

STUDII ȘI CERCETĂRI MATEMATICE

EXTRAS

9

TOMUL 17
1965

EDITURA ACADEMIEI REPUBLICII SOCIALISTE ROMANIA

O PROPRIETATE A SCHEMELOR MATRICIALE DE ALGORITMI ¹⁾

DE

L. NEGRESCU și I. PĂVĂLOIU
(Cluj)

În lucrare se dă o condiție necesară și suficientă pe care trebuie să o satisfacă o schemă logică de algoritmi pentru ca schema matricială de algoritmi obținută din ea să fie astfel încît dacă α_{ij} sînt elementele schemei matriciale, atunci ele să satisfacă condiția :

$$\bigvee_{i=1}^{p+1} \alpha_{ij} \equiv 1, \quad j = \overline{0, p}.$$

Rezultatul găsit se bazează pe unele rezultate obținute anterior în lucrarea [2].

În lucrarea [2] am expus un procedeu care permitea ca, fiind dată o schemă logică de algoritmi să se construiască cu ajutorul procedurii dat de noi schema logică matricială echivalentă cu prima. De asemenea, am precizat noțiunea de schemă logică de algoritmi și am dat definiția ei. În această notă vom reține pentru schemele logice de algoritmi și pentru schemele logice matriciale aceleași definiții. La început prin cîteva exemple vom încerca să punem o problemă la care vom răspunde mai tîrziu.

Exemplul 1. Fie dată schema logică de algoritmi [2]

$$(1) \quad A_0(\alpha_1, 3) (\alpha_2, 5) A_3(\alpha_4, 1) (\alpha_5, 3) A_6 A_7.$$

Aplicînd procedeuul P [2] schemei (1), obținem următoarea schemă matricială :

$$(2) \quad \begin{array}{c|ccc} & A_3 & A_6 & A_7 \\ \hline A_0 & \alpha_1 \cdot \alpha_2 \vee \alpha_1 \bar{\alpha}_2 \bar{\alpha}_5 \vee \alpha_1 & \alpha_1 \cdot \bar{\alpha}_2 \cdot \alpha_5 & 0 \\ A_3 & \bar{\alpha}_4 \cdot \alpha_1 \cdot \alpha_2 \vee \alpha_4 \bar{\alpha}_5 \vee \bar{\alpha}_4 \alpha_1 \vee \bar{\alpha}_4 \alpha_1 \cdot \alpha_2 \bar{\alpha}_5 & \alpha_4 \alpha_5 \vee \bar{\alpha}_4 \alpha_1 \bar{\alpha}_2 \alpha_5 & 0 \\ A_6 & 0 & 0 & 1 \end{array}$$

¹⁾ Lucrare prezentată la Sesiunea științifică a Academiei R.P.R., Filiala Cluj, din 11 — 12 decembrie 1964.

Să notăm cu β_{ij} elementele matricei (2); atunci vom arăta că :

$$\bigvee_{i=1}^3 \beta_{ij} \equiv 1 \text{ pentru } i = 1, 2, 3 \text{ și } \beta_{is} \& \beta_{it} \equiv 0 \text{ pentru } s \neq t, i = 1, 2, 3.$$

În adevăr, pentru prima linie avem :

$$(\alpha_1 \alpha_2 \vee \alpha_1 \bar{\alpha}_2 \bar{\alpha}_5 \vee \bar{\alpha}_1) \vee (\alpha_1 \bar{\alpha}_2 \bar{\alpha}_5) \vee 0 = \alpha_1 (\alpha_2 \vee \bar{\alpha}_2 (\alpha_5 \vee \bar{\alpha}_5)) \vee \bar{\alpha}_1 \equiv 1$$

dacă ținem seamă că $\alpha \vee \bar{\alpha} \equiv 1$.

Pentru linia a doua a matricei (2) avem :

$$\begin{aligned} (\bar{\alpha}_4 \alpha_1 \alpha_2 \vee \alpha_4 \bar{\alpha}_5 \vee \bar{\alpha}_4 \bar{\alpha}_1 \vee \bar{\alpha}_4 \alpha_1 \bar{\alpha}_2 \bar{\alpha}_5) \vee (\alpha_4 \alpha_5 \vee \bar{\alpha}_4 \alpha_1 \bar{\alpha}_2 \alpha_5) \vee 0 = \\ = \bar{\alpha}_4 (\alpha_1 \alpha_2 \vee \bar{\alpha}_1 \vee \alpha_1 \bar{\alpha}_2 \bar{\alpha}_5 \vee \alpha_1 \bar{\alpha}_2 \alpha_5) \vee \alpha_4 \bar{\alpha}_5 \vee \alpha_4 \alpha_5 = \\ = \bar{\alpha}_4 (\alpha_1 (\alpha_2 \vee \bar{\alpha}_2 (\alpha_5 \vee \bar{\alpha}_5)) \vee \bar{\alpha}_1) \vee \alpha_4 (\alpha_5 \vee \bar{\alpha}_5) \equiv 1. \end{aligned}$$

Pentru linia a treia afirmația este evidentă.

Cea de-a doua afirmație este demonstrată în general în [2].

Exemplul 2. Fie schema logică de algoritmi

$$(3) \quad A_0(\alpha_1, 3) (\alpha_2, 4) A_3(\alpha_4, 1) A_5 A_6.$$

Aplicând același procedeu ca în exemplul 1, obținem următoarea schemă matricială

$$(4) \quad \begin{array}{c|ccc} & A_3 & A_5 & A_6 \\ \hline A_0 & \alpha_1 \alpha_2 \vee \bar{\alpha}_1 \vee \alpha_1 \bar{\alpha}_2 \bar{\alpha}_4 \bar{\alpha}_1 & \alpha_1 \bar{\alpha}_2 \bar{\alpha}_4 & 0 \\ A_3 & \bar{\alpha}_4 \alpha_1 \alpha_2 \vee \bar{\alpha}_4 \bar{\alpha}_1 & \alpha_4 & 0 \\ A_5 & 0 & 0 & 1 \end{array}$$

Pentru această schemă matricială calculăm disjuncția elementelor primei linii și avem

$$\begin{aligned} (\alpha_1 \alpha_2 \vee \bar{\alpha}_1 \vee \alpha_1 \bar{\alpha}_2 \bar{\alpha}_4 \bar{\alpha}_1) \vee (\alpha_1 \bar{\alpha}_2 \bar{\alpha}_4) \vee 0 = \alpha_1 \alpha_2 \vee \bar{\alpha}_1 \vee \alpha_1 \bar{\alpha}_2 \bar{\alpha}_4 (\bar{\alpha}_1 \vee 1) = \\ = \alpha_1 \alpha_2 \vee \bar{\alpha}_1 \vee \alpha_1 \bar{\alpha}_2 \bar{\alpha}_4 = \alpha_1 (\alpha_2 \vee \bar{\alpha}_2 \bar{\alpha}_4) \vee \bar{\alpha}_1 \end{aligned}$$

care după cum reiese din tabelul 1 nu este identic egală cu unitatea.

Tabelul 1

α_1	α_2	α_4	$\bar{\alpha}_2 \bar{\alpha}_4$	$\alpha_2 \vee \bar{\alpha}_2 \bar{\alpha}_4$	$\alpha_1 (\alpha_2 \vee \bar{\alpha}_2 \bar{\alpha}_4) \vee \bar{\alpha}_1$
1	1	1	0	1	1
1	0	0	1	1	1
0	1	0	0	1	1
0	0	1	0	0	1
1	1	0	0	1	1
1	0	1	0	0	0
0	1	1	0	1	1
0	0	0	1	1	1

Din tabelul 1 se observă că pentru $\alpha_1 = 1$, $\alpha_2 = 0$ și $\alpha_4 = 1$ funcția logică $\alpha_1 (\alpha_2 \vee \bar{\alpha}_2 \bar{\alpha}_4) \vee \bar{\alpha}_1$ ia valoarea zero și deci ea nu este identic adevărată.

Pentru elementele celei de-a doua linii avem :

$$\bar{\alpha}_4 \alpha_1 \alpha_2 \vee \bar{\alpha}_4 \alpha_1 \vee \alpha_4 = \bar{\alpha}_4 \alpha_1 (\alpha_2 \vee 1) \vee \alpha_4 = \bar{\alpha}_4 \alpha_1 \vee \alpha_4$$

care pentru orice valoare a lui α_3 și $\alpha_4 \equiv 0$ și $\alpha_1 = 0$ este egală cu zero ceea ce ne arată că nici această funcție nu este identic adevărată.

Problema pe care o punem în această lucrare este aceea că fiind dată o schemă logică de algoritmi S definită ca în [2], schema logică matricială obținută din ea prin procedeul P satisface sau nu satisface următoarele proprietăți :

$$(5) \quad \bigvee_{j=1}^{p+1} \alpha_{ij} (X) \equiv 1, \quad i = \overline{1, p}$$

$$(6) \quad \alpha_{is} (X) \& \alpha_{it} (X) \equiv 0 \quad t \neq s \text{ și } i = \overline{1, p},$$

oricare ar fi X , stare a memoriei [2], [3]? Se observă că schema logică de algoritmi dată în exemplul 1 satisface aceste proprietăți, însă schema logică dată în exemplul 2 nu satisface decât proprietatea (6). De altfel în [2] este demonstrat că proprietatea (6) este satisfăcută de orice schemă matricială care se obține cu ajutorul procedurii P dintr-o schemă logică de algoritmi. Rămâne deci să lămurim ce condiții trebuie să îndeplinească o schemă logică de algoritmi, pentru ca elementele schemei matriciale obținute din ea să satisfacă proprietatea (5)? Pentru a rezolva această problemă reamintim mai întâi câteva noțiuni din [2].

Procedeul P dat în [2] ne permitea ca cu ajutorul lui dintr-o schemă logică de algoritmi să obținem toate subșirurile de forma

$$(7) \quad B_i H_{i_1} H_{i_2} \dots H_{i_k} B_j, \quad 0 \leq i \leq n+1,$$

cu următoarele proprietăți :

1. B_i este un operator aritmetic și dacă $j = i + 1$, atunci și B_j este tot un operator aritmetic.

2. $H_{i_1}, H_{i_2}, \dots, H_{i_k}$ ($k > 0$) sînt condiții logice astfel încît dacă $H_{i_{t+1}}$ este condiția logică care urmează după H_{i_t} în subșirul (7), atunci sau $i_{t+1} = i_t + 1$ sau i_{t+1} este indicele de transfer [2] al elementului B_{i_t} după cum în acest element funcția logică apare ca și în schema logică de algoritmi sau apare negația ei.

3. B_j este un operator aritmetic dacă elementele subșirului $H_{i_1}, H_{i_2}, \dots, H_{i_k}$ ($k > 0$) au proprietatea că oricare ar fi un element H_{i_ν} din acest subșir, cu funcția sa logică α_{i_ν} atunci atît această funcție cît și negația ei nu mai apar între elementele acestui subșir.

4. Dacă B_j este o condiție logică cu funcția logică α_j , atunci neapărat există în subșirul $H_{i_1} H_{i_2}, \dots, H_{i_k}$ ($k > 0$) un indice i_μ astfel că $i_\mu = j$. Evident, dacă notăm cu B_{i_μ} funcția logică din elementul H_{i_μ} al subșirului (7), atunci între β_{i_μ} și α_j avem una din relațiile $\beta_{i_\mu} = \alpha_j$ sau $\beta_{i_\mu} = \bar{\alpha}_j$. Cu aceste precizări putem enunța următoarea

TEOREMĂ. *Condiția necesară și suficientă ca unei scheme logice de algoritmi să i se poată atașa o schemă logică matricială ale cărei elemente*

să satisfacă proprietatea (5) este ca schema logică de algoritmi să fie astfel încât în orice subșir de forma (7) obținut cu procedeul P ultimul element să fie un operator aritmetic.

Demonstrație. Necesitatea. Considerăm șirul tuturor subșirurilor de forma (7) care încep cu un oarecare operator B_i , și presupunem că în acest subșir există cel puțin un subșir de forma (7) care conține ca ultim element o funcție logică. Fie subșirul considerat de forma

$$(8) \quad B_i \alpha_{i_1} \alpha_{i_2} \dots \alpha_{i_{s-1}} \alpha_{i_s},$$

unde α_{i_s} este o funcție logică. Fie C_i^0 mulțimea tuturor subșirurilor care încep cu operatorul B_i și fie X o stare a memoriei în așa fel ca :

$$(9) \quad \alpha_{i_1}(X) = \alpha_{i_2}(X) = \dots \alpha_{i_{s-1}}(X) = 1.$$

Vom arăta că pentru această stare a memoriei nu are loc condiția (5). În adevăr, pentru aceasta vom împărți mulțimea C_i^0 în două mulțimi disjuncte $C_i^{0,1}$ și $C_i^{0,2}$ astfel : din mulțimea $C_i^{0,1}$ fac parte acele subșiruri care încep cu B_i și pentru care funcția logică α_{i_1} apare ca și în șirul (8) (adică în elementele mulțimii $C_i^{0,1}$ după operatorul B_i apare chiar α_{i_1} și nu $\bar{\alpha}_{i_1}$), și considerăm mulțimea $C_i^{0,2} = C_{C_i^0}(C_i^{0,1})$. Elementele mulțimii $C_i^{0,2}$ evident vor conține după operatorul aritmetic B_i funcția logică $\bar{\alpha}_{i_1}$.

Din modul cum am construit mulțimile $C_i^{0,1}$ și $C_i^{0,2}$ rezultă că subșirul (8) aparține mulțimii $C_i^{0,1}$ și deoarece $\bar{\alpha}_{i_1}(x) = 0$ are loc relația

$$(10) \quad \bigvee_{j \in C_i^{0,2}} \beta_{ij} = 0,$$

unde $\beta_{ij} = \bar{\alpha}_{i_1} \& \alpha_{i_2}^j \& \dots \& \alpha_{i_k}^j$ și $B_i \bar{\alpha}_{i_1} \alpha_{i_2}^j \dots \alpha_{i_k}^j B_j$ aparține mulțimii $C_i^{0,2}$. Cu mulțimea $C_i^{0,1}$ procedăm la fel ca și cu mulțimea C_i^0 și fie $C_i^{0,1,1}$ și $C_i^{0,1,2}$, unde de data aceasta mulțimea $C_i^{0,1,1}$ conține în subșirurile sale ca al treilea element tocmai funcția logică α_{i_2} așa cum apare ea în subșirul (8), adică șirul (8) aparține mulțimii $C_i^{0,1,1}$ iar $C_i^{0,1,2} = C_{C_i^{0,1}}(C_i^{0,1,1})$. Se observă imediat că deoarece $\bar{\alpha}_{i_2}(X) = 0$, avem relația

$$(11) \quad \bigvee_{j \in C_i^{0,1,2}} \beta_{ij} = 0,$$

unde β_{ij} au aceeași semnificație ca în (10).

Continuând analog și aplicînd de $s-2$ ori acest procedeu, vom

construi mulțimile $C_i^{\overbrace{s-2}^{0,1,1,1,\dots,1}} = C_i^0(1, s-1)$ și $C_i^{\overbrace{s-2}^{0,1,1,1,\dots,1,2}} = C_i^0(2, s-2)$ astfel ca subșirul (8) să aparțină lui $C_i^0(1, s-2)$ și $C_i^0(2, s-2) = C_{C_i^0(1, s-2)}(C_i^0(1, s-3))$

și evident avem :

$$(12) \quad \bigvee_{j \in C_i^0(2, s-2)} \beta_{ij} = 0.$$

Din construcția procedurii P rezultă că clasa $C_i^0(1, s-1)$ care se construiește ca mai sus din clasa $C_i^0(1, s-2)$ va conține un singur element și anume subșirul (8).

Din ceea ce am considerat mai sus, rezultă că dacă au loc relațiile (9) pentru o stare a memoriei convenabil aleasă, atunci avem :

$$(13) \quad \bigvee_{\substack{j=1 \\ i \in C_i^0(2, i)}}^{s-1} \beta_{ij} = 0$$

Deoarece

$$(14) \quad C_i^0 = \left[\bigcup_{h=1}^{s-1} C_i^0(2, h) \right] \cup (C_i^0(1, s-1)),$$

urmează că nu are loc relația (5) după ce ms-a arătat în [2] conjuncția $\alpha_{i_1} \& \alpha_{i_2} \& \dots \& \alpha_{i_s}$ nu intră în nici un element din linia i a schemei matriciale.

Suficiența. Fie S este o schemă logică de algoritmi astfel încât prin procedeul P dat în [2] orice subșir de forma (7), conține ca ultim element un operator aritmetic diferit.

De aici urmează că pentru orice stare a memoriei după executarea unui oarecare operator diferit de operatorul de oprire urmează să se execute în mod obligatoriu un alt operator. Dacă ținem seamă de echivalența între schemele logice de algoritmi și schemele logice matriciale [1], rezultă că cel puțin pentru o stare a memoriei există în orice linie a schemei matriciale un element care are valoarea 1 și de aceea rezultă suficiența.

Primită la redacție la 29 mai 1965

Institutul de calcul al
Academiei Republicii Socialiste România — Filiala Cluj

BIBLIOGRAFIE

1. И. И. ЯНОВ, О логических схемах алгоритмов. Проблемы кибернетики, I, п. 75—127, (1958).
2. L. NEGRESCU și I. PĂVĂLOIU, Asupra verificării formale a schemelor logice de algoritmi. St. cerc. mat., 17, 2, 271—286 (1965).
3. Р. И. ПОДЛОВЕНКО, О преобразованиях схем программ и их применений в программировании. Проблемы кибернетики, 7 (1962).