河南工业大学

《智能系统》实验报告

专业班级：物联网 1603　　学号：201616070320　姓名：郭治洪

实验一 Prolog 语言编程（4 学时）

实验时间：2019.4

【实验目的】

（1）掌握 SWI-Prolog 安装与卸载；

（2）掌握 SWI-Prolog 的基本特性；

（3）理解项目文件结构以及之间的关系；

（4）掌握调试项目的步骤。

【实验环境】

SWI-Prolog

【实验内容】

（1）编写一个描述亲属关系的 Prolog 程序，然后再给予出一些事实数据，建立一个小型演绎数据库；

（2）掌握调试 Prolog 程序的调试步骤。

【详细分析】

1. 去 SWI-Prolog 官网下载并且安装 SWI-Prolog

地址 http://www.swi-prolog.org/download/stable

并且配置环境变量（略）。

2. 学习 Prolog 简易教程

Prolog 语言参考：https://draveness.me/prolog-ji-chu-1

Learn Prolog 翻译版：https://www.cnblogs.com/seaman-h-zhang/p/4592913.html

Prolog 语言入门教程：http://www.ruanyifeng.com/blog/2019/01/prolog.html

3. 尝试查阅百年孤独家谱图，并且用 prolog 实现

源码见下页。

```
% One Hundred Years of Solitude 's family tree in Prolog
% Wiki picture:
https://en.wikipedia.org/wiki/One_Hundred_Years_of_Solitude#/media/File
:One_Hundred_Years_Of_Solitude_Buendia%27s_Family_Tree.svg
% Author : GuoZhiHong
% Class : IoT1603
% StudentID : 201616070320


% See
here:https://codereview.stackexchange.com/questions/143116/family-tree-
in-prolog

% predicate(X,Y) means "X is Y 's XXX ."
% eg. father(X,Y) means "X is Y 's father."

:- encoding(utf8).
:- discontiguous man/1, woman/1,
marry/2,amorousAffairsy/2,parent/2,father/2,mother/2,child/2,illegitima
teChild/2.

% Generation I
man('José Arcadio Buendía').
woman('Úrsula Iguarán').

marry('José Arcadio Buendía','Úrsula Iguarán').

% Generation II
man('José Arcadio').
man('Colonel Aureliano Buendía').
woman('Amaranta').
woman('Remedios Moscote').
woman('Rebeca').
woman('Pilar Ternera').

parent('José Arcadio Buendía','José Arcadio').
parent('José Arcadio Buendía','Colonel Aureliano Buendía').
parent('José Arcadio Buendía','Amaranta').
parent('Úrsula Iguarán','José Arcadio').
parent('Úrsula Iguarán','Colonel Aureliano Buendía').
parent('Úrsula Iguarán','Amaranta').
marry('Colonel Aureliano Buendía','Remedios Moscote').
marry('Rebeca','José Arcadio').
amorousAffairsy('Pilar Ternera','José Arcadio').
amorousAffairsy('Pilar Ternera','Colonel Aureliano Buendía').
```

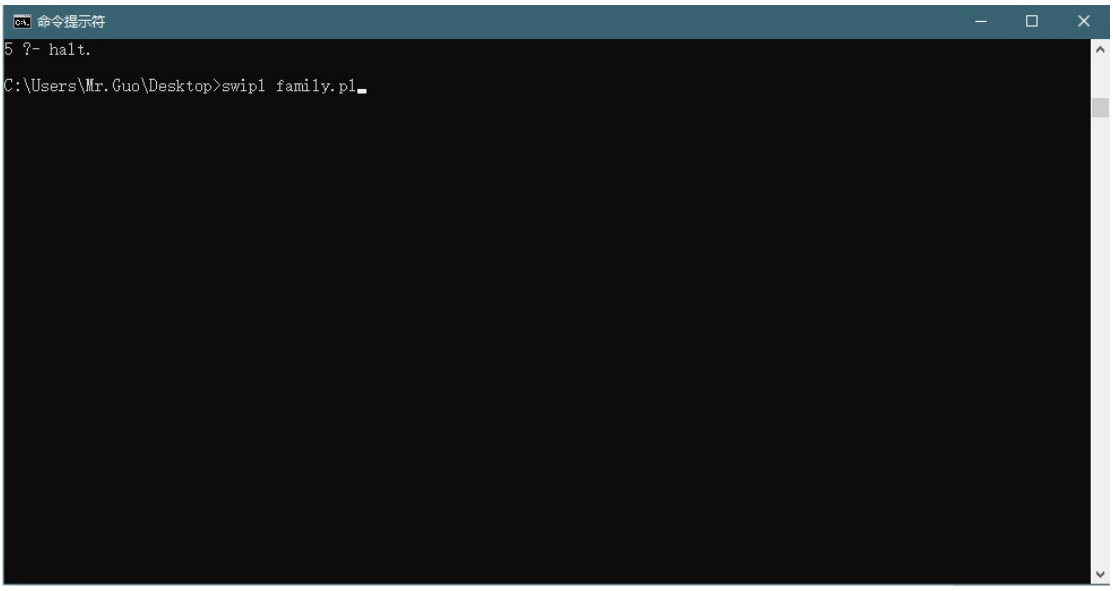
```
% Generation III
woman('Aureliano José').
woman('Arcadio').
man('Santa Sofía de la Piedad').

marry('Santa Sofía de la Piedad','Arcadio').
illegitimateChild('17 sons by unknown women','Colonel Aureliano
Buendía').
parent('Pilar Ternera','Arcadio').
parent('José Arcadio Buendía','Arcadio').
% Generation IV
woman('Remedios the Beauty').
man('José Arcadio II').
man('Aureliano II').
woman('Petra Cotes').
woman('Fernanda del Carpio').

marry('Aureliano II','Fernanda del Carpio').
amorousAffairsy('Petra Cotes','Aureliano II').
parent('Santa Sofía de la Piedad','Remedios the Beauty').
parent('Santa Sofía de la Piedad','José Arcadio II').
parent('Santa Sofía de la Piedad','Aureliano II').
parent('Arcadio','Remedios the Beauty').
parent('Arcadio','José Arcadio II').
parent('Arcadio','Aureliano II').
% Generation V
man('Gastón').
man('José Arcadio').
man('Mauricio Babilonia').
woman('Amaranta Úrsula').
woman('Renata Remedios').

marry('Amaranta Úrsula','Gastón').
amorousAffairsy('Renata Remedios','Mauricio Babilonia').
parent('Aureliano II','Amaranta Úrsula').
parent('Aureliano II','José Arcadio').
parent('Aureliano II','Renata Remedios').
parent('Fernanda del Carpio','Amaranta Úrsula').
parent('Fernanda del Carpio','José Arcadio').
parent('Fernanda del Carpio','Renata Remedios').
% Generation VI
man('Aureliano Babilonia').
parent('Mauricio Babilonia','Aureliano Babilonia').
```

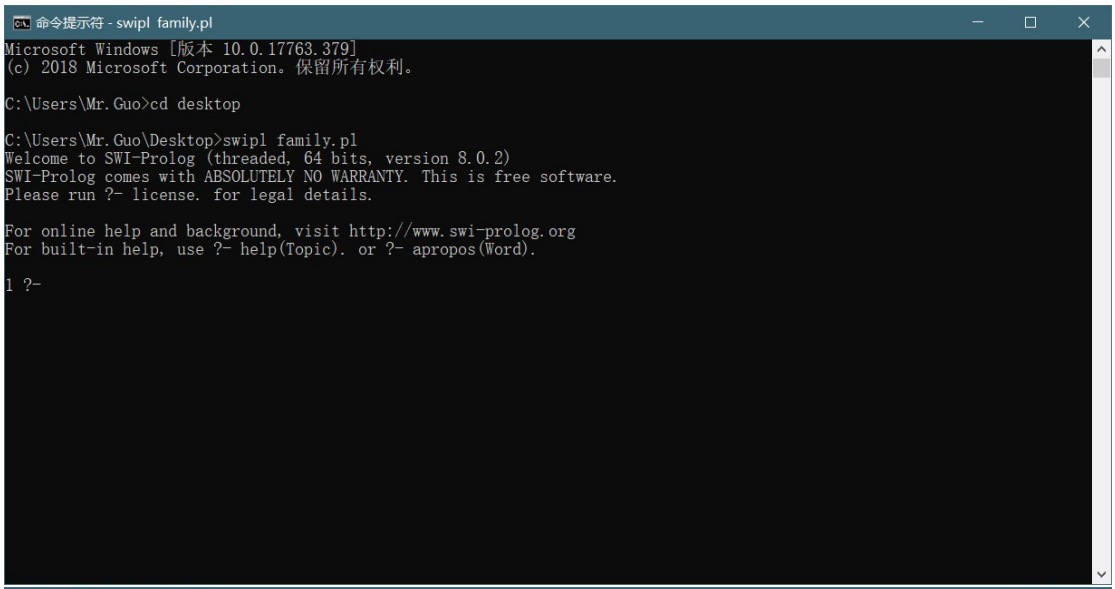
```
parent('Renata Remedios','Aureliano Babilonia').

amorousAffairsy('Amaranta Úrsula','Aureliano Babilonia').
% Generation VII
man('Aureliano').
parent('Aureliano Babilonia','Aureliano').
parent('Amaranta Úrsula','Aureliano').

% --------  Start defining the relationships --------

% The couple relation is in fact symmetric.
couple(X,Y) :- marry(X,Y), X\=Y.
couple(X,Y) :- marry(Y,X), X\=Y.

% The amorous affairsy relation is also in fact symmetric.
illegitimateLover(X,Y) :- amorousAffairsy(X,Y); amorousAffairsy(Y,X),
X\=Y.

father(X,Y) :- parent(X,Y), man(X), X\=Y.
mother(X,Y) :- parent(X,Y), woman(X), X\=Y.

% X or Y is Z's parents. X may be Z's father or mother,Y may be Z's
mother or father.
parents(X,Y,Z) :- (father(X,Z), mother(Y,Z)); (father(Y,Z),
mother(X,Z)), X\=Y, X\=Z, Z\=Y.
parents(X,Y,Z) :- (father(X,Z), mother(Y,Z)); (father(Y,Z),
mother(X,Z)), X\=Y, X\=Z, Z\=Y.

% X is Y's child. => Y is X's father or mother.
child(X,Y) :- (father(Y,X); mother(Y,X)), X\=Y.
illegitimateChild(X,Y) :- illegitimateChild(X,Y), X\=Y.
son(X,Y) :- child(X,Y), man(X), X\=Y.
daughter(X,Y) :- child(X,Y), woman(X), X\=Y.

% Grandma or grandpa is someone's mother or father 's mother or father.
grandma(X,Y) :- (mother(X,M), mother(M,Y)); (mother(X,F),father(F,Y)),
X\=Y.
grandpa(X,Y) :- (mother(X,M), father(M,Y)); (father(X,F), father(F,Y)),
X\=Y.
grandparents(X,Y,Z) :- (grandma(X,Z), grandpa(Y,Z)); (grandma(Y,Z),
grandpa(X,Z)), X\=Y, X\=Z, Z\=Y.

% Siblings need having same father and mother.
```

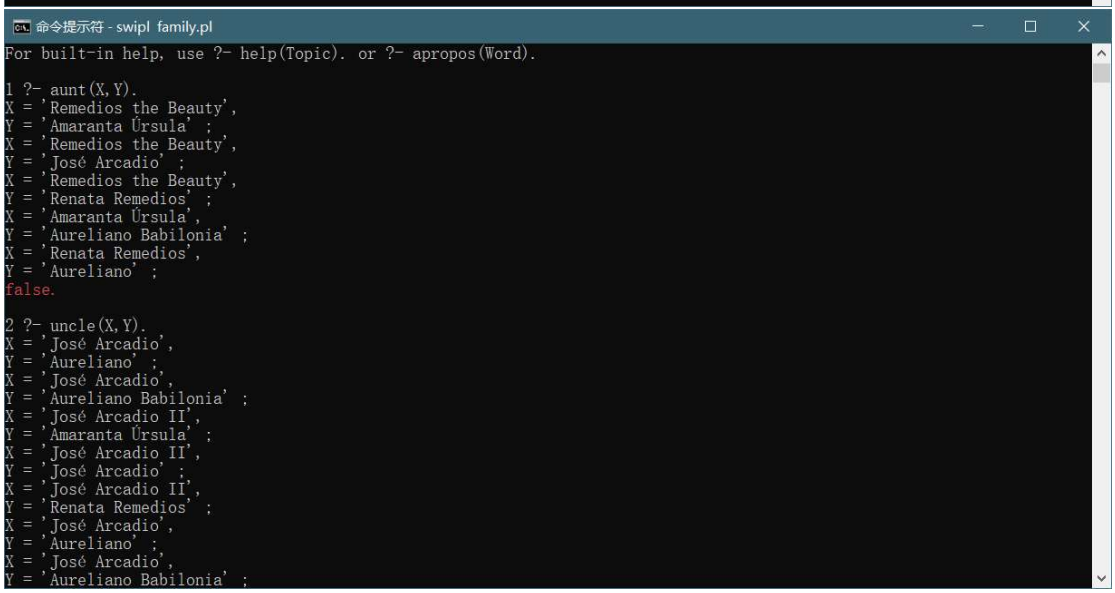
```
siblings(A,B) :- father(F,A), father(F,B), mother(M,A), mother(M,B),
A\=B.

% Uncle or aunt is someone's mother or father 's siblings.
uncle(U,N) :- man(U), ((siblings(U,F), father(F,N)); (siblings(U,M),
mother(M,N))), U\=N.
aunt(A,N) :- woman(A), ((siblings(A,F), father(F,N)); (siblings(A,F),
mother(F,N))), A\=N.

% Y's sister or brother is woman or man.
sister(X,Y) :- siblings(X,Y), woman(X), X\=Y.
brother(X,Y) :- siblings(X,Y), man(X), X\=Y.

% Ancestor may be someone's parent or they are someone's parent's
ancestor.
ancestor(X,Y):- parent(X,Y), X\=Y.
ancestor(X,Y):- parent(X,Z), ancestor(Z,Y).

% --------  End defining the relationships --------
```

执行方法在命令提示符打开并且定位到源码目录。执行 `swipl family.pl` 。然后按照 Prolog 语法进行查询。

截图如下：




命令提示符 - swipl family.pl
Microsoft Windows [版本 10.0.17763.379]
(c) 2018 Microsoft Corporation。保留所有权利。
C:\Users\Mr.Guo>cd desktop
C:\Users\Mr.Guo\Desktop>swipl family.pl
Welcome to SWI-Prolog (threaded, 64 bits, version 8.0.2)
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).
1 ?-


命令提示符 - swipl family.pl
For built-in help, use ?- help(Topic). or ?- apropos(Word).
1 ?- aunt(X,Y).
X = 'Remedios the Beauty',
Y = 'Amaranta Úrsula' ;
X = 'Remedios the Beauty',
Y = 'José Arcadio' ;
X = 'Remedios the Beauty',
Y = 'Renata Remedios' ;
X = 'Amaranta Úrsula',
Y = 'Aureliano Babilonia' ;
X = 'Renata Remedios',
Y = 'Aureliano' ;
false.
2 ?- uncle(X,Y).
X = 'José Arcadio',
Y = 'Aureliano' ;
X = 'José Arcadio',
Y = 'Aureliano Babilonia' ;
X = 'José Arcadio II',
Y = 'Amaranta Úrsula' ;
X = 'José Arcadio II',
Y = 'José Arcadio' ;
X = 'José Arcadio II',
Y = 'Renata Remedios' ;
X = 'José Arcadio',
Y = 'Aureliano' ;
X = 'José Arcadio',
Y = 'Aureliano Babilonia' ;