

# Kombinatorika

U okviru PROLOG-a dosta značaja se posvećuje kombinatornim problemima. Kombinatorni problemi se pogodno rešavaju korišćenjem rekurzije i u proceduralnim jezicima.

## 1. Napisati PROLOG program za štampanje svih listi koje se sastoje od elemenata 0 i 1 date dužine.

glavni(N) :- generisi(L,N), write(L), nl, fail.

generisi([],0) :- ! .

generisi([G|R],N) :- cifra(G), M is N-1, generisi(R,M).

cifra(0).

cifra(1).

Program se poziva sa glavni(N).

Napomena: Dati program generiše sve liste u prirodnom poretku zdesna nalevo, sa malom modifikacijom moguće je generisanje sleva udesno.

Izlaz:

1 ?- glavni(4).

[0,0,0,0]

[0,0,0,1]

[0,0,1,0]

[0,0,1,1]

[0,1,0,0]

[0,1,0,1]

[0,1,1,0]

[0,1,1,1]

[1,0,0,0]

[1,0,0,1]

[1,0,1,0]

[1,0,1,1]

[1,1,0,0]

[1,1,0,1]

[1,1,1,0]

[1,1,1,1]

false.

**Zadatak: Modifikovati prethodni program u smislu gore navedene napomene.**

## 2. Napisati PROLOG program koji će generisati sve podskupove date liste.

genpod([],[]).

genpod([G|R],L):-genpod(R,L).

genpod([G|R],[G|R1]):-genpod(R,R1).

Izlaz:

1 ?- genpod([4,5,6],X).

X = [] ;

X = [6] ;

X = [5] ;

X = [5, 6] ;

X = [4] ;

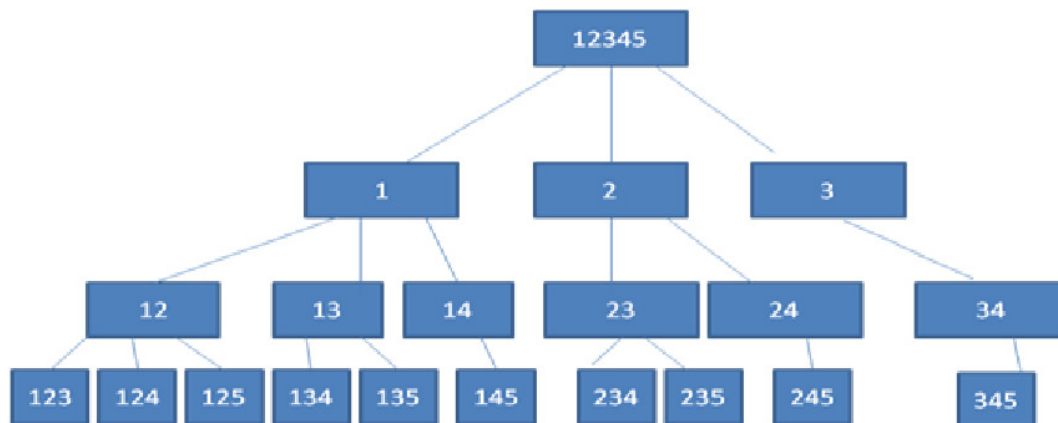
X = [4, 6] ;

X = [4, 5] ;

X = [4, 5, 6].

### 3. Napisati PROLOG program koji će generisati sve kombinacije K-te klase.

IDEJA za L=[1,2,3,4,5], k=3 => rekurzivno stablo



komb(L,0,[]).

komb([G|R],K,[G|R1]):-K>0, K1 is K-1, komb(R,K1,R1).

komb([G|R],K,L):-K>0,komb(R,K,L).

Izlaz:

1 ?- komb([2,4,6,8,10,12],3,X).

X = [2, 4, 6] ;

X = [2, 4, 8] ;

X = [2, 4, 10] ;

X = [2, 4, 12] ;

X = [2, 6, 8] ;

X = [2, 6, 10] ;

X = [2, 6, 12] ;

X = [2, 8, 10] ;

X = [2, 8, 12] ;

X = [2, 10, 12] ;

```

X = [4, 6, 8] ;
X = [4, 6, 10] ;
X = [4, 6, 12] ;
X = [4, 8, 10] ;
X = [4, 8, 12] ;
X = [4, 10, 12] ;
X = [6, 8, 10] ;
X = [6, 8, 12] ;
X = [6, 10, 12] ;
X = [8, 10, 12] ;
false.

```

**Zadatak: Napisati u C/C++/Java odgovarajući program. Izvršiti poredenje.**

PODSECANJE – pseudo kod

Kako redosled elemenata u kombinaciji nije bitan, pri ispisu je dovoljno voditi računa da element sa pozicije  $i$ ,  $i < j$  nikad ne stavimo posle elementa na poziciji  $j$ . Ostaje jos da se kodira algoritam za ispisivanje svih kombinacija je sličan prethodno opisanim algoritmima.

```

=====
function Combinations( position, start, ref combination, a, n, k )
  if position = k then
    print array combination
  end if

  if start = n+1 then
    return // potrošili smo sve elemente
  end if

  for i = begin to n do // begin je jedan element odmah posle prethodno stavljenog
    combination[ position ] = a[i]
    Combinations( positions+1, i+1, combination, a, n, k )
  end for
end function

```

**4. Definis predikat kojim se odredjuju sve permutacije date liste.**

/\*prvo resenje\*/

```

perm([],[]).
perm([G|R],L):-perm(R,L1), ubaci(G,L1,L).
/* dodaj element X na pocetak liste */
ubaci(X, [], [X]).
ubaci(X, [G|R], [X,G|R]).
ubaci(X, [G|R], [G|R1]):-ubaci(X, R, R1).

```

```

IZLAZ (1. nacin)
1 ?- perm([1,2,3,4], L).
L = [1, 2, 3, 4] ;

```

```

L = [2, 1, 3, 4] ;
L = [2, 3, 1, 4] ;
L = [2, 3, 4, 1] ;
L = [1, 3, 2, 4] ;
L = [3, 1, 2, 4] ;
L = [3, 2, 1, 4] ;
L = [3, 2, 4, 1] ;
L = [1, 3, 4, 2] ;
L = [3, 1, 4, 2] ;
L = [3, 4, 1, 2] ;
L = [3, 4, 2, 1] ;
L = [1, 2, 4, 3] ;
L = [2, 1, 4, 3] ;
L = [2, 4, 1, 3] ;
L = [2, 4, 3, 1] ;
L = [1, 4, 2, 3] ;
L = [4, 1, 2, 3] ;
L = [4, 2, 1, 3] ;
L = [4, 2, 3, 1] ;
L = [1, 4, 3, 2] ;
L = [4, 1, 3, 2] ;
L = [4, 3, 1, 2] ;
L = [4, 3, 2, 1] ;
false.

```

```

/* drugo resenje*/
perm1([],[]).
perm1(L,[X|R]):-izbor(L,X,Ost), perm1(Ost, R).

```

```

izbor([G|R], G, R).
izbor([G|R], X, [G|R1]):- izbor(R,X,R1).

```

```

1 ?- izbor([1,2,3,4],X,L).
X = 1,
L = [2, 3, 4] ;
X = 2,
L = [1, 3, 4] ;
X = 3,
L = [1, 2, 4] ;
X = 4,
L = [1, 2, 3] ;
false.

```

## 5. Odredi sve varijacije bez ponavljanja klase K date liste L.

```

/* 1.nacin */

```

var1(L,K,NL):- komb(L,K,L1), perm(L1,NL).

IZLAZ

2 ?- var1([1,2,3,4],3,L).

L = [1, 2, 3] ;

L = [2, 1, 3] ;

L = [2, 3, 1] ;

L = [1, 3, 2] ;

L = [3, 1, 2] ;

L = [3, 2, 1] ;

L = [1, 2, 4] ;

L = [2, 1, 4] ;

L = [2, 4, 1] ;

L = [1, 4, 2] ;

L = [4, 1, 2] ;

L = [4, 2, 1] ;

L = [1, 3, 4] ;

L = [3, 1, 4] ;

L = [3, 4, 1] ;

L = [1, 4, 3] ;

L = [4, 1, 3] ;

L = [4, 3, 1] ;

L = [2, 3, 4] ;

L = [3, 2, 4] ;

L = [3, 4, 2] ;

L = [2, 4, 3] ;

L = [4, 2, 3] ;

L = [4, 3, 2] ;

false.

/\* 2. nacin\*/

var2(L,0,[]).

var2(L,K,[X|R]):-K>0, izbor(L,X,Ost), K1 is K-1, var2(Ost,K1,R).

**6. Odredi sve varijacije sa ponavljanjem klase K date liste L.**

varp(L,0,[]).

varp(L,K,[X|R]):-K>0, izbor(L,X,Ost), K1 is K-1, varp(L,K1,R).

IZLAZ

3 ?- varp([1,2],3,L).

L = [1, 1, 1] ;

L = [1, 1, 2] ;

L = [1, 2, 1] ;

L = [1, 2, 2] ;

L = [2, 1, 1] ;

L = [2, 1, 2] ;

L = [2, 2, 1] ;

L = [2, 2, 2] ;

false.

## Neki primeri primene kombinatorike

1. Odredi listu operacija {+,-,\*,//} koje treba staviti umesto ? u izrazu  $((\dots(((1 ? 2) ? 3) ? 4) ? \dots) ? N)$  tako da vrednost izraza bude X.

Očekivani izlaz

N=4, X=6, znamo da  $((((1-2)+3)+4) = 6$

1 ?- form1(4,6,L).

L = [-, +, +] ;

false.

N=4, X=20, znamo da  $((((1*2)+3)*4) = 20$

2 ?- form1(4,20,L).

L = [\* , + , \*] ;

false

Ideja:

Zapisimo dati izraz u obliku  $S = ((\dots(((1 a_2 2) a_3 3) a_4 4) a_5 5) a_6 6) \dots) ? N)$

Dakle, elementi niza  $a_2 \dots a_6$  su znaci aritmetickih operacija {+,-,\*,//} koje predstavimo vednostima

1,2,3,4

Svih mogucih varijacija za 5 clanova niza sa 4 operacija moze biti veoma mnogo ( $4^5$ ), onda pokusavamo da implementoramo efikasno resenje.

Izracunavajuci izraz S pamticemo medjurezultat u elementima niza (liste)  $b_1, \dots, b_6$  tako da

$b_1 = 1$

$b_2 = b_1 a_2 2, \dots, b_6 = b_5 a_6 6, S = b_6$

Menjajuci varijante, najpre cemo menjati operatore  $a_6 = +, -, *, //$  tj.  $a_6 = 1, 2, 3, 4$

Za svaku od tih vrednosti  $a_6$ , izracunava se samo  $b_6$

izborop(+, A, B, C):-C is A+B.

izborop(-, A, B, C):-C is A-B.

izborop(\*, A, B, C):-C is A\*B.

izborop(/, A, B, C):-C is A/B.

Potom povecavamo  $a_5 = a_5 + 1$ , preračunavajuci  $b_5$  i ponovo posmatramo  $a_6 = 1, 2, 3, 4$  i tako dalje ...

Od N dobijamo izraz vrednosti X menjajuco  $Op = +, -, *, //$  i pamtimo listu tekucih rezultata  $Tr$  za  $I = 1..N$   
 $p1(N, X, [Op|R], Tr, I):-I \leq N, izborop(Op, Tr, I, Tr1), I1 is I+1, p1(N, X, R, Tr1, I1).$

Prilozeno resenje je opste i radi za bilo koje X (koje treba dobiti) i bilo koje brojeve od 1 do N (cijom „kombinacijom“ se proizvodi broj X)

**form1(N, X, LO):-p1(N, X, LO, 1, 2).**

**p1(N, X, [], X, I):-I > N.**

**p1(N, X, [Op|R], Tr, I):-I \leq N, izborop(Op, Tr, I, Tr1), I1 is I+1, p1(N, X, R, Tr1, I1).**

**izborop(+, A, B, C):-C is A+B.**

**izborop(-, A, B, C):-C is A-B.**

**izborop(\*, A, B, C):-C is A\*B.**

**izborop(/, A, B, C):-C is A/B.**

2. Odredi listu operacija  $\{+, -, *, /\}$  koje treba staviti između brojeva date liste L da bi dobili rezultat X, pri tome se operacije izvršavaju redom bez obzira na prioritet.

TEST PRIMER

$L=[4,2,7]$   $X=14$

$L0=[-, *]$  jer vrednost izraza  $4-2*7$  ako se operacije izvršavaju redom je  $(4-2)*7$  tj. 14

3. Odredi listu operacije LO (+/-) koje treba redom staviti između brojeva liste L da bi dobili rezultat S.

TEST PRIMER

1 ?-  $p([1,2,3,4],L,6)$ .

$L = [-, +, +]$

Jer  $1-2+3+4=6$

2 ?-  $p([1,2,3,4,5,6,7,8],L,20)$ .

$L = [+ , + , + , + , + , + , -]$

3 ?-  $p([1,2,3,4,5,6,7,8],L,-4)$ .

$L = [+ , + , + , - , + , - , -]$ .

4. Data je lista L brojeva i lista LO operacija  $\{+, -, *, /\}$ , lista LO ima jedan element manje od liste LO. Definisi predikat kojim se određuje vrednost izraza koji se dobija redom umetanjem operacija iz LO između brojeva u listi L. Pri tome se operacije izvršavaju redom i nema prioriteta.

TEST PRIMER

$L=[10,1,4]$   $LO=[+, *]$

rezultat je 44 (jer  $(10+1)*4=44$ )

5. Data je lista L brojeva i lista LO operacija  $\{+, -, *, /\}$ , lista LO ima jedan element manje od liste LO. Definisi predikat kojim se na osnovu datih lista L, LO i datog broja X formira nova lista NL brojeva od brojeva iz liste L tako da primenom operacija iz liste LO na listu NL dobijamo rezultat X (nema prioriteta operacije se izvršavaju redom).

$L=[1,4,10]$   $LO=[+, *]$   $X=44$  rezultat  $NL=[10,1,4]$ , jer kao u prethodnom test primeru  $(10+1)*4=44$

Dakle, u ovom zadatku je potrebno  
odrediti permutacije liste L,  
na svaku permutaciju primeniti listu operacija LO  
i proveriti da li je rezultat jednak datom X

Priprema pred kontrolni

6. Definisi predikat kojim se od liste L celih brojeva

a) formira lista NL od K članova (bez ponavljanja članova liste L) i lista operacija koje treba redom staviti između elemenata liste NL da bi dobili rezultat R (za dato R, K), pri tome operacije se izvršavaju redom (nema prioriteta)

b)) formira najkracu listu NL (bez ponavljanja clanova liste L ) i lista operacija koje treba redom staviti izmedju elemenata liste NL da bi dobili rezultat R (za dato R), pri tome operacijese izvrsavaju redom (nema prioriteta)

#### HINTOVI

a) formiramo varijacije bez ponavljanja klase K od elemenata liste L, a zatim pozivamo predikat form2 (zadatak 2 iz poglavlja Neke primene kombinatorike)

b) pokusavamo prvo da nadjemo listu duzine 1, zatim listu duzine 2, ..., listu duzine N

2. Kada su jednog oca upitali koliko godina ima svako od njegovih dvoje dece, otac je izjavio:  
Ako se proizvodu brojeva koji predstavljaju godine dece doda zbir njihovih godina, dobija se broj a.  
Napisati PROLOG program za odredjivanje broja godina dece ako pretpostavimo da se broj a ucitava sa tastature.

#### TEST PRIMER

23

#### IZLAZ

Prvo i drugo dete mogu imati

1 i 11 godina

2 i 7 godina

3 i 5 godina

5 i 3 godina

7 i 2 godina

11 i 1 godina