

ANALIZA LOGICĂ A ALGORITMILOR (II)

DE

E. MUNTEANU

(Cluj)

Prezenta Notă este o continuare a lucrării [1]. Pentru rezolvarea problemei de analiză a algoritmilor formulată în [1], este necesar ca algoritmul schemă-graf să fie adus la forma canonică.

În lucrare se definește forma canonică a unui algoritm schemă-graf (2.3) și se dau metode de aducerea unui algoritm la forma canonică. Prima metodă (teorema 2) este un procedeu universal care permite aducerea oricărui algoritm la forma canonică. Neajunsul acestui procedeu constă în aceea că se obține o formă canonică cu o structură complicată a ciclului.

A doua metodă, bazată pe transformările schemelor-graf cu două intrări, cu două ieșiri și cu cicluri ce se intersectează, permite obținerea formei canonice în care ciclurile nu au structură simplă, ea se poate aplica însă numai la anumite clase de algoritmi care sînt descrise în lucrare.

În lucrarea [1] a fost formulată problema analizei algoritmilor ca o corespondență între sistemul de condiții logice care descrie formularea unei probleme și algoritmul de rezolvare a acestei probleme.

Pentru rezolvarea problemei de analiză, în aceeași lucrare a fost introdusă noțiunea de formulă atașată unui algoritm și a fost dată o metodă de construire a formulei atașate unui algoritm schemă-graf fără cicluri.

Construirea formulei atașate unui algoritm cu cicluri se poate face, deocamdată, numai în cazul cînd algoritmul este adus la așa-numita formă canonică.

În această Notă se definește forma canonică a algoritmilor schemă-graf și se studiază transformările cu ajutorul cărora se poate obține în anumite cazuri forma canonică a unui algoritm dat.

1. DESCOMPUNEREA UNEI SCHEME-GRAF

Vom generaliza noțiunea de schemă-graf introdusă de A.L. Kalujnin [2] în felul următor:

Printr-o *schemă-graf generalizată* înțelegem un obiect matematic, descris în felul următor.

Fie dată mulțimea $T = \{T_1, T_2, \dots, T_n\}$, ale cărei elemente le vom numi *operatori de transformare* și mulțimea $L = \{L_1, L_2, \dots, L_n\}$, ale cărei elemente le vom numi *operatori logici*.

O schemă-graf generalizată este un graf finit, G , legat și orientat, ale cărui virfuri le vom numi *noduri*. Anumite noduri sînt legate între ele prin *săgeți*.

Graful G satisface următoarele proprietăți:

1. Există anumite noduri marcate care se numesc *intrări* și care sînt singurele noduri în care nu intră nici o săgeată; din fiecare intrare pleacă numai o săgeată. Notăm aceste noduri cu $I_1, I_2, \dots, I_p$.

2. Există anumite noduri marcate care se numesc *ieșiri* și care sînt singurele noduri din care nu pleacă nici o săgeată. Notăm aceste noduri cu $E_1, E_2, \dots, E_q$.

3. Oricărui nod al grafului G , diferit de o intrare sau de o ieșire, i se atașază un operator de transformare T_i sau un operator logic L_i . În primul caz nodul se numește *T-nod* iar în al doilea caz *L-nod*.

4. Din fiecare *T-nod* pleacă o singură săgeată și din fiecare *L-nod* pleacă două săgeți. În ultimul caz cele două săgeți sînt marcate una cu semnul plus (+) și cealaltă cu semnul minus (-).

Interpretarea unei scheme-graf generalizate este analogă cu interpretarea dată în lucrarea [1]. Vom păstra în continuare notațiile introduse în această lucrare.

Este evident că orice schemă-graf este o schemă-graf generalizată pentru care $p = 1$ și $q = 1$.

Să considerăm o schemă-graf S . Fie operatori O_i , care diferă de intrare și O_1 , care diferă de ieșire.

DEFINIȚIA 1. Operatorii O_i și O_1 se numesc separabili dacă O_i este succesul operatorului O_i și nu există un drum invers de la operatorul O_i la operatorul O_1 .

Dacă operatorii O_i și O_1 sînt separabili, vom spune că operatorul O_1 se separă de operatorul O_i .

Pentru fiecare operator de atribuire al schemei-graf S , există cel mult un operator care se separă de operatorul respectiv, iar pentru fiecare operator logic există cel mult doi operatori care se separă de operatorul logic considerat.

Dacă operatorii O_{q_k} și O_{p_k} sînt separabili, vom spune că ei formează o pereche și vom nota această pereche în felul următor:

$$(1) \quad D_k = (O_{p_k}, O_{q_k})$$

înțelegînd prin aceasta că operatorul O_{q_k} se separă de operatorul O_{p_k} .

Vom nota mulțimea tuturor perechilor schemei-graf S astfel

$$(2) \quad V = \{V_1, V_2, \dots, V_h\}$$

unde fiecare element V_i ($i = 1, 2, \dots, h$) este de forma (1).

DEFINIȚIA 2. Vom spune că submulțimea nevidă

$$V^* = \{V_{k_1}, V_{k_2}, \dots, V_{k_m}\}, \quad m \leq h$$

a mulțimii V formează o *tăietură* în schema-graf S , dacă după înlăturarea săgeților care leagă operatorii $O_{p_{k_1}}, O_{p_{k_2}}, \dots, O_{p_{k_m}}$ cu operatorii $O_{q_{k_1}}, O_{q_{k_2}}, \dots, O_{q_{k_m}}$ respectiv, schema-graf S se descompune în două părți nelegate între ele cu săgeți.

Numim aceste două părți subschemă-graf determinate de tăietura V . Vom nota cu $\Gamma_s^1(V^*)$, subschemă-graf a cărei intrare coincide cu intrarea schemei-graf S , și cu $\Gamma_s^2(V^*)$, schema-graf a cărei ieșire coincide cu ieșirea schemei-graf S .

Următoarele proprietăți sînt evidente:

1.1. Subschemă-graf $\Gamma_s^1(V^*)$ are cel puțin o ieșire.

1.2. Subschemă-graf $\Gamma_s^2(V^*)$ are cel puțin o intrare.

1.3. Fiecărei intrări a subschemei-graf $\Gamma_s^2(V^*)$ îi corespunde cel puțin o ieșire a subschemei-graf $\Gamma_s^1(V^*)$. Din proprietățile 1.1 și 1.2 rezultă că schemele-graf $\Gamma_s^1(V^*)$ și $\Gamma_s^2(V^*)$ sînt scheme-graf generalizate.

TEOREMA 1. *Condiția necesară și suficientă pentru ca o schemă-graf S să admită cel puțin o tăietură este ca mulțimea perechilor schemei să fie nevidă.*

Este evident că dacă schema-graf S admite cel puțin o tăietură, atunci mulțimea V este nevidă.

Fie $V = \{V_1, V_2, \dots, V_h\}$, mulțimea tuturor perechilor schemei-graf S . Presupunem că V este nevidă. Vom arăta că în acest caz S admite cel puțin o tăietură. Fie $V_1 = (O_{p_1}, O_{q_1})$. Înlăturăm din S săgeata care unește pe O_{p_1} cu O_{q_1} . Dacă S se descompune în două subscheme-graf, atunci $V^* = \{V_1\}$ și teorema este demonstrată. Dacă nu, înseamnă că în graful neorientat al schemei S , există un drum de la operatorul O_{q_1} la operatorul O_{p_1} . Fie $O_{i_1}, \dots, O_{i_k}$, totalitatea operatorilor care aparțin acestui drum. Pentru fiecare $0 \leq j \leq k$, căutăm în graful orientat un drum de la operatorul O_{i_j} la operatorul $O_{i_{j+1}}$. Dacă pentru orice $0 \leq j \leq k$ există un astfel de drum, atunci există un drum, în graful orientat, de la operatorul O_{q_1} la O_{p_1} . Această concluzie este falsă deoarece $(O_{p_1}, O_{q_1}) \in V$. Rezultă atunci că pentru un anumit j nu există un drum orientat de la O_{i_j} la $O_{i_{j+1}}$, dar acești operatori sînt legați printr-o săgeată. Înseamnă deci că această săgeată pleacă de la operatorul $O_{i_{j+1}}$ la operatorul O_{i_j} . Prin urmare $(O_{i_{j+1}}, O_{i_j}) \in V$. Înlăturăm și această săgeată și dacă S se descompune în două subscheme, atunci $V^* = \{V_1, (O_{i_{j+1}}, O_{i_j})\}$, iar dacă nu, procesul se continuă.

După un număr finit de pași procesul de construire a mulțimii V se termină, fie atunci cînd se obține o mulțime ce determină o tăietură

în schema S , fie atunci când $V^* = V$. În ultimul caz mulțimea V determină o tăietură. Într-adevăr, după înlăturarea săgeților, care leagă operatorii $O_{p_1}, O_{p_2}, \dots, O_{p_h}$ cu operatorii $O_{q_1}, O_{q_2}, \dots, O_{q_h}$ nu există nici un drum din operatorul O_{p_i} în operatorul O_{q_i} și nici operatorul O_{q_i} în operatorul O_{p_i} , deoarece V este mulțimea tuturor perechilor de operatori separabili.

DEFINIȚIA 3. Orice schemă-graf generalizată, care admite cel puțin o tăietură se numește *decompozabilă*.

Din definiția 3 și teorema 2 rezultă că orice schemă-graf generalizată a cărei mulțime a perechilor de operatori separabili este nevidă, este o schemă decompozabilă.

O schemă-graf generalizată care nu este decompozabilă se numește *indecompozabilă*. Mulțimea perechilor de operatori separabili a unei scheme-graf generalizate indecompozabilă este vidă.

Descompunerea unei scheme-graf în subscheme-graf se face efectuând toate tăieturile posibile în schema dată.

2. FORMA CANONICĂ A UNEI SCHEME-GRAF

2.1. Schema-graf S se numește *elementară*, dacă ea are forma din figura 1, unde blocul 1 este o schemă-graf generalizată cu o singură intrare (1) și care nu conține cicluri. Vom numi această schemă, *subscheme-graf interioară ciclului*.

2.2. Schema-graf S este de tip elementar, dacă ea are aceeași formă (fig. 1) și subscheme-graf interioară ciclului este o schemă-graf generalizată oarecare cu o singură intrare.

Evident că orice schemă-graf elementară este o schemă-graf de tip elementar.

Noțiunea de formă canonică a unei scheme-graf S se definește în felul următor:

2.3. Vom spune că o schemă-graf S este adusă la forma canonică dacă ea satisface una din următoarele cerințe:

- S nu conține cicluri.
- S este o schemă-graf elementară.
- S este o schemă-graf de tip elementar, a cărei subscheme-graf interioară ciclului este adusă la forma canonică.
- S are forma din figura 2 unde subschemele-graf 1 și 2 sînt aduse la forma canonică.

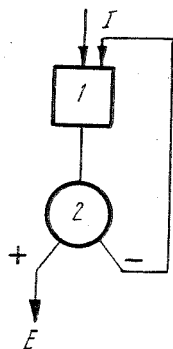


Fig. 1



Fig. 2

DEFINIȚIA 4. Algoritmul schemă-graf $\mathcal{A}$ este adus la forma canonică, dacă schema-graf corespunzătoare este adusă la forma canonică.

TEOREMA 2. Orice algoritm schemă-graf poate fi adus la forma canonică.

Pentru a demonstra această teoremă vom da o metodă de construire a formei canonice a unui algoritm schemă-graf oarecare. Fie algoritmul $\mathcal{A}$, care constă din operatorii $O_0, O_1, \dots, O_n$, unde O_n este ieșirea.

Dacă O_j este un operator de atribuire și O_i este succesorul lui, iar i este o variabilă care nu intră în descrierea algoritmului $\mathcal{A}$, atunci vom nota cu K_j ($j = 1, 2, \dots, n-1$) schema-graf din figura 3.

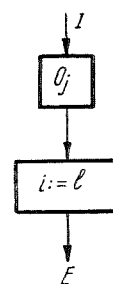


Fig. 3

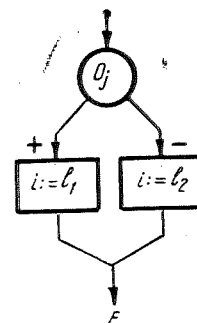


Fig. 4

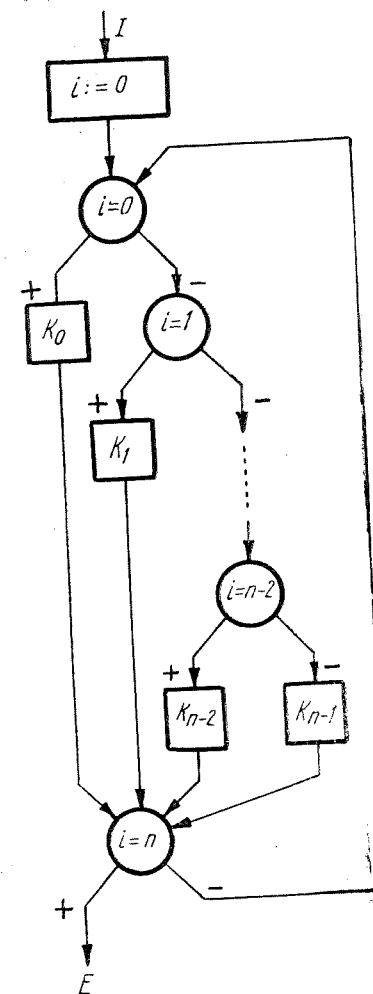


Fig. 5

Dacă O_i este un operator logic iar O_{i_1} și O_{i_2} sînt succesorii lui, atunci vom nota cu K_i schema-graf generalizată din figura 4.

Construim algoritmul schemă-graf $\mathcal{A}^*$ a căruia schemă-graf S^* este cea din figura 5.

Este evident că algoritmul $\mathcal{A}^*$ este echivalent cu algoritmul $\mathcal{A}$, iar schema-graf S este elementară. Rezultă deci că algoritmul $\mathcal{A}^*$ este forma canonică a algoritmului $\mathcal{A}$.

Metoda aceasta nu corespunde întrutotul scopului pe care îl urmărim, deoarece se obține o schemă-graf elementară, cu o structură foarte complicată a ciclului chiar și în cazurile simple. Or, pentru a construi formula atașată unui algoritm, noi tindem să obținem o formă canonică cu o structură simplă a ciclurilor.

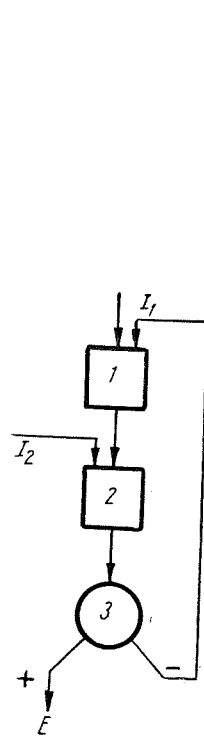


Fig. 6

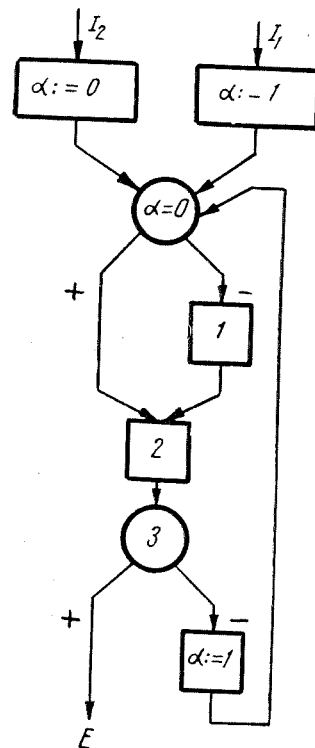


Fig. 7

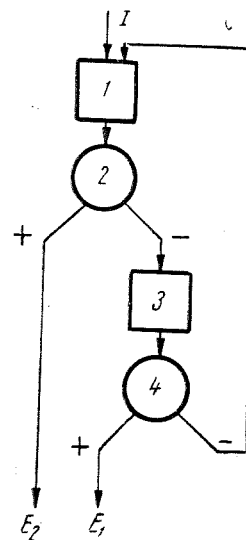


Fig. 8

Vom numi procedeul descris mai sus *procedeul universal* de construire a formei canonice a unui algoritm dat.

Alături de acest procedeu, vom introduce trei tipuri de transformări, care ne permit să definim un alt procedeu de construire a formei canonice, care în cazurile cele mai frecvente ne dă o formă canonică cu o structură a ciclurilor mai simplă decât cea obținută prin procedeul universal.

2.4. Schema-graf generalizată S se numește *schemă-graf cu două intrări*, dacă este de forma din figura 6 unde subschemele 1 și 2 sînt scheme-graf oarecare, care nu au alte intrări și ieșiri decît cele reprezentate în figură.

Fie $\mathcal{A}$ algoritmul definit de S și de o interpretare dată și α o variabilă ce nu intră în descrierea lui $\mathcal{A}$. Atunci $\mathcal{A}$ este echivalent cu algoritmul definit de schema-graf din figura 7 și de aceeași interpretare.

Schema-graf generalizată din figura 7 se descompune într-o schemă-graf generalizată fără cicluri și o schemă-graf de tip elementar.

2.5. Schema-graf generalizată S se numește *schemă-graf cu două ieșiri* dacă este de forma din figura 8 unde schemele-graf 1 și 2 sînt oare-

care și nu conțin alte intrări și alte ieșiri decît cele reprezentate în figură.

Fie $\mathcal{A}$ algoritmul definit de S și de o interpretare dată, iar α și β variabile ce nu intră în descrierea lui $\mathcal{A}$. Atunci $\mathcal{A}$ este echivalent cu algoritmul definit de schema-graf din figura 9 și de aceeași interpretare.

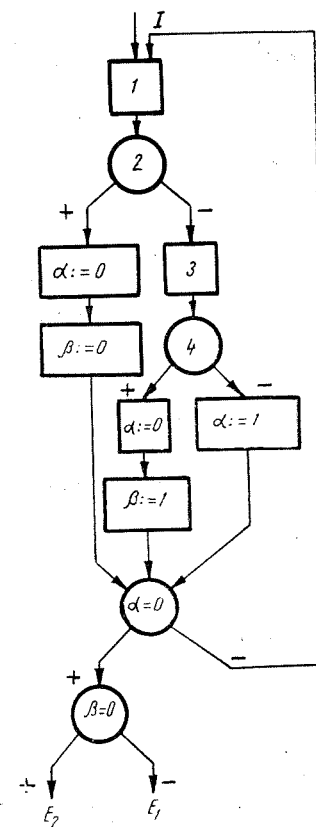


Fig. 9

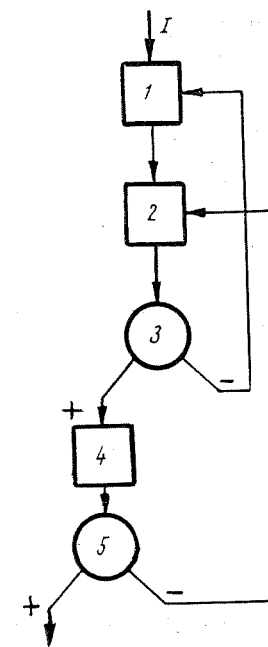


Fig. 10

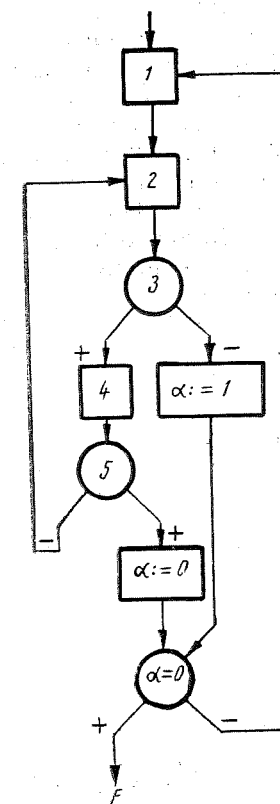


Fig. 11

Schema-graf generalizată din figura 9 se descompune într-o schemă-graf de tip elementar și o schemă-graf generalizată fără cicluri.

2.6. Schema-graf S se numește *schemă cu cicluri ce se intersectează* dacă este de forma din figura 10 unde subschemele 1, 2 și 4 sînt oarecare și nu au alte intrări și ieșiri decît cele reprezentate în figură.

Fie $\mathcal{A}$ algoritmul definit de schema-graf S și de o interpretare dată, iar α o variabilă ce nu intră în descrierea lui $\mathcal{A}$. Atunci $\mathcal{A}$ este echivalent cu algoritmul definit de aceeași interpretare și de schema-graf din figura 11. Schema-graf din figura 11 este o schemă-graf de tip elementar.

Schemele cu două intrări, cu două ieșiri și schemele cu cicluri ce se intersectează nu epuizează toate tipurile de scheme-graf neaduse la forma canonică.

Pe baza acestor transformări descrise mai sus, definim următorul procedeu de construire a formei canonice.

2.7. Fie algoritmul schemă-graf $\mathfrak{A}$ definit de schema-graf S și de o interpretare dată.

2.7.1. Descompunem schema-graf S în subschemele indecompozabile $S_1, S_2, \dots, S_k$.

2.7.2. Pentru fiecare schemă S_i ($i = 1, 2, \dots, k$), care nu este adusă la forma canonică, se verifică dacă nu se poate aplica una din transformările (2.4), (2.5) sau (2.6), adică se verifică dacă S_i nu este o schemă cu două intrări sau o schemă cu două ieșiri sau o schemă cu cicluri ce se intersectează și în caz afirmativ se transformă S_i în mod corespunzător. Dacă S_i nu este de unul din cele trei tipuri indicate, atunci se aplică procedeul universal pentru aducerea ei la forma canonică.

2.7.3. Dacă S_i a fost transformată conform uneia din transformările (2.4), (2.5) sau (2.6), atunci în fiecare caz se obține o schemă care sau este de tip elementar (transformarea schemelor cu cicluri ce se intersectează) sau se descompune în două subscheme dintre care una este de tip elementar și este adusă la forma canonică.

Notăm cu S'_i schema-graf interioară ciclului unei scheme-graf de tip elementar obținută printr-una din aceste transformări.

Procesul se aplică schemei S'_i pînă cînd aceasta se aduce la forma canonică.

2.7.4. Aplicarea acestui procedeu se termină cînd toate schemele S_i au fost aduse la forma canonică.

Primită la redacție la 9 august 1966

*Institutul de calcul al Academiei Republicii Socialiste
România — Filiala Cluj.*

BIBLIOGRAFIE

1. E. MUNTEANU, *Analiza logică a algoritmilor (I)*. St. cerc. matem. **18**, (1966).
2. Л. А. КАЛУЖНИН, *Об алгоритмизации математических задач*. Проблемы кибернетики, вып. 3. (1959).