

Ministerul Educației al Republicii Moldova
Universitatea Tehnică a Moldovei

Departamentul Informatică și Ingineria Sistemelor

RAPORT

Lucrarea de laborator Nr.1
la Inteligența Artificială

Tema: Modelarea unui arbore genealogic în Prolog

A elaborat:

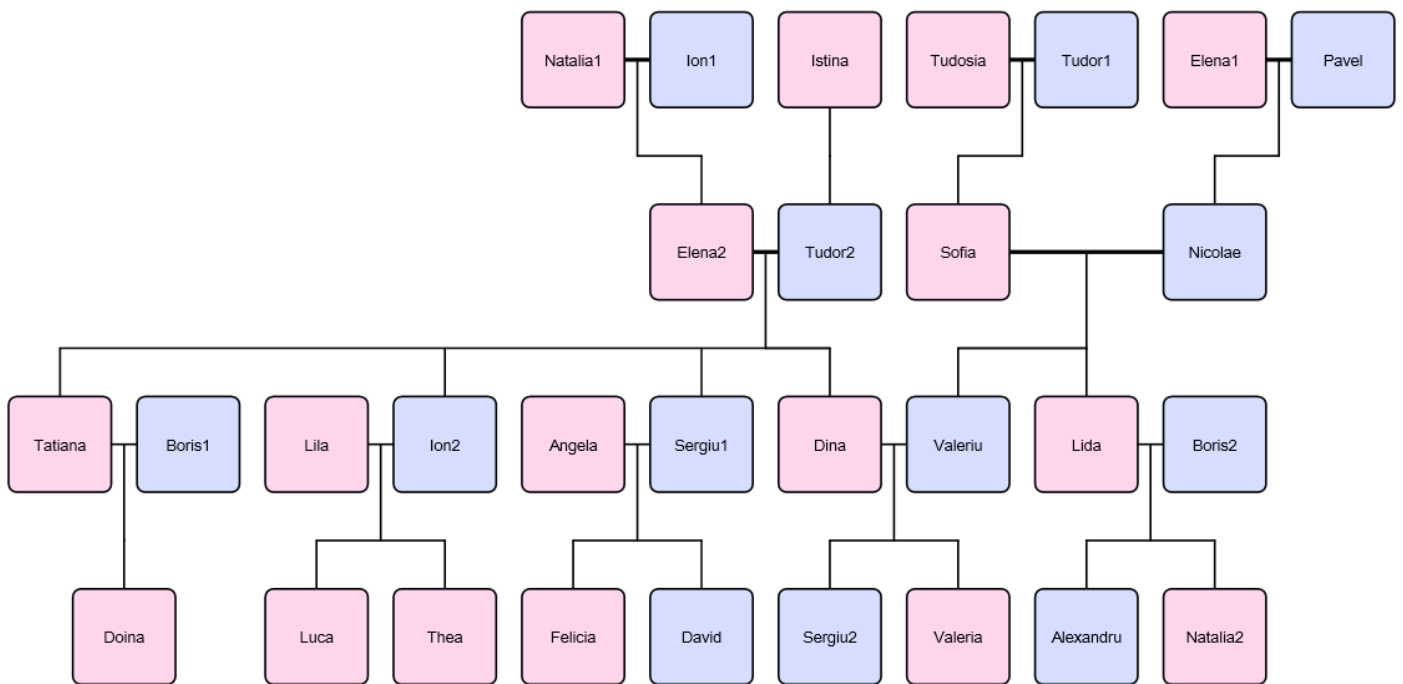
Ungureanu Valeria, C-161

A verificat:

Bumbu Tudor

Chișinău-2019

Arborele genealogic



Baza de cunoștințe:

Fapte:

```
barbat(pavel).
barbat(tudor1).
barbat(nicolae).
barbat(tudor2).
barbat(valeriu).
barbat(ion2).
barbat(sergiu1).
barbat(alexandru).
barbat(sergiu2).
barbat(david).
barbat(ion1).
barbat(boris1).
barbat(boris2).
```

```
femeie(natalia1).
femeie(istina).
femeie(elena1).
femeie(tudosia).
```

femeie(sofia).
femeie(elena2).
femeie(lida).
femeie(tatiana).
femeie(dina).
femeie(natalia2).
femeie(doina).
femeie(thea).
femeie(luca).
femeie(valeria).
femeie(felicia).
femeie(lila).
femeie(angela).

parinte(istina, tudor2).

parinte(pavel, nicolae).
parinte(elena1, nicolae).

parinte(ion1, elena2).
parinte(natalia1, elena2).

parinte(tudosia, sofia).
parinte(tudor1, sofia).

parinte(elena2, tatiana).
parinte(tudor2, tatiana).

parinte(elena2, ion2).
parinte(tudor2, ion2).

parinte(elena2, dina).
parinte(tudor2, dina).

parinte(elena2, sergiu1).
parinte(tudor2, sergiu1).

parinte(sofia, lida).
parinte(nicolae, lida).

parinte(sofia, valeriu).
parinte(nicolae, valeriu).

parinte(tatiana, doina).
parinte(boris1, doina).

parinte(ion2, thea).
parinte(lila, thea).

parinte(ion2, luca).
parinte(lila, luca).

parinte(dina, sergiu2).
parinte(valeriu, sergiu2).

parinte(dina, valeria).
parinte(valeriu, valeria).

parinte(sergiu1, felicia).
parinte(angela, felicia).

parinte(sergiu1, david).
parinte(angela, david).

parinte(lida, natalia2).
parinte(boris2, natalia2).

parinte(lida, alexandru).
parinte(boris2, alexandru).

Reguli:

1. Am creat regulile care definesc relația de paternitate și maternitate în baza celor 2 fapte declarate anterior. Dacă X este bărbat (femeie) și părintele lui Y, atunci X este tata (mama) lui Y.

tata(X, Y) :- barbat(X), parinte(X,Y).
mama(X, Y) :- femeie(X), parinte(X,Y)

2. În continuare am elaborat un set de reguli care determină existența relației de matrimoniu dintre 2 persoane. Respectiv, X este soțul(soția) lui Y, dacă X este bărbat (femeie), și dacă X și Y sunt ambii părinții lui Z. Pentru a evita duplicatele care apar în cazul în care soții au 2 sau mai mulți copii, am utilizat predicatul *setof*.

sot_cu_duplicati(X, Y) :- barbat(X), parinte(X, Z) , parinte(Y, Z), X \= Y.
sot(X, Y) :- setof(X-Y, sot_cu_duplicati(X,Y), Soti), member(X-Y, Soti).

sotie_cu_duplicati(X, Y) :- femeie(X), parinte(X, Z) , parinte(Y, Z), X \= Y.
sotie(X, Y) :- setof(X-Y, sotie_cu_duplicati(X,Y), Sotii), member(X-Y, Sotii).

3. Următorul set de reguli stabilește că X este bunelul (bunica) lui Y în cazul în care X este bărbat (femeie), X este părintele lui Z, iar Z este părintele lui Y.

bunel(X, Y) :- barbat(X), parinte(X,Z), parinte(Z,Y).
bunica(X, Y) :- femeie(X), parinte(X,Z), parinte(Z,Y).

4. Am continuat cu elaborarea regulilor care stabilesc că X este străbunicul (străbunica) lui Y : X este tatăl (mama) bunicului sau bunicii în raport cu Y.

strabunel(X, Y) :- tata(X, Z), (bunel(Z, Y) ; bunica(Z, Y)).
strabunica(X, Y) :- mama(X, Z), (bunel(Z, Y) ; bunica(Z, Y)).

5. Urmează relația de fraternitate dintre X și Y: X este bărbat (femeie), X și Y au mamă comună, X și Y nu pot fi identici, respectiv, X este fratele (sora) lui Y. Adicional, am putea completa predicatul frate(X, Y) sau sora(X, Y) cu regulile care stabilesc că X și Y au tată comun, în cazul în care mama lui X și Y nu este definită în baza de cunoștințe.

frate(X, Y) :- barbat(X), mama(Z, Y), mama(Z, X), X \= Y.
sora(X, Y) :- femeie(X), mama(Z, Y), mama(Z, X), X \= Y .

6. În continuare am elaborat setul de reguli care determină că X este unchiul (mătușa) lui Y. Această relație intervine în 2 cazuri:

- X este fratele (sora) părintelui lui Y;
 - X este soțul (soția) fratelui sau surorii părintelui lui Y.
- Între aceste 2 seturi de reguli am utilizat predicat SAU.

matusa(X, Y) :- femeie(X), parinte(Z, Y), sora(X, Z) ; matusa2(X, Y).
unchi(X, Y) :- (barbat(X), parinte(Z, Y), frate(X, Z)) ; unchi2(X, Y).
unchi2(X, Y) :- barbat(X), parinte(Z, Y), (sora(F, Z) ; frate(F, Z)), sot(X, F).
matusa2(X, Y) :- femeie(X), parinte(Z, Y), (sora(F, Z) ;
frate(F, Z)), sotie(X, F).

7. În final, am creat setul de reguli care stabilește că X este verișorul (verișoara) lui Y: X este bărbat(femeie), părintele lui Y este unchiul sau mătușa lui X. Am utilizat din nou predicatul *setof*, pentru a evita duplicatele în cazul în care pentru X în baza de cunoștințe se găsește atât unchiul, cât și mătușa care sunt părinții comuni ai lui Y.

verisor_cu_duplicati(X, Y) :- barbat(X), parinte(Z, Y),
(unchi(Z, X) ; matusa(Z, X)).
verisor(X, Y) :- setof(X-Y, verisor_cu_duplicati(X, Y), Verisori),
member(X-Y, Verisori).
verisoara_cu_duplicati(X, Y) :- femeie(X), parinte(Z, Y),
(matusa(Z, X) ; unchi(Z, X)).
verisoara(X, Y) :- setof(X-Y, verisoara_cu_duplicati(X, Y), Verisoare),
member(X-Y, Verisoare).

Rezultate:

Vom verifica care sunt străbuneii, buneii, părinții, frații, unchii și mătușile, verișorii și verișoarele predicatului *Valeria*:

?- **strabunel**(X, valeria).

X = pavel ;

X = tudor1 ;

X = ion1 ;

false.

?- **strabunica**(X, valeria).

X = natalia1 ;

X = istina ;

X = elena1 ;

X = tudosia ;

false.

?- **bunel**(X, valeria).

X = nicolae ;

X = tudor2 ;

false.

?- **bunica**(X, valeria).

X = sofia ;

X = elena2 ;

false.

?- **tata**(X, valeria).

X = valeriu ;

false

?- **mama**(X, valeria).

X = dina ;

false.

?- **frate**(X, valeria).

X = sergiu2 ;

false.

?- **unchi**(X, valeria).

X = ion2 ;

X = sergiu1 ;

X = boris1 ;

X = boris2 ;

false.

?- **matusa**(X, valeria).

X = lida ;

X = tatiana ;

X = lila ;

X = angela ;

false.

?- **verisor**(X, valeria).

X = alexandru ;

X = david.

false

?- **verisoara**(X, valeria).

X = doina ;

X = felicia ;

X = luca ;

X = natalia2 ;

X = thea.

false

Predicatul Valeria nu are soț, copii și soră:

?- **sot**(X, valeria).

false

?- **mama**(valeria, X).

false.

?- **sora**(X, valeria).

false

Concluzie: Pentru realizarea lucrării de laborator nr.1 am studiat despre limbajul de programare Prolog și elementele de bază a logicii predicatelor de ordinul 1, în baza căreia funcționează Prolog.