

Daniela ROȘCA

MATEMATICI SPECIALE:
PROBLEME DE NUMĂRARE,
PROBABILITĂȚI DISCRETE,
TEORIA GRAFURILOR

U.T.PRESS
Cluj-Napoca, 2016
ISBN 978-606-737-187-1

Daniela ROȘCA

MATEMATICI SPECIALE:

PROBLEME DE NUMĂRARE,

PROBABILITĂȚI DISCRETE,

TEORIA GRAFURILOR

U.T. PRESS

Cluj-Napoca, 2016

ISBN 978-606-737-187-1



Editura U.T.PRESS
Str.Observatorului nr. 34
C.P.42, O.P. 2, 400775 Cluj-Napoca
Tel.:0264-401.999 / Fax: 0264 - 430.408
e-mail: utpress@biblio.utcluj.ro
www.utcluj.ro/editura

Director: Ing. Călin D. Câmpean

Recenzia: Prof.dr. Dorian Popa
 Prof.dr. Neculae Vornicescu

Copyright © 2016 Editura U.T.PRESS

Reproducerea integrală sau parțială a textului sau ilustrațiilor din această carte este posibilă numai cu acordul prealabil scris al editurii U.T.PRESS.

ISBN 978-606-737-187-1

Prefață

Cursul de **Matematici Speciale**, incluzând capitolele *Combinatorică*, *Probabilități discrete* și *Teoria grafurilor*, este inclus în programa de studii a secției de Calculatoare și Tehnologia Informației în primul semestru al anului întâi. Capitolele mai sus menționate sunt ramuri ale așa-numitelor *matematici discrete*, numite și *matematici finite*, care se ocupă cu studiul obiectelor matematice care au o structură discretă, în sensul că nu necesită noțiunea de continuitate. Majoritatea obiectelor matematice studiate în matematicile discrete sunt mulțimi finite sau infinit numărabile, cum ar fi grafurile finite sau mulțimi de numere întregi.

Matematicile discrete au căpătat un mare interes în ultimii ani datorită aplicațiilor sale în științele calculatoarelor. Conceptele din matematicile discrete sunt utile în studiul sau descrierea obiectelor sau problemelor din algoritmi și limbaje de programare.

Pe lângă ramurile studiate aici, matematicile discrete mai includ: lanțuri Markov, algoritmica (studiul metodelor de calcul), teoria complexității (studiul limitărilor teoretice și practice ale algoritmilor), logica (studiul raționamentelor), teoria mulțimilor, geometria discretă și computațională, teoria codificării și criptologia, optimizarea discretă, algebra liniară etc.

Această carte este accesibilă oricărei persoane care are cunoștințe de matematică dobândite în liceu.

Cluj-Napoca, octombrie 2016

Daniela Roșca

Cuprins

1	Metode de numărare	7
1.1	Numărarea ca și bijecție. Principiul cutiei	7
1.2	Principii de bază	8
1.3	Numărarea perechilor	9
1.4	Funcții, cuvinte și selecții	11
1.4.1	Selecții ordonate cu repetiții (cuvinte)	11
1.4.2	Injectii ca și selecții ordonate fără repetiții	12
1.4.3	Permutări	13
1.4.4	Selecții neordonate fără repetiții	14
1.4.5	Selecții neordonate cu repetiții	15
1.5	Design-uri	16
1.6	Partiții și distribuții	17
1.6.1	Partiții ale unei mulțimi în submulțimi (partiții de mulțimi)	18
1.6.2	Distribuții, surjecții și numere multinomiale	19
1.6.3	Partiții întregi	22
1.7	Recurențe și funcții generatoare	22
1.8	Relații de recurență și funcții generatoare	23
1.8.1	Recurențe liniare	23
1.8.2	Funcții generatoare	24
1.8.3	Recurențe liniare neomogene	25
1.8.4	Recurența lui Catalan	27
1.8.5	Câteva proprietăți ale funcțiilor generatoare	28
1.9	Probleme rezolvate	29
2	Probabilități discrete	32
2.1	Evenimente aleatoare	32
2.2	Axiomele teoriei probabilităților	33
2.3	Probabilități condiționate și independență	35
2.4	Probabilitatea totală. Formula lui Bayes	37
2.5	Scheme probabilistice	39
2.5.1	Schema binomială	39
2.5.2	Schema multinomială	39
2.5.3	Schema hipergeometrică	40
2.5.4	Schema hipergeometrică generalizată	41
2.5.5	Schema urnelor lui Poisson	41
2.6	Variabile aleatoare	42
2.6.1	Variabile aleatoare discrete	43
2.6.2	Variabile aleatoare continue	44
2.7	Suma variabilelor aleatoare	45
2.8	Exemple de v.a.d.	46

2.8.1	Distribuția binomială	46
2.8.2	Distribuția hipergeometrică	46
2.8.3	Distribuția lui Poisson	47
2.8.4	Distribuția geometrică	47
2.8.5	Distribuția binomială negativă	48
2.8.6	Distribuția urnelor lui Poisson	49
2.9	Valoare medie și varianță	49
2.10	Covarianța	52
2.11	Valori medii și varianțe pentru câteva v.a.d.	53
2.12	Teorema lui Chebâșev	57
2.13	Legile numerelor mari	58
2.14	Probleme rezolvate	61
3	Teoria grafurilor	67
3.1	Introducere	67
3.1.1	Grafuri	67
3.1.2	Gradul unui vârf. Teorema lui Euler	68
3.1.3	Lanțuri, lanțuri simple, lanțuri elementare, cicluri în grafuri	69
3.1.4	Conectivitate în grafuri	71
3.1.5	Multigrafuri, grafuri orientate și multigrafuri orientate	72
3.1.6	Drumuri, drumuri simple, drumuri elementare, cicluri în grafuri orientate	74
3.1.7	Conectivitate și conectivitate tare în grafuri orientate .	75
3.2	Reprezentarea grafurilor	77
3.2.1	Matrice de adiacență	77
3.2.2	Matrice de incidență	82
3.3	Arbori, sortare și căutare	83
3.3.1	Arbori cu rădăcină	85
3.3.2	Arbori de decizie	86
3.3.3	Arbori de sortare	87
3.4	Arbori binari și coduri binare	89
3.4.1	Arbori binari	89
3.4.2	Coduri binare	91
3.5	Arbori de acoperire	96
3.5.1	Căutare în adâncime și căutare în lărgime	99
3.5.2	Grafuri cu ponderi și arbori economici	101
3.5.3	Arbore de acoperire minimă în grafuri orientate	105
3.5.4	Lanț minim. Algoritmul lui Dijkstra	107
3.6	Algoritmi de tip greedy	110
3.6.1	Algoritm greedy pentru problema ponderii maxime . .	111
3.6.2	Algoritm greedy pentru colorarea vârfurilor	112
3.7	Grafuri bipartite	114
3.7.1	Cuplaje	115
3.7.2	Cuplaje în grafuri bipartite	116
3.7.3	Cuplaj maxim	119
3.8	Grafuri hamiltoniene și euleriene	121
3.8.1	Grafuri hamiltoniene	122
3.8.2	Grafuri euleriene	123
3.8.3	Problema poștaşului	126
3.9	Rețele	128

3.9.1	Drumuri critice	128
3.9.2	Flux și tăietură	130
3.9.3	Flux maxim, tăietură minimă	132
3.9.4	Algoritmi pentru găsirea unui flux maxim cu valori întregi	135
Anexa A: Relații binare		137
A.1	Relații de echivalență	137
A.2	Relații de ordine	138

Capitolul 1

Metode de numărare

În multe probleme din teoria probabilităților sau știința calculatoarelor avem nevoie să calculăm numărul de evenimente rezultate în urma efectuării unui experiment, a obiectelor sau a operațiilor necesare unui algoritm. Acest capitol prezintă câteva metode de bază legate de cele mai comune situații întâlnite în problemele de numărare.

1.1 Numărarea ca și bijecție. Principiul cutiei

Cum putem număra o mulțime finită? În practică, de obicei indicăm obiectele și spunem *unu*, *doi*, *trei* etc, oprindu-ne atunci când am ajuns la ultimul obiect. Din punct de vedere matematic, această metodă înseamnă a construi o funcție de la mulțimea pe care vrem să o numărăm la mulțimea $\mathbb{N}_n = \{1, 2, \dots, n\}$. Această funcție trebuie să fie bijectivă: *fiecare* obiect trebuie numărat *o singură dată*. În consecință, prima definiție va fi:

Definiția 1.1 Fie $m \in \mathbb{N}$. Mulțimea A are m membri dacă există o bijecție de la A la \mathbb{N}_m .

Pentru a arăta că prin numărare nu putem obține rezultate diferite, avem nevoie de așa-numitul «principiu al cutiei» (al lui Dirichlet).

Teorema 1.1 (Principiul cutiei) Fie $m \in \mathbb{N}$. Atunci pentru orice $n \in \mathbb{N}$ este adevărată următoarea afirmație:

Dacă există o injecție de la \mathbb{N}_n la \mathbb{N}_m , atunci $n \leq m$.

O formulare echivalentă este următoarea: Dacă $n > m$, atunci nu există nici o injecție de la \mathbb{N}_n la \mathbb{N}_m . Aceasta înseamnă că, dacă $n > m$ și n obiecte sunt introduse în m cutii, atunci (cel puțin) o cutie va conține mai mult decât un obiect.

Folosind principiul cutiei, următoarea definiție determină o valoare unică pentru m .

Definiția 1.2 Dacă există o bijecție între A și \mathbb{N}_m , atunci spunem că A are **măsura** sau **cardinalul** m și scriem

$$|A| = m.$$

În continuare ne vom numi **n -mulțime** o mulțime A cu n elemente și o **k -submulțime** a lui A o submulțime cu k elemente.

Câteva aplicații ale principiului cutiei

Exemplul 1.1 *In orice mulțime cu mai mult de 13 persoane, există cel puțin două care au ziua de naștere în aceeași lună.*

«Cutile» sunt aici cele 12 luni și «obiectele» sunt persoanele.

Exemplul 1.2 *La o petrecere, ori de câte ore două persoane sunt prezentate una alteia, ele își dau mâna. Să se explice de ce la petrecere trebuie să fie (cel puțin) două persoane care dau mâna de același număr de ori.*

Soluție. Notăm cu A mulțimea persoanelor de la petrecere, $m = |A|$, și fie

$$f(x) = \text{numărul de persoane din } A \text{ cu care } x \text{ dă mâna.}$$

Valorile posibile ale lui $f(x)$ sunt $0, 1, \dots, m-1$, deci putem defini funcția $f : A \rightarrow B$, cu $B = \{0, 1, \dots, m-1\}$. Nu putem aplica principiul cutiei, deoarece $|A| = |B|$. Dar, dacă există o persoană x^* care dă mâna cu $m-1$ persoane, atunci fiecare dă mâna cu x^* , și prin urmare nu există nici o persoană care să nu fi dat mâna. În consecință, valorile 0 și $m-1$ nu pot fi luate simultan de către o funcție fixată f , deci $|B| \leq m-1$. Din principiul cutiei deducem că f nu este injectivă, deci există două persoane x_1, x_2 pentru care $f(x_1) = f(x_2)$. Deci x_1 și x_2 dau mâna de același număr de ori.

Această problemă poate fi formulată și în felul următor: Să se arate că în orice mulțime A de persoane există două care au același număr de prieteni în A . Presupunem că $|A| \geq 2$ și că, dacă x este prieten cu x' , atunci x' este prieten cu x .

Desigur, metoda «a arăta și a indica numărul» nu este suficientă pentru a număra mulțimi mai sofisticate, prin urmare e nevoie să studiem alte metode.

1.2 Principii de bază

Primul principiu este foarte simplu și poate fi formulat matematic astfel:

Principiul 1 (principiul adunării) Dacă A și B sunt mulțimi nevide și disjuncte, atunci

$$|A \cup B| = |A| + |B|.$$

Evident, rezultatul poate fi extins la orice reuniune de mulțimi două câte două disjuncte A_1, A_2, \dots, A_n adică

$$|A_1 \cup A_2 \cup \dots \cup A_n| = |A_1| + |A_2| + \dots + |A_n|. \quad (1.1)$$

Observația 1.1 *Principiul adunării spune că, atunci când există n cazuri astfel încât al i -lea caz are k_i opțiuni, pentru $i = 1, 2, \dots, n$, și oricare două cazuri nu au nici o opțiune în comun, atunci numărul total de opțiuni este*

$$k_1 + k_2 + \dots + k_n.$$

O aplicație simplă a formulei (1.1) este o formă generalizată a principiului cutiei: presupunem că un număr de obiecte sunt distribuite în n cutii și A_i reprezintă mulțimea obiectelor din cutia i , $1 \leq i \leq n$. Intrucât mulțimile A_i sunt disjuncte, numărul total de obiecte este $|A_1| + |A_2| + \dots + |A_n|$. Dacă presupunem că nici o cutie nu conține mai mult de k obiecte, numărul total de obiecte este cel mult

$$k + k + \dots + k = nk.$$

Exprimând contrariul afirmației de mai sus, obținem **principiul generalizat al cutiei**:

Teorema 1.2 *Dacă m obiecte sunt distribuite în n cutii și $m > nk$, atunci cel puțin o cutie conține cel puțin $k + 1$ obiecte.*

Exemplul 1.3 *Intr-o competiție în care echipele au cinci membri, regula este că membrii fiecărei echipe trebuie să aibă zilele de naștere în aceeași lună. Câte persoane sunt necesare pentru a garanta că se poate construi o echipă?*

Soluție. Aplicând principiul generalizat al cutiei, numărul de persoane este strict mai mare decât $4 \cdot 12$, deci numărul minim de persoane este 49. ■

În cazul când mulțimile A_i nu sunt în mod necesar disjuncte, avem următorul rezultat.

Principiul 2 (Principiul includerii și al excluderii) Dacă A_1, A_2, \dots, A_n sunt mulțimi finite, atunci

$$|A_1 \cup A_2 \cup \dots \cup A_n| = \alpha_1 - \alpha_2 + \alpha_3 - \dots (-1)^{n-1} \alpha_n, \quad (1.2)$$

unde α_i este suma cardinalelor intersecțiilor de i mulțimi:

$$\begin{aligned} \alpha_1 &= \sum_{i=1}^n |A_i|, \\ \alpha_2 &= \sum_{1 \leq i < j \leq n} |A_i \cap A_j|, \\ &\dots \\ \alpha_n &= \left| \bigcap_{i=1}^n A_i \right|. \end{aligned}$$

Demonstrarea principiului includerii și al excluderii se poate face prin inducție după n . Ca și o consecință imediată are loc următorul rezultat.

Corolarul 1.3 *Dacă A_1, A_2, \dots, A_n sunt submulțimi ale unei N -mulțimi date X , atunci numărul de elemente care nu aparțin nici uneia din aceste submulțimi este*

$$\begin{aligned} |X \setminus (A_1 \cup \dots \cup A_n)| &= |X| - |A_1 \cup \dots \cup A_n| \\ &= N - \alpha_1 + \alpha_2 - \dots + (-1)^n \alpha_n. \end{aligned} \quad (1.3)$$

O aplicație a acestui principiu este *problema deranjamentelor* (a se vedea secțiunea 1.9, pag. 30).

1.3 Numărarea perechilor

În practică avem nevoie câteodată să numărăm lucruri care pot fi descrise în mod natural ca și perechi de obiecte. De exemplu, vrem să calculăm încărcarea didactică pentru un an de studiu. Pentru a face acest lucru, construim un tabel, unde fiecare linie corespunde unui student și fiecare coloană corespunde unei discipline. Dacă un student x participă la cursul y atunci marcăm poziția corespunzătoare (x, y) din tabel. Numărul total de semne

este încărcarea didactică.

student \ curs	C_1	C_2	C_3	\dots	C_n	suma pe linii
S_1	✓	✓				.
S_2		✓			✓	.
S_3	✓	✓	✓			$r_x(S)$
\vdots						\vdots
S_m	✓				✓	.
suma pe coloane	.	.	$c_y(S)$	\dots	.	$ S $

Problema noastră va fi numărarea unei submulțimi S a produsului cartezian $X \times Y$, unde X este mulțimea studenților și Y este mulțimea disciplinelor.

Primul mod constă în a număra semnele de pe linia x și a găsi totalul pe linii $r_x(S)$, pentru fiecare $x \in X$. Numărul de elemente din S poate fi obținut adunând toate sumele pe linii,

$$|S| = \sum_{x \in X} r_x(S). \quad (1.4)$$

Al doilea mod constă în a calcula semnele din coloana y și a calcula totalul pe coloane $c_y(S)$, pentru fiecare $y \in Y$. În acest caz $|S|$ poate fi obținută prin adunarea tuturor sumelor pe coloane,

$$|S| = \sum_{y \in Y} c_y(S). \quad (1.5)$$

Evident, cele două sume trebuie să coincidă și aceste două expresii pentru $|S|$ sunt deseori utilizate în practică pentru a verifica rezultate, sau în teorie pentru a deduce anumite proprietăți ale cardinalelor mulțimilor X și Y .

Exemplul 1.4 *Intr-o grupă sunt 24 fete. Fiecare fată cunoaște 7 din băieții din grupă și fiecare băiat cunoaște 8 fete din grupă. Câți băieți sunt în grupă?*

Soluție. Notăm cu n numărul de băieți. Construim un tablou unde plasăm băieții pe coloane și fetele pe linii. De fiecare dată când băiatul x o cunoaște pe fata y , marcăm elementul (x, y) . Egalând suma de pe linii cu suma de pe coloane obținem $8n = 7 \cdot 24$, deci există $n = 21$ băieți.

Exemplul 1.5 *Este posibil să se găsească o mulțime de submulțimi ale lui \mathbb{N}_7 cu proprietatea că fiecare are 5 elemente și fiecare element al lui \mathbb{N}_7 aparține la exact 4 din aceste submulțimi?*

Soluție. Construim un tablou cu elementele $1, 2, \dots, 7$ pe linii și submulțimile cerute A_1, \dots, A_k ale lui \mathbb{N}_7 (k necunoscut în acest moment) pe coloane. Ori de câte ori $i \in A_k$, marcăm poziția (k, i) . Totalul pe linii va fi $4 \cdot 7 = 28$, în timp ce totalul pe coloane va fi $5 \cdot k$. Deoarece 5 nu este divizor al lui 28, problema are răspuns negativ.

O consecință simplă a egalităților (1.4) și (1.5) este al doilea principiu de bază:

Principiul 3 (Principiul multiplicării) Date fiind mulțimile X și Y , cardinalul mulțimii $X \times Y$ este dat de

$$|X \times Y| = |X| \cdot |Y|.$$

Evident, rezultatul poate fi extins la un produs finit de mulțimi:

$$|X_1 \times X_2 \times \dots \times X_n| = |X_1| \cdot |X_2| \cdot \dots \cdot |X_n|.$$

Observația 1.2 *Principiul multiplicării spune că, atunci când o procedură poate fi despărțită în n pași, astfel încât există k_1 opțiuni pentru pasul 1, și astfel încât, după efectuarea pasului $i - 1$ ($i = 1, 2, \dots, n$) există k_i opțiuni pentru pasul i , atunci numărul de moduri în care poate fi realizată procedura este*

$$k_1 \cdot k_2 \cdot \dots \cdot k_n.$$

1.4 Funcții, cuvinte și selecții

În cele mai multe probleme de numărare avem un număr n de obiecte care sunt distribuite în k clase sau grupuri. Numărul de moduri în care poate fi făcută această distribuție depinde de următoarele:

1. obiectele sunt distincte sau nu;
2. clasele sunt distincte sau nu;
3. ordinea obiectelor dintr-o clasă este relevantă sau nu;
4. ordinea claselor este relevantă sau nu;
5. obiectele pot fi folosite mai mult de o singură dată sau nu;
6. sunt admise clase vide sau nu.

Exercițiu 1.4 *Să se găsească un exemplu practic pentru fiecare din cazurile enumerate mai sus.*

1.4.1 Selecții ordonate cu repetiții (cuvinte)

Pentru o mulțime dată $Y \neq \emptyset$, considerăm mulțimea $\{f, f : \mathbb{N}_m \rightarrow Y\}$. Valorile luate de funcția f determină m -uplul

$$(f(1), f(2), \dots, f(m)) \in Y \times \dots \times Y := Y^m.$$

Reciproc, fiecare element din Y^m este un m -uplu (y_1, y_2, \dots, y_m) și corespunde unei funcții $f : \mathbb{N}_m \rightarrow Y$ definită prin

$$f(1) = y_1, f(2) = y_2, \dots, f(m) = y_m.$$

În concluzie, a defini o funcție $f : \mathbb{N}_m \rightarrow Y$ este echivalent cu a alege un element din Y^m . Dacă privim elementele mulțimii Y ca pe litere ale unui alfabet, atunci secvența $f(1), f(2), \dots, f(m)$ poate fi privită ca cele m litere ale unui *cuvânt*. De exemplu, dacă $Y = \{a, b, c, d\}$, cuvintele «*pom*» și «*alb*» corespund, respectiv, funcțiilor $f, g : \mathbb{N}_3 \rightarrow Y$ definite prin

$$\begin{aligned} f(1) &= p, f(2) = o, f(3) = m, \\ g(1) &= a, g(2) = l, g(3) = b. \end{aligned}$$

Definiția 1.3 *Un **cuvânt** de lungime m în alfabetul Y este o funcție $f : \mathbb{N}_m \rightarrow Y$.*

Teorema 1.5 *Fie X, Y mulțimi finite nevide și fie*

$$\mathcal{F} = \{f, f : X \rightarrow Y\}.$$

Dacă $|X| = m$ și $|Y| = n$, atunci

$$|\mathcal{F}| = n^m.$$

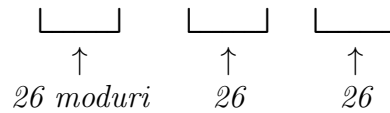
Demonstrație. Orice $f \in \mathcal{F}$ este determinat de m -uplul valorilor sale

$$(f(x_1), f(x_2), \dots, f(x_m)) \in Y^m.$$

Prin urmare, $|\mathcal{F}| = |Y^m| = n^m$ ■

O formulare echivalentă a acestei teoreme este: *Numărul cuvintelor de lungime m dintr-un alfabet Y cu n simboluri este n^m .*

Exemplul 1.6 *Intr-un alfabet cu 26 litere, există 26^3 cuvinte cu 3 litere, presupunând că nu există nici o restricție la formarea cuvintelor. O ilustrare intuitivă a obținerii acestui număr este următoarea:*



Aici cele trei cutii reprezintă cele trei «poziții» ale literelor cuvintelor, fiecare din ele fiind «completat» independent în 26 moduri.

Prin urmare, există un alt mod de a interpreta o funcție $f : \mathbb{N}_m \rightarrow Y$ (sau, echivalent, un cuvânt de lungime m în alfabetul Y): Pentru a construi cuvântul «*pom*», mai întâi am selectat litera p din stoc, apoi o , apoi m . Presupunem că avem un stoc nelimitat de litere. Astfel, fiecare cuvânt reprezintă o **selecție ordonată** a literelor alfabetului $Y = \{a, b, \dots, z\}$, **cu repetiții** permise. În concluzie, putem enunța următoarea teoremă.

Teorema 1.6 *Numărul de selecții ordonate cu repetiție a m obiecte din n -mulțimea Y este n^m .*

Exemplul 1.7 *Dacă X este o m -mulțime, atunci numărul total de submulțimi ale lui X este 2^m .*

Soluție. Presupunem că $X = \{x_1, x_2, \dots, x_m\}$ și fie Y alfabetul $\{0, 1\}$. Orice submulțime S a lui X corespunde unui cuvânt de lungime n din alfabetul Y , definit de funcția

$$f(i) = \begin{cases} 1, & \text{dacă } x_i \in S, \\ 0, & \text{dacă } x_i \notin S. \end{cases}$$

De exemplu, dacă $m = 7$ și $S = \{x_2, x_4, x_5\}$, cuvântul este 0101100. În concluzie, numărul de submulțimi distincte ale lui X coincide cu numărul de cuvinte de lungime m din alfabetul $\{0, 1\}$, care este 2^m .

1.4.2 Injecții ca și selecții ordonate fără repetiții

În unele situații ne este permis să avem doar un obiect de fiecare tip. De exemplu, dacă selectăm o echipă, nici un jucător nu poate fi selectat mai mult de o singură dată.

A realiza o *selecție ordonată* înseamnă a defini o funcție $f : \mathbb{N}_m \rightarrow Y$, cu $f(i)$ reprezentând al i -lea obiect selectat. A permite repetiții înseamnă că este posibil să avem $f(j) = f(k)$ pentru $j \neq k$. A nu permite repetiții înseamnă a impune ca f să fie *injectivă*. Suntem interesați să determinăm numărul de injecții. Acest număr este dat în teorema următoare.

Teorema 1.7 Numărul de selecții ordonate fără repetiții, a m obiecte dintr-o mulțime Y cu n elemente, este egal cu numărul de injecții $f : \mathbb{N}_m \rightarrow Y$ și este dat de formula

$$n(n-1)(n-2)\dots(n-m+1) = \frac{n!}{(n-m)!}. \quad (1.6)$$

Demonstrație. Valoarea $f(1)$ poate fi atribuită în n moduri. Dacă nu admitem repetiții, valoarea $f(2)$ poate fi atribuită în $n-1$ moduri, și în final, valoarea $f(m)$ poate fi atribuită în $n-m+1$ moduri. Din regula de multiplicare rezultă formula (1.6). ■

Observația 1.3 Numerele $n(n-1)\dots(n-m+1)$ apar destul de frecvent în probleme de numărate, astfel că au primit următoarele denumiri:

$$n^{\underline{m}} := n(n-1)(n-2)\dots(n-m+1)$$

sunt numite **factoriale descendente**¹ (de n de lungime m). Analog se consideră

$$n^{\overline{m}} := n(n+1)(n+2)\dots(n+m-1)$$

care se vor numi **factoriale ascendente**.²

Observația 1.4 Numărul $n^{\underline{m}}$ este de fapt numărul de cuvinte de lungime m ale căror litere, elemente din Y (cu $|Y| = n$), sunt diferite. Din acest motiv, aceste cuvinte se numesc **m -permutări** ale lui Y sau **aranjamente** de n luate câte m . De exemplu, 1234 și 6512 sunt două 4-permutări ale lui $Y = \{1, 2, \dots, 6\}$.

Exemplul 1.8 Câte cuvinte din 4 litere se pot forma dintr-un alfabet de 9 simboluri, dacă nici o literă nu se poate folosi mai mult de o singură dată?

Răspuns. $9^{\underline{4}} = 9 \cdot 8 \cdot 7 \cdot 6 = 3024$.

Exemplul 1.9 Pentru o clasă formată din 30 elevi, în câte moduri se pot acorda premii la sfârșit de an, știind că există un premiu întâi, un premiu doi și un premiu trei?

Răspuns. $30^{\underline{3}} = 30 \cdot 29 \cdot 28 = 24360$.

1.4.3 Permutări

Presupunem că un număr de obiecte sunt așezate aleator unul după altul. Din punct de vedere matematic, acest lucru înseamnă a defini o funcție de la mulțimea de «poziții» ocupate de obiecte, la mulțimea de obiecte.

Definiția 1.4 O **permutare** a unei mulțimi finite $X \neq \emptyset$ este o bijecție $\sigma : X \rightarrow X$.

Pentru a simplifica scrierea, în mod frecvent se consideră, în loc de o n -mulțime X arbitrară, mulțimea \mathbb{N}_n . Mulțimea tuturor permutărilor lui \mathbb{N}_n va fi notată prin Π_n . Ca o consecință imediată a teoremei 1.7, avem:

¹In limba engleză *falling factorials*.

²In limba engleză *rising factorials*.

Teorema 1.8 Numărul permutărilor mulțimii \mathbb{N}_n este

$$|\Pi_n| = n(n-1) \cdot \dots \cdot 2 \cdot 1 = n! \quad (1.7)$$

O permutare $\pi \in \Pi_n$ va fi notată cu

$$(\pi(1), \pi(2), \dots, \pi(n)).$$

Exemplul 1.10 Patru persoane se pot așeza în șir indian în $4! = 24$ moduri.

Definiția 1.5 O permutare $\pi \in \Pi_n$ se numește **deranjament** dacă $\pi(i) \neq i$ pentru orice $i \in \mathbb{N}_n$.

Astfel, $(3, 2, 1)$ nu este deranjament al lui \mathbb{N}_3 , dar $(2, 3, 1)$ este deranjament. Numărul de deranjamente va fi calculat în secțiunea 1.9, pag. 30.

1.4.4 Selecții neordonate fără repetiții

Presupunem că avem o mulțime X cu n obiecte și selectăm k din ele. Rezultatul selecției este o k -submulțime Y . În acest model, important este rezultatul selecției, și nu procesul de selecție. De asemenea, nu există posibilitatea de repetiție (fiecare membru al lui X este în Y sau nu și nici un membru nu poate fi selectat mai mult de o singură dată). De exemplu, dintr-o clasă de 30 elevi, vrem să selectăm 3 elevi pentru a participa la o excursie gratuită în jurul lumii. Desigur nimeni nu va fi interesat de ordinea în care s-a făcut selecția, ci doar de rezultatul selecției.

Numărul de selecții neordonate fără repetiții a k obiecte dintr-o n -mulțime X , este numărul de k -submulțimi ale mulțimii X . În general, acest număr este notat cu $\binom{n}{k}$ (se citește *combinări de n luate câte k*) și se numește **număr binomial**³.

Teorema 1.9 Numărul binomial $\binom{n}{k}$ are expresia

$$\binom{n}{k} = \frac{n(n-1) \dots (n-k+1)}{k!} = \frac{n^{\underline{k}}}{k!} = \frac{n!}{k!(n-k)!}. \quad (1.8)$$

Demonstrație. Cele k elemente ale k -submulțimii pot fi selectate în $n(n-1) \dots (n-k+1)$ moduri, la fel cum am făcut pentru selecțiile ordonate fără repetiții. Aceste k obiecte pot fi permutate între ele în $k!$ moduri. Deoarece ordinea nu contează, alegem doar unul din cele $k!$ moduri, astfel că în final numărul de k -submulțimi ale unei n -mulțimi va fi

$$\frac{n(n-1) \dots (n-k+1)}{k!}.$$

■

O proprietate imediată este $\binom{n}{k} = \binom{n}{n-k}$.

Observația 1.5 Numărul binomial poate fi definit într-un caz mai general, ca

$$\binom{x}{k} = \frac{x(x-1) \dots (x-k+1)}{k!}, \quad (1.9)$$

pentru $x \in \mathbb{R}$, $k \in \mathbb{N} \cup \{0\}$.

³In cele mai multe din cărțile românești se folosește notația C_n^k pentru numărul binomial $\binom{n}{k}$.

Exemplul 1.11 *Dintr-o clasă de 30 elevi putem selecta o echipă de 7 elevi care să participe la o excursie în $\binom{30}{7} = 2035800$ moduri.*

Exemplul 1.12 (Numărarea șirurilor de biți de lungime n cu exact k zerouri) *Care este numărul de șiruri de biți de lungime n cu exact k zerouri?*

Soluție. Fiecare șir de biți este determinat prin alegerea unei k -submulțimi a n -mulțimii de poziții. Astfel, zerourile sunt plasate în aceste k poziții și unitățile în cele $n - k$ poziții rămase, așa cum s-a făcut în exemplul 1.7, pag. 12. În concluzie, numărul cerut este $\binom{n}{k}$.

1.4.5 Selecții neordonate cu repetiții

Suntem interesați acum de cazul selecțiilor neordonate cu repetiții.

Exemplul 1.13 *Să se scrie selecțiile neordonate de patru obiecte din mulțimea $\{a, b, c\}$, cu repetiții permise.*

Soluție. Există 15 asemenea selecții și acestea sunt:

aaaa	aaab	aaac	aabb	aabc
aacc	abbb	abbc	abcc	accc
bbbb	bbbc	bbcc	bccc	cccc

Cum se pot număra în cazul general? Ideea este să reprezentăm asemenea selecții ca și cuvinte cu litere din alfabetul $\{0, 1\}$. De exemplu, cuvântul asociat selecției $abcc$ va fi 101011. Zerourile sunt *markeri* care separară tipurile de obiecte, iar unitățile ne specifică numărul de obiecte din fiecare tip. Dacă nu există nici obiect de un anumit tip, nu se pune nimic. Astfel, în exemplul nostru, această asociere va fi

a	b	c	c
1	0	1	0
1	0	1	1

Problema se reduce astfel la a determina numărul de șiruri de biți de lungime șase, care conțin exact două zerouri. Acest număr este $\binom{6}{2} = 15$ (a se vedea exemplul 1.12, pag. 15).

În general, putem enunța următoarea teoremă.

Teorema 1.10 *Numărul de selecții neordonate, cu repetiții, a k obiecte dintr-o mulțime de n obiecte este $\binom{n+k-1}{k}$.*

Demonstrație. Intrucât selecțiile sunt neordonate, fără a altera generalitatea problemei putem aranja obiectele astfel ca, în cadrul fiecărei selecții, toate obiectele de un anumit tip apar primele, urmate de obiectele de un alt tip etc. După aceea, se poate asocia fiecărei selecții câte un cuvânt de lungime $n + (k - 1)$ din alfabetul $\{0, 1\}$, așa cum am explicat în exemplul anterior. Presupunem că avem n_i obiecte de tipul i . Primele n_1 litere ale cuvântului vor fi 1, urmate de un 0, următoarele n_2 litere vor fi din nou 1, urmate de un 0, și așa mai departe. Dacă pentru un i avem $n_i = 0$, atunci vom avea doi markeri consecutivi. De exemplu, pentru selecțiile $accc$ și $cccc$ cuvintele vor fi 100111 și respectiv 000111. Asocierea definită prin această regulă este o

bijecție de la mulțimea de selecții la mulțimea de cuvinte de lungime $n+k-1$ care conțin exact $n-1$ zerouri. În concluzie, numărul de cuvinte este

$$\binom{n+k-1}{n-1} = \binom{n+k-1}{k}. \quad (1.10)$$

■

Observația 1.6 Putem remarca faptul că a defini o selecție neordonată cu repetiții a k obiecte dintr-o mulțime de n obiecte este echivalent cu a introduce k bile identice în n cutii distincte. Intr-adevăr, construcția de șiruri de biți poate fi făcută luând câte un 1 pentru fiecare bilă și 0 ca marker care separă cutiile.

Exemplul 1.14 Avem 6 feluri de prăjituri. Câte platouri diferite care să conțină 10 prăjituri se pot alcătui? (presupunând că avem cel puțin 10 prăjituri din fiecare fel).

Răspuns. $\binom{15}{5} = 3003$.

Exemplul 1.15 Să se arate că, dacă sunt aruncate trei zaruri identice, există 56 rezultate posibile. Câte rezultate există în cazul când sunt aruncate n asemenea zaruri?

Răspuns. Acest număr este $\binom{n+5}{5}$, în particular $\binom{8}{5} = 56$.

Exemplul 1.16 Presupunem că expresia $(x_1+x_2+x_3)^n$ este dezvoltată și că termenii asemenea sunt restrânși, folosind regulile algebrice cunoscute. Care este numărul de termeni din formula care rezultă?

Răspuns. Avem n de 1 (obiecte, corespunzând monoamelor x_i), și 2 markeri, separând pe x_1 de x_2 și respectiv pe x_2 de x_3 . Astfel, numărul de șiruri de biți de lungime $n+2$ conținând 2 zerouri și n unități va fi $\binom{n+2}{n}$.

1.5 Design-uri

Un fabricant a dezvoltat un produs nou și dorește să evalueze n varietăți ale lui cerând anumitori consumatori să testeze aceste varietăți. Este imposibil din punct de vedere practic ca fiecare consumator să testeze toate varietățile, deci ar fi rezonabil să se impună următoarele condiții:

- (i) Fiecare consumator ar trebui să testeze același număr k de varietăți;
- (ii) Fiecare varietate ar trebui să fie testată de același număr r de consumatori.

De exemplu, dacă $n = 8$, $k = 4$, $r = 3$, o schemă posibilă ar fi să cerem la 6 persoane să testeze varietățile, după cum urmează:

$$1234 \quad 5678 \quad 1357 \quad 2468 \quad 1247 \quad 3658,$$

unde numărul $1, 2, \dots, 8$ reprezintă varietățile.

Definiția 1.6 Fie X o n -mulțime. O mulțime \mathcal{B} de k -submulțimi ale lui X este un **design** cu parametrii (n, k, r) dacă fiecare membru al lui X aparține la exact r dintre submulțimile lui \mathcal{B} .

Un element $B \in \mathcal{B}$ se numește un **bloc** al design-ului.

Așa cum am văzut în exemplul 1.5, pag.10, nu există nici un design cu parametrii $(7, 5, 4)$, deci este necesar să găsim condiții asupra parametrilor (n, k, r) care să asigure existența unui design cu acești parametri.

Prin urmare, să considerăm o n -mulțime X și fie \mathcal{C} o mulțime de k -submulțimi ale lui X , nu neapărat un design. Ca și în secțiunea 1.3, construim un tablou (pentru cazul particular $n = 6, k = 3$). Un semn pe poziția (x, C_i) înseamnă $x \in C_i$. Pentru fiecare $x \in X$, fie $r(x)$ numărul de apariții ale lui x ca și element al unei submulțimilor C_i .

$x \setminus C$	C_1	C_2	C_3	C_4	$r(x)$
1	✓	✓		✓	3
2	✓		✓		2
3		✓			1
4	✓		✓	✓	3
5			✓		1
6		✓		✓	2
	3	3	3	3	

Dacă presupunem că fiecare C_i este o k -submulțime a lui X , atunci pe fiecare coloană totalul va fi k . Egalând totalul pe linii cu totalul pe coloane obținem

$$\sum_{x \in X} r(x) = |\mathcal{C}| \cdot k. \quad (1.11)$$

Să presupunem acum că avem un design B cu parametrii (n, k, r) . În acest caz în egalitatea (1.11) înlocuim $r(x) = r$, obținând

$$n \cdot r = k \cdot b, \quad (1.12)$$

unde $b = |\mathcal{B}|$ este numărul de blocuri.

Mai mult, numărul total de k -submulțimi ale lui X este $\binom{n}{k}$, deci numărul b de blocuri ar trebui să fie mai mic decât acest număr. În concluzie

$$b = \frac{nr}{k} \leq \binom{n}{k}. \quad (1.13)$$

Se poate demonstra că (1.12) și (1.13) sunt de asemenea condiții suficiente pentru existența unui design. Mai precis, are loc teorema 1.11. Pentru demonstrația implicației inverse și alte proprietăți ale designs-urilor, a se consulta [2].

Teorema 1.11 *Există un design cu parametrii (n, k, r) dacă și numai dacă*

$$k \mid nr \quad \text{și} \quad r \leq \binom{n-1}{k-1}.$$

1.6 Partiții și distribuții

În acest paragraf vom studia partiții ale unei mulțimi în submulțimi (partiții de mulțimi), partiții întregi, și distribuții ale unei mulțimi de obiecte într-o mulțime de cutii.

1.6.1 Partiții ale unei mulțimi în submulțimi (partiții de mulțimi)

Definiția 1.7 Fie $I \neq \emptyset$ o mulțime de indici, finită sau infinită. O **partiție** a unei mulțimi X este o familie $\mathcal{H} = \{X_i, i \in I\}$ de submulțimi nevide ale lui X , astfel încât

1. $X = \bigcup_{i \in I} X_i$,
2. X_i sunt disjuncte două câte două.

Cu alte cuvinte, fiecare element $x \in X$ trebuie să aparțină uneia și numai uneia din submulțimile X_i . Submulțimile X_i se mai numesc și **clasele** sau **părțile** partiției \mathcal{H} .

Observația 1.7 A defini o partiție a unei n -mulțimi X în k părți este echivalent cu a plasa n bile distincte în k cutii identice astfel încât să nu rămână goală nici o cutie (plasare surjectivă).

Exemplul 1.17 Familia $\mathcal{H} = \{X_1, X_2, X_3, X_4\}$, unde

$$X_1 = \{1, 2, 9\}, X_2 = \{3, 5, 7, 10\}, X_3 = \{4\}, X_4 = \{6, 8\}$$

este o partiție a lui \mathbb{N}_{10} .

Un alt exemplu de partiție este dat în Apendicele A, pag.137, unde am arătat și cum o relație de echivalență determină o partiție.

Desigur ne interesează să calculăm numărul de partiții. Un prim rezultat privind acest număr este următorul.

Teorema 1.12 Fie $S(n, k)$ numărul de partiții ale unei n -mulțimi X în k părți (numite și k -partiții), $1 \leq k \leq n$. Atunci

1. $S(n, 1) = 1$,
2. $S(n, n) = 1$,
3. $S(n, k) = S(n - 1, k - 1) + kS(n - 1, k)$, pentru $2 \leq k \leq n - 1$.

Demonstrație. Mai întâi se constată cu ușurință că $S(n, 1) = 1$, deoarece există doar o partiție a lui X într-o singură parte, și anume $\mathcal{H} = \{X\}$. La fel, $S(n, n) = 1$, întrucât din nou există doar o partiție a lui X în n clase, și anume $\mathcal{H} = \{\{x\}, x \in X\}$.

Rămâne de demonstrat 3. Pentru aceasta, să fixăm $z \in X$. Cu z fixat, distingem următoarele tipuri de partiții:

Tipul 1: partiții în care mulțimea $\{z\}$ este o parte,

Tipul 2: partiții în care partea care îl conține pe z are și alte elemente.

Să calculăm numărul de partiții de tipul 1. Dacă mulțimea $\{z\}$ este înlăturată din partiție, obținem o partiție a $(n - 1)$ -mulțimii $X \setminus \{z\}$ în $k - 1$ părți, și există $S(n - 1, k - 1)$ astfel de partiții. Reciproc, dată fiind o astfel de partiție, putem reconstitui partea $\{z\}$.

Să calculăm acum numărul de partiții de tipul 2. Presupunem că \mathcal{H} este o partiție de tipul 2 cu părțile X_1, X_2, \dots, X_k . Putem defini o pereche de obiecte (i, \mathcal{H}_0) , cu i luat astfel încât $z \in X_i$ și cu \mathcal{H}_0 luată ca și partiție a $(n - 1)$ -mulțimii $X \setminus \{z\}$ cu cele k părți X_1, \dots, X_{i-1} ,

$X_i \setminus \{z\}, X_{i+1}, \dots, X_k$. Există k valori posibile pentru i și $S(n-1, k)$ partiții posibile \mathcal{H}_0 , deci avem $kS(n-1, k)$ asemenea perechi. Reciproc, dată fiind o asemenea perche (i, \mathcal{H}_0) , putem reda z părții X_i și reconstitui \mathcal{H} .

Intrucât pentru un z fixat o partiție poate fi fie de tipul 1 fie de tipul 2, prin adunarea numerelor obținute pentru fiecare tip de partiție se obține concluzia 3. ■

Observația 1.8 Numerele $S(n, k)$ sunt cunoscute ca și numerele lui Stirling⁴ de speța a doua. Din nefericire nu există o formulă simplă pentru aceste numere, dar folosind recurența 3 putem construi următorul tabel:

$n \backslash k$	1	2	3	4	5	6	7	8	...
1	1								
2	1	1							
3	1	3	1						
4	1	7	6	1					
5	1	15	25	10	1				
6	1	31	90	65	15	1			
7	1	63	301	350	140	21	1		
8	1	127	966	1701	1050	266	28	1	
...									

Exemplul 1.18 Mulțimea N_5 are următoarele 2-partiții, unde am înlocuit \cup cu + și am omis acoladele și virgulele pentru a simplifica scrierea:

$$\begin{aligned} 12345 = & 1234 + 5, 1235 + 4, 1245 + 3, 1345 + 2, 2345 + 1, \\ & 124 + 35, 125 + 34, 134 + 25, 135 + 24, 145 + 23, \\ & 234 + 15, 235 + 14, 245 + 13, 345 + 12. \end{aligned}$$

Intr-adevăr, am obținut $S(5, 2) = 15$ partiții.

1.6.2 Distribuții, surjecții și numere multinomiale

Am văzut că o partiție este echivalentă cu o plasare surjectivă de bile distincte în cutii identice (neordonate). În cazul în care cutiile sunt ordonate, partiția se va numi **distribuție**. Astfel, a defini o distribuție înseamnă a plasa obiecte distincte în cutii distincte, astfel încât nici o cutie să nu rămână goală, sau, echivalent, a defini o surjecție de la mulțimea bilelor la mulțimea cutiilor.

În exemplul 1.17, pag. 18 avem plasarea

$$\text{cutia 1: } 1, 2, 9; \quad \text{cutia 2: } 3, 5, 7, 10; \quad \text{cutia 3: } 4; \quad \text{cutia 4: } 6, 8,$$

care definește surjecția $\mathcal{D} : X \rightarrow \mathbb{N}_4$ dată de

x	1	2	3	4	5	6	7	8	9	10
$\mathcal{D}(x)$	1	1	2	3	2	4	2	4	1	2

În concluzie, problema numărării distribuțiilor este echivalentă cu problema numărării surjecțiilor. În acest moment putem demonstra următorul rezultat.

⁴James Stirling (1692-1770), matematician scoțian.

Teorema 1.13 Fie \mathcal{S} mulțimea surjecțiilor de la o n -mulțime X la o k -mulțime Y . Atunci

$$|\mathcal{S}| = k! \cdot S(n, k). \quad (1.14)$$

Demonstrație. Fiecare surjecție $\mathcal{D} : X \rightarrow Y$ induce o partiție a lui X în k părți. Dacă o asemenea partiție este dată, există $k!$ surjecții care o pot genera, întrucât cele k părți se pot atribui celor k elemente ale lui Y un orice mod bijectiv. În concluzie, are loc formula (1.14). ■

Exemplul 1.19 Să se calculeze numărul de moduri în care se pot împărți cărțile la jocul de Bridge.

Re-formulare. Avem 52 obiecte (cărți de joc) și patru cutii (cei patru jucători), și trebuie să distribuim câte 13 cărți fiecărui jucător. În acest joc contează ce cărți primește fiecare jucător. Astfel, nu vom cere determinarea numărului de partiții ale 52-mulțimii X în patru 13-submulțimi, ci numărul de funcții surjective de la 52-mulțimea X la 4-mulțimea $Y = \{N, E, S, W\}$ cu proprietatea că fiecare «cutie» a lui Y primește 13 obiecte din X .

Mai general, putem cere numărul de surjecții de la o n -mulțime la o k -mulțime de cutii $\{y_1, y_2, \dots, y_k\}$ cu proprietatea că

$$\begin{aligned} n_1 \text{ obiecte sunt introduse în prima cutie } y_1, \\ n_2 \text{ obiecte sunt introduse în a doua cutie } y_2, \\ \dots \\ n_k \text{ obiecte sunt introduse în a } k\text{-a cutie } y_k. \end{aligned}$$

Acest număr este notat cu

$$\binom{n}{n_1, n_2, \dots, n_k} \quad (1.15)$$

și este numit **număr multinomial**. În continuare vom da o formulă pentru acest număr.

Teorema 1.14 Date fiind numerele întregi pozitive (> 0) n, n_1, n_2, \dots, n_k satisfăcând condiția $n_1 + n_2 + \dots + n_k = n$, avem

$$\binom{n}{n_1, n_2, \dots, n_k} = \frac{n!}{n_1! \cdot n_2! \cdot \dots \cdot n_k!}. \quad (1.16)$$

Demonstrație. Mai întâi aranjăm obiectele în $X = \{x_1, x_2, \dots, x_n\}$ astfel:

$$x_{\pi(1)}, x_{\pi(2)}, \dots, x_{\pi(n)},$$

unde π este o permutare a lui \mathbb{N}_n . Apoi definim o surjecție de la X la Y prin introducerea primelor n_1 obiecte din listă în prima cutie (y_1), a următoarelor n_2 obiecte în a doua cutie (y_2), \dots , a ultimelor n_k obiecte în ultima cutie (y_k). Se obține aceeași surjecție dacă se permută primele n_1 obiecte între ele în orice mod. Acest lucru poate fi făcut în $n_1!$ moduri. Astfel, din cele $n!$ permutări π , există $n_1! \cdot n_2! \cdot \dots \cdot n_k!$ care vor induce o surjecție dată. În concluzie, numărul de surjecții proprietatea specificată este într-adevăr cel din formula (1.16). ■

Observația 1.9 Numărul multinomial (1.15) mai apare și sub terminologia numărul de **permutări cu repetiții**, a $n = \sum_{i=1}^k n_i$ simboluri de k tipuri distincte, unde n_i sunt de tipul i , $i = 1, 2, \dots, k$.

Observația 1.10 Teorema 1.14 rămâne adevărată și în cazul când una sau mai multe din numerele n_i sunt zero. În acest caz, simbolul din (1.15) este definit ca și numărul de funcții de la o n -mulțime la o mulțime de k «cutii», cu proprietatea că n_i obiecte sunt plasate în cutia i , pentru $i = 1, \dots, k$. În acest caz funcțiile nu sunt în mod necesar surjecții, întrucât admitem ca anumite numere n_i să fie zero. Cu convenția $0! = 1$, formula (1.16) rămâne adevărată.

Exemplul 1.20 Câte cuvinte de 10 litere se pot forma din literele cuvântului MATEMATICA?

Soluție. Fiecare cuvânt corespunde unei surjecții de la mulțimea $\{x_1, x_2, \dots, x_{10}\}$ la mulțimea de 6 «cutii» $\{A, C, E, I, M, T\}$, astfel încât trei obiecte merg în «cutia» A, un obiect în «cutia» C, un obiect în «cutia» E, un obiect în «cutia» I, două obiecte în «cutia» M și două obiecte în «cutia» T. Numărul total de cuvinte va fi

$$\binom{10}{3, 1, 1, 1, 2, 2} = \frac{10!}{3!1!1!1!2!2!} = 151200.$$

Să observăm că în cazul particular $k = 2$ numărul multinomial coincide cu numărul binomial. Din acest motiv este de așteptat să existe o generalizare a teoremei binomiale (a binomului lui Newton). Această generalizare este cunoscută ca și *teorema multinomială*.

Teorema 1.15 (Teorema multinomială) Fie n, k numere întregi pozitive (> 0). Atunci

$$(x_1 + x_2 + \dots + x_k)^n = \sum \binom{n}{n_1, n_2, \dots, n_k} x_1^{n_1} x_2^{n_2} \dots x_k^{n_k}, \quad (1.17)$$

unde suma este luată după toate k -uplele de numere întregi nenegative (≥ 0) (n_1, n_2, \dots, n_k) astfel încât $n_1 + n_2 + \dots + n_k = n$.

Demonstrație. Atunci când se calculează membrul stâng, termenul $x_1^{n_1} x_2^{n_2} \dots x_k^{n_k}$ apare prin alegerea de n_i ori a factorului x_i , $i = 1, 2, \dots, k$. Cu alte cuvinte, termenul $x_1^{n_1} x_2^{n_2} \dots x_k^{n_k}$ corespunde unei funcții de la mulțimea de n factori (monoame) la mulțimea $\{x_1, x_2, \dots, x_k\}$, cu proprietatea că n_i factori merg în x_i , pentru $i = 1, 2, \dots, k$. Există

$$\binom{n}{n_1, n_2, \dots, n_k}$$

asemenea funcții, și prin urmare acesta este numărul de termeni $x_1^{n_1} x_2^{n_2} \dots x_k^{n_k}$ din produs. ■

1.6.3 Partiții întregi

Definiția 1.8 Fie $n \in \mathbb{N}$. Suma $n = n_1 + n_2 + \dots + n_k$, cu $n_i \in \mathbb{N}$, se va numi o **k -partiție (întreagă)** a numărului n . Numărul de k -partiții ale lui n va fi notat cu $P(n, k)$.

Observația 1.11 A defini o k -partiție întreagă a lui n este echivalent cu a introduce n bile identice în k cutii identice, astfel încât nici o cutie să nu rămână goală (plasare surjectivă).

Pentru numărul $P(n, k)$ nu există nici o formulă de recurență acceptabilă, similară celei din teorema 1.12 pentru numerele lui Stirling $S(n, k)$, care să ne permită calculul imediat al numerelor $P(n, k)$.

Exemplul 1.21 4-partițiile lui 8 sunt:

$$5 + 1 + 1 + 1, 4 + 2 + 1 + 1, 3 + 3 + 1 + 1, 3 + 2 + 2 + 1, 2 + 2 + 2 + 2.$$

Se pot considera de asemenea k -partiții întregi **ordonate**, dacă, de exemplu, se consideră că $3 + 3 + 1 + 1 \neq 3 + 1 + 3 + 1$. Astfel, construcția unei k -partiții întregi ordonate a lui n este echivalentă cu plasarea a n bile identice în k cutii distincte, astfel încât nici o cutie să nu rămână goală.

Problema numărării partițiilor întregi ordonate nu poate fi rezolvată în mod similar cu cazul partițiilor ordonate de mulțimi, unde s-a înmulțit numărul $S(n, k)$ de partiții neordonate de mulțimi cu numărul $k!$ în care părțile se pot permuta între ele (a se vedea teorema 1.13). Într-adevăr, pentru $3 + 1 + 1$ nu există $6 = 3!$ partiții ordonate distincte, cu numai trei: $3 + 1 + 1, 1 + 3 + 1, 1 + 1 + 3$. Prin urmare, trebuie folosită o altă abordare.

Teorema 1.16 Numărul k -partițiilor întregi ordonate ale lui n este $\binom{n-1}{k-1}$.

Demonstrație. Observăm că fiecare k -partiție ordonată a lui n este echivalentă cu introducerea a n bile identice în k cutii distincte, astfel încât nici o cutie să nu rămână goală. Deci, începem prin a plasa câte o bilă în fiecare cutie. Astfel, rămân de plasat arbitrar cele $n - k$ bile rămase în k cutii. Șirul de biți asociat (a se vedea secțiunea 1.4.5, pag. 15) va conține $n - k$ unități și $k - 1$ zerouri. Numărul de șiruri de biți de lungime $n - k + k - 1 = n - 1$, cu exact $k - 1$ zerouri, este într-adevăr $\binom{n-1}{k-1}$. ■

Exemplul 1.22 3-partițiile ordonate ale lui 6 sunt:

$$4 + 1 + 1, 1 + 4 + 1, 1 + 1 + 4, 3 + 2 + 1, 3 + 1 + 2, \\ 2 + 3 + 1, 2 + 1 + 3, 1 + 3 + 2, 1 + 2 + 3, 2 + 2 + 2.$$

Avem într-adevăr $\binom{5}{2} = 10$ partiții ordonate.

1.7 Rezolvarea problemelor de numărare folosind relații de recurență

Exemplul 1.23 (Problema turnurilor din Hanoi) Avem trei suporturi numerotate 1, 2, 3 și pe unul din suporturi avem așezate n discuri, fiecare mai mic în diametru decât cel situat sub el. O mișcare permisă constă în a muta un disc de pe un suport pe altul, astfel încât un disc să nu fie așezat peste un disc cu suport mai mic. Câte mutări permise sunt necesare pentru a muta cele trei discuri de pe un suport pe altul?

Soluție. Pentru a rezolva problema mutării tuturor discurilor de pe suportul 1 pe suportul 2, executăm următoarele:

1. (recursiv) Rezolvăm problema mutării a $n - 1$ discuri de pe suportul 1 pe suportul 3;
2. mutăm discul n pe suportul 2;
3. (recursiv) Rezolvăm problema mutării a $n - 1$ discuri de pe suportul 3 pe suportul 2.

Fie a_n numărul de mutări necesare pentru a muta n discuri de pe un suport pe altul. Obținem următoarea relație de recurență

$$a_n = 2a_{n-1} + 1, \quad a_1 = 1.$$

Avem, $a_2 = 3$, $a_3 = 7$, deci se poate ghici cu ușurință că $a_n = 2^n - 1$, fapt care se poate demonstra imediat prin inducție.

Ce se întâmplă însă dacă se obține o relație de recurență unde este imposibil să ghicim termenul general? Pentru aceste situații avem nevoie de teorii mai sofisticate.

1.8 Relații de recurență și funcții generatoare

1.8.1 Recurențe liniare

Fie $k \in \mathbb{N}$. Un tip simplu de recurență este

$$u_{n+k} = \alpha_1 u_{n+k-1} + \alpha_2 u_{n+k-2} + \dots + \alpha_k u_n, \quad n \geq 0. \quad (1.18)$$

cu α_i , $i = 1, 2, \dots, k$, numere reale date astfel încât $\alpha_k \neq 0$. Aceasta este numită **recurență liniară** de ordin k și o formulă explicită pentru termenul general u_n se poate găsi ori de câte ori u_0, u_1, \dots, u_{k-1} sunt dați. Pentru a dovedi acest lucru e nevoie să demonstrăm mai întâi următoarea leamă.

Lema 1.17 *Mulțimea de șiruri care satisfac relația de recurență (1.18) este un spațiu vectorial de dimensiune k .*

Demonstrație. Dacă $(a_n)_n$ și $(b_n)_n$ satisfac (1.18), atunci un calcul simplu arată că $\alpha \cdot (a_n)_n$ și $(a_n + b_n)_n$ satisfac (1.18), deci avem un spațiu liniar. Pentru a găsi dimensiunea lui, e nevoie să găsim o bază formată din k șiruri.

Ideea este să asociem *ecuația caracteristică* $R(x) = 0$, unde

$$R(x) = x^k - \alpha_1 x^{k-1} - \dots - \alpha_{k-1} x - \alpha_k. \quad (1.19)$$

Sunt posibile următoarele situații privind rădăcinile lui R :

1. Dacă r_1 este o rădăcină a lui R , atunci șirul $(u_n)_n$, unde $u_n = r_1^n$, satisface (1.18). Intr-adevăr, avem $r_1^k = \alpha_1 r_1^{k-1} + \dots + \alpha_{k-1} r_1 + \alpha_k$ și, după înmulțirea acestei egalități cu r_1^n , rezultă concluzia.
2. Dacă r_1 este o rădăcină dublă nenulă a lui R , atunci șirul $(b_n)_n$, unde $b_n = n r_1^n$, satisface (1.18). Intr-adevăr, din $R(r_1) = R'(r_1) = 0$ obținem

$$\begin{aligned} k r_1^{n+k} &= \alpha_1 (k-1) r_1^{n+k-1} + \dots + \alpha_{k-1} r_1^{n+1}, \\ r_1^{n+k} &= \alpha_1 r_1^{n+k-1} + \dots + \alpha_{k-1} r_1^{n+1} + \alpha_k r_1^n. \end{aligned}$$

Dacă se înmulțește a doua egalitate cu n și se adună aceste două egalități, se obține concluzia.

3. Dacă r_1 este o rădăcină cu multiplicitatea p pentru R , atunci şirurile $(u_n)_n$ cu

$$u_n \in \{r_1^n, n r_1^n, n^2 r_1^n, \dots, n^{p-1} r_1^n\}$$

sunt soluţii pentru (1.18). Pentru a arăta acest lucru se folosesc aceleaşi argumente ca şi în cazul anterior.

4. Dacă $r_{1,2} = \alpha \pm i\beta = e^{a \pm bi}$ sunt rădăcini complexe ale lui R , atunci putem scrie

$$r_1 = e^a e^{bi} = e^a (\cos b + i \sin b),$$

şi prin urmare $r_{1,2}^n = e^{an} (\cos bn \pm i \sin bn)$. În acest caz şirurile $(c_n)_n$ şi $(d_n)_n$, date de

$$\begin{aligned} c_n &= \frac{1}{2}(r_1^n + r_2^n) = e^{an} \cos bn, \\ d_n &= \frac{1}{2i}(r_1^n - r_2^n) = e^{an} \sin bn, \end{aligned}$$

sunt soluţii reale pentru (1.18).

5. Dacă $r_{1,2} = \alpha \pm \beta bi$ sunt rădăcini complexe pentru R cu multiplicitatea p , atunci în acelaşi mod putem arăta că

$$c_n, d_n, n c_n, n d_n, n^2 c_n, n^2 d_n, \dots, n^{p-1} c_n, n^{p-1} d_n$$

satisfac relaţia de recurenţă (1.18).

În concluzie, celor k rădăcini ale lui R li se pot asocia soluţii liniar independente ale recurenţei. Soluţia generală $(Y_n)_n$ a recurenţei (1.18) va fi o combinaţie liniară a acestor soluţii. Coeficienţii combinaţiei liniare vor fi determinaţi în mod unic din condiţiile «iniţiale» $Y_0 = u_0, \dots, Y_{k-1} = u_{k-1}$. De exemplu, în cazul când R sunt rădăcini reale şi distincte, soluţia generală este $(Y_n)_n$, cu $Y_n = \beta_1 r_1^n + \beta_2 r_2^n + \dots + \beta_k r_k^n$. Constantele reale β_i vor fi determinate în mod unic din condiţiile $Y_i = u_i$, $i = 1, \dots, k$, care se pot scrie astfel:

$$\begin{aligned} \beta_1 + \beta_2 + \dots + \beta_k &= u_0, \\ \beta_1 r_1 + \beta_2 r_2 + \dots + \beta_k r_k &= u_1, \\ \vdots \\ \beta_1 r_1^{n-1} + \beta_2 r_2^{n-1} + \dots + \beta_k r_k^{n-1} &= u_{k-1}. \end{aligned}$$

Necunoscutele β_1, \dots, β_k sunt unic determinate deoarece determinantul sistemului este nenul (fiind un determinant Vandermonde asociat rădăcinilor distincte r_i). Se poate demonstra că acest determinant este nenul şi în restul cazurilor. ■

1.8.2 Funcţii generatoare

Aşa cum s-a văzut deja în exemplul 1.23, soluţia unei probleme de numărare (şi nu numai) poate fi exprimată ca şi un şir $(u_n)_n$. În asemenea cazuri, un mod uzual de a rezolva acest tip de probleme se bazează pe reprezentarea lui $(u_n)_n$ ca şi o serie de puteri

$$g_u(x) = \sum_{n=0}^{\infty} u_n x^n. \quad (1.20)$$

În această situație, g_u se folosește sub denumirea de **funcție generatoare**⁵ pentru șirul $(u_n)_n$. (Este desigur necesar ca seria să convergă dacă g_u este definită ca și funcție de x . Dacă g_u este privită ca și un element al unui inel de polinoame, o asemenea convergență nu este necesară). În unele cazuri e posibil ca seria (1.20) să se reducă la sumă finită, și atunci g_u este un polinom.

Exemplul 1.24 Fie $m \in \mathbb{N}$ fixat și fie u_n numărul de n -submulțimi ale unei m -mulțimi. Se știe (a se vedea secțiunea 1.4.4, pag. 14) că

$$u_n = \binom{m}{n},$$

de unde funcția generatoare va fi

$$g_u(x) = \sum_{n=0}^m \binom{m}{n} x^n = (1+x)^m.$$

În cazul când formula de recurență este dată pentru $(u_n)_n$, funcția generatoare poate fi utilă pentru determinarea unei expresii generale pentru u_n . Metoda are trei pași:

- (i) din relația de recurență deducem o ecuație pentru $g_u(x)$,
- (ii) se rezolvă ecuația pentru $g_u(x)$,
- (iii) se găsește o formulă pentru coeficienții u_n ai lui $g_u(x)$ (prin descompunere în fracții simple sau prin teorema binomială) în felul următor: Dacă se pot determina g_u și seria sa de puteri, pentru a găsi u_n vom identifica coeficienții seriei de puteri.

Observația 1.12 Pentru recurența liniară și omogenă studiată în paragraful precedent, se pot utiliza de asemenea funcții generatoare, dar această metodă necesită mai multe calcule decât folosirea polinomului caracteristic.

1.8.3 Recurențe liniare neomogene

În unele cazuri metoda funcțiilor generatoare poate fi folosită pentru rezolvarea recurențelor liniare și neomogene. O asemenea recurență este

$$u_{n+2} = au_{n+1} + bu_n + f(n), \quad u_0, u_1 \text{ dați.} \quad (1.21)$$

Aplicabilitatea metodei depinde de forma particulară a funcției f . Fie

$$g_u(x) = \sum_{n=0}^{\infty} u_n x^n.$$

Dacă se înmulțește această identitate cu $(a + bx)$ se obține

$$\begin{aligned} (a + bx)g_u(x) &= au_0 + (au_1 + bu_0)x + (au_2 + bu_1)x^2 + \dots \\ &\quad + (au_n + bu_{n-1})x^n + \dots \end{aligned}$$

⁵Funcțiile generatoare au fost introduse de către De Moivre și Euler la începutul secolului XVIII.

Din (1.21) deducem că

$$\begin{aligned} au_1 + bu_0 &= u_2 - f(0), \\ au_2 + bu_1 &= u_3 - f(1), \\ &\vdots \\ au_n + bu_{n-1} &= u_{n+1} - f(n-1). \end{aligned}$$

Înlocuind această expresie se obține, după gruparea termenilor și după înmulțirea egalității cu x ,

$$x(a + bx)g_u(x) = au_0x + \sum_{n=1}^{\infty} u_{n+1}x^{n+1} - \sum_{n=1}^{\infty} f(n-1)x^{n+1}. \quad (1.22)$$

În acest moment e nevoie să determinăm suma seriei de puteri

$$\sum_{n=1}^{\infty} f(n-1)x^{n+1} = h(x).$$

Dacă nu putem găsi această sumă h , metoda eșuează.

Presupunem că am găsit h . Ne concentrăm pe (1.22) și înlocuim prima sumă din membrul drept cu $g_u(x) - u_0 - u_1x$. Astfel, (1.22) devine

$$g_u(x) = \frac{-u_0 + (au_0 - u_1)x - h(x)}{bx^2 + ax - 1}. \quad (1.23)$$

Dezvoltând această funcție în serie de puteri și identificând coeficientul lui x^n , vom obține o expresie generală pentru u_n .

Exemplul 1.25 *Să se găsească termenul general u_n al recurenței*

$$u_{n+2} = -u_{n+1} + 2u_n + n, \text{ cu } u_0 = -2, u_1 = 1.$$

Soluție. Din calculele de mai sus, efectuate pentru cazul general, găsim

$$h(x) = \sum_{n=1}^{\infty} (n-1)x^{n+1} = x^3 \sum_{n=2}^{\infty} (n-1)x^{n-2} = \frac{x^3}{(1-x)^2},$$

Apoi, din (1.23),

$$g_u(x) = \frac{-2 + 3x}{(1-x)^3(2x+1)} = \frac{A}{(1-x)^3} + \frac{B}{(1-x)^2} + \frac{C}{1-x} + \frac{D}{2x+1},$$

cu $A = \frac{1}{3}, B = -\frac{7}{9}, C = -\frac{14}{27}, D = -\frac{28}{27}$. Mai departe, scriem

$$\begin{aligned} \frac{1}{1-x} &= \sum_{n=0}^{\infty} x^n \quad | \cdot C \\ \frac{1}{(1-x)^2} &= \sum_{n=0}^{\infty} (n+1)x^n \quad | \cdot B \\ \frac{1}{(1-x)^3} &= \sum_{n=0}^{\infty} (n+1)(n+2)x^n \quad | \cdot A \\ \frac{1}{2x+1} &= \sum_{n=0}^{\infty} (-1)^n (2x)^n \quad | \cdot D \end{aligned}$$

Identificând coeficientul lui x^n se obține

$$\begin{aligned} u_n &= \frac{A}{2}(n+1)(n+2) + B(n+1) + C + D(-1)^n 2^n \\ &= \frac{1}{54} (9n^2 - 15n - 52 - 7(-1)^n 2^{n+3}). \end{aligned}$$

Observația 1.13 *Intr-un caz mai general, când recurența liniară și neomogenă este*

$$u_{n+k} = a_1 u_{n+k-1} + a_2 u_{n+k-2} + \dots + a_k u_n + g(n),$$

cu u_0, u_1, \dots, u_{k-1} dați, începem cu calculul lui

$$(1 + a_1 x + a_2 x^2 + \dots + a_k x^{k-1}) g_u(x)$$

și sperăm din nou că termenii care conțin $g(n)$ pot fi manipulați în mod convenabil.

1.8.4 Recurența lui Catalan

Dacă vrem să calculăm numărul u_n de arbori binari cu n vârfuri (a se vedea paragraful 3.4.1), ajungem la următoarea relație de recurență cunoscută sub numele de **recurența lui Catalan**⁶.

$$u_{n+1} = u_0 u_n + u_1 u_{n-1} + \dots + u_k u_{n-k} + \dots + u_n u_0.$$

cu $u_0 = u_1 = 1$. Numărul u_n se numește **al n -lea număr al lui Catalan**. Să observăm că numărul de termeni din această recurență nu este constant, ca și în cazurile precedente.

Considerăm funcția generatoare

$$g_u(x) = \sum_{n=0}^{\infty} u_n x^n.$$

Avem

$$\begin{aligned} g_u^2(x) &= (u_0 + u_1 x + u_2 x^2 + \dots)(u_0 + u_1 x + u_2 x^2 + \dots) \\ &= u_0 + 2u_0 u_1 x + (u_0 u_2 + u_1^2 + u_2 u_0) x^2 + \dots \\ &\quad + (u_0 u_n + u_1 u_{n-1} + \dots + u_n u_0) x^n + \dots \\ &= u_0 + u_2 x + u_3 x^2 + \dots + u_{n+1} x^n + \dots, \end{aligned}$$

de unde deducem că

$$x g_u^2(x) = x + \sum_{n=2}^{\infty} u_n x^n,$$

și mai departe

$$x g_u^2(x) - g_u(x) + 1 = 0.$$

Rezolvând această ecuație obținem

$$g_{u1}(x) = \frac{1 + \sqrt{1 - 4x}}{2x}, \quad g_{u2}(x) = \frac{1 - \sqrt{1 - 4x}}{2x}.$$

⁶Eugène Charles Catalan (1814-1894), matematician belgian.

Deoarece $g_u(0) = u_0 = 1$, singura soluție va fi

$$g_u(x) = \frac{1 - \sqrt{1 - 4x}}{2x}.$$

Dezvoltăm acum funcția g_u folosind formula

$$(1 - y)^\alpha = 1 + \frac{\alpha}{1!}y + \frac{\alpha(\alpha - 1)}{2!}y^2 + \dots + \frac{\alpha(\alpha - 1) \dots (\alpha - n + 1)}{n!}y^n + \dots$$

pentru $\alpha = 1/2$ și $y = -4x$.

Coeficientul lui x^n va fi

$$u_n = \frac{1 \cdot 3 \cdot \dots \cdot (2n - 1)}{2^{n+2}(n + 1)!} 4^{n+1} = \frac{1}{n + 1} \binom{2n}{n}.$$

1.8.5 Câteva proprietăți ale funcțiilor generatoare

Teorema 1.18 (Convoluția) Dacă u_n se poate scrie ca și o convoluție de șiruri $(a_n)_n$ și $(b_n)_n$, adică

$$u_n = \sum_{k=0}^n a_k b_{n-k},$$

atunci

$$g_u(x) = g_a(x)g_b(x).$$

Demonstrație. Dacă evaluăm $g_a g_b$,

$$g_a(x)g_b(x) = (a_0 + a_1x + \dots + a_nx^n + \dots)(b_0 + b_1x + \dots + b_nx^n + \dots),$$

coeficientul lui x^n va fi într-adevăr

$$a_0b_n + a_1b_{n-1} + \dots + a_nb_0 = \sum_{k=0}^n a_kb_{n-k}.$$

■

Teorema 1.19 Dacă

$$b_n = \sum_{k=0}^n a_k,$$

atunci

$$g_b(x) = \frac{g_a(x)}{1 - x}.$$

Demonstrație. Folosind seria geometrică obținem

$$\begin{aligned} \frac{g_a(x)}{1 - x} &= (a_0 + a_1x + \dots + a_nx^n + \dots)(1 + x + x^2 + \dots + x^n + \dots) \\ &= a_0 + (a_0 + a_1)x + (a_0 + a_1 + a_2)x^2 + \dots \\ &\quad + (a_0 + a_1 + \dots + a_n)x^n + \dots \end{aligned}$$

deci coeficientul lui x^n va fi într-adevăr $b_n = \sum_{k=0}^n a_k$.

■

Teorema 1.20 *Dacă*

$$b_n = \sum_{k=1}^{\infty} a_{n+k}, \quad n \geq 0,$$

atunci

$$g_b(x) = \frac{g_a(1) - g_a(x)}{1 - x}.$$

Demonstrație. Fie $h_a(x) = g_a(1) - g_a(x)$. Seria sa de puteri se poate scrie $\sum_{n=0}^{\infty} c_n x^n$, cu

$$\begin{aligned} c_0 &= g(1) - a_0 = a_1 + a_2 + \dots, \\ c_n &= -a_n, \text{ pentru } n \geq 1. \end{aligned}$$

Conform teoremei 1.19 avem

$$\frac{h_a(x)}{1-x} = \sum_{n=0}^{\infty} b_n x^n, \text{ cu } b_n = \sum_{k=0}^n c_k = c_0 + \sum_{k=1}^n (-a_k) = \sum_{k=1}^{\infty} a_{n+k}.$$

■

1.9 Probleme rezolvate

Problema 1.1 (*Bile identice în cutii distincte cu un număr arbitrar de bile în cutii*) Numărul de moduri în care k bile identice pot fi introduse în n cutii distincte, cu un număr arbitrar de bile în fiecare cutie este $\binom{n+k-1}{k}$.

Soluție. Numărul cerut poate fi găsit prin aplicarea directă a teoremei 1.10, pag. 15.

Problema 1.2 *Care este numărul de soluții întregi nenegative ale ecuației*

$$x_1 + x_2 + \dots + x_6 = 10?$$

Soluție. A se vedea exemplul 1.14, pag. 16 sau problema 1.1. Numărul cerut va fi din nou $\binom{15}{5}$.

Problema 1.3 (*Bile identice în cutii distincte cu nici o cutie goală*) Numărul de moduri în care k bile identice pot fi introduse în n cutii distincte, cu un număr arbitrar de bile în fiecare cutie și nici o cutie goală, este $\binom{k-1}{n-1}$.

Soluție. Folosim ideea din demonstrația teoremei 1.10, pag. 15. Introducem o bilă în fiecare cutie, întrucât nu admitem cutii goale. Astfel, rămâne să introducem cele $k - n$ bile rămase în n cutii, în același mod în care s-a făcut în problema precedentă. Acest număr este prin urmare

$$\binom{n-1+k-n}{n-1} = \binom{k-1}{n-1}.$$

Problema 1.4 *Care este numărul de soluții întregi pozitive (> 0) ale ecuației*

$$x_1 + x_2 + \dots + x_6 = 10?$$

Soluție. Conform problemei 1.3, numărul cerut este $\binom{9}{5}$.

Problema 1.5 Care este numărul de soluții întregi nenegative ale ecuației

$$x_1 + x_2 + x_3 + x_4 + x_5 = 36,$$

unde $x_1 \geq 4$, $x_4 \geq 7$.

Soluție. Problema poate fi reformulată astfel: Care este numărul de moduri în care 36 bile identice pot fi introduse în 5 cutii distincte, cu cel puțin 4 bile în prima cutie și cel puțin 7 bile în a patra cutie?

Incepem prin a introduce 4 bile în prima cutie și 7 bile în cutia a patra. Rămân 25 bile care trebuie introduse arbitrar în 5 cutii, și acest lucru se poate face în $\binom{29}{4}$ moduri (a se vedea problema 1.1).

Problema 1.6 (Problema deranjamentelor) O secretară ineficientă trebuie să introducă n scrisori în n plicuri adresate. În câte moduri poate realiza faptul de a introduce fiecare scrisoare într-un plic greșit?

Soluție. Trebuie să calculăm numărul de deranjamente ale permutării Π_n (definiția 1.5, pag. 14). Fie $\pi : \mathbb{N}_n \rightarrow \mathbb{N}_n$ permutarea definită prin

$$\pi(i) = j, \quad \text{dacă scrisoarea } i \text{ este introdusă în plicul } j.$$

Conform corolarului 1.3, pag. 9, numărul de deranjamente va fi

$$d_n = n! - \alpha_1 + \alpha_2 - \alpha_3 \cdots + (-1)^n \alpha_n,$$

unde α_k este numărul de permutări care fixează k poziții. Spunem că o permutare π fixează poziția j dacă $\pi(j) = j$.

Putem alege k poziții în $\binom{n}{k}$ moduri. Pentru fiecare fixare a k poziții, există $(n-k)!$ permutări ale pozițiilor rămase, prin urmare

$$\alpha_k = \binom{n}{k} (n-k)! = \frac{n!}{k!}.$$

Astfel

$$d_n = n! \left(1 - \frac{1}{1!} + \frac{1}{2!} - \frac{1}{3!} + \cdots + (-1)^n \frac{1}{n!} \right).$$

Problema 1.7 (Poker) În jocul de poker există 52 cărți de joc și un jucător primește 5 cărți. Să se găsească numărul de cazuri posibile pentru fiecare din următoarele mâini:

- Chintă regală ($A, K, Q, J, 10$ de aceeași culoare);
- Chintă de culoare (5 cărți consecutive de aceeași culoare, dar nu chintă regală);
- Careu (oricare patru cărți de același rang și a cincea carte arbitrară);
- Full house (3 cărți de același rang și încă o pereche de alt rang);
- Culoare (oricare 5 cărți de aceeași culoare, dar nu chintă regală sau chintă de culoare);
- Chintă (oricare 5 cărți consecutive, dar nu toate de aceeași culoare);
- Trei de același fel (3 cărți de același rang și 2 cărți de 2 ranguri diferite);
- Două perechi (câte 2 cărți de 2 ranguri diferite și a cincea carte de alt rang);
- O pereche (2 cărți de un rang, plus 3 cărți de 3 alte ranguri diferite).

Soluție.

a. 4 (există 4 alegeri pentru o culoare ($\clubsuit, \diamondsuit, \heartsuit, \spadesuit$) și câte o chintă regală pentru fiecare culoare).

b. $4 \cdot 9 = 36$ (4 alegeri pentru o culoare, pentru fiecare culoare 9 moduri de a obține 5 cărți în ordine: $(A, 2, 3, 4, 5), (2, 3, 4, 5, 6), (3, 4, 5, 6, 7), \dots, (9, 10, J, Q, K)$).

c. $13 \cdot 48 = 624$ (13 alegeri pentru un rang, un mod de a selecta 4 cărți din acest rang, 48 moduri de a selecta a cincea carte).

d. $13 \cdot \binom{4}{3} \cdot 12 \cdot \binom{4}{2} = 3744$ (13 moduri de a selecta un rang pentru cele 3 cărți, $\binom{4}{3}$ moduri de a alege 3 cărți din acest rang, 12 moduri de a selecta un rang pentru pereche și $\binom{4}{2}$ moduri de a obține o pereche din acest rang).

e. $4\binom{13}{5} - 40 = 5108$ (4 moduri de a selecta o culoare, $\binom{13}{5}$ moduri de a alege 5 cărți dintr-o culoare; scădem chinta regală și chinta de culoare).

f. $10 \cdot 4^5 - 40 = 10200$ (10 moduri de a alege 5 ranguri consecutive și 4 moduri de a alege o carte din fiecare rang; scădem chinta regală și chinta de culoare).

g. $13\binom{4}{3}\binom{12}{2} \cdot 4^2 = 54912$ (13 moduri de a selecta un rang, $\binom{4}{3}$ moduri de a alege 3 cărți dintr-un rang, $\binom{12}{2}$ moduri de a alege celelalte ranguri și 4^2 moduri de a alege o carte din fiecare din aceste 2 ranguri).

h. $\binom{13}{2}\binom{4}{2}\binom{4}{2} \cdot 44 = 123\,552$ ($\binom{13}{2}$ moduri de a selecta 2 ranguri, $\binom{4}{2}$ moduri de a alege 2 cărți din fiecare din aceste ranguri, 44 moduri de a alege a cincea carte).

i. $13\binom{4}{2}\binom{12}{3} \cdot 4^3 = 1\,098\,240$ (13 moduri de a alege un rang, $\binom{4}{2}$ moduri de a alege 2 cărți din acest rang, $\binom{12}{3}$ moduri de a alege câte o carte din fiecare din aceste ranguri).

Să menționăm că există $\binom{52}{5} = 2\,598\,960$ moduri de a alege 5 cărți. Numărul de mâini care nu aparțin nici uneia din situațiile menționate anterior va fi $1\,302\,540$, care este puțin mai mare decât $\binom{52}{5}/2 = 1\,299\,480$.

Capitolul 2

Probabilități discrete

2.1 Evenimente aleatoare

Multe procese din natură, societate, economie, medicină etc. depind de șansă, adică rezultatul unui proces nu poate fi prevăzut cu exactitate. Cursul de schimb valutar dintr-o anumită zi din viitor, presiunea arterială a unei persoane la un moment dat, sunt doar două exemple. Asemenea evenimente aleatoare nu pot fi prevăzute cu exactitate, dar frecvența relativă cu care apar într-o serie lungă de repetări ale experimentului este de obicei stabilă. Evenimentele care au această proprietate se numesc *evenimente aleatoare* sau *stochastice*. Teoria probabilităților ne furnizează modele matematice pentru asemenea fenomene aleatoare.

Un **experiment aleator** este un proces în care mai multe rezultate sunt posibile, astfel că nu putem spune dinainte care va fi rezultatul experimentului.

Spunem că două rezultate sunt **incompatibile** dacă ele nu pot avea loc simultan. Posibilele rezultate două câte două incompatibile ale unui experiment se numesc **evenimentele elementare** ale experimentului. Mulțimea evenimentelor elementare este notată cu E și este numită **spațiu de selecție**. De exemplu, în experimentul aruncării unui zar o singură dată, evenimentele elementare sunt e_1, e_2, \dots, e_6 , unde e_i este evenimentul «apare fața i ». Deci, spațiul de selecție va fi $E = \{e_1, e_2, e_3, e_4, e_5, e_6\}$. Dacă se consideră experimentul testării duratei de funcționare a unui bec, există o infinitate de evenimente elementare e_x , $x \in [0, \infty)$, e_x fiind evenimentul «durata de funcționare este x minute».

Intrucât acest material este dedicat matematicilor *discrete*, vom considera în continuare doar experimente cu un număr *finit* sau *numărabil* de rezultate.

Pe lângă evenimentele elementare, suntem desigur interesați de evenimente mai complexe, de exemplu, atunci când este aruncat un zar, apariția unui număr par. Fie un experiment dat și fie E mulțimea evenimentelor sale elementare. Orice submulțime a lui E este numită **eveniment**. Un eveniment A are loc dacă și numai dacă unul din evenimentele elementare care intră în componența lui A are loc. De exemplu, în experimentul aruncării unui zar, mulțimea $A = \{e_2, e_4, e_6\}$ poate fi interpretată ca fiind evenimentul «apare un număr par». Într-adevăr, acest eveniment are loc în aruncarea unui zar dacă și numai dacă unul din evenimentele elementare conținute în A are loc.

Deoarece $E \subseteq E$ și $\emptyset \subseteq E$, E și \emptyset pot de asemenea fi privite ca și evenimente, conform definiției unui eveniment. Intrucât mulțimea E este

alcătuită din toate evenimentele elementare și în orice experiment are loc exact un eveniment elementar, E are loc *întotdeauna*. De aceea E va fi numit **evenimentul sigur**. Pe de altă parte, \emptyset nu conține nici un eveniment elementar, prin urmare nu are loc niciodată. Prin urmare el va fi numit **evenimentul imposibil**.

Fie $A_1, A_2, \dots, A_n \subseteq E$ evenimente. Atunci:

1. $A_1 \cup \dots \cup A_n$ este un eveniment numit **suma** sau **reuniunea** evenimentelor A_i , care are loc exact atunci când cel puțin unul din evenimentele A_1, \dots, A_n are loc.
2. $A_1 \cap \dots \cap A_n$ este un eveniment numit **produsul** sau **intersecția** evenimentelor A_i , care are loc dacă și numai dacă A_1, \dots, A_n au loc simultan.

Evenimentele $A_1, A_2 \subseteq E$ se numesc **incompatibile** dacă $A_1 \cap A_2 = \emptyset$ (A_1 și A_2 nu pot avea loc simultan).

Deoarece $\bar{A} = E \setminus A \subseteq E$, \bar{A} este de asemenea un eveniment. El are loc exact atunci când A nu are loc. Evenimentul \bar{A} va fi numit evenimentul **complementar** lui A .

În final să menționăm că toate proprietățile operațiilor \cup, \cap și complementaritate valabile pentru mulțimi vor fi de asemenea valabile pentru evenimente.

2.2 Axiomele teoriei probabilităților

Definiția 2.1 (Kolmogorov)¹ Fie E un spațiu de selecție asociat unui experiment aleator și fie $\mathcal{P}(E) = \{A, A \subseteq E\}$. O **măsură de probabilitate** pe E este o funcție $P : \mathcal{P}(E) \rightarrow [0, 1]$ care satisface următoarele condiții (axiome):

1. $P(E) = 1$,
2. $P\left(\bigcup_{k=1}^{\infty} A_k\right) = \sum_{k=1}^{\infty} P(A_k)$ dacă A_k sunt evenimente două câte două disjuncte (incompatibile).

Consecințe

Fie E un spațiu de selecție asociat unui experiment aleator și fie $A, B \subseteq E$ evenimente. Au loc următoarele proprietăți:

1. $P(\emptyset) = 0$.
2. $P\left(\bigcup_{k=1}^n A_k\right) = \sum_{k=1}^n P(A_k)$ dacă A_k , $k = 1, \dots, n$, sunt evenimente două câte două incompatibile.
3. $P(\bar{A}) = 1 - P(A)$,
4. $P(B \setminus A) = P(B) - P(A \cap B)$.
Dacă $A \subseteq B$, atunci $P(B \setminus A) = P(B) - P(A)$.
5. $P(A \cup B) = P(A) + P(B) - P(A \cap B)$.
6. Dacă $A \subseteq B$, atunci $P(A) \leq P(B)$.

¹Andrey Nikolaevich Kolmogorov (1903–1987), matematician rus.

7. Dacă A_i , $i = 1, \dots, n$, sunt evenimente arbitrare, atunci

$$P\left(\bigcup_{i=1}^n A_i\right) = \sum_{i=1}^n P(A_i) - \sum_{1 \leq i < j \leq n} P(A_i \cap A_j) + \sum_{1 \leq i < j < k \leq n} P(A_i \cap A_j \cap A_k) - \dots - (-1)^n P(A_1 \cap \dots \cap A_n).$$

8. $P(A \cup B) \leq P(A) + P(B)$.

9. $P\left(\bigcup_{k=1}^{\infty} A_k\right) \leq \sum_{k=1}^{\infty} P(A_k)$.

Demonstrație.

1. Luând în condiția **2**. $A_1 = E$, $A_k = \emptyset$, pentru $k > 1$, obținem

$$P(E) = P(E) + \sum_{i=2}^{\infty} P(\emptyset), \text{ deci } P(\emptyset) = 0.$$

2. Luăm în condiția **2**. $A_k = \emptyset$, pentru $k > n$.

3. Avem $1 = P(E) = P(A \cup \bar{A}) = P(A) + P(\bar{A})$.

4. Scriem $B = (A \cap B) \cup (B \setminus A)$. Deoarece $A \cap B$ și $B \setminus A$ sunt disjuncte, avem $P(B) = P(A \cap B) + P(B \setminus A)$, de unde concluzia.

5. Scriem $A \cup B = A \cup (B \setminus A)$, deci $P(A \cup B) = P(A) + P(B \setminus A)$. Înlocuind $P(B \setminus A)$ din proprietatea precedentă, obținem concluzia.

6. Folosind proprietatea 4 pentru cazul $A \subseteq B$, obținem $0 \leq P(B \setminus A) = P(B) - P(A)$, de unde inegalitatea cerută.

7. Această proprietate poate fi demonstrată prin inducție după n . ■

Observația 2.1 Definiția 2.1 a fost dată de Kolmogorov și ne vom referi la **1.** și **2.** ca și la axiomele lui Kolmogorov. Există și alte definiții ale probabilității, folosind unele din proprietățile 1.-9., caz în care axiomele lui Kolmogorov apar ca și consecințe.

Propoziția 2.1 Dacă natura unui experiment este astfel încât acesta are doar un număr finit de evenimente elementare, și dacă ele sunt egal probabile, atunci din axiomele lui Kolmogorov rezultă că probabilitatea unui eveniment A este

$$P(A) = \frac{\text{numărul de evenimente elementare favorabile lui } A}{\text{numărul tuturor evenimentelor elementare}}. \quad (2.1)$$

Formula (2.1) a fost folosită de Laplace² pentru a defini probabilitatea.

Demonstrație. Vom arăta că o probabilitate definită prin axiomele lui Kolmogorov satisface (2.1).

²Pierre-Simon, Marquis de Laplace (1749-1827), matematician și astronom francez.

Pentru început considerăm spațiul de selecție $E = \{e_1, \dots, e_n\}$ cu e_i evenimente elementare egal probabile. Atunci

$$1 = P(E) = P(e_1 \cup \dots \cup e_n) = \sum_{i=1}^n P(e_i) = nP(e_j), \quad (2.2)$$

de unde

$$P(e_j) = \frac{1}{n} \text{ pentru orice } j \in \mathbb{N}_n.$$

În a treia egalitate din (2.2) am folosit faptul că evenimentele elementare sunt incompatibile. Mai departe, fie $A \subseteq E$, $A = e_1 \cup e_2 \cup \dots \cup e_k$. Probabilitatea lui A va fi

$$P(A) = P(e_1 \cup e_2 \cup \dots \cup e_k) = \sum_{j=1}^k P(e_j) = k \cdot \frac{1}{n} = \frac{k}{n}.$$

Intr-adevăr, k este numărul de evenimente elementare favorabile lui A și n este numărul total de evenimente elementare. În concluzie, (2.1) este demonstrată. ■

Exemplul 2.1 Două zaruri sunt aruncate simultan și există un premiu atunci când suma fețelor este ≥ 10 . Care este probabilitatea de a câștiga premiul?

Solution. Spațiul de selecție asociat experimentului va fi

$$E = \{e_{11}, e_{12}, \dots, e_{16}, e_{21}, e_{22}, \dots, e_{26}, \dots, e_{61}, e_{62}, \dots, e_{66}\},$$

în timp ce evenimentul A de a avea suma fețelor ≥ 10 poate fi scris ca și

$$A = \{e_{46}, e_{55}, e_{56}, e_{64}, e_{65}, e_{66}\}.$$

Astfel,

$$P(A) = \frac{|A|}{|E|} = \frac{6}{36} = \frac{1}{6}.$$

2.3 Probabilități condiționate și evenimente aleatoare independente

Probabilitatea unui eveniment aleator A este alterată, în general, când se cunoaște faptul că un alt eveniment aleator B cu $P(B) \neq 0$ a avut loc. Probabilitatea evenimentului A , cu condiția că B (cu $P(B) \neq 0$) a avut loc deja, este notată cu $P(A|B)$ și este numită **probabilitatea evenimentului A condiționată de evenimentul B** .

Exemplul 2.2 Presupunem că avem două urne, U_1 conținând 5 albe și 5 bile negre și U_2 conținând o bilă albă și 9 bile negre. Realizăm experimentul de a alege la întâmplare una din urne și de a scoate, la întâmplare, o bilă din această urnă. Apoi considerăm următoarele evenimente:

$$\begin{aligned} B &: \text{«bila extrasă este albă»}, \\ A_i &: \text{«bila a fost extrasă din urna } i\text{»}, \quad i = 1, 2. \end{aligned}$$

Atunci

$$P(B|A_1) = \frac{5}{10} = \frac{1}{2}, \quad P(B|A_2) = \frac{1}{10}.$$

În general, probabilitățile condiționate sunt definite prin relațiile

$$\begin{aligned} P(A|B) &= \frac{P(A \cap B)}{P(B)}, \text{ dacă } P(B) \neq 0, \\ P(B|A) &= \frac{P(A \cap B)}{P(A)}, \text{ dacă } P(A) \neq 0, \end{aligned}$$

care implică

$$P(A \cap B) = P(B) \cdot P(A|B) = P(A) \cdot P(B|A). \quad (2.3)$$

Definiția 2.2 Două evenimente aleatoare A, B se numesc **independente** (unul de celălalt) atunci când realizarea unuia nu influențează în nici un fel probabilitatea de realizare a celuilalt, adică atunci când $P(A|B) = P(A)$.

Astfel, pentru evenimentele independente A, B avem

$$P(A \cap B) = P(A) \cdot P(B). \quad (2.4)$$

Definiția 2.3 Evenimentele aleatoare A_1, \dots, A_n se numesc **total independente** dacă pentru orice $m \leq n$ și orice m -uplu (i_1, i_2, \dots, i_m) , $1 \leq i_1 < i_2 < \dots < i_m \leq n$ are loc

$$P(A_{i_1} \cap A_{i_2} \cap \dots \cap A_{i_m}) = P(A_{i_1})P(A_{i_2}) \dots P(A_{i_m}).$$

Definiția 2.4 Evenimentele aleatoare A_1, \dots, A_n se numesc **două câte două independente** dacă A_i și A_j sunt independente pentru orice $i \neq j$.

Este imediat faptul că evenimentele total independente sunt independente două câte două. Totuși, implicația inversă nu este adevărată în general, așa cum arată următorul exemplu.

Exemplul 2.3 Fie $E = \{e_1, e_2, e_3, e_4\}$ un spațiu de selecție, unde toate evenimentele elementare e_i au probabilitatea $P(e_i) = 1/4$. Considerăm evenimentele $A = e_1 \cup e_2$, $B = e_1 \cup e_3$, $C = e_1 \cup e_4$. Vom arăta că evenimentele A, B, C sunt două câte două independente, dar nu sunt total independente.

Pe de o parte, $P(A) = P(B) = P(C) = 1/2$,

$$P(A \cap B) = P(e_1 \cup (e_2 \cap e_3)) = P(e_1 \cup \emptyset) = P(e_1) = \frac{1}{4},$$

și analog $P(A \cap C) = P(B \cap C) = 1/4$. În concluzie

$$\begin{aligned} P(A \cap B) &= P(A) \cdot P(B), \\ P(A \cap C) &= P(A) \cdot P(C), \\ P(B \cap C) &= P(B) \cdot P(C), \end{aligned}$$

deci evenimentele A, B, C sunt două câte două independente.

Pe de altă parte,

$$P(A \cap B \cap C) = P(e_1) = \frac{1}{4}, \quad P(A) \cdot P(B) \cdot P(C) = \frac{1}{8},$$

deci A, B, C nu sunt total independente.

Teorema 2.2 Fie A_1, A_2, \dots, A_n evenimente aleatoare. Atunci

$$P(A_1 \cap A_2 \cap \dots \cap A_n) = P(A_1) \cdot P(A_2|A_1) \cdot P(A_3|A_1 \cap A_2) \cdot \dots \cdot P(A_n|A_1 \cap \dots \cap A_{n-1}).$$

Demonstrație. Demonstrația este imediată folosind inducția după n și formula (2.3). ■

O consecință imediată este

Corolarul 2.3 Dacă evenimentele A_1, A_2, \dots, A_n sunt total independente, atunci

$$P(A_1 \cap A_2 \cap \dots \cap A_n) = P(A_1) \cdot P(A_2) \cdot \dots \cdot P(A_n).$$

2.4 Formula probabilității totale. Formula lui Bayes³

Definiția 2.5 Fie E un spațiu de selecție asociat unui experiment aleator. O mulțime $\mathcal{S} = \{A_1, A_2, \dots, A_n\}$ de evenimente aleatoare se numește **sistem complet de evenimente (SCE)** pentru E dacă

1. $E = A_1 \cup A_2 \cup \dots \cup A_n$,
2. $A_i \cap A_j = \emptyset$, pentru $i \neq j$.

Un (SCE) se poate defini în mod analog pentru o mulțime numărabilă de evenimente.

Teorema 2.4 (Legea probabilității totale) Fie E un spațiu de selecție asociat unui experiment, $\mathcal{S} = \{A_1, A_2, \dots, A_n\}$ un (SCE) și B un eveniment aleator. Atunci

$$P(B) = \sum_{i=1}^n P(A_i) \cdot P(B|A_i). \quad (2.5)$$

Formula (2.5) este cunoscută sub numele de **formula probabilității totale (FTP)**.

Demonstrație. Avem

$$\begin{aligned} B = E \cap B &= (A_1 \cup A_2 \cup \dots \cup A_n) \cap B \\ &= (A_1 \cap B) \cup (A_2 \cap B) \cup \dots \cup (A_n \cap B). \end{aligned}$$

Intrucât evenimentele $A_i \cap B$ din ultima sumă sunt două câte două incompatibile, se obține că

$$P(B) = \sum_{i=1}^n P(A_i \cap B).$$

Mai departe, folosirea formulei (2.3) pentru probabilitățile $P(A_i \cap B)$ ne conduce la concluzia (2.5). ■

Observația 2.2 Să menționăm că evenimentul B are loc împreună cu unul (și numai unul) din evenimentele unui SCE.

³Thomas Bayes (1702-1762), matematician britanic.

Exemplul 2.4 *Avem trei urne de tipul 1 (conținând două bile albe și șase bile negre) și o urnă de tipul 2 (conținând o bilă albă și opt bile negre). Se alege o urnă la întâmplare și apoi din urna aleasă se extrage o bilă. Care este probabilitatea evenimentului B : «bila extrasă este albă»?*

Soluție. Considerăm evenimentele A_i : «este aleasă o urnă de tipul i », $i = 1, 2$. Atunci $\mathcal{S} = \{A_1, A_2\}$ va fi un SCE, întrucât $E = A_1 \cup A_2$ și $A_1 \cap A_2 = \emptyset$. Evenimentul B are loc împreună cu unul din evenimentele din \mathcal{S} . Conform formulei probabilității totale avem

$$P(B) = P(A_1) \cdot P(B|A_1) + P(A_2) \cdot P(B|A_2).$$

Inlocuind $P(A_1) = 3/4$, $P(A_2) = 1/4$, $P(B|A_1) = 2/8$, $P(B|A_2) = 1/9$, obținem $P(B) = 31/144$.

Presupunem că ipotezele teoremei probabilității totale sunt satisfăcute. Atunci putem cere să se calculeze probabilitatea evenimentului A_i , condiționată de faptul că evenimentul B a avut loc. Această probabilitate va fi

$$P(A_i|B) = \frac{P(A_i) \cdot P(B|A_i)}{P(B)} = \frac{P(A_i) \cdot P(B|A_i)}{\sum_{k=1}^n P(A_k) \cdot P(B|A_k)}. \quad (2.6)$$

Această formulă este cunoscută sub numele de **formula lui Bayes**. Ea este utilizată pentru a evalua probabilitățile evenimentelor A_i care formează un (SCE), după ce se cunoaște rezultatul unui experiment. De fapt acestea sunt probabilitățile *cauzelor* realizării unui eveniment.

Formula lui Bayes conține probabilitățile

$$P(A_i), P(B|A_i), \quad i \in \mathbb{N}_n,$$

care pot fi calculate *înaintea* efectuării experimentului. Din acest motiv ele se numesc **probabilități a priori**.

Executând experimentul, constatăm că evenimentul B a avut loc și vrem să stabilim probabilitățile

$$P(A_i|B), \quad \text{pentru } i = 1, \dots, n,$$

unde $P(A_i|B)$ reprezintă probabilitatea ca realizarea lui B să fie datorată apariției evenimentului A_i (B să se fi realizat împreună cu A_i). Aceste probabilități se numesc **probabilități a posteriori**.

Exemplul 2.5 *In experimentul din exemplul 2.4, presupunem că a fost extrasă o bilă albă (B s-a realizat). Care este probabilitatea ca bila extrasă să fi provenit dintr-o urnă de tipul 1 (probabilitatea ca să se fi realizat datorită evenimentului A_1)?*

Soluție. Aplicând formula lui Bayes obținem

$$P(A_1|B) = \frac{P(A_1) \cdot P(B|A_1)}{P(B)} = \frac{\frac{3}{4} \cdot \frac{1}{4}}{\frac{31}{144}} = \frac{27}{31}.$$

2.5 Scheme probabilistice

2.5.1 Schema binomială

Presupunem că avem o urnă conținând N_1 bile albe și N_2 bile negre. Extragem o bilă, observăm culoarea sa și apoi o reintroducem în urnă. Evident, compoziția urnei rămâne neschimbată de la o extragere la alta. Dacă notăm cu A evenimentul «bila extrasă este albă», atunci \bar{A} va fi evenimentul «bila extrasă este neagră» și probabilitățile acestor evenimente vor fi

$$P(A) = \frac{N_1}{N} = p, \quad P(\bar{A}) = \frac{N_2}{N} = q, \quad \text{cu } N = N_1 + N_2.$$

Desigur, $p + q = 1$ și p, q rămân neschimbate de la o extragere la alta.

Efectuăm n asemenea extracții conform procedurii descrise mai sus. Cărem probabilitatea evenimentului

X_k : «printre cele n bile extrase, k sunt albe și $n - k$ sunt negre» .

Soluție. O succesiune favorabilă de evenimente A, \bar{A} este evenimentul

$$B = \underbrace{A A \dots A}_{k \text{ ori}} \underbrace{\bar{A} \bar{A} \dots \bar{A}}_{n - k \text{ ori}}.$$

Astfel, $P(B) = p^k q^{n-k}$. Întrucât există $\binom{n}{k}$ asemenea evenimente favorabile cu k apariții ale lui A și $n - k$ apariții ale lui \bar{A} (vezi exemplul 1.12, pag. 15) și întrucât aceste $\binom{n}{k}$ evenimente sunt incompatibile, avem

$$P(X_k) = \binom{n}{k} p^k q^{n-k}, \quad \text{pentru } k = 0, 1, \dots, n.$$

Observația 2.3 Probabilitatea $P(X_k)$ este coeficientul lui x^k din polinomul $\varphi_n(x) = (px + q)^n$. Din acest motiv, funcția φ_n se numește funcția generatoare pentru schema binomială.

Schema binomială a fost studiată pentru prima oară de Bernoulli⁴. Din acest motiv, ea este se mai numește și **schema lui Bernoulli**. O aplicare a acestei scheme este prezentată în paragraful 2.14, pag. 62.

2.5.2 Schema multinomială

Această schemă este o generalizare a schemei binomiale. Presupunem că se dă o urnă care conține N bile de k culori notate c_1, c_2, \dots, c_k . Mai precis, există N_i bile de culoarea c_i , pentru $i \in \mathbb{N}_k$, $N_1 + \dots + N_k = N$. Ca în cazul anterior, extragem o bilă, observăm culoarea ei și o reintroducem în urnă. Dacă se notează cu A_i evenimentul «bila extrasă are culoarea c_i », atunci

$$p_i = P(A_i) = \frac{N_i}{N}, \quad \text{pentru } i \in \mathbb{N}_k$$

și $\mathcal{S} = \{A_1, A_2, \dots, A_k\}$ va fi un sistem complet de evenimente, cu evenimentul sigur E : «o bilă este extrasă». Evident, $p_1 + p_2 + \dots + p_k = 1$.

⁴James Bernoulli (cunoscut și ca Jacob I) (1654-1705), matematician elvețian.

Efectuăm n extrageri în modul descris mai sus. Cerem probabilitatea evenimentului

$X_{n_1, \dots, n_k} : \ll \text{printre cele } n \text{ bile extrase, } n_i \text{ au culoarea } c_i, \text{ pentru } i \in \mathbb{N}_k \gg.$

Desigur, $n_1 + n_2 + \dots + n_k = n$.

Soluție. Vom arăta că

$$P(X_{n_1, n_2, \dots, n_k}) = \frac{n!}{n_1! \cdot n_2! \cdot \dots \cdot n_k!} \cdot p_1^{n_1} p_2^{n_2} \dots p_k^{n_k}. \quad (2.7)$$

O succesiune favorabilă de evenimente A_1, A_2, \dots, A_k este evenimentul

$$B = \underbrace{A_1 A_1 \dots A_1}_{n_1 \text{ ori}} \underbrace{A_2 A_2 \dots A_2}_{n_2 \text{ ori}} \dots \underbrace{A_k A_k \dots A_k}_{n_k \text{ ori}}.$$

Probabilitatea acestui eveniment este $P(B) = p_1^{n_1} p_2^{n_2} \dots p_k^{n_k}$. Intrucât există

$$\binom{n}{n_1, n_2, \dots, n_k}$$

asemenea evenimente favorabile cu n_i apariții ale culorii c_i pentru $1 \leq i \leq k$ (vezi teorem 1.14, pag. 20) și întrucât aceste evenimente sunt incompatibile, formula (2.7) are loc.

Teorema 1.15 ne permite să facem următoarea observație.

Observația 2.4 Probabilitatea în (2.7) este coeficientul lui $x_1^{n_1} x_2^{n_2} \dots x_k^{n_k}$ al polinomului în k variabile

$$\Phi_n(x_1, x_2, \dots, x_k) = (p_1 x_1 + p_2 x_2 + \dots + p_k x_k)^n.$$

Datorită acestui fapt, această funcție se numește **funcția generatoare a schemei multinomiale**.

2.5.3 Schema hipergeometrică

Presupunem că avem o urnă conținând N bile, din care N_1 sunt albe și $N_2 = N - N_1$ sunt negre. Se extrag n bile succesiv, fără a le reintroduce în urnă. Se cere probabilitatea evenimentului

$Y_{n_1, n_2} : \ll \text{printre cele } n \text{ bile extrase, } n_1 \text{ sunt albe și } n_2 \text{ sunt negre} \gg.$

Soluție. Dacă notăm bilele albe cu a_1, a_2, \dots, a_{N_1} , și bilele negre cu b_1, b_2, \dots, b_{N_2} , există $\binom{N}{n}$ moduri de a extrage n bile dintr-o urnă de N bile și o succesiune favorabilă de bile este de forma

$$a_{i_1}, a_{i_2}, \dots, a_{i_{n_1}}, b_{j_1}, b_{j_2}, \dots, b_{j_{n_2}}.$$

Există $\binom{N_1}{n_1}$ moduri de a extrage n_1 bile albe dintr-un total de N_1 bile albe. Pentru fiecare succesiune fixată de bile albe, există $\binom{N_2}{n_2}$ moduri de a extrage n_2 bile negre dintr-un total de N_2 bile negre. Deci, în total există

$$\binom{N_1}{n_1} \binom{N_2}{n_2}$$

extracții favorabile și prin urmare probabilitatea cerută este

$$P(Y_{n_1, n_2}) = \frac{\binom{N_1}{n_1} \binom{N_2}{n_2}}{\binom{N}{n}} = \frac{\binom{N_1}{n_1} \binom{N_2}{n_2}}{\binom{N_1 + N_2}{n_1 + n_2}}. \quad (2.8)$$

2.5.4 Schema hipergeometrică generalizată

Schema hipergeometrică se poate generaliza în felul următor: se presupune că o urnă conține N bile de k culori, notate c_1, c_2, \dots, c_k . Mai precis, există N_i bile de culoarea c_i ($1 \leq i \leq k$) și $N_1 + \dots + N_k = N$. Se extrag n bile și se cere probabilitatea evenimentului

$$Y_{n_1, \dots, n_k} : \text{«printre cele } n \text{ bile extrase, } n_i \text{ au culoarea } c_i, i \in \mathbb{N}_k \text{»}.$$

Desigur, $n_1 + n_2 + \dots + n_k = n$.

Argumente similare cu cele folosite la schema hipergeometrică ne conduc la formula

$$P(Y_{n_1, n_2, \dots, n_k}) = \frac{\binom{N_1}{n_1} \binom{N_2}{n_2} \dots \binom{N_k}{n_k}}{\binom{N}{n}}. \quad (2.9)$$

2.5.5 Schema urnelor lui Poisson

În această schemă avem n urne notate U_i , $1 \leq i \leq n$, fiecare conținând bile albe și negre, dar în proporții diferite. Introducem următoarele notații:

$$\begin{aligned} p_i &= \text{probabilitatea să extragem o bilă albă din urna } U_i, \\ q_i &= \text{probabilitatea să extragem o bilă neagră din urna } U_i, \end{aligned}$$

pentru $1 \leq i \leq n$ și cu $p_i + q_i = 1$. Extragem n bile, câte una din fiecare urnă, și vrem să calculăm probabilitatea evenimentului

$$Z_k : \text{«printre cele } n \text{ bile extrase, } k \text{ sunt albe și } n - k \text{ sunt negre»}.$$

Soluție. Pentru a simplifica prezentarea, considerăm pentru început cazul particular $n = 4$ și calculăm probabilitatea de a avea $k = 3$ bile albe și o bilă neagră ($n - k = 1$). Pentru $i \in \mathbb{N}_4$, considerăm evenimentele A_i : «bila extrasă din urna U_i este albă». Atunci evenimentele complementare \bar{A}_i vor fi «bila extrasă din urna U_i este neagră».

Evenimentele care exprimă extracțiile favorabile sunt:

$$\begin{aligned} B_1 &= A_1 \cap A_2 \cap A_3 \cap \bar{A}_4, \\ B_2 &= A_1 \cap A_2 \cap \bar{A}_3 \cap A_4, \\ B_3 &= A_1 \cap \bar{A}_2 \cap A_3 \cap A_4, \\ B_4 &= \bar{A}_1 \cap A_2 \cap A_3 \cap A_4, \end{aligned}$$

evenimentele din componența fiecărui produs fiind total independente. Prin urmare, probabilitățile evenimentelor favorabile sunt

$$\begin{aligned} P(B_1) &= p_1 p_2 p_3 q_4, & P(B_2) &= p_1 p_2 q_3 p_4, \\ P(B_3) &= p_1 q_2 p_3 p_4, & P(B_4) &= q_1 p_2 p_3 p_4. \end{aligned}$$

Mai departe, evenimentul Z_3 poate fi scris ca și

$$Z_3 = B_1 \cup B_2 \cup B_3 \cup B_4.$$

Intrucât evenimentele din această sumă sunt incompatibile, probabilitatea cerută va fi suma probabilităților $P(B_i)$,

$$P(Z_3) = p_1 p_2 p_3 q_4 + p_1 p_2 q_3 p_4 + p_1 q_2 p_3 p_4 + q_1 p_2 p_3 p_4. \quad (2.10)$$

Dacă se consideră polinomul

$$Q_4(x) = (p_1x + q_1)(p_2x + q_2)(p_3x + q_3)(p_4x + q_4),$$

atunci $P(Z_3)$ reprezintă coeficientul lui x^3 din polinomul $Q_4(x)$.

În cazul general, considerăm polinomul

$$Q_n(x) = \prod_{i=1}^n (p_i x + q_i)$$

și folosim un raționament similar pentru a arăta că probabilitatea $P(Z_k)$ este coeficientul lui x^k din $Q_n(x)$.

Observația 2.5 Dacă toate urnele sunt identice, atunci $p_i = p$, $q_i = q$ și polinomul $Q_n(x)$ se reduce la

$$Q_n(x) = (px + q)^n.$$

Coeficientul lui x^k va fi

$$\binom{n}{k} p^k q^{n-k},$$

care este exact probabilitatea din schema binomială. Într-adevăr, faptul că urnele sunt identice este echivalent cu situația când avem o urnă și reintroducem bila în urnă după fiecare extragere.

2.6 Variabile aleatoare

Există multe definiții pentru variabila aleatoare (v.a.), din care o vom adopta pe următoarea:

Definiția 2.6 O variabilă care ia valori care depind de rezultatul unui experiment, astfel depinzând de șansă, este numită **variabilă aleatoare**. Aceasta este de fapt o funcție cu valori reale definită pe spațiul de selecție.

De exemplu, numărul de bacterii pe o unitate de suprafață în studiul efectului unui medicament, numărul de votanți în favoarea anumitui candidat sunt variabile aleatoare.

Definiția 2.7 Fie X o variabilă aleatoare. **Funcția de distribuție** (sau **funcția de repartiție**) a v.a. X este definită prin $F : \mathbb{R} \rightarrow [0, 1]$,

$$F(x) = P(X < x), \text{ pentru orice } x \in \mathbb{R}.$$

Din punctul de vedere al teoriei probabilităților, o variabilă aleatoare este complet caracterizată de funcția sa de repartiție. Ea se consideră cunoscută atunci când funcția sa de repartiție este cunoscută. Funcția de repartiție F definită mai sus următoarele proprietăți:

P1. $P(a \leq X < b) = F(b) - F(a)$, pentru orice $a, b \in \mathbb{R}$, $a < b$.

Demonstrație. Dacă se consideră evenimentele

$$A : a \leq X < b,$$

$$B : X < b,$$

$$C : X < a,$$

avem $B = A \cup C$ și $A \cap C = \emptyset$. Astfel, $P(B) = P(A) + P(C)$ și prin urmare

$$P(A) = P(B) - P(C) = P(X < b) - P(X < a) = F(b) - F(a).$$

P2. Funcția F este crescătoare.

Demonstrație. Fie $x_1 \leq x_2$. Atunci $P(x_1 \leq X < x_2) \geq 0$, de unde, folosind **P1**, $F(x_2) - F(x_1) \geq 0$.

P3. Funcția F este continuă la stânga:

$$F(x - 0) = F(x), \quad \text{pentru orice } x \in \mathbb{R}.$$

P4. $\lim_{x \rightarrow -\infty} F(x) = 0, \quad \lim_{x \rightarrow \infty} F(x) = 1.$

În practică există două tipuri de variabile aleatoare: *discrete* și *continue*.

2.6.1 Variabile aleatoare discrete

Definiția 2.8 O variabilă aleatoare X se numește **discretă** (v.a.d.) dacă poate lua doar un număr finit sau numărabil de valori. Astfel, ea este determinată de valorile sale x_1, x_2, \dots, x_n (sau x_1, x_2, \dots) și de probabilitățile $p_i = P(X = x_i)$ cu care ia aceste valori. Valorile p_i trebuie să satisfacă $\sum_i p_i = 1$.

Deci, o v.a.d. poate fi scrisă ca și

$$X : \begin{pmatrix} x_1 & x_2 & \dots & x_n \\ p_1 & p_2 & \dots & p_n \end{pmatrix}, \quad p_i > 0, \quad \sum_{i=1}^n p_i = 1,$$

dacă ia un număr finit de valori, sau ca și

$$X : \begin{pmatrix} x_1 & x_2 & \dots & x_n & \dots \\ p_1 & p_2 & \dots & p_n & \dots \end{pmatrix}, \quad p_i > 0, \quad \sum_{i=1}^{\infty} p_i = 1,$$

dacă mulțimea valorilor sale este numărabilă. Pentru a nu considera de fiecare dată cazuri separate, vom adopta uneori a doua notație pentru o v.a. finită, admitând că $p_i = 0$ pentru $i = n + 1, n + 2, \dots$. În acest caz vom nota

$$\mathcal{X} = \{x_1, x_2, \dots, x_n, \dots\} \text{ sau } \mathcal{X} = \{x_1, x_2, \dots, x_n\}.$$

Definiția 2.9 Funcția $f_X : \mathcal{X} \rightarrow \{p_1, p_2, \dots, p_n, \dots\}$,

$$f_X(x_i) = p_i = P(X = x_i),$$

se numește **funcție de probabilitate** a v.a.d. X .

Funcția de probabilitate f_X definește complet variabila aleatoare X .

Propoziția 2.5 Cu notațiile de mai sus avem

$$F(x) = \sum_{x_i < x} p_i.$$

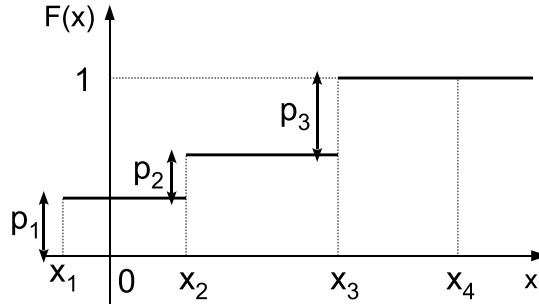


Figura 2.1: Funcția de repartiție a unei variabile aleatoare discrete.

Demonstrație. Fie $k \in \mathbb{N}$ astfel încât $x_k < x \leq x_{k+1}$. Atunci

$$\begin{aligned}
 F(x) &= P(X < x) \\
 &= P(X = x_1 \cup X = x_2 \cup \dots \cup X = x_k) \\
 &= p_1 + p_2 + \dots + p_k = \sum_{i=1}^k p_i = \sum_{x_i < x} p_i.
 \end{aligned}$$

■

Definiția 2.10 Variabilele aleatoare X și Y se numesc **independente** dacă

$$P(X = x, Y = y) = P(X = x) \cdot P(Y = y).$$

Variabilele aleatoare X_1, \dots, X_n se numesc **total independente** dacă pentru orice $m \leq n$ și orice m -uplu (i_1, i_2, \dots, i_m) , $1 \leq i_1 < i_2 < \dots < i_m \leq n$ avem

$$\begin{aligned}
 &P(X_{i_1} = x_1, X_{i_2} = x_2, \dots, X_{i_m} = x_m) \\
 &= P(X_{i_1} = x_1) \cdot P(X_{i_2} = x_2) \cdot \dots \cdot P(X_{i_m} = x_m).
 \end{aligned}$$

2.6.2 Variabile aleatoare continue

Definiția 2.11 O v.a. se numește **absolut continuă** dacă funcția sa de distribuție F poate fi scrisă ca și

$$F(x) = \int_{-\infty}^x f(t) dt.$$

In acest caz, funcția $f : \mathbb{R} \rightarrow \mathbb{R}$ se numește funcția **densitate** a distribuției.

Intrucât $\lim_{x \rightarrow \infty} F(x) = 1$, funcția densitate trebuie să satisfacă egalitatea

$$\int_{-\infty}^{\infty} f(x) dx = 1.$$

Alte proprietăți imediate ale unei v.a. continue sunt:

$$\begin{aligned}
 P(a \leq X < b) &= F(b) - F(a) = \int_a^b f(x) dx, \\
 P(X = a) &= 0.
 \end{aligned}$$

Un exemplu de distribuție continuă des întâlnit în practică este **distribuția normală**, numită și distribuția lui Gauss:⁵

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \cdot e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}.$$

Această distribuție de probabilitate are o importanță majoră în multe do-

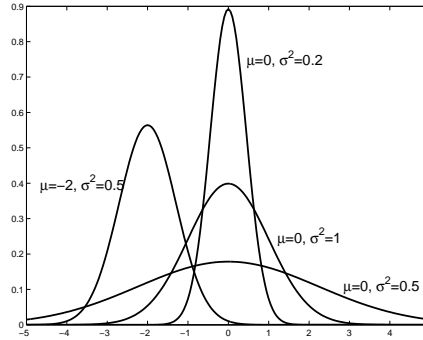


Figura 2.2: Câteva funcții densitate normale.

menii. Ea constă dintr-o familie de distribuții de aceeași formă generală, depinzând de doi parametri: media μ și deviația standard σ . Graficul său se numește deseori *clopotul lui Gauss*, datorită faptului că graficul funcției sale densitate de probabilitate are forma unui clopot (vezi figura 2.2).

În continuare ne vom limita la variabile aleatoare discrete.

2.7 Suma variabilelor aleatoare

Definiția 2.12 Fie X, Y două variabile aleatoare discrete. Suma lor este de asemenea o v.a.d., notată $X + Y$, și definită prin

$$P(X + Y = z) = \sum_{x \in \mathcal{X}} P(X = x, Y = z - x), \quad (2.11)$$

Pentru suma a n variabile aleatoare X_1, \dots, X_n , definiția poate fi adaptată cu ușurință:

$$P(X_1 + \dots + X_n = z) = \sum_{x_1 \in \mathcal{X}_1, \dots, x_{n-1} \in \mathcal{X}_{n-1}} P(X_1 = x_1, \dots, X_{n-1} = x_{n-1}, X_n = z - (x_1 + \dots + x_{n-1})).$$

Dacă X, Y sunt variabile aleatoare independente, suma (2.11) devine

$$P(X + Y = z) = \sum_{x \in \mathcal{X}} P(X = x) \cdot P(Y = z - x).$$

Exemplul 2.6 Fie $p \in (0, 1)$, $q = 1 - p$ și fie X, Y v.a. independente generate de schema binomială:

$$X : \binom{n}{k} p^k q^{n-k}, \quad k = 0, \dots, n, \quad Y : \binom{m}{j} p^j q^{m-j}, \quad j = 0, \dots, m.$$

⁵Carl Friedrich Gauss (Gauß) (1777-1855), matematician și fizician german, care a adus contribuții semnificative în multe domenii, printre care teoria numerelor, analiză, geometrie diferențială, geodezie, magnetism, astronomie și optică.

Pentru suma $X + Y$ avem

$$\begin{aligned}
 P(X + Y = k) &= \sum_{j=0}^k P(X = j) \cdot P(Y = k - j) \\
 &= \sum_{j=0}^k \binom{n}{j} p^j q^{n-j} \cdot \binom{m}{k-j} p^{k-j} q^{m-(k-j)} \\
 &= p^k q^{n+m-k} \sum_{j=0}^k \binom{n}{j} \binom{m}{k-j} \\
 &= p^k q^{n+m-k} \binom{m+n}{k}.
 \end{aligned}$$

In concluzie,

$$X + Y : \left(\binom{m+n}{k} p^k q^{m+n-k} \right), \quad i = 0, \dots, m + n.$$

2.8 Exemple de variabile aleatoare discrete

2.8.1 Distribuția binomială

Definiția 2.13 Variabila aleatoare discretă X are o **distribuție binomială** cu parametrii n, p ($n \in \mathbb{N}$, $p \in (0, 1)$) dacă funcția sa de probabilitate este $f_X : \{0, 1, \dots, n\} \rightarrow (0, 1]$,

$$f_X(k) = \binom{n}{k} p^k q^{n-k} \text{ pentru orice } k \in \mathbb{N}_n \cup \{0\}, \text{ unde } q = 1 - p. \quad (2.12)$$

Definiția este corectă întrucât $f_X > 0$ și $\sum_{k=0}^n f_X(k) = 1$, ultima egalitate rezultând din

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

Cu notațiile din definiția 2.8, această v.a. se poate scrie de asemenea

$$X : \left(\binom{n}{k} p^k q^{n-k} \right), \quad k \in \{0, 1, \dots, n\}. \quad (2.13)$$

Distribuția binomială poate fi definită astfel: presupunem că repetăm un experiment de n ori și că experimentele individuale în această serie de repetări sunt total independente. Presupunem de asemenea că în toate experimentele un eveniment A are loc cu probabilitatea p , independentă de numărul experimentului. Atunci v.a. X va reprezenta numărul de apariții ale lui A într-o asemenea serie de n experimente. Cum am văzut deja în paragraful 2.5.1, funcția de probabilitate f_X este într-adevăr cea dată în (2.12).

2.8.2 Distribuția hipergeometrică

Definiția 2.14 Variabilă aleatoare discretă X are o **distribuție hipergeometrică** cu parametrii n, a, b ($n, a, b \in \mathbb{N}$, $n \leq a$, $n \leq b$) dacă funcția sa de probabilitate este $f_X : \{0, 1, \dots, n\} \rightarrow (0, 1]$,

$$f_X(k) = \frac{\binom{a}{k} \binom{b}{n-k}}{\binom{a+b}{n}}, \text{ pentru orice } k \in \{0, 1, \dots, n\}. \quad (2.14)$$

Definiția este corectă, întrucât $f_X > 0$ și $\sum_{k=0}^n f_X(k) = 1$. Ultima egalitate are loc din identitatea lui Vandermonde⁶

$$\sum_{k=0}^n \binom{a}{k} \binom{b}{n-k} = \binom{a+b}{n}, \quad (2.15)$$

care rezultă prin identificarea coeficientului lui x^n din egalitatea

$$(1+x)^{a+b} = (1+x)^a (1+x)^b.$$

V. a. hipergeometrică poate fi generată folosind schema hipergeometrică (a se vedea paragraful 2.5.3).

2.8.3 Distribuția lui Poisson

Definiția 2.15 Variabila aleatoare discretă X are o **distribuție Poisson**⁷ cu parametrul $\lambda > 0$ dacă funcția sa de probabilitate este $f_X : \{0, 1, 2, \dots\} \rightarrow (0, 1]$,

$$f_X(k) = \frac{\lambda^k}{k!} e^{-\lambda}. \quad (2.16)$$

Avem $f_X > 0$ și

$$\sum_{k=0}^{\infty} f_X(k) = e^{-\lambda} \sum_{k=0}^{\infty} \frac{\lambda^k}{k!} = e^{-\lambda} \cdot e^{\lambda} = 1,$$

deci definiția este corectă.

2.8.4 Distribuția geometrică

Definiția 2.16 Variabila aleatoare discretă X are o **distribuție geometrică** cu parametrul $p \in (0, 1]$ dacă funcția sa de probabilitate este $f_X : \mathbb{N} \rightarrow (0, 1]$,

$$f_X(k) = p q^{k-1}, \text{ pentru orice } k \in \mathbb{N} \text{ și cu } q = 1 - p. \quad (2.17)$$

Distribuția geometrică poate fi generată astfel: considerăm un experiment și fie A un eveniment care poate avea loc ca și rezultat al experimentului. Fie $p = P(A)$. Repetăm experimentul independent, până la prima apariție a lui A și notăm cu A_i evenimentul care constă în apariția lui A în al i -lea experiment, $i = 1, 2, \dots$. Atunci, pentru variabila aleatoare X reprezentând numărul de experimente efectuate până la prima apariție a lui A , avem

$$f_X(k) = P(X = k) = P(\overline{A}_1 \cap \overline{A}_2 \cap \dots \cap \overline{A}_{k-1} \cap A_k) = p q^{k-1}.$$

Desigur, $f_X > 0$ și

$$\sum_{k=1}^{\infty} p q^{k-1} = \frac{p}{1-q} = 1,$$

seria implicată fiind cea geometrică.

⁶Alexandre-Théophile Vandermonde (1735–1796), muzician și chimist francez.

⁷Siméon Denis Poisson (1781–1840), matematician și fizician francez.

2.8.5 Distribuția binomială negativă

Definiția 2.17 Variabila aleatoare discretă X are o **distribuție binomială negativă**, sau **binomială cu exponent negativ** cu parametrul $p \in (0, 1]$, dacă funcția sa de probabilitate este $f_X : \{n, n+1, \dots\} \rightarrow (0, 1]$,

$$f_X(k) = \binom{k-1}{n-1} p^n q^{k-n}, \quad (2.18)$$

pentru orice $k \in \{n, n+1, \dots\}$, $q = 1 - p$.

Pentru a demonstra corectitudinea, folosim seria binomială generalizată

$$\begin{aligned} (1+x)^\alpha &= 1 + \alpha x + \frac{\alpha(\alpha-1)}{2!} x^2 + \frac{\alpha(\alpha-1)(\alpha-2)}{3!} x^3 + \dots \\ &= \sum_{j=0}^{\infty} \binom{\alpha}{j} x^j \quad \text{cu } \alpha \in \mathbb{R}, |x| < 1. \end{aligned} \quad (2.19)$$

Aici $\binom{\alpha}{j}$ reprezintă numărul binomial

$$\binom{\alpha}{j} = \frac{\alpha(\alpha-1)\dots(\alpha-j+1)}{j!},$$

definit în formula (1.9), pag. 14.

În cazul particular $x = -q$, $\alpha = -n$, seria (2.19) devine

$$\begin{aligned} (1-q)^{-n} &= \sum_{j=0}^{\infty} \binom{n+j-1}{j} q^j = \sum_{j=0}^{\infty} \binom{n+j-1}{n-1} q^j \\ &= \sum_{k=n}^{\infty} \binom{k-1}{n-1} q^{k-n}, \end{aligned}$$

de unde

$$\sum_{k=n}^{\infty} f_X(k) = p^n (1-q)^{-n} = 1$$

și prin urmare v.a. este definită corect.

V.a. binomială cu exponent negativ poate fi generată astfel: considerăm un experiment și fie A un eveniment care poate avea loc ca și rezultat al acestuia. Fie $p = P(A) > 0$. Repetăm experimentul independent, până când evenimentul A are loc de n ori, după care ne oprim. X este variabila aleatoare care reprezintă numărul de experimente efectuate până când A s-a realizat de n ori. Valorile posibile pentru X sunt într-adevăr $n, n+1, n+2, \dots$. Atunci, pentru un k ($k \geq n$) fixat, $f_X(k)$ poate fi calculată astfel:

$$f_X(k) = P(X = k) = P(A_{k-1}^{n-1} \cap A_k^1),$$

unde A_k^1 și A_{k-1}^{n-1} sunt evenimentele

$$\begin{aligned} A_k^1 &: \text{«}A \text{ are loc în al } k\text{-lea experiment}\text{»}, \\ A_{k-1}^{n-1} &: \text{«}A \text{ are loc de } n-1 \text{ ori în primele } k-1 \text{ experimente}\text{»}. \end{aligned}$$

Probabilitățile lor sunt, folosind schema binomială (vezi paragraful 2.5.1),

$$\begin{aligned} P(A_k^1) &= p, \\ P(A_{k-1}^{n-1}) &= \binom{k-1}{n-1} p^{n-1} (1-p)^{(k-1)-(n-1)} = \binom{k-1}{n-1} p^{n-1} q^{k-n}. \end{aligned}$$

Intrucât aceste evenimente sunt independente, formula (2.18) este demonstrată.

2.8.6 Distribuția urnelor lui Poisson

Definiția 2.18 Variabila aleatoare discretă X urmează **distribuția urnelor lui Poisson** cu parametrii $p_i \in (0, 1]$, $i \in \{1, \dots, n\}$ dacă funcția sa de probabilitate este $f_X : \{0, 1, \dots, n\} \rightarrow (0, 1]$,

$$f_X(k) = \tilde{p}_k, \quad (2.20)$$

pentru toți $k \in \{0, 1, \dots, n\}$, unde \tilde{p}_k este coeficientul lui x^k din polinomul

$$Q_n(x) = \prod_{i=1}^n (p_i x + q_i).$$

Aici $q_i = 1 - p_i$. Această v.a. este bine definită întrucât $f_X > 0$ și

$$\sum_{k=0}^n \tilde{p}_k = Q_n(1) = \prod_{i=1}^n (p_i + q_i) = 1.$$

Această distribuție este generată de schema urnelor lui Poisson (vezi paragraful 2.5.5).

2.9 Valoarea medie și varianța unei variabile aleatoare discrete

Fie X o v.a.d. cu valorile posibile $x_1, x_2, \dots, x_n, \dots$, și fie $p_k = P(X = x_k)$ și $i \in \mathbb{N}$.

Definiția 2.19 Numărul

$$\alpha_i = \sum_k x_k^i p_k,$$

dacă seria converge absolut, se numește **momentul de ordin i al lui X** . Numărul

$$\mu_i = \sum_k (x_k - \alpha_1)^i p_k,$$

se numește **momentul centrat de ordin i al lui X** .

De o importanță deosebită sunt momentele α_1 și μ_2 .

Definiția 2.20 Numărul

$$\alpha_1 = \sum_k x_k p_k,$$

se numește **expectața** sau **valoarea așteptată** sau **valoarea medie** sau **media** lui X și este notat cu $E(X)$ sau $M(X)$.

Valoarea medie este o măsură a centrului de distribuție, în sensul următor: dacă p_k sunt interpretate ca și mase asociate punctelor $x_k \in \mathbb{R}$, atunci $E(X)$ este centrul de masă al acestui sistem, pe axa reală:

$$\alpha_1 = E(X) = \frac{\sum x_k p_k}{\sum p_k}.$$

Exemplul 2.7 Dacă se consideră variabila aleatoare

$$X : \left(\begin{array}{c} k \\ 1/6 \end{array} \right)_{k \in \mathbb{N}_6}$$

reprezentând numărul feței apare când se aruncă un zar, atunci valoarea sa medie este

$$E(X) = \sum_{k=1}^6 k \cdot p_k = \frac{1}{6} \sum_{k=1}^6 k = 3.5.$$

Definiția 2.21 Numărul μ_2 se numește **varianța** lui X și este notat cu $\text{var}(X)$ sau $D^2(X)$. Astfel,

$$\text{var}(X) = \sum_k (x_k - \alpha_1)^2 p_k = E((X - E(X))^2).$$

Rădăcina pătrată a varianței este numită **deviația standard** sau **dispersia**, și este notată cu $\sigma(X)$ sau $D(X)$:

$$\sigma(X) = \sqrt{\text{var}(X)}.$$

Înainte de a prezenta proprietățile de bază ale expectanței și varianței, dăm următoarea definiție.

Definiția 2.22 Fie $X : \left(\begin{array}{c} x_k \\ p_k \end{array} \right)$, $k \in I$, o v.a.d., $i \in \mathbb{N}$ și $a \in \mathbb{R}$. Atunci variabilele aleatoare $a + X$, aX și X^i sunt definite, respectiv, ca și

$$\begin{aligned} a + X &: \left(\begin{array}{c} a + x_k \\ p_k \end{array} \right), \quad k \in I, \\ aX &: \left(\begin{array}{c} a x_k \\ p_k \end{array} \right), \quad k \in I, \\ X^i &: \left(\begin{array}{c} x_k^i \\ p_k \end{array} \right), \quad k \in I. \end{aligned}$$

Propoziția 2.6 Expectanța și varianța au următoarele proprietăți:

P1. $E(a) = a$, dacă a este v.a. constantă $a : \left(\begin{array}{c} a \\ 1 \end{array} \right)$.

P2. $E(aX) = aE(X)$.

P3. $E(X + Y) = E(X) + E(Y)$; Într-adevăr, dacă $Z = X + Y$,

$$X : \left(\begin{array}{c} x_k \\ p_k \end{array} \right), \quad k \in \mathbb{N}_n, \quad Y : \left(\begin{array}{c} y_k \\ r_k \end{array} \right), \quad k \in \mathbb{N}_m, \quad \text{atunci } Z : \left(\begin{array}{c} x_i + y_j \\ p_{ij} \end{array} \right),$$

cu $p_{ij} = P(X = x_i, Y = y_j)$, $i \in \mathbb{N}, j \in \mathbb{N}_m$. Astfel,

$$E(Z) = \sum_{i=1}^n \sum_{j=1}^m (x_i + y_j) p_{ij} = \sum_{i=1}^n x_i \sum_{j=1}^m p_{ij} + \sum_{j=1}^m y_j \sum_{i=1}^n p_{ij}.$$

Dar

$$\begin{aligned}\sum_{j=1}^m p_{ij} &= P(Z = x_i + y_1) + \dots + P(Z = x_i + y_m) \\ &= P(Z = x_i + y_1 \cup \dots \cup Z = x_i + y_m) \\ &= P(X = x_i) = p_i,\end{aligned}$$

pentru toți $i \in \mathbb{N}_n$. Analog calculăm $\sum_{i=1}^n p_{ij} = r_j$, pentru orice $j \in \mathbb{N}_m$, de unde concluzia

$$E(X + Y) = \sum_{i=1}^n x_i p_i + \sum_{j=1}^m y_j r_j = E(X) + E(Y).$$

P4. $E(aX + bY) = aE(X) + bE(Y), \quad E\left(\sum_{i=1}^n a_i X_i\right) = \sum_{i=1}^n a_i E(X_i)$

P5. $E(X - E(X)) = 0.$

P6. Dacă $X \geq 0$, atunci $E(X) \geq 0.$

P6'. Dacă $X \geq Y$, atunci $E(X) \geq E(Y).$

P7. $\text{var}(X) = \alpha_2 - \alpha_1^2 = E(X^2) - (E(X))^2$. Intr-adevăr,

$$\begin{aligned}\text{var}(X) &= E((X - E(X))^2) \\ &= E(X^2 - 2E(X) \cdot X + (E(X))^2) \\ &= E(X^2) - 2E(X) \cdot E(X) + (E(X))^2 \\ &= E(X^2) - (E(X))^2.\end{aligned}$$

P8. $\text{var}(aX + b) = a^2 \text{var}(X)$. Intr-adevăr,

$$\begin{aligned}\text{var}(aX + b) &= E((aX + b - E(aX + b))^2) \\ &= E((aX + b - aE(X) - b)^2) \\ &= E(a^2(X - E(X))^2) \\ &= a^2 \text{var}(X).\end{aligned}$$

Consecințe:

$$\begin{aligned}\text{var}(b) &= 0, \\ \text{var}(X + b) &= \text{var}(X), \\ \text{var}(aX) &= a^2 \text{var}(X).\end{aligned}$$

P9. Dacă X, Y sunt v.a. independente, $E(X \cdot Y) = E(X)E(Y).$

Intr-adevăr, fie

$$X : \begin{pmatrix} x_k \\ p_k \end{pmatrix}, \quad k \in \mathbb{N}_n, \quad Y : \begin{pmatrix} y_k \\ r_k \end{pmatrix}, \quad k \in \mathbb{N}_m. \text{ Atunci } X \cdot Y : \begin{pmatrix} x_i y_j \\ p_{ij} \end{pmatrix},$$

cu

$$p_{ij} = P(X = x_i \cap Y = y_j) = P(X = x_i) \cdot P(Y = y_j) = p_i r_j.$$

Atunci

$$\begin{aligned} E(X \cdot Y) &= \sum_{i=1}^n \sum_{j=1}^m x_i y_j p_{ij} = \sum_{i=1}^n \sum_{j=1}^m x_i y_j p_i r_j \\ &= \left(\sum_{i=1}^n x_i p_i \right) \left(\sum_{j=1}^m y_j r_j \right) = E(X)E(Y). \end{aligned}$$

P9'. Dacă X_i , $i \in \mathbb{N}_s$ sunt v.a.d. total independente, atunci

$$E \left(\prod_{i=1}^s X_i \right) = \prod_{i=1}^s E(X_i).$$

Observația 2.6 Dacă două v.a.d. nu sunt independente, proprietatea **P9** nu are loc în general.

Deviația standard și varianța sunt o măsură a împrăstierii unei distribuții în jurul valorii medii.

2.10 Covarianța

Cum am văzut, este important să calculăm varianța unei sume de v.a., cunoscând varianțele termenilor sumei. Pentru aceasta, avem nevoie câteva rezultate preliminare. Începem cu următoarea definiție.

Definiția 2.23 Fie X, Y două v.a. astfel încât $E(X), E(Y)$, $E(X \cdot Y)$, $\text{var}(X)$ și $\text{var}(Y)$ există. **Covarianța** lui X și Y este definită prin

$$\text{cov}(X, Y) = E((X - E(X)) \cdot (Y - E(Y))). \quad (2.21)$$

Propoziția 2.7 Covarianța are următoarele proprietăți:

P1. $\text{cov}(X, X) = \text{var}(X)$.

P2. $\text{cov}(X, Y) = E(X \cdot Y) - E(X)E(Y)$.

P2'. Dacă X și Y sunt independente, atunci $\text{cov}(X, Y) = 0$.

P3. Dacă X_i , $i \in \mathbb{N}_n$, atunci

$$\text{var} \left(\sum_{i=1}^n X_i \right) = \sum_{i=1}^n \text{var}(X_i) + 2 \sum_{1 \leq i < j \leq n} \text{cov}(X_i, X_j). \quad (2.22)$$

Demonstrație. Avem, folosind **P4** din propoziția 2.6 pag. 50,

$$\begin{aligned} \text{var} \left(\sum_{i=1}^n X_i \right) &= E \left(\left(\sum_{i=1}^n X_i - \sum_{i=1}^n E(X_i) \right)^2 \right) \\ &= E \left(\left(\sum_{i=1}^n (X_i - E(X_i)) \right)^2 \right). \end{aligned}$$

Folosind apoi formula

$$\left(\sum_{i=1}^n a_i \right)^2 = \sum_{i=1}^n a_i^2 + 2 \sum_{1 \leq i < j \leq n} a_i a_j,$$

pentru $a_i = X_i - E(X_i)$, se obține concluzia.

Definiția 2.24 Variabilele aleatoare X_1, X_2, \dots, X_n se numesc **necorelate** dacă

$$\text{cov}(X_i, X_j) = 0, \text{ pentru } i \neq j.$$

În acest moment putem să dăm un rezultat privind varianța sumei de variabile aleatoare. Mai precis, are loc următorul rezultat.

Propoziția 2.8 Sunt adevărate următoarele afirmații:

P4. V.a. total independente sunt necorelate.

P5. Dacă X_1, \dots, X_n sunt v.a. necorelate, atunci

$$\text{var}(X_1 + X_2 + \dots + X_n) = \text{var}(X_1) + \text{var}(X_2) + \dots + \text{var}(X_n).$$

Demonstrație Al doilea termen din membrul drept al egalității (2.22) se anulează.

P6. $\text{var}(X + Y) = \text{var}(X) + \text{var}(Y)$ dacă și numai dacă $\text{cov}(X, Y) = 0$.

2.11 Valori medii și varianțe pentru câteva variabile aleatoare discrete

Distribuția binomială

Teorema 2.9 Dacă X este a v.a.d. având o distribuție binomială dată în (2.13), atunci

$$E(X) = np, \text{ var}(X) = npq.$$

Demonstrație.

$$E(X) = \sum_{k=0}^n k \binom{n}{k} p^k q^{n-k}.$$

Pentru a calcula această sumă, derivăm formula binomială în raport cu x

$$(q + px)^n = \sum_{k=0}^n \binom{n}{k} p^k x^k q^{n-k},$$

obținând

$$n(q + px)^{n-1} p = \sum_{k=1}^n \binom{n}{k} p^k q^{n-k} k x^{k-1}. \quad (2.23)$$

Pentru $x = 1$, avem

$$E(X) = n(p + q)^{n-1} p = np.$$

Apoi, folosind proprietatea P7 din propoziția 2.6,

$$\text{var}(X) = E(X^2) - (E(X))^2, \quad (2.24)$$

unde

$$E(X^2) = \sum_{k=0}^n k^2 \binom{n}{k} p^k q^{n-k}.$$

Dacă înmulțim identitatea (2.23) cu x , o derivăm (în raport cu x) și apoi înlocuim $x = 1$, obținem

$$E(X^2) = np(1 + (n-1)p).$$

înlocuind în (2.24), în final obținem concluzia

$$\text{var}(X) = np(1 + np - p) - n^2p^2 = npq.$$

■

Distribuția hipergeometrică

Teorema 2.10 *Dacă X este o v.a.d. având o distribuție hipergeometrică dată în (2.14), atunci*

$$E(X) = np, \text{ var}(X) = npq \frac{a+b-n}{a+b-1}, \text{ cu } p = \frac{a}{a+b}, q = \frac{b}{a+b}.$$

Demonstrație.

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

Întrucât

$$k \binom{a}{k} = a \binom{a-1}{k-1}, \quad (2.25)$$

putem scrie

$$E(X) = \frac{1}{\binom{a+b}{n}} \sum_{k=1}^n a \binom{a-1}{k-1} \binom{b}{n-k} = a \frac{\binom{a+b-1}{n-1}}{\binom{a+b}{n}},$$

unde în ultima egalitate am folosit identitatea Vandermonde (2.15). După simplificări, are loc concluzia.

Pentru varianță mai întâi evaluăm

$$\begin{aligned} E(X^2) &= \frac{1}{\binom{a+b}{n}} \sum_{k=1}^n k^2 \binom{a}{k} \binom{b}{n-k} \\ &= \frac{1}{\binom{a+b}{n}} \left(\sum_{k=1}^n k \binom{a}{k} \binom{b}{n-k} + \sum_{k=2}^n k(k-1) \binom{a}{k} \binom{b}{n-k} \right). \end{aligned}$$

Folosind (2.25) și relația

$$k(k-1) \binom{a}{k} = a(a-1) \binom{a-2}{k-2}. \quad (2.26)$$

din nou dintr-o identitate de tip Vandermonde, obținem

$$\begin{aligned} E(X^2) &= \frac{a}{\binom{a+b}{n}} \binom{a+b-1}{n-1} + \frac{a(a-1)}{\binom{a+b}{n}} \binom{a+b-2}{n-2} \\ &= \frac{n}{a+b} + n(n-1) \frac{a(a-1)}{(a+b)(a+b-1)}. \end{aligned}$$

Înlocuind acum $E(X)$ și $E(X^2)$ în formula (2.24) și efectuând calculele, obținem concluzia. ■

Distribuția Poisson

Teorema 2.11 *Dacă X este o v.a.d. având o distribuție Poisson dată în (2.16), atunci*

$$E(X) = \lambda, \quad \text{var}(X) = \lambda.$$

Demonstrație.

$$E(X) = e^{-\lambda} \sum_{k=0}^{\infty} k \frac{\lambda^k}{k!} = \lambda e^{-\lambda} \sum_{k=1}^{\infty} k \frac{\lambda^{k-1}}{(k-1)!} = \lambda e^{-\lambda} e^{\lambda} = \lambda.$$

Apoi evaluăm $E(X^2)$, obținând

$$\begin{aligned} E(X^2) &= e^{-\lambda} \sum_{k=0}^{\infty} k^2 \frac{\lambda^k}{k!} = \lambda e^{-\lambda} \sum_{k=1}^{\infty} k \frac{\lambda^{k-1}}{(k-1)!} \\ &= \lambda e^{-\lambda} \left(\sum_{k=1}^{\infty} (k-1) \frac{\lambda^{k-1}}{(k-1)!} + \sum_{k=1}^{\infty} \frac{\lambda^{k-1}}{(k-1)!} \right) \\ &= \lambda^2 e^{-\lambda} \sum_{k=2}^{\infty} \frac{\lambda^{k-2}}{(k-2)!} + \lambda e^{-\lambda} \sum_{k=1}^{\infty} \frac{\lambda^{k-1}}{(k-1)!} \\ &= \lambda^2 e^{-\lambda} e^{\lambda} + \lambda e^{-\lambda} e^{\lambda} = \lambda(\lambda + 1). \end{aligned}$$

În final, $\text{var}(X) = E(X^2) - (E(X))^2 = \lambda(\lambda + 1) - \lambda^2 = \lambda$. ■

Distribuția geometrică

Teorema 2.12 *Dacă X este o v.a.d. având o distribuție geometrică dată în (2.17), atunci*

$$E(X) = \frac{1}{p}, \quad \text{var}(X) = \frac{q}{p^2}.$$

Demonstrație.

$$E(X) = p \sum_{k=1}^{\infty} k q^{k-1} = p \left(\frac{1}{1-q} \right)'_q = \frac{p}{(1-q)^2} = \frac{1}{p}.$$

$$E(X^2) = p \sum_{k=1}^{\infty} k^2 q^{k-1}.$$

Pentru a evalua această sumă, considerăm seria de puteri

$$\sum_{k=1}^{\infty} k q^{k-1} = \frac{1}{(1-q)^2},$$

o înmulțim cu q și apoi derivăm egalitatea în raport cu q . Obținem

$$\sum_{k=1}^{\infty} k^2 q^{k-1} = \left(\frac{q}{(1-q)^2} \right)'_q = \frac{1+q}{(1-q)^3}.$$

Prin înlocuire în $E(X^2)$ obținem concluzia. ■

Distribuția binomială negativă

Teorema 2.13 *Dacă X este o v.a.d. având o distribuție binomială negativă, dată în (2.18), atunci*

$$E(X) = \frac{n}{p}, \quad \text{var}(X) = \frac{nq}{p^2}.$$

Demonstrație. În acest caz, dacă folosim metodele anterioare, calculele se complică destul de mult. De aceea vom folosi o altă metodă pentru calculul expectanței și dispersiei. Mai precis, vom scrie v.a. X ca și o sumă de variabile aleatoare total independente ale căror valori medii și varianțe pot fi calculate mai ușor.

Deci, considerăm următoarea variabilă aleatoare:

$$\begin{aligned} X_1 &= \text{nr. de experimente efectuate până la prima apariție a lui } A, \\ X_k &= \text{nr. of experimente efectuate între a } (k-1)\text{-a} \\ &\quad \text{și a } k\text{-a apariție a lui } A, \end{aligned}$$

pentru $k = 2, 3, \dots, n$. Aceste v.a. sunt total independente și au o distribuție geometrică cu același parametru p .

Conform teoremei 2.17 avem

$$E(X_k) = \frac{1}{p}, \quad \text{var}(X_k) = \frac{q}{p^2},$$

pentru orice $k \in \mathbb{N}_n$.

Intrucât valoarea medie a sumei de v.a. este egală cu suma valorilor medii (vezi **P9'** pag. 52), avem

$$E(X) = E(X_1) + \dots + E(X_n) = nE(X_1) = \frac{n}{p},$$

și astfel formula pentru $E(X)$ este demonstrată.

Pentru varianță avem

$$\text{var}(X) = \text{var}(X_1) + \dots + \text{var}(X_n) = n \cdot \text{var}(X_1) = \frac{nq}{p^2},$$

datorită faptului că v.a. X_1, X_2, \dots, X_n sunt total independente (vezi formula (2.8)). ■

Distribuția urnelor lui Poisson

Teorema 2.14 *Dacă X este o v.a.d. având distribuția urnelor lui Poisson, dată în definiția 2.18, atunci*

$$E(X) = \sum_{k=1}^n p_k, \quad \text{var}(X) = \sum_{k=1}^n p_k q_k.$$

Demonstrație. În acest caz metodele directe duc din nou la calcule foarte complicate, deci vom folosi aceeași idee de a scrie X ca și sumă de n variabile aleatoare total independente.

Deci, fie X_k ($1 \leq k \leq n$) v.a. reprezentând numărul de bile albe extrase din urna U_k . Atunci X_k sunt total independente și X poate fi scris ca și

$$X = X_1 + X_2 + \dots + X_n.$$

Variabila aleatoare X_k are distribuția

$$X_k : \begin{pmatrix} 1 & 0 \\ p_k & q_k \end{pmatrix}.$$

Un calcul simplu arată că

$$E(X_k) = p_k, \quad E(X_k^2) = p_k, \quad \text{var}(X_k) = p_k q_k, \quad \text{pentru } k \in \mathbb{N}_n,$$

de unde

$$\begin{aligned} E(X) &= \sum_{i=1}^n E(X_k) = \sum_{i=1}^n p_k, \\ \text{var}(X) &= \sum_{i=1}^n \text{var}(X_k) = \sum_{i=1}^n p_k q_k. \end{aligned}$$

În final să observăm că, în cazul când urnele sunt identice ($p_k = p$), obținem, așa cum era de așteptat, expectanța și variația distribuției binomiale. ■

2.12 Inegalitatea lui Markov. Teorema lui Chebâșev

Propoziția 2.15 (Inegalitatea lui Markov) ⁸Fie X o variabilă aleatoare și fie $a > 0$. Atunci

$$P(|X| \geq a) \leq \frac{E(|X|^2)}{a^2}. \quad (2.27)$$

Demonstrație. Fie Z_a o v.a.

$$Z_a = \begin{cases} 1, & \text{dacă } |X| \geq a, \\ 0, & \text{în rest.} \end{cases}$$

Această v.a. satisface inegalitatea $|X|^2 \geq a^2 \cdot Z_a$, din care rezultă că

$$E(|X|^2) \geq a^2 E(Z_a) = a^2 P(|X| \geq a).$$

■

Următorul rezultat, cunoscut sub numele de teorema lui Cebâșev⁹, poate fi folosită pentru determinarea unei margini inferioare pentru probabilitatea ca o variabilă aleatoare Y să aparțină unui interval $[\mu - k\sigma, \mu + k\sigma]$ din jurul valorii medii μ .

⁸Andrey Markov (1856-1922), matematician rus.

⁹Pafnuty Lvóvich Cebâșev (1821-1894), matematician rus.

Teorema 2.16 (Cebâșev) Fie Y o variabilă aleatoare cu valoarea medie μ și varianța σ^2 finite. Atunci, pentru orice constantă $\varepsilon_1 > 0$, avem

$$P(|Y - \mu| \geq \varepsilon_1 \sigma) \leq \frac{1}{\varepsilon_1^2}. \quad (2.28)$$

Demonstrație. Dacă în inegalitatea (2.27) se ia $X = Y - E(Y)$ și $a = \varepsilon_1 \sigma$, concluzia se obține imediat. ■

Inegalitatea (2.28) se poate scrie de asemenea ca și

$$P(|Y - \mu| < \varepsilon_1 \sigma) \geq 1 - \frac{1}{\varepsilon_1^2},$$

sau, dacă se notează $\varepsilon = \sigma \varepsilon_1$,

$$\begin{aligned} P(|Y - E(Y)| \geq \varepsilon) &\leq \frac{\text{var}(Y)}{\varepsilon^2}, \text{ sau} \\ P(|Y - E(Y)| < \varepsilon) &\geq 1 - \frac{\text{var}(Y)}{\varepsilon^2}. \end{aligned} \quad (2.29)$$

Exemplul 2.8 Numărul de clienți care intră într-un magazin într-o zi, notat cu Y , a fost observat pentru o perioadă lungă de timp și s-a găsit că există în medie de 20 clienți, cu o deviație standard de 2 clienți¹⁰. Probabilitatea distribuției lui Y este necunoscută. Ce se poate spune despre probabilitatea ca a doua zi Y să fie între 16 și 24?

Soluție. Trebuie să găsim $P(16 < Y < 24)$. Pentru orice $\varepsilon > 0$, inegalitatea (2.29) se poate scrie ca și

$$P(\mu - \varepsilon < Y < \mu + \varepsilon) \geq 1 - \frac{\sigma^2}{\varepsilon^2}.$$

Pentru $\mu = 20$ și $\sigma = 2$, rezultă că $\mu - \varepsilon = 16$ și $\mu + \varepsilon = 24$ dacă $\varepsilon = 4$. Astfel,

$$P(16 < Y < 24) \geq 1 - \frac{2^2}{4^2} = \frac{3}{4}.$$

Aceasta înseamnă că totalul clienților de a doua zi va fi între 16 și 24 cu o probabilitate de cel puțin $3/4$.

Să observăm că, dacă σ ar fi fost egal cu 1,

$$P(16 < Y < 24) \geq 1 - \frac{1}{4^2} = \frac{15}{16}.$$

Deci valoarea lui σ are un efect considerabil asupra probabilităților asociate cu intervale.

2.13 Legile numerelor mari

Legile numerelor mari (LNM), numite și *teoremele de aur ale calculului probabilităților*, sunt teoreme care descriu stabilitatea pe termen lung a unei variabile aleatoare. De exemplu, când se aruncă un zar de un anumit număr de ori, media punctelor va fi stabilă în jurul valorii medii 3.5 (vezi exemplul 2.7, pag. 50), atunci când numărul de aruncări devine mare.

¹⁰Modul de estimare al deviației standard după multe observații este subiectul statisticii matematice, netratat în acest material.

Forme ale LNM

Legile numerelor mari există în două forme: **legea slabă a numerelor mari (LSNM)** și **legea tare a numerelor mari (LTNM)**. Ambele afirmă că media de selecție converge către valoarea medie, diferența dintre ele fiind tipul de convergență.

Definiția 2.25 *Un șir $(X_n)_{n \in \mathbb{N}}$ de variabile aleatoare **converge în probabilitate** către o variabilă aleatoare X (și notăm $X_n \xrightarrow{p} X$) dacă*

$$\lim_{n \rightarrow \infty} P(|X_n - X|) < \varepsilon = 1, \text{ pentru orice } \varepsilon > 0. \quad (2.30)$$

*Un șir $(X_n)_{n \in \mathbb{N}}$ de variabile aleatoare **converge aproape sigur** la o variabilă aleatoare X (și notăm $X_n \xrightarrow{a.s.} X$) dacă*

$$P\left(\lim_{n \rightarrow \infty} X_n = X\right) = 1, \text{ pentru orice } \varepsilon > 0. \quad (2.31)$$

Se poate arăta că, dacă $X_n \xrightarrow{a.s.} X$, atunci $X_n \xrightarrow{p} X$, dar implicația inversă nu este adevărată în general. Totuși, se poate arăta că, dacă $X_n \xrightarrow{p} X$, atunci există un subșir $(X_{n_k})_{k \in \mathbb{N}}$ astfel încât $X_{n_k} \xrightarrow{a.s.} X$. Din aceste motive, convergența în probabilitate este numită convergență **slabă**, în timp ce convergența aproape sigură este cunoscută cu numele de convergență **tare**.

Definiția 2.26 (Legea slabă a numerelor mari) *Un șir $(X_n)_{n \in \mathbb{N}}$ de variabile aleatoare cu $E(X_n) < \infty$ pentru orice $n \in \mathbb{N}$ **se supune LSNM** dacă*

$$\begin{aligned} & \frac{1}{n} \sum_{k=1}^n (X_k - E(X_k)) \xrightarrow{p} 0, \quad \text{adică} \\ & \lim_{n \rightarrow \infty} P\left(\frac{1}{n} \sum_{k=1}^n (X_k - E(X_k)) < \varepsilon\right) = 1, \quad \forall \varepsilon > 0. \end{aligned} \quad (2.32)$$

În continuare dăm câteva teoreme referitoare la LSNM.

Teorema 2.17 (LSNM, Markov) *Fie $(X_n)_{n \in \mathbb{N}}$ un șir de variabile aleatoare satisfăcând următoarea condiție (cunoscută ca și condiția lui Markov):*

$$\lim_{n \rightarrow \infty} \frac{1}{n^2} \text{var}(X_1 + X_2 + \dots + X_n) = 0.$$

Atunci $(X_n)_{n \in \mathbb{N}}$ se supune LSNM.

Demonstrație. Pentru $n \in \mathbb{N}$ considerăm media

$$\bar{X}_n = \frac{1}{n} (X_1 + X_2 + \dots + X_n).$$

Atunci, avem

$$\begin{aligned} P\left(\frac{1}{n} \sum_{k=1}^n X_k - E(X_k) \geq \varepsilon\right) & \leq P\left(\left|\frac{1}{n} \sum_{k=1}^n X_k - E(X_k)\right| \geq \varepsilon\right) \\ & = P(|\bar{X}_n - E(\bar{X}_n)| \geq \varepsilon) \\ & \leq \frac{\text{var}(\bar{X}_n)}{\varepsilon^2} = \frac{\text{var}(X_1 + \dots + X_n)}{n^2 \varepsilon^2}. \end{aligned}$$

În ultima egalitate am folosit inegalitatea lui Cebâșev dată în teorema 2.16, pentru $Y = \overline{X}_n$. Dacă facem $n \rightarrow \infty$ se obține

$$\lim_{n \rightarrow \infty} P \left(\frac{1}{n} \sum_{k=1}^n X_k - E(X_k) \geq \varepsilon \right) = 0, \text{ pentru orice } \varepsilon > 0,$$

care este echivalent cu (2.32), de unde deducem că șirul $(X_n)_{n \in \mathbb{N}}$ se supune LSNM. ■

Teorema 2.18 (LSNM, Cebâșev) *Fie $(X_n)_{n \in \mathbb{N}}$ un șir de variabile aleatoare independente două câte două, pentru care există constanta reală $0 < M < \infty$ astfel încât*

$$\text{var}(X_n) \leq M, \text{ pentru orice } n \in \mathbb{N}.$$

Atunci șirul $(X_n)_{n \in \mathbb{N}}$ se supune LSNM.

Demonstrație. Concluzia rezultă imediat din faptul că

$$\frac{1}{n^2} \text{var}(X_1 + \dots + X_n) = \frac{1}{n^2} (\text{var}(X_1) + \dots + \text{var}(X_n)) \leq \frac{M}{n}$$

și din teorema precedentă. ■

Teorema 2.19 (LSNM, Poisson) *Dacă într-un șir de repetări independente ale unui experiment probabilitatea ca un eveniment A să aibă loc la fiecare încercare este p , atunci pentru orice $\varepsilon > 0$ avem*

$$\lim_{n \rightarrow \infty} P \left(\left| \frac{m_n}{n} - p \right| < \varepsilon \right) = 1,$$

unde m_n este numărul de apariții ale evenimentului A în n încercări.

Teorema lui Poisson afirmă că șirul de frecvențe relative m_n/n ale apariției unui eveniment tinde în probabilitate către probabilitatea evenimentului.

Demonstrație. Pentru un experiment considerăm variabila aleatoare

$$X : \begin{pmatrix} 1 & 0 \\ p & q \end{pmatrix}, \text{ cu } q = 1 - p,$$

care ia valoarea 1 dacă evenimentul A are loc și 0 în caz contrar. Un calcul simplu arată că $E(X) = p$ și $\text{var}(X) = pq$. Atunci, pentru fiecare încercare considerăm o variabilă aleatoare $X_n = X$ și observăm că

$$\overline{X}_n = \frac{1}{n} (X_1 + X_2 + \dots + X_n) = \frac{m_n}{n} \text{ și } E(\overline{X}_n) = \frac{np}{n} = p.$$

Șirul de v.a. $(X_n)_{n \in \mathbb{N}}$ satisface condiția lui Markov:

$$\frac{1}{n^2} \sum_{i=1}^n \text{var}(X_i) = \frac{pq}{n} \longrightarrow 0, \text{ } n \rightarrow \infty,$$

prin urmare, aplicând teorema lui Markov concluzia rezultă imediat. ■

Definiția 2.27 (Legea tare a numerelor mari) *Un șir $(X_n)_{n \in \mathbb{N}}$ de variabile aleatoare, cu $E(X_n) < \infty$ pentru orice $n \in \mathbb{N}$, **se supune LTNM** dacă*

$$\begin{aligned} \frac{1}{n} \sum_{k=1}^n (X_k - E(X_k)) &\xrightarrow{a.s.} 0, \quad \text{adică} \\ P\left(\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{k=1}^n (X_k - E(X_k)) = 0\right) &= 1. \end{aligned} \quad (2.33)$$

Vom enunța fără demonstrație două rezultate privind LTNM.

Teorema 2.20 (Kolmogorov) *Dacă $(X_n)_{n \in \mathbb{N}}$ este un șir de variabile aleatoare independente satisfăcând*

$$\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} \text{var}(X_k) < \infty,$$

atunci

$$\frac{1}{n} \sum_{k=1}^n (X_k - E(X_k)) \xrightarrow{a.s.} 0,$$

adică șirul $(X_n)_{n \in \mathbb{N}}$ se supune LTNM.

Teorema 2.21 (Kolmogorov) *Fie $(X_n)_{n \in \mathbb{N}}$ un șir de variabile aleatoare independente și identic distribuite, astfel încât $E(X_n) = \mu < \infty$. Atunci*

$$\frac{1}{n} \sum_{k=1}^n X_k \xrightarrow{a.s.} \mu,$$

adică șirul $(X_n)_{n \in \mathbb{N}}$ se supune LTNM.

2.14 Probleme rezolvate

Problema 2.1 (Problema zilei de naștere) *Câte persoane trebuie să fie într-o sală ca să putem paria în mod favorabil (cu o probabilitate de succes mai mare decât 0.5) că există cel puțin două persoane care au aceeași zi de naștere?*

În teoria probabilităților această problemă se mai numește *paradoxul zilei de naștere*. Există 365 zile de naștere posibile, deci am fi tentați să credem că ar fi nevoie de jumătate din acest număr, 183. Rezultatul este oarecum surprinzător, de fapt numărul cerut fiind 23. Pentru 60 sau mai multe persoane, probabilitatea este mai mare decât 99%, totuși nu poate fi 100% decât dacă există 366 persoane.

Soluție. Fie p_n probabilitatea ca, într-o sală cu n persoane, să nu existe nici o zi de naștere care să se repete. Presupunem că pentru fiecare persoană există 365 zile de naștere posibile, toate egal probabile.¹¹ Atunci

$$p_n = \frac{365 \cdot 364 \cdot \dots \cdot (365 - n + 1)}{365^n}.$$

¹¹ Chiar dacă am presupus că cele 365 zile de naștere pot să cadă în orice zi, evidențele statistice arată că acest lucru nu este adevărat.

Pentru problema noastră, trebuie să găsim valoarea minimă a lui n pentru care $1 - p_n > 0.5$ (sau, echivalent $p_n < 0.5$). Evaluând $q_n = 1 - p_n$ pentru diferite valori ale lui n , găsim că

$n = 15$	$q_n = 25.29\%$	$n = 30$	$q_n = 70.63\%$
$n = 20$	$q_n = 41.14\%$	$n = 40$	$q_n = 89.12\%$
$n = 21$	$q_n = 44.36\%$	$n = 50$	$q_n = 97.03\%$
$n = 22$	$q_n = 47.56\%$	$n = 57$	$q_n = 99.01\%$
$n = 23$	$q_n = 50.72\%$	$n = 100$	$q_n = 99.99996\%$

Problema 2.2 (Aceeși zi de naștere cu tine) *Care este probabilitatea ca într-o sală cu n persoane cineva să aibă aceeași zi de naștere ca și tine?*

Soluție. Fie q_n probabilitatea evenimentului contrar A_n , ca nimeni din sală să nu aibă aceeași zi de naștere cu tine. Atunci A_n poate fi scris ca și un produs de n evenimente independente,

$$A_n = B_1 \cap B_2 \cap \cdots \cap B_n,$$

unde B_k ($1 \leq k \leq n$) este evenimentul «*persoana k nu are aceeași zi de naștere ca tine*». Probabilitatea fiecărui B_k este

$$P(B_k) = \frac{364}{365},$$

întrucât există 364 cazuri favorabile (toate zilele dintr-un an, cu excepția zilei tale) și 365 cazuri posibile. Prin urmare, probabilitatea cerută va fi

$$s_n = 1 - q_n = 1 - \left(\frac{364}{365}\right)^n.$$

Calculul arată că, pentru $n = 23$ persoane, probabilitatea este $s_{23} = 6.1\%$, și s_n începe să devină mai mare decât 0.5 pentru $n \geq 253$, care este semnificativ mai mare decât $365/2 = 182.5$.

Problema 2.3 (Partida neterminată) *Aceasta este o veche și clasică problemă din teoria probabilităților.*

Considerăm două echipe, A și B, presupuse de forțe egale. Ele sunt în competiție pentru un premiu, primit de prima echipă care câștigă n jocuri. După ce A a câștigat i jocuri și B a câștigat j jocuri ($i, j < n$), competiția este întreruptă în mod neașteptat și trebuie să decidem cum să se împartă premiul în bani.

Soluție. Această problemă a stârnit mult interes încă de la începuturile istoriei teoriei probabilităților. Lista următoare ne dă sugestii despre proporția pe care A trebuie să o primească.

$$\begin{aligned} \text{Pacioli (1494)} & : i/(i+j), \\ \text{Tartaglia (1556)} & : (n+i-j)/2n, \\ \text{Forestani (1603)} & : (2n-1+i-j)/(4n-2). \end{aligned}$$

Problema a fost rezolvată de către Fermat și Pascal în 1654, corespondența dintre aceștia care a condus în final la găsirea soluției problemei este prezentată în [5]. Ideea a fost ca fiecare echipă să primească o fracție din premiu egală cu probabilitatea ca aceasta să câștige jocurile rămase. Deci, numărul

de jocuri de care A are nevoie ca să câștige este $k_1 = n - i$ și numărul de jocuri de care B are nevoie ca să câștige este $k_2 = n - j$. Numărul maxim de jocuri care trebuie jucate în meciul rămas este $k_1 + k_2 - 1$.

Pentru a câștiga competiția în exact m jocuri, A trebuie să câștige ultimul joc și $k_1 - 1$ din primele $m - 1$ jocuri. Conform schemei binomiale (pag. 39), probabilitatea de a câștiga $k_1 - 1$ din cele $m - 1$ jocuri rămase este

$$\binom{m-1}{k_1-1} \left(\frac{1}{2}\right)^{m-1},$$

prin urmare probabilitatea ca A să câștige în m jocuri va fi $2^{-m} \binom{m-1}{k_1-1}$.

Posibilele valori pentru m sunt $k_1, k_1 + 1, \dots, k_1 + k_2 - 1$. În concluzie, probabilitatea ca A să câștige este

$$\sum_{m=k_1}^{k_1+k_2-1} \frac{1}{2^m} \binom{m-1}{k_1-1}.$$

Analog, probabilitatea ca B să câștige este

$$\sum_{m=k_2}^{k_1+k_2-1} \frac{1}{2^m} \binom{m-1}{k_2-1}.$$

Problema 2.4 (Experimente de capturare și recapturare) *Un*

biolog merge la un iaz și capturează $K = 60$ pești, îi marchează pe fiecare cu o pată de vopsea și le dă drumul înapoi în iaz. După o vreme se întoarce și capturează o altă mulțime de $n = 50$ pești, găsind $k = 12$ pești marcați și $n - k = 38$ pești nemarcați. Care este cea mai bună estimare a biologului despre mărimea populației de pești din iaz? (Presupunem că nici un pește nu intră sau iese din iaz între cele două vizite).

Soluție. Fie N numărul de pești din iaz. Probabilitate de a obține k pești marcați și $n - k$ nemarcați dintr-o selecție de n pești este (folosind schema hipergeometrică pag. 40)

$$p_N = \frac{\binom{K}{k} \binom{N-K}{n-k}}{\binom{N}{n}}.$$

Pentru a estima populația, alegem N astfel încât să maximizeze p_N . Un calcul simplu ne arată că

$$p_N = p_{N-1} \cdot \frac{N-K}{N-K-(n-k)} \cdot \frac{N-n}{N},$$

deci inegalitatea

$$\frac{p_N}{p_{N-1}} \geq 1$$

va fi echivalentă cu

$$N \leq \frac{Kn}{k}.$$

În concluzie, pentru

$$N = \left\lceil \frac{Kn}{k} \right\rceil$$

probabilitatea p_N va fi maximizată.

Să menționăm că, în cazul când $N = \frac{Kn}{k}$, avem egalitatea

$$\frac{K}{N} = \frac{k}{n},$$

care înseamnă că proporția de pești marcați din populație coincide cu proporția de pești marcați din selecție.

În exemplul nostru $N = \frac{60 \cdot 50}{12} = 250$, deci probabilitatea p_N este maximizată pentru $N = 250$.

Problema 2.5 5% din bărbați și 0.25% din femei au o formă de daltonism. Care este probabilitatea ca o persoană daltonistă să fie bărbat?

Soluție. Considerăm experimentul de a testa o persoană la întâmplare și fie evenimentele

$$\begin{aligned} A &: \text{«persoană testată este daltonistă»}, \\ B_1 &: \text{«persoană testată este femeie»}, \\ B_2 &: \text{«persoană testată este bărbat»}. \end{aligned}$$

Atunci $S = \{B_1, B_2\}$ este un (SCE) întrucât $B_1 \cap B_2 = \emptyset$ și $B_1 \cup B_2 = E$, E fiind evenimentul sigur «o persoană este testată». Deci, A are loc împreună cu unul (și numai unul) din evenimentele lui S . Conform formulei probabilității totale avem

$$P(A) = P(B_1) \cdot P(A|B_1) + P(B_2) \cdot P(A|B_2).$$

Probabilitățile implicate aici sunt

$$P(B_1) = \frac{1}{2}, \quad P(A|B_1) = \frac{5}{100}, \quad P(B_2) = \frac{1}{2}, \quad P(A|B_2) = \frac{0.25}{100},$$

prin urmare $P(A) = 2.62\%$.

Presupunem acum că experimentul de testare a fost realizat și evenimentul A a avut loc, adică persoana testată este daltonistă. Ni se cere probabilitatea $P(B_2|A)$, adică probabilitatea ca persoana să fie bărbat, știind deja că este daltonistă. Aceasta poate de asemenea fi interpretată ca și probabilitatea ca realizarea evenimentului A să fie datorată evenimentului B_2 din (SCE) S . Pentru a calcula probabilitatea cerută folosim formula lui Bayes

$$P(B_2|A) = \frac{P(B_1) \cdot P(A|B_1)}{P(A)} = 0.9523 \dots,$$

astfel probabilitatea ca o persoană daltonistă să fie bărbat este 95.23%.

Problema 2.6 (Genetică) O femeie are un frate bolnav de hemofilie și doi părinți care nu au hemofilie. Dacă ea are doi băieți sănătoși, care este probabilitatea ca ea să fie purtătoare?

Soluție. Este cunoscut faptul că hemofilia este cauzată de o genă recisivă h pe cromozomul X și că o femeie are ultima pereche de cromozomi XX , în timp ce un bărbat are XY . Un copil va moșteni un cromozom de la mamă și unul de la tată.

Intrucât fratele este bolnav, el trebuie să fie X^hY (unde X^h este de la mamă și Y de la tată). Din faptul că mama este sănătoasă, deducem că ea este X^hX^H . Aceasta pentru că gena sănătoasă H este dominantă peste gena recisivă h și deci, întrucât mama este purtătoare, ea are o genă bolnavă h . Tata trebuie să aibă prin urmare X^HY , pentru că dacă ar avea X^hY ar însemna că este bolnav.

În concluzie, părinții sunt X^hX^H și X^HY .

În această situație, copiii lor ar putea fi:

$$X^hX^H \quad X^HX^H \quad X^hY \quad X^HY \quad .$$

Deci, pentru o fiică, șansele să fie purtătoare sunt 50%. Dacă ea este purtătoare, atunci probabilitatea ca fiii săi să aibă boala este 50% (întrucât fiii săi pot fi sau X^hY , sau X^HY).

Deci, să considerăm următoarele evenimente:

B : «femeia este purtătoare »

A : «femeia are doi fii sănătoși »

Ni se cere probabilitatea $P(B|A)$, care poate fi calculată cu formula lui Bayes. Mai întâi folosim formula probabilității totale

$$P(A) = P(A|B)P(B) + P(A|\bar{B})P(\bar{B}) = \frac{1}{4} \cdot \frac{1}{2} + 1 \cdot \frac{1}{2} = \frac{5}{8}.$$

Probabilitatea $P(A|B)$ a fost calculată în felul următor: ea este X^hX^H , prin urmare un fiu poate fi sau X^hY (caz în care este bolnav), sau X^HX^H (caz în care este sănătos). Deci probabilitatea ca un fiu să fie sănătos este $1/2$.

Acum, formula lui Bayes ne dă

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)} = \frac{\frac{1}{8}}{\frac{5}{8}} = \frac{1}{5}.$$

Problema 2.7 Dacă cineva joacă la ruletă și pariază 1 € pe negru, atunci câștigă 1 € cu probabilitatea $\frac{18}{37}$ și pierde 1 € cu probabilitatea $\frac{19}{37}$.

- Care este valoarea așteptată a câștigului?
- Dacă ruleta este una americană, unde probabilitatea de câștig este $\frac{18}{38}$, în timp ce probabilitatea de a pierde este $\frac{20}{38}$, care este în acest caz valoarea așteptată a câștigului?
- O persoană joacă la ruleta americană și pariază 1 \$ pe negru de 19 ori. Care este valoarea așteptată a câștigului său?

Soluție.

- V.a. reprezentând câștigul este

$$X : \begin{pmatrix} 1 & -1 \\ \frac{18}{37} & \frac{19}{37} \end{pmatrix}.$$

Valoarea ei medie va fi

$$E(X) = 1 \cdot \frac{18}{37} + (-1) \cdot \frac{19}{37} = -0.0270.$$

Aceasta înseamnă că pierde în medie 2.70 cenți în fiecare joc.

b) În cazul al doilea, v.a. reprezentând câștigul este

$$Y : \begin{pmatrix} 1 & -1 \\ \frac{18}{38} & \frac{20}{38} \end{pmatrix}$$

și valoarea sa medie va fi

$$E(Y) = 1 \cdot \frac{18}{38} + (-1) \cdot \frac{20}{38} = -0.0526 \$.$$

(dublul celei din primul caz !!!).

c) Pentru $i \in \mathbb{N}_{19}$, fie Y_i v.a. reprezentând câștigul la al i -lea joc. Atunci $E(Y_i) = E(Y) = -\frac{2}{38}$, de unde expectanța cerută va fi

$$E(Y_1 + \dots + Y_{19}) = E(Y_1) + \dots + E(Y_{19}) = 19 \cdot \left(\frac{-2}{38} \right) = -1 \$.$$

Problema 2.8 *Un bărbat nu cumpără niciodată bilet de autobuz. El presupune că există o probabilitate de 0.05 de a fi prins. Prima amendă nu costă nimic, a doua costă 20 lei, și amenziile următoare costă 40 lei fiecare. În aceste ipoteze, care este costul așteptat a 100 de călătorii fără să plătească, în comparație cu costul a 100 bilete de autobuz, dacă biletul pentru o călătorie costă 2 lei?*

Soluție. Fie $p = 0.05$ probabilitatea să fie prins o dată. V.a. reprezentând suma datorată în cazul când nu plătește niciodată bilet este

$$X : \begin{pmatrix} 0 & 20 & 20 + 40 \cdot 1 & \dots & 20 + 40 \cdot k & \dots & 20 + 40 \cdot 98 \\ p_0 + p_1 & p_2 & p_3 & \dots & p_{k+2} & \dots & p_{100} \end{pmatrix},$$

unde $p_k = \binom{100}{k} p^k (1-p)^{100-k}$. Atunci

$$E(X) = \sum_{k=0}^{98} (20 + 40k) \binom{100}{k+2} p^{k+2} (1-p)^{98-k} = 140.98 \text{ lei},$$

care este mai mic decât suma cheltuită în cazul în care plătește fiecare călătorie (200 lei). Pentru a calcula suma, am folosit argumente similare cu cele din calculul valorii medii a distribuției binomiale.

Problema 2.9 *Presupunem că, atunci când cineva călătorește cu autobuzul, costul unei călătorii este 2 lei și că există o probabilitate p de a fi controlat. De fiecare dată când o persoană este prinsă fără bilet, amenda este 40 lei. Care este valoarea maximă a lui p pentru costul așteptat a 100 călătorii fără să plătească biletul este mai mică decât 200 lei (suma plătită în cazul în care cumpără de fiecare dată bilet)?*

Soluție. V.a. reprezentând suma plătită în cazul când nu plătește niciodată bilet este

$$X : \left(\binom{100}{k} p^k (1-p)^{100-k} \right), \quad k = 0, 1, \dots, 100,$$

prin urmare dacă se consideră v.a. $\tilde{X} = \frac{1}{40} X$, atunci \tilde{X} va avea o distribuție binomială. Conform teoremei 2.9, valoarea sa medie va fi $E(X) = 100p$, de unde $E(X) = 4000p$. Inegalitatea $4000p \leq 200$ implică

$$p_{\max} = \frac{200}{4000} = 0.05.$$

Capitolul 3

Teoria grafurilor

3.1 Introducere

3.1.1 Grafuri

În lumea reală, multe din situații pot fi descrise prin intermediul unei diagrame care constă dintr-o mulțime de puncte împreună cu segmente care unesc anumite perechi de puncte. De exemplu, punctele pot fi centre de comunicații și liniile pot fi legături, sau punctele pot fi persoane și segmentele pot uni perechi de prieteni. În asemenea diagrame suntem interesați în principal dacă două puncte sunt sau nu unite printr-un segment. O reprezentare matematică a situațiilor de acest tip ne conduce la conceptul de graf.

Definiția 3.1 Un **graf** G constă dintr-o mulțime (finită) V , ale cărei elemente se numesc **vârfuri**, și o mulțime E de perechi neordonate cu componentele în V , ale cărei elemente se numesc **muchii**. De obicei se scrie $G = (V, E)$ iar muchiile se notează $\{x, y\}$ sau mai scurt xy , unde $x, y \in V$.

Observația 3.1 Într-un graf, muchia xy coincide cu muchia yx și admitem ca muchie și o pereche xx .

Exemplul 3.1 $G = (V, E)$, cu

$$V = \{a, b, c, d, f\}, \quad E = \{ab, ad, bf, cd, df\}.$$

Definiția 3.2 Fie $G = (V, E)$ un graf.

Spunem că două vârfuri $x, y \in V$ sunt **adiacente** (sau **vecine**) dacă xy este muchie.

O muchie este o mulțime de unul sau două vârfuri distincte numite **extremități**.

Dacă un vârf v este extremitate pentru muchia e , atunci spunem că v este **incident** muchiei e și despre e spunem că este **incidentă** vârfului v .

O **bucă** într-un vârf v este o muchie care unește vârful v cu el însuși. O **muchie proprie** este o muchie care unește două extremități diferite.

Un graf fără bucle se numește **graf simplu**.

Graful $G' = (V', E')$ este numit **subgraf** al lui G dacă $V' \subseteq V$ și $E' \subseteq E$.

Dacă $V_1 \subseteq V$, graful format de vârfurile lui V_1 și toate muchiile din E între aceste vârfuri este numit subgraf **indus de mulțimea vârfurilor** V_1 .

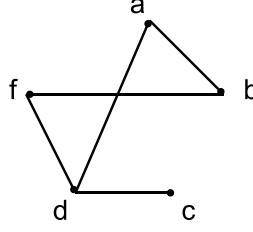


Figura 3.1: Reprezentarea grafului din exemplul 3.1.

Un **subgraf de acoperire** al grafului G este un subgraf care conține toate vârfurile lui G .

Graful G se numește **complet** dacă $xy \in E$ pentru orice $x, y \in V$, $x \neq y$.

Observația 3.2 Fie $G = (V, E)$ un graf și fie \mathcal{G} mulțimea tuturor subgrafurilor grafului G . Pe \mathcal{G} se poate defini următoarea relație:

$$H_1 \preceq H_2 \iff H_1 \text{ subgraf al lui } H_2. \quad (3.1)$$

Se poate demonstra cu ușurință că (\mathcal{G}, \preceq) este mulțime parțial ordonată. Pentru definiția relației de ordine parțiale și mai multe detalii despre relații, a se vedea anexa A, pag. 138.

3.1.2 Gradul unui vârf. Teorema lui Euler pentru grafuri

Definiția 3.3 **Gradul** unui vârf v într-un graf simplu $G = (V, E)$ este numărul de muchii ale lui G incidente în v . Acesta va fi notat cu $\delta(v)$ sau $\deg(v)$.

Intr-un graf cu bucle, se consideră că fiecare buclă în v aduce o contribuție egală cu 2 la $\delta(v)$.

Pentru graful din exemplul 3.1, gradele sunt

$$\delta(a) = 2, \delta(b) = 2, \delta(c) = 1, \delta(d) = 3, \delta(f) = 2.$$

Relația dintre suma gradelor și numărul de muchii este dat de teorema lui Euler¹.

Teorema 3.1 (Euler) Suma valorilor lui $\delta(v)$, luată după toate vârfurile unui graf $G = (V, E)$, este egală cu dublul numărului de muchii:

$$\sum_{v \in V} \delta(v) = 2|E|.$$

Demonstrație. O muchie $e = xy$ aduce o contribuție 2 la gradul total: 1 la $\delta(x)$ și 1 la $\delta(y)$. ■

O consecință imediată a teoremei lui Euler este următorul corolar:

Corolarul 3.2 Intr-un graf, există un număr par de vârfuri care au grad impar.

¹Leonhard Euler (1707-1783), matematician și fizician elvețian.

3.1.3 Lanțuri, lanțuri simple, lanțuri elementare, cicluri în grafuri

Frecvent folosim grafurile ca și modele pentru situații practice în care sunt implicate trasee. În aceste situații, vârfurile reprezintă orașe/joncțiuni și fiecare muchie reprezintă un drum sau o altă formă de comunicație. Prin urmare avem nevoie să introducem câteva noțiuni suplimentare, pentru a putea manevra aceste situații practice.

Definiția 3.4 ²Fie $G = (V, E)$ un graf.

Un **lanț** de **lungime** k ($k \in \mathbb{N}$) în G este un șir

$$\omega = \langle v_0, e_1, v_1, e_2, v_2, \dots, e_k, v_k \rangle,$$

ai cărui termeni sunt alternativ vârfuri și muchii, astfel încât, pentru $1 \leq i \leq k$, extremitățile lui e_i sunt v_{i-1} și v_i . Spunem că ω este un lanț între v_0 și v_k sau un (v_0, v_k) – lanț. Lanțul ω este deseori reprezentat doar prin șirul de vârfuri $\langle v_0, v_1, \dots, v_k \rangle$ sau prin șirul de muchii $\langle e_1, e_2, \dots, e_k \rangle$.

Un lanț ω_1 în G este numit un **sublanț** al lui ω dacă ω_1 este un șir $\omega_1 = \langle v_i, e_{i+1}, v_{i+1}, \dots, e_j, v_j \rangle$ de termeni consecutivi ai lui ω .

Un **lanț simplu** în G este un lanț lanț în care toate muchiile (dar nu neapărat și vârfurile) sunt distincte.

Un **lanț elementar** în G este un lanț în care toate vârfurile (și implicit toate muchiile) sunt distincte, cu excepția primului și al ultimului, care pot să coincidă.

Vom folosi de asemenea cuvintele lanț/lanț simplu/lanț elementar pentru a numi un graf sau subgraf ale cărui vârfuri și muchii aparțin lanțului respectiv.

Un lanț/lanț simplu/lanț elementar în G este numit **închis** dacă primul și ultimul vârf coincid. Un lanț/lanț simplu/lanț elementar care nu este închis se numește **deschis**.

Un **ciclu** în G este un lanț simplu închis. Un **ciclu elementar** în G este un lanț elementar închis din G .

Distanța dintre două vârfuri $x, y \in V$, notată $d(x, y)$ sau $\text{dist}(x, y)$, este minimul lungimilor tuturor lanțurilor din G , dintre x și y .

Definiția 3.5 Fie $\omega_1 = \langle x_1, x_2, \dots, x_k \rangle$, $\omega_2 = \langle y_1, y_2, \dots, y_p \rangle$ două lanțuri dintr-un graf $G = (V, E)$, astfel încât $x_k = y_1$. Lanțul

$$\omega_1 \cup \omega_2 = \langle x_1, x_2, \dots, x_k, y_2, \dots, y_p \rangle$$

se numește **reuniunea** sau **concatenarea** lanțurilor ω_1 și ω_2 .

Dacă un lanț ω conține un sublanț închis ω' , atunci ω poate fi **reduc** la un lanț notat $\omega - \omega'$, prin îndepărtarea din ω a vârfurilor și muchiilor din ω' , cu excepția primului și ultimului vârf.

²Noțiunile corespunzătoare din limba engleză sunt: walk = lanț, closed walk = lanț închis, trail = lanț simplu, path = lanț elementar, cycle = ciclu elementar, closed trail = ciclu

În continuare dăm câteva proprietăți ale lanțurilor unui graf.

Propoziția 3.3 *Orice lanț ω între două vârfuri distincte x, y este fie un lanț elementar, fie conține un sublanț închis al lui ω .*

Demonstrație. Dacă ω nu este un lanț elementar, atunci un subșir al lui ω cuprins între două vârfuri repetate definește un sublanț închis al lui ω . ■

Propoziția 3.4 *Dacă există lanț între x și y în G , atunci există de asemenea un lanț elementar între x și y în G .*

Demonstrație. Dacă lanțul ω nu este elementar, el conține un sublanț închis ω_1 . Lanțul $\omega - \omega_1$ este fie elementar, fie conține un sublanț ω_2 . În cazul al doilea, considerăm $(\omega - \omega_1) - \omega_2$ și repetăm procedura (un număr finit de ori), până când obținem un lanț elementar. ■

Propoziția 3.5 *Orice ciclu C conține un ciclu elementar.*

Demonstrație. Fie C^* un lanț închis de lungime minimă. Atunci C^* nu are nici un lanț închis propriu deoarece singurele sale vârfuri care se repetă sunt primul și ultimul. În concluzie, C^* este un ciclu elementar. ■

Observația 3.3 *Această proprietate nu este adevărată dacă C este doar un lanț închis.*

Pentru rezultatul următor, trebuie să introducem o nouă terminologie.

Definiția 3.6 *O mulțime de cicluri elementare care nu au muchii comune C_1, C_2, \dots, C_n este numită **descompunere** a lanțului simplu închis τ , dacă aceste cicluri elementare sunt sublanțuri pentru τ și dacă reuniunea mulțimilor muchiilor lor coincide cu mulțimea muchiilor din τ (cu alte cuvinte $\{C_1, C_2, \dots, C_n\}$ formează o partiție a mulțimii de muchii din τ).*

Propoziția 3.6 *Un lanț simplu închis poate fi descompus în cicluri elementare fără muchii în comun.*

Demonstrație. Demonstrăm prin inducție după k (k = numărul de muchii). Pentru $k = 1$, lanțul simplu este o buclă. Presupunem proprietatea adevărată pentru orice lanțuri simple închise cu $\leq m$ muchii și fie τ un lanț simplu închis cu $m + 1$ muchii. Conform propoziției 3.5, τ conține un ciclu elementar C . Atunci $\tau - C$ este un lanț simplu închis având $\leq m$ muchii. Din ipoteza inducției, $\tau - C$ poate fi descompus în cicluri elementare fără muchii comune și prin urmare τ poate fi scris ca și reuniunea acestor cicluri elementare disjuncte și C . ■

Definiția 3.7 (Operații cu grafuri) *Fie $G = (V, E)$ un graf.*

*Prin **ștergerea unei muchii** $e \in E$ din graful graf G , se obține un subgraf notat $G - e$, care conține toate vârfurile lui G și toate muchiile lui G , cu excepția lui e .*

*Prin **ștergerea unui vârf** $x \in V$ din graful G , se obține un subgraf notat $G - x$, care conține toate vârfurile lui G cu excepția lui x și toate muchiile lui G cu excepția celor incidente în x .*

Prin **adăugarea unei muchii** $e_1 = xy \notin E$ la graful G , se obține un alt graf, notat $G + e_1$, cu mulțimea vârfurilor $V \cup \{x\} \cup \{y\}$ și mulțimea muchiilor $E \cup \{e_1\}$.

3.1.4 Conectivitate în grafuri

Definiția 3.8 Fie $G = (V, E)$ un graf.

G se numește **conex** dacă, pentru oricare două vârfuri x, y , există un lanț între x și y .

O **componentă conexă** (sau simplu **componentă**) a lui G este un subgraf conex maximal³ S al lui G , în sensul că S nu este un subgraf propriu al nici unui subgraf conex al lui G .

O muchie $e \in E$ se numește **muchie tăietură** sau **bridge** dacă prin înlăturarea lui e crește numărul de componente conexe.

Când un graf de dimensiuni mici este dat printr-un desen, este destul de ușor să precizăm dacă este conex sau nu. Totuși, când un graf are dimensiuni mari sau când este dat în alt mod (de exemplu prin matricea sa de incidență), avem nevoie de algoritmi eficienți pentru a decide dacă este conex sau nu. Un asemenea algoritm este prezentat în paragraful 3.2.1, pag. 81.

Următoarea teoremă afirmă că orice graf G care nu este conex poate fi descompus în componente conexe. Importanța sa este semnificativă deoarece multe proprietăți ale grafurilor pot fi stabilite pentru fiecare componentă conexă a sa. Din acest motiv, teoremele despre grafuri sunt deseori demonstrate doar pentru grafuri conexe.

Teorema 3.7 (de descompunere în subgrafuri conexe) Pentru orice graf $G = (V, E)$ există un număr (finit) de k subgrafuri conexe $G_i = (V_i, E_i)$, $i \in \mathbb{N}_k$, cu următoarele proprietăți:

1. $V_i \cap V_j = \emptyset$, $E_i \cap E_j = \emptyset$, pentru orice $i, j \in \mathbb{N}_k$, $i \neq j$,
2. $V = \bigcup_{i=1}^k V_i$, $E = \bigcup_{i=1}^k E_i$.

Demonstrație. Pe mulțimea de vârfuri V definim următoarea relație:

$$x \sim y \iff x = y \text{ sau există un } (x, y) - \text{lanț.} \quad (3.2)$$

Această relație este o relație de echivalență pe V , numită *relația de accesibilitate*. Ea induce o partiție $\{V_1, V_2, \dots, V_k\}$ pe mulțimea de vârfuri V (vezi anexa 3.9.4). Pentru fiecare $i \in \mathbb{N}_k$, notăm cu $G_i = (V_i, E_i)$ graful indus de submulțimea de vârfuri V_i .

Apoi trebuie să arătăm că $E_s \cap E_t = \emptyset$ și $\bigcup_{i=1}^k E_i = E$. Presupunem că există $s \neq t$ astfel încât $E_s \cap E_t \neq \emptyset$ și fie $e = xy \in E_s \cap E_t$. Atunci $x, y \in V_s \cap V_t$, care este o contradicție. Rămâne să arătăm incluziunea $E \subseteq \bigcup_{i=1}^k E_i$. Fie $e = xy \in E$. Dacă $x \in V_s$ și $y \in V_t$ cu $s \neq t$, atunci din conexitatea grafului

³Pentru definiția unui element maximal a se vedea anexa 3.9.4, pag. 139. Relația de ordine în acest caz este cea definită în (3.1).

deducem că există un lanț între x și y , adică între un vârf din V_s și un vârf din V_t , care este o contradicție cu faptul că V_s și V_t aparțin unor clase de echivalență diferite. În concluzie x, y aparțin aceleiași clase de echivalență V_j . Astfel, $e \in E_j \subseteq \bigcup_{i=1}^k E_i$. ■

O consecință imediată a teoremei 3.7 este următorul corolar.

Corolarul 3.8 (*Descompunerea în componente conexe*)

Grafurile $G_i = (E_i, V_i)$, $i \in \mathbb{N}_k$, induse de clasele de echivalență ale relației de accesibilitate (3.2), sunt componentele conexe ale grafului G .

Demonstrație. Presupunem că există un subgraf conex al lui G , notat $G' = (V', E')$, astfel încât $G_i \preceq G'$. Vom demonstra că $G' = G_i$.

Din $G_i \preceq G$ avem $V_i \subseteq V'$ și $E_i \subseteq E'$. Presupunem că există $x \in V' \setminus V_i$. Deoarece $G_i \preceq G'$ și G', G_i sunt conexe, vârful x este de asemenea conectat cu toate vârfurile din V_i , în contradicție cu construcția lui V_i . Apoi, fie $e = xy \in E'$. Din $V' = V_i$ avem $x, y \in V_i$, și prin urmare $e \in E_i$, deoarece (V_i, E_i) este graful indus pe submulțimea de vârfuri V_i . În concluzie, $E' \subseteq E_i$, și astfel demonstrația este încheiată. ■

Acest corolar ne sugerează următoarea definiție alternativă pentru o componentă conexă.

Definiția 3.9 O **componentă conexă** a unui graf $G = (V, E)$ este un subgraf indus de o clasă de echivalență a relației de accesibilitate pe V .

3.1.5 Multigrafuri, grafuri orientate și multigrafuri orientate

În practică există situații (ca de exemplu rețelele electrice), când trebuie considerate mai multe muchii între două vârfuri. În această situație este nevoie de extinderea noțiunii de graf la o nouă noțiune, care să admită mai multe muchii între aceleași pereche de vârfuri.

Definiția 3.10 *Muchiile multiple* (de asemenea numite **muchii paralele**), sunt două sau mai multe muchii care sunt incidente acelorași două vârfuri.

Un **multigraf** este un graf care poate avea muchii multiple.

Matematic, un multigraf este alcătuit din:

- o mulțime V de vârfuri,
- o mulțime E de perechi neordonate cu componentele în V , ale cărei elemente se numesc muchii,
- o funcție $\mu : E \rightarrow \mathbb{N} \cup \{0\}$, $\mu(xy)$ = numărul de muchii cu extremitățile x, y .

De obicei, un multigraf se notează cu $G = (V, E, \mu)$. Terminologia introdusă în definițiile 3.2, 3.4, este valabilă și pentru multigrafuri, cu observația că din cele trei notații date pentru lanțuri (vezi pag. 69), este corectă doar cea în care sunt specificate atât vârfurile cât și muchiile. Vom admite de asemenea existența buclilor, iar pentru un multigraf fără bucle folosim terminologia *multigraf simplu*.

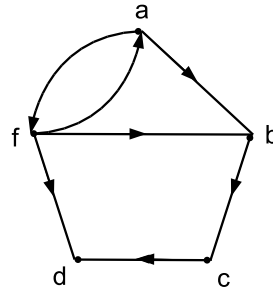


Figura 3.2: Graful orientat din exemplul 3.2.

Conceptele de grafuri și multigrafuri pot să nu fie suficiente. De exemplu, când avem de-a face cu probleme de trafic, este necesar să cunoaștem care străzi ale rețelei au sens unic și în ce direcție se poate circula. Evident, doar un graf al rețelei de străzi nu ne este de folos într-o asemenea situație. Ceea ce avem nevoie este un graf în care fiecare muchie are o anumită orientare – un graf orientat.

Definiția 3.11 Un **graf orientat** (sau un **digraf**) D constă dintr-o mulțime (finită) V , ale cărei elemente se numesc vârfuri, și o mulțime A , $A \subseteq V \times V$, ale cărei elemente se numesc **arce**. De obicei se scrie $D = (V, A)$ și arcele se notează (x, y) , cu $x, y \in V$.

Spre deosebire de muchii, arcele sunt perechi *ordonate*. Astfel, un arc (x, y) unește vârfurile x și y într-o anumită direcție (de la x spre y). Să mai menționăm faptul că, pentru $x \neq y$, arcul (x, y) nu coincide cu arcul (y, x) .

Exemplul 3.2 $D = (V, A)$, cu

$$\begin{aligned} V &= \{a, b, c, d, e, f\}, \\ A &= \{(a, b), (b, c), (c, d), (f, d), (f, b), (f, a), (a, f), (f, b)\}. \end{aligned}$$

Definiția 3.12 Fie $D = (V, A)$ un graf orientat.

Două sau mai multe arce care unesc aceeași pereche de vârfuri în aceeași direcție se numesc **arce multiple** (Dar două arce care unesc aceeași pereche de vârfuri în direcții opuse nu sunt arce multiple!).

Un arc care unește un vârf cu el însuși se numește **bucă**.

Un graf orientat fără bucle se numește **simplu**.

Două vârfuri unite printr-un arc într-o direcție oarecare se numesc **adiacente**.

Vârfurile sunt **incidente** arcului care le unește.

Un arc este **incident** spre și dinspre vârfurile pe care le unește.

Extremitatea inițială a unui arc este vârful din care pleacă arcul.

Extremitatea finală a unui arc este vârful spre care se îndreaptă arcul.

Un **subgraf orientat** (sau **subdigraf**) al lui D este un graf orientat ale cărui vârfuri și arce sunt vârfuri și arce ale lui D .

Dacă $V_1 \subset V$, subgraful orientat construit cu vârfurile lui V_1 și toate arcele din A care unesc aceste vârfuri se numește **subgraf orientat indus** de V_1 .

Graful **suport** al unui graf orientat este graful obținut prin înlocuirea tuturor arcelor din graf orientat prin muchii.

Gradul exterior al unui vârf x este numărul de arce incidente **dinspre** x . **Gradul interior** al unui vârf x este numărul de arce incidente **spre** x . Buclele aduc o contribuție egală cu 1 la fiecare din aceste grade.

Observația 3.4 Fie $D = (V, A)$ un graf orientat și fie \mathcal{D} mulțimea tuturor subgrafurilor digrafului D . Ca și în cazul grafurilor, pe \mathcal{D} se poate introduce următoarea relație:

$$H_1 \preceq H_2 \iff H_1 \text{ subgraf orientat al lui } H_2. \quad (3.3)$$

Se poate arăta că (\mathcal{D}, \preceq) este o mulțime **parțial ordonată**.

Teorema 3.9 (Teorema lui Euler pentru grafuri orientate)

Suma gradelor interioare și a gradelor exterioare ale vârfurilor unui graf sunt ambele egale cu numărul de arce.

Demonstrație. Fiecare arc aduce următoarele contribuții:

- unu, la gradul exterior al vârfului spre care este incident,
- unu, la gradul interior al vârfului dinspre care este incident. ■

3.1.6 Drumuri, drumuri simple, drumuri elementare, cicluri în grafuri orientate

Definiția 3.13⁴ Fie $D = (V, A)$ un graf orientat.

Un **drum** de lungime k ($k \in \mathbb{N}$) în D este o succesiune finită

$\omega = \langle v_0, a_1, v_1, \dots, a_k, v_k \rangle$ ai cărei termeni sunt alternativ vârfuri și arce, astfel încât, pentru $i \in \mathbb{N}_k$, arcul a_i are extremitatea inițială v_{i-1} și extremitatea finală v_i . Spunem că ω este un drum de la v_0 la v_k sau un (v_0, v_k) -drum.

Ca și în cazul grafurilor, un drum $\omega = \langle v_0, a_1, v_1, \dots, a_k, v_k \rangle$ poate fi dat de asemenea doar prin șirul vârfurilor sale $\omega = \langle v_0, v_1, \dots, v_k \rangle$ sau prin șirul arcelor sale $\omega = \langle a_1, \dots, a_k \rangle$.

Un **drum simplu orientat** în D este un drum în care toate arcele (dar nu neapărat vârfurile) sunt distincte.

Un **drum elementar orientat** în D este un drum în care toate vârfurile (și implicit toate arcele) sunt distincte, cu excepția primului și ultimului, care pot să coincidă.

Un drum/drum simplu/drum elementar în D se numește **închis** dacă primul și ultimul vârf coincid.

Un drum simplu închis în D se numește **ciclu** (câteodată **ciclu orientat** sau **circuit**).

⁴Noțiunile corespunzătoare din limba engleză în cazul grafurilor orientate sunt: digraph = graf orientat, (dar există de asemenea terminologia «oriented graf», care înseamnă și nu se traduce prin graf orientat !), walk = drum, closed walk = drum închis, trail = drum simplu, path = drum elementar, cycle = ciclu elementar, closed trail = ciclu orientat, circuit.

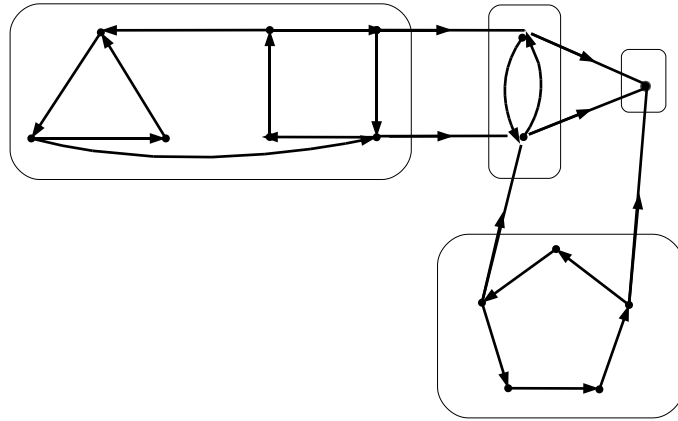


Figura 3.3: Un graf orientat și componentele sale conexe.

Un drum elementar închis în D se numește **ciclu elementar**.

Terminologia introdusă în definițiile 3.5 și 3.7 este de asemenea valabilă, înlocuind muchiile cu arce.

3.1.7 Conectivitate și conectivitate tare în grafuri orientate

Privind conectivitatea unui graf orientat, lucrurile sunt puțin diferite față de cazul grafurilor.

Definiția 3.14 Un **graf orientat conex** este un graf orientat al cărui graf suport este un graf conex. Un **graf orientat neconex** este un graf orientat care nu este conex.

Un graf orientat este **tare conex** dacă există un drum care unește orice pereche orientată de vârfuri.

Un graf orientat este **conex** dacă graful său suport este conex.

O **componentă tare conexă** a unui graf orientat D este un subgraf tare conex maximal S al lui D , în sensul că S nu este un subgraf propriu al nici unui subgraf orientat tare conex al lui D .

Să menționăm faptul că dacă un graf orientat este tare conex, atunci el este conex, dar implicația inversă nu este adevărată în general.

Teorema 3.10 (Descompunerea unui graf orientat în componente tare conexe) Pentru orice graf orientat $D = (V, A)$, există o partiție $\{V_1, V_2, \dots, V_k\}$ a mulțimii V , astfel încât subgrafurile orientate $D_i = (V_i, A_i)$, induse de V_i , sunt componentele tare conexe ale lui D .

Demonstrație. Definim relația \sim pe V , prin

$$v_i \sim v_j \iff i = j \quad \text{sau} \quad \begin{array}{l} \text{există un drum, atât de la} \\ v_i \text{ la } v_j, \text{ cât și de la } v_j \text{ la } v_i. \end{array}$$

Atunci \sim este o relație de echivalență pe V , numită *relația de accesibilitate reciprocă*. Aceasta induce o partiție $\{V_l, l \in \mathbb{N}_k\}$ a lui V . Pentru $l \in \mathbb{N}_k$,

considerăm subgrafurile orientate $D_l = (V_l, A_l)$ induse de V_l . Fie $s \in \mathbb{N}_k$. Vom demonstra că D_s este tare conex.

Deci fie $v_i, v_j \in V_s$, $v_i \neq v_j$. Deoarece v_i, v_j aparțin aceleiași clase de echivalență, vom avea $v_i \sim v_j$. Astfel, există drumurile $\omega_1 = \langle v_i, \dots, v_j \rangle$ și $\omega_2 = \langle v_j, \dots, v_i \rangle$, cu arce din A . Demonstrăm că aceste drumuri au toate arcele în A_s . Presupunem contrariul, că există $x^* \notin V_s$ astfel încât

$$\omega_1 = \langle v_i, \dots, x^*, \dots, v_j \rangle.$$

Atunci $\omega_3 = \langle v_i, \dots, x^* \rangle$ este un drum de la v_i la x^* și $\omega_4 = \langle x^*, \dots, v_j \rangle \cup \omega_2$ este un drum de la x^* la v_i . Prin urmare, $x^* = v_i$, care este o contradicție cu $x^* \notin V_s$. În concluzie, atât ω_1 cât și ω_2 sunt drumuri cu arce din A_s , deci $D_s = (V_s, A_s)$ este un subgraf tare conex.

Faptul că D_s este maximal poate fi arătat folosind același raționament ca și în demonstrația corolarului 3.8. ■

Ca și pentru grafuri, are loc următoarea definiție alternativă a unei componente tare conexe.

Definiția 3.15 *O componentă tare conexă a unui graf orientat $D = (V, A)$ este un subgraf orientat indus de o clasă de echivalență a relației de accesibilitate reciprocă pe V .*

Observația 3.5 *În cazul grafurilor orientate, nu există în mod necesar o partiție a mulțimii de arce, adică*

$$\bigcup_{i=1}^k A_i \neq A, \quad \text{în general.}$$

Aceasta înseamnă că în anumite grafuri orientate există arce care nu aparțin niciunuia din subgrafurile orientate ale partiției. Aceste arce au următoarea proprietate (vezi figura 3.3):

Dacă V_1, V_2 sunt mulțimile de vârfuri a două componente tare conexe ale lui D , atunci toate arcele între V_1 și V_2 au aceeași orientare (fie toate **dinspre** V_1 fie toate **spre** V_1).

Spre deosebire de grafurile orientate, în grafuri există întotdeauna o partiție a muchiilor, așa cum am putut vedea în teorema 3.7.

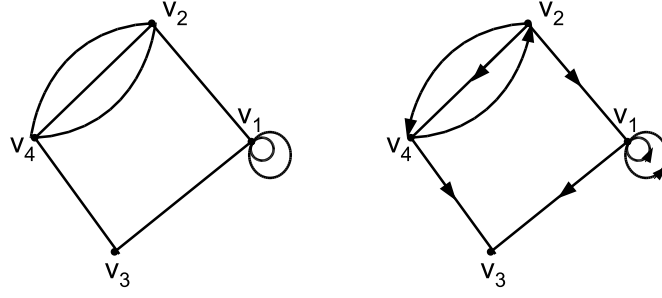
Desigur, în practică este necesar câteodată să considerăm multigrafuri orientate. Definiția va fi similară cu cea a multigrafurilor.

Definiția 3.16 *Un multigraf orientat este un graf orientat care poate avea arce multiple.*

Matematic, un multigraf orientat constă din:

- o mulțime V de vârfuri,
- o mulțime $A \subseteq V \times V$ de perechi ordonate cu componentele în V , ale cărei elemente se numesc arce,
- o funcție $\mu : A \rightarrow \mathbb{N}$, $\mu((x, y)) = \text{numărul de arce de la } x \text{ la } y$.

De obicei, un multigraf orientat se notează cu $D = (V, A, \mu)$. Terminologia introdusă în definițiile 3.2 și 3.4 este de asemenea valabilă pentru multigrafuri orientate.

Figura 3.4: Graful G și graful orientat D din exemplul 3.3.

3.2 Reprezentarea grafurilor

Există mai multe moduri de a reprezenta un graf/graf orientat: desen, matrice de adiacență, matrice de incidență, lista muchiilor incidente (respectiv listele arcelor care vin și care pleacă).

3.2.1 Matrice de adiacență

Definiția 3.17 Matricea de adiacență A_G a grafului (grafului orientat, multigrafului, multigrafului orientat) G cu mulțimea de vârfuri $V = \{v_1, v_2, \dots, v_n\}$ este o matrice de dimensiune $n \times n$ cu elementele

$$(A_G)_{ij} = \begin{cases} \text{numărul de } (v_i, v_j)\text{-muchii (arce), dacă } i \neq j, \\ \text{numărul de bucle în } v_i \text{ dacă } i = j, \end{cases}$$

pentru $i, j \in \mathbb{N}_n$.

Pe parcursul acestui paragraf vom folosi termenul de graf (resp. graf orientat) și pentru multigrafuri (resp. multigrafuri orientate).

Exemplul 3.3 Pentru graful G și graful orientat D din figura 3.4, matricele de adiacență sunt, respectiv,

$$A_G = \begin{matrix} & \begin{matrix} v_1 & v_2 & v_3 & v_4 \end{matrix} \\ \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{matrix} & \begin{pmatrix} 2 & 1 & 1 & 0 \\ 1 & 0 & 0 & 3 \\ 1 & 0 & 0 & 1 \\ 0 & 3 & 1 & 0 \end{pmatrix} \end{matrix}, \quad A_D = \begin{matrix} & \begin{matrix} v_1 & v_2 & v_3 & v_4 \end{matrix} \\ \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{matrix} & \begin{pmatrix} 2 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 \end{pmatrix} \end{matrix}.$$

Câteva proprietăți imediate ale matricelor de adiacență sunt:

1. Matricea de adiacență a unui graf (graf orientat) simplu sunt 0 și 1, cu 0 pe diagonala principală;
2. Matricea de adiacență a unui graf este simetrică;
3. Dacă $D = (V, A)$ este un graf orientat cu $V = \{v_1, \dots, v_n\}$ și elementele matricei de adiacență sunt a_{ij} , $i, j \in \mathbb{N}_n$, atunci

$$\sum_{j=1}^n a_{ij} = \text{gradul exterior}(v_i), \quad \sum_{i=1}^n a_{ij} = \text{gradul interior}(v_j).$$

Dacă în matricea din exemplul 3.3 vrem numărul de lanțuri de lungime 2 dintre două vârfuri, putem să numărăm cu ușurință că există două asemenea (v_1, v_2) -lanțuri, niciun (v_2, v_4) -lanț, 4 (v_2, v_3) -lanțuri, ș.a.m.d. Dacă se construiește o matrice al cărei element $\alpha_{i,j}$ ($i, j \in \mathbb{N}_n$) reprezintă numărul de (v_i, v_j) -lanțuri de lungime 2, această matrice va fi egală cu

$$\begin{pmatrix} 6 & 2 & 2 & 4 \\ 2 & 10 & 4 & 0 \\ 2 & 4 & 2 & 0 \\ 4 & 0 & 0 & 10 \end{pmatrix},$$

reprezentând exact A_G^2 . În general, are loc următoarea proprietate:

Propoziția 3.11 *Fie G un graf (graf orientat) cu mulțimea de vârfuri $V = \{v_1, v_2, \dots, v_n\}$ și fie A_G matricea sa de adiacență. Atunci valoarea elementului $a_{ij}^{(r)}$ al puterii r a matricei A_G este egală cu numărul de (v_i, v_j) -lanțuri (drumuri) de lungime r .*

Demonstrație. Prin inducție după r . Pentru $r = 1$, proprietatea este adevărată. Presupunem că ea este adevărat pentru r . Avem

$$a_{ij}^{(r+1)} = \sum_{k=1}^n a_{ik}^{(r)} a_{kj}.$$

În al k -lea termen al acestei sume, factorul $a_{ik}^{(r)}$ reprezintă numărul de (v_i, v_k) -lanțuri de lungime r și factorul a_{kj} reprezintă numărul de (v_k, v_j) -lanțuri de lungime 1. Prin urmare, produsul $a_{ik}^{(r)} a_{kj}$ este egal cu numărul de (v_i, v_j) -lanțuri de lungime $r + 1$, cu penultimul vârf v_k . Însumând după k , obținem numărul total de (v_i, v_j) -lanțuri de lungime $r + 1$, și astfel concluzia este demonstrată. ■

Observația 3.6 *Egalitatea $a_{ij}^{(p)} = 0$ nu implică existența unui (v_i, v_j) -lanț de lungime $< p$.*

Propoziția 3.11 ne permite să decidem dacă un graf este conex (sau dacă un graf orientat este tare conex). Dacă un graf este conex, trebuie să existe lanțuri (elementare) între oricare două vârfuri. Lungimea fiecărui lanț va fi $\leq n - 1$, altfel un vârf ar fi vizitat mai mult de o singură dată. În concluzie, are loc următorul rezultat.

Propoziția 3.12 *Dacă \mathcal{A} este matricea de adiacență a unui graf (sau graf orientat) G cu n vârfuri și*

$$\mathcal{T}_{n-1} = \mathcal{A} + \mathcal{A}^2 + \dots + \mathcal{A}^{n-1},$$

atunci G este conex (respectiv tare conex) dacă și numai dacă fiecare element ne-diagonal al lui \mathcal{T}_{n-1} este > 0 .

Demonstrație. Fie $\mathcal{A}^k = (a_{ij}^{(k)})_{i,j \in \mathbb{N}_n}$ pentru $k \in \mathbb{N}_{n-1}$, $\mathcal{T}_{n-1} = (t_{ij})_{i,j \in \mathbb{N}_n}$ și fie $i, j \in \mathbb{N}_n$, $i \neq j$.

\Rightarrow Presupunem că G este tare conex. Atunci există un (v_i, v_j) -lanț, de o anumită lungime k , $k \in \mathbb{N}_{n-1}$. Astfel, $a_{ij}^{(k)} > 0$ și prin urmare $t_{ij} > 0$.

\Leftarrow Dacă $t_{ij} > 0$, atunci există $k \in \mathbb{N}_{n-1}$ astfel încât $a_{ij}^{(k)} > 0$. Deci, există un (v_i, v_j) -lanț (de lungime k), prin urmare G este tare conex. ■

Pentru graful orientat din exemplul 3.3 avem

$$A_D^2 = \begin{pmatrix} 4 & 0 & 2 & 0 \\ 2 & 2 & 2 & 0 \\ 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 2 \end{pmatrix}, \quad A_D^3 = \begin{pmatrix} 8 & 0 & 4 & 0 \\ 6 & 0 & 2 & 4 \\ 0 & 0 & 0 & 0 \\ 4 & 4 & 4 & 0 \end{pmatrix}, \quad \mathcal{T}_3 = \begin{pmatrix} 4 & 0 & 2 & 0 \\ 2 & 2 & 2 & 0 \\ 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 2 \end{pmatrix},$$

prin urmare graful orientat D nu este tare conex, și aceasta pentru că nu există nici un lanț între v_3 și orice alt vârf.

În cazul în care suntem interesați doar dacă există *cel puțin un* (v_i, v_j) -lanț și nu ne interesează numărul lanțurilor, putem folosi **matrice booleene**. O matrice de adiacență booleană a unui graf (graf orientat) G cu mulțimea de vârfuri $V = \{v_1, \dots, v_n\}$ are elementele

$$b_{ij} = \begin{cases} 1, & \text{dacă există cel puțin o muchie (arc) } (v_i, v_j), \\ 0, & \text{în rest.} \end{cases}$$

Pentru graful din exemplul 3.3, matricele booleene sunt

$$B_G = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}, \quad B_D = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix}.$$

Puterile și sumele acestor matrice vor fi calculate utilizând aritmetica *booleană*⁵. În această aritmetică, numerele sunt 1 și 0 (câteodată folosite ca și *adevărat* și *fals*), suma și produsul notate \oplus și \odot , și definite ca și

$$\begin{array}{c|cc} \oplus & 0 & 1 \\ \hline 0 & 0 & 1 \\ 1 & 1 & 1 \end{array} \quad \begin{array}{c|cc} \odot & 0 & 1 \\ \hline 0 & 0 & 0 \\ 1 & 0 & 1 \end{array}$$

Avantajul aritmeticii booleene în programare este că necesită mai puțină memorie și timp. Operațiile \oplus și \odot pentru matrice booleene sunt definite ca și în cazul matricelor reale, cu \oplus în loc de $+$, \odot în loc de \cdot , și au proprietățile

$$\begin{aligned} A \oplus A &= A, \\ A \odot I &= I \odot A = A, \\ A \odot (B \oplus C) &= A \odot B \oplus A \odot C. \end{aligned}$$

Pentru o matrice booleană B vom folosi notațiile

$$B^{[2]} = B \odot B, \quad B^{[p]} = B^{[p-1]} \odot B, \quad \text{pentru } p \geq 3.$$

Exemplul 3.4 Considerăm graful orientat din figura 3.5. Matricea sa booleană de adiacență va fi

$$B = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \end{pmatrix},$$

⁵După numele lui George Boole (1815-1864), matematician și filozof britanic, inventatorul algebrei booleene, baza aritmeticii calculatoarelor.

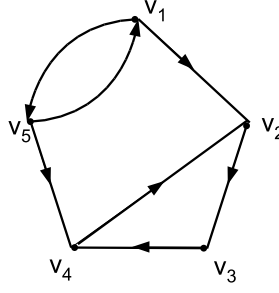


Figura 3.5: Graful orientat din exemplul 3.4.

și atunci, cu notația $\mathcal{R}_4 = B \oplus B^{[2]} \oplus B^{[3]} \oplus B^{[4]}$,

$$B^{[2]} = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix}, \quad B^{[3]} = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 \end{pmatrix},$$

$$B^{[4]} = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{pmatrix}, \quad \mathcal{R}_4 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

Are loc un rezultat similar celui dat în propoziția 3.11, anume:

Propoziția 3.13 Fie B matricea booleană a unui graf (graf orientat) cu n vârfuri și fie $p \in \mathbb{N}$. Notăm cu $a_{ij}^{[p]}$ elementele matricei $B^{[p]}$ și cu $r_{ij}^{[p]}$ elementele matricei $\mathcal{R}_p = B \oplus B^{[2]} \oplus \dots \oplus B^{[p]}$. Sunt adevărate următoarele afirmații, pentru $i \neq j$:

1. Dacă $a_{ij}^{[p]} = 1$, atunci există un (v_i, v_j) -lanț de lungime $= p$,
2. Dacă $a_{ij}^{[p]} = 0$, atunci nu există niciun (v_i, v_j) -lanț de lungime $= p$,
3. Dacă $r_{ij}^{[p]} = 1$, atunci există un (v_i, v_j) -lanț de lungime $\leq p$,
4. Dacă $r_{ij}^{[p]} = 0$, atunci nu există niciun (v_i, v_j) -lanț de lungime $\leq p$.

Demonstrație. Demonstrația se face analog cu cea din propoziția 3.11. ■

Calcularea sumei $\mathcal{R}_p = B \oplus B^{[2]} \oplus \dots \oplus B^{[p]}$ poate fi prea costisitoare pentru n mare. Putem deduce aceleași concluzii ca și în 3. și 4., folosind matricea $\mathcal{B}_p = I \oplus \mathcal{R}_p$ în loc de \mathcal{R}_p . Această matrice schimbă doar diagonală principală, în schimb are următoarea proprietate:

Propoziția 3.14 Dacă B este o matrice booleană și $C = I \oplus B$, atunci

$$C^{[p]} = I \oplus B \oplus B^{[2]} \oplus \dots \oplus B^{[p]}, \quad \text{pentru orice } p \in \mathbb{N}.$$

Demonstrație. Prin inducție. Pentru $p = 1$, concluzia este adevărată. Presupunem că este adevărată pentru p . Avem

$$\begin{aligned} C^{[p+1]} &= C^{[p]} \odot C = (I \oplus B \oplus B^{[2]} \oplus \dots \oplus B^{[p]}) \odot (I \oplus B) \\ &= I \oplus (B \oplus B) \oplus (B^{[2]} \oplus B^{[2]}) \oplus \dots \oplus (B^{[p]} \oplus B^{[p]}) \oplus B^{[p+1]} \\ &= I \oplus B \oplus B^{[2]} \oplus \dots \oplus B^{[p+1]}, \end{aligned}$$

și folosind principiul inducției, concluzia este demonstrată. ■

O aplicație importantă a acestei proprietăți este următoarea: Date fiind două vârfuri distincte v_i, v_j ale unui graf (graf orientat), se poate determina lungimea unui (v_i, v_j) – lanț de lungime minimă, calculând successiv puterile booleene ale lui C și înregistrând prima apariție a unui 1 pe poziția (i, j) . Odată ce 1 apare pe o anumită poziție (i, j) din $C^{[p]}$, va apărea pe aceeași poziție în toate matricele $C^{[p']}$ cu $p' \geq p$.

De fapt, există o metodă mai rapidă, în care se evaluează puterile $C^{[2]}$, $C^{[4]}$, \dots , $C^{[2^s]}$, \dots și anumite puteri booleene intermediare dintre puterile 2^s și 2^{s+1} , unde poziția (i, j) se schimbă din 0 în 1. De exemplu, să presupunem că minimumul lungimii unui (v_i, v_j) – lanț este 51, dar este necunoscut. Mai întâi calculăm $C^{[2]}$, $C^{[4]}$, $C^{[8]}$, $C^{[16]}$, $C^{[32]}$, găsim elementele

$$c_{ij}^{[2]} = c_{ij}^{[4]} = c_{ij}^{[8]} = c_{ij}^{[16]} = c_{ij}^{[32]} = 0.$$

Apoi, pentru $C^{[64]}$ găsim $c_{ij}^{[64]} = 1$ (deci $s = 5$), prin urmare există un (v_i, v_j) – lanț de lungime l , cu $32 < l \leq 64$.

Mai departe, calculăm $C^{[32]} \odot C^{[2^k]}$, pentru $k = 1, 2, 3, 4$ (în general calculăm pentru $k \leq s - 1$, pâna la prima apariție a lui 1). În cazul nostru

$$c_{ij}^{[32+2]} = c_{ij}^{[32+4]} = c_{ij}^{[32+8]} = c_{ij}^{[32+16]} = 0,$$

deci $48 < l \leq 64$. Apoi, calculăm $C^{[48]} \odot C^{[2^k]}$, pentru $k = 1, 2$ (în general pentru $k \leq s - 2$, până apare 1) și găsim $c_{ij}^{[48+2]} = 0$, $c_{ij}^{[48+4]} = 1$. Astfel, $50 < l \leq 52$, și avem nevoie de încă un calcul să găsim l , anume $c_{ij}^{[51]} = 1$. În total, am efectuat $6+4+2+1=13$ operații, în loc de 50 necesare pentru a calcula puterile booleene succesive ale lui C până la 51.

Puterile booleene ale matricei C sunt de asemenea utile pentru determinarea componentelor tare conexe ale unui graf orientat (respectiv componentele unui graf). Mai precis are loc următorul rezultat.

Teorema 3.15 (Metoda lui Foulkes de găsim a componentelor) *Fie G un graf (graf orientat) simplu cu m muchii (arce), B matrice sa booleană de adiacență, $C = I \oplus B$, și $p \in \mathbb{N}$ astfel încât $2^p \geq m$. Atunci două vârfuri distincte v_i, v_j sunt situate în aceeași componentă (tare) conexă a lui G dacă și numai dacă liniile i și j din $C^{[2^p]}$ sunt identice.*

Demonstrație. Fie $i \neq j$ și fie α_{ij} elementul (i, j) al matricei $C^{[2^p]}$. Orice lanț din G are cel mult m arce, deci, dacă $\alpha_{ij} = 0$, atunci nu există niciun (v_i, v_j) – lanț.

\Rightarrow Presupunem că v_i, v_j sunt situate în aceeași componentă. Atunci este imediat faptul că $\alpha_{ii} = \alpha_{ij} = \alpha_{ji} = \alpha_{jj} = 1$. Fie $\nu \neq i, \nu \neq j$.

Dacă $\alpha_{i\nu} = 1$, există un (v_i, v_ν) – lanț în G . Deoarece există de asemenea un (v_j, v_i) – lanț, reuniunea acestor două lanțuri va fi un (v_j, v_ν) – lanț, și prin urmare $\alpha_{j\nu} = 1$.

Dacă $\alpha_{i\nu} = 0$, presupunem că $\alpha_{j\nu} = 1$, deci există un (v_j, v_ν) – lanț. Ca și înainte, deducem că există un (v_i, v_ν) – lanț, care este o reuniune a (v_i, v_j) – lanțului cu acest (v_j, v_ν) – lanț. Astfel, $\alpha_{i\nu} = 1$, care este o contradicție. Prin urmare, $\alpha_{j\nu} = 0$.

În concluzie, în ambele cazuri obținem $\alpha_{i\nu} = \alpha_{j\nu}$, însemnând că liniile i și j din $C^{[2^p]}$ coincid.

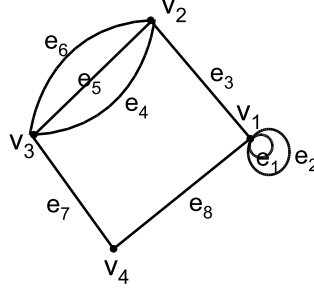


Figura 3.6: Graful din exemplul 3.5.

\Leftarrow Presupunem că $\alpha_{i\nu} = \alpha_{j\nu}$ pentru toți ν .

Din $\alpha_{ii} = 1$ deducem $\alpha_{ji} = 1$, prin urmare există un (v_j, v_i) – lanț. În același mod, din $\alpha_{jj} = 1$ deducem că $\alpha_{ij} = 1$, de unde există un (v_i, v_j) – lanț. În concluzie, v_i și v_j sunt situate în aceeași componentă conexă.

■

Pentru graful din exemplul 3.4 avem $C^{[4]} = \mathcal{R}_4$, deci graful are două componente tare conexe, una conținând vârfurile v_1, v_5 și cealaltă conținând vârfurile v_2, v_3, v_4 .

3.2.2 Matrice de incidență

Definiția 3.18 (Matrice de incidență a unui graf) Fie

$G = (V, E)$ un graf cu $V = \{v_1, v_2, \dots, v_n\}$ și $E = \{e_1, e_2, \dots, e_m\}$. Matricea de incidență a grafului G este o matrice I_G de dimensiune $n \times m$ cu elementele

$$(I_G)_{ij} = \begin{cases} 0, & \text{dacă } v_i \text{ nu este extremitate a muchiei } e_j, \\ 1, & \text{dacă } v_i \text{ e extremitatea lui } e_j \text{ și } e_j \text{ e muchie proprie,} \\ 2, & \text{dacă } e_j \text{ este o buclă în } v_i. \end{cases}$$

pentru $i, j \in \mathbb{N}_n$.

Exemplul 3.5 Pentru graful din figura 3.6, matricea de incidență este

$$I_G = \begin{matrix} & \begin{matrix} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 \end{matrix} \\ \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{matrix} & \begin{pmatrix} 2 & 2 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \end{matrix}.$$

Câteva proprietăți ale matricei de incidență a unui graf sunt:

1. Suma elementelor de pe orice linie a matricei de incidență este egală cu gradul vârfului corespunzător, dacă o buclă este considerată ca având gradul 2;
2. Suma elementelor de pe orice coloană a unei matrice de incidență este 2;
3. Matricea de incidență a unui graf G cu $n \geq 2$ vârfuri are rangul $n - p$, unde p este numărul de componente conexe ale lui G .

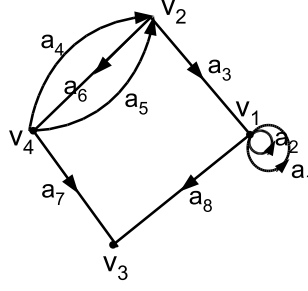


Figura 3.7: Graful din exemplul 3.6.

Definiția 3.19 (Matricea de incidență a unui graf orientat) Fie $D = (V, A)$ un graf orientat cu $V = \{v_1, v_2, \dots, v_n\}$ și $A = \{a_1, a_2, \dots, a_m\}$. Matricea de incidență a grafului D este o matrice I_D de dimensiune $n \times m$ cu elementele

$$(I_D)_{ij} = \begin{cases} 0, & \text{dacă } v_i \text{ nu este extremitate a arcului } a_j, \\ 1, & \text{dacă } v_i \text{ este extremitatea finală a arcului } a_j, \\ -1, & \text{dacă } v_i \text{ este extremitatea inițială a arcului } a_j, \\ 2, & \text{dacă } a_j \text{ este o buclă în } v_i. \end{cases}$$

pentru $i, j \in \mathbb{N}_n$.

Exemplul 3.6 Pentru graful din figura 3.7, matricea de incidență este

$$I_D = \begin{matrix} & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & a_8 \\ \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{matrix} & \begin{pmatrix} 2 & 2 & 1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 1 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \end{matrix}.$$

O proprietate imediată este că într-un graf orientat fără bucle, suma pe fiecare coloană este zero.

3.3 Arbori, sortare și căutare

Definiția 3.20

Un **arbore** este un graf conex fără cicluri.

Orice subgraf conex al unui arbore se numește **subarbore**.

O **pădure** este un graf fără cicluri (sau, echivalent, o mulțime de arbori).

În figura 3.8 sunt date exemple de arbori. În continuare vom da câteva proprietăți simple ale arborilor.

Teorema 3.16 Într-un arbore T , oricare două vârfuri distincte x, y sunt conectate printr-un lanț unic.

Demonstrație. Intrucât T este conex, există un (x, y) -lanț, notat $\langle v_0, v_1, \dots, v_r \rangle$, cu $v_0 = x, v_r = y$. Dacă există un alt (x, y) -lanț, $\langle u_0, u_1, \dots, u_s \rangle$, cu $u_0 = x, u_s = y$, atunci fie i cel mai mic indice pentru care $u_{i+1} \neq v_{i+1}$. Deoarece ambele lanțuri se termină în y , ele se vor întâlni încă o dată, și fie j cel mai mic indice astfel încât $j > i$ și $v_j = u_k$ pentru un anumit k . Atunci $\langle v_i, v_{i+1}, \dots, v_j, u_{k-1}, \dots, u_{i+1}, v_i \rangle$ este un ciclu în T , contrar ipotezei. ■

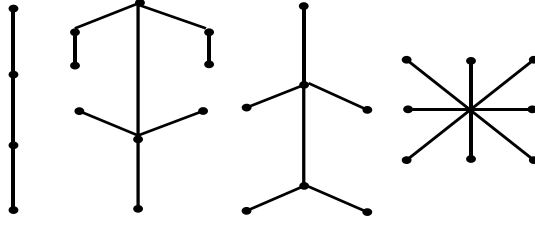
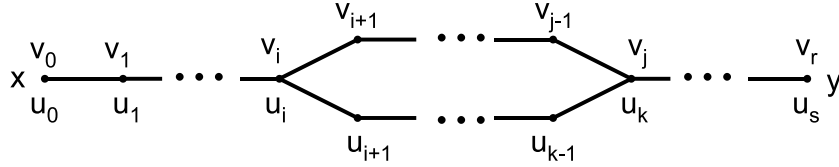


Figura 3.8: Exemple de arbori.



Teorema 3.17 *Graful obținut din arborele $T = (V, E)$ prin eliminarea oricărei muchii are două componente, fiecare dintre acestea fiind un arbore.*

Demonstrație. Presupunem că $uv \in E$, $E' = E \setminus \{uv\}$, și fie $S = (V, E')$. Definim

$$V_1 = \{x \in V : \text{unicul } (x, v) - \text{lanț din } T \text{ trece prin } u\}.$$

Pentru orice $x \in V_1$, unicul (x, v) – lanț trebuie să se termine și să înceapă cu muchia uv , altfel T ar conține un ciclu $\langle x, \dots, u, v, \dots, x \rangle$.

Fie $V_2 = V \setminus V_1$. Fiecare vârf din V_1 este unit cu u printr-un lanț din S și fiecare vârf din V_2 este unit cu v în S . În concluzie V_1 și V_2 sunt mulțimile de vârfuri ale celor două componente ale lui S . Fiecare componentă este (prin definiție) conexă și ea nu conține niciun ciclu, întrucât în T nu există cicluri. Astfel, cele două componente sunt arbori. ■

Teorema 3.18 *Dacă $T = (V, E)$ este un arbore, atunci $|E| = |V| - 1$.*

Demonstrație. Prin inducție după $|V|$.

Dacă $|V| = 1$, concluzia are loc, deoarece unicul arbore care există în acest caz nu are cicluri.

Presupunem concluzia adevărată pentru un arbore cu $\leq k$ vârfuri. Fie T un arbore cu $|V| = k + 1$ și fie uv o muchie a lui T . Dacă $T_1 = (V_1, E_1)$ și $T_2 = (V_2, E_2)$ sunt arborii obținuți prin eliminarea muchiei uv din T (vezi teorema 3.17), avem

$$|V_1| + |V_2| = |V| \quad \text{și} \quad |E_1| + |E_2| = |E|.$$

Astfel,

$$|E| = |E_1| + |E_2| + 1 = |V_1| - 1 + |V_2| - 1 + 1 = |V| - 1.$$

În a doua egalitate aplicăm ipoteza inducției pentru arborii T_1 și T_2 . ■

Observația 3.7 *Concluziile teoremelor 3.16–3.18 furnizează câteva moduri alternative de a defini un arbore.*

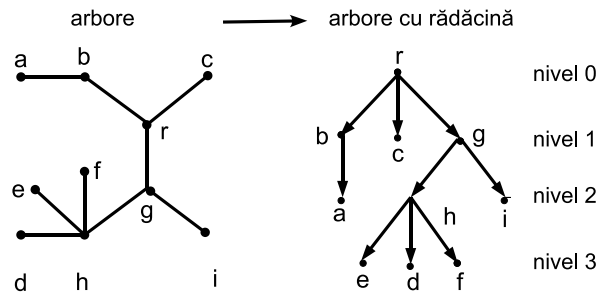


Figura 3.9: Construcția arborelui cu rădăcina într-un vârf dat r .

3.3.1 Arbori cu rădăcină

Frecvent, un vârf al unui arbore poate avea un rol special dintr-un anumit punct de vedere. De exemplu, într-un arbore genealogic al unei familii regale, putem pune în evidență poziția specială a regelui punându-l în vârful arborelui, și pe toți fiii «sub» părinții lor.⁶ În general vârful cu rol special se numește **rădăcină** și se asociază o **direcție** fiecărei muchii, anume de la părinte la fiu.

Definiția 3.21 Un **arbore orientat** este un graf orientat al cărui graf suport este un arbore.

Un **arbore cu rădăcină** sau **arborescență** este un arbore orientat în care există un vârf special r , numit **rădăcină**, astfel încât pentru orice alt vârf v , există un drum (unic) de la r la v .

Atunci când se desenează un arbore cu rădăcină, săgețile de pe arce de obicei se omit, deoarece direcția arcului este întotdeauna «de la» rădăcină. De asemenea rezultă că drumul de la rădăcină la un vârf dat este unic.

Definiția 3.22 Într-un arbore cu rădăcină, **adâncimea** sau **nivelul** unui vârf v , notată $\text{depth}(v)$, este distanța de la rădăcină, adică lungimea unicului drum de la rădăcină la v . Astfel, rădăcina are nivelul 0.

Un vârf într-un arbore cu rădăcină se numește **frunză** dacă se află la nivelul i ($i \geq 0$) și nu este adiacent niciunui vârf de la nivelul $i + 1$.

Un vârf care nu este o frunză este un vârf **interior**.

Înălțimea unui arbore cu rădăcină este cel mai mare număr k pentru care nivelul k nu este vid.

Dacă vârful x precede imediat vârful y pe drumul de la rădăcină la y , atunci x este **părintele** sau **tatăl** lui y și y este **copilul** sau **fiul** lui x .

Fie x, y vârfuri distincte ale unui arbore cu rădăcină. Atunci y se numește **descendent** al lui x (și x se numește **predecesor** al lui y) dacă x este situat pe unicul drum de la rădăcină la y .

Fiecare vârf (cu excepția rădăcinii) are un unic tată, dar un vârf poate avea un număr arbitrar de fii (inclusiv 0). Deci, un vârf este o frunză dacă și numai dacă nu are nici un fiu. De asemenea, un vârf este tată dacă și numai

⁶Să menționăm că, atunci când se alcătuiește un arbore genealogic, istoricii pun de obicei regele pe poziția cea mai joasă și descendenții deasupra lui, tocmai ca să se reproducă forma unui arbore adevărat. Cele două versiuni sunt, desigur, echivalente.

dacă este un vârf interior. Arborii cu rădăcină sunt de fapt grafuri orientate, direcțiile arcelor fiind de la un tată la fiul său. Pentru simplitate, vom omite săgețile și terminologia pentru grafuri orientate.

Definiția 3.23 *Dacă într-un arbore cu rădăcină fiecare tată are cel mult m fii, spunem că avem un arbore **m-ar** cu rădăcină. În particular, pentru $m = 2$, arborele se va numi **binar**, iar când $m = 3$ se va numi **ternar**. Un arbore cu rădăcină m -ar, în care fiecare tată are exact m fii, se numește **complet**.*

3.3.2 Arbori de decizie

Incepem acest paragraf cu următoarea teoremă:

Teorema 3.19 *Înălțimea unui arbore m -ar cu rădăcină cu k frunze este cel puțin $\log_m k$.*

Demonstrație. Fie h înălțimea arborelui. Inegalitatea cerută

$$h \geq \log_m k \quad (3.4)$$

este echivalentă cu $m^h \geq k$, deci ar fi suficient să arătăm că un arbore m -ar cu rădăcină, de înălțime h , are cel mult m^h frunze.

Arătăm acest lucru prin inducție după h . Dacă $h = 0$, arborele are un singur vârf, care este atât frunză cât și tată, deci $k = 1$, și astfel inegalitatea este adevărată. Presupunem acum că inegalitatea este adevărată pentru $0 \leq h \leq h_0$ și fie T un arbore m -ar cu rădăcină cu înălțime $h_0 + 1$. Dacă notăm rădăcina cu r , atunci arborele $T - r$ (vezi definiția 3.7, pag. 70) va fi format din cel mult m arbori T_1, \dots, T_m ale căror rădăcini sunt vârfurile de la nivelul 1 din T . Fiecare T_i este astfel arbore cu rădăcină, de înălțime $\leq h_0$. Din ipoteza inducției, fiecare din acești T_i arbori are cel mult m^{h_0} frunze. Frunzele lui T coincid cu frunzele arborilor T_1, \dots, T_m , prin urmare numărul de frunze din T va fi mai mic decât $m \cdot m^{h_0} = m^{h_0+1}$.

În concluzie, conform principiului inducției, inegalitatea $k \leq m^h$ este adevărată pentru orice $h \geq 0$. ■

Observația 3.8 *În general numărul $\log_m k$ nu este întreg. Inegalitatea (3.4) poate fi rescrisă ca și*

$$h \geq \lceil \log_m k \rceil, \quad (3.5)$$

unde pentru un număr real x , se notează cu $\lceil x \rceil$ cel mai mic număr întreg z astfel încât $z \geq x$.

Teorema 3.19 se poate aplica în studiul arborilor de decizie. Un arbore de decizie este un arbore m -ar în care:

- fiecare vârf interior reprezintă o decizie,
- rezultatele posibile ale deciziei sunt reprezentate prin muchiile care unesc vârfurile situate la nivelul următor,
- frunzele arborelui reprezintă rezultatele finale ale procedurii.

În cazul când rezultatele deciziei sunt doar **adevărat** sau **fals** (0 sau 1), avem de-a face cu un arbore de decizie binar.

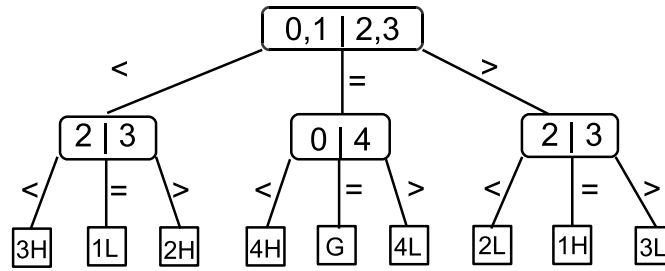


Figura 3.10: Arborele de decizie pentru exemplul 3.7.

Exemplul 3.7 (Problema falsei monede) Presupunem că avem o monedă autentică etichetată cu 0 și n alte monede, indentice cu 0 în aparență, cu excepția faptului că sunt etichetate $1, 2, \dots, n$. Se suspectează că una din monezi poate fi falsă – fie prea ușoară fie prea grea. Să se arate că sunt necesare cel puțin $\lceil \log_3(2n + 1) \rceil$ cântăriri pe o balanță, pentru a decide care monedă (dacă există vreuna) este falsă, și de asemenea să se decidă dacă ea este mai ușoară sau mai grea. Să se realizeze o procedură care folosește exact acest număr de cântăriri, când $n = 4$.

Soluție. Există $2n + 1$ rezultate (frunze ale arborelui de decizie), anume

$$G \quad 1H \quad 1L \quad 2H \quad 2L \quad \dots \quad nH \quad nL,$$

unde G înseamnă că toate monezile sunt bune, kH înseamnă că moneda k este mai grea și kL înseamnă că moneda k este mai ușoară.

Avem nevoie de un arbore ternar de decizie, deoarece există trei rezultate posibile ale fiecărei decizii, atunci când se cântăresc două mulțimi de monezi:

- $<$: mulțimea de monezi din stânga este mai ușoară
- $=$: mulțimile de monezi au aceeași greutate
- $>$: mulțimea de monezi din stânga este mai grea

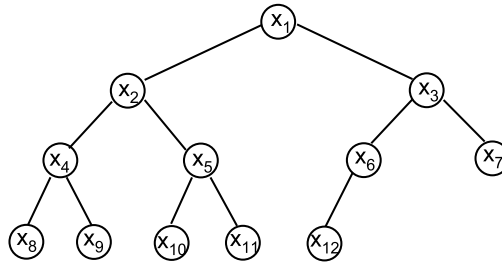
Conform teoremei 3.19, înălțimea arborelui de decizie va fi $\geq \lceil \log_3(2n + 1) \rceil$.

În cazul $n = 4$, înălțimea va fi ≥ 2 . Procedura care efectuează doar două cântăriri este ilustrată în figura 3.10.

În finalul acestui paragraf menționăm că în problemele de decizie marginea inferioară $\lceil \log_m k \rceil$ dată în (3.5) este una teoretică. În anumite situații este posibilă construcția unei proceduri din care să se poată trage concluzia cu doar $\lceil \log_m k \rceil$ decizii, dar există de asemenea situații când acest număr nu poate fi atins.

3.3.3 Arbori de sortare

Vrem să ordonăm crescător o listă x_1, x_2, \dots, x_n de numere două câte două distincte. Algoritmii folosiți pentru a rezolva această problemă presupun compararea a două numere și (dacă e cazul) schimbarea între ele. Acest fel de procedură se poate reprezenta printr-un arbore de decizie, unde un vârf conține o comparație a două numere, x_i și x_j . Deoarece există două rezultate posibile ($x_i < x_j$ și $x_i > x_j$), arborele de decizie este unul binar.

Figura 3.11: Arborele binar asociat listei X .

Un algoritm simplu folosit pentru sortare este așa-numitul *algorithm bobble sort*, unde se compară de fiecare dată două numere consecutive, schimbându-le între ele dacă nu sunt în ordinea cerută.

Algoritm 1 (Algoritm bobble sort)

pentru $j = 1, \dots, n$
 pentru $i = 1, \dots, n - j$
 dacă $x_i > x_{i+1}$ schimbă x_i, x_{i+1}

Un algoritm mult mai eficient, care folosește arbori binari și are nevoie de mai puține comparații, este algoritmul *heap-sort*. În continuare vom descrie pașii acestui algoritm.

Construirea unui arbore neordonat

Începem prin a atașa elementele listei $X = \{x_1, x_2, \dots, x_n\}$ vârfurilor unui arbore binar, astfel:

x_1 rădăcina – nivel 0
 x_2, x_3 – nivel 1
 x_4, x_5, x_6, x_7 – nivel 2
 \dots

Astfel, în acest arbore, x_k este tată pentru x_{2k} și x_{2k+1} , când $2k + 1 \leq n$. Ultimul nivel este de obicei incomplet, iar dacă n este par, atunci $x_{n/2}$ are doar un fiu.

Algoritmul heap-sort continuă în felul următor: mai întâi, arborele nesortat este transformat într-un tip special de arbore pe care îl vom numi *heap*. Apoi heap-ul este transformat într-o listă ordonată y_1, y_2, \dots, y_n .

Definiția 3.24 Un **heap** este un arbore binar, cu un număr x_i asociat fiecărui vârf și cu proprietatea că fiecare tată este mai mic decât fiii săi ($x_i < x_{2i}$, $x_i < x_{2i+1}$).

Transformarea unui arbore neordonat într-un heap

Vom începe să comparăm tații (vârfurile interne) în ordine inversă (de la cel mai mare nivel spre rădăcină). Presupunem că, atunci când ajungem la vârful x_k , cei doi subarbori cu rădăcinile x_{2k} și x_{2k+1} au fost deja transformați în heap-uri.

Dacă $x_k < x_{k+1}$ și $x_k < x_{2k+1}$ nu facem nimic, întrucât subarborii cu rădăcina x_k este deja heap.

Dacă nu, stocăm x_k temporar și mutăm pe cel mai mic dintre fiii săi x_{2k} și x_{2k+1} în vârful vacant. Acest lucru creează o nouă poziție vacantă.

Dacă x_k este mai mic decât fiii vârfului vacant sau dacă nu are fii, vom ocupa poziția vacantă cu x_k .

Dacă nu, ocupăm poziția vacantă cu cel mai mic dintre fii și continuăm. În final vom găsi un loc pentru x_k când am ajuns la o frunză, dacă nu înainte.

Fie **heap(k,n)** procedura care, dat fiind un vârf x_k cu proprietatea că arborii cu rădăcinile în x_{2k} și x_{2k+1} sunt heap-uri, transformă subarborii cu rădăcina x_k într-un heap. Aplicând această procedură pentru $k = \lfloor \frac{1}{2}n \rfloor, \lfloor \frac{1}{2}n \rfloor - 1, \dots, 2, 1$, întregul arbore va fi transformat într-un heap.

Transformarea unui heap într-o listă ordonată

Intr-un heap, rădăcina are cea mai mică valoare, deci ea va fi primul element y_1 al listei ordonate.

Locul vacant rămas la rădăcină este ocupat de ultima valoare x_n și vechea frunză x_n este înlăturată.

Avem acum un arbore cu $n - 1$ vârfuri și doi subarbori cu rădăcinile x_2 și x_3 , care sunt heap-uri. Aplicăm acum procedura **heap(1,n-1)** pentru a reface proprietatea de heap a întregului arbore.

În noul arbore (care este heap) rădăcina are cea mai mică dintre valorile rămase și aceasta este atribuită lui y_2 .

Ultima frunză este înlăturată.

Refacem acum heap-ul prin **heap(1,n-2)**

și așa mai departe.

Exemplul 3.8 *Considerăm lista*

$$X = \{79, 27, 81, 49, 50, 11, 99, 87, 40, 95, 55, 73\}$$

Arborele binar asociat acestei liste este reprezentat în figura 3.11, iar pașii construcției primului heap sunt prezentați în figura 3.12. Pasul următor este prezentat în figura 3.13.

3.4 Arbori binari și coduri binare

3.4.1 Arbori binari

Definiția unui arbore binar a fost dată la pag. 86. Într-un arbore binar, considerăm că un fiu poate fi sau un **fiu stâng**, sau un **fiu drept**, cu alte cuvinte vom face distincție între fii.

Definiția 3.25 Fie $T = (V, E)$ un arbore binar și $v \in V$. Un **subarbore stâng (drept)** al vârfului v este un subarbore binar indus de fiul stâng (drept) al lui v și de toți descendenții acestora.

Deci, putem considera un arbore ca având trei componente: rădăcina și subarborii săi stâng și drept. Acest lucru ne permite să enunțăm următoarea *proprietate recursivă a unui arbore binar*: Dacă T este un arbore binar de înălțime h , atunci subarborii săi stâng și drept au ambii înălțimea $\leq h - 1$, cu egalitate pentru cel puțin unul din ei. Demonstrația acestei proprietăți este imediată, prin inducție după h .

Observația 3.9 Alte două proprietăți imediate ale arborilor binari sunt următoarele:

1. Un arbore binar complet de înălțime h are $2^{h+1} - 1$ vârfuri.

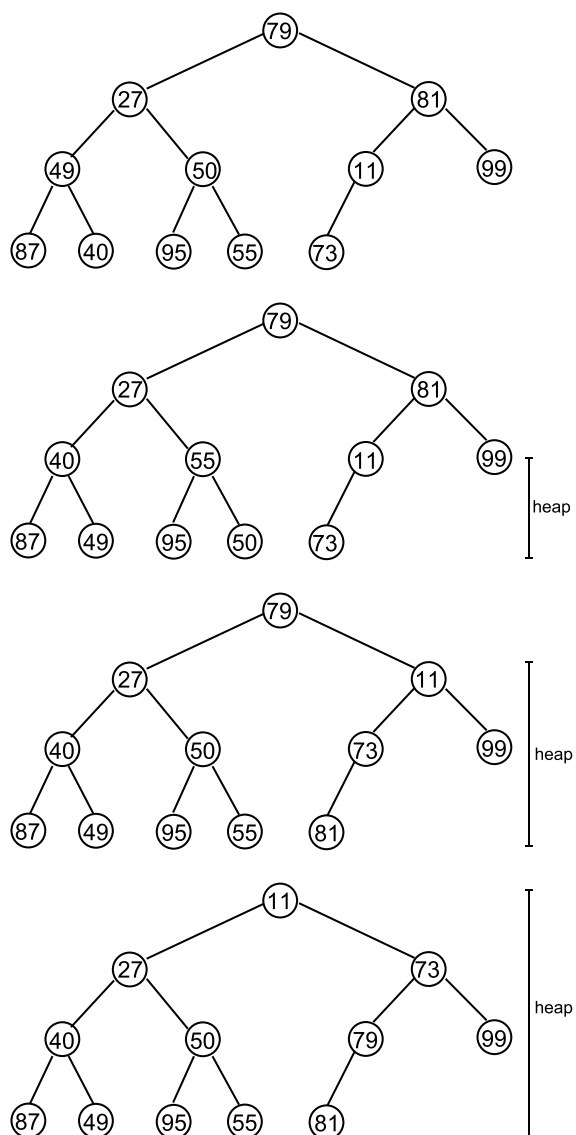


Figura 3.12: Pasul 1: Transformarea listei neordonate X într-un heap.

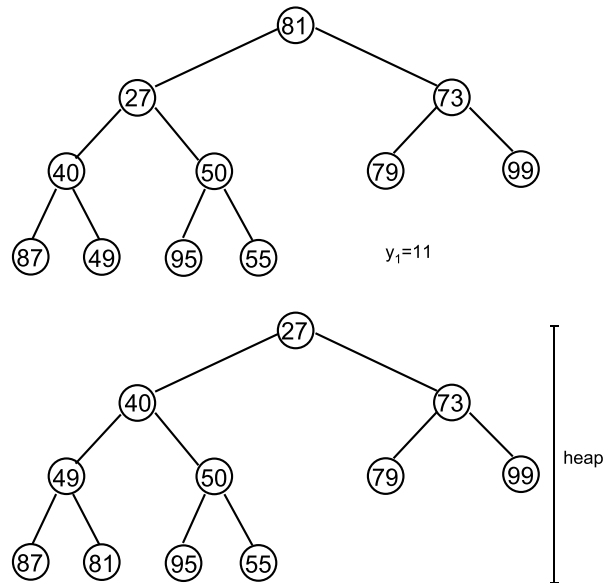


Figura 3.13: Pasul 2: Extrgerea rădăcinii $y_1 = 11$, ștergerea ultimei frunze și construcția din nou a unui heap.

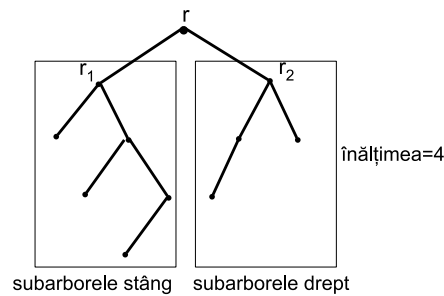


Figura 3.14: Un arbore binar de înălțime 4, cu rădăcina în r , și subarborii săi stâng și drept de înălțime 3 resp. 2, cu rădăcinile r_1 resp. r_2 .

2. Orice arbore binar de înălțime h are cel mult $2^{h+1} - 1$ vârfuri.

Demonstrațiile acestor proprietăți rezultă imediat prin inducție.

3.4.2 Coduri binare

Pentru a codifica informația, computerul folosește șiruri de cifre binare (biți). Cel mai cunoscut mod de codificare este codificarea ASCII, care asociază fiecărui simbol șir format din 7 biți.

Definiția 3.26 Un **cod binar** este o asociere de șiruri de biți unei mulțimi de simboluri.

Pentru anumite aplicații, este de dorit să avem codificări în care codurile să aibă lungime variabilă. Deci să considerăm următoarele trei exemple de codificări:

- (c_1) $x_1 \rightarrow 00$ $x_2 \rightarrow 01$ $x_3 \rightarrow 10$ $x_4 \rightarrow 11$,
- (c_2) $x_1 \rightarrow 0$ $x_2 \rightarrow 10$ $x_3 \rightarrow 110$ $x_4 \rightarrow 111$,
- (c_3) $x_1 \rightarrow 0$ $x_2 \rightarrow 1$ $x_3 \rightarrow 00$ $x_4 \rightarrow 01$.

Codurile asociate șirului $x_1x_3x_1x_4$ sunt:

- (c_1) 00100011,
 (c_2) 01100111,
 (c_3) 000001.

Decodificarea se poate face în mod unic în primele două codificări, în timp ce în a treia codificare, șirul de biți se poate decodifica $x_1x_1x_1x_1x_1x_2$, dar și $x_3x_3x_4$. În concluzie, trebuie evitate aceste situații generate de faptul că un cod este începutul unui alt cod, și astfel nu se știe unde începe și unde se termină un cod. Pentru a evita această ambiguitate, introducem următoarea proprietate:

Definiția 3.27 *Un **cod prefix** este un cod binar cu proprietatea că niciun cod al unui simbol nu este un subșir inițial al unui cod asociat altui simbol. Numărul de biți dintr-un cod prefix al unui simbol se numește **lungimea codului**.*

Se poate vedea că în a treia codificare, codul $x_1 \rightarrow 0$ este șir inițial al codurilor $x_3 \rightarrow 00$ și $x_4 \rightarrow 01$, prin urmare (c_3) nu este un cod prefix.

Construcția codurilor prefix poate fi realizată prin folosirea arborilor binari.

Construcția codurilor prefix

Fie T un arbore binar. Dacă etichetăm fiecare muchie stângă cu 0 și fiecare muchie dreaptă cu 1, atunci fiecărei frunze i se asociază un cod format dintr-un șir de etichete ale muchiilor care compun unicul drum de la rădăcină la acea frunză. De exemplu,

literă	a	b	c	d	e	f	g
cod	000	0010	0011	0101	011	100	101

Se poate constata că orice drum de la rădăcină la frunză *nu* este un subdrum inițial pentru un alt drum de la rădăcină la o altă frunză, deci codurile sunt coduri prefix.

Coduri Huffman

Intr-un cod prefix, este natural să se dorească folosirea mai puținor biți pentru codificarea simbolurilor mai frecvente. O măsură a eficienței unei codificări este **lungimea medie ponderată (LMP)** a sa, unde lungimea fiecărui cod este înmulțită cu frecvența simbolului pe care îl codifică. În exemplul de mai sus, dacă frecvențele sunt

$$\begin{array}{c|ccccccc} \text{literă} & a & b & c & d & e & f & g \\ \hline \text{frecvență} & 0.2 & 0.05 & 0.1 & 0.1 & 0.25 & 0.15 & 0.15 \end{array}, \quad (3.6)$$

atunci lungimea medie ponderată este

$$3 \cdot 0.2 + 4 \cdot 0.05 + 4 \cdot 0.1 + 4 \cdot 0.1 + 3 \cdot 0.25 + 3 \cdot 0.15 + 3 \cdot 0.15 = 3.25.$$

Coeficienții frecvențelor sunt adâncimile frunzelor corespunzătoare din arbore.

Definiția 3.28 Fie T un arbore binar și s_1, s_2, \dots, s_l frunzele sale, astfel încât frunzei s_i îi este asociată ponderea w_i . Atunci **adâncimea medie ponderată (AMP)** a arborelui binar T este

$$\mathcal{W}(T) = \sum_{i=1}^l \text{depth}(s_i) \cdot w_i.$$

Un algoritm pentru construcția unui cod prefix, ale cărui cuvinte au cea mai mică LMP posibilă, a fost dat în 1952 de Huffman⁷ în teza sa de doctorat la MIT.

Algoritm 2 (Coduri prefix Huffman)

Intrare: o mulțime $S = \{s_1, \dots, s_l\}$ de simboluri

o listă $W = \{w_1, \dots, w_l\}$ de ponderi asociate lui S ($w_i \rightarrow s_i$)

Ieșire: un arbore binar reprezentând un cod prefix pentru mulțimea de simboluri S , cu codurile având LMP minimă

- se inițializează F ca o pădure de vârfuri izolate cu etichetele s_1, \dots, s_l , cu ponderile respective w_1, \dots, w_l
- pentru $i = 1, \dots, l - 1$
 - alegem din pădurea F doi arbori, T_0 și T_1 cu cele mai mici ponderi
 - se crează un nou arbore binar a cărui rădăcină are pe T_0 și T_1 ca și subarbori stâng, respectiv drept
 - se etichetează muchia spre T_0 cu 0 și muchia spre T_1 cu 1
 - se asociază noului arbore ponderea $w(T_0) + w(T_1)$
 - se înlocuiesc arborii T_0 și T_1 din pădurea F cu noul arbore
- retur un arbore binar asociat listei W

Definiția 3.29 Arborele binar rezultat prin aplicarea algoritmului 2 va fi numit **arbore Huffman** pentru lista de simboluri S . Codul prefix care corespunde acestui arbore se va numi **cod Huffman**.

Exemplul 3.9 Considerăm frecvențele mulțimii de simboluri

$S = \{a, b, c, d, e, f, g\}$, date prin tabelul (3.6). Pașii algoritmului Huffman sunt ilustrați în figurile 3.15 și 3.16. Codificarea Huffman va fi următoarea:

literă	a	b	c	d	e	f	g
cod	00	0010	0101	011	10	110	111

Lungimea medie ponderată a acestei codificări este

$$2 \cdot 0.2 + 4 \cdot 0.05 + 4 \cdot 0.1 + 3 \cdot 0.1 + 2 \cdot 0.25 + 3 \cdot 0.15 + 3 \cdot 0.15 = 2.7.$$

Schema de decodificare folosind arborele Huffman

Un cod dat determină un drum unic de la rădăcină la frunza corespunzătoare aceluși simbol. Pentru decodificare, se citește codul de la stânga la dreapta și se parcurge arborele pornind de la rădăcină. Când se ajunge la o frunză, simbolul corespunzător este înregistrat. Următorul bit începe un nou drum de la rădăcină la următorul simbol. Continuăm până când toți biții codului au fost citați.

⁷David Albert Huffman (1925-1999), inginer electronist american.

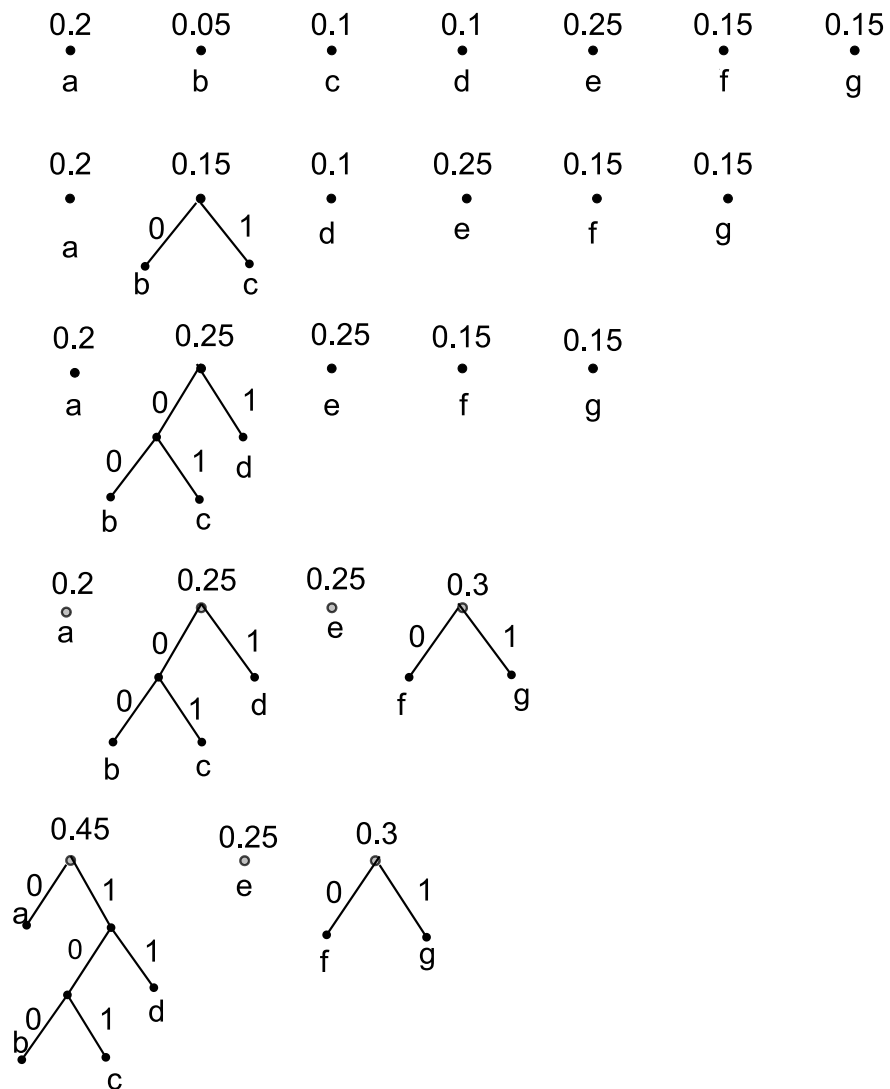


Figura 3.15: Iterațiile 1-4 din algoritmul lui Huffman pentru exemplul 3.9.

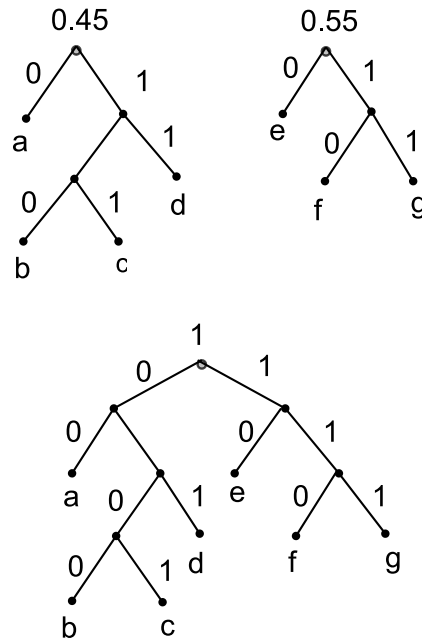


Figura 3.16: Iterațiile 5-6 din algoritmul lui Huffman pentru exemplul 3.9.

Exemplu: Prin decodificarea următoarelor șiruri de biți

1100001010001110, 0100001111110011110, 11000010110011

se obțin respectiv șirurile de simboluri *facade*, *baggage*, *faced*.

Correctitudinea algoritmului Huffman

Trebuie să demonstrăm că algoritmul Huffman produce un cod prefix *optimal* (cu cea mai mică LMP). Acest lucru se poate arăta prin inducție, dar mai întâi avem nevoie de câteva rezultate preliminare.

Lema 3.20 *Dacă frunzelelor unui arbore binar le sunt asociate ponderi și dacă fiecărui vârf interior îi este asociată o pondere egală cu suma ponderilor copiilor săi, atunci AMP a arborelui va fi egală cu suma ponderilor vârfurilor interioare:*

$$\mathcal{W}(T) = \sum_{v \in I} w(v)$$

(am notat cu I mulțimea vârfurilor interioare ale lui T).

Demonstrație. Pentru fiecare frunză (simbol) s_k , notăm $d_k = \text{depth}(s_k)$. Contribuția simbolului s_k la membrul drept este $w_k \cdot d_k$, deoarece ponderea w_k apare la nivelele $0, 1, 2, \dots, d_k-1$, deci de d_k ori. Contribuția lui s_k la membrul stâng va fi (din definiția 3.28) $\text{depth}(s_k) \cdot w_k = d_k w_k$. ■

Teorema 3.21 *Fie $S = \{s_1, \dots, s_l\}$ o mulțime de simboluri. Pentru orice listă dată de ponderi $W = \{w_1, \dots, w_l\}$, un arbore Huffman are cea mai mică AMP posibilă dintre toți arborii binari care au aceste ponderi asociate frunzelor.*

Demonstrație. Folosim inducția după l .

Pentru $l = 2$, arborele are o rădăcină cu ponderea $w_1 + w_2$, suma ponderilor fiilor săi stâng și drept, prin urmare $AMP = 1 \cdot w_1 + 1 \cdot w_2$ și este minimă.

Pentru $l_0 \geq 2$ dat, presupunem că algoritmul 2 produce un arbore Huffman H cu AMP minimă pentru orice listă l_0 de ponderi. Fie $w_1, w_2, \dots, w_{l_0+1}$ o listă de $l_0 + 1$ ponderi și presupunem că w_1 și w_2 sunt cele mai mici două ponderi, alese primele de către algoritm. Astfel, frunzele s_1 și s_2 sunt frați. Fie y tatăl lor și fie $\overline{H} = H - s_1 - s_2$. Atunci \overline{H} coincide cu arborele obținut prin aplicarea algoritmului Huffman pentru mulțimea $\{y, s_3, s_4, \dots, s_{l_0+1}\}$, cu $w(y) = w_1 + w_2$. Dar,

$$\text{depth}(s_1) = \text{depth}(s_2) = \text{depth}(y) + 1, \quad (\text{în } H)$$

prin urmare

$$\begin{aligned} \mathcal{W}(H) &= \sum_{i=1}^{l_0+1} \text{depth}(s_i) \cdot w_i \\ &= \text{depth}(s_1) \cdot w_1 + \text{depth}(s_2) \cdot w_2 + \sum_{i=3}^{l_0+1} \text{depth}(s_i) \cdot w_i \\ &= w_1 + w_2 + \text{depth}(y) \cdot (w_1 + w_2) + \sum_{i=3}^{l_0+1} \text{depth}(s_i) \cdot w_i \\ &= w_1 + w_2 + \mathcal{W}(\overline{H}). \end{aligned}$$

Din ipoteza inducției, \overline{H} este optimal între toți arborii binari care au ponderile $w_1 + w_2, w_3, \dots, w_{l_0+1}$ asociate frunzelor.

Presupunem acum că T^* este un arbore binar optimal pentru ponderile $w_1, w_2, w_3, \dots, w_{l_0+1}$. Fie x un vârf interior al lui T^* , cu cea mai mare adâncime, și notăm cu z și v copiii săi. Fără să pierdem generalitatea, putem presupune că ponderile lui z și v sunt w_1 și w_2 , deoarece în caz contrar am fi putut schimba ponderile lor cu w_1 și w_2 pentru a produce un arbore cu AMP mai mică.

Considerăm arborele $\overline{T} = T^* - z - v$. Atunci

$$\mathcal{W}(T^*) = \mathcal{W}(\overline{T}) + w_1 + w_2.$$

Dar \overline{T} este un arbore binar cu frunzele având ponderile $w_1 + w_2, w_3, \dots, w_{l_0+1}$, și prin urmare

$$\mathcal{W}(\overline{T}) \geq \mathcal{W}(\overline{H}).$$

Astfel, $\mathcal{W}(T^*) \geq \mathcal{W}(H)$ și în consecință H este optimal. ■

3.5 Arbori de acoperire

Definiția 3.30 Un **arbore de acoperire** pentru graful $G = (V, E)$ este un subgraf de acoperire⁸ care este un arbore.

Intrebarea care se pune în mod natural este: dat fiind un graf conex, există un arbore de acoperire, și în caz afirmativ, cum se poate construi un arbore de acoperire?

Înainte de a prezenta câteva metode de construcție a arborilor de acoperire, avem nevoie de câteva noțiuni.

⁸Definiția 3.2, pag. 67.

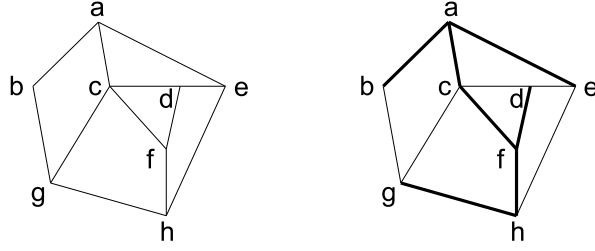


Figura 3.17: Un graf și unul din arborii săi de acoperire.

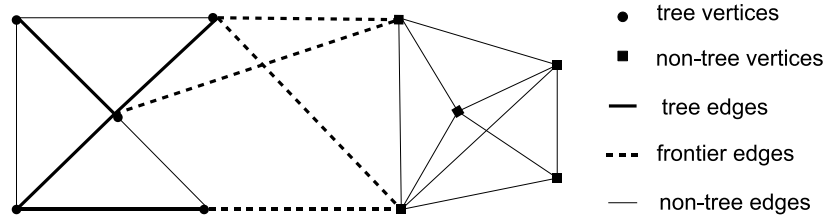


Figura 3.18: Un arbore într-un graf și muchiile sale frontieră.

Definiția 3.31 Fie $G = (V, E)$ un graf și $G_1 = (V_1, E_1)$ un subgraf al lui G . O muchie a lui G se numește o **muchie frontieră** pentru G_1 în G , dacă ea are o extremitate în V_1 și cealaltă extremitate în $V \setminus V_1$.

Mulți algoritmi de construcție a arborilor de acoperire se bazează pe următorul rezultat.

Propoziția 3.22 Fie $G = (V, E)$ un graf, T un arbore care este un subgraf al grafului G , și e o muchie frontieră pentru T . Atunci subgraful $T + e$ este de asemenea un arbore.

Demonstrație. Adăugarea unei muchii frontieră e la arborele T nu poate crea un ciclu, întrucât una din extremitățile lui e nu aparține lui T . ■

Să menționăm că adăugarea muchiei e la T implică și adăugarea unui vârf nou în T .

Construcția de bază a arborilor

În schema de bază pentru construcția arborilor se folosesc etichete pentru vârfuri, pentru a marca ordinea în care vârfurile au fost adăugate arborelui. Prin **etichetarea vârfurilor** înțelegem atribuirea etichetelor $0, 1, 2, \dots, n-1$ (câteodată $1, 2, \dots, n$) celor n vârfuri ale grafului.

Definiția 3.32 Fie $G = (V, E)$ un graf și $x \in V$ un vârf. **Componenta lui x în G** , notată $C_G(x)$, este subgraful lui G indus de mulțimea de vârfuri care sunt accesibile din x .

⁹Vezi definiția 3.7, pag. 70.

Astfel, dacă un graf $G = (V, E)$ este conex, atunci $C_G(x) = G$ pentru orice $x \in V$. Reciproc, dacă există $x \in V$ astfel încât $C_G(x) = G$, atunci graful G este conex.

În continuare prezentăm un algoritm de bază pentru construcția unui arbore de acoperire pentru componenta $C_G(x)$. Acesta începe cu o muchie și adaugă arborelui muchii frontieră succesiv, până când sunt adăugate toate vârfurile grafului dat.

Algoritm 3 (Algoritm de bază pentru construcția unui arbore de acoperire, cu etichetarea vârfurilor)

Intrare: un graf $G = (V, E)$ și un vârf inițial $u \in V$.

Ieșire: un arbore de acoperire T pentru $C_G(u)$ și o etichetare a vârfurilor pentru $C_G(u)$.

- se inițializează arborele T cu vârfurile u
- se etichetează cu 0 vârfurile u
- se inițializează contorul de etichetare $i = 1$
- cât timp T nu acoperă $C_G(u)$
 - se alege o muchie frontieră e pentru T
 - fie v extremitatea muchiei e care nu aparține lui T
 - se adaugă muchia e și vârfurile v arborelui T
 - se etichetează cu i pe vârfurile v
 - $i = i + 1$
- retur arbore T și etichetarea vârfurilor lui $C_G(u)$

Observația 3.10 (unicitatea) Să menționăm pentru început că, fără reguli de alegere a unei muchii frontieră, arborele generat de algoritmul 3 nu este unic. Unicitatea depinde de anumite priorități asociate muchiilor sau vârfurilor.

Observația 3.11 Ori de câte ori este aplicat algoritmul 3, arborele de acoperire rezultat poate fi privit ca și un arbore orientat cu rădăcina u .

Propoziția 3.23 Dacă o execuție a algoritmului 3 începe cu un vârf v al grafului G , atunci subgraful constând din vârfurile etichetate și muchiile arborelui este un arbore de acoperire al componentei $C_G(v)$.

Corolarul 3.24 Un graf este conex dacă și numai dacă algoritmul 3 etichetează toate vârfurile sale.

În cazul în care graful G nu este conex, vom obține un algoritm de construcție a unei păduri.

Definiția 3.33 O **pădure de acoperire** a unui graf G este o mulțime de arbori, astfel încât fiecare arbore este un arbore de acoperire a câte unei componente conexe a lui G .

Pentru construcția unei păduri dăm următorul algoritm de bază.

Algoritm 4 (Algoritm de bază pentru construcția unei păduri, cu etichetarea vârfurilor)

Intrare: un graf $G = (V_G, E_G)$

Ieșire: o pădure de acoperire F a lui G și o etichetare a vârfurilor lui G .

- se inițializează pădurea F cu graful vid

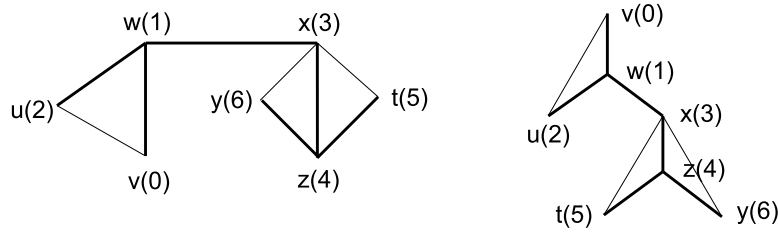


Figura 3.19: Arborele DFS.

- se inițializează contorul componente cu $t = 1$
- cât timp pădurea F nu acoperă pe G (nu are încă $|V_G|$ vârfuri)
 - se alege un vârf v care nu este în pădurea F
 - se folosește algoritmul pentru arbori, pentru a construi un arbore de acoperire T_t în $C_G(v)$
 - se adună $|V_F|$ la fiecare etichetă a vârfurilor arborelui T_t
 - se adaugă arborele T_t la pădurea F
 - $t = t + 1$
- retur pădurea F și etichetarea vârfurilor lui G

3.5.1 Căutare în adâncime și căutare în lărgime

Căutarea în adâncime și căutarea în lărgime sunt de fapt algoritmi de construire de arbori de acoperire, bazați pe algoritmul 3. Chiar dacă ei sunt folosiți câteodată pentru a căuta ceva, cuvântul «de căutare» ne poate oarecum induce în eroare, dar este prea înrădăcinat ca să-l putem schimba.

Căutare în adâncime (DFS)

Ideea algoritmului DFS¹⁰ este de a selecta o muchie frontieră incidentă celui mai recent vârf adăugat în arborele parțial (adică vârfului cu cea mai mare etichetă). Când acest lucru nu este posibil, algoritmul se întoarce la penultimul vârf adăugat (adică părintele) și încearcă din nou. Astfel, căutarea avansează adânc în graf (de unde numele *în adâncime*).

În continuare vom descrie algoritmul DFS. Un exemplu de arbore DFS este ilustrat în figura 3.19. În timpul algoritmului, eticheta asociată unui vârf w se notează cu $df(w)$.

Algoritm 5 (Căutare în adâncime)

Intrare: un graf conex $G = (V, E)$ și un vârf de plecare $v \in V$

Ieșire: un arbore de acoperire în adâncime T și o etichetare a vârfurilor lui G

- se inițializează arborele T cu vârful v
- se inițializează mulțimea de muchii frontieră pentru T cu \emptyset
- $df(v) = 0$
- se inițializează contorul de etichete $i = 1$
- cât timp T nu acoperă pe G (nu are încă $|V|$ vârfuri)
 - se actualizează mulțimea muchiilor frontieră pentru T
 - fie e o muchie frontieră pentru T a cărei extremitate etichetată

¹⁰DFS este prescurtarea termenului englez *Depth-First Search*, folosit pentru căutarea în adâncime.

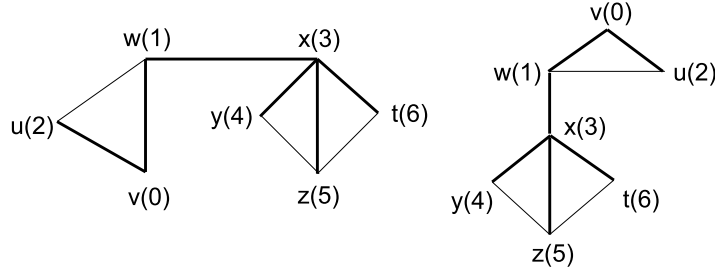


Figura 3.20: Arborele BFS.

- are **cel mai mare** număr df posibil.
- fie w extremitatea neetichetată a muchiei e
- se adaugă muchia e (și vârful w) arborelui T
- $df(w) = i$
- $i = i + 1$
- retur arborele T împreună cu numerele df

Căutare în lărgime (BFS)

În timp ce în DFS se avansează de la vârful curent la un nou vârf ori de câte ori este posibil, în BFS¹¹ se verifică toate vârfurile adiacente vârfului curent înainte de a trece la următorul. Selecționarea muchiilor frontieră se va face cât mai aproape cu putință de vârful inițial, sau, în termeni de priorități, muchiile frontieră incidente vârfurilor având cea mai mică etichetă posibilă vor avea cea mai mare prioritate. Un exemplu de arbore BFS este dat în figura 3.20.

Algoritm 6 (Căutare în lărgime)

Algoritmul este identic cu algoritmul 5 (DFS), cu excepția cuvintelor **cel mai mare**, care se înlocuiesc cu **cel mai mic**.

Propoziția 3.25 Când BFS este aplicat unui graf conex $G = (V, E)$, extremitățile fiecărei muchii din G care nu aparține arborelui sunt fie la același nivel, fie la nivele consecutive (sau, echivalent, dacă se notează cu r vârful de plecare, distanțele¹² la r sunt fie egale, fie diferă cu unu).

Demonstrație. Fie $T = (V, E')$ un arbore de acoperire rezultat prin aplicarea algoritmului BFS pornind din vârful r .

Presupunem că există muchia $e = xy \in E \setminus E'$ astfel încât

$$d(r, x) = p, \quad d(r, y) = p + s, \quad s > 1.$$

Dacă se notează cu ω unicul lanț în T dintre r și x , atunci $\omega \cup e$ formează un lanț între r și y de lungime $p + 1 < p + s$, contradicție cu proprietatea că $p + s$ este lungimea celui mai scurt lanț dintre r și y . ■

Teorema 3.26 (proprietatea de arbore al distanțelor) Fie $G = (V, E)$ graf și fie T arborele construit aplicând algoritmul BFS, pornind din vârful $v \in V$. Atunci pentru orice vârf x , $d(v, x)$ este egală cu lungimea unicului

¹¹BFS este prescurtarea termenului englez *Breadth-First Search* folosit pentru căutarea în lărgime.

¹²Definiția distanței dintre două vârfuri a fost dată la pag. 69.

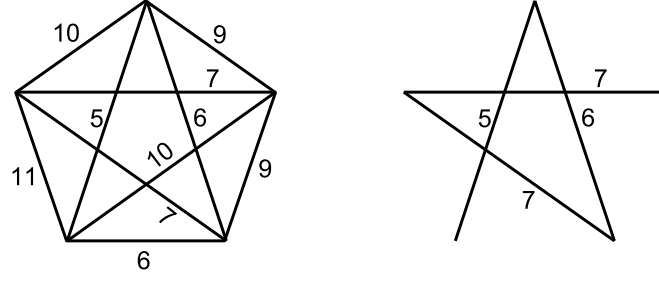


Figura 3.21: Un graf și un arbore economic al său.

lanț în T dintre v și x . Cu alte cuvinte, arborele obținut prin algoritmul BFS pornind din v este un arbore al distanțelor față de v .

Demonstrație. Reamintim că $d(v, x)$ este distanța dintre v și x , adică lungimea celui mai scurt lanț în G dintre v și x .

Notăm $k = d(v, x)$. Construim partiția lui V alcătuită din clasele

$$\ell_v(p) = \{y \in V, d(v, y) = p\}, \quad p = 0, 1, 2, \dots, m.$$

Prin urmare $x \in \ell_v(k)$ și părintele său în T va fi un vârf $y_1 \in \ell_v(k-1)$. Părintele lui y_1 în T va fi un vârf $y_2 \in \ell_v(k-2)$, iar continuând în acest mod ajungem la vârful $y_k \in \ell_v(0)$, adică $y_k = v$. Prin urmare am construit lanțul $\langle v, y_{k-1}, y_{k-2}, \dots, y_2, y_1, x \rangle$ în T , de lungime k . ■

3.5.2 Grafuri cu ponderi și arbori economici

Definiția 3.34 Un **graf cu ponderi** este un graf în care fiecărei muchii i se asociază un număr, numit ponderea sa. Ponderea muchiei e va fi notată cu $w(e)$.

Grafurile cu ponderi apar frecvent în aplicații; în grafurile de comunicații ele pot reprezenta costuri de transport, timp de călătorie, distanța spațială, prețul comunicației sau al întreținerii.

Presupunem că mai multe calculatoare având o locație fixă trebuie legate între ele pentru a forma o rețea (vezi figura 3.21). Costul creării unei conexiuni directe între o pereche de calculatoare este cunoscut pentru fiecare pereche posibilă și este reprezentat prin ponderea muchiei. Desigur avem nevoie de cea mai ieftină conexiune. Dacă asociem un graf în care vârfurile sunt calculatoarele, iar muchiile sunt cablurile care le leagă și ponderea fiecărei muchii este costul conexiunii, atunci scopul nostru va fi să determinăm un arbore de acoperire de pondere minimă.

Definiția 3.35 Fie $G = (V, E)$ un graf conex cu ponderi, $T = (V, E_1)$ un arbore de acoperire al lui G și $\omega = \langle e_1, e_2, \dots, e_n \rangle$ un lanț în G .

Ponderea lanțului $\omega = \langle e_1, e_2, \dots, e_n \rangle$ este suma ponderilor muchiilor sale:

$$w(\omega) = \sum_{i=1}^n w(e_i).$$

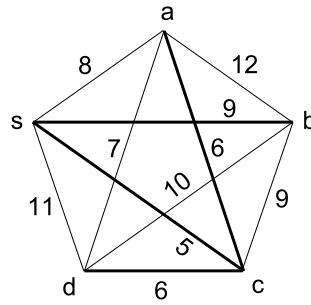


Figura 3.22: Arborele economic obținut cu algoritmul 7.

Ponderea arborelui de acoperire T este suma ponderilor muchiilor sale:

$$w(T) = \sum_{e \in E_1} w(e).$$

Un arbore T_{\min} pentru care $w(T_{\min})$ este minim se numește **arbore de acoperire minimă** sau **arbore economic**. Deci,

$$w(T_{\min}) = \min\{w(T), T \text{ arbore de acoperire al lui } G\}.$$

Problema arborelui economic: Fie G graf conex cu ponderi. Să se găsească un arbore de acoperire al lui G a cărui pondere să fie minimă.

O soluție naivă ar fi să scriem toți arborii de acoperire, să calculăm ponderile lor și să-l alegem pe cel de pondere minimă. În acest caz, dacă graful G are n vârfuri și este complet, atunci există n^{n-2} arbori de acoperire,¹³ care sunt imposibil de analizat în timp real. Prin urmare este nevoie de algoritmi mai eficienți pentru găsirea unui arbore economic.

Un asemenea algoritm este cunoscut sub numele de **algoritmul lui Prim**. Acest algoritm a fost descoperit pentru prima oară în 1930 de către Jarník¹⁴ și mai târziu, independent, de către Prim¹⁵ în 1957 și Dijkstra în 1959. De aceea mai este câteodată numit **algoritmul DJP** sau **algoritmul lui Jarník**.

Algoritm 7 (Algoritmul lui Prim)

Intrare: un graf conex cu ponderi $G = (V, E)$

Ieșire: un arbore economic T

- se alege un vârf arbitrar s al grafului G
- se inițializează arborele Prim T cu vârful s
- se inițializează mulțimea muchiilor frontieră cu \emptyset
- cât timp T nu acoperă graful G (nu are încă $|V|$ vârfuri)
 - se actualizează mulțimea de muchii frontieră pentru T
 - fie e o muchie frontieră cu cea mai mică pondere $w(e)$
 - se adaugă muchia e la arborele T
- retur arborele Prim T

¹³Acest rezultat este cunoscut ca și *formula lui Cayley* referitoare la arbori de acoperire.

¹⁴Vojtěch Jarník (1897-1970), matematician ceh.

¹⁵Robert Clay Prim (b. 1921), matematician și calculatorist american.

Pentru graful din figura 3.22, pașii algoritmului Prim pornind cu vârful s sunt dați în următorul tabel:

pas	muchii frontieră	muchii adăugate
1	sd, sa, sc	sc
2	cd, ca, cb, sa, sd, sb	ac
3	ab, bc, cd, sb, sd	cd
4	ab, bc, db, sb	sb

Teorema 3.27 (Corectitudinea algoritmului lui Prim) *Arborele rezultat în urma aplicării algoritmului lui Prim este un arbore economic.*

Demonstrație. Fie $G = (V, E)$ un graf conex cu ponderi, T arborele rezultat prin aplicarea algoritmului Prim grafului G și Y un arbore economic al lui G . Notăm cu $T_k = (V_k, E_k)$ arborele lui Prim construit după k iterații.

Pentru simplitate începem cu cazul în care toate muchiile au ponderi diferite.

⊗ Dacă $Y = T$, atunci T este un arbore economic. Dacă nu, ar trebui să obținem o contradicție.

Deci, presupunem $Y \neq T$. Fie e prima muchie considerată de algoritm, care este în T dar nu în Y . Fie $T_i = (V_i, E_i)$ mulțimea de vârfuri conectate de muchiile adăugate înaintea lui e . Prin urmare, o extremitate a lui e este în V_i și cealaltă nu este în V_i . Deoarece Y este un arbore de acoperire al lui G , există un lanț în Y care unește cele două extremități ale lui e . Când ne deplasăm de-a lungul acestui lanț, trebuie să întâlnim o muchie f care unește un vârf din V_i cu un vârf care nu e în V_i . La a $(i + 1)$ -a iterație, când e a fost adăugată la T , f a fost o muchie frontieră, deci f ar fi putut fi și ea adăugată la T . Deoarece f nu a fost adăugată, putem trage concluzia că

$$w(f) > w(e). \quad (3.7)$$

Considerăm acum subgraful $Y_1 = Y - f + e$, care este un arbore.

Intr-adevăr, $Y + e$ are un ciclu care conține pe f , și prin urmare $Y + e - f$ nu are nici un ciclu și este conex. Să mai observăm că Y_1 are în T cu o muchie mai mult decât avea Y . Pentru Y_1 avem

$$w(Y_1) = w(Y) - w(f) + w(e) < w(Y), \quad (3.8)$$

care este o contradicție cu faptul că Y este un arbore economic. Deci, $T = Y$ și prin urmare T este un arbore economic.

Pentru a finaliza demonstrația, trebuie să considerăm cazul în care admitem ponderi egale. Demonstrația de mai sus funcționează, până în punctul în care tragem concluzia (3.7). Aceasta va fi înlocuită cu $w(f) \geq w(e)$, și prin urmare inegalitatea (3.8) se schimbă în $w(Y_1) \leq w(Y)$. În acest punct nu vom mai avea o contradicție, ci concluzia că Y_1 este de asemenea un arbore economic. Vom «înlocui» acum Y cu Y_1 și vom repeta demonstrația din punctul ⊗, pentru «noul» arbore economic Y . Intrucât Y_1 are în T cu o muchie mai mult decât Y , algoritmul se va termina după cel mult $|V| - 1$ pași, unde în punctul ⊗ trebuie să avem $Y = T$. ■

Incheiem cu mențiunea că, în cazul în care oricare două muchii au ponderi diferite, arborele economic este unic. În cazul când avem ponderi egale, este

posibil să existe mai mulți arbori economici, iar algoritmul lui Prim va găsi pe unul dintre ei.

Un alt algoritm folosit pentru găsirea unui arbore economic a fost dat de Kruskal¹⁶ în 1956. Algoritmul ordonează la început muchiile după ponderile lor și apoi ia din această listă câte o muchie și o adaugă la graful parțial, cu condiția ca aceasta să nu închidă un ciclu.

Algoritm 8 (Algoritmul lui Kruskal)

Intrare: *un graf conex cu ponderi* $G = (V, E)$.

Ieșire: *un arbore economic* T

- se inițializează graful Kruskal K ca o pădure unde fiecare vârf din V este un arbore separat
- se crează o mulțime S conținând toate muchiile din G
- cât timp S este nevidă
 - se elimină muchia cu cea mai mică pondere din S
 - dacă acea muchie nu închide un ciclu în K (în cazul când ar fi adăugată la K), atunci se adaugă la K
 - altfel se elimină
- retur arbore Kruskal K

Observația 3.12 *La pasul în care se verifică dacă muchia e închide un ciclu, se verifică de fapt dacă ea conectează doi arbori diferiți ai pădurii K . Când e adăugată grafului K , ea va uni cei doi arbori într-un singur arbore. Dacă graful G nu este conex, atunci algoritmul 8 nu creează un arbore economic, ci o pădure economică.*

Teorema 3.28 (Corectitudinea algoritmului Kruskal) *Fie G un graf conex cu ponderi și fie K un subgraf al lui G care rezultă din aplicarea algoritmului Kruskal. Atunci K este un arbore economic.*

Demonstrație. Mai întâi, prin construcție, K nu poate avea cicluri. Apoi K este conex. Intr-adevăr, dacă presupunem că K are două componente, atunci într-un punct al algoritmului se adaugă muchia de pondere minimă care unește cele două componente. Astfel, K este un arbore de acoperire al lui G .

Pentru simplitate, începem cu cazul în care toate muchiile au ponderi diferite. Fie Y un arbore economic.

★ Dacă $Y = K$, atunci K este un arbore economic. Dacă nu, ar trebui să obținem o contradicție.

Deci să presupunem că $Y \neq K$. Fie e prima muchie considerată de algoritm, care este în K dar nu în Y și fie K_i graful construit până în momentul când a fost adăugată muchia e . Atunci $Y + e$ are un ciclu (nu se poate adăuga o muchie la un arbore de acoperire și să se obțină tot un arbore). Acest ciclu conține o altă muchie f care, la pasul din algoritm în care e a fost adăugată la K_i , nu a fost considerată de către algoritm. Motivul pentru care f nu a fost considerată este că $w(f) > w(e)$, și nu pentru că f ar fi închis un ciclu. (Intr-adevăr, dacă f ar fi închis un ciclu dacă ar fi fost adăugată la K_i , atunci Y ar fi conținut un ciclu, anume $K_i + f$. Deci f a fost o muchie frontieră la începutul iterației $(i + 1)$).

¹⁶Josef Bernard Kruskal (b. 1928), matematician și statistician american.

Subgraful $Y_1 = Y + e - f$ este de asemenea un arbore de acoperire, cu ponderea

$$w(Y_1) = w(Y) + w(e) - w(f) < w(Y), \quad (3.9)$$

care este o contradicție cu faptul că Y este un arbore economic. Astfel presupunerea $K \neq Y$ este falsă, deci $K = Y$, prin urmare K este un arbore economic.

În încheiere, trebuie să considerăm de asemenea cazul când anumite ponderi sunt egale. Repetăm raționamentul de mai sus, dar în momentul când am ajuns la inegalitatea (3.9), aceasta se transformă în

$$w(Y_1) = w(Y) + w(e) - w(f) \leq w(Y).$$

Aceasta va implica faptul că Y_1 este de asemenea arbore economic. În acest caz vom «înlocui» Y cu Y_1 și vom repeta demonstrația din punctul ★ cu «noul» Y , până când vom obține $Y = K$. ■

Observația 3.13 *O altă argumentare pentru corectitudine va fi dată în exemplul 3.16, pag. 112.*

3.5.3 Arbore de acoperire minimă în grafuri orientate

Definiția 3.36 *Fie $D = (V, A)$ un graf orientat. Un **arbore de acoperire cu rădăcină** al lui D este un arbore cu rădăcină (vezi definiția 3.21, pag. 85) care conține toate vârfurile lui D .*

Propoziția 3.29 *Fie $D = (V, A)$ un graf orientat tare conex și fie r un vârf arbitrar. Atunci există un arbore de acoperire al lui D , cu rădăcina r .*

Ca și în cazul grafurilor, putem asocia unui graf orientat $D = (V, A)$ o funcție pondere $w : A \rightarrow \mathbb{R}_+$, și putem obține un **graf orientat cu ponderi**. **Ponderea** grafului D este definită ca și

$$w(D) = \sum_{a \in A} w(a).$$

Pentru un graf orientat tare conex cu ponderi $D = (V, A)$ și un vârf fixat $r \in V$, vrem să găsim un arbore orientat de acoperire minimă $T^*(r)$, cu rădăcina r , în sensul că

$$w(T^*(r)) \leq w(T(r)),$$

pentru orice alt arbore de acoperire $T(r)$ al lui D , cu rădăcina r . Un algoritm pentru a construi un asemenea arbore cu rădăcină a fost descoperit de Chu și Liu în 1965 și apoi independent de Edmonds în 1967. Înainte de a da acest algoritm, introducem câteva notații.

Pentru o mulțime de vârfuri $U \subseteq V$, introducem următoarele notații:

$$\begin{aligned} A^-(U) &= \{x \in V : (x, u) \in A \text{ și } u \in U\}, \\ A^+(U) &= \{x \in V : (u, x) \in A \text{ și } u \in U\}. \end{aligned}$$

Astfel, $A^-(U)$ este mulțimea extremităților inițiale ale arcelor cu extremitatea finală în U și $A^+(U)$ este mulțimea extremităților finale ale arcelor cu extremitatea finală în U .

Algoritm 9 (Chu, Liu, Edmonds)

Intrare: - un graf orientat tare conex $D = (V, A)$ cu
 ponderea $w : A \rightarrow \mathbb{R}_+$
 - un vârf r

Ieșire: un arbore de acoperire minimă orientat, cu rădăcina r

- pentru fiecare vârf v diferit de r , alegem arcul cu extremitatea finală v , de pondere minimă
- fie S mulțimea arcelor selectate (câte un arc pentru fiecare v)
- se inițializează $M = S$
- cât timp $T = (V, M)$ nu este arbore
 - fie C un ciclu orientat în T , notat $C = \{v_1, v_2, \dots, v_k, v_1\}$
 - pentru orice $v_i \in A^-(C) \setminus C$ se calculează
 $w(i, C) = \min\{w(i, v_j) + w(C) - w(v_{j-1}, v_j), v_j \in A^+(i) \cap C\}$,
unde v_{j-1} este extremitatea inițială a arcului (v_{j-1}, v_j) al lui C , cu extremitatea finală v_j
 - alegem $i_0 \in A^-(C)$ pentru care $w(i_0, C)$ este minim
 - fie $v_{j_0} \in C$ pentru care
 $w(i_0, C) = w(i_0, v_{j_0}) + w(C) - w(v_{j_0-1}, v_{j_0})$
 - $M = M - (v_{j_0-1}, v_{j_0}) + (i_0, v_{j_0})$
- retur M , arborele de acoperire cerut, cu rădăcina r

Teorema 3.30 (Corectitudinea algoritmului Chu-Liu-Edmonds) *Arborele cu rădăcină obținut prin aplicarea algoritmului 9 unui graf orientat $D = (V, A)$ pentru un vârf de plecare $r \in V$ este arborele de acoperire minimă pentru D , cu rădăcina r .*

Demonstrație. Mai întâi observăm că numărul de arce din S este $|V| - 1$ și rămâne neschimbat în timpul algoritmului. Prin urmare, în momentul când graful T nu mai conține cicluri, el va fi un arbore de acoperire cu rădăcină. Arătăm că, atunci când funcția pondere este injectivă, arborele de acoperire cu rădăcină are pondere minimă. Când ponderea nu este injectivă, folosim același raționament ca și în demonstrația teoremei 3.27 (corectitudinea algoritmului Prim).

Fie T arborele rezultat în urma aplicării algoritmului 9 și fie $T^* = T^*(r)$ arborele de acoperire minimă cu rădăcina r . Presupunem că $T^*(r) \neq T$. Atunci există arcele $a_1 = (i, j) \in T^* \setminus T$ și $a_2 = (k, j) \in T$.

Dacă $a_2 \in S$, atunci $w(a_2) < w(a_1)$. Dacă $a_2 \notin S$, atunci j este vârf al unui ciclu $C = \{v_j, v_{j+1}, \dots, v_{j-1}, v_j\}$ din S , $v_j = j$. Din criteriul de selecție al arcului $a_2 \in T$, tragem concluzia că

$$w(k, j) + w(C) - w(v_{j-1}, v_j) < w(i, j) + w(C) - w(v_{j-1}, v_j),$$

de unde $w(a_2) < w(a_1)$. Notăm cu T_1 arborele cu rădăcină obținut din T^* prin înlocuirea arcului a_1 cu arcul a_2 . Atunci $w(T_1) < w(T^*)$, care contrazice minimalitatea lui T^* . ■

Exemplul 3.10 *Considerăm graful orientat din figura 3.23 (stânga) și aplicăm algoritmul 9, cu vârf de plecare r .*

La primul pas, mulțimea S este

$$S = \{(r, 1), (4, 2), (2, 3), (8, 4), (2, 5), (5, 6), (1, 7), (6, 8)\}.$$

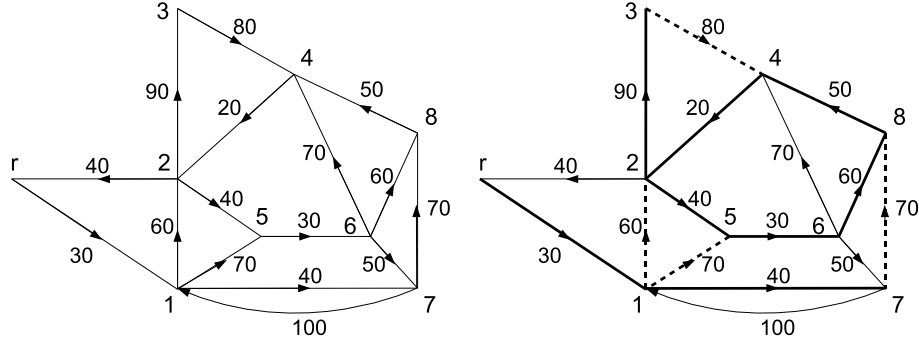


Figura 3.23: Stânga: Graful tare conex orientat din exemplul 3.10. Dreapta: mulțimea S (linie groasă) și arcele candidate pentru înlocuirea unui arc din ciclul C (linie punctată).

Această mulțime de arce formează ciclul $C = \langle 2, 5, 6, 8, 4, 2 \rangle$ (figura 3.23, dreapta) de pondere $w(C) = 200$. Mulțimea $A^-(C)$ a extremităților inițiale ale arcelor cu extremitatea finală în C este

$$A^-(C) = \{1, 3, 7\},$$

și pentru fiecare din vârfurile acestei mulțimi calculăm

$$A^+(1) \cap C = \{2, 5\}, \quad A^+(3) \cap C = \{4\}, \quad A^+(7) \cap C = \{8\}.$$

Atunci

$$\begin{aligned} w(1, C) &= \min\{w(1, v_j) + 200 - w(v_{j-1}, v_j), \quad v_j = 2, 5\} \\ &= \min\{60 + 200 - 20, 70 + 200 - 40\} = 230, \\ w(3, C) &= w(3, 4) + 200 - w(8, 4) = 230, \\ w(7, C) &= w(7, 8) + 200 - w(6, 8) = 210. \end{aligned}$$

Deci, $i_0 = 7, v_{j_0} = 8$ și arcul $(6, 8)$ va fi înlocuit cu arcul $(7, 8)$. În acest moment

$$M = \{(r, 1), (4, 2), (2, 3), (8, 4), (2, 5), (5, 6), (1, 7), (7, 8)\},$$

și formează un arbore de acoperire orientat, care are ponderea minimă egală cu 210, deci algoritmul se oprește.

3.5.4 Lanț minim. Algoritmul lui Dijkstra

O altă problemă de optimizare este problema lanțului de lungime minimă. Dată fiind, de exemplu, o rețea de căi ferate care conectează anumite orașe, se cere să se determine cel mai scurt drum dintre două orașe date din această rețea. Formularea problemei este următoarea:

Problema lanțului minim (PLM): Fie s, t vârfuri ale unui graf conex cu ponderi G . Să se găsească un lanț între s și t a cărui pondere să fie minimă.

Observația 3.14 Dacă ponderile sunt toate egale cu 1, atunci problema se reduce la găsirea unui lanț între s și t care să conțină un număr minim de muchii. Această versiune a problemei lanțului minim a fost deja rezolvată cu algoritmul BFS.

Problema (PLM) a fost rezolvată în 1959 de către Dijkstra.¹⁷ De fapt algoritmul său realizează mai mult decât cere problema: găsește cel mai scurt lanț de la un vârf dat s la *fiecare* din celelalte vârfuri ale grafului.

Definiția 3.37 Fie $G = (V, E)$ un graf cu ponderi. **Distanța** dintre două vârfuri $x, y \in V$, notată cu $\text{dist}(x, y)$, este lungimea celui mai scurt lanț dintre cele două vârfuri.

Ideea algoritmului este să construim un arbore Dijkstra, pornind de la un vârf dat s , adăugând, la fiecare iterație, o muchie frontieră a cărei extremitate care nu e în arbore să fie cât mai aproape de s . Pentru simplitate, odată ce vârful s este fixat, pentru un vârf arbitrar x vom nota $\text{dist}(s, x)$ cu $\text{dist}(x)$. La fiecare iterație se adaugă o muchie (și implicit un vârf) la arborele Dijkstra, și atunci se scrie o etichetă pe noul vârf. Astfel, vârfurile din arborele Dijkstra la o iterație dată sunt acelea pentru care cel mai scurt lanț de la s a fost deja găsit.

Fie P -valoarea unei muchii frontieră e

$$P(e) = \text{dist}(x) + w(e),$$

unde x este extremitatea etichetată a lui e . Muchiei cu cea mai mică P -valoare îi este dată cea mai mare prioritate. Mai mult, P -valoarea muchiei cu cea mai mare prioritate va da distanța de la s la extremitatea neetichetată a lui e . Algoritmul etichetează fiecare vârf al arborelui cu distanța dintre vârful s și acel vârf.

Algoritm 10 (lanț minim Dijkstra)

Intrare: un graf conex $G = (V, E)$ ponderi nenegative și $s \in V$

Ieșire: – un arbore de acoperire T al lui G , cu rădăcina s , în care drumul (unic) de la s la fiecare vârf v este cel mai scurt drum de la s la v în G

– o etichetare a vârfurilor, care dă distanța de la s la fiecare vârf

- se inițializează arborele Dijkstra T cu vârful s
- se inițializează mulțimea de muchii frontieră pentru T cu \emptyset
- $\text{dist}(s) = 0$
- se scrie eticheta 0 pe vârful s
- cât timp arborele T nu acoperă încă graful G
 - pentru fiecare e muchie frontieră pentru T
 - fie x extremitatea etichetată a muchiei e
 - $P(e) = \text{dist}(x) + w(e)$
 - fie e^* o muchie frontieră pentru T cu cea mai mică P -valoare
 - fie y extremitatea neetichetată a muchiei e^*
 - se adaugă muchia e^* (și vârful y) la arborele T
 - $\text{dist}(y) = P(e^*)$
 - se scrie eticheta $\text{dist}(y)$ pe vârful y
- retur arbore Dijkstra T și o etichetare a vârfurilor acestuia

Teorema 3.31 (corectitudinea algoritmului Dijkstra) Fie G un graf conex cu ponderi și T arborele Dijkstra rezultat din aplicarea algoritmului 10, pornind din vârful s . Atunci, pentru orice vârf x al lui T , unicul lanț în T între s și x este cel mai scurt lanț în G între s și x .

¹⁷Edsger Wybe Dijkstra [ˈdɛɪkstra] (1930-2002), calculatorist olandez.

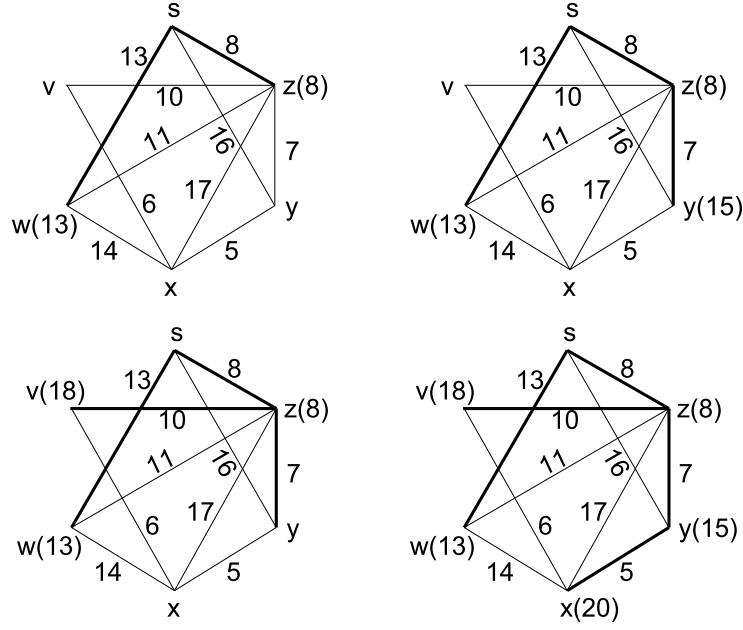


Figura 3.24: Arborii Dijkstra (linii groase) după pașii 2,3,4,5.

Demonstrație. Demonstrația urmează același raționament ca cel folosit în demonstrația teoremei 3.27. ■

Exemplul 3.11 Considerăm graful cu ponderi din figura 3.24. Dacă aplicăm algoritmul 10 pornind din vârful s , pașii sunt următorii:

Pasul 1 $dist(s) = 0$,

$$P(sz) = 8$$

$$P(sy) = 16 \Rightarrow \min = 8, \Rightarrow \text{adăugăm } sz, dist(z) = 8.$$

$$P(sw) = 13$$

Pasul 2

$$P(sw) = 13$$

$$P(sy) = 16$$

$$P(zy) = 15$$

$$P(zv) = 18$$

$$P(zw) = 19$$

$$P(zx) = 25$$

$$\Rightarrow \min = 13 \Rightarrow \text{adăugăm } sw, dist(w) = 13.$$

Pasul 3

$$P(sy) = 16$$

$$P(zy) = 15$$

$$P(zx) = 25$$

$$P(zv) = 18$$

$$P(wx) = 27$$

$$\Rightarrow \min = 15 \Rightarrow \text{adăugăm } zy, dist(y) = 15.$$

Pasul 4

$$P(yx) = 20$$

$$P(zx) = 25$$

$$P(zv) = 18$$

$$P(wx) = 27$$

$$\Rightarrow \min = 18 \Rightarrow \text{adăugăm } zv, dist(v) = 18.$$

Pasul 5

$$\begin{aligned}
P(wx) &= 27 \\
P(yx) &= 20 \\
P(vx) &= 24 \\
P(zx) &= 25
\end{aligned}
\Rightarrow \min = 20 \Rightarrow \text{adăugăm } yx, \text{ dist}(x) = 20.$$

Pentru grafuri orientate, există mai mulți algoritmi pentru găsirea unui arbore orientat al lanțurilor minime. Printre aceștia menționăm algoritmi construiți de Moore-Dijkstra, Bellman-Kalaba și Ford, împreună cu variațiile lor.

3.6 Algoritmi de tip greedy

O problemă de optimizare este o problemă un care vrem să găsim nu doar o soluție, ci *cea mai bună* soluție.

Un algoritm de tip greedy (AG) este un algoritm care «se repede» imediat la ce este «mai bun», fără să-i pese de consecințe. Acesta repetă o singură procedură, până când nu mai rămâne nimic de făcut.

Matematic, spunem că alege soluția cea mai bună *imediat* (sau **local**), fără să ia în considerare efectele acestei decizii în viitor, dar sperând că alegând un *optim local* la fiecare pas, va putea sfârși prin a găsi un **optim global**.

Exemplul 3.12 *Presupunem că vrem să dăm rest o sumă de €6.39 folosind cât mai puține bancnote și monede. Un algoritm greedy ar proceda astfel: la fiecare pas, luăm cea mai mare bancnotă sau monedă posibilă. Piese existente sunt*

$$5\text{€}, 2\text{€}, 1\text{€}, 50\text{c}, 20\text{c}, 10\text{c}, 5\text{c}, 2\text{c}, 1\text{c},$$

deci putem da restul astfel:

$$6.39 = 5\text{€} + 1\text{€} + 20\text{c} + 10\text{c} + 5\text{c} + 2 \cdot 2\text{c}. \quad (7 \text{ piese})$$

Pentru euro, algoritmul de tip greedy dă întotdeauna soluția optimă.

Dar, presupunem că avem un sistem monetar imaginar, cu monedele 10u, 7u, 1u. Cu un AG, suma de 15u se plătește

$$15u = 1 \cdot 10u + 5 \cdot 1u \quad (6 \text{ piese}).$$

Există totuși un mod de a plăti cu doar 3 monede, anume

$$15u = 2 \cdot 7u + 1 \cdot 1u.$$

Deci, algoritmul greedy dă o soluție, dar nu o soluție optimă.

Alte exemple de algoritmi greedy sunt algoritmi lui Prim, Kruskal și Dijkstra, dar așa cum am putut vedea, ei ne conduc întotdeauna la soluția optimă. Deci, pentru anumite probleme de optimizare, un algoritm de tip greedy poate totuși să găsească soluția optimă.

Prin urmare, întrebările care se pun în mod natural sunt: Când un algoritm greedy găsește soluția optimă? Dacă știm că este posibil ca să nu găsească o soluție optimă, de ce se mai folosește? Vom da un răspuns parțial la prima întrebare în paragraful următor. Privitor la a doua întrebare, motivul pentru care utilizăm algoritmi greedy este că de obicei acești algoritmi sunt rapizi, ușor de implementat, și deseori ne conduc la o aproximare relativ bună a soluției optime.

3.6.1 Algoritm greedy pentru problema ponderii maxime

Fie E o mulțime finită și \mathcal{I} o familie de submulțimi ale lui E , numite **mulțimi admisibile**. Atunci perechea $\mathcal{S} = (E, \mathcal{I})$ se numește **sistem admisibil de (sub)mulțimi**. Dacă $w : E \rightarrow \mathbb{R}_+$ este o funcție pondere, atunci pentru submulțimea $N \in \mathcal{I}$ definim ponderea sa ca și

$$w(N) = \sum_{x \in N} w(x).$$

Problema P_{\max} a găsirii ponderii maxime este următoarea: Să se găsească o mulțime admisibilă cu pondere maximă.

Un AG pentru rezolvarea acestei probleme este următorul:

Algoritm 11 (Algoritm greedy pentru P_{\max})

Intrare: $\mathcal{S} = (E, \mathcal{I})$ un sistem de submulțimi admisibile

Ieșire: o soluție a lui P_{\max} (cel puțin sperăm să obținem maximalitatea)

- se inițializează $M = \emptyset$ și $A = E$
 - cât timp $A \neq \emptyset$
 - alegem $e \in A$ cu ponderea maximă
 - $A = A - e$
 - dacă $M + e \in \mathcal{I}$, atunci $M = M + e$
- retur mulțimea M

Deci, fiecare element $e \in E$ a fost considerat o singură dată, fiind fie adăugat, fie eliminat definitiv. Astfel, algoritmul este rapid și simplu, dar nu e sigur că ne conduce la soluția optimă.

Exemplul 3.13 Fie $E = \{e_1, e_2, e_3\}$, cu ponderile

$$w(e_1) = 3, \quad w(e_2) = w(e_3) = 2,$$

și $\mathcal{I} = \{\{e_1\}, \{e_2\}, \{e_2, e_3\}\}$.

Soluția problemei P_{\max} este $M = \{e_2, e_3\}$, având $w(M) = 4$, în timp ce soluția obținută cu algoritmul 11 este $M = \{e_1\}$.

Motivul pentru care AG nu a găsit soluția optimă este cauzat de faptul că nu există nicio legătură între mulțimile din \mathcal{I} .

Definiția 3.38 Un sistem de mulțimi $\mathcal{S} = (E, \mathcal{I})$ se numește **ereditar** dacă este închis în raport cu incluziunea, adică,

$$\text{dacă } A \in \mathcal{I} \text{ și } A' \subseteq A, \text{ atunci } A' \in \mathcal{I}.$$

Exemplul 3.14 Fie $E = \mathbb{R}^3$ și \mathcal{I} mulțimea tuturor mulțimilor de vectori liniar independenți. Atunci $\mathcal{S} = (\mathbb{R}^3, \mathcal{I})$ este un sistem ereditar de submulțimi, deoarece orice submulțime a unei mulțimi de vectori liniar independenți este liniar independentă.

Exemplul 3.15 Fie $G = (V_G, E_G)$ un graf și \mathcal{I} mulțimea de submulțimi ale lui E_G ale căror subgrafuri induse sunt subgrafuri aciclice ale lui G . Atunci $\mathcal{S} = (E_G, \mathcal{I})$ este un sistem ereditar de mulțimi.

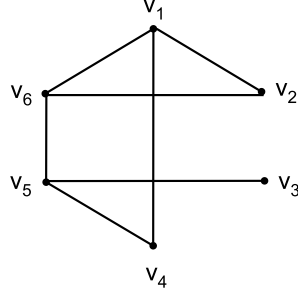


Figura 3.25: Graful pentru problema planificării.

În continuare vom considera problema P_{\max} pentru sisteme ereditare de mulțimi.

Propoziția 3.32 Dacă sistemul $\mathcal{S} = (E, \mathcal{I})$ are proprietatea că pentru orice $w : E \rightarrow \mathbb{R}_+$ algoritmul de tip greedy 11 ne conduce la o soluție a lui P_{\max} , atunci (E, \mathcal{I}) este un sistem ereditar de mulțimi.

Propoziția 3.33 Aplicând AG unui sistem ereditar de mulțimi (E, \mathcal{I}) și unei funcții pondere $w : E \rightarrow \mathbb{R}$, obținem o mulțime **maximală**, în sensul că

$$\text{nu există } x \in E \text{ astfel încât } M \cup \{x\} \in \mathcal{I}.$$

Definiția 3.39 Un sistem ereditar de mulțimi $\mathcal{M} = (E, \mathcal{I})$ se numește **matroid** dacă satisface următoarea condiție, pe care o vom numi **proprietatea de creștere**:

$$(AP) : \quad \text{Dacă } I, J \in \mathcal{I} \text{ și } |I| < |J|, \text{ atunci există } e \in J \setminus I \\ \text{astfel încât } I \cup \{e\} \in \mathcal{I}.$$

Propoziția 3.34 Dacă un sistem ereditar de mulțimi (E, \mathcal{I}) are proprietatea că, pentru orice pondere $w : E \rightarrow \mathbb{R}_+$, AG conduce la o soluție a problemei P_{\max} , atunci acest sistem are de asemenea proprietatea de creștere.

Teorema 3.35 Un sistem de mulțimi (E, \mathcal{I}) are proprietatea că, pentru orice $w : E \rightarrow \mathbb{R}_+$ un algoritm de tip greedy conduce la o soluție a problemei P_{\max} dacă și numai dacă (E, \mathcal{I}) este un matroid.

Exemplul 3.16 (matroid) Fie $G = (V, E)$ un graf și fie \mathcal{I} mulțimea de muchii cu proprietatea că graful generat de aceste muchii nu conține cicluri. Atunci $\mathcal{S} = (E, \mathcal{I})$ este un matroid.

Astfel, algoritmul lui Kruskal (pag. 104) ne conduce la un arbore economic.

3.6.2 Algoritm greedy pentru colorarea vârfurilor

Presupunem că vrem să planificăm șase lecții de câte o oră $V = \{v_1, v_2, \dots, v_6\}$. Printre participanți, există persoane care vor să participe la v_1 și v_2 , v_1 și v_4 , v_3 și v_5 , v_2 și v_6 , v_4 și v_5 , v_5 și v_6 , v_1 și v_6 . Câte ore sunt necesare pentru a programa cele 6 lecții fără suprapuneri? Situația poate fi reprezentată printr-un graf (figura 3.25), ale cărui vârfuri sunt cele șase lecții și în care muchiile

reprezintă posibilele suprapuneri. Un orar posibil care să evite suprapunerile este:

Ora 1	Ora 2	Ora 3	Ora 4
v_1 și v_3	v_2 și v_4	v_5	v_6

Din punct de vedere matematic, construim o partiție a mulțimii V în patru părți, astfel încât nici o parte să nu conțină perechi de vârfuri adiacente ale grafului. De fapt, definim o funcție

$$c : \{v_1, v_2, v_3, v_4, v_5, v_6\} \rightarrow \{1, 2, 3, 4\},$$

care asociază fiecărui vârf (lecție) ora programată pentru ea. De obicei vorbim despre culori asociate vârfurilor și ceea ce este important este că vârfurile care sunt adiacente în graf trebuie să aibă culori diferite.

Definiția 3.40 *O colorare a vârfurilor unui graf $G = (V, E)$ este o funcție $c : V \rightarrow \mathbb{N}$ cu proprietatea*

$$c(x) \neq c(y) \quad \text{dacă } xy \in E.$$

Numărul cromatic al lui G , notat $\chi(G)$, este definit ca și cel mai mic număr natural k pentru care există o colorare a vârfurilor lui G folosind k culori.

Revenind la figura 3.25, orarul nostru este echivalent cu o colorare a vârfurilor folosind patru culori. Dar numărul cromatic al grafului este trei, întrucât există colorarea

Culoare 1	Culoare 2	Culoare 3
v_1	v_2 și v_5	v_3, v_4 și v_6

Mai mult, nu există nici o colorare cu doar două culori deoarece v_1, v_2, v_6 sunt mutual adiacente, deci ele trebuie să aibă culori diferite. În concluzie, numărul cromatic al grafului este 3.

În general găsirea lui $\chi(G)$ este o problemă dificilă și nu există nici un algoritm rapid (executabil în timp polinomial) care să realizeze o asemenea colorare. O metodă pentru colorarea vârfurilor cu un număr rezonabil de culori este de a asocia culori vârfurilor în ordine, în așa fel încât fiecare vârf să primească prima culoare care nu a fost asociată niciunuia din vecinii săi. Astfel, se face cea mai bună alegere la fiecare pas, fără a privi înainte pentru a vedea dacă acea alegere nu va crea probleme mai târziu. Cu alte cuvinte se aplică un algoritm de tip greedy.

Algoritm 12 (Algoritm greedy pentru colorarea vârfurilor)

Intrare: un graf $G = (V, E)$ cu $|V| = n$

Ieșire: o colorare a vârfurilor lui G

- asociază culoarea 1 vârfului v_1
- pentru $i = 2, \dots, n$
 - fie S o mulțime vidă de culori
 - pentru $j = 1, \dots, i - 1$
 - dacă v_j și v_i sunt adiacente, atunci se adaugă în S culoarea lui v_j
 - $k = 1$
 - cât timp culoarea k este în S , execută $k = k + 1$
 - se atribuie culoarea k vârfului v_i
- retur o culoare pentru fiecare vârf din V

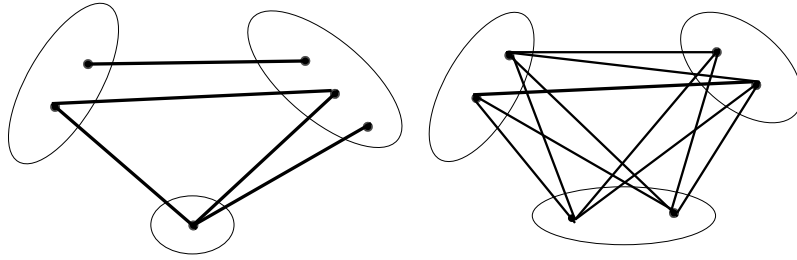


Figura 3.26: A 3-partite graf (left) și a 3-partite complete graf (right).

În final vom da un rezultat privind numărul cromatic al unei clase particulare de grafuri.

Definiția 3.41 *Un graf se numește **planar** dacă el poate fi desenat în plan fără ca muchiile sale să se intersecteze.*

Teorema 3.36 (Teorema celor patru culori) *Orice hartă plană poate fi colorată folosind patru culori, astfel încât acele regiuni care au în comun o frontieră diferită de un punct să nu aibă aceeași culoare.*

Acest rezultat a fost pentru prima oară formulat ca și conjectură¹⁸ în 1853 de către Guthrie. La acea vreme nu a fost dificil să se arate că sunt suficiente cinci culori. Teorema celor patru culori este prima teoremă importantă care a fost demonstrată cu ajutorul calculatorului. Deoarece a fost formulată ca și conjectură, au existat multe încercări de a o demonstra, fiecare demonstrație dovedindu-se după ani incorectă. În final, ea a fost demonstrată în 1976 de către Appel și Haken, folosind un algoritm al lui Koch. Demonstrația lor a redus infinitatea de hărți posibile la 1936 configurații (mai târziu reduse la 1476), verificate una câte una de către calculator. Mai târziu, în 1996, s-a demonstrat că există de fapt doar 633 configurații posibile.

3.7 Grafuri bipartite

Definiția 3.42 *Fie $r \in \mathbb{N}$, $r \geq 2$. Un graf $G = (V, E)$ se numește **r -partit** dacă V admite o partiție în r clase astfel încât fiecare muchie are extremitățile în clase diferite (sau, echivalent, vârfurile din aceeași clasă nu trebuie să fie adiacente).*

Observația 3.15 *Un graf 2-partit este numit **bipartit** și se notează $G = (V_1 \cup V_2, E)$, unde V_1 și V_2 sunt cele două clase ale partiției. O proprietate imediată a unui graf bipartit este*

$$\sum_{v \in V_1} \deg(v) = \sum_{v \in V_2} \deg(v).$$

În continuare vom da o caracterizare importantă a grafurilor bipartite.

¹⁸În matematică, o conjectură este o afirmație care pare a fi adevărată, dar acest lucru nu a fost încă demonstrat. De îndată ce o conjectură a fost demonstrată, este ridicată la rangul de teoremă și poate fi folosită ulterior fără riscuri în alte demonstrații matematice. Până în acel moment, matematicienii pot folosi conjectura, dar orice rezultat dedus din aceasta este unul conjectural până când conjectura respectivă este clarificată.

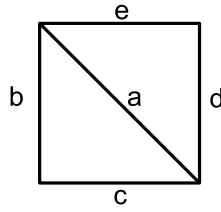


Figura 3.27: Graful din exemplul 3.17.

Teorema 3.37 (de caracterizare a grafurilor bipartite) *Un graf este bipartit dacă și numai dacă nu conține cicluri de lungime impară.*

Demonstrație.

- \Rightarrow Fie $G = (V_1 \cup V_2, E)$ graf bipartit. Atunci sau G nu conține cicluri, sau, dacă există un ciclu, el are vârfurile succesiv în V_1 și V_2 , cu ultimul vârf în aceeași clasă cu primul. Un asemenea ciclu are un număr par de muchii.
- \Leftarrow Fie $G = (V, E)$ un graf fără cicluri impare (G are prin urmare niciun ciclu sau cicluri pare). Fie $G_0 = (V_0, E_0)$ o componentă conexă a lui G și notăm T_0 arborele de acoperire obținut prin algoritmul BFS, pornind dintr-un vârf oarecare $s \in V_0$. Considerăm mulțimile

$$\begin{aligned} V_1 &= \{s\} \cup \{x \in V, d(x, s) = \text{par}\}, \\ V_2 &= \{x \in V, d(x, s) = \text{impar}\}. \end{aligned}$$

Atunci orice muchie $e_0 \in T_0$ are o extremitate în V_1 și cealaltă în V_2 . Fie $e = xy$ o muchie care nu aparține arborelui de acoperire. Atunci $d(x, s)$ și $d(y, s)$ sunt fie egale, fie diferă prin 1 (vezi propoziția 3.25). Dacă ar fi egale, e ar închide un ciclu de lungime impară, fapt care ar contrazice ipoteza. În consecință

$$|d(x, s) - d(y, s)| = 1,$$

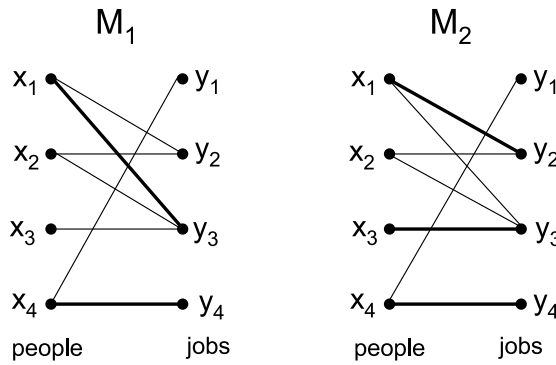
care implică faptul că unul din vârfurile x, y este în V_1 și celălalt este în V_2 . ■

O aplicație a grafurilor bipartite este *atribuirea de locuri de muncă (job assignment)*. Dacă X este o mulțime de persoane și Y o mulțime de locuri de muncă, astfel încât fiecare persoană din X este calificată pentru anumite locuri de muncă din Y , atunci întrebarea care se pune în mod firesc este: cum se pot atribui persoanelor locuri de muncă, astfel încât un număr maxim de persoane să primească locuri de muncă pentru care sunt calificate?

În limbajul grafurilor, considerăm graful bipartit $G = (X \cup Y, E)$, unde $xy \in E$ dacă și numai dacă x este calificat pentru locul de muncă y . O asemenea atribuire se va numi **cuplaj** în G .

3.7.1 Cuplaje

Definiția 3.43 Fie $G = (V, E)$ un graf.

Figura 3.28: Cuplajele M_1 și M_2 .

O mulțime $M \subseteq E$ se numește **cuplaj** în G dacă, oricum am alege două muchii din M , ele nu sunt adiacente în G .

Spunem că un cuplaj M **saturează** vârful $v \in V$, și că v este saturat de M , dacă există o muchie din M care este incidentă în v ; în caz contrar spunem că v este **nesaturat** de M .

Un cuplaj **perfect** este un cuplaj M în care orice vârf al lui G este saturat de M .

Dacă $e = xy \in M$, spunem că M **cuplează** pe x cu y (sau x este cuplat cu y de către M).

Un cuplaj **maxim** este un cuplaj cu număr maxim de muchii (nu există niciun alt cuplaj care să aibă mai multe muchii).

Un cuplaj **maximal** este un cuplaj care nu este o submulțime proprie a unui alt cuplaj din G .

Exemplul 3.17 În graful din figura 3.27,

$\{b\}, \{c\}, \{d\}, \{e\}$ sunt cuplaje care nu sunt maximale.

$\{b, d\}, \{c, e\}$ sunt cuplaje maxime,

$\{a\}$ este cuplaj maximal, dar nu cuplaj maxim.

Cel mai des întâlnite aplicații ale cuplajelor sunt programarea utilajelor și atribuirea locurilor de muncă.

3.7.2 Cuplaje în grafuri bipartite

Considerăm graful bipartit din figura 3.28. Mulțimea $M_1 = \{x_1y_3, x_4y_4\}$ nu este un cuplaj maxim, dar $M_2 = \{x_1y_2, x_3y_3, x_4y_4\}$ este un cuplaj maxim, deoarece nu există un alt cuplaj cu mai mult de trei muchii. Într-adevăr, dacă se consideră că X este o mulțime de persoane și Y este o mulțime de locuri de muncă, atunci este imposibil să găsim pentru toate cele patru persoane locuri de muncă pentru care acestea să fie calificate. Din cauză că persoanele $\{x_1, x_2, x_3\}$ sunt colectiv calificate doar pentru două locuri de muncă y_2 și y_3 , una din ele nu poate primi loc de muncă, cu toate că există pretendenți pentru toate locurile de muncă.

În multe aplicații se dorește, dacă e posibil, găsirea unui cuplaj al lui G care să satureze toate vârfurile din X .

Definiția 3.44 Dacă $|M| = |X|$ (toate persoanele primesc locuri de muncă), spunem că M este un **cuplaj complet**.

Desigur, M_1 și M_2 nu sunt cuplaje complete. Primul pas în studierea cuplajelor este să decidem dacă este posibilă găsirea unui cuplaj complet. Am văzut în exemplul anterior că dacă există trei persoane și două locuri de muncă, nu poate exista un cuplaj complet. Apoi, suntem interesați de găsirea de condiții necesare și suficiente pentru existența unui astfel de cuplaj.

Fie $G = (X \cup Y, E)$ un graf bipartit și $S \subseteq X$. Definim mulțimea tuturor vârfurilor adiacente vârfurilor din S ,

$$J(S) = \{y \in Y, \text{ pentru care există } x \in S \text{ a.î. } xy \in E\},$$

cu alte cuvinte mulțimea de locuri de muncă pentru care persoanele din S sunt colectiv calificate.

Observația 3.16 Dacă $|J(S)| < |S|$, atunci cineva din S nu va primi loc de muncă. Deci, dacă J este un cuplaj complet, atunci

$$|J(S)| \geq |S|, \text{ pentru orice } S \subseteq X. \quad (3.10)$$

Condiția dată în (3.10) este cunoscută sub numele de condiția lui Hall¹⁹.

Teorema 3.38 (Hall, 1935) In graful bipartit $G = (X \cup Y, E)$ există un cuplaj complet dacă și numai dacă este satisfăcută condiția lui Hall (3.10).

Demonstrație.

\Rightarrow Fie M un cuplaj complet și fie $S \subseteq X$. Vârfurile din Y cuplate de M cu cele din S formează o submulțime a lui $J(S)$ de măsură $|S|$. Astfel, $|J(S)| \geq |S|$.

\Leftarrow Presupunem că $|J(S)| \geq |S|$ pentru orice $S \subseteq X$. Dat fiind un cuplaj M pentru care $|M| < |X|$, vom arăta cum se poate construi un cuplaj M' cu $|M'| = |M| + 1$ (un astfel de cuplaj M' va exista întotdeauna!).

Fie x_0 orice vârf necuplat de către M . Atunci

$$|J(\{x_0\})| \geq |\{x_0\}| = 1,$$

deci există cel puțin o muchie $e = x_0 y_1$ în M .

Dacă y_1 este necuplat, atunci $M' = M \cup \{e\}$.

Dacă y_1 este cuplat de către M , să spunem cu x_1 , atunci

$$|J(\{x_0, x_1\})| \geq |\{x_0, x_1\}| = 2,$$

deci există y_2 , $y_2 \neq y_1$, adiacent la x_0 sau x_1 .

Dacă y_2 este necuplat, stop.

Dacă y_2 este cuplat, să spunem cu x_2 , repetăm

raționamentul, deci există y_3 , diferit de y_1 și y_2 ,

adiacent la cel puțin unul din vârfurile x_0, x_1, x_2 .

...

Continuând în acest fel, trebuie în cele din urmă să ne oprim la un vârf necuplat y_r (deoarece G este finit).

¹⁹Philip Hall (1904-1982), matematician englez.

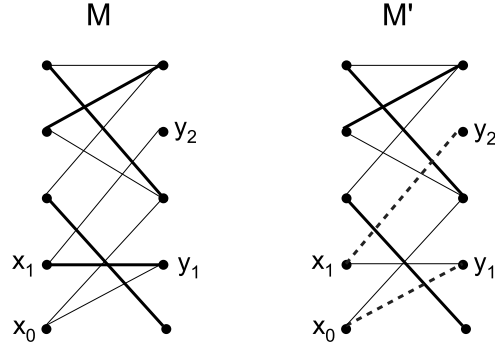


Figura 3.29: Cuplajele M și M' din demonstrația teoremei lui Hall. Muchia x_1y_1 din M este înlăturată și muchiile x_0y_1 , x_1y_2 care nu sunt în M sunt adăugate.

Fiecare y_i , $i \in \mathbb{N}_r$, este adiacent cel puțin unuia din vârfurile x_0, x_1, \dots, x_{i-1} , prin urmare avem un lanț

$$p = \langle y_r, x_{r-1}, y_{r-1}, \dots, x_1, y_1, x_0 \rangle, \quad \text{în care } x_iy_i \in M.$$

Construim cuplajul M' astfel încât muchia x_iy_i a lanțului p să nu aparțină lui M' , dar muchia alternantă să aparțină lui M' (figura 3.29).

Intrucât muchiile terminale y_rx_{r-1} și y_1x_0 sunt ambele în M' , avem $|M'| = |M| + 1$ (numărul muchiilor adăugate este egal cu unu plus numărul muchiilor înlăturate).

■

Pe scurt, ideea demonstrației este construcția unui lanț ale cărui muchii sunt alternativ în M și în $E \setminus M$. În general, pentru un graf $G = (V, E)$ și pentru un cuplaj M din G , spunem că un lanț este **lanț alternant** pentru M dacă muchiile sale sunt alternativ în $E \setminus M$ și M . Un **lanț crescător** pentru M este un lanț alternant pentru M cu primul și ultimul vârf nesaturate de M . De exemplu, un lanț crescător pentru cuplajul M din figura 3.29 este $\langle x_0, y_1, x_1, y_2 \rangle$.

Să observăm că, întrucât prima și ultima muchie a unui lanț crescător nu aparțin lui M , lanțul are cu o muchie mai puțin în M decât în $E \setminus M$.

Din demonstrația teoremei putem de asemenea deduce că, în cazul în care condiția lui Hall este satisfăcută și M nu este complet, atunci există un lanț crescător pentru M .

Ideea demonstrației constituie de asemenea un instrument practic pentru construcția cuplajelor complete, furnizând de fapt un algoritm pentru această construcție.

Observația 3.17 *Teorema lui Hall este cunoscută și sub numele de **teorema mariajului**, datorită următoarei interpretări: X este o mulțime de bărbați și Y este o mulțime de femei, unii dintre ei cunoscându-se reciproc. Se cere condiția ca fiecare bărbat să se însoare cu una din femeile pe care o cunoaște. În teorema lui Hall, $J(S)$ reprezintă mulțimea de femei cunoscute de bărbații din S . Un cuplaj este o mulțime de perechi care se căsătoresc.*

3.7.3 Cuplaj maxim

În general, un graf bipartit nu va avea un cuplaj complet, deci problemele care se ridică în acest caz sunt: cum se poate găsi măsura maximă a unui cuplaj și cum se poate face o atribuire din care să rezulte că cel mai mare număr posibil de persoane primesc locuri de muncă. Soluția se poate deduce teorema lui Hall.

Mai întâi să observăm că, dacă $|S| > |J(S)|$, atunci anumite persoane nu vor primi locuri de muncă, de fapt există cel puțin $|S| - |J(S)|$ persoane care nu vor primi locuri de muncă. Această observație ne sugerează următoarea definiție.

Definiția 3.45 *Deficiența d a unui graf bipartit $G = (X \cup Y, E)$ este definită ca și*

$$d = \max\{|S| - |J(S)|, \quad S \subseteq X\}.$$

Observația 3.18 *Pentru $S = \emptyset$ avem $|S| = |J(S)| = 0$, deci $d \geq 0$. Atunci, folosind deficiența, teorema lui Hall poate fi reformulată astfel: Există un cuplaj complet dacă și numai dacă $d = 0$.*

În următoarea teoremă vom da un rezultat privind măsura unui cuplaj maxim în cazul general.

Teorema 3.39 *Măsura unui cuplaj maxim M într-un graf bipartit $G = (X \cup Y, E)$ este*

$$|M| = |X| - d,$$

unde d este deficiența grafului G .

Demonstrație. Din definiția 3.45 deducem că există $S_0 \subseteq X$ astfel încât

$$|S_0| - |J(S_0)| = d.$$

În orice cuplaj, cel puțin d elemente din S_0 rămân necuplate, deci

$$|M| \leq |X| - d.$$

Rămâne să arătăm că există un cuplaj cu măsura egală cu $|X| - d$.

Fie D o nouă mulțime, cu măsura $|D| = d$. Construim graful $G^* = (X^* \cup Y^*, E^*)$, cu

$$\begin{aligned} X^* &= X, \\ Y^* &= Y \cup D, \\ E^* &= E \cup K, \end{aligned}$$

unde K este mulțimea tuturor muchiilor posibile care leagă X și D . Atunci, pentru graful G^* avem $J^*(S) = D \cup J(S)$, de unde

$$|J^*(S)| - |S| = d + |J(S)| - |S| \geq 0.$$

Astfel, G^* satisface condiția lui Hall și prin urmare G^* are un cuplaj complet M^* .

Eliminând din M^* acele d muchii care au un vârf în D , obținem cuplajul cerut în G . ■

Observația 3.19 *Teorema 3.39 nu este prea eficientă pentru găsirea măsurii unui cuplaj maxim, din cauză că pentru a calcula d trebuie examinate mai întâi toate cele $2^{|X|}$ submulțimi ale lui X .*

O abordare mai eficientă se bazează pe faptul că, dacă avem un lanț alternant pentru un cuplaj M , atunci putem construi un cuplaj mai bun M' . Pentru a putea folosi această idee, avem nevoie de următorul rezultat, cunoscut sub numele de teorema lui Berge.

Teorema 3.40 (Berge,²⁰ 1957) *Cuplajul M dintr-un graf G este un cuplaj maxim dacă și numai dacă G nu conține niciun lanț crescător pentru M .*

Demonstrație.

\Rightarrow Fie M este un cuplaj maxim. Vom demonstra că G nu conține niciun lanț crescător pentru M . Presupunem contrariul, anume că G conține un lanț crescător în M , notat $P = \langle e_1, e_2, \dots, e_{2s+1} \rangle$. Intrucât $e_1 \notin M$, deducem că

$$e_1, e_3, \dots, e_{2s+1} \notin M \quad \text{și} \quad e_2, e_4, \dots, e_{2s} \in M.$$

Definim mulțimea $M_1 \subseteq E$ obținută din M prin îndepărtarea muchiilor e_2, e_4, \dots, e_{2s} și adăugarea muchiilor $e_1, e_3, \dots, e_{2s+1}$. Atunci M_1 este un cuplaj în G care conține cu o muchie mai mult decât în M , fapt ce contrazice maximalitatea lui M .

\Leftarrow Arătăm că, dacă G nu conține niciun lanț alternant pentru M , atunci M este maxim. Presupunem că M nu este maxim și fie M^* un cuplaj maxim. Atunci $|M^*| > |M|$.

Fie F mulțimea de muchii din

$$M \Delta M^* = (M \cup M^*) \setminus (M \cap M^*) = (M \setminus M^*) \cup (M^* \setminus M).$$

Cu muchiile din F formăm un graf H , în care orice vârf va avea gradul unu sau doi, deoarece el poate fi incident cel mult unei muchii din M și unei muchii din M^* . Deci, componentele lui H sunt fie lanțuri elementare, fie cicluri elementare. În fiecare din aceste lanțuri/cicluri, muchiile din M alternează cu muchii care nu sunt în M . Astfel, în orice ciclu, numărul de muchii din M (din $M \setminus M^*$) este egal cu numărul de muchii care nu sunt în M (sunt în $M^* \setminus M$). Dar,

$$|M^*| > |M| \quad \Rightarrow \quad |M^* \setminus M| > |M \setminus M^*|,$$

deci este imposibil ca în F să existe numai cicluri. În concluzie, există cel puțin o componentă care este un lanț cu un număr impar de muchii, și acesta este un lanț crescător pentru M . Aceasta este o contradicție. ■

Din demonstrația teoremei lui Berge, putem deduce următorul algoritm pentru găsirea unui cuplaj maxim:

²⁰Claude Berge (1926-2002), matematician francez.

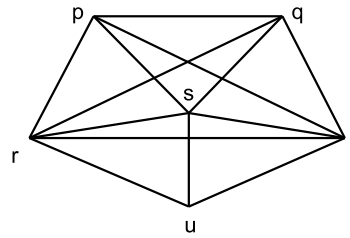


Figura 3.30: O hartă a orașelor și străzile care fac legătura între ele.

Algoritm 13 (pentru găsirea unui cuplaj maxim)

Intrare: *un graf G*

Ieșire: *un cuplaj maxim în G*

- se inițializează M cu orice muchie
- \otimes se caută un lanț crescător pentru M
Dacă se găsește un lanț crescător, se construiește un cuplaj mai bun M' în modul descris în demonstrația teoremei, și se revine la \otimes , cu M' înlocuind pe M
Dacă nu poate fi găsit niciun lanț crescător, STOP: M este cuplaj maxim.

Căutarea unui lanț crescător se poate face printr-o procedură BFS modificată. Se alege un vârf necuplat x_0 și se construiește un arbore de lanțuri alternante «parțiale», pornind din x_0 , astfel:

1. La nivelul 1, inserăm toate vârfurile y_1, y_2, \dots, y_k adiacente lui x_0 în G . Dacă oricare din aceste vârfuri y_i este necuplat, STOP: $\langle x_0, y_i \rangle$ este un lanț crescător.
2. Dacă toate vârfurile de la nivelul 1 sunt cuplate, se inserează la nivelul 2 vârfurile x_1, x_2, \dots, x_k cu care acestea sunt cuplate.
3. La nivelul 3, se inserează toate vârfurile noi adiacente vârfurilor de la nivelul 2. Dacă oricare din ele este necuplat, STOP: lanțul care ne conduce la acest vârf x_0 este un lanț crescător.
4. Dacă toate vârfurile de la nivelul 3 sunt cuplate, inserăm vârfurile cu care acestea sunt cuplate la nivelul 4, și așa mai departe.

Construcția se poate opri din cauză ca nu mai există vârfuri noi de inserat la un nivel impar. Când se întâmplă acest lucru, nu există niciun lanț crescător din vârful ales x_0 . Trebuie, totuși, să repetăm procedura pentru orice vârf necuplat din X înainte să fim siguri că nu putem găsi în G niciun lanț alternant G .

3.8 Grafuri hamiltoniene și euleriene

Pentru început să menționăm că toate rezultatele date în această secțiune pentru grafuri sunt de asemenea valabile și pentru multigrafuri, în afară de cazul în care specificăm contrariul.

Considerăm harta din figura 3.30, a șase orașe împreună cu străzile dintre ele. Un inginer constructor de autostrăzi (E) dorește să inspecteze toate

străzile, indiferent de direcție, fiind pregătit să înceapă și să termine în locuri diferite. Prietenul său, inspectorul (H), dorește să ia masa la restaurant în fiecare oraș. Fiecare din ei dorește să-și realizeze scopul în mod cât mai eficient cu putință. Deci, scopul lui (E) se poate formula astfel: «Doresc, dacă este posibil, să vizitez fiecare stradă o singură dată», în timp ce (H) spune «Doresc, dacă este posibil, să vizitez fiecare oraș o singură dată și să mă întorc în punctul de plecare. Intrebarea este: Se pot găsi trasee convenabile pentru fiecare dintre ei?

Pentru inspectorul (H), o posibilitate este lanțul închis

$$\omega = \langle p, q, t, s, u, r, p \rangle.$$

Pentru inginerul (E), fie x vârful de plecare și y vârful final, și presupunem pentru moment că $x \neq y$. El folosește o muchie incidentă în x atunci când pleacă și apoi, de fiecare dată când se întoarce în x trebuie să sosească și să plece folosind muchii diferite. În acest mod, el folosește un număr impar de muchii în x , și astfel gradul lui x ar trebui să fie un număr impar (x ar trebui să fie un vârf impar). Analog, gradul lui y ar trebui să fie și el număr impar. Toate vârfurile rămase ar trebui să fie pare, deoarece de fiecare dată când ajunge la un vârf intermediar trebuie să și plece din el, și prin urmare folosește două muchii diferite.

În concluzie, un traseu pentru (E), pornind și terminând în vârfuri distincte x și y este posibil dacă și numai dacă x și y sunt vârfuri impare și restul vârfurilor sunt pare. În cazul nostru, gradele sunt

$$\begin{array}{c|cccccc} v & p & q & r & s & t & u \\ \hline \delta(v) & 4 & 4 & 5 & 5 & 5 & 3 \end{array},$$

deci nu există niciun traseu pentru (E).

Dacă $x = y$, din nou nu există nicio soluție, întrucât toate vârfurile ar trebui să fie pare.

3.8.1 Grafuri hamiltoniene

În general, traseul lui (H) este un ciclu elementar care conține toate vârfurile unui graf dat. Asemenea cicluri au fost studiate pentru prima oară de către Hamilton,²¹ deci un ciclu cu această proprietate se numește **ciclu hamiltonian**. Un graf care este conex și conține un ciclu hamiltonian se numește **graf hamiltonian**. În cazul în care nu impunem restricția ca vârful inițial să coincidă cu vârful final, un asemenea lanț elementar care vizitează fiecare vârf exact o dată se numește **lanț hamiltonian**.

În general, nu este ușor să decidem dacă un graf este hamiltonian sau nu. Câteva rezultate care dau condiții suficiente pentru existența unui lanț hamiltonian sunt următoarele:

Teorema 3.41 (Dirac,²² 1952) *Un graf conex cu $n > 2$ vârfuri este hamiltonian dacă fiecare vârf al său are gradul $\geq n/2$. Această teoremă este valabilă doar pentru grafuri, nu și pentru multigrafuri.*

²¹William Rowan Hamilton (1805-1865), matematician, fizician și astronom irlandez.

²²Gabriel Andrew Dirac (1925-1984), matematician britanic.

Teorema 3.42 (Ore,²³ 1960) *Un graf conex cu $n > 2$ vârfuri este hamiltonian dacă suma gradelor oricăror vârfuri neadiacente este $\geq n$. O consecință imediată este că toate grafurile complete sunt hamiltoniene.*

Cea mai bună caracterizare a grafurilor hamiltoniene, care generalizează rezultatele precedente obținute de Ore și Dirac, a fost dată în 1972 de către Bondy²⁴ și Chvátal.²⁵ Înainte de a da teorema trebuie să dăm următoarea definiție.

Definiția 3.46 *Dat fiind un graf G cu n vârfuri, definim **închiderea** lui G ca fiind graful construit în mod unic din G prin adăugarea, pentru orice pereche de vârfuri neadiacente x, y cu $\delta(x) + \delta(y) \geq n$, muchia xy .*

Teorema 3.43 (Bondy-Chvátal, 1972) *Un graf este hamiltonian dacă și numai dacă închiderea sa este graf hamiltonian.*

O consecință a acestei teoreme este că toate poliedrele regulate (tetraedru, cub, octaedru, dodecaedru, icosaedru), considerate ca și grafuri, sunt hamiltoniene.

Printre algoritmi folosiți pentru construcția unui ciclu hamiltonian menționăm algoritmul lui Ham.

3.8.2 Grafuri euleriene

Problema găsirii unui traseu pentru (E) a fost rezolvată rapid: răspunsul a fost «nu». Această problemă a fost discutată pentru prima oară în 1736 de către Euler, în timp ce rezolva faimoasa problemă a celor *șapte poduri din Königsberg*.

Problema celor șapte poduri din Königsberg este o problemă renumită de matematică, inspirată dintr-un fapt real. Orașul prusian Königsberg (acum Kaliningrad, Rusia) era așezat pe râul Pregel, și includea două insule mari care erau conectate între ele și cu orașul principal prin șapte poduri (figura 3.31). Întrebarea pe care și-o puneau locuitorii era dacă este posibilă construirea unui traseu care să traverseze fiecare pod exact o dată. În 1736, Leonhard Euler a demonstrat că acest lucru nu este posibil. În demonstrația lui, Euler a formulat problema în limbajul teoriei grafurilor, abstractizând problema astfel: mai întâi a eliminat toate elementele cu excepția bucăților de pământ și a podurilor care le conectează, și apoi a înlocuit fiecare bucată de pământ cu un punct (vârf), și fiecare pod cu o linie (muchie), ca în figura 3.32. Astfel, el avea să introducă o nouă structură matematică – graful.

Dacă se calculează gradele vârfurilor din graful problemei celor șapte poduri (figura 3.32), se poate observa că cerința ca toate cele șapte vârfuri să aibă grad par nu este satisfăcută.

Definiția 3.47 *Un lanț simplu care parcurge fiecare muchie a grafului exact o dată se numește **lanț eulerian**.*

*Un lanț eulerian care este închis se numește **ciclu eulerian**.*

*Un graf se numește **eulerian** dacă el conține un ciclu eulerian.*

²³Øystein Ore (1899-1968), matematician norvegian.

²⁴John Adrian Bondy, matematician american.

²⁵Vašek Chvátal (n. 1946), matematician ceh.

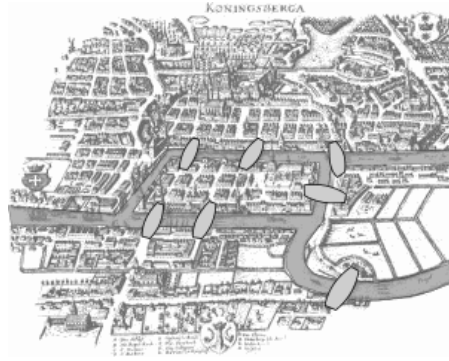


Figura 3.31: O hartă a orașului Königsberg și a renumitelor sale poduri.

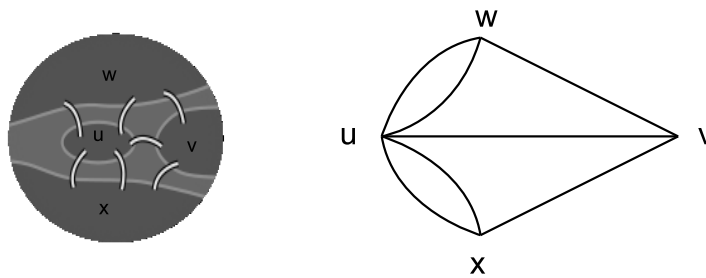


Figura 3.32: Graful Königsberg.

Așa cum am menționat deja, Euler a observat că o condiție necesară pentru existența ciclurilor euleriene este ca toate vârfurile grafului să aibă grad par; aceasta înseamnă că graful Königsberg nu este eulerian. În ceea ce privește existența unui lanț eulerian, fie toate vârfurile, fie toate cu excepția a două dintre ele ar trebui să aibă gradul par. Întrebarea care se pune în mod natural este dacă aceste condiții neceasare sunt și suficiente.

Carl Hierholzer a publicat în 1873 prima caracterizare completă a grafurilor euleriene, demonstrând că de fapt grafurile euleriene sunt exact grafurile care sunt conexe și în care orice vârf are gradul par.

Teorema 3.44 (Hierholzer, 1873) *Fie $G = (V, E)$ un graf conex. Atunci următoarele afirmații sunt echivalente:*

- a) G este eulerian,
- b) Orice vârf al lui G are grad par,
- c) G este reuniune de cicluri care nu au muchii comune.

Demonstrație.

a) \Rightarrow b) Această implicație a fost deja discutată.

b) \Rightarrow c) Deoarece fiecare vârf are grad par, G nu poate fi arbore, prin urmare G conține un ciclu, notat C_1 .

Dacă $C_1 = G$, atunci concluzia este demonstrată.

Dacă nu, considerăm graful $G_1 = G - C_1$. Deoarece muchiile înlăturate din G formează un ciclu, gradul în G_1 al fiecărui vârf din C_1 este redus cu 2, și prin urmare fiecare vârf al lui G_1 este par. Astfel, G_1 nu este arbore, prin urmare conține un ciclu C_2 .

Dacă $G = C_1 \cup C_2$, concluzia este demonstrată.

Dacă nu, continuăm în același mod și ne vom opri în momentul în care vom obține faptul că $G = C_1 \cup C_2 \cup \dots \cup C_n$.

$c) \Rightarrow a)$ Fie T^* un lanț simplu închis în G de lungime maximă. Conform propoziției 3.6, T^* este reuniune de cicluri fără muchii în comun.

Dacă T^* include toate muchiile din G , atunci concluzia este demonstrată.

Dacă nu, atunci există un ciclu, notat C , care nu face parte din T^* . Deoarece G este conex, există un vârf $v \in C^* \cap C$ și un lanț elementar închis obținut în felul următor:

- traversând T^* până când v este întâlnit pentru prima oară
- «întorcându-ne» în jurul lui C înapoi în v
- traversând apoi porțiunea rămasă din T^*

Construind acest lanț simplu, obținem o contradicție cu maximalitatea lui T^* . În concluzie, existența lui C care nu face parte din T^* este falsă, și prin urmare T^* include toate muchiile din G . ■

Construcția lanțurilor și ciclurilor euleriene

Fiind dat un graf conex cu cel mult două vârfuri de grad impar, putem construi un lanț eulerian sau un ciclu eulerian în acest graf cu ajutorul algoritmului lui Fleury, care datează din 1883. Începem cu un vârf de grad impar (dacă graful nu are nici unul, atunci începem cu orice vârf). La fiecare pas alegem o muchie, care să nu fie muchie tăietură decât în cazul în care nu avem altă posibilitate. Muchia aleasă este apoi ștearsă. La sfârșitul algoritmului nu mai există muchii rămase, iar succesiunea de muchii alese formează un ciclu eulerian dacă graful nu are vârfuri de grad impar, sau un lanț eulerian dacă graful are două vârfuri de grad impar.

Algoritm 14 (Fleury, 1883)

Intrare: - un graf conex $G = (V, E)$, ale cărui vârfuri (a) au grade pare; (b) exact două vârfuri x, y au grade impare

- un vârf de pornire v_0

Ieșire: (a) un ciclu eulerian; (b) un lanț eulerian între x și y

- se setează lanțul curent ca vid, vârful curent ca și v_0 și $A = E$
- cât timp $A \neq \emptyset$
 - se selectează o muchie e incidentă vârfului curent, dar alegând o muchie tăietură numai dacă nu avem altă alternativă
 - se adaugă e la lanțul curent
 - se setează vârful curent ca și cealaltă extremitate muchiei e
 - $A = A \setminus \{e\}$
 - se șterg toate vârfurile izolate
- retur un ciclu eulerian în cazul (a) și un lanț eulerian între x și y în cazul (b)

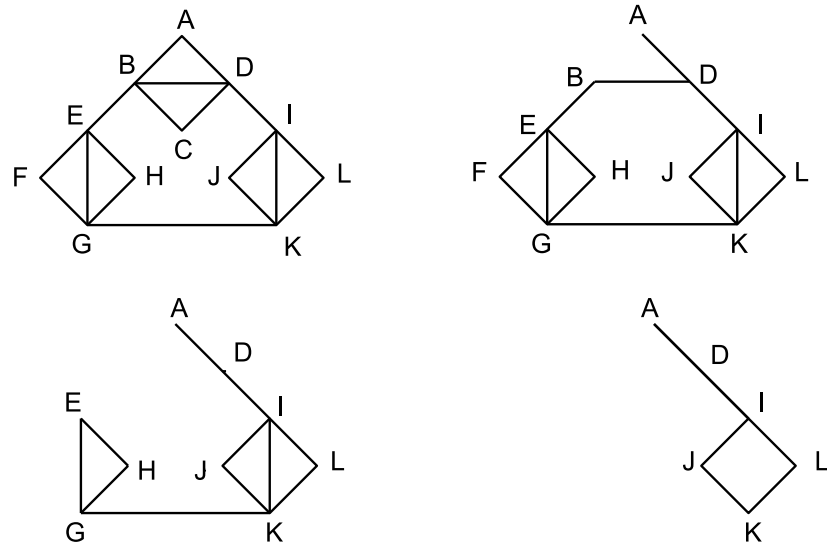


Figura 3.33: Pași din exemplul 3.18.

Exemplul 3.18 Considerăm graful din figura 3.33 (stânga sus). Vom lua vârful A ca și vârful de pornire. Primele muchii considerate în ciclul eulerian C vor fi AB, BC, CD . În acest moment, graful rămas este desenat în figura 3.33 (dreapta sus) și vârful curent este D . Muchia DA devine muchie tăietură, deci vom lua în C muchiile DB, BE, EF, FG , graful rămas fiind cel din figura 3.33 (stânga jos). Acum GK este o muchie tăietură, prin urmare vom alege GE, EH, HG, GK, KI . Apoi, ID este muchie tăietură, deci alegem IJ, JK, KL, LI, ID, DA .

Ca și aplicații ale grafurilor euleriene să menționăm colectarea resturilor menajere, măturarea străzilor, curățarea zăpezii, efectuarea marcajelor rutiere, împărțirea corespondenței.

3.8.3 Problema poștaşului

În 1962, matematicianul chinez Meigu Guan a introdus problema găsirii celui mai scurt ciclu care să traverseze fiecare muchie a unui graf cel puțin o dată. El avea în vedere un poștaş care vrea să împartă corespondența într-o rețea de străzi și să se întoarcă la oficiul poștal cât mai repede cu putință. J. Edmonds a poreclit această problemă *problema poștaşului chinez*.

Definiția 3.48 Într-un graf cu ponderi, un **ciclu optimal** este un ciclu care parcurge fiecare muchie a lui G cel puțin o dată și a cărui pondere totală este minimă.

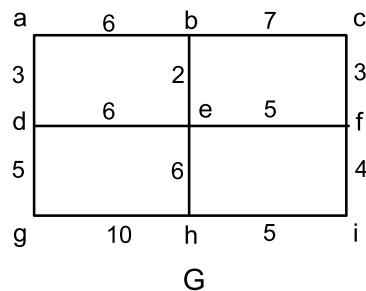
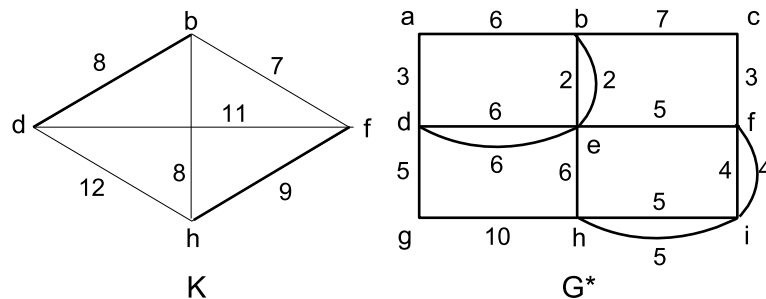
Desigur, dacă fiecare vârf al grafului are grad par, un ciclu eulerian este un ciclu optimal. În caz contrar, anumite muchii trebuie parcurse de mai multe ori. Deci, scopul este să găsim un ciclu ale cărui muchii repetate să aibă pondere minimă. Acesta corespunde, de fapt, unui ciclu eulerian în graful G^* , format din G prin adăugarea unui număr de muchii egal cu numărul de ori în care se re-parcurge o muchie.

În 1973, Edmonds și Johnson au rezolvat problema poștaşului chinez folosind următorul algoritm:

Algoritm 15 (Construcția unui ciclu optimal)Intrare: un graf conex cu ponderi G Ieșire: un ciclu optimal W

- se găsește mulțimea S de vârfuri impare din G
- pentru fiecare pereche de vârfuri impare $u, v \in S$, se găsește $d(u, v) = \text{distanța dintre } u \text{ și } v$
- se formează un graf complet K cu vârfurile lui S și în K se asociază ponderea $d(u, v)$ fiecărei muchii uv
- se găsește un cuplaj perfect M în K a cărui pondere totală este minimă
- pentru fiecare muchie e din M
 - fie p lanțul corespunzător cel mai scurt în G între extremitățile muchiei e
 - pentru fiecare muchie f a lanțului p , se adaugă la graful G o dublură a muchiei f , incluzând ponderea acesteia
- Fie G^* graful eulerian format prin adăugarea la graful G a muchiilor dublate din pasul anterior
- se construiește un ciclu eulerian W în G^* . Acest ciclu va corespunde ciclului optimal din graful original G .

Observația 3.20 Optimalitatea ciclului rezultă din faptul că am ales un cuplaj perfect de pondere minimă, format din acele muchii care corespund celui mai scurt lanț în G .

Figura 3.34: Graful cu ponderi G din exemplul 3.19.Figura 3.35: Graful complet K și graful eulerian G^* .

Exemplul 3.19 Considerăm graful cu ponderi G din figura 3.34. Vârfurile de grad impar sunt b, d, f, h . Ele vor forma graful complet K din figura 3.35 (stânga). Un cuplaj perfect în K cu pondere minimă este $M = \{bd, fh\}$. Fiecare muchie din M reprezintă un drum în G : pentru muchia bd de pondere 8 avem drumul în G $\langle b, e, d \rangle$ (de lungime 8), în timp ce pentru muchia fh

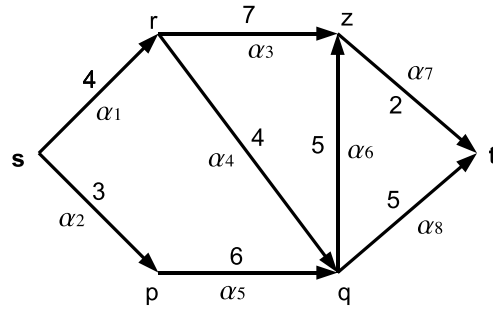


Figura 3.36: Graful asociat proiectului din exemplul 3.20.

de lungime 9 avem drumul $\langle f, i, h \rangle$. Graful G^* este reprezentat în figura 3.35 (dreapta).

În final, pentru a construi un ciclu Eulerian aplicăm algoritmul lui Fleury și obținem

$$W = \langle a, b, c, f, e, b, e, d, e, h, i, f, i, h, g, d, a \rangle.$$

3.9 Rețele

3.9.1 Drumuri critice

Fie $D = (V, A)$ un graf orientat și $w : A \rightarrow \mathbb{N}$ o funcție pondere reprezentând costuri, distanțe sau timpuri. Acest graf orientat poate fi folosit în planificarea unui proiect, atunci când proiectul poate fi împărțit în activități mai mici legate între ele. De exemplu, proiectul de construcție a unei case poate fi împărțit în mai multe activități mici, cum ar fi turnarea fundației, ridicarea zidurilor, montarea acoperișului, instalarea rețelei electrice, etc. Aceste activități sunt desigur legate, în sensul că anumite activități nu pot începe până când altele nu au fost realizate. Graful orientat asociat unui asemenea tip de proiect se construiește astfel:

- arcele reprezintă activități,
- vârfurile reprezintă evenimente (un eveniment este îndeplinirea unor activități),
- ponderea unui arc este timpul necesar activității reprezentate de arcul respectiv.

Exemplul 3.20 Vrem să planificăm activitățile $\alpha_1, \dots, \alpha_8$ astfel încât timpul total necesar pentru proiect să fie minim, cunoscând condițiile de începere pentru fiecare din activitățile α_i :

Activitate	α_1	α_2	α_3	α_4	α_5	α_6	α_7	α_8
Timp necesar (zile)	4	3	7	4	6	5	2	5
Condiții de începere	—	—	α_1	α_1	α_2	α_4	α_3	α_4
						α_5	α_6	α_5

De exemplu, activitatea α_6 nu poate începe până când activitățile α_4 și α_5 nu sunt îndeplinite. Suntem interesați de găsirea numărului minim de zile necesare pentru realizarea întregului proiect. Graful asociat proiectului este cel din figura 3.36. În acest graf, vârful s reprezintă evenimentul «începerea proiectului», t reprezintă evenimentul «proiectul este îndeplinit», iar un

vârf intermediar, de exemplu q , reprezintă evenimentul «activitățile α_4 și α_5 sunt îndeplinite».

Pentru $v \in V$ notăm cu $E(v)$ momentul cel mai devreme la care poate începe activitatea corespunzătoare lui v . Astfel, avem

$$E(s) = 0, \quad E(p) = 3, \quad E(r) = 4.$$

Apoi, pentru q , ambele activități α_4 și α_5 trebuie să fie îndeplinite, deci

$$E(q) = \max\{E(r) + w(r, q), E(p) + w(p, q)\} = \max\{4 + 4, 3 + 6\} = 9.$$

În general, pentru calculul momentelor $E(v)$ avem formulele

$$\begin{cases} E(s) = 0, \\ E(v) = \max_{u \in A^-(v)} \{E(u) + w(u, v)\}, \end{cases}$$

unde $A^-(v)$ este mulțimea extremităților inițiale ale arcelor cu extremitatea finală în v . Pentru restul vârfurilor avem

$$\begin{aligned} E(z) &= \max\{E(q) + w(q, z), E(r) + w(r, z)\} = 14, \\ E(t) &= \max\{E(z) + w(z, t), E(q) + w(q, t)\} = 16. \end{aligned}$$

Prin urmare, cel mai devreme moment pentru îndeplinirea lui t este $E(t) = 16$, însemnând că numărul minim de zile necesare proiectului este 16. De fapt, acest număr reprezintă lungimea celui mai lung drum de la s la t , și se poate găsi și prin aplicarea algoritmului BFS grafului orientat D .

Metoda descrisă anterior este o parte a unei tehnici numită *analiza drumului critic*. Restul tehnicii este în felul următor: pentru fiecare $v \in V$, se calculează numerele $L(v)$, reprezentând cele mai târzii momente până la care trebuie începute activitățile (v, x) , astfel încât proiectul să fie terminat la timp. Formulele pentru $L(v)$ sunt:

$$\begin{cases} L(t) = E(t), \\ L(v) = \min_{x \in A^+(v)} \{L(x) - w(v, x)\}, \end{cases}$$

unde $A^+(v)$ este mulțimea extremităților finale ale arcelor cu extremitatea inițială în v . Astfel, despre o activitate (y, z) știm că:

- nu poate începe înainte de momentul $E(y)$, cel mai devreme;
- trebuie să fie finalizată până în momentul $L(z)$, cel mai târziu;
- timpul de realizare este $w(y, z)$.

Definiția 3.49 Pentru o activitate (y, z) , **timpul de lansare** $F(y, z)$ este definit prin

$$F(y, z) = L(z) - E(y) - w(y, z).$$

Astfel, (y, z) poate începe în orice moment după $E(y)$ și înainte de $E(y) + F(y, z)$, fără a întârzia proiectul.

Definiția 3.50 O activitate (y, z) pentru care timpul de lansare $F(y, z)$ este zero se numește **critică**.

O activitate critică trebuie începută la cel mai devreme moment posibil $E(y)$, dacă vrem ca proiectul să fie terminat la timp. În graful orientat asociat proiectului va exista cel puțin un drum elementar de la s la t care constă în întregime din activități critice, și acest drum se numește **drum critic**. Acest drum elementar este de fapt unicul drum în arborele de acoperire BFS asociat digrafului, unde prioritatea considerată este cea mai mare pondere.

În exemplul nostru, numerele L sunt:

$$\begin{aligned} L(t) &= 16, \\ L(q) &= \min\{L(t) - w(q, t), L(z) - w(q, z)\} = 9, \\ L(p) &= L(q) - w(p, q) = 3, \\ L(r) &= \min\{L(z) - w(r, z), L(q) - w(r, q)\} = 5, \\ L(s) &= \min\{L(r) - w(s, r), L(p) - w(s, p)\} = 0. \end{aligned}$$

Timpii de lansare vor fi:

$$\begin{aligned} F(s, r) &= L(r) - E(s) - w(r, s) = 5 - 0 - 4 = 1, \\ F(s, p) &= L(p) - E(s) - w(p, s) = 3 - 0 - 3 = 0, \\ F(p, q) &= L(q) - E(p) - w(p, q) = 9 - 3 - 6 = 0, \\ F(r, q) &= L(q) - E(r) - w(r, q) = 9 - 4 - 4 = 1, \\ F(r, z) &= L(z) - E(r) - w(r, z) = 14 - 4 - 7 = 3, \\ F(z, t) &= L(t) - E(z) - w(t, z) = 16 - 14 - 2 = 0, \\ F(q, z) &= L(z) - E(q) - w(q, z) = 14 - 9 - 5 = 0, \\ F(q, t) &= L(t) - E(q) - w(q, t) = 16 - 9 - 5 = 2. \end{aligned}$$

Prin urmare, activitățile $(s, p) = \alpha_2$, $(p, q) = \alpha_5$, $(q, z) = \alpha_6$, $(z, t) = \alpha_7$, sunt activități critice. Despre activitățile care nu sunt critice putem spune următoarele:

$$\begin{aligned} (s, r) &= \alpha_1 \text{ poate începe după momentul } E(s) = 0 \text{ și înainte de momentul } E(s) + 1 = 1, \\ (r, z) &= \alpha_3 \text{ poate începe după momentul } E(r) = 4 \text{ și înainte de momentul } E(r) + 3 = 7, \\ (r, q) &= \alpha_4 \text{ poate începe după momentul } E(r) = 4 \text{ și înainte de momentul } E(r) + 1 = 5, \\ (q, t) &= \alpha_8 \text{ poate începe după momentul } E(r) = 9 \text{ și înainte de momentul } E(r) + 2 = 11. \end{aligned}$$

Se poate construi cu ușurință un drum critic:

$$\langle s, p, q, z, t \rangle.$$

3.9.2 Flux și tăietură

Intr-un graf orientat, mulțimea arcelor poate fi privită ca și o rețea de conducte prin care se transportă fluide, ponderea unui arc reprezentând în acest caz capacitatea conductei. Într-un asemenea graf, există două vârfuri, s (sursa) și t (scurgerea, destinația), având un rol special:

- toate arcele care conțin s sunt orientate dinspre s ;
- toate arcele care conțin t sunt orientate spre t .

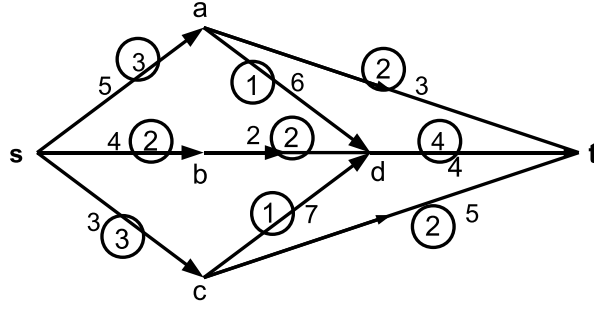


Figura 3.37: Rețeaua din exemplul 3.21.

Deci, avem de-a face cu un graf conex orientat $D = (V, A)$, o **funcție de capacitate** $c : A \rightarrow \mathbb{R}_+$, o **sursă** s și o **destinație** t . Un asemenea graf se va numi **rețea de transport**, sau pe scurt **rețea**.

Presupunem că de-a lungul arcelor curge un fluid și fie $f(x, y)$ cantitatea de fluid care curge prin arcul (x, y) . Cantitatea de fluid care ajunge la un vârf v ar trebui să fie egală cu cantitatea care iese din v , exceptând vârfurile s și t . Dacă definim

$$f^+(v) = \sum_{(x,v) \in A} f(x, v),$$

$$f^-(v) = \sum_{(v,y) \in A} f(v, y),$$

această cerință se poate scrie $f^+(v) = f^-(v)$, pentru $v \neq s, t$. Trebuie cerut de asemenea ca nici un arc să nu transporte mai mult decât capacitatea sa. În concluzie, putem da următoarea definiție:

Definiția 3.51 *Un **flux** de la sursa s la destinația t într-o rețea de transport este o funcție care atribuie arcului (x, y) un număr nenegativ $f(x, y)$, astfel încât:*

- a) $f^+(v) = f^-(v)$, pentru $v \neq s, t$ (legea conservării);
- b) $f(x, y) \leq c(x, y)$, pentru orice $(x, y) \in A$ (legea fezabilității).

Desigur, nu este permisă acumularea de fluid în vârfurile intermediare și aceasta înseamnă

$$f^-(s) = f^+(t).$$

Valoarea comună se va numi **valoarea fluxului** f și va fi notată cu $val(f)$.

Exemplul 3.21 *Considerăm rețeaua dată în figura 3.37. Funcția f definită în tabelul următor este un flux în această rețea de transport și are valoarea $val(f) = 8$.*

(x, y)	(s, a)	(s, b)	(s, c)	(a, d)	(b, d)	(c, t)	(a, t)	(c, t)	(d, t)
$f(x, y)$	3	2	3	1	2	1	2	2	4

În continuare vom calcula valoarea maximă a unui flux pentru rețeaua dată în figura 3.37.

Primul pas este găsirea unei margini superioare pentru această valoare, în funcție de capacități. Intrucât din s se poate transporta cantitatea $5 + 4 + 3 =$

12, fluxul ar trebui să fie ≤ 12 . Ideea este să partiționăm mulțimea vârfurilor în două părți, S conținând s și T conținând t . Astfel, fluxul de la S la T va fi egal cu fluxul de la s la t și această valoare comună va fi $val(f)$. Valoarea $val(f)$ poate fi scrisă ca și

$$val(f) = \sum_{x \in S, y \in T} f(x, y) - \sum_{u \in T, v \in S} f(u, v),$$

unde prima sumă reprezintă fluxul total de la S la T și a doua sumă reprezintă fluxul total în direcție inversă. În exemplul nostru, dacă $S = \{s, a, b, c\}$ și $T = \{d, t\}$, atunci într-adevăr

$$val(f) = f(a, d) + f(a, t) + f(b, d) + f(c, d) + f(c, t) - 0 = 8.$$

Desigur, aceeași valoare se obține pentru partiția $\{S, T\}$ cu $S = \{s, b\}$, $T = \{a, c, d, t\}$:

$$val(f) = f(s, a) + f(s, c) + f(b, d) - 0 = 8.$$

De asemenea, să menționăm că, pentru ultima partiție

$$val(f) \leq c(s, a) + c(s, c) + c(b, d) = 10. \quad (3.11)$$

În acest moment este necesar să dăm următoarea definiție.

Definiția 3.52 *Intr-o rețea $D = (V, A)$, cu sursa s , destinația t și funcția de capacitate $c : A \rightarrow \mathbb{R}_+$, perechea (S, T) se numește **tăietură** dacă $\{S, T\}$ este o partiție a lui V astfel încât $s \in S$ și $t \in T$. **Capacitatea tăieturii** este definită ca și*

$$cap(S, T) = \sum_{x \in S, y \in T} c(x, y).$$

Putem deduce imediat următorul rezultat:

Teorema 3.45 *Fie s sursa și t destinația unei rețele de transport. Dacă $f : A \rightarrow \mathbb{R}_+$ este un flux arbitrar de la s la t și (S, T) este o tăietură, atunci*

$$val(f) \leq cap(S, T).$$

3.9.3 Flux maxim, tăietură minimă

O consecință imediată a teoremei anterioare este următoarea: dacă f_0 este un flux cu valoare maximă și (S_0, T_0) este o tăietură cu capacitate minimă, atunci

$$val(f_0) \leq cap(S_0, T_0)$$

însemnând că flux maxim \leq tăietură minimă. De fapt, cele două valori sunt egale, acest lucru fiind demonstrat mai târziu în teorema 3.46.

Ideea folosită pentru construcția unui flux maxim este de a îmbunătăți valoarea unui flux dat, dacă acesta nu are cea mai mare valoare posibilă. Vom ilustra această idee pentru rețeaua din figura 3.37 și pentru fluxul din exemplul 3.21.

Există două tipuri de îmbunătățiri:

Tipul 1: Considerăm drumul elementar $\langle s, a, t \rangle$. Nici (s, a) , nici (a, t) nu transportă fluid la capacitatea lor maximă, deci putem mări fluxul pe ambele arce, până când este atinsă capacitatea unuia dintre ele. Dacă definim

$$\begin{aligned} f_1(s, a) &= 4, \\ f_1(a, t) &= 3, \end{aligned}$$

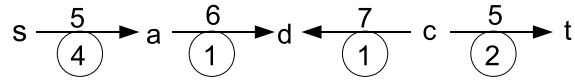
atunci arcul (a, t) este saturat. Mai departe, deoarece fluxurile pe ambele arce au fost mărite cu aceeași cantitate, datorită legii conservării pentru vârful a avem

$$f_1(x, y) = f(x, y)$$

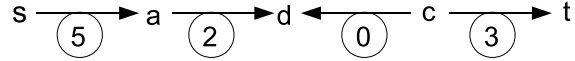
pe arcele rămase. Prin urmare, am obținut un nou flux f_1 cu

$$val(f_1) = val(f) + 1 = 9.$$

Tipul 2: Considerăm lanțul $\langle s, a, d, c, t \rangle$ în graful suport.



Acest lanț nu este un drum în rețea din cauză că (d, c) nu este arc. Arcul (c, d) are direcție opusă drumului de la s la t , prin urmare putem reduce cu 1 fluxul pe (c, d) și putem mări cu 1 fluxurile pe celelalte arce ale lanțului, fără să violăm legea conservării. Astfel, putem defini un nou flux f_2 ca și



și $f_2 = f_1$ pentru restul arcelor. Arcul (s, a) este astfel saturat și fluxul pe (c, d) nu poate fi negativ, deci nu putem face o schimbare mai bună în acest lanț. Valoarea noului flux f_2 va fi

$$val(f_2) = val(f_1) + 1 = 10,$$

și reprezintă valoarea maximă, așa cum s-a constatat în (3.11). Prin urmare, f_2 este un flux maxim.

Lanțurile $\langle s, a, t \rangle$ și $\langle s, a, d, c, t \rangle$ folosite pentru creșterea fluxurilor f și f_1 sunt *lanțuri crescătoare* ale fluxurilor respective.

Definiția 3.53 *Dat fiind fluxul f într-o rețea, un lanț*

$$p = \langle s = x_1, x_2, \dots, x_{k-1}, x_k = t \rangle$$

*în graful suport se numește **lanț crescător** pentru f dacă*

$$f(x_i, x_{i+1}) < c(x_i, x_{i+1}) \text{ și } (x_i, x_{i+1}) \in A \quad (3.12)$$

$$\text{sau } f(x_{i+1}, x_i) > 0 \text{ și } (x_{i+1}, x_i) \in A, \quad (3.13)$$

pentru $1 \leq i \leq k - 1$.

De fapt, (3.12) înseamnă că arcele orientate direct nu sunt folosite la întreaga lor capacitate, în timp ce (3.13) înseamnă că arcele în sens invers transportă «contra-fluid». De fapt, un lanț crescător pentru f este un lanț care poate fi «îmbunătățit». Dat fiind un asemenea lanț, putem mări fluxul pe arcele

directe și micșora fluxul pe arcele contrare cu aceeași cantitate, fără a viola legea conservării.

Cea mai mare schimbare (fără a supraîncărca arcele directe și fără a face negativ fluxul prin arcele contrare) este minimul, pentru $1 \leq i \leq k-1$, al cantităților

$$\alpha(i) = \begin{cases} c(x_i, x_{i+1}) - f(x_i, x_{i+1}), & \text{dacă } (x_i, x_{i+1}) \in A, \\ f(x_{i+1}, x_i), & \text{dacă } (x_{i+1}, x_i) \in A. \end{cases}$$

Acest minim va fi notat cu α și va fi numit **capacitate reziduală** a lanțului p . Dacă adunăm α la fluxul prin arcele directe și scădem α din fluxul prin arcele contrare, obținem fluxul f^* cu

$$val(f^*) = val(f) + \alpha > val(f),$$

prin urmare f a fost mărit.

În concluzie, existența unui lanț crescător de la s la t ne permite găsirea unui nou flux f^* cu valoarea $val(f^*) > val(f)$. Această idee va fi folosită pentru a demonstra teorema următoare. În demonstrație avem nevoie de următoarea definiție.

Definiția 3.54 *Un lanț crescător **incomplet** pentru f este un lanț care satisface condițiile (3.12) – (3.13) de lanț crescător pentru f , cu excepția faptului că vârful final nu este t .*

Teorema 3.46 (Teorema flux maxim tăietură minimă)

Valoarea maximă a unui flux de la s la t într-o rețea de transport este egală cu minimul capacității unei tăieturi care separă s și t :

$$\text{flux maxim} = \text{tăietură minimă}.$$

Demonstrație. Fie f un flux maxim. Definim mulțimile de vârfuri

$$\begin{aligned} S &= \{x \in V, \text{ există lanț crescător incomplet pentru } f, \text{ de la } s \text{ la } x\}, \\ T &= V \setminus S. \end{aligned}$$

Atunci $t \in T$, altfel ar exista un lanț crescător pentru f , de la s la t și f ar putea fi mărit, contrar ipotezei că f este un flux maxim. Prin urmare (S, T) este o tăietură.

Trebuie să demonstrăm că

$$cap(S, T) = val(f).$$

• Fie (x, y) un arc cu $x \in S$, $y \in T$. Din definiția lui S deducem că există un lanț crescător incomplet pentru f , de la s la x . Dacă $f(x, y) < c(x, y)$, atunci am putea extinde acest lanț la y , contradicție cu $y \in T$. În concluzie,

$$f(x, y) = c(x, y).$$

• Fie (u, v) un arc cu $u \in T$, $v \in S$. Atunci există un lanț crescător incomplet pentru f , de la s la v . Dacă $f(u, v) > 0$, am putea extinde lanțul incomplet la u , contradicție cu $u \in T$. Astfel,

$$f(u, v) = 0.$$

În concluzie,

$$\text{val}(f) = \sum_{x \in S, y \in T} f(x, y) - \sum_{u \in T, v \in S} f(u, v) = \sum_{x \in S, y \in T} c(x, y) = \text{cap}(S, T).$$

Dacă (S', T') este o altă tăietură, atunci

$$\text{cap}(S', T') \geq \text{val}(f) = \text{cap}(S, T),$$

de unde rezultă că tăietura (S, T) are capacitate minimă. În consecință teorema este demonstrată. ■

3.9.4 Algoritmi pentru găsirea unui flux maxim cu valori întregi

Pe parcursul acestui paragraf vom considera rețeaua $D = (V, A)$ cu sursa s , destinația t și funcția de capacitate $c : A \rightarrow \mathbb{R}_+$.

Primul algoritm, construit de Ford și Fulkerson, se bazează pe ideea folosită în demonstrația teoremei 3.46: dat fiind un flux f , căutăm un lanț crescător pentru f și vom construi un nou flux f^* , cum s-a descris la pagina 134.

Pentru al doilea algoritm, construit de Edmonds și Karp, avem nevoie de următoarea definiție.

Definiția 3.55 Pentru un flux f și un arbore T într-o rețea $D = (V, A)$, un arc (x, y) se numește **arc frontieră util** pentru f dacă:

$$\begin{aligned} & x \in T, y \notin T \text{ și } f(x, y) < c(x, y) \\ \text{sau} & x \notin T, y \in T \text{ și } f(x, y) > 0. \end{aligned}$$

Vom da în continuare algoritmul pentru construcția unui arbore T , folosind arce frontieră utile.

Algoritm 16 (Edmonds, Karp)

Intrare: o rețea $D = (V, A)$ cu sursa s , destinația t , funcția de capacitate $c : A \rightarrow \mathbb{N}$ și un flux $f : A \rightarrow \mathbb{N}$

Ieșire: un arbore T

- se inițializează $T = \{s\}$
- se scrie eticheta 0 pe vârful s
- $i=1$
- cât timp T nu conține vârful t și mai există arce frontieră utile pentru T
 - se actualizează mulțimea arcelor frontieră utile
 - fie $a = (x, y)$ arcul frontieră util cu vârful etichetat având cea mai mică valoare posibilă
 - se adaugă la arborele T arcul a și extremitatea neetichetată a arcului a
 - se atașează eticheta i vârfului neetichetat al lui a
 - $i=i+1$
- retur arbore T

Legătura dintre arborele T și fluxul maxim este dată în următoarea teoremă.

Teorema 3.47 *Fie $D = (V, A)$ o rețea cu sursa s , destinația t , funcția de capacitate $c : A \rightarrow \mathbb{N}$ și fie $f : A \rightarrow \mathbb{N}$ un flux. Dacă arborele T rezultat prin aplicarea algoritmului 16 conține destinația t , atunci unicul lanț în T de la s la t este un lanț crescător pentru f . Dacă T nu conține t , atunci f este un flux maxim.*

Demonstrația acestei teoreme poate fi găsită de exemplu în [13], iar faptul că fluxul ia valori întregi este o condiție esențială.

Anexa A: Relații binare

- **Produsul cartezian** a două mulțimi A și B este mulțimea

$$A \times B = \{(x, y), x \in A, y \in B\}.$$

- O **relație binară** este un triplet (A, B, R) , unde A, B sunt mulțimi arbitrare și $R \subseteq A \times B$.
- Dacă $(x, y) \in R$, atunci spunem că x este **în relație** cu y (prin R). Acest lucru se notează deseori cu $x R y$.
- Relația (A, A, R) , $R \subseteq A \times A$ se numește relație binară pe mulțimea A .
- O relație binară pe A se numește:
 - **reflexivă** dacă $x R x$ pentru orice $x \in A$
 - **tranzitivă** dacă $x R y$ și $y R z$ implică $x R z$
 - **simetrică** dacă $x R y$ implică $y R x$
 - **antisimetrică** dacă $x R y$ și $y R x$ implică $x = y$

A.1 Relații de echivalență

Definiția .1 O relație $R \subseteq A \times A$ este o **relație de echivalență** pe A dacă R este reflexivă, tranzitivă și simetrică.

Exemplul .1 Fie $R \subseteq \mathbb{Z} \times \mathbb{Z}$, dată prin

$$x R y \iff x - y \in 3\mathbb{Z},$$

unde $3\mathbb{Z} = \{3k, k \in \mathbb{Z}\}$. Atunci R este o relație de echivalență pe \mathbb{Z} .

Definiția .2 Fie R o relație de echivalență pe o mulțime A și fie $x \in A$. **Clasa de echivalență a lui x** , notată \hat{x} sau $R\langle x \rangle$, este mulțimea

$$\hat{x} = \{y \in A, x R y\}.$$

Mulțimea tuturor claselor de echivalență, notată A/R , se numește **mulțime cât**:

$$A/R = \{\hat{x}, x \in A\}.$$

Exemplul .2 Pentru relația de echivalență definită în exemplul .1,

$$\begin{aligned}\hat{0} &= \{3k, k \in \mathbb{Z}\}, \\ \hat{1} &= \{3k + 1, k \in \mathbb{Z}\}, \\ \hat{2} &= \{3k + 2, k \in \mathbb{Z}\}.\end{aligned}$$

Observăm că clasele de echivalență sunt două câte două disjuncte și orice număr întreg aparține uneia din aceste clase.

Definiția .3 O colecție de submulțimi $\{S_1, S_2, \dots, S_n\}$ ale unei mulțimi A se numește **partiție** a lui A dacă satisface următoarele două condiții:

- $S_i \cap S_j = \emptyset$, pentru orice $1 \leq i < j \leq n$,
- $\bigcup_{i=1}^n S_i = A$.

Propoziția .48 Fie R o relație de echivalență pe o mulțime A și fie $x, y \in A$. Atunci următoarele afirmații sunt echivalente:

1. $x R y$,
2. $\hat{x} = \hat{y}$,
3. $\hat{x} \cap \hat{y} \neq \emptyset$.

Corolarul .49 Fie R o relație de echivalență pe o mulțime A . Atunci clasele de echivalență formează o partiție a lui A :

$$\{\hat{x}, x \in A\} \quad \text{este o partiție.}$$

Reciproc, dată fiind o partiție $\{S_1, S_2, \dots, S_n\}$ a unei mulțimi A , există o relație de echivalență pe A . Aceasta este definită astfel:

$$x R y \iff \exists i \in \mathbb{N}_n \text{ astfel încât } x, y \in S_i.$$

Exemplul .3 Clasele de echivalență $\hat{0}, \hat{1}, \hat{2}$ din exemplul .2 formează o partiție a lui \mathbb{Z} . Intr-adevăr,

$$\hat{0} \cup \hat{1} \cup \hat{2} = \mathbb{Z} \text{ și } \hat{0} \cap \hat{1} = \emptyset, \hat{0} \cap \hat{2} = \emptyset, \hat{1} \cap \hat{2} = \emptyset.$$

În acest caz, mulțimea cât este

$$\mathbb{Z}/R = \{\hat{0}, \hat{1}, \hat{2}\} = \mathbb{Z}_3.$$

A.2 Relații de ordine

Definiția .4 Fie A o mulțime și $R \subseteq A \times A$.

R este o **relație de ordine parțială** pe A dacă R este reflexivă, tranzitivă și antisimetrică. În acest caz, perechea (A, R) se numește **mulțime ordonată parțial**, sau simplu **mulțime ordonată**.

R este **relație de ordine totală** dacă este o relație de ordine parțială și pentru orice $x, y \in A$ avem $x R y$ sau $y R x$ (adică oricare două elemente sunt comparabile prin relația de ordine). În acest caz, perechea (A, R) se numește **mulțime total ordonată**.

Exemplul .4 (\mathbb{R}, \leq) este o mulțime total ordonată, dar $(\mathcal{P}(\mathbb{R}), \subseteq)$ este doar o mulțime parțial ordonată. Aici $\mathcal{P}(\mathbb{R}) = \{S, S \subseteq \mathbb{R}\}$.

Elemente speciale

În mulțimile parțial ordonate pot exista anumite elemente care joacă un rol special.

Definiția .5 Fie (A, \leq) o mulțime parțial ordonată.

Elementul $x \in A$ se numește **cel mai mic element** al lui A dacă

$$x \leq y, \text{ pentru orice } y \in A.$$

Elementul $z \in A$ se numește **cel mai mare element** al lui A dacă

$$y \leq z, \text{ pentru orice } y \in A.$$

Elementul $m \in A$ se numește **element minimal** pentru A dacă

$$x \leq m \text{ pentru un anumit } x \in A \implies x = m.$$

Elementul $M \in A$ se numește **element maximal** pentru A dacă

$$M \leq x \text{ pentru un anumit } x \in A \implies x = M.$$

Ceea ce este important de menționat despre un element maximal este faptul că nu este în general cel mai mare element, adică el nu este întotdeauna mai mare decât toate celelalte elemente. Intr-adevăr, considerăm (A, \subseteq) , cu

$$A = \{\{n\}, n \in \mathbb{N}\}.$$

Ea constă doar din elemente maximale, dar nu are un cel mai mare element. Mai mult, toate elementele lui A sunt minimale. Acest exemplu ne arată de asemenea că elementul maximal poate să nu fie unic, și că se poate întâmpla ca un element să fie maximal și minimal în același timp.

Dacă o mulțime are un cel mai mare element, atunci acesta este unicul element maximal. Reciproc, chiar dacă o mulțime are doar un element maximal, acesta nu este în mod necesar cel mai mare element.

Într-o mulțime total ordonată, termenii element maximal și cel mai mare element coincid.

Bibliografie

- [1] M. Aigner, *Discrete Mathematics*, American Mathematical Society, Rhode Island, 2007.
- [2] N. L. Biggs, *Discrete Mathematics*, Oxford University Press, 2005.
- [3] J.A. Bondy and U.S. Murty, *Graph Theory with Applications*, Elsevier North-Holland, 1982.
- [4] I.N. Bronshtein, K.A. Semendyayev, *Handbook of Mathematics*, Springer, 1997.
- [5] K. Devlin *Partida neterminată*, Ed. Humanitas, București 2015.
- [6] R. Durrett, *The Essentials of Probability*, Duxbury Press, 1994.
- [7] J. Gross, J. Yellen, *Graph Theory and its Applications*, CRC Press, 1999.
- [8] H. Lisei, *Probability Theory*, Casa cărții de Știință, Cluj-Napoca, 2004.
- [9] H. Lisei, S. Micula, A. Soós, *Probability Theory through Problems and Applications*, Cluj University Press, 2006.
- [10] J. Michael, J. Gross, J. Grossman, S. Douglas (editors), *Handbook of Discrete and Combinatorial Mathematics*, CRC Press, 2000.
- [11] I. Mihoc, *Calculul probabilităților și statistică matematică*, litogr. Univ. Babeș-Bolyai, 1994.
- [12] D. Stirzaker, *Elementary Probability*, Cambridge University Press, 1995.
- [13] N. Vornicescu, *Grafe: teorie și algoritmi*, Ed. Mediamira, 2005.
- [14] D. Wackerly, W. Mendenhall and R.L. Scheaffer, *Mathematical Statistics with Applications*, Duxbury Press, 2001.
- [15] *Wikipedia: The Free Encyclopedia*, <http://www.wikipedia.org>