Capitolul 1

Funcţii booleene

Scopul acestui capitol îl constituie familiarizarea cititorului cu o clasă particulară de funcţii – **clasa funcţiilor booleene**. Aceste funcţii, deşi au o mulţime de definiţie (un **domeniu**) şi o mulţime de valori (un **codomeniu**) aparent simple, au proprietăţi locale şi globale foarte utile. Teoria dezvoltată pentru ele constituie de fapt **baza** conceptuală şi concretă a **semanticii** logicii propoziţionale în sens clasic (şi nu numai). Schimbând semnificaţia simbolurilor 0 şi 1 din **cifră** în **valoare de adevăr** (**fals,** respectiv **adevărat**), a operaţiilor + şi $\overline{\ \ }$, etc., din **adunare** (de exemplu, *adunarea în mulţimea numerelor naturale* sau adunarea *modulo 2*) şi **opus**, în **disjuncţie**, respectiv **negaţie** (ca operaţii cu valori de adevăr)**,** etc., multe rezultate din logică pot fi ulterior deduse printr-o simplă „traducere”. Anumite noţiuni şi proprietăţi specifice funcţiilor booleene nu sunt direct şi neapărat necesare în studiul logicii formale, astfel încât subiectul nu este tratat în detaliu.

Vom introduce - cât mai succint şi la un nivel informal - şi alte noţiuni, notaţii sau rezultate necesare pentru îmbunătăţirea înţelegerii majoritatea de *natură algebrică* ([DID]) sau de *informatică elementară* ([SOR]). În primul rând, chiar din manualele de matematică de liceu sunt bine cunoscute cel puţin două modalităţi de a *prezenta* o mulţime:

- **Prin enumerarea elementelor sale**. **N** = {0, 1, 2, ...} este mulţimea numerelor naturale.

- **Prin specificarea unei proprietăţi caracteristice**. **A** = {x ∈ **R** | | $x^2 + 9x - 8 = 0$}, este mulţimea rădăcinilor reale ale unei ecuaţii polinomiale de gradul al II-lea.

Mai există o modalitate de specificare, care, fără a fi fost tratată în mod explicit, a fost totuşi suficient de des utilizată (în ideea constructivistă, [CAZ2], [RIC]). Aceasta poate fi descrisă pe scurt astfel: **A *este cea mai mică mulţime care conţine elementele ... şi care este închisă la operaţiile ...*** . De exemplu, dacă notăm cu 0 *cel mai mic (primul)* număr natural şi, pentru fiecare n, număr natural, cu s(n) *succesorul său imediat*, mulţimea **N** poate fi definită (în ideea de mai sus) **constructiv** sau **structural** (ea este de fapt o *mulţimebine-ordonată*, [ŢIP]), astfel:

Baza. 0 ∈ **N** (*zero* este număr natural).

Pas constructiv (**structural**). Dacă n ∈ **N**, atunci s(n) ∈ **N** (dacă n este număr natural, atunci *succesorul său imediat* este număr natural).

Nimic altceva nu mai este număr natural.

Prin urmare, **N** este o mulţime care conţine (*iniţial*) elementul numit 0. Se introduc apoi *elemente noi folosind elemente „vechi" (deja existente în mulţime) şi simbolul (operatorial)* s. Procesul continuă *cât timp este posibil* (în cazul de mai sus, el *continuă „la infinit"*). Pentru ca **N** să fie într-adevăr cea mai mică mulţime construită în felul descris, am adăugat, în plus faţă de ***Bază*** şi ***Pasul constructiv***, condiţia **nimic altceva nu mai este număr natural** (în cele ce urmează, ultimul text va fi implicit presupus a fi prezent.

Soluţia adoptată pentru această a treia cale de descriere a unei mulţimi are avantajul de a avea şi o caracteristică de natură *(semi)algoritmică*. Acceptăm astfel ***paradigma imperativă*** propusă de D. Knuth ([KNU]), *Algoritm = Date + Operaţii*. Mai exact, un **algoritm (imperativ)** *reprezintă o secvenţă finită de paşi (instrucţiuni), care descriu operaţii precise asupra unor informaţii (date) iniţiale (de intrare) sau intermediare (de lucru, temporare), în vederea obţinerii unor informaţii (rezultate) finale (de ieşire)*. Paşii ***se execută*** *(operaţiile se efectuează în mod concret) în ordinea scrierii lor în secvenţă*. Un algoritm **calculează o funcţie** sau **rezolvă o problemă** ([CRO], [SOR]). Intuitiv, datele de intrare reprezintă elemente din domeniul de definiţie al funcţiei de calculat (sau *informaţiile iniţiale* din realitatea în care îşi are originea problema pe care vrem să o rezolvăm), iar datele de ieşire sunt elemente din codomeniul funcţiei (respectiv, *soluţiile* problemei). Un algoritm **se termină** *pentru toate intrările admise, prin urmare* ***există întotdeauna un ultim pas****, a cărui execuţie marchează de obicei şi obţinerea rezultatelor de ieşire*. Din motive tehnice, vom lua uneori în considerare şi *algoritmi care nu se termină pentru toate intrările*, pe care-i vom numi **semialgoritmi (proceduri)**. Un (semi)algoritm poate fi descris sub mai multe forme, printre care se numără şi **pseudocodul** (limbaj intermediar între limbajul natural şi un limbaj de programare comercial). Astfel, definiţia constructivă a lui **N** poate deveni, în *limbaj algoritmic* (pseudocod, [IVA]):

Intrare: -.

Ieşire: **N**.

Metodă:

Pas 1: **N** .{0} =:

Pas 2: **Cât_timp** (*este posibil*) **execută**

Pas 3: Alege n ∈ **N**.

Pas 4: **N** =:**N** ∪ {s(n)}.

Sf_Cât_timp.

Desigur că în cele de mai sus 0 putea fi introdus „din exterior”, adică în **Intrare**. Înafara *instrucţiunii de ciclare cu un număr necunoscut de paşi* (**Cât_timp** (<*condiţie*>) **execută** <corp> **Sf_Cât_timp**) şi a celei de *asignare* (<variabilă> **=:** <valoare>), în descrierea algoritmilor imperativi vom mai folosi *selecţia* (**Dacă** (<condiţie>) **atunci** <corp> **altfel** <corp> **Sf_Dacă**) şi *instrucţiunea de ciclare cu un număr cunoscut de paşi* (**Pentru** <variabilă> = <valoare iniţială>, <indice>, <valoare finală> **execută** <corp> **Sf_Pentru**). Câteodată vom întrebuinţa şi varianta **Repetă** <corp> **Până_când** (<condiţie>) **Sf_Repetă** pentru o instrucţiune de ciclare. Presupunând că semnificaţia intuitivă a instrucţiunilor de mai sus este cunoscută de către cititor, vom mai face precizări şi completări pe parcursul lucrării legate de conceptul de algoritm, cum ar fi prezentarea pe scurt a conceptelor de **corectitudine, complexitate, tratabilitate**. Orice manual de introducere în programarea imperativă poate fi folositor celor care nu stăpânesc asemenea noţiuni (de exemplu, [COR], [AHO],

[LUC]). Revenind la definiţiile constructive – care „ascund” şi ideea de **recursivitate** în programare - ele au şi alte avantaje. *Un prim* avantaj este acela că se poate folosi aceeaşi metodă pentru a introduce alte definiţii, care sunt legate de mulţimea respectivă în totalitatea ei. Putem da astfel o definiţie constructivă (recursivă) a adunării numerelor naturale, *pe care o vom folosi de altfel în **Capitolul 5***, după cum urmează.

Baza: $x + 0 = x$, pentru fiecare $x \in \mathbf{N}$ (a aduna 0 la orice număr natural înseamnă a-l lăsa neschimbat).
Pas constructiv: $x + s(y) = s(x + y)$, pentru fiecare $x, y \in \mathbf{N}$ (dacă ştim să calculăm $x + y$ şi cunoaştem succesorul imediat al numărului natural y, atunci ştim să calculăm şi suma $x + s(y)$; mai exact, aceasta coincide cu succesorul imediat al numărului care reprezintă suma $x + y$).

Un al doilea – şi cel mai important – avantaj este posibilitatea folosirii în demonstraţii a **principiului inducţiei structurale**:

*Fie **A** o mulţime definită constructiv, **A’** $\subseteq$ **A** mulţimea **elementelor iniţiale** (definite prin pasul **Baza** al definiţiei) şi $\mathcal{P}$ o afirmaţie care trebuie demonstrată pentru toate elementele lui **A**. Acceptăm că $\mathcal{P}(a)$ este adevărată pentru fiecare $a \in$ **A** dacă şi numai dacă:*

***1. (Baza.)** Arătăm că $\mathcal{P}(a)$ este adevărată pentru fiecare $a \in$ **A’**.*

***2. (Pas inductiv.)** Fie orice $b \in$ **A**, element nou obţinut din elementele deja construite $a_1, a_2, \ldots, a_n$, cu ajutorul operatorului f (vom conveni să* *scriem acest lucru prin b = f(a_1, a_2, ... , a_n), deşi relaţia dintre elementele vechi şi cele noi nu este întotdeauna de natură funcţională) şi presupunem că este adevărată $\mathcal{P}(a_i)$ pentru fiecare $a_i \in \{1, 2, ..., n\}$. Arătăm că este adevărată $\mathcal{P}(b)$.*

Observaţie. Principiul inducţiei matematice (naturale) aşa cum este el cunoscut din matematica de liceu, este un caz particular al principiului inducţiei structurale (după cum se observă imediat). Folosim cuvântul *principiu* în loc de *teoremă* (care trebuie să aibă şi o *demonstraţie* aferentă) deoarece în cele de mai sus doar ***stipulăm*** că formulele $(\forall n)P(n)$ şi respectiv $P(0) \wedge (\forall n)(P(n)$ implică $P(n+1))$ sunt *tare echivalente* (în sensul care va fi precizat în **Capitolul 2**). În cele de mai sus, $P(0)$ poate fi înlocuit şi cu orice $P(k)$, k – număr natural fixat, deoarece şi submulţimea lui **N**, $\{k, k + 1, ... \}$ este *bine-ordonată*; în acest caz, locul lui $(\forall n)$ este luat de $(\forall n \geq k)$). Echivalenţa amintită, deşi *adevărată în anumite situaţii („structuri") particulare, nu este adevărată în sensul general al logicii formale*. Forme particulare ale principiului inducţiei structurale sunt şi **metoda aserţiunilor invariante** ([DIJ]), folosită pentru demonstrarea *corectitudinii algoritmilor* imperativi, sau **metoda inducţiei asupra unei demonstraţii** ([WIN]), folosită pentru demonstrarea unor teoreme de tip „*corectitudine şi completitudine*". ■

Noţiunile de axiomă, teoremă, regulă de inferenţă, demonstraţie, raţionament, sunt utilizate în acest capitol la modul informal/descriptiv, ele urmând a fi precizate în capitolele următoare.

§1. Algebre booleene

Se presupun cunoscute noţiunile şi notaţiile de bază din matematica de liceu. În plus, submulţimea lui **N,** {1, 2, ... ,n} se va nota şi cu [n], iar pentru indicarea unui element al unui produs cartezian (numit şi *tuplu* sau *n-uplu* în cazul în care numărul n de componente este cunoscut) se vor folosi parantezele ascuţite, nu cele rotunde (exceptând cazul în care este vorba de aplicarea unei funcţii unui tuplu).

Notăm cu **B** mulţimea {0, 1} şi cu $FB^{(n)} = \{f \mid f : \mathbf{B}^n \to \mathbf{B}\}$, $\mathbf{B}^n$ reprezentând *produsul cartezian* al lui **B** cu el însuşi, luat de $n \in \mathbf{N}$ ori ($\mathbf{B}^n = \mathbf{B} \times \mathbf{B} \times ... \times \mathbf{B}$). $FB^{(0)}$ va coincide cu **B**, prin convenţie. Vom pune deci:

$$\begin{cases} FB = \bigcup\limits_{n \geq 0} FB^n \\ FB^{(0)} = \mathbf{B} \end{cases}$$

Observaţie. *card(FB^n)* $= 2^{2^n}$. Cardinalul unei mulţimi **A** va mai fi notat, atunci când nu există confuzii şi cu |**A**|. Mai mult, dacă atât domeniul cât şi codomeniul unei funcţii sunt mulţimi finite, funcţia poate fi dată tabelar. O asemenea tabelă de definiţie va fi numită încă de pe acum **tabelă de adevăr**, deşi semnificaţia acestui termen poate crea anumite confuzii de moment. În cazul unei funcţii $f \in FB^{(n)}$ (operaţie n-ară), tabela de definiţie va fi de forma:

x_1	x_2	...	x_n	$f(x_1, x_2, \ldots, x_n)$
0	0		0	f(0, 0, ... , 0)
0	0	...	1	f(0, 0, ... , 1)
...	...	...	...	...
1	1	1	1	f(1, 1, ... , 1)

■

Se mai observă că am folosit o *ordine standard* pe $\mathbf{B}^n$, de unde se poate deriva o ordine standard pentru valorile din codomeniul funcţiei. Acest lucru face posibilă o reprezentare a funcţiilor booleene ca numere în baza 2 şi (desigur) ca numere în baza 10.

Întrebare. *Puteţi justifica egalitatea* card (FB^n) = 2^{2^n} ? ■

Indicăm câteva funcţii importante din FB. După cum am precizat deja, în $FB^{(0)}$ avem doar *constantele* corespunzătoare, elemente ale lui **B** (prin convenţie, acestea sunt *funcţii de 0 variabile*).

• **Pentru n = 1**, cele 4 funcţii de o variabilă (operaţii 1-are sau *unare*) sunt $\mathbf{c_0}$ (funcţia ***indentic 0***), $\mathbf{c_1}$ (funcţia ***identic 1***), $\mathbf{1_B}$ (***identitatea***) şi $\overline{}$ (***negaţia, opusul***), date prin:

x	$\mathbf{c_0}$	$\mathbf{c_1}$	$\mathbf{1_B}$	$\bar{\ }$
0	0	1	0	1
1	0	1	1	0

• **Pentru n = 2**, din totalul celor 16 funcţii de două variabile posibile (operaţii 2-are; *binare*), câteva dintre cele mai importante sunt: + (***suma,*** sau ***adunarea*** *booleană* sau ***disjuncţia***), • (***produsul*** *boolean* sau ***conjuncţia***), ⊕ (***suma modulo 2*** sau ***disjuncţia exclusivă***) şi | (***anticonjuncţia*** sau ***operaţia lui Sheffer***):

x	**y**	**x + y**	**x • y**	**x ⊕ y**	**x | y**
0	0	0	0	0	1
0	1	1	0	1	1
1	0	1	0	1	1
1	1	1	1	0	0

Întrebare. *Câte funcţii sunt în* $FB^{(3)}$ *? Puteţi da vreun exemplu de asemenea funcţie, care să aibă şi o „semnificaţie cunoscută” ?* ■

Întrebare. *Puteţi descoperi singuri metoda „standard” de construcţie a liniilor unui tabel ca cel de mai sus (ordinea standard pe* $\mathbf{B}^n$)*?* ■

Observaţie. Funcţiile binare •, + şi funcţia unară ¯, pot fi privite ca *legi de compoziţie interne* pe mulţimea **B**. Astfel, într-un mod cu totul similar cu cazurile cunoscute ale *grupului*, *inelului* sau *corpului*, tuplul $\mathcal{B}$ = < **B**, •, +, ¯ > formează o *algebră booleană, sau algebră Boole* (după numele *matematicianului G. Boole*, 1815 – 1864). ■

Definiţia 1.1. Se numeşte **algebră booleană** un 4-uplu $\mathcal{M}$, $\mathcal{M}$ = <**M**, ⊥, ∇, ~ >, format din orice mulţime nevidă **M** (*suportul* algebrei) două operaţii binare ⊥, ∇ : **M** × **M** → **M** şi o operaţie unară ~ : **M** → **M**, care satisfac condiţiile (*legile*):

1) x ⊥ y = y ⊥ x	*comutativitate* (a lui ⊥)
2) (x ⊥ y) ⊥ z = x ⊥ (y ⊥ z)	*asociativitate* (a lui ⊥)
3) x ⊥ (y ∇ z) = (x ⊥ y) ∇ (x ⊥ z)	*distributivitate* (a lui ⊥ faţă de ∇)
4) (x ⊥ y) ∇ y = y	*absorbţie*
5) (x ⊥ (~x)) ∇ y = y	*legea contradicţiei*

şi respectiv

1’) x ∇ y = y ∇ x.	*comutativitate* (a lui ∇)
2’) (x ∇ y) ∇ z = x ∇ (y ∇ z)	*asociativitate* (a lui ∇)
3’) x ∇ (y ⊥ z) = (x ∇ y) ⊥ (x ∇ z)	*distributivitate* (a lui ∇ faţă de ⊥)
4’) (x ∇ y) ⊥ y = y	*absorbţie*
5’) (x ∇ (~x)) ⊥ y = y	*legea tautologiei*

■

Legile (numite impropriu şi *axiome*) de mai sus nu reprezintă identităţi, ele trebuind să fie înţelese ca nişte *ecuaţii satisfăcute pentru toate valorile variabilelor* x, y, z, care sunt *nume generice* pentru elemente oarecare din **M**. Fiecare dintre cei doi membri reprezintă de fapt (expresiile unor) funcţii booleene (numărul de argumente fiind dat de numărul de nume de variabile distincte care apar în intreaga ecuaţie). *Egalitatea înseamnă prin urmare egalitatea de funcţii*. Mai mult, vom admite fără demonstraţie, că ***ecuaţiile reprezintă scheme de axiome***, adică legile anterioare – precum şi cele „derivate" care vor urma - sunt adevărate şi dacă înlocuim (textual) orice apariţie a unei funcţii (subexpresii) prin altă functie (subexpresie). O asemenea ***Teoremă de substituţie*** va fi demonstrată în capitolul următor, în contextul logicii formale.

În general, considerând *afirmaţii* (notate generic cu $\mathcal{A}$) *peste o mulţime* ***M*** *(suport al unei algebre booleene),* care depind doar de variabile cu valori în **M** şi folosesc doar operaţiile amintite, afirmaţii care sunt reprezentate fie prin axiome (**Baza** definiţiei structurale), fie obţinute din axiome printr-un anumit *raţionament utilizând reguli de inferenţă* (**Pasul constructiv**: cu ajutorul regulilor se obţin afirmaţii noi, numite şi *teoreme*, din afirmaţii vechi), putem defini **dualele** lor, $\mathcal{A}^\delta$, în felul următor: *$\mathcal{A}^\delta$ se obţine din $\mathcal{A}$ prin înlocuirea* ***simultană*** *(textuală) a tuturor apariţiilor lui $\perp$ cu ∇ şi a tuturor apariţiilor lui ∇ cu $\perp$.*

Putem extinde conceptul şi notaţia anterioară la *obiecte* oarecare (afirmaţii, dar şi elemente din **M**, funcţii peste **M**, texte, etc.). Astfel, în

B, 1 este dualul lui 0 (evident şi reciproc, *relaţia de dualitate fiind o relaţie simetrică*), duala sumei este produsul, duala unei funcţii oarecare se obţine prin dualizarea ***întregii*** *tabele de adevăr*, etc. Într-o algebră booleană oarecare $\mathcal{M}$, **se poate arăta** (demonstraţia formală nefiind esenţială pentru scopul acestei lucrări) că există un (unic) element în **M** (notat **0**) care satisface în plus ecuaţia $x \perp (\tilde{x}) =$ **0,** precum şi un (unic) element $\mathbf{1} \in \mathbf{M}$, care este dualul lui **0**, satisfăcând $x \nabla (\tilde{x}) = \mathbf{1}$ **(0** fiind desigur distinct de **1)**. Mai mult, relaţia de dualitate este şi *idempotentă* (avem $(o^{\delta})^{\delta} = o$, pentru fiecare obiect o), existând şi obiecte *autoduale*, adică obiecte care satisfac $o^{\delta} = o$ (de exemplu, funcţiile $\mathbf{1_B}$, $\overline{}$ şi $f \in FB^{(3)}$, dată prin $f(x, y, z) = x \oplus y \oplus z$, sunt autoduale). Fiecare axiomă i) din **Definiţia 1.1**, $i \in [5]$ are astfel duala sa, notată acolo prin i'). Mai mult, înlocuind într-un raţionament prin care se obţine o teoremă $\mathcal{A}$, orice axiomă cu duala ei, vom găsi un raţionament (dual), prin care se obţine (*deduce*, *demonstrează*) afirmaţia $\mathcal{A}^{\delta}$. Este justificat atunci să adoptăm **principiul dualităţii pentru** $\mathcal{B}$ (care, la o privire atentă, este şi el un caz particular al principiului inducţiei structurale). De fapt, pentru fiecare *text (secvenţă finită de caractere grafice)* se poate afla dualul său, după schema sugerată anterior. Admitem deci fără demonstraţie formală că:

N.B. *O afirmaţie booleană $\mathcal{A}$ este adevărată dacă şi numai dacă duala sa $\mathcal{A}^{\delta}$ este adevărată.*

Întrebare. *Puteţi arăta că funcţia f(x, y, z) = x ⊕ y ⊕ z este autoduală ?* ■

Teorema 1.1. Tuplul $\mathcal{B} = < \mathbf{B}, \bullet, +, \bar{\ } >$ este o algebră booleană şi pentru fiecare $x \in \mathbf{B}$ avem $x \bullet (\bar{x}) = 0$ şi $x + (\bar{x}) = 1$.

Demonstraţie. Conform principiului dualităţii, este suficient să arătăm că sunt adevărate doar axiomele 1) – 5) şi $x \bullet (\bar{x}) = 0$ (în cazul nostru, $\perp$ este înlocuit de • iar ∇ de către +). Vom privi atât membrul stâng cât şi membrul drept al ecuaţiilor ca expresiile unor funcţii şi vom folosi tabelele de adevăr pentru reprezentarea acestora. Datorită simplităţii calculelor, dintre axiome vom arăta doar validitatea lui 4). Avem:

x	**y**	**x • y**	**(x • y) + y**	**y**
0	0	0	0	0
0	1	0	1	1
1	0	0	0	0
1	1	1	1	1

şi respectiv:

x	$\bar{\mathbf{x}}$	$\mathbf{x \bullet (\bar{x})}$
0	1	0
1	0	0

Adevărul axiomei 4) rezultă din primul tabel prin compararea penultimei coloane (care este membrul stâng al ecuaţiei) cu ultima

coloană (membrul drept), linie cu linie. Se observă imediat că acestea coincid, adică funcţiile date de expresiile respective sunt egale *(două funcţii sunt egale dacă au acelaşi domeniu şi codomeniu şi valorile lor coincid pe fiecare valoare a argumentului).* Similar pentru $x \bullet (\bar{x}) = 0$ şi cel de-al doilea tabel, cu observaţia că nu am mai explicitat coloana care reprezintă membrul drept (şi care este de fapt expresia funcţiei $\mathbf{c_0}$).
■

O algebră booleană cunoscută este dată de *mulţimea părţilor (submulţimilor)* unei mulţimi oarecare **V**, notată $2^{\mathbf{V}}$, împreună cu *intersecţia, reuniunea* şi *complementara faţă de* ***V***, $\mathcal{V} = < 2^{\mathbf{V}}, \cap, \cup, C_{\mathbf{V}} >$.

Observaţie. Conceptul de algebră booleană este prezent în matematică prin mai multe definiţii, nu toate echivalente în orice context ([BIR]). Să menţionăm faptul că o definiţie echivalentă cu **Definiţia 1.1** este:

O *algebră booleană* este o *latice* $\mathcal{M}$ = <**M**, ⊥, ∇> care satisface condiţiile suplimentare:

- Există (măcar) un *prim element*, $\mathbf{0} \in \mathbf{M}$, astfel încât $x \nabla \mathbf{0} = x$.
- Există (măcar) un *ultim element*, $\mathbf{1} \in$ M, astfel încât $x \perp \mathbf{1} = x$.
- Operaţia ⊥ este *distributivă* faţă de operaţia ∇.
- Pentru fiecare $x \in \mathbf{M}$, există un element $\bar{x} \in \mathbf{M}$ (numit şi *complementul* lui x), astfel încât $x \nabla \bar{x} = \mathbf{1}$ şi $x \perp \bar{x} = \mathbf{0}$.

O *latice* (şi aici sunt mai multe accepţiuni matematice ale termenului şi câteva definiţii echivalente pentru o aceeaşi accepţiune) este un triplet $\mathcal{M} = \langle \mathbf{M}, \perp, \nabla \rangle$, în care ambele operaţii satisfac proprietăţile de *idempotenţă*, *comutativitate*, *asociativitate* şi *absorbţie*. În plus, în orice latice (deci şi în orice algebră booleană), se poate defini o *relaţie de ordine parţială (relaţie binară, reflexivă, tranzitivă şi antisimetrică)*, prin: $x \leq y$ dacă şi numai dacă $x = x \perp y$ (sau, dual, $y = x \nabla y$). Datorită acestui fapt, o latice se mai defineşte şi ca fiind o *mulţime parţial ordonată (poset)* în care toate submulţimile finite, nevide, admit măcar o *cea mai mică margine superioară (l.u.b.,* ⌡*)* şi o *cea mai mare margine inferioară (g.l. b.,* ⌠*).* ■

§2. Teoreme de reprezentare şi forme normale pentru funcţiile booleene

Într-o algebră booleană (în particular, în $\mathcal{B}$) sunt valabile şi alte teoreme. Ele pot fi demonstrate fie utilizând tabelele de adevăr, fie construind un raţionament, adică pornind de la axiome (şi/sau de la alte teoreme, demonstrate anterior) şi utilizând anumite reguli de inferenţă. Sumarizăm câteva dintre ele în tabelul următor (teoremele sunt notate cu 6) – 13) iar dualele lor respectiv cu 6') – 13'); am neglijat uneori, de exemplu în 13) şi 13'), scrierea lui •).

6) $\overline{\overline{x}} = x$	6') $\overline{\overline{x}} = x$
7) $x \bullet \overline{x} = 0$	7') $x + \overline{x} = 1$
8) $x \bullet x = x$	8') $x + x = x$
9) $x \bullet 0 = 0$	9') $x + 1 = 1$
10) $x \bullet 1 = x$	10') $x + 0 = x$
11) $x_1 \bullet x_2 \bullet \ldots \bullet x_n = 0$ dacă şi numai dacă există $i \in [n]$ astfel încât $x_i = 0$ (oricare ar fi $n \geq 2$ şi oricare ar fi $x_1, x_2, \ldots, x_n \in$ **B**)	11') $x_1 + x_2 + \ldots + x_n = 1$ dacă şi numai dacă există $i \in [n]$ astfel încât $x_i = 1$ (oricare ar fi $n \geq 2$ şi oricare ar fi $x_1, x_2, \ldots, x_n \in$ **B**)
12) $x_1 \bullet x_2 \bullet \ldots \bullet x_n = 1$ dacă şi numai dacă pentru fiecare $i \in [n]$ avem $x_i = 1$ (oricare ar fi $n \geq 2$ şi oricare ar fi $x_1, x_2, \ldots, x_n \in$ **B**)	12') $x_1 + x_2 + \ldots + x_n = 0$ dacă şi numai dacă pentru fiecare $i \in [n]$ avem $x_i = 0$ (oricare ar fi $n \geq 2$ şi oricare ar fi $x_1, x_2, \ldots, x_n \in$ **B**)
13) $\overline{x_1 \cdot x_2 \cdot \ldots \cdot x_n} = \overline{x}_1 + \overline{x}_2 + \ldots + \overline{x}_n$ (oricare ar fi $n \geq 2$ şi oricare ar fi $x_1, x_2, \ldots, x_n \in$ **B**)	13') $\overline{x_1 + x_2 + \ldots + x_n} = \overline{x}_1 \cdot \overline{x}_2 \cdot \ldots \cdot \overline{x}_n$ (oricare ar fi $n \geq 2$ şi oricare ar fi $x_1, x_2, \ldots, x_n \in$ **B**)

Tabelul 1.1.

Din tabel se observă că afirmaţia 6) este autoduală şi acesta putea fi completat cu generalizarea la $n \geq 3$ elemente a asociativităţii, comutativităţii, distributivităţii, precum şi cu alte teoreme, etc. Afirmaţiile 13) şi 13') se mai numesc *legile lui deMorgan*.

Exerciţiul 1.1. Să se demonstreze adevărul afirmaţiilor care urmează folosind atât tabelele de adevăr cât şi raţionamente, implicând axiome (sau alte afirmaţii, demonstrate în prealabil) şi reguli de inferenţă (deducţie, demonstraţie), cunoscute din matematica de liceu (de exemplu, cele legate de faptul că *egalitatea* este o *relaţie de echivalenţă*, adică este *reflexivă, simetrică şi tranzitivă*):

a) 11) din tabelul anterior.

b) $x \bullet (x + y) = x$.

c) $x + x \bullet y = x$.

d) $x + \overline{x} \bullet y = x + y$.

e) $\overline{x} + x \bullet y = \overline{x} + y$.

f) $x \bullet (\overline{x} + y) = x \bullet y$.

g) $\overline{x} \bullet (x + y) = \overline{x} \bullet y$.

Rezolvare. Vom lăsa aplicarea metodei care utilizează tabelele de adevăr pe seama cititorului. De asemenea, vom presupune deja demonstrate celelalte afirmaţii din **Tabelul 1.1**.

a) Procedăm prin inducţie matematică, afirmaţia de demonstrat fiind $(\forall n \in \mathbf{N})(n \geq 2$ implică $P(n))$, unde:

$P(n)$: $(\forall x_1, x_2, ..., x_n \in \mathbf{B})(x_1 \bullet x_2 \bullet \ldots \bullet x_n = 0$ dacă şi numai dacă $(\exists i \in [n])(x_i = 0))$.

Baza. $n = 2$. Se folosesc 9) şi 10) din **Tabelul 1.1**.

Pas inductiv. Să presupunem că pentru (orice) $k \geq 2$ şi oricare elemente $x_1, x_2, ..., x_k \in$ **B** avem:

$x_1 \bullet x_2 \bullet \ldots \bullet x_k = 0$ atunci şi numai atunci când există $i \in [k]$ astfel încât $x_i = 0$.

Să presupunem faptul că este adevărată P(k) şi să arătăm că P(k + 1) este adevărată. Fie orice element din **B**, notat x_{k+1} şi să notăm $y = x_1 \bullet x_2 \bullet \ldots \bullet x_k$. Atunci avem de demonstrat că este adevărată afirmaţia

$y \bullet x_{k+1} = 0$ dacă şi numai dacă există $i \in [k + 1]$ astfel încât $x_i = 0$,

ceea ce este echivalent cu a arăta că:

$y \bullet x_{k+1} = 0$ dacă şi numai dacă există $i \in [k]$ astfel încât $x_i = 0$ sau $x_{k+1} = 0$.

Aplicând acum ipoteza inductivă, rezultă că mai trebuie să arătăm că:

$y \bullet x_{k+1} = 0$ dacă şi numai dacă $y = 0$ sau $x_{k+1} = 0$.

Ultima afirmaţie este însă adevărată din cele deja demonstrate (P(2) este adevărată).

b) $x \bullet (x + y) = x$. Pornim cu membrul stâng şi folosind axioma 3) (distributivitate) găsim $x \bullet (x + y) = x \bullet x + x \bullet y$. Folosim acum 8) din **Tabelul 1.1**, distributivitatea şi faptul că egalitatea este o relaţie tranzitivă, pentru a obţine $x \bullet (x + y) = x \bullet (1 + y)$. Din comutativitate şi 9') (**Tabelul 1.1**), se deduce că $x \bullet (x + y) = x \bullet 1$. Ceea ce trebuia arătat urmează acum imediat, prin aplicarea lui 10) (**Tabelul 1.1**).

c) $x + x \bullet y = x$. Rezultă din ultima parte a demonstraţiei anterioare.

d) $x + \overline{x} \bullet y = x + y$. Pornim cu membrul stâng al egalităţii şi îl înlocuim pe x cu $x + x \bullet y$, ceea ce putem face folosind punctul anterior şi faptul că egalitatea este o relaţie simetrică. Găsim $x + \overline{x} \bullet y = x + x \bullet y + \overline{x} \bullet y$.

Folosind comutativitatea şi distributivitatea, rezultă că trebuie să arătăm $x + y \bullet (x + \overline{x}) = x + y$. Aplicăm acum 7') şi apoi 10) (**Tabelul 1.1**), pentru a obţine ceea ce se cere.

e) $\overline{x} + x \bullet y = \overline{x} + y$. În relaţia precedentă se înlocuiesc toate apariţiile lui x cu $\overline{x}$ şi se foloseşte apoi 6).

f) $x \bullet (\overline{x} + y) = x \bullet y$. Se folosesc - în ordine – distributivitatea, afirmaţia 7) (**Tabelul 1.1**), comutativitatea şi 10') (**Tabelul 1.1**).

g) $\overline{x} \bullet (x + y) = \overline{x} \bullet y$. Din nou, se înlocuiesc simultan toate apariţiile lui x cu $\overline{x}$ în relaţia precedentă şi se aplică 6). ■

Să trecem în revistă şi câteva rezultate importante din teoria generală a funcţiilor booleene, pregătind un suport abstract adecvat pentru capitolele următoare. *O primă parte* dintre enunţuri vor fi reluate pe parcursul lucrării, într-un alt cadru. *O a doua parte* este prezentată mai detaliat în alte cursuri (cum ar fi ***Arhitecturi şi sisteme de operare***). În sfârşit, *a treia parte* necesită cunoştinţe suplimentare (din acest motiv, unele demonstraţii vor fi omise).

O clasă de proprietăţi interesante se referă la o metodă generală de reprezentare „standard" a funcţiilor din FB. Începând cu teorema următoare introducem notaţiile $x^1 = x$ şi $x^0 = \overline{x}$ (***în sensul că „puterea" 1 ataşată unei expresii o lasă neschimbată, iar „puterea" 0 îi „adaugă" o bară***). *Să remarcăm că indicii superiori precedenţi nu se supun principiului dualităţii (de exemplu, nu este adevărat că $(x^1 = x)^\delta$ coincide cu $(x^0 = x)$).*

Teorema 1.2 ([CAZ1], *de descompunere, în sumă de „termeni”*). Pentru fiecare $n \in \mathbf{N^*}$, $f \in FB^{(n)}$ şi fiecare $k \in [n]$, avem:

$$f(x_1, x_2, ..., x_n) = \sum_{\alpha_1, \alpha_2, ..., \alpha_k \in B} x_1^{\alpha_1} \cdot x_2^{\alpha_2} \cdot ... \cdot x_k^{\alpha_k} \cdot f(\alpha_1, \alpha_2, ..., \alpha_k, x_{k+1}, ..., x_n)$$

oricare ar fi $x_1, x_2, ..., x_n \in \mathbf{B}$.

Demonstraţie. Dacă x_i, $\alpha_i \in \mathbf{B}$ atunci, direct din notaţiile de mai sus, rezultă că:

(∗) $\mathbf{(x^0)^\alpha = (x^\alpha)^0}$ ***precum şi*** $\mathbf{x^\alpha = 1}$ **dacă şi numai dacă** $\mathbf{x = \alpha}$.

Folosind 12) (**Tabelul 1.1**), rezultă imediat că, în condiţiile teoremei, $x_1^{\alpha_1} \cdot x_2^{\alpha_2} \cdot ... \cdot x_k^{\alpha_k} = 1$ dacă şi numai dacă $x_i = \alpha_i$ pentru fiecare $i \in [k]$. Fie acum elementele $a_1, a_2, ..., a_n \in \mathbf{B}$, oarecare, fixate. Conform (∗), în suma

$$\sum_{\alpha_1, \alpha_2, ..., \alpha_k \in B} a_1^{\alpha_1} \cdot a_2^{\alpha_2} \cdot ... \cdot a_k^{\alpha_k} \cdot f(\alpha_1, \alpha_2, ..., \alpha_k, a_{k+1}, ..., a_n)$$

unul şi numai unul dintre factorii $a_1^{\alpha_1} \cdot a_2^{\alpha_2} \cdot ... \cdot a_k^{\alpha_k}$ va fi egal cu 1, adică cel pentru care $\alpha_i = a_i$, pentru fiecare $i \in [k]$. Datorită comutativităţii şi legilor 10), 9) şi 10’) (**Tabelul 1.1**), rezultă că suma este egală exact cu $f(a_1, a_2, ..., a_n)$. ■

Este adevărată şi *teorema duală (de descompunere, în produs de „factori”)*, ambele rezultate fiind folosite pentru demonstrarea *existenţei formelor normale* pentru funcţiile booleene. În enunţul teoremei duale, înafara înlocuirii lui + cu • şi a lui Σ cu Π, numele

$\alpha_1, \alpha_2, \dots \alpha_k$ (ca argumente ale lui f) se înlocuiesc cu aceleaşi elemente, dar barate.

Definiţia 1.2. Fie $n \in$ **N*** şi $x_1, x_2, \dots, x_n \in$ **B** variabile (booleene) distincte (putem nota mulţimea acestora cu $X = \{x_1, x_2, \dots, x_n\}$, ideea fiind însă că lucrăm cu o listă, adică cu o colecţie *ordonată* de elemente distincte). Se numeşte ***termen (n-ar, peste X)*** orice produs (uneori, operatorul de produs este omis, sau supradimensionat) $t = x_{i_1}^{\alpha_1} \cdot x_{i_2}^{\alpha_2} \cdot \dots \cdot x_{i_k}^{\alpha_k}$, unde $0 \le k \le n$, $\alpha_1, \alpha_2, \dots, \alpha_k \in$ **B** şi $1 \le i_1 < i_2 < \dots < i_k \le n$. ■

În definiţia precedentă, termenul generat pentru k=0 este 1 (prin convenţie). Pentru k = n obţinem aşa-numiţii *termeni maximali* ***(maxtermeni)***, adică *acei termeni în care fiecare dintre variabilele considerate apare o dată şi numai o dată (barată sau nebarată), în ordinea precizată*.

Observaţie. Între mulţimea termenilor n-ari t (peste X) şi mulţimea n-uplelor peste {0, 1, 2} (aceasta „coincide” de fapt cu mulţimea $\{f \mid f : [n] \to \{0, 1, 2\}\}$) se poate stabili o corespondenţă biunivocă g, dată de $g(t) = <e_1, e_2, \dots, e_n>$, unde, pentru fiecare $i \in [n]$, avem $e_i = 0$ dacă x_i apare barată în t, $e_i = 1$ dacă x_i apare nebarată în t şi $e_i = 2$ în rest (x_i nu apare în t). Mulţimea termenilor n-ari consideraţi va avea atunci 3^n elemente. Raţionând similar pentru cazul maxtermenilor n-ari (în acest caz, nu este posibil ca vreo variabilă considerată să nu apară), rezultă că există 2^n maxtermeni n-ari distincţi (indiferent de numele celor n variabile diferite fixate prin X). ■

Consideraţiile de natură combinatorială sunt practic indispensabile în vederea obţinerii unor rezultate convenabile.

Întrebare. *Puteţi rescrie atât teorema de descompunere cu termeni cât şi teorema de descompunere cu factori pentru cazul n = 2 şi k = 1 ?* ■

Exemplu. Dacă luăm n = 2 şi notăm x_1 cu x şi x_2 cu y, atunci cei $3^2 = 9$ termeni sunt: x, y, $\bar{x}$, $\bar{y}$, x•y, $\bar{x}$•y, x•$\bar{y}$, $\overline{x \cdot y}$, **1**. Cei $2^2 = 4$ maxtermeni sunt: x•y, $\bar{x}$•y, x•$\bar{y}$, $\overline{x \cdot y}$. ■

Întrebare. *Sunt adevărate afirmaţiile făcute în precedenta **Observaţie**? Justificaţi răspunsul.* ■

Întrebare. *Fie X = {x, y, z, t}. Este* $\bar{x}$•z•t *un maxtermen (peste X) ?* ■

Definiţia 1.3. Se numeşte *formă normală disjunctivă (n-ară, n* $\in$ ***N*)***, sau (n-)**FND** pe scurt, orice sumă (finită) de termeni n-ari distincţi. Se numeşte *formă normală disjunctivă perfectă (n-ară, n* $\in$ ***N*)***, sau (n-)**FNDP** pe scurt, orice sumă de maxtermeni n-ari distincţi. ■

Orice **FND** se poate reprezenta şi grafic, ca un *arbore* ([KNU], [LUC]). Datorită comutativităţii adunării putem face abstracţie de ordinea (max)termenilor dintr-o sumă, ***mai exact, considerând oricare două sume care diferă doar prin ordinea termenilor, le vom privi ca fiind identice***. Vor exista astfel $C_{3^n}^{k}$ forme normale disjunctive n-are având k termeni, $0 \le k \le 3^n$ (prin convenţie, pentru k = 0, unica formă care este acceptată şi este şi perfectă, se notează tot cu 0). În consecinţă, numărul total al n-**FND** – urilor va fi:

$$C_{3^n}^{0} + C_{3^n}^{1} + \ldots + C_{3^n}^{k} + \ldots + C_{3^n}^{3^n} = 2^{3^n} .$$

Analog, numărul total al n-**FNDP** – urilor va fi:

$$C_{2^n}^{0} + C_{2^n}^{1} + \ldots + C_{2^n}^{k} + \ldots + C_{2^n}^{2^n} = 2^{2^n} .$$

Teorema 1.3 ([CAZ1]). Orice funcţie booleană se poate „reprezenta” în mod unic ca o **FNDP**.

Demonstraţie. Fie fixate $n \in \mathbf{N^*}$, $f \in FB^{(n)}$ şi $X = \{x_1, x_2, \ldots , x_n\}$. Aplicând **Teorema de descompunere** cu termeni pentru f şi k = n, găsim că f se poate scrie sub forma:

$$f(x_1, x_2, \ldots, x_n) = \sum_{\alpha_1, \alpha_2, \ldots, \alpha_n \in B} x_1^{\alpha_1} \cdot x_2^{\alpha_2} \cdot \ldots \cdot x_n^{\alpha_n} \cdot f(\alpha_1, \alpha_2, \ldots, \alpha_n)$$

oricare ar fi valorile lui $x_1, x_2, \ldots , x_n$ din **B** şi prin urmare avem:

$$f(x_1, x_2, \ldots, x_n) = \sum_{\alpha_1, \alpha_2, \ldots, \alpha_n \in B} x_1^{\alpha_1} \cdot x_2^{\alpha_2} \cdot \ldots \cdot x_n^{\alpha_n} ,$$

oricare ar fi $x_1, x_2, ..., x_n \in \mathbf{B}$, $\alpha_1, \alpha_2, ..., \alpha_n \in \mathbf{B}$, astfel încât $f(\alpha_1, \alpha_2, ..., \alpha_n) = 1$, ceea ce înseamnă exact o reprezentare a lui f ca o n-**FNDP**. Unicitatea reprezentării provine din faptul că mulţimea n-**FNDP**–urilor şi mulţimea $FB^{(n)}$ au acelaşi cardinal (adică număr de elemente). Mai spunem că expresia din membrul drept al reprezentării *este* ***(o) FND(P) pentru f***. ■

Prin dualizare se obţin noţiunile de ***(n-)factor peste X*** (orice sumă de n variabile din X, acestea apărând barate sau nu), ***maxfactor (n-ar, peste X)*** – un (n-)factor în care apar toate variabilele, *formă normală conjunctivă (n-ară)* ((n-)**FNC**, adică orice produs de factori dictincţi), *formă normală conjunctivă (n-ară) perfectă* ((n-)**FNCP**, adică orice produs de maxfactori distincţi). Nu uităm că se aplică asociativitatea generalizată şi comutativitatea, peste tot, atât pentru sumă cât şi pentru produs, astfel încât ***două sume (produse) nu vor fi considerate distincte dacă diferă doar prin ordinea componentelor***. Aplicând principiul dualităţii, rezultă că este adevărată şi duala **Teoremei 1.3**, adică:

Teorema 1.4. Fie orice $n \in \mathbf{N^*}$, orice $f \in FB^{(n)}$ şi oricare nume distincte de variabile $x_1, x_2, ..., x_n$. Atunci f se poate reprezenta în mod unic ca o **FNCP** peste $X = \{x_1, x_2, ..., x_n\}$, sub forma:

$$f(x_1, x_2, ..., x_n) = \prod_{\alpha_1, \alpha_2, ..., \alpha_n \in \mathbf{B}} (x_1^{\alpha_1} + x_2^{\alpha_2} + ... + x_n^{\alpha_n}),$$

oricare ar fi $x_1, x_2, \ldots, x_n \in \mathbf{B}$, $\alpha_1, \alpha_2, \ldots, \alpha_n \in \mathbf{B}$, astfel încât $f(\overline{\alpha}_1, \overline{\alpha}_2, \ldots, \overline{\alpha}_n) = 0$. ■

§3. Clase speciale de funcţii booleene

Deşi rezultatele teoretice anterioare sunt încurajatoare (funcţiile booleene, date tabelar, pot fi reprezentate şi prin *expresii standard*, cum ar fi **FNDP**, **FNCP**; acestea sunt *unice* dacă se folosesc anumite convenţii; se pot *construi „algoritmic”*, conform **Teoremei 1.3** şi **Teoremei 1.4**, etc.), s-ar putea ca din punct de vedere practic să nu fie chiar convenabile. Astfel, ne-am putea pune problema găsirii *celei mai „scurte”* forme normale, funcţie de o anumită *măsură* fixată. Există numeroase „măsuri candidat” pentru o **FND** sau **FNC,** suficient de simple (vom nota orice asemenea formă prin φ): lungimea ca text (număr de caractere grafice conţinute, poate chiar excluzând parantezele); numărul de operatori folosiţi (poate chiar exceptând *negaţia*); numărul de nivele din arborele ataşat; numărul de termeni (factori); numărul de componente „elementare” ale unui termen (factor); **numărul total de *apariţii* ale variabilelor (apariţia unei aceleiaşi variabile pe *poziţii* diferite se numără distinct)**, etc. Considerând ultima măsură (pe care o vom nota cu $\mathsf{n}(\varphi)$), putem numi *formă normală disjunctivă minimală* (**FNDM**) pentru $f \in$ FB, orice **FND** φ' astfel încât:

$$\mathsf{n}(\varphi') = \mathit{min}\ \{\mathsf{n}(\varphi) \mid \varphi \text{ este } \mathbf{FND} \text{ pentru } f\}.$$

Dată o funcţie booleană $f \in$ FB, se poate pune **problema determinării *tuturor* FNDM pentru f**, sau a uneia *standard* (ceea ce este posibil deoarece – reamintim - pentru fiecare număr natural n numărul funcţiilor booleene n-are este 2^{2^n} iar numărul formelor disjunctive n-are este 2^{3^n}). **Problema anterioară este rezolvabilă cu ajutorul algoritmului lui W. Quine** ([CAZ1]). Algoritmul lui Quine intră sub incidenţa principiului dualităţii, astfel încât făcând modificările de rigoare el poate determina şi toate *formele normale conjunctive minimale (**FNCM**)* pentru orice funcţie booleană.

O problemă similară, prin rezolvarea căreia s-ar putea reduce în anumite cazuri timpul de procesare a unor texte (expresii, formule, etc.), este **găsirea unui număr minim de operaţii booleene *convenabile***, cu ajutorul cărora să se „reprezinte” **orice** funcţie booleană.

Definiţia 1.4. Clasa *funcţiilor booleene* ***elementare*** este:

$$\mathbf{E} = \{ i_p^n \mid n \in \mathbf{N}^*, 1 \le p \le n, i_p^n : \mathbf{B}^n \to \mathbf{B},$$

$$i_p^n (x_1, x_2, \ldots , x_p, \ldots , x_n) = x_p\}.$$

Fie $n \in \mathbf{N}^*$, t un număr natural, f, $h_1, h_2, \ldots , h_t \in FB^{(n)}$ şi $g \in FB^{(t)}$. Spunem că *f se obţine din g, $h_1, h_2, \ldots , h_t$ prin **superpoziţie*** dacă pentru fiecare $x = \langle x_1, x_2, \ldots , x_n \rangle$ avem:

$$f(x) = g(h_1(x), h_2(x), \ldots , h_t(x)).$$

Fie $M \subseteq FB$. Se numeşte *M-şir* orice *secvenţă (listă) finită* $f_0, f_1, \ldots , f_r$ de funcţii booleene în care fiecare f_i este fie din $\mathbf{E} \cup M$ fie se obţine prin superpoziţie din alte funcţii, aflate în aceeaşi listă dar înaintea lui f_i. ■

Funcţiile i_p^n se mai numesc şi *proiecţii*. Pentru superpoziţie vom utiliza notaţia f = SUP(g, h_1, h_2, ... , h_t), iar $\overline{\mathrm{M}}$ va denota *mulţimea funcţiilor care apar ca elemente în M-şiruri*. Pentru fiecare M dat, $\overline{\mathrm{M}}$ *va fi practic o mulţime definită constructiv*, în care **E** ∪ M constituie mulţimea funcţiilor *de bază* iar *operatorul de superpoziţie* este singura modalitate de a se obţine *funcţii noi din funcţii vechi*. Prin urmare, $\overline{\mathrm{M}}$ este cea mai mică mulţime care conţine proiecţiile, elementele lui M şi este *închisă la superpoziţie*. Algebric vorbind, se mai spune că $\overline{\mathrm{M}}$ *este* ***închiderea*** *prin superpoziţie a mulţimii* **E** ∪ M. M *se va numi* ***închisă*** *dacă coincide cu închiderea sa*.

Teorema 1.5 ([CAZ1]). O mulţime M ⊆ FB este închisă dacă şi numai dacă conţine funcţiile elementare şi orice superpoziţie de funcţii din M se află în M.

Demonstraţie. Este imediată din definiţii, demonstraţia reprezintând o aplicare directă a principiului inducţiei structurale. Astfel, este suficient să arătăm că, dată M, avem:

Baza. **E** ⊆ M şi M ⊆ M .

Pas inductiv. Pentru fiecare n ∈ N*, fiecare t ∈ N*, fiecare g ∈ $\mathrm{FB}^{(t)}$, fiecare h_1, h_2, ... , h_t ∈ $\mathrm{FB}^{(n)}$, fiecare f ∈ $\mathrm{FB}^{(n)}$, astfel încât f = SUP(g, h_1, h_2, ... , h_t), avem: *Dacă g, h_1, h_2, ... , h_t ∈ M, atunci f ∈ M.* ■

Exemple imediate de mulţimi închise sunt Ø, **E** şi FB. Se poate arăta şi că următoarele mulţimi (infinite) sunt mulţimi închise (a se consulta şi exerciţiile din finalul capitolului, împreună desigur cu rezolvările din **Anexă**):

- **T_0**: mulţimea funcţiilor booleene de oricâte argumente care *păstrează pe 0*, adică satisfac $f(0, 0, \dots, 0) = 0$.
- **T_1**: dual, mulţimea funcţiilor care *păstrează pe 1*, adică satisfac $f(1, 1, \dots, 1) = 1$.
- **Aut**: mulţimea funcţiilor *autoduale* ($f^{\delta} = f$). Să notăm ca o proprietate de caracterizare interesantă pentru această clasă de funcţii şi faptul că $f^{\delta}(x_1, x_2, \dots, x_n) = \overline{f(\overline{x}_1, \overline{x}_2, \dots, \overline{x}_n)}$. Acest lucru se obţine imediat deoarece tabela de definiţie pentru f^{δ} se obţine din tabela pentru f, înlocuind simultan, peste tot, pe 0 cu 1 şi pe 1 cu 0.
- **Mon**: mulţimea funcţiilor *monotone*. Pe **B** putem defini o relaţie de ordine „naturală”: $0 \leq 1$ (putem pune chiar $0 < 1$). Relaţia precedentă poate fi extinsă „pe componente” la orice produs cartezian. Să considerăm două n-uple $\alpha = <\alpha_1, \alpha_2, \dots, \alpha_n>$ şi $\beta = <\beta_1, \beta_2, \dots, \beta_n>$, $\alpha_i, \beta_i \in \mathbf{B}$, $i \in [n]$. Vom spune că $\alpha \leq \beta$ dacă şi numai dacă $\alpha_i \leq \beta_i$ pentru fiecare $i \in [n]$ (desigur că vom avea $\alpha < \beta$ dacă şi numai dacă $\alpha \leq \beta$ şi $\alpha \neq \beta$ adică α şi β diferă prin măcar o componentă). O funcţie $f \in FB^{(n)}$ este monotonă dacă pentru fiecare $\alpha, \beta \in B^n$, din $\alpha \leq \beta$ rezultă $f(\alpha) \leq f(\beta)$.
- **Lin:** mulţimea funcţiilor *liniare*. Se poate arăta mai întâi că tripletul $\mathcal{I} = <\mathbf{B}, \oplus, \bullet>$ este un inel comutativ cu unitatea 1, *izomorf cu inelul claselor de resturi modulo 2 (cele două mulţimi suport şi operaţiile „corespondente” se pot*

„identifica”). Urmează că *orice funcţie booleană se poate reprezenta unic ca un polinom (eventual, de mai multe variabile) cu coeficienţi în* ***I****.* Fie astfel o funcţie booleană f, despre care ştim deja că se poate scrie ca o sumă (booleană) de termeni. Putem acum folosi egalităţile (se pot demonstra uşor, folosind tabelele de adevăr): $\overline{x} = x \oplus 1$ şi $x + y = \overline{\overline{x} \cdot \overline{y}} =$ $= (x \oplus 1) \bullet (y \oplus 1) \oplus 1 = x \bullet y \oplus x \oplus y$, precum şi proprietăţile inelelor, pentru a observa că orice **FND** a lui f devine o sumă *modulo 2* de termeni (care sunt produse de variabile distincte). Numărul acestor produse este 2^n, deci *numărul polinoamelor modulo 2* este 2^{2^n}, acelaşi ca şi numărul funcţiilor booleene n-are (de unde urmează *unicitatea* reprezentării). Spunem că o funcţie $f \in FB^{(n)}$ este liniară dacă reprezentarea sa (unică) sub formă de polinom modulo 2 are aspectul:

$$c_0 \oplus c_1 \bullet x_1 \oplus c_2 \bullet x_2 \oplus \ldots \oplus c_n \bullet x_n,\ c_i \in \mathbf{B},\ i \in [n].$$

Observaţie. Se poate arăta că toate mulţimile anterioare sunt nebanale, adică nu coincid nici cu FB, nici cu **E**, nici cu Ø. ■

Exemplu ([CAZ1]). Funcţia $f(x, y, z) = x \bullet z + \overline{x} \bullet y \bullet \overline{z} + \overline{x} \bullet \overline{y} \bullet \overline{z}$ este liniară deoarece avem succesiv:

$f(x, y, z) =$	*comutativitate, distributivitate*
$= x \bullet z + \overline{x} \bullet \overline{z} \bullet (y + \overline{y}) =$	*ştim că* $y + \overline{y} = 1$
$= x \bullet z + \overline{x} \bullet \overline{z} =$	*folosind* $a + b = a \bullet b \oplus a \oplus b$
$= x \bullet z \bullet \overline{x} \bullet \overline{z} \oplus x \bullet z \oplus \overline{x} \bullet \overline{z} =$	*ştim că* $x \bullet z \bullet \overline{x} \bullet \overline{z} = 0$

$= x \bullet z \oplus \bar{x} \bullet \bar{z} =$ *folosind $\bar{a} = a \oplus 1$*

$= x \bullet z \oplus (x \oplus 1)(z \oplus 1) =$ *distributivitate*

$= x \bullet z \oplus x \bullet z \oplus x \oplus z \oplus 1 =$ *folosind $a \oplus a = 0$*

$= x \oplus z \oplus 1.$ ■

Definiţia 1.5. O mulţime de funcţii booleene este *completă* dacă închiderea sa coincide cu FB. O mulţime completă se numeşte *bază* dacă este *maximală* (adică nici o submulţime proprie a sa nu mai este completă). ■

Observaţie. Se arată relativ simplu că mulţimea {$\mathbf{c_0}$, $\mathbf{c_1}$, f, •}, unde f este dată prin $f(x, y, z) = x \oplus y \oplus z$, este o bază. Mulţimea {•, +, ¯ } este completă (ceea ce se arată direct, folosind definiţiile), dar nu este bază pentru că atât {¯ , •} cât şi {¯ , +} sunt complete (acest lucru rezultă din legile lui deMorgan, care „ne spun” că disjuncţia se exprimă cu ajutorul negaţiei şi conjuncţiei şi dual, conjuncţia se exprimă cu ajutorul negaţiei şi disjuncţiei). Poate surprinzător, {+, •} nu este completă, dar { | } (care nu este inclusă în nici una dintre mulţimile $\mathbf{T_0}$, $\mathbf{T_1}$, **Aut, Mon, Lin**) este completă şi desigur bază (ne bazăm pe egalităţile $\bar{x} = x \mid x$ şi $x \bullet y = (x \mid y) \mid (x \mid y)$). Prin urmare, dacă ne reamintim de unul dintre scopurile enunţate (exprimarea tuturor funcţiilor booleene cu ajutorul unui număr minim de funcţii), mulţimea **{|} ar fi candidatul ideal**. Din păcate, operatorul lui Sheffer („|”) nu este prea „comod” (din punct de vedere al gândirii umane) astfel încât exprimările celorlalte funcţii devin complicate şi greu de reţinut. Este de dorit să se găsească o *cale de mijloc*, prin care să se păstreze într-adevăr cât mai *puţine funcţii într-o bază*, dar acestea să fie şi *uşor de înţeles/manipulat*. ■

Acceptăm fără demonstraţie (**Teorema 1.7** a fost însă deja demonstrată implicit, prin comentariile anterioaree), următoarele rezultate ([CAZ1], această referinţă nefiind sursa primară):

Teorema 1.6 (E. L. Post). O mulţime de funcţii booleene este completă dacă şi numai dacă nu este inclusă în nici una dintre mulţimile $\mathbf{T_0}$, $\mathbf{T_1}$, **Aut, Mon, Lin**. ■

Teorema 1.7. Orice bază conţine cel mult patru funcţii şi există baze compuse din una, două, trei şi patru funcţii. ■

Teorema 1.8. Orice mulţime închisă de funcţii booleene fie este inclusă cel puţin în una dintre mulţimile $\mathbf{T_0}$, $\mathbf{T_1}$, **Aut, Mon, Lin**, fie coincide cu FB. ■

Teorema 1.9. Mulţimile $\mathbf{T_0}$, $\mathbf{T_1}$, **Aut, Mon, Lin** sunt *precomplete* (o mulţime M de funcţii booleene este precompletă dacă pentru orice altă funcţie $f \notin M$, mulţimea $M \cup \{f\}$ este completă). ■

E. L. Post a determinat chiar **toate** mulţimile închise de funcţii booleene încă din anul 1941, cât şi relaţiile de incluziune între ele şi anumite baze. Încă se fac cercetări în legătură cu extinderea rezultatelor de acest tip la mulţimi de cardinal mai mare decât doi (cardinalul lui **B**). Importanţa unor asemenea cercetări constă în speranţa reducerii complexităţii unor algoritmi clasici şi dezvoltării unor teorii similare, implementabile şi care să fie valabile şi pentru logicile neclasice multivaluate).

§4. Recapitulare şi Index

Teoria funcţiilor booleene constituie *suportul pentru semantica logicii clasice*. Înţelegerea problematicii capitolelor următoare va fi astfel uşurată, anumite paragrafe (cum ar fi cel privind formele normale) fiind simple transpuneri într-un alt limbaj ale unor concepte şi rezultate deja întâlnite. Principalele teme abordate au fost:

- Reprezentarea funcţiilor booleene. Funcţii booleene particulare importante.
- Proprietăţile globale ale clasei funcţiilor booleene. Principiul dualităţii. Algebre booleene.
- Forme normale pentru funcţiile booleene. Forme minimale.
- Mulţimi închise şi (pre)complete de funcţii booleene. Baze.

Deşi ne-am bazat în principal pe cunoştinţele de matematică de liceu, am fost nevoiţi să abordăm tangenţial – într-un mod intuitiv, pentru coerenţa materialului – subiecte colaterale cum ar fi cele privind algoritmii (imperativi), mulţimile definite structural (constructiv), sau principiul inducţiei structurale. Folosirea nediscreţionară, în acest moment, de către cititor a unor termeni introduşi doar informal (cum sunt, până acum: axiomă, teoremă, raţionament, demonstraţie, etc.)

poate fi dăunătoare, mărind confuziile care se fac în mod uzual între sintaxă şi semantică sau între limbaj şi metalimbaj.

Indexul (neexhaustiv) este:

Este poate bine să menţionăm faptul că am preferat un **Index** în ordinea paginilor şi nu în ordine alfabetică, tocmai pentru că insistăm asupra necesităţii parcurgerii secvenţiale a materialului.

§5. Exerciţii

1. Completaţi demonstraţia **Teoremei 1.1**, adică arătaţi validitatea legilor 1), 2), 3), 5), 1') – 5') şi $x + (\overline{x}) = 1$, utilizând tabelele de adevăr.
2. Arătaţi că $\mathcal{V} = < 2^{\mathbf{V}}, \cap, \cup, C_{\mathbf{V}}>$ este o algebră booleană, oricare ar fi mulţimea (nevidă) **V**. În plus, demonstraţi că **0** = Ø şi **1** = **V**.
3. Arătaţi adevărul afirmaţiilor rămase nedemonstrate din **Tabelul 1.1**.
4. Justificaţi egalitatea $\mathrm{card}(FB^n) = 2^{2^n}$.
5. Fie mulţimea termenilor n-ari t, construiţi peste mulţimea variabilelor booleene distincte (ordonate) $X = \{x_1, x_2, \ldots , x_n\}$, precum şi mulţimea n-uplelor peste {0, 1, 2}. Arătaţi că există o funcţie bijectivă g, care „identifică" aceste mulţimi, dată prin $g(t) = <e_1, e_2, \ldots , e_n>$, unde, pentru fiecare $i \in [n]$, avem $e_i = 0$ dacă x_i apare barată în t, $e_i = 1$ dacă x_i apare (nebarată) în t şi $e_i = 2$ în rest (adică x_i nu apare în t). De asemenea, arătaţi că există (măcar) o corespondenţă bijectivă între mulţimea n-uplelor peste {0, 1, 2} şi mulţimea de funcţii $\{f \mid f : [n] \to \{0, 1, 2\}\}$. Deduceţi că mulţimea termenilor n-ari

considerați va avea 3^n elemente şi că vor exista 2^n maxtermeni n-ari distincţi.

6. Să se găsească **FNDP** şi **FNPC** pentru funcţia definită prin tabelul:

x	**y**	**z**	**f(x, y, z)**
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

7. Justificaţi faptul că mulţimile $\mathbf{T_0}$, $\mathbf{T_1}$, **Aut, Mon, Lin** sunt infinite. Arătaţi că $\mathbf{T_0}$ este mulţime închisă.
8. Arătaţi că mulţimea M = {•, +, ‾ } este o mulţime completă de funcţii booleene.
9. Arătaţi că funcţia booleană „ +" este o funcţie monotonă.
10. Arătaţi că $\mathcal{I}$ = <**B**, ⊕, •> este un inel comutativ cu unitatea **1** şi că el este izomorf cu $\mathcal{R}$ = <$\mathcal{Z}_2$, ⊕, ⊗>, inelul claselor de resturi *modulo 2*.
11. Arătaţi că funcţia booleană $f(x, y, z) = x\bar{y} + y\bar{z}$ nu este liniară (acolo unde nu există confuzii, operatorul • nu va fi scris explicit).