

Combinatorică și Teoria Grafurilor

Mircea Marin



Editura Universității de Vest
Timișoara, 2021

Cuprins

1	Combinatorică	1
1.1	Tehnici de numărare	2
1.1.1	Regula bijecției, regula produsului, și regula sumei	2
1.1.2	Tipuri de aranjamente combinatoriale	6
1.1.3	Coeficienți binomiali	13
1.1.4	Coeficienți multinomiali	15
1.1.5	Concluzii	17
1.1.6	Exerciții	18
1.2	Tehnici de generare și enumerare	22
1.2.1	Ordonarea lexicografică	22
1.2.2	Alte ordonări	23
1.2.3	Algoritmi de ranking, unranking și enumerare	24
1.2.4	Concluzii	33
1.2.5	Exerciții	33
1.3	Principiul cutiei	36
1.3.1	Aplicații remarcabile	37
1.3.2	Concluzii	41
1.3.3	Exerciții	41
1.4	Principiul incluziunii și excluziunii	43
1.4.1	O notație convenabilă	46
1.4.2	Aplicații remarcabile	50
1.4.3	Concluzii	52
1.4.4	Exerciții	52
1.5	Tehnici avansate de numărare	54
1.5.1	Relații de recurență	54
1.5.2	Concluzii	70
1.5.3	Exerciții	71
1.6	Teoria de enumerare a lui Pólya	74
1.6.1	Ansambluri și grupuri de simetrii	76

1.6.2	Cicluri. Reprezentarea ciclică a permutărilor . . .	77
1.6.3	Grupurile C_n și D_n	79
1.6.4	Simetrii ale unor ansambluri tridimensionale . . .	81
1.6.5	Colorări	83
1.6.6	Colorări cu număr prestabilit pentru fiecare culoare	87
1.6.7	Concluzii	92
1.6.8	Exerciții	93
1.7	Probleme de ocupare	95
1.7.1	Partiții	95
1.7.2	Numere Stirling de cicluri	98
1.7.3	Numere Stirling de mulțimi	102
1.7.4	Concluzii	103
1.7.5	Exerciții	104
2	Teoria Grafurilor	107
2.1	Aplicații ale teoriei grafurilor	108
2.2	Definiții și noțiuni	115
2.2.1	Noțiuni fundamentale	116
2.2.2	Grafuri izomorfe	118
2.2.3	Conectivitate	119
2.2.4	Proprietăți fundamentale	122
2.2.5	Clase de grafuri	122
2.2.6	Subgrafuri și subdiviziuni	125
2.2.7	Operații cu grafuri	125
2.2.8	Statistici descriptive	127
2.2.9	Concluzii	130
2.2.10	Exerciții	130
2.3	Reprezentarea grafurilor în calculator	135
2.3.1	Proprietăți ale acestor reprezentări	139
2.3.2	Un API Java de lucru cu grafuri	143
2.3.3	Concluzii	148
2.3.4	Exerciții	149
2.4	Traversarea grafurilor	152
2.4.1	Traversarea în adâncime de la un nod sursă (DFS)	153
2.4.2	Traversarea în lățime de la un nod sursă (BFS) . .	155
2.4.3	Ordini de traversare a nodurilor în adâncime . . .	156
2.4.4	Aplicații ale traversării în adâncime	158
2.4.5	Concluzii	170
2.4.6	Exerciții	171
2.5	Grafuri ponderate	178

2.5.1	Căi cu lungime ponderată minimă de la un nod sursă	181
2.5.2	Algoritmul lui Bellman-Ford	183
2.5.3	Algoritmul lui Dijkstra	185
2.5.4	Căi cu lungime ponderată minimă între toate nodurile	189
2.5.5	Algoritmul lui Floyd-Warshall	189
2.5.6	Concluzii	193
2.5.7	Exerciții	193
2.6	Arbori	195
2.6.1	Enumerarea arborilor de ordinul n	196
2.6.2	Arbori de acoperire	199
2.6.3	Algoritmul lui Prim	201
2.6.4	Algoritmul lui Kruskal	202
2.6.5	Exerciții	204
2.7	Grafuri planare	206
2.7.1	Teorema lui Euler	208
2.7.2	Teorema lui Kuratowski	209
2.7.3	Poliedre regulate	211
2.7.4	Concluzii	214
2.7.5	Exerciții	214
2.8	Grafuri hamiltoniene și euleriene	217
2.8.1	Drumuri și cicluri hamiltoniene	217
2.8.2	Criterii de detecție a grafurilor hamiltoniene	218
2.8.3	Drumuri și cicluri euleriene	222
2.8.4	Detecția grafurilor euleriene	222
2.8.5	Construcția ciclurilor euleriene	223
2.8.6	Concluzii	225
2.8.7	Exerciții	226
2.9	Colorări	229
2.9.1	Polinoame cromatice	230
2.9.2	Problema celor patru culori	233
2.9.3	Aplicații	235
2.9.4	Exerciții	237
2.10	Rețele de transport	239
2.10.1	Rețele de transport cu muchii antiparalele	240
2.10.2	Rețele de transport cu surse și destinații multiple	241
2.10.3	Metoda lui Ford-Fulkerson	243
2.10.4	Algoritmul Edmonds-Karp	254

2.10.5	Aplicații	254
2.10.6	Concluzii	267
2.10.7	Exerciții	268
2.11	Cuplaje	271
2.11.1	Cuplaje maxime	272
2.11.2	Cuplaje perfecte minime. Algoritmul ungar	274
2.11.3	Exerciții	279
A	Noțiuni de bază	281
A.1	Multiseturi și mulțimi	281
A.2	Relații	286
A.3	Funcții	289
	Bibliografie	293

1

Combinatorică

Combinatorica este o parte importantă a matematicii discrete care se ocupă cu studiul aranjamentelor de obiecte: împerecheri și grupări, clasamente și ordonări, selecții și alocări. Studiul ei a început în secolul XVII ca o activitate recreațională de estimare a șanselor de câștig la jocurile de noroc.

Sunt trei direcții principale de studiu al combinatoricii:

1. **Combinatorica enumerativă** este știința numărării și enumerării unor aranjamente de obiecte în prezența unor anumite constrângeri. Combinatorica enumerativă are aplicații în rezolvarea unor probleme precum: determinarea complexității algoritmilor, calculul probabilităților de apariție a unor evenimente, etc.

Se disting două tipuri de aranjamente: aranjamente ordonate (sau *permutări*) și aranjamente neordonate (sau *combinări*).

2. **Combinatorica existențială** cuprinde studiul problemelor de determinare a existenței unor aranjamente cu anumite proprietăți.
3. **Combinatorica constructivă** cuprinde studiul algoritmilor de creare a aranjamentelor cu anumite proprietăți.

Pentru aranjamente combinatoriale de un anumit fel, tehnicile de numărare oferă formule de calcul a numărului lor, iar cele de enumerare oferă metode de generare a lor în o anumită ordine. Metodele de enumerare joacă un rol important în simulările computerizate.

1.1 Tehnici de numărare

Teoria numărării caută răspuns la întrebarea “Cât de multe?” fără să enumere toate alternativele posibile. Exemple de întrebări de acest gen sunt:

- Câte numere diferite se pot reprezenta cu n biți?
- Câte parole distincte de 7 caractere se pot forma, dacă este permisă doar folosirea cifrelor zecimale și a literelor din mulțimea $\{a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z\}$?
- În câte feluri putem ordona n elemente distincte?
- Câte submulțimi cu cel mult 3 elemente are o mulțime cu 10 elemente distincte?

În această secțiune sunt prezentate:

1. Principiile de numărare ale combinatoricii, cu exemple de utilizare a acestora în raționamentul combinatorial.
2. Tipuri de aranjamente combinatoriale: permutări și combinații cu și fără repetiție.
3. Formule de calcul al numărului de aranjamente combinatoriale de diferite tipuri și exemple de probleme care se rezolvă cu aceste formule.

Anexa A conține cunoștințe de bază despre multiseturi, mulțimi, relații și funcții, care sunt necesare pentru a înțelege materialul din acest volum.

1.1.1 Regula bijecției, regula produsului, și regula sumei

Regula bijecției spune că dacă A, B sunt două mulțimi finite și există o funcție bijectivă de la A la B , atunci mulțimile A și B au același număr de elemente. **Regula produsului** spune că dacă o procedură poate fi descompusă în o secvență de 2 proceduri astfel încât prima se poate efectua în n_1 feluri iar a doua în n_2 feluri, atunci există $n_1 \cdot n_2$ feluri de a efectua acea procedură. Mai general, dacă o procedură p poate fi descompusă în o secvență de proceduri $p_1, \dots, p_m$ și fiecare p_i se poate efectua în n_i feluri, atunci p se poate efectua în $n_1 \cdot n_2 \cdot \dots \cdot n_m$ feluri. În particular, dacă $A_1, A_2, \dots, A_m$ sunt mulțimi finite, atunci

alegerea unui element $\langle a_1, a_2, \dots, a_m \rangle$ din $A_1 \times A_2 \times \dots \times A_m$ (sau a unui șir $a_1 a_2 \dots a_m$ cu $a_1 \in A_1, a_2 \in A_2, \dots, a_m \in A_m$) se descompune în o secvență de n operații: se alege a_1 din A_1 , apoi a_2 din A_2 , și așa mai departe până se alege a_m din A_m . Conform regulii produsului, $\langle a_1, a_2, \dots, a_m \rangle$ se poate alege în $|A_1| \cdot |A_2| \cdot \dots \cdot |A_m|$ feluri. De aici deducem că $|A_1 \times A_2 \times \dots \times A_m| = |A_1| \cdot |A_2| \cdot \dots \cdot |A_m|$.

Exemplul 1. Scaunele unei săli de seminar sunt etichetate cu o literă din alfabetul A de 26 litere al limbii engleze, urmată de un număr întreg pozitiv mai mic sau egal ca 100. Câte scaune pot fi etichetate diferit?

RĂSPUNS: O etichetă este de forma ℓn cu $\ell \in A$ și $n \in N$, unde N este mulțimea întregilor de la 1 la 100. Conform regulii produsului, sunt $|A| \cdot |N| = 26 \cdot 100$ etichete diferite. Deci răspunsul este 2600. $\square$

Exemplul 2. Câte șiruri diferite de 7 biți se pot forma?

RĂSPUNS: Un astfel de șir este de forma $b_1 b_2 b_3 b_4 b_5 b_6 b_7$ cu $b_i \in \{0, 1\}$ pentru $1 \leq i \leq 7$, Conform regulii produsului, sunt $|\{0, 1\}|^7 = 2^7 = 128$ astfel de șiruri. $\square$

Exemplul 3. O serie a unui bilet de tombolă este o secvență de trei litere din mulțimea $\{a, b, c, x, y, z\}$ urmată de o secvență de trei cifre zecimale. Câte bilete de tombolă pot fi marcate cu serii diferite de acest fel?

RĂSPUNS: O serie de bilet este un șir de forma $\ell_1 \ell_2 \ell_3 d_1 d_2 d_3$ cu $\ell_1, \ell_2, \ell_3 \in A$ și $d_1, d_2, d_3 \in D$, unde $A = \{a, b, c, x, y, z\}$ și $D = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$. Conform regulii produsului, sunt $|A|^3 \cdot |D|^3 = 6^3 \cdot 10^3 = 216000$ astfel de serii. Deci numărul căutat este 216000. $\square$

Exemplul 4. Câte funcții $f : A \rightarrow B$ există dacă A și B sunt finite?

RĂSPUNS: Deoarece A, B sunt mulțimi finite, putem presupune că $A = \{a_1, a_2, \dots, a_m\}$ unde $m = |A|$. Pentru a defini $f : A \rightarrow B$ trebuie să alegem valoarea lui $f(a_1)$, apoi pe a lui $f(a_2)$, până la valoarea lui $f(a_m)$. Altfel spus, trebuie ales un șir $b_1 b_2 \dots b_m \in B^m$ care pentru fiecare $1 \leq i \leq m$ ne indică faptul că $f(a_i) = b_i$. De exemplu, dacă $A = \{a_1 = a, a_2 = b, a_3 = c\}$ și $B = \{1, 2, 3, 4\}$, atunci funcția

$$f(a) = f(a_1) = 3, \quad f(b) = f(a_2) = 1, \quad f(c) = f(a_3) = 4$$

este reprezentată de șirul $b_1 b_2 b_3 = 314$.

Conform regulii produsului, sunt $|B|^m = |B|^{|A|}$ șiruri diferite de acest fel. Deci există $|B|^{|A|}$ funcții $f : A \rightarrow B$. $\square$

Exemplul 5. Presupunem că A și B sunt mulțimi finite și $|A| \leq |B|$. Câte funcții injective $f : A \rightarrow B$ există?

RĂSPUNS: Fie $|A| = m$, $|B| = n$ și $A = \{a_1, \dots, a_m\}$. Pentru a defini o funcție injectivă $f : A \rightarrow B$ trebuie să alegem

$$\begin{aligned} f(a_1) &\in B_1 = B, \\ f(a_2) &\in B_2 = B - \{f(a_1)\}, \\ &\vdots \\ f(a_m) &\in B_m = B - \{f(a_1), f(a_2), \dots, f(a_{m-1})\} \end{aligned}$$

Conform regulii produsului, sunt $|B_1| \cdot |B_2| \cdot \dots \cdot |B_m|$ posibilități de a defini f . Se observă că $|B_i| = n - i + 1$ pentru $1 \leq i \leq m$. Deci numărul de funcții injective $f : A \rightarrow B$ este $n(n-1) \dots (n-m+1)$. $\square$

Exemplul 6. Ce valoare va avea k după ce se execută fragmentul de program de mai jos?

```
k := 0;
for i1 := 1 to n1
  for i2 := 1 to n2
    ⋮
    for im := 1 to nm
      k := k + 1
```

RĂSPUNS: Acest fragment de program este format din m bucle **for** imbricate. Fiecare incrementare a lui k se produce pentru o valoare diferită a tuplului de valori $(i_1, i_2, \dots, i_m)$. Numărul de incrementări ale lui k coincide cu numărul de tuple $(i_1, i_2, \dots, i_m)$ cu $i_1 \in \{1, 2, \dots, n_1\}$, $i_2 \in \{1, 2, \dots, n_2\}$, $\dots$, $i_m \in \{1, 2, \dots, n_m\}$, adică cu $n_1 n_2 \dots n_m$. Valoarea inițială a lui k este 0, deci valoarea lui finală va fi $n_1 n_2 \dots n_m$.

Exemplul 7. Câte submulțimi are o mulțime finită S ?

RĂSPUNS: Fie $S = \{a_1, a_2, \dots, a_n\}$ unde $n = |S|$ și B_n este mulțimea șirurilor de n biți. Definim funcția bijectivă $f : 2^S \rightarrow B_n$,

$$f(A) = b_1 b_2 \dots b_n \text{ unde } b_i = \begin{cases} 1 & \text{dacă } a_i \in A, \\ 0 & \text{dacă } a_i \notin A. \end{cases}$$

Conform regulii bijecției, $|2^S| = |B_n|$. Conform regulii produsului, $|B_n| = \underbrace{2 \cdot 2 \cdot \dots \cdot 2}_n = 2^n$. Deci S are $2^n = 2^{|S|}$ submulțimi.

Regula sumei spune că dacă o procedură se poate efectua în 2 feluri, pentru felul i sunt n_i variante, și nici una din variantele de primul fel nu coincide cu vreo variantă de felul 2, atunci există $n_1 + n_2$ variante de a efectua procedura. Mai general, dacă o procedură poate fi efectuată în m feluri, pentru felul i sunt n_i variante, și variantele efectuate în feluri diferite sunt diferite, atunci există $n_1 + n_2 + \dots + n_m$ variante de a efectua procedura respectivă.

Exemplul 8. Un student își poate alege un proiect din una din trei liste: prima listă conține 21 proiecte, a doua 12 proiecte, iar a treia 18 proiecte. Se presupune că cele trei liste nu au proiecte în comun. Câte posibilități de a alege un proiect are studentul respectiv?

RĂSPUNS: Conform regulii sumei, sunt $21 + 12 + 18 = 51$ posibilități.

Exemplul 9. Ce valoare are k după ce se execută fragmentul de program de mai jos?

```
k := 0;
for i1 := 1 to n1
    k := k + 1
for i2 := 1 to n2
    k := k + 1
    :
for im := 1 to nm
    k := k + 1
```

RĂSPUNS: Fragmentul de program este o secvență de m bucle **for**. Pentru fiecare $j \in \{1, 2, \dots, m\}$, fie T_j mulțimea de execuții a buclei pentru variabila i_j . Se observă că $|T_j| = n_j$ pentru toți $j \in \{1, 2, \dots, m\}$ și că fiecare execuție a unei bucle incrementează valoarea lui k . Deci numărul de incrementări a variabilei k coincide cu numărul N de posibilități de a executa o buclă. Conform regulii sumei

$$N = |T_1| + |T_2| + \dots + |T_m| = n_1 + n_2 + \dots + n_m.$$

k are valoarea inițială 0 și este incrementat de N ori, deci valoarea lui k va fi $N = n_1 + n_2 + \dots + n_m$. $\square$

Unele probleme de numărare se pot rezolva combinând regula sumei cu regula produsului.

Exemplul 10. Un elev are de citit două cărți în limbi diferite și le poate alege dintr-o colecție de 10 cărți în engleză, 11 cărți în franceză și

12 cărți în germană. Câte posibilități de a-și îndeplini sarcina are elevul respectiv?

RĂSPUNS: Sarcina elevului este un șir de 2 operații: (1) să aleagă două limbi diferite din mulțimea $L = \{E, F, G\}$ unde E, F, G reprezintă limbile engleză, franceză și germană, și (2) să aleagă câte o carte pe care s-o citească în fiecare limbă aleasă. Pentru prima operație are 3 posibilități: $\{E, F\}$, $\{E, G\}$ și $\{F, G\}$. Conform regulii sumei, numărul de posibilități de a-și îndeplini sarcina este $N_{\{E, F\}} + N_{\{E, G\}} + N_{\{F, G\}}$ unde $N_{\{X, Y\}}$ este numărul de posibilități de a citi o carte în limba X și altă carte în limba Y . Conform regulii produsului $N_{\{X, Y\}} = N_X \cdot N_Y$ unde N_X este numărul de cărți în limba X și N_Y este numărul de cărți în limba Y . Rezultă că $N_{\{E, F\}} = 10 \cdot 11$, $N_{\{E, G\}} = 10 \cdot 12$ și $N_{\{F, G\}} = 11 \cdot 12$. Deci elevul are $110 + 120 + 132 = 362$ posibilități să-și îndeplinească sarcina.

Exemplul 11. Parolele de acces la un calculator sunt șiruri de 6, 7 sau 8 caractere care pot fi litere din alfabetul englez A de 26 de litere, sau cifre din mulțimea $D = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$. Fiecare parolă trebuie să conțină cel puțin o cifră. Câte parole diferite de acest fel se pot forma?

RĂSPUNS: Fie P_m mulțimea parolelor de lungime $m \in \{6, 7, 8\}$. Conform regulii sumei, sunt posibile $|P_6| + |P_7| + |P_8|$ parole. Fie S_m mulțimea șirurilor de m caractere din $A \cup D$. Conform regulii produsului, $|S_m| = |A \cup D|^m = (|A| + |D|)^m = 36^m$. Deasemenea, $|S_m| = |P_m| + |S_m - P_m|$ fiindcă $S_m = P_m \cup (S_m - P_m)$ și $P_m \cap (S_m - P_m) = \emptyset$. Rezultă că $|P_m| = |S_m| - |S_m - P_m| = 36^m - |S_m - P_m|$.

$S_m - P_m$ conține șirurile de m caractere care nu sunt parole. Acestea sunt formate doar din caractere din A . Conform regulii produsului, $|S_m - P_m| = 26^m$. Deci $P_m = 36^m - 26^m$ iar numărul căutat este $|P_6| + |P_7| + |P_8| = (36^6 + 36^7 + 36^8) - (26^6 + 26^7 + 26^8)$.

1.1.2 Tipuri de aranjamente combinatoriale

În această secțiune presupunem că S este o mulțime finită cu n elemente și vom considera următoarele tipuri de aranjamente pentru $r \leq n$:

- un **r -aranjament ordonat** (sau **r -permutare**) al elementelor lui S este un r -tuplu $\pi = \langle p_1, p_2, \dots, p_r \rangle$ cu $p_i \in S$ și $p_i \neq p_j$ pentru orice $i, j \in \{1, 2, \dots, r\}$, $i \neq j$. De exemplu, $\langle 2, 1, 4 \rangle$ și $\langle 4, 2, 1 \rangle$ sunt 3-permutări ale mulțimii $\{1, 2, 3, 4\}$.

O **permutare** este o n -permutare. De exemplu, $\langle a, c, b \rangle$ și $\langle c, a, b \rangle$ sunt permutări ale mulțimii $\{a, b, c\}$.

- un **r -aranjament ordonat cu repetiție** (sau **r -permutare cu repetiție**) al elementelor lui S este un r -tuplu $\langle s_1, s_2, \dots, s_r \rangle \in S^r$.
- un **aranjament neordonat** (sau **combinare**) al lui S este o submulțime a lui S . De pildă, combinările lui $\{1, 2\}$ sunt $\emptyset, \{1\}, \{2\}, \{1, 2\}$.
- Dacă $r \leq n$, un **r -aranjament neordonat** (sau **r -combinare**) a lui S este o submulțime cu r elemente a lui S . De exemplu, 2-combinările lui $\{a, b, c\}$ sunt $\{a, b\}, \{a, c\}, \{b, c\}$.
- un **r -aranjament neordonat cu repetiție** (sau **r -combinare cu repetiție**) al elementelor lui S este un multiset $\{s_1, s_2, \dots, s_r\}$ cu $s_i \in S$ pentru toți $i \in \{1, 2, \dots, r\}$.

Reamintim faptul că un **multiset** este o structură de date asemănătoare cu o mulțime, însă în care fiecare element poate să apară de mai multe ori. Numărul de apariții al unui element într-un multiset se numește *multiplicitatea* elementului respectiv. De exemplu, $\{1, 1, 2, 4\}$ este un multiset în care elementul 1 are multiplicitatea 2, iar elementele 2 și 4 au multiplicitatea 1. Două multiseturile sunt **egale** dacă conțin aceleași elemente, iar elementele au aceeași multiplicitate. De exemplu, $\{1, 2, 2, 4\} = \{2, 1, 4, 2\}$, dar $\{1, 2, 4\} \neq \{1, 2, 2, 4\}$.

În subsecțiunile următoare vom determina formule de calcul al numărului de aranjamente de fiecare tip. Vom folosi notația $n!$ pentru produsul $1 \cdot 2 \cdot \dots \cdot n$ atunci când $n \in \mathbb{N}$, $n \geq 1$, și vom presupune că $0! = 1$. Valoarea lui $n!$ pentru $n \in \mathbb{N}$ se numește **factorialul** lui n .

Permutări și r -permutări

Orice r -permutare $\langle p_1, p_2, \dots, p_r \rangle$ a elementelor unei mulțimi finite S cu n elemente corespunde unei funcții injective $\pi : \{1, 2, \dots, r\} \rightarrow S$ cu $\pi(i) = p_i$. Am demonstrat în Exemplul 5 de la pagina 4 că numărul de astfel de funcții π este $n(n-1) \cdots (n-r+1)$. Deci

Numărul de r -permutări al unei mulțimi cu n elemente este

$$P(n, r) = n(n-1) \cdots (n-r+1) = \frac{n!}{(n-r)!} \quad (1.1)$$

Orice permutare a lui S este o n -permutare, deci

Numărul de permutări al unei mulțimi cu n elemente este

$$P(n, n) = n(n-1) \cdots 1 = n!$$

Exemplul 12. Se consideră alfabetul A de 26 litere latine mici de la a la z . Câte șiruri diferite de trei litere din A se pot forma?

RĂSPUNS: $P(26, 3) = 26!/(26-3)! = 26 \cdot 25 \cdot 24 = 15600$. $\square$

Exemplul 13. În câte feluri pot fi puși într-un rând n bărbați și n femei astfel încât să nu fie două femei una lângă alta?

RĂSPUNS. Rândul este fie de forma $b_1 f_1 b_2 f_2 \dots b_n f_n$, sau de forma $f_1 b_1 f_2 b_2 \dots f_n b_n$, unde $B = \{b_1, b_2, \dots, b_n\}$ este mulțimea celor n bărbați, iar $F = \{f_1, f_2, \dots, f_n\}$ este mulțimea celor n femei. Conform regulii produsului, sunt $(n!)^2$ posibilități pentru fiecare din cele 2 cazuri. Conform regulii sumei, numărul căutat este $(n!)^2 + (n!)^2 = 2(n!)^2$. $\square$

Combinări și r -combinări

Reamintim faptul că o combinare a unei mulțimi S este o submulțime a lui S , iar o r -combinare a lui S este o submulțime cu r elemente a lui S . Am demonstrat în Exemplul 7 de la pagina 4 că

O mulțime cu n elemente are 2^n submulțimi.

Vom scrie $C(n, r)$ sau $\binom{n}{r}$ pentru a ne referi la numărul de r -combinări al unei mulțimi cu n elemente.

$$P(n, r) = C(n, r) \cdot P(r, r). \quad (1.2)$$

DEMONSTRAȚIE: Fie S o mulțime cu n elemente. Orice r -permutare π a lui S este determinată în mod unic de un șir $o_1 o_2$ de două operații:

o_1 : se alege o submulțime C cu r elemente a lui S . S are n elemente, deci sunt $C(n, r)$ posibilități de a efectua operația o_1 .

o_2 : π se obține alegând o r -permutare a mulțimii C . C are r elemente, deci sunt $P(r, r)$ posibilități de a efectua operația o_2 .

Conform regulii produsului, numărul de r -permutări π ale elementelor lui S coincide cu $C(n, r) \cdot P(r, r)$, adică, $P(n, r) = C(n, r) \cdot P(r, r)$. $\square$

O consecință imediată a acestui rezultat este că