

# TRATAREA ALGEBRICĂ A LIMBAJELOR FORMALIZATE

DE

TEODOR RUS

În lucrare, pornindu-se de la definiția unui limbaj formalizat ca și o anumită submulțime a unui semigrup liber, se face o legătură între tratarea obișnuită a acestuia și tratarea lui ca și un anumit sistem relațional. Sistemul respectiv este indus în limbaj de gramatica acestuia. Pornindu-se de la considerații din teoria relațiilor, se ajunge la definirea celor două probleme fundamentale ale unui limbaj, și anume problema analizei și problema ambiguității, în limbajul sistemelor relaționale. De asemenea, cu ajutorul acestor noțiuni se construiește un algoritm de analiză a unui limbaj sub forma unui algoritm normal.

## § 1. NOTIUNEA DE LIMBAJ FORMALIZAT

Fie  $V$  o mulțime finită nevidă de simboluri care admite o descompunere de forma  $V = V_N \cup V_T$ , unde  $V_N \cap V_T = \emptyset$ ,  $\emptyset$  fiind mulțimea vidă. Elementele mulțimii  $V_N$  se vor numi în continuare categorii sintactice, iar elementele mulțimii  $V_T$  se vor numi litere ale alfabetului limbajului formalizat definit peste mulțimea  $V$ .

Notăm cu litere mari latine elementele lui  $V_N$  și cu litere mici latine elementele mulțimii  $V_T$ .

Să considerăm semigrupul liber generat de mulțimea  $V$ , pe care îl notăm cu  $W(V)$ . Elementele semigrupului liber se vor numi în continuare cuvinte, și se vor nota de asemenea cu litere mici latine de la sfârșitul alfabetului însoțite de indici dacă este nevoie. Notăm cu  $P$  o mulțime finită de perechi de forma  $(A_i, w_i)$ , unde  $A_i \in V_N$ ,  $w_i \in W(V)$  pe care le vom numi mulțimea producțiilor. Dacă  $(A_i, w_i) \in P$ , atunci aceasta se mai poate scrie și sub forma  $A_i \rightarrow w_i$ , ceea ce înseamnă că avem de-a face cu o regulă de producere. Oricare ar fi regula de producere  $(A_i, w_i) \in P$ , ea furnizează o aplicație definită în semigrupul liber  $W(V)$  cu valori în semigrupul liber  $W(V)$ . Dacă notăm cu  $f$  această aplicație, atunci  $x \in W(V)$  aparține domeniului de definiție al lui  $f$  dacă  $x$  are forma  $x = tA_i v$ . Valoarea aplicației  $f$  asupra lui  $x$  se va scrie sub forma  $xf$  și este  $tw_i v$ .

Faptul că  $(A_i w_i) \in P$  furnizează aplicația  $f$  în semigrupul liber,  $W(V)$  se scrie sub forma  $f_{(A_i, w_i)}$ . Domeniul lui  $f$  se notează cu  $D_{(A_i, w_i)} = \{x | x = tA_i v\}$  iar mulțimea valorilor lui  $f_{(A_i, w_i)}$  se notează cu

$$O_{(A_i, w_i)} = \{y | y = tw_i v\}.$$

DEFINIȚIA 1.1. Numim gramatică un cvadruplu ordonat de forma  $(W(V), V_T, \{P, \{A_1, A_2, \dots, A_n | A_i \in V_N\}\})$  și notăm gramaticile cu litera  $G$ .

DEFINIȚIA 1.2. Dacă  $x, y \in W(V)$ , vom spune că  $y$  este o derivare a lui  $x$  cu ajutorul unei gramatici  $G$  dacă  $x = y$  sau dacă există șirul de cuvinte  $x_0, x_1, \dots, x_n$  astfel ca să avem :

$$x_0 = x$$

$$x_i f_{(A_j, w_j)} = x_{i+1} \quad (i = 0, 1, \dots, n-1)$$

$$x_n = y,$$

unde  $f_{(A_j, w_j)}$  este o aplicație furnizată de una din perechile  $(A_j, w_j) \in P$ .

Dacă se folosește notația  $A_i \rightarrow w_i$  pentru  $(A_i, w_i) \in P$  și dacă  $x_i f_{(A_j, w_j)} = x_{i+1}$  se scrie sub forma  $x_i \rightarrow x_{i+1}$ , atunci noțiunea de mai sus se poate prezenta sub forma

$$x_0 = x$$

$$x_i \rightarrow x_{i+1} \quad (i = 0, 1, \dots, n-1)$$

$$x_n = y$$

ceea ce coincide cu notațiile obișnuite [1], [2], [3], [4].

Dacă  $y$  este o derivare a lui  $x$ , atunci uneori vom nota acest lucru prin  $x \Rightarrow y$ .

DEFINIȚIA 1.3. O submulțime  $L \subseteq W(V)$  se numește limbaj formalizat generat de gramatica  $G$  dacă elementele lui  $L$  sînt cuvinte în  $W(V_T)$  și dacă oricare ar fi  $x \in L$ , există o categorie sintactică  $A_i \in V_N$  astfel ca  $x$  să fie o derivare a lui  $A_i$  cu ajutorul gramaticii  $G$ .

Mai scurt, limbajul  $L_G$  generat de gramatica  $G$ , se poate defini astfel

$$L_G = \{x | x \in W(V_T) \text{ \& } \exists A_i (A_i \in V_N \text{ \& } A_i \Rightarrow x)\}.$$

## §.2 RELAȚII BINARE ÎN LIMBAJELE FORMALIZATE

DEFINIȚIA 2.1. Fie  $M$  o mulțime nevidă de elemente. Considerăm produsul cartezian  $M \times M$ . O submulțime  $\Phi \subseteq M \times M$  se va numi relație binară definită pe mulțimea  $M$ .

Din teoria relațiilor [5], [6], [7] este bine cunoscută următoarea teoremă.

TEOREMA 2.1. *Mulțimea relațiilor n-are (în particular mulțimea relațiilor binare) definite peste o mulțime  $M$  formează o algebră conținută în algebra submulțimilor mulțimii  $M$ .*

Date fiind relațiile  $\Phi_1$  și  $\Phi_2$  definite pe mulțimea  $M$ , avem :

1.  $\Phi_1 \subseteq \Phi_2$  în sensul teoriei mulțimilor.
2. Reuniunea  $\Phi_1 \cup \Phi_2 = \Phi$  se definește în felul următor : dacă  $(x, y) \in \Phi_1 \cup \Phi_2$ , atunci  $(x, y) \in \Phi_1$  sau  $(x, y) \in \Phi_2$ . Deci  $(x, y) \in \Phi$ . Dacă  $(x, y) \in \Phi$ , atunci  $(x, y) \in \Phi_1$  sau  $(x, y) \in \Phi_2$  deci  $(x, y) \in \Phi_1 \cup \Phi_2$ . Prin urmare reuniunea a două relații  $\Phi_1, \Phi_2$  se definește ca relația care conține atît submulțimea produsului cartezian care definește pe  $\Phi_1$ , cît și submulțimea produsului cartezian care definește pe  $\Phi_2$ .
3. Intersecția  $\Phi_1 \cap \Phi_2 = \Phi$  a două relații  $\Phi_1$  și  $\Phi_2$  se definește în mod asemănător ca submulțimea produsului cartezian care conține elementele comune lui  $\Phi_1$  și  $\Phi_2$ , adică

$$(x, y) \in \Phi_1 \cap \Phi_2 \text{ implică } (x, y) \in \Phi_1 \text{ \& } (x, y) \in \Phi_2 \text{ deci } (x, y) \in \Phi.$$

$$(x, y) \in \Phi \text{ implică } (x, y) \in \Phi_1 \text{ \& } (x, y) \in \Phi_2, \text{ deci } (x, y) \in \Phi_1 \cap \Phi_2.$$

4. Relația complementară  $\bar{\Phi}$  a unei relații  $\Phi$  se definește în felul următor :

$$(x, y) \in \bar{\Phi} \text{ dacă } (x, y) \in M \times M \text{ și } (x, y) \notin \Phi.$$

Date fiind relațiile  $\Phi_1$  și  $\Phi_2$  pe mulțimea  $M$ , vom defini produsul relațiilor  $\Phi_1$  și  $\Phi_2$  astfel :

$$\Phi = \Phi_1 \circ \Phi_2 = \{(x, z) | (x, y) \in \Phi_1 \text{ \& } (y, z) \in \Phi_2\}.$$

Mulțimea relațiilor astfel definite posedă următoarele proprietăți :

1.  $\Phi_1 \circ (\Phi_2 \circ \Phi_3) = (\Phi_1 \circ \Phi_2) \circ \Phi_3$
2.  $(\bigcup_{i \in I} \Phi_i) \circ \Phi = \bigcup_{i \in I} \Phi_i \circ \Phi$
3.  $\Phi \circ (\bigcup_{i \in I} \Phi_i) = \bigcup_{i \in I} \Phi \circ \Phi_i$
4.  $(\bigcap_{i \in I} \Phi_i) \circ \Phi \subseteq \bigcap_{i \in I} \Phi_i \circ \Phi$ .

Din 2 și 3 de mai sus avem și proprietatea : dacă  $\Phi_1 \subseteq \Phi_2$ , atunci  $\Phi \circ \Phi_1 \subseteq \Phi \circ \Phi_2$ .

Noțiunea de relație inversă a unei relații se poate defini astfel : dată fiind relația  $\Phi = \{(x, y) | (x, y) \in M \times M\}$ , inversa ei este  $\Phi^{-1} = \{(y, x) | (x, y) \in \Phi\}$ .

Relația inversă a unei relații astfel definită se bucură de următoarele proprietăți :

1.  $(\Phi^{-1})^{-1} = \Phi$ ,
2.  $\Phi_1 \subseteq \Phi_2$  implică  $\Phi_1^{-1} \subseteq \Phi_2^{-1}$
3.  $(\bigcup_{i \in I} \Phi_i)^{-1} = \bigcup_{i \in I} \Phi_i^{-1}$
4.  $(\bigcap_{i \in I} \Phi_i)^{-1} = \bigcap_{i \in I} \Phi_i^{-1}$
5.  $(\Phi_1 \circ \Phi_2)^{-1} = \Phi_2^{-1} \circ \Phi_1^{-1}$

Prin  $I$  am notat o mulțime finită de indici.

Revenind acum la noțiunea definită în (1.1) și anume aceea de gramatică, vom avea următoarea leamnă:

LEMA 2.1. *O gramatică  $G$  induce în semigrupul liber  $W(V)$  care intră în definiția ei o mulțime finită de relații binare.*

Demonstrația acestei leme este aproape banală. Pentru realizarea ei să considerăm perechea  $(A_i, w_i) \in P$ , unde  $P$  este mulțimea perechilor de producție care intră în definiția gramaticii  $G$ . Atunci este aproape evident că fiecare asemenea pereche furnizează o relație binară definită în semigrupul liber  $W(V)$ . Să notăm cu  $\Phi_i$  relația furnizată de  $(A_i, w_i)$ .

Spunem că  $(x, y) \in W(V) \times W(V)$  aparține relației  $\Phi_i$  dacă aplicația  $f_{(A_i, w_i)}$  definită de  $(A_i, w_i)$  are proprietatea că

$$x f_{(A_i, w_i)} = y$$

sau cu alte cuvinte dacă  $x = tA_i v$  și  $y = tw_i v$ .

Să notăm cu  $\Phi$  mulțimea relațiilor binare furnizate de o gramatică  $G$  în semigrupul liber  $W(V)$  împreună cu diagonală lui  $W(V)$ . Atunci dacă  $\Phi_i$  și  $\Phi_j$  sînt elementele mulțimii  $\Phi$  și dacă  $(x, y) \in W(V) \times W(V)$  și  $(x, y) \in \Phi_i$  &  $(x, y) \in \Phi_j$ , avem următoarea definiție:

DEFINIȚIA 2.2. Spunem că  $\Phi_j$  precede pe  $\Phi_i$  pentru perechea  $(x, y) \in W(V) \times W(V)$ , și notăm cu  $\Phi_j < \Phi_i$ , dacă au loc proprietățile:

1.  $(x, y) \in \Phi_i$ , unde  $x = t_1 A_i v_1$ ,  $y = t_1 w_i v_1$ , iar  $(A_i, w_i) \in P$  și furnizează relația  $\Phi_i$ .

2.  $(x, y) \in \Phi_j$ , unde  $x = t_2 A_j v_2$ ,  $y = t_2 w_j v_2$ , iar  $(A_j, w_j) \in P$  și furnizează relația  $\Phi_j$ .

3.  $t_1 = t'_1 A_j t'_1$ , unde  $A_i$  nu posedă intrări în cuvîntul  $t_1$ .

DEFINIȚIA 2.3. Spunem că  $\Phi_i$  precede pe  $\Phi_j$  pentru perechea  $(x, y) \in W(V) \times W(V)$  și notăm cu  $\Phi_i < \Phi_j$ , dacă au loc proprietățile:

1.  $(x, y) \in \Phi_i$ , unde  $x = t_1 A_i v_1$ ,  $y = t_1 w_i v_1$ , iar  $(A_i, w_i) \in P$  și furnizează relația  $\Phi_i$ .

2.  $(x, y) \in \Phi_j$ , unde  $x = t_2 A_j v_2$ ,  $y = t_2 w_j v_2$ , iar  $(A_j, w_j) \in P$  și furnizează relația  $\Phi_j$ .

3.  $t_2 = t'_2 A_i t'_2$ , unde  $A_j$  nu posedă intrări în cuvîntul  $t_2$ .

Fie  $(x, y) \in W(V) \times W(V)$  și  $(x, y) \in \Phi_i$ . Să considerăm descompunerea cuvîntului  $x$  de forma

$$x = t_1 A_i v_1 t_2 A_i v_2 \dots t_n A_i v_n,$$

unde subcuvintele  $t_j, v_j$ ,  $j = 1, 2, \dots, n$ , nu conțin intrări ale categoriei sintactice  $A_i$ . Atunci faptul că  $(x, y) \in \Phi_i$ ,  $\Phi_i$  fiind relația furnizată de perechea  $(A_i, w_i)$ , se poate considera în  $n$  moduri, adică există  $n$  descompuneri ale cuvîntului  $x$  de forma

$$x = tA_i v \quad \text{cu } y = tw_i v.$$

Să considerăm prima intrare a categoriei sintactice  $A_i$  în cuvîntul  $x$  astfel ca să avem  $x = t_1 A_i v_1$ ,  $t_1$  neconținînd intrări ale altor categorii sintactice. Atunci spunem că  $(x, y) \in \Phi_i^{(1)}$ ,  $\Phi_i^{(1)}$  fiind relația determinată de  $\Phi_i$ , astfel că dacă  $(x, y) \in \Phi_i^{(1)}$ , atunci  $x = t_1 A_i v_1$ ,  $y = t_1 w_i v_1$  și  $t_1$  nu mai conține intrări ale categoriilor sintactice  $A_i \in V_N$ . Putem acum defini pe  $\Phi_i^{(2)}$  în același mod în felul următor: dacă  $(x, y) \in \Phi_i^{(2)}$ , atunci  $x = t_1 A_i v_1 t_2 A_i v_2$ , astfel ca  $t_1, v_1, t_2$  nu mai conțin intrări ale categoriei

sintactice  $A_i$ . În mod analog se poate construi  $\Phi_i^{(n)}$ , unde  $n$  este numărul intrărilor categoriei sintactice  $A_i$  în cuvîntul  $x$ .

Din definiția relațiilor  $\Phi_i^{(1)}, \Phi_i^{(2)}, \dots, \Phi_i^{(n)}$ , avem următoarea proprietate față de perechea  $(x, y) \in \Phi_i$ :

$$\Phi_i^{(1)} < \Phi_i^{(2)} < \dots < \Phi_i^{(n)}.$$

Convenim în continuare să considerăm relațiile  $\Phi_i^{(1)}, \Phi_i^{(2)}, \dots, \Phi_i^{(n)}$  ca și un sistem de relații furnizat de relația  $\Phi_i$ , pentru o pereche de elemente  $(x, y) \in \Phi_i$ , astfel ca  $x = t_1 A_i v_1 \dots t_n A_i v_n$ .

DEFINIȚIA 2.4. Fie  $(x, y) \in W(V) \times W(V)$  și  $(x, y) \in \Phi_i$ ,  $\Phi_i$  fiind furnizată de  $(A_i, w_i) \in P$ . Atunci numărul  $n$  al relațiilor furnizate de  $\Phi_i$  peste perechea  $(x, y) \in \Phi_i$  se numește gradul lui  $A_i$  în  $x$  respectiv gradul lui  $w_i$  în  $y$  și se notează cu

$$\text{grad}_{A_i} x = n \quad \text{și} \quad \text{grad}_{w_i} y = n.$$

Are loc următoarea leamnă

LEMA 2.2. *Dacă  $(x, y) \in \Phi_i$  și  $(x, y) \in \Phi_j$ ,  $i \neq j$  și dacă  $x_1$  este un subcuvînt al cuvîntului  $x$  și avem  $\text{grad}_{A_i} x_1 = 1$  și  $\text{grad}_{A_j} x_1 = 1$ , atunci are loc una din subordonările  $\Phi_i < \Phi_j$  sau  $\Phi_j < \Phi_i$ .*

Pentru demonstrație să considerăm  $(x, y) \in \Phi_i$ ,  $(x, y) \in \Phi_j$  și fie  $x_1 \subseteq x$ , astfel ca să avem  $x_1 = t_1 A_i v_1$ ,  $x_1 = t_2 A_j v_2$ ,  $t_1$  și  $t_2$  ne mai conținînd intrări ale categoriilor sintactice  $A_i$ , și respectiv  $A_j$ .

Dacă  $t_1 = t'_1 A_j t'_1$  sau  $v_1 = v'_1 A_j v'_1$ , atunci  $\text{grad}_{A_j} x_1 > 1$  contrar ipotezei. La fel dacă  $t_2 = t'_2 A_i t'_2$  sau  $v_2 = v'_2 A_i v'_2$ , atunci  $\text{grad}_{A_i} x_1 > 1$  contrar ipotezei lemei. Prin urmare categoriile sintactice  $A_i$  și  $A_j$  au numai cîte o intrare în  $x_1$ . Să notăm cu  $l(x)$  lungimea cuvîntului  $x \in W(V)$ , sau cu alte cuvinte numărul simbolurilor conținute în cuvîntul  $x$ . Atunci, în cazul de mai sus, folosind noțiunea de lungime a unui cuvînt, avem una din următoarele inegalități

$$l(t_1) < l(t_2)$$

$$l(t_2) < l(t_1)$$

$$l(t_1) = l(t_2).$$

Dacă  $l(t_1) < l(t_2)$  înseamnă că  $x_1$  admite descompunerea  $x_1 = t'_1 t'_1 A_j v_1$  cu  $l(t'_1) = l(t_2) - l(t_1)$ . Atunci se vede imediat că avem  $t'_1 t'_1 = u_1 A_i u_2$  și conform definiției 2.2 avem  $\Phi_i < \Phi_j$ . În mod analog se ajunge la concluzia că dacă  $l(t_2) < l(t_1)$ , atunci  $\Phi_j < \Phi_i$ .

Dacă  $l(t_1) = l(t_2)$ , atunci vom avea  $x_1 = t_1 A_i v_1$  și  $x_1 = t_2 A_j v_2$  și prin urmare  $t_1 A_i v_1 = t_2 A_j v_2$ . Conform definiției noțiunii de egalitate a două cuvinte, avem  $t_1 = t_2$ . În acest caz,  $A_i = A_j$ , ceea ce contrazice ipoteza  $i \neq j$ . Aceasta demonstrează complet lema 2.2.

DEFINIȚIA 2.5. Dat fiind semigrupul liber  $W(V)$  și o gramatică  $G$ , vom numi început sintactic al unui cuvînt  $x \in W(V)$  relativ la gramatica  $G$ , un subcuvînt  $x_1$  al cuvîntului  $x$  care se bucură de următoarele proprietăți:

(1)  $x = x'_1 x''$ ,  $x'$  și  $x''$  putînd fi cuvintele vide.

(2)  $\exists \Phi_i \in \Phi$ , astfel că  $\Phi_i$  este furnizată de perechea  $(A_i, w_i)$ , unde  $w_i = x_1$ .

(3)  $\text{grad}_{A_i} x' = 0$  și  $\text{grad}_{w_i} x' = 0$  pentru orice pereche  $(A_i, w_i) \in P$ .

(4) Oricare ar fi o altă descompunere a cuvîntului  $x$  care satisface condițiile 1—3, de forma  $x = y'y_1y''$ ,  $y_1 = w_i$  avem  $l(x') < l(y')$ .

DEFINIȚIA 2.6. Dat fiind semigrupul liber  $W(V)$  și o gramatică  $G$ , vom numi sfîrșit sintactic al unui cuvînt  $x \in W(V)$  relativ la gramatica  $G$ , un subcuvînt  $x_1$  al lui  $x$  care are următoarele proprietăți:

(1)  $x = x'x_1x''$ ,  $x'$  și  $x''$  putînd fi cuvintele vide.

(2)  $\exists \Phi_i (\Phi_i \in \Phi)$  astfel că  $\Phi_i$  este furnizată de perechea  $(A_i, w_i)$ , unde  $w_i = x_1$ .

(3)  $\text{grad}_{A_i} x'' = 0$  și  $\text{grad}_{w_i} x'' = 0$  pentru orice pereche  $(A_i, w_i) \in P$ .

(4) Oricare ar fi o altă descompunere a lui  $x$  de forma  $x = y'y_1y''$  care satisface condițiile 1—3 cu  $y_1 = w_j$ ,  $(A_j, w_j) \in P$  avem  $l(x'') < l(y_1')$ .

LEMA 2.3. Dacă un cuvînt  $x \in W(V)$  admite un început (sfîrșit) sintactic, atunci acesta este unic.

Pentru demonstrație fie  $x \in W(V)$  pentru care sînt îndeplinite condițiile definiției 2.5,  $x_1$  fiind începutul său sintactic. Să presupunem că  $x_1$  nu este unic și că mai există încă un început sintactic  $y_1$ , al lui  $x$  astfel ca să avem  $x = y'y_1y''$  și  $x_1 \neq y_1$ , fiind îndeplinite condițiile definiției 2.5 și pentru acesta. Atunci, deoarece  $\text{grad}_{A_i} x' = 0$ ,  $\text{grad}_{A_i} y' = 0$  și  $\text{grad}_{w_i} x' = 0$ ,  $\text{grad}_{w_i} y' = 0$ ,  $x'$  și  $y'$  nu admit începuturi sintactice. Deoarece  $x_1 \neq y_1$ , avem  $l(x') \neq l(y')$ . Atunci sau  $l(x') < l(y')$  și  $y_1$  nu este început sintactic sau  $l(y') < l(x')$  și  $x_1$  nu este început sintactic. Ambele aceste variante contrazic ipoteza de mai sus. De aici rezultă că  $x_1 = y_1$ .

LEMA 2.4. Dacă  $(x, y) \in W(V) \times W(V)$  și  $G$  este o gramatică  $G = (W(V), V_T, P, \{A_1, \dots, A_n | A_i \in V_N\})$  și dacă  $x \Rightarrow y$ , atunci există o mulțime finită de relații  $\Phi_{i_1}, \Phi_{i_2}, \dots, \Phi_{i_p}$  din mulțimea  $\Phi$  astfel ca  $(x, y) \in \Phi_{i_1} \circ \Phi_{i_2} \circ \dots \circ \Phi_{i_p}$ , indicii  $i_j, j = 1, 2, \dots, p$ , nefiind obligatoriu diferiți între ei.

Să presupunem că  $x \Rightarrow y$  și  $(x, y) \in W(V) \times W(V)$ . Atunci, conform definiției 1.2, avem  $x = y$  sau există șirul de cuvinte  $x_0, x_1, \dots, x_p$ , astfel ca

$$x_0 = x$$

$$x_i f_{(A_j, w_j)} = x_{i+1}$$

$$x_p = y$$

pentru anumite aplicații  $f_{(A_j, w_j)}$  definite de perechile  $(A_j, w_j) \in P$ . Dacă  $x = y$ , atunci  $(x, x) \in \Phi_\Delta \in \Phi$ . Dacă  $x \neq y$  și există șirul  $x_0, x_1, \dots, x_p$  astfel ca să fie îndeplinite condițiile din definiția 1.2, atunci avem

$$x_0 = t_1 A_{i_1}, v_1$$

$$x_1 = t_1 w_{i_1}, v_1$$

și deci  $(x_0, x_1) \in \Phi_{i_1}$ ,  $\Phi_{i_1}$  fiind relația furnizată de perechea  $(A_{i_1}, w_{i_1}) \in P$ . În mod analog pentru orice  $j = 0, 1, \dots, p$  avem

$$x_i = t_i A_{i_j} w_i$$

$$x_{i+1} = t_i w_{i_j} w_i$$

și deci  $(x_i, x_{i+1}) \in \Phi_{i_j}$ ,  $\Phi_{i_j}$  fiind relația furnizată de perechea  $(A_{i_j}, w_{i_j}) \in P$ . Dar relațiile  $\Phi_{i_1}, \Phi_{i_2}, \dots, \Phi_{i_p}$  sînt cuprinse toate în mulțimea  $\Phi$ . Rezultă imediat că  $(x_0, x_2) \in \Phi_{i_1} \circ \Phi_{i_2}$  și în general  $(x_0, x_p) \in \Phi_{i_1} \circ \Phi_{i_2} \circ \dots \circ \Phi_{i_p}$ . Deoarece  $x_0 = x$  și  $x_p = y$ , avem  $(x, y) \in \Phi_{i_1} \circ \Phi_{i_2} \circ \dots \circ \Phi_{i_p}$ .

TEOREMA 2.1. Dacă  $L_G$  este un limbaj formalizat definit de gramatica  $G = (W(V), V_T, P, \{A_1, \dots, A_n | A_i \in V_N\})$ , atunci oricare ar fi cuvîntul  $x \in L_G$  și categoria sintactică  $A_i \in V_N$  astfel ca  $A_i \Rightarrow x$ , există un produs de relații  $\Phi_{i_1}, \Phi_{i_2}, \dots, \Phi_{i_p}$ ,  $\Phi_{i_j} \in \Phi$ ,  $j = 1, 2, \dots, p$  astfel ca să avem  $(x, y) \in \Phi_{i_1} \circ \dots \circ \Phi_{i_p}$ .

TEOREMA 2.1 este o consecință directă a lemei 2.4 și a definiției de limbaj formalizat.

LEMA 2.5. Dacă  $x \in L_G$ ,  $L_G$  fiind un limbaj formalizat generat de o gramatică  $G$ , atunci  $x$  admite un început sintactic.

Deoarece  $x \in L_G$ , există o categorie sintactică  $A_i \in V_N$  astfel ca  $A_i \Rightarrow x$  sau  $(A_i, x) \in \Phi_{i_1} \circ \Phi_{i_2} \circ \dots \circ \Phi_{i_p}$ , unde  $\Phi_{i_1}, \Phi_{i_2}, \dots, \Phi_{i_p}$  este o submulțime a mulțimii  $\Phi$  a relațiilor furnizate de  $G$  peste semigrupul liber  $W(V)$ ,  $V = V_N \cup V_T$ ,  $V_N \cap V_T = \emptyset$ . Să presupunem că  $p = 1$ . Atunci  $(A_i, x) \in \Phi_{i_1}$  și deci  $x = t, w_{i_1}, w_i$  fiind începutul sintactic al lui  $x$ . Prin inducție, dacă presupunem lema adevărată pentru un  $p$  oarecare, atunci rezultă ușor că ea este adevărată și pentru  $p + 1$ , ceea ce înseamnă că lema este adevărată pentru orice  $p$ . Pentru a arăta acest lucru să presupunem că  $p > 1$  și că lema este adevărată pentru orice  $q \leq p$ . Atunci avem

$$(A_i, x) \in \Phi_{i_1} \circ \Phi_{i_2} \circ \dots \circ \Phi_{i_p}$$

$$(A_i, z_1) \in \Phi_{i_1}$$

$$(z_1, z_2) \in \Phi_{i_2}$$

$$\dots \dots \dots$$

$$(z_{p-1}, x) \in \Phi_{i_p}$$

Deoarece  $(z_{p-1}, x) \in \Phi_{i_p}$  înseamnă că

$$z_{p-1} = t A_{i_p} v,$$

$$x = t w_{i_p} v.$$

Dacă  $t \in L_G$ , atunci, deoarece  $(A_i, z_{p-1}) \in \Phi_{i_1} \circ \dots \circ \Phi_{i_{p-1}}$ , rezultă că  $t$  are un început sintactic. După cum se vede ușor, acesta va fi și începutul sintactic al lui  $x$ . Dacă  $t \notin L_G$  și nu are început sintactic, atunci  $x = t w_{i_p} v$  iar  $t$  și  $w_{i_p}$  îndeplinesc condițiile definiției 2.5,  $w_{i_p}$  fiind începutul sintactic al lui  $x$ .

DEFINIȚIA 2.7. Dacă  $L_G$  este un limbaj formalizat definit de o gramatică  $G$  în semigrupul liber  $W(V)$ ,  $V = V_N \cup V_T$ ,  $V_N \cap V_T = \emptyset$  și dacă  $x \in L_G$ , atunci dacă  $x$  admite o descompunere de forma  $x = x'x_1x''$  astfel ca să avem  $A_k \Rightarrow x_s$ , spunem că  $x_1$  aparține categoriei sintactice  $A_k$  și notăm acest lucru cu  $x_1 \in [A_k]$ .

LEMA 2.6. Dacă  $L_G$  este un limbaj formalizat definit de gramatica  $G$  în semigrupul liber  $W(V)$ ,  $V = V_N \cup V_T$ ,  $V_N \cap V_T = \emptyset$  și dacă  $x \in L_G$  și conține un subcuvînt  $y \in [A_k]$ , atunci cuvîntul  $x_1$  obținut din  $x$  prin înlocuirea lui  $y$  cu  $A_k$  admite un început sintactic, dacă  $x_1 \neq A_i$ ,  $A_i \in V_N$ .

Deoarece  $x \in L_G$ , există un  $A_j \in V_N$  și relațiile  $\Phi_1, \Phi_2, \dots, \Phi_p$  în mulțimea  $\Phi$  astfel ca să avem  $(A_j, x) \in \Phi_1 \circ \Phi_2 \circ \dots \circ \Phi_p$ . Conform ipotezei lemei avem

$$x = tx'yx''v$$

$$x_1 = tx'A_kx''v.$$

Deoarece  $x_1 \neq A_j$ , rezultă că  $tx' \neq \emptyset$  sau  $x''v \neq \emptyset$ . Să admitem că  $tx' \neq \emptyset$ . Atunci, deoarece  $(A_j, x) \in \Phi_1 \circ \dots \circ \Phi_p$  avem  $p > 1$ . Dacă  $p > 1$ , am avea  $x = w_j$  și  $x_1 = A_j$  contrar ipotezei  $x_1 \neq A_j$ . Înseamnă că  $x_1 = x'y_1x''$ ,  $y \in [A_i]$ ,  $A_i \in V_N$ . Fie  $y_1$  cea mai de la stînga intrare a cuvîntului  $y$  în cuvîntul  $x_1$  cu această proprietate. Atunci  $\text{grad}_{A_i} x'_1 = 0$  și  $\text{grad}_{w_i} x'_1 = 0$ . Aceasta înseamnă că  $y_1$  este începutul sintactic al lui  $x_1$ , ceea ce demonstrează lema.

LEMA 2.7. Fie semigrupul liber  $W(V)$  definit peste o mulțime finită nevidă  $V = V_N \cup V_T$ ,  $V_N \cap V_T = \emptyset$ . O mulțime finită de relații binare  $\Phi_1, \Phi_2, \dots, \Phi_n$  definite în  $W(V)$  de către o mulțime de perechi de forma  $(A_i, w_i)$ ,  $A_i \in V_N$ ,  $w_i \in W(V)$  astfel încît dacă  $(x, y) \in W(V)$ ,  $(x, y) \in \Phi_i$  dacă  $x = t_1 A_i v_1$ ,  $y = t_2 w_i v_2$ , induce în  $W(V)$  o gramatică.

Demonstrația lemei 2.7 este aproape banală. Pentru aceasta este suficient să considerăm mulțimea perechilor  $(A_i, w_i)$  care definesc o gramatică  $G$ , în semigrupul liber  $W(V)$ , chiar mulțimea perechilor  $(A_i, w_i)$  care definesc relațiile  $\Phi_1, \dots, \Phi_n$ .

Lemele 2.1, 2.4, 2.7 și teorema 2.1 ne permit acum să formulăm definițiile noțiunilor de gramatică și de limbaj formalizat în termenii sistemelor de relații.

DEFINIȚIA 2.8. Numim gramatică peste semigrupul liber  $W(V)$ ,  $V = V_N \cup V_T$ ,  $V_N \cap V_T = \emptyset$ ,  $V \neq \emptyset$ , un sistem de relații binare  $\Phi_1, \Phi_2, \dots, \Phi_n$  definite de condițiile următoare: dacă  $x, y \in W(V)$ , atunci  $(x, y) \in \Phi_i$  dacă și numai dacă  $x_1$  admit descompuneri de forma  $x = t A_i V$ ,  $y = t W_i v$ , unde  $A_i \in V_N$ ,  $w_i \in W(V)$ .

Să notăm cu  $\Phi$  o astfel de gramatică.

DEFINIȚIA 2.9. Numim limbaj formalizat generat de gramatica  $\Phi$ , mulțimea cuvintelor semigrupului liber  $W(V_T)$  astfel ca oricare ar fi  $x \in L$ , există un  $A_i \in V_N$  cu proprietatea că

$$(A_i, x) \in \Phi_{i_1} \circ \Phi_{i_2} \circ \dots \circ \Phi_{i_p}, \quad \Phi_{i_j} \in \Phi, \quad j = 1, 2, \dots, p.$$

Prin urmare studiul limbajelor formalizate, în acest fel, revine la studiul sistemelor de relații binare definite peste un semigrup liber  $W(V)$ , unde  $V \neq \emptyset$ ,  $V = V_N \cup V_T$ ,  $V_N \cap V_T = \emptyset$ .

Vom formula acum cele două probleme fundamentale referitoare la limbajele formalizate și anume problema ambiguității unui limbaj și problema analizei unui limbaj în termenii definițiilor de mai sus.

DEFINIȚIA 2.10. Un limbaj  $L_\Phi$  definit peste semigrupul liber  $W(V)$ ,  $V \neq \emptyset$ ,  $V = V_N \cup V_T$ ,  $V_N \cap V_T = \emptyset$  se va numi neambiguu dacă oricare ar fi  $x \in L_\Phi$  și  $A_i \in V_N$  astfel ca  $(A_i, x) \in \Phi_{i_1} \circ \Phi_{i_2} \circ \dots \circ \Phi_{i_p}$  produsul  $\Phi_{i_1} \circ \Phi_{i_2} \circ \dots \circ \Phi_{i_p}$  este unic.

Problema construirii limbajelor formalizate neambigue revine în acest fel la construirea relațiilor  $\Phi_1, \Phi_2, \dots, \Phi_n$  în semigrupul liber  $W(V)$ ,  $V \neq \emptyset$ ,  $V = V_N \cup V_T$ ,  $V_N \cap V_T = \emptyset$  astfel încît oricare ar fi  $A_i \in V_N$

și  $x \in W(V_T)$  dacă există un produs de relații dintre relațiile  $\Phi_1, \Phi_2, \dots, \Phi_n$  astfel ca  $(A_i, x) \in \Phi_{i_1} \circ \dots \circ \Phi_{i_p}$ , atunci acest produs trebuie să fie unic. Noi nu vom studia în această lucrare astfel de sisteme de relații.

DEFINIȚIA 2.11. Dacă  $L_\Phi$  este un limbaj formalizat definit peste semigrupul liber  $W(V)$ ,  $V \neq \emptyset$ ,  $V = V_N \cup V_T$ ,  $V_N \cap V_T = \emptyset$  și dacă  $x \in L_\Phi$ , atunci vom numi analiză a lui  $x$  o mulțime de relații  $\Phi_{i_1}, \Phi_{i_2}, \dots, \Phi_{i_p}$ ,  $\Phi_{i_j} \in \Phi$ ,  $j = 1, 2, \dots, p$ , astfel ca să avem

$$(A_i, x) \in \Phi_{i_1} \circ \Phi_{i_2} \circ \dots \circ \Phi_{i_p}, \quad A_i \in V_N.$$

În felul acesta apare clară legătura dintre cele două probleme fundamentale ale unui limbaj formalizat. Se vede acum limpede că dacă un limbaj formalizat este ambiguu, atunci el nu are o analiză unică și invers, dacă un limbaj formalizat nu are o analiză unică, atunci el este ambiguu.

În cele ce urmează, bazați pe noțiunile construite mai sus, vom descrie un algoritm de analiză pentru un limbaj formalizat.

### § 3. ALGORITM DE ANALIZĂ AL LIMBAJELOR FORMALIZATE

După cum se știe [2], majoritatea algoritmilor de analiză pentru limbajele formalizate s-au născut din necesitatea verificării dacă un anumit cuvînt al unui limbaj formalizat este sau nu corect din punct de vedere sintactic. Acest lucru este foarte important în problema construirii translatoarelor pentru diferite mașini electronice cifrice de calcul.

Pentru construirea algoritmului, în cele ce urmează vom grupa mulțimea relațiilor  $\Phi_1, \Phi_2, \dots, \Phi_n$  care definesc limbajul  $L_\Phi$ , astfel ca să fie posibil să construim algoritmul respectiv ca și un algoritm normal.

Fie  $(A_1, w_1), (A_2, w_2), \dots, (A_n, w_n)$ , perechile care furnizează relațiile  $\Phi_1, \Phi_2, \dots, \Phi_n$ .

DEFINIȚIA 3.1. Spunem că relația  $\Phi_i \in \Phi$  este o relație inițială dacă perechea  $(A_i, w_i)$  care o definește are proprietatea că  $w_i \in W(V_T)$ .

Să considerăm toate relațiile inițiale ale mulțimii  $\Phi$  și să le nume-  
rotăm cu indicii  $i_0, i_1, \dots, i_p$ , astfel ca

$$\Phi_{i_0} \text{ este furnizată de } (A_{i_0}, w_{i_0})$$

$$\Phi_{i_1} \text{ este furnizată de } (A_{i_1}, w_{i_1})$$

$$\dots$$

$$\Phi_{i_p} \text{ este furnizată de } (A_{i_p}, w_{i_p}).$$

Să notăm acum cu  $\Phi_\Delta$  mulțimea acelor relații din  $\Phi$  pentru care dacă  $\Phi_k \in \Phi_\Delta$ , atunci perechea  $(A_k, w_k)$  care furnizează relația  $\Phi_k$  are proprietatea că  $w_k$  posedă intrări ale categoriei sintactice  $A_k$ . Considerăm mulțimea  $\Phi - \Phi_\Delta$  ale cărei relații le plasăm într-o mulțime de grupe de relații  $\mathcal{Q}_0, \mathcal{Q}_1, \dots, \mathcal{Q}_N$  după următorul algoritm:

(1) În grupa  $\mathcal{Q}_0$  plasăm toate relațiile inițiale  $\Phi_{i_0}, \Phi_{i_1}, \dots, \Phi_{i_p}$ , adică acelea pentru care  $w_{i_0}, w_{i_1}, \dots, w_{i_p} \in W(V_T)$ .

(2) Dacă grupa  $\mathcal{Q}_i$  a fost construită, atunci grupa  $\mathcal{Q}_{i+1}$  se construiește din acele relații  $\Phi_{i_0}, \Phi_{i_1}, \dots, \Phi_{i_q}$  care au proprietatea că  $w_{i_0},$

$w_{i_1}, \dots, w_{i_q}$  sînt cuvinte în  $W(V)$  care conțin, ca și intrări ale categoriilor sintactice din  $V_N$ , numai pe acelea care au intrat în relațiile care constituie grupele  $\mathcal{Q}_0, \mathcal{Q}_1, \dots, \mathcal{Q}_{i-1}$ .

(3) Repetăm procesul descris de pasul (2) al algoritmului pînă cînd toate relațiile mulțimii  $\Phi - \Phi_\Delta$  sînt plasate într-o grupă.

Să notăm cu  $N$  numărul grupelor astfel obținute. Atunci avem următoarea lema:

**LEMA 3.1.** *Dacă  $\Phi$  este mulțimea relațiilor binare în semigrupul liber  $W(V)$  care definește gramatica  $G$ , atunci în urma aplicării algoritmului descris mai sus mulțimii  $\Phi - \Phi_\Delta$ , fiecare relație este plasată într-o grupă și numai în una singură.*

Deoarece  $\Phi$  este o mulțime finită,  $\Phi - \Phi_\Delta$  este o mulțime finită și după un număr finit de aplicări ale pasului (2) al algoritmului descris mai sus, fiecare element al mulțimii  $\Phi - \Phi_\Delta$  este examinat. Aceasta înseamnă că fiecare dintre aceste elemente este plasat în cel puțin o grupă  $\mathcal{Q}_i$ . Să arătăm că dacă  $\Phi_i \in \Phi - \Phi_\Delta$ , atunci  $\Phi_i$  nu poate aparține la două grupe distincte. Să presupunem pentru aceasta că  $\Phi_i \in \mathcal{Q}_k$  și  $\Phi_i \in \mathcal{Q}_l$ ,  $k \neq l$ . Atunci avem sau  $k < l$  sau  $l < k$ . Dacă  $k < l$ , atunci perechea  $(A_i, w_i)$  care furnizează  $\Phi_i$  are proprietatea că  $w_i$  conține numai intrări ale categoriilor sintactice care intră în perechile de definiție ale relațiilor din grupele  $\mathcal{Q}_0, \mathcal{Q}_1, \dots, \mathcal{Q}_{k-1}$ . Dacă  $\Phi_i \in \mathcal{Q}_l$ , deoarece  $l > k$ ,  $w_i$  conține intrări ale unor categorii sintactice care intră în definiția relațiilor din grupele  $\mathcal{Q}_k, \mathcal{Q}_{k+1}, \dots, \mathcal{Q}_{l-1}$ , contrar ipotezei care plasează pe  $\Phi_i$  în grupa  $\mathcal{Q}_k$ . Deci în cazul  $k < l$  și  $\Phi_i \in \mathcal{Q}_k$ ,  $\Phi_i$  nu poate aparține grupei  $\mathcal{Q}_l$ .

Să presupunem că  $l > k$  și  $\Phi_i \in \mathcal{Q}_l$ . Atunci  $w_i$  conține intrări ale categoriilor sintactice care intră în definirea relațiilor plasate în grupele  $\mathcal{Q}_0, \mathcal{Q}_1, \dots, \mathcal{Q}_{l-1}$ , altfel  $\Phi_i$  nu ar aparține grupei  $\mathcal{Q}_l$ . Deoarece  $l > k$ , avem  $l > k$  sau  $l-1 = k$ . În ambele cazuri  $\Phi_i$  nu poate aparține grupei  $\mathcal{Q}_k$ , deoarece  $w_i$  conține intrări ale cel puțin uneia din categoriile sintactice care definește o relație din grupa  $\mathcal{Q}_{l-1}$ .

Pentru a extinde algoritmul de ordonare la mulțimea  $\Phi$  în întregime, îl vom completa cu pașii (1'), (2'), (3') pe care îi descriem mai jos, și care au menirea de a plasa relațiile mulțimii  $\Phi_\Delta$  în cel puțin o grupă.

(1') În grupa  $\mathcal{Q}_0$  nu intră nici un element al mulțimii  $\Phi_\Delta$ .

(2') Dacă  $\Phi_k \in \Phi_\Delta$  și dacă, abstracție făcînd de intrările categoriei sintactice  $A_k$  în cuvîntul  $w_k$ , relația  $\Phi_k$  se poate plasa în grupa  $\mathcal{Q}_j$ , atunci o plasăm pe aceasta în grupa  $\mathcal{Q}_j$  ca ultim element al acestei grupe.

(3') Dacă  $\Phi_\Delta = \{\Phi_{k_1}, \Phi_{k_2}, \dots, \Phi_{k_s}\}$ , atunci se aplică pasul (2') fiecăreia din relațiile  $\Phi_{k_i}$ ,  $i = 1, 2, \dots, s$ .

Notăm cu  $\mathfrak{A}_1$  algoritmul descris de pașii (1), (2), (3), (1'), (2'), (3') care va examina toate relațiile mulțimii  $\Phi$  și vom avea următoarea teoremă:

**TEOREMA 3.1.** *Dacă  $\Phi$  este mulțimea relațiilor care definesc o gramatică  $G$  în semigrupul liber  $W(V)$ , atunci  $\mathfrak{A}_1$  plasează elementele mulțimii  $\Phi$  în grupele  $\mathcal{Q}_0, \mathcal{Q}_1, \dots, \mathcal{Q}_N$  astfel încît dacă  $\mathcal{Q}_i$  și  $\mathcal{Q}_j$  sînt oricare două astfel de grupe cu  $i \neq j$ , atunci:  $\mathcal{Q}_i \cap \mathcal{Q}_j = \emptyset$ ,  $\emptyset$  fiind mulțimea vidă.*

Demonstrația teoremei se face în felul următor: dacă  $\Phi_\Delta = \emptyset$ , atunci teorema rezultă din lema 3.1. Dacă  $\Phi_\Delta \neq \emptyset$ , atunci dacă  $\Phi_k \in \Phi_\Delta$ ,

$\Phi_k$  nu poate fi plasat în două grupe distincte. Aceasta rezultă din descrierea pasului (2') al algoritmului  $\mathfrak{A}_1$ , ceea ce demonstrează teorema.

Să considerăm acum mulțimea  $V \neq \emptyset$ ,  $V = V_N \cup V_T$ ,  $V_N \cap V_T = \emptyset$  ca și un alfabet și să construim un algoritm normal  $\mathfrak{A}$  în acest alfabet în felul următor:

(1) Se consideră grupa  $\mathcal{Q}_0$  a relațiilor care definesc gramatica unui limbaj formalizat  $L_G$  în semigrupul liber  $W(V)$  și se stabilește următoarea listă de formule de înlocuire:

$$w_{i_0} \rightarrow A_{i_0}$$

$$w_{i_1} \rightarrow A_{i_1}$$

$$\dots$$

$$w_{i_p} \rightarrow A_{i_p}.$$

În mod analog se procedează pentru grupele  $\mathcal{Q}_1, \mathcal{Q}_2, \dots, \mathcal{Q}_N$ . Considerăm atunci algoritmul normal în alfabetul  $V$  cu formulele de înlocuire stabilite mai sus și să-l notăm cu  $\mathfrak{A}$ .

Fie relațiile  $\Phi_1, \Phi_2, \dots, \Phi_N$  furnizate de gramatica  $G$  și ordonate după algoritmul  $\mathfrak{A}_1$ . Notăm cu  $\Phi_{N_1}, \Phi_{N_2}, \dots, \Phi_{N_k}$  toate relațiile grupei  $\mathcal{Q}_N$ .

**DEFINIȚIA 3.2.** Limbajul formalizat  $L_G$  generat de gramatica  $G$  în semigrupul liber  $W(V)$  se numește CF-limbaj dacă perechile  $(A_{N_1}, W_{N_1}), (A_{N_2}, W_{N_2}), \dots, (A_{N_k}, W_{N_k})$  care definesc relațiile  $\Phi_{N_1}, \Phi_{N_2}, \dots, \Phi_{N_k}$  au proprietatea că  $A_{N_1}, A_{N_2}, \dots, A_{N_k}$  reprezintă aceeași categorie sintactică din  $V_N$ . Să notăm cu  $S$  categoria sintactică respectivă.

**DEFINIȚIA 3.3.** Limbajul formalizat  $L_G$  generat de gramatica  $G$  în semigrupul liber  $W(x)$  se numește CF-limbaj generalizat dacă perechile  $(A_{N_1}, W_{N_1}), (A_{N_2}, W_{N_2}), \dots, (A_{N_k}, W_{N_k})$  care definesc relațiile  $\Phi_{N_1}, \Phi_{N_2}, \dots, \Phi_{N_k}$  au proprietatea că  $A_{N_1}, A_{N_2}, \dots, A_{N_k}$  nu reprezintă toate aceeași categorie sintactică din  $V_N$ .

Să considerăm acum un CF-limbaj  $L_G$  și să presupunem că relațiile furnizate de gramatica  $G$  în  $W(V)$  au fost ordonate cu ajutorul algoritmului  $\mathfrak{A}_1$ . Considerăm toate lanțurile de perechi  $(A_{N_i}, w_{N_i}), (w_{N_i}, w_{N_{i-1}}), \dots, (w_{N_{i-r_i+1}}, w_{N_{i-r_i}})$  astfel ca

$$w_{N_{i-r_i}} = t_1 w_1 t_2 w_2 \dots t_n w_n t_{n+1}$$

$$w_i \in W(V_N), i = 1, 2, \dots, n$$

$$t_i \in W(V_T), i = 1, 2, \dots, n+1$$

$$\exists i (w_i \neq \emptyset \text{ \& } t_i \neq \emptyset)$$

Dacă notăm cu  $S$  categoria sintactică reprezentată de  $A_{N_i}$ ,  $i = 1, 2, \dots, k$ , atunci se vede imediat că relațiile furnizate de  $(A_{N_i}, w_{N_i}), (w_{N_i}, w_{N_{i-1}}), \dots, (w_{N_{i-r_i+1}}, w_{N_{i-r_i}})$  adică  $\Phi_{N_i}, \Phi_{N_{i-1}}, \dots, \Phi_{N_{i-r_i}}$  au proprietatea că produsul  $\Phi_{N_i} \circ \Phi_{N_{i-1}} \circ \dots \circ \Phi_{N_{i-r_i}}$ ,  $i = 1, 2, \dots, k$  conține acele elemente  $x, y \in W(V)$  care se obțin din derivarea lui  $S$ ,  $y$  avînd forma  $t_1 w_1 t_2 w_2 \dots t_n w_n t_{n+1}$ . Adăugăm acum (ca și următoarele formule de înlocuire la algoritmul normal  $\mathfrak{A}$ ) toate formulele de forma

$$t_1 S t_2 S \dots t_n S t_{n+1} \rightarrow S.$$

Să notăm cu  $\mathfrak{A}$ , algoritmul normal astfel obținut. Atunci are loc următoarea teoremă:

**TEOREMA 3.2.** *Dacă  $L_G$  este un CF-limbaj și dacă  $x \in L_G$ , atunci  $\mathfrak{A}_v(x) = S$ ,  $S \in V_N$ ,  $S$  fiind categoria sintactică unică care apare în partea stângă a perechilor  $(A_{N_i}, w_{N_i})$ , care formează grupa  $\mathcal{Q}_N$ .*

Să presupunem că  $x \in L_G$ . Conform lemei 2.5, rezultă că el admite un început sintactic. Fie acesta  $t_1$ . Atunci  $x = x' t_1 x''$ ,  $t_1 = w_i$ ,  $w_i \in W(V_T)$ . Aceasta înseamnă că există o formulă de înlocuire în schema algoritmului  $\mathfrak{A}$  astfel ca această formulă să se aplice cuvîntului  $x$  și să-l transforme în cuvîntul  $x_1$ , unde  $x_1 = x' A_i x''$ .

Conform lemei 2.6, cuvîntul  $x_1$  va admite un început sintactic. Dacă acesta este cuvîntul  $t_2 \in W(V_T)$ , atunci din construcția algoritmului  $\mathfrak{A}$  urmează că există o formulă cu proprietatea că se aplică cuvîntului  $x$  înlocuind cuvîntul  $t_2$  cu categoria sintactică  $A_k$ ,  $t_2 \in [A_k]$ . În felul acesta după un număr finit de pași algoritmul  $\mathfrak{A}$  va înlocui toate intrările cuvintelor  $x_i \in W(V_T)$  din cuvîntul  $x$  cu categoriile sintactice la care aparțin acestea.

Conform lemei 2.5, dacă cuvîntul  $x_p$  astfel obținut este diferit de o categorie sintactică  $A_j$ ,  $A_j \in V_N$ , atunci el admite un început sintactic și prin urmare există o formulă în schema algoritmului  $\mathfrak{A}$  care înlocuiește începutul său sintactic cu categoria sintactică la care aparține. Se observă că lungimea cuvîntului  $x$  fiind finită, după un număr finit de pași cuvîntul  $x$  se transformă fie într-o categorie sintactică  $S \in V_N$ , și atunci el este un cuvînt al limbajului  $L_G$ , fie într-un cuvînt  $x_k$ ,  $l(x_k) < l(x)$  astfel că  $x_k$  nu mai admite început sintactic printre formulele algoritmului  $\mathfrak{A}$ , dar el admite început sintactic printre formulele algoritmului  $\mathfrak{A}_v$  de forma

$$t_1 S t_2 S \dots t_n S t_{n+1}.$$

În acest caz se aplică cuvîntului  $x_k$  formula pentru care s-a pus în evidență începutul sintactic al cuvîntului, înlocuindu-se  $t_1 S t_2 S \dots t_n S t_{n+1}$  cu  $S$ . Dacă după un număr finit de aplicări a pașilor algoritmului  $\mathfrak{A}_v$  se obține un cuvînt  $x_p$  astfel ca el să difere de  $S$  și nici o formulă a algoritmului  $\mathfrak{A}_v$  nu se aplică la acesta, atunci  $x \in L_G$ . Dacă  $x_p$  este  $S$ , atunci  $x \in L_G$ . Pentru a demonstra aceste două variante este suficient să considerăm în ordine inversă șirul relațiilor furnizate de perechile care au definit algoritmul normal și care s-au aplicat la cuvîntul  $x$  și pînă la obținerea cuvîntului  $x_k$ . Fie acestea  $\Phi_1, \Phi_2, \dots, \Phi_r$ . Să considerăm apoi în ordine inversă șirul relațiilor care sînt furnizate de formulele algoritmului care au condus de la  $x_k$  la  $x_p$ . Fie acestea  $\Phi_{r+1}, \Phi_{r+2}, \dots, \Phi_s$ . Atunci este vizibil că

$$(S, x) \in \Phi_1 \circ \Phi_2 \circ \dots \circ \Phi_s$$

și deci  $x \in L_G$ .

Dacă se ajunge la cuvîntul  $x_p \neq S$ , atunci într-un mod asemănător se arată că nu există o derivare a lui  $S$  care să conducă la  $x$  și prin urmare  $x \in L_G$ .

Să observăm acum că dacă algoritmul descris mai sus îl modificăm în așa fel încît la fiecare pas începutul sintactic (sau sfîrșitul sintactic) al cuvîntului obținut la etapa precedentă aplicării algoritmului să fie înlocuit

cu categoria sintactică la care aparține acesta, atunci se obține chiar șirul transformărilor normale [2] al cuvîntului respectiv.

Considerăm acum următorul exemplu: fie  $V = \{S, A, B, C, t, *, (, ), \lambda\}$ , unde  $V_N = \{S, A, B, C\}$  și  $V_T = \{+, *, (, ), \lambda\}$  și gramatica furnizată de perechile

$$(S, A)$$

$$(A, A + B)$$

$$(A, B)$$

$$(B, B * C)$$

$$(C, (A))$$

$$(C, \lambda).$$

Algoritmul de ordonare va aranja aceste perechi (sau relațiile furnizate de ele) sub următoarea formă:

$$\mathcal{Q}_0 = \{(C, \lambda)\}$$

$$\mathcal{Q}_1 = \{(B, C), (B, B * C)\}$$

$$\mathcal{Q}_2 = \{(A, B), (A, A + B)\}$$

$$\mathcal{Q}_3 = \{(C, (A))\}$$

$$\mathcal{Q}_4 = \{(S, A)\}.$$

Algoritmul normal  $\mathfrak{A}$  corespunzător acestei ordonări va fi

$$\lambda \rightarrow C$$

$$C \rightarrow B$$

$$B * C \rightarrow B$$

$$B \rightarrow A$$

$$A + B \rightarrow A$$

$$(A) \rightarrow C$$

$$A \rightarrow S.$$

Deoarece pornind de la elementele grupei  $\mathcal{Q}_4$  putem construi lanțurile de forma

$$(1) (S, A), (A, A + B)$$

$$(2) (S, A), (A, B), (B, B * C)$$

$$(3) (S, A), (A, B), (B, C), (C, (A))$$

algoritmul normal  $\mathcal{A}$ , va avea schema

$$\begin{aligned}\lambda &\rightarrow C \\ C &\rightarrow B \\ B * C &\rightarrow B \\ B &\rightarrow A \\ A + B &\rightarrow A \\ (A) &\rightarrow C \\ A &\rightarrow S \\ S + S &\rightarrow S \\ S * S &\rightarrow S \\ (S) &\rightarrow S\end{aligned}$$

Dacă considerăm cuvîntul  $(x + \lambda) * \lambda$ , atunci aplicînd acestui cuvînt algoritmul  $\mathcal{A}$ , precizat de formulele de înlocuire de mai sus obținem :

- (1)  $(C + C) * C$  în urma aplicării primei formule.
- (2)  $(B + B) * B$  în urma aplicării celei de-a doua formule.
- (3)  $(A + A) * B$  în urma aplicării celei de-a patra formule.
- (4)  $(S + S) * S$  în urma aplicării celei de-a șaptea formule.
- (5)  $(S) * S$  în urma aplicării celei de-a opta formule.
- (6)  $S * S$  în urma aplicării celei de-a zecea formule.
- (7)  $S$  în urma aplicării celei de-a doua formule.

Se vede deci că  $(\lambda + \lambda) * \lambda \in L_G$ .

Primită la redacție la 20 mai 1966

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

#### BIBLIOGRAFIE

1. N. CHOMSKY, *On certain formal properties of grammars*. Inf. and control, **2**, 2, 137—167 (1959).
2. I. EICKEL, M. PAUL, EL. BAUER, K. SAMUELSON, *A syntax controled generator of formal language procesor*. Comm. A.C.M., **7**, 8, 451—455 (1963).
3. R.W. FLOYD, *Bounded context syntactic analysis*. Comm. A.C.M. **7**, 2, 62—67 (1964).
4. E. I. IRONS, *Structurale connected in the formal languages*. Comm. A.C.M. **7**, 2, 67—71 (1964).
5. Mc.KINSEY, *Postulates for the calculus of binar relations*. Journal of the Symbolic Logic, **6**, 3, 85—97 (1941).
6. A. TARSKY, *On the calculus of relations*. Journal of the Symbolic Logic. **6**, 3, 73—85 (1941).
7. I. PURDEA, *Généralisation du produit de deux relations binaires*. Mathematica.