ARHITECTURA SISTEMELOR DE CALCUL ŞI SISTEME DE OPERARE

LUCRĂRILE DE LABORATOR Nr. 12, 13 şi 14

ELEMENTE DE LOGICĂ NUMERICĂ. REDUCEREA EXPRESIILOR LOGICE.

I. SCOPUL LUCRĂRILOR

Lucrările prezintă câteva elemente de logica propoziţiilor. Scopul lucrării constă în a familiariza studenţii cu modalităţile de reducere ale expresiilor logice pentru funcţii logice în forma normală disjunctivă utilizând legile algebrice pentru expresii logice şi metoda diagramelor Karnaugh de ordin 2, 3 şi 4.

II. NOŢIUNI TEORETICE

1. Elemente de logică numerică

Logica propoziţională este o algebră al cărei obiectiv iniţial este modelarea raţionamentului.

Mai recent această algebră şi-a demonstrat utilitatea ca instrument de concepţie (concepţia circuitelor calculatorului).

O a treia utilizare a logicii constă în a servi ca model de calcul pentru limbajele de programare (**Prolog**).

Logica propoziţională este un model matematic care ne permite să raţionăm asupra naturii adevărate sau false a **expresiilor logice**.

O **propoziţie** este un enunţ care poate lua una din cele două valori de adevăr: **adevărat** sau **fals**. Simbolurile care pot reprezenta o propoziţie se numesc **variabile propoziţionale**.

Expresii logice

O primă mulţime de expresii logice se defineşte recursiv astfel:

1) sunt operanzi atomici:
   - variabilele propoziţionale;
   - constantele logice **true** şi **false**;
2) Orice operand atomic este expresie logică;
3) Dacă E şi F sunt expresii logice atunci **E and F** este expresie logică;
4) Dacă E şi F sunt expresii logice atunci **E or F** este expresie logică;
5) Dacă E este expresie logică atunci **not E** este expresie logică;

Funcţii booleene

"Semnificaţia**"** unei expresii logice poate fi descrisă formal ca o funcţie care dă o valoare **adevărat** sau **fals** pentru expresia întreagă pornind de la valoarea argumentelor, numită **funcţie logică** sau **booleană**.

Tabele de adevăr

O funcţie booleană poate fi reprezentată în practică printr-o tabelă de adevăr ale cărei linii corespund tuturor combinaţiilor de valori de adevăr pentru argumente. Există o coloană pentru fiecare argument şi una pentru valoarea funcţiei.

Figura următoare prezintă tabelele de adevăr pentru operaţiile logice **and**, **or**, **not**, **xor**.

p	q	p **and** q
0	0	0
0	1	0
1	0	0
1	1	1

p	q	p **or** q
0	0	0
0	1	1
1	0	1
1	1	1

p	**not** p
0	1
1	0

p	q	p **xor** q
0	0	1
0	1	0
1	0	0
1	1	1

Tabela de adevăr a unei funcţii cu k argumente posedă 2^k linii. Fiecare linie asignează pentru funcţie valoarea 0 sau 1, deci există 2^{2k} funcţii cu k argumente.

Operatori logici suplimentari

- **implicaţia → "** dacă p este adevărat atunci q este adevărat**"**;
- **echivalenţa** ≡ **"** dacă şi numai dacă**"**;
- operatorul **nonand "** not (p **and** q)**"**, notat p **nand** q;
- operatorul **nonor "** not (p **or** q)**"**, notat p **nor** q.

Figura următoare prezintă tabelele de adevăr pentru **→**, **≡**, **nand**, **nor**.

p	q	p → q		p	q	p ≡ q		p	q	p **nand** q		p	q	p **nor** q
0	0	1		0	0	1		0	0	1		0	0	1
0	1	1		0	1	0		0	1	1		0	1	0
1	0	0		1	0	0		1	0	1		1	0	0
1	1	1		1	1	1		1	1	0		1	1	0

Asociativitatea şi precedenţa operatorilor logici

Operatorii logici **and** şi **or** sunt asociativi şi comutativi şi vor fi grupaţi de la stânga la dreapta. Ceilalţi operatori nu sunt asociativi.

Precedenţa operatorilor logici:

1. **not**;
2. **nand;**
3. **nor;**
4. **and;**
5. **or;**
6. **→;**
7. **≡**.

Funcţii booleene ale expresiilor logice

Vom construi expresii logice pornind de la tabelele de adevăr. Deşi se pot defini o infinitate de expresii logice, de obicei, se va încerca pe cât posibil să se găsească cea mai simplă expresie logică.

Notaţii prescurtate

* **and** se reprezintă prin juxtapunerea operanzilor;
* **or** se reprezintă prin +;
* **not** se reprezintă prin ¬ sau prin bararea variabilei.

Construcţia unei expresii logice din tabela de adevăr

Forma normală disjunctivă este o sumă logică de **mintermi** pentru care funcţia ia valoarea logică adevărat (1). Un **minterm** este un produs logic de literale (variabile propoziţionale) ale unei linii, astfel: dacă p are valoarea 0 în coloana k, se utilizează ¬p, altfel se utilizează p.

Forma normală conjunctivă este un produs logic de **maxtermi** pentru care funcţia ia valoarea 0. Un **maxterm** este o sumă logică de literale ale unei linii, astfel: dacă p ia valoarea 0 se utilizează p, altfel ¬p.

Legi algebrice pentru expresii logice

Legi ale echivalenţei

1. Reflexivitate: p ≡ p;
2. Comutativitate: (p ≡ q) ≡ (q ≡ p);
3. Tranzitivitate: (p ≡ q) and (q ≡ r) → (p ≡ r);
4. Echivalenţa negaţiilor: (p ≡ q) → (¬p ≡ ¬q);

Legi analoage aritmeticii

5. Comutativitate and: $pq \equiv qp$;
6. Asociativitate and: $p(qr) \equiv (pq)r$;
7. Comutativitate or: $(p+q) \equiv (q+p)$;
8. Asociativitate or: $(p+(q+r)) \equiv ((p+q)+r)$;
9. Distributivitate and faţă de or: $p(q+r) \equiv pq+pr$;
10. 1 (true) este identitate pentru and: $(p1) \equiv p$;
11. 0 (false) este identitate pentru or: $(p + 0) \equiv p$;
12. 0 este adsorbant pentru and: p and $0 \equiv 0$;
13. Eliminare duble negaţii: $(\neg\neg p) \equiv p$.

Diferenţe faţă de legile aritmeticii

14. Distributivitate or faţă de and: $(p+qr) \equiv ((p+q)(p+r))$;
15. 1 este adsorbant pentru or: $1 + p \equiv 1$;
16. Idempotenţa operatorului and: $pp \equiv p$;
17. Idempotenţa operatorului or: $p+p \equiv p$;
18. Subsumarea:
    a) $(p+pq) \equiv p$;
    b) $p(p+q) \equiv p$.
19. Eliminarea anumitor negaţii:
    a) $p(\neg p+q) \equiv pq$;
    b) $p+\neg pq \equiv p+q$.
20. Legile lui de Morgan:
    a) $\neg(pq) \equiv \neg p+\neg q$;
    b) $\neg(p+q) \equiv \neg p\neg q$;
    c) $\neg(p_1p_2...p_n) \equiv \neg p_1+\neg p_2+...+ \neg p_n$;
    d) $\neg(p_1+p_2+...+p_n) \equiv \neg p_1\neg p_2...\neg p_n$.

Legi ale implicaţiei

21. $((p\to q)$ and $(q\to p)) \equiv (p \equiv q)$;
22. $(p \equiv q) \to (p\to q)$;
23. $(p\to q)$ and $(q\to r) \to (p\to r)$;
24. $(p\to q) \equiv (\neg p+q)$.

Tautologii şi metode de demonstraţie

25. Legea terţului exclus: $(p+\neg p) \equiv 1$;
26. Analiza de caz: $(p\to q)$ and $(\neg p\to q) \equiv q$;
27. Contrara reciprocei: $(p\to q) \equiv (\neg q\to\neg p)$;
28. Reducere la absurd: $(\neg p\to 0) \equiv p$;
29. Demonstraţie prin reducere: $(p \equiv 1) \equiv p$;

Exemple:

1. Funcţii de o variabilă a:

a	z_0	z_1	z_2	z_3
0	0	0	1	1
1	0	1	0	1

$z_0 = 0$ constantă;
$z_1 = a$ identitate;
$z_2 = \neg a$ negaţie;
$z_3 = 1$ constantă;

2) Funcţii logice de 2 variabile a şi b:

00	01	10	11	ab
0	0	0	0	F0 = 0
0	0	0	1	F1 = ab
0	0	1	0	F2 = a¬b
0	0	1	1	F3 = a
0	1	0	0	F4 = ¬ab
0	1	0	1	F5 = b
0	1	1	0	F6 = a⊕b
0	1	1	1	F7 = a+b
1	0	0	0	F8 = ¬ (a+b) = ¬a¬b
1	0	0	1	F9 = ¬(a⊕b)
1	0	1	0	F10 = ¬b
1	0	1	1	F11 = a+¬b
1	1	0	0	F12 = ¬a
1	1	0	1	F13 = ¬a+b
1	1	1	0	F14 = ¬(ab) = ¬a+¬b
1	1	1	1	F15 = 1

Tabelele Karnaugh (TK)

Tabelele sau diagramele lui Karnaugh permit simplificarea funcţiilor logice. Metoda se bazează pe inspectarea vizuală a tabelelor judicios construite (metoda este utilă cu un număr de variabile ≤ 6).

TK se poate considera ca o transformare a tabelei de adevăr.

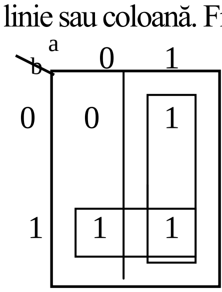
TK cu 2 variabile

Cele 4 căsuţe ale TK corespund celor 4 linii ale tabelei de adevăr:

- fiecare variabilă logică completează o linie sau o coloană;
- un produs de 2 variabile completează o căsuţă;

Pentru a completa TK pornind de la tabela de adevăr se atribuie valoarea 1 căsuţelor corespunzătoare stărilor din intrare în care funcţia are valoarea 1. Metoda de simplificare constă din a încadra mulţimea căsuţelor ocupate, adiacente pe aceeaşi linie sau coloană. Figura următoare prezintă o TK cu 2 variabile.

a	b	z
0	0	0
0	1	1
1	0	1
1	1	1



În urma reducerii avem vom obţine z = a+b.

TK cu 3 variabile

Tabela de adevăr a funcţiei logice de 3 variabile se transformă într-o TK cu două dimensiuni, grupând două variabile pe linie sau coloană. Trecerea de la o linie (coloană) la alta diferă printr-o singură variabilă (se consideră tabela ca înfăşurătoarea unui cilindru.

Pentru construirea TK cu 3 variabile:

- fiecare variabilă completează un bloc de 4 căsuţe;
- un produs logic de două variabile completează un bloc de 2 căsuţe;

- un produs logic de 3 variabile completează o căsuţă.

Exemplu: z (a, b, c) = ¬a¬b¬c+a¬bc+a¬b¬c+abc;

b \ ac	00	01	11	10
0	1		1	1
1				

În urma reducerii se obţine z = ac+¬b¬c.

TK cu 4 variabile

Se construieşte TK, precum înfăşurătoarea unui cilindru, atât orizontal cât şi vertical, astfel:

- fiecare variabilă completează un bloc de 8 căsuţe;
- un produs logic de 2 variabile completează un bloc de 4 căsuţe;
- un produs logic de 3 variabile completează un bloc de 2 căsuţe;
- un produs logic de 4 variabile completează o căsuţă.

Exemplu:

z(a,b,c,d)=¬a¬b¬c¬d+¬a¬b¬cd+¬a¬bcd+¬ab¬cd+ab¬cd+
a¬b¬cd+a¬b¬c¬d+¬abcd+abcd+a¬bcd+¬abc¬d.

cd \ ab	00	01	11	10
00	1			1
01	1	1	1	1
11	1	1	1	1
10		1		

Expresia simplificată este z = d+¬b¬c+¬abc.

În general, metoda de simplificare a unei funcţii de 4 variabile prin TK este următoarea:

- încadrarea căsuţelor cu 1 care nu sunt adiacente altora cu 1 şi deci nu pot forma blocuri de 2 căsuţe;
- încadrarea căsuţelor care pot forma grupe de 2 căsuţe dar nu pot forma blocuri de 4 căsuţe;
- încadrarea căsuţelor care pot forma grupe de 4 căsuţe dar nu pot forma blocuri de 8 căsuţe;
- încadrarea grupelor de 8 căsuţe adiacente;

Pentru reducerea funcţiilor logice cu mai mult de 4 variabile trebuie create mai multe tabele Karnaugh.

III. MODUL DE LUCRU

Se realizează aplicaţii care reduc expresiile în forma normală disjunctivă a funcţiilor logice fie prin aplicarea axiomelor algebrei booleene fie prin utilizarea diagramelor Karnaugh.

IV. CONŢINUTUL REFERATULUI

1. Sumarul noţiunilor întâlnite.
2. Rezolvaţi următoarele probleme:

    2.1 Fie un sumator binar, dispozitiv care efectuează suma S_i a 2 biţi A_i şi B_i deordin i şi a unei cifre de transport C_{i-1} de ordin i-1 imediat inferior.

    a) Daţi tabela de adevăr pentru funcţiile S_i şi C_i;

    b) Deduceţi şI simplificaţi expresiile booleene pentru variabilele de ieşire S_i şi C_i.

    2.2 Figura următoare prezintă două funcţii booleene a şi b în funcţie de variabilele p, q şi r. Scrieţi expresiile de tip formă normală disjunctivă şi realizaţi reducerea:

p	q	r	a	b
0	0	0	1	1
0	0	1	0	1
0	1	0	0	1
0	1	1	0	0
1	0	0	1	0
1	0	1	1	0
1	1	0	1	0
1	1	1	1	0

    2.3 Să se construiască tabela Karnaugh pentru funcţia de două variabile implică **“$\rightarrow$”.**

    2.4 Să se găsească tabela de adevăr şi să se reducă expresia formei normale disjunctive prin metoda diagramelor Karnaugh pentru funcţia NAND (p, q, r) = $\neg$(pqr).

    2.5 Să se construiască tabela de adevăr şi să se reducă expresia formei normale disjunctive pentru funcţia de patru variabile logice: a, b, c, d, care este 1 dacă configuraţia binară abcd conţine cel mult un bit egal cu 1 şi zero în rest.

    2.6 Să se construiască tabela de adevăr şi să se reducă expresia formei normale disjunctive pentru funcţia de patru variabile logice: a, b, c, d, care este 1 dacă una, două sau trei dintre variabile sunt 1 şi zero dacă nici-una sau toate sunt 1.

    2.7 Să se construiască tabela de adevăr şi să se reducă expresia formei normale disjunctive pentru funcţia de patru variabile logice: a, b, c, d, care este 1 dacă cel mult două dintre variabile sunt 1 şi zero dacă trei sau patru sunt 1.

    2.8 Să se construiască tabela de adevăr şi să se reducă expresia formei normale disjunctive pentru funcţia de patru variabile logice: a, b, c, d, care este 1 dacă una, trei sau patru dintre variabile sunt 1 şi zero dacă zero sau două sunt 1.

    2.9 Să se construiască tabela de adevăr şi să se reducă expresia formei normale disjunctive pentru funcţia de patru variabile logice: a, b, c, d, care este 1 dacă configuraţia binară abcd văzută ca un număr binar este o valoare mai mică decât 10 şi zero în rest.

    2.10 Să se găsească tabela de adevăr şi să se reducă expresia formei normale disjunctive prin metoda diagramelor Karnaugh pentru funcţia de patru variabile, pqr $\rightarrow$ s.

3. Observaţii şi concluzii personale.