Faculty of Computers, Informatics and Microelectronics

Technical University of Moldova

PLIA

Laboratory work # 1

Introduction of Programming Language Prolog

Author:
Petru Negrei

Supervisor:
L. Luchianov

Octomber 2014

1 Introduction

1.1 Topic

Introduction of Programming Language Prolog

1.2 Objective

Main principles of working with programming language Prolog.

1.3 Generic requirements

1.3.1 Task

- Read the instructions and necessary theory.
- Analyze given examples and find alternative solutions (questions).
- Elaborate your family tree and a knowlegde base in Prolog which will describe existing relations in your family and that will provide questions to them. Your family tree contain at least three generations. In order to check relations will be used at least six scopes.

1.3.2 Report

Report will containt a short description of work done, and will present necessary information about tools, algorithms used or studied.

2 Implementation

- ○ *data* (or "raw" data) is computer data without any concern for its meaning.
- ○ *Information* is data with a meaning, data that means something.
- ○ A *data base* is a collection of information that allows one to extract meaningful data in many different ways.
- ○ A *knowledge base* has even more: in addition to the information (called *facts* in prolog), and the ability to extract information, there is also the ability to deduce new facts using prolog *rules*.

2.1 Facts

Here are some facts for a simple prolog knowledge base. It gives facts about a small family tree, with part of three generations. Below you can see a graphical representation of family tree.

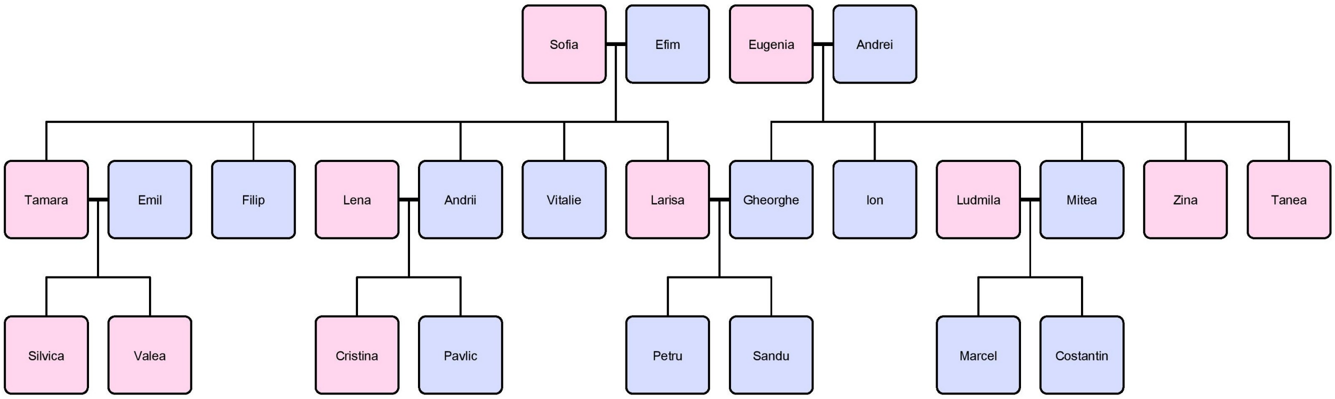

Fig. 1 Genealogical tree

Below is represented the full list of facts:

```
male(petru).
male(sandu).
male(gheorghe).
male(ion).
male(mitea).
male(andrei).
male(vitalie).
male(efim).
male(andrii).
male(filip).
male(emil).
male(pavlic).
male(marcel).
male(constantin).
female(larisa).
female(zina).
female(tanea).
female(eugenia).
female(sofia).
female(tamara).
female(silvica).
female(valea).
female(cristina).
female(lena).
female(ludmila).
parent(gheorghe, petru).
parent(gheorghe, sandu).
parent(larisa, petru).
parent(larisa, sandu).
parent(andrei, gheorghe).
parent(andrei, ion).
parent(andrei, mitea).
parent(andrei, zina).
parent(andrei, tanea).
parent(eugenia, gheorghe).
parent(eugenia, ion).
parent(eugenia, mitea).
parent(eugenia, zina).
parent(eugenia, tanea).
parent(efim, larisa).
parent(efim, filip).
parent(efim, andrii).
parent(efim, tamara).
parent(efim, vitalie).
parent(sofia, larisa).
parent(sofia, filip).
parent(sofia, andrii).
parent(sofia, tamara).
parent(sofia, vitalie).
parent(tamara, silvica).
parent(tamara, valea).
parent(emil, silvica).
parent(emil, valea).
parent(lena, cristina).
parent(lena, pavlic).
parent(adrii, cristina).
parent(adrii, pavlic).
parent(ludmila, marcel).
parent(ludmila, constantin).
parent(mitea, marcel).
parent(mitea, constantin).
married(andrei, eugenia).
married(eugenia, andrei).
married(efim, sofia).
married(sofia, efim).
married(gheorghe, larisa).
married(larisa, gheorghe).
married(emil, tamara).
married(tamara, emil).
married(lena, andrii).
married(andrii, lena).
married(ludmila, mitea).
married(mitea, ludmila).
```

2.2 Rules

```
husband(X, Y):- male(X), married(X, Y).
wife(X, Y) :- female(X), married(X, Y).
father(X, Y):- male(X), parent(X, Y).
mother(X, Y):- female(X), parent(X, Y).

sibling(X, Y) :- father(Z, X), father(Z, Y),
                     mother(W, X), mother(W, Y), not(X = Y).
brother(X, Y) :- male(X), sibling(X, Y).
sister(X, Y):- female(X), sibling(X, Y).
```

```
grandparent(X, Z):- parent(X, Y), parent(Y, Z).
grandfather(X, Z) :- male(X), grandparent(X, Z).
grandmother(X, Z):- female(X), grandparent(X, Z).
grandchild(X, Z) :- grandparent(Z, X).
grandson(X, Z) :- male(X), grandchild(X, Z).
granddaughter(X, Z) :- female(X), grandchild(X, Z).

child(Y, X) :- parent(X, Y).
son(Y, X) :- male(Y), child(Y, X).
daughter(Y, X) :- female(Y), child(Y, X).

auntoruncle(X, W) :- sibling(X, Y), parent(Y, W).
uncle(X, W) :- male(X), auntoruncle(X, W).
aunt(X, W) :- female(X), auntoruncle(X, W).

cousin(X, Y) :- parent(Z, X), auntoruncle(Z, Y).
nieceornephew(X, Y) :- parent(Z, X), sibling(Z, Y).
nephew(X, Y) :- male(X), nieceornephew(X, Y).
niece(X, Y) :- female(X), nieceornephew(X, Y).
```

3 Conclusion

In the following laboratory work I learn more about the fundamental part of the programming language Prolog. I build a knowledge base of my family tree using Prolog, writing facts and rules in order to provide necessary information based on external queries.