**Ministerul Educaţiei al Republicii Moldova**

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare Informatică şi Microelectronică**

**Catedra Automatică şi Tehnologii Informaţionale**

**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](#_Toc463336762)

[1.2 Sarcina 3](#_Toc463336763)

[2 Noțiuni teoretice 5](#_Toc463336764)

[3 Realizarea 7](#_Toc463336765)

[Concluzia 11](#_Toc463336766)

[Bibliografia 12](#_Toc463336767)

[Anexe A 13](#_Toc463336768)

# 1 Scopul lucrării

Însuşirea noţiunilor privind mecanismele specifice limbajului Prolog pentru controlul procesului de backtracking: cut şi fail.

# 1.2 Sarcina

1. Citiţi breviarul teoretic. Se atrage atenţia asupra faptului că toate cunoştinţele din această lucrare vor fi necesare şi la efectuarea celorlalte lucrări.

2. Lansaţi la executare programul elaborat în lucrarea 1 şi cercetaţi schimbările semanticii procedurale:

- prin schimbarea ordinii propoziţiilor-fapte;

- prin schimbarea ordinii propoziţiilor-reguli; (două variante);

- prin schimbarea subscopurilor în reguli; (două variante);

- trageţi concluzii.

3. Rezolvaţi următoarele probleme propuse şi urmăriţi execuţia lor corectă.

3.1. Elaboraţi şi testaţi un program pentru determinarea unei valori minime din două numere (X şi Y) fără utilizarea predicatului cut.

3.2. Elaboraţi şi testaţi un program pentru determinarea unei valori minime din două numere (X şi Y), utilizând predicatul cut roşu şi cut verde.

3.3. Care vor fi răspunsurile programului

р( 1).

р( 2) :- !.

р( 3).

Efectuaţi o analiză comparativă între utilizarea predicatelor cut în spaţiul bazei de cunoştinţe şi spaţiul scopurilor pentru întrebările formulate în lista de scopuri ce urmează:

р( X).

р( X), p(Y).

р( X), !, p(Y).

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

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

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

4. Introduceţi schimbările corespunzătoare în programul din punctul 2., utilizând cut verde cel puţin în două reguli din baza de cunoştinţe.

5. Introduceţi schimbările corespunzătoare în programul din punctul 2., utilizând cut roşu în reguli din baza de cunoştinţe. Trageţi concluzii.

6. Prezentaţi 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 situaţii 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 funcţii:

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).

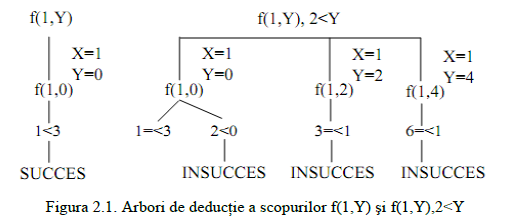
Y = 0

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

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

no

sistemul semnalează un eşec.



Se observă că se încearcă resatisfacerea primului scop cu regulile 2 şi 3, iar acest lucru este inutil datorită semanticii acestor reguli. Astfel, dacă o valoare mai mică decât 3 pentru X duce la eşecul scopului S din conjuncţia 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 deducţie a scopurilor f(1,Y) şi f(1,Y),2<Y

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

Predicatul cut are următoarele efecte laterale:

• La apariţia predicatului cut, toate selecţiile făcute între scopul antet al regulii şi cut sunt "îngheţate", deci marcajele de satisfacere a scopurilor sunt eliminate, ceea ce duce la eliminarea oricăror altor soluţii alternative pentru această porţiune. O încercare de resatisfacere a unui scop între scopul antet de regula şi scopul curent va eşua.

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

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

(C1) H :- D1, D2, …, Dm, !, Dm+1, …, Dn.

(C2) H :- A1, …,Ap.

(C3) H.

Dacă D1, D2, …, Dm sunt satisfăcute, ele nu mai pot fi resatisfăcute datorită lui cut. Dacă D1, …, Dm sunt satisfăcute, C2 şi 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, …, Dn, dacă acestea au mai multe soluţii.

# 3 Realizarea

Prima sarcină e să facem arborele genealogin 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 formez 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.

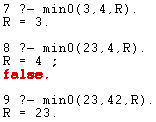


Figura 3.2 – Rezultat la interogarea min fara cut

- Folosirea cut-ului verde:

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

min1(X,Y,X):-X<Y.

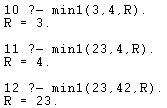


Figura 3.3 – Rezultatul interogării

- Folosirea cut-ului roșu:

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

min2(X,\_,X).

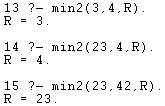


Figura 3.4 – Rezultatul interogării

Realizarea punctului trei punct trei din sarcină:

p(1).

p(2):-!.

p(3).

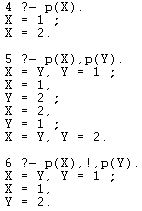


Figura 3.5 – Rezultatul interogării

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.

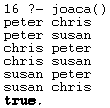


Figura 3.6 – Rezultatul interogării

Realizarea sarcinii patru și cinci:



Figura 3.7 – Rezultatul interogării cut verde



Figura 3.8 – Rrezultatul interiogării 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 asupara 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. **Studierea 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(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).

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).