

ASUPRA UNUI SISTEM FORMAL. II¹⁾

DE
T. RUS

În această lucrare vom construi câțiva algoritmi care să decidă asupra compatibilității cuvintelor în sistemul formal construit în [5].

În primul rând vom considera cazul când în cuvântul P dat în sistemul formal nu există intrări ale simbolurilor din $\mathcal{C} \cup \mathcal{R}$. În acest caz se poate ușor vedea că algoritmul nominal $\mathfrak{A}_1$ cu schema

$$\mathfrak{A}_1 \left\{ \begin{array}{l} \alpha \xi_1 \xi_2 \rightarrow \cdot \alpha \xi_1 \xi_2 \quad (\xi_1 \in \mathcal{C}, \xi_2 \in \mathcal{R}) \\ \alpha \xi_1 \xi_2 \rightarrow \cdot \alpha \xi_1 \xi_2 \quad (\xi_1 \in \mathcal{R}, \xi_2 \in \mathcal{C}) \\ \alpha \xi \rightarrow \alpha \quad (\xi \in \mathcal{C} \cup \mathcal{R}) \\ \alpha \rightarrow \cdot \\ \rightarrow \alpha \quad (\xi \in \mathcal{A}) \end{array} \right.$$

în alfabetul $\mathcal{A} \cup \{\alpha\}$ aplicat la oricare cuvânt P în alfabetul $\mathcal{A}$ transformă acest cuvânt în cuvântul vid dacă el este compatibil în $\mathcal{A}$ și nu conține intrări ale simbolurilor din $\mathcal{C} \cup \mathcal{R}$ și într-un cuvânt R diferit de cuvântul vid în caz contrar.

Pentru demonstrarea acestei afirmații este suficient să observăm că dacă în cuvântul P nu avem intrări ale simbolurilor din $\mathcal{C} \cup \mathcal{R}$, el este compatibil dacă și numai dacă este sau un cuvânt în $\mathcal{C}$ sau un simbol în alfabetul $\mathcal{C} \cup \mathcal{R}$.

Prima formulă care se aplică cuvântului P este formula $\rightarrow \alpha$ care transformă cuvântul P în cuvântul αP . Pe urmă simbolul α se „trage” prin cuvântul P ștergînd succesiv fiecare simbol (literă) al alfabetului $\mathcal{C}$ care apare în P , dacă nu există intrări ale altor simboluri din alfabetul $\mathcal{A}$ în P . Dacă cuvântul P conține intrări ale altor simboluri din $\mathcal{A}$, diferite de literele alfabetului $\mathcal{C}$, atunci pentru a fi compatibil el trebuie

¹⁾ Lucrare prezentată la Colectivul de aproximare a funcțiilor cu aplicații la calculul numeric din 15–19 noiembrie 1963, Cluj.

să conțină un singur simbol din $\mathcal{X}$ care se șterge cu ajutorul formulei $\alpha\xi \rightarrow \alpha$. Algoritmul se oprește fie semnalizând apariția în P a simbolurilor din $\mathcal{C}$ și din $\mathcal{X}$ fără nici un simbol de legătură între ele, fie semnalizând că în P apar simboluri diferite de simbolurile alfabetului $\mathcal{C} \cup \mathcal{X}$. Dacă algoritmul se oprește în acest de-al doilea caz (ceea ce se recunoaște după faptul că în cuvântul obținut nu intră simbolul α) și este diferit de cuvântul vid, se va aplica de la început algoritmul $\mathcal{U}_2$ care va transforma cuvântul P în cuvântul vid dacă el este compatibil și conține intrări de simboluri diferite de simbolurile alfabetului $\mathcal{C} \cup \mathcal{X}$ sau într-un cuvânt diferit de cuvântul vid în caz contrar.

Schema algoritmului normal $\mathcal{U}_2$ în alfabetul $\mathcal{A} \cup \{\alpha\} \cup \{\beta\}$ este următoarea :

$$\mathcal{U}_2 \left\{ \begin{array}{ll} \alpha((\rightarrow (\alpha(\\ \alpha(Q) \rightarrow \alpha & \left(\begin{array}{l} Q \sqsubseteq \xi_0 \xi_1 \dots \xi_K, \quad \xi_i \in \mathcal{C} \text{ dacă } K > 0 \\ \text{sau } Q \sqsubseteq \xi, \quad \xi \in \mathcal{C} \cup \mathcal{X} \text{ dacă } K = 0 \end{array} \right) \\ \alpha\xi \rightarrow \xi\alpha & (\xi \in \mathcal{C} \cup \mathcal{X}) \\ \alpha\xi \rightarrow \beta\alpha & (\xi \in \mathcal{C} \cup \mathcal{X}) \\ (\beta) \rightarrow & (\beta \notin \alpha, \quad \beta, \alpha \in \mathcal{A}) \\ \alpha \rightarrow & \\ \beta \rightarrow & \\ \rightarrow \alpha & \end{array} \right.$$

Să demonstrăm că are loc una din egalitățile :

$$\mathcal{U}_2 \sqsubseteq P \sqsubseteq \sqsubseteq \left\{ \begin{array}{l} \Lambda \quad \text{dacă } P \text{ este compatibil în } \mathcal{A} \text{ și conține intrări de} \\ \text{simboluri diferite de simbolurile alfabetului lui } \mathcal{C} \cup \mathcal{X} \\ R \notin \Lambda \text{ în caz contrar.} \end{array} \right.$$

Fie deci cuvântul P compatibil în alfabetul $\mathcal{A}$ și conținând și simboluri diferite de simbolurile alfabetului $\mathcal{C} \cup \mathcal{X}$. Să arătăm atunci că $\mathcal{U}_2 \sqsubseteq P \sqsubseteq \sqsubseteq \Lambda$.

Prima formulă a algoritmului normal $\mathcal{U}_2$ care se va aplica cuvântului P va fi formula $\rightarrow \alpha$. După aceea prima formulă a algoritmului $\mathcal{U}_2$ se va aplica de atâtea ori pînă ce se va găsi primul element de bază care intră în cuvântul P . Deoarece cuvântul P este compatibil, acest element de bază intră în cuvântul P sub forma (Q) , în condițiile noastre. Se va aplica apoi formula a doua care va înlocui cuvântul (Q) cu simbolul α și se trece la căutarea altor elemente de bază care intră în cuvântul P .

În felul acesta, după un număr finit de pași, se vor scoate toate elementele de bază care intră în cuvântul P , iar după formula a patra toate simbolurile din $\mathcal{C} \cup \mathcal{X}$ care intră în cuvântul P se vor înlocui cu simbolul β . Formula a cincea va înlocui apoi toate cuvintele de forma (β) cu cu-

vinte vid, ceea ce este echivalent cu ștergerea tuturor simbolurilor din $\mathcal{C} \cup \mathcal{X}$ care aparțin domeniului de acțiune al altor simboluri din $\mathcal{C} \cup \mathcal{X}$ care intră în P . Deoarece cuvântul P este compatibil, înseamnă că există un singur simbol din $\mathcal{C} \cup \mathcal{X}$ care nu aparține domeniului de acțiune al nici unui alt simbol din $\mathcal{C} \cup \mathcal{X}$ care intră în cuvântul P . Prin urmare dacă primele cinci formule nu se mai aplică, atunci formula a șasea $\alpha \rightarrow$ va scoate unicul simbol α care apare în cuvântul P transformat iar formula a șaptea $\beta \rightarrow \cdot$ va înlocui simbolul din $\mathcal{C} \cup \mathcal{X}$ care nu aparține domeniului de acțiune al nici unui alt simbol din $\mathcal{C} \cup \mathcal{X}$ care intră în P cu cuvântul vid, final, obținînd în felul acesta cuvântul vid.

Să arătăm acum că dacă cuvântul P nu este compatibil, are loc egalitatea $\mathcal{U}_2 \sqsubseteq P \sqsubseteq \sqsubseteq R$, $R \notin \Lambda$.

Această egalitate rezultă din faptul că dacă în cuvântul P nu există intrări ale simbolurilor din $\mathcal{C} \cup \mathcal{X}$, atunci deoarece cuvântul conține intrări de simboluri diferite de simbolurile din $\mathcal{C} \cup \mathcal{X}$, după un număr finit de pași se va aplica de un număr infinit de ori ultima formă, ceea ce înseamnă că la un astfel de cuvânt algoritmul nu se aplică. Dacă însă în cuvântul P există intrări ale simbolurilor din $\mathcal{C} \cup \mathcal{X}$, atunci oricare ar fi cauza incompatibilității algoritmul se oprește după aplicarea formulei $\beta \rightarrow \cdot$, semnalînd această incompatibilitate prin faptul că cuvântul obținut este un cuvânt diferit de cuvântul vid.

Cuvântul R , obținut în acest fel, poate fi sau un cuvânt în alfabetul $\{\beta\}$ și atunci sîntem în cazul în care există mai multe simboluri din $\mathcal{C} \cup \mathcal{X}$ care intră în P și care nu aparțin domeniului de acțiune al nici unui alt simbol din $\mathcal{C} \cup \mathcal{X}$ care intră în P , sau poate fi un cuvânt în alfabetul $\mathcal{A}$ ceea ce înseamnă că există simboluri diferite de simbolurile din $\mathcal{C} \cup \mathcal{X}$ care nu intră în domeniul de acțiune al nici unui simbol din $\mathcal{C} \cup \mathcal{X}$ care intră în P , sau poate fi un cuvânt în alfabetul $\mathcal{A} \cup \{\beta\}$ ceea ce înseamnă că au loc ambele incompatibilități semnalate mai sus. În oricare din cazurile analizate mai sus, P nu este un cuvânt compatibil.

Să construim acum un algoritm normal care va decide dacă într-un cuvânt dat în alfabetul $\mathcal{A}$ intră sau nu intră simboluri din alfabetul $\mathcal{C} \cup \mathcal{X}$. Dacă notăm cu $\mathcal{U}_3$ acest algoritm, atunci schema lui este

$$\mathcal{U}_3 \left\{ \begin{array}{ll} \alpha\xi \rightarrow \alpha & (\xi \in \mathcal{C} \cup \mathcal{X}), \\ \alpha \rightarrow \cdot & \\ \rightarrow \alpha & (\alpha \in \mathcal{A}). \end{array} \right.$$

Se vede imediat că au loc egalitățile

$$\mathcal{U}_3 \sqsubseteq P \sqsubseteq \sqsubseteq \left\{ \begin{array}{l} \Lambda \text{ dacă în } P \text{ nu există intrări ale simbolurilor din } \mathcal{C} \cup \mathcal{X}, \\ \notin \Lambda \text{ în caz contrar.} \end{array} \right.$$

Algoritmul care decide dacă un cuvânt arbitrar P în alfabetul $\mathcal{A}$ este compatibil sau nu în acest alfabet, este atunci algoritmul normal $\mathfrak{A}$ construit astfel ca

$$\mathfrak{A} \perp P \perp \overline{\alpha} \begin{cases} \mathfrak{A}_1 \perp P \perp \text{ dacă } \mathfrak{A}_3 \perp P \perp \overline{\alpha} \Lambda, \\ \mathfrak{A}_2 \perp P \perp \text{ dacă } \mathfrak{A}_3 \perp P \perp \not\subseteq \Lambda. \end{cases}$$

Se poate demonstra ușor că

$$\mathfrak{A} \perp P \perp \overline{\alpha} \begin{cases} \Lambda \text{ dacă } P \text{ este un cuvânt compatibil în } \mathcal{A}, \\ R \not\subseteq \Lambda \text{ în caz contrar.} \end{cases}$$

Vom construi în cele ce urmează un algoritm normal care ne va pune în evidență într-un cuvânt compatibil simbolul $s \in \mathcal{O} \cup \mathfrak{R}$ care nu aparține domeniului de acțiune al nici unui alt simbol din $\mathcal{O} \cup \mathfrak{R}$ care intră în cuvântul compatibil dat. Pentru aceasta este suficient să observăm că în schema algoritmului normal $\mathfrak{A}_2$ formula de oprire $\beta \rightarrow \cdot$ scoate cu oprire chiar acest simbol din cuvântul compatibil dat. Prin urmare algoritmul căutat în alfabetul $\mathcal{A} \cup \{\alpha\}$ va avea schema

$$\begin{cases} \alpha((\rightarrow (\alpha(\\ \alpha(Q) \rightarrow \alpha & (Q - \text{același ca și în cazul algoritmului } \mathfrak{A}_2) \\ \alpha\xi \rightarrow \xi\alpha & (\xi \in \mathcal{A}) \\ (s) \rightarrow & (s \in \mathcal{O} \cup \mathfrak{R}) \\ \alpha \rightarrow \cdot \\ \rightarrow \alpha & (\alpha \in \mathcal{A}) \end{cases}$$

Dacă notăm cu $\mathfrak{B}_1$, acest algoritm, atunci din cele de mai sus avem

$$\mathfrak{B}_1 \perp P \perp \overline{\alpha} \begin{cases} s, & \text{unde } s \text{ este simbolul din } \mathcal{O} \cup \mathfrak{R} \text{ care nu aparține do-} \\ & \text{menului de acțiune al nici unui alt simbol din } \mathcal{O} \cup \mathfrak{R} \\ & \text{care intră în } P, \text{ dacă } P \text{ este compatibil,} \\ R \not\subseteq s & \text{în caz contrar.} \end{cases}$$

Deoarece însă în cuvântul P există în general intrări ale aceluiași simbol s în mai multe locuri, evident că pentru a preciza acest simbol avem nevoie de aripa dreapta, respectiv stângă a intrării lui în P , adică trebuie să cunoaștem cuvintele P_1 și P_2 astfel ca

$$P \overline{\alpha} P_1 s P_2, \quad P_1 \overline{\alpha} (P'_1), \quad P_2 \overline{\alpha} (P'_2).$$

Pentru determinarea cuvintelor P_1 și P_2 ne vom folosi de noțiunea de lungime a unui cuvânt. Prin lungimea unui cuvânt vom înțelege numărul simbolurilor (literelor) care compun cuvântul respectiv. Astfel dacă notăm cu $l \perp P \perp$ lungimea cuvântului P , avem

$$l \perp \wedge \perp \overline{\alpha} 0, \\ l \perp \xi_1 \dots \xi_K \perp \overline{\alpha} K$$

unde 0 și K sînt scrierile numărului zero respectiv K_2 .

Vom construi în cele ce urmează algoritmul normal $\mathfrak{B}_2$ astfel ca:

$$\mathfrak{B}_2 \perp P \perp \overline{\alpha} Q_1 s Q_2$$

unde

$$l \perp Q_1 \perp \overline{\alpha} l \perp P_1 \perp, \quad l \perp Q_2 \perp \overline{\alpha} l \perp Q_2 \perp$$

iar Q_1 și Q_2 sînt cuvinte în alfabetul $\{|\}$.

Schema algoritmului normal $\mathfrak{B}_2$ în alfabetul $\mathcal{A} \cup \{\alpha\} \cup \{|\}$ este atunci următoarea:

$$\mathfrak{B}_2 \begin{cases} \alpha((\rightarrow (\alpha(, \\ \alpha(Q) \rightarrow |Q'| \alpha & \begin{cases} (Q' - \text{cuvînt în } \{|\} \text{ și } l \perp Q' \perp \overline{\alpha} l \perp Q \perp, \\ Q \text{ fiind același ca și în algoritmul } \mathfrak{B}_1, \end{cases} \\ \alpha\xi \rightarrow \xi\alpha & (\xi \in \mathcal{A}), \\ (| \rightarrow |(& (| \in \mathcal{A}), \\ |) \rightarrow |) \\ (\xi) \rightarrow ||| & (\xi \in \mathcal{O} \cup \mathfrak{R}), \\ \alpha \rightarrow \cdot \\ \rightarrow \alpha & \alpha \in \mathcal{A} \cup \{|\}. \end{cases}$$

Se poate vedea simplu că algoritmul normal $\mathfrak{B}_2$ astfel construit satisface condițiile cerute.

Cuvintele Q_1 și Q_2 în alfabetul $\{|\}$ ne vor servi în cele ce urmează pentru a pune în evidență cuvintele P_1 și P_2 aripa stînga, respectiv dreapta a simbolului s din cuvîntul P .

Pentru aceasta vom construi algoritmi normali $\mathfrak{R}_1$ și $\mathfrak{R}_2$ cu proprietatea că

$$\mathfrak{R}_1 \perp |Q_1 * P \perp \overline{\alpha} P_2,$$

$$\mathfrak{R}_2 \perp |Q_2 * P \perp \overline{\alpha} P_1.$$

Fie $A = \mathcal{A} \cup \{\alpha\}$ alfabetul obținut prin adăugarea la alfabetul $\mathcal{A}$ al sistemului formal a simbolului α . Presupunem că $*$ $\in \mathcal{S}$ și deci $*$ $\in \mathcal{A}$. Vom considera alfabetul $B \overline{\alpha} A \cup \{|\}$ și vom construi în acest alfabet algoritmi normali [3]

$$\mathfrak{R} \begin{cases} \alpha | \rightarrow \cdot \\ \alpha * \xi \rightarrow \alpha * & (\xi \in \mathcal{A}) \\ \alpha * \rightarrow \\ \rightarrow \alpha \end{cases}$$

$$\mathfrak{C} \begin{cases} \alpha | \rightarrow \alpha \\ \alpha * \rightarrow \cdot \\ \rightarrow \alpha \end{cases}$$

$$\mathcal{D} \begin{cases} * \xi \rightarrow * \\ * \rightarrow \\ \mathcal{F} \{ \mid \rightarrow \cdot \\ \mathcal{G} \{ * \rightarrow \cdot \end{cases}$$

Construim pe urmă compozițiile normale $\mathcal{F} \circ \mathcal{D}$ și $\mathcal{U}_1 \circ \mathcal{G}$ respectiv $\mathcal{U}_2 \circ \mathcal{G}$ unde algoritmi normalii $\mathcal{U}_1$ și $\mathcal{U}_2$ sînt dați de schemele

$$\mathcal{U}_1 \begin{cases} \xi \rightarrow \cdot & (\xi \in \mathcal{A}) \\ \beta \xi \rightarrow \xi \beta & (\xi \in \mathcal{A}, \beta \in \mathcal{A}) \\ \xi \beta \rightarrow \cdot & \\ \rightarrow \beta & \end{cases}$$

Construim în continuare algoritmi normalii $\mathcal{S}_1$ și $\mathcal{S}_2$ după teorema reuniunii [3] astfel ca:

$$\mathcal{S}_1 \sqsubset R \sqsubset \simeq \sqsubset \mathcal{F} \circ \mathcal{D} \sqsubset \sqsubset R \sqsubset * \sqsubset \mathcal{U}_1 \circ \mathcal{G} \sqsubset \sqsubset R \sqsubset$$

$$\mathcal{S}_2 \sqsubset R \sqsubset \simeq \sqsubset \mathcal{F} \circ \mathcal{D} \sqsubset \sqsubset R \sqsubset * \sqsubset \mathcal{U}_2 \circ \mathcal{G} \sqsubset \sqsubset R \sqsubset$$

unde simbolurile $\simeq$ și $\circ$ nu fac parte din alfabetul $\mathcal{A}$, ele avînd sensul precizat în [3].

Vom construi apoi algoritmi normalii $\mathcal{N}_1$ și $\mathcal{N}_2$ peste alfabetul algoritmilor $\mathcal{S}_1$ și $\mathcal{S}_2$ în așa fel ca $\mathcal{N}_1$ și $\mathcal{N}_2$ să posedă următoarele proprietăți:

$\mathcal{N}_1$ respectiv $\mathcal{N}_2$ atunci și numai atunci prelucrează cuvîntul R în cuvîntul S cînd există șirul cuvintelor $R_0, R_1, \dots, R_n$ ($n \geq 0$) care satisfac condițiile

$$\begin{aligned} R_0 &\sqsupseteq R, \\ R_i &\sqsupseteq \mathcal{S}_1 \sqsubset R_{i-1} \sqsubset \text{ respectiv } R_i \sqsupseteq \mathcal{S}_2 \sqsubset R_{i-1} \sqsubset, \\ R_n &\sqsupseteq S, \\ \mathcal{S} \sqsubset R_i \sqsubset &\not\sqsupseteq \Lambda \quad (0 \leq i < n), \\ \mathcal{S} \sqsubset R_n \sqsubset &\not\sqsupseteq \Lambda. \end{aligned}$$

Algoritmi $\mathcal{N}_1$ și $\mathcal{N}_2$ vor fi atunci dați de compozițiile normale $\mathcal{G} \circ \mathcal{N}_1$ și respectiv $\mathcal{G} \circ \mathcal{N}_2$ și prin urmare

$$\begin{aligned} \mathcal{N}_1 &\simeq \mathcal{G} \circ \mathcal{N}_1, \\ \mathcal{N}_2 &\simeq \mathcal{G} \circ \mathcal{N}_2. \end{aligned}$$

Să demonstrăm acum că au loc următoarele egalități

$$\begin{aligned} \mathcal{N}_1 \sqsubset | Q_1 * P \sqsubset \sqsupseteq P_2 \\ \mathcal{N}_2 \sqsubset | Q_2 * P \sqsubset \sqsupseteq P_1 \end{aligned}$$

astfel ca $P = P_1 s P_2$, s fiind simbolul care nu aparține domeniului de acțiune al nici unui alt simbol din $\mathcal{O} \cup \mathcal{A}$ care intră în P , P fiind un cuvînt compatibil în alfabetul $\mathcal{A}$.

Pentru demonstrarea acestor egalități să observăm că cuvintele $|Q_1$ și $|Q_2$ pot fi privite ca numere naturale scrise în alfabetul $\{|\}$ pe care în continuare le putem nota prin n respectiv m .

Atunci se vede direct din schemele algoritmilor că

$$\begin{aligned} \mathcal{S} \sqsubset n * P \sqsubset \sqsupseteq \sqsubset n - 1 \sqsubset * P, \\ \mathcal{S} \sqsubset * P \sqsubset \sqsupseteq \Lambda, \\ \mathcal{D} \sqsubset n * P \sqsubset \sqsupseteq n, \\ \mathcal{G} \sqsubset n * P \sqsubset \sqsupseteq P, \\ \mathcal{F} \sqsubset n \sqsubset \sqsupseteq \sqsubset n - 1 \sqsubset, \\ \mathcal{G} \sqsubset * P \sqsubset \sqsupseteq P. \end{aligned}$$

De asemenea se vede ușor că:

$$\mathcal{S}_1 \sqsubset n * P \sqsubset \sqsupseteq \mathcal{F} \sqsubset \mathcal{D} \sqsubset n * P \sqsubset \sqsupseteq * \mathcal{U}_1 \sqsubset \mathcal{G} \sqsubset n * P \sqsubset \sqsupseteq \sqsubset n - 1 \sqsubset * \mathcal{U}_1 \sqsubset P \sqsubset$$

și în mod analog

$$\mathcal{S}_2 \sqsubset m * P \sqsubset \sqsupseteq \sqsubset m - 1 \sqsubset * \mathcal{U}_2 \sqsubset P \sqsubset,$$

unde $m - 1$ și respectiv $n - 1$ înseamnă numărul natural cu o unitate mai mic decît m respectiv n sau cuvinte în alfabetul $\{|\}$ a căror lungime sînt cu o unitate mai mică ca a cuvintelor de lungime m respectiv n .

Considerînd acum în calitate de cuvinte R_i necesare în definiția algoritmilor $\mathcal{N}_1$ respectiv $\mathcal{N}_2$ cuvintele

$$R_i \sqsupseteq \sqsubset n - 1 \sqsubset * P_i \quad (0 \leq i \leq n) \text{ pentru } \mathcal{N}_1$$

și

$$R_i \sqsupseteq \sqsubset m - 1 \sqsubset * P_i \quad (0 \leq i \leq m) \text{ pentru } \mathcal{N}_2,$$

unde prin P_i am notat cuvîntul obținut după a i -a aplicare a algoritmilor $\mathcal{N}_1$ respectiv $\mathcal{N}_2$ și atunci

$$\begin{aligned} R_0 &\sqsupseteq n * P \text{ sau } R_0 \sqsupseteq | Q_1 * P \\ R_n &\sqsupseteq * P_2 \end{aligned}$$

pentru algoritmul $\mathcal{N}_1$ și

$$\begin{aligned} R_0 &\sqsupseteq m * P \text{ sau } R_0 \sqsupseteq | Q_2 * P \\ R_m &\sqsupseteq P_1 * \end{aligned}$$

pentru algoritmul $\mathcal{N}_2$. Aceasta deoarece $\mathcal{U}_1 \sqsubset \xi P \sqsubset \sqsupseteq P$ iar $\mathcal{U}_2 \sqsubset P \xi \sqsubset \sqsupseteq P$, unde ξP respectiv $P \xi$ sînt cuvinte în $\mathcal{A}$.

Se vede imediat că

$$\mathcal{N}_1 \sqsubset | Q_1 * P \sqsubset \sqsupseteq * P_2$$

și

$$\mathcal{N}_2 \sqsubset | Q_2 * P \sqsubset \sqsupseteq P_1 *$$



și deci

$$\mathfrak{R}_1 \sqsubset |Q_1 * P \sqsubset \simeq \mathfrak{G} \sqsubset \mathfrak{R}_1 \sqsubset |Q_1 * P \sqsubset \simeq \mathfrak{G} \sqsubset * P_2 \sqsubset \sqsupseteq P_2,$$

$$\mathfrak{R}_2 \sqsubset |Q_2 * P \sqsubset \simeq \mathfrak{G} \sqsubset \mathfrak{R}_2 \sqsubset |Q_2 * P \sqsubset \simeq \mathfrak{G} \sqsubset P_1 * \sqsubset \sqsupseteq P_1.$$

Cuvintele P_1 și P_2 sînt însă astfel că $P_1 \sqsupseteq (P'_1)$ iar $P_2 \sqsupseteq (P'_2)$ după cum am observat mai sus. Dacă asupra cuvintelor P_1 și P_2 ne-am pune aceeași problemă ca și asupra cuvîntului P , atunci pentru noi ar fi important de cunoscut de fapt cuvintele P'_1 și P'_2 . Pentru a obține aceste cuvinte noi putem construi algoritmul normal $\mathfrak{S}$ care șterge prima și ultima literă a unui cuvînt. Schema acestui algoritm normal în alfabetul $\mathcal{A} \cup \{\alpha\} \cup \{\beta\}$ este:

$$\mathfrak{S} \begin{cases} \alpha \xi \rightarrow \xi \alpha & (\xi \in \mathcal{A}), \\ \xi \alpha \rightarrow \beta, \\ \xi \beta \rightarrow \beta \xi, \\ \beta \xi \rightarrow \cdot, \\ \rightarrow \alpha. \end{cases}$$

Construind acum compoziția normală a algoritmilor $\mathfrak{S}$ și $\mathfrak{R}_1$ respectiv $\mathfrak{R}_2$, $\mathfrak{S} \circ \mathfrak{R}_1$, $\mathfrak{S} \circ \mathfrak{R}_2$ și notînd acești algoritmi normali (compozițiile) prin $\mathfrak{R}'_1$ respectiv $\mathfrak{R}'_2$ vom avea:

$$\mathfrak{R}'_1 \sqsubset |Q_1 * P \sqsubset \simeq \gamma \sqsubset \mathfrak{R}_1 \sqsubset |Q_1 * P \sqsubset \sqsupseteq P'_1,$$

$$\mathfrak{R}'_2 \sqsubset |Q_2 * P \sqsubset \simeq \mathfrak{S} \sqsubset \mathfrak{R}_2 \sqsubset |Q_2 * P \sqsubset \sqsupseteq P'_2.$$

Pentru a rezolva însă complet problema găsirii cuvintelor P_1 și P_2 trebuie să mai construim o serie de algoritmi care să ne ducă de la cuvîntul P la cuvîntul $|Q_1 * P$ și respectiv $|Q_2 * P$.

Pentru aceasta este suficient să observăm că algoritmul $\mathfrak{B}_2$ are proprietatea că $\mathfrak{B}_2 \sqsubset P \sqsubset \sqsupseteq Q_1 s Q_2$.

Construind algoritmi $\mathfrak{B}_3$ și $\mathfrak{B}_4$ în alfabetul $\mathcal{A} \cup \{\cdot\} \cup \{\alpha\}$ cu schemele:

$$\mathfrak{B}_3 \begin{cases} \alpha | \rightarrow | \alpha \\ \alpha s \rightarrow | s \\ s | \rightarrow s \\ s \rightarrow \cdot \\ \rightarrow \alpha \end{cases}$$

$$\mathfrak{B}_4 \begin{cases} \alpha | \rightarrow \alpha \\ \alpha s \rightarrow \cdot | \\ \rightarrow \alpha \end{cases}$$

Se vede imediat că

$$\mathfrak{B}_3 \sqsubset Q_1 s Q_2 \sqsubset \sqsupseteq |Q_1$$

iar

$$\mathfrak{B}_4 \sqsubset Q_1 s Q_2 \sqsubset \sqsupseteq |Q_2.$$

Compozițiile normale $\mathfrak{B}_3 \circ \mathfrak{B}_2$ și $\mathfrak{B}_4 \circ \mathfrak{B}_2$ aplicate la cuvîntul P nu vor duce prin urmare la cuvintele $|Q_1$ și respectiv $|Q_2$. Dacă construim acum după teorema reuniunii [3] algoritmi $\mathfrak{B}_6$ și $\mathfrak{B}_7$ astfel ca

$$\mathfrak{B}_6 \sqsubset P \sqsubset \simeq \sqsubset \mathfrak{B}_3 \circ \mathfrak{B}_2 \sqsubset \sqsubset P \sqsubset * \mathfrak{B}_5 \sqsubset P \sqsubset,$$

$$\mathfrak{B}_7 \sqsubset P \sqsubset \simeq \sqsubset \mathfrak{B}_4 \circ \mathfrak{B}_2 \sqsubset \sqsubset P \sqsubset * \mathfrak{B}_5 \sqsubset P \sqsubset,$$

unde $\mathfrak{B}_5$ este algoritmul identic, atunci se vede că algoritmi normali

$$\mathfrak{R}_1 \circ \mathfrak{B}_6 \text{ și } \mathfrak{R}_2 \circ \mathfrak{B}_7$$

au proprietatea că

$$\sqsubset \mathfrak{R}_1 \circ \mathfrak{B}_6 \sqsubset \sqsubset P \sqsubset \sqsupseteq P_1, P_1 \sqsupseteq (P'_1),$$

$$\sqsubset \mathfrak{R}_2 \circ \mathfrak{B}_7 \sqsubset \sqsubset P \sqsubset \sqsupseteq P_2, P_2 \sqsupseteq (P'_2),$$

unde $P \sqsupseteq P_1 s P_2$ sau folosindu-se de algoritmi $\mathfrak{R}'_1$ și $\mathfrak{R}'_2$ avem

$$\sqsubset \mathfrak{R}'_1 \circ \mathfrak{B}_6 \sqsubset \sqsubset P \sqsubset \sqsupseteq P'_1,$$

$$\sqsubset \mathfrak{R}'_2 \circ \mathfrak{B}_7 \sqsubset \sqsubset P \sqsubset \sqsupseteq P'_2.$$

Vom nota în cele ce urmează algoritmi $\sqsubset \mathfrak{R}'_1 \circ \mathfrak{B}_6 \sqsubset$ și $\mathfrak{R}'_2 \circ \mathfrak{B}_7$ respectiv cu $\mathfrak{M}_1$ și $\mathfrak{M}_2$.

Construind apoi algoritmul normal $\mathfrak{P}$ după teorema reuniunii astfel ca

$$\mathfrak{P} \simeq \mathfrak{B}_1 \beta \mathfrak{M}_1 \alpha \mathfrak{M}_2.$$

După cum se vede din definiția algoritmului $\mathfrak{P}$ avem

$$\mathfrak{P} \sqsubset P \sqsubset \sqsupseteq s \beta P_1 \alpha P_2.$$

Să notăm cu $\mathfrak{P}_{\mathcal{O} \cup \mathcal{A}}^P$ proiecția cuvîntului P pe alfabetul $\mathcal{O} \cup \mathcal{A}$. Dacă avem

$$l \sqsubset \mathfrak{M}_1 \sqsubset P_1 \sqsubset \sqsupseteq l \sqsubset \mathfrak{M}_2 \sqsubset P_2 \sqsubset,$$

atunci

$$\mathfrak{P} \simeq \mathfrak{B}_1 \beta \mathfrak{M}_1 \alpha \mathfrak{M}_2 \text{ (a)}$$

iar dacă

$$l \sqsubset \mathfrak{M}_1 \sqsubset P_1 \sqsubset \sqsupseteq l \sqsubset \mathfrak{M}_2 \sqsubset P_2 \sqsubset$$

atunci

$$\mathfrak{P} \simeq \mathfrak{B}_1 \sigma \alpha \mathfrak{M}_2 \beta \mathfrak{M}_1$$

$$\sigma \sqsupseteq \mathcal{A}, \sigma \not\sqsupseteq \alpha, \beta, \delta, \gamma, \rho. \text{ (b)}$$

Se vede deci că avem

$$\mathfrak{P} \sqsubset P \sqsubset \sqsupseteq s \beta P'_1 \alpha P'_2$$

în primul caz și

$$\mathfrak{P} \sqsubset P \sqcup \subseteq s \sigma \alpha P'_2 \beta P'_1,$$

în al doilea caz.

Admitem în cele ce urmează că avem de-a face numai cu cazul în care algoritmul $\mathfrak{P}$ este definit de echivalența (a). Dacă el este definit de echivalența (b), atunci pentru a nu complica metasimbolismul schimbăm notațiile algoritmilor $\mathfrak{M}_1$ și $\mathfrak{M}_2$ respectivi, adică pe $\mathfrak{M}_1$ îl notăm cu $\mathfrak{M}_2$ iar pe $\mathfrak{M}_2$ îl notăm cu $\mathfrak{M}_1$. De aceste precizări vom ține seama de asemenea mai jos și în definirea algoritmilor $\mathfrak{P}_i$.

Se vede că $\mathfrak{P} \sqsubset P \sqcup \subseteq s \beta P'_1 \alpha P'_2$ sau $\mathfrak{P} \sqsubset P \sqcup = s \sigma \alpha P'_2 \beta P'$, unde $s \in \mathcal{O} \cup \mathcal{A}$, $s \uparrow \in D_{s_i}$, s_i fiind toate celelalte simboluri din $\mathcal{O} \cup \mathcal{A}$ care intră în cuvîntul P iar P'_1 și P'_2 sînt astfel ca

$$P \subseteq P_1 s P_2, \quad P_1 \subseteq (P'_1)_1, \quad P_2 \subseteq (P'_2)_1.$$

Simbolul s astfel obținut îl vom numi simbol de ordinul întâi, și-l vom nota cu s'_1 . De asemenea cuvintele P'_1 și P'_2 vor fi numite cuvinte de ordinul întâi. Se știe [5] că într-un cuvînt compatibil există unul și numai un simbol de ordinul întâi. Aplicînd din nou algoritmul $\mathfrak{P}$ cuvintelor P'_1 și P'_2 respectiv, vom obține

$$\mathfrak{P} \sqsubset P'_1 \sqcup \subseteq s \beta P''_1 \alpha P''_2,$$

$$\mathfrak{P} \sqsubset P'_2 \sqcup \subseteq s \beta P''_3 \alpha P''_4.$$

Simbolurile s obținute în felul acesta se vor numi simboluri de ordinul doi și se vor nota respectiv cu s'_1 și s'_2 . Dacă algoritmul $\mathfrak{P}$ nu se aplică la unul din cuvintele P'_1 sau P'_2 , atunci acesta este un element de bază al sistemului formal, adică un cuvînt în $\mathcal{C}$ sau un simbol în $\mathcal{O} \cup \mathcal{A}$ iar unul din simbolurile s'_i pot lipsi. (Se poate vedea simplu că ambele simboluri s'_i pot lipsi, ceea ce este echivalent cu faptul că ambele cuvinte P'_1 și P'_2 sînt elemente de bază.)

Pentru a decide dacă cuvintele P'_1 respectiv P'_2 sînt sau nu elemente de bază vom aplica acestor cuvinte algoritmul $\mathfrak{B}$ în alfabetul $\mathcal{A}$ care are schema

$$\mathfrak{B} \begin{cases} \xi \rightarrow (\xi \uparrow \in \mathcal{O} \cup \mathcal{A}, \xi \in \mathcal{A}), \\ \xi \rightarrow \cdot \xi (\xi \in \mathcal{O} \cup \mathcal{A}). \end{cases}$$

Putem acum construi algoritmi $\mathfrak{P}_1$ respectiv $\mathfrak{P}_2$ echivalenți cu algoritmi $\mathfrak{P} \sqsubset P'_1 \sqcup$ respectiv $\mathfrak{P} \sqsubset P'_2 \sqcup$ dacă avem

$$\mathfrak{B} \sqsubset P'_1 \sqcup \not\subseteq \Lambda, \quad \mathfrak{B} \sqsubset P'_2 \sqcup \not\subseteq \Lambda$$

cu ajutorul compunerii algoritmilor $\mathfrak{P}$ și $\mathfrak{M}_1$ respectiv $\mathfrak{M}_2$ și echivalenți cu algoritmul identic $\mathfrak{B}_5$ în caz contrar.

Vom avea deci

$$\mathfrak{P}_1 \sqsubset P \sqcup \simeq \begin{cases} \mathfrak{B} \sqsubset \mathfrak{M}_1 \sqsubset P \sqcup \sqcup & \text{dacă } \mathfrak{B} \sqsubset \mathfrak{M}_1 \sqsubset P \sqcup \sqcup \not\subseteq \Lambda \\ \mathfrak{B}_5 \sqsubset \mathfrak{M}_1 \sqsubset P \sqcup \sqcup & \text{în caz contrar} \end{cases}$$

$$\mathfrak{P}_2 \sqsubset P \sqcup \simeq \begin{cases} \mathfrak{B} \sqsubset \mathfrak{M}_2 \sqsubset P \sqcup \sqcup & \text{dacă } \mathfrak{B} \sqsubset \mathfrak{M}_2 \sqsubset P \sqcup \sqcup \not\subseteq \Lambda, \\ \mathfrak{B}_5 \sqsubset \mathfrak{M}_2 \sqsubset P \sqcup \sqcup & \text{în caz contrar.} \end{cases}$$

În mod absolut analog se construiesc apoi algoritmi $\mathfrak{P}_1^2, \mathfrak{P}_2^2, \mathfrak{P}_3^2, \mathfrak{P}_4^2$ astfel ca :

$$\mathfrak{P}_i^2 \sqsubset P \sqcup \simeq \begin{cases} \mathfrak{B} \sqsubset \mathfrak{M}_k \sqsubset \mathfrak{M}_j \sqsubset P \sqcup \sqcup \sqcup & \text{dacă } \mathfrak{B} \sqsubset \mathfrak{M}_k \sqsubset \mathfrak{M}_j \sqsubset P \sqcup \sqcup \sqcup \not\subseteq \Lambda \\ \mathfrak{B} \sqsubset \mathfrak{M}_k \sqsubset \mathfrak{M}_j \sqsubset P \sqcup \sqcup \sqcup & \text{în caz contrar} \end{cases}$$

$$i = 1, 2, 3, 4; \quad k = 1, 2; \quad j = 1, 2.$$

Se vede ușor că dacă notăm cu p un ordin oarecare al simbolurilor s obținute, respective al algoritmilor $\mathfrak{P}_i^p$, atunci numărul maxim de asemenea simboluri, respectiv numărul maxim de algoritmi $\mathfrak{P}_i^p$ diferiți de algoritmi identici va fi cel mult 2^p .

Deoarece cuvîntul P inițial conține un număr finit de simboluri și deci nu număr finit de intrări ale simbolurilor din $\mathcal{O} \cup \mathcal{A}$ după un număr finit de pași, ceea ce echivalează cu construirea unui număr finit de algoritmi de forma $\mathfrak{P}_i^p$ diferiți de algoritmul identic $\mathfrak{B}_5$, toate cuvintele de acel ordin vor fi elemente de bază ale sistemului formal. Să notăm cu μ ordinul acestor cuvinte. Atunci evident că șirul algoritmilor $\{\mathfrak{P}_i^p\}$, $\max i = 2^p$ iar $p = 0, 1, 2, \dots, \mu$, $\mathfrak{P}_0^0 \subseteq \mathfrak{P}$, va transforma cuvîntul inițial P într-un șir de cuvinte de forma $s \beta Q \alpha S$, unde fiecare dintre cuvintele Q și S sînt sau cuvinte compatibile în sistemul formal care conține intrări de simboluri din $\mathcal{O} \cup \mathcal{A}$ sau sînt elemente de bază ale sistemului formal, după cum algoritmi $\mathfrak{P}_i^p$ care se aplică la aceste cuvinte sînt diferiți de algoritmul identic $\mathfrak{B}_5$ sau sînt chiar algoritmul identic $\mathfrak{B}_5$. Se poate acum construi algoritmul $\mathfrak{U}$ pentru alfabetele tuturor algoritmilor $\mathfrak{P}_i^p$ astfel ca

$$\mathfrak{U} \simeq \mathfrak{P} * \mathfrak{P}_1^1 \mathfrak{P}_2^1 * \dots * \mathfrak{P}_1^p \mathfrak{P}_2^p \dots \mathfrak{P}_{2^p}^p * \dots * \mathfrak{P}_1^\mu \dots \mathfrak{P}_{2^\mu}^\mu.$$

Se vede ușor că

$$\mathfrak{U} \sqsubset P \sqcup \subseteq s'_1 \beta P'_1 \alpha P'_2 * \dots * s'_1^\mu \beta P_1^\mu \alpha P_2^\mu \dots s_{2^\mu}^\mu \beta P_{2^\mu-1}^\mu \alpha P_{2^\mu}^\mu,$$

unde P_i^μ sînt toate elementele de bază. Dacă unul dintre $s'_i \in \mathcal{C}_1$, atunci cuvîntul

$$s'_i \beta P_i^j P_{i+1}^j$$

este astfel că

$$s'_i \beta P_i^j \alpha P_{i+1}^j \subseteq s'_i \beta P_i^j \alpha \wedge.$$

Bineînțeles că printre cuvintele P_i^j , $j < \mu$ pot să existe cuvinte care sînt elemente de bază ale sistemului formal.

Din cele de mai sus rezultă că are loc următoarea afirmație pentru cuvintele compatibile în sistemul formal.

Oricare ar fi cuvântul compatibil P în alfabetul $\mathcal{A}$, există un algoritm normal Π care transformă cuvântul D într-un șir de sisteme de cuvinte astfel ca:

- 1) cuvântul P se numește cuvânt de ordinul zero;
- 2) Cuvintele de ordin maxim ale sistemului sînt elemente de bază ale sistemului formal;

3) Oricare ar fi cuvântul R de ordin v , ($0 \leq v < \mu$), el se obține cel mult din două cuvinte de ordinul $v + 1$ cu ajutorul uneia din regulile de formare [5] și cu ajutorul unui simbol $s \in \mathcal{O} \cup \mathcal{R}$ de ordinul v .

Prin urmare, oricărui cuvânt compatibil P în alfabetul $\mathcal{A}$ i se poate atașa în mod unic un arbore A [6] cu următoarele proprietăți:

- 1° Toate nodurile arborelui A sînt simboluri ale alfabetului $\mathcal{O} \cup \mathcal{R}$.
- 2° Toate muchiile arborelui A sînt cuvinte compatibile în alfabetul $\mathcal{A}$.
- 3° Toate vîrfurile arborelui A sînt elemente de bază ale alfabetului $\mathcal{A}$.
- 4° Toate nodurile corespunzătoare simbolurilor din $\mathcal{O} \cup \mathcal{R}$ de același ordin se găsesc pe o dreaptă.

Dacă notăm cu d_i dreapta nodurilor corespunzătoare simbolurilor de ordin i și d_j a celor de ordin j , atunci d_i și d_j sînt paralele.

Arborele A corespunzător cuvântului compatibil P într-o scriere liniară simbolică nu este altceva decît rezultatul aplicării algoritmului Π la cuvântul P dacă în fața acestui rezultat punem cuvântul P_0^0 , unde $P_0^0 \subseteq P^*$, și dacă în definiția lui Π nu intră decît algoritmi diferiți de $\mathcal{B}_5$.

Separatorii $*$ care apar în acest rezultat marchează ordinul nodurilor arborelui respectiv, iar separatorii α și β indică respectiv muchia dreapta și stînga care pleacă din nodul din stînga simbolului β . Se observă simplu că din fiecare nod al arborelui A pleacă cel mult trei muchii.

Unicitatea arborelui A corespunzător cuvântului compatibil P rezultă din faptul că oricare ar fi algoritmul normal $\Pi_1 \simeq \Pi$ peste alfabetul $\mathcal{A}$, $\Pi_1 \sqsubseteq P \sqsubseteq \Pi$ egalitatea înțelegîndu-se în sensul grafic introdus în [3].

Dacă ne folosim de un metasimbolism special, care nu păstrează forma liniară a cuvintelor despre care se raționează cu el, atunci cu ajutorul acestui metasimbolism, arborele A corespunzător cuvântului compatibil P se poate reprezenta sub forma dată în figură, unde liniile orizontale punctate marchează ordinele simbolurilor s care se găsesc pe acestea, iar $\max k_\mu = 2^\mu$.

DEFINIȚIE. Spunem că cuvântul P este echivalent cu cuvântul Q relativ la alfabetul $\mathcal{A}$ dacă sînt îndeplinite condițiile:

1) Dacă cuvântul P nu conține intrări ale simbolurilor din $\mathcal{O} \cup \mathcal{R}$, atunci nici Q nu conține intrări ale simbolurilor din $\mathcal{O} \cup \mathcal{R}$ și $P \subseteq Q$.

2) Dacă cuvântul P conține intrări ale simbolurilor din $\mathcal{O} \cup \mathcal{R}$, atunci și Q conține intrările unor astfel de simboluri în așa fel încît dacă $s \in \mathcal{O} \cup \mathcal{R}$ intră în P , atunci s intră și în Q și reciproc.

3) Dacă s-au definit domeniile de acțiune ale simbolului s în cuvântul P și în cuvântul Q , notate prin D_s^P și respectiv D_s^Q , atunci oricare ar fi elementul de bază P_1 , care intră în D_s^P , el intră și în D_s^Q și reciproc.

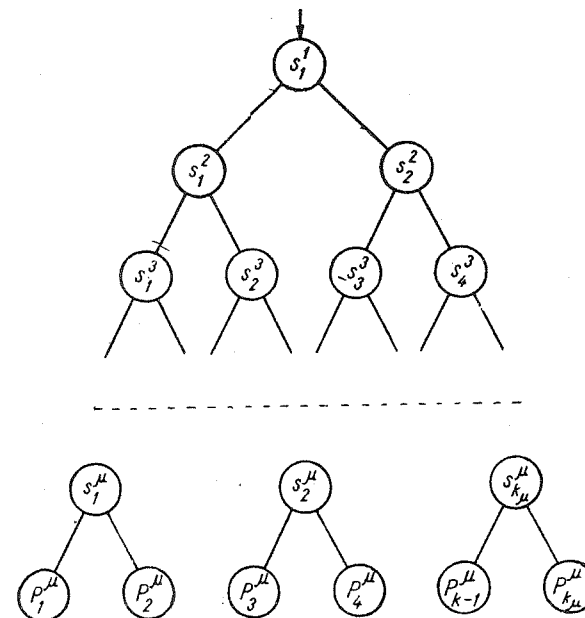


Fig. 1

4) Dacă cuvântul P are un denotat, atunci și cuvântul Q are un denotat și aceste denotate coincid.

Notînd cu $\sim$ simbolul echivalenței introduse, vom avea

$$P \sim P. \quad (a)$$

$$\text{Dacă } P \sim Q, \text{ atunci } Q \sim P. \quad (b)$$

$$\text{Dacă } P \sim Q \text{ și } Q \sim R, \text{ atunci } P \sim R. \quad (c)$$

TEOREMĂ. Oricare ar fi cuvântul compatibil P în alfabetul $\mathcal{A}$ există un singur cuvânt Q de forma

$$Q \subseteq Q_1 * Q_2 * \dots * Q_\mu$$

echivalent cu P , Q_i ($i = 1, 2, \dots, \mu$) fiind cuvinte în alfabetul

$$\mathcal{C} \cup \mathcal{X} \cup \mathcal{O} \cup \mathcal{R} \cup \{\alpha, \beta, \sigma\}$$

α, β, σ sînt diferite între ele și $\alpha, \beta, \sigma \in \mathcal{A}$ astfel ca:

1) $\mathcal{O}_\mu$ este cuvînt în alfabetul $\mathcal{C} \cup \mathcal{X} \cup \mathcal{O} \cup \mathcal{R}$ de forma

$$Q_\mu \subseteq s_1 P_1 R_1 \dots s_{k_\mu} P_{k_\mu} R_{k_\mu},$$

P_i și R_i ($i = 1, 2, \dots, k_\mu$) fiind elemente de bază ale sistemului formal astfel încât dacă $s_i \in \mathcal{O}_2 \cup \mathcal{R}$, atunci $P_i, R_i \notin \Lambda$ iar dacă $s_i \in \mathcal{O}_1$, atunci $P_i \notin \Lambda$ iar $R_i \in \Lambda$, acesta din urmă lipsind din Q_μ .

2) Q_i , $i < \mu$ este cuvânt în alfabetul

$$\mathcal{O} \cup \mathcal{X} \cup \mathcal{O} \cup \mathcal{R} \cup \{\alpha, \beta, \sigma\}$$

astfel ca :

Dacă β și α intră în Q_i , atunci ele pot intra în acest cuvânt numai sub forma $s\beta P_1$ respectiv $s\alpha P_1$, unde P_1 este un element de bază al sistemului formal iar $s \in \mathcal{O} \cup \mathcal{R}$ sau $s \in \sigma$.

Dacă P_1 este element de bază al sistemului formal și intră în Q_i și dacă el nu este precedat în acest cuvânt de β sau α , atunci el poate intra în acest cuvânt numai sub una din formele sP_1R_1 sau sR_1P_1 dacă $s \in \mathcal{O}_2 \cup \mathcal{R}$, R_1 fiind un alt element de bază care intră în Q_i , sau sP_1 dacă $s \in \mathcal{O}_1$.

3) Simbolurile din $\mathcal{O} \cup \mathcal{R}$ care intră în cuvântul Q_i ($i = 1, 2, \dots, \mu$) sînt acelea și numai acelea care au ordinul i în cuvântul P și sînt așezate de la stînga la dreapta în ordinea în care apare în arborele A atașat cuvîntului P .

Să demonstrăm în primul rînd că cuvîntul Q precizat în această teoremă este unic. Pentru aceasta să presupunem că nu ar fi unic și că ar exista și cuvîntul Q' de forma indicată mai sus astfel ca $P \sim Q'$. Se vede atunci imediat că $Q \sim Q'$. Prin urmare dacă

$$Q \subseteq Q_1 * \dots * Q_\mu$$

și

$$Q' \subseteq Q'_1 * \dots * Q'_\mu,$$

atunci din proprietățile cuvintelor Q_μ și Q'_μ , rezultă că $Q_\mu \subseteq Q'_\mu$, iar din faptul că în cuvîntul P există un singur ordin cel mai înalt al simbolurilor din $\mathcal{O} \cup \mathcal{R}$ care intră în el avem $\mu = \mu'$.

Suprimînd acum cuvintele Q_μ și Q'_μ din cuvintele Q și Q' , obținem cuvintele Q_1^s și $Q_1'^s$ de forma:

$$Q_1^s \subseteq Q_1 * \dots * Q_{\mu-1},$$

$$Q_1'^s \subseteq Q'_1 * \dots * Q'_{\mu-1}.$$

Să arătăm că $Q_1^s \sim Q_1'^s$ iar $Q_{\mu-1} \subseteq Q'_{\mu-1}$.

Prima afirmație este evidentă deoarece $Q \sim Q'$, iar din aceste cuvinte pentru a obține cuvintele Q_1^s și $Q_1'^s$ am suprimat aceleași intrări de simboluri. Prin urmare rămîne de arătat că $Q_{\mu-1} \subseteq Q'_{\mu-1}$.

Pentru aceasta să considerăm cuvintele

$$\mathfrak{Q}_{\mathcal{O} \cup \mathcal{R}}^{\mu-1}, \mathfrak{Q}_{\mathcal{O} \cup \mathcal{X}}^{\mu-1}, \mathfrak{Q}_{\{\alpha, \beta, \sigma\}}^{\mu-1}$$

și

$$\mathfrak{Q}_{\mathcal{O} \cup \mathcal{R}}^{\mu-1}, \mathfrak{Q}_{\mathcal{O} \cup \mathcal{X}}^{\mu-1}, \mathfrak{Q}_{\{\alpha, \beta, \sigma\}}^{\mu-1}.$$

Din cauza condiției (3) din teoremă și din faptul că ordinul unui simbol $s \in \mathcal{O} \cup \mathcal{R}$ care intră în P este unic determinat, rezultă imediat egalitatea

$$\mathfrak{Q}_{\mathcal{O} \cup \mathcal{R}}^{\mu-1} \subseteq \mathfrak{Q}_{\mathcal{O} \cup \mathcal{R}}^{\mu-1}.$$

Să arătăm că avem de asemenea și

$$\mathfrak{Q}_{\mathcal{O} \cup \mathcal{X}}^{\mu-1} \subseteq \mathfrak{Q}_{\mathcal{O} \cup \mathcal{X}}^{\mu-1}.$$

Să presupunem că nu ar fi așa. Atunci, evident, există cel puțin un element de bază al sistemului formal care intră în $Q_{\mu-1}$ și nu intră în $Q'_{\mu-1}$ sau invers. Să presupunem pentru precizarea ideilor că P_1 este elementul de bază respectiv și că el intră în $Q_{\mu-1}$ dar nu intră în $Q'_{\mu-1}$. El poate intra în cuvîntul $Q_{\mu-1}$ numai sub una din formele

$$sP_1R_1, sR_1P_1, s\beta P_1, s\alpha P_1, sP_1.$$

În oricare din aceste cazuri dacă $s \in \mathcal{O} \cup \mathcal{R}$, este de ordinul $\mu - 1$ în cuvîntul P . Prin urmare el intră și în cuvîntul $Q'_{\mu-1}$. Deoarece P_1 nu intră în cuvîntul $Q'_{\mu-1}$, rezultă că un simbol $s \in \mathcal{O} \cup \mathcal{R}$ are în cuvîntul P pe de o parte o intrare de una din formele

$$(P_1) s(R_1), (R_1) s(P_1), s(P_1), (P_1) s(\cdot) s(P_1)$$

după cum se vede din definiția lui D_s^P și aceeași intrare a simbolului s poate avea și o altă formă, deoarece $Q_1^s \sim Q_1'^s$. De aici va rezulta că domeniul de acțiune al simbolului s nu este unic în cuvîntul P ceea ce este în contradicție cu faptul că P fiind compatibil este bine construit.

Prin urmare dacă P_1 intră în $Q_{\mu-1}$, el intră și în $Q'_{\mu-1}$ și reciproc, de unde avem $\mathfrak{Q}_{\mathcal{O} \cup \mathcal{X}}^{\mu-1} \subseteq \mathfrak{Q}_{\mathcal{O} \cup \mathcal{X}}^{\mu-1}$. În mod analog va rezulta că $\mathfrak{Q}_{\{\alpha, \beta, \sigma\}}^{\mu-1} \subseteq \mathfrak{Q}_{\{\alpha, \beta, \sigma\}}^{\mu-1}$.

Ținînd acum seama de-a doua parte a condiției 3) a teoremei rezultă imediat că $Q_{\mu-1} \subseteq Q'_{\mu-1}$.

Procedînd acum cu cuvintele Q_1^s și $Q_1'^s$ ca și cu cuvintele Q și respectiv Q' , vom ajunge, după un număr μ de pași, la egalitatea $Q_1 \subseteq Q_1'$, de unde va rezulta că $Q \subseteq Q'$, ceea ce demonstrează unicitatea cuvîntului Q .

Pentru demonstrarea existenței cuvîntului Q să facem în prealabil cîteva observații.

Observația 1. După cum se vede din definiția echivalenței, dacă cuvîntul P este un element de bază al sistemului formal, atunci Q este și el un element de bază al sistemului formal și avem $P \subseteq Q$ în sensul introdus în [3].

Observația 2. Dacă cuvîntul P este de forma $(P_1)s(P_2)$ sau $s(P_1)$, unde P_1 și P_2 sau P_1 , sînt element de bază ale sistemului formal, atunci rezultă evident din definiția echivalenței că cuvîntul Q în acest caz trebuie să conțină dintre elementele de bază ale sistemului formal numai elementele P_1 și P_2 sau P_1 iar dintre simbolurile din $\mathcal{O} \cup \mathcal{R}$ numai simbolul s , al cărui domeniu de acțiune în cuvîntul Q trebuie să conțină elementele

P_1 și P_2 sau P_1 . Convenim ca domeniul de acțiune al simbolului s în cuvântul Q să-l scriem sub forma $D_s^Q \subseteq P_1 P_2$ în primul caz și $D_s^Q \subseteq P_1$, în al doilea caz. Dacă folosim simbolul $\sim$ pentru a nota echivalența introdusă, atunci avem

$$(P_1) s(P_2) \sim sP_1 P_2 \quad \text{sau} \quad s(P_1) \sim sP_1.$$

Se vede acum ușor că oricare ar fi cuvintele P_1 și P_2 dacă $P_1 \sim Q_1$ și $P_2 \sim Q_2$, atunci și

$$(P_1) s(P_2) \sim sQ_1 Q_2$$

sau

$$s(P_1) \sim sQ_1$$

Demonstrația existenței cuvântului Q din teorema enunțată va consta acum în construirea unui algoritm normal care să transforme orice cuvânt compatibil P în sistemul formal în cuvântul Q echivalent cu el.

Pentru aceasta să observăm că dacă șirul algoritmilor $\{\mathfrak{P}_i^j\}$ cu ajutorul cărora construim algoritmul $\mathfrak{U}$ se iau în așa fel ca printre ei să nu existe nici un algoritm identic cu $\mathfrak{B}_5$, atunci

$$\mathfrak{U} \sqsubset P \sqsubset \subseteq s'_1 \beta P'_1 \alpha P'_2 * \dots * s'_\mu \beta P'_\mu \alpha P'_2 \dots s'_\mu \beta P'^{\mu-1}_{k_\mu} \alpha P'^\mu_{k_\mu},$$

unde fiecare element de bază care apare în cuvântul $\mathfrak{U} \sqsubset P \sqsubset$ apare sau printre cuvintele P'_i dacă elementul de bază respectiv aparține domeniului de acțiune al unui simbol de ordinul μ sau în cuvântul P'_i dacă el aparține domeniului de acțiune al unui simbol $s \in \mathcal{O} \cup \mathfrak{A}$ de ordin j mai mic ca μ și simbolul s nu conține în domeniul său de acțiune simboluri din $\mathcal{O} \cup \mathfrak{A}$ de ordin μ care să aibă ca domeniu de acțiune simbolul de bază considerat.

Să considerăm acum algoritmul normal $\mathfrak{R}$ dat de schema:

$$\left\{ \begin{array}{ll} \beta(P) \rightarrow \beta & (P \text{ cuvânt în } \mathcal{A}) \\ \alpha(P) \rightarrow \alpha & (\beta \sqsupset \in \mathcal{A}, \beta \not\subseteq \alpha) \\ \beta \rightarrow \beta \\ \beta(\rightarrow \beta \\ \alpha \rightarrow \alpha \\ \alpha(\rightarrow \alpha \\ \beta s \rightarrow \beta & (s \in \mathcal{O} \cup \mathfrak{A}) \\ \alpha s \xi \rightarrow \alpha \xi & (s \in \mathcal{O} \cup \mathfrak{A}, \xi \not\subseteq \beta) \\ \beta \alpha \xi \rightarrow \alpha \xi & (\xi \not\subseteq \Lambda, \xi \in \mathcal{A}) \\ \beta \xi \alpha \rightarrow \beta \xi & (\xi \not\subseteq \Lambda, \xi \in \mathcal{A}) \\ \beta \xi \alpha \eta \rightarrow \xi \eta & (\xi, \eta \in \mathcal{A}) \\ \beta \alpha \rightarrow \\ \Lambda \rightarrow \end{array} \right.$$

Din schema acestui algoritm rezultă ușor egalitatea

$$\mathfrak{R} \sqsubset \mathfrak{U} \sqsubset P \sqsubset \sqsubset Q,$$

unde Q are forma

$$Q \subseteq Q_1 * Q_2 * \dots * Q_i * \dots * Q_\mu.$$

Cuvintele $Q_1, Q_2, \dots, Q_\mu$ au următoarele proprietăți:

1. Cuvântul Q_μ conține numai simboluri din $\mathcal{O} \cup \mathfrak{A}$ împreună cu elementele de bază care formează domeniul lor de acțiune dispuse sub forma $s \xi \eta$ sau $s \xi$ după cum $s \in \mathcal{O}_2 \cup \mathfrak{A}$ sau $s \in \mathcal{O}_1$.

2. Orice simbol s din $\mathcal{O} \cup \mathfrak{A}$ care apare în cuvântul $Q_{\mu-i}$, i fiind mai mare sau egal cu 1, are domeniul de acțiune sau sub forma $s \xi \eta$ sau $s \xi$ dacă el nu conține în domeniul său de acțiune simboluri din $\mathcal{O} \cup \mathfrak{A}$ de ordin $\mu-i+1$. Dacă în domeniul de acțiune al simbolului s în cuvântul P intră simboluri din $\mathcal{O} \cup \mathfrak{A}$ de ordinul $\mu-i+1$, atunci are loc unul din cazurile:

$$a) Q_{\mu-i} \subseteq Q'_{\mu-i} s Q''_{\mu-i}, \quad Q'_{\mu-i} \subseteq s_1 Q''_{\mu-i}, \quad s \in \mathcal{O} \cup \mathfrak{A}.$$

În acest caz domeniul de acțiune al simbolului s conține simboluri s de ordin $\mu-i+1$.

$$b) Q_{\mu-i} \subseteq Q'_{\mu-i} s Q''_{\mu-i}, \quad Q'_{\mu-i} \subseteq \beta \xi Q''_{\mu-i} \text{ sau } Q'_{\mu-i} \subseteq \alpha \xi Q''_{\mu-i},$$

ξ fiind un element de bază al sistemului formal. În acest caz simbolul $s \in \mathcal{O} \cup \mathfrak{A}$ și avem în primul subcaz $Q'_{\mu-i} \subseteq \beta \xi Q''_{\mu-i}$, ξ primul element al domeniului de acțiune al simbolului s , în al doilea subcaz $Q'_{\mu-i} \subseteq \alpha \xi Q''_{\mu-i}$, ξ fiind al doilea element al domeniului său de acțiune. Din aceste proprietăți și din definiția algoritmului $\mathfrak{U}$ rezultă imediat că cuvintele $Q_1, Q_2, \dots, Q_\mu$ de mai sus posedă proprietățile enunțate în teoremă.

Să arătăm acum că $P \sim Q$. Pentru aceasta să arătăm că sînt îndeplinite condițiile din definiția echivalenței. Condiția 1) este evidentă, deoarece dacă P nu conține intrări ale simbolurilor din $\mathcal{O} \cup \mathfrak{A}$, atunci $\mathfrak{U} \sqsubset P \sqsubset \subseteq P$, $\mathfrak{R} \sqsubset \mathfrak{U} \sqsubset P \sqsubset \subseteq P$ ceea ce se vede simplu din definițiile algoritmilor normali $\mathfrak{U}$ și $\mathfrak{R}$.

Să presupunem acum că simbolul $s \in \mathcal{O} \cup \mathfrak{A}$ intră în cuvântul P , și să arătăm atunci că el intră și în cuvântul Q . Deoarece algoritmul $\mathfrak{U} \sqsubset P \sqsubset$ nu șterge nici un simbol din cuvântul P (în afara unor simboluri din S), rezultă că simbolul s intră și în $\mathfrak{U} \sqsubset P \sqsubset$. Avem acum două cazuri posibile:

1) s intră într-un cuvânt P'_i din cele care formează cuvântul $\mathfrak{U} \sqsubset P \sqsubset$.
2) s intră în cuvântul $\mathfrak{U} \sqsubset P \sqsubset$ sub forma $s\beta$ sau $s\sigma$. Să arătăm acum că toate simbolurile din $\mathcal{O} \cup \mathfrak{A}$ care intră în P intră în cuvântul $\mathfrak{U} \sqsubset P \sqsubset$ sub forma $s\beta$ sau $s\sigma$. Pentru aceasta este suficient să arătăm că primul caz semnalat mai sus se reduce la al doilea.

Dacă simbolul s intră în cuvântul P'_i și el nu aparține domeniului de acțiune al nici unui alt simbol din $\mathcal{O} \cup \mathfrak{A}$ care intră în P'_i , atunci $P'_i \subseteq P'^{i+1}_{i+1} s P'^{i+1}_{i+1}$ și prin urmare s apare în cuvântul $\mathfrak{U} \sqsubset P \sqsubset$ sub forma $s\beta$

sau $s\sigma$ într-un cuvânt de ordin superior ordinului j . Dacă s aparține domeniului de acțiune al altor simboluri din $\mathcal{O} \cup \mathcal{R}$ care intră în P_i , atunci

$$P_i \subseteq P_{i+1}^{i+1} s_1 P_{i+1}^{i+1}$$

iar s intră fie în cuvântul P_{i+1}^{i+1} fie în cuvântul P_{i+1}^{i+1} .

În oricare din aceste cazuri vom repeta raționamentul asupra cuvântului respectiv. După un număr finit de pași vom ajunge la concluzia că s intră în cuvântul $\mathbb{U} \perp P \perp$ sub forma $s\beta$ sau $s\sigma$. Rezultă deci că orice simbol din $\mathcal{O} \cup \mathcal{R}$ care intră în P intră și în $\mathbb{U} \perp P \perp$ sub forma $s\beta$ sau $s\sigma$. Se poate arăta în mod analog că orice element de bază care intră în cuvântul P , intră și în cuvântul $\mathbb{U} \perp P \perp$ fie sub forma $\beta\xi$ fie sub forma $\alpha\xi$, ξ fiind elementul de bază respectiv.

Examinînd acum șirul formulelor algoritmului $\mathcal{R}$ observăm că printre aceste formule nu există nici o formulă care să șteargă din cuvântul $\mathbb{U} \perp P \perp$ cuvinte de forma $s\beta$, $\beta\xi$, $s\sigma$ sau $\alpha\xi$. Rezultă, prin urmare, că toate simbolurile din $\mathcal{O} \cup \mathcal{R}$ care intră în P intră și în Q și analog pentru elementele de bază. Reciproca acestei afirmații are loc prin faptul că algoritmul $\mathcal{R}$ nu atașează cuvântului $\mathbb{U} \perp P \perp$ nici un simbol nou, și scoate toate simbolurile alfabetului $\mathcal{A}$ care apar în $\mathbb{U} \perp P \perp$ și care nu au forma $s\beta$ dacă $s \in \mathcal{O} \cup \mathcal{R}$ și $\beta\xi$ sau $\alpha\xi$ dacă ξ este un element de bază.

Să arătăm acum că oricare ar fi $s \in \mathcal{O} \cup \mathcal{R}$ care intră în cele două cuvinte P și Q , are loc și condiția 3) din definiția echivalenței.

Pentru aceasta să arătăm în primul rînd cum se determină domeniul de acțiune al unui simbol oarecare s care intră în cuvântul Q .

Deoarece cuvântul Q are forma

$$Q \subseteq Q_1 * Q_2 * \dots * Q_i * \dots * Q_\mu,$$

simbolul s intră într-unul din cuvintele Q_i , avînd ordinul i în cuvântul P . Dacă s intră în Q_μ , atunci domeniul său de acțiune este format din elementele de bază care stau imediat în dreapta lui. Deoarece în acest caz s este de ordinul μ , rezultă că în cuvântul P are forma $(P_1)s(P_2)$ sau $s(P_1)$, unde P_1 și P_2 sau P_1 sînt elemente de bază. Din observația 2) se vede atunci că pentru acest simbol are loc condiția 3).

Fie acum simbolul s , care intră în cuvântul Q_i . Din observațiile de mai sus asupra cuvintelor Q_i punem prin definiție

$$D_{s_j}^Q \subseteq \begin{cases} s_{k_{i+1}}^{i+1} - r D_{s_{k_{i+1}}^{i+1}-r}^{Q_{i+1}} s_{k_{i+1}^{i+1}-r+1}^{i+1} D_{s_{k_{i+1}^{i+1}-r+1}}^{Q_{i+1}} & \text{dacă } s_j^i \in \mathcal{O}_2 \cup \mathcal{R} \text{ și sîntem în cazul a)} \\ s_{k_{i+1}^{i+1}-r+1}^{i+1} D_{s_{k_{i+1}^{i+1}-r+1}}^{Q_{i+1}} & \text{dacă } s_j^i \in \mathcal{O}_1 \text{ și sîntem în cazul a)} \end{cases}$$

sau

$$D_{s_j}^Q \subseteq \begin{cases} \xi s_{k_{i+1}^{i+1}-r+1}^{i+1} D_{s_{k_{i+1}^{i+1}-r+1}}^{Q_{i+1}} & \text{în primul subcaz b)} \\ s_{k_{i+1}^{i+1}-r+1}^{i+1} D_{s_{k_{i+1}^{i+1}-r+1}}^{Q_{i+1}} \xi & \text{în al doilea subcaz b)}. \end{cases}$$

Numărul r este astfel ales încît oricare dintre simbolurile $s_{k_{i+1}^{i+1}-l}^{i+1}$ care intră în cuvântul Q_{i+1} , $l > r$ aparține domeniului de acțiune al unui simbol s_p^i , $p > j$.

Prin urmare vom avea

$$D_{s_{k_i}^i}^Q \subseteq \begin{cases} s_{k_{i+1}^{i+1}-1}^{i+1} D_{s_{k_{i+1}^{i+1}-1}}^{Q_{i+1}} s_{k_{i+1}^{i+1}}^{i+1} D_{s_{k_{i+1}^{i+1}}}^{Q_{i+1}} s_{k_i^i}^i \in \mathcal{O}_2 \cup \mathcal{R} \text{ în cazul a)} \\ s_{k_{i+1}^{i+1}}^{i+1} D_{s_{k_{i+1}^{i+1}}}^{Q_{i+1}} s_{k_i^i}^i \in \mathcal{O}_1 \text{ în cazul a)} \end{cases}$$

și

$$D_{s_{k_i}^i}^Q \subseteq \begin{cases} \xi s_{k_{i+1}^{i+1}}^{i+1} D_{s_{k_{i+1}^{i+1}}}^{Q_{i+1}} & \text{în primul subcaz b)}, \\ s_{k_{i+1}^{i+1}}^{i+1} D_{s_{k_{i+1}^{i+1}}}^{Q_{i+1}} \xi & \text{în al doilea subcaz b)}. \end{cases}$$

Deoarece însă

$$D_{s_j^i}^Q \subseteq \begin{cases} \xi \eta \text{ dacă } s_j^i \in \mathcal{O}_2 \cup \mathcal{R}, \\ \xi \text{ dacă } s_j^i \in \mathcal{O}_1, \end{cases}$$

unde ξ și η sînt elemente de bază ale sistemului formal, rezultă clar că oricare ar fi simbolul s care intră în cuvântul Q , D_s^Q conține numai intrări de simboluri din $\mathcal{O} \cup \mathcal{R}$ și elemente de bază. Se poate acum arăta simplu (recursiv) că dacă ξ , η intră în D_s^Q , atunci ξ , η intră și în D_s^P și reciproc. În mod analog se poate arăta că dacă $s_r \in \mathcal{O} \cup \mathcal{R}$ intră în D_s^Q , atunci el intră și în D_s^P și reciproc.

Rezultă deci că condiția 3) din definiția echivalenței este îndeplinită pentru cuvintele P și Q .

Pentru a demonstra îndeplinirea celei de-a patra condiții din definiția echivalenței pentru cuvintele P și Q , vom construi un algoritm normal $\mathcal{T}$ care pornind de la definiția domeniului de acțiune al unui simbol s oarecare din $\mathcal{O} \cup \mathcal{R}$ care intră în Q va transforma acest cuvânt în cuvântul compatibil P corespunzător lui astfel ca $P \sim Q$. Algoritmul $\mathcal{T}$ dă de asemenea o cale de obținere a denotatului cuvântului Q , care conduce la denotatul cuvântului P .

Să construim acum un algoritm $\mathcal{T}$ cu proprietatea că $\mathcal{T} \perp Q \perp \subseteq P$ iar $P \sim Q$ în sensul introdus mai sus. Pentru aceasta vom construi un șir de algoritmi a cărui compoziție normală ne va da algoritmul căutat $\mathcal{T}$.

Vom construi în primul rînd un algoritm $\mathcal{T}_1$, care transformă toate intrările din cuvântul Q de forma $s\xi\eta$ și $s\xi$ $s \in \mathcal{O} \cup \mathcal{R}$ iar ξ și η fiind elemente de bază, în cuvintele echivalente cu ele, respectiv $(\xi)s(\eta)$ și $s(\xi)$. Schema acestui algoritm este

$$\mathcal{T}_1 \begin{cases} s\xi\eta \rightarrow (\xi)s(\eta) & (s \in \mathcal{O}_2 \cup \mathcal{R}, \xi, \eta - \text{elemente de bază}, \\ s\xi \rightarrow s(\xi) & (s \in \mathcal{O}_1, \xi - \text{element de bază}). \end{cases}$$

În cuvîntul rămas vom schimba toate intrările simbolului $*$ cu simbolul δ , $\delta \in \mathcal{A}$, $\delta \notin \alpha, \beta, \sigma$. Schema acestui algoritm, notat cu $\mathcal{I}_2$, va fi atunci

$$\mathcal{I}_2 \{ * \rightarrow \delta.$$

Cuvîntului obținut îi vom aplica algoritmul $\mathcal{I}_3$, care schimbă ultima intrare a simbolului δ cu $*$.

Schema acestui algoritm este următoarea:

$$\mathcal{I}_3 \left\{ \begin{array}{ll} \rho\xi \rightarrow \xi\rho & (\xi \in \mathcal{A} \cup \{\alpha, \beta, \delta\}, \rho \in \mathcal{A}, \rho \notin \alpha, \beta, \delta, \sigma) \\ \xi\rho \rightarrow \xi\tau & (\tau \notin \rho, \tau \in \mathcal{A}) \\ \xi\tau \rightarrow \tau\xi & (\xi \notin \delta) \\ \delta\tau \rightarrow \cdot * \\ \rightarrow \rho \end{array} \right.$$

Cuvîntului obținut îi vom aplica algoritmul $\mathcal{I}_4$, care va transforma pe rînd toate simbolurile s de ordin $\mu - i$ ($i = 1, 2, \dots, \mu - 1$) împreună cu domeniul lor de acțiune în cuvintele echivalente cu ele. Schema algoritmului $\mathcal{I}_4$ este următoarea:

$$\mathcal{I}_4 \left\{ \begin{array}{ll} s\sigma * P \rightarrow * \delta s\sigma P & (P - \text{cuvînt compatibil în } \mathcal{A}) \\ s\sigma P \rightarrow Ps\sigma & \sigma \notin \Lambda \text{ sau } \sigma \notin \Lambda, \sigma \in \mathcal{A}, \sigma \notin \alpha, \beta, \gamma, \delta \\ Ps\sigma \rightarrow \gamma s(P) & (s \in \mathcal{O}_1, \gamma \in \mathcal{A}, \gamma \notin \alpha, \beta, \delta) \\ QPs\sigma \rightarrow \gamma(Q)s(P) & (\text{dacă } \sigma \notin \Lambda) \\ QPs\sigma \rightarrow \gamma(P)s(Q) & (P, Q \text{ cuvinte compatibile în } \mathcal{A}) \\ s\alpha\xi * P \rightarrow * \delta s\alpha\xi P & (\text{dacă } \sigma \notin \Lambda) \\ s\alpha\xi P \rightarrow Ps\alpha\xi \\ Ps\alpha\xi \rightarrow \gamma(P)s(\xi) \\ s\beta\xi * P \rightarrow * \delta s\beta\xi P \\ s\beta\xi P \rightarrow Ps\beta\xi \\ Ps\beta\xi \rightarrow \gamma(\xi)s(P) \\ P * \delta \rightarrow * \delta P \\ * \delta \rightarrow * \\ \gamma \rightarrow \\ \delta * \rightarrow * \\ * \rightarrow \\ \delta \rightarrow \end{array} \right.$$

Se vede acum simplu că algoritmul $\mathcal{I} \simeq \mathcal{I}_4 \circ \mathcal{I}_3 \circ \mathcal{I}_2 \circ \mathcal{I}_1$ are proprietatea că

$$\mathcal{I} \mathcal{L} Q \mathcal{J} \bar{\mathcal{O}} P, P \sim Q$$

P fiind un cuvînt compatibil în sistemul formal, al cărui denotat este același cu al cuvîntului Q , dacă el există, după cum se vede simplu din definiția denotatului [5].

Prin urmare între mulțimea $\{P\}$ a cuvintelor compatibile în sistemul formal construit în [5] și mulțimea $\{Q\}$ a cuvintelor echivalente cu cuvintele mulțimii $\{P\}$ în sensul introdus mai sus, se poate stabili o corespondență biunivocă.

Scopul sistemului formal construit în lucrarea [5] și a algoritmilor de decizie din această lucrare este de a arăta rezolvabilitatea cu ajutorul mașinilor de calcul a unor clase de probleme, care admit o reprezentare sub formă de cuvinte compatibile în sistemul formal construit. Aceste probleme pentru o interpretare corespunzătoare a sistemului formal constituie probleme obișnuite din matematică, cum ar fi de exemplu posibilitatea folosirii mașinii de calcul în cazul problemelor nearitmetice, construirea programelor interpretative la o mașină de calcul, mecanizarea demonstrației unor teoreme etc. Într-o lucrare ulterioară, vom preciza și demonstra rezolvabilitatea acestor clase de probleme în sistemul formal construit.

Primită la radacție la 25 ianuarie 1964

BIBLIOGRAFIE

1. EDMUND C. BERKELEY, *Symbolic Logic and Intelligent Machines*. Reinhold Publishing Corporation, New York, 1959.
2. Б. В. ГНЕДЕНКО, *Элементы программирования*. Государственное изд. физ.-матем. литературы, Москва, 1963.
3. А. А. МАРКОВ, *Теория алгарифмов*. Труды матем. инст. им. В. А. Стеклова, XLII, 1954.
4. Z. PAWLAK, *Organization of addressless Computers Working in Parenthesis Notation*. Zeitschr. f. Math.-Logik und Grundlagen d. Math., 9 S. 243—250 (1963).
5. T. RUS, *Asupra unui sistem formal*. I. Studii și cercetări matematice, 15, 4 (1964).
6. C. BERGE, *Theorie des Graphes et ses applications*. Dunod, Paris, 1958.