Algebrele booleene

Note de curs
Dr.Ing.Mat. ION I. BUCUR

În anul 1854 apare la New York, în editura Dover Publication, lucrarea *An Investigation of the Laws of Thought* a matematicianului şi logicianului englez George Boole. Această lucrare conţine introducerea unei algebre (o structură algebrică) intenţionată să descrie relaţiile logice complexe ale limbajului natural.
Focalizarea acestei lucrări asupra limbajului natural şi relaţiilor logice complexe ale acestuia este deosebit de importantă dacă se priveşte această abordare prin prisma translatării unei descrieri, în limbaj natural, a funcţionării unui circuit (bloc funcţional) oarecare, într-o descriere riguroasă, ne-ambiguă şi echivalentă funcţional acesteia.
În linii mari activitatea de proiectare actuală în domeniul circuitelor digitale este compusă, între altele, dintr-o translatare de acest fel. Termenii au evoluat în complexitate dar ideia este, în principiu, aceeaşi.

După aproape o sută de ani de la apariţia lucrării lui Boole, în 1938, matematicianul Claude Shannon, care lucra pentru Bell Telephone Laboratories, propune în lucrarea *Symbolic Analysis of Relay and Switching Circuits,* o algebră a comutatorilor destinată analizei şi proiectării sistemelor binare discrete.

Algebra introdusă de Shannon este o extindere a algebrei logice introduse de Boole. Blocurile funcţionale fundamentale ale acestei algebre ŞI, SAU şi NU sunt cele mai convenabile implementări ale acestei algebre. Un comutator închis (conduce) are valoarea 1, iar un comutator deschis are valoare 0. Un astfel de comutator poate reprezenta o variabilă a algebrei lui Shannon. Primele două blocuri funcţionale admit două sau mai multe variabile, iar cel de-al treilea bloc funcţional este definit doar pentru o singură variabilă. Definiţiile acestor blocuri funcţionale sunt simple, în termenii introduşi de Shannon. Astfel, oricare bloc fundamental are o descriere succintă a valorii liniei (unice) de ieşire în raport cu liniile (argumentele, variabilele) de intrare:
(a) Blocul ŞI are valoarea 1 dacă şi numai dacă toate variabilele sale (două sau mai multe) au valoarea 1;
(b) Blocul SAU are valoarea 0 dacă şi numai dacă toate variabilele sale (două sau mai multe) au valoarea 0;
(c) Blocul NU are valoarea 1 dacă şi numai dacă variabila sa (unică) are valoarea 0.

În tabelul 1 sunt exemplificate succint aceste trei blocuri funcţionale, primele blocuri (ŞI şi SAU) fiind prezentate, pentru simplitate, doar pentru două variabile.

x	*y*	ŞI(*x*,*y*)	*x*	*y*	SAU(*x*,*y*)	*x*	NU(*x*)
0	0	0	0	0	0	0	1
0	1	0	0	1	1	1	0
1	0	0	1	0	1		
1	1	1	1	1	1		

Tabelul 1. Blocurile funcţionale ŞI, SAU (de două varaibile) şi NU.

Cu toate că, aşa cum se va vedea în continuare, algebra comutatorilor este doar un caz particular al algebrei lui Boole se obişnuieşte să se utilizeze termenul *algebră booleană* pentru întreg aparatul matematic utilizat în sistemele şi circuitele digitale.
Faptul că tehnologic, blocurile funcţionale ale algebrei comutatorilor, sunt construibile fizic constituie raţiunea utilizării acestui aparat matematic.

Exemplul 1. Se consideră următoarea descriere verbală a unui bloc simplu de comandă al uşii unei săli:
- uşa este deschisă atâta timp cât există în imediata sa proximitate (aproximativ 50 cm de-o parte şi de alta a uşii) o persoană şi rămâne deschisă pentru un interval, relativ scurt, de timp *t* chiar şi după plecarea oricărei persoane din zona de proximitate.
Se doreşte o scriere concisă exactă, în termenii algebrei comutatorilor, a acestei descrieri (făcută în termenii limbajului natural) a funcţionării uşii.
Fiecare element relevant din descrierea în limbaj natural va fi notat printr-o variabilă care poate lua doar două valori 0 şi 1.
Uşa este deschisă. Se notează prin variabila U. Câtă vreme uşa este deschisă, U = 1, iar când uşa este închisă U = 0. Închiderea şi deschiderea uşii are loc în urma unei acţionări, printr-un motor electric – spre exemplu.
Există o persoană în zona de proximitate. Determinarea unei persoane în zona de proximitate se poate realiza printr-un senzor cu traductor inductiv, fotoelectric etc. Prezenţa unei persoane în zona de proximitate se notează prin variabila P. Ori de câte ori este o persoana în apropierea uşii P = 1, altfel P = 0.
Temporizatorul este activ pentru durata prestabilită *t*. Se notează prin T. Atunci când temporizatorul este activ T = 1, altfel T = 0. Temporizatorul este activat de îndată ce este sesizată o persoană în zona de proximitate. Activarea temporizatorului se mai numeşte şi startarea acestuia.
Prezenţa unei persoane în zona de proximitate face ca uşa să se deschidă, pe de-o parte şi activează temporizatorul, pe de-altă parte. Apariţia unei persoane, în zona de proximitate, pe durata temporizării are ca efect numai reactivarea temporizării, uşa fiind deja deschisă.
Se poate deduce uşor că translatarea descrierii iniţiale poate fi făcută succint astfel:

U = SAU(P,T).

Sintetic funcţionarea uşii poate fi exprimată, utilizând expresia anterioară cu blocul funcţional SAU. Astfel, uşa este deschisă dacă este sesizată, în apropiere, o persoană sau este activ temporizatorul.
Limbajul natural are, spre deosebire de cel matematic, un *anumit grad de ambiguitate*. Enumerări de forma *„Marcajul poate fi alb şi roşu sau galben"*, nu au întotdeauna o precedenţă bine stabilită a operatorilor fundamentali. Operatorii fundamentali au particularităţi care pot depinde de context, în limbajul natural. Astfel, dacă se consideră fraza *„Colegul meu este imbrăcat cu o jachetă roşie sau cu un sacou verde"*, spre exemplu, aceasta are două dependenţe în raport cu care este adevărată. Este adevărată atunci cand colegul poartă o jachetă roşie sau când poartă un sacou verde dar, ambele situaţii nu pot fi simultan satisfăcute. Deosebirile dintre limbajul natural şi cel

matematic fac din procesul de translatare al acestora, în proiectarea sistemelor şi circuitelor digitale, o activitate ne-trivială.

Modul cel mai uzitat de introducere al algebrei booleene face apel la setul de postulate introdus de Huntington în anul 1904. Primul postulat poate fi considerat ca stabilind sistemul aflat în studiu.

I. *Există o mulţime **K** de obiecte sau elemente, satisfăcând o relaţie de echivalenţă notată prin „=”, îndeplinind principiul substituţiei.*

Prin substituţie se înţelege că dacă $a = b$ atunci a poate fi substituit prin b în orice expresie care conţine elementul a fără să fie afectată validitatea expresiei respective.

IIa. *Este definită o lege de compoziţie „+” astfel încât expresia $a + b$ este în **K**, $\forall\ a, b \in$ **K**.*

IIb. *Este definită o lege de compoziţie „*” astfel încât expresia $a * b$ (abreviat ab) este în **K**, $\forall\ a, b \in$ **K**.*

IIIa. *Există un element $0 \in$ **K**, astfel încât $a + 0 = a$, $\forall\ a \in$ **K**.*

IIIb. *Există un element $1 \in$ **K**, astfel încât $a * 1 = a$, $\forall\ a \in$ **K**.*

IVa. *$a + b = b + a$, comutativitatea legii +.*

IVb. *$a * b = b * a$, comutativitatea legii *.*

Va. $a + (b * c) = (a + b) * (a + c)$, *distributivitatea + faţă de **.

Vb. $a * (b + c) = (a * b) + (a * c)$, *distributivitatea * faţă de +.*

VI. $\forall a \in$ **K**, $\exists\ a' \in$ **K**, *astfel încât*

$a * a' = 0$, şi

$a + a' = 1$.

$\exists\ x,y \in$ **K**, *astfel încât* $x \neq y$.

Se poate remarca faptul că nu s-a precizat nimic în legătură cu numărul sau tipul elementelor care alcătuiesc mulţimea ***K***. Există mai multe mulţimi care satisfac aceste postulate. Câteva dintre acestea vor fi exemplificate în continuare.

Pentru ca un set de postulate să fie valid acesta trebuie să fie consistent. Aceasta revine la faptul că nici unul dintre postulate nu contrazice oricare dintre celelalte postulate din setul considerat. Verificarea consistenţei se poate face prin examinarea fiecărui postulat, pentru a demonstra că nici un postulat nu contravine oricărui grup posibil de postulate, dar abordarea este extrem de laborioasă. Există, însă, o alte cale mult mai simplă, pentru verificarea consistenţei. Pentru aceasta este necesar să se găsească doar un singur exemplu de algebră booleană despre care se ştie, în mod independent, că este consistentă. Dacă o astfel de structură algebrică satisface toate postulatele lui Huntington, atunci postulatele în sine sunt consistente.

Cea mai simplă algebră booleană constă din numai două elemente, notate prin 1 şi 0, definite că satisfac:

$1' = 0,\ 0' = 1$

$1 * 1 = 1 + 1 = 1 + 0 = 0 + 1 = 1,$

şi

$0 + 0 = 0 * 0 = 1 * 0 = 0 * 1 = 0.$

Se remarcă faptul că postulatele I, II, III şi VII sunt satisfăcute prin definiţie. Satisfacerea legilor de comutativitate (IVa şi IVb) este evidentă iar verificarea legilor de distributivitate (Va şi Vb) necesită doar alcătuirea listelor de valori de-o parte şi de alta a ecuaţiilor, pentru toate combinaţiile de valori ale variabilelor *a, b* şi *c.* Postulatul VI este imediat verificabil dând valori (0 şi 1) variabilei *a*.

O altă cerinţă importantă este independenţa postulatelor. Aceasta revine la verificarea faptului că nici unul dintre postulate nu poate fi dedus din celelalte. Postulatele, aşa cum au fost exprimate sunt independente. Această verificare este mult mai complexă, si nu este esenţială pentru această lucrare. Pe de altă parte, nu este necesară abordarea algebrelor booleene printr-un set independent de axiome. Există multe tratări ale subiectului care includ între axiome anumite teoreme, din raţiuni de simplificare a modului de prezentare, nu face obiectul acestei abordări şi poate fi găsită în lucrările de specialitate.
Se poate trage o primă concluzie pe marginea aparatului formal introdus.
O algebră booleană se defineşte, în general, peste o mulţime $\boldsymbol{K} \supseteq \mathbf{B} \equiv \{0,1\}$ înzestrată cu două operaţii, notate aditiv + şi respectiv multiplicativ *, care satisfac legile de comutativitate şi distributivitate. Mulţimea **B** conţine întotdeauna cele două elemente notate 0 şi 1. Acestea sunt *elementele neutre* ale operatorului aditiv şi, respectiv, multiplicativ:

$$a * 1 = a,\ a + 0 = a\ ,\forall a \in \mathbf{B}.$$

În fine, oricare element *a* din **B** are un complement, notat în cele ce urmează prin *a'*. Relaţiile importante dintre un element şi complementul său sunt enunţate astfel:

$$a * a' = 0 \text{ şi } a + a' = 1,\ \forall a \in \mathbf{B}.$$

Este important de reţinut că ambele elemente ale mulţimii **B** nu trebuie privite ca fiind numere, sunt doar notate prin două numere. La fel de bine se pot utiliza alte două simboluri distincte, dar tradiţional se utilizează 0 şi 1. Aceste simboluri corespund, din punct de vedere tehnologic, unor stări distincte ale unor dispozitive fizice care implementează operatorii acestor algebre.
Se poate remarca cu uşurinţă faptul că algebra booleană diferă, ca structură algebrică, de algebra obişnuită prin distributivitatea ambilor operatori, pe de-o parte, şi prin apariţia complementului, pe de-altă parte.
O privire mai atentă asupra postulatelor lui Huntington relevă faptul că anumite postulate sunt grupate în perechi.

Fiecare postulat dintr-o pereche poate fi obţinut din celălalt postulat prin interschimbarea simbolurilor 0 şi 1, ca şi a operatorilor + şi *. Astfel:
$a + 0 = a$, prin interschimbarea amintită devine $a * 1 = a$, iar
$a + (b * c) = (a + b) * (a + c)$ se transformă în $a * (b + c) = (a * b) + (a * c)$.

Oricare teoremă care poate fi demonstrată în algebra booleană are o teoremă duală care este de asemenea adevărată. Cu alte cuvinte, fiecare pas din demonstraţia unei teoreme poate fi înlocuit de dualul acestuia producând demonstraţia teoremei duale.

Teoremele fundamentale ale algebrei booleene

În continuare sunt enunţate principalele teoreme, cele care permit o manipulare convenabilă a algebrei booleene. Unele dintre teoreme sunt numite leme deoarece

acestea au un rol mai limitat de aplicabilitate, furnizând relaţii utilizate în construcţia demonstraţiei unor rezultate cu grad ridicat de generalitate, teoremele.
Atât lemele cât şi teoremele vor fi enunţate dar vor fi demonstrate numaio parte (cele mai semnificative) demonstraţiile celorlalte leme şi teoreme fiind lăsate ca exerciţii.

Lema 1
Elementele 0 şi 1 sunt unice.

Demonstraţie: se presupune că există două elemente 0, notate prin 0_1 şi 0_2. În baza postulatului IIIa, pentru orice elemente w_1 şi w_2 din ***K*** au loc relaţiile:

$$w_1 + 0_1 = w_1 \text{ şi } w_2 + 0_2 = w_2.$$

Acum fie $w_1 = 0_2$ şi $w_2 = 0_1$:

$$0_2 + 0_1 = 0_2 \text{ şi } 0_1 + 0_2 = 0_1.$$

Utilizând comutativitatea operatorului + şi proprietatea de tranzitivitate a egalităţii, rezultă:

$$0_1 = 0_2.$$

Prin dualitate se poate demonstra şi unicitatea elementului 1. □

Lema 2
Pentru orice element $w \in \boldsymbol{K}$ au loc relaţiile:

$$w + w = w \text{ şi}$$
$$w * w = w.$$

Lema 3
Pentru orice element $w \in \boldsymbol{K}$ au loc relaţiile:

$$w + 1 = 1 \text{ şi}$$
$$w * 0 = 0.$$

Lema 4
Elementele 0 şi 1 sunt distincte iar 1' = 0.

Lema 5
Pentru orice elemente w_1 şi w_2 din ***K*** au loc relaţiile:

$$w_1 + w_1w_2 = w_1, \text{ şi}$$
$$w_1(w_1 + w_2) = w_1.$$

Lema 6
Complementul unui element $w \in \boldsymbol{K}$, w', este unic.

Lema 7
Pentru orice element $w \in \boldsymbol{K}$, $(w')' = w$.

Lema 8
Oricare ar fi elementele u, v şi $w \in \boldsymbol{K}$, are loc relaţia:

$$u * ((u + v) + w) = ((u + v) + w) * u.$$

Teorema 1
Oricare ar fi elementele u, v şi $w \in \boldsymbol{K}$, au loc relaţiile:

$$u + (v + w) = (u + v) + w, \text{ şi}$$
$$u * (v * w) = (u * v) * w.$$

Teorema 2
Pentru orice pereche de elemente u şi $v \in \boldsymbol{K}$, se verifică relaţiile:

$$u + u'v = u + v \text{ şi}$$
$$u(u' + v) = uv.$$

Teorema 3

Următoarele două relaţii sunt adevărate oricare ar fi elementele u şi $v \in \boldsymbol{K}$:

$$(u + v)' = u' * v' \text{ şi}$$
$$(u * v)' = u' + v'.$$

Proprietăţile algebrelor booleene

Spre deosebire de postulate, proprietăţile sunt, în fapt, teoreme şi de aceea sunt demonstrabile. Metoda generală de demonstrare a acestor proprietăţi se bazează pe postulatele algebrelor booleene utilizându-se mult inducţia matematică.

(I) Asociativitatea:

$$a + (b + c) = (a + b) + c = a + b + c,$$
$$a * (b* c) = (a * b) * c = a * b * c.$$

(II) Idempotenţa:

$$a + a = a,$$
$$a * a = a.$$

(III)

$$a + 1 = 1,$$
$$a * a = a.$$

(IV) Absorbţia:

$$a + (a * b) = a,$$
$$a * (a + b) = a.$$

(V) Involuţia:

$$(a')' = a$$

(VI) Legile De Morgan:

$$(a + b)' = a' * b',$$
$$(a * b)' = a' + b'.$$

(VII)

$$a + a' * b = a + b,$$
$$a * (a' + b) = a * b.$$

(VIII) Consensus:

$$a * b + a' * c + b * c = a * b + a' * c ,$$
$$(a + b) * (a' + c) * (b + c) = (a + b) * (a' + c) .$$

Exceptând proprietatea (V) Involuţia, se poate remarca o dublă reprezentare a fiecărei proprietăţi (identităţi, în fapt) ale algebrelor booleene. Această remarcă se concretizează în „*Principiul Dualităţii”,* aşa cum este acesta formulat în definiţia următoare.

Principiul Dualităţii

Orice identitate dintr-o algebră booleană se transformă într-o altă identitate dacă au loc următoarele inter-schimbări:

Operatorii + şi *,
Elementele 0 şi 1.

Exemplul 2: Identitatea $a + 1 = 1$, se transformă, prin aplicarea principiului dualităţii, în identitatea: $a * 0 = 0$.

□

O algebră booleană se identifică, tradiţional, prin numele mulţimii suport, ***B***. Există, în acest sens, următoarele exemple clasice:

Algebra comutatorilor,
Algebra submulţimilor unei mulţimi (numită şi algebra claselor),
Algebra booleană aritmetică,
Algebra booleană a funcţiilor booleene.

Algebra comutatorilor este un sistem algebric care constă din mulţimea {0,1}, doi operatori binari, notaţi ŞI (*), respectiv SAU (+), şi un operator unar, notat NU (’), unde:

$0 + 0 = 0$,	$0 * 0 = 0$,	
$0 + 1 = 1$,	$0 * 1 = 0$,	$0' = 1$,
$1 + 0 = 1$,	$1 * 0 = 0$,	$1' = 0$,
$1 + 1 = 1$,	$1 * 1 = 1$.	

Se remarcă faptul că, în conformitate cu definiţia dată, algebra comutatorilor poate fi enunţată astfel: ({0,1}, +, *, 0,1).
Se poate verifica uşor validitatea postulatelor pentru această algebră booleană.
De reţinut că există o proprietate exclusivă a acestei algebre:
$a + b = 1$ dacă şi numai dacă $a = 1$ ori $b = 1$,
$a * b = 0$ dacă şi numai dacă $a = 0$ ori $b = 0$.

Algebra submulţimilor este un sistem algebric construit peste o mulţime nevidă S ($S \neq \emptyset$) numită mulţimea univers, pentru care se consideră toate submulţimile distincte posibile ale acesteia. Dacă notăm prin $|S|$ cardinalitatea mulţimii S, atunci mulţimea tuturor submulţimilor mulţimii S are $2^{|S|}$ elemente.
Această algebră are drept operator aditiv (+) reuniunea ($\cup$), iar intersecţia ($\cap$) este operatorul multiplicativ (*).
Cu alte cuvinte, se poate scrie: (**K**, +, *, 0, 1) = ($2^{|S|}$, $\cup$, $\cap$, $\emptyset$, S), unde s-a notat mulţimea tuturor submulţimilor distincte ale mulţimii S prin $2^{|S|}$ (notaţie tradiţională).
Dacă $S=\{a,b\}$ atunci **K** $=\{\emptyset,\{a\},\{b\},\{a,b\}\}$. Verificarea postulatelor este, de asemenea, simplă. Relativ la această algebră este de reţinut teorema Stone de reprezentare:

Teorema de izomorfism (Stone)
Orice algebră booleană finită este izomorfă algebrei booleene a submulţimilor unei anumite mulţimi finite S.

Algebra booleană aritmetică se construieşte peste numerele naturale.
Astfel se consideră n numere prime distincte şi fie p produsul acestora. Se notează prin D_n mulţimea tuturor divizorilor acestui produs (p). Se folosesc notaţiile aritmetice consacrate, *cmmdc* şi *cmmmc* pentru cel mai mare divizor comun şi, respectiv, cel mai mic multiplu comun.
Se notează prin **1,** numărul întreg **1** şi va fi distinct de elementul 1 boolean evidenţiat în definiţia algebrelor booleene.
Atunci se poate scrie: (**B**, +, *, 0, 1) = (D_n, *cmmmc*, *cmmdc*, **1**, p).
Dacă, spre exemplu, $p = 2 \times 3 \times 5 = 30$, atunci $D_n = \{1,2,3,5,6,10,15,30\}$.

Este interesant de remarcat că atunci când D_n este privită ca fiind alcătuită din produse de divizori primi netriviali, şi sunt utilizate în clar mulţimile acestor divizori, atunci se poate scrie o formulă apropiată de cea realizată pentru algebra submulţimilor:
$D_n=\{\emptyset,\{2\},\{3\},\{5\},\{2,3\},\{2,5\},\{3,5\},\{2,3,5\}\}$,
deoarece **1** nu are nici un divizor netrivial.
Acum apare evident izomorfismul acestei algebre cu algebra submulţimilor unei mulţimi, în cazul de faţă exemplificându-se – pentru raţiuni de simplitate - cu doar trei elemente distincte.

Algebra binară este construită peste mulţimea suport minimală $\mathbf{B} = \{0,1\}$ iar operaţiile binare + şi * sunt disjuncţia, respectiv, conjuncţia, adesea denumite *sumă*, respectiv, *produs* sau, încă, SAU, respectiv ŞI. Spaţiul multidimensional descris de variabilele cu valori binare este notat prin $\mathbf{B}^n$. Acest spaţiu mai este referit prin *cubul* n-dimensional (*hipercub*) din cauza generalizării reprezentării spaţiului $\mathbf{B}^3$ care geometric descrie un cub peste triedrul natural. Un punct din $\mathbf{B}^n$ este reprezentat printr-un vector cu n ranguri binare. Atunci când sunt asociate variabile binare dimensiunilor spaţiului boolean $\mathbf{B}^n$, un punct se poate identifica prin valorile corespunzătoare acestor variabile. Un literal este o valoare, o instanţiere, a unei variabile sau a complementului acesteia. Un produs de n literali reprezintă un punct din spaţiul şi se spune că reprezintă un cub zero-dimensional. Se utilizează curent referirea unui produs de literali prin denumirea generică *cub*.

Algebra funcţiilor booleene de n variabile este construită astfel:
$(F_n(\mathbf{B}), +, *, 0, 1)$, unde:
$F_n(\mathbf{B})$ este mulţimea funcţiilor booleene de n variabile definite peste $\mathbf{B}^n$, cu valori în $\mathbf{B}$, $n \geq 1$.
\+ este adunarea acestor funcţii,
* este multiplicarea acestor funcţii,
0 este funcţia identic nulă, iar
1 este funcţia constantă 1.

Această algebră a funcţiilor scalare joacă un rol deosebit în fundamentele matematice ale analizei şi sintezei dispozitivelor numerice pentru că operatorii şi elementele acesteia se pot implementa prin dispozitive care se pot produce industrial în mod eficient. Studiul proprietăţilor acestei algebre arată modul în care sunt utilizabili operatorii şi elementele acesteia pentru modelarea conceptelor dezvoltate în calculul algebrico-analitic tradiţional.

Funcţiile booleene

O funcţie scalară din $F_n(\mathbf{B})$ *incomplet specificată* este definită peste o partiţie a mulţimii $\mathbf{B}^n$. Valorile din domeniul de definiţie (adesea referite generic prin *puncte*) unde funcţia nu este definită, se numesc valori neprecizate ale funcţiei, sau, condiţii neprecizate (*don't care,* este termenul anglo-saxon). Aceşti vectori din domeniul de definiţie al unei funcţii se referă, în fapt, la valori care nu sunt utilizate şi, prin urmare valorile funcţiei respective nu sunt observate în acele puncte.

Un interes deosebit îl prezintă funcţiile vectoriale definite peste $\mathbf{B}^n$ cu valori în $\mathbf{B}^m$ unde $m > 1$. O astfel de funcţie este considerată echivalentă unui vector de funcţii din $F_n(\mathbf{B})$. Valorile neprecizate ale funcţiilor vectoriale pot fi distincte pentru fiecare dintre cele *m* componente ale respectivei funcţii. Acest aspect se datorează faptului că punctele neprecizate pot fi proprii unor anumite componente ale vectorului de funcţii. Din acest motiv funcţiile vectoriale incomplet specificate sunt considerate ca fiind definite peste $\mathbf{B}^n$, dar cu valori în mulţimea $\{0,1,*\}^m$, unde prin simbolul * s-a notat condiţia de nedefinire, de neprecizare, a funcţiilor scalare. Pentru fiecare componentă scalară a unei funcţii vectoriale se poate defini o partiţie a domeniului de definiţie după valorile luate. Astfel, toate punctele în care o funcţie scalară ia valorile 0, 1 şi * se numesc, tradiţional, 0-set, 1-set, şi respectiv *-set (*off set*, *on set* şi respectiv *dc set, în literatura anglo-saxonă*). Nu există pericolul unei confuzii între notaţia punctelor din codomeniu desemnând valori nedefinite, pe de-o parte, şi operatorul multiplicativ al acestei algebre deoarece tradiţional operatorul multiplicativ este omis utilizându-se simpla juxtapunere (juxtapunerea a două variabile, spre exemplu, este echivalentă produsului acestora).

O generalizare, într-un anumit sens, a noţiunii de funcţie booleană este *relaţia booleană*. O relaţie booleană este definită, în principiu, între spaţii booleene. Astfel, unui punct dintr-un domeniu se asociază mai multe puncte din codomeniu (spre deosebire de funcţii unde se asociază un singur punct din codomeniu). Relaţiile booleene şi optimizarea acestora joacă un rol deosebit în sinteza şi optimizarea circuitelor, reţelelor, multi-nivel.

Pentru funcţiile booleene, în speţă funcţiile de variabile binare cu valori binare, au fost introduse numeroase definiţii şi notaţii în vederea evidenţierii unor aspecte analitice mai importante ale acestora. O parte dintre acestea, cele mai semnificative, vor fi, succint, enunţate în cele ce urmează.
Mulţimea variabilelor unei funcţii este numită *mulţimea suport*, sau pe scurt, *suportul* funcţiei.

Reprezentări ale funcţiilor booleene

Sunt posibile mai multe modalităţi de descriere ale unei funcţii booleene. O caracteristică comună a acestor funcţii, care se va reflecta în toate aceste modalităţi de descriere, este faptul că funcţiile discrete de variabilă discretă au domeniul de definiţie finit şi implicit, astfel de funcţii sunt reprezentabile prin enumerări finite. Adeseori, însă volumul enumerării acestor funcţii poate fi considerabil sau chiar prohibitiv. Modalităţile de reprezentare pot fi clasificate ca fiind *formele tabelare*, *expresiile sau formulele booleene* şi *diagramele de decizii binare*.

Formele tabelare, cronologic fiind primele utilizate, pot fi privite ca fiind alcătuite din enumerări ordonate de perechi de valori constituite din punctul domeniului de definiţie şi valoarea funcţiei în respectivul punct. Ca mod de implementare au suport evident în alcătuirea memoriilor. Punctele sunt vectori binari, la fel şi valorile funcţiei fiind binare scalare sau vectoriale. Au fost utilizate în aplicaţiile de asistenţă automată a proiectării calculatoarelor în primele tentative de acest fel. Odată cu creşterea suportului funcţiilor reprezentarea devine incomodă iar metodele dezvoltate pentru acest mod de reprezentare nu s-au dovedit a fi printre cele mai performante.

Formulele booleene, pe scurt formulele sau expresiile, au avantajul unei dezvoltări analitice de referinţă. Întreg aparatul teoretic utilizează această formă de reprezentare. Există numeroase aplicaţii care, sub o formă sau alta, sunt dezvoltate având la bază obiecte implementate prin structuri echivalente acestei forme de reprezentare. Ca formă de reprezentare are cele mai multe metode dezvoltate şi cu performanţe printre cele mai bune. Dimensiunile reale ale funcţiilor reprezentabile prin formule sunt superioare celor reprezentabile prin formele tabelare. Printre obiectele dezvoltate cu această reprezentare cele mai cunoscute sunt cele corespunzătoare sumelor de produse (produselor de sume) în două sau mai multe nivele.

Diagramele de decizii binare (*Binary Decision Diagrams*, termenul în original, abreviat *BDD*) au fost iniţial introduse de Lee şi apoi reluate de Akers, pentru reprezentarea prin arbori sau grafuri aciclice direcţionate cu rădăcini, a funcţiilor binare de variabile binare, scalare. Aceste reprezentări permit manipulări de funcţii cu complexitate superioară celor reprezentabile prin formule booleene.

Reprezentarea propusă de Akers utilizează pentru o decizie evaluarea unei variabile din suportul funcţiei. Utilizând această structură, ulterior, Bryant a introdus diagramele de decizii ordonate (*Ordered Binary Decision Diagrams*, cu abrevierea *OBDD*) şi a demonstrat existenţa unor algoritmi performanţi de manipulare a acestor forme de reprezentare.
Schimbând maniera de decizie prin evaluarea unei funcţii în locul evaluării unei variabile, Karplus a introdus grafele aciclice direcţionate numite ITE (abrevierea sintagmei *if-then-else*) apreciate ca fiind o generalizare a OBDD. Există funcţii ale căror reprezentări prin ITE sunt mai compacte decât prin OBDD.

Formulele booleene

Există numeroase situaţii în care anumite proprietăţi ale unor dispozitive, reale sau simulate, virtuale, sunt exprimate prin formule construite peste algebrele booleene. O legătură de mare importanţă există între formulele şi funcţiile booleene.

O variabilă booleană este o variabilă care ia valori din mulţimea ***B***. Prin *literali* se înţeleg variabile booleene care pot fi asertate sau complementate (negate). Variabilele sunt cele mai simple exemple de funcţii booleene.

Expresiile construite prin simboluri legate prin operatorii * şi + sunt cele mai simple exemple de formule booleene. Formulele pot fi dezvoltate ierarhic utilizând paranteze. O formulă booleană se defineşte astfel:

Definiţia 2: Dată o algebră booleană B şi n simboluri $x_1, x_2, \ldots, x_n$, atunci mulţimea formulelor booleene peste cele n simboluri este alcătuită din:

(1) Elementele mulţimii B care sunt formule,
(2) Simbolurile $x_1, x_2, \ldots, x_n$ care sunt formule,
(3) Dacă g şi h sunt formule, atunci tot formule sunt şi
(3.1) (g) + (h),
(3.2) (g) * (h) şi
(3.3) (h)'.

Un şir de caractere este o formulă booleană dacă şi numai dacă aceasta se obţine printr-un număr finit de aplicări ale regulilor 1, 2 şi 3.
□

Se poate remarca din definiţia anterioară că numărul formulelor booleene de n variabile este *infinit*.
Este esenţial de văzut modul în care unei formule booleene F, corect definite, i se poate asocia o funcţie de asemenea corect definită.
Dată F, o formulă booleană de n simboluri, atunci funcţia f, de n variabile, corespunzătoare acestei formule se defineşte astfel:

Definiţia 3: Dacă $F = b \in B$, atunci formula reprezintă funcţia constantă definită prin:
$f(x_1, x_2, \ldots, x_n) = b, \forall (x_1, x_2, \ldots, x_n) \in B^n$;
Dacă $F = x_i$, atunci formula reprezintă *proiecţia funcţiei* definită prin:
$f(x_1, x_2, \ldots, x_n) = x_i, \forall (x_1, x_2, \ldots, x_n) \in B^n$;
Dacă formula este de forma $G + H$, ori $G * H$, ori G', atunci funcţia de n variabile corespunzătoare se defineşte după cum urmează:
$(g + h) (x_1, x_2, \ldots, x_n) = g(x_1, x_2, \ldots, x_n) + h(x_1, x_2, \ldots, x_n)$,
$(g * h) (x_1, x_2, \ldots, x_n) = g(x_1, x_2, \ldots, x_n) * h(x_1, x_2, \ldots, x_n)$,
$(g') (x_1, x_2, \ldots, x_n) = (g(x_1, x_2, \ldots, x_n))'$
$(x_1, x_2, \ldots, x_n) \in B^n$.

Numărul de funcţii booleene de n variabile definite peste o algebră booleană finită este *finit*.

Exemplul 2. Se consideră algebra construită peste mulţimea $B = \{0,1, a',a\}$.
Se consideră mulţimea simbolurilor ca fiind $\{x,y\}$, atunci, o formulă booleană cu aceste două simboluri poate fi, spre exemplu:

$$F = a' * x + a * y'.$$

Tradiţional operatorul * se poate omite, fiind subînţeles, utilizându-se simpla juxtapunere a simbolurilor.
Funcţia booleană de două variabile $f: B^2 \to B$**,** corespunzătoare formulei booleene $F = a' * x + a * y'$, va avea domeniul de definiţie:
$B^2 = \{(0,0), (0,1), (0,a'), (0,a), (1,0), (1,1), (1,a'), (1,a), (a',0), (a',1), \ldots\}$
şi poate fi definită punctual (deoarece domeniul de definiţie este finit) evaluând expresia considerată în fiecare punct al domeniului de definiţie:
$f(0,0) = a$, $f(0,1) = 0$, $f(0,a') = a$, $f(0,a) = 0$, $f(1,0) = 1$, ….
□

Teorema de descompunere a unei funcţii booleene (Claude Shannon).

Fie $f: B^n \to B$ o funcţie booleană, atunci are loc identitatea:
$\forall (x_1, x_2, \ldots, x_n) \in B^n$,
$f(x_1, x_2, \ldots, x_n) = x_1' f(0, x_2, \ldots, x_n) + x_1 f(1, x_2, \ldots, x_n)$,
□

Formulele canonice

Formulele canonice de reprezentare ale unei funcţii sunt unice pentru o funcţie dată. Există forme canonice din categoria expresiilor booleene, dar există şi forme canonice

din categoria diagramelor de decizii binare, acestea sunt diagramele de decizii binare ordonate.
Atunci când se compară două forme canonice se ţine seama de specificul respectiv. Astfel, dacă se compară reprezentări canonice OBDD atunci comparaţia va avea loc între doi arbori sau două grafe aciclice orientate cu rădăcini.

Astfel de forme sunt utile atunci când se compară două funcţii în vederea stabilirii identităţii lor. Adeseori în proiectare se aleg anumite formule de reprezentare ale unei funcţii în funcţie de natura tehnologiei utilizate, criteriile de performanţă etc. Pentru evitarea unor erori care pot apărea în procesul de calcul al unei formule de implementare pentru o funcţie dată se procedează, în general, la validarea funcţională a formulei implementate. Există mai multe metode de validare funcţională. Una dintre cele mai simple este calculul formei canonice atât pentru funcţia considerată cât şi pentru formula sa de implementare.

Teorema de reprezentare

O funcţie $f\colon B^n \to B$ este booleană dacă şi numai dacă aceasta poate fi exprimată prin identitatea:

$$f(X) = \sum_{A\in\{0,1\}^n} f(A)X^A$$

În identitatea de mai înainte s-au utilizat notaţiile :
$X = (x_1, x_2, \ldots, x_n)$, $A = (a_1, a_2, \ldots, a_n) \in \{0,1\}^n$, iar
$X^A = x_1^{a_1} x_2^{a_2} \cdots x_n^{an}$.

Prin expresia $x_i^{a_i}$ se înţelege x_i' dacă $a_i = 0$, şi x_i dacă $a_i = 1$. Termenii produs X^A astfel construiţi se numesc mintermi şi sunt fiecare în parte unic asociaţi unui anumit punct din domeniul de definiţie al funcţiei. Prezenţa lor în forma canonică disjunctivă este validată de valorile (nenule) corespunzătoare ale funcţiei.
Demonstraţia acestei teoreme decurge simplu, utilizând sistematic teorema Shannon.

□

Exemplul 3
Se consideră o funcţie de trei variabile (cazul $n = 3$). Aplicând teorema de reprezentare se obţine următoarea formulă canonică:

$$f(x_1, x_2, x_3) = f(0,0,0)x_1'x_2'x_3' + f(0,0,1)x_1'x_2'x_3 + f(0,1,0)x_1'x_2x_3' +$$
$$f(0,1,1)x_1'x_2x_3 + f(1,0,0)x_1 x_2'x_3' + f(1,0,1)x_1 x_2'x_3 + f(1,1,0)x_1 x_2x_3' +$$
$$f(1,1,1)x_1 x_2x_3$$

Termenii produs utilizaţi, mintermii, sunt prezenţi în formula canonică disjunctivă a unei funcţiei oarecare dacă funcţia respectivă are valori nenule în punctele corespunzătoare.
Se remarcă faptul că formula reflectă valorile funcţiei în toate punctele domeniului său de definiţie. Particularizarea sau *personalizarea* formulei canonice este exclusiv dependentă de aceste valori.

□