UNIWERSYTET

W SIEDLCACH

FACULTY OF EXACT AND NATURAL SCIENCES

The Faculty of Informatics

Laboratory exercises

Selected Programming Paradigms

Family tree

Report lab 7 and 8

Developed by

Abdullakim

2nd year, Erasmus student computer science and math studies

Course Supervisor

MSc. Maciej Nazarczuk

Siedlce, 2024

Contents

Formulation of the task

The aim of the project is to create an anonymized "Family Tree" for my family, spanning 4 generations.

The scope of the tree should include great-grandparents on both sides, grandparents, parents, siblings of parents and children of all branches of the family tree.

As part of this project, my task will be to model this tree in the language Prolog programming.

The designed Knowledge Base is intended to identify family members, their gender and gender through the Fact Base direct descendants, the Rules Database contains rules determining the relationship between different family members.

Family tree

The Family Tree was created based on my family.

The spanned family contains 25 members.

A Family Tree containing 25 members (shown in Fig. 1).

Knowledge Base Code Based on the Family Tree, which is a specific Tree of Goals, it was developed by Knowledge Base, which consists of a Facts Base and a Rules Base.

The Fact base is as follows:

```
% Facts

:- discontiguous male/2.

:- discontiguous female/2.


male(me, Abdullakim).

female(sibling1, Dori).

male(sibling2, Uluk).

male(sibling3, Bekus).

male(father, Zamirbek).

female(mother, Manzuura).

male(grandfather, Maken).

female(grandmother, Inaiat).

female(aunt1, Kuka).

female(aunt2, Kuka).

male(mother_father, Kaisa).

female(mother_mother, Elina).

male(father_father, Akil).

female(father_mother, Aisulu).

male(grandfather_father, Erkin).

female(grandfather_mother, Kubat).

male(grandmother_father, Alimbekov).

female(grandmother_mother, Aisha).

male(mother_father_father, Azim).

female(mother_father_mother, Aigerim).
```

```
male(mother_mother_father, Ubaydulla).

female(mother_mother_mother, Kenzhe).


% Rules

:- discontiguous parent/2.

:- discontiguous sibling/2.

:- discontiguous aunt_and_uncle/2.

:- discontiguous aunt_and_aunt/2.

:- discontiguous grandfather/2.

:- discontiguous grandmother/2.

:- discontiguous great_grandparent/2.

:- discontiguous great_grandfather/2.

:- discontiguous great_grandmother/2.

:- discontiguous father_father/2.

:- discontiguous father_mother/2.

:- discontiguous mother_father/2.

:- discontiguous mother_mother/2.

:- discontiguous mother_sibling/2.

:- discontiguous father_sibling/2.

:- discontiguous grandfather_grandfather/2.

:- discontiguous grandfather_grandmother/2.

:- discontiguous grandfather_father/2.

:- discontiguous grandfather_mother/2.

:- discontiguous grandmother_father/2.

:- discontiguous grandmother_mother/2.

:- discontiguous mother_father_grandfather/2.

:- discontiguous mother_father_grandmother/2.

:- discontiguous mother_mother_grandfather/2.
```

```
:- discontiguous mother_mother_grandmother/2.

 %  Rules

 parent(me, father).

 parent(me, mother).


 parent(sibling1, father).

 parent(sibling1, mother).


 parent(sibling2, father).

 parent(sibling2, mother).


 parent(father, grandfather).

 parent(father, grandmother).


 parent(mother, aunt1).

 parent(mother, aunt2).


 parent(mother, mother_father).

 parent(mother, mother_mother).


 parent(father, father_father).

 parent(father, father_mother).


 % Rules for sibling relationships

 sibling(me, sibling1).

 sibling(me, sibling2).
```

```
sibling(sibling1, me).
sibling(sibling1, sibling2).

sibling(sibling2, me).
sibling(sibling2, sibling1).

% Rules for aunt relationships
aunt(X, Y) :- parent(X, P), sibling(P, Y).

% Rules for grandfather/grandmother relationships
grandfather(X, Y) :- parent(X, P), parent(P, Y), male(X).
grandmother(X, Y) :- parent(X, P), parent(P, Y), female(X).

% Rules for mother's siblings
mother_sibling(X, Y) :- parent(X, P), mother(P, M), sibling(M, Y).

% Rules for father's siblings
father_sibling(X, Y) :- parent(X, P), father(P, F), sibling(F, Y).

% Rules for grandfather's father/mother
grandfather_grandfather(X, Y) :- grandfather(X, G), father(G, Y).
grandfather_grandmother(X, Y) :- grandfather(X, G), mother(G, Y).

% Rules for grandmother's father/mother
grandmother_father(X, Y) :- grandmother(X, G), father(G, Y).
grandmother_mother(X, Y) :- grandmother(X, G), mother(G, Y).
```

% Rules for mother's father's father/mother

mother_father_grandfather(X, Y) :- mother_father(X, MF), father(MF, F), father(F, Y).

mother_father_grandmother(X, Y) :- mother_father(X, MF), father(MF, F), mother(F, Y).

% Rules for mother's mother's father/mother

mother_mother_grandfather(X, Y) :- mother_mother(X, MM), father(MM, F), father(F, Y).

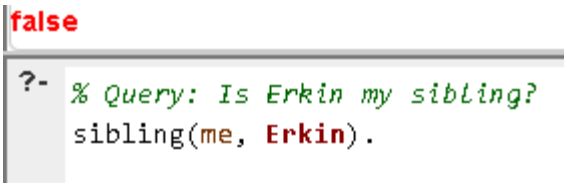
mother_mother_grandmother(X, Y) :- mother_mother(X, MM), father(MM, F), mother(F, Y).

Testing Phase sample:

In this query we are going to test if Erkin is my sibling or not. The command to do this as follows:

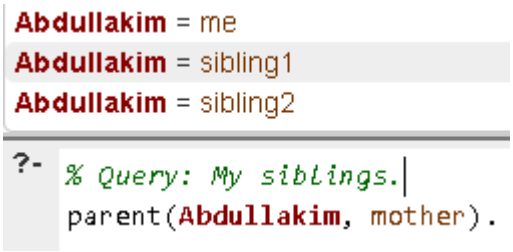
sibling(me, Erkin).

The outcome is false he is not my sibling.



In this code below, it checks my siblings and that is equal to me. The code to test this relation is:

parent(Abdullakim, mother).



Below I am testing the code to know my mother's siblings. With the following code:

parent(mother, Manzuura).

```
Manzuura = aunt1
Manzuura = aunt2
```

```
?- % Query: My mother's siblings?
   parent(mother, Manzuura).
```

The Facts Database is composed of:

1. facts about parenting.

2. facts about gender.

The Rules Database is composed of:

1. rules about family relationships.

2. rules about traits.

Predicates have been written in the convention:

Is X the father of Y?

father(X,Y):-parent(X,Y), male(X).

Is X the grandson of Y?

grandson(X,Y):-grandparents(Y,X), man(X).

Example overview:

aunt(X,Y):-woman(X),parent(P,Y),(sibling(X,P),sibling(P,X)).

For:

X = Kuka

Y =Maken

If

1. woman (Kuka) if Kuka is a woman

2. , and

3. parent(P,Maken) Maken has parent Y

4. , and

5. (siblings(Maken,Y) Maken is a sibling of P

6. ; or

7. siblings(P,Maken). P is a sibling of P

the sibling relationship is shown once, e.g.

siblings (Kuka, Cholpon).

so it checks two cases.

Literature

1. Tchórzewski J., Laboratory exercises in selected programming paradigms

declarative, Siedlce, September 2021