

Ministerul Educației al Republicii Moldova
Universitatea Tehnică a Moldovei
Facultatea Calculatoare Informatică și Microelectronică
Catedra Automatică și Tehnologii Informaționale

RAPORT

Lucrare de laborator nr 2

Disciplina: Programarea Funcțională și Inteligența Artificială

Tema: Mecanisme de control al procesului de backtracking.

A efectuat:

Vovc Artemie st. TI-133

A verificat:

Lazu Victoria lect. super.

Chișinău 2016

Cuprins

1 Scopul lucrării	3
1.2 Sarcina.....	3
2 Noțiuni teoretice.....	5
3 Realizarea.....	7
Concluzia	11
Bibliografia	12
Anexe A	13

1 Scopul lucrării

Însușirea noțiunilor privind mecanismele specifice limbajului Prolog pentru controlul procesului de backtracking: cut și fail.

1.2 Sarcina

1. Citiți breviarul teoretic. Se atrage atenția asupra faptului că toate cunoștințele din această lucrare vor fi necesare și la efectuarea celorlalte lucrări.

2. Lansați la executare programul elaborat în lucrarea 1 și cercetați schimbările semanticii procedurale:

- prin schimbarea ordinii propozițiilor-fapte;
- prin schimbarea ordinii propozițiilor-reguli; (două variante);
- prin schimbarea subscopurilor în reguli; (două variante);
- trageți concluzii.

3. Rezolvați următoarele probleme propuse și urmăriți execuția lor corectă.

3.1. Elaborați și testați un program pentru determinarea unei valori minime din două numere (X și Y) fără utilizarea predicatului cut.

3.2. Elaborați și testați un program pentru determinarea unei valori minime din două numere (X și Y), utilizând predicatul cut roșu și cut verde.

3.3. Care vor fi răspunsurile programului

p(1).

p(2) :- !.

p(3).

Efectuați o analiză comparativă între utilizarea predicatelor cut în spațiul bazei de cunoștințe și spațiul scopurilor pentru întrebările formulate în lista de scopuri ce urmează:

p(X).

p(X), p(Y).

p(X), !, p(Y).

3.4. Doi copii pot juca un meci într-un turneu de tenis dacă au aceeași vârstă. Fie următorii copii și vârstele lor:

`copil(peter,9). copil(paul,10). copil(chris,9). copil(susan,9).`

Definiți un predicat din care rezultă toate perechile de copii care pot juca un meci într-un turneu de tenis.

4. Introduceți schimbările corespunzătoare în programul din punctul 2., utilizând cut verde cel puțin în două reguli din baza de cunoștințe.

5. Introduceți schimbările corespunzătoare în programul din punctul 2., utilizând cut roșu în reguli din baza de cunoștințe. Trageți concluzii.

6. Prezentați darea de seamă.

2 Noțiuni teoretice

Sistemul Prolog se implică automat într-un proces de backtracking atunci când acest lucru este necesar pentru satisfacerea unui scop. În unele situații acest comportament poate fi deosebit de util, pe când în altele poate deveni foarte ineficient. Se consideră următorul exemplu de program în care se definesc valorile unei funcții:

```
f(X, 0) :- X < 3. % 1
f(X, 2) :- 3 =< X, X < 6. % 2
f(X, 4) :- 6 =< X. % 3
```

La întrebarea:

```
?- f(1, Y).
```

```
Y = 0
```

răspunsul sistemului indică faptul că valoarea funcției pentru $X=1$ este $Y=0$. Dacă se pune întrebarea formată din conjuncția de scopuri:

```
?- f(1, Y), 2 < Y.
```

```
no
```

sistemul semnalează un eșec.

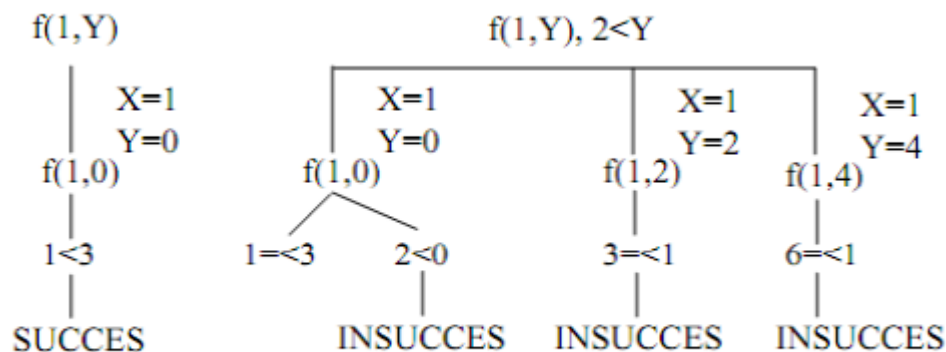


Figura 2.1. Arbori de deducție a scopurilor $f(1,Y)$ și $f(1,Y), 2<Y$

Se observă că se încearcă resatisfacerea primului scop cu regulile 2 și 3, iar acest lucru este inutil datorită semanticii acestor reguli. Astfel, dacă o valoare mai mică decât 3 pentru X duce la eșecul scopului S din conjuncția de scopuri $f(X, Y), S$, este inutil să se încerce resatisfacerea scopului f , deoarece aceasta nu este posibilă. Încercarea de resatisfacere a scopului $f(X, Y)$ poate fi împiedicată prin introducerea predicatului cut.

16 Figura 2.1. Arbori de deducție a scopurilor $f(1,Y)$ și $f(1,Y), 2<Y$

Predicatul cut, notat cu atomul special $!$, este un predicat standard, fără argumente, care poate fi îndeplinit (este adevărat) întotdeauna și nu poate fi resatisfăcut.

Predicatul cut are următoarele efecte laterale:

- La apariția predicatului cut, toate selecțiile făcute între scopul antet al regulii și cut sunt "înghețate", deci marcajele de satisfacere a scopurilor sunt eliminate, ceea ce duce la eliminarea oricăror altor soluții alternative pentru această porțiune. O încercare de resatisfacere a unui scop între scopul antet de regula și scopul curent va eșua.

- Dacă clauza în care s-a introdus predicatul cut reușește, toate clauzele cu același antet, succesive clauzei în care a apărut cut, vor fi ignorate.

- Printr-o descriere succintă, comportamentul predicatului cut este următorul:

(C1) $H :- D1, D2, \dots, Dm, !, Dm+1, \dots, Dn.$

(C2) $H :- A1, \dots, Ap.$

(C3) $H.$

Dacă $D1, D2, \dots, Dm$ sunt satisfăcute, ele nu mai pot fi resatisfăcute datorită lui cut. Dacă $D1, \dots, Dm$ sunt satisfăcute, C2 și C3 nu vor mai fi utilizate pentru re-satisfacerea lui H. Resatisfacerea lui H se poate face numai prin resatisfacerea unuia din scopurile $Dm+1, \dots, Dn$, dacă acestea au mai multe soluții.

3 Realizarea

Prima sarcină e să facem arborele genealogic vezi mai jos in figura 1 arborele genealogic.

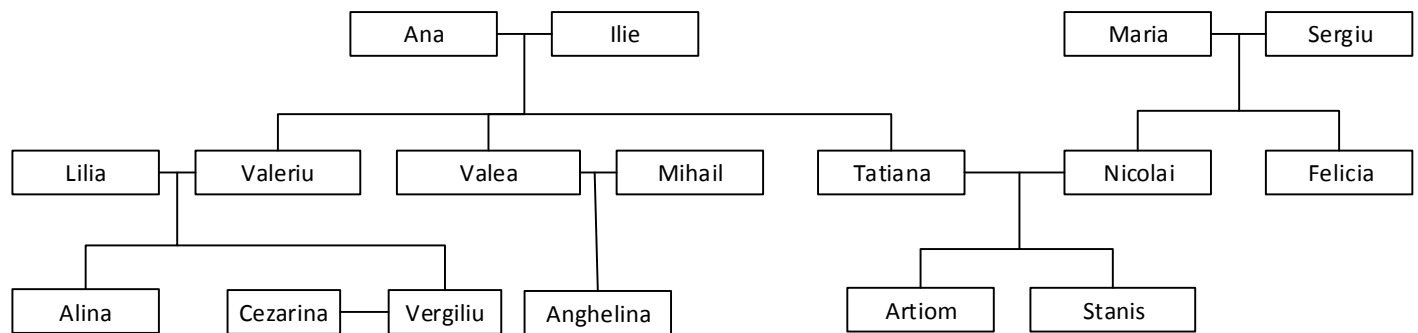


Figura 3.1 – Arborele genealogic

Baza de cunoștințe pentru figura 3.1 de mai sus.

```
femeie(ana).
femeie(maria).
femeie(lilia).
femeie(valea).
femeie(tatiana).
femeie(felicia).
femeie(alina).
femeie(anghelina).
femeie(cezarina).
barbat(ilie).
barbat(sergiu).
barbat(valeriu).
barbat(nicolai).
barbat(mihail).
barbat(vergiliu).
barbat(artiom).
barbat(stanis).
parinti(ana,ilie,valeriu).
parinti(ana,ilie,tatiana).
parinti(ana,ilie,valea).
parinti(maria,sergiu,nicolai).
parinti(maria,sergiu,felicia).
parinti(lilia,valeriu,alina).
parinti(lilia,valeriu,vergiliu).
parinti(valea,mihail,anghelina).
parinti(tatiana,nicolai,stanis).
parinti(tatiana,nicolai,artiom).
```

Și formează interogările:

```
mama(X,Y):-
parinti(X,_,Y),femeie(X),write(X),write(""),write(Y),nl,fail;true.
tata(X,Y):- parinti(_,X,Y),barbat(X).
frati(X,Y):- parinti(M,T,X),parinti(M,T,Y),X\=Y.
sora(X,Y):- frati(X,Y),femeie(X).
frate(X,Y):- frati(X,Y),barbat(X).
dumneaieiAreFrate(X,Y):- sora(X,Y),!,write("Pentru: "),write(X),write("
frate este: "),write(Y),nl.
dumnealuiAreSora(X,Y):- sora(X,Y),!,write("Sora: "),write(X),write("
pentru: "),write(Y),nl.
frate0(X,Y,X):- parinti(M,T,X),parinti(M,T,Y),X\=Y,!.
frate0(_,Y,Y).
verisori(X,Y):- parinti(M1,T1,X),parinti(M2,T2,Y),
(
    frati(M1,M2);
    frati(M1,T2);
    frati(T1,M2);
    frati(T1,T2)
),X\=Y,not(frati(X,Y)),write(X),write(" "),write(Y),nl,fail;true.
bunici(X,Y,N):- parinti(X,Y,Z),(parinti(Z,_,N);parinti(_,Z,N)).
bunica(X,N):- bunici(X,_,N).
bunel(X,N):-bunici(_,X,N).
nepot(X):-parinti(M,T,X),(parinti(_,_,M);parinti(_,_,T)).
casatoriti(X,Y):- parinti(X,Y,_).
unchi(X,Y):-parinti(M,T,Y),
((frate(X,M);frate(X,T));
( parinti(M1,T1,M),parinti(M1,T1,C1),
casatoriti(C1,X),not(parinti(_,X,Y))));
( parinti(M2,T2,T),parinti(M2,T2,C2),
casatoriti(C2,X),not(parinti(_,X,Y))
)).
matusa(X,Y):-parinti(M,T,Y),
((sora(X,M);sora(X,T));
( parinti(M1,T1,M),parinti(M1,T1,C1),
casatoriti(X,C1),not(parinti(X,_,Y))));
( parinti(M2,T2,T),parinti(M2,T2,C2),
casatoriti(X,C2),not(parinti(X,_,Y))
)).
p(1).
p(2):-!.
p(3).
copil(peter,9). copil(paul,10). copil(chris,9). copil(susan,9).
joaca():- copil(X,M),copil(Y,M),X\=Y,write(X),write("
"),write(Y),nl,fail;true.
min0(X,Y,Y):-X>=Y.
min0(X,Y,X):-X<Y.
min1(X,Y,Y):-X>=Y,!.

```



```

min1 (X, Y, X) :-X<Y.
min2 (X, Y, Y) :-X>=Y, !.
min2 (X, _, X) .

```

Realizarea punctului trei din sarcină: determinarea valorii minime.

```

min0 (X, Y, Y) :-X>=Y.
min0 (X, Y, X) :-X<Y.

```

```

7 ?- min0(3,4,R).
R = 3.

8 ?- min0(23,4,R).
R = 4 ;
false.

9 ?- min0(23,42,R).
R = 23.

```

Figura 3.2 – Rezultat la interogarea min fara cut

- Folosirea cut-ului verde:

```

min1 (X, Y, Y) :-X>=Y, !.
min1 (X, Y, X) :-X<Y.

```

```

10 ?- min1(3,4,R).
R = 3.

11 ?- min1(23,4,R).
R = 4.

12 ?- min1(23,42,R).
R = 23.

```

Figura 3.3 – Rezultatul interogării

- Folosirea cut-ului roșu:

```

min2 (X, Y, Y) :-X>=Y, !.
min2 (X, _, X) .

```

```

13 ?- min2(3,4,R).
R = 3.

14 ?- min2(23,4,R).
R = 4.

15 ?- min2(23,42,R).
R = 23.

```

Figura 3.4 – Rezultatul interogării

Realizarea punctului trei punct trei din sarcină:

```

p(1) .
p(2) :-!.
p(3) .

```

```

4 ?- p(X) .
X = 1 ;
X = 2 .

5 ?- p(X), p(Y) .
X = Y, Y = 1 ;
X = 1,
Y = 2 ;
X = 2,
Y = 1 ;
X = Y, Y = 2 .

6 ?- p(X), !, p(Y) .
X = Y, Y = 1 ;
X = 1,
Y = 2 .

```

Figura 3.5 – Rezultatul interogării

Realizarea sarcinii trei punct patru:

```

copil(peter,9) . copil(paul,10) . copil(chris,9) . copil(susan,9) .
joaca() :-
copil(X,M),copil(Y,M),X\=Y,write(X),write(""),write(Y),nl,fail,true.
16 ?- joaca()
peter chris
peter susan
chris peter
chris susan
susan peter
susan chris
true.

```

Figura 3.6 – Rezultatul interogării

Realizarea sarcinii patru și cinci:

```

17 ?- dumneaieiAreFrate(alina,Frate)
Pentru: alina frate este: vergiliu
Frate = vergiliu.

```

Figura 3.7 – Rezultatul interogării cut verde

```

18 ?- frate0(_,artiom,Resultat).
Resultat = stanis.

```

Figura 3.8 – Rrezultatul interioogării cut roșu

Concluzia

Lucrarea dată a avut ca scop să ne facă cunoscuți cu programarea logică la general și programarea în prolog esențial. În lucrarea dată am lucrat cu concepția de cut roșu și verde asupra arborelui genialogic format în laboratorul precedent. S-a utilizat și fail clauza pentru a produce backtrackingul.

Bibliografia

- 1 **Prolog la general** : <http://biblioteca.regielive.ro/referate/limbaje-de-programare/prolog-limbaj-de-programare-logica-114586.html>
- 2 **Studiarea SWI-Prolog** : <http://www.swi-prolog.org/pldoc/index.html>

Anexe A

Baza de cunoștințe

```
femeie(ana).
femeie(maria).
femeie(lilia).
femeie(valea).
femeie(tatiana).
femeie(felicia).
femeie(alina).
femeie(anghelina).
femeie(cezarina).
```

```
barbat(ilie).
barbat(sergiu).
barbat(valeriu).
barbat(nicolai).
barbat(mihailescu).
barbat(vergiliu).
barbat(artiom).
barbat(stanis).
```

```
parinti(ana,ilie,valeriu).
parinti(ana,ilie,tatiana).
parinti(ana,ilie,valea).
parinti(maria,sergiu,nicolai).
parinti(maria,sergiu,felicia).
parinti(lilia,valeriu,alina).
parinti(lilia,valeriu,vergiliu).
parinti(valea,mihailescu,angelina).
parinti(tatiana,nicolai,stanis).
parinti(tatiana,nicolai,artiom).
```

```
mama(X,Y):- parinti(X,_,Y),femeie(X),write(X),write("
"),write(Y),nl,fail;true.
tata(X,Y):- parinti(_,X,Y),barbat(X).
frati(X,Y):- parinti(M,T,X),parinti(M,T,Y),X\=Y.
sora(X,Y):- frati(X,Y),femeie(X).
frate(X,Y):- frati(X,Y),barbat(X).
```

```
dumneaieiAreFrate(X,Y):- sora(X,Y),!,write("Pentru: "),write(X),write("
frate este: "),write(Y),nl.
dumnealuiAreSora(X,Y):- sora(X,Y),!,write("Sora: "),write(X),write("
pentru: "),write(Y),nl.
```

```
frate0(X,Y,X):- parinti(M,T,X),parinti(M,T,Y),X\=Y,!.
frate0(_,Y,Y).
```

```
verisori(X,Y):- parinti(M1,T1,X),parinti(M2,T2,Y),
```

```

(
    frati(M1,M2);
    frati(M1,T2);
        frati(T1,M2);
        frati(T1,T2)
    ),X\=Y,not(frati(X,Y)),write(X),write(" "),write(Y),nl,fail;true.

bunici(X,Y,N):- parinti(X,Y,Z), (parinti(Z,_,N);parinti(_,Z,N)).
bunica(X,N):- bunici(X,_,N).
bunel(X,N):-bunici(_,X,N).
nepot(X):-parinti(M,T,X), (parinti(_,_,M);parinti(_,_,T)).
casatoriti(X,Y):- parinti(X,Y,_).
unchi(X,Y):-parinti(M,T,Y),
    ((frate(X,M);frate(X,T));
    ( parinti(M1,T1,M),parinti(M1,T1,C1),
    casatoriti(C1,X),not(parinti(_,X,Y)) );
    ( parinti(M2,T2,T),parinti(M2,T2,C2),
    casatoriti(C2,X),not(parinti(_,X,Y))
    ) ).
matusa(X,Y):-parinti(M,T,Y),
    ((sora(X,M);sora(X,T));
    ( parinti(M1,T1,M),parinti(M1,T1,C1),
    casatoriti(X,C1),not(parinti(X,_,Y)) );
    ( parinti(M2,T2,T),parinti(M2,T2,C2),
    casatoriti(X,C2),not(parinti(X,_,Y))
    ) ).

p(1).
p(2):-!.
p(3).

copil(peter,9). copil(paul,10). copil(chris,9). copil(susan,9).
joaca():- copil(X,M),copil(Y,M),X\=Y,write(X),write("
"),write(Y),nl,fail;true.

min0(X,Y,Y):-X>=Y.
min0(X,Y,X):-X<Y.

min1(X,Y,Y):-X>=Y,!.
min1(X,Y,X):-X<Y.

min2(X,Y,Y):-X>=Y,!.
min2(X,_,X).

```