INFORMATICA, 1999, Vol. 10, No. 1, 27–44

A Generalization of Regular Expressions

Aleksei L. GOMOZOV, Larisa I. STANEVICHENE

Department of Computational Mathematics and Cybernetics
Moscow State University, Moscow, Russia

Received: January 1999

Abstract. CF-expressions are defined which generalize the regular one. It is established that so called pseudo-coiterating CF-expressions characterize the regular sets. The results are used to develop some more characterizations of the regular sets: the pseudo-coiterating D-graphs and the pseudo-coiterating pushdown automata (PDAs). An algorithm is presented for deciding whether a device of three mentioned types is pseudo-coiterating or not. Apparently, the pseudo-coiterating PDAs form the most large of classes of PDAs the solvability of the question of belonging to which was proved and which are known as characterizations of the regular sets.

Key words: D-graphs; regular expression generalization; characterization of context-free languages; pushdown automata; characterization of regular sets.

1. Introduction

In this development we attempt to express adequately the common essence of the following formal descriptions of the context-free (in shorthand form, CF) languages: D-graphs, PDAs, and CF-grammars. Defined here CF-expressions summarize the essence of the mentioned formalisms. Now it is possible the most accurate description of the primitives for pumping of CF-words. The stated here Growth Theorem for the CF-expressions indicates such a description.

As a consequence, we define easily a subclass of the CF-expressions that characterizes the regular sets. This subclass is remarkable by its wideness that is compatible to the wideness of the CF-grammars without self-embedding symbols.

Further, this result indicates the subclass of PDAs (the so-called pseudo-coiterating PDAs) that characterizes the regular sets. There exists an algorithm for deciding, once any PDA is given, whether it is pseudo-coiterating or not. Thus, the present paper contains the sufficient condition of CF-language regularity which can be checked.

The notion of a CF-expression was prepared by the investigation of the D-graphs (see (Stanevichene, 1997) and its references). We use terms and denotations from (Stanevichene, 1997).

In Section 2 of the present paper agreements are accepted about the denotations of objects that define a D-graph. The notion of the core is slightly extended.

Section 3 considers CF-expressions which generalize the regular expressions. Pseudo-coiterating CF-expressions are defined. It is stated that they characterize the regular sets. This fact is important to the following Section.

In Section 4, the pseudo-coiterating PDAs are defined and investigated. It is established that a pseudo-coiterating PDA accepts a regular set. The core of a PDA M indicates, whether M is pseudo-coiterating or not. Consequently, we have a partial algorithm of checking of $L(M)$'s regularity. Hence, the paper is also a contribution to the regularity question that, since (Stearns, 1967) appearance, has not received a great attention.

2. D-graph and its core

The following definitions introduce convenient denotations of some notions defined in (Stanevichene, 1997).

For any D-set $\mathcal{P} \subseteq \Sigma_(\times \Sigma_)$ let $Left(\mathcal{P})$ and $Right(\mathcal{P})$ denote respectively the sets $\{a \mid \exists b \in \Sigma_) \ (a, b) \in \mathcal{P}\}$, $\{b \mid \exists a \in \Sigma_(\ (a, b) \in \mathcal{P}\}$.

DEFINITION 1. Let Σ, V be finite sets, $P_0 \in V$, $F \subseteq V$, $\mathcal{P}$ be a D-set (finite, also), λ be a function

$$Left(\mathcal{P}) \cup Right(\mathcal{P}) \rightarrow V \times (\Sigma \cup \{\Lambda\}) \times V.$$

Then the sixtuple

$$D = (\Sigma, V, \mathcal{P}, \lambda, P_0, F)$$

is called a *D-graph* with the input alphabet Σ, set of vertices V, input vertex P_0, set of the output vertices F, D-set $\mathcal{P}$, set of edges $E(D) = Left(\mathcal{P}) \cup Right(\mathcal{P})$, and location function λ.

DEFINITION 2. Let $\pi \in E(D)$, $P, Q \in V$, $a \in \Sigma \cup \{\Lambda\}$, $\lambda(\pi) = (P, a, Q)$. Then P, Q are called the initial and final, respectively, vertices of π, a is called π's label.

Occasionally, we omit location functions within D-graph formulas implying that edge weights are complicated to avoid the confusion of multiple edges.

The label $\omega(T)$ of a path T is defined as the natural sequence of its edge labels.

The intersection $Sentences(D)$ of the D-language $\mathcal{L}_P$ and the set of all paths from P_0 to P, $P \in F$, defines a language

$$L(D) = \{\omega(T) | T \in Sentences(D)\}.$$

DEFINITION 3. $Core(D, w, d)$ is defined as the set of all (w, d)-canons. In (Stanevichene, 1997) (see Definition 15), elements of a core were, in addition, sentences.

The Growth Theorem (see Theorem 5) in (Stanevichene, 1997) is generalized as follows.

Theorem 1. *Let $D = (\Sigma, V, \mathcal{P}, \lambda, P_0, F)$ be a D-graph. For every of its paths $T \in \mathcal{L}_P$ there exists $T_0 \in Core(D,1,1)$ such that $(T_0, T) \in \Uparrow_D^*$. In each element $\langle T_1, T_2, T_3, T_4, T_5 \rangle$ of T's genealogy from the ancestor T_0, the formantis $T_2T_3T_4$ is contained in some element of $Core(D,2,1)$.*

3. CF-Expressions

Ogden's lemma and our growth notion induce to add to regular operations a "coiterating" operation which groups pairs of words around the "center", similarly to the following CF-productions:

$$S \to z|xSy, \quad z \in Z \subseteq \Sigma^*, \quad (x,y) \in X \times Y \subseteq \Sigma^* \times \Sigma^*.$$

Here it can be said that z and xzy are the alternative centers of a growing word, and the replacement of z by xzy pumps the word. Clearly, it is necessary to find a means of marking subwords the replacement of which is valid. Hence, coiteration may be defined through lists of some primitive alternations and through techniques for pointing of positions where an element of a list may be substituted.

Recursive ties of coiterated objects require to distinguish one coiteration instance from another. Usual methods of expressing of hierarchies of subwords within CF-words influenced the following Definition 4.

Let Σ_1, Σ_2 be alphabets, $x \in (\Sigma_1 \cup \Sigma_2)^*$. Then $projection(x, \Sigma_2)$ denotes the result of the deletion all symbols of $\Sigma_1 - \Sigma_2$ from x.

Let us agree that the truth of $b_i \dots b_j \in \mathcal{L}_P$, $1 \leqslant i < j \leqslant k$, implies $projection(x_i b_{i+1} x_{i+1} \dots b_{j-1} x_{j-1}, \mathcal{B}) \in \mathcal{L}_P$ every time we consider alphabets Δ, $\mathcal{B}$, a D-set $\mathcal{P} \subseteq \mathcal{B}^2$, and a formula $x_0 b_1 x_1 \dots b_k x_k$ where $k \geqslant 2$, $b_l \in \mathcal{B}$ for $1 \leqslant l \leqslant k$, and x_m is a denotation of a word over $\Delta \cup \mathcal{B}$ for $0 \leqslant m \leqslant k$.

DEFINITION 4. Let $Names$ be a nonempty finite set, $\mathcal{B} = \{(_\iota,)_\iota | \iota \in Names\}$. Let Σ, $\mathcal{B}$, and $\{+, \epsilon, \emptyset, (,)\}$ be pairwise disjoint, $Alph = \Sigma \cup \mathcal{B} \cup \{+, \epsilon, \emptyset, (,)\}$. Let us define recursively a CF-expression ζ over Σ (as a specific word over $Alph$):

1. $a \in \Sigma \cup \{\epsilon, \emptyset\}$ is a CF-expression;
2. if α, β are CF-expressions, then:
   (a) $\zeta = \alpha + \beta$ is a CF-expression; the subexpressions α and β are called addends of ζ; ζ is called a sum;
   (b) $\zeta = (\alpha)(\beta)$ is a CF-expression; the subexpressions α and β are called factors of ζ; if a factor is not a sum, then round brackets are redundant;
   (c) if $\iota \in Names$, then $(_\iota \beta)_\iota$ is a CF-expression of the name ι (or a ι-nest);
3. all CF-expressions over Σ are constructed by 1–2.

Further $Alph(\zeta)$, $\mathcal{B}(\zeta)$, and $Names(\zeta)$ denote the sets $\{c \in Alph | \exists\ (u, v \in Alph^*)\ \zeta = ucv\}$, $\mathcal{B} \cap Alph(\zeta)$, and $\{\iota \in Names | (_\iota \in \mathcal{B}(\zeta)\}$ respectively.

The following notions of a fraction and clan help formulate the language defined by a CF-expression.

DEFINITION 5. Let ζ be a CF-expression over Σ. Define a set $Fractions(\zeta)$ of "fractions" of ζ recursively:

1) $Fractions(a) = \{a\}$ for $a \in \Sigma \cup \{\epsilon, \emptyset\}$;

2) if α, β are CF-expressions, then

$$Fractions((\beta)) = Fractions(\beta),$$

$$Fractions(\alpha + \beta) = Fractions(\alpha) \cup Fractions(\beta),$$

$$Fractions((\alpha)(\beta)) = Fractions(\alpha)Fractions(\beta),$$

$$Fractions((_\iota\beta)_\iota) = \{(_\iota\}Fractions(\beta)\{)_\iota\}.$$

Hence, a fraction appears as a CF-expression having no sums and nameless parenthesis. By this reason every word of such a form will be called a fraction.

DEFINITION 6. Let ζ be a fraction over Σ. Then its label $\omega(\zeta) \subset \Sigma^*$ is defined recursively as follws:

1) $\omega(a) = \{a\}$, $a \in \Sigma$;

2) $\omega(\epsilon) = \{\Lambda\}$;

3) $\omega(\emptyset) = \emptyset$;

4) if α and β are fractions, then:

a) $\omega(\alpha\beta) = \omega(\alpha)\omega(\beta)$;

b) $\omega((_\iota\beta)_\iota) = \omega(\beta)$.

Note that for any set of words S the catenations $S\emptyset$ and $\emptyset S$ are empty. It justifies the following notion of a trim fraction.

DEFINITION 7. Let ξ be a fraction. Let $\xi = \emptyset$ or $\emptyset \notin Alph(\xi)$. Then ξ is called a trim fraction.

The following formula indicates the method of trimming:

$$trim(\xi) = \begin{cases} \emptyset, & \xi \text{ contains the symbol } \emptyset, \\ \xi & \text{otherwise.} \end{cases}$$

For any CF-expression ζ let $Trim(\zeta) = \{trim(\xi) | \emptyset \neq \xi \text{ is a fraction of } \zeta\}$.

DEFINITION 8. Let us define recursively a clan being generated by a CF-expression ζ:

$$Clan(\zeta) = Trim(\zeta) \cup \{\beta_1(_\iota\alpha_2)_\iota\beta_3 | \exists(\beta_2, \alpha_1, \alpha_3 \in Alph(\zeta)^*)$$

$$\alpha_1(_\iota\alpha_2)_\iota\alpha_3, \beta_1(_\iota\beta_2)_\iota\beta_3 \in Clan(\zeta)\}.$$

DEFINITION 9. Let ζ be a CF-expression. Then

$$L(\zeta) = \bigcup_{\xi \in Clan(\zeta)} \omega(\xi)$$

is called the language being defined by ζ. If $Clan(\zeta) = \emptyset$, then $L(\zeta) = \emptyset$.

DEFINITION 10. Let ζ be a CF-expression, $\iota \in Names(\zeta)$. Then $h(\iota, \zeta) =$ $= depth(projection(\zeta, \{(_\iota,)_\iota\}))$ is called the ι-height of ζ.

DEFINITION 11. The number

$$height(\zeta) = \max\{h(\iota, \zeta) | \iota \in Names(\zeta)\}$$

is called the height of the CF-expression ζ.

DEFINITION 12. Let $d \geqslant 0$, ζ be a CF-expression. Then $Core(\zeta, d) = \{\xi \in Clan(\zeta) | height(\xi) \leqslant d\}$ is called a d-core of ζ.

From Definitions 5 and 8 it follows that

$$width(projection(\xi, \mathcal{B}(\zeta))) \leqslant width(projection(\zeta, \mathcal{B}(\zeta)))$$

for any $\xi \in Clan(\zeta)$. Besides the equality $\xi = xyz$ implies $|y| \leqslant |\zeta|$ for any $x, z \in Alph(\zeta)^*$, $y \in (\Sigma \cup \{\epsilon\})^*$. From Definition 11 it follows that

$$depth(projection(\xi, \mathcal{B}(\zeta))) \leqslant d \cdot |Names(\zeta)|.$$

Hence, $Core(\zeta, d)$ is a finite set by Theorem 1 of (Stanevichene,1997).

DEFINITION 13. Let ζ be a CF-expression, $\xi \in Clan(\zeta)$ contain a m-nest u, $h(m, u) =$ 2, and $h(k, u) < 2$ for every name $k \neq m$. Then u is called a formantis of ζ.

Lemma 1 (see below) implies that the 2-core of a CF-expression reveals every formantis.

DEFINITION 14. Let $u = v(_m x(_m y)_m z)_m w \in Clan(\zeta)$, $(_m x(_m y)_m z)_m$ be a formantis. Then we write $(v(_m y)_m w, u) \in \Uparrow_\zeta$. The relation $\Uparrow_\zeta^*$ is called the growth relation. The quintuple $(v, (_m x, (_m y)_m, z)_m, w)$ is called the history of the growth of u from $v(_m y)_m w$.

DEFINITION 15. Let $u(_m x(_m y)_m z)_m v$ be a CF-expression. Then the subwords $(_m x$ and $z)_m$ are called its (matching) cycles (more precisely, m-cycles), left and right respectively.

Similarly to the case of D-graphs, it is easy to define the analog of the reduction function mapping a factorization of a CF-expression in a set of CF-expressions. This function (let its name be $reduction$, also) deletes in the given CF-expression all cycles that do not intersect with even-numbered parts of the given factorization.

Lemma 1. *If β is a formantis and $\alpha\beta\gamma \in Clan(\zeta)$, then*

$$reduction(\alpha, \beta, \gamma) \subseteq Core(\zeta, 2).$$

Proof. Let $\alpha'\beta\gamma' \in reduction(\alpha, \beta, \gamma)$ where α' and γ' are images of α and γ respectively. Suppose that $\alpha'\beta\gamma'$ contains a k-nest ξ for some k and $h(k, \xi) = 3$. Observe that if different nests are subexpressions of the same expression, then either one of the nests contains another, or they do not overlap. Consequently, ξ is not within α', γ' (by the construction of them), and β (by the reason that each subexpression of a formantis has the height at most 2). If ξ contains β, then matching cycles of ξ are within α' and γ' (left and right, respectively), contrary to construction of $\alpha'\beta\gamma'$.

Lemma 2. *If a fraction $\xi \in Clan(\zeta)$ contains a cycle, then there exists $\xi' \in Clan(\zeta)$ such that $(\xi', \xi) \in\Uparrow_\zeta$ and $|\xi'| < |\xi|$.*

Proof. Prove that ξ contains a formantis. Let us choose a nest $\beta = ({}_m x ({}_m y)_m z)_m \in Nests = \{u | \exists (v, w \in Alph(\zeta)^*, m \in Names(\zeta))\ (\xi = vuw$, u is a m-nest, $h(m, u) \geqslant 2)\}$ having the minimal length. By the choice it contains no proper subexpression being an element of $Nests$, i.e., is a formantis. Let $\xi = \alpha\beta\gamma$ for some α, γ. Then, by the definition of growth relation, we have $(\xi', \xi) \in\Uparrow_\zeta$ for $\xi' = \alpha({}_m y)_m\gamma$ and $|\xi'| < |\xi|$.

Theorem 2 (CF-Expression Growth Theorem). *Let ζ be a CF-expression. For every $\xi \in Clan(\zeta)$ there exist number $k \geqslant 0$ and a sequence of fractions $\xi_0, \ldots, \xi_k$ such that $\xi_0 \in Core(\zeta, 1)$, $\xi_k = \xi$, $(\xi_{i-1}, \xi_i) \in\Uparrow_\zeta$ with a history $(u_i, x_i, y_i, z_i, v_i)$, $0 < i \leqslant k$, and the formantis $x_i y_i z_i$ is a subexpression of some element of $Core(\zeta, 2)$.*

Proof. If $\xi \in Core(\zeta, 1)$, then the fact holds for $k = 0$. Now let $\xi \in Clan(\zeta) - Core(\zeta, 1)$. Then ξ has a cycle. By Lemma 2 there exists $\xi' \in Clan(\zeta)$ such that $(\xi', \xi) \in\Uparrow_\zeta$ and $|\xi'| < |\xi|$. The last inequality implies the existence of number $k \geqslant 1$ and sequence $\xi_0, \ldots, \xi_k$ such that ξ_0 has no cycle (i.e., is of $Core(\zeta, 1)$), $\xi_k = \xi$, and $(\xi_{i-1}, \xi_i) \in\Uparrow_\zeta$ for $0 < i \leqslant k$ with a history $(u_i, x_i, y_i, z_i, v_i)$. By Lemma 1 each formantis $x_i y_i z_i$, $0 < i \leqslant k$, is a subexpression of an element of $Core(\zeta, 2)$.

The following theorem, in essence, generalizes Cleene's theorem.

Theorem 3. $L = L(\zeta)$ *for some CF-expression ζ iff $L = L(D)$ for some D-graph D.*

Our proof of this theorem is constructive. It uses a series of new terms and denotations. The following sections 3.1, 3.2 present two parts of this proof.

3.1. *Transformation of a D-graph to an Equivalent CF-expression*

As above D-graph is denoted by

$$D = (\Sigma, V, \mathcal{P}, \lambda, P_0, F).$$

DEFINITION 16. Let $T = \pi_1 T_1 \pi_2 T_2 \in \mathcal{L}_P$ be a path of D, $\pi_1, \pi_2 \in \mathcal{P}$, $T_1 \in \mathcal{L}_P$. Then $rlf(T) = (\pi_1, T_1, \pi_2, T_2)$ is called the right-linear factorization of T.

Note that if $rlf(T) = (\pi_1, T_1, \pi_2, T_2)$, then $Core(D, 1, 1)$ contains a path having a right-linear factorization $(\pi_1, T_1', \pi_2, T_2')$ where $pair(T_i') = pair(T_i)$ for $i = 1, 2$. (Indeed, $reduction(\pi_1, T_1, \pi_2, T_2) \subseteq Core(D, 1, 1)$.) As it will be seen, this fact implies that $Core(D, 1, 1)$ provides all necessary information for the constructions proposed below. In this section we write $Core(D)$ instead of $Core(D, 1, 1)$.

DEFINITION 17. Let $Phrases(D) = Sentences(D) \cup \{T_1, T_2 | \exists((\pi_1, \pi_2) \in \mathcal{P}, T \in Phrases(D))\ (\pi_1, T_1, \pi_2, T_2) = rlf(T)\}$. Then every element of $Phrases(D)$ is called a phrase of D.

In the following subsets of $Phrases(D)$, empty elements are synonyms of the denotation Λ, and the subsets themselves are languages in $E(D)^*$.

Let

$$Alt(P, Q) = \{T \in Phrases(D) | pair(T) = (P, Q)\},$$
$$Productions(P, Q) = Alt(P, Q) \cap Core(D).$$

DEFINITION 18. Let $T \in Phrases(D)$. Then a fraction $scheme(T)$ is defined recursively:

1) if T is empty, then $scheme(T) = (_{pair(T)}\epsilon)_{pair(T)}$;

2) if $rlf(T) = (\pi_1, T_1, \pi_2, T_2)$, then

$$scheme(T) = (_{pair(T)}\pi_1 scheme(T_1)\pi_2 scheme(T_2))_{pair(T)}.$$

Transform a D-graph D to a CF-expression $\mathcal{E}(D)$ over $E(D)$ where $Names(\mathcal{E}(D)) = \{pair(T) | T \in Phrases(D)\}$:

$$\mathcal{E}(D) = \sum_{Q \in F} \sum_{T \in Productions(P_0, Q)} scheme(T).$$

Definition 18 implies that $Alt(P, Q)$ is equal to the set $\{\pi_1 T_1 \pi_2 T_2 |\ beg(\pi_1) = P;\ parti(\pi_1 T_1 \pi_2) = 1; T_1 \in Alt(end(\pi_1), beg(\pi_2)); T_2 \in Alt(end(\pi_2), Q)\}$, being united to $\{P\}$ in case of $P = Q$.

Let

$$\tau(P, Q) = \{scheme(T) | T \in Productions(P, Q)\},\ L_{P,Q} = L(\sum_{\xi \in \tau(P,Q)} \xi).$$

Clearly, $L_{P,Q} \subseteq Alt(P,Q)$. The fact $Alt(P,Q) \subseteq L_{P,Q}$ holds by the following lemma.

Lemma 3. *If T is a phrase, then $T \in L_{pair(T)}$.*

Proof. Let $\zeta = \sum_{\xi \in \tau(pair(T))} \xi$. Note that the construction of schemes implies the equalities $Fractions(\zeta) = Trim(\zeta) = \tau(pair(T))$. By Theorem 2 it is sufficient to prove that $T \in \omega(\psi)$ where $(\xi, \psi) \in \Uparrow_\zeta^m$ for some $\xi \in \tau(pair(T))$ and $m \geqslant 0$.

Let's use induction on $k \geqslant 0$ where $2k = |T|$. If $k = 0$, then T is an empty phrase. Hence, $\xi = ({}_{pair(T)}\epsilon)_{pair(T)} \in \tau(pair(T))$. Note that $(\xi, \xi) \in \Uparrow_\zeta^0$ and $T \in \omega(\xi)$, therefore our assertion holds in this case. Let $k > 0$. Assume the assertion for every phrase the length of which is less than $2k$. Consider a phrase T having the length $2k$. Suppose that $rlf(T) = (\pi_1, T_1, \pi_2, T_2)$. By induction hypothesis we have for $i = 1, 2$: $T_i \in \omega(\psi_i)$ where $(\xi_i, \psi_i) \in \Uparrow_\zeta^{m_i}$ for some $\xi_i \in \tau(pair(T_i))$ and $m_i \geqslant 0$. It may be supposed (cf. the formula for ξ_0 in Theorem 2) that $height(\xi_i) = 1$, $i = 1, 2$. Then $\xi = \pi_1\xi_1\pi_2\xi_2 \in \tau(pair(T))$. As $(\xi, \pi_1\psi_1\pi_2\psi_2) \in \Uparrow_\zeta^{m_1+m_2}$ and $T = \pi_1 T_1 \pi_2 T_2 \in \omega(\pi_1\psi_1\pi_2\psi_2)$, the assertion holds for $k \geqslant 0$.

Hence, $T \in L_{pair(T)}$.

This lemma implies that $L(\mathcal{E}(D)) = Sentences(D)$. Define a morphism φ by: $\varphi(\pi) = \omega(\pi)$, $\pi \in E(D)$, $\omega(\pi) \in \Sigma$; $\varphi(\pi) = \epsilon$, $\pi \in E(D)$, $\omega(\pi) = \Lambda$; $\varphi(a) = a$, $a \notin E$. By Theorem 3 of (Stanevichene, 1997) the CF-expression $Expr(D) = \varphi(\mathcal{E}(D))$ describes the language $L(D)$.

3.2. *Transformation of a CF-expression to a D-graph*

Now prove that every CF-expression defines a CF-language. It is sufficient to consider the case of a nonempty clan. Otherwise a CF-expression defines the empty (i.e., CF) language.

First transform a CF-expression to simplify the construction of the equivalent PDA or D-graph.

DEFINITION 19. Let a CF-expression ζ over Σ be a nest or $\zeta \in \Sigma \cup \{\epsilon, \emptyset\}$. Then ζ is called an item.

Let ζ be a CF-expression. To provide the uniformity of nests within the below defined CF-expression $Augm(\zeta)$, we require the following: 1) every element of $Clan(\zeta)$ has the form $\alpha\beta$ for an item α and fraction β; 2) if $x({}_{\iota_1}y({}_{\iota_2}\beta)_{\iota_2})_{\iota_1}z \in Clan(\zeta)$ or $x({}_{\iota_1}({}_{\iota_2}\beta)_{\iota_2}y)_{\iota_1}z \in Clan(\zeta)$ for some $\iota_1, \iota_2 \in Names(\zeta)$ and $x, y, z, \beta \in Alph(\zeta)^*$, then y is a fraction. To reach that, it is sufficient to add the "artificial" factors ϵ within elements of $Fractions(\zeta)$ which violate this requirement.

DEFINITION 20. Let $\xi \neq \emptyset$ be a trim fraction, κ be a finite sequence of natural numbers. Define *a marking of ζ by means of κ* recursively as the following fraction $rlm(\zeta, \kappa)$:

1) if $a \in \Sigma \cup \{\epsilon\}$, then $rlm(a, \kappa) = (_\kappa a)_\kappa$ and $rlm((_\iota a)_\iota, \kappa) = (_\iota a)_\iota$ for any name ι;

2) if α is an item, then

$$rlm(\alpha\beta, \kappa) = (_\kappa rlm(\alpha, (\kappa, 1))rlm(\beta, (\kappa, 2)))_\kappa,$$

$$rlm((_\iota \alpha\beta)_\iota, \kappa) = (_\iota rlm(\alpha, (\kappa, 1))rlm(\beta, (\kappa, 2)))_\iota$$

for any fraction β and name ι.

DEFINITION 21. Let ζ be a CF-expression such that $Clan(\zeta) \neq \emptyset$. Let the elements of $Core(\zeta, 2)$ be anyhow numbered, $\nu(\xi)$ denote number of $\xi \in Core(\zeta, 2)$, and

$$Mark(\zeta) = \{rlm((_0\xi)_0, \nu(\xi))|\xi \in Core(\zeta, 2)\}.$$

Then the CF-expression

$$Augm(\zeta) = \sum_{\xi \in Mark(\zeta)} \xi$$

is called ζ's augmentation.

DEFINITION 22. Let $u\xi v \in Clan(Augm(\zeta))$, ξ be a nest. Then ξ is called an active nest.

The following lemma is useful to prove the equivalence of $Augm(\zeta)$ to ζ.

Lemma 4. *Let ζ have a nonempty clan. Then:*

1) $Core(\zeta, 1) = \{projection(\xi, Alph(\zeta))|\xi \in Core(Augm(\zeta), 1)\}$*;*

2) $\{\psi \in Alph(\zeta)^*|\psi$ *is a formantis of* $\zeta\}$ = $\{projection(\xi, Alph(\zeta))|\xi$ *is a formantis of* $Augm(\zeta)\}$.

Proof. Let $\kappa \in Names(Augm(\zeta)) - Names(\zeta)$. Remark that $h(\kappa, \psi) \leqslant 1$ for any $\psi \in Mark(\zeta)$. Consequently, every formantis of an element of $Mark(\zeta)$ is a marking of a formantis of an element of $Core(\zeta, 2)$. This implies the equality $Core(Augm(\zeta), 1) = \{\xi \in Mark(\zeta)|projection(\xi, Alph(\zeta)) \in Core(\zeta, 1)\}$. Hence, the first assertion of the lemma holds, as $Core(\zeta, 1) \subseteq Core(\zeta, 2)$ and $Core(\zeta, 2) = \{projection(\xi, Alph(\zeta))|\xi \in Mark(\zeta)\}$.

The last equality implies $\{\psi \in Alph(\zeta)^*|\psi$ is a formantis of $\zeta\} = FS$ where FS denotes the set $\{projection(\xi, Alph(\zeta))|\xi$ is a formantis, $\exists(u, v \in Alph(Augm(\zeta))^*)$ $u\xi v \in Mark(\zeta)\}$. Show that FS coincides

$$\{projection(\xi, Alph(\zeta))|\xi \text{ is a formantis of } Augm(\zeta)\}.$$

From Lemma 1 it follows that FS presents the projections onto $Alph(\zeta)$ of every $Augm(\zeta)$'s formantis having its name from $Names(\zeta)$. Consequently, it is sufficient to show that each of κ-nests,

$$\kappa \in Names(Augm(\zeta)) - Names(\zeta),$$

is not a formantis. If $h(\kappa, v) = 2$ for an active κ-nest v, then (see the remark in the beginning of the proof) a genealogy of v contains two quintuples

$$(u_0, ({}_\iota u_1({}_\kappa u_2, ({}_\iota u_3)_\iota, u_4)_\kappa u_5)_\iota, u_6)$$

and

$$(u'_0, ({}_\iota u_1({}_\kappa u_2, ({}_\iota u'_3)_\iota, u_4)_\kappa u_5)_\iota, u'_6)$$

displaying the same matching cycles. Consequently, $h(\iota, v) \geqslant 2$, i.e., v is not a formantis.

Lemma 5. $L(\zeta) = L(Augm(\zeta))$.

Proof. On the base of Lemma 4, it is not difficult to construct the proof of the following assertion:

$$\begin{aligned}\{projection(\xi, Alph(\zeta)) | \exists \xi_0 \in Core(Augm(\zeta), 1)\ (\xi_0, \xi) \in\Uparrow^k_{Augm(\zeta)}\} \\ = \{\psi | \exists \psi_0 \in Core(\zeta, 1)\ (\psi_0, \psi) \in\Uparrow^k_\zeta\}\end{aligned}$$

for $k \geqslant 0$.

This fact implies that there exists a surjection

$$\varphi : Clan(Augm(\zeta)) \to Clan(\zeta)$$

such that

$$projection(\xi, Alph(\zeta)) = \varphi(\xi)$$

for $\xi \in Clan(Augm(\zeta))$. Consequently,

$$L(Augm(\zeta)) = \cup_{\xi \in Clan(Augm(\zeta))} \omega(\xi) = \cup_{\xi \in Clan(Augm(\zeta))} \omega(\varphi(\xi)) = L(\zeta).$$

Lemma 6. *Let* $({}_m({}_k\alpha)_k({}_l\beta)_l)_m$ *be an active nest. Then an element of* $Core(Augm(\zeta), 3)$ *has a nest* $({}_m({}_k\alpha')_k({}_l\beta')_l)_m$ *as its subexpression.*

Proof. From Definition 22 it follows that $Clan(Augm(\zeta))$ contains $\xi = u({}_m({}_k\alpha)_k({}_l\beta)_l)_m v$ for some $u, v \in Alph(Augm(\zeta))^*$. Let

$$\xi' \in reduction(u, ({}_m({}_k, \alpha)_k, ({}_l, \beta)_l)_m v).$$

Observe that $projection(\xi, \mathcal{B}(Augm(\zeta)))$ is a balanced word. Consequently, if a cycle contains a parenthesis, then the matching parenthesis lies within either this cycle, or the matching one.

Besides, remember that a deletion of matching cycles must not imply a deletion of preserved parts of ξ'.

Taking into account these observations, it is not difficult to imagine all possible positions of parentheses $(_\iota$, $)_\iota$, $\iota \in Names(Augm(\zeta))$, within ξ and to conclude that $h(t,\xi)$ reaches its maximum in the following cases:

$$\xi = u_1(_t u_2(_m(_k\alpha_1(_t\alpha_2)_t\alpha_3)_k(_l\beta)_l)_m v_1)_t v_2, \quad t = k,$$

$$\xi = u_1(_t u_2(_m(_k\alpha)_k(_l\beta_1(_t\beta_2)_t\beta_3)_l)_m v_1)_t v_2, \quad t = l.$$

Consequently, $h(\iota,\xi') \leqslant 3$ for every $\iota \in Names(Augm(\zeta))$.

DEFINITION 23. Let

$$k,l,m \in Names(Augm(\zeta)), \quad \alpha,\beta \in Alph(Augm(\zeta))^*,$$

and $\xi = (_m(_k\alpha)_k(_l\beta)_l)_m$ be an active nest. Then (k,l,m) is called *an alliance (conditioned by ξ)*.

Note that every active nest either conditions an alliance, or has the form $(_\iota a)_\iota$, $a \in \Sigma \cup \{\epsilon\}$, $\iota \in Names(Augm(\zeta))$.

Lemma 6 signifies that every alliance is conditioned by a nest of an element of $Core(Augm(\zeta),3)$.

Now convert $Core(Augm(\zeta),3)$ into a D-graph

$$D(\zeta) = (\Sigma, V, \mathcal{P}, \lambda, [(_0, Z_0], \{[)_0, Z_0]\}).$$

Permit "neutral" edge as a shorthand symbol for: 1) path being a word of $\mathcal{L}_P$ the length of which is equal to 2; 2) the element of $\mathcal{P}$ consisting of the edges of this path.

Construct V and $\mathcal{P}$ progressively:

(i) assign $\emptyset$ to $\mathcal{P}$; assign $\{[(_0, Z_0], [)_0, Z_0]\}$ to V;

(ii) while the following actions augment V, repeat them:

– assign $\{([(_m, Z]\frac{\Lambda}{+m}[(_k, m], [)_l, m]\frac{\Lambda}{-m}[)_m, Z]), [)_k, m]\Lambda[(_l, m] \,|[(_m, Z] \in V;$ (k,l,m) is an alliance $\} \cup \{[(_m, Z]a[)_m, Z]|[(_m, Z] \in V;\ a \in \Sigma;\ (_m a)_m$ is an active nest $\} \cup \{[(_m, Z]\Lambda[)_m, Z]|[(_m, Z] \in V;\ (_m\epsilon)_m$ is an active nest $\}$ to $NewP$;

– assign $\mathcal{P} \cup NewP$ to $\mathcal{P}$;

– assign $V \cup \{P | \exists \nu \in NewP$ (an edge of ν has P as its vertex) $\}$ to V.

The function λ is defined by way of the ignorance of edge charge.

$D(\zeta)$ can be considered as the graph of the PDA

$$M(\zeta) = (\mathcal{B}(Augm(\zeta)), \Sigma, Names(Augm(\zeta)) \cup \{Z_0\}, Z_0, \delta(\zeta), (_0, \{)_0\}),$$

where

$$\begin{aligned}\delta(\zeta) &= \{(p,a,Z) \to (q,ZX)|(p,Z)\frac{a}{+X}(q,X) \in E(D(\zeta))\}\\ &\cup \{(p,a,Z) \to (q,\Lambda)|\exists X \in N(\zeta)\ (p,Z)\frac{a}{-Z}(q,X) \in E(D(\zeta))\}\\ &\cup \{(p,a,Z) \to (q,Z)|(p,Z)a(q,Z) \in E(D(\zeta))\}.\end{aligned}$$

To verify the equivalence of ζ and $D(\zeta)$, define objects that can be correlated with paths of $D(\zeta)$.

DEFINITION 24. Let ζ be a CF-expression. An element of the recursively defined set $Paths(\zeta) = Clan(\zeta) \cup \{x[({}_m u]^k ({}_m y)_m [v)_m]^l z | k, l \geqslant 0, x({}_m u({}_m y)_m v)_m z \in Paths(\zeta)\}$ is called a path of ζ.

For $b_1, b_2 \in \mathcal{B}(Augm(\zeta))$ let

$$DisBal(b_1, b_2, 0) = \begin{cases} \{b_1\}, & b_1 = b_2, \\ \emptyset, & b_1 \neq b_2, \end{cases}$$

$DisBal(b_1, b_2, s) = \{b_1 y b_2 | s = |projection(b_1 y b_2, \mathcal{B}(Augm(\zeta)))| - 1;\ \exists(x, z \in Alph(Augm(\zeta))^*)\ x b_1 y b_2 z \in Paths(Augm(\zeta))\}$ for $s \geqslant 0$. Let $DisBal(b_1, b_2) = \cup_{s \geqslant 0} DisBal(b_1, b_2, s)$.

For $P, Q \in V$ let

$$\mathcal{T}(P, Q, 0) = \begin{cases} \{P\}, & P = Q, \\ \emptyset, & P \neq Q, \end{cases}$$

$\mathcal{T}(P, Q, s) = \{T\pi | \pi \in E(D(\zeta));\ end(\pi) = Q;\ T \in \mathcal{T}(P, beg(\pi), s-1)\}$ for $s > 0$. Let $\mathcal{T}(P, Q) = \cup_{s \geqslant 0} \mathcal{T}(P, Q, s)$.

For any vertex $P = (p, Z)$, let $state(P)$ denote the state symbol p. Define a morphism

$$\Phi : E(D(\zeta))^* \to (\{state(beg(\pi)) | \pi \in E(D(\zeta))\}(\Sigma \cup \{\Lambda\}))^*$$

by $\Phi(\pi) = state(beg(\pi))\omega(\pi)$.

Lemma 7. *For any $s \geqslant 0$ and $P, Q \in V$, if $\mathcal{T}(P, Q, s) \neq \emptyset$, then there exists a bijection*

$$\varphi : \mathcal{T}(P, Q, s) \to DisBal(state(P), state(Q), s),$$

such that every $T \in \mathcal{T}(P, Q, s)$ satisfies the equality

$$\Phi(T) state(end(T)) = \varphi(T).$$

Proof. By induction on s. If $s = 0$, then the lemma follows from the definitions of $\mathcal{T}(P, Q, 0)$ and $DisBal(state(P), state(Q), 0)$. Let $s > 0$. Assume that for any $0 \leqslant s' < s$ and $P', Q' \in V$ satisfying the condition $\mathcal{T}(P', Q', s') \neq \emptyset$ there exists a bijection

$$\varphi' : \mathcal{T}(P', Q', s') \to DisBal(state(P'), state(Q'), s'),$$

such that every $T \in \mathcal{T}(P', Q', s')$ satisfies the equality

$$\Phi(T) state(end(T)) = \varphi'(T).$$

Let $\mathcal{T}(P, Q, s) \neq \emptyset$. By construction of elements of $E(\zeta)$ $\mathcal{T}(P, Q, s)$ is equal to one of the following unions:

$\{[(_m, Z]\frac{\Lambda}{+m}[(_k, m]T | P = [(_m, Z];\ (\exists l \in Names(Augm(\zeta))\ (k, l, m)$ is an alliance); $T \in \mathcal{T}([(_k, m], Q, s-1)\} \cup \{[(_m, Z]a[)_m, Z]T | P = [(_m, Z];\ a \in \Sigma;\ (_m a)_m$ is an active nest; $T \in \mathcal{T}([)_m, Z], Q, s-1)\} \cup \{[(_m, Z]\Lambda[)_m, Z]T | P = [(_m, Z];\ (_m \epsilon)_m$ is an active nest; $T \in \mathcal{T}([)_m, Z], Q, s-1)\}$;

$\{[)_k, m]\Lambda[(_l, m]T | P = [)_k, m];\ (k, l, m)$ is an alliance; $T \in \mathcal{T}([(_l, m], Q, s-1)\}$ $\cup \{[)_k, m]\frac{\Lambda}{-m}[)_m, Z]T | P = [)_k, m];\ (\exists l \in Names(Augm(\zeta))\ (l, k, m)$ is an alliance); $T \in \mathcal{T}([)_m, Z], Q, s-1)\}$.

Let $p = state(P)$, $q = state(Q)$. By construction of $Augm(\zeta)$ $DisBal(p, q, s)$ is equal to one of the following unions:

$\{(_m(_k y | p = (_m;\ \exists l \in Names(Augm(\zeta))\ (k, l, m)$ is an alliance; $(_k y \in DisBal((_k, q, s-1)\} \cup \{(_m a)_m y | p = (_m;\ a \in \Sigma \cup \{\epsilon\};\ (_m a)_m$ is an active nest; $)_m y \in DisBal()_m, q, s-1)\}$;

$\{)_k(_l y | p =)_k;\ \exists m \in Names(Augm(\zeta))\ (k, l, m)$ is an alliance; $(_l y \in DisBal((_l, q, s-1)\} \cup \{)_k)_m y | p =)_k;\ \exists l \in Names(Augm(\zeta))\ (l, k, m)$ is an alliance; $)_m y \in DisBal()_m, q, s-1)\}$.

Consequently, the existence of φ' implies that the function φ defined by

$$\varphi(\pi T) = \Phi(\pi)\varphi'(T),\ \pi T \in \mathcal{T}(P, Q, s)$$

is an appropriate bijection.

Lemma 7 implies the following proposition.

Consequence 1. Let $P, Q \in V$. If $\mathcal{T}(P, Q) \neq \emptyset$, then there exists a bijection

$$\varphi : \mathcal{T}(P, Q) \to DisBal(state(P), state(Q)),$$

such that any $T \in \mathcal{T}(P, Q)$ satisfies the equality $\Phi(T)state(end(T)) = \varphi(T)$.

Consequence 2. $L(D(\zeta)) = L(\zeta)$.

Proof. By Consequence 1 there exists a bijection

$$\varphi : Sentences(D(\zeta)) \to Clan(Augm(\zeta)),$$

such that every sentence T satisfies the condition $\Phi(T)state(end(T)) = \varphi(T)$. Hence, $\omega(T) = projection(\varphi(T), \Sigma)$. So,

$$L(D(\zeta)) = L(Augm(\zeta)).$$

Now by Lemma 5 $L(D(\zeta)) = L(\zeta)$.

3.3. *Coiterating and Pseudo-coiterating CF-expressions*

The following Definition 27 breaks CF-expressions into two subclasses.

DEFINITION 25. Let w be a cycle within a fraction over Σ,

$$projection(w, \Sigma) \in \Sigma^+.$$

Then w is called a labelled cycle.

DEFINITION 26. Let $\xi = u(_m x(_m y)_m z)_m v$ is a fraction, both the cycles $(_m x$ and $z)_m$ be labelled. Then ξ is called an inserting fraction.

DEFINITION 27. Let ζ be a CF-expression. If $Clan(\zeta)$ has an inserting element, then ζ is called a coiteraing expression, else pseudo-coiterating.

Lemma 8. *If ζ is a coiterating expression, then there exists an inserting fraction $\xi \in Clan(\zeta)$ such that $height(\xi) \leqslant 5$.*

Proof. Suppose that ξ' is an inserting element of $Clan(\zeta)$. By Definition 26 $\xi' = u'(_m x'_1 a x'_2(_m y')_m z'_1 b z'_2)_m v'$ for some name $m \in Names(\zeta)$, words $u', x'_1, x'_2, y', z'_1, z'_2, v' \in Alph(\zeta)^*$, and symbols $a, b \in \Sigma$. Let

$$\begin{aligned}\xi = {} & u\ \ (_m x_1 a x_2 (_m y)_m z_1 b z_2)_m v) \\ & \in reduction(u', (_m, x'_1, a, x'_2, (_m, y',)_m, z'_1, b, z'_2,)_m, v')\end{aligned}$$

where $s \in \{u, x_1, x_2, y, z_1, z_2, v\}$ denotes the image of s' under this reduction.

Consideration of all possible positions of parentheses $(_k,)_k$, $k \in Names(\zeta)$, within ξ leads to the conclusion that $h(k, \xi)$ reaches its maximum in the following cases:

$$\begin{aligned}
\xi &= u_1(_k u_2(_m x_{11}(_k x_{12} a x_{21}(_k x_{22}(_k x_{23}(_m y_1(_k y_2)_k \\
&\quad\ y_3)_m z_{11})_k z_{12} b z_{21})_k z_{22})_k z_{23})_m v_1)_k v_2, \quad k \neq m, \\
\xi &= u_1(_k u_2(_m x_{11}(_k x_{12}(_k x_{13} a x_{21}(_k x_{22}(_m y_1(_k y_2)_k \\
&\quad\ y_3)_m z_{11})_k z_{12})_k z_{13} b z_{21})_k z_{22})_m v_1)_k v_2, \quad k \neq m, \\
\xi &= u(_m x_{11}(_m x_{12} a x_2(_m y)_m z_{11})_m z_{12} b z_2)_m v, \quad k = m, \\
\xi &= u(_m x_1 a x_{21}(_m x_{22}(_m y)_m z_1 b z_{21})_m z_{22})_m v, \quad k = m.
\end{aligned}$$

In any case $h(k, \xi) \leqslant 5$, $k \in Names(\zeta)$, i.e., $height(\xi) \leqslant 5$. The cycles $(_m x_1 a x_2$, $z_1 b z_2)_m$ are labelled, as $projection((_m x_1 a x_2, \Sigma)$ and $projection(z_1 b z_2)_m, \Sigma)$ contain a and b respectively. Consequently, ξ is an inserting fraction.

Lemma 9. *CF-expression ζ is pseudo-coiterating iff every element of $Core(\zeta, 5)$ is not an inserting fraction.*

Proof. Let ζ be a pseudo-coiterating CF-expression. Then $Clan(\zeta)$ does not have an inserting element. But $Core(\zeta, 5)$ forms a subset of $Clan(\zeta)$.

Now let $Core(\zeta, 5)$ contain no inserting fractions. Then Lemma 8 implies that every element of $Clan(\zeta)$ does not satisfy Definition 26. Consequently, ζ is pseudo-coiterating CF-expression.

Theorem 4. *If ζ is a pseudo-coiterating CF-expression, then $L(\zeta)$ is regular.*

Proof. Let the finite automaton $\mathcal{A}(\zeta)$ have the set of the edges $\{\lambda(\pi)|\pi \in E(D(\zeta))\}$, the input vertex $[(_0, Z_0]$, and the output vertex $[)_0, Z_0]$. Then $L(\zeta) = L(\mathcal{A}(\zeta))$. Indeed, the paths of a pseudo-coiterating ζ can be used instead of the fractions for the definition of $L(\zeta)$:

$$L(\zeta) = \cup_{\xi \in Paths(\zeta)} \omega(\xi).$$

Note that $Paths(\zeta)$ and $Paths(Augm(\zeta)) = DisBal((_0,)_0)$ define the same language. By Consequence 1 there exists a bijection

$$\varphi : \mathcal{T}([(_0, Z_0], [)_0, Z_0]) \to DisBal((_0,)_0)$$

such that every $T \in \mathcal{T}([(_0, Z_0], [)_0, Z_0])$ satisfies the condition

$$\Phi(T) state(end(T)) = \varphi(T).$$

The last equality implies $\omega(\varphi(T)) = \{\omega(T)\}$. But $\mathcal{T}([(_0, Z_0], [)_0, Z_0])$ was converted to the set of all successful paths of $\mathcal{A}(\zeta)$ by means of the deletion of edge charges. Consequently, $L(\zeta) = \{\omega(T)|T \in \mathcal{T}([(_0, Z_0], [)_0, Z_0])\} = L(\mathcal{A}(\zeta))$.

4. Inserting Nests and Language Regularity Condition

Define a pseudo-coiterating D-graph and pseudo-coiterating PDA. The last notion was defined in (Vylitok, 1998) by a different way.

DEFINITION 28. Let D be a D-graph, and T be its sentence such that there exist cycles T_1, T_2 forming a nest in T and satisfying the inequalities $\omega(T_1) \neq \Lambda$, $\omega(T_2) \neq \Lambda$. Then this nest is called an inserting nest of D, and D is called a coiterating D-graph. A D-graph without inserting nests is called a pseudo-coiterating D-graph.

DEFINITION 29. If a PDA M defines an inserting nest, then M is called a coiterating PDA, else it is called a pseudo-coiterating PDA.

Theorem 5. *If D is a pseudo-coiterating D-graph, then $Expr(D)$ (see Section 3.1) is a pseudo-coiterating CF-expression.*

Proof. Note that $\mathcal{E}(D)$ is a coiterating CF-expression iff D has cycles forming a nest. Indeed, let a fraction $\xi = x_1 (_\iota x_2 (_\iota x_3)_\iota x_4)_\iota x_5 \in Clan(\mathcal{E}(D))$ be such that $\omega(x_2) \neq \Lambda$

and $\omega(x_4) \neq \Lambda$. Then, by the construction of schemes, $(_\iota x_2 = y_1\pi_1 y_2, x_4)_\iota = y_3\pi_2 y_4$, and

$$projection(\pi_1 y_2 (_\iota x_3)_\iota y_3 \pi_2, E(D))$$

is one of the nests of the sentence corresponding to ξ.

Hence, the fraction $\psi = x_1[(_\iota x_2]^2(_\iota x_3)_\iota [x_4)_\iota]^2 x_5 \in Clan(\mathcal{E}(D))$ corresponds to the sentence having the nest

$$projection(\pi_1 y_2 y_1 \pi_1 y_2 (_\iota x_3)_\iota y_3 \pi_2 y_4 y_3 \pi_2, E(D))$$

that is formed by cycles $\pi_1 y_2 y_1$, $y_4 y_3 \pi_2$.

The fraction $\varphi(\psi) \in Clan(Expr(D))$ ought to satify the equalities $\omega(projection(\varphi(x_2), E(D))) = \omega(projection(\varphi(x_4), E(D))) = \Lambda$. The contrary would imply that D is not a pseudo-coiterating D-graph. Consequently, $Expr(D)$ is a pseudo-coiterating CF-expression.

Theorem 5 and Consequence 2 imply

Theorem 6. *If D is a pseudo-coiterating D-graph, then $L(D)$ is a regular set.*

The following important theorem was proved in (Vylitok, 1998) in terms of PDAs. It implies a partial algorithm for regularity testing.

Theorem 7. *Let D be a D-graph. It defines an inserting nest iff so does an element of $Core(D, 5, 5)$.*

Thus, we have the algorithmically checked condition of CF-language regularity. This condition is not necessary. Indeed, constructed in (Stanevichene, 1994) coiterating unambiguous deterministic PDA and corresponding D-graph define a regular set.

References

Stanevichene, L.I. (1997). D-graphs in context-free language theory. *Informatica*, **8**, 0–0.

Stearns, R.E. (1967) . A regularity test for pushdown machines. *Information and Control*, **11**, 323–340.

Vylitok, A.A. (1998). Pushdown automata and characterization of regular languages. *Candidate thesis*. Moscow State University (in Russian).

Stanevichene, L.I. (1994). A recursively unsolvable problem in deterministic pushdown acceptors. *Reports of the Russian Academy of Sciences* (This journal issues in English as *Doklady Mathematics*), **6**, 744–746 (in Russian).

L. Stanevichene received the Degree of Candidate of Physical and Mathematical Sciences from the Computer Center of the USSR Academy of Sciences in 1972. She is a senior tutor of the Computational Mathematics and Cybernetics Department of the Moscow State University. Her research interests include formal language theory and its applications.

A. Gomozov is a student of the Computational Mathematics and Cybernetics Department of the Moscow State University. He works under the supervision of L. Stanevichene.

Reguliariųjų reiškinių apibendrinimas

Aleksei GOMOZOV, Larisa STANEVIČIENĖ

Straipsnyje pateikta regualiariuosius reiškinius apibendrinančių nuo konteksto nepriklausančių reiškinių apibrėžtis. Parodyta, kad vadinamieji pseudo-ko-iteraciniai nuo konteksto nepriklausantys reiškiniai charakterizuoja reguliariąsias aibes. Šis rezultatas panaudotas kitoms reguliariųjų aibių charakterizacijoms, pseudo-ko-iteraciniams D grafams ir pseudo-ko-iteraciniams automatams su steko atmintimi, formuluoti. Sudarytas algoritmas, leidžiantis nustatyti, ar aukščiau nurodytų tipų charakterizacijos yra pseudo-ko-iteracinės, ar ne.