中山大学计算机学院

人工智能

本科生实验报告

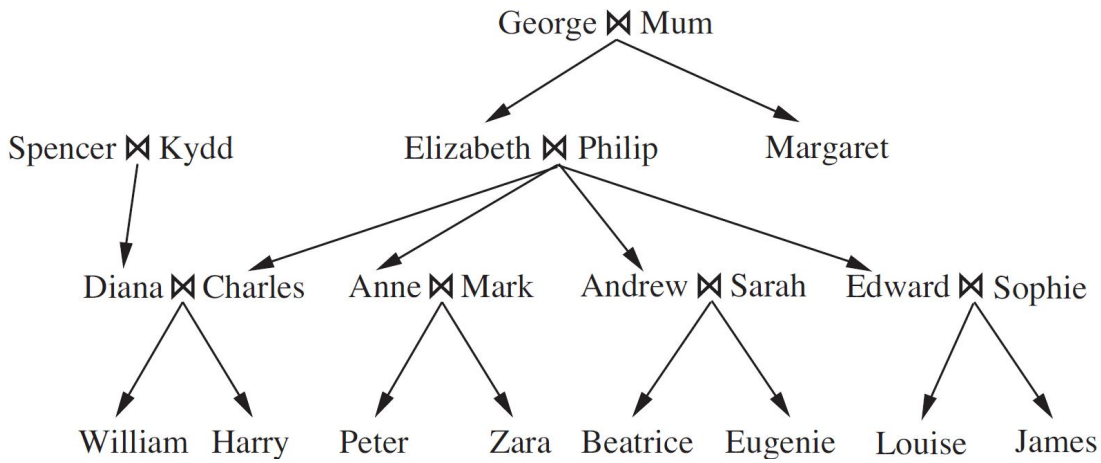
（2022 学年春季学期）

课程名称：Artificial Intelligence

教学班级	信息与计算科学班	专业（方向）	信息与计算科学
学号	21311359	姓名	何凯迪

一、 实验题目

1. 家族树下图所示，要求列出家族树中显示的基本事实。
2. 写出对应的谓词来预测以下的关系：Grandchild, Greatgrandparent, Ancestor,Brother,Sister,Daughter,Son,FirstCousin, Brother-In-Law, Sister-In-Law, Aunt, Uncle. 从中任选三个来预测。
3. 定义 mth cousin n times removed 关系。
4. 提问 (1)Elizabeth’s grandchildren,(2)Diana’s brothers-in-law, (3)Zara’s greatgrandparents, (4) Eugenie’s ancestors.



二、 实验内容

1. 算法原理

Prolog 语言是一种以一阶谓词为基础的逻辑性语言，以一阶谓词逻辑的 Horn 子句集为语法，以 Robinson 的消解原理为工具，加上深度优先的控制策略而形成的人工智能通用程序设计语言。

Prolog 语言的特点如下：

- **是一种描述性语言。**只需要告诉系统“做什么”，不要告诉系统“如何做”。

• **数据与程序的统一表达。**提供一种统一的符号结构 “项” ，数据与程序都是由项
组成。
• **自动实现模式匹配与回溯。**这是人工智能中最常用的两项操作，Prolog 自动实现这
些操作。
• **程序易于编写与阅读。**它是面向人的自然语言。
• **语句句型少，语法简明。**只有三种句型。

2. 伪代码

事实：

```
couple(x,y).
male(x).
female(x).
father(x,y).
child_fact(x,y).
```

定义关系：

```
mother(X,Z) :- father(Y,Z),(couple(X,Y)|couple(Y,X)).
child(X,Y) :- parent(Y,X).
parent(X,Y) :- father(X,Y)|mother(X,Y).
……
```

含各类需要或实验要求的关系

输出方式：

```
list_son :- son(X,Y), write(X),write(' is '),write(Y),write("'s son"),nl,fail.
find_son :- read(X),son(Y,X),write(Y),nl,fail.
list_Elizabeth_grandchildren :- grandchild(X,'Elizabeth'),write(X),write(" is Elizabeth's grandchild"),nl,fail.
……
```

含各要求所需的输出。

3. 关键代码展示（带注释）

事实在 1-71 行列出，对应谓词预测关系在 79-92 行列出（list_relation 实现列出所有符合该 relation 关系人物对，find_relation 实现手动输入人名后查找所有符合该 relation 关系的人物），定义 mth cousin n times removed 关系在 115-120 行列出，第四小问的具体人物关系提问在 100-113 列出。

```
%facts of couples
couple('George','Mum').
couple('Spencer','Kydd').
couple('Elizabeth','Philip').
couple('Diana','Charles').
couple('Anne','Mark').
couple('Andrew','Sarah').
couple('Edward','Sophie').
```

```
10. %facts of gender
11. male('George').
12. male('Philip').
13. male('Spencer').
14. male('Charles').
15. male('Mark').
16. male('Andrew').
17. male('Edward').
18. male('William').
19. male('Harry').
20. male('Peter').
21. male('James').
22.
23. female('Mum').
24. female('Kydd').
25. female('Elizabeth').
26. female('Margaret').
27. female('Diana').
28. female('Anne').
29. female('Sarah').
30. female('Sophie').
31. female('Zara').
32. female('Beatrice').
33. female('Eugenie').
34. female('Louise').
35.
36. %father(X,Y)--X is Y's father
37. father('George','Elizabeth').
38. father('George','Margaret').
39. father('Spencer','Diana').
40. father('Philip','Charles').
41. father('Philip','Anne').
42. father('Philip','Andrew').
43. father('Philip','Edward').
44. father('Charles','William').
45. father('Charles','Harry').
46. father('Mark','Peter').
47. father('Mark','Zara').
48. father('Andrew','Beatrice').
49. father('Andrew','Eugenie').
50. father('Edward','Louise').
51. father('Edward','James').
52.
```

```
%mother(X,Y)--X is Y's mother
mother(X,Z) :- father(Y,Z),(couple(X,Y)|couple(Y,X)).

%facts of children;child(X,Y)--X is Y's child
child_fact('Elizabeth',couple('George','Mum')).
child_fact('Margaret',couple('George','Mum')).
child_fact('Diana',couple('Spencer','Kydd')).
child_fact('Charles',couple('Elizabeth','Philip')).
child_fact('Anne',couple('Elizabeth','Philip')).
child_fact('Andrew',couple('Elizabeth','Philip')).
child_fact('Edward',couple('Elizabeth','Philip')).
child_fact('William',couple('Diana','Charles')).
child_fact('Harry',couple('Diana','Charles')).
child_fact('Peter',couple('Anne','Mark')).
child_fact('Zara',couple('Anne','Mark')).
child_fact('Beatrice',couple('Andrew','Sarah')).
child_fact('Eugenie',couple('Andrew','Sarah')).
child_fact('Louise',couple('Edward','Sophie')).
child_fact('James',couple('Edward','Sophie')).

child(X,Y) :- parent(Y,X).

%list_mother :- mother(X,Y), write(X),write(' is '),write(Y),write("'s mother"),nl,fail.
%find_mother :- read(X),mother(Y,X),write(Y),fail.
%judge_mother :- read(X),read(Y),mother(X,Y),fail.

%Son
son(X,Y) :- (father(Y,X)|mother(Y,X)),male(X).
list_son :- son(X,Y), write(X),write(' is '),write(Y),write("'s son"),nl,fail.
find_son :- read(X),son(Y,X),write(Y),nl,fail.

%Daughter
daughter(X,Y) :- (father(Y,X)|mother(Y,X)),female(X).
list_daughter :- daughter(X,Y), write(X),write(' is '),write(Y),write("'s daughter"),nl,fail.
find_daughter :- read(X),daughter(Y,X),write(Y),nl,fail.

%Sister
sister(X,Y) :- father(Z,X),father(Z,Y),X\=Y,female(X).
list_sister :- sister(X,Y), write(X),write(' is '),write(Y),write("'s sister"),nl,fail.
find_sister :- read(X),sister(Y,X),write(Y),nl,fail.

%Brother
brother(X,Y) :- father(Z,X),father(Z,Y),X\=Y,male(X).
```

```
96.
97.  %parent
98.  parent(X,Y) :- father(X,Y)|mother(X,Y).
99.
100. %Elizabeth's grandchildren;Diana's brother-in-law;Zara's greatgrandparents;Eugenie's anc
     estors
101. %grandchild(X,Y) :- (father(Y,Z),father(Z,X))|(father(Y,Z),mother(Z,X))|(mother(Y,Z),father(Z,
     X))|(mother(Y,Z),mother(Z,X)).
102.
103. grandchild(X,Y) :- parent(Y,Z),parent(Z,X).
104. list_Elizabeth_grandchildren :- grandchild(X,'Elizabeth'),write(X),write(" is Elizabeth's gran
     dchild"),nl,fail.
105.
106. brother_in_law(X,Y) :- brother(X,Z),(couple(Z,Y)|couple(Y,Z)) | (couple(X,Z)|couple(Z,X)),sist
     er(Z,Y) | (couple(Z,Y)|couple(Y,Z)),sister(W,Z),(couple(X,W)|couple(W,X)).
107. list_Diana_brother_in_law :- brother_in_law(X,'Diana'),write(X),write(" is Diana's brother-in-l
     aw"),nl,fail.
108.
109. greatgrandparent(X,Y) :- parent(X,Z),parent(Z,W),parent(W,Y).
110. list_Zara_grandparents :- greatgrandparent(X,'Zara'),write(X),write(" is Zara's greatgrandp
     arent"),nl,fail.
111.
112. ancestor(X,Y) :- parent(X,Y)|grandchild(Y,X)|greatgrandparent(X,Y).
113. list_Eugenie_ancestor :- ancestor(X,'Eugenie'),write(X),write(" is Eugenie's ancestor"),nl,fai
     l.
114.
115. %mth cousin n times removed
116. distance(X, Y, N) :- (X = Y, N = 0);(ancestor(X, Y), child(Y, Z), distance(X, Z, N1), N is N1 + 1);
     (\+ancestor(X, Y), ancestor(Z, X), ancestor(Z, Y), distance(Z, Y, N1), distance(Z, X, N2), N is (N
     1 - N2)).
117.
118. mthCousin(X, Y, M) :- (ancestor(Z, X), ancestor(Z, Y), distance(Z, X, N1), distance(Z, Y, N2),
     N1 = N2, M is (N1 - 1)).
119.
120. mthCousinNremoved(X, Y, M, N) :- ((mthCousin(X, Y, M1), M is M1, N is 0)|(ancestor(Z, X), m
     thCousin(Z, Y, M), distance(Z, X, N2), N is N2)|(ancestor(Z, Y), mthCousin(Z, X, M), distance(Z,
     Y, N3), N is N3)).
```

4. 创新点&优化（如果有）

1、对于任务 2.1 要求写出对应谓词来预测三个关系，程序中既有 list_relation 实现了列出所有符合该 relation 关系人物对，又有 find_relation 实现了手动输入人名后查找所有符合该 relation 关系的人物。

2、除了设置了 distance(X,Y,N)，还多设置了一个 mthCousin(X, Y, M)，可以计算 X 与 Y 是几代 Cousin，便于后面对 mthCousinNremoved(X, Y, M, N)。

3、

对于mother(X,Z),原本打算当成事实列出，最后发现过于繁琐。故在已列出father的基础上，定义了mother(X,Z):-father(Y,Z),(couple(X,Y)|couple(Y,X))

4、

对于grandchild(X,Y),最先是通过father(Y,Z),father(Z,X)|father(Y,Z),mother(Z,X)|mother(Y,Z),father(Z,X)|mother(Y,Z),mother(Z,X)来定义的。后来突然想到新定义一个中介parent(X,Y):-father(X,Y)|mother(X,Y)，则对于grandchild(X,Y)可以新定义为parent(Y,Z),parent(Z,X)即可。

三、 实验结果及分析

实验结果展示示例（可图可表可文字，尽量可视化）

1、要求列出家族树中显示的基本事实

关键代码中已给出基本事实（couple、gender、child 等）。

2、写出对应的谓词来预测以下的关系：Grandchild, Greatgrandparent, Ancestor, Brother, Sister, Daughter, Son, FirstCousin, BrotherInLaw, SisterInLaw, Aunt, Uncle. 从中任选三个来预测。

（实际代码中已实现大部分关系预测，此处仅展示其中三个）

```
?- list_son.
Charles is Philip's son
Andrew is Philip's son
Edward is Philip's son
William is Charles's son
Harry is Charles's son
Peter is Mark's son
James is Edward's son
Charles is Elizabeth's son
Andrew is Elizabeth's son
Edward is Elizabeth's son
William is Diana's son
Harry is Diana's son
Peter is Anne's son
James is Sophie's son
false.

?- list_daughter.
Elizabeth is George's daughter
Margaret is George's daughter
Diana is Spencer's daughter
Anne is Philip's daughter
Zara is Mark's daughter
Beatrice is Andrew's daughter
Eugenie is Andrew's daughter
Louise is Edward's daughter
Elizabeth is Mum's daughter
Margaret is Mum's daughter
Diana is Kydd's daughter
Anne is Elizabeth's daughter
Zara is Anne's daughter
Beatrice is Sarah's daughter
Eugenie is Sarah's daughter
Louise is Sophie's daughter
false.

?- list_sister.
Elizabeth is Margaret's sister
Margaret is Elizabeth's sister
Anne is Charles's sister
Anne is Andrew's sister
Anne is Edward's sister
Zara is Peter's sister
Beatrice is Eugenie's sister
Eugenie is Beatrice's sister
Louise is James's sister
false.
```

```
?- find_son.
|: 'Elizabeth'.
Charles
Andrew
Edward
false.

?- find_daughter.
|: 'Philip'.
Anne
false.

?- find_sister.
|: 'Beatrice'.
Eugenie
false.
```

3、定义 mth cousin n times removed 关系

```
?- mthCousinNremoved('Peter','James',M,N).
M = 1,
N = 0 .

?- mthCousinNremoved('Edward','Zara',M,N).
M = 0,
N = 1 .

?- mthCousinNremoved('Harry','Margaret',M,N).
M = 0,
N = 2 .
```

4、提问(1)Elizabeth's grandchildren,(2)Diana's brothers-in-law, (3)Zara's greatgrandparents, (4) Eugenie's ancestors.

```
kaddy@kaddy-VirtualBox:~/AI-exp2$ prolog
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

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

?- ['exp2.pl'].
true.

?- list_Elizabeth_grandchildren.
William is Elizabeth's grandchild
Harry is Elizabeth's grandchild
Peter is Elizabeth's grandchild
Zara is Elizabeth's grandchild
Beatrice is Elizabeth's grandchild
Eugenie is Elizabeth's grandchild
Louise is Elizabeth's grandchild
James is Elizabeth's grandchild
false.

?- list_Diana_brother_in_law.
Andrew is Diana's brother-in-law
Edward is Diana's brother-in-law
Mark is Diana's brother-in-law
false.

?- list_Zara_greatgrandparents.
George is Zara's greatgrandparent
Mum is Zara's greatgrandparent
false.

?- list_Eugenie_ancestor.
Andrew is Eugenie's ancestor
Sarah is Eugenie's ancestor
Philip is Eugenie's ancestor
Elizabeth is Eugenie's ancestor
George is Eugenie's ancestor
Mum is Eugenie's ancestor
false.
```

四、 参考资料

https://blog.csdn.net/qq_38237214/article/details/73613903?ops_request_misc=%257B%2522request%255Fid%2522%253A%252216797405881680018219127 9%2522%252C%2522scm%2522%253A%252220140713.130102334..%2522%257D&request_id=167974058816800182191279&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduend~default-4-73613903-null-null.142^v76^insert_down1,201^v4^add_ask,239^v2^insert_chatgpt&utm_term=prolog&spm=1018.2226.3001.4187

https://blog.csdn.net/qq_33017507/article/details/109502720?ops_request_misc=%257B%2522request%255Fid%2522%253A%252216797405881680018219127 9%2522%252C%2522scm%2522%253A%252220140713.130102334..%2522%257D&request_id=167974058816800182191279&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~top_positive~default-1-109502720-null-null.142^v76^insert_down1,201^v4^add_ask,239^v2^insert_chatgpt&utm_term=prolog&spm=1018.2226.3001.4187

https://blog.csdn.net/m0_37816922/article/details/100912825