

Programare Logică – Laboratorul 3

Recursivitate

Isabela Drămnesc

1 Întrebări din curs

- Ce sunt variabilele anonime? Când le folosim?
- Ce înțelegeți prin unificare? Cum se produce unificarea?
- Ce sunt structurile? Cum reprezentăm:

```
curs( datep( nume( ardelean ), prenume( ionel ) ),  
      datec( marti, cursn( pl ), sala( a11 ) ) ),  
biblioteca( universitate( deVest ),  
            carte( poezii, autor( mihai, eminescu ) ) ).
```

- De ce avem nevoie pentru a afla soluții pentru întrebările:
 1. Cine are curs de pl (Programare Logică) marți în sala 102?
 2. Ce cărți găsim la biblioteca universității din cluj?
- Cum reprezentăm și cum interogăm în Prolog:
 1. $3X^2 + (7N/Z^2)$.
 2. $X^{10} - 2X^4$

2 Concepte

- Trace
- Domeniu inductiv
- Recursivitate
- Proceduri recursive
- Condiție limită (de oprire)
- Apel recursiv
- Operații recursive pe liste
 - Mapare recursivă
 - Comparare recursivă
 - Îmbinare recursivă

3 Trace

Introduceți următoarea bază de date:

```
vizitat(ionel,spania).  
vizitat(maria,spania).  
vizitat(ioana,italia).  
vizitat(ioana,cehia).
```

```
calator(ionel).  
calator(ioana).  
calator(victoria).
```

```
calatorie(X,Y):-calator(X),vizitat(X,Y).
```

Pentru fiecare din următoarele interogări urmăriți ce se întâmplă cu ajutorul comenzii *trace* :

```
?-calatorie(ionel,X).  
?-calatorie(maria,X).  
?-calatorie(N,spania).  
?-calatorie(C,cehia).  
?-calatorie(X,Y).
```

4 Inducție/Recursie

4.1 Domeniu inductiv

Este un domeniu compus din obiecte compuse care se pot descompune într-un număr finit de obiecte "mai simple". Se continuă procedeul până se ajunge la "cele mai simple" obiecte.

Exemplu de domeniu inductiv:

```
[]      /*-lista vida; */  
.(h,t)  /* -lista generica, unde h=head (capul),  
        t=tail (coada) listei.  
        Coada listei trebuie sa fie o lista. */
```

4.2 Proceduri recursive

O procedură recursivă constă în:

- Condițiile la limită (se tratează cazul când calculul se oprește, pe cele mai simple obiecte).
- Apelul recursiv se face doar pe argumente "mai simple", în cazul nostru apelul recursiv se face pe coada listei.

Exemplu: Predicatul care determină dacă un anumit termen este listă sau nu:

```

1)
este_lista ([ ]). /* conditia la limita */
este_lista ([H|T]:- este_lista (T). /* apel recursiv */

/* Cate solutii obtin pentru interogarea: */
?- este_lista (A).

```

Ordinea declarării clauzelor este foarte importantă:

Atenție la cazuri de genul:

```

2)
este_lista ([H|T]:- este_lista (T).
este_lista ([ ]).

```

```

?- este_lista (X).

```

```

3)
parinte (X,Y):- copil (Y,X).
copil (X,Y):- parinte (Y,X).

```

```

4)
persoana (X):- persoana (Y) , tata (Y,X).
persoana (ion).
tata (gabriel , ion).

```

```

?-persoana (X).

```

4.3 Exerciții:

5) Pentru exercițiul ”Scrieți un predicat schimbă primele 2/2 (binar) care acceptă o listă și generează o listă similară, numai cu primele două elemente schimbate” (din temă) interogați în Prolog:

```

?- schimba ([1,2,3,4],X).
?- schimba (X,Y) , schimba (Y,X).

```

Ce observați? Folosiți comanda *trace* pentru a vedea ce se întâmplă la ultima interogare.

6) Predicatul pentru a vedea dacă un element este membru al listei. Exemplu:

```

membru (X,[X,-]).
membru (X,[_|Y]):- membru (X,Y).

```

```

?-membru (3,[1,2,3]).

```

7) Exemplu pentru un predicat care elimină un element dintr-o listă.

```

eliminare (X,[X|T],T).
eliminare (X,[Y|T1],[Y|T2]):- eliminare (X,T1,T2).

```

Testați în Prolog toate posibilitățile!

8) Scrieți un predicat care calculează $n!$ Știm că $0! = 1$.

9) Scrieți un predicat care să calculeze cmmdc a două numere. Se folosește definiția recursivă a lui Euclid: Fie a și b două numere întregi pozitive. Dacă $b=0$, atunci $\text{cmmdc}(a,b)=a$; altfel $\text{cmmdc}(a,b)=\text{cmmdc}(b,r)$, unde r este restul împărțirii lui a la b .

5 Mapare recursivă

Mapare: fiind date 2 structuri similare să se schimbe prima în a doua după niște reguli date.

Exemplu: "you are a computer" se mapează în "i am not a computer", "do you speak french" se mapează în "i do not speak german".

```
10)
change(you, i).
change(are, [am, not]).
change(french, german).
change(do, no).
change(X, X).

alter([], []).
alter([H|T], [X|Y]):- change(H, X), alter(T, Y).

?- alter([you, are, a, computer], W).
?- alter([i, do, like, you], W).
```

6 Îmbinarea structurilor (Concatenare)

11) Se dau două liste A și B , scrieți un predicat care să returneze o listă care conține elementele listei A urmate de elementele listei B . Exemplu: pentru $A=[a,b,c]$, $B=[7,8,9]$, lista returnată va fi $C=[a,b,c,7,8,9]$.

Testați și următoarele interogări:

```
?- concatenare([1,2,3], [s,d,3,4], X).

?- concatenare(X, [1,2,3], [1,2,s,d,3,4]).

?- concatenare(X, [1,2,3], [d,f,g,ssss,1,2,3]).

?- concatenare(X,Y, [d,f,g,ssss,1,2,3]).
```

7 Tema:

[Tema 3.](#)

Termen de realizare: laboratorul următor.