3. ELEMENTE DE LOGICĂ MATEMATICĂ

Logica matematică reuneşte teoria mulţimilor, teoria algoritmilor, teoria modelelor şi teoria demonstraţiilor. Vom face cunoştinţă cu unele momente legate de **teoria algoritmilor**, care studiază modelele matematice ale operaţiilor mecanice executate de oameni sau dispozitive speciale atunci când rezolvă probleme de masă de acelaşi tip, în capitolul 5. **Teoria modelelor** a apărut prin aplicarea metodelor logicii matematice la algebră, transformându-se într-o disciplină de-sine-stătătoare, care studiază modelele matematice ale teoriilor ştiinţifice, punând în evidenţă legile comune tuturor teoriilor ce se exprimă printr-un limbaj formalizat dat. **Teoria demonstraţiilor**, care reprezintă partea principală a logicii matematice, studiază modelele matematice ale procesului gândirii, structura gândirii, ale raţionamentelor utilizate în matematică. Orice proces de gândire, printre care şi cel utilizat în matematică, este legat de următoarele patru obiecte:

1. Un limbaj L în care se exprimă premisele iniţiale (datele) ale gândirii, diferite momente ale gândirii, rezultatele obţinute prin raţionament. De obicei, L este limba unui popor, îmbogăţită cu termeni şi concepte caracteristice teoriei obiectelor studiate.
2. O clasă K a obiectelor studiate N.
3. Un concept de adevăr al enunţului α în limbajul L privind obiectul studiat $N \in K$.
4. Un proces de elaborare a enunţurilor folosite în raţionament, constând în trecerea de la unele enunţuri, numite **premise**, la un enunţ nou, numit **consecinţă** a enunţurilor iniţiale.

Modelul matematic al procesului gândirii constă din următoarele modele:

- Limbajul formalizat L_f - modelul matematic al limbajului L;
- Modelul matematic N_m al obiectului studiat N;
- Definiţia exactă a conceptului de adevăr a enunţului α din limbajul L_f în modelul N_m;
- Calculul logic sau modelul matematic al trecerii de la premise la consecinţe.

Dintre principalele orientări ale logicii matematice pot fi menţionate logica clasică, logica intuiţionistă, logica polivalentă, logica hibridă, logica fuzzy, etc.

Definirea riguroasă a problemelor legate de ştiinţa calculatoarelor, informatica teoretică şi aplicată este bazată pe principiile logicii matematice. Problemele tehnice privind circuitele logice şi comenzile secvenţiale, majoritatea modelelor matematice utilizate în inteligenţa artificială nu pot fi concepute în afara acestui domeniu.

3.1. Funcţiile algebrei logicii

Vom evidenţia în mod deosebit mulţimea B care conţine două elemente, notate de obicei prin 0 şi 1 $B=\{0,1\}$. Aceste elemente nu vor fi considerate numere în sensul obişnuit. Ne vom opri la interpretarea logică: 0 în sens de *nu* sau *fals* şi 1 în sens de *da* sau *adevărat*; (vă amintiţi de programare - *FALS* şi *TRUE*). Într-un caz mai general această interpretare nu este obligatorie - adesea elementele mulţimii B pot fi considerate simboluri formale care nu au un sens aritmetic.

Algebra $A = <B, F>$, în care F este mulţimea operaţiilor $f: B^n \to B$, $n = 1, 2,..., m$ se numeşte **algebra logicii** sau **booleană**, după numele matematicianului şi logicianului englez George Boole (1815 - 1864). Operaţiile $f: B^n \to B$ se numesc **funcţii ale algebrei logicii** sau **funcţii logice**, **funcţii booleene (FB).** Cu alte cuvinte, o funcţie booleană de n argumente $f(x_1, x_2,..., x_n)$ are domeniul de valori şi domeniul de definiţie mulţimea $B=\{0,1\}$. Mulţimea tuturor funcţiilor logice de n argumente notăm prin $P_2(n)$.

Luând în locul lui B o mulţime finită M de cardinal k de simboluri formale împreună cu toate operaţiile definite pe M vom ajunge la **logica polivalentă**. Mulţimea tuturor funcţiilor logice în logica polivalentă se va nota prin $P_k(n)$.

O **algebră booleană** mai poate fi definită ca o **latice complementară şi distributivă** cu axiomele şi proprietăţile respective (v.p. 2.8).

O funcţie logică $f(x_1, x_2,..., x_n)$ poate fi definită cu ajutorul unui tabel care conţine $n+1$ coloane. Primele n coloane vor enumera toate combinaţiile posibile ale argumentelor, iar ultima determină valorile posibile ale funcţiei.

Exemplul 3.1. Pentru o funcţie de trei argumente $f(x_1, x_2, x_3)$ putem avea următorul tabel:

Tabelul 3.1. Tabel de adevăr

x_1	x_2	x_3	$f(x_1,x_2,x_3)$	x_1	x_2	x_3	$f(x_1,x_2,x_3)$
0	0	0	**1**	1	0	0	**1**
0	0	1	**0**	1	0	1	**0**
0	1	0	**0**	1	1	0	**0**
0	1	1	**0**	1	1	1	**1**

Se va mai spune că combinaţia *(000)* este aplicată în *1*, *(001)* - în *0* ş.a.m.d. Combinaţiile au fost enumerate în ordine lexicografică, începând de la *0*, reprezentat în binar prin *(000)* până la *7* - *(111)*. Avem 8 combinaţii posibile în cazul unei funcţii de trei argumente. ◄

Pentru cazul unei funcţii de n argumente observăm că combinaţiile posibile ale argumentelor sunt vectori binari de lungime n. Este uşor de demonstrat, că mulţimea tuturor combinaţiilor posibile este $k = 2^n$ (v. teorema 2.1). Din aceleaşi considerente pot exista $2^k = 2^{2^n}$ funcţii booleene distincte. Întradevăr, mulţimea tuturor vectorilor binari de lungime n prin definiţie reprezintă produsul cartezian B^n. Cardinalul acestui produs este $|B| = 2^n$. Analogic, deoarece oricare funcţie booleană de n argumente este un vector binar de lungime $k = 2^n$, există 2^{2^n} funcţii booleene.

Tabelele de tipul lui 3.1 se numesc **tabele de adevăr**.

Există situaţii când pentru unele combinaţii ale valorilor argumentelor valoarea FB nu este determinată. **Funcţiile Booleene nedeterminate** pentru una sau mai multe combinaţii ale valorilor argumentelor se numesc **incomplet definite**. În tabelul de adevăr valorile nedefinite le vom indica cu asterisc (*).

Exemplul 3.2. Fie $f(x_1, x_2, x_3)$ o FB dată prin tabelul de adevăr 3.2.

Tabelul 3.2. Exemplu de FB

x_1	x_2	x_3	$f(x_1,x_2,x_3)$	x_1	x_2	x_3	$f(x_1,x_2,x_3)$
0	0	0	**1**	1	0	0	*
0	0	1	*	1	0	1	**0**
0	1	0	**0**	1	1	0	**0**
0	1	1	**0**	1	1	1	*

Funcţia este nedeterminată pentru combinaţiile *(001)*, *(100)* şi *(111)* ale valorilor argumentelor, ele putând fi aplicate uneori arbitrar în *0* sau *1*. Funcţiile incomplet definite se întâlnesc frecvent în practica comenzilor secvenţiale, evidenţierea situaţiilor nedefinite şi atribuirea unor valori la necesitate fiind extrem de importantă pentru simplificarea lor. ◄

Funcţii booleene de un singur argument pot fi patru:

Tabelul 3.3. FB de un argument

x	$f_0(x)$	$f_1(x)$	$f_2(x)$	$f_3(x)$
0	0	0	1	1
1	0	1	0	1

Funcţiile f_0 şi f_3 se numesc constanta *0* si, respectiv, constanta *1*. Valorile lor nu depind de valoarea argumentului. În acest caz se va spune că argumentul este nesemnificativ, redundant sau fictiv. Funcţia f_1 repetă valoarea argumentului x:

vom scrie $f_1(x) = x$. Funcţia $f_2(x)$ se numeşte negaţia lui x (*NONx*) şi se notează $\bar{x}$ (*x barat*) sau $\rceil x$.

Exista 16 *FB* de două argumente:

Tabelul 3.4. FB de două argumente

x_1	x_2	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7
0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1
x_1	x_2	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1

FB $f_0(x_1,x_2)$ şi $f_{15}(x_1,x_2)$ sunt aceleaşi constante *0* şi *1*, respectiv, numai că în acest caz ambele argumente sunt fictive, iar funcţiile mai pot fi numite degenerate.

Funcţia $f_1(x_1,x_2)$, care aplică combinaţiile celor două argumente în *1* atunci şi numai atunci, când ambele argumente au valoarea *1*, se numeşte **conjuncţie, produs logic** sau **funcţia logică ŞI, (AND)**. Se notează $x_1\&x_2$, x_1x_2 sau $x_1\wedge x_2$.

Funcţia $f_7(x_1,x_2)$ are valoarea *1* dacă cel puţin unul din argumente este *1* şi se numeşte **disjuncţie, sumă logică, funcţia logică SAU (OR).** Se notează $x_1 \vee x_2$ sau x_1+x_2.

FB $f_6(x_1,x_2)$ se numeşte **suma modulo 2**, funcţia *SAU-EXCLUSIV* sau *neechivalenţă* şi se notează $f_6(x_1,x_2) = x_1\oplus x_2$.

Funcţiile $f_3(x_1,x_2)$ şi $f_5(x_1,x_2)$ corespund valorilor argumentelor: $f_3(x_1,x_2) = x_1$ şi $f_5(x_1,x_2) = x_2$ şi se numesc funcţii identitate, iar $f_{10}(x_1,x_2)$ şi $f_{12}(x_1,x_2)$ corespund funcţiilor $f_3(x_1,x_2)$ şi $f_5(x_1,x_2)$ negate.

Funcţia $f_8(x_1,x_2) = \overline{f_7(x_1,x_2)}$ poartă denumirea de **funcţia lui Pierce** şi se notează: $f_8(x_1,x_2) = x_1\uparrow x_2$. Se mai numeşte această funcţie *NICI* sau *NOR* din cauza că coincide cu $\overline{x_1+x_2}$.

FB $f_9(x_1,x_2)$ care are valoarea 1 atunci şi numai atunci când valorile argumentelor coincid se numeşte funcţia de **echivalenţă** şi se notează prin $f_9(x_1,x_2) = x_1\leftrightarrow x_2$ sau $x_1\sim x_2$.

Funcţia $f_{13}(x_1,x_2)$ are valoarea 0 numai în cazul când x_1 este 1, iar $x_2 = 0$. Ea se numeşte **implicaţie**. Se va mai spune că x_1 **implică** x_2 sau **dacă** x_1 **atunci** x_2 şi vom nota $x_1 \to x_2$. Analogic, funcţia $f_{11}(x_1,x_2)$ este implicaţia lui x_2, *în* x_1: $x_2 \to x_1$.

FB $f_{14}(x_1,x_2)$ este 0 dacă şi numai dacă ambele argumente sunt 1 şi se numeşte funcţia lui **Sheffer** sau *ŞI-NU* (în engleză *NAND*). Se notează $f_{14}(x_1,x_2) = x_1|x_2$ sau $\overline{x_1 x_2}$.

Observăm, că în cazul funcţiilor de un argument jumătate din FB sunt degenerate (cu argumente fictive). În cazul a două argumente din 16 funcţii booleene şase sunt cu argumente fictive. Odată cu creşterea lui n numărul funcţiilor degenerate descreşte, tinzând către 0, când n tinde spre infinit.

3.2. Transformări echivalente şi decompoziţia FB

3.2.1. Formule echivalente

În 2.6 am numit superpoziţie a funcţiilor $f_1, f_2,..., f_n$ funcţia f obţinută prin substituiri şi redenumiri de argumente în aceste funcţii, iar prin formulă înţelegem expresia care descrie această superpoziţie. Vom concretiza noţiunea de formulă pentru funcţiile logice, introducând noţiunea de formulă peste $\Omega = \{f_1, f_2,..., f_m,...\}$, Ω fiind o mulţime de funcţii logice date.

Considerăm *formulă* peste Ω toate expresiile care conţin numai simboluri de variabile, simboluri de funcţii şi paranteze.

Exemplul 3.3. Poate fi considerată formulă expresia: $f=((x_1 \oplus x_3)(x_2 \to x_4))$. Valoarea funcţiei dată de o formulă poate fi calculată, cunoscând valorile argumentelor şi tabelele de adevăr ale funcţiilor logice elementare (v. tab.3.4). Dacă $x_1=1, x_2=1, x_3=0$ *şi* $x_4=0$, avem $x_1 \oplus x_3=1, x_2 \to x_4=0$ şi $f=0$. ◄

Deci, o formulă, pune în corespondenţă fiecărui set de valori ale argumentelor o anumită *valoare de adevăr* şi poate servi, împreună cu tabelele de adevăr, drept metodă de definire şi calculare a valorilor funcţiilor logice. Se mai spune că o formulă reprezintă sau **realizează** o funcţie logică. Calculând valorile *FB* pentru toate 2^n combinaţii ale argumentelor restabilim tabelul de adevăr al acestei funcţii. Însă, spre deosebire de tabelele de adevăr, o funcţie logică poate fi realizată prin mai multe formule. Cu alte cuvinte, dacă între mulţimea tabelelor de adevăr şi mulţimea funcţiilor logice există o corespondenţă biunivocă, alta este situaţia cu formulele: o FB poate fi prezentată printr-o infinitate de formule. De exemplu, funcţia Pierce $f_8(x_1,x_2) = x_1 \uparrow x_2$ mai poate fi realizată prin formulele

$\overline{x_1 + x_2}$ sau $\overline{x_1}\,\overline{x_2}$, iar funcţia Sheffer $f_{14}(x_1,x_2) = x_1|x_2$ prin $\overline{x_1} + \overline{x_2}$ sau $\overline{x_1 x_2}$.

Formulele, care realizează aceeaşi funcţie logică se numesc ***echivalente***. Echivalenţa a două formule se va nota prin simbolul = sau ≡. Metoda generală de stabilire a echivalenţei a două formule constă în construirea tabelelor lor de adevăr şi compararea acestora. Altă metodă, denumită metoda transformărilor echivalente, presupune transformarea uneia dintre formule (sau a ambelor) până se ajunge la o formă evident comună. Un interes aparte îl prezintă procedeele de reprezentare a funcţiilor booleene prin aşa numitele forme normale sau canonice şi forme minimale.

3.2.2. Decompoziţia funcţiilor booleene

Notăm $x^0 = \overline{x}$, $x^1 = x$. Este simplu să ne convingem (de exemplu prin construirea tabelului de adevăr), că pentru un parametru α egal cu 0 sau cu 1, reieşind din notaţia introdusă, avem

$$x^{\alpha}=1 \text{ când } x=\alpha$$

şi
$$x^{\alpha}=0, \text{ dacă } x\neq\alpha \qquad (3.1)$$

Teorema 3.1. Orice funcţie booleană $f(x_1, x_2,..., x_n)$ poate fi pusă sub forma:

$$f(x_1,..., x_m, x_{m+1},..., x_n)= \bigvee_{\alpha_1 \cdots \alpha_m} x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_m^{\alpha_m} f(\alpha_1,..., \alpha_m, x_{m+1},..., x_n) \qquad (3.2)$$

în care $m \leq n$, iar disjuncţiile se vor lua pentru toate 2^m seturi, formate din variabilele $x_1, x_2,..., x_m$.

Relaţia (3.2) se numeşte ***decompoziţia*** funcţiilor booleene pentru variabilele $x_1, x_2,..., x_m$.

Demonstrăm prin introducerea unui set arbitrar de valori $(\sigma_1,..., \sigma_m, \sigma_{m+1},..., \sigma_n)$ ale argumentelor în ambele părţi ale relaţiei (3.2). Deoarece $x^{\alpha}=1$ numai în cazul în care $x=\alpha$, dintre toate cele 2^m conjuncţii ale părţii drepte a relaţiei (3.2) numai una este egală cu 1 (restul sunt 0), cea pentru care $\alpha_1=\sigma_1$, $\alpha_2=\sigma_2$,..., $\alpha_m=\sigma_m$. Obţinem

$$f(\sigma_1, \sigma_2,..., \sigma_n) = \sigma_1^{\alpha_1} \sigma_2^{\alpha_2} \cdots \sigma_m^{\alpha_m} f(\sigma_1, \sigma_2,..., \sigma_m, \sigma_{m+1},..., \sigma_n) = f(\sigma_1, \sigma_2,..., \sigma_n),$$

şi, deoarece setul de valori a fost luat arbitrar, teorema este demonstrată.

Pentru $m=1$ avem decompoziţia FB pentru o singură variabilă (de exemplu x_1):

$$f(x_1, x_2,..., x_n)= \overline{x_1}\, f(0, x_2,..., x_n) \vee x_1 f(1, x_2,..., x_n).$$

Prezintă interes deosebit cazul $m = n$ - decompoziţia FB pentru toate variabilele $x_1, x_2,..., x_n$. Avem:

$$f(x_1, x_2,..., x_n) = \bigvee_{\alpha_1 \cdots \alpha_n} x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n} f(\alpha_1,..., \alpha_n) = \bigvee_{f(\alpha_1 \cdots \alpha_n)=1} x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n}, \qquad (3.3)$$

în care disjuncţiile se iau pentru combinaţiile argumentelor aplicate de f în 1, iar conjuncţiile respective se numesc *elementare, termeni canonici conjunctivi* sau *termeni minimali (mintermi)*.

Definiţie. Numim ***formulă booleană*** orice formulă care în afară de variabile şi paranteze conţine doar funcţiile disjuncţie, conjuncţie şi negaţie.

Teorema 3.2. Orice funcţie logică poate fi prezentată printr-o formulă booleană.

Demonstraţia este imediată şi reiese din teorema 3.1, relaţia (3.3) şi ultima definiţie. Constanta 0 poate fi reprezentată prin formula $x\bar{x}$.

3.2.3. Algebra booleană. Proprietăţile operaţiilor booleene

Algebra $A = \langle P_2, \{\vee, \wedge, \bar{\ }\}\rangle$, suportul căreia este mulţimea tuturor funcţiilor logice, iar în calitate de operaţii sunt luate disjuncţia, conjuncţia şi negaţia, se numeşte algebra booleană a funcţiilor logice. Operaţiile acestei algebre se mai numesc operaţii booleene.

Pentru operaţiile booleene s-a convenit a se considera următoarea **ordine de îndeplinire**: mai întâi vor fi îndeplinite toate operaţiile de negaţie (cea mai înaltă prioritate), apoi conjuncţiile, iar cea mai mică prioritate o are disjuncţia. La necesitate pot fi utilizate parantezele.

Este simplu de stabilit (prin construirea tabelelor de adevăr, de exemplu), că operaţiile booleene posedă proprietăţile:

- asociativitate $x_1(x_2x_3) = (x_1x_2)x_3;$
  $x_1 \vee (x_2 \vee x_3) = (x_1 \vee x_2) \vee x_3;$ (3.4)
- comutativitate $x_1x_2 = x_2x_1;$
  $x_1 \vee x_2 = x_2 \vee x_1;$ (3.5)
- distributivitate $x_1(x_2 \vee x_3) = x_1x_2 \vee x_1x_3;$
  $x_1 \vee (x_2x_3) = (x_1 \vee x_2)(x_1 \vee x_3);$ (3.6)
- idempotenţă $xx = x;$
  $x \vee x = x;$ (3.7)
- principiul involuţiei $\bar{\bar{x}} = x;$ (3.8)
- operaţii cu constante $x\&1=x;\ x\&0=0;$
  $x \vee 1=1;\ x \vee 0=x;\ \bar{0}=1;\quad \bar{1}=0;$ (3.9)

- legile lui de Morgan $\overline{x_1 x_2} = \overline{x_1} + \overline{x_2}$;

  $\overline{x_1 + x_2} = \overline{x_1} . \overline{x_2}$ (3.10)
- contradicţie $x\overline{x} = 0$; (3.11)
- legea terţului exclus $x + \overline{x} = 1$. (3.12)

Operaţiile booleene posedă proprietăţile (3.4) - (3.12) chiar dacă argumentele x_i sunt la rândul lor funcţii sau vor fi obţinute în rezultatul unor calcule. Ca rezultat, aceste proprietăţi se mai numesc relaţii de echivalenţă, iar transformările care pot fi operate cu ajutorul lor se numesc transformări echivalente. Relaţiile de echivalenţă (3.4) - (3.12), împreună cu unele relaţii suplimentare, obţinute din primele, prezintă un interes deosebit în tratarea problemei de echivalenţă a formulelor, datorită eficacităţii mai mari în comparaţie cu tabelele de adevăr.

Mai aducem câteva relaţii de echivalenţă suplimentare (demonstraţiile se bazează pe proprietăţile de mai sus şi se propun ca exerciţiu):

- absorbţie $x \vee xy = x$; $x(x \vee y) = x$;
- alipire $xy \vee x\overline{y} = x$;
- alipire generalizată $xz \vee y\overline{z} \vee xy = xz \vee y\overline{z}$

3.3. Forme canonice

3.3.1. Forma canonică disjunctivă

Decompoziţia (3.3) se numeşte ***formă canonică disjunctivă (FCD)*** sau ***formă normal disjunctivă perfectă (FNDP)*** a funcţiei $f(x_1, x_2,..., x_n)$. *FCD* conţine tot atâtea conjuncţii elementare câte unităţi sunt în tabelul de adevăr al funcţiei date: fiecărui set de valori ale argumentelor pentru care $f = 1$ îi corespunde o conjuncţie în care x_i este negat, dacă $\alpha=0$ şi fără negaţie, dacă $\alpha=1$. Cu alte cuvinte, există o corespondenţă biunivocă între tabelul de adevăr al funcţiei $f(x_1, x_2,..., x_n)$ şi *FCD* a acestei funcţii. Unica funcţie booleană care nu posedă *FCD* este constanta *0*.

Relaţia (3.3) conduce la un algoritm simplu de elaborare a FCD pentru o FB arbitrară:

1°. Se construieşte tabelul de adevăr al funcţiei;

2°. Pentru fiecare combinaţie de valori ale argumentelor aplicate în 1 se scriu termenii canonici conjunctivi în care argumentul x_i este luat ca atare sau negat după cum valoarea lui în combinaţia respectivă este 1 sau 0;

3°. Se reunesc toţi termenii canonici conjunctivi, obţinuţi la pasul precedent, cu operaţia disjuncţie.

Dacă pentru un termen canonic conjunctiv fiecare variabilă va fi înlocuită prin *1* sau *0*, după cum este prezentă în acest TCC (fără negaţie sau negată, respectiv) vom avea reprezentarea binară a unei conjuncţii elementare. De exemplu, reprezentarea binară a conjuncţiei elementare $\overline{x_1}x_2\overline{x_3}\,\overline{x_4}$ este 0100. Reprezentarea binară poate fi transformată în zecimală, adică conjuncţia elementară de mai sus poate fi considerată *4* zecimal. Prin utilizarea acestor notaţii FCD este mult mai compactă. Notând prin Σ disjuncţia unui set de TCC, forma canonică disjunctivă a unei funcţii booleene oarecare poate avea forma $f(x_1,x_2,...,x_n) = \Sigma(0,2,4,...)$.

3.3.1. Forma canonică conjunctivă

Reprezentarea unei *FB* se poate face şi sub o altă formă, numită *forma canonică **conjunctivă***.

Pentru aceasta numim *numărul combinaţiei* numărul *i* ataşat unei combinaţii de valori $(\sigma_1, \sigma_2,..., \sigma_n)$ conform relaţiei:

$$i = \sigma_1 2^{n-1} + \sigma_2 2^{n-2} + ... + \sigma_n 2^0$$

şi introducem funcţia $S_i(x_1, x_2,..., x_n)$ definită astfel:

$$S_i = \begin{cases} 0, \text{ dacă numărul combinaţiei este } i \\ 1, \text{ în caz contrar.} \end{cases} \qquad (3.13)$$

Această funcţie se numeşte *funcţia caracteristică a lui zero* sau *constituentul lui zero.*

Poate fi demonstrată

Teorema 3.3. Orice *FB* poate fi scrisă sub următoarea formă

$$f(x_1, x_2,..., x_n) = \underset{i \in M_0}{\&} S_i \qquad (3.14)$$

unde M_0 este mulţimea combinaţiilor valorilor argumentelor pentru care funcţia $f(x_1, x_2,..., x_n)$ ia valoarea *0*.

Demonstraţie. Se consideră un set arbitrar de valori ale argumentelor $(\sigma_1, \sigma_2,..., \sigma_n)$. Sunt posibile două cazuri distincte: funcţia să aplice acest set de valori a) în *0* şi b) în *1*.

Dacă valoarea funcţiei este 0, atunci în partea dreaptă a relaţiei (3.14) se află funcţia S_{ik} al cărei indice i_k corespunde numărului setului considerat. În acest caz, conform cu (3.13), pentru setul respectiv de valori ale argumentelor S_{ik} va fi egală cu *0*. Conform proprietăţii $x\&0 = 0$ partea dreaptă a relaţiei (3.14) va fi egală cu 0. Dacă însă pentru setul considerat valoarea funcţiei este 1, atunci conform formulării teoremei, printre funcţiile S_{ik} din (3.14) nu va fi nici una pentru care indicele să coincidă cu numărul combinaţiei. În acest caz, conform cu (3.13), toţi membrii conjuncţiei din partea dreaptă a relaţiei (3.14) vor fi egali cu 1, deci partea dreaptă va fi egală cu 1. Deoarece s-a demonstrat că ambele părţi ale relaţiei (3.14) sunt identice pentru un set **arbitrar** de valori ale argumentelor rezultă că este adevărată pentru orice alt set de valori. Teorema este demonstrată.

Pentru a stabili expresiile funcţiilor S_{ik} se consideră expresia booleană

$$x_1^{\alpha_1} \vee x_2^{\alpha_2} \vee \ldots \vee x_n^{\alpha_n}. \qquad (3.15)$$

Conform relaţiei (3.1) funcţia (3.15) este *0* atunci şi numai atunci, când $x_1 \neq \alpha_1$, $x_2 \neq \alpha_2, \ldots, x_n \neq \alpha_n$, fiind *1* pentru toate celelalte cazuri. Aplicând (3.13) pentru funcţia constituentul lui *0*, rezultă:

$$S_i(x_1, x_2, \ldots, x_n) = x_1^{\overline{\alpha_1}} \vee x_2^{\overline{\alpha_2}} \vee \ldots \vee x_n^{\overline{\alpha_n}}$$

cu condiţia ca $i = \alpha_1 2^{n-1} + \alpha_2 2^{n-2} + \ldots + \alpha_n 2^0$.

Deci, orice *FB* poate fi descrisă printr-o expresie de forma

$$f(x_1, x_2, \ldots, x_n) = \&_0 \,(x_1^{\overline{\alpha_1}} \vee x_2^{\overline{\alpha_2}} \vee \ldots \vee x_n^{\overline{\alpha_n}}) \qquad (3.16)$$

unde prin $\&_0$ s-a notat faptul că se consideră conjuncţia termenilor disjunctivi (3.15) pentru care funcţia ***f*** ia valoarea ***0***.

Reprezentarea *FB* sub forma (3.16) se numeşte ***forma canonică conjunctivă (FCC)***, iar termenii (3.15) cu condiţia $x_i \neq \alpha_i$ ***termeni canonici disjunctivi (TCD)***, ***termeni maximali*** sau ***maxtermi***.

Relaţia (3.16) permite stabilirea algoritmului realizării FCC dacă se cunoaşte tabelul de adevăr al funcţiei, care conţine următorii paşi:

1^0. Din tabelul de adevăr al funcţiei se consideră toate combinaţiile pe care funcţia le aplică în 0;

2^0. Se scriu TCD, care corespund acestor combinaţii. În expresia TCD argumentul x_i intră direct sau negat după cum în combinaţia considerată are valoarea 0 sau 1;

3^0. TCD obţinuţi la pasul 2 se reunesc prin semnul conjuncţiei.

Formele canonice sunt unice pentru o funcţie logică complet definită. Este recomandată FCD în cazul în care tabelul de adevăr al funcţiei conţine un număr mai mic de valori 1, decât de valori 0, şi FCC, în caz contrar.

3.4. Alte forme de reprezentare a funcţiilor booleene

În afară de reprezentarea FB cu ajutorul tabelelor de adevăr şi a formelor canonice mai sunt cunoscute alte forme cum ar fi diagramele Karnaugh, schemele logice, diagramele de timp, etc.

3.4.1. Diagrame Karnaugh

Diagramele Karnaugh au fost concepute pentru simplificarea FB şi reprezintă un tablou bidimensional, care pentru o funcţie de n argumente conţine 2^p linii şi 2^q coloane, iar $p+q = n$. Dacă n este par, atunci $p = q$ şi $p = q+1$, în caz contrar. De obicei, pot fi utilizate cu succes pentru $n = 4, 5,$ mai dificil pentru $n = 6$ şi mai mare. Într-un tabel bidimensional Karnaugh titlurile coloanelor şi liniilor sunt formate din combinaţiile de valori posibile dispuse în cod Gray (binar reflectat). Acest cod fiind continuu şi ciclic asigură relaţia de adiacenţă între câmpurile diagramei (numim adiacente două câmpuri dacă titlurile lor diferă printr-un singur rang).

Exemplul 3.4. Pentru FB de 4 argumente cu tabelul de adevăr 3.5 diagrama Karnaugh este prezentată în figura 3.1.◄

Tabelul 3.5. Tabelul de adevăr al unei FB

x_1	x_2	x_3	x_4	$f(x_1,x_2,x_3,x_4)$	x_1	x_2	x_3	x_4	$f(x_1,x_2,x_3,x_4)$
0	0	0	0	0	1	0	0	0	0
0	0	0	1	1	1	0	0	1	0
0	0	1	0	1	1	0	1	0	1
0	0	1	1	0	1	0	1	1	1
0	1	0	0	0	1	1	0	0	0
0	1	0	1	0	1	1	0	1	0
0	1	1	0	1	1	1	1	0	1
0	1	1	1	1	1	1	1	1	1

x_3x_4 \ x_1x_2	00	01	11	10
00	0	0	0	0
01	1	0	0	0
11	0	1	1	1
10	1	1	1	1

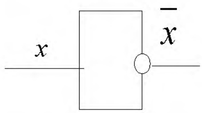
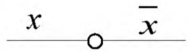
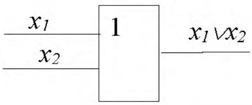
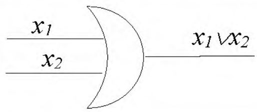
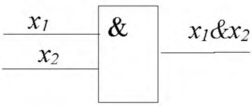
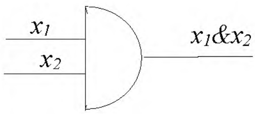
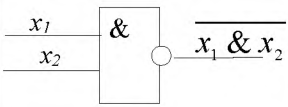
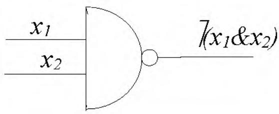
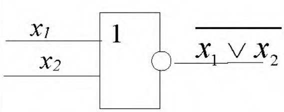
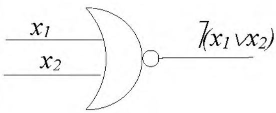
Fig. 3.1. Diagramă Karnaugh

Combinaţiile valorilor argumentelor x_1 şi x_2 sunt dispuse în partea superioară a diagramei, iar cele ale argumentelor x_3 şi x_4 vertical în partea stângă. La intersecţia unei coloane şi a unei linii este câmpul diagramei în care se trece *0* sau *1*, conform valorii funcţiei în tabelul de adevăr.

3.4.2. Circuite logice

Circuitul logic (sau schema logică) este o reprezentare grafică a FB, obţinută prin adoptarea unor semne convenţionale pentru

Tab.3.6. Reprezentarea FB prin scheme logice

Denumirea funcţiei	Reprezentarea grafică	
	Standardele fostei URSS	Standarde internaţionale
Negaţia $f(x) = \bar{x}$		
Disjuncţia $f(x_1,x_2) = x_1 \vee x_2$		
Conjuncţia $f(x_1,x_2) = x_1 \wedge x_2$		
Pierce $f(x_1,x_2) = x_1 \uparrow x_2$		
Sheffer $f(x_1,x_2) = x_1 / x_2$		

operaţiile logice. Simbolurile grafice adoptate constituie o reprezentare a circuitelor logice, care materializează funcţiile logice elementare. **Prin**