Combinatorică cu aplicaţii

Colecţia
INFORMATICĂ TEORETICĂ

Zoltán Kása

Combinatorică cu aplicaţii

Presa Universitară Clujeană
Cluj-Napoca, 2003

Universitatea Babeş-Bolyai Cluj-Napoca
Facultatea de Matematică şi Informatică

Universitatea Babeş-Bolyai
Presa Universitară Clujeană
Director: Horia Cosma
Str. Republicii, nr. 24 Cluj-Napoca, Romania
Tel.: +40 264 597401 Fax: +40 264 591906
E-mail: presa_universitara@email.ro

ISBN 973-610-152-5

Cuprins

Notaţii utilizate

A^∞	mulţimea cuvintelor finite sau infinite peste alfabetul A
A^n	mulţimea cuvintelor finite de lungime n peste alfabetul A
A^ω	mulţimea cuvintelor infinite peste alfabetul A
A^+	mulţimea cuvintelor finite nevide peste alfabetul A
A^*	mulţimea cuvintelor finite peste alfabetul A
a/b	a se înlocuieşte cu b (la translatoare)
$\alpha \to \beta$	producţie (la gramatici)
$\alpha \Longrightarrow \beta$	derivare directă (la gramatici)
$\alpha \overset{*}{\Longrightarrow} \beta$	derivare (la gramatici)
$\|u\|$	lungimea cuvântului u
$\binom{n}{m}$	combinări de n luate câte m
$\left\langle {n \atop m} \right\rangle$	combinări cu repetiţii de n luate câte m
$C(u)$	complexitatea maximă a cuvântului finit u
C_n	numărul lui Catalan
δ_{ij}	simbolul lui Kronecker (egal cu 1 când $i = j$ şi 0 altfel)
$\#A$	cardinalul mulţimii A
F_n	numărul lui Fibonacci
$F_n(u)$	mulţimea factorilor de lungime n ai cuvântului u
$f_u(n)$	numărul factorilor de lungime n ai cuvântului u
$G(n)$	complexitatea maximă globală în mulţimea cuvintelor de lungime n
$K(u)$	complexitatea totală a cuvântului u
$K_d(u)$	d-complexitatea totală a cuvântului u
$L(A)$	limbajul acceptat de automatul A
$L(G)$	limbajul generat de gramatica G
λ	cuvântul vid
$N(k, d)$	d-complexitatea totală a unui cuvânt de lungime k cu litere distincte
$\mathcal{P}(A)$	mulţimea părţilor lui A (mulţimea submulţimilor lui A)

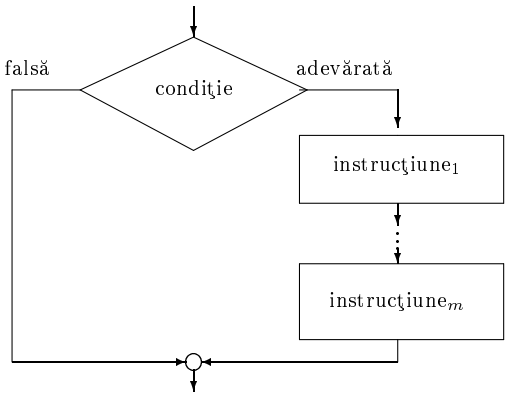
Lista aplicaţiilor

Introducere

În descrierea algoritmilor prezentaţi vom folosi o variantă de pseudocod care se citeşte uşor şi nu foloseşte marcaje care să îngreuneze înţelegerea. Pentru atribuirea vom folosi semnul uzual :=. Pentru marcarea corpului unei intructţiuni vom folosi indentarea. Toate instructţiunile cu aceeaşi indentare aparţin corpului instructţiunii la care se referă. Vom descrie pe scurt forma instructţiunilor folosite împreună cu semantica lor, dată sub forma unor diagrame uşor de înţeles.

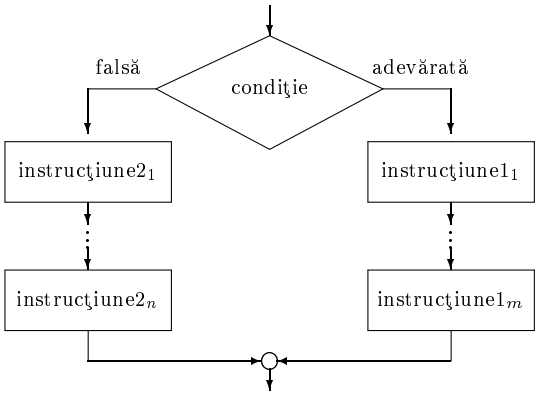
Instructţiunea **dacă**

dacă condiţie **atunci**
    instructţiune$_1$
    instructţiune$_2$
    ...
    instructţiune$_m$



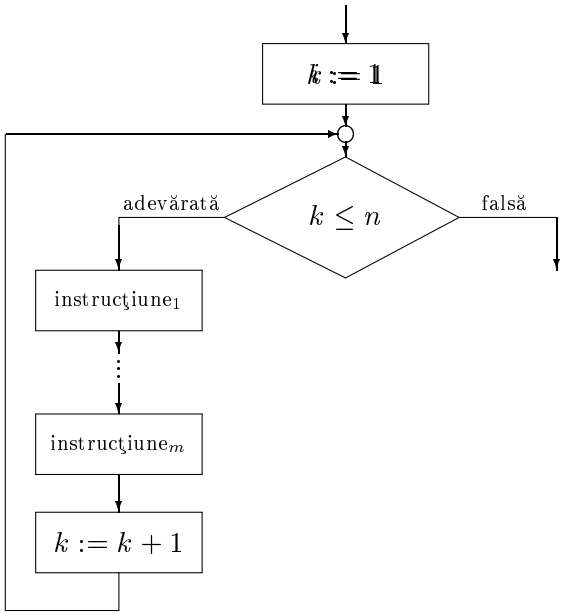
sau

dacă condiţie **atunci**
    instructţiune1$_1$
    instructţiune1$_2$
    ...
    instructţiune1$_m$
altfel
    instructţiune2$_1$
    instructţiune2$_2$
    ...
    instructţiune2$_n$



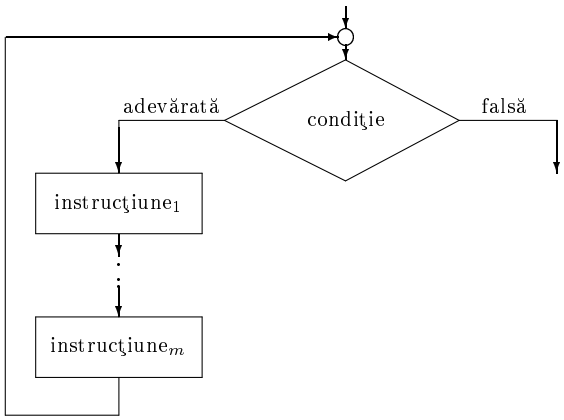
Instrucţiunea **pentru**

pentru $k = 1, 2, \ldots, n$ **execută**
    instrucţiune$_1$
    instrucţiune$_2$
    ...
    instrucţiune$_m$



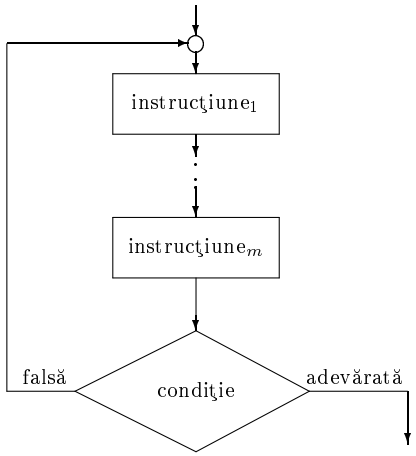
Instrucţiunea **cât timp**

cât timp condiţie **execută**
    instrucţiune$_1$
    instrucţiune$_2$
    ...
    instrucţiune$_m$



Instrucţiunea **repetă**

repetă
    instrucţiune$_1$
    instrucţiune$_2$
    ...
    instrucţiune$_m$
până când condiţie



La fiecare instrucţiune, dacă corpul ei conţine o singură instrucţiune, atunci aceasta din urmă se poate scrie pe aceeaşi linie cu descrierea instrucţiunii respective.
De exemplu, în loc de

pentru $i = 1, 2, \ldots, n$ **execută**
    $v_i := i$

se poate scrie mai simplu

pentru $i = 1, 2, \ldots, n$ **execută** $v_i := i$

Pentru descrierea procedurilor vom folosi structura

procedura nume (parametri)
    instrucţiune$_1$
    instrucţiune$_2$
    ...
    instrucţiune$_m$
sfârşit procedură

Algoritmii prezentaţi în carte au fost programaţi în Turbo Pascal şi pot fi obţinuţi de la adresa

`http://www.cs.ubbcluj.ro/~kasa/combinatorica.html`

Tot la aceeaşi adresă se va păstra o erată actualizată cu greşelile cunoscute. Observaţiile şi sugestiile pot fi trimise la adresa autorului:

`kasa@cs.ubbcluj.ro`

1 Permutări, aranjamente, combinări

1.1 Formula binomului

Pornind de la formulele cunoscute: $(a+b)^2 = a^2 + 2ab + b^2$ şi $(a+b)^3 = = a^3 + 3a^2b + 3ab^2 + b^3$ încercăm să găsim o formulă generală pentru $(a+b)^n$. Prin înmulţiri succesive cu $(a+b)$ se obţin formulele:

$$(a+b)^4 = a^4 + 4a^3b + 6a^2b^2 + 4ab^3 + b^4$$

$$(a+b)^5 = a^5 + 5a^4b + 10a^3b^2 + 10a^2b^3 + 5ab^4 + b^5.$$

Se poate observa (ceea ce a făcut-o prima dată Blaise Pascal) că în dezvoltarea de mai sus coeficienţii se pot obţine uşor din coeficienţii din dezvoltarea anterioară. Adică, aceşti coeficienţi pot fi aranjaţi într-un triunghi, numit triunghiul lui Pascal, în felul următor:

$$\begin{array}{ccccccccccccc}
 & & & & & & 1 & & & & & & \\
 & & & & & 1 & & 1 & & & & & \\
 & & & & 1 & & 2 & & 1 & & & & \\
 & & & 1 & & 3 & & 3 & & 1 & & & \\
 & & 1 & & 4 & & 6 & & 4 & & 1 & & \\
 & 1 & & 5 & & 10 & & 10 & & 5 & & 1 & \\
1 & & 6 & & 15 & & 20 & & 15 & & 6 & & 1 \\
1 & & 7 & & 21 & & 35 & & 35 & & 21 & & 7 & & 1 \\
\multicolumn{13}{c}{\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots}
\end{array}$$

În acest triunghi fiecare număr se obţine prin adunarea celor două numere care se găsesc în linia precedentă în stânga şi dreapta lui. Putem enunţa următoarea teoremă.

Teorema 1. *Pentru orice număr natural $n \geq 1$ avem formula*

$$(a+b)^n = \sum_{k=0}^{n} \binom{n}{k} a^{n-k} b^k \tag{1}$$

unde coeficienţii $\binom{n}{k}$ satisfac relaţiile:

$$\binom{n}{0} = \binom{n}{n} = 1; \quad \binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}, \quad 0 < k \leq n \tag{2}$$

Demonstraţie. Demonstraţia formulei se face prin inducţie matematică completă. Pentru $n = 1, 2$ formula se verifică uşor. Să presupunem adevărată pentru $n-1$:

$$(a+b)^{n-1} = \sum_{k=0}^{n-1} \binom{n-1}{k} a^{n-1-k} b^k.$$

Pentru a obţine $(a+b)^n$ să înmulţim $(a+b)^{n-1}$ cu $(a+b)$. Termenul care conţine $a^{n-k}b^k$ în dezvoltarea lui $(a+b)^n$ se obţine înmulţind termenul în $a^{n-1-k}b^k$ din dezvoltarea lui $(a+b)^{n-1}$ cu a, sau înmulţind termenul în $a^{n-1}b^{k-1}$ cu b, deci

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}, \qquad \text{iar} \qquad \binom{n-1}{0} = \binom{n}{0}.$$

□

Coeficienţii $\binom{n}{k}$ se numesc *coeficienţi binomiali.*

Pentru calcularea acestor coeficienţi să vedem ce se întâmplă dacă efectuăm următorul produs:

$$(a_1 + b_1)(a_2 + b_2) \dots (a_n + b_n).$$

Un termen oarecare din acest produs conţine k a_i-uri (i oarecare) şi $n-k$ b_i-uri (i oarecare), deci are forma

$$a_{i_1} a_{i_2} \dots a_{i_k} b_{j_1} b_{j_2} \dots b_{j_{n-k}}.$$

Câţi termeni cu k a_i-uri şi $n-k$ b_i-uri există? a_{i_1} poate fi ales din toţi cei n factori, a_{i_2} din $n-1$ factori rămaşi, etc. Deci în total sunt $n(n-1)\dots(n-k+1)$ termeni de acest fel. Dacă înlocuim fiecare a_i (i=1,2,...,n) cu a, vom obţine mulţi termeni identici. Câţi? a_{i_1} poate fi pe oricare din cei k poziţii, a_{i_2} pe cei $k-1$ poziţii rămase, etc. Deci din $k(k-1)\dots 2\cdot 1$ termeni cu k a_i-uri vom obţine un termen cu k de a-uri. Deci

$$\binom{n}{k} = \frac{n(n-1)\dots(n-k+1)}{1\cdot 2\dots(k-1)k}.$$

Dacă folosim notaţia:

$$n\cdot(n-1)\cdot(n-2)\cdots 3\cdot 2\cdot 1 = n! \text{ (} n \text{ factorial)} \quad \text{şi} \quad 0! = 1$$

se poate scrie

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} \tag{3}$$

ceea ce se poate demonstra uşor şi prin inducţie matematică. Se pot verifica direct şi următoarele formule:

$$\binom{n}{k} = \binom{n}{n-k} \tag{4}$$

$$\binom{n}{k} = \frac{n-k+1}{k}\binom{n}{k-1} \qquad \text{şi} \qquad \binom{n}{k} = \frac{n}{k}\binom{n-1}{k-1} \tag{5}$$

În general, *coeficientul binomial* $\binom{n}{k}$ ne arată în câte moduri distincte posibile pot fi scrise unul după altul $n-k$ de a-uri şi k de b-uri.

1.2 Permutări

Coeficientul binomial $\binom{n}{k}$ ne arată în câte moduri posibile pot fi înşirate $n-k$ de a-uri şi k de b-uri. Care este numărul de moduri posibile dacă considerăm elemente distincte? Problema este: în câte moduri distincte pot fi înşirate fără repetare n elemente? Primul element poate fi ales din n elemente, cel de al doilea din $n-1$ elemente şi aşa mai departe. Deci numărul căutat este

$$P_n = n \cdot (n-1) \cdots 2 \cdot 1 = n!$$

Spunem că *permutăm* cele n elemente, iar P_n se numeşte *numărul permutărilor* a n elemente. Pentru calculul lui $n!$ când n este mare se poate utiliza o formulă aproximativă dată de *J. Stirling* în 1730:

$$n! \approx \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$$

unde e reprezintă baza logaritmilor naturali. În tabelul următor se pot găsi valorile calculate cu formula lui Stirling şi comparate cu cele exacte pentru n până la valoarea 10.

n	$n!$	$\sqrt{2\pi n}\left(\frac{n}{e}\right)^n$	$\frac{n!-\sqrt{2\pi n}\left(\frac{n}{e}\right)^n}{n!}$
1	1	0.92	0.07786299
2	2	1.92	0.04049782
3	6	5.84	0.02729840
4	24	23.51	0.02057604
5	120	118.02	0.01650693
6	720	710.08	0.01378030
7	5040	4980.40	0.01182622
8	40320	39902.40	0.01035726
9	362880	359536.87	0.00921276
10	3628800	3598695.62	0.00829596

Euler a găsit o formulă interesantă pentru $n!$:

$$n! = \lim_{m\to\infty} \frac{m^n m!}{(n+1)(n+2)\dots(n+m)}$$

ceea ce permite generalizarea factorialului pentru orice x real. Se notează pentru orice x real:

$$\Gamma(x) = \lim_{m\to\infty} \frac{m^x m!}{x(x+1)(x+2)\dots(x+m)}$$

şi se defineşte $x!$ pentru un x real oarecare ca: $x! = \Gamma(x+1) = x\Gamma(x)$

Ce se întâmplă însă dacă permitem ca unele elemente să se repete de un număr de ori? Să considerăm de exemplu elementele 1,1,2,3, şi să găsim toate permutările posibile ale acestora (adică permitem ca elementul 1 să se repete de două ori în fiecare permutare). Să luăm odată numai elementele 1,1,2. Găsim următoarele permutări

112 121 211

Elementul 3 poate fi introdus între oricare două elemente în fiecare permutare, deci:

1123	1213	2113
1132	1231	2131
1312	1321	2311
3112	3121	3211

Notăm cu $P_{i_1,i_2,\ldots i_k}$, unde $i_1+i_2+\cdots+i_k=n$, numărul permutărilor cu repetiţii a n elemente, în care numai k elemente sunt distincte, celelalte se repetă: primul element se poate repeta de i_1 ori, elementul al doilea de i_2 ori etc. Din exemplele de mai sus rezultă că $P_{2,1}=3$, $P_{2,1,1}=12$.

Pentru a calcula valoarea lui $P_{i_1,i_2,\ldots i_k}$, la început să facem distincţie între elementele care se repetă. De exemplu să le scriem cu culori diferite. În acest caz ajungem la permutările obişnuite, deci putem forma $n!$ grupe. În fiecare grupă elementul 1 apare de i_1 ori, dar cu diferite culori, la fel şi celelalte elemente. Deci sunt $i_1!$ grupe care se deosebesc doar prin faptul că elementul 1 apare cu diferite culori. La fel şi pentru celelalte elemente. Deci dacă vrem să obţinem numărul grupelor, care apar dacă eliminăm colorarea elementelor, ajungem la formula:

$$P_{i_1,i_2,\ldots,i_k}=\frac{n!}{i_1!i_2!\ldots i_k!}, \qquad \text{unde } i_1+i_2+\cdots i_k=n,$$

ceea ce reprezintă formula de calcul pentru *permutările cu repetiţii*. Se mai foloseşte şi notaţia:

$$P_{i_1,i_2,\ldots,i_k}=\binom{n}{i_1,i_2,\ldots,i_k}.$$

Se poate observa că

$$\binom{n}{k}=\frac{n!}{k!(n-k)!}=P_{k,n-k}=\binom{n}{k,n-k}$$

reprezintă numărul permutărilor cu repetiţii a n elemente, din care două sunt distincte, primul element se repetă de k ori, iar cel de al doilea de $n-k$ ori.

1.3 Formula multinomului

Formula binomului poate fi generalizată uşor. Să calculăm coeficienţii dezvoltării lui $(a_1+a_2+\cdots a_k)^n$. Printr-o analogie cu formula binomului se obţine:

$$(a_1+a_2+\cdots a_k)^n=\sum_{n_1+n_2+\cdots n_k=n}\frac{n!}{n_1!n_2!\ldots n_k!}a_1^{n_1}a_2^{n_2}\ldots a_k^{n_k} \tag{6}$$

unde evident $n_1,n_2,\ldots n_k$ sunt numere naturale. Pentru $k=2$ obţinem formula binomului.

Coeficienţii $P_{n_1,n_2,\ldots,n_k} = \binom{n}{n_1, n_2, \ldots, n_k} = \dfrac{n!}{n_1! n_2! \ldots n_k!}$ se numesc *coeficienţi multinomiali*.

Sunt adevărate următoarele formule (date ca probleme la sfirşitul capitolului):

$$\sum_{n_1+\ldots+n_k=n} \binom{n}{n_1, n_2, \ldots, n_k} = k^n, \quad n_1, n_2, \ldots, n_k \in \mathbf{N} \tag{7}$$

$$\binom{n}{n_1, n_2, \ldots, n_k} = \sum_{n_1+\ldots+n_k=n-1} \binom{n-1}{n_1, n_2, \ldots, n_{i-1}, n_i - 1, n_{i+1}, \ldots, n_k}, \tag{8}$$

unde $n_1, n_2, \ldots, n_k \in \mathbf{N}^*$ (adică numere naturale nenule).

Numărarea arborilor [17]. Câţi arbori distincţi cu vârfurile şi gradele date există? Să fie vârfurile date $x_1, x_2, \ldots, x_n$ astfel încât $\delta(x_1) = d_1, \delta(x_2) = d_2, \ldots, \delta(x_n) = d_n$, unde $\delta(x)$ reprezintă gradul vârfului x. Evident gradele d_i sunt numere naturale nenule a căror sumă este $2(n-1)$ (Arborele are $n-1$ muchii şi fiecare se numără de două ori).

Să notăm cu $T(n; d_1, d_2, \ldots, d_n)$ numărul arborilor cu vârfurile $x_1, x_2, \ldots, x_n$, cu proprietatea că $\delta(x_1) = d_1, \delta(x_2) = d_2, \ldots, \delta(x_n) = d_n$. Evident $1 \leq d_i \leq n-1$ $(i = 1, 2, \ldots, n)$ şi $\sum_{i=1}^{n} d_i = 2(n-1)$. Atunci

$$T(n; d_1, d_2, \ldots, d_n) = \binom{n-2}{d_1 - 1, d_2 - 1, \ldots, d_n - 1}. \tag{9}$$

Pentru a demonstra formula de mai sus, să observăm în primul rând că

$$\sum_{i=1}^{n} (d_i - 1) = \sum_{i=1}^{n} d_i - n = 2(n-1) - n = n - 2.$$

În continuare vom presupune că $d_1 \geq d_2 \geq \ldots \geq d_{n-1} \geq d_n = 1$ (Fiecare arbore are cel puţin două vârfuri de grad 1.)

Este adevărată formula

$$T(n; d_1, d_2, \ldots, d_n) = \sum_{i=1}^{n-1} T(n-1; d_1, d_2, \ldots, d_{i-1}, d_i - 1, d_{i+1}, \ldots, d_{n-1}) \tag{10}$$

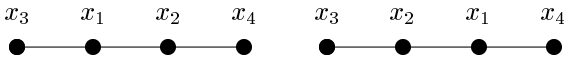
unde $d_1 \geq 2, \ldots, d_{n-1} \geq 2$. Acest lucru se justifică deoarece $T(n-1; d_1, d_2, \ldots, d_i - 1, \ldots, d_{n-1})$ reprezintă arborii care se obţin din arborii iniţiali prin eliminarea vârfului x_i care este legat printr-o singură muchie cu vârful x_n.

După aceste pregătiri, demonstrăm formula (9) prin inducţie completă. Pentru $n = 1$ formula este trivială, iar pentru $n = 2$ este evidentă. Să presupunem adevărată

pentru $n-1$ ($n \geq 3$) şi să demonstrăm pentru n.

$$\begin{aligned} T(n; d_1, d_2, \ldots, d_n) &= \sum_{i=1}^{n-1} T(n-1; d_1, d_2, \ldots, d_{i-1}, d_i - 1, d_{i+1}, \ldots, d_{n-1}) \\ &= \sum_{i=1}^{n-1} \binom{n-3}{d_1 - 1, d_2 - 1, \ldots, d_i - 2, \ldots, d_{n-1} - 1} \quad \text{(ipot. ind.)} \\ &= \binom{n-2}{d_1 - 1, d_2 - 1, \ldots, d_{n-1} - 1} \quad \text{(cf. form. (8))} \\ &= \binom{n-2}{d_1 - 1, d_2 - 1, \ldots, d_n - 1} \quad \text{(pt. că } d_n = 1\text{).} \end{aligned}$$

Pentru $T(4; 2, 2, 1, 1) = 2$ avem arborii:

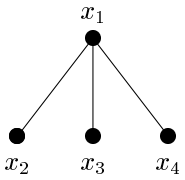


Din formula

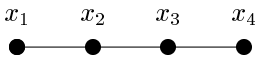
$$\sum_{d_1, \ldots, d_n \geq 1} \binom{n-2}{d_1 - 1, d_2 - 1, \ldots, d_n - 1} = \underbrace{(1 + 1 + \ldots + 1)}_{n}^{n-2} = n^{n-2}$$

rezultă că numărul arborilor cu n vârfuri date este egal cu n^{n-2} (formula lui Cayley).

Pentru $n = 4$ avem 16 arbori. Patru dintre aceştia sunt de forma



(fiecare x_i poate fi rădăcină) şi 12 de forma



unde în $\binom{4}{2} = 6$ moduri pot fi alese vârfurile de grad 1, şi pentru fiecare caz vârfurile interioare pot fi alese în două moduri.

1.4 Aranjamente

Aranjamentele reprezintă o generalizare a permutărilor (în engleză nici nu există un termen pentru aranjamente, se numesc tot permutări – permutări de n luate câte k). Dacă în loc de permutarea a n elemente totdeaunea "permutăm" doar k din cele n, ajungem la formarea grupelor de k elemente distincte. În câte moduri pot fi făcute aceste grupări? Primul element poate fi ales din n,

al doilea din $n-1$, ş.a.m.d. Ultimul element din grupă poate fi ales din cele $n-k+1$ rămase. Astfel formula pentru *aranjamente de n elemente luate câte k* se obţine astfel:

$$A_n^k = n(n-1)(n-2)\dots(n-k+1) = \frac{n!}{(n-k)!}, \qquad k \leq n$$

Dacă un element se poate repeta de oricâte ori în aceste grupe, ajungem la noţiunea de *aranjamente cu repetiţii*. În acest caz, deoarece fiecare element se poate repeta, atât primul, cât şi următoarele elemente pot fi alese din toate. Deci numărul aranjamentelor cu repetiţii a n elemente luate câte k este egal cu n^k.

$$\overline{A}_n^k = n^k$$

De exemplu, numărul aranjamentelor cu repetiţii a 2 elemente câte 8 este 2^8, ceea ce reprezintă numărul numerelor binare formate cu cel mult 8 cifre (ţinând cont de faptul că eliminăm cifrele 0 de la început). (De remarcat, că aici nu mai avem nici o restricţie asupra numărului natural k.)

1.5 Combinări

Sunt probleme în care nu are importanţa ordinea elementelor în cadrul unei grupe. Dacă considerăm aranjamentele de n luate câte k, şi nu vrem să ţinem cont de ordinea elementelor, atunci fiecare grupă apare de $k!$ ori. Astfel numărul *combinărilor de n elemente luate câte k* este egal cu

$$C_n^k = \frac{A_n^k}{P_k} = \frac{n(n-1)\dots(n-k+1)}{k!} = \frac{n!}{k!(n-k)!}$$

Ceea ce demostrează că

$$C_n^k = \binom{n}{k},$$

adică C_n^k reprezintă coeficienţii binomiali.

Dacă admitem că în grupele de mai sus elementele se pot şi repeta, ajungem la noţiunea de *combinări cu repetiţii*. Să considerăm combinările cu repetiţii a 3 elemente luate câte 2:

$$11, \quad 12, \quad 13; \quad 22, \quad 23; \quad 33.$$

Aceste grupe pot fi codificate în felul următor. Pentru fiecare grupă scriem atâtea cifre 1 de câte ori apare primul element în grupa respectivă, după care scriem un 0, pe urmă scriem atâtea cifre 1 de câte ori apare elementul al doilea în grupă, şi iarăşi folosim cifra 0 pentru despărţirea grupelor de cifre 1. Dacă un element nu apare în grupă se scrie numai cifra 0 pentru despărţire, astfel vor fi două cifre 0 alăturate. Pentru grupele de mai sus avem:

$$1100, \quad 1010, \quad 1001, \quad 0110, \quad 0101, \quad 0011.$$

În general, pentru cazul combinărilor cu repetiţii a n elemente luate câte k, în aceste grupe totdeauna există k cifre 1, şi $n-1$ cifre 0 (pentru ultimul element nu mai este necesară utilizarea cifrei 0). Se poate vedea că există o corespondenţă biunivocă între numărul combinărilor cu repetiţii a n elemente luate câte k şi a numărului de permutări cu repetiţii a k elemente 1 şi $n-k$ elemente 0. Deci

$$P_{k,n-1} = \frac{(k+n-1)!}{k!(n-1)!} = \binom{n+k-1}{k}.$$

Astfel:

$$\overline{C}_n^k = C_{n+k-1}^k = \binom{n+k-1}{k}.$$

Pentru combinările cu repetiţii de n elemente luate câte k se mai foloseşte şi notaţia $\left\langle {n \atop k} \right\rangle$. Uşor se poate demonstra următoarea formulă de recurenţă:

$$\left\langle {n \atop k} \right\rangle = \left\langle {n \atop k-1} \right\rangle + \left\langle {n-1 \atop k} \right\rangle$$

Ideea demonstraţiei: Fixăm un anumit element, sunt $\left\langle {n \atop k-1} \right\rangle$ combinări care îl conţin şi sunt $\left\langle {n-1 \atop k} \right\rangle$ combinări care nu-l conţin.

1.6 Generalizarea combinărilor

Definim simbolul $\binom{r}{k}$ pentru orice r real şi pentru orice k întreg:

$$\binom{r}{k} = \begin{cases} \dfrac{r(r-1)(r-2)\dots(r-k+1)}{k(k-1)\dots 1}, & \text{dacă } k > 0 \\ 1 & \text{dacă } k = 0 \\ 0 & \text{dacă } k < 0 \end{cases} \tag{11}$$

Pentru cazurile particulare avem formulele:

$$\binom{r}{0} = 1, \quad \binom{r}{1} = r, \quad \binom{r}{2} = \frac{r(r-1)}{2}$$

Printr-un calcul direct se obţine:

$$\binom{r}{m}\binom{m}{k} = \binom{r}{k}\binom{r-k}{m-k}, \qquad m, k \text{ întregi} \tag{12}$$

$$\binom{-r}{k} = (-1)^k \binom{r+k-1}{k}, \qquad k \text{ întreg} \tag{13}$$

Formula binomului poate fi scrisă pentru orice r real şi k întreg:

$$(x+y)^r = \sum_k \binom{r}{k} x^k y^{r-k} \tag{14}$$

Simbolul $\sum_k$ înseamnă că se face însumarea pentru orice număr k întreg. Termenii din această sumă infinită pentru $k < 0$ şi $k > r$sunt egali cu 0.

Această formula se poate demonstra, folosind dezvoltarea în serie MacLaurin a funcţiei $f(x) = (x+y)^r$ (unde y şi r sunt considerate constante reale) în jurul originii. Dezvoltarea MacLaurin este următoarea:

$$f(x) = f(0) + \frac{f'(0)}{1!}x + \ldots + \frac{f^{(k)}(0)}{k!}x^k + \ldots = \sum_{k\geq 0} \frac{f^{(k)}(0)}{k!}x^k \tag{15}$$

Derivatele succesive ale lui f sunt (pentru $x + y \neq 0$):

$f'(x) = r(x+y)^{r-1}$

$f''(x) = r(r-1)(x+y)^{r-2}$

...

$f^{(k)}(x) = r(r-1)(r-2)\ldots(r-k+1)(x+y)^{r-k}$

Avem:

$$f(0) = y^r,\ f'(0) = ry^{r-1}, \ldots,\ f^{(k)}(0) = r(r-1)\ldots(r-k+1)y^{r-k}$$

Deci, după înlocuire în (15), obţinem

$$(x+y)^r = \sum_{k\geq 0} \frac{r(r-1)\ldots(r-n+1)y^{r-k}}{k!}x^k = \sum_k \binom{r}{k} x^k y^{r-k}$$

ceea ce demonstrează formula (14).

Să demonstrăm două formule interesante ([42]), de care vom avea nevoie ulterior:

$$\sum_k \binom{n}{k}\binom{k}{i}(-1)^k = (-1)^n \delta_{in} \tag{16}$$

unde δ_{in} este simbolul lui Kronecker:

$$\delta_{in} = \begin{cases} 1 & \text{dacă } i = n \\ 0 & \text{dacă } i \neq n \end{cases} \qquad \text{şi}$$

$$\sum_k \binom{n}{k}(-1)^k(b_0 + b_1 k + \ldots + b_{n-1}k^{n-1}) = 0,\ \forall b_0, \ldots b_{n-1} \in \mathbf{R} \tag{17}$$

Prima se demonstrează, plecând de la formula (12). În cazul nostru această formulă se scrie:

$$\binom{n}{k}\binom{k}{i} = \binom{n}{i}\binom{n-i}{k-i}$$

Atunci

$$\sum_k \binom{n}{k}\binom{k}{i}(-1)^k = \sum_k \binom{n}{i}\binom{n-i}{k-i}(-1)^k$$

$$= \binom{n}{i}\sum_{k=0}^{n}\binom{n-i}{k-i}(-1)^k = \binom{n}{i}(-1)^i \sum_{k=-i}^{k-n}\binom{n-i}{k}(-1)^k$$

$$= \binom{n}{i}(-1)^i \sum_{k=0}^{k-n}\binom{n-i}{k}(-1)^k = \begin{cases} 0 & \text{dacă } n \neq i \\ (-1)^n & \text{dacă } n = i \end{cases}$$

Formula (17) se demonstrează plecând de la (16). Înmulţim ambii membri ai formulei (16) cu o constantă $c_i \in \mathbf{R}$, şi pe urmă le adunăm pentru $i = 0, 1, \ldots, n-1$:

$$\sum_{i=0}^{n-1} c_i \sum_k \binom{n}{k}\binom{k}{i}(-1)^k = 0 \qquad \text{sau}$$

$$\sum_k \binom{n}{k}(-1)^k \sum_i^{n-1} c_i \binom{k}{i} = 0$$

Constantele c_i pot fi alese arbitrar, deci $\sum_i^{n-1} c_i \binom{k}{i}$ este un polinom de gradul $n-1$ în k, adică un $b_0 + b_1 k + \ldots b_{n-1}k^{n-1}$.

N. Abel a găsit o generalizare interesantă a formulei binomului:

$$(x+y)^n = \sum_k \binom{n}{k} x(x-kz)^{k-1}(y+kz)^{n-k}, \quad n \text{ natural şi } x \neq 0. \quad (18)$$

Pentru $z = 0$ se obţine formula binomului.

Demonstraţia acestei formule se poate face prin inducţie matematică completă. Evident pentru $n = 0$ formula este adevărată. Presupunem adevărată pentru $n-1$ şi demonstrăm pentru n. Să considerăm funcţiile:

$$f(y) = \sum_k \binom{n}{k} x(x-kz)^{k-1}(y+kz)^{n-k} \text{ şi } g(y) = (x+y)^n$$

Derivând cele două funcţii se obţine:

$$f'(y) = \sum_k \binom{n}{k} x(x-kz)^{k-1}(n-k)(y+kz)^{n-1-k}$$

$$g'(y) = n(x+y)^{n-1} \overset{ip.ind.}{=} n\sum_k \binom{n-1}{k} x(x-kz)^{k-1}(y+kz)^{n-1-k}$$

$$= \sum_k (n-k)\binom{n}{k} x(x-kz)^{k-1}(y+kz)^{n-1-k}$$

Am ţinut cont de formula $n\binom{n-1}{k} = (n-k)\binom{n}{k}$. Se vede că $f'(y) = g'(y)$, deci $f(y) = g(y) + C$, unde C este o constantă, pe care o vom determina luând $y = -x$. Obţinem

$$C = f(-x) - g(-x) = \sum_k \binom{n}{k} x(x-kz)^{k-1}(-x+kz)^{n-k}$$

$$= \sum_k \binom{n}{k} (-1)^{n-k} x(x-kz)^{n-1} = x\sum_k \binom{n}{k} (-1)^{n-k}(x-kz)^{n-1}$$

Pentru orice $x, z \in \mathbf{R}$, expresia $(x-kz)^{n-1}$ este un polinom în k de grad $n-1$, deci după formula (17):

$$\sum_k \binom{n}{k} (-1)^k (x-kz)^{n-1} = 0,$$

deci şi

$$\sum_k \binom{n}{k} (-1)^{n-k} (x-kz)^{n-1} = 0,$$

ceea ce demonstrează că $f(y) = g(y)$, deci formula lui Abel este adevărată.

Vom prezenta şi o altă demonstraţie ingenioasă dată de Lucas ([17]).

Pornim de la polinoamele lui Abel:

$$A_k(x,y) = \frac{x(x-ky)^{k-1}}{k!}, \quad k \geq 1 \quad \text{şi} \quad A_0(x,y) = 1$$

Se poate vedea uşor, printr-un calcul direct, că

$$\frac{\partial A_k(x,y)}{\partial x} = A_{k-1}(x-y,y)$$

Derivând mai departe, se poate observa, că

$$\frac{\partial^\ell A_k(x,y)}{\partial x^\ell} = A_{k-\ell}(x-\ell y,y),$$

ceea ce se demonstrează uşor prin inducţie asupra lui ℓ.

Orice polinom $P(x)$ de orice grad se poate exprima sub forma

$$P(x) = \sum_{k\geq 0} \lambda_k A_k(x,z), \qquad \text{pentru } z \in \mathbf{R},$$

unde valorile λ_k depind numai de z. Derivând polinomul P de ℓ ori obţinem

$$P^{(\ell)}(x) = \frac{d^\ell P(x)}{dx^\ell} = \lambda_\ell + \lambda_{\ell+1} A_1(x-\ell z, z) + \ldots$$

De unde, prin înlocuire cu $x = \ell z$, se obţine $\lambda_\ell = P^{(\ell)}(\ell z)$. Astfel

$$P(x) = \sum_{k\geq 0} A_k(x,z) P^{(k)}(kz).$$

În care, punând $P(x) = (x+y)^n$, se obţine exact formula lui Abel, deoarece

$$\frac{\partial^k (x+y)^n}{\partial x^k} = n(n-1)\ldots(n-k+1)(x+y)^{n-k} = k!\binom{n}{k}(x+y)^{n-k}.$$

Să calculăm coeficienţii dezvoltării expresiei $\left(1 + x + x^2 + \ldots + x^{q-1}\right)^n$. Folosim notaţia:

$$(1 + x + x^2 + \ldots + x^{q-1})^n = \sum_{k=0}^{n(q-1)} \binom{n}{k}_q x^k \tag{19}$$

ceea ce este *formula polinomului*. Membrul stâng este egal cu $(1-x^q)^n(1-x)^{-n}$. Deci

$$\begin{aligned}
&\left(1+x+x^2+\ldots+x^{q-1}\right)^n = \left(1-x^q\right)^n\left(1-x\right)^{-n} = \\
&= \sum_i \binom{n}{i}(-x^q)^i \sum_j \binom{-n}{j}(-x)^j = \\
&= \sum_i \sum_j (-1)^{i+j}\binom{n}{i}\binom{-n}{j}x^{qi+j} = \\
&= \sum_i \sum_j (-1)^{i+j}\binom{n}{i}(-1)^j\binom{n+j-1}{j}x^{qi+j} \\
&= \sum_i \sum_j (-1)^i \binom{n}{i}\left\langle{n \atop j}\right\rangle x^{qi+j}
\end{aligned}$$

De unde rezultă că

$$\binom{n}{k}_q = \sum_{qi+j=k} (-1)^i \binom{n}{i}\left\langle{n \atop j}\right\rangle.$$

Coeficienţii $\binom{n}{k}_q$ se numesc *coeficienţi polinomiali*. Este evident că avem $\binom{n}{k}_2 = \binom{n}{k}$.
Avem următoarea relaţie:

$$\binom{n}{k}_q = \binom{n-1}{k}_q + \binom{n-1}{k-1}_q + \ldots + \binom{n-1}{k-q+1}_q,$$

care se poate demonstra identificând coeficenţii lui x^k în ambii membri ai expresiei:

$$\sum_{k\geq 0}\binom{n}{k}_q x^k = \left(1+x+x^2+\ldots+x^{q-1}\right)^{n-1}\left(1+x+x^2+\ldots+x^{q-1}\right).$$

Se pot verifica uşor următoarele formule:

$$\binom{n}{0}_q = 0, \quad \binom{n}{k}_q = \left\langle{n \atop k}\right\rangle \text{ pentru } k < q.$$

Probleme.

1. Să se demonstreze formula:

$$\binom{k}{k} + \binom{k+1}{k} + \binom{k+2}{k} + \cdots \binom{n}{k} = \binom{n+1}{k+1}, \quad k \leq n$$

2. Să se demonstreze formulele:

a) $\binom{n}{0} + \binom{n}{1} + \binom{n}{2} + \cdots + \binom{n}{n} = 2^n$

b) $\binom{n}{1} + 2\binom{n}{2} + 3\binom{n}{3} + \cdots + n\binom{n}{n} = 2^{n-1}n$

c) $\binom{n}{0} + \frac{1}{2}\binom{n}{1} + \frac{1}{3}\binom{n}{2} \cdots + \frac{1}{n+1}\binom{n}{n} = \frac{2^{n+1}-1}{n+1}$

3. Să se demonstreze că suma coeficienţilor polinomiali $P_{i_1,i_2,\ldots,i_k}$ este egală cu k^n.

4. Să se demonstreze formula:

$$P_{i_1,i_2,\ldots,i_k} = P_{i_1-1,i_2,\ldots,i_k} + P_{i_1,i_2-1,\ldots,i_k} + \ldots + P_{i_1,i_2,\ldots,i_k-1}$$

5. Să se demonstreze formula:

$$\binom{\frac{1}{2}}{n+1} = \frac{(-1)^n}{2^{2n+1}(n+1)}\binom{2n}{n},$$

pentru n natural.

6. Să se demonstreze formula lui Vandermonde pentru orice n întreg, r şi s reali:

$$\sum_k \binom{r}{k}\binom{s}{n-k} = \binom{r+s}{n}$$

7. Să se demonstreze formulele:

a) $\sum_{k=0}^{n} \binom{2n}{k} = 2^{2n-1} + \frac{1}{2}\binom{2n}{n}$

b) $\sum_{k=0}^{n-1} \binom{2n-1}{k} = 2^{2n-2}$

8. Să se demonstreze identitatea:

$$\sum_{k=0}^{n}(n-k)\binom{2n}{k} = \frac{n}{2}\binom{2n}{n}$$

9. Să se demonstreze formula:

$$\sum_{k}\binom{n}{k}^2 = \binom{2n}{n}$$

10. Să se demonstreze formula:

$$\sum_{k=0}^{n(q-1)}\binom{n}{k}_q = q^n$$

11. Să se demonstreze formula:

$$\binom{n}{k}_q = \binom{n}{n(q-1)-k}_q$$

2 Generarea combinărilor, permutărilor şi a unor mulţimi

Pentru generarea combinărilor, permutărilor, aranjamentelor etc. se poate folosi o procedura generală, care la fiecare apel generează într-un vector $V = (v_1, v_2, \ldots, v_n)$ elementele respective. În vectorul respectiv vom păstra numere naturale, care pot fi chiar elementele sau indicele elementelor. Vom folosi un indicator, care la început primeşte valoarea 0, la primul apel se modifică în 1 şi devine din nou 0 când numai avem ce genera. Procedura va avea următoarea structură generală [44]:

procedura generează (V, n, IND):
dacă $IND = 0$ **atunci** $V :=$ valoarea iniţială
$IND := 1$
exit
dacă există următorul **atunci** $V :=$ următorul
altfel $IND := 0$
sfârşit procedură

De remarcat că în această procedură variabila IND este o variabilă de intrare-ieşire. Valoarea ei obţinută în procedură se foloseşte în următorul apel. Apelul general al procedurii este următorul:

$IND := 0$
repetă
cheamă generează(V, n, IND)
dacă $IND = 1$ **atunci cheamă** scrie (V, n)
până când $IND = 0$

2.1 Combinări

Pentru generarea combinărilor de n elemente luate câte k pornim de la vectorul $V = (1, 2, \ldots, k)$, care reprezintă prima combinare. În general, dacă la un moment dat avem combinarea $V = (v_1, v_2, \ldots, v_k)$, vom căuta cel mai mare indice i pentru care $v_i < n - k + i$ şi generăm combinarea

$$(v_1, v_2, \ldots, v_{i-1}, v_i + 1, v_i + 2, \ldots, v_i + n - i + 1).$$

Algoritmul se termină când nu mai există nici un element v_i care să satisfacă condiţia de mai sus.

Se poate vedea uşor că algoritmul de mai sus va generea toate combinările. Elementele vectorului V sunt în ordine crescătoare şi această proprietate se conservă după fiecare pas (după fiecare apel al procedurii), deci algoritmul generează vectori dintincţi. Totodată, datorită modului de generare a următoarei combinări, algoritmul generează toată combinările.

Algoritmul de generare este următorul:

procedura combin (V, n, k, IND)
dacă $IND = 0$ **atunci**
    pentru $i := 1, 2, \ldots, k$ **execută** $v_i := i$
    $IND := 1$
    exit
pentru $i := k, k-1, \ldots, 1$ **execută**
    dacă $v_i < n - k + i$ **atunci**
        $v_i := v_i + 1$
        pentru $j := i+1, i+2, \ldots, k$ **execută** $v_j := v_{j-1} + 1$
        exit
$IND := 0$
sfârşit procedură

Apel:

$IND := 0$
repetă
    cheamă combin(V, n, k, IND)
    dacă $IND = 1$ **atunci cheamă** scrie (V, k)
până când $IND = 0$

Dacă aplicăm algoritmul pentru $n = 6$, $k = 4$ obţinem combinările din tabelul de mai jos.

```
1 2 3 4     1 2 3 5     1 2 3 6     1 2 4 5     1 2 4 6
1 2 5 6     1 3 4 5     1 3 4 6     1 3 5 6     1 4 5 6
2 3 4 5     2 3 4 6     2 3 5 6     2 4 5 6     3 4 5 6
```

2.2 Permutări

Pentru generarea tuturor permutărilor vom prezenta trei metode.

Metoda 1.

Se porneşte de la vectorul iniţial $V = (1, 2, \ldots, n)$. Se caută cel mai mare indice i pentru care $v_i < v_{i+1}$ şi $v_{i+1} > v_{i+2} > \ldots > v_n$, apoi cel mai mare indice k pentru care $v_k > v_i$. Se schimbă între ele elementele v_i şi v_k, şi apoi ultimele $n - i$ elemente (v_{i+j} cu v_{n+1-j} pentru $j = 1, 2, \ldots, \frac{n-i}{2}$)

procedura perm1 (V, n, IND)
  dacă $IND = 0$ **atunci**
    pentru $i = 1, 2, \ldots, n$ **execută** $v_i := i$
    $IND := 1$
    exit
  $i := n - 1$
  cât timp $v_i > v_{i+1}$ **execută**
    $i := i - 1$
    dacă $i = 0$ **atunci**
      $IND := 0$
      exit
  $k := n$
  cât timp $v_i > v_k$ **execută** $k := k - 1$
  $v_i \leftrightarrow v_k$
  pentru $i := 1, 2, \ldots, \lfloor \frac{n-i}{2} \rfloor$ **execută** $v_{i+j} \leftrightarrow v_{n+1-j}$
sfârşit procedură

Apel:

$IND := 0$
repetă
  cheamă perm1 (V, n, IND)
  dacă $IND = 1$ **atunci cheamă** scrie (V, n)
până când $IND = 0$

Exemplu. Pentru $n = 3$ algoritmul ne dă următoarele:

```
1 2 3     1 3 2     2 1 3     2 3 1     3 1 2     3 2 1
```

Metoda 2.

Se porneşte de la vectorul iniţial $V = (1, 2, \ldots, n)$. Se permută circular elementele vectorului, sau numai a unei porţiuni din vector. De exemplu din (1,2,3,4) se obţin prin permutări circulare următoarele: (4,1,2,3), (3,4,1,2) şi (2,3,4,1). Pe urmă din (1,2,3,4) printr-o permutare a ultimelor trei elemente se obţine (1,4,2,3), care se permută din nou în întregime etc.

procedura perm2 (V, n, IND)
  dacă $IND = 0$ **atunci**
      pentru $i = 1, 2, \ldots, n$ **execută** $v_i := i$
      $IND := 1$
      exit
  pentru $k := n, n-1, \ldots, 2$ **execută**
        cheamă circular (V, n, k)
        dacă $V_{n-k+1} \neq n-k+1$ **atunci exit**
  $IND := 0$
sfârşit procedură

Procedura **circular** face o permutare circulară a ultimelor k elemente.

procedura circular (V, n, k)
  $p := v_n$
  pentru $i := n, n-1, \ldots, n-k+2$ **execută** $v_i := v_{i-1}$
  $v_{n-k+1} := p$
sfârşit procedură

Apel:

$IND := 0$
repetă
    cheamă perm2 (V, n, IND)
    dacă $IND = 1$ **atunci cheamă** scrie (V, n)
până când $IND = 0$

Exemplu.

```
1 2 3     3 1 2     2 3 1     1 3 2     2 1 3     3 2 1
```

Metoda 3.

Generarea recursivă a permutărilor se face de jos în sus. Se porneşte cu un element (a_1), din care se formează grupele (a_2, a_1), (a_1, a_2). În general dintr-o grupă $(a_1, a_2, \ldots, a_k)$ se formează grupele:

$(a_{k+1}, a_1, a_2, \ldots, a_k)$,
$(a_1, a_{k+1}, a_2, \ldots, a_k)$,
$(a_1, a_2, a_{k+1}, \ldots, a_k)$,
... ...
$(a_1, a_2, \ldots, a_{k+1}, a_k)$,
$(a_1, a_2, \ldots, a_k, a_{k+1})$.

În descrierea algotimului recursiv se folosesc vectorii intermediari $B = (b_1, b_2, \ldots, b_n)$ şi $C = (c_1, c_2, \ldots, c_n)$.

procedura perm (k,b)
  dacă $k <= n$ **atunci**
    pentru $i := 1, 2, \ldots, k$ **execută**
      pentru $j := 1, 2, \ldots, i-1$ **execută** $c_j := b_j$
      $c_i := a_k$
      pentru $j := i+1, i+2, \ldots, k$ **execută** $c_j := b_{j-1}$
      cheamă perm $(k+1, c)$
  altfel scrie $b_1, b_2, \ldots, b_n$
sfârşit procedură

Apel:

cheamă perm $(1, a)$

```
3 2 1     2 3 1     2 1 3     3 1 2     1 3 2     1 2 3
```

2.3 Combinări cu repetiţii

Programul generează combinările cu repetiţii de n elemente luate câte m. Se porneşte cu vectorul iniţial (1,1, ..., 1) (vector cu m elemente). La fiecare pas din vectorul $(v_1, v_2, \ldots, v_m)$ se formează următorul, după regula: se caută cel mai mare i cu proprietatea $v_i < n$ şi se formează vectorul: $(v_1, v_2, \ldots v_{i-1}, v_i + 1, v_i + 1, \ldots, v_i + 1)$. Evident, astfel se obţin toate combinările cu repetiţii.

procedura combrep (V, n, m, IND)
  dacă $IND = 0$ **atunci**
    pentru $i = 1, 2, \ldots, m$ **execută** $v_i := 1$
    $IND := 1$
    exit
  pentru $i := m, m-1, \ldots, 1$ **execută**
    dacă $v_i \neq n$ **atunci**
      pentru $j := i, i+1, \ldots, m$ **execută** $v_j := v_i + 1$
      exit
  $IND := 0$
sfârşit procedură

Apel:

$IND := 0$
repetă
  cheamă comrep (V, n, m, IND)
  dacă $IND = 1$ **atunci cheamă** scrie (V, m)
până când $IND = 0$

Pentru $n = 4, m = 3$ avem:

```
1 1 1    1 1 2    1 1 3    1 1 4    1 2 2
1 2 3    1 2 4    1 3 3    1 3 4    1 4 4
2 2 2    2 2 3    2 2 4    2 3 3    2 3 4
2 4 4    3 3 3    3 3 4    3 4 4    4 4 4
```

Pentru $n = 2, m = 3$ avem:

```
1 1 1    1 1 2    1 2 2    2 2 2
```

2.4 Generarea produsului cartezian

Fiind date mulţimile $A_i = \{1, 2, \ldots, n_i\}$ (pentru $i = 1, 2, \ldots, m$) produsul cartezian este mulţimea: $\prod_{i=1}^{m} A_i = A_1 \times A_2 \times \cdots \times A_m =$

$$= \{(a_1, a_2, \ldots, a_m) \mid a_1 \in A_1,\ a_2 \in A_2, \ldots, a_m \in A_m\}$$

Algoritmul următor ([44]) generează elementele produsului cartezian pe rând într-un vector $V = (v_1, v_2, \ldots, v_m)$. Se pleacă de la vectorul $V = (1, 1, \ldots, 1)$. Succesorul vectorului V se determină astfel: se caută cel mai mare indice i pentru care $v_i < n_i$ şi dacă el există, atunci succesorul lui V va fi vectorul $(v_1, v_2, \ldots, v_{i-1}, v_i+1, 1, \ldots, 1)$. Când un astfel de indice i nu există, înseamnă că s-au generat toate elementele produsului cartezian în ordine lexicografică.

procedura prodcart (V, m, N, IND)
dacă $IND = 0$ **atunci**
    pentru i:=1,2,...,m **execută** $v_i := 1$
    $IND := 1$
    exit
altfel pentru $i := m, m-1, \ldots, 1$ **execută**
        dacă$v_i < n_i$ **atunci**
            $v_i := v_i + 1$
            exit
        altfel $v_i := 1$
$IND := 0$
sfârşit procedură

Apel:

$IND := 0$
repetă
  cheamă prodcart (V, m, N, IND)
  dacă $IND = 1$ **atunci** scrie (V, m)
până când $IND = 0$

Pentru $m = 3, n_1 = 2, n_2 = 3, n_3 = 2$ obţinem următoarele:

```
1 1 1    1 1 2    1 2 1    1 2 2    1 3 1    1 3 2
2 1 1    2 1 2    2 2 1    2 2 2    2 3 1    2 3 2
```

2.5 Generarea tuturor submulţimilor unei mulţimi

Algoritmul următor ([44]) generează submulţimile pe baza vectorului caracteristic. Fie dată mulţimea $X = \{x_1, x_2, \ldots, x_n\}$ (unde elementele sunt considerate într-o ordine) şi o submulţime Y a lui X. Vectorul caracteristic $(c_1, c_2, \ldots, c_n)$ al submulţimii Y a lui X se defineşte prin:

$$c_i = \begin{cases} 1 & \text{dacă} \quad x_i \in Y \\ 0 & \text{dacă} \quad x_i \notin Y \end{cases} \qquad i = 1, 2, \ldots, n$$

Problema generării vectorilor caracterici se reduce la generarea elementelor produsului cartezian: $\underbrace{\{0,1\} \times \{0,1\} \times \cdots \times \{0,1\}}_{n}$

procedura submult1 (V, m, IND)
dacă $IND = 0$ **atunci**
    pentru i:=1,2,...,m **execută** $v_i := 0$
    $IND := 1$
    exit
altfel pentru $i := m, m-1, \ldots, 1$ **execută**
        dacă $v_i < 1$ **atunci**
            $v_i := 1$
            exit
        altfel $v_i := 0$
$IND := 0$
sfârşit procedură

Apel:

$IND := 0$
repetă
  cheamă submult1 (V, m, N, IND)
  dacă $IND = 1$ **atunci** scrie (V, m)
până când $IND = 0$

Pentru o mulţime de 3 elemente obţinem:

$\emptyset$ `{1}` `{2}` `{1,2}` `{3}` `{1,3}` `{2,3}` `{1,2,3}`

Următorul algoritm generează submulţimile în ordinea crescătoare a numărului lor de elemente. Se va folosi un vector V, care la început are toate componentele egale cu 0. Succesorul vectorului $V = (v_1, v_2, \ldots, v_m)$ se determină astfel: se caută cel mai mare indice i pentru care $v_i < i$ (în acest caz avem şi următoarele relaţii: $v_{i+1} = i+1, \ldots, v_{m-1} = m-1, v_m = m$). Succesorul lui V va fi: $(v_1, v_2 \ldots, v_{i-1}, v_i + 1, v_i + 2, \ldots v_i + m - i + 1)$. În interpretarea vectorului V se elimină elementele nule, de exemplu: vectorul $(0, 0, 1, 2)$ reprezintă submulţimea: $\{1, 2\}$.

procedura submult2 (V, m, IND)
dacă $IND = 0$ **atunci**
    pentru i:=1,2,...,m **execută** $v_i := 0$
    $IND := 1$
    exit
altfel pentru $i := m, m-1, \ldots, 1$ **execută**
        dacă $v_i < i$ **atunci**
            $v_i := v_i + 1$

pentru $j := i+1, i+2, \ldots, m$ **execută** $v_j := v_{j-1} + 1$
exit
altfel $v_i := 0$
$IND := 0$
sfârşit procedură

Apel:

$IND := 0$
repetă
cheamă submult2 (V, m, IND)
dacă $IND = 1$ **atunci** scrie (V, m)
până când $IND = 0$

Tot pentru o mulţime cu 3 elemente avem:

$\emptyset$ `{1}` `{2}` `{3}` `{1,3}` `{2,3}` `{1,2,3}`

3 Partiţiile unui număr natural

Prin partiţia unui număr întreg înţelegem descompunera lui într-o sumă de numere naturale. De exemplu 5 poate fi descompus în sumă de numere naturale în următoarele moduri:

```
5
4+1
3+2
3+1+1
2+2+1
2+1+1+1
1+1+1+1+1
```

Astfel 5 are 7 partiţii distincte. Într-o partiţie nu contează ordinea elementelor. Ne interesează numărul partiţiilor unui număr natural.

3.1 Partiţiile unui număr natural fără restricţii

Următorul program generează toate partiţiile unui număr natural, folosind un vector $(v_1, v_2, \ldots, v_n)$, pe baza procedurii generale din capitolul 2. Ideea algoritmului [44] este că la început se iniţializează $v_n := n$ şi $v_i := 0$ pentru $i := 1, 2, \ldots, n-1$. Apoi se caută cel mai mare indice i pentru care $v_i + 1 < v_n$, se atribuie valoarea $v_i + 1$ elementelor $v_i, v_{i+1}, \ldots, v_{n-1}$ şi se modifică corespunzător valoarea lui v_n.

procedura partitie (V, n, IND)
dacă $IND = 0$ **atunci**
    pentru $i := 1, 2, \ldots, n-1$ **execută** $v_i := 0$
    $v_n := n$
    $IND := 1$
    exit
$x := v_n$
pentru $i := n-1, n-2, \ldots, 1$ **execută**
    dacă $v_i + 1 < v_n$ **atunci**
        $m := v_i + 1$
        pentru $j := i, i+1, \ldots, n-1$ **execută** $v_j := m$
        $v_n := x - (n-i-1)m - 1$
        exit

altfel $x := x + v_i$
$IND := 0$
sfârşit procedură

Apel:

$IND := 0$
repetă
cheamă partitie(V, n, IND)
dacă $IND = 1$ **atunci** scrie (V, n)
până când $IND = 0$

Exemplu. Pentru $n = 4$ avem:

```
4
1 3
2 2
1 1 2
1 1 1 1
```

3.2 Partiţiile lui n în numere mai mici sau egale cu m

Suntem interesaţi în numărul partiţiilor în care fiecare număr poate fi cel mult egal cu un număr m dat. Să notăm cu $P(n, m)$ numărul partiţiilor lui n în care fiecare element poate fi cel mult egal cu m. De exemplu: $P(5, 2)$=3, iar partiţiile sunt: $2 + 2 + 1$, $2 + 1 + 1 + 1$ şi $1 + 1 + 1 + 1 + 1$.

Pentru $P(n, m)$ se poate găsi uşor o formulă de recurenţă. Împărţind aceste partiţii în două categorii: cele care nu conţin numere egale cu m (numărul lor este egal cu $P(n, m - 1)$) şi cele care conţin numere egale cu m (numărul lor este egal cu $P(n - m, m)$) obţinem următoarea formulă de recurenţă:

$$P(n, m) = P(n, m - 1) + P(n - m, m), \quad \text{unde} \quad n > m > 1$$

Ţinând cont şi de cazurile particulare, obţinem următoarele formule:

$$\begin{aligned} &P(n, m) = P(n, m - 1) + P(n - m, m) && \text{dacă} \quad n > m > 1 \\ &P(n, m) = 1 + P(n, n - 1) && \text{dacă} \quad 1 < n \leq m \\ &P(1, m) = P(n, 1) = 1 \end{aligned}$$

Următoarea procedură (concepută pe baza unui program din [34]) se bazează pe aceste formule. Ea foloseşte un vector $s_1, s_2, \ldots s_{ind}$ care păstrează elementele partiţiei.

procedura partitie (n, m)
dacă $m = 1$ sau $n = 1$ **atunci**
  scrie $\underbrace{1, 1, \ldots 1}_{n}, s_1, s_2, \ldots, s_{ind}$
altfel **dacă** $n \leq m$ **atunci**
  scrie $n, s_1, s_2, \ldots, s_{ind}$
  cheamă partitie $(n, n-1)$
  altfel **cheamă** partitie $(n, m-1)$
  $ind := ind + 1$, $s_{ind} := m$
  cheamă partitie $(n-m, m)$
  $ind := ind - 1$
sfârşit procedură

Apelul procedurii se face prin:

$ind := 0$
cheamă partitie (n, m)

Exemplu. Pentru $n = 5, m = 3$ avem:

```
1 1 1 1 1
1 1 1 2
1 2 2
2 3
1 1 3
```

Acest algoritm poate fi folosit şi pentru rezolvarea problemei precedente dacă se pune $m := n$. De exemplu pentru $n = 4$, $m = 4$ avem:

```
4
1 1 1 1
2 2
1 1 2
1 3
```

3.3 Partiţiile lui n în numere distincte mai mici sau egale cu m

Să notăm cu $Q(n, m)$ numărul partiţiilor numărului n în numere distincte nu mai mari de m. În acest caz se deduc uşor următoarele formule de recurenţe:

$$\begin{aligned} &Q(n,m) = Q(n,m-1) + Q(n-m,m-1) && \text{dacă } n > m > 1 \\ &Q(n,m) = 1 + Q(n,n-1) && \text{dacă } 1 < n \leq m \\ &Q(1,m) = Q(n,1) = 0 \end{aligned}$$

Se poate scrie o procedură asemănătoare celei din secţiunea 3.2.

Exemplu. Pentru $n = 10, m = 5$ avem:

```
1 4 3 2
2 5 3
1 5 4
```

3.4 Partiţiile lui n în k numere mai mici sau egale cu m

Dacă $S(n, m, k)$ reprezintă numărul partiţiilor numărului n în k numere (nu neaparat distincte) mai mici sau egale cu m, avem următoarele formule:

$$\begin{aligned} &S(n,m,k) = S(n,m-1,k) + S(n-m,m,k-1) \\ &\qquad\qquad && \text{dacă } n > m > 1, k > 1 \\ &S(n,m,1) = S(n,m-1,1) + 1 && \text{dacă } 1 < n \leq m \\ &S(1,n,k) = 1, S(n,1,k) = 0 \\ &S(n,m,1) = 1 && \text{dacă } n \leq m \\ &S(n,m,1) = 0 && \text{dacă } n > m \end{aligned}$$

Se poate scrie o procedură asemănătoare celei din secţiunea 3.2.

Exemplu. Pentru $n = 8, m = 4, k = 5$ avem:

```
1 2 2 2 1
1 3 2 1 1
1 4 1 1 1
```

3.5 Partiţiile lui n în k numere distincte mai mici sau egale cu m

Dacă $R(n,m,k)$ reprezintă numărul partiţiilor numărului n în k numere distincte mai mici sau egale cu m, avem următoarele formule:

$$R(n,m,k) = R(n,m-1,k) + R(n-m,m-1,k-1) \quad \text{dacă } n > m > 1, k > 1$$
$$R(n,m,1) = R(n,m-1,1) + 1 \quad \text{dacă } 1 < n \leq m$$
$$R(1,n,k) = 1, R(n,1,k) = 0$$
$$R(n,m,1) = 1 \quad \text{dacă } n \leq m$$
$$R(n,m,1) = 0 \quad \text{dacă } n > m$$

Se poate scrie o procedură asemănătoare celei din secţiunea 3.2.

Exemplu. Pentru $n = 10, m = 5, k = 3$ avem:

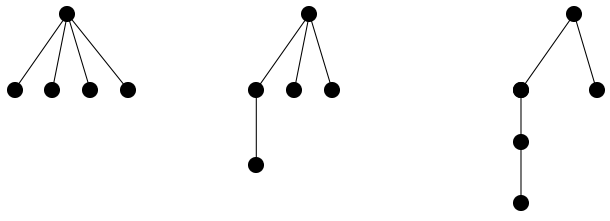
```
2 5 3
1 5 4
```

Enumerarea arborilor. Partiţiile unui număr natural pot fi folosite pentru enumerarea arborilor. Dacă un arbore are n vârfuri atunci numărul muchiilor este $n-1$, deci suma gradelor vârfurilor este $2(n-1)$ (dublul numărului muchiilor). Vom partiţiona acest număr $2(n-1)$ în $k = n$ numere nu neaparat distincte nu mai mari de $n-1$. Aceste numere vor reprezenta gradul vârfurilor. Se poate demonstra că o astfel de partiţie totdeauna reprezintă gradul vârfurilor unui arbore.

De exemplu, dorim să enumerăm toţi arborii cu 5 vârfuri (Vom considera arbori neetichetaţi). Atunci partiţionăm numărul 8 în 5 numere nu mai mari de 4. Aceste partiţii sunt (cum am văzut în secţiunea precedentă):

```
4 1 1 1 1
3 2 1 1 1
2 2 2 1 1
```

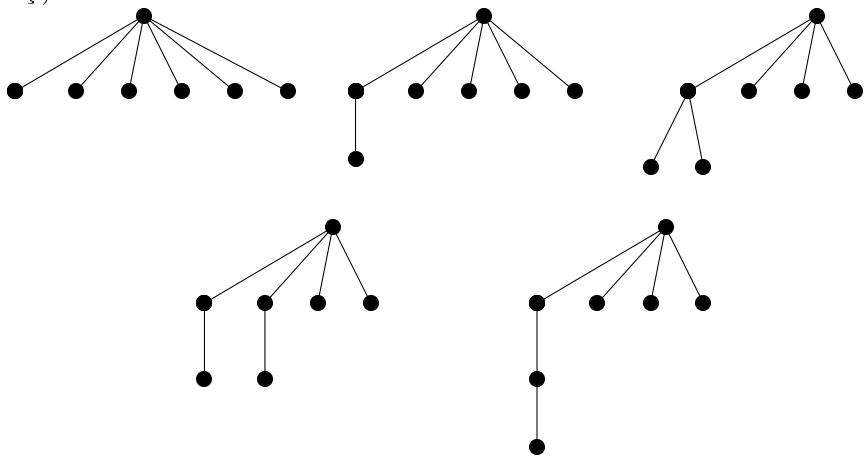
Arborii corespunzători sunt:



Să enumerăm toţi arborii cu 7 vârfuri în care gradul maxim este mai mare decât 4. Partiţionăm numărul 12 în 7 numere nu mai mari de 6, şi eliminăm partiţiile în care numărul cel mai mare este mai mic de 4. Partiţiile sunt următoarele:

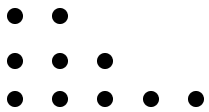
```
6 1 1 1 1 1 1      3 3 2 1 1 1 1
5 2 1 1 1 1 1      3 2 2 2 1 1 1
4 3 1 1 1 1 1      2 2 2 2 2 1 1
4 2 2 1 1 1 1
```

Cele 3 partiţii din partea dreapta vor fi eliminate. Arborii corespunzători celor 4 partiţii rămase sunt următoarele (pentru partiţia 4 2 2 1 1 1 1 există doi arbori distincţi):

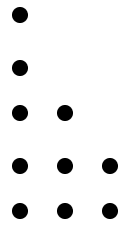


3.6 Diagrama lui Ferrers

Orice partiţie a unui număr întreg poate fi reprezentată printr-o diagramă, numită *diagrama lui Ferrers*. Dacă un număr se partiţionează în numerele $\pi_1, \pi_2, \ldots, \pi_k$, astfel încât $\pi_1 \geq \pi_2 \geq \ldots \geq \pi_k$, atunci diagrama lui Ferrers va fi formată din π_1 puncte în prima linie, π_2 puncte în linia a doua, ..., π_k puncte în linia a k-a, punctele fiind aliniate şi pe coloane. De exemplu partiţia $(5, 3, 2)$ se va reprezenta prin diagrama:



Evident că oricărei partiţii reprezentate printr-o diagramă a lui Ferrers i se poate asocia o altă partiţie, care se obţine prin numărarea punctelor pe coloane. Astfel partiţiei de mai sus i se asociază *partiţia conjugată*: $(3, 3, 2, 1, 1)$. Diagrama lui Ferrers a acestei partiţii se obţine din cea originală prin aşezarea coloanelor pe linii.



Fie $\pi = (\pi_1, \pi_2, \ldots, \pi_k)$ este o partiţie a lui n, adică $n = \pi_1 + \pi_2 + \ldots + \pi_k$ şi $\pi' = (\pi'_1, \pi'_2, \ldots, \pi'_m)$ partiţia ei conjugată (unde $m = \pi_1$). Se defineşte *ponderea partiţiei* ca fiind egală cu suma elementelor din partiţie, deci $|\pi| = \pi_1 + \pi_2 + \ldots + \pi_k = n$. *Pătratul lui Durfee* ataşat partiţiei π este cel mai pătrat de puncte din diagrama lui Ferrers a partiţiei respective. Numărul liniilor al acestui pătrat se notează cu $d(\pi)$ şi se numeşte *diagonala diagramei*. În diagramele următoare conjugate avem $d(\pi) = d(\pi') = 3$.

$\pi = (6, 4, 3, 1)$ $\qquad$ $\pi' = (4, 3, 3, 2, 1, 1)$

Rangul unei partiţii π de defineşte în felul următor:

$$r(\pi) = [\pi_1 - \pi'_1, \pi_2 - \pi'_2, \ldots, \pi_{d(\pi)} - \pi'_{d(\pi)}]$$

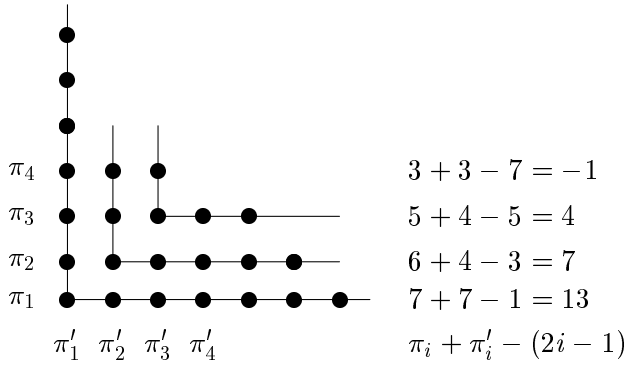
Pentru exemplul de mai sus avem: $r(\pi) = (2, 1, 0)$.
Pentru o partiţie $\pi = (\pi_1, \pi_2, \ldots, \pi_k)$ valoarea lui $d(\pi)$ (notată numai prin d) se calculează în felul următor:

$i := 0$
cât timp $\pi_{i+1} + \pi'_{i+1} - (2i + 1) > 0$ **execută** $i := i + 1$
$d := i$

sau prin formulă matematică

$$d(\pi) = \max_{i=1,2,\ldots,k} \left\{ i \;\middle|\; \pi_i + \pi'_i - (2i - 1) > 0 \right\}$$

Pentru că în algoritmul de mai sus rolul lui π şi π' se poate schimba fără ca rezultatul să fie afectat, rezultă că $d(\pi) = d(\pi')$.



Dintre toate partiţiile care au acelaşi rang cea care are ponderea minimă se numeşte *partiţie de bază*. Să consiredăm următoarele partiţii împreună cu conjugatele lor:

$\pi = (5, 3, 3, 2, 1), \quad |\pi| = 14$
$\pi' = (5, 4, 3, 1, 1)$
$r_\pi = (0, -1, 0)$

$\rho = (6, 4, 3, 2, 2, 1), \quad |\rho| = 18$
$\rho' = (6, 5, 3, 2, 1, 1)$
$r_\rho = (0, -1, 0)$

$\tau = (4, 4, 3, 1), \quad |\tau| = 12$
$\tau' = (4, 3, 3, 2)$
$r_\tau = (0, -1, 0)$

Dintre acestea ultima este de bază. Unui rang dat i se pot asocia o infinitate de partiţii. În mulţimea tuturor partiţiilor cu acelaşi rang există una singură cu pondere minimă. În [54] se dau două condiţii necesare şi suficiente pentru ca o partiţie să fie de bază. În teorema următoare π este o partiţie, π' conjugata ei, iar d diagonala diagramei corespunzătoare lui π.

Teorema 2. *O partiţie $\pi = (\pi_1, \pi_2, \ldots, \pi_k)$ cu diagonala d este de bază dacă şi numai dacă sunt adevărate următoarele două afirmaţii:*

a) $\pi_d = d$ sau $\pi'_d = d$,

b) Pentru $1 \leq i < d$, dacă $\pi_i > \pi_{i+1}$ atunci $\pi'_i = \pi'_{i+1}$.

Pentru exemplul de mai sus

$\tau = (4, 4, 3, 1), \quad |\tau| = 12$

$\tau' = (4, 3, 3, 2)$

$r_\tau = (0, -1, 0)$

avem $d = 3$, $\tau_3 = \tau'_3 = 3$ şi $\tau_2 > \tau_3 \Rightarrow \tau'_2 = \tau'_3 = 3$.

Pentru a doua teoremă avem nevoie de următoarele. Pentru o partiţie π cu d dat definim partiţiile ρ şi σ în felul următor:

$$\rho = (\pi_1 - d, \pi_2 - d, \ldots, \pi_d - d) \text{ şi } \sigma = (\pi'_1 - d, \pi_2 - d, \ldots, \pi'_d - d)$$

(Acestea se obţin din diagrama lui Ferrers după eliminarea pătratului lui Durfee.) Părţile egale cu 0 se elimină din partiţiile definite. Astfel orice partiţie π se poate defini prin $\pi = (d, \rho, \sigma)$.

Teorema 3. *O partiţie $\pi = (d, \rho, \sigma)$ este de bază dacă şi numai dacă partiţiile conjugate ρ' şi σ' nu au elemente comune.*

Exemple.

1. $\pi = (6, 4, 3, 2, 2, 1), \quad |\pi| = 18$

   $\pi' = (6, 5, 3, 2, 1, 1)$

   $r_\pi = (0, -1, 0)$

Partiţia π nu este de bază pentru că $d = 3$ şi

$\rho = (3, 1, 0)$, deci $\rho = (3, 1)$, $\rho' = (2, 1, 1)$

$\sigma = (3, 2, 0)$, adică $\sigma = (3, 2)$, $\sigma' = (2, 2, 1)$

şi partiţiile σ' şi ρ' au elemente comune.

2. Partiţia $\tau = (4, 4, 3, 1)$ este de bază. $\tau' = (4, 3, 3, 2)$, $d = 3$, $\rho = (1, 1)$, $\sigma = (1)$ iar conjugatele sunt: $\rho' = (2)$, $\sigma' = (1)$ care nu au elemente comune.

4 Funcţii generatoare

4.1 Definiţie şi operaţii

Un şir infinit de numere reale $(a_n)_{n\geq 0} =< a_0, a_1, a_2, \ldots, a_n, \ldots >$ poate fi reprezentat cu ajutorul unei serii de puteri:

$$A(z) = a_0 + a_1 z + a_2 z^2 + \ldots + a_n z^n + \ldots = \sum_{n\geq 0} a_n z^n,$$

care se numeşte *funcţia generatoare* a şirului $< a_n >$.

De exemplu pentru numerele lui Fibonacci $F_n = F_{n-1} + F_{n-2}$, cu $F_0 = 0$, şi $F_1 = 1$ avem următoarea funcţie generatoare:

$$F(z) = \sum_{n\geq 0} F_n z^n = z + z^2 + 2z^3 + 3z^4 + 5z^5 + 8z^6 + 13z^7 + \ldots$$

Vom înmulţi formal membrul I şi II cu z apoi cu z^2. Deci avem următoarele formule:

$$\begin{aligned} F(z) &= F_0 + F_1 z + F_2 z^2 + F_3 z^3 + \ldots + F_n z^n + \ldots \\ zF(z) &= F_0 z + F_1 z^2 + F_2 z^3 + \ldots + F_{n-1} z^n + \ldots \\ z^2 F(z) &= F_0 z^2 + F_1 z^3 + \ldots + F_{n-2} z^n + \ldots \end{aligned}$$

Dacă scădem membru cu membru cele două formule din urmă din prima, şi ţinem cont de formula de definiţie a numerelor lui Fibonacci, obţinem:

$$F(z)(1 - z - z^2) = z$$

de unde

$$F(z) = \frac{z}{1 - z - z^2} \tag{20}$$

Calculele de mai sus pot fi justificate matematic, dar nu insistăm asupra acesteia. Totuşi, dacă dezvoltăm funcţia de mai sus în serie MacLauren, obţinem exact seria generatoare a numerelor lui Fibonacci. Pentru seriile de puteri se pot stabile valorile lui z pentru care seria converge. Dacă $z \in \mathbf{R}$ şi $\lim_{n\to\infty} \sqrt[n]{|a_n|} = r$ atunci seria este convergentă pentru orice $|z| < r$. Pentru cele ce urmează convergenţa nu are importanţă, dacă totuşi ne îndoim de justeţea calculelor efectuate, rezultatul obţinut se poate verifica totdeaunea şi prin alte căi.

Funcţiile generatoare au multe aplicaţii. Cu ajutorul lor pot fi demonstrate o serie de identităţi, pot fi obţinute formule interesante.

Fie date funcţiile generatoare: $A(z) = \sum_{n\geq 0} a_n z^n$ şi $B(z) = \sum_{n\geq 0} b_n z^n$, înşirăm în continuare câteva operaţii care se pot efectua cu ele.

a) *egalitate*

$A(z) = B(z)$ dacă şi numai dacă $a_n = b_n$ pentru orice $n \in \mathbf{N}$.

b) *adunare*

$$\alpha A(z) + \beta B(z) = \sum_{n\geq 0} (\alpha a_n + \beta b_n) z^n$$

c) *deplasare*

Funcţia

$$z^k A(z) = \sum_{n\geq 0} a_n z^{n+k} = \sum_{n\geq k} a_{n-k} z^n$$

reprezintă şirul $\underbrace{0, 0, \ldots, 0}_{k}, a_0, a_1, \ldots$, iar funcţia

$$\frac{1}{z^k}(A(z) - a_0 - a_1 z - a_2 z^2 - \ldots - a_{k-1} z^{k-1}) = \sum_{n\geq k} a_n z^{n-k} = \sum_{n\geq 0} a_{k-n} z^n$$

reprezintă şirul $a_k, a_{k+1}, a_{k+2}, \ldots$

Exemplu: Fie $A(z) = 1 + z + z^2 + \ldots$. Avem:

$$\frac{1}{z}\Big(A(z) - 1\Big) = A(z), \quad \text{de unde} \quad \mathrm{A(z)} = \frac{1}{1 - \mathrm{z}}$$

d) *înmulţire*

$$A(z)B(z) = (a_0 + a_1 z + \ldots + a_n z^n + \ldots)(b_0 + b_1 z + \ldots + b_n z^n + \ldots) =$$

$$= a_0 b_0 + (a_0 b_1 + a_1 b_0) z + (a_0 b_2 + a_1 b_1 + a_2 b_0) z^2 + \ldots = \sum_{n\geq 0} s_n z^n$$

unde $s_n = \sum_{k=0}^{n} a_k b_{n-k}$.

Caz particular: Dacă $b_n = 1$ pentru orice n natural, atunci

$$A(z)\frac{1}{1-z} = \sum_{n\geq 0} (\sum_{k=0}^{n} a_k) z^n \tag{21}$$

Dacă avem şi $a_n = 1$ pentru orice n natural, obţinem formula

$$\frac{1}{(1-z)^2} = \sum_{n\geq 0}(n+1)z^n. \tag{22}$$

e) *derivare*

$$A'(z) = a_1 + 2a_2 z + 3a_3 z^2 + \ldots = \sum_{n\geq 0}(n+1)a_{n+1}z^n$$

Exemplu: Plecând de la funcţia generatoare

$$A(z) = \sum_{n\geq 0} z^n = \frac{1}{1-z},$$

derivând membru cu membru, obţinem:

$$A'(z) = \sum_{n\geq 1} n z^{n-1} = \frac{1}{(1-z)^2}.$$

f) *integrare*

$$\int_0^z A(t)dt = a_0 z + \frac{1}{2}a_1 z^2 + \frac{1}{3}a_2 z^3 + \ldots = \sum_{n\geq 1}\frac{1}{n}a_{n-1}z^n$$

Exemplu:

Fie funcţia generatoare:

$$\frac{1}{1-z} = 1 + z + z^2 + z^3 + \ldots$$

Prin integrare membru cu membru obţinem:

$$\ln\frac{1}{1-z} = z + \frac{1}{2}z^2 + \frac{1}{3}z^3 + \ldots = \sum_{n\geq 1}\frac{1}{n}z^n$$

Prin înmulţirea celor două funcţii generatoare de mai sus, obţinem:

$$\frac{1}{1-z}\ln\frac{1}{1-z} = \sum_{n\geq 1} H_n z^n$$

unde $H_n = 1 + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{n}$, $(H_0 = 0,\ \ H_1 = 1)$ sunt aşa-numitele numere armonice.

g) *schimbarea argumentului*

Fie dată funcţia generatoare $A(z) = \sum_{n\geq 0} a_n z^n$ reprezentând şirul $a_0, a_1, a_2, \ldots$, atunci funcţia $A(cz) = \sum_{n\geq 0} c^n a_n z^n$ reprezintă şirul $a_0, ca_1, c^2 a_2, \ldots c^n a_n, \ldots$. Avem şi următoarele formule:

$$\frac{1}{2}\left(A(z) + A(-z)\right) = a_0 + a_2 z^2 + \ldots + a_{2n} z^{2n} + \ldots$$

$$\frac{1}{2}\left(A(z) - A(-z)\right) = a_1 z + a_3 z^3 + \ldots + a_{2n-1} z^{2n-1} + \ldots$$

Exemplu: Fie $A(z) = 1 + z + z^2 + z^3 + \ldots = \dfrac{1}{1-z}$. Atunci

$$1 + z^2 + z^4 + \ldots = \frac{1}{2}\left(A(z) + A(-z)\right) = \frac{1}{2}\left(\frac{1}{1-z} + \frac{1}{1+z}\right) = \frac{1}{1-z^2}$$

care se poate obţine şi direct înlocuind z cu z^2 în $A(z)$.

La fel se poate obţine o formulă pentru suma termenilor cu puteri impari:

$$z + z^3 + z^5 + \ldots = \frac{1}{2}\left(A(z) - A(-z)\right) = \frac{1}{2}\left(\frac{1}{1-z} - \frac{1}{1+z}\right) = \frac{z}{1-z^2}.$$

Cu ajutorul funcţiilor generatoare se pot obţine formule interesante. Fie $A(z) = \dfrac{1}{1-z} = 1 + z + z^2 + z^3 + \ldots$. Atunci $zA(z(1+z)) = F(z)$, adică tocmai funcţia generatoare a numerelor lui Fibonacci. Din formula de mai sus obţinem

$$zA(z(1+z)) = z + z^2(1+z) + z^3(1+z)^2 + z^4(1+z)^3 + \ldots$$

Coeficientul lui z^{n+1} din membrul stâng este F_{n+1} (adică, al $(n+1)$-lea număr Fibonacci), iar coeficientul lui z^{n+1} din membrul drept, după aplicarea formulei binomului în fiecare termen, este $\sum_{k\geq 0} \binom{n-k}{k}$. De unde obţinem formula

$$F_{n+1} = \sum_{k\geq 0} \binom{n-k}{k} = \sum_{k=0}^{\lfloor \frac{n+1}{2} \rfloor} \binom{n-k}{k} \tag{23}$$

Reamintim că formula binomului se poate generaliza pentru orice r real, adică

$$(1+z)^r = \sum_{n\geq 0} \binom{r}{n} z^n,$$

care reprezintă funcţia generatoare pentru coeficienţii binomiali.

Cu ajutorul formulei de mai sus (pentru r întreg negativ) se poate obţine o formulă asemănătoare utilă în multe demonstraţii. Să pornim de la funcţia

$$\frac{1}{(1-z)^m} = (1-z)^{-m} = \sum_{k\geq 0} \binom{-m}{k} (-z)^k$$

Deoarece după formula (13) avem

$$\binom{-m}{k} = (-1)^k \binom{m+k-1}{k},$$

se poate obţine următoarea formulă:

$$\frac{1}{(1-z)^{m+1}} = \sum_{k\geq 0} \binom{m+k}{k} z^k$$

Atunci

$$\frac{z^m}{(1-z)^{m+1}} = \sum_{k\geq 0} \binom{m+k}{k} z^{m+k} = \sum_{k\geq 0} \binom{m+k}{m} z^{m+k} = \sum_{k\geq 0} \binom{k}{m} z^k$$

Se obţine următoarea formulă interesantă:

$$\sum_{k\geq 0} \binom{k}{m} z^k = \frac{z^m}{(1-z)^{m+1}}, \qquad \text{pentru } m \text{ întreg nenegativ.} \tag{24}$$

4.2 Aplicaţii ale funcţiilor generatoare

Funcţiile generatoare, printre altele, pot fi aplicate la numărarea unor obiecte, la demonstrarea unor identităţi, la rezolvarea problemelor de partiţionare.

Numărarea unor obiecte se face prin obţinerea unor formule recursive, care pe urmă sunt folosite în funcţii generatoare, care după dezvoltarea lor ne dau valorile căutate.

Demonstrarea unor identităţi se poate face în mai multe moduri:

– prin dezvoltarea unor funcţii generatoare în două moduri şi egalând coeficienţii obţinuţi,

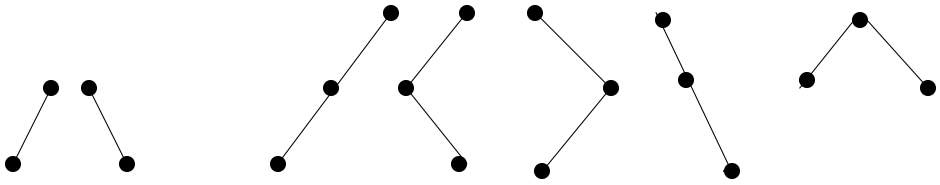
– prin schimbarea ordinii de însumare, dacă coeficienţii sunt deasemenea sume,

– prin diferite operaţii (derivare, integrare) asupra unor funcţii generatoare,

– prin folosirea formulelor recursive (înlocuirea valorilor cunoscute).

4.2.1 Numărarea arborilor binari

Câţi arbori binari cu n vârfuri există? Cu un vârf există un singur arbore binar. Cu două vârfuri există 2 arbori binari, iar cu trei 5.



n=2 n=3

Vom nota cu b_n numărul arborilor binari cu n vârfuri. Prin convenţie $b_0 = 1$. Dacă fixăm rădăcina arborelui, ne mai rămân $n-1$ vârfuri care pot apărea în subarborele stâng sau drept. Dacă în subarborele stâng sunt k vârfuri, în subarborele drept trebuie să fie $n-1-k$. Cu aceşti subarbori se pot forma în total $b_k b_{n-1-k}$ arbori, adunând aceste valori pentru $k = 0, 1, \ldots, n-1$ vom obţine exact valoarea lui b_n. Deci pentru $n \geq 1$

$$b_n = b_0 b_{n-1} + b_1 b_{n-2} + \ldots + b_{n-1} b_0 = \sum_{k=0}^{n-1} b_k b_{n-1-k} \tag{25}$$

Înmulţind ambii membri cu z şi însumând după n, obţinem:

$$\sum_{n\geq 1} b_n z^n = \sum_{n\geq 1} \left(\sum_{k=0}^{n-1} b_k b_{n-1-k} \right) z^n \tag{26}$$

Fie $B(z) = \sum_{n\geq 0} b_n z^n$ funcţia generatoare a numerelor b_n. În acest caz membrul stâng de mai sus este egal cu $B(z) - 1$ (deoarece $b_0 = 1$). Membrul drept seamănă foarte mult cu produsul a două funcţii generatoare. Fie

$$A(z) = zB(z) = \sum_{n\geq 0} b_n z^{n+1} = \sum_{n\geq 1} b_{n-1} z^n$$

Atunci membrul drept din formula (26) este egal cu $A(z)B(z) = zB^2(z)$. Deci

$$B(z) - 1 = zB^2(z), \quad B(0) = 1$$

Rezolvând această ecuaţie în $B(z)$ obţinem:

$$B(z) = \frac{1 \pm \sqrt{1-4z}}{2z}$$

Din cauza că $B(0) = 1$ ne convine numai soluţia cu semnul $-$. Deci

$$B(z) = \frac{1}{2z}\left(1 - \sqrt{1-4z}\right) = \frac{1}{2z}\left(1 - (1-4z)^{\frac{1}{2}}\right) =$$

$$= \frac{1}{2z}\left(1 - \sum_{n\geq 0}\binom{\frac{1}{2}}{n}(-4z)^n\right) = \frac{1}{2z}\left(1 - \sum_{n\geq 0}\binom{\frac{1}{2}}{n}(-1)^n 2^{2n} z^n\right) =$$

$$= \frac{1}{2z} - \binom{\frac{1}{2}}{0}\frac{2^0 z^0}{2z} + \binom{\frac{1}{2}}{1}\frac{2^2 z}{2z} - \ldots - \binom{\frac{1}{2}}{n}(-1)^n \frac{2^{2n} z^n}{2z} + \ldots =$$

$$= \binom{\frac{1}{2}}{1}2 - \binom{\frac{1}{2}}{2}2^3 z + \ldots - \binom{\frac{1}{2}}{n}(-1)^n 2^{2n-1} z^{n-1} + \ldots =$$

$$= \sum_{n\geq 0}\binom{\frac{1}{2}}{n+1}(-1)^n 2^{2n+1} z^n = \sum_{n\geq 0}\frac{1}{n+1}\binom{2n}{n} z^n$$

De unde $b_n = \dfrac{1}{n+1}\binom{2n}{n}$.

Observaţie. Pentru ultima transformare se aplică formula din problema 1 de la pagina 26:

$$\binom{\frac{1}{2}}{n+1} = \frac{(-1)^n}{2^{2n+1}(n+1)}\binom{2n}{n}$$

În demonstraţia de mai sus am obţinut următoarea funcţie generatoare

$$\sum_{n\geq 0}\frac{1}{n+1}\binom{2n}{n} z^n = \frac{1-\sqrt{1-4z}}{2z}, \tag{27}$$

care poate fi utilă în demonstrarea altor formule.

Avem şi formula:

$$\sqrt{1-4z} = \sum_{n\geq 0}\frac{-1}{2n-1}\binom{2n}{n} z^n \tag{28}$$

care se poate demonstra printr-un calcul direct (a se vedea problema 3 la pag. 64).

4.2.2 Numărarea frunzelor în mulţimea arborilor binari

Să calculăm numărul frunzelor (vârfurilor terminale) în mulţimea arborilor binari cu n vârfuri. Să notăm acest număr cu f_n. De remarcat că rădăcina arborelui, chiar dacă are gradul 1, nu se consideră frunză. Se poate vedea uşor că $f_2 = 2$, $f_3 = 6$, iar pentru cazurile $n = 0$ şi $n = 1$ vom considera $f_0 = 0$, $f_1 = 1$. (Aceste valori vor fi justificate ulterior.)

Să considerăm arborii binari cu n vârfuri care au în subarborele stâng k vârfuri, iar în subarborele drept $n - k - 1$ vârfuri. În total sunt b_k astfel de subarbori stângi, al i-lea dintre aceştia are v_i frunze. Deci, dacă fixăm subarborele stâng (ca fiind al i-lea), arborele corespunzător are $v_i b_{n-1-k} + f_{n-1-k}$ frunze. Însumând după i de la 1 la b_k obţinem numărul frunzelor pentru toţi arborii care au un arborele stâng format din k vârfuri, acest număr fiind $b_{n-1-k} f_k + b_k f_{n-1-k}$. Atunci

$$f_n = \sum_{k=0}^{n-1} (f_k b_{n-1-k} + b_k f_{n-1-k})$$

Printr-un calcul simplu se obţine că

$$f_n = 2(f_0 b_{n-1} + f_1 b_{n-2} + \ldots f_{n-1} b_0) \quad \text{pentru} \quad n \geq 2. \tag{29}$$

Fie

$$F(z) = \sum_{n \geq 0} f_n z^n \quad \text{şi} \quad B(z) = \sum_{n \geq 0} b_n z^n.$$

Înmulţind în (29) cu z^n în ambii membri şi însumând după n, obţinem

$$\sum_{n \geq 2} f_n z^n = 2 \sum_{n \geq 2} \left(\sum_{k=0}^{n-1} f_k b_{n-1-k} \right) z^n$$

Ţinând cont de $f_0 = 0$ şi $f_1 = 1$, obţinem

$$F(z) - z = 2zF(z)B(z)$$

De unde

$$F(z) = \frac{z}{1 - 2zB(z)},$$

dar ştiind că

$$B(z) = \frac{1}{2z}\left(1 - \sqrt{1-4z}\,\right),$$

obţinem

$$F(z) = \frac{z}{\sqrt{1-4z}} = z(1-4z)^{-1/2} = z\sum_{n\geq 0}\binom{-\frac{1}{2}}{n}(-4z)^n$$

Făcând calculele, se obţine

$$F(z) = \sum_{n\geq 0}\binom{2n}{n}z^{n+1} = \sum_{n\geq 1}\binom{2n-2}{n-1}z^n$$

de unde se obţine

$$f_n = \binom{2n-2}{n-1} \quad \text{sau} \quad f_{n+1} = \binom{2n}{n} = (n+1)b_n.$$

Conform definiţiei combinărilor generalizate se obţin valorile corecte pentru f_0 şi f_1.

4.2.3 Numărarea arborilor binari cu n vârfuri şi k frunze

O problemă puţin mai dificilă este următoarea: câţi arbori binari cu n vârfuri există care au câte k frunze (vârfuri terminale) fiecare. Să notăm cu $b_n^{(k)}$ numărul arborilor binari cu n vârfuri şi k frunze. Se poate vedea uşor că $b_n^{(k)} = 0$ pentru $k > \lfloor (n+1)/2 \rfloor$. Printr-un raţionament simplu se poate calcula cazul particular $k = 1$, adică $b_n^{(1)} = 2^{n-1}$ pentru $n \geq 1$. Prin convenţie se poate considera $b_0^{(0)} = 1$. Ca şi la problemele anterioare, vom număra arborii binari în cauză prin a studia subarborele stâng, respectiv subarborele drept. Dacă în subarborele stâng avem i vârfuri cu j frunze, atunci în subarborele drept avem $n-i-1$ vârfuri cu $k-j$ frunze. Produsul $b_i^{(j)}b_{n-i-1}^{(k-j)}$ reprezintă numărul acestor arbori. Făcând suma după k şi j, obţinem următoarea formulă de recurenţă:

$$b_n^{(k)} = 2b_{n-1}^{(k)} + \sum_{i=1}^{n-2}\sum_{j=1}^{k-1} b_i^{(j)}b_{n-i-1}^{(k-j)}. \tag{30}$$

Să considerăm funcţia generatoare $B^{(k)}(z) = \sum_{n\geq 0} b_n^{(k)}z^n$ (pentru $k \geq 1$ suma se poate considera şi de la 1, deoarece $b_0^{(k)} = 0$). Să înmulţim ambii membri

ai formulei (30) cu z^n şi să le adunăm aceste egalităţi (pentru $n = 0, 1, 2, \ldots$) membru cu membru. Obţinem:

$$\sum_{n\geq 1} b_n^{(k)} z^n = 2\sum_{n\geq 1} b_{n-1}^{(k)} z^n + \sum_{n\geq 1}\left(\sum_{i=1}^{n-2}\sum_{j=1}^{k-1} b_i^{(j)} b_{n-i-1}^{(k-j)}\right) z^n$$

Schimbând ordinea de însumare, obţinem:

$$\sum_{n\geq 1} b_n^{(k)} z^n = 2\sum_{n\geq 1} b_{n-1}^{(k)} z^n + \sum_{j=1}^{k-1}\sum_{n\geq 1}\left(\sum_{i=1}^{n-2} b_i^{(j)} b_{n-i-1}^{(k-j)}\right) z^n$$

De unde se obţine:

$$B^{(k)}(z) = 2zB^{(k)}(z) + z\left(\sum_{j=1}^{k-1} B^{(j)}(z)B^{(k-j)}(z)\right)$$

sau

$$B^{(k)}(z) = \frac{z}{1-2z}\left(\sum_{j=1}^{k-1} B^{(j)}(z)B^{(k-j)}(z)\right) \tag{31}$$

Din aproape în aproape, obţinem:

$$B^{(2)}(z) = \frac{z}{1-2z}\Big(B^{(1)}(z)\Big)^2$$

$$B^{(3)}(z) = \frac{2z^2}{(1-2z)^2}\Big(B^{(1)}(z)\Big)^3$$

$$B^{(4)}(z) = \frac{5z^3}{(1-2z)^3}\Big(B^{(1)}(z)\Big)^4$$

Căutăm soluţia generală sub forma:

$$B^{(k)}(z) = \frac{c_k z^{k-1}}{(1-2z)^{k-1}}\Big(B^{(1)}(z)\Big)^k,$$

unde, după cum am văzut, avem $c_2 = 1$, $c_3 = 2$, $c_4 = 5$. După înlocuire în (31), obţinem următoarea relaţie de recurenţă pentru numerele c_k:

$$c_k = \sum_{i=1}^{k-1} c_i c_{k-i}.$$

Pentru $k = 2$ avem $c_2 = c_1 c_1$, de unde $c_1 = 1$. Să considerăm $c_0 = 1$. (Vom vedea mai târziu că această alegere este benefică.) Dacă $C(z) = \sum_{n\geq 0} c_n z^n$ este

funcţia generatoare a numerelor c_n, atunci, ţinând cont de produsul funcţiilor generatoare, avem:

$$C(z) - 1 - z = (C(z) - 1)^2 \quad \text{adică} \quad C^2(z) - 3C(z) + z + 2 = 0,$$

care după rezolvare în $C(z)$, ne dă:

$$C(z) = \frac{3 - \sqrt{1 - 4z}}{2} \qquad \text{(Se alege semnul minus pentru că C(0)=1.)}$$

După dezvoltarea în serie (a se vedea formula (28)), se obţine:

$$C(z) = \frac{3}{2} - \frac{1}{2}(1 - 4z)^{\frac{1}{2}} = \frac{3}{2} - \frac{1}{2}\sum_{n\geq 0}\frac{-1}{2n-1}\binom{2n}{n}z^n$$

$$= \frac{3}{2} + \sum_{n\geq 0}\frac{1}{2(2n-1)}\binom{2n}{n}z^n = 1 + \sum_{n\geq 1}\frac{1}{2(2n-1)}\binom{2n}{n}z^n.$$

De unde se obţine:

$$c_n = \frac{1}{2(2n-1)}\binom{2n}{n}, \quad \text{pentru } n \geq 1.$$

Tinând cont de faptul că $b_n^{(1)} = 2^{n-1}$ pentru $n \geq 1$, se poate verifica uşor că $B^{(1)} = \dfrac{z}{1-2z}$. Deci

$$B^{(k)}(z) = \frac{1}{2(2k-1)}\binom{2k}{k}\frac{z^{2k-1}}{(1-2z)^{2k-1}}$$

Ştiind că

$$\frac{1}{(1-z)^m} = \sum_{n\geq 0}\binom{n+m-1}{n}z^n,$$

avem

$$B^{(k)}(z) = \frac{1}{2(2k-1)}\binom{2k}{k}\sum_{n\geq 0}\binom{2k+n-2}{n}2^n z^{2k+n-1}$$

$$= \frac{1}{2(2k-1)}\binom{2k}{k}\sum_{n\geq 2k-1}\binom{n-1}{n-2k+1}2^{n-2k+1}z^n$$

De unde se obţine

$$b_n^{(k)} = \frac{1}{2k-1}\binom{2k}{k}\binom{n-1}{2k-2}2^{n-2k}$$

sau

$$b_n^{(k)} = \frac{1}{n}\binom{2k}{k}\binom{n}{2k-1}2^{n-2k}.$$

Numărul stărilor în sistemul cu camarazi. *Sistemul cu camarazi* (*buddy system*) reprezintă o metodă de alocare dinamică a memoriei calculatoarelor. Vom folosi formula de mai sus pentru a număra câte stări există în sistemul de alocare a memoriei cu camarazi. În sistemul de alocare cu camarazi se alocă blocuri de memorie cu dimensiunea 2^k, unde k este un număr natural oarecare. La început toată memoria formează un bloc de dimensiune o putere a lui 2. Dacă apare o cerere de memorie, se caută un bloc liber de dimensiune 2^k cât mai apropiată de cererea respectivă. Dacă un astfel de bloc nu există, se consideră un bloc liber de dimensiune 2^{k+1}, care se împarte în două blocuri egale, unul se declară liber, iar celălalt se alocă. Dacă toate blocurile de dimensiune 2^{k+1} sunt ocupate, se consideră un bloc disponibil de dimensiunea 2^{k+2}, care se împarte în două, unul se declară liber, celălalt se împarte din nou ş.a.m.d. Cele două blocuri care rezultă din împărţirea unui bloc se numesc camarazi. Două blocuri libere alăturate se pot unifica într-un singur, dacă ele sunt camarazi. Numim o k-stare a memoriei o configuraţie a ei în care există în total k blocuri (alocate sau libere).

Numărul k-stărilor se poate determina cu ajutorul numerelor $b_n^{(k)}$. Să notăm cu S_k numărul k-stărilor într-un sistem cu camarazi. Vom folosi noţiunea de arbore binar regular, în care fiecare vârf interior are exact doi descendenţi. Dacă dintr-un arbore binar regular cu $k+1$ frunze ştergem toate frunzele, obţinem un arbore binar cu k vârfuri, numărul acestora este, după cum se ştie, $\frac{1}{k+1}\binom{2k}{k}$.

Pentru ca formulele să fie mai simple, vom calcula S_{k+1}. Dacă în a i-a $(k+1)$-stare p_i reprezintă numărul perechilor de camarazi, iar n_i numărul celorlalte blocuri, atunci $2p_i + n_i = k+1$. Dacă notăm cu $s = \frac{1}{k+1}\binom{2k}{k}$, avem

$$S_{k+1} = \sum_{i=1}^{s} 2^{n_i} 3^{p_i},$$

deoarece în cazul camarazilor nu putem avea situaţia ca ambele să fie disponibile, pentru că în acest caz ele se unesc într-un bloc mai mare (deci pentru fiecare pereche există trei posibilităţi: numai primul alocat, numai cel de al doilea alocat, ambele alocate). De unde:

$$S_{k+1} = 2^{k+1} \sum_{i=1}^{s} \left(\frac{3}{4}\right)^{p_i}.$$

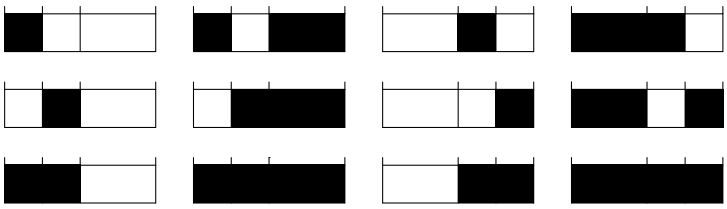
Dacă eliminăm toate frunzele ale unui arbore binar regular cu $2k+1$ vârfuri, obţinem un arbore binar (obişnuit) cu k vârfuri. Dacă le punem înapoi frunzele ca să obţinem din nou arborele regular iniţial, fiecărei frunze din arbore binar îi corespunde o pereche de frunze (camarazi) în arborele binar regular. Deci

$$S_{k+1} = 2^{k+1} \sum_{i=1}^{\lfloor \frac{k+1}{2} \rfloor} b_k^{(i)} \left(\frac{3}{4}\right)^i.$$

Printr-un calcul simplu [37] se obţine:

$$S_{k+1} = \frac{3 \cdot 2^{2k-2}}{k} \sum_{i=0}^{\lfloor \frac{k-1}{2} \rfloor} \binom{k}{i} \binom{k-i}{i+1} 2^{-4i} 3^i, \quad \text{pentru } k \geq 1.$$

Avem: $S_2 = 3, S_3 = 12, S_4 = 57$ etc. Pentru $k = 3$ avem următoarele stări posibile:



Se poate deduce şi formula recursivă:

$$S_{k+1} = \frac{4}{k+1}\Big((2k-1)S_k - (k-2)S_{k-1}\Big), \quad \text{pentru } k \geq 3.$$

Forma poloneză postfixată. Pentru evaluarea expresiilor algebrice în timpul compilării (mai precis în timpul generării codului) este foarte potrivită forma poloneză postfixată (precum şi forma poloneză prefixată) care reprezintă expresia fără a folosi paranteze.

În forma poloneză postfixată în loc de operaţia $a \circ b$ (unde a, b sunt operanzii şi $\circ$ operaţia binară) se scrie $ab\circ$. La fel în forma poloneză prefixată, unde se scrie $\circ ab$ în loc de $a \circ b$.

Expresia $a + (b + a * c) * b$ se reprezintă prin

$$+a * +b * acb$$

în forma poloneză prefixată, şi prin

$$abac * +b * +$$

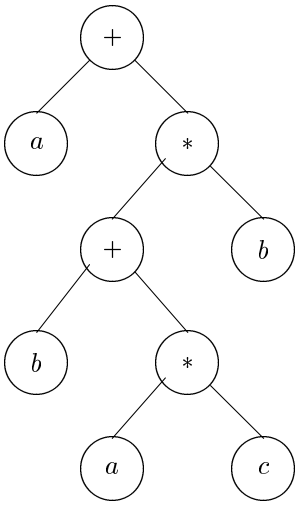
în forma poloneză postfixată.

Adică

$$\underbrace{+a * \underbrace{+b \underbrace{*ac}\, b}}\qquad \underbrace{a\, b\, \underbrace{ac*}\, +b*}+$$

Expresiile pot fi reprezentate uşor cu ajutorul arborilor binari, iar diferite parcurgeri în arbore corespund la diferite forme ale expresiilor.

Parcurgerea în preordine ne dă forma poloneză prefixată a expresiei iar parcurgerea în postordine ne dă forma poloneză postfixată. Parcurgerea în ordine ne dă expresia în sine (fără paranteze).



parcurgere în postordine

$$abac*+b*+$$

parcurgere în preordine

$$+a*+b*acb$$

parcurgerea în ordine (unde am pus şi parantezele)

$$a+(b+a*c)*b$$

Scrierea poloneză se poate extinde şi la anumite instrucţiuni din limbajele de programare.

4.2.4 Demonstrarea unor identităţi

Funcţiile generatoare pot fi utilizate pentru demonstrarea unor identităţi, care altfel se demonstrează destul de greu.

Prima idee este de a dezvolta o funcţie generatoare în două moduri diferite şi pe urmă de a egala coeficienţii lui z^n din ambele dezvoltări.

Considerăm următoarea funcţie generatoare:

$$\frac{1}{\sqrt{1-4z}} = (1-4z)^{-\frac{1}{2}} = \sum_{n\geq 0} \binom{-\frac{1}{2}}{n} (-4z)^n =$$

$$= \sum_{n\geq 0} (-1)^n \binom{-\frac{1}{2}}{n} 2^{2n} z^n = \sum_{n\geq 0} \binom{2n}{n} z^n$$

Folosind această formulă la dezvoltarea funcţiei $A(z) = \dfrac{z}{\sqrt{1-4z}}$ în două moduri diferite, vom obţine o formulă interesantă.

Conform expresiei de mai sus avem:

$$A(z) = \sum_{n\geq 0} a_n z^n = \frac{z}{\sqrt{1-4z}} = \sum_{n\geq 0} \binom{2n}{n} z^{n+1} = \sum_{n\geq 1} \binom{2n-2}{n-1} z^n,$$

de unde obţinem $a_n = \binom{2n-2}{n-1}$.

Pe de altă parte

$$A(z) = \frac{z}{\sqrt{1-4z}} = \frac{z\sqrt{1-4z}}{1-4z} = \frac{z}{1-4z}\sqrt{1-4z}$$

Avem:

$$\frac{z}{1-4z} = z\sum_{n\geq 0}(4z)^n = \sum_{n\geq 0}4^n z^{n+1} = \sum_{n\geq 1}4^{n-1}z^n \tag{32}$$

şi formula (28):

$$\sqrt{1-4z} = \sum_{n\geq 0}\frac{-1}{2n-1}\binom{2n}{n}z^n \tag{33}$$

Prin înmulţirea celor două funcţii din (32) şi (33) obţinem:

$$A(z) = \left(\sum_{n\geq 1}4^{n-1}z^n\right)\left(\sum_{n\geq 0}\frac{-1}{2n-1}\binom{2n}{n}z^n\right)$$

De unde, ţinând cont de formula produsului funcţiilor generatoare, obţinem:

$$a_n = \sum_{k=0}^{n-1}4^{n-1-k}\cdot\frac{-1}{2k-1}\binom{2k}{k}$$

Egalând cele două expresii ale lui a_n, obţinem:

$$-4^{n-1}\sum_{k=0}^{n-1}\frac{1}{4^k(2k-1)}\binom{2k}{k} = \binom{2n-2}{n-1}$$

care poate fi scrisă şi în felul următor (înlocuind $n-1$ cu n):

$$\sum_{k=1}^{n}\frac{4^{n-k}}{2k-1}\binom{2k}{k} = 4^n - \binom{2n}{n} \qquad \text{pentru } n\geq 1.$$

Pentru demonstarea unor identităţi combinatoriale se poate folosi şi metoda inversării ordinii de însumare în formule ([42], [70]).

Să începem cu o identitate simplă. Să demonstrăm că:

$$\sum_{k\geq 0}\binom{n}{2k+1} = 2^{n-1}$$

Se formează funcţia generatoare care are ca termen general membrul stâng al identităţii de mai sus, se schimbă ordinea de însumare, se aplică formula (24) şi se obţine:

$$\sum_{n\geq 0}\left(\sum_{k\geq 0}\binom{n}{2k+1}z^n\right)=\sum_{k\geq 0}\left(\sum_{n\geq 0}\binom{n}{2k+1}z^n\right)=\sum_{k\geq 0}\frac{z^{2k+1}}{(1-z)^{2k+2}}$$

$$=\frac{z}{(1-z)^2}\sum_{k\geq 0}\left(\frac{z^2}{(1-z)^2}\right)^k=\frac{z}{(1-z)^2}\cdot\frac{1}{1-\frac{z^2}{(1-z)^2}}$$

$$=\frac{z}{1-2z}=z\frac{1}{1-2z}=z(1+2z+(2z)^2+\ldots+(2z)^k+\ldots)$$

Coeficentul lui z^n în dezvoltarea de mai sus este egal cu 2^{n-1}, ceea ce demonstrează identitatea. Exact la fel se poate demonstra şi identitatea următoare (care de fapt se poate obţine şi direct, ţinând cont de faptul că suma coeficienţilor binomiali este egală cu 2^n):

$$\sum_{k\geq 0}\binom{n}{2k}=2^{n-1}$$

Să demonstrăm o identitate mai complexă ([70]):

$$\sum_{k\geq 0}\binom{n+k}{m+2k}\binom{2k}{k}\frac{(-1)^k}{k+1}=\binom{n-1}{m-1},\text{ pentru } m,n \text{ întregi nenegative.}$$

Considerăm funcţia generatoare pentru care termenul general în n este membrul stâng al identităţii (considerăm m fixat):

$$A(z)=\sum_{n\geq 0}\left(\sum_{k\geq 0}\binom{n+k}{m+2k}\binom{2k}{k}\frac{(-1)^k}{k+1}\right)z^n$$

$$=\sum_{k\geq 0}\binom{2k}{k}\frac{(-1)^k}{k+1}z^{-k}\sum_{n\geq 0}\binom{n+k}{m+2k}z^{n+k}$$

Pentru ultima sumă se foloseşte formula (24), şi se obţine

$$A(z)=\sum_{k\geq 0}\binom{2k}{k}\frac{(-1)^k}{k+1}z^{-k}\frac{z^{m+2k}}{(1-z)^{m+2k+1}}$$

$$=\frac{z^m}{(1-z)^{m+1}}\sum_{k\geq 0}\binom{2k}{k}\frac{1}{k+1}\left(\frac{-z}{(1-z)^2}\right)^k$$

Ţinând cont de formula (27) obţinem:

$$A(z) = \frac{z^m}{(1-z)^{m+1}} \cdot \frac{(1-z)^2}{-2z} \left(1 - \sqrt{1 + \frac{4z}{(1-z)^2}}\right)$$

Făcând calculele se ajunge la

$$A(z) = \frac{z^m}{(1-z)^m} = z\frac{z^{m-1}}{(1-z)^m} \tag{34}$$

Membrul stâng al identităţii originale va fi egal cu coeficientul lui z^n în (34), care se obţine ţinând cont de formula (24) (de fapt, din cauza lui z din faţa fracţiei, se ia coeficientul lui z^{n-1} din dezvoltarea fracţiei $\frac{z^{m-1}}{(1-z)^m}$, astfel acest coeficient este $\binom{n-1}{m-1}$.

Se pot obţine identităţi şi prin utilizarea unor formule de recurenţe. De exemplu, dacă se înlocuieşte $b_n = \frac{1}{n+1}\binom{2n}{n}$ (numărul arborilor binari cu n vârfuri) în formula (25), se obţine:

$$\frac{1}{n+1}\binom{2n}{n} = \sum_{i=0}^{n-1} \frac{1}{i+1}\binom{2i}{i}\frac{1}{n-i}\binom{2n-2i-2}{n-i-1},$$

adică

$$\sum_{i=1}^{n} \frac{1}{(n-i+1)i}\binom{2i-2}{i-1}\binom{2n-2i}{n-i} = \frac{1}{n+1}\binom{2n}{n}, \quad \text{pentru } n \geq 1.$$

Asemănător, dacă se înlocuiesc valorile pentru b_n (numărul arborilor binari cu n vârfuri) şi f_n (numărul frunzelor în mulţimea arborilor binari cu n vârfuri) în formula (29), se obţine următoarea identitate:

$$\sum_{k=1}^{n} \frac{1}{k}\binom{2k-2}{k-1}\binom{2n-2k}{n-k} = \frac{1}{2}\binom{2n}{n}, \quad \text{pentru } n \geq 1.$$

Dacă ţinem cont de faptul că numărul f_n al frunzelor în mulţimea arborilor binari cu n vârfuri este egal cu suma $kb_n^{(k)}$ pentru $k = 1, 2 \ldots$, adică

$$f_n = \sum_{k=1}^{\lfloor \frac{n+1}{2} \rfloor} kb_n^{(k)},$$

obţinem identitatea:

$$\binom{2n-2}{n-1} = \sum_{k=1}^{\lfloor \frac{n+1}{2} \rfloor} \frac{k}{n} \binom{n}{2k-1} \binom{2k}{k} 2^{n-2k}$$

sau

$$\sum_{k\geq 1} \frac{k}{2^{2k}} \binom{n}{2k-1} \binom{2k}{k} = \frac{n}{2^n} \binom{2n-2}{n-1} \quad \text{pentru } n \geq 1.$$

Câteva funcţii generatoare mai uzuale

$$\sum_{n\geq 0} z^n = 1 + z + z^2 + z^3 + \ldots = \frac{1}{1-z}$$

$$\sum_{n\geq 0} (-1)^n z^n = 1 - z + z^2 - z^3 + \ldots = \frac{1}{1+z}$$

$$\sum_{n\geq 0} \binom{n+p}{p} z^n = \binom{p}{p} + \binom{p+1}{p} z + \binom{p+2}{p} z^2 + \ldots = \frac{1}{(1-z)^{p+1}}, \quad p \in \mathbf{N}$$

$$\sum_{n\geq 1} \frac{1}{n} z^n = z + \frac{1}{2} z^2 + \frac{1}{3} z^3 + \ldots = \ln \frac{1}{1-z}$$

$$\sum_{n\geq 0} z^{2n} = 1 + z^2 + z^4 + z^6 \ldots = \frac{1}{1-z^2}$$

$$\sum_{n\geq 0} \binom{r}{n} z^n = 1 + \binom{r}{1} z + \binom{r}{2} z^2 + \binom{r}{3} z^3 \ldots = (1+z)^r, \quad r \in \mathbf{R}$$

$$\sum_{n\geq 0} \binom{n}{p} z^n = \binom{p}{p} z^p + \binom{p+1}{p} z^{p+1} + \binom{p+2}{p} z^{p+2} + \ldots = \frac{x^p}{(1+x)^{p+1}}, \quad p \in \mathbf{N}$$

$$\sum_{n\geq 0} \binom{2n}{n} z^n = 1 + \binom{2}{1} z + \binom{4}{2} z^2 + \binom{6}{3} z^3 + \ldots = \frac{1}{\sqrt{1-4z}}$$

$$\sum_{n\geq 0} \frac{-1}{2n-1} \binom{2n}{n} z^n = 1 - \binom{2}{1} z - \frac{1}{3} \binom{4}{2} z^2 - \frac{1}{5} \binom{6}{3} z^3 - \ldots = \sqrt{1-4z}$$

$$\sum_{n\geq 0} \frac{1}{n+1} \binom{2n}{n} z^n = 1 + \frac{1}{2} \binom{2}{1} z + \frac{1}{3} \binom{4}{2} z^2 + \frac{1}{4} \binom{6}{3} z^3 + \ldots = \frac{1-\sqrt{1-4z}}{2z}$$

$$\sum_{n\geq 0} 2^n z^n = 1 + 2z + 2^2 z^2 + 2^3 z^3 + \ldots = \frac{1}{1-2z}$$

4.2.5 Probleme de partiţionare

Cu ajutorul funcţiilor generatoare se pot rezolva şi problemele de partiţionare a unui număr natural în sume de numere naturale.

Să pornim de la produsul următor de funcţii generatoare:

$$\frac{1}{1-z}\cdot\frac{1}{1-z^2}\cdot\frac{1}{1-z^3}\cdots\frac{1}{1-z^n}$$
$$=(1+z+z^2+z^3\ldots)(1+z^2+z^4+z^6\ldots)\ldots(1+z^n+z^{2n}+z^{3n}\ldots)$$
$$=\sum_{k_1\geq 0}\sum_{k_2\geq 0}\cdots\sum_{k_n\geq 0}z^{k_1+2k_2+\ldots nk_n}$$

În această dezvoltare coeficientul lui z^n reprezintă numărul în câte moduri poate fi descompus n în sumă de numere naturale mai mici decât el. Exponentul $k_1+2k_2+\ldots nk_n=n$ ne arată că în descompunere apare 1 de k_1 ori, 2 de k_2 ori ş.a.m.d. Evident că dacă k_n este 1 atunci toate celelalte k_i-uri trebuie să fie 0. De exemplu pentru $n=3$ avem următoarele descompuneri $3,\ 1+2,\ 1+1+1$ (adică $3=0\cdot 1+0\cdot 2+1\cdot 3,\ \ 3=1\cdot 1+1\cdot 2+0\cdot 3$, $3=3\cdot 1+0\cdot 2+0\cdot 3$). Dacă $k_1+k_2+\ldots k_n=m$, atunci în descompunerea numărului n se face folosind m numere.

Dacă în descompunere fiecare număr poate să apară de cel mult o dată, atunci funcţia generatoare corespunzătoare este:

$$(1+z)(1+z^2)(1+z^3)\ldots(1+z^n).$$

Dacă dorim ca în descompunerea lui n să apară numerele $n_1,n_2,\ldots n_k$, atunci funcţia generatoare este:

$$\frac{1}{1-z^{n_1}}\frac{1}{1-z^{n2}}\cdots\frac{1}{1-z^{n_2}}.$$

Probleme.

1. Să se demonstreze formula

$$\sum_{k\geq 0}\binom{n+k-1}{k}z^k=\frac{1}{(1-z)^n}$$

2. Să se demonstreze cu ajutorul funcţiilor generatoare că

$$\overline{C}_n^k=\binom{n+k-1}{k},$$

unde $\overline{C}_n^k$ reprezintă combinările cu repetiţii de n elemente luate câte k.

3. Să se demonstreze formula:

$$\sqrt{1-4z} = \sum_{n\geq 0} \frac{-1}{2n-1}\binom{2n}{n} z^n$$

4. Să se demonstreze formula:

$$\sum_{k=0}^{n} \binom{2k}{k}\binom{2n-2k}{n-k} = 2^{2n}$$

5. Să se demonstreze formula:

$$\sum_{k=0}^{n} \frac{1}{2k-1}\binom{2k}{k}\binom{2n-2k}{n-k} = -\delta_{n0}$$

5 Automatizarea demonstrării identităţilor

Uneori suntem puşi în faţa situaţiei de a calcula o sumă finită sau infinită. Vom prezenta în continuare anumite tehnici care prin natura lor uşor pot fi mecanizate, şi deci putem folosi calculatorul. Există programe care pot face şi calcule simbolice, cum ar fi Mathematica sau Maple. Aceste programe ne porûşura munca. Următoarele metode se pot folosi în cazul seriilor hipergeometrice. O serie $\sum_{k\geq 0} t_k$ este hipergeometrică dacă raportul a doi termeni consecutiv este o expresie raţională.

5.1 Metoda Sorei Celine

Să prezentăm metoda Sorei Celine (numele ei complet este Mary Celine Fasenmyer) care se găseşte descrisă în [57]. Calculele deobicei fiind laborioase, încercăm să facem o prezentare pe un exemplu simplu. Presupunem că avem de calculat suma[1]:

$$f(n) = \sum_k \binom{n}{k}.$$

Să-l notăm termenul de însumat prin $F(n,k)$, deci

$$F(n,k) = \binom{n}{k},$$

şi să încercăm să găsim o formulă de recurenţă de tipul

$$a(n)F(n,k) + b(n)F(n-1,k) + c(n)F(n,k-1) + d(n)F(n-1,k-1) = 0,$$

unde coeficienţii $a(n), b(n), c(n), d(n)$ depind numai de n nu şi de k. În calculele următoare, pentru a simplifica scrierea, omitem scrierea argumentului n. După ce împărţim cu $F(n,k)$, obţinem:

$$a + b\frac{F(n-1,k)}{F(n,k)} + c\frac{F(n,k-1)}{F(n,k)} + d\frac{F(n-1,k-1)}{F(n,k)} = 0,$$

de unde după un calcul direct se obţine

$$a + b\frac{n-k}{n} + c\frac{k}{n-k+1} + d\frac{k}{n} = 0.$$

[1] Dacă într-o formulă apare numai variabila de însumare, fără specificarea limitelor, atunci însumarea se consideră de la 0 la ∞.

După aducerea la numitor comun, obţinem:

$$an(n-k+1)+b(n-k)(n-k+1)+cnk+dk(n-k+1)=0,$$

expresie care se mai poate scrie:

$$k^2\ (b-d)+$$

$$+k\ (-an-2bn-b+cn+dn+d)+$$

$$+\ (an^2+an+bn^2+bn)=0,$$

care trebuie să fie egală cu 0 pentru orice k, deci ajungem la următorul sistem de ecuaţii:

$$\begin{cases} b-d=0 \\ -an-2bn+cn+dn-b+d=0 \\ an^2+bn^2+an+bn=0 \end{cases}$$

Înlocuind $d=b$ în ecuaţia a doua, obţinem $c=a+b$. Din ecuaţia a treia obţinem $(n^2+1)(a+b)=0$, deci $b=-a$, $c=0$, iar $d=-a$. După înlocuirea în relaţia de recurenţă originală, obţinem:

$$a(n)F(n,k)-a(n)F(n-1,k)-a(n)F(n-1,k-1)=0,$$

de unde:

$$F(n,k)-F(n-1,k)-F(n-1,k-1)=0.$$

Adunând aceste relaţii pentru toţi k şi ţinând cont de faptul că

$$\sum_k F(n,k)=\sum_k F(n,k-1),$$

onţinem

$$f(n)-f(n-1)-f(n-1)=0.$$

Deci

$$f(n)=2f(n-1)$$

Dar, prin calcul direct se poate vedea că $f(0)=1$, şi deci $f(n)=2^n$.

Metoda generală a Sorei Celine se poate aplica la găsirea unei formule pentru sumele de forma: $f(n) = \sum_k F(n,k)$, unde

$$\frac{F(n+1,k)}{F(n,k)} \quad \text{şi} \quad \frac{F(n,k+1)}{F(n,k)}$$

sunt ambele raţionale atât în k cât şi în n. Încercăm să găsim o formulă de recurenţă de forma

$$\sum_{i=0}^{I} \sum_{j=0}^{J} a_{ij}(n) F(n-i, k-j) = 0. \tag{35}$$

Paşii metodei Sorei Celine sunt:

1. Fixăm valorile lui I şi J, la început ambele sunt egale cu 1.
2. Scriem formula de recurenţă respectivă (cf. formulei (35))
3. Împărţim toţi membrii formulei cu $F(n,k)$ şi calculăm şi simplificăm toate fracţiile $F(n-i,k-j)/F(n,k)$.
4. Aducem fracţiile la numitor comun, şi aranjăm numărătorul sub forma unui polinom în k.
5. Scriem sistemul de ecuaţii prin anularea fiecărui coeficient al polinomului.
6. Dacă sistemul de ecuaţii are soluţii nenule, am găsit o formulă de recurenţă. Dacă nu există soluţie nenulă, continuăm cu alte valori pentru I şi J.

Calculele fiind mecanice, ne putem folosi de calculator. În sistemele Mathematica şi Maple sunt posibilităţi de a rezolva probleme nenumerice. În Maple există un packet numit EKHAD[2] care rezolvă problema găsirii unei relaţii de recurenţă. Apelul procedurii `Celine` se face după citirea pachetul EKHAD prin

```
read EKHAD,
```

sub forma

```
celine((n,k)-> F(n,k),I,J).
```

[2]Pachetul EKHAD pentru Maple se poate obţine gratuit de la următoarele adrese:
`http://www.central.cis.upenn.edu/~wilf/AeqB.html` sau
`http://www.math.temple.edu/~zeilberg`

Să continuăm ilustrarea utilizării algoritmului şi a pachetului EKHAD cu o problemă mai complexă. Să calculăm suma

$$f(n) = \sum_k \binom{n}{k}^2$$

Se notează

$$F(n,k) = \sum_k \binom{n}{k}^2$$

şi se încearcă obţinerea uneui formule recursive pentru $I = 1, J = 1$:

```
celine((n,k)-> binomial(n,k)*binomial(n,k),1,1);

The full  recurrence is
0, '==0'
```

Nu obţinem soluţie, deci trebuie să modificăm valorile pentru I şi J. Modificăm I în 2 şi încercăm din nou.

```
> celine((n,k)-> binomial(n,k)*binomial(n,k),2,1);

The full  recurrence is
0, '==0'
```

Iar nu s-a ajums la soluţie. Modificăm şi J în 2.

```
> celine((n,k)-> binomial(n,k)*binomial(n,k),2,2);

The full  recurrence is
-b[8]*(-n+1)*F(n-2,k-2)-(2*n-2)*b[8]*F(n-2,k-1)
-(2*n-1)*b[8]*F(n-1,k-1)-b[8]*(-n+1)*F(n-2,k)
-(2*n-1)*b[8]*F(n-1,k)+b[8]*n*F(n,k), '==0'
```

Deci am obţinut următoarea recurenţă (după omiterea constantei $b[8]$.):

$$(n-1)F(n-2,k-2) - (2n-2)F(n-2,k-1) - (2n-1)F(n-1,k-1) +$$

$$+(n-1)F(n-2,k) - (2n-1)F(n-1,k) + nF(n,k) = 0$$

sau, după însumare pentru toţi k:

$$(n-1)f(n-2)-(2n-2)f(n-2)-(2n-1)f(n-1)+(n-1)f(n-2)-$$

$$-(2n-1)f(n-1)+nf(n)=0$$

deci

$$\Big((n-1)-(2n-2)+(n-1)\Big)f(n-2)-2(2n-1)f(n-1)+nf(n)=0$$

adică

$$f(n)=\frac{2(2n-1)}{n}f(n-1)$$

De unde

$$f(n)=\frac{2(2n-1)}{n}\cdot\frac{2(2n-3)}{n-1}\cdot\frac{2(2n-5)}{n-2}\cdots\frac{2\cdot 1}{1}=$$

$$=\frac{2^n(2n-1)!!}{n!}\cdot\frac{n!}{n!}=\frac{(2n)!}{n!n!}=\binom{2n}{n}.$$

adică obţinem formula cunoscută

$$\sum_k\binom{n}{k}^2=\binom{2n}{n}$$

Următorul exemplu ne arată că uneori trebuie să folosim tehnica de mai sus cu precauţie. Să calculăm suma:

$$f(n)=\sum_{k\geq 0}\frac{1}{k+1}\binom{n}{k}.$$

Folosim notaţia

$$F(n,k)=\frac{1}{k+1}\binom{n}{k}.$$

Folosind procedura `celine` din pachetul EKHAD, obţinem următoarea relaţie de recurenţă:

$$(n+1)F(n,k)-nF(n-1,k)-nF(n-1,k-1)=0$$

Însumarea se poate face numai pentru $k\geq 1$ din cauza ultimului termen din membrul stâng, şi ştiind că

$$f(n)=1+\sum_{k\geq 1}\frac{1}{k+1}\binom{n}{k}$$

obţinem:

$$(n+1)\Big(f(n)-1\Big)-n\Big(f(n-1)-1\Big)-nf(n)-0,$$

ceea ce ne dă

$$f(n)=\frac{2n}{n+1}f(n-1)+\frac{1}{n+1}.$$

Prin calcul direct se ajunge la

$$f(n)=\frac{2^{n+1}-1}{n+1}.$$

5.2 Metoda WZ

Metoda WZ (după Wilf şi Zeilberger) ([70], [57]) se poate folosi la demonstrarea unor identităţi de forma $\sum_k t(n,k)=d(n)$ unde termenii membrului stâng sunt nenuli numai pentru un număr finit de valori ale lui k.

Paşii algoritmului sunt:

1. Se transformă formula de demonstrat, prin împărţire cu membrul drept, în forma

$$\sum_k F(n,k)=1.$$

2. Se caută o funcţie raţională $R(n,k)$ în două variabile numită *funcţie certificat* şi se formează $G(n,k)=R(n,k)F(n,k)$. Expresia lui $R(n,k)$ deobicei se determină cu ajutorul calculatorului.

3. Se verifică dacă are loc

$$F(n-1,k)-F(n,k)=G(n,k)-G(n,k-1). \tag{36}$$

Dacă da, atunci se verifică dacă $\sum_k F(0,k)=1$. Dacă ambele sunt adevărate, atunci formula de demonstrat este adevărată.

Cum se justifică aceste afirmaţii? Se poate observa uşor că

$$\sum_k G(n,k)=\sum_k G(n,k-1).$$

Dar atunci, dacă formula (36) are loc, avem

$$\sum_k F(n-1,k)=\sum_k F(n,k),$$

ceea ce demonstrează că $\sum_k F(n,k)$ nu depinde de n, deci este constantă, care se poate determina luând $n = 0$.

Să ilustrăm metoda WZ pe un exemplu simplu, să demonstrăm identitatea cunoscută:

$$\sum_k \binom{n}{k} = 2^n.$$

Deducem că

$$F(n,k) = 2^{-n}\binom{n}{k}.$$

Fie

$$R(n,k) = \frac{n-k}{n},$$

atunci

$$G(n,k) = \frac{1}{2^n}\binom{n}{k}\frac{n-k}{n} = \frac{1}{2^n}\binom{n-1}{k}.$$

După un calcul simplu:

$$G(n,k) - G(n,k-1) = \frac{1}{2^n}\binom{n-1}{k} - \frac{1}{2^n}\binom{n-1}{k-1} =$$

$$= \frac{1}{2^n}\cdot\frac{(n-1)!}{(k-1)!(n-k-1)!}\left(\frac{1}{k} - \frac{1}{n-k}\right) =$$

$$= \frac{1}{2^n}\cdot\frac{(n-1)!}{k!(n-k)!}(n-2k) = \frac{n-2k}{n2^n}\binom{n}{k}.$$

Dar

$$F(n-1,k) - F(n,k) = \frac{1}{2^{n-1}}\binom{n-1}{k} - \frac{1}{2^n}\binom{n}{k} =$$

$$= \frac{1}{2^n}\frac{(n-1)!}{k!(n-1-k)!}\left(2 - \frac{n}{n-k}\right) = \frac{n-2k}{n2^n}\binom{n}{k}$$

Iar $F(0,k) = 1$ pentru $k = 0$, în rest este 0. Deci $\sum_k F(0,k) = 1$, ceea ce demonstrează formula.

Să vedem cum am putea rezolva problema pusă într-o sesiune Maple. Folosim pachetul EKHAD pentru a determina funcţia certificat.

```
> read EKHAD;
> celine((n,k)->binomial(n,k)*2^(-n),1,1);
```

ne dă funcţia certificat $R(n,k) = \frac{n-k}{n}$. În continuare definim funcţiile $F(n,k)$, $R(n,k)$ şi $G(n,k)$. Pe urmă calculăm valoarea expresiei $F(n-1,k) - F(n,k) - G(n,k) + G(n,k-1)$.

```
> F:=(n,k)->binomial(n,k)*2^(-n);
> R:=(n,k)->(n-k)/n;
> G:=(n,k)->R(n,k)*F(n,k);
> simplify(expand(F(n-1,k)-F(n,k)-G(n,k)+G(n,k-1)));
```

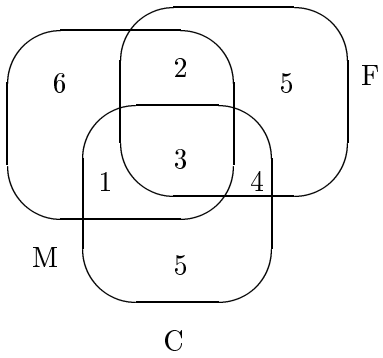
Rezultatul este 0, ceea ce demostrează identitatea.

6 Principiul includerii şi al excluderii

Să rezolvăm următoarea problemă. *Într-o clasă de 30 elevi sunt 12 elevi cărora le place matematica, 14 cărora le place fizica şi 13 cărora le place chimia. Dintre aceştia sunt 5 cărora le place şi matematica şi fizica, 4 cărora le place şi matematica şi chimia şi 7 cărora le place şi fizica şi chimia. În total există 3 elevi cărora le plac toate cele trei din materiile amintite. Întrebarea este: câţi elevi sunt în clasă cărora nu le plac nici una din cele trei materii?*

În figura alăturată am desenat cele trei grupe de elevi: M – cărora le place matematica, F – cărora le place fizica şi C – cărora le place chimia. În porţiunea comună a celor trei grupe apare 3, adică numărul celor cărora le plac toate cele trei discipline. Partea comună dintre M şi F (2+3) reprezintă pe cei 5 elevi cărora le place şi matematica dar şi fizica. Adunând numerele aflate pe figură obţinem 26, adică numărul celor cărora le place cel puţin una din cele trei materii. Deci sunt 4 elevi $(30-26)$ cărora nu le place nici matematica, nici fizica şi nici chimia.

Acest număr s-ar fi putut obţine şi în felul următor. Scădem din 30 suma celor cărora le place câte o materie (12+14+13), dar astfel scăzând de două ori pe cei cărora le plac două materii, le adunăm înapoi (5+4+7), din care pe urmă scădem 3 (care a fost adăugat în plus), adică numărul celor cărora nu le plac nici una din cele trei materii este egal cu $30-(12+14+13)+(5+4+7)-3=4$.



Generalizând, obţinem următoarea problemă. Se dă o mulţime finită A şi submulţimile $A_1, A_2, \ldots, A_n$ ale sale. Să notăm cu $\#A$ cardinalul mulţimii A (adică numărul elementelor sale). Atunci numărul elementelor lui A care nu apar în nici una din submulţimile A_i $(i=1,2,\ldots,n)$ este egal cu

$$\#A-\sum_{i=1}^{n}\#A_i+\sum_{1\le i<j\le n}\#(A_i\cap A_j)-\sum_{1\le<i<j<k\le n}\#(A_i\cap A_j\cap A_k)+\ldots$$

$$\ldots+(-1)^n\#(A_1\cap A_2\cap\ldots\cap A_n) \tag{37}$$

În continuare vom folosi notaţiile uzuale:

$$\bigcup_{i=1}^{n}A_i=A_1\cup A_2\cup\ldots\cup A_n,\quad \bigcap_{i=1}^{n}A_i=A_1\cap A_2\cap\ldots\cap A_n.$$

Se pot demonstra prin inducţie matematică completă şi următoarele formule:

$$\#\left(\bigcup_{i=1}^{n} A_i\right) = \sum_{i=1}^{n} \#A_i - \sum_{1\le i<j\le n} \#(A_i \cap A_j) + \ldots + (-1)^{n+1} \#\left(\bigcap_{i=1}^{n} A_i\right) \quad (38)$$

$$\#\left(\bigcap_{i=1}^{n} A_i\right) = \sum_{i=1}^{n} \#A_i - \sum_{1\le i<j\le n} \#(A_i \cup A_j) + \ldots + (-1)^{n+1} \#\left(\bigcup_{i=1}^{n} A_i\right) \quad (39)$$

Demonstraţia formulei (38):

Vom folosi metoda inducţiei matematice. Pentru $n = 2$ avem formula evidentă

$$\#(A_1 \cup A_2) = \#A_1 + \#A_2 - \#(A_1 \cap A_2). \quad (40)$$

Conform ipotezei inducţiei avem:

$$\#\left(\bigcup_{i=1}^{n-1} A_i\right) = \sum_{i=1}^{n-1} \#A_i - \sum_{1\le i<j\le n-1} \#(A_i \cap A_j) + \ldots + (-1)^n \#\left(\bigcap_{i=1}^{n-1} A_i\right)$$

Vom folosi descompunerea $\bigcup_{i=1}^{n} A_i = \left(\bigcup_{i=1}^{n-1} A_i\right) \cup A_n$ şi vom aplica formula (40):

$$\#\left(\bigcup_{i=1}^{n} A_i\right) = \#\left(\left(\bigcup_{i=1}^{n-1} A_i\right) \cup A_n\right) = \#\left(\bigcup_{i=1}^{n-1} A_i\right) + \#A_n - \#\left(\left(\bigcup_{i=1}^{n-1} A_i\right) \cap A_n\right) =$$

$$= \sum_{i=1}^{n-1} \#A_i - \sum_{1\le i<j\le n-1} \#(A_i \cap A_j) + \ldots + (-1)^n \#\left(\bigcap_{i=1}^{n-1} A_i\right) +$$

$$+ \#A_n - \#\left(\left(\bigcup_{i=1}^{n-1} A_i\right) \cap A_n\right) \quad (41)$$

Ultima expresie se mai poate scrie:

$$\left(\bigcup_{i=1}^{n-1} A_i\right) \cap A_n = \bigcup_{i=1}^{n-1} (A_i \cap A_n),$$

care după înlocuire în (41) (ţinând cont şi de ipoteza inducţiei) şi regruparea termenilor ne dă tocmai ceea ce trebuia demonstrat.

Aceste două formule reprezintă principiul includerii şi al excluderii. Formula (37) este o formulă de tip ciur (ca idee seamănă cu ciurul lui Eratostene).

Să rezolvăm următoarea problemă. *Câte dintre primele 100 numere naturale nu se divid nici cu 2, nici cu 3 şi nici cu 5?*

Fie A mulţimea numerelor naturale până la 100 şi A_2, A_3 şi A_5 mulţimile numerelor care nu se divid cu 2, 3 respectiv 5. Printr-un calcul simplu se obţine: $\#A_2 = \frac{100}{2} = 50$, $\#A_3 = \lfloor \frac{100}{3} \rfloor = 33$, $\#A_5 = 20$, $\#(A_2 \cap A_3) = \lfloor \frac{100}{2 \cdot 3} \rfloor = 16$, $\#(A_2 \cap A_5) = 10$, $\#(A_3 \cap A_5) = 6$, $\#(A_2 \cap A_3 \cap A_5) = 3$. Atunci numărul numerelor până la 100 care nu se divid cu 2, 3 şi 5 este:

$$\#A - \Big(\#A_2 + \#A_3 + \#A_5\Big) + \Big(\#(A_2 \cap A_3) + \#(A_2 \cap A_5) + \#(A_3 \cap A_5)\Big) -$$

$$-\#\Big(A_2 \cap A_3 \cap A_5\Big) = 100 - (50 + 33 + 20) - (16 + 10 + 6) - 3 = 32$$

Un caz special al formulei (38) este acela când toate submulţimele A_i au acelaşi cardinal: $\#A_i = N^{(1)}$, toate intersecţiile $A_i \cap A_j$ deasemenea au acelaşi cardinal: $\#(A_i \cap A_j) = N^{(2)}$ ş.a.m.d. Se notează cu $N = \#(A_1 \cup A_2 \cup \ldots \cup A_n)$. În acest caz formula (38) devine:

$$N = \binom{n}{1} N^{(1)} - \binom{n}{2} N^{(2)} + \ldots + (-1)^k \binom{n}{k} N^{(k)} + \ldots + (-1)^n \binom{n}{n} N^{(n)}.$$

6.1 Aplicaţie la determinarea funcţiei lui Euler

Funcţia $\varphi(n)$ al lui Euler ne dă numărul numerelor naturale mai mici ca n şi prime cu el. (Două numere naturale sunt prime între ele dacă cel mai mare divizor comun al lor este 1.) De exemplu $\varphi(10) = 4$ pentru că numerele mai mici ca 10 şi prime cu el sunt: 1, 3, 7, 9.

Pentru a calcula valoarea lui $\varphi(n)$ vom aplica formula (37) a ciurului. Numărul n poate fi descompus în factori primi: $n = p_1^{\alpha_1} p_2^{\alpha_2} \ldots p_r^{\alpha_r}$. Să notăm cu A_i mulţimea numerelor naturale mai mici ca n care sunt multipli de p_i. Atunci avem:

$$\#A_i = \frac{n}{p_i}, \quad \#(A_i \cap A_j) = \frac{n}{p_i p_j}, \quad \ldots \quad \#(A_i \cap A_j \cap A_k) = \frac{n}{p_i p_j p_k}, \quad \ldots$$

Atunci conform formulei (37) avem

$$\varphi(n) = n - \sum_{i=1}^{r} \frac{n}{p_i} + \sum_{1 \le i < j \le r} \frac{n}{p_i p_j} - \ldots + (-1)^r \frac{n}{p_1 p_2 \ldots p_r}$$

care este tocmai dezvoltarea produsului:

$$\varphi(n) = n \left(1 - \frac{1}{p_1}\right) \left(1 - \frac{1}{p_2}\right) \ldots \left(1 - \frac{1}{p_r}\right).$$

Se poate vedea uşor că de exemplu: $\varphi(10) = 10\left(1-\frac{1}{2}\right)\left(1-\frac{1}{5}\right) = 4$, deoarece $10 = 2 \cdot 5$.

6.2 Aplicaţie în teoria grafurilor

Vom demonstra următorul rezultat interesant.

Teorema 4. [Zarankiewicz] *Dacă G este un graf cu n vârfuri, fără bucle şi muchii multiple, care nu conţine subgrafuri complete cu k vârfuri, atunci pentru gradul minim δ al vârfurilor avem relaţia:*

$$\delta \leq \left\lfloor \frac{(k-2)n}{k-1} \right\rfloor$$

Înainte de a demonstra teorema, să vedem un exemplu. Pentru $n = 4$ şi $k = 3$, graful cu cele mai multe muchii şi deci şi cu grade ale vârfurilor cele mai mari este un ciclu cu 4 vârfuri (care nu conţine triunghi), deci gradul fiecărui vârf este 2, deci şi $\delta = 2$. Din teoremă rezultă tot 2, ca limita superioară pentru gradul minim.

Demonstraţie. Fie $f = \left\lfloor \frac{(k-2)n}{k-1} \right\rfloor$. De unde rezultă că avem următoarea relaţie:

$$(k-2)n = f(k-1) + r, \quad \text{unde } 0 \leq r < k-1 \tag{42}$$

Să presupunem că graful $G = (V, E)$ (unde V este mulţimea vârfurilor, E mulţimea muchiilor) satisface condiţiile teoremei, totuşi gradul fiecărui vârf este cel puţin $f + 1$. Vom demosntra că această presupunere ne conduce la o contradicţie. Vom nota cu $N(x)$ mulţimea vârfurilor adiacente lui x (vârfurile vecine). Fie

$$x_1 \in V, \quad x_2 \in N(x_1)$$

Cu ajutorul principiului includerii şi al excluderii să calculăm numărul vecinilor comune celor două vârfuri considerate:

$$\#\big(N(x_1) \cap N(x_2)\big) = \#N(x_1) + \#N(x_2) - \#\big(N(x_1) \cup N(x_2)\big) \geq$$
$$\geq (f+1) + (f+1) - n = 2(f+1) - n > 0$$

Pentru justificarea faptului că $2(f+1)-n>0$ să pornim de la (42), şi să-l exprimăm pe n, majorându-l pe r cu $k-1$:

$$n<\frac{(f+1)(k-1)}{k-2}=(f+1)\left(1+\frac{1}{k-2}\right) \tag{43}$$

De unde, pentru $k\geq 3$ obţinem $n<2(f+1)$.

Deci există un vârf x_3 în mulţimea $N(x_1)\cap N(x_2)$ (ceea ce înseamnă că $k>3$). Printr-un calcul analog se obţine:

$$\#\big(N(x_1)\cap N(x_2)\cap N(x_3)\big)\geq 3(f+1)-2n>0$$

Ultima inegalitate rezultă tot din (43) pentru $k\geq 4$.

Continuând, se obţine:

$$\#\big(N(x_1)\cap\ldots\cap N(x_{k-1})\big)=\#N(x_1)+\#\left(\bigcap_{j=1}^{k-2}N(x_j)\right)-$$

$$-\#\left(N(x_1)\cup\left(\bigcap_{j=1}^{k-2}N(x_j)\right)\right)\geq(f+1)+(k-2)(f+1)-(k-3)n-n=$$

$$=(k-1)(f+1)-(k-2)n=(k-1)(f+1)-f(k-1)-r=$$

$$=k-1-r>0$$

De aici rezultă că există un vârf $x_k\in N(x_1)\cap N(x_2)\cap\ldots\cap N(x_{k-1})$, iar aceste vârfuri $x_1,x_2,\ldots,x_k$ formează un sufgraf complet cu k vârfuri, ceea ce contrazice ipoteza şi astfel teorema este demonstrată.

6.3 Aplicaţie la determinarea numărului funcţiilor surjective

Se dau mulţimile $X=\{x_1,x_2,\ldots x_m\}$ şi $Y=\{y_1,y_2,\ldots,y_n\}$. Se cere să se determine numărul funcţiilor surjective $f:X\to Y$. O funcţie este surjectivă dacă toate elementele din Y sunt imagini ale unor elemente din X, adică $Y=f(X)$. Să notăm cu A mulţimea tuturor funcţiilor f de la X la Y, adică $A=\{f\mid f:X\to Y\}$. Fie A_i mulţimea funcţiilor pentru care y_i nu este imaginea nici unui element din X, adică $A_i=\{f\mid f:X\to Y, y_i\notin f(X)\}$. Numărul $S_{m,n}$ al funcţiilor surjectice de la X la Y este: $\#A-\#\left(\bigcup_{i=1}^{n}A_i\right)$. Folosind principiul includerii şi al excluderii, obţinem:

$$S_{m,n}=\#A-\sum_{i=1}^{n}\#A_i+\sum_{1\leq i<j\leq n}\#(A_i\cap A_j)-\ldots+(-1)^n\#\left(\bigcap_{i=1}^{n}A_i\right) \tag{44}$$

Se poate vedea uşor că $\#A = n^m$, deoarece fiecărui $x \in X$ îi putem asocia oricare element din Y şi de oricâte ori. La fel $\#A_i = (n-1)^m$, $\#(A_i \cap A_j) = (n-2)^m$, etc. Din Y putem elimina k elemente în $\binom{n}{k}$ moduri, deci

$$\sum_{1 \le i_1 < i_2 < \ldots < i_k \le n} \# \left(\bigcap_{j=1}^{k} A_{i_j} \right) = \binom{n}{k} (n-k)^m$$

După înlocuire în formula (44) obţinem:

$$S_{m,n} = n^m - \binom{n}{1}(n-1)^m + \binom{n}{2}(n-2)^m + \ldots + (-1)^{n-1}\binom{n}{n-1}$$

(Deoarece $A_1 \cap A_2 \cap \ldots \cap A_n = \emptyset$ — nu poate exista o funcţie care să nu ia nici o valoare — ultimul termen lipseşte.)

Dacă $m = n$ atunci numărul funţiilor surjective este egal cu cel al funcţiilor bijective, deci $S_{n,n} = n!$, deoarece pentru fiecare funcţie avem o permutare a celor n elemente. Obţinem în acest caz o formulă interesantă pentru $n!$:

$$n! = n^n - \binom{n}{1}(n-1)^n + \binom{n}{2}(n-2)^n - \ldots + (-1)^{n-1}\binom{n}{n-1}$$

sau într-o formă mai concisă:

$$n! = \sum_{k=0}^{n-1} (-1)^k \binom{n}{k} (n-k)^n.$$

6.4 Alte principii importante în combinatorică

Asemănător principiului includerii şi al excluderii există şi alte principii care se pot folosi la rezolvarea unor probleme. Vom prezenta în continuare două din ele.

6.4.1 Principiul cutiei

Acest principiu a fost formulat prima dată de matematicianul francez Dirichlet (1805–1859). În forma cea mai simplă se enunţă astfel: *Dacă n obiecte trebuie împărţite în mai puţin de n clase, atunci există cel puţin o clasă în care vor fi cel puţin două obiecte.*

Mai general se poate enunţa astfel:

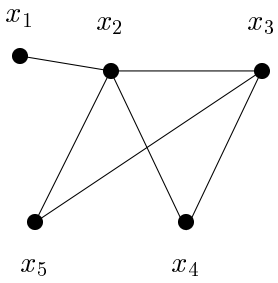
Fie date m obiecte, care trebuie împărţite în n clase şi un număr natural k astfel încât $m > kn$. Atunci în cazul oricărei împărţiri va exista cel puţin o clasă cu cel puţin $k+1$ obiecte.

Pentru $k = 1$ obţinem cazul de mai sus.

Vom prezenta câteva exemple de aplicare a acestui principiu. Primul exemplu îl vom lua din teoria grafurilor. Să demonstrăm următoarea afirmaţie:

Teorema 5. *În orice graf simplu (graf neorientat fără muchii multiple şi bucle) există cel puţin două vârfuri care au acelaşi grad.*

Demonstraţie. Să reamintim că gradul unui vârf este egal cu numărul muchiilor adiacente vârfului respectiv. Dacă notăm cu $\varphi(x)$ gradul vârfului x, în graful alăturat avem următoarele: $\varphi(x_1) = 1$, $\varphi(x_2) = 4$, $\varphi(x_3) = 3$, $\varphi(x_4) = 2$, $\varphi(x_5) = 2$. Pentru demonstrarea afirmaţiei să considerăm un graf neorientat cu n vârfuri. Atunci gradele vârfurilor pot fi între 0 şi $n-1$. Dacă avem un vârf de grad 0, el fiind izolat, nu putem avea niciun vârf cu grad $n-1$ (un vârf cu grad $n-1$ este legat cu toate celelalte vârfuri).



Deci gradele pot fi, fie

$$0, 1, \ldots, n-2,$$

fie

$$1, 2, \ldots, n-1,$$

adică în ambele cazuri avem $n-1$ grade diferite şi n vârfuri, deci conform principiului cutiei (forma simplă) există cel puţin două vârfuri cu acelaşi grad. Astfel teorema este demostrată. □

Să rezolvăm următoarea problemă:

Se dă un şir finit $a_1, a_2, \ldots, a_n$ de numere întregi. Să se determine un subşir $a_i, a_{i+1}, \ldots, a_j$ cu proprietatea că $a_i + a_{i+1} + \cdots a_j$ să fie multiplu de n.

Să considerăm următoarele sume:
$s_1 = a_1$,
$s_2 = a_1 + a_2$,

...

$s_i = a_1 + a_2 + \cdots + a_i$,

...

$s_n = a_1 + a_2 + \cdots + a_n$.

Sunt două cazuri:

1) Există un k astfel ca s_k este multiplu de n. În acest caz pentru subşirul căutat $i = 1, j = k$.

2) Dacă nici o sumă parţială s_k nu este multiplu de n, atunci resturile împărţirii acestor sume parţiale la n pot fi $1, \ldots, n-1$. Fiindcă avem n sume parţiale şi doar $n-1$ resturi, cel puţin două sume parţiale (fie s_x şi s_y, $x < y$) dau acelaşi rest, deci diferenţa lor se împarte la n. Atunci pentru şirul căutat $i = x + 1, j = y$.

Exemplu: Fie şirul dat:

$a_1 = -1, a_2 = 2, a_3 = 3, a_4 = 1, a_5 = 5, a_6 = 7$, deci $n = 6$.

Sumele parţiale vor fi:

$s_1 = -1, s_2 = 1, s_3 = 4, s_4 = 5, s_5 = 10, s_6 = 17$.

Resturile, pe rând, sunt: $5, 1, 4, 5, 4, 5$. Sunt patru soluţii:

$x = 1, y = 4 \Longrightarrow i = 2, j = 4$, deci: $2 + 3 + 1 = 6$

$x = 1, y = 6 \Longrightarrow i = 2, j = 6$, deci: $2 + 3 + 1 + 5 + 7 = 18$

$x = 3, y = 5 \Longrightarrow i = 4, j = 5$, deci: $1 + 5 = 6$

$x = 4, y = 6 \Longrightarrow i = 5, j = 6$, deci: $5 + 7 = 12$.

Principiul cutiei are următoarea consecinţă uşor de demonstrat:

Fie date numerele naturale $a_1, a_2, \ldots a_n$ şi x. Să notăm cu A media aritmetică a numerelor $a_1, a_2, \ldots, a_n$. Atunci

1) dacă $A > x$, există un i astfel încât $a_i > x$,

2) dacă $A < x$, există un i astfel încât $a_i < x$.

Pentru utilizarea acestei consecinţe, vom considera următoarea problemă:

Dacă ataşăm vârfurilor unui decagon numerele $0, 1, \ldots, 9$, să se demonstreze că există trei vârfuri consecutive pentru care suma numerelor ataşate este mai mare de 13.

Fie numerele ataşate celor 10 vârfuri: $x_1, x_2, \ldots, x_{10}$. Se consideră sumele:

$a_1 = x_1 + x_2 + x_3$

$a_2 = x_2 + x_3 + x_4$

...

$a_9 = x_9 + x_{10} + x_1$

$a_{10} = x_{10} + x_1 + x_2$.

Fie A media aritmetică a acestor numere. În sumele de mai sus fiecare număr apare de exact trei ori, deci media aritmetică este $\frac{3(0+1+2+\cdots+9)}{10} = 13,5$. Conform consecinţei principiului cutiei există un i ca $a_i > 13$, care ne dă soluţia problemei.

Prezentăm în continuare două probleme ale lui P. Erdős, care se rezolvă folosind principiul cutiei.

Oricum alegem $n+1$ numere dintre numerele naturale $1, 2, \ldots, 2n$, cel puţin două dintre ele vor fi relativ prime.

Să le grupăm numerele în felul următor:

$$\{1, 2\}, \{3, 4\}, \ldots \{2k-1, 2k\}, \ldots, \{2n-1, 2n\}$$

Alegând $n+1$ numere, cel puţin două vor fi din aceeaşi grupă (în total avem n grupe), deci vor fi numere consecutive, şi ca atare, prime între ele.

Oricum alegem $n+1$ numere dintre numerele naturale $1, 2, \ldots, 2n$, cel puţin două dintre ele vor fi unul multiplul celuilalt.

Formăm grupele următoare astfel: scriem numerele impare pe prima coloană, înmulţim prima coloană cu 2 şi le scriem numerele respective în coloana al doilea. Repetăm procedeul cu coloana a doua, etc.

$1, 2, 4, 8, \ldots$
$3, 6, 12, \ldots$
$5, 10, 20, \ldots$
$\ldots$
$2n-1$

De exemplu, dacă $n = 15$, obţinem următorul tabel:

1, 2, 4, 8, 16
3, 6, 12, 24
5, 10, 20
7, 14, 28
9, 18
11, 22
13, 26
15, 30
17
19
21
23
25
27
29

Se poate vedea că astfel toate numerele între 1 şi $2n$ apar în tabelul de mai sus. Numerele impare apar pe prima coloană, iar numerele pare pe celelalte. Un număr par totdeaunea se poate scrie ca $2^l(2k-1)$ pentru un l şi un k dat.

Având n linii, oricum alegem cele $n+1$ numere, cel puţin două vor fi din aceeaşi linie, care sunt unul multiplul celuilalt.

Aplicaţii în teoria limbajelor

Vom demonstra lema de pompare (lema Bar-Hillel) pentru limbajele regulare (pentru noţiuni a se vedea de exemplu [33] sau [45]).

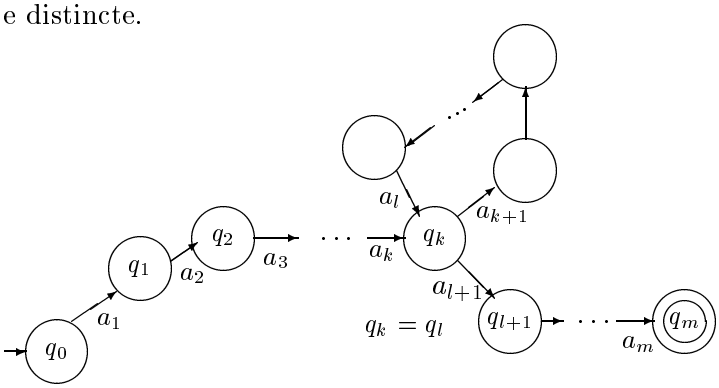
Teorema 6 (Bar-Hillel). *Pentru orice limbaj regular L există un număr natural $n > 0$, care depinde numai de L, astfel încât orice cuvânt $u \in L$ de lungime cel puţin n se poate descompune în $u = xyz$ cu următoarele proprietăţi:*

a) $|y| \geq 1$

b) $|xy| \leq n$

c) $xy^iz \in L$ *pentru orice* $i = 0, 1, 2, \ldots$

Demonstraţie. Presupunem că automatul finit care recunoaşte limbajul L are n stări (n-ul cerut de lemă): $q_0, q_1, \ldots, q_{n-1}$. Să considerăm un cuvânt oarecare $a_1a_2 \ldots a_m \in L$, cu proprietatea că $m \geq n$. Acest cuvânt este recunoscut de automatul finit considerat. În urma recunoaşterii automatul trece prin stările $q_0, q_1, \ldots, q_m$, unde q_m este o stare finală. Deoarece $m \geq n$, iar în şirul stărilor sunt $m+1$ stări, rezultă că aceste stări $q_0, q_1, \ldots, q_m$ nu pot fi toate distincte.



Fie $k < l$ cele mai mici indici pentru care $q_k = q_l$. În acest caz descompunerea cuvântului este următoarea:

$$x = a_1a_2 \ldots a_k, \;\; y = a_{k+1}a_{k+2} \ldots a_l, \;\; z = a_{l+1}a_{l+2} \ldots a_m.$$

Se verifică uşor că cele trei condiţii sunt satisfăcute. □

Lema de tip Bar-Hillel pentru limbajele independente de contex deasemenea foloseşte principiul cutiei în demostraţie.

6.4.2 Principiul celui mai lung drum în grafuri

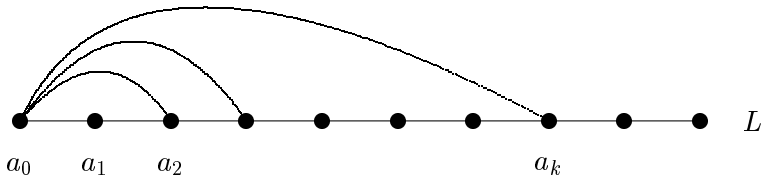
Vom folosi noţiunea de graf simplu, care înseamnă un graf neorientat fără muchii multiple şi fără bucle. Dacă nu se specifică că graful este orientat, în continuare, prin graf vom înţelege totdeaunea un graf neorientat. Printr-un drum într-un graf (orientat sau nu) în continuare vom înţelege un drum elementar, adică un drum în care toate vârfurile sunt distincte. (În cazul grafurilor neorientate drumul deobicei este numit lanţ.) De exemplu, în graful de la pagina 80 drumul x_1, x_2, x_3, x_4 este elementar, dar x_1, x_2, x_3, x_4, x_2 nu mai este elementar.

Într-un graf deci totdeaunea există un cel mai lung drum (adică cel care are cele mai multe vârfuri).

Să rezolvăm următoarele probleme, care se pot enunţa sub forma unor teoreme.

Teorema 7. *Dacă într-un graf simplu fiecare vârf are gradul cel puţin k, atunci graful are un ciclu de lungime cel puţin $k+1$.*

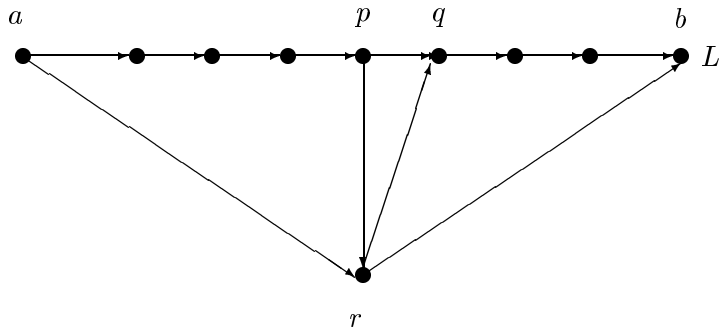
Demonstraţie. Să considerăm un cel mai lung drum notat cu L, format din vârfurile $a_0, a_1, a_2, \ldots$.



Vârful a_0 din drum are gradul cel puţin k, dar el nu poate fi legat decât cu vârfuri din drum, pentru că altfel drumul nu ar fi cel mai lung. Graful fiind simplu, a doua muchie (prima este muchia $\{a_0, a_1\}$) poate fi legat cu vârful a_2 (sau cu un vârf cu un indice mai mare), la fel muchia a k-a trebuie să fie legat în cel mai rău caz cu vârful a_k. Astfel se formează ciclul $a_0, a_1, a_2, \ldots, a_k, a_0$, care are lungime cel puţin $k+1$, ceea ce demonstrează afirmaţia. □

Teorema 8. [L. RÉDEI] *Indiferent cum orientăm muchiile unui graf complet în graful orientat obţinut va exista un drum hamiltonian.*

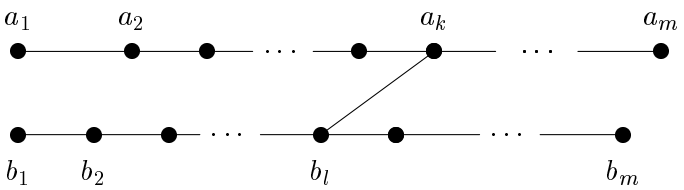
Demonstraţie. Un drum hamiltonian este un drum (elementar) care trece prin toate vârfurile grafului. Pentru demonstrarea teoremei se consideră un cel lung drum orientat L. Fie a şi b primul şi respectiv ultimul vârf din acest drum. Dacă acest drum nu este unul hamiltonian, atunci există un vârf r care nu este pe drumul considerat. Acest vârf r este legat cu toate vârfurile din drumul L cu arce de la r sau spre r. Nu poate exista un arc de la r la a, că în acest caz drumul nu ar fi de lungime maximă. Deci există arcul de la a la r. La fel trebuie să existe arcul de la r la b.



Pentru că avem arcele (a, r) şi (r, b), trebuie să existe vârfurile p şi q adiacente pe drumul L, astfel încât să existe arcele (p, r) şi (r, q). Dar în acest caz, înlocuind în L arcul (p, q) cu arcele (p, r) şi (r, q), obţinem un drum mai lung decât L, ceea ce este o contradicţie cu presupunerea că L este de lungime maximă. Deci presupunerea că drumul L de lungime maximă nu ar fi hamiltonian este greşită, ceea ce demonstrează afirmaţia. □

Un alt exemplu de utilizare a acestui principiu este următorul.

Să se demonstreze că dacă într-un graf conex nu există drumuri mai lungi de m, atunci oricare două drumuri de lungime m au cel puţin un punct comun.



Presupunem contrariul, adică existenţa a două drumuri de lungime m distincte: $a_1, a_2, \ldots, a_m$ şi $b_1, b_2, \ldots, b_m$. Dar graful fiind conex, trebuie să există

între oricare două vârfuri ale celor două drumuri câte un drum. Un astfel de drum care uneşte două vârfuri ale celor două drumuri trebuie să fie format din cel puţin o muchie. Fie această muchie (a_k, b_l). Drumul format din porţiunea mai lungă dintre $a_1, \ldots, a_k$ şi $a_k, \ldots, a_m$ (de lungime cel puţin $m/2$), muchia a_k, b_l, porţiunea mai lungă dintre $b_1, \ldots, b_l$ şi $b_l, \ldots, b_m$ (de lungime cel puţin $m/2$), este mai lung decât m, ceea ce este o contradicţie cu ipoteza că nu există drumuri mai lungi de m.

Probleme.

1. O permutare a elementelor $1, 2, \ldots, n$ are un punct fix, dacă un element i se găseşte în poziţia i. Să se calculeze numărul permutărilor a n elemente fără puncte fixe.

2. În câte moduri pot dansa n perechi (soţ–soţie) astfel încât nici un soţ să nu danseze cu soţia lui.

3. Să se demonstreze, folosind principiul includerii şi al excluderii, formula următoare:

$$\sum_{i=0}^{k} (-1)^i \binom{n}{i} \binom{n-i}{k-i} = 0.$$

4. Să se demonstreze că

$$\sum_{d|n} \varphi(n) = n,$$

unde φ este funcţia lui Euler (vezi pag. 76), iar $d|n$ înseamnă că d divide pe n.

7 Numere remarcabile

7.1 Numerele lui Fibonacci

Am mai văzut că numerele lui Fibonacci se pot defini recursiv prin formulele:

$$F_0 = 0, \quad F_1 = 1, \quad F_n = F_{n-1} + F_{n-2} \quad \text{pentru } n \geq 2. \tag{45}$$

Primele numere Fibonacci sunt:

$$0,\ 1,\ 1,\ 2,\ 3,\ 5,\ 8,\ 13,\ 21,\ 34,\ 55,\ 89,\ 144,\ 233,\ 377,\ 610,\ 987,\ \ldots$$

Funcţia generatoare a numerelor lui Fibonacci este (vezi formula (20) pe pagina 46):

$$F(z) = \sum_{n\geq 0} F_n z^n = \frac{z}{1 - z - z^2} \tag{46}$$

Rezolvând ecuaţia $1 - z - z^2 = 0$ obţinem rădăcinile $z_1 = \frac{1}{2}(-1 - \sqrt{5})$ şi $z_2 = \frac{1}{2}(-1 + \sqrt{5})$. Atunci fracţia $\dfrac{z}{1 - z - z^2}$ se poate descompune în fracţii simple de forma $\dfrac{A}{z - z_1} + \dfrac{B}{z - z_2}$. Printr-un calcul simplu se poate arăta că $F(z)$ se poate scrie şi sub forma:

$$F(z) = \frac{1}{\sqrt{5}} \left(\frac{1}{1 - \alpha z} - \frac{1}{1 - \beta z} \right) \tag{47}$$

unde

$$\alpha = \frac{1}{2}(1 + \sqrt{5}), \quad \beta = \frac{1}{2}(1 - \sqrt{5}) \tag{48}$$

Se ştie că

$$\frac{1}{1 - \alpha z} = 1 + \alpha z + \alpha^2 z^2 + \ldots + \alpha^n z^n + \ldots$$

Deci

$$F(z) = \frac{1}{\sqrt{5}}(1 + \alpha z + \alpha^2 z^2 + \alpha^3 z^3 + \ \ldots\ - 1 - \beta z - \beta^2 z^2 - \beta^3 z^3 - \ \ldots)$$

Identificând coeficienţii lui z^n din ambii membri, obţinem:

$$F_n = \frac{1}{\sqrt{5}}(\alpha^n - \beta^n) = \frac{1}{2^n \sqrt{5}} \left((1 + \sqrt{5})^n - (1 - \sqrt{5})^n \right)$$

La acelaşi rezultat se ajunge dacă rezolvăm relaţia de recurenţă care defineşte şirul lui Fibonacci.

Dezvoltând cei doi termeni din paranteză după formula binomului, obţinem următoarea formulă:

$$F_n = \frac{1}{2^n\sqrt{5}}\left(\sum_{k=0}^{n}\binom{n}{k}(\sqrt{5})^n - \sum_{k=0}^{n}\binom{n}{k}(-\sqrt{5})^n\right) =$$

$$= \frac{1}{2^n\sqrt{5}}\sum_{k=0}^{n}\binom{n}{k}(\sqrt{5})^n\left(1-(-1)^k\right) = \frac{1}{2^{n-1}\sqrt{5}}\sum_{k\ \text{impar}}\binom{n}{k}(\sqrt{5})^n$$

Deci al n-lea număr Fibonacci se poate scrie:

$$F_n = \frac{1}{2^{n-1}\sqrt{5}}\sum_{k=1}^{\lfloor\frac{n+1}{2}\rfloor}\binom{n}{2k-1}(\sqrt{5})^{2k-1} \tag{49}$$

La capitolul de funcţii generatoare am demonstrat deja că:

$$F_n = \sum_{k=0}^{\lfloor\frac{n}{2}\rfloor}\binom{n-1-k}{k}$$

Numerele lui Fibonacci satisfac multe identităţi interesante. Următoarea identitate matriceală se poate demonstra prin inducţie matematică:

$$\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^n = \begin{pmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{pmatrix} \tag{50}$$

Calculând determinantul celor doi membri, obţinem următoarea identitate, care se poate demonstra şi prin inducţie:

$$F_{n+1}F_{n-1} - F_n^2 = (-1)^n \tag{51}$$

Pornind de la formula de bază (45) a numerelor lui Fibonacci se poate ajunge pas cu pas la următoarea identitate:

$$F_{n+m} = F_mF_{n+1} + F_{m-1}F_n, \tag{52}$$

dar care se poate demonstra şi prin inducţie de ex. asupra lui m.

Vom demonstra prin inducţie matematică următoarea proprietate:

$$F_{nk} \quad \text{este multiplu de} \quad F_k. \tag{53}$$

Demonstraţia o vom face prin inducţie asupra lui n. Pentru $n = 2$ avem conform formulei (52): $F_{2k} = F_{k+k} = F_k F_{k+1} + F_{k-1} F_k$, care evident este multiplu de F_k. Presupunem adevărată afirmaţia pentru $(n-1)$ şi demonstrăm pentru n. Vom folosi din nou formula (52):

$$F_{nk} = F_{(n-1)k+k} = F_k F_{(n-1)k+1} + F_{k-1} F_{(n-1)k}$$

care este multiplu de F_k, deoarece în primul termen apare F_k, iar în termenul al doilea termen, conform ipotezei inducţiei, $F_{(n-1)k}$ este multiplu de F_k.

În continuare vom calcula următoarea sumă

$$\sum_{k=0}^{n} F_k F_{n-k}$$

Pornind de la formula produsului funcţiilor generatoare, se vede că această sumă este egală cu coeficientul termenului z^n în dezvoltarea funcţiei $F^2(z)$. Vom dezvolta această funcţie şi vom încerca să găsim coeficientul lui z^n în dezvoltare. Ţinând cond de formulele (46) şi (47) obţinem

$$F^2(z) = \left(\frac{z}{1-z-z^2}\right)^2 = \frac{1}{5}\left(\frac{1}{1-\alpha z} - \frac{1}{1-\beta z}\right)^2$$

unde $\alpha + \beta = 1$ conform formulelor (48). Deci

$$F^2(z) = \frac{1}{5}\left(\frac{1}{(1-\alpha z)^2} - \frac{2}{(1-\alpha z)(1-\beta z)} + \frac{1}{(1-\beta z)^2}\right)$$

Deoarece

$$\frac{1}{(1-\alpha z)^2} = \sum_{n\geq 0}(n+1)\alpha^n z^n \quad \text{şi} \quad \frac{2}{(1-\alpha z)(1-\beta z)} = \frac{2}{z}F(z)$$

obţinem

$$F^2(z) = \frac{1}{5}\sum_{n\geq 0}(n+1)(\alpha^n + \beta^n)z^n - \frac{2}{5}\sum_{n\geq 0} F_{n+1} z^n$$

Deoarece $\alpha + \beta = 1$ şi $\alpha\beta = -1$, avem

$$\sum_{n\geq 0}(\alpha^n + \beta^n)z^n = \sum_{n\geq 0}\alpha^n z^n + \sum_{n\geq 0}\beta^n z^n = \frac{1}{1-\alpha z} + \frac{1}{1-\beta z} = \frac{2-z}{1-z-z^2}$$

Deci

$$F^2(z) = \frac{1}{5}\sum_{n\geq 0}(n+1)(2F_{n+1} - F_n)z^n - \frac{2}{5}\sum_{n\geq 0} F_{n+1} z^n$$

De unde, după egalarea coeficienţilor termenilor în z^n în ambii membri, se obţine

$$\sum_{k=0}^{n} F_k F_{n-k} = \frac{2n}{5} F_{n+1} - \frac{n+1}{5} F_n.$$

Să calculăm şi suma primelor n numere Fibonacci:

$$\sum_{k=0}^{n} F_k$$

Dacă notăm suma primelor n numere Fibonacci cu s_n, funcţia generatoare corespunzătoare este:

$$S(z) = \sum_{n \geq 0} s_n z^n.$$

Se poate observa că, dacă folosim notaţia

$$F(z) = \sum_{n \geq 0} F_n z^n = \frac{z}{1 - z - z^2}, \qquad A(z) = \sum_{n \geq 0} z^n = \frac{1}{1 - z},$$

se obţine

$$S(z) = F(z)A(z) = \frac{z}{(1 - z)(1 - z - z^2)},$$

care, printr-un calcul simplu, se transformă în

$$S(z) = \frac{z + 1}{1 - z - z^2} - \frac{1}{1 - z} = \frac{z}{1 - z - z^2} + \frac{1}{1 - z - z^2} - \frac{1}{1 - z},$$

din care se obţine

$$s_n = F_n + F_{n+1} - 1 = F_{n+2} - 1,$$

adică

$$\sum_{k=0}^{n} F_k = F_{n+2} - 1.$$

Teorema 9. *Orice număr natural n se poate descompune într-o sumă de numere Fibonacci. Dacă nu se folosesc în descompunere numerele F_0 şi F_1, şi nici două numere Fibonacci consecutive, atunci abstracţie făcând de ordinea numerelor, această descompunere este unică.*

Înainte de demonstraţie să vedem câteva exemple:

$29 = 21 + 8 = F_8 + F_6$,

$100 = 89 + 8 + 3 = F_{11} + F_6 + F_4$,

$499 = 377 + 89 + 21 + 8 + 3 + 1 = F_{14} + F_{11} + F_8 + F_6 + F_4 + F_2$.

Demonstraţie. Dacă n este un număr Fibonacci demonstraţia este imediată. Altfel se demonstrează prin inducţie asupra lui n. Numerele 1, 2, 3 toate sunt numere Fibonacci, deci descompunerea lor se reduce la câte un număr Fibonacci: $1 = F_2, 2 = F_3, 3 = F_4$. Presupunem că orice număr natural mai mic de n se poate descompune într-o sumă de numere Fibonacci, respectând condiţiile enunţului. Scăzând din n cel mai mare număr Fibonacci (fie acesta F_k) mai mic decât n, se obţine un număr mai mic decât n, care prin ipoteza inducţiei se descompune într-o sumă de numere Fibonacci în mod unic. Adăugând F_k la această sumă obţinem descompunerea lui n. Dacă, prin aceasta apar două cifre 1 consecutive în descompunere, prin adunarea celor două numere Fibonacci (care sunt numere Fibonacci consecutive) apare un alt număr Fibonacci în decompunere. Prin acest procedeu se pot elimina cifrele 1 alăturate. □

În algoritmul de descompunere totdeaunea punem în evidenţă cel mai mare număr Fibonacci, care intră în număr, pe urmă scăzându-l din n se continuă la fel. În exemplele de mai sus s-a procedat în acest fel.

Folosind această descompunere, numerele naturale por fi reprezentate asemănător reprezentării numerelor în baza 2. Dacă 29 în baza 2 se scrie

$$29 = 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = (11101)_2$$

atunci 29 în scrierea Fibonacci este:

$$29 = 21 + 8 = 1 \cdot F_8 + 0 \cdot F_7 + 1 \cdot F_6 + 0 \cdot F_5 + 0 \cdot F_4 + 0 \cdot F_3 + 0 \cdot F_2 = \\ = (1010000)_F$$

În această scriere nu pot exista niciodată două cifre 1 alăturate, deoarece atunci suma celor două numere Fibonacci corespunzătoare ar da un alt număr Fibonaci mai mare, care ar trebui să apară în descompunere.

În descompunerea Fibonacci prima cifră totdeaunea este 1, iar ultima cifră reprezintă prezenţa or lipsa numărului F_2 în descompunere.

Uşor se pot imagina operaţii cu reprezentările Fibonacci ale numerelor naturale. Adunând doi de 1 apare în sumă un anumit număr Fibonacci F_k de două ori, care se scrie: $F_k + F_k = F_k + F_{k-1} + F_{k-2} = F_{k+1} + F_{k-2}$, deci în sumă va apare un 1 în poziţiile $k+1$ şi $k-2$. Deasemenea, plecând de la formula de bază a numerelor lui Fibonacci, se elimină din sumă cifrele 1 alăturate. De exemplu: $29 + 100 = (F_8 + F_6) + (F_{11} + F_6 + F_4) = (1010000)_F + (1000010100)_F$

			1	0	1	0	0	0	0
1	0	0	0	0	1	0	1	0	0
1	0	0	1	1	0	0	1	0	0
							1		
1	0	1	0	0	0	1	0	0	1

Deci suma este $F_{11} + F_9 + F_5 + F_2 = 129$.

Generalizarea numerelor lui Fibonacci. Numerele lui Fibonacci pot fi generalizate în multe feluri. Vom prezenta trei generalizări posibile.

1. Să calculăm numărul secvenţelor de lungime k formate din 0 şi 1, care încep şi se termină în 1, şi nu conţin d cifre 0 alăturate (deci conţin cel mult $d-1$ cifre 0 alăturate). Să numim aceste secvenţe *corecte*, şi notăm cu $F_{k,d}$ numărul lor. O secvenţă corectă de lungime k se poate obţine dintr-o secvenţă corectă de lungime $k-1$ prin adăugarea la dreapta ei a unei cifre 1, sau drintr-o secvenţă corectă de lungime $k-2$ prin adăugarea la dreapta a secvenţei 01, şi aşa mai departe până la secvenţe de lungime $k-d$, la care se poate adăuga secvenţa $\underbrace{00\ldots01}_{d}$. Astfel se obţine următoarea relaţie cu condiţiile iniţiale date:

$$\begin{aligned} F_{k,d} &= F_{k-1,d} + F_{k-2,d} + \ldots F_{k-d,d}, \qquad \text{pentru } k > 1 \\ F_{1,d} &= 1 \\ F_{k,d} &= 0, \qquad \text{pentru } k \leq 0. \end{aligned}$$

Dacă considerăm funcţia generatoare $F_d(z) = \sum_{n \geq 0} F_{n,d} z^n$, printr-un calcul simplu, asemănător celui făcut la obţinerea funcţiei generatoare a numerelor lui Fibonacci (pag. 46), se obţine următoarea formulă:

$$F_d(z)(1 - z - z^2 - \ldots - z^d) = F_{0,d} + z(F_{1,d} - F_{0,d}) + z^2(F_{2,d} - F_{1,d} - F_{0,d})$$
$$+ \ldots + z^{d-1}(F_{d-1,d} - F_{d-2,d} - \ldots - F_{0,d}) + \ldots$$

Conform formulei de recurenţă de mai sus şi a condiţiilor iniţiale respective, toate parantezele, mai puţin prima, sunt egale cu 0. Deoarece şi $F_{0,d} = 0$, se obţine formula:

$$F_d(z) = \sum_{n \leq 0} F_{n,d} z^n = \frac{z}{1 - z - \ldots - z^d} = \frac{z(1-z)}{1 - 2z + z^{d+1}}.$$

Pentru $d = 2$ se obţine funcţia generatoare a numerelor lui Fibonacci, deci $F_{n,2} = F_n$. Astfel, numerele $F_{n,d}$ pot fi considerate ca numere Fibonacci generalizate.

2. Pornim de la următoarea problemă: *Câte secvenţe de lungime n formate numai din 0 şi 1 există, în care nu sunt două cifre 1 consecutive?*

Să notăm cu a_n numărul acestor secvenţe. O astfel de secvenţă se va numi secvenţă corectă. O secvenţă corectă de lungime n evident se poate termina în 0 sau 1. Dacă ultima poziţie este 0, atunci penultima poate fi orice cifră (0 sau 1), dacă pe ultima poziţie este un 1, atunci înaintea lui nu poate fi decât 0. Pe primele $n-1$ respectiv $n-2$ poziţii poate figura orice secvenţă corectă de lungime $n-1$ respectiv $n-2$. De aici rezultă că $a_n = a_{n-1} + a_{n-2}$, şi cum $a_1 = 2$ (secvenţele corecte sunt 0 şi 1), $a_2 = 3$ (secvenţele corecte sunt $00, 01, 10$), rezultă că $a_n = F_{n+2}$ (Se poate considera că $a_0 = 1$).

Să formulăm şi o altă problemă: *În mulţimea tuturor secvenţelor corecte de acest fel câte cifre 1 există?* Să notăm acest număr cu r_n. Considerăm la început toate secvenţele care se termină în 0. În mulţimea acestora sunt r_{n-1} cifre 1. Secvenţele care se termină în 1 de fapt se termină în 01, în total sunt a_{n-2} astfel de secvenţe, în care sunt deci $r_{n-2} + a_{n-2}$ cifre 1. Deci $r_n = r_{n-1} + r_{n-2} + a_{n-2}$, adică

$$r_n = r_{n-1} + r_{n-2} + F_n, \quad r_0 = 0, r_1 = 1.$$

Aceste numere pot sugera o generalizare a numerelor lui Fibonacci în felul următor:

$$F_n^{(k)} = F_{n-1}^{(k)} + F_{n-2}^{(k)} + F_n^{(k-1)}, \quad F_0^{(k)} = 0,\ F_1^{(k)} = 1 \text{ şi } F_n^{(0)} = F_n. \qquad (54)$$

Pentru $k = 1$ obţinem $F_n^{(1)} = a_n$. În tabelul următor se dau câteva valori pentru acest număr Fibonacci generalizat.

n	$k=0$	$k=1$	$k=2$	$k=3$
0	0	0	0	0
1	1	1	1	1
2	1	2	3	4
3	2	5	9	14
4	3	10	22	40
5	5	20	51	105
6	8	38	111	256
7	13	71	233	594
8	21	130	474	1324
9	34	235	942	2860
10	55	420	1836	6020

Funcţia generatoare a acestor numere $F_m^{(k)}$ este:

$$F^{(k)}(z) = \sum_{n\geq 0} F_n^{(k)} z^n.$$

Din definiţia acestor numere, conform formulei (54) rezultă

$$F^{(k)}(z) = zF^{(k)}(z) + z^2F^{(k)}(z) + F^{(k-1)}(z).$$

De unde

$$F^{(k)}(z)(1 - z - z^2) = F^{(k-1)}(z).$$

Deoarece

$$F^{(0)}(z) = F(z) = \frac{z}{1 - z - z^2},$$

rezultă printr-un calcul simplu că

$$F^{(k)}(z) = \frac{z}{(1 - z - z^2)^{k+1}}.$$

Folosind formula cunoscută (vezi pag. 63):

$$\sum_{n\geq 0} \binom{n+k}{k} z^n = \frac{1}{(1-z)^{k+1}},$$

şi înlocuind în ea z cu $z(1+z)$, obţinem

$$F^{(k)}(z) = \sum_{n\geq 0} \binom{n+k}{k} \sum_{i=0}^{n} \binom{n}{i} z^{n+i+1}.$$

Iar de aici, identificând coeficienţii lui z^m în ambii termeni, obţinem:

$$F_m^{(k)} = \sum_{n=\lceil \frac{m-1}{2} \rceil}^{m-1} \binom{n+k}{k} \binom{n}{m-n-1}.$$

Am ţinut cont de faptul că $i \leq n$, de unde $2n+1 \geq m$, şi pentru un n fixat va exista un singur i pentru care exponentul este egal cu m.

3. O altă posibilitate de generalizare a numerelor lui Fibonacci ar fi să considerăm formula de recurenţă

$$f_n^{(k)} = k f_{n-1}^{(k)} + f_{n-2}^{(k)}, \quad \text{cu} \quad f_0^{(k)} = 0,\ f_1^{(k)} = 1, \tag{55}$$

pentru orice $k \geq 1$. Evident, pentru $k = 1$ se obţin numerele lui Fibonacci obişnuite, adică $f_n^{(1)} = F_n$. În tabelul următor sunt adte cîteva valori ale acestor numere.

n	$k=1$	$k=2$	$k=3$	$k=4$
0	0	0	0	0
1	1	1	1	1
2	1	2	3	4
3	2	5	10	17
4	3	12	33	72
5	5	29	109	305
6	8	70	360	1292
7	13	169	1189	5473
8	21	408	3927	23184
9	34	985	12970	98209
10	55	2378	42837	416020

Ataşăm acestei formule de recurenţă ecuaţia caracteristică $r^2 - kr - 1 = 0$, care are ca rădăcini

$$r_1 = \frac{k + \sqrt{k^2+4}}{2} \qquad r_2 = \frac{k - \sqrt{k^2+4}}{2}.$$

De unde avem

$$f_n^{(k)} = C_1 \left(\frac{k + \sqrt{k^2+4}}{2} \right)^n + C_2 \left(\frac{k - \sqrt{k^2+4}}{2} \right)^n,$$

şi ţinând cont de condiţiile iniţiale, obţinem

$$f_n^{(k)} = \frac{1}{\sqrt{k^2+4}} \left(\frac{k + \sqrt{k^2+4}}{2} \right)^n - \frac{1}{\sqrt{k^2+4}} \left(\frac{k - \sqrt{k^2+4}}{2} \right)^n, \quad n \geq 0.$$

Cazul $k = 2$ poate prezenta un interes mai mare. Să folosim în acest caz notaţia f_n în loc de $f_n^{(2)}$.

$$f_n = \frac{1}{2\sqrt{2}}\left(1+\sqrt{2}\right)^n - \frac{1}{2\sqrt{2}}\left(1-\sqrt{2}\right)^n, \text{ pentru } n \geq 0.$$

Dezvoltând după formula binomului, obţinem

$$f_n = \frac{1}{2\sqrt{2}}\left(\sum_{k\geq 0}\binom{n}{k}\left(\sqrt{2}\right)^k - \sum_{k\geq 0}\binom{n}{k}\left(-\sqrt{2}\right)^k\right)$$

$$= \frac{1}{\sqrt{2}}\sum_{k \text{ impar}}\binom{n}{k}\left(\sqrt{2}\right)^k = \sum_{k\geq 0}\binom{n}{2k+1}2^k.$$

Calculele de mai sus ne conduc la următoarea identitate:

$$\sum_{k\geq 0}\binom{n}{2k+1}2^k = \frac{\left(1+\sqrt{2}\right)^n - \left(1-\sqrt{2}\right)^n}{2\sqrt{2}}.$$

Se poate demonstra uşor (de exemplu prin inducţie asupra lui n) următoarea formulă:

$$f_{n+m} = f_{n+1}f_m + f_n f_{m-1}, \qquad \text{pentru } n \geq 0,\ m \geq 1.$$

Un caz particular ar acestei formule este:

$$f_{2n+1} = f_{n+1}^2 + f_n^2.$$

Aceste formule sunt adevărate şi pentru un k oarecare, deci:

$$f_{n+m}^{(k)} = f_{n+1}^{(k)}f_m^{(k)} + f_n^{(k)}f_{m-1}^{(k)}, \qquad \text{pentru } n \geq 0,\ \ m \geq 1,\ \ k \geq 1, \tag{56}$$

ceea ce reprezintă o generalizare a formulei (52). Avem şi formula

$$f_{2n+1}^{(k)} = \left(f_{n+1}^{(k)}\right)^2 + \left(f_n^{(k)}\right)^2.$$

Demonstrând formula

$$\begin{pmatrix} f_{n+1}^{(k)} & f_n^{(k)} \\ f_n^{(k)} & f_{n-1}^{(k)} \end{pmatrix} = \begin{pmatrix} k & 1 \\ 1 & 0 \end{pmatrix}^n$$

se obţine

$$f_{n+1}^{(k)}f_{n-1}^{(k)} - \left(f_n^{(k)}\right)^2 = (-1)^n.$$

Aceste numere pot calculate cu ajutorul funcţiilor generatoare. Fie

$$f^{(k)}(z) = \sum_{n \geq 0} f_n^{(k)} z^n.$$

Pornind de la formula (55) se obţine

$$f^{(k)}(z) = \frac{z}{1 - kz - z^2}.$$

Ţinând cont de dezvoltarea

$$\frac{1}{1-z} = 1 + z + z^2 + \ldots + z^m + \ldots$$

prin înlocuirea $z \to z(k+z)$ obţinem

$$\frac{z}{1 - z(k+z)} = z + z^2(k+z) + \ldots + z^{m+1}(k+z)^m + \ldots$$

de unde:

$$f_{n+1}^{(k)} = \sum_{i \geq 0} \binom{n-i}{i} k^{n-2i}$$

sau

$$f_n^{(k)} = \sum_{i \geq 0} \binom{n-1-i}{i} k^{n-1-2i}$$

Sistemul cu camarazi pentru alocarea dinamică a memoriei calculatoarelor. Sistemul cu camarazi (buddy system) este o metodă de alocarea dinamică a memoriei calculatoarelor în care zonele de memorie (numite blocuri) sunt alocate în dimensiuni predefinite [36]. În sistemul general există următoarea relaţie de recurenţă între dimensiunile respective:

$$u_m = a_1 u_{m-1} + a_2 u_{m-2} + \ldots a_r u_{m-r}, \qquad r > 0$$

ceea ce înseamnă că un bloc de dimensiune u_m se împarte în a_1 blocuri de dimensiune u_{m-1}, în a_2 blocuri de dimensiune u_{m-2} etc. (Dimensiunile $u_0, u_1, \ldots, u_{r-1}$ sunt date.) Blocurile vecine de memorie pot fi comasate într-unul singur dacă ele provin dintr-o astfel de împărţire (aceste blocuri se numesc camarazi). Un caz particular important este cel în care $a_1 = 1, a_2 = 0, \ldots a_{r-1} = 0, a_r = 1$, adică relaţia de mai sus devine

$$u_m = u_{m-1} + u_{m-r}, \qquad r > 0.$$

Pentru $r = 1$ obţinem sistemul binar clasic (fiecare bloc de dimensiune 2^k se împarte în două blocuri egale), iar pentru $r = 2$ se obţine *sistemul Fibonacci*.

Sortare externă. Când elementele de sortat nu încap în întregime în memoria calculatorului ele sunt păstrate pe suporturi externe şi sortate şi interclasate pe porţiuni. În [43] şi [5] se descrie *metoda polifazată* pentru sortarea externă, care stă în faptul că porţiuni din fişierul iniţial sunt sortate şi depuse pe suporturi externe (numite benzi pentru faptul că iniţial benzile magnetice erau folosite în acest scop) în fişiere numite monotonii. Aceste monotonii sunt depuse pe două benzi, pe urmă două câte două (câte una de pe fiecare bandă) sunt interclasate, iar monotoniile rezultate sunt depuse pe o a treia bandă. Când una din benzile se eliberează, celelalte două sunt considerate ca iniţiale, şi se continuă cu interclasarea până când se ajunge la o singură monotonie. În această metodă este important ca la un moment dat să se elibereze o singură bandă, ceea ce este asigurat prin faptul că iniţial pe cele două benzi se depun F_{k+1} respectiv F_k monotonii (unde F_{k+1} şi F_k sunt numere Fibonacci).

7.2 Numerele lui Catalan

Numerele $C_n = \dfrac{1}{n+1}\binom{2n}{n}$ se numesc numerele lui Catalan. Se poate observa că aceste numere sunt identice cu numerele F_n, care reprezintă numărul arborilor binari cu n vârfuri.

Numerele lui Catalan apar în foarte multe probleme. Printre altele C_n reprezintă:

- numărul arborilor binari cu n noduri,
- numărul de modalităţi de a plasa perechi de paranteze într-o secvenţă de $n+1$ numere pentru a le înmulţi două câte două,
- numărul expresiilor corecte în formă poloneză posfixată formate din n operanzi şi $n+1$ operatori,
- numărul drumurilor sub diagonală într-o grilă care unesc punctele $(0,0)$ şi (n,n) formate din segmente orizontale şi verticale,
- numărul secvenţelor cu n biţi în care numărul cifrelor 1 nu depăşeşte numărul cifrelor 0 în nici o poziţie plecând de la stânga spre dreapta (aceste secvenţe le vom numi secvenţe de tip Catalan),
- numărul segmentelor care unesc $2n$ puncte în plan fără să se intersecteze,
- numărul şirurilor $(x_1, x_2, \ldots, x_{2n})$ în care $x_i \in \{1, -1\}$ şi $x_1 + x_2 + \cdots + x_{2n} = 0$ cu proprietatea că pentru toate sumele parţiale avem: $x_1 \geq 0$, $x_1 + x_2 \geq 0$, $\ldots$, $x_1 + x_2 + \cdots + x_{2n-1} \geq 0$,

- numărul modurilor de a împărţi un poligon cu $n+2$ vârfuri în n triunghiuri.

Afirmaţiile de mai sus pot fi demonstrate uşor dacă dăm o codificare a obiectelor de fiecare categorie, astfel încât fiecare obiect să se codifice printr-o secvenţă de tip Catalan şi fiecărei secvenţe de tip Catalan să-i corespundă un obiect din categoria respectivă.

Numerele lui Catalan reprezintă soluţia următoarei ecuaţii de recurenţă:

$$C_{n+1} = C_0C_n + C_1C_{n-1} + \cdots + C_nC_0, \quad \text{pentru } n \geq 0$$

cu $C_0 = 1$ (a se vedea argumentarea la calculul numărului arborilor binari).

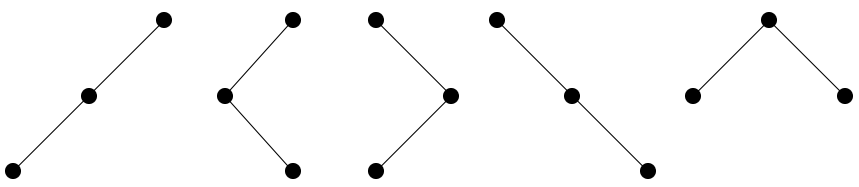
Numerele lui Catalan verifică şi relaţia: $(n+2)C_{n+1} = (4n+2)C_n$ pentru $n \geq 0$ (cu $C_0 = 1$).

Funcţia generatoare corespunzătoare este:

$$\sum_{n\geq 0} C_n x^n = \frac{1-\sqrt{1-4x}}{2x}.$$

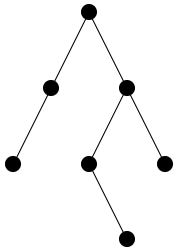
Codificarea arborilor binari

Plecând de la rădăcină, codificăm muchiile arborelui stâng şi pe urmă cele ale arborelui drept. Se pune 00 pentru o muchie stângă şi 11 pentru una de dreaptă, dacă există ambele; se pune 01 pentru o muchie stângă şi 10 pentru cea de dreaptă, dacă lipseşte muchia pereche. Completăm secvenţa cu 0 la început şi 1 la sfârşit, şi evident vom obţine o secvenţă de tip Catalan. Unui arbore binar cu n vârfuri îi corespunde o secvenţă de tip Catalan de $2n$ cifre (din care n 0-uri şi n 1-uri). În cazul arborilor binari cu 3 vârfuri avem următoarea codificare (spaţiile în coduri s-au pus numai pentru a evidenţia formarea lor, ele nu au nici o altă semnificaţie):



0 01 01 1 0 01 10 1 0 10 01 1 0 10 10 1 0 00 11 1

Un exemplu mai complex:



0 00 01 11 00 10 11 1

Algoritm pentru codificarea unui arbore binar

Intrare: un arbore binar B
Ieşire: o secvenţă Catalan
scrie 0
cheamă codificare (B)
scrie 1

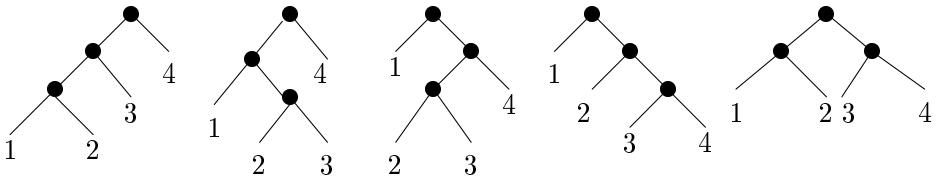
Iar procedura *codificare* este:

procedura codificare (B):
Fie B_L subarborele stâng şi B_R subarborele drept al arborelui B
dacă $B_L \neq \emptyset$ şi $B_R = \emptyset$ **atunci**
scrie 01
cheamă codificare (B_L)
dacă $B_L = \emptyset$ şi $B_R \neq \emptyset$ **atunci**
scrie 10
cheamă codificare (B_R)
dacă $B_L \neq \emptyset$ şi $B_R \neq \emptyset$ **atunci**
scrie 00
cheamă codificare (B_L)
scrie 11
cheamă codificare (B_R)
sfârşit procedură

Codificarea pentru înmulţire

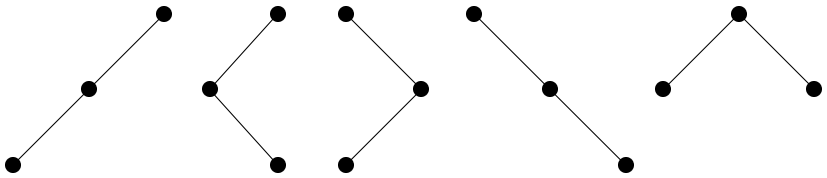
Pentru codificarea înmulţirilor, la început ataşăm unei modalităţi de înmulţire a celor n numere un arbore binar în felul următor: dacă înmulţim a cu b, construim un subarbore format dintr-un nod rădăcină (care corespunde operatorului de înmulţire) şi două noduri corespunzătoare operanzilor. Arborele obţinut este un arbore binar regular (fiecare nod diferit de frunze, are exact

doi descendenţi). Ştergem din acest arbore toate frunzele cu muchiile corespunzătoare şi pe urmă codificăm arborele binar obţinut. Pentru $n = 4$ avem următoarele expresii şi arbori binari regularii:



(((1 2) 3) 4) ((1 (2 3)) 4) (1 ((2 3) 4)) (1 (2 (3 4))) ((1 2) (3 4))

Dacă ştergem frunzele din aceşti arbori obţinem arborii binari de mai sus, adică



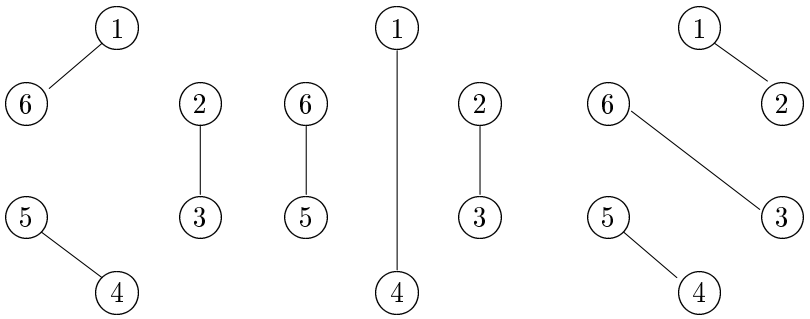
001011 001101 010011 010101 000111

Codificarea expresiilor poloneze postfixate

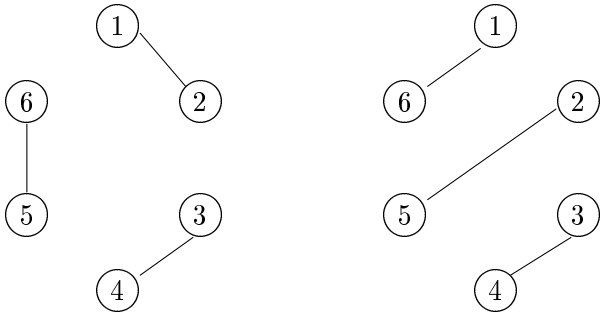
Se codifică fiecare operand cu 0 şi fiecare operator cu 1 şi se adaugă un 1 la sfârşit. De exemplu pentru expresia $abc \times d \times \times$ — care corespunde expresiei algebrice $(a \times ((b \times c) \times d))$ — codul rezultat este 00010111.

Codificarea pentru segmente

Dacă avem $2n$ puncte în plan pe care le unim prin n segmente care două câte două nu se intersectează, folosim următoarea codificare: Numerotăm punctele de la 1 la $2n$. Pentru un segment care uneşte punctele i şi j $(i < j)$ punem 0 în poziţia a i-a în secvenţă şi 1 în poziţia a j-a. Pentru $n = 3$ avem următoarele figuri şi coduri:



001011 001101 010011

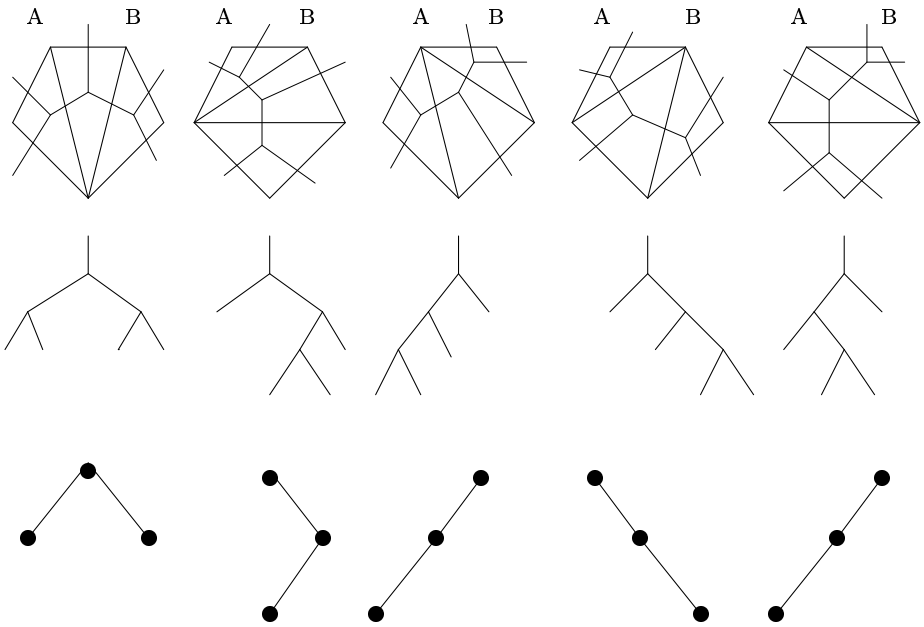


010101 000111

Codificarea pentru poligoane

Se împarte poligonul în triunghi, după care se ia câte un nod în fiecare triunghi şi atâte noduri exterioare câte laturi sunt, se duce câte o muchie care uneşte punctele din triunghiurile care au câte o latură comună. Deasemena se duce câte o muchie care uneşte un nod dintr-un triunghi care are o latură identică cu o latură a poligonului şi unul din afara laturii. Arborele astfel obţinut va sta la baza codificării împărţirii poligonului în triunghiuri. Vom exemplica pentru $n = 3$, adică considerăm cazul poligonului cu 5 laturi (pentagon). După obţinerea arborelui ataşat se fixează o latură (în exemplul de mai jos latura AB), se desenează arborele folosind ca rădăcină nodul în partea exterioară a laturii considerate. Muchia corespunzătoare se desenează în jos, pe urmă se continuă cu desenarea arborelui sub forma unui arbore binar. Dacă se elimină din acest arbore toate muchiile care au o extremitate un nod de tip frunză (adică care are gradul 1) se obţine un arbore binar, care se poate codifica cum

s-a mai prezentat.



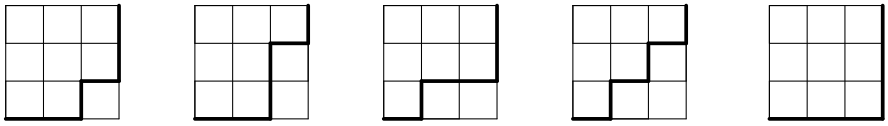
Codificarea pentru şiruri

Se codifică numărul 1 prin 0 şi numărul -1 prin 1. Evident, se obţine o secvenţă în care numărul 1-urilor nu depăşeşte numărul 0-urilor în nici o poziţie (din cauza că sumele parţiale sunt ≥ 0). Pentru $n = 3$ avem următoarele şiruri corecte şi codurile corespunzătoare:

`1, 1, 1, -1, -1, -1` codificat: 000111
`1, 1, -1, 1, -1, -1` codificat: 001011
`1, 1, -1, -1, 1, -1` codificat: 001101
`1, -1, 1, 1, -1, -1` codificat: 010011
`1, -1, 1, -1, 1, -1` codificat: 010101

Codificarea în grilă

Punem 0 pentru o unitate de drum orizontală şi 1 pentru una verticală. Pentru o grilă de 3×3 avem următoarele drumuri şi coduri:



001011 001101 010011 010101 000111

Decodificarea

Decodificarea secvenţelor de tip Catalan se poate face uşor. Vom exemplifica decodificarea pe secvenţa: 00010111.

Dacă vrem să obţinem un arbore binar, ştergem primul şi ultimul bit. Secvenţa 00 înseamnă o muchie stângă pentru care există şi o pereche dreaptă (păstrăm poziţia ei într-o stivă), urmează o secvenţă 10 care corespunde la o muchie dreaptă, secvenţa 11 rămasă este perechea dreaptă a muchiei păstrate în stivă. Se obţine arborele binar din figura *a*) de mai jos.

Algoritm pentru decoficarea unei secvenţe Catalan în arbore binar

Intrare: o secvenţă Catalan
Ieşire: un arbore binar
Observaţie: În urma desenării unei muchii dintr-un vârf numit vârf curent, noul vârf desenat devine noul vârf curent.
şterge un 0 de la începutul şi un 1 de la sfârşitul secvenţei, şi desenează un vârf (rădăcina) ca vârf curent
cheamă decodificare (c)

Iar procedura *decodificare* este:

procedura decodificare (c):
citeşte ab
şterge ab din c
dacă $ab = 01$ **atunci**
    desenează o muchie stângă din vârful curent
    cheamă decodificare (c)
dacă $ab = 10$ **atunci**
    desenează o muchie dreaptă din vârful curent
    cheamă decodificare (c)
dacă $ab = 00$ **atunci**
    pune în stivă poziţia vârfului curent
    desenează o muchie stângă din vârful curent
    cheamă decodificare (c)
dacă $ab = 11$ **atunci**

scoate din stivă poziţia vârfului curent,
acesta va fi vârful curent
desenează o muchie dreaptă din vârful curent
cheamă decodificare (c)

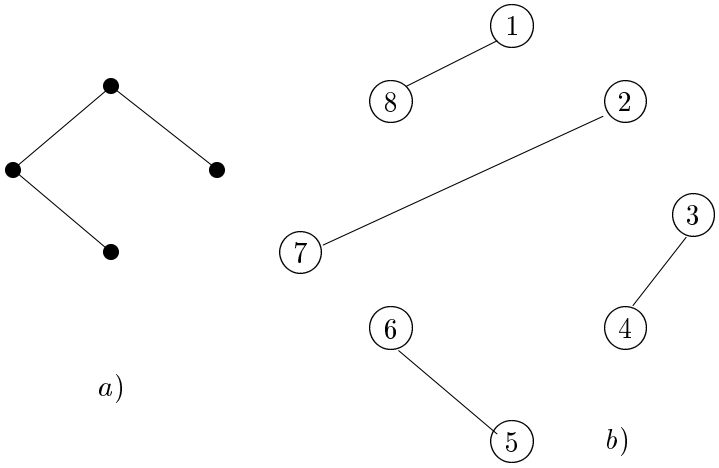
sfârşit procedură

Dacă dorim să obţinem segmente de dreaptă din acest cod, odată căutăm prima subsecvenţă 01, trasăm segmentul (între punctele 3 şi 4), le ştergem din şir şi continuăm cu următoarea secvenţă 01 (păstrând poziţiile originale). Se obţine segmentele din figura b).

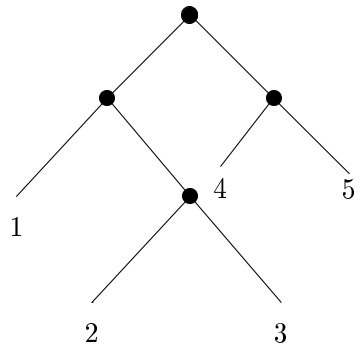
Pentru înmulţire, odată construim arborele binar (fig. a), pe urmă îl completăm cu noduri, ca fiecare nod original să aibă exact doi descendenţi. Arborele binar regular ne dă ordinea de înmulţire (fig. c).

Drumul în grilă se obţine imediat trasând unităţi orizontale şi verticale, după cum avem 0 sau 1 în secvenţă (fig. d).

Decodificarea pentru şiruri şi poligoane este deasemenea uşoară.

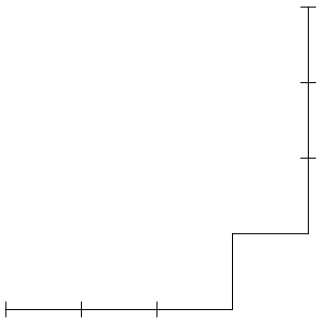


a) b)



((1 (2 3)) (4 5))

c)



d)

Numerele lui Catalan pot fi generalizate [15]. De exemplu, se pot defini numerele $C_n^{(k)}$ care reprezintă numărul drumurilor de la $(0, 0, \ldots, 0)$ la $(n, n, \ldots, n)$ într-o grilă k-dimensională care trec prin punctele $(x_1, x_2, \ldots, x_k)$ care satisfac condiţia $x_1 \geq x_2 \geq \ldots \geq x_k$. Acest număr se poate exprima cu ajutorul formulei:

$$C_n^{(k)} = \frac{1}{\binom{n+1}{1}} \cdot \frac{1}{\binom{n+2}{2}} \cdots \frac{1}{\binom{n+k-1}{k-1}} \cdot \frac{(kn)!}{n!n! \cdots n!}.$$

(În formula de mai sus $n!$ apare de k ori.) Pentru $k = 2$ obţinem numerele clasice ale lui Catalan.

7.3 Numerele lui Stirling

Vom folosi următoarea notaţie: $[x]_n = x(x-1)(x-2)\ldots(x-n+1)$ pentru $x \in \mathbf{R}, n \in \mathbf{N}^*$. Evident că $[x]_n$ fiind un polinom în x se poate scrie sub forma obişnuită:

$$[x]_n = s(n,0) + s(n,1)x + s(n,2)x^2 + \ldots + s(n,k)x^k + \ldots + s(n,n)x^n$$

unde coeficienţii $s(n,k)$ se numesc *numerele lui Stirling de speţa întâi.* Pornind de la relaţia evidentă $[x]_{n+1} = [x]_n(x-n)$ obţinem următoarea formulă de recurenţă pentru numerele $s(n,k)$:

$$s(n+1,k) = s(n,k-1) - ns(n,k)$$

cu $s(n,0) = 0$, $s(n,n) = 1$. Iar pentru $n < k$ avem $s(n,k) = 0$.

Să exprimăm x^n în funcţie de $[x]_n$:

$$x^n = S(n,1)[x]_1 + S(n,2)[x]_2 + \ldots + S(n,k)[x]_k \ldots + S(n,n)[x]_n \quad (57)$$

Numerele $S(n,k)$ se numesc *numerele lui Stirling de speţa a doua*. Să vedem cum am putea interpreta numerele $S(n,k)$. Să înlocuim în (57) x cu m ($m \in \mathbf{N}^*$):

$$m^n = S(n,1)[m]_1 + \ldots + S(n,k)[m]_k \ldots + S(n,n)[m]_n \quad (58)$$

În membrul stâng m^n poate reprezenta numărul tuturor funcţiilor $f : X \to Y$, unde $X = \{x_1, x_2, \ldots x_n\}$ iar $Y = \{y_1, y_2, \ldots, y_m\}$ (aranjamente cu repetiţii a m elemente în grupe de n). Aceste funcţii pot fi grupate după numărul de elemente folosite din Y ca imagini de valori din X. Să calculăm numărul funcţiilor care folosesc k elemente din Y. Din Y pot fi alese k elemente în A_m^k moduri ($A_m^k = [m]_k$ — aranjamente de m luate cu câte k) deoarece contează şi ordinea lor). Aceste valori sunt imagini a n elemente din X ($n \geq k$), deci mai multe elementele din X pot avea aceeaşi imagine. Să împărţim mulţimea X în k submulţimi, elementele unei submulţimi vor avea aceeaşi imagine din Y. Aceste submulţimi reprezintă o partiţie a mulţimii X (adică sunt disjuncte şi reuniunea lor este tocmai X). Astfel după formula (58) $S(n,k)$ reprezintă numărul partiţiilor mulţimii X cu n elemente în k clase (submulţimi).

Se poate arăta că avem: $k!S(n,k) = s_{n,k}$, unde $s_{n,k}$ reprezintă numărul funcţiilor surjective $f : A \to B$ cu $A = \{a_1, a_2, \ldots a_n\}$ şi $B = \{F_1, F_2, \ldots F_k\}$. Într-adevăr orice funcţie surjectivă induce o partiţie a mulţimii A în k clase:

$$f^{-1}(F_1), f^{-1}(F_2), \ldots, f^{-1}(F_k)$$

Evident, dacă permutăm elementele mulţimii B obţinem aceeaşi partiţie, deci la o partiţie corespund $k!$ funcţii surjecticve. De unde rezultă formula de mai sus. Timând cont de formula pentru numărul funcţiilor surjective, obţinem următoare formulă pentru numerele lui Stirling de speţa a doua:

$$S(n,k) = \frac{1}{k!} \sum_{i=0}^{k-1} (-1)^i \binom{k}{i} (k-i)^n$$

Numerele lui Stirling de speţa a doua verifică următoarea relaţie de recurenţă:

$$S(n+1,k) = \sum_{i=k-1}^{n} \binom{n}{i} S(i,k-1), \qquad \text{pentru } k \geq 2. \quad (59)$$

7.4 Numerele lui Bell

După cum ştim, numărul $S(n,k)$ reprezintă numărul partiţiilor unei mulţimi de n elemente în k clase. Atunci numărul

$$F_n = \sum_{k=1}^{n} S(n,k)$$

reprezintă numărul partiţiilor unei mulţimi de n elemente, şi se numeşte *numărul lui Bell.*

Numerele lui Bell verifică următoarea relaţie de recurenţă:

$$F_{n+1} = \sum_{k=0}^{n} \binom{n}{k} F_k$$

unde prin definiţie $F_0 = 1$. Plecând de la definiţia numerelor F_{n+1} şi folosind formula (59), obţinem:

$$F_{n+1} = \sum_{k=1}^{n+1} S(n+1,k) = 1 + \sum_{k=2}^{n+1} \sum_{i=k-1}^{n} \binom{n}{i} S(i,k-1)$$

Prin schimbarea ordinii de însumare obţinem:

$$F_{n+1} = 1 + \sum_{i=1}^{n} \binom{n}{i} \sum_{k=2}^{i+1} S(i,k-1) = 1 + \sum_{i=1}^{n} \binom{n}{i} F_i,$$

de unde, ţinând cont de faptul că $F_0 = 1$ se obţine formula dorită.

7.5 Generarea secvenţelor Catalan

Pentru generarea tuturor secvenţelor Catalan de o anumită lungime pornim de la ideea că dacă într-secvenţă Catalan schimbăm o secvenţă 10 în 01 obţinem tot o secvenţă Catalan.

Următorul algoritm recursiv foloseşte următoarele proceduri şi funcţii:

– procedura **push(x)** pune secvenţa x în stivă,

– funcţia **pop** scoate un element din stivă,

– funcţia **Cauta(x,i)** caută în secvenţa x începând de la poziţia i prima secvenţa 10 şi returnează poziţia respectivă,

– funcţia **Schimbă(x,j)** schimbă în x secvenţa 10 în poziţiile j şi $j+1$ în secvenţa 01, şi returnează noua secvenţă.

Algoritmul este următorul:

procedura Catalan (x):
    push (x)
    cât timp stiva nu este vidă **execută**
      $x :=$ pop
      $i := 1$
      cât timp $i \leq$ lungime(x) **execută**
            $j :=$ Caută(x, i)
            dacă $j > 0$ **atunci**
                  cheamă Catalan(Schimbă(x, j))
            $i := i + 2$
sfârşit procedură

Procedura se apelează pentru secvenţa $0101\ldots01$. Pentru 01010101 secvenţele generate vor fi următoarele:

```
01010101
00110101
00101101
00011101
00011011
00010111
00001111
00101011
00100111
00110011
01001101
01001011
01000111
01010011
```

Probleme.

1. Să se demonstreze următoarele identităţi:

a. $F_2 + F_4 + \ldots + F_{2n} = F_{2n+1} - 1$

b. $F_1 + F_3 + \ldots + F_{2n-1} = F_{2n}$

c. $F_1^2 + F_2^2 + \ldots + F_n^2 = F_n F_{n+1}$

d. $F_1F_2 + F_2F_3 + \ldots + F_{2n-1}F_{2n} = F_{2n}^2$

e. $F_1F_2 + F_2F_3 + \ldots + F_{2n}F_{2n+1} = F_{2n+1}^2 - 1$

f. $F_3 + F_6 + \ldots + F_{3n} = \dfrac{F_{3n+2} - 1}{2}$

2. Să se demonstreze că $f_{nk}^{(l)}$ se divide prin $f_k^{(l)}$.

3. Să se demonstreze formulele:

a. $\sum_{k\geq 0} \binom{n}{k} F_k = F_{2n}$

b. $\sum_{k\geq 0} \binom{n}{k} F_{m+k} = F_{m+2n}$

4. Să se demonstreze că F_{3k} este par, iar $F_{3k\pm 1}$ impar pentru orice $k \in \mathbf{N}$.

8 Combinatorica cuvintelor

8.1 Cuvinte finite

Fie A o mulţime finită de simboluri, numită *alfabet*. Elementele lui A se numesc *litere* sau *simboluri*. O secvenţă $a_1a_2\ldots a_n$ de elemente din A se numeşte *cuvânt*. Lungimea cuvântului $u = a_1a_2\ldots a_n$ este n şi se notează cu $|u|$. Cuvântul care nu conţine nici-o literă se numeşte *cuvânt vid* şi se notează cu λ (în unele lucrări cu ε). Mulţimea tuturor cuvintelor finite care se poate forma cu literele alfabetului A se notează cu A^*. Se mai folosesc şi notaţiile:

$$A^+ = A^* \setminus \{\lambda\}, \qquad A^n = \Big\{u \in A^* \;\Big|\; |u| = n\Big\} = \Big\{a_1a_2\ldots a_n \mid a_i \in A\Big\},$$

adică A^+ este mulţimea tuturor cuvintelor finite şi nevide formate cu literele alfabetului A, iar A^n este mulţimea tuturor cuvintelor de lungime n. (Evident $A^0 = \{\lambda\}$.)

Introducem în mulţimea A^* operaţia binară de *concatenare* sau *produs*. Dacă $u = a_1a_2\ldots a_n$ şi $v = b_1b_2\ldots b_m$, atunci

$$w = uv = a_1a_2\ldots a_nb_1b_2\ldots b_m, \qquad |w| = |u| + |v|.$$

Această operaţie este asociativă, dar nu este comutativă. Are ca element neutru λ: $\lambda u = u\lambda = u$. A^* înzestrată cu această operaţie este un monoid. Puterea unui cuvânt u se defineşte astfel:

- $u^0 = \lambda$
- $u^n = u^{n-1}u$, pentru $n \geq 1$.

Un cuvânt se numeşte *cuvânt primitiv*, dacă nu este puterea a nici unui cuvânt, adică u este primitiv dacă

$$u = v^n,\ v \neq \lambda \quad \Rightarrow \quad n = 1.$$

Cuvântul $u = abcab$ este primitiv, pe când $v = abcabc = (abc)^2$ nu.

Două cuvinte u şi v se numesc *conjugate*, dacă există cuvintele p şi q astfel încât $u = pq$ şi $v = qp$. În acest caz u se obţine din v prin permutare circulară. Conjugarea este evident o relaţie de echivalenţă. Cuvintele $u = abcda$ şi $v = cdaab$ sunt conjugate.

Un cuvânt $u = a_1a_2\ldots a_n$ este *periodic* dacă există un $p \geq 1$ astfel încât

$$a_i = a_{i+p},\ \text{pentru } i = 1, 2\ldots n-p.$$

Cel mai mic p cu proprietatea de mai sus se numeşte *perioada* cuvântului. Orice alt p se numeşte *o perioadă*. Cuvântul $u = abcabca$ este periodic cu perioada $p = 3$.

Să notăm, ca deobicei, prin (a, b) cel mai mare divizor comun al numerelor naturale a şi b. Avem următorul rezultat imediat.

Propoziţia 1. *Dacă u este periodic cu o perioadă p şi cu o perioadă q, atunci este periodic şi cu perioada (p, q).*

Introducem şi operaţia unară de *oglindire*. Dacă $u = a_1a_2 \dots a_n$, atunci $u^R = a_n a_{n-1} \dots a_1$, adică u^R este *reversul* lui u. Evident că $\left(u^R\right)^R = u$. Dacă $u = u^R$, atunci u se numeşte *palindrom*.

Atât mulţimea A^* cât şi A^+ sunt mulţimi infinite numărabile. Cuvintele din A^* pot fi aranjate în ordine după lungime, iar în cazul lungimilor egale, în ordine alfabetică, dacă în mulţimea A se introduce o ordine a literelor.

Cuvântul u este un *factor* sau *subcuvânt* al cuvântului v, dacă există cuvintele p şi q astfel încât $v = puq$. Dacă $pq \neq \lambda$, atunci u este un factor propriu al lui v. Dacă $p = \lambda$, atunci u este un *prefix* al lui v, iar dacă $q = \lambda$, atunci u este un *sufix* al lui v. Mulţimea factorilor de lungime n ale cuvântului u se notează cu $F_n(u)$. $F(u)$ este mulţimea tuturor factorilor ale lui u. Deci

$$F(u) = \bigcup_{n=1}^{|u|} F_n(u).$$

De exemplu, dacă $u = abaab$, atunci

$F_1(u) = \{a, b\},\ F_2(u) = \{ab, ba, aa\},\ F_3(u) = \{aba, baa, aab\},$
$F_4(u) = \{abaa, baab\},\ F_5(u) = \{abaab\}.$

Două cuvinte $u = a_1a_2 \dots a_m$ şi $v = b_1b_2 \dots b_n$ sunt egale dacă

- $m = n$ şi
- $a_i = b_i$ pentru $i = 1, 2, \dots, n$.

Teorema 10 (Fine–Wilf). *Fie u şi v două cuvinte de lungime n respectiv m. Dacă există p şi q astfel încât u^p şi v^q să aibă un prefix comun de lungime $n + m - (n, m)$, atunci u şi v sunt puteri ale aceluiaşi cuvânt.*

Demonstraţie. Este suficient să facem demonstraţia pentru cazul $(n, m) = 1$. În acest caz ambele cuvinte vor fi puteri ale unei litere. Dacă $(n, m) = d \neq 1$,

atunci în loc de alfabetul A se consideră ca alfabet mulţimea A^d şi demontraţia rămâne valabilă. Fie $u = a_1a_2 \dots a_n,\ v = b_1b_2 \dots b_m,\ n < m$. Distingem două cazuri:

1) $\frac{m}{2} < n < m$
2) $n < \frac{m}{2}$.

Cazul egalitate nu avem, deoarece $(m, n) = 1$.

1) Cazul $\frac{m}{2} < n < m$. În acest caz avem situaţia:

$$\begin{array}{llllllllll} x = u^p = a_1 & a_2 & \dots & a_n & a_1 & a_2 & \dots & a_{m-n} & a_{m-n+1} \dots \\ y = v^q = b_1 & b_2 & \dots & b_n & b_{n+1} & b_{n+2} & \dots & b_m & b_1 \dots \end{array}$$

Se obţine:

$$a_1 = b_{n+1} = b_1$$
$$a_2 = b_{n+2} = b_2$$
$$\dots$$
$$a_{m-n} = b_{n+(m-n)} = b_m = b_{m-n},$$

deci cuvântul $b_1b_2 \dots b_m$ este periodic cu o perioadă n. Continuând, avem

$$a_{m-n+1} = b_1 = b_{m-n+1}$$
$$a_{m-n+2} = b_2 = b_{m-n+2}$$
$$\dots$$
$$a_{m-n+(2n-m)} = b_{2n-m} = b_{m-n+(2n-m)} = b_n$$

Deci cuvântul $b_1b_2 \dots b_n$ este periodic cu o periodă $m - n$, dar $(m, n) = (m, m - n) = 1$, deci cuvântul este periodic şi cu 1, adică toate literele sale sunt egale. Dar în acest caz toate cele n litere ale cuvântului u sunt egale, deci şi cele ale lui v, din cauza prefixului comun de lungime $m + n - 1$.

2) Cazul $n < \frac{m}{2}$. Avem

$$\begin{array}{llllllll} x = u^p = a_1 \dots a_n & a_1 & \dots a_n & a_1 \dots & a_n & a_1 \dots & & a_{m-ln} \\ y = v^q = b_1 \dots b_n & b_{n+1} & \dots b_{2n} & b_{2n+1} \dots & b_{3n} & b_{3n+1} \dots & & b_m \end{array}$$

Demostraţia este asemănătoare, doar la partea a doua avem

$$b_{ln+1} = a_1 = b_1$$
$$b_{ln+2} = a_2 = b_2$$
$$\dots$$
$$b_{ln+(m-ln)} = a_{m-ln} = b_m,$$

deci cuvântul $b_1b_2\dots b_m$ este periodic cu o perioadă $m-ln$, dar $(m,n) = (m, m-nl) = 1$, dar el este periodic şi cu o perioadă n, deci toate literele ale cuvântului sunt egale între ele. □

În teoremă, valoarea $n+m-(n,m)$ este optimă. Dăm un exemplu în care cele două puteri au un prefix comun de lungime $n+m-(n,m)-1$ şi u şi v nu sunt puteri ale aceluiaşi cuvânt. Fie

$u = abaab, \quad m = |u| = 5, \quad u^2 = abaababaab$
$v = aba, \quad n = |v| = 3, \quad v^3 = abaabaaba$

Conform teoremei, un prefix comun de lungime 7 ar asigura ca cele două cuvinte să fie puteri ale aceluiaşi cuvânt. Se vede că u^2 şi v^3 au un prefix comun de lungime 6 ($abaaba$) şi nu există nici un cuvânt x ca u şi v să fie puteri ale lui x. Deci lungimea prefixului comun nu poate fi micşorată.

8.2 Cuvinte infinite

Pe lângă cuvinte finite vom considera şi cuvinte infinite la dreapta

$$u = u_1u_2\dots u_n\dots$$

Vom nota cu A^ω mulţimea cuvintelor infinite peste alfabetul A. Uneori vom avea nevoie să le tratăm împreună cuvintele finite şi infinite, pentru care vom utiliza notaţia:

$$A^\infty = A^* \cup A^\omega.$$

Şi în acest caz se definesc noţiunile de factor, sufix, prefix, asemănător cazului finit.

Fie $u \in A^\omega$. Cuvântul $v \in A^+$ este un factor al lui u, dacă există cuvintele $p \in A^*$, $q \in A^\omega$, astfel încât $u = pvq$. Dacă $p \neq \lambda$, atunci p este un prefix al lui u, iar q un sufix al lui u. Deasemenea $F_n(u)$ reprezintă mulţimea factorilor de lungime n ale lui u.

Exemple de cuvinte infinite:

1) ***Cuvântul putere***, definit prin

$$p = 010011000111\dots\underbrace{0\dots0}_{n}\underbrace{1\dots1}_{n}\dots = 010^21^20^31^3\dots0^n1^n\dots$$

Se poate vedea că

$F_1(p) = \{0, 1\},\ F_2(p) = \{01, 10, 00, 11\},$
$F_3(p) = \{010, 100, 001, 011, 110, 000, 111\}, \ldots$

2) ***Cuvântul Fibonacci*** se defineşte prin procedeul următor:

$f_0 = 0,\ f_1 = 01$
$f_n = f_{n-1}f_{n-2}$ pentru $n \geq 2$

Se obţin următoarele cuvinte finite:

$f_0 = 1$
$f_1 = 01$
$f_2 = 010$
$f_3 = 01001$
$f_4 = 01001010$
$f_5 = 0100101001001$
$f_6 = 010010100100101001010$
$f_7 = 01001010010010100101001001010010010$

Cuvântul Fibonacci infinit se obţine dacă se trece la limită:

$$f = \lim_{n\to\infty} f_n.$$

Factorii acestui cuvânt sunt:

$F_1(f) = \{0, 1\},\ F_2(f) = \{01, 10, 00\},\ F_3(f) = \{010, 100, 001, 101\},$
$F_4(f) = \{0100, 1001, 0010, 0101, 1010\}, \ldots$

Denumirea se justifică prin modul de generare a acestor cuvinte finite, cât şi prin faptul că $|f_n| = F_{n+2}$, adică lungimea cuvântului f_n este un număr Fibonacci, al $(n+2)$-lea număr Fibonacci.

Acest cuvânt Fibonacci infinit are o serie de proprietăţi interesante. Din construcţia cuvântului rezultă că nu conţine nici un factor care să aibă două 1-uri consecutive (totdeaunea se concatenează două cuvinte care încep cu câte un 0).

Pentru un factor x al lui f se notează cu $h(x)$ numărul cifrelor 1 din u. Un cuvânt infinit u se numeşte *balansat*, dacă pentru orice factori x şi y de aceeaşi lungime din u avem $|h(x) - h(y)| \leq 1$, adică

$$x, y \in F_n(u) \ \Rightarrow\ |h(x) - h(y)| \leq 1.$$

Propoziţia 2. *Cuvântul Fibonacci f este balansat.*

Demonstraţie. Se demonstrează prin inducţie completă asupra lungimii factorilor. Evident pentru $n = 1$ enunţul are loc. presupenem adevărat pentru orice factor de lungime mai mică de n.

Un factor de lungime n se poate termina în 00, 01 sau 10. Considerăm cazurile:

1) $x = u00$ şi $y = v10$.

Factorul x se poate continua doar cu 1, deci $x' = u001$, factorul y poate fi continuat şi cu 0 şi cu 1: $y' = v100$, $y'' = v101$. Avem

$h(x') - h(y') = h(x) + 1 - h(y) = h(x) - h(y) + 1$, dar $h(x) - h(y) \neq 1$, deoarece altfel $h(u) - h(v) = 2$, ceea ce contravine ipotezei inducţiei. Deci $|h(x') - h(y')| \leq 1$.

$h(x') - h(y'') = h(x) + 1 - h(y) - 1 = h(x) - h(y)$, deci conform ipotezei inducţiei $|h(x') - h(y'')| \leq 1$.

2) $x = u00$ şi $y = v01$.

În acest caz ambii factori pot fi continuaţi într-un singur mod, deci $x' = u001$, $y' = v010$. Avem:

$h(x') - h(y') = h(x) + 1 - h(y)$, dar $h(x) - h(y) \neq 1$, deoarece atunci $h(u) - h(v) = 2$, ceea ce nu se poate, din cauza ipotezei inducţiei. Deci şi $|h(x') - h(y')| \leq 1$.

3) $x = u10$ şi $y = v01$.

Ca şi în primul caz $x' = u100$ şi $x'' = u101$, iar $y' = v010$. Avem cazurile:

$h(x') - h(y') = h(x) - h(y)$

$h(x'') - h(y') = h(x) + 1 - h(y)$, dar $h(x) - h(y) = 1$ ar implica $h(u1) - h(v0) = 2$, ceea ce este o contradicţie cu ipoteza inducţiei. Deci, avem:

$|h(x') - h(y')| \leq 1$ şi $|h(x'') - h(y')| \leq 1$,

ceea ce demonstrează propoziţia. □

Propoziţia 3. *$F_n(f)$ are $n + 1$ elemente.*

Demonstraţie. Am văzut în demonstraţia precedentă că orice factor se termină în 00, 01 sau 10. Dintre aceştia, factorii care se termină în 00 sau 01 se pot continua numai într-un singur fel, iar factorii care se termină în 10 se pot continua în două moduri. Deci $\#F_{n+1}(f) \leq \#F_n(f) + 1$. Toate aceste cuvinte sunt într-adevăr factori, deci $\#F_{n+1}(f) = \#F_n(f) + 1$. De unde rezultă că $\#F_n(f) = n + 1$. □

Dacă un cuvânt finit u se concatenează cu el însuşi de un număr infinit de ori, acest lucru se notează prin u^ω.

Un cuvânt infinit u se numeşte *periodic*, dacă există un cuvânt finit v, astfel încât $u = v^\omega$. Această noţiune este o generalizare a noţiunii de periodicitate de la cuvintele finite. Cuvântul u este *sufixperiodic*, dacă există cuvintele finite v şi w astfel încât $u = vw^\omega$.

Cuvântul Fibonacci poate fi generat şi cu ajutorul unui homomorfism. Se defineşte un homomorfism ca fiind funcţia

$$h : A^* \to A^*, \quad h(uv) = h(u)h(v), \quad \forall u, v \in A^*.$$

Este suficient să definim homomorfismul numai pentru literele alfabetului. Un homomorfism poate fi extins şi la cuvintele infinite

$$h : A^\omega \to A^\omega, \quad h(uv) = h(u)h(v), \quad \forall u \in A^*, v \in A^\omega.$$

Cuvintele Fibonacci f_n pot fi obţinute pornind de la homomorfismul:

$$\sigma(0) = 01,\ \sigma(1) = 0.$$

În acest caz avem

Propoziţia 4. $f_{n+1} = \sigma(f_n)$

Demonstraţie. Demonstraţia se face prin inducţie completă. Evident $f_1 = \sigma(f_0)$. Presupunem adevărată afirmaţia $f_k = \sigma(f_{k-1})$ pentru orice $k \leq n$. Deorece

$$f_{n+1} = f_n f_{n-1},$$

conform ipotezei inducţiei avem

$$f_{n+1} = \sigma(f_{n-1})\sigma(f_{n-2}) = \sigma(f_{n-1}f_{n-2}) = \sigma(f_n).$$

□

De unde imediat rezultă

Propoziţia 5. $f_n = \sigma^n(0)$

Cuvântul Fibonacci f este punctul fix al homomorfismul σ:

$$f = \sigma(f).$$

8.3 Grafuri de cuvinte

Fie $X \subseteq A^m$ o mulţime de cuvinte de lungime m peste alfabetul A şi $E \subseteq AX \cap XA$. Definim graful orientat ataşat acestor două mulţimi ca fiind graful care are ca vârfuri cuvintele din X şi ca arce cuvintele din E. Există arc de la vârful $a_1a_2 \ldots a_m$ la vârful $b_1b_2 \ldots b_m$ dacă

$$a_2 = b_1, \;\; a_3 = b_2, \;\; \ldots, \;\; a_m = b_{m-1} \text{ şi } a_1a_2 \ldots a_mb_m \in E$$

adică ultimele $m-1$ litere ale primului cuvânt se suprapun cu primele $m-1$ litere ale celui de al doilea cuvânt. Acest arc este etichetat cu $a_1a_2 \ldots a_mb_m$ (sau, ceea ce este acelaşi lucru, cu $a_1b_1 \ldots b_m$).

Grafuri de Bruijn

Dacă $X = A^m$ şi $E = A^{m+1}$, unde $A = \{a_1, a_2, \ldots a_n\}$, graful obţinut se numeşte *graf de Bruijn* $B(n, m)$.

Grafurile de Bruijn $B(2, 2)$ şi $B(2, 3)$ se găsesc în figurile 1 şi 2. În figura 3 este reprezentat graful $B(3, 2)$.

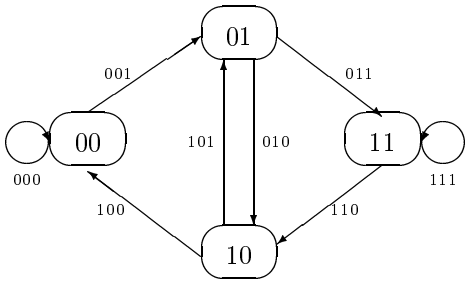


Figura 1: Graful de Bruijn $B(2, 2)$.

Ataşăm unui drum $x_1x_2 \ldots x_m, \;\; x_2x_3 \ldots x_mx_{m+1}, \;\; \ldots, \;\; z_1z_2 \ldots z_m$ într-un graf de Bruijn cuvântul $x_1x_2 \ldots z_{m-1}z_m$, adică concatenarea cu suprapunere maximă a cuvintelor care corespund vârfurilor. În graful $B(2, 3)$ din figura 2 drumului $001, 011, 111, 110$ îi corespunde cuvântul 001110. Cuvântul corespunzător unui drum Hamiltonian (drum ce trece prin fiecare vârf o singură dată) în graful $B(2, 3)$ se numeşte *cuvânt de Bruijn de tip (n,m)*. De exemplu 0001110100 şi 0001011100 sunt cuvinte de Bruijn de tip $(2, 3)$. Un cuvânt de Bruijn de tip (n, m) conţine toate subcuvintele de lungime m.

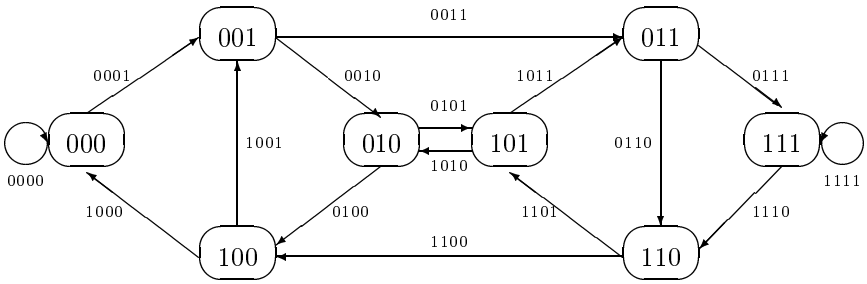


Figura 2: Graful de Bruijn $B(2,3)$.

Un graf orientat conex[1] este Eulerian[2] dacă fiecare vârf are gradul interior egal cu gradul exterior[3].

Teorema 11. *Graful de Bruijn $B(n,m)$ este Eulerian.*

Demonstraţie. a) Graful este conex, deoarece pentru oricare două vârfuri $x_1x_2\ldots x_m$ şi $z_1z_2\ldots z_m$ există un drum între ele. Din vârful $x_1x_2\ldots x_m$ pornesc n arce către vârfuri ce au toate primul caracter diferit. Astfel există drumul: $x_1x_2\ldots x_m,\ \ x_2x_3\ldots x_mz_1,\ \ \ldots,\ \ \ x_mz_1\ldots z_{m-1},\ \ z_1z_2\ldots z_m$.

b) În vârful oarecare $x_1x_2\ldots x_m$ intră arce din vârfurile $yx_1\ldots x_{m-1}$, unde $y\in A$ (A alfabetul de bază al grafului, adică $X=A^m$). Arcele ce pornesc din vârful $x_1x_2\ldots x_m$ au extremitatea finală în $x_2x_3\ldots x_my$ cu $y\in A$. Deci graful este Eulerian. □

Avem şi rezultatul imediat:

Teorema 12. *Cuvintele ataşate arcelor într-un drum Eulerian în graful $B(n,m)$ corespund, în aceeaşi ordine, unui drum Hamiltonian în graful $B(n,m+1)$.*

De exemplu drumul format din arcele 000, 001, 010, 101, 011, 111, 110, 100 este Eulerian în $B(2,2)$, iar aceleaşi cuvinte considerate ca vârfuri, corespunde unui drum Hamiltonian în graful $B(2,3)$.

[1]Un graf orientat este conex dacă între oricare două vârfuri ale lui există drum orientat cel puţin într-o direcţie.

[2]Un graf orientat este Eulerian dacă conţine un circuit ce trece prin fiecare arc al grafului o singură dată.

[3]Adică numărul arcelor ce au extremitatea iniţială în vârful dat este egal cu cel al arcelor ce au extremitatea finală în acel vârf.

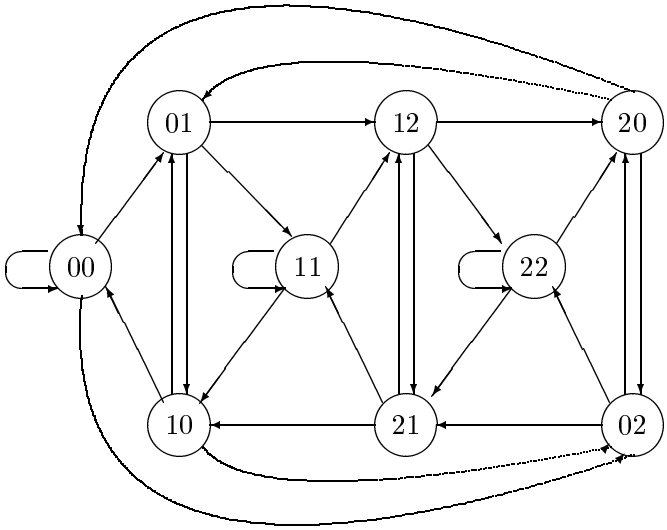


Figura 3: Graful de Bruijn $B(3,2)$.

Pentru generarea cuvintelor de Bruijn există multe metode. Vom prezenta algoritmul din [51]. Fie alfabetul $A = \{a_1, a_2, \ldots, a_n\}$. Dorim să generăm un cuvânt de Bruijn de tip (n, m) peste acest alfabet.

Se porneşte cu secvenţa $\underbrace{a_1 a_1 \ldots a_1}_{m ori}$, la care se adaugă prin concatenare la dreapta câte un caracter în felul următor: se adaugă caracterul a_k cu indice k cel mai mare posibil pentru care subcuvântul de pe ultimele m poziţii este unic în cuvântul deja generat (nu apare de două ori în cuvântul generat). Se continuă până când nici un caracter nu mai poate fi adăugat.

Evident, lungimea cuvântului generat va fi $n^m + m - 1$.

Exemplu. Fie $A = \{0, 1\}$. Se caută un cuvânt de Bruijn de tip $(2, 3)$.
Se porneşte cu 000.
Se adaugă 1. Cuvântul generat: 0001.
Se adaugă 1. Cuvântul generat: 00011.
Se adaugă 1. Cuvântul generat: 000111.

Se adaugă 0, pentru că altfel 111 apare de două ori (suprapus). Cuvântul generat: 0001110.
Se adaugă 1. Cuvântul generat: 00011101.
Se adaugă 0, pentru că altfel 011 apare de două ori. Cuvântul generat: 000111010.
Se adaugă 0, pentru că altfel 101 apare de două ori (suprapus). Cuvântul generat: 0001110100. Nu se mai poate continua, nici cu 1, nici cu 0.

Deci cuvântul 0001110100 este cel căutat. Se poate observa că şi cuvântul 0001011100 este cuvânt de Bruijn de acelaşi tip. Din aceste două, prin permutări circulare, se pot obţine toate cuvintele de Bruijn de tip (2,3).

Pe baza algoritmului de mai sus uşor se poate demonstra şi rezultatul următor.

Teorema 13. *Un cuvânt de Bruijn de tip (n, m) este cuvântul cel mai scurt care conţine toate subcuvintele de lungime m peste un alfabet de n caractere.*

Reţele de calculatoare. Grafurile de Bruijn reprezintă un model corespunzător pentru reţelele de calculatoare. Varianta neorientată (fig. 4) poate sta la baza unor reţele de calculatoare. Graful este multiplu conex, adică fiecare muchie se găseşte pe cel puţin un ciclu, deci chiar prin eliminarea unei muchii (întreruperea legăturii directe între cele două noduri) graful rămâne conex. Diametrul grafului este m, adică se poate ajunge din orice nod în orice nod printr-un drum de lungime cel mult m. Evident, în cazul reţelelor muchiile multiple şi buclele se elimină.

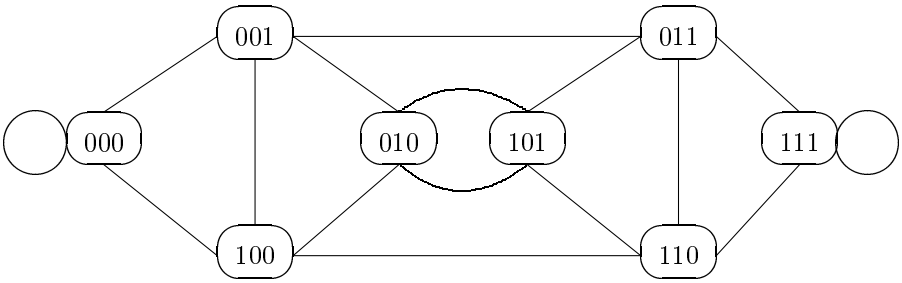


Figura 4: Graful de Bruijn neorientat $B^*(2, 3)$.

Un alt model eficient de reţele de calculatoare este hipercubul (fig. 5). În [55] se studiază proprietăţile grafurilor de Bruijn generalizate.

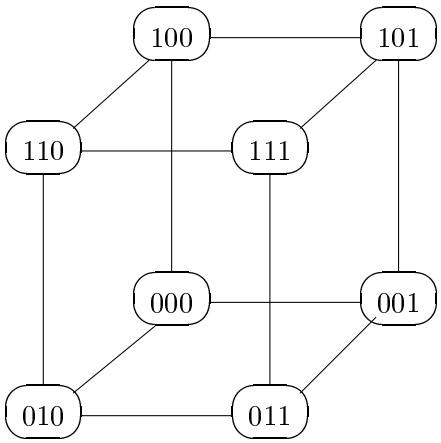


Figura 5: Hipercub trideminsonal.

Grafuri Rauzy

Dacă u este un cuvânt infinit, şi se consideră $X = F_n(u)$, $E = F_{n+1}(u)$ atunci graful de cuvinte obţinut se numeşte *graf Rauzy* (sau *graf factor*). În figura 6 sunt reprezentate grafurile Rauzy pentru cuvântul Fibonacci în cazul $n = 1, 2, 3, 4, 5$. Cuvântul Fibonacci, cum am mai văzut, este

$$f = 01001010010010100 10 \ldots,$$

şi $F_1(f) = \{0, 1\}$, $F_2(f) = \{01, 10, 00\}$,
$F_3(f) = \{010, 100, 001, 101\}$, $F_4(f) = \{0100, 1001, 0010, 0101, 1010\}$,
$F_5(f) = \{01001, 10010, 00101, 01010, 10100, 00100\}$.
Pentru cuvântul putere

$$p = 0100110001110000111100000111 11 \ldots \underbrace{0 \ldots 0}_{n} \underbrace{1 \ldots 1}_{n} \ldots$$

avem de exemplu
$F_1(p) = \{0, 1\}$, $F_2(p) = \{01, 10, 00, 11\}$,
$F_3(p) = \{010, 100, 000, 001, 011, 111, 110\}$,
$F_4(p) = \{0100, 1001, 0011, 0110, 1100, 1000, 0000, 0001, 0111, 1110, 1111\}$,
iar grafurile Rauzy corepunzătoare se găsesc în figura 7.

După cum se poate vedea uşor din exemplele din figurile 6 şi 7, există factori de lungime n care se pot continua într-un singur fel în cuvântul infinit

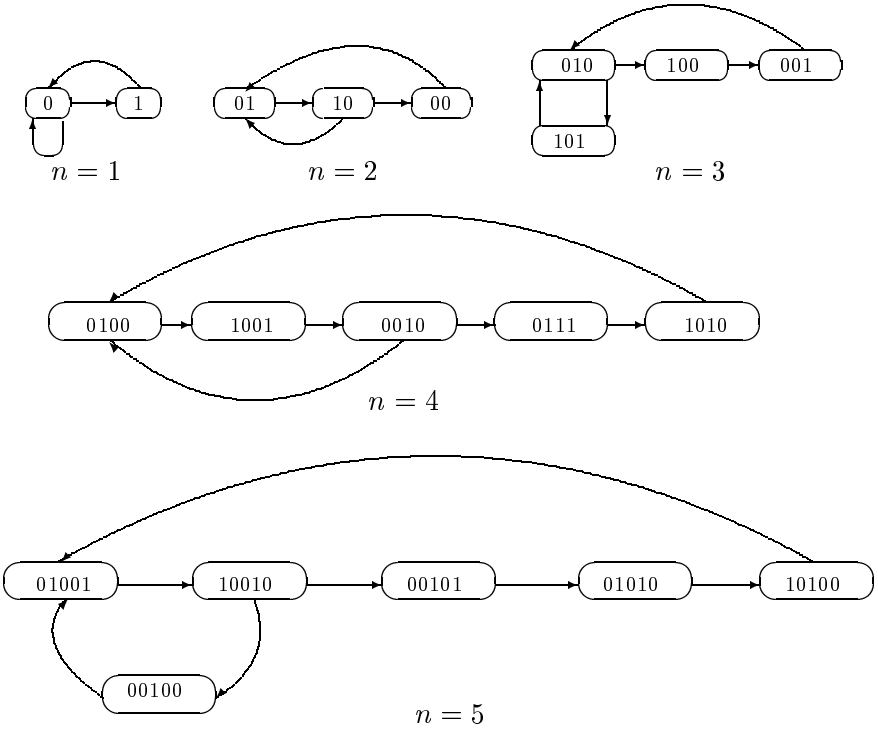


Figura 6: Graful Rauzy pentru cuvântul Fibonacci

(prin adăugarea unei singure litere), acestea corespund nodurilor din care iese câte un singur arc, şi sunt factori care se pot continua în două moduri, care corespund nodurilor din care ies două arce. Astfel se poate defini *factorul special la dreapta* ca fiind un factor $v \in F_n(u)$ pentru care există cel puţin două litere $a \in A$ astfel ca $va \in F_{n+1}(u)$. *Factorul special la stânga* este un factor $v \in F_n(u)$ pentru care există cel puţin două litere $a \in A$ astfel ca $av \in F_{n+1}(u)$. Un factor este *bispecial* dacă este factor special la dreapta şi totodată şi factor special la stânga. Câteva exemple de factori speciali din figurile 6 şi 7:

factori speciali la stânga: 0100, 01001 (*fig. 6*), 10, 110, 1110, 0001 (*fig. 7*)

factori speciali la dreapta: 0010, 10010 (*fig. 6*), 01, 011, 0111 (*fig. 7*)

factori bispeciali: 010, 00100 (*fig. 6*), 000, 111, 1111, 0011 (*fig. 7*)

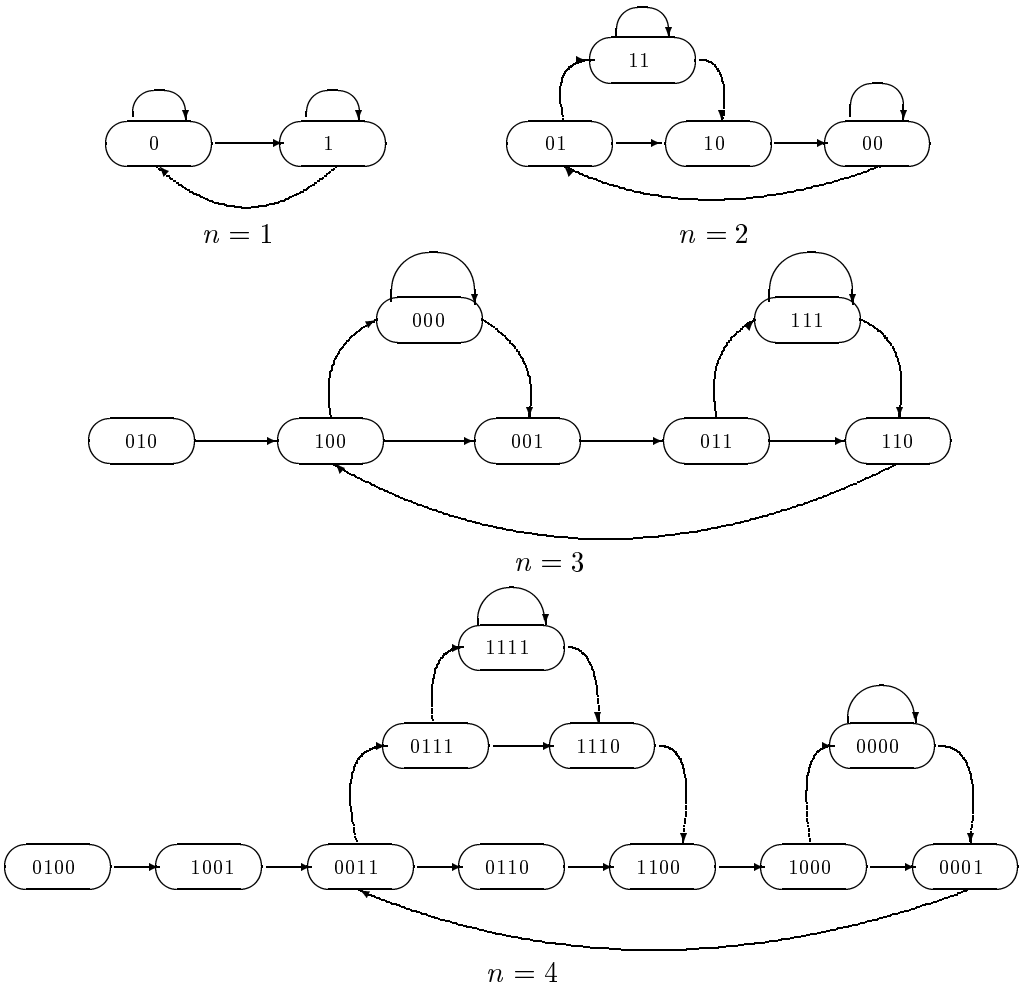


Figura 7: Graful Rauzy pentru cuvântul putere.

8.4 Complexitatea cuvintelor

Prin complexitatea cuvintelor înţelegem o măsură care poate caracteriza diversitatea factorilor. Vom folosi următoarele măsuri pentru complexitate.

1) *Complexitatea factorială* sau simplu *complexitatea* unui cuvânt se defineşte ca fiind numărul tuturor factorilor distincţi de o anumită lungime ale unui cuvânt dat, care se notează cu $f_u(n)$.

$$f_u(n) = \#F_n(u), \quad u \in A^\infty$$

Avem $f_u(n) = 0$ pentru $n > |u|$ sau $n = 0$.

2) *Complexitatea maximă* se defineşte numai pentru cuvinte finite, ca fiind

$$C(u) = \max\{f_u(n) \mid n \geq 1\}, \quad u \in A^*$$

Pentru cuvinte infinite se defineşte *complexitatea maximă superioară* $C_u^+(n)$ respectiv *complexitatea maximă inferioară* $C_u^-(n)$.

$$C_u^+(n) = \max_i C(u_i u_{i+1} \dots u_{i+n-1}), \quad C_u^-(n) = \min_i C(u_i u_{i+1} \dots u_{i+n-1})$$

3) *Complexitatea maximă globală* în A^n se defineşte ca fiind

$$G(n) = \max\{C(u) \mid u \in A^n\}$$

4) *Complexitatea totală* se defineşte pentru cuvinte finite şi numără toţi factorii distincţi ai unui cuvânt.

$$K(u) = \sum_{i=1}^{|u|} f_u(i), \quad u \in A^*$$

Pentru cuvinte infinite se defineşte *complexitatea totală superioară* $K_u^+(n)$ respectiv *complexitatea totală inferioară* $K_u^+(n)$.

$$K_u^+(n) = \max_i K(u_i u_{i+1} \dots u_{i+n-1}), \quad K_u^-(n) = \min_i K(u_i u_{i+1} \dots u_{i+n-1})$$

5) *d-complexitatea*

Pentru a defini d-complexitatea [35] să generalizăm noţiunea de factor. Fie $d, k, s \in \mathbf{N}$, $u = a_1 a_2 \dots a_k \in A^k$. Un cuvânt $v = a_{i_1} a_{i_2} \dots a_{i_s}$ este un *d-factor* al lui u dacă

$$\begin{aligned} &i_1 \geq 1, \\ &1 \leq i_{j+1} - i_j \leq d, \qquad \text{pentru } j = 1, 2, \dots, s-1 \\ &i_s \leq k. \end{aligned}$$

Pentru cuvântul $u = abab$ d-factorii sunt:

1-factorii: a, b
2-factorii: ab, aa, ba, bb
3-factorii: aba, abb, aab, bab
4-factorii: $abab$

Vom defini toate măsurile de mai sus şi pentru d-factori, dar numai pentru cuvinte finite.

– *d-complexitatea factorială* sau simplu *d-complexitatea* unui cuvânt se defineşte ca fiind numărul tuturor d-factorilor distincţi de o anumită lungime ale unui cuvânt dat. Se notează cu $f_{u,d}(n)$ această complexitate pentru un cuvânt u, dacă d-factorii sunt de lungime n. Avem $f_{u,d}(n) = 0$ pentru $n > |u|$ sau $n = 0$.

– *d-complexitatea maximă* se defineşte numai pentru cuvinte finite, ca fiind

$$C_d(u) = \max\{f_{u,d}(n) \mid n \geq 1\}$$

– *d-complexitatea maximă globală* în A^n se defineşte ca fiind

$$G_d(n) = \max\{C_d(u) \mid u \in A^n\}$$

– *d-complexitatea totală* numără toţi d-factorii distincţi ai unui cuvânt.

$$K_d(u) = \sum_{i=1}^{|u|} f_{u,d}(i)$$

8.4.1 Complexitatea factorială

Complexitatea factorială este

$$f_u(n) = \#F_n(u), \quad \forall u \in A^\infty,\ n \in \mathbf{N}.$$

Avem $f_u(n) = 0$ pentru $n > |u|$ sau $n = 0$.

De exemplu, pentru cuvântul $u = abacab$:

$$f_u(1) = 3,\ f_u(2) = 4,\ f_u(3) = 4,\ f_u(4) = 3, f_u(5) = 2,\ f_u(6) = 1.$$

Pentru cuvântul Fibonacci, cum am mai văzut în propoziţia 3, avem:

$$f_f(n) = n + 1$$

Pentru cuvântul putere $p = 010011\ldots0^k1^k\ldots$, complexitatea factorială este

$$f_p(n) = \frac{n(n+1)}{2} + 1.$$

Acest lucru se poate demonstra calculând diferenţa $f_p(n+1) - f_p(n)$, care este numărul acelor factori de lungime n care se pot continua în două moduri

pentru a se ajunge la factori de lungime $n+1$. Singurii factori care se pot continua în două moduri sunt de forma 0^k1^{n-k} pentru $k \leq n-k$ şi de forma 1^k0^{n-k} pentru $k < n-k$. Considerând separat cazurile când n este par, respectiv impar, se poate obţine uşor că

$$f_p(n+1) - f_p(n) = n+1.$$

De unde

$$\begin{aligned} f_p(n) &= n + f_p(n-1) = n + (n-1) + f_p(n-2) = \dots \\ &= n + (n+1) + \dots + 2 + f_p(1) = \frac{n(n+1)}{2} + 1 \end{aligned}$$

Şirul

$$\begin{aligned} u_C = u_0u_1 \dots u_n \dots &= 0.1.10.11.100.101.110.111.1000\dots \\ &= 0110111001011101111000\dots, \end{aligned}$$

unde u_i este reprezentarea binară a numărului natural i, care se numeşte *şirul Champernowne*, are complexitatea factorială $f_{u_C}(n) = 2^n$, ceea ce se poate vedea imediat.

Teorema 14. *Dacă pentru un $u \in A^\omega$ există un $n \in \mathbf{N}$ astfel încât $f_u(n) \leq n$ atunci u este sufixperiodic.*

Demonstraţie. Trebuie ca $f_u(1) \geq 2$, pentru că altfel cuvântul u este trivial (constant). Deci trebuie să existe un $k \leq n$ pentru care $f_u(k) = f_u(k+1)$. Dar

$$f_u(k+1) - f_u(k) = \sum_{v \in F_k(u)} \Big(\#\{a \in \Sigma \mid va \in F_{k+1}(u)\} - 1 \Big).$$

Deci orice subcuvânt $v \in F_k(u)$ se poate continua numai într-un singur fel ca să obţinem un $va \in F_{k+1}(u)$. Astfel dacă $v = u_iu_{i+1}\dots u_{i+k-1} = u_ju_{j+1}\dots u_{j+k-1}$ atunci neaparat şi $u_{i+k} = u_{j+k}$. Pentru că $F_k(u)$ este o mulţime finită, iar cuvântul u infinit, vor exista indicii i şi j $(i < j)$ pentru care $u_iu_{i+1}\dots u_{i+k-1} = u_ju_{j+1}\dots u_{j+k-1}$, dar atunci şi $u_{i+k} = u_{j+k}$. Dar atunci din $u_{i+1}u_{i+2}\dots u_{i+k} = u_{j+1}u_{j+2}\dots u_{j+k}$ rezultă $u_{i+k*1} = u_{j+k+1}$, deci $u_{i+l} = u_{j+l}$ pentru orice $l \geq 0$. Şi atunci cuvântul u este sufixperiodic. □

Definiţia 1. *Un cuvânt $u \in A^\omega$ pentru care $f_u(n) = n+1$ pentru orice $n \geq 1$ natural, se numeşte cuvânt sturmian.*

Cuvintele sturmiene sunt cuvintele infinite neperiodice care au complexitate factorială cea mai mică. Evident cuvântul Fibonacci este un cuvânt sturmian. Din $f_u(1) = 2$ rezultă că aceste cuvinte sunt formate din două litere.

Din teorema 14 rezultă că orice cuvânt infinit care nu este sufixperiodic are complexitate factorială cel puţin $n + 1$, adică

$$u \in A^\omega, u \text{ nu este sufixperiodic } \Rightarrow \ f_u(n) \geq n + 1.$$

Cuvintele pentru care avem egalitate sunt cuvintele sturmiene.

Este interesant, că aceeaşi caracterizarea a şirurilor infinite se poate da şi cu ajutorul complexităţilor maxime superioare şi a complexităţilor totale superioare. În [26] se demonstrează următoarele rezultate.

Teorema 15. *Pentru orice cuvânt infinit u care nu este sufixperiodic şi pentru orice $n \geq 1$ avem:*

$$C_u^+(n) \geq \left[\frac{n}{2}\right] + 1, \quad K_u^+(n) \geq \left[\frac{n^2}{4} + n\right].$$

Pentru cuvinte sturmiene avem egalitate.

Să notăm cu $\{x\}$ partea fracţionară a unui număr. Evident că avem $x = \lfloor x \rfloor + \{x\}$. Dacă R este o funcţie, vom nota prin R^n compunerea lui R de n ori cu ea însăşi. Deci $R^n = R \circ R \circ \ldots \circ R$. Atunci cuvintele sturmiene se pot caracteriza în felul următor:

Teorema 16. *Un cuvânt u este sturmian atunci şi numai atunci dacă există un număr iraţional α şi un număr real z astfel încât pentru $R(x) = \{x + \alpha\}$ avem*

$$u_n = \begin{cases} 0, & \text{dacă } R^n(z) \in (0, 1-\alpha) \\ 1, & \text{dacă } R^n(z) \in [1-\alpha, 1) \end{cases}$$

sau

$$u_n = \begin{cases} 1, & \text{dacă } R^n(z) \in (0, 1-\alpha) \\ 0, & \text{dacă } R^n(z) \in [1-\alpha, 1) \end{cases}$$

Pentru cuvântul Fibonacci aceste numere sunt: $\alpha = z = \frac{\sqrt{5}-1}{2}$ (raportul de aur).

Cuvintele sturmiene pot fi generate şi cu ajutorul următorului procedeu. Se lansează o bilă de biliard de pe o latură a unei mese pătrate sub un unghi iraţional. Presupunem că bila se reflectă pe fiecare latură şi îşi continuă mişcarea indefinit. Se notează cu 0 reflecţia pe o latură orizontală şi cu 1 pe o latură verticală. Şirul obţinut este sturmian. Problema se poate generaliza pentru cazul multidimensional, adică cuvântul infinit peste un alfabet cu $s+1$ litere se obţine pe baza traiectoriei bilei de biliard în hipercubul $(s+1)$-dimensional. În [3] şi [6] se demonstrează că în acest caz complexitatea factorială este egală cu

$$f_u(n, s+1) = \sum_{i=0}^{\min(n,x)} \frac{n!s!}{(n-i)!i!(s-i)!}$$

Pentru $s = 1$ se obţine $f_u(n, 2) = f_u(n) = n + 1$, iar pentru $s = 2$, $f_u(n, 3) = n^2 + n + 1$.

8.4.2 Complexitatea maximă

Pentru un cuvânt finit u

$$C(u) = \max\{f_u(n) \mid n \geq 1\}$$

reprezintă complexitatea maximă. În tabelul următor sunt date valorile complexităţilor factoriale, din care rezultă de exemplu: $C(11211122) = 5$, $C(11211211) = 3$ etc.

Cuvântul u	$f_u(1)$	$f_u(2)$	$f_u(3)$	$f_u(4)$	$f_u(5)$	$f_u(6)$	$f_u(7)$	$f_u(8)$
11211122	2	4	5	5	4	3	2	1
11211211	2	3	3	3	3	3	2	1
11211212	2	3	4	4	4	3	2	1
11211221	2	4	5	5	4	3	2	1
11211222	2	4	5	5	4	3	2	1
11212111	2	3	5	5	4	3	2	1
11212112	2	3	4	5	4	3	2	1
11212122	2	4	4	4	4	3	2	1

În [50] sunt studiate complexităţile factoriale şi cea maximă a cuvintelor finite. Un rezultat interesant este următorul.

Teorema 17. *Dacă w este un cuvânt finit, $f_w(n)$ complexitatea lui factorială, atunci există numerele naturale m şi M cu proprietatea că $1 \leq m \leq M \leq |w|$ astfel încât*

- $f_w(n+1) > f_w(n)$ *pentru* $1 \leq n < m$,
- $f_w(n+1) = f_w(n)$ *pentru* $m \leq n < M$,
- $f_w(n+1) = f_w(n) - 1$ *pentru* $M \leq n \leq |w|$.

În tabelul de mai sus se poate vedea că pentru

$w = 11211122$ avem $m = 3$, $M = 4$,

$w = 11212112$ avem $m = 4$, $M = 4$,

$w = 11212122$ avem $m = 2$, $M = 5$.

8.4.3 Complexitatea maximă globală

Cum am văzut, complexitatea maximă globală este

$$G(n) = \max\{C(u) \mid u \in A^n\},$$

adică cea mai mare complexitate (factorială) în mulţimea tuturor cuvintelor de lungimea n pe un alfabet dat. Se pun următoarele întrebări [2]:

– ce lungime au subcuvintele pentru care această complexitate este atinsă,

– câte cuvinte de lungime dată există pentru care această complexitate este atinsă.

Exemple.

Pentru alfabetul $A = \{0, 1\}$ tabelele următoare conţin complexitatea factorială pentru toate cuvintele de lungime 3 respectiv 4.

	$f_u(i)$		
u	$i = 1$	$i = 2$	$i = 3$
000	1	1	1
001	2	2	1
010	2	2	1
011	2	2	1
100	2	2	1
101	2	2	1
110	2	2	1
111	1	1	1

Se vede că în acest caz ($n = 3$) complexitatea maximă globală este 2, şi este atinsă pentru subcuvinte de lungime 1 şi 2. Numărul total al cuvintelor care au complexitatea maximă este 6.

	$f_u(i)$			
u	$i=1$	$i=2$	$i=3$	$i=4$
0000	1	1	1	1
0001	2	2	2	1
0010	2	3	2	1
0011	2	3	2	1
0100	2	3	2	1
0101	2	2	2	1
0110	2	3	2	1
0111	2	2	2	1
1000	2	2	2	1
1001	2	3	2	1
1010	2	2	2	1
1011	2	3	2	1
1100	2	3	2	1
1101	2	3	2	1
1110	2	2	2	1
1111	1	1	1	1

În acest caz al cuvintelor de lungime 4 complexitatea maximă globală este 3, şi este atinsă pentru subcuvinte de lungime 2. Numărul total al acestor cuvinte este 8.

Pentru rezolvarea celor două probleme puse vom folosi următoarele notaţii:

$$R(n) = \{i \in \{1, 2, \ldots, n\} \mid \exists u \in A^n : f_u(i) = G(n)\}$$

$$M(n) = \#\{u \in A^n : C(u) = G(n)\}$$

Tabelul din figura 8 conţine valorile $G(n)$, $R(n)$, $M(n)$ până la lungime de 20 pe un alfabet cu 2 litere.

Teorema 18. *Dacă* $\#A = q$ *şi* $q^k + k \leq n \leq q^{k+1} + k$ *atunci* $G(n) = n - k$.

Demonstraţie. Să considerăm mai întâi cazul $n = q^{k+1} + k,\ k \geq 1$. Există cuvântul de Bruijn w de lungime $q^{k+1} + k$ care conţine toate cele q^{k+1} subcuvinte de lungime $k+1$. Deci $f_w(k+1) = q^{k+1}$. Este evident că

$f_w(l) = q^l < f_w(k+1), \quad l = 1, 2, \ldots, k,$
$f_w(k+1+j) = q^{k+1} - j < f_w(k+1), \quad j = 1, 2, \ldots, q^{k+1} - 1.$

Orice alt cuvânt de lungime $q^{k+1} + k$ va avea complexitatea factorială cel mult $f_w(k+1)$. Deci $G(n) = n - k$.

Să considerăm acum cazul $n = q^{k+1} + k - r$, unde $r = 1, 2, \ldots, q^{k+1} - q^k$, deoarece $q^k + k \leq n \leq q^{k+1} + k$. Dacă eliminăm ultimele r caractere din

n	$G(n)$	$R(n)$	$M(n)$
1	1	1	2
2	2	1	2
3	2	1, 2	6
4	3	2	8
5	4	2	4
6	4	2, 3	36
7	5	3	42
8	6	3	48
9	7	3	40
10	8	3	16
11	8	3, 4	558
12	9	4	718
13	10	4	854
14	11	4	920
15	12	4	956
16	13	4	960
17	14	4	912
18	15	4	704
19	16	4	256
20	16	4, 5	79006

Figura 8: Complexitatea maximă globală atinsă pentru subcuvinte ce au lungime din mulţimea $R(n)$. $M(n)$ reprezintă numărul cuvintelor pentru care complexitatea maximă este atinsă.

cuvântul de Bruijn w de mai sus, obţinem un cuvânt w_n de lungime $n = q^{k+1} + k - r$, care are complexitate factorială $f_{w_n}(k+1) = q^{k+1} - r$. Avem

$$f_{w_n}(l) = q^l \leq q^k \leq q^{k+1} = f_{w_n}(k+1), \quad l = 1, 2, \ldots, k,$$
$$f_{w_n}(k+1+j) = f_{w_n}(k+1) - j < f_{w_n}(k+1), \quad j = 1, 2 \ldots, n-k-1.$$

Orice alt cuvânt de lungime $n = q^{k+1}+k-r$ poate avea complexitate factorială cel mult $q^{k+1} - r$, deci avem $G(n) = q^{k+1} - r = n - k$. □

Teorema 19. *Dacă* $\#A = q$ *şi* $q^k + k < n \leq q^{k+1} + k$ *atunci* $R(n) = \{k+1\}$, *iar dacă* $n = q^k + k$ *atunci* $R(n) = \{k, k+1\}$.

Demonstraţie. Vom folosi metoda utilizată în demonstrarea teoremei 18. În prima parte a demonstraţiei am văzut că pentru $n = q^{k+1} + k$, $k \geq 1$ există un cuvânt w de lungime n pentru care $G(n) = f_w(k+1) = n - k$, ceea ce

înseamnă că $k+1 \in R(n)$. Pentru cuvântul w şi orice alt cuvânt de lungime n avem $f_w(l) \leq f_w(k+1)$, $l \neq k+1$. Rezultă că $R(n) = \{k+1\}$.

Ca şi în partea a doua a demonstraţiei teoremei 18 considerăm $n = q^{k+1} + k - r$ cu $r = 1, 2, \ldots, q^{k+1} - q^k$, şi cuvântul w_n pentru care $G(n) = f_{w_n}(k+1) = q^{k+1} - r$. Avem deci $k+1 \in R(n)$. Pentru $l > k+1$ este evident că pentru orice cuvânt de lungime n avem $f_{w_n}(l) < f_{w_n}(k+1)$. Încercăm să găsim un cuvânt w cu $f_w(k+1) = n-k$ pentru care $f_w(l) = n-k$ pentru $l \leq k$. Avem $f_w(l) \leq q^l \leq q^k \leq q^{k+1} - r$, deci egalitatea $f_w(l) = n - k = q^{k+1} - r$ poate exista numai pentru $l = k$ şi $r = q^{k+1} - q^k$, adică $n = q^k + k$. Pentru $n = q^k + k$ vom avea într-adevăr $R(n) = \{k, k+1\}$. Pornim de la un cuvânt de Bruijn (vezi pag. 118) şi îi adăugăm a literă din alfabet, obţinem evident un cuvânt v de lungime $n = q^k + k$, care conţine toate cele q^k cuvinte de lungime k şi $q^k = n - k$ cuvinte de lungime $k+1$, deci $f_v(k) = f_v(k+1) = G(n)$. □

Teorema 20. *Dacă* $\#A = q$ *şi* $q^k + k \leq n \leq q^{k+1} + k$ *atunci* $M(n)$ *este egal cu numărul drumurilor distincte de lungime* $n - k + 1$ *ale grafului de Bruijn* $B(q, k+1)$.

Demonstraţie. Numărul $M(n)$ al cuvintelor de lungime n care au complexitate maximă globală este egal cu numărul cuvintelor $w \in A^n$ pentru care $f_w(k+1) = n-k$. Aceste cuvinte conţin $n-k$ subcuvinte de lungime $k+1$ toate distincte. Aceste cuvinte corespund unor drumuri de lungime $n-k-1$ în graful de Bruijn corespunzător. Şi fiecărui drum de lungime $n-k-1$ în graful de Bruijn corespunzător îi putem ataşa un cuvânt de lungime n, care conţine toate cele $n-k$ subcuvinte distincte de lungime $k+1$. □

Numărul $M(n)$ se poate calcula şi cu ajutorul arborilor de Bruijn ([2]). Un *arbore de Bruijn* $T(q, w)$ este un arbore q-ar cu rădăcina w şi se defineşte recursiv în felul următor.

a) Cuvântul $w \in A^k$ unde $A = \{a_1, a_2, \ldots, a_q\}$ este rădăcina arborelui $T(q, w)$.

b) Dacă la un moment dat în construcţia recursivă a arborelui, $x_1 x_2 \ldots x_k$ este o frunză,[1] acele cuvinte dintre $x_1 x_2 \ldots x_k a_1$, $x_1 x_2 \ldots x_k a_2$, $\ldots$, $x_1 x_2 \ldots x_k a_q$ care nu apar pe drumul de la rădăcină la nodul $x_1 x_2 \ldots x_k$, vor fi descendenţii nodului $x_1 x_2 \ldots x_k$.

[1] nod fără descendenţi

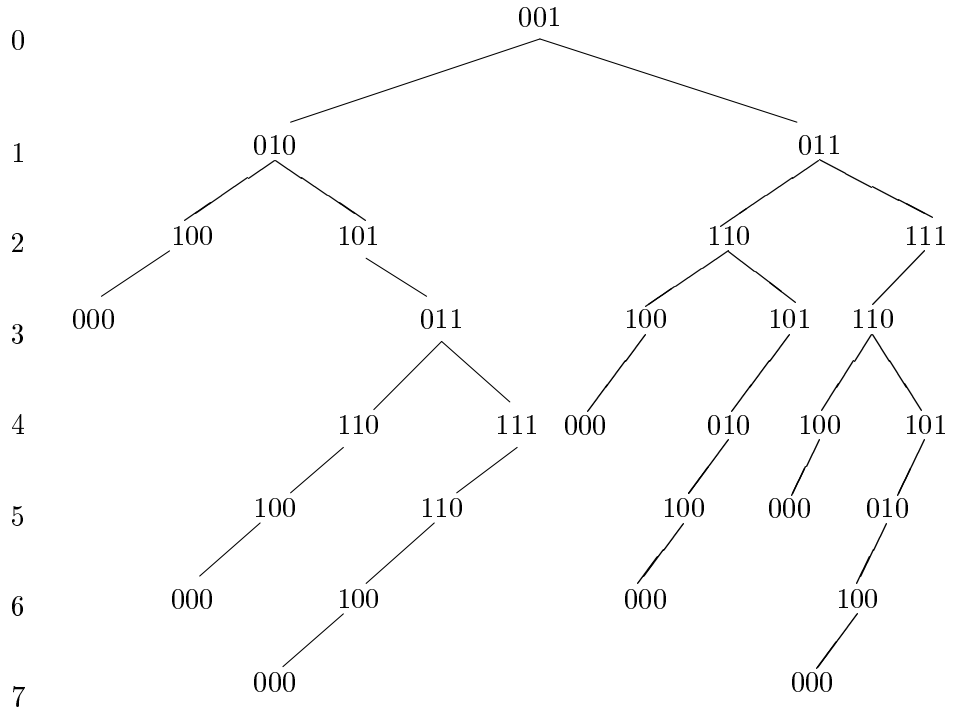


Figura 9: Arborele de Bruijn $T(2, 000)$.

c) Regula *b*) se aplică cât timp este posibil.

Pe baza definiţiei de mai sus şi a teoremei 20 se poate enunţa şi teorema următoarea.

Teorema 21. *$M(n)$ este egal cu numărul nodurilor (vârfurilor) de pe nivelul $n-k-1$ în mulţimea arborilor $\{T(q,w) \mid w \in A^{k+1}\}$. (Rădăcina este de nivel 0, descendenţii ei de nivel 1, etc.)*

Figurile 9–12 conţin arborii $T(2, 000)$ $T(2, 001)$ $T(2, 010)$ $T(2, 100)$. Să calculăm, de exemplu, valoarea lui $M(6)$. În acest caz $k = 2$, în figurile 9–12 sunt cei patru arbori $T(2, 000)$ $T(2, 001)$ $T(2, 010)$ $T(2, 100)$, ceilalţi patru sunt oglindirile acestora. La nivelul 3 în arborile din figurile 9–12 sunt în total 18 noduri, deci $M(6) = 2 \cdot 18 = 36$. Alte exemple: $M(7) = 2 \cdot 21 = 42$, $M(10) = 2 \cdot 8 = 16$.

Plecând de la arborii în care toate nodurile au exact doi descendenţi, se poate găsi o margine superioară pentru calculul lui $M(n)$.

$$M(n) \leq 2^{k+1} \cdot 2^{n-k-1} = 2^n$$

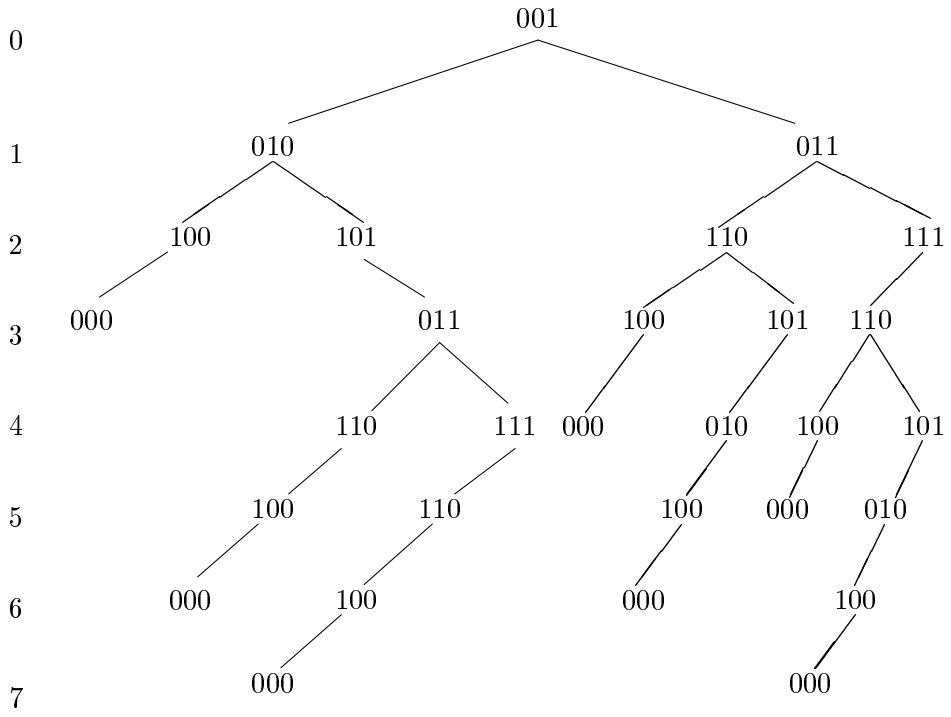


Figura 10: Arborele de Bruijn $T(2, 001)$.

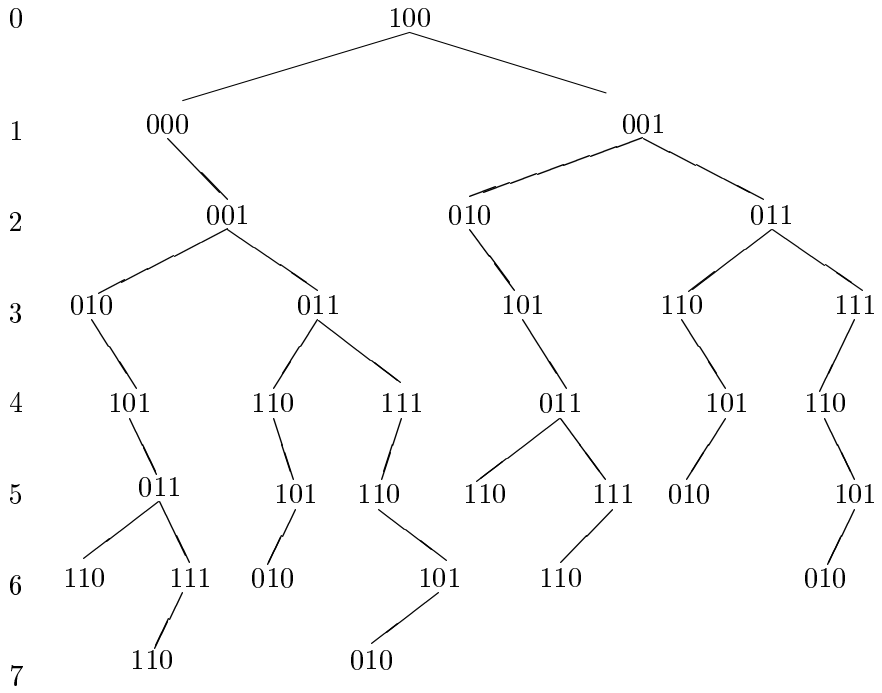


Figura 11: Arborele de Bruijn $T(2, 100)$.

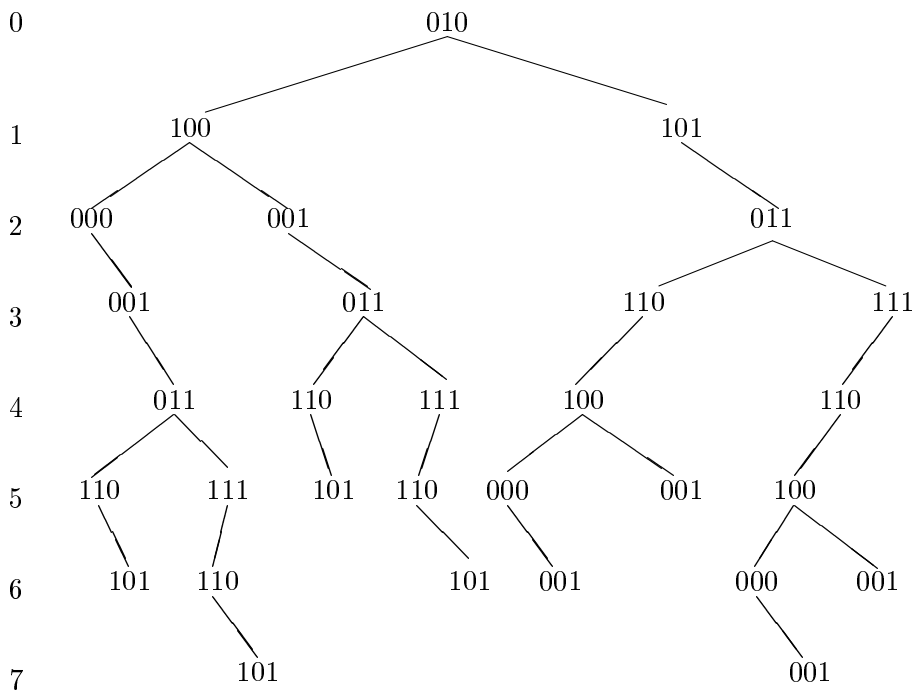


Figura 12: Arborele de Bruijn $T(2, 010)$.

Pentru unele valori ale lui n se poate găsi uşor o formulă exactă.

Teorema 22. *Dacă* $n = 2^k + k - 1$ *atunci* $M(n) = 2^{2^{k-1}}$.

Demonstraţie. Numărul ciclurilor hamiltoniene în graful de Bruijn $B(2, k)$ este egal cu $2^{2^{k-1}-k}$ [11]. Cu fiecare vârf al unui ciclu hamiltonian începe un cuvânt de Bruijn, care conţine toţi factorii de lungime k, şi care are complexitate factorilă maximă, deci $M(n) = 2^k \cdot 2^{2^{k-1}-k} = 2^{2^{k-1}}$. □

Teorema de mai sus se poate genereliza uşor pentru un alfabet cu $q \geq 2$ litere, şi se obţine:

Teorema 23. *Dacă* $n = q^k + k - 1$ *atunci* $M(n) = (q!)^{q^{k-1}}$.

Calculul valorii lui $M(n)$ pentru un n oarecare este o problemă deschisă.

8.4.4 Complexitatea totală

Complexitatea totală se defineşte pentru cuvinte finite şi numără toţi factorii distincţi ai unui cuvânt.

$$K(u) = \sum_{i=1}^{|u|} f_u(i)$$

Un cuvânt $\underbrace{a \ldots a}_{k}$ pentru $k > 1$ se numeşte *cuvânt trivial* (foloseşte numai o singură literă). Un astfel de cuvânt de lungime k are complexitatea totală egală cu k (conţine câte un singur factor de fiecare lungime între 1 şi k).

Apar următoarele două probleme:

1. Să se găsească cel mai scurt cuvânt cu o complexitate totală dată.

Această problemă totdeauna are soluţie. (Dacă complexitatea totală este C, atunci în cel mai rău caz cuvântul trivial de lungime C este cel cerut).

2. Să se găsească un cuvânt de lungime k cu o complexitate totală dată, dacă există.

Pentru rezolvarea acestei probleme se poate folosi o metodă de tip *branch-and-bound*, dar problema nu are soluţie totdeauna. Următorul algoritm recursiv găseşte toate cuvintele de lungime k cu complexitatea totală C dacă există. Folosim alfabetul $a_1, a_2, \ldots, a_k$, iar semnul + reprezintă concatenarea a două cuvinte. La primul apel w este cuvântul vid.

procedura generează (w):
  dacă $K(w) < C$ şi $|w| < k$
    atunci pentru $i = 2, 3, \ldots, k$ **execută** generează $(w + a_i)$
    altfel dacă $K(w) = C$ şi $|w| = k$ **atunci** scrie (w)
sfârşit procedură

Prima problemă se poate rezolva renunţând la criteriul lungime în algoritmul de mai sus. Dacă nu impunem restricţie la lungime totdeaunea există un cuvânt trivial (cu toate litere egale intre ele) de lungime C care are complexitatea totală C.

Se pune problema dacă există un cuvânt netrivial cu o complexitate totală dată. Răspunsul este afirmativ ([39]).

Teorema 24. *Dacă C este un număr natural diferit de 2 şi 4, atunci totdeauna există un cuvânt netrivial care are complexitatea C.*

Demonstraţie. Să considerăm următoarele cuvinte de lungime k şi complexitatea lor totală, care se poate obţine prin calcul direct.

$$K(a^{k-1}b) = 2k-1 \qquad \text{pentru } k \geq 1,$$
$$K(ab^{k-3}aa) = 4k-8 \qquad \text{pentru } k \geq 4,$$
$$K(abcd^{k-3}) = 4k-6 \qquad \text{pentru } k \geq 3.$$

1. Dacă C este un număr impar, se poate scrie $C = 2k-1$ pentru un k. De aici avem $k = \frac{C+1}{2}$ şi cuvântul $a^{k-1}b$ are complexitatea C.

2. Dacă C este un număr par, atunci poate fi scris ca $C = 2\ell$. Avem următoarele două cazuri:

2.1. Dacă $\ell = 2h$, atunci din $4k-8 = C$ rezultă $4k-8 = 4h$, adică $k = h+2$. În acest caz cuvântul $ab^{k-3}aa$ are complexitatea totală egală cu C (pentru $k \geq 4$).

2.2. Dacă $l = 2h+1$ atunci $4k-6 = C$ ne dă $4k-6 = 4h+2$. De aici rezultă $k = h+2$ şi cuvântul $abcd^{k-3}$ are complexitatea totală egală cu C (pentru $k \geq 3$). □

Mai mult, este adevărată şi următoarea teoremă.

Teorema 25. *Dacă C este un număr natural diferit de 1, 2, 4, 6, 10, 18 şi 22, atunci există un cuvânt format numai din două simboluri, cu complexitatea dată C.*

Demonstraţie. În demonstraţia teoremei de mai sus am folosit mai mult de două litere numai în cazul 2.2. Deci trebuie să demonstrăm afirmaţia numai când C este de forma $4h+2$. Dacă $C = 4h+2$ şi $C \geq 34$, atunci avem:

$$K(ab^{k-7}abbabb) = 8k-46 \qquad \text{pentru } k \geq 10,$$
$$K(ab^{k-7}ababba) = 8k-42 \qquad \text{pentru } k \geq 10,$$

Dacă $h = 2s$, atunci $8k-46 = 4h+2$, adică $k = s+6$ şi cuvântul $ab^{k-7}abbabb$ are complexitatea totală $4h+2$.

Dacă $h = 2s + 1$, atunci $8k - 42 = 4h + 2$, adică $k = s + 6$ şi cuvântul $ab^{k-7}ababba$ are complexitatea totală $4h + 2$.

Pentru $C < 34$ numai pentru numerele 14, 26 şi 30 de forma cerută există soluţie: $K(ab^4a) = 14$, $K(ab^6a) = 26$, $K(ab^5aba) = 30$. □

În legătură cu problema a doua apare a problemă nouă: câte cuvinte de lungime k şi complexitate totală C există? Pentru k mic, această problemă poate fi studiată exhaustiv. Fie A un alfabet din k litere, şi să considerăm toată cuvintele de lungime k peste A.

Fie $|A| = k$ şi să notăm cu $f_k(C)$ frecvenţa cuvintelor de lungime k peste A care au complexitatea totală C. În tabelul de mai jos sunt date câteva frecvenţe (k lungimea cuvintelor, C complexitatea totală, f frecvenţa).

k = 2

C:	2	3
$f_k(C)$:	2	2

k = 3

C:	3	4	5	6
$f_k(C)$:	3	0	18	6

k = 4

C:	4	5	6	7	8	9	10
$f_k(C)$:	4	0	0	36	48	144	24

k = 5

C:	5	6	7	8	9	10	11	12	13	14	15
$f_k(C)$:	5	0	0	0	60	0	200	400	1140	1200	120

Următoarea teoremă este adevărată şi se poate demonstra uşor.

Teorema 26.

$$f_k(C) = 0 \qquad dacă\ C < k\ sau\ C > \frac{k(k+1)}{2},$$

$$f_k(k) = k,$$

$$f_k(2k-1) = 3k(k-1),$$

$$f_k\left(\frac{k(k+1)}{2} - 1\right) = \frac{k(k-1)k!}{2},$$

$$f_k\left(\frac{k(k+1)}{2}\right) = k!$$

$$Dacă\ C = k+1, k+2, \cdots, 2k-2, \quad atunci\ \ f_k(C) = 0.$$

$$Dacă\ C\ =\ 2k, 2k+1, \cdots, 3k-5, \quad atunci\ \ f_k(C) = 0.$$

Cuvintele de lungime k pot avea complexitatea totală între k şi $\frac{k(k+1)}{2}$. Fie b_k cel mai mic număr natural pentru care

$$f_k(C) \neq 0 \quad \text{pentru orice } C \text{ astfel încât} \quad b_k \leq C \leq \frac{k(k+1)}{2}.$$

Numărul b_k există pentru orice k (în cel mai rău caz este egal cu $\frac{k(k+1)}{2}$). Se poate vedea uşor că $f_k(b_k) \neq 0$ pentru $k \geq 5$, pentru că $K(ab^{k-\ell}ab^{\ell-2}) = = b_k$.

De examplu: $b_3 = 5$, $b_4 = 7$, $b_5 = 11$ şi $b_6 = 14$.

Dăm fără demonstraţie următoarea teoremă (dată sub forma de conjectură în [39] şi demonstrată în [46]).

Teorema 27. *Dacă* $k = \frac{\ell(\ell+1)}{2} + 2 + i$, *unde* $\ell \geq 2$ *şi* $0 \leq i \leq \ell$ *atunci*

$$b_k = \frac{\ell(\ell^2-1)}{2} + 3\ell + 2 + i(\ell+1).$$

8.4.5 *d*-complexitatea totală

Noţiunea de factor (subcuvânt) se generalizează după cum urmează.

Fie d, k *şi* s *numere întregi pozitive,* $p = x_1x_2\cdots x_k \in X^k$. *Un* d*-factor al lui* p *se defineşte ca* $q = x_{i_1}x_{i_2}\cdots x_{i_s}$ *unde*

$$\begin{aligned} &i_1 \geq 1, \\ &1 \leq i_{j+1} - i_j \leq d, \qquad \text{pentru} \quad j = 1, 2, \cdots, s-1, \\ &i_s \leq k. \end{aligned}$$

Aceasta se notează prin $q \subseteq_d p$. *Dacă* $q \subseteq_d p$ *şi* $q \neq p$ *atunci* q *este un* d*-factor propriu al lui* p *şi se notează prin* $q \subset_d p$.

d-complexitatea totală $K_d(p)$ al cuvântului p este numărul tuturor d-factorilor ai lui p.

Să remarcăm că avem $K_1(p) = K(p)$.

Exemplu. Fie $X = \{a, b\}$ şi $p = abab$. În acest cuvânt există doi 2-factori de lungime 1 (a, b), patru 2-factori de lungime 2 (ab, aa, ba, bb), patru 2-factori de lungime 3 (aba, abb, aab, bab), şi un singur 2-factor de lungime 4 $(abab)$. Deci $K_2(p) = 2 + 4 + 4 + 1 = 11$.

În cazul cuvintelor de lungime k, care constau din simboluri diferite, d-complexitatea se va nota cu $N(k,d)$.

Dacă $|X| \geq 2,\ \ k \geq 1,\ \ d \geq 1$ şi $p \in X^k$ atunci

$$k \leq K_d(p) \leq 2^k - 1.$$

Dacă p este un cuvânt, consistând din simboluri diferite şi d este un întreg pozitiv, atunci vom nota cu $a_{i,d}(p)$ numărul d-subcuvintelor ale lui p care se termină în poziţia i. Dacă $k \geq 1$ şi $p \in X^k$ este format din simboluri distincte atunci

$$a_{i,d}(p) = 1 + a_{i-1,d}(p) + a_{i-2,d}(p) + \ldots + a_{i-d,d}(p), \qquad i = 1, 2, \ldots, k \quad (60)$$

d-complexitatea a unui cuvânt de lungime k cu simboluri diferite se poate obţine cu formula

$$N(k,d) = \sum_{i=1}^{k} a_{i,d}(p)$$

unde p este un cuvânt oarecare de k simboluri diferite. Din cauza formulei (60) se poate scrie (când $d \geq 2$):

$$a_{i,d} + \frac{1}{d-1} = \left(a_{i-1,d} + \frac{1}{d-1}\right) + \cdots + \left(a_{i-d,d} + \frac{1}{d-1}\right).$$

Fie

$$b_{i,d} = a_{i,d} + \frac{1}{d-1}, \qquad \text{şi} \qquad c_{i,d} = (d-1)b_{i,d}$$

atunci

$$c_{i,d} = c_{i-1,d} + c_{i-2,d} + \ldots + c_{i-d,d}$$

şi secvenţa $c_{i,d}$ este de tip Fibonacci. Pentru orice d avem $a_{1,d} = 1$ şi de aici rezultă $c_{1,d} = d$. Numerele $c_{i,d}$ pot fi definite cu ajutorul următoarei formule de recurenţe:

$$\begin{aligned} &c_{n,d} = c_{n-1,d} + c_{n-2,d} + \ldots + c_{n-d,d} && \text{pentru} \quad n > 0, \\ &c_{n,d} = 1 && \text{pentru} \quad n \leq 0. \end{aligned}$$

Aceste numere pot fi obţinute cu ajutorul următoarei funcţii generatoare.

Teorema 28.

$$F_d(z) = \sum_{n\geq 0} c_{n,d} z^n = \frac{1 + (d-3)z - (d-1)z^2 + z^{d+1}}{(1-z)(1-2z+z^{d+1})}.$$

Demonstraţie. Funcţia generatoare a numerelor $c_{n,d}$ este

$$F_d(z) = \sum_{n\geq 0} c_{n,d} z^n$$

Din (60) avem

$$F_d(z) = c_{0,d} + c_{1,d}z + \cdots + c_{d-1,d}z^{d-1} + z\sum_{n\geq d} c_{n-1,d}z^{n-1} +$$

$$+z^2 \sum_{n\geq d} c_{n-2,d}z^{n-2} + \cdots + z^n \sum_{n\geq d} c_{n-d,d}z^{n-d} =$$

$$= \sum_{n=0}^{d-1} c_{n,d}z^n + z\left(F_d(z) - \sum_{n=0}^{d-2} c_{n,d}z^n\right) + z^2\left(F_d(z) - \sum_{n=0}^{d-3} c_{n,d}z^n\right) +$$

$$+\cdots + z^{d-1}(F_d(z) - c_{0,d}) + z^d F_d(z)$$

Atunci

$$F_d(z)(1 - z - z^2 - \cdots - z^d) = c_{0,d} + z(c_{1,d} - c_{0,d}) + z^2(c_{2,d} - c_{1,d} - c_{0,d}) +$$

$$+\cdots + z^{d-1}(c_{d-1,d} - c_{d-2,d} - \cdots - c_{0,d}).$$

Dar $c_{0,d} = 1$, $c_{1,d} = d$, $c_{2,d} = c_{1,d} + c_{0,d} + c_{-1,d} + \ldots + c_{2-d,d} = 2d - 1$ etc., deci obţinem

$$F_d(z) = \frac{1 + (d-1)z + (d-2)z^2 + \cdots 2z^{d-2} + z^{d-1}}{1 - z - z^2 - \cdots - z^d}$$

Deoarece

$$(1 - z - z^2 - \cdots z^d)(1 - z) = 1 - 2z + z^{d+1},$$

se obţine:

$$F_d(z) = \frac{1 + (d-2)z - z^2 - \cdots - z^d}{1 - 2z + z^{d+1}} = \frac{1 + (d-3)z - (d-1)z^2 + z^{d+1}}{(1-z)(1-2z+z^{d+1})},$$

ceea ce demonstrează teorema. □

d-complexitatea $N(k,d)$ poate fi exprimată cu ajutorul acestor numere $c_{n,d}$ cu ajutorul formulei:

$$N(k,d) = \frac{1}{d-1}\left(\sum_{i=1}^{k} c_{i,d} - k\right), \qquad \text{pentru } d > 1$$

şi

$$N(k,1) = \frac{k(k+1)}{2}$$

sau

$$N(k,d) = N(k-1,d) + \frac{1}{d-1}(c_{k,d} - 1), \qquad \text{for } d > 1, \quad k > 1.$$

Dacă $d = 2$ atunci

$$F_2(z) = \frac{1-z^2}{1-2z+z^3} = \frac{1+z}{1-z-z^2} = \frac{F(z)}{z} + F(z)$$

unde $F(z)$ este funcţia generatoare a numerelor Fibonacci F_n (cu $F_0 = 0$, $F_1 = 1$). Atunci. din această formulă obţinem

$$c_{n,2} = F_{n+1} + F_n = F_{n+2}$$

şi

$$N(k,2) = \sum_{i=1}^{k} F_{i+2} - k = F_{k+4} - k - 3$$

Tabelul 1 listează valorile lui $N(k,d)$ pentru $k \leq 10$ şi $d \leq 10$.

$k \backslash d$	1	2	3	4	5	6	7	8	9	10
1	1	1	1	1	1	1	1	1	1	1
2	3	3	3	3	3	3	3	3	3	3
3	6	7	7	7	7	7	7	7	7	7
4	10	14	15	15	15	15	15	15	15	15
5	15	26	30	31	31	31	31	31	31	31
6	21	46	58	62	63	63	63	63	63	63
7	28	79	110	122	126	127	127	127	127	127
8	36	133	206	238	250	254	255	255	255	255
9	45	221	383	464	494	506	510	511	511	511
10	55	364	709	894	974	1006	1018	1022	1023	1023

Tabelul 1

Din definiţia d-factorului rezultă că

$$N(k,d) = N(k,d+1), \qquad \text{for} \quad d \geq k-1$$

dar

$$N(k,k-1) = 2^k - 1$$

şi atunci

$$N(k,d) = 2^k - 1, \qquad \text{pentru orice} \quad d \geq k-1.$$

Următoarea teoremă ne dă valoarea lui $N(k,d)$ pentru o gamă largă de valori [38].

Teorema 29. *Pentru $k \geq 2d-2$ avem*

$$N(k,k-d) = 2^k - (d-2)\cdot 2^{d-1} - 2.$$

Demonstraţie. Fie $k \geq 2d-2$. Atunci $N(k,k-d-1)$ poate fi calculat în felul următor. Printre cei $N(k,k-d)$ factori sunt exact $d\cdot 2^{d-1}$ pentru care $i_{j+1} - i_j = k-d$ pentru un j, deooarece sunt d posibilităţi de a alege poziţii cu distanţă $k-d$, şi 2^{d-1} posibilităţi de a alege celelalte litere. (Aceste distanţe între litere trebuie să fie mai mici decât $k-d$, deci $k-d+1 \geq d-1$ adică $k \geq 2d-2$). Deci

$$N(k,k-d-1) = N(k,k-d) - d\cdot 2^{d-1}.$$

Pentru $d = 1,2,\ldots,$ avem:

$$N(k,k-2) = N(k,k-1) - 1$$

$$N(k,k-3) = N(k,k-2) - 2\cdot 2^1$$

$$N(k,k-4) = N(k,k-3) - 3\cdot 2^2$$

$$\ldots$$

$$N(k,k-d) = N(k,k-d-+1) - (d-1)\cdot 2^{d-2}$$

la care se mai adaugă formula evidentă

$$N(k,k-1) = 2^k - 1$$

Adunând acestea membru cu membru, obţinem:

$$N(k, k-d) = 2^k - 1 - \left(1 + 2 \cdot 2^1 + 3 \cdot 2^2 + \ldots + (d-1) \cdot 2^{d-2}\right)$$

De unde, printr-un calcul simplu, se obţine:

$$N(k, k-d) = 2^k - (d-2)2^{d-1} - 2,$$

ceea ce trebuia demonstrat. □

Valoarea $N(K, d)$ se poate calcula şi numărând secvenţele de 0-uri şi 1-uri de lungime k care nu au mai mult $d-1$ zerouri adiacente. Într-o astfel de secvenţă 1 reprezintă prezenţa, iar 0 absenţa simbolului respectiv într-un d-factor. Fie $b_{k,d}$ numărul secvenţelor de zero şi unu, de lungime k în care prima şi ultima poziţie este 1 şi numărul zerourilor adiacente este cel mult $d-1$. Uşor se poate demonstra că

$$b_{k,d} = b_{k-1,d} + b_{k-2,d} + \ldots + b_{k-d,d}, \qquad \text{pentru} \quad k > 1,$$

$$b_{1,d} = 1,$$

$$b_{k,d} = 0, \qquad \text{pentru } k \leq 0,$$

pentru că secvenţele de lungime $k-i$ $(i = 1, 2, ..., d)$ pot fi continuate numai într-un singur mod pentru a obţine o secvenţă similară de lungime k (adăugând la dreapta secvenţe de forma $0^{i-1}1$). Pentru $b_{k,d}$ tse poate obţine şi formula:

$$b_{k,d} = 2b_{k-1,d} - b_{k-1-d,d}.$$

Funcţia generatoare pentru $b_{n,d}$ este următoarea ([40]):

Teorema 30.

$$B_d(z) = \sum_{n \geq 0} b_{n,d} z^n = \frac{z}{1 - z - \cdots - z^d} = \frac{z(1-z)}{1 - 2z + z^{d+1}}.$$

Observaţie. Pentru $b_{n,2}$ se obţin numerele Fibonacci cunoscute.

Adăugând zerouri la stânga sau/şi la dreapta acestor secvenţe putem obţine numerele $N(k, d)$ ca numărul acestor secvenţe. Astfel

$$N(k, d) = b_{k,d} + 2b_{k-1,d} + 3b_{k-2,d} + \cdots + kb_{1,d}.$$

(pot fi adăugate i zerouri în $i+1$ moduri: 0 la stânga şi i la dreapta, 1 la stânga şi $i-1$ la dreapta, şi aşa mai departe).

Din formula de mai sus se poate obţine funcţia generatoare $N_d(z)$ care corespunde complexităţii $N(k,d)$ ca produs a celor două funcţii generatoare $B_d(z)$ şi

$$A(z) = \sum_{n\geq 0}(n+1)z^n = \frac{1}{(1-z)^2},$$

din care rezultă următoarea teoremă.

Teorema 31.

$$N_d(z) = \sum_{n\geq 0} N(n,d)z^n = \frac{z}{(1-z)(1-2z+z^{d+1})}$$

În cazul $d=1$ avem

$$N(k,1) = \frac{k(k+1)}{2},$$

şi

$$N_1(z) = \sum_{n\geq 0}\frac{n(n+1)}{2}z^n = \frac{z}{(1-z)^3}.$$

Pentru $d=2$ avem

$$N(k,2) = F_{k+4} - k - 3.$$

Funcţia generatoare corespunzătoare este:

$$N_2(z) = \sum_{n\geq 0}(F_{n+4} - n - 3)z^n = \frac{z}{(1-z)^2(1-z+z^2)}.$$

8.5 Limbaje şi automate

Până acum am studiat cuvintele mai mult din punctul de vedere al structurii, al complexităţii lor. Mulţimile de cuvinte pot fi însă studiate şi în totalitatea lor. Există anumite reguli cu ajutorul cărora toate cuvintele din mulţimea respectivă pot fi generate? Există reguli prin care cuvintele din mulţime pot fi acceptate?

O mulţime finită sau infinită de cuvinte se numeşte *limbaj formal* sau simplu *limbaj*. Dacă dorim specificarea alfabetului spunem că limbajul este peste alfabetul respectiv.

Răspunsul la cele două întrebări puse mai sus este afirmativ: sunt limbaje care pot fi generate de *gramatici* şi sunt limbaje care por fi acceptate de *automate*.

Automatele sunt de două tipuri: *automate acceptoare* şi *automate translatoare*. Automatele acceptoare sunt cele care acceptă anumite limbaje definite peste un alfabet dat. Automatele acceptoare sunt denumite deobicei pe scurt *automate*. Automatele translatoare transformă anumite limbaje definite peste un alfabet de intrare în anumite limbaje peste un albafet de ieşire, cele două alfabete putând fi identice. Automatele translatoare sunt denumite deobicei pe scurt *translatoare*.

Gramatici

Concatenarea a două mulţimi de cuvinte (limbaje) este operaţia prin care obţinem o mulţime formată din toate cuvintele concatenate, primul cuvânt fiind luat din prima, iar cel de al doilea din a doua mulţime. Adică, dacă A şi B sunt două mulţimi de cuvinte, atunci

$$AB = \{uv \mid u \in A, v \in B\}.$$

O gramatică este un ansamblu de patru elemente $G = (N, T, P, S)$ unde

• N şi T sunt două alfabete (mulţimi finite şi nevide) disjuncte, elementele alfabetului N se numesc *neterminale* sau *variabile*, pe când elementele lui T sunt numite *terminale*.

• $P \subseteq (N \cup T)^* N (N \cup T)^* \times (N \cup T)^*$, adică elementele lui P sunt perechi (u, v) unde $u \in (N \cup T)^* N (N \cup T)^*$ (adică u este un cuvânt format cu elementele din $N \cup T$, dar conţine cel puţin o variabilă), v este un cuvânt peste alfabetul $N \cup T$, fără nici o restricţie, deci poate fi şi chiar cuvântul vid λ. Perechea (u, v) se numeşte *producţie* sau *regulă de rescriere* şi uneori se mai notează şi prin $u \to v$. Această regulă are semnificaţia că primul element u într-un cuvânt se înlocuieşte cu v, astfel putându-se genera cuvinte noi.

• Simbolul S se numeşte *simbol iniţial* (simbol de start). Generarea cuvintelor începe cu acest simbol, şi se continue prin înlocuiri cu ajutorul

producţiilor. Interesul este de a obţine cuvinte care conţin numai simboluri terminale.

Vom defini relaţia $\underset{G}{\Longrightarrow}$ (numită *derivare directă*), în felul următor $x \underset{G}{\Longrightarrow} y$ dacă $x = \alpha u \beta$, $y = \alpha v \beta$ şi $(u, v) \in P$, unde $\alpha, \beta \in (N \cup T)^*$. Închiderea reflexiv tranzitivă a acestei relaţii se notează cu $\overset{*}{\underset{G}{\Longrightarrow}}$ şi se numeşte *derivare*.

Avem $x \overset{*}{\underset{G}{\Longrightarrow}} z$ dacă există $\alpha_1, \alpha_2, \ldots, \alpha_n \in (N \cup T)^*$ şi $x \underset{G}{\Longrightarrow} \alpha_1 \underset{G}{\Longrightarrow} \alpha_2 \underset{G}{\Longrightarrow} \ldots \underset{G}{\Longrightarrow} \alpha_n \underset{G}{\Longrightarrow} z$ sau $x = z$.

Dacă nu există nici o confuzie, litera G poate fi omisă peste tot. Dacă se face cel puţin o înlocuire atunci vom folosi notaţia $\overset{+}{\Longrightarrow}$ în loc de $\overset{*}{\Longrightarrow}$ (Notaţia $\overset{+}{\Longrightarrow}$ reprezintă închiderea tranzitivă a relaţiei $\Longrightarrow$).

Se numeşte limbaj generat de gramatica $G = (N, T, P, S)$ mulţimea

$$L(G) = \left\{ w \in T^* \mid S \overset{*}{\underset{G}{\Longrightarrow}} w \right\}.$$

Exemplu. Fie $G = (N, T, P, S)$, unde
$N = \{S\}$,
$T = \{a, b\}$,
$P = \{S \to ab,\ S \to aSb\}$.
Se poate vedea uşor că $L(G) = \{a^n b^n \mid n \geq 1\}$, deoarece

$$S \underset{G}{\Longrightarrow} aSb \underset{G}{\Longrightarrow} a^2 S b^2 \underset{G}{\Longrightarrow} \ldots \underset{G}{\Longrightarrow} a^{n-1} S b^{n-1} \underset{G}{\Longrightarrow} a^n b^n,$$

unde am folosit până la penultima derivare directă regula $S \to aSb$, pe când la ultima, regula $S \to ab$. Putem scrie $S \overset{*}{\underset{G}{\Longrightarrow}} a^n b^n$. Deci $a^n b^n$ pentru orice n poate fi derivat din S şi nici un alt cuvânt nu poate fi derivat din S.

Gramaticile G_1 şi G_2 sunt *echivalente* dacă $L(G_1) = L(G_2)$. Acest lucru se notează prin $G_1 \cong G_2$.

Exemple.

1. Se dau gramaticile
$G_1 = (\{S\}, \{a, b\}, \{S \to aSb,\ S \to \lambda\}, S)$ şi
$G_2 = (\{S\}, \{a, b\}, \{S \to aSb,\ S \to ab\}, S)$.
Atunci $L(G_1) \setminus \{\lambda\} = L(G_2)$. Deci $G_1 \ncong G_2$, adică cele două gramatici nu sunt echivalente.

2. Următoarele două gramatici sunt echivalente, deoarece ambele generează limbajul $\{a^n b^n c^n \mid n \geq 1\}$.

$G_1 = (N_1, T, P_1, S_1)$, unde

$N_1 = \{S_1, X, Y\},\ T = \{a, b, c\},$

$P_1 = \{S_1 \to abc,\ S_1 \to aXbc,\ Xb \to bX,\ Xc \to Ybcc,\ bY \to Yb,$

$aY \to aaX,\ aY \to aa\}.$

$G_2 = (N_2, T, P_2, S_2)$, unde

$N_2 = \{S_2, A, B, C\},$

$P_2 = \{S_2 \to aS_2BC,\ S_2 \to aBC,\ CB \to BC,\ aB \to ab,\ bB \to bb,$

$bC \to bc,\ cC \to cc\}.$

Prima dată vom demonstra prin inducţie matematică completă că pentru $n \geq 2$ avem $S_1 \underset{G_1}{\overset{*}{\Longrightarrow}} a^{n-1}Yb^nc^n$. Dacă $n = 2$, atunci

$$S_1 \underset{G_1}{\Longrightarrow} aXbc \underset{G_1}{\Longrightarrow} abXc \underset{G_1}{\Longrightarrow} abYbcc \underset{G_1}{\Longrightarrow} aYb^2c^2$$

Presupunem că $S_1 \underset{G_1}{\overset{*}{\Longrightarrow}} a^{n-2}Yb^{n-1}c^{n-1}$. Folosim regula $aY \to aaX$, iar pe urmă de $(n-1)$ ori regula $Xb \to bX$, apoi regula $Xc \to Ybcc$, după care din nou folosim de $(n-1)$ ori regula $bY \to Yb$. Deci

$$S_1 \underset{G_1}{\Longrightarrow} a^{n-2}Yb^{n-1}c^{n-1} \underset{G_1}{\Longrightarrow} a^{n-1}Xb^{n-1}c^{n-1} \underset{G_1}{\overset{*}{\Longrightarrow}} a^{n-1}b^{n-1}Xc^{n-1}$$

$$\underset{G_1}{\Longrightarrow} a^{n-1}b^{n-1}Ybc^n \underset{G_1}{\overset{*}{\Longrightarrow}} a^{n-1}Yb^nc^n$$

Dacă acum folosim regula $aY \to aa$, obţinem că $S_1 \underset{G_1}{\overset{*}{\Longrightarrow}} a^nb^nc^n$ pentru $n \geq 2$, dar $S_1 \underset{G_1}{\Longrightarrow} abc$ pe baza regulii $S_1 \to abc$, deci $a^nb^nc^n \in L(G_1)$ pentru orice $n \geq 1$. Mai trebuie să demonstrăm că alte cuvinte decât cele de forma $a^nb^nc^n$ nu pot fi derivate. Este uşor să ne convingem de acest lucru, deoarece derivare cu succes (adică cea care se termină cu un cuvânt format numai din simboluri terminale) nu poate exista în afara celei de mai sus.

Asemănător pentru $n \geq 2$ avem

$$S_2 \underset{G_2}{\Longrightarrow} aS_2BC \underset{G_2}{\overset{*}{\Longrightarrow}} a^{n-1}S_2(BC)^{n-1} \underset{G_2}{\Longrightarrow} a^n(BC)^n \underset{G_2}{\overset{*}{\Longrightarrow}} a^nB^nC^n$$

$$\underset{G_2}{\Longrightarrow} a^nbB^{n-1}C^n \underset{G_2}{\overset{*}{\Longrightarrow}} a^nb^nC^n \underset{G_2}{\Longrightarrow} a^nb^ncC^{n-1} \underset{G_2}{\overset{*}{\Longrightarrow}} a^nb^nc^n$$

Aici am folosit, în ordine, următoarele reguli: $S_2 \to aS_2BC$ (de $n-1$ ori), $S_2 \to aBC$, $CB \to BC$ (de $n-1$ ori), $aB \to ab$, $bB \to bb$ (de $n-1$ ori),

$bC \to bc$, $cC \to cc$ (de $n-1$ ori). Totosdată $S_2 \underset{G_2}{\Longrightarrow} aBC \underset{G_2}{\Longrightarrow} abC \underset{G_2}{\Longrightarrow} abc$, deci $S_2 \overset{*}{\underset{G_2}{\Longrightarrow}} a^n b^n c^n$, $n \geq 1$. Şi aici se poate vedea uşor că alte cuvinte nu pot fi derivate.

Clasificarea lui Chomsky

Dacă impunem anumite restricţii asupra producţiilor (a regurilor de rescriere) ajungem la următoarele clase de gramatici.

Gramatica $G = (N, T, P, S)$ este

- de tipul 0 dacă nu se pune nici o restricţie asupra producţiilor.
- de tipul 1 (*dependentă de context*), dacă fiecare producţie este de forma $uXv \to uwv$, unde $X \in N$, $u, v \in (N \cup T)^*$, $w \in (N \cup T)^+$. Se admite şi producţia $S \to \lambda$, dacă S nu apare în partea dreaptă a nici unei producţii.
- de tipul 2 (*independentă de context*), dacă fiecare producţie este de forma $X \to w$, unde $X \in N$, $w \in (N \cup T)^+$. Se admite şi producţia $S \to \lambda$, dacă S nu apare în partea dreaptă a nici unei producţii.
- de tipul 3 (*regulară*), dacă fiecare producţie este de forma $X \to aY$ sau $X \to a$, unde $a \in T$ şi $X, Y \in N$. Se admite şi producţia $S \to \lambda$, dacă S nu apare în partea dreaptă a nici unei producţii.

Dacă gramatica G este de tipul i, atunci limbajul $L(G)$ este deasemenea de tipul i (limbaj general, dependent de context, independent de context, regular).

Un limbaj L este de tipul i ($i = 0, 1, 2, 3$), dacă există o gramatică G de tipul i care îl generează, adică $L = L(G)$.

Să notăm cu $\mathcal{L}_i$ ($i = 0, 1, 2, 3$) familia limbajelor de tipul i. Se poate demonstra că $\mathcal{L}_0 \supset \mathcal{L}_1 \supset \mathcal{L}_2 \supset \mathcal{L}_3$.

Exemple. În exemplele următoare în loc de producţiile

$$X \to \alpha_1,\ X \to \alpha_2,\ \ldots,\ X \to \alpha_n$$

vom scrie pe scurt

$$X \to \alpha_1 \mid \alpha_2 \mid \ldots \mid \alpha_n$$

Gramatică dependentă de context

$G_1 = (N_1, T_1, P_1, S_1)$, unde $N_1 = \{S_1, A, B, C\}$, $\quad T_1 = \{a, 0, 1\}$

Elementele lui P_1 sunt:

$S_1 \to ACA$

$AC \to AACA \mid ABa \mid AaB$

$B \to AB \mid A$

$A \to 0 \mid 1$

Limbajul $L(G_1)$ constă din cuvintele de forma uav, unde $u, v \in \{0,1\}^*$ şi $|u| \neq |v|$.

Gramatică independentă de context

$G_2 = (N_2, T_2, P_2, E)$, unde $N_2 = \{E, T, F\}$, $\quad T_2 = \{+, *, (,), a\}$

Elementele lui P_2 sunt:

$E \to E + T \mid T$

$T \to T * F \mid F$

$F \to (E) \mid a$

Limbajul $L(G_2)$ constă din expresii algebrice, care pot fi formate din elementele alfabetului T_2.

Gramatică regulară

$G_3 = (N_3, T_3, P_3, S_3)$, unde $N_3 = \{S_3, A, B\}$, $\quad T_3 = \{a, b\}$

Elementele lui P_3 sunt:

$S_3 \to aA$

$A \to aB \mid a$

$B \to aB \mid bB \mid a \mid b$

Limbajul $L(G_3)$ conţine cuvinte care se pot forma cu literele a şi b şi care încep cu cel puţin doi de a.

Vom demonstra următorul rezultat interesant.

Teorema 32. *Există limbaje care nu se pot genera cu gramatici.*

Demonstraţie. Să considerăm mulţimea tuturor limbajelor peste un alfabet A notată cu $\mathcal{L}_A = \{L \mid L \subseteq A^*\}$. Mulţimea A^* este o mulţime infinită dar numărabilă. Cuvintele acestei mulţimi se pot înşira după lungimea lor, iar cele de aceeaşi lungime se pot pune în ordine alfabetică. Fie acest şir al cuvintelor din A^*: $s_0, s_1, s_2, \ldots$, unde evident $s_0 = \lambda$. Atunci fiecărui limbaj $L \subseteq A^*$ îi

putem ataşa un şir binar infinit $b_0, b_1, b_2, \ldots$ în felul următor:

$$b_i = \begin{cases} 1, & \text{dacă } s_i \in L \\ 0, & \text{dacă } s_i \notin L \end{cases} \qquad i = 0, 1, 2, \ldots$$

Se poate vedea uşor că mulţimea acestor şiruri binare este infinită şi nenumărabilă (adică de puterea continuului), deoarece fiecare şir poate fi considerat ca partea fracţionară (în binar) a unui număr real pozitiv subunitar. Este adevărat şi faptul că orice număr real pozitiv subunitar se poate considera ca un astfel de şir, dacă considerăm scrierea lui în binar şi luăm partea de după virgulă. Deoarece intervalul $(0, 1)$ este de puterea continuului, rezultă că şi mulţimea $\mathcal{L}_A$ este la fel.

Pentru demonstrarea afirmaţiei să codificăm gramaticile. Pentru o gramatică dată $G = (N, T, P, S)$ fie $N = \{S_1, S_2, \ldots S_n\}$, $T = \{a_1, a_2, \ldots a_m\}$ şi $S = S_1$. Codificare este următoarea:

$$\text{Codul lui } S_i \text{ este } 10\underbrace{11\ldots11}_{\text{de } i \text{ ori}}01, \quad \text{codul lui } a_i \text{ este } 100\underbrace{11\ldots11}_{\text{de } i \text{ ori}}001$$

Simbolurile sunt despărţite prin 000, codul săgeţii este 0000, iar producţiile le despărţim prin 00000.

Astfel, pentru a codifica o gramatică este suficient să le codificăm producţiile. Să luăm un exemplu:

$G = (\{S\}, \{a, b\}, \{S \to aSb,\ S \to ab\}, S)$.

Codul S este 10101, codul lui a este 1001001, iar al lui b 10011001. Atunci gramatica se codifică astfel:

$$\underbrace{10101}0000\underbrace{1001001}000\underbrace{10101}000\underbrace{10011001}00000\underbrace{10101}0000\underbrace{1001001}$$
$$000\underbrace{10011001}$$

Pe baza acestei codificări gramaticile pot fi ordonate lexicografic, adică odată după lungime, iar în cadrul cuvintelor (gramaticilor codificate) de aceeaşi lungime, în ordine alfabetică: $G_1, G_2, \ldots, G_k, \ldots$. Am văzut că mulţimea limbajelor formale este de puterea continuului, iar gramaticile formează o mulţime numărabilă, rezultă că există limbaje care nu se pot descrie cu ajutorul gramaticilor. □

Automate finite

Un *automat* este un ansamblu $\mathcal{A} = (A, Q, E, I, F)$, unde

- A este un *albafet*,
- Q este mulţimea *stărilor* automatului,
- E este mulţimea *arcelor etichetate*, $E \subseteq Q \times A \times Q$, care reprezintă tranziţiile,
- $I, F \subseteq Q$ sunt mulţimea *stărilor iniţiale* respectiv *finale*.

Un arc este notat prin (p, a, q) unde p este *extremitatea iniţială*, q *extremitatea finală*, iar a *eticheta* arcului. Un şir de arce consecutive

$$(q_0, a_1, q_1), (q_1, a_2, q_2), \ldots, (q_{n-2}, a_{n-1}, q_{n-1}), (q_{n-1}, a_n, q_n)$$

se numeşte un *drum* al automatului, iar cuvântul $a_1 a_2 \ldots a_n$ este *eticheta drumului*. Acest drum se notează prin $q_0 \xrightarrow{w} q_n$, unde $w = a_1 a_2 \ldots a_n$ este eticheta drumului. Drumul începe în q_0 şi se termină în q_n. Un drum este *productiv* dacă începe într-o stare iniţială şi se termină într-o stare finală. Cuvintele care sunt etichete ale unor drumuri productive sunt cuvinte acceptate de automatul respectiv. Cuvintele acceptate de un automat formează limbajul acceptat de automatul respectiv. Adică

$$L(\mathcal{A}) = \{w \in A^* \mid \exists p \in I, q \in F \text{ şi } p \xrightarrow{w} q\}$$

Dacă mulţimea Q a stărilor automatului este finită automatul se numeşte *automat finit*. În acest caz, deoarece alfabetul totdeauna este o mulţime finită, rezultă că şi mulţimea arcelor E este finită.

Două automate $\mathcal{A}_1$ şi $\mathcal{A}_2$ sunt echivalente dacă $L(\mathcal{A}_1) = L(\mathcal{A}_2)$, adică dacă limbajele acceptate de cele două automate sunt identice.

Automatele finite pot fi reprezentate şi cu ajutorul grafurilor. Stările vor fi vârfurile grafului, iar arcele (tranziţiile) automatului vor fi arcele grafului, care vor fi etichetate corespunzător. Stările iniţiale vor fi marcate cu săgeţi, iar stările finale cu cercuri concentrice. În loc de arce multiple între două stări vom folosi un singur arc etichetat cu etichetele tuturor arcelor între cele două stări.

stare iniţială

stare finală

În figura 13 sunt prezentate două automate finite. Dacă într-un automat o stare inițială este și finală, atunci automatul acceptă și cuvântul vid λ.

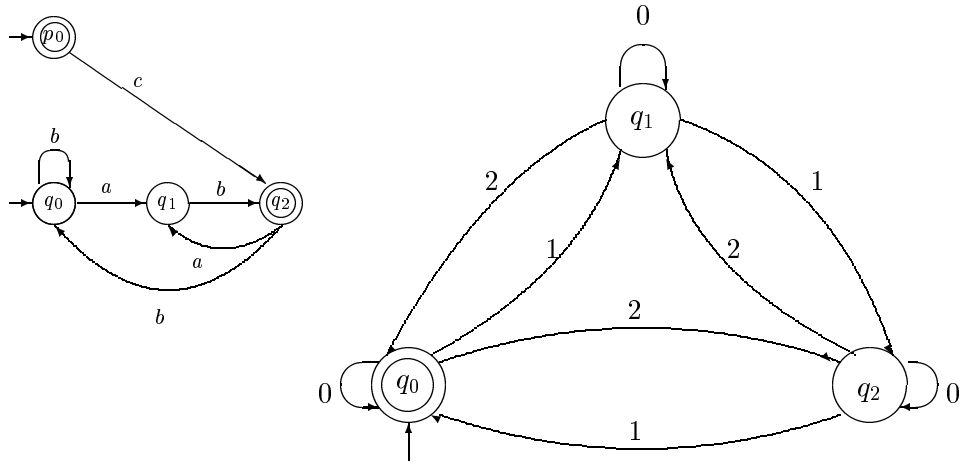


Figura 13: Automate finite

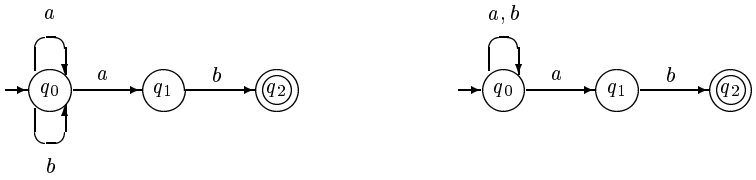
Pentru tratarea mai ușoară a tranzițiilor se pot defini următoarele mulțimi:

$$\forall p \in Q, a \in A \qquad \delta(p, a) = \{q \mid (p, a, q) \in E\}$$

Un automat se numește *determinist* dacă

$$\#I = 1 \text{ și } \forall q \in Q, a \in A \text{ avem } \#\delta(q, a) \leq 1$$

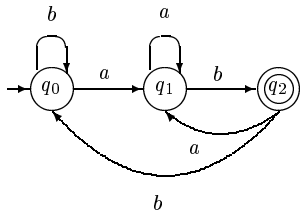
În caz contrar automatul este un *automat nedeterminist*.
Un exemplu de automat finit nedeterminist (cele două grafuri reprezintă același automat):



Mulțimile δ sunt date în tabelul următor, numit și *tabelul de tranziții*:

	a	b
q_0	$\{q_0, q_1\}$	$\{q_0\}$
q_1	$\emptyset$	$\{q_2\}$
q_2	$\emptyset$	$\emptyset$

Un exemplu de automat finit determinist:



O stare p este *accesibilă* dacă există un drum de la o stare inițială la starea p. O stare q este *productivă* dacă există un drum de la starea q la o stare finală. Stările inaccesibile şi neproductive se pot elimina din automat fără a afecta limbajul acceptat.

Algoritm pentru eliminarea stărilor inaccesibile

Definim următoarele mulţimi:

$U_0 = I$

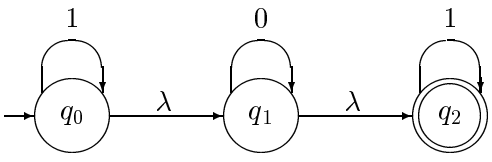
$$U_i = U_{i-1} \bigcup_{\substack{a \in \Sigma \\ q \in U_{i-1}}} \delta(q, a), \quad \text{pentru } i \geq 1.$$

Continuăm crearea mulţimilor U_i, până când pentru un k obţinem $U_k = U_{k-1}$. Mulţimea Q fiind finită, cu siguranţă vom ajunge la o astfel de situaţie. Fie notată această mulţime U_k cu U.

Elementele mulţimii $Q \setminus U$ sunt stările inaccesibile şi pot fi eliminate din automat.

Definim automatul finit cu λ-*mişcări*, care se deosebeşte de automatul finit obişnuit prin faptul că admitem ca arcele să fie etichetate şi cu λ. Adică mulţimea arcelor se defineşte ca $E \subseteq Q \times (A \cup \{\lambda\}) \times Q$.

Următorul automat cu λ-mişcări acceptă cuvintele de forma uvw, unde $u \in \{1\}^*$, $v \in \{0\}^*$ s $w \in \{1\}^*$.



Teorema 33. *Unui automat finit oarecare cu λ-mişcări i se poate ataşa un automat echivalent fără λ-mişcări.*

Fie automatul finit $\mathcal{A} = (Q, A, E, I, F)$ cu λ-mişcări. Definim automatul echivalent fără λ-mişcări ca fiind $\overline{\mathcal{A}} = (Q, A, \overline{E}, I, \overline{F})$. Pentru a defini elementele automatului trebuie să definim următoarele:

$\Lambda(q) \subseteq Q$ – mulţimea acelor stări în care se poate ajunge din starea q folosind numai λ-mişcări (inclusiv starea q).

$$\Lambda(S) = \bigcup_{q \in S} \Lambda(q), \quad \forall S \subseteq Q.$$

$$\overline{F} = \begin{cases} F \cup I, & \text{dacă } \Lambda(I) \cap F \neq \emptyset \\ F, & \text{altfel.} \end{cases}$$

Pentru a defini tranziţiile $\overline{\delta}$ să definim mulţimile:

$$\Delta(q, a) = \bigcup_{p \in \Lambda(q)} \delta(p, a)$$

$$\overline{\delta}(q, a) = \Delta(q, a) \cup \left(\bigcup_{p \in \Delta(q,a)} \Lambda(p) \right), \quad \forall q \in Q, \ \forall a \in \Sigma.$$

Arcele din $\overline{E}$ se definesc astfel: pentru fiecare $q \in \overline{\delta}(p, a)$ se defineşte arcul $(p, a, q) \in \overline{E}$. Se poate vedea uşor că automatul iniţial şi cel obţinut prin construcţia dată sunt echivalente.

În cazul exemplului de mai sus tabelul de tranziţii al automatului $\mathcal{A}$ este:

δ	0	1	λ
q_0	$\emptyset$	$\{q_0\}$	$\{q_1\}$
q_1	$\{q_1\}$	$\emptyset$	$\{q_2\}$
q_2	$\emptyset$	$\{q_2\}$	$\emptyset$

Atunci:

$\Lambda(q_0) = \{q_0, q_1, q_2\}$

$\Lambda(q_1) = \{q_1, q_2\}$

$\Lambda(q_2) = \{q_2\}$

$\Lambda(I) = \Lambda(q_0)$, şi intersecţia acesteia cu F nu este vidă, şi deci

$$\overline{F} = F \cup \{q_0\} = \{q_0, q_2\}.$$

$\Delta(q_0, 0) = \delta(q_0, 0) \cup \delta(q_1, 0) \cup \delta(q_2, 0) = \{q_1\}$

$$\{q_1\} \cup \Lambda(q_1) = \{q_1, q_2\} = \overline{\delta}(q_0, 0).$$

$\Delta(q_0, 1) = \delta(q_0, 1) \cup \delta(q_1, 1) \cup \delta(q_2, 1) = \{q_0, q_2\}$

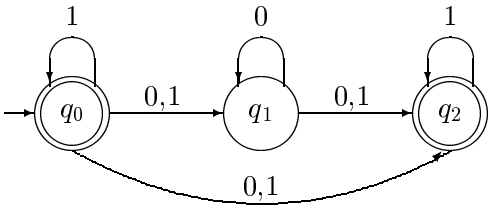
$$\{q_0, q_2\} \cup (\Lambda(q_0) \cup \Lambda(q_2)) = \{q_0, q_1, q_2\} = \overline{\delta}(q_0, 1)$$

$$\Delta(q_1,0) = \delta(q_1,0) \cup \delta(q_2,0) = \{q_1\}$$
$$\{q_1\} \cup \Lambda(q_1) = \{q_1,q_2\} = \overline{\delta}(q_1,0)$$
$$\Delta(q_1,1) = \delta(q_1,1) \cup \delta(q_2,1) = \{q_2\}$$
$$\{q_2\} \cup \Lambda(q_2) = \{q_2\} = \overline{\delta}(q_1,1)$$
$$\Delta(q_2,0) = \delta(q_2,0) = \emptyset = \overline{\delta}(q_2,0)$$
$$\Delta(q_2,1) = \delta(q_2,1) = \{q_2\}$$
$$\{q_2\} \cup \Lambda(q_2) = \{q_2\} = \overline{\delta}(q_2,1)$$

Deci automatul $\overline{\mathcal{A}}$ are următorul tabel de tranziţii:

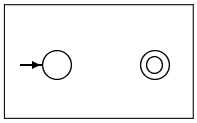
$\overline{\delta}$	0	1
q_0	$\{q_1,q_2\}$	$\{q_0,q_1,q_2\}$
q_1	$\{q_1,q_2\}$	$\{q_2\}$
q_2	$\emptyset$	$\{q_2\}$

iar automatul este:



Operaţii cu automate. Vom defini operaţiile de reuniune, produs, iteraţie pentru automate finite deterministe. Aceste operaţii pot fi definite foarte uşor cu ajutorul automatelor cu λ-mişcări. Operaţiile le vom da şi cu ajutorul unor diagrame, care permit o înţelegere mai uşoară.

Vom nota un automat determinist cu ajutorul următoarei diagrame:



unde s-a notat cu un cerculeţ şi o săgeată spre interior starea iniţială, şi cu două cerculeţe concentrice stările finale.

Fie $\mathcal{A}_1 = (A_1, Q_1, E_1, \{i_1\}, F_1)$ şi $\mathcal{A}_2 = (A_2, Q_2, E_2, \{i_2\}, F_2)$ cele două automate care reprezintă operanzii operaţiilor şi $\mathcal{A} = (A, Q, E, \{i\}, F)$ automatul rezultat.

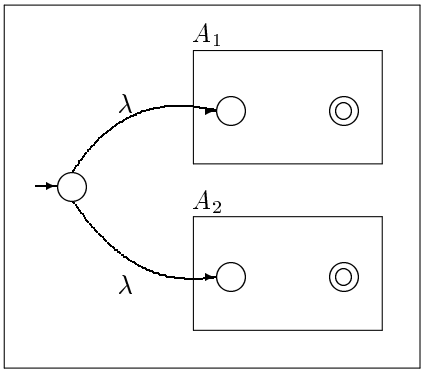
Reuniune

$\mathcal{A} := \mathcal{A}_1 \cup \mathcal{A}_2$, unde

$Q := Q_1 \cup Q_2 \cup \{i\}$,
$A := A_1 \cup A_2$,
$F := F_1 \cup F_2$,
$E := E_1 \cup E_2 \cup \{(i,\lambda,i_1),(i,\lambda,i_2)\}$

Sub forma de diagramă operaţia de reuniune se prezintă astfel;

$A_1 \cup A_2$

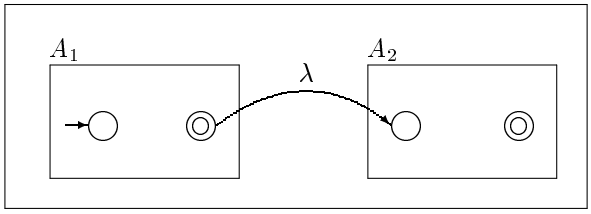


Produs

$\mathcal{A} := \mathcal{A}_1 \cdot \mathcal{A}_2$, unde

$Q := Q_1 \cup Q_2$,
$A := A_1 \cup A_2$,
$F := F_2$,
$i := i_1$,
$E := E_1 \cup E_2 \cup \{(p, \lambda, i_2) \mid p \in F_1\}$

$A_1 \cdot A_2$

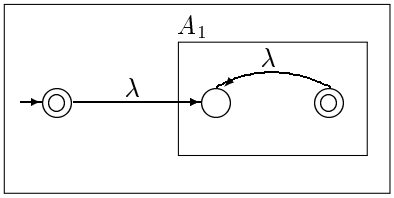


Iteraţie

$\mathcal{A} := \mathcal{A}_1^*$, unde

$Q := Q_1 \cup \{i\}$,
$A := A_1$,
$F := F_1 \cup \{i\}$,
$E := E_1 \cup \{i, \lambda, i_1)\} \cup \{p, \lambda, i_1) \mid p \in F_2\}$

A_1^*



Minimizarea automatelor finite

Prezentăm un algoritm care oricărui automat finit determinist îi asociează un automat finit determinist echivalent, care însă are un număr minim de stări. S-ar putea ca cele două să coincidă, dacă automatul respectiv are deja un număr minim de stări.

Două stări distincte p şi q al unui automat se numesc *echivalente* dacă ambele sunt productive sau ambele sunt neproductive.

Dacă două stări nu sunt echivalente, atunci ele se numesc *diferenţiate*. În algoritmul următor marcăm cu o câte stea stările diferenţiate, iar cele echivalente le comasăm. Pe parcursul algoritmului unor perechi de stări le vom ataşa liste de alte perechi de stări.

Algoritmul de mai jos se aplică automatelor finite deterministe din care s-au eliminat stările inaccesibile.

Algoritm pentru minimizarea automatelor finite

• Să le marcăm cu câte o stea perechile de stări $\{p, q\}$ pentru care avem $p \in F$ şi $q \notin F$ sau invers.

• Pentru fiecare pereche $\{p, q\}$ nemarcată, să verificăm toate perechile $\{p', q'\}$ pentru care avem $(p, a, p') \in E$ şi $(q, a, q') \in E$ pentru fiecare $a \in A$. Dacă cel puţin o pereche $\{p', q'\}$ este marcată, se marchează şi perechea $\{p, q\}$, împreună cu toate elementele listei ataşate perechii $\{p, q\}$, dacă o astfel de listă există. Dacă nici una din perechile studiate nu este marcată, tuturor acestor perechi le asociem în câte o listă perechea iniţială $\{p, q\}$. Acest pas se continuă atât timp cât este posibil.

La sfârşitul algoritmului perechile nemarcate vor fi cele echivalente, deci ele pot fi comasate. Astfel se reduce numărul stărilor.

Exemplu. Să se consideră automatul din figura 14. Vom folosi un tabel pentru marcare cu stea a perechilor de stări. Marcarea perechii $\{p, q\}$ se face printr-o stea pusă la intersecţia liniei p şi a coloanei q (sau a liniei q şi a coloanei p).

Prima dată le marcăm perechile: $\{c, a\}$, $\{c, b\}$, $\{c, d\}$, $\{c, e\}$ şi $\{c, f\}$ (c este singura stare finală). Pe urmă le luăm pe rând perechile nemarcate şi încercăm să le marcăm conform algoritmului.

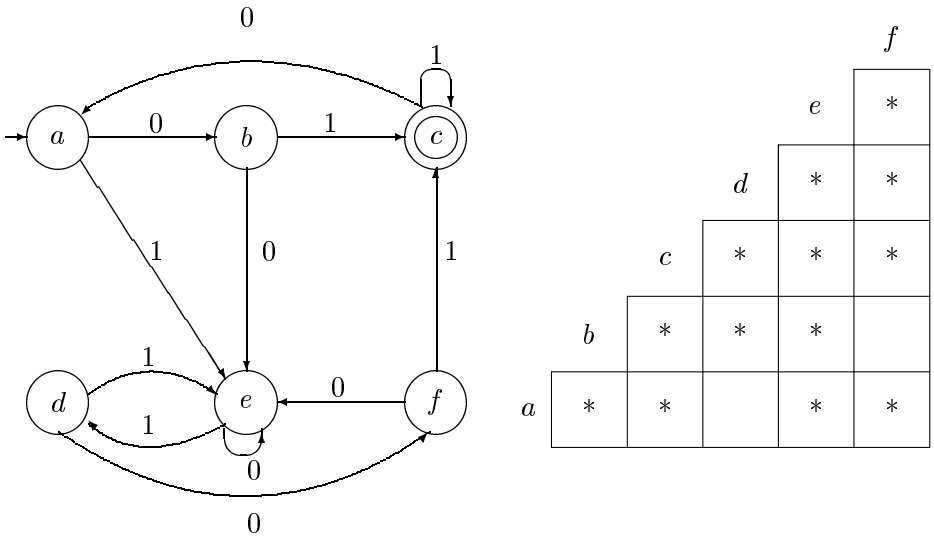


Figura 14: Minimizarea automatelor finite

Să începem cu perechea $\{a, b\}$. Îi asociem perechile :
– $\{b, e\}$ deoarece există în automat arcele $(a, 0, b)$ şi $(b, 0, e)$,
– $\{e, c\}$ deoarece există în automat arcele $(a, 1, e)$ şi $(b, 1, c)$
şi deoarece $\{e, c\}$ este deja marcată, se marchează şi perechea $\{a, b\}$.

În cazul perechii $\{a, d\}$ cele două perechi rezultate vor fi $\{b, f\}$ şi $\{e, e\}$. Perechii $\{b, f\}$ asociem o lista cu singurul element $\{a, d\}$.

Acum continuând cu $\{b, f\}$, obţinem perechiile $\{e, e\}$ şi $\{c, c\}$, cărora conform algoritmului nu i se asociează nimic.

Continuăm cu $\{a, e\}$. Perechiile corespunzătore vor fi $\{b, e\}$ şi $\{e, d\}$. Nici una din ele nu este marcată, deci le asociăm amândurora perechea $\{a, e\}$.

Continuând cu perechea $\{b, e\}$, obţinem perechiile $\{e, e\}$ şi $\{c, d\}$, şi deoarece perechea din urmă este marcată, se marchează şi perechea $\{b, e\}$, şi din lista asociată acesteia, şi perechea $\{a, e\}$. Continuând mai departe, se ajunge la tabelul de mai sus, din care se vede că $a \equiv d$ şi $b \equiv f$. După comasarea stărilor echivalente, ajungem la automatul din figura 15, echivalent cu cel iniţial, şi care are un număr minim de stări.

Teorema 34. *Oricărui automat finit nedeterminist i se poate ataşa un automat finit determinist echivalent.*

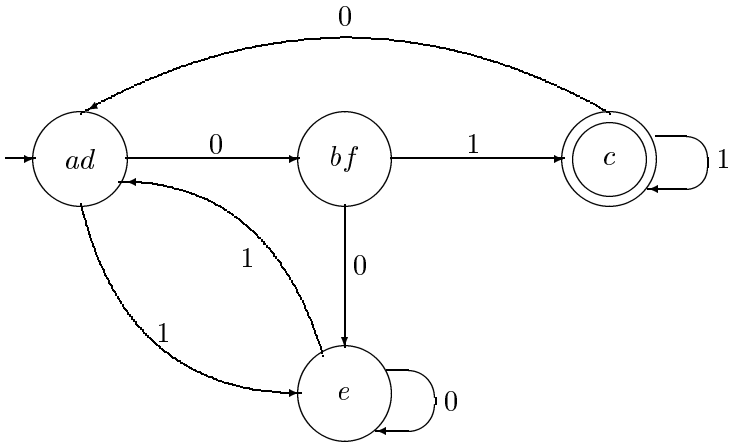


Figura 15: Automat minimizat

Vom da un algoritm, care transformă orice automat finit nedeterminst într-unul echivalent cu el dar care este determinist. Fie $\mathcal{A} = (A, Q, E, I, F)$ automatul nedeterminist şi să construim automatul determinist $\overline{\mathcal{A}} = (A, \overline{Q}, \overline{E}, i, \overline{F})$ echivalent cu el. Am folosit i pentru a marca singura stare iniţială. Deci $i \in \overline{Q}$. Definim:

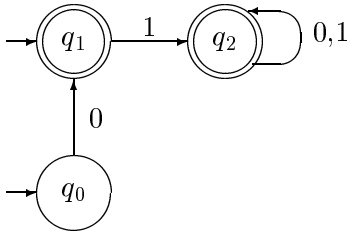
$\overline{Q} := \mathcal{P}(Q) \setminus \emptyset$

$(S, a, R) \in \overline{E}$ pentru $a \in A$, $S, R \in \overline{Q}$, unde $R = \bigcup_{q \in S} \delta(q, a)$

$i := I$

$\overline{F} := \{S \subseteq Q \mid S \cap F \neq \emptyset\}$

Exemplu. Se dă automatul următor:



cu tabelul de tranziţii

δ	0	1
q_0	$\{q_1\}$	$\emptyset$
q_1	$\emptyset$	$\{q_2\}$
q_2	$\{q_2\}$	$\{q_2\}$

Tabelul de tranziţii al noului automat este:

$\overline{\delta}$	0	1
$\{q_0\}$	$\{q_1\}$	$\emptyset$
$\{q_1\}$	$\emptyset$	$\{q_2\}$
$\{q_2\}$	$\{q_2\}$	$\{q_2\}$
$\{q_0, q_1\}$	$\{q_1\}$	$\{q_2\}$
$\{q_0, q_2\}$	$\{q_1, q_2\}$	$\{q_2\}$
$\{q_1, q_2\}$	$\{q_2\}$	$\{q_2\}$
$\{q_0, q_1, q_2\}$	$\{q_1, q_2\}$	$\{q_2\}$

Dacă folosim notaţiile:

$s_0 := \{q_0, q_1\}$,
$s_1 := \{q_0\}$,
$s_2 := \{q_1\}$,
$s_3 := \{q_2\}$,
$s_4 := \{q_0, q_2\}$,
$s_5 := \{q_1, q_2\}$,
$s_6 := \{q_0, q_1\}$.

obţinem tabelul următor:

$\overline{\delta}$	0	1
s_0	$\{s_2\}$	$\{s_3\}$
s_1	$\{s_2\}$	$\emptyset$
s_2	$\emptyset$	$\{s_3\}$
s_3	$\{s_3\}$	$\{s_3\}$
s_4	$\{s_5\}$	$\{s_3\}$
s_5	$\{s_3\}$	$\{s_3\}$
s_6	$\{s_5\}$	$\{s_3\}$

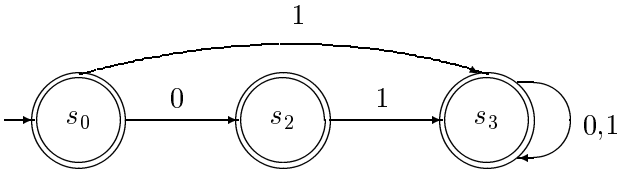
Stările accesibile sunt:

$U_0 = \{s_0\}$,
$U_1 = \{s_0, s_2, s_3\}$,
$U_2 = \{s_0, s_2, s_3\} = U_1 = U$

deci tabelul devine:

$\overline{\delta}$	0	1
s_0	$\{s_2\}$	$\{s_3\}$
s_2	$\emptyset$	$\{s_3\}$
s_3	$\{s_3\}$	$\{s_3\}$

Toate stările sunt şi stări finale. Noul automat este:



Există mai multe tipuri de automate acceptoare. Acestea corespund claselor de limbaje din ierarhia lui Chomsky. Limbajele acceptate de autoamte finite sunt cele regulare. Acest lucru rezultă din teorema următoare.

Teorema 35. *Pentru orice automat finit $\mathcal{A}$ există o gramatică regulară $\mathcal{G}$ care generează limbajul acceptat de automatul $\mathcal{A}$, şi invers, pentru orice gramatică regulară $\mathcal{G}$ se poate construi un automat finit $\mathcal{A}$, care acceptă limbajul gerenat de gramatica $\mathcal{G}$.*

Demonstraţie.
Cazul $\mathcal{A} \to \mathcal{G}$:

Dacă automatul $\mathcal{A}$ este nedeterminist, i se poate asocia un automat finit determinist echivalent cu el. Deci putem considera că $\mathcal{A}$ este un automat finit determinist: $\mathcal{A} = (A, Q, E, i, F)$. Să construim gramatica regulară $\mathcal{G} = (N, T, P, S)$:

- $N := Q$
- $T := A$
- $S := i$
- Se defineşte regula $p \to aq$ pentru fiecare arc $(p, a, q) \in E$. Dacă $q \in F$, atunci pe lângă regula de mai înainte se defineşte şi regula $p \to a$. Toate aceste reguli formează mulţimea regulilor P.

Se poate vede imediat că $L(\mathcal{G}) = L(\mathcal{A}) \setminus \{\lambda\}$, deoarece unui drum productiv din automatul $\mathcal{A}$ îi corespunde o derivare în $\mathcal{G}$ care începe cu simbolul start şi se termină într-un cuvânt cu litere terminale. Deci, dacă $w \in L(\mathcal{A})$ atunci $w \in L(\mathcal{G})$.

Cazul $\mathcal{G} \to \mathcal{A}$:

Plecând de la gramatica regulară $\mathcal{G} = (N, T, P, S)$ se defineşte automatul finit (nedeterminist în general) $\mathcal{A} = (A, Q, E, I, F)$:

- $Q := N \cup \{W\}$, unde $W \notin N \cup T$ (deci este un simbol nou),
- definim arcul (X, a, Y) pentru fiecare regulă $X \to aY$,
- definim arcul (X, a, W) pentru fiecare regulă $X \to a$,
- $I := \{S\}$,
- $F := \begin{cases} \{W\} & \text{dacă nu există regula } S \to \lambda, \\ \{W, S\} & \text{dacă există regula } S \to \lambda. \end{cases}$

Şi acum se poate vede imediat că $L(\mathcal{A}) = L(\mathcal{G})$, deoarece unei derivări în $\mathcal{G}$ care începe cu simbolul de start şi se termină într-un cuvânt cu litere terminale îi corespunde în $\mathcal{A}$ un drum productiv. Deci, dacă $w \in L(\mathcal{G})$ atunci $w \in L(\mathcal{A})$. □

Automatele finite pot accepta şi cuvinte infinite. Considerăm un drum infinit într-un automat finit:

$$(p_0, a_1, p_1), (p_1, a_2, p_2), \ldots, (p_{n-1}, a_n, p_n), \ldots$$

Acest drum infinit se numeşte *productiv*, dacă $p_0 \in I$ şi dacă conţine o infinitate de stări finale, adică $p_n \in F$ pentru o infinitate de n-uri. Se spune despre cuvântul infinit corespunzător $a_1 a_2 \ldots a_n \ldots$ că este acceptat de automatul respectiv. Astfel de automate se numesc *automate în sens Büchi*. Automatul din figura 16 este un automat finit care acceptă cuvintele peste un alfabet binar, care repetă secvenţa ab de o infinitate de ori.

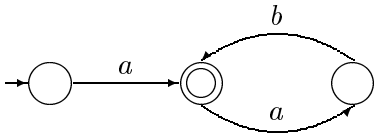


Figura 16: Automat finit în sens Büchi

Automatele finite pot fi generalizate în sensul că arcelor sunt ataşate nu litere, ci cuvinte peste alfabetul automatului. Adică un *automat finit generalizat* este un ansamblu (A, Q, E, I, F), unde $E \subseteq Q \times E^* \times Q$. Celelalte elemente sunt definite ca la automatul finit. Automatul generalizat din figura 17 acceptă limbajul $L = \{x(cx)^* \mid x = aba \text{ sau } x = abb\}$.

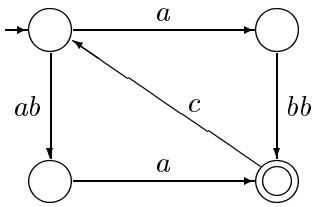
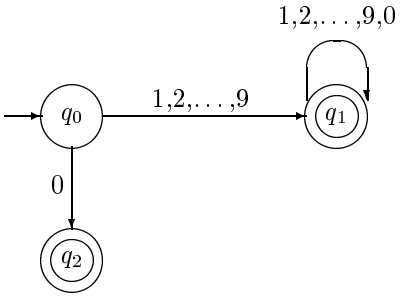
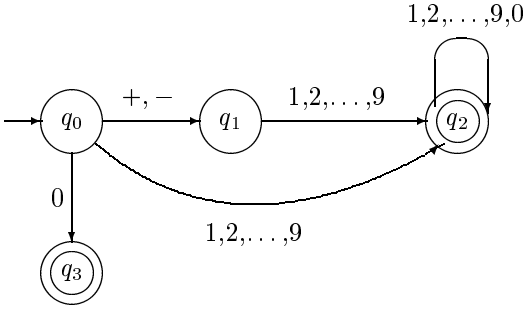


Figura 17: Automat finit generalizat

Aplicaţii ale automatelor finite în analiza lexicală. În prima fază a procesului de compilare trebuie recunoscute elementele lexicale. Acest lucru se poate face cu ajutorul automatelor finite. De exemplu, automat următor recunoaşte numerele naturale:



Automatul următor recunoaşte numerele întregi cu sau fără semn:

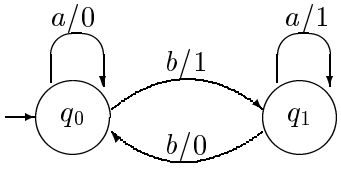


Translatoare

Definiţia unui translator se deosebeşte de cea a unui automat finit numai prin faptul că utilizează două alfabete A şi B, iar arcele sunt definite ca fiind elementele mulţimii $E \subseteq Q \times A \times B \times Q$. Un arc (p, a, b, q) înseamnă că trecând din starea p în starea q litera a se transformă în b, şi se mai notează prin $p \xrightarrow{a/b} q$. În cazul unui translator nu există stări finale. Pentru orice drum

$$q_0 \xrightarrow{a_1/b_1} q_1 \xrightarrow{a_2/b_2} \ldots \xrightarrow{a_{n-1}/b_{n-1}} q_{n-1} \xrightarrow{a_n/b_n} q_n$$

unde $q_0 \in I$, automatul transformă cuvântul $a_1 a_2 \ldots a_n$ în cuvântul $b_1 b_2 \ldots b_n$. Un exemplu de translator se găseşte în figura 18, împreună cu tabelul de tranziţii.



	a	b
q_0	$q_0, 0$	$q_1, 1$
q_1	$q_1, 1$	$q_0, 0$

Figura 18: Un translator

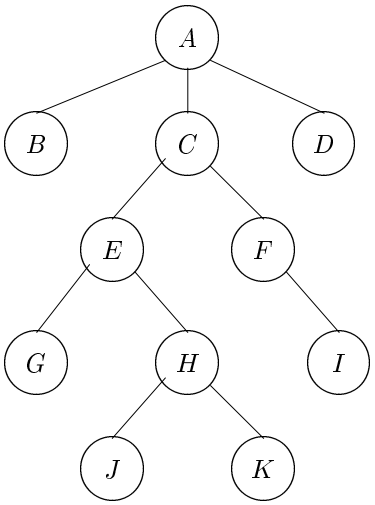
Acest translator „traduce" cuvântul *baabbab* în 1110110. Se poate vedea uşor că transformă un cuvânt $a_1 a_2 \ldots a_m$ în $b_1 b_2 \ldots b_m$, unde

$$b_k = \begin{cases} 1, & \text{dacă } a_1 a_2 \ldots a_k \text{ conţine un număr impar de } b\text{-uri} \\ 0, & \text{dacă } a_1 a_2 \ldots a_k \text{ conţine un număr par de } b\text{-uri} \end{cases}$$

Fără a intra în detalii, prezentăm în tabelul următor corespondenţa dintre automate şi clase de limbaje.

tip automat	**clasă de limbaje**
automate finite	limbaje regulare (de tipul 3)
automate pushdown nedeterministe	limbaje independente de context (de tipul 2)
automate liniar mărginite	limbaje dependente de context (de tipul 1)
automate Turing	limbaje generale (de tipul 0)

Codificarea arborilor cu rădăcină. Dacă vârfurile unui arbore cu rădăcină sunt din mulţimea V, atunci arborele poate fi codificat cu ajutorul unui cuvânt peste alfabetul A care este format din elementele lui V la care se adaugă parantezele (,) şi ,(vigula). De exemplu pentru arborele



cuvântul ataşat este: $A(B, C(E(G, H(J, K)), F(I)), D)$

Să notăm descendenţii vârfului v prin $D(v)$, iar codul arborelui cu rădăcina x prin $\lambda(x)$. Codificarea se face după următoarea descriere recursivă.

- Dacă v este o frunză, atunci $\lambda(v) = v$.
- Dacă $D(v) = \{v_1, v_2, \ldots, v_n\}$, atunci $\lambda(v) = v(\lambda(v_1), \lambda(v_2), \ldots, \lambda(v_n))$.

Se poate verifica uşor dacă o astfel de formulă este corectă. Nu pot fi corecte formulele care conţin două paranteze stânga alăturate, două litere din V alăturate sau două virgule alăturate. Se verifă cu ajutorul următoarei funcţii dacă parantezele sunt aşezate corect. Se defineşte $\delta(v_1 v_2 \ldots v_k)$ (unde v_i reprezintă literele şi parantezele din codul arborelui, după omiterea virgulelor) astfel:

$$\delta(v_i) = \begin{cases} 0 & \text{dacă } i = 1 \\ \delta(v_{i-1}) & \text{dacă } v_i \in V \\ \delta(v_{i-1}) + 1 & \text{dacă } v_i \text{ este (} \\ \delta(v_{i-1}) - 1 & \text{dacă } v_i \text{ este)} \end{cases}$$

Cuvântul reprezintă un cod corect dacă:

- $v \notin D(v)$ pentru orice $v \in V$,
- $\delta(v_i) > 0$ pentru $1 < i < n$
- $\delta(v_k) = 0$.

Pentru exemplul anterior avem:

A	(	B	C	(	E	(	G	H	(	J	K	)	)	F	(	I	)	)	D	)
0	1	1	1	2	2	3	3	3	4	4	4	3	2	2	3	3	2	1	1	0

Codificarea se poate realiza şi fără paranteze, folosind indici pentru a marca numărul descendenţilor unui vârf. Dacă notăm codul în acest caz prin $\nu(v)$ pentru arborele cu rădăcina v, atunci

- Dacă x este o frunză, atunci $\nu(x) = x_0$.
- Dacă $D(x) = \{v_1, v_2, \ldots, v_n\}$, atunci $\nu(x) = x_n(\nu(v_1), \nu(v_2), \ldots, \nu(v_n))$.

Pentru exemplul anterior avem: $\nu(A) = A_3B_0C_2G_0H_2J_0K_0F_1I_0D_0$. Şi în acest caz se poate defini o funcţie cu ajutorul căreia se poate determina dacă un cuvânt de forma de mai sus este sau nu corect.

Probleme.

1. Se consideră homomorfismul: $\sigma(0) = 01,\ \sigma(1) = 02,\ \sigma(2) = 0$ şi se porneşte cu 0. Fie r cuvântul infinit care este punctul fix al acestui homomorfism, adică pentru care avem $\sigma(r) = r$. Să se demonstreze că: $f_r(n) = 2n + 1$.

2. Se consideră cuvintele $w_0 = 0$ şi $w_1 = 12$, şi regula de generare $w_n = w_{n-1}^2 w_{n-2}$ pentru $n \geq 2$. Să se demonstreze că pentru cuvântul $w = \lim_{n\to\infty} w_n$ avem $f_w(n) = n + 2$.

3. Se consideră homomorfismul: $\sigma(0) = 0010,\ \sigma(1) = 1$ şi (evident) se porneşte cu 0. Dacă $s = \sigma(s)$ (cuvântul lui Chacon), să se demonstreze că: $f_s(n) = 2n - 1$ pentru $n \geq 2$.

4. Să se demonstreze că pentru orice cuvânt u sturmian numărul factorilor de lungime n speciali la dreapta este egal cu numărul factorilor de lungime n speciali la stânga şi acest număr este egal cu $f_u(n + 1) - f_u(n)$.

5. Să se demonstreze că în cazul cuvintelor sturmiene orice factor bispecial este un palindrom (cuvânt egal cu reversul lui) şi pentru orice factor special la dreapta reversul factorului este special la stânga.

Rezolvarea problemelor

Pag. 26. Permutări, aranjamente, combinări

1. Se poate porni de la formula (2) şi prin aplicare succesivă se ajunge la formula dorită.

$$\binom{n+1}{k+1} = \binom{n}{k} + \binom{n}{k+1} = \binom{n}{k} + \binom{n-1}{k} + \binom{n-1}{k+1} = \dots$$

Formula se poate demonstra şi prin inducţie asupra lui n. Pasul principal, în care se foloseşte ipoteza inducţiei:

$$\underbrace{\binom{k}{k} + \binom{k+1}{k} + \dots + \binom{n}{k}} + \binom{n+1}{k} = \binom{n+1}{k+1} + \binom{n+1}{k} =$$

$$= \binom{n+2}{k+1}$$

A treia metodă foloseşte formula binomului sub forma: $(1+x)^n = \sum_{k=0}^{n} \binom{n}{k} x^k$. Membrul I al identităţii de demonstrat este egal cu coeficientul lui x^k în expresia:

$$(1+x)^k + (1+x)^{k+1} + \dots (1+x)^n,$$

ceea ce, folosind formula de însumare a unei progresii geometrice, se transformă în

$$\frac{(1+x)^k \left((1+x)^{n-k+1} - 1\right)}{x} = \frac{(1+x)^{n+1}}{x} - \frac{(1+x)^k}{x},$$

în care termenul în x^k are coeficientul egal cu $\binom{n+1}{k+1}$, ceea ce demonstrează formula dorită.

2.a). În formula (1) se pune $a = 1$ şi $b = 1$ şi se obţine formula căutată:

$$2^n = \sum_{k=0}^{n} \binom{n}{k}$$

2.b). Se porneşte de la formula (1) a binomului pentru $a = 1, b = x$:

$$(1+x)^n = \sum_{k=0}^{n} \binom{n}{k} x^k \tag{61}$$

După derivarea ambilor membri, se obţine:

$$n(1+x)^{n-1} = \sum_{k=1}^{n} k\binom{n}{k} x^{k-1}$$

Formula se obţine pentru $x = 1$.

2.c). Se integrează ambii membri din formula (61):

$$\frac{1}{n+1}(1+x)^{n+1} = \sum_{k=0}^{n} \frac{1}{k+1}\binom{n}{k} x^{k+1} + C \tag{62}$$

Constanta C se poate determina luând $x = 0$. Se obţine: $C = \frac{1}{n+1}$. Punând pe urmă $x = 1$ în formula (62) se obţine exact formula care trebuie demonstrată.

3. În formula (6) a polinomului se pune $a_1 = a_2 = \ldots = a_n = 1$.

4. Se porneşte de la membrul drept:

$$\begin{aligned}
&P_{i_1-1,i_2,\ldots,i_k} + P_{i_1,i_2-1,\ldots,i_k} + \ldots + P_{i_1,i_2,\ldots,i_k-1} = \\
&= \frac{(n-1)!}{(i_1-1)!i_2!\ldots i_k!} + \frac{(n-1)!}{i_1!(i_2-1)!\ldots i_k!} + \ldots + \frac{(n-1)!}{i_1!i_2!\ldots(i_k-1)!} = \\
&= \frac{1}{n}\left(i_1\frac{n(n-1)!}{i_1(i_1-1)!i_2!\ldots i_k!} + \ldots + i_k\frac{n(n-1)!}{i_1!i_2!\ldots i_k(i_k-1)!}\right) = \\
&= \frac{1}{n}(i_1+i_2+\ldots+i_k)P_{i_1,i_2,\ldots,i_k} = P_{i_1,i_2,\ldots,i_k}
\end{aligned}$$

5. Se porneşte de la definiţia combinărilor generalizate:

$$\begin{aligned}
&\binom{\frac{1}{2}}{n+1} = \frac{\frac{1}{2}\left(\frac{1}{2}-1\right)\left(\frac{1}{2}-2\right)\cdots\left(\frac{1}{2}-(n+1)+1\right)}{(n+1)\cdot n\cdots 2\cdot 1} = \\
&= \frac{\frac{1}{2}\cdot\left(-\frac{1}{2}\right)\left(-\frac{3}{2}\right)\cdots\left(-\frac{2n-1}{2}\right)}{1\cdot 2\cdots n\cdot(n+1)} =
\end{aligned}$$

$$= \frac{(-1)^n}{2^{n+1}} \cdot \frac{1 \cdot 3 \cdots (2n-1)}{1 \cdot 2 \cdots n \cdot (n+1)} \cdot \frac{2 \cdot 4 \cdot 6 \cdots 2n}{2 \cdot 4 \cdot 6 \cdots 2n} =$$
$$= \frac{(-1)^n}{2^{n+1}} \cdot \frac{(2n)!}{n!(n+1)} \cdot \frac{1}{2^n \cdot n!} = \frac{(-1)^n}{2^{2n+1}} \cdot \frac{(2n)!}{n!n!(n+1)} =$$
$$= \frac{(-1)^n}{2^{2n+1}(n+1)} \binom{2n}{n}$$

6. Se pleacă de la formula $(1+x)^{r+s} = (1+x)^r(1+x)^s$, şi se identifică coeficienţii termenului x^n în ambii membri, astfel

$$\binom{r+s}{n} = \sum_{k=0}^{n} \binom{r}{k} \binom{s}{n-k}$$

7.a) Ţinem cont de formula $\binom{n}{k} = \binom{n}{n-k}$. În rândul al doilea combinările sunt scrise în ordinea inversă, astfel încât combinările egale între ele apar una deasupra celeilalte.

$$2^{2n} = \sum_{k=0}^{2n} \binom{2n}{k} = \binom{2n}{0} + \binom{2n}{1} + \ldots + \binom{2n}{n-1} + \binom{2n}{n}$$
$$+ \binom{2n}{2n} + \binom{2n}{2n-1} + \ldots + \binom{2n}{n+1}$$
$$= 2\sum_{k=0}^{n} \binom{2n}{k} - \binom{2n}{n}. \quad \text{De unde: } \sum_{k=0}^{n} \binom{2n}{n} = 2^{2n-1} + \frac{1}{2}\binom{2n}{n}.$$

7.b) Se aplică tehnica de la punctul anterior.

8. $\displaystyle\sum_{k=0}^{n} (n-k)\binom{2n}{k} = n\sum_{k=0}^{n} \binom{2n}{k} - \sum_{k=0}^{n} k\binom{2n}{k}.$

Dar $\displaystyle k\binom{2n}{k} = 2n\binom{2n-1}{k-1}$,

şi ţinând cont de formulele de la punctul anterior, suma cerută este egală cu:

$$n2^{2n-1} + \frac{n}{2}\binom{2n}{n} - 2n2^{2n-2} = \frac{n}{2}\binom{2n}{n}.$$

9. Se aplică formula lui Vandermonde pentru $r = s = n$ şi se ţine cont de formula $\binom{n}{k} = \binom{n}{n-k}$.

10. Se pune $x = 1$ în formula (19).

11. Se foloseşte substituţia $x = \frac{1}{y}$.

Pag. 64. Funcţii generatoare

---

1. *Metoda 1.* Se porneşte de la membrul stâng şi se aplică formula (24):

$$\sum_{k\geq 0}\binom{n+k-1}{k}z^k = z^{-n+1}\sum_{k\geq 0}\binom{n+k-1}{n-1}z^{n+k-1}$$

$$= \frac{1}{z^{n-1}}\cdot\frac{z^{n-1}}{(1-z)^n} = \frac{1}{(1-z)^n}$$

Metoda 2. Se porneşte de la membrul drept şi se aplică formula (13):

$$\frac{1}{(1-z)^n} = (1-z)^{-n} = \sum_{k\geq 0}\binom{-n}{k}(-z)^k$$

$$= \sum_{k\geq 0}(-1)^k\binom{n+k-1}{k}(-1)^k z^k = \sum_{k\geq 0}\binom{n+k-1}{k}z^k$$

2. Se foloseşte formula cunoscută: $\overline{C}_n^k = \overline{C}_n^{k-1} + \overline{C}_{n-1}^k$. Fie funcţia generatoare $C_n(z) = \sum_{k\geq 0}\overline{C}_n^k z^k$. Se poate scrie

$$C_n(z) = \overline{C}_n^0 + \sum_{k\geq 1}\overline{C}_n^k z^k = \overline{C}_n^0 + \sum_{k\geq 1}\overline{C}_n^{k-1}z^k + \sum_{k\geq 1}\overline{C}_{n-1}^k z^k$$

$$= z\sum_{k\geq 0}\overline{C}_n^{k-1}z^k + \overline{C}_n^0 + \sum_{k\geq 1}\overline{C}_{n-1}^k z^k = zC_n(z) + C_{n-1}(z)$$

De unde se obţine:

$$C_n(z) = \frac{1}{1-z}C_{n-1}(z).$$

Prin convenţie $C_0(z) = 1$. Ţinând cont şi de problema precedentă, se obţine:

$$C_n(z) = \frac{1}{(1-z)^n} = \sum_{k\geq 0}\binom{n-k+1}{k}z^k.$$

3. Folosim formula binomului pentru $(1-4z)^{\frac{1}{2}}$.

$$\sqrt{1-4z} = \sum_{n\geq 0} \binom{\frac{1}{2}}{n} (-4z)^n$$

Dar, printr-un calcul direct sau folosind formula din problema 5 pag. 26, avem

$$\binom{\frac{1}{2}}{n} = \frac{(-1)^{n-1}}{2^{2n-1}n} \binom{2n-2}{n-1}$$

Dar

$$\binom{2n-2}{n-1} = \frac{n}{2(2n-1)} \binom{2n}{n}$$

Astfel

$$\binom{\frac{1}{2}}{n} = \frac{(-1)^{n-1}}{4^n(2n-1)} \binom{2n}{n}.$$

Deci

$$\begin{aligned} \sqrt{1-4z} &= \sum_{n\geq 0} \binom{\frac{1}{2}}{n} (-4z)^n = \sum_{n\geq 0} \frac{(-1)^{n-1}}{4^n(2n-1)} \binom{2n}{n} (-4)^n z^n \\ &= \sum_{n\geq 0} \frac{-1}{2n-1} \binom{2n}{n} z^n \end{aligned}$$

4. Pornim de la formula cunoscută

$$\sum_{n\geq 0} \binom{2n}{n} z^n = \frac{1}{\sqrt{1-4z}}$$

Înmulţim această formulă cu ea însăşi, membru cu membru. Obţinem

$$\sum_{n\geq 0} \sum_{k=0}^{n} \binom{2k}{k} \binom{2n-2k}{n-k} z^n = \frac{1}{1-4z}$$

Membrul drept se dezvoltă în serie:

$$\frac{1}{1-4z} = 1 + 4z + 4^2 z^2 + \ldots + 4^n z^n + \ldots$$

De unde, egalând coeficienţii termenilor în z^n, obţinem:

$$\sum_{k=0}^{n} \binom{2k}{k} \binom{2n-2k}{n-k} = 4^n.$$

5. Înmulţim membru cu membru următoarele egalităţi:

$$\sqrt{1-4z} = \sum_{n\geq 0} \frac{-1}{2n-1}\binom{2n}{n} z^n$$

$$\frac{1}{\sqrt{1-4z}} = \sum_{n\geq 0} \binom{2n}{n} z^n$$

şi obţinem:

$$1 = \sum_{n\geq 0}\sum_{k=0}^{n} \frac{-1}{2k-1}\binom{2k}{k}\binom{2n-2k}{n-k} z^n$$

de unde, egalând coeficienţii termenilor corespunzători, rezultă:

$$\sum_{k=0}^{n} \frac{-1}{2k-1}\binom{2k}{k}\binom{2n-2k}{n-k} = \delta_{n0},$$

deoarece în membrul stâng numai coeficientul lui z^0 diferă de 0.

Pag. 86. Principiul includerii şi al excluderii

---

1. Să notăm cu A_i mulţimea tuturor permutărilor a n elemente care au un punct fix în i. Atunci avem $\#A_i = (n-1)!$. Evident că pentru cazul a k puncte fixe avem: $\#(A_{i_1} \cap \ldots \cap A_{i_k}) = (n-k)!$ pentru indici diferiţi (fixăm k poziţii şi celelalte pot fi permutate). Folosind varianta specială de la pagina 76 a formulei (38), obţinem pentru numărul $P(n)$ al permutărilor fără puncte fixe, formula:

$$P(n) = \sum_{i=0}^{n} (-1)^i \binom{n}{i} (n-i)! = n! \left(\sum_{i=0}^{n} (-1)^i \frac{1}{i!} \right).$$

2. Perechile pot dansa în $P(n)$ moduri, unde $P(n)$ este numărul permutărilor a n elemente fără puncte fixe de la problema anterioară.

3. Considerăm toate combinările de n elemente luate câte k. Numărul $N^{(i)}$ al combinărilor care nu conţin i elemente specificate este $\binom{n-i}{k-i}$. Aplicând varianta specială a formulei (38) de la pagina 76 obţinem exact formula cerută.

4. Să considerăm descompunerea lui n în factori primi, deci $n = p_1^{\alpha_1} p_2^{\alpha_2} \dots p_r^{\alpha_r}$. Demonstraţia se face prin inducţie după $\alpha_1 + \alpha_2 + \ldots + \alpha_r$ (vezi [68]). Se verifică pentru $\alpha_1 = 1$, $\alpha_2 = 0$, $\ldots$, $\alpha_2 = 0$. Se presupune adevărată pentru orice $\alpha_1 + \alpha_2 + \ldots + \alpha_r \leq q - 1$ şi se demonstrează pentru $\alpha_1 + \alpha_2 + \ldots + \alpha_r = q$. Mulţimea divizorilor lui n se împarte în două: divizorii care conţin pe p_1 până la puterea $\alpha_1 - 1$ (aceştia de fapt sunt divizorii lui $\dfrac{n}{p_1}$ şi divizorii care conţin pe $p_1^{\alpha_1}$. Să notăm mulţimea celor din categoria întâi cu D_1 şi mulţimea celorlalţi cu D_2. Deasemenea să mai folosim notaţiile: $m_1 = \dfrac{n}{p_1}$ şi $m_2 = \dfrac{n}{p_1^{\alpha_1}}$. Atunci

$$\begin{aligned}
\sum_{d|n} \varphi(d) &= \sum_{d \in D_1} \varphi(d) + \sum_{d \in D_2} \varphi(d) \\
&= \sum_{d|m_1} \varphi(d) + \varphi(p_1^{\alpha_1}) \sum_{d|m_2} \varphi(d) \\
&= m_1 + p_1^{\alpha_1} \left(1 - \frac{1}{p_1}\right) m_2 \\
&= \frac{n}{p_1} + n - \frac{n}{p_1} = n
\end{aligned}$$

Am folosit următoarele:

$\displaystyle\sum_{d|m_1} \varphi(d) = m_1 = \frac{n}{p_1}$ (conform ipotezei inducţiei),

$\displaystyle\sum_{d|m_2} \varphi(d) = m_2$ (conform ipotezei inducţiei),

$\varphi(p_1^{\alpha_1} \cdot m_2) = \varphi(p_1^{\alpha_1}) \cdot \varphi(m_2)$ (pentru că cei doi factori sunt relativ primi),

$\varphi(p_1^{\alpha_1}) = p_1^{\alpha_1} \left(1 - \dfrac{1}{p_1}\right)$ (vezi formula pentru funcţia φ la pag. 76).

Pag. 109. Numere remarcabile

---

1. Toate se demonstrează prin inducţie asupra lui n.

a. Pentru $n = 1$ egalitatea se verifică. Presupunem adevărată formula pentru n şi demonstrăm pentru $n + 1$:

$$\underbrace{F_2 + F_4 + \cdots + F_{2n}}_{} + F_{2n+2} = (F_{2n+1} - 1) + F_{2n+2} = F_{2n+3} - 1 =$$

$= F_{2(n+1)+1} - 1$

b. Pentru $n = 1$ avem $F_1 = F_2 = 1$.

$\underbrace{F_1 + F_3 + \cdots F_{2n-1}} + F_{2n+1} = F_{2n} + F_{2n+1} = F_{2n+2}$

c. Pentru $n = 1$ avem $F_1^2 = F_1 F_2 = 1$.

$\underbrace{F_1^2 + F_2^2 + \cdots + F_n^2} + F_{n+1}^2 = F_n F_{n+1} + F_{n+1}^2 = F_{n+1}(F_n + F_{n+1}) =$

$= F_{n+1} F_{n+2}$

d. Pentru $n = 1$ avem $F_1 F_2 = F_2^2 = 1$.

$\underbrace{F_1 F_2 + F_2 F_3 \cdots + F_{2n-1} F_{2n}} + F_{2n} F_{2n+1} + F_{2n+1} F_{2n+2} =$

$= F_{2n}^2 + F_{2n} F_{2n+1} + F_{2n+1} F_{2n+2} = F_{2n}(F_{2n} + F_{2n+1}) + F_{2n+1} F_{2n+2} =$

$= F_{2n} F_{2n+2} + F_{2n+1} F_{2n+2} = (F_{2n} + F_{2n+1}) F_{2n+2} = F_{2n+2}^2$

e. Pentru $n = 1$ avem $F_1 F_2 + F_2 F_3 = 1 + 2 = 3 = F_3^2 - 1$

$\underbrace{F_1 F_2 + F_2 F_3 + \cdots + F_{2n} F_{2n+1}} + F_{2n+1} F_{2n+2} + F_{2n+2} F_{2n+3} =$

$= (F_{2n+1}^2 - 1) + F_{2n+1} F_{2n+2} + F_{2n+2} F_{2n+3} =$

$= F_{2n+1}(F_{2n+1} + F_{2n+2}) - 1 + F_{2n+2} F_{2n+3} =$

$= F_{2n+1} F_{2n+3} + F_{2n+2} F_{2n+3} - 1 = (F_{2n+1} + F_{2n+2}) F_{2n+3} - 1 =$

$= F_{2n+3}^2 - 1$

f. Pentru $n = 1$ avem $F_3 = 2 = \dfrac{F_5 - 1}{2}$.

$\underbrace{F_3 + F_6 + \cdots + F_{3n}} + F_{3n+3} = \dfrac{F_{3n+2} - 1}{2} + F_{3n+3} =$

$= \dfrac{(F_{3n+2} + F_{3n+3}) + (F_{3n+3} - 1)}{2} = \dfrac{F_{3n+4} + F_{3n+3} - 1}{2} = \dfrac{F_{3n+5} - 1}{2}$

2. Se demonstrează prin inducţie asupra lui n. Pentru $n = 1$ proprietatea este evident adevărată. Se foloseşte formula (56) din pagina 96:

$$f_{nk}^{(l)} = f_{(n-1)k+k}^{(l)} = f_{(n-1)k+1}^{(l)} f_k^{(l)} + f_{(n-1)k}^{(l)} f_{k-1}^{(l)}.$$

Primul termen evident se divide prin $f_k^{(l)}$, iar al doilea deasemenea din cauza ipotezei inducţiei.

3. a. Se ţine cont de următoarele: $F_n = \frac{1}{\sqrt{5}}(\alpha^n - \beta^n)$, $\quad 1 + \alpha = \alpha^2$, $1 + \beta = \beta^2$:

$$\sum_{k\geq 0} \binom{n}{k} F_k = \frac{1}{\sqrt{5}} \sum_{k\geq 0} \binom{n}{k} (\alpha^k - \beta^k) = \frac{1}{\sqrt{5}} \Big((1+\alpha)^n - (1+\beta)^n \Big)$$

$$= \frac{1}{\sqrt{5}} (\alpha^{2n} - \beta^{2n}) = F_{2n}$$

3. b. Se folosesc formulele de la punctul anterior.

$$\sum_{k\geq 0} \binom{n}{k} F_{m+k} = \frac{1}{\sqrt{5}} \sum_{k\geq 0} \binom{n}{k} (\alpha^{m+k} - \beta^{m+k})$$

$$= \frac{\alpha^m}{\sqrt{5}} \sum_{k\geq 0} \binom{n}{k} \alpha^k - \frac{\beta^m}{\sqrt{5}} \sum_{k\geq 0} \binom{n}{k} \beta^k = \frac{\alpha^m}{\sqrt{5}} (1+\alpha)^n - \frac{\beta^m}{\sqrt{5}} (1+\beta)^n$$

$$= \frac{1}{\sqrt{5}} (\alpha^m \alpha^{2n} - \beta^m \beta^{2n}) = \frac{1}{\sqrt{5}} (\alpha^{m+2n} - \beta^{m+2n}) = F_{m+2n}$$

4. Demonstraţia se face prin inducţie. $F_3 = 2$ este par. Presupunem că $F_{3(k-1)}$ este par. Atunci $F_{3k} = F_{3k-1} + F_{3k-2} = \underbrace{2F_{3k-2}}_{\text{par}} + \underbrace{F_{3k-3}}_{\text{par cf. ip. ind}}$ este par.

$F_1 = 1, F_2 = 1$ sunt impare. Presupunem $F_{3(k-1)+1} = F_{3k-2}$ impar. Atunci $F_{3k+1} = F_{3k} + F_{3k-1} = \underbrace{2F_{3k-1}}_{\text{par}} + \underbrace{F_{3k-2}}_{\text{impar cf. ipt. ind}}$ impar. La fel şi pentru F_{3k-1}.

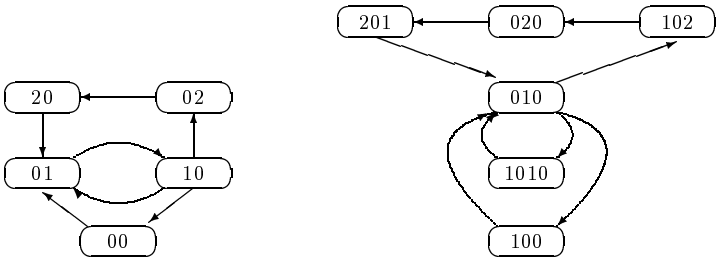
Pag. 168. Combinatorica cuvintelor

---

1. Cuvântul r foloseşte 3 litere, deci $f_r(1) = 3$. Cuvântul este

$$r = 0102010010201010201001020\ldots$$

Se poate observa din forma homomorfismului că 0 poate fi urmat de 0, 1 sau 2, pe când atât 1 cât şi 2 poate fi urmat numai de 0.



Încercăm să calculăm diferenţa $f_r(n+1) - f_r(n)$. În graful Rauzy pentru $n = 2$ şi $n = 3$ (vezi figura alăturată) se observă că există un singur cuvânt care se poate continua în trei feluri şi unul singur care este continuarea a trei cuvinte (cele două nu neaparat distincte). Toate celelalte se pot continua într-un singur fel. Prin inducţie completă demonstrăm că acest lucru se întâmplă şi în cazul general. Fie $a_1a_2 \dots a_n \in F_n(r)$ care se poate continua în trei feluri, deci $a_1a_2 \dots a_nb \in F_{n+1}(r)$, unde $b \in \{0, 1, 2\}$. Dar în acest caz $ca_1a_2 \dots a_n \in F_{n+1}$ se poate continua exact de trei ori (de remarcat că din construcţia grafului rezultă că c este unic determinat). Toate celelalte cuvinte se pot continua într-un singur fel. Deci putem afirma că $f_r(n+1) = f_r(n)+2$, de unde rezultă imediat (ţinând cont şi de $f_r(1) = 3$) că $f_r(n) = 2n+1$. Cuvântul r se numeşte cuvântul Arnoux-Rauzy.

Pentru k litere avem generalizarea:

$\sigma(a_0) = a_0a_1,$

$\sigma(a_1) = a_0a_2,$

$\dots$

$\sigma(a_{k-2}) = a_0a_{k-1},$

$\sigma(a_{k-1}) = 0.$

Dacă s este punctul fix al homomorfismului, atunci $f_s(n) = (k-1)n + 1$.

2. Se procedează ca în problema anterioară. Şi această problemă poate fi generalizată la un alfabet cu k litere, şi în acest caz $f_w(n) = n + k - 1$ [27].

3. Se procedează ca la problemele anterioare.

4. Pentru un cuvânt sturmian u orice factor de lungime n special la dreapta se poate prelungi la dreapta cu o literă în două moduri, iar ceilalţi factori

numai într-un singur mod. Dar prin aceste prelungiri se obţin toţi factorii de lungime $n+1$. Fie $d_u(n)$ numărul factorilor speciali la dreapta de lungime n, atunci $f_u(n) - d_u(n)$ reprezintă numărul factorilor de lungime n care nu sunt speciali la dreapta. Atunci $f_u(n+1) = 2d_u(n) + \big(f_u(n) - d_u(n)\big)$, de unde $d_u(n) = f_u(n+1) - f_u(n)$. Se procedează la fel şi pentru factorii speciali la stânga.

5. Se demonstrează prin inducţie asupra lungimii factorilor că dacă $v \in F_n(u)$ atunci şi $v^R \in F_n(u)$. Pentru $n = 2$ avem factorii 00, 01, 10, pentru care afirmaţia este adevărată. Presupunem adevărată pentru n. Fie $v = a_1a_2 \ldots a_n \in F_n(u)$ şi fie $a_1a_2 \ldots a_nb \in F_{n+1}(u)$, dar atunci $a_2 \ldots a_nb \in F_n(u)$, şi conform ipotezei inducţiei avem $a_n \ldots a_2a_1 \in F_n(u)$ şi $ba_n \ldots a_2 \in F_n(u)$, de unde rezultă că $ba_n \ldots a_2a_1 \in F_{n+1}(u)$.

Bibliografie

[1] Andrásfai, B., *Introductory Graph Theory*, Akadémiai Kiadó, Budapest & Adam Hilger Ltd., Bristol & Pergamon Press Inc. Elmsford, New York, 1977.

[2] Anisiu, M-C., Blázsik, Z., Kása, Z., Maximal Complexity of Finite Words, *Pure Math. and Appl.*, **13**, 1–2 (2002) pp. 39–48.

[3] Arnoux, P., Mauduit, C., Shiokawa, I., Tamura, J-I., Complexity of Sequences Defined by Billiards in the Cube, *Bull. Soc. Math. France*, **122** (1994) pp. 1–12.

[4] Arnoux, P., Rauzy, G., Représentation géométrique de suites de complexité $2n + 1$, *Bull. Soc. Math. France*, **119** (1991) pp. 199–215.

[5] Baase, S., *Computer Algorithms: Introduction to Design and Analysis*, Addison-Wesley Publ. Co., 1988.

[6] Baryshnikov, Yu., Complexity of Trajectories in Rectangular Billiards, *Commun. Math. Phys.*, **174** (1995) pp. 43–56.

[7] Bege, A., Kása, Z., Coding Objects Related to Catalan Numbers, *Studia Universitatis Babeş-Bolyai, Informatica*, **46**, 1 (2001) pp. 31–40.

[8] Berge, C., *Teoria grafurilor şi aplicaţii*, Editura Tehnică, 1969.

[9] Berge, C., *Graphes et hypergraphes*, Dunod, Paris, 1970.

[10] Berstel, J., An Exercise on Fibonacci Representations, *Theor. Inform. Appl.*, **35**, 6 (2002) pp. 491–498.

[11] Bond, J., Iványi, A., Modeling of Interconnection Networks Using de Bruijn Graphs, *Third Conference of Program Designers,* July 1-3, 1987, Eötvös Loránd University, Faculty of Natural Sciences, Budapest, 1987, pp. 75-88.

[12] Cassaigne, J., Complexité et facteurs spéciaux, *Bull. Belg. Math. Soc. Simon Stevin*, **4**, 1 (1997) pp. 67–88.

[13] Choffrut, C., Karhumäki, J., Combinatorics of Words, *Handbook of Formal Languages, vol. I–III.*, Springer Verlag, 1997. ed. G. Rozemberg, A. Salomaa.

[14] Chartrand, G., Oellermann, O. R., *Applied and Algorithmic Graph Theory*, McGraw-Hill, Inc., 1993.

[15] Cofman, J., Catalan Numbers for the Classroom?, *Elemente der Mathematik*, **52** (1997) pp. 108–117.

[16] Cofman, J., Fibonacci-féle számoktól fraktálokig, *Polygon* (Szeged), **3**, 1 (1993) pp. 91–102.

[17] Comtet, L., *Advanced Combinatorics. The Art of Finite and Infinite Expansions*, D. Reidel Publ. Co., Dordrecht–Boston, 1974.

[18] Cormen, T. H., Leiserson, C. E., Rivest, R. R., *Introducere în algoritmi*, Editura Computer Libris Agora, Cluj, 2000.

[19] Cormen, T. H., Leiserson, C. E., Rivest, R. R., Stein, C., *Introduction to Algorithms*, Second edition, Mit Press—McGraw-Hill, 2001.

[20] Csákány, B., *Diszkrét matematikai játékok*, Polygon, Szeged, 1998.

[21] Demetrovics, J., Denev, J., Pavlov, R., *A számítástudomány matematikai alapjai*, Nemzeti Tankönyvkiadó, Budapest, 1999.

[22] Eggleton, R. B, Guy, R. K., Catalan Strikes Again! How Likely is a Function to be Convex?, *Mathematics Magazine*, **67**, 4 (1988) pp. 211–219.

[23] Ferenczi, S., Les tranformation de Chacon: combinatoire, structure géométrique, lien avec les systèmes de complexité $2n+1$, *Bull. Soc. math. France*, **123** (1995) pp. 271–292.

[24] Ferenczi, S., Complexity of Sequences and Dynamical Systems, *Discrete Math.*, **206**, 1–3 (1999) pp. 145–154.

[25] Ferenczi, S., Substitutions and Symbolic Dynamical Systems, Preprint, 1996.

[26] Ferenczi, S., Kása, Z., Complexity for Finite Factors of Infinite Sequences, *Theoretical Computer Science*, **218** (1999) pp.l 177–195.

[27] Ferenczi, S., Mauduit, C., Transcendence of Numbers with a Low Complexity Expansion, *Journal of Number Theory*, **67**, 2 (1997) pp. 146–161.

[28] Georgescu, H., Principiul cutiei al lui Dirichlet, *Gazeta de Informatică*, 1994, nr. 9-10.

[29] Giammarresi, D., Montalbano, R., Deterministic Generalized Automata, *Theoretical Computer Science*, **215** (1999) pp. 191–208.

[30] Gould, H. W., *Combinatorial Identities*, Morgantown, W. Va. 1972.

[31] Graham, R. L., Knuth, D. E., Patashnik, O., *Concrete Mathematics. A Foundation for Computer Science*, Addison-Wesley, 1994.

[32] Hajnal, P., *Összeszámlálási problémák*, Polygon, Szeged, 1997.

[33] Hopcroft, J., Ullmann, J. D., *Introduction to Automata Theory, Languages and Computation*, Addison–Wesley Publ. Co. 1979.

[34] Iorga, V., Fătu, I., Asupra partiţiilor unui număr natural, *Gazeta de Informatică*, 1993, nr. 2.

[35] Iványi, A, On the d-complexity of Words, *Annales Univ. Sci. Budapest. Sect. Comput.* **8** (1987) pp. 69-90.

[36] Kása, Z., Locating the Buddies in the Gneral Bddy Sstems, Studia Univ. Babeş-Bolyai, Math., **26**, 4 (1981) pp. 46–50.

[37] Kása, Z., Ţâmbulea, L., Binary Trees and Number of States in Buddy Systems, *Annales Universitatis Scientarium Budapestinensis de Rolando Eötvös Nominatae, Computatorica*, **7** (1987), pp. 1–10.

[38] Kása, Z. Computing the d-complexity of Words by Fibonacci-like Sequences, *Studia Univ. Babeş-Bolyai, Math.* **35**, 3 (1990) pp. 49-53.

[39] Kása, Z., On the d-complexity of Strings, *Pure Math. and Appl.*, **9**, 1–2 (1998) pp. 119–128.

[40] KÁSA, Z., *d*-complexity of Words and Generating Functions, *Babeş-Bolyai University, Faculty of Mathematics and Computer Science, Seminar on Computer Science, Preprint No.* 5, 1998 pp. 65–72.

[41] KÁSA, Z., BEGE, A., *Matematică discretă*, Centrul de Formare Continuă şi Învăţământ la Distanţă, Universitatea Babeş-Bolyai, Cluj-Napoca, 2003.

[42] KNUTH, D. E., *Tratat de programarea calculatoarelor* Vol. 1. *Algoritmi fundamentali*, Editura Tehnică, Bucureşti, 1974.

[43] KNUTH, D. E., *Tratat de programarea calculatoarelor* Vol. 3. *Căutare şi sortare*, Editura Tehnică, Bucureşti, 1974.

[44] LIVOVSCHI, L., GEORGESCU, H., *Sinteza şi analiza algoritmilor*, Editura Ştiinţifică şi Enciclopedică, Bucureşti, 1986.

[45] LIVOVSCHI, L., GEORGESCU, H., POPOVICI, C. P., ŢĂNDĂREANU, N., *Bazele informaticii*, Editura Didactică şi Pedagogică, Bucureşti, 1981.

[46] LEVÉ, F., SÉÉBOLD, P., Proof of a Conjecture on Word Complexity, *Bull. Belg. Math. Soc. Simon Stevin*, **8**, 2 (2001) pp. 277–291 .

[47] LOTHAIRE, M. *Combinatorics on Words*, Addison-Wesley Publishing Company, 1983.

[48] LOTHAIRE, M. *Algebraic Combinatorics on Words*, Cambridge University Press, 2002.

[49] LOVÁSZ L., *Combinatorial Problems and Exercises*, Akadémiai Kiadó, Budapest & North-Holland, Amsterdam, 1979.

[50] DE LUCA, A., On the Combinatorics of Finite Words, *Theoretical Computer Science*, **218** (1999) pp. 13–39.

[51] MARTIN, M. H., A Problem in Arrangements, *Bull. A. M. S.*, **40** (1934) pp. 859–864.

[52] MATEESCU, E., MAXIM, I., *Arbori*, Editura Ţara Fagilor, 1996.

[53] Moldovan, G., Cioban, V., Lupea, M., *Limbaje formale şi teoria automatelor. Culegere de probleme*, Editura Mesagerul, Cluj-Napoca, 1997. `http://math.ubbcluj.ro/~infodist/alf/INDEX.HTM`

[54] Nolan, J. M., Savage, C. D., Wilf, H. S., Basis Partitions, *Discrete Math.*, **179**, 1–3 (1998) pp. 277–283.

[55] Paterson, K. G., Interconnection Networks Based on Two-dimensional de Bruijn Graphs. *Applications of Combinatorial Mathematics. Based on the proceedings of a conference, Oxford, UK, December 14–16, 1994*. Oxford: Clarendon Press. Inst. Math. Appl. Conf. Ser., New Ser. 60, 169-184 (1997) Mitchell, Chris (ed.).

[56] Pawlak, Z., *Matematyczne aspekty procesu produkcyjnego*, Panstwowe Wydawnictwo Ekonomiczne, Warsaw, 1969. (trad. în maghiară, Budapest, 1971)

[57] Petkovšek, M., Wilf, H. S., Zeilberger, D., *A=B*, Wellesley, MA; A. K. Peters, 1996.

[58] Pósa, L., Előadása, Erdős-konferencia, 1999, *Matematikai Lapok*, **7**, 3–4 (1997) pp. 33–40.

[59] Révész, Gy., *Bevezetés a formális nyelvek elméletébe*, Akadémiai Kiadó, Budapest, 1979.

[60] Ralston, A., De Bruijn Sequences – A Model Example of the Interaction of Discrete Mathematics and Computer Science, *Math. Magazine*, **55**, 3 (1982) pp. 131–143.

[61] Riordan, J., *An Introduction to Combinatorial Analysis*, John Wiley & Sons. Inc., 1958.

[62] Riordan, J., *Combinatorial Identities*, John Wiley & Sons. Inc., 1968.

[63] Risley, R. N., Zamboni, L. Q., A Generalization of Sturmian Flows; Combinatorial Structure and Transcendence, *Acta Arith.*, **95**, 2 (2000) pp. 167–184.

[64] Şerbănaţi, L. D., *Limbaje de programare şi compilatoare*, Editura Academiei R. S. România , Bucureşti, 1987.

[65] Sipser, M., *Introduction to the Theory of Computation*, PWS Publishing Company, 1997.

[66] Szele, T., *Bevezetés az algebrába*, Tankönyvkiadó, Budapest, 1967.

[67] Tomescu, I., *Introducere în combinatorică*, Editura Tehnică, Bucureşti, 1972.

[68] Tomescu, I., *Probleme de combinatorică şi teoria grafurilor*, Editura Didactică şi Pedagogică, Bucureşti, 1981,

[69] Vilenkin, N. I., *Combinatorica*, Editura Nauka, Moscova, 1969. (trad. în maghiară. Budapest, 1971.)

[70] Wilf, H. S., The 'Snake Oil' Method for Proving Combinatorial Identities, *Surveys in Combinatorics*, 1989 (Norwich, 1989) pp. 208–217. London Math. Soc. Lecture Note, Ser. 141, Cambridge Univ. Press, Cambridge 1989.

[71] Wilf, H. S., *Generatingfunctionology*, Academic Press Inc., Boston, MA, 1996 (second edition).

Index