KYBERNETIKA — VOLUME 21 (1985), NUMBER 5

TWO THEOREMS ABOUT GALIUKSCHOV SEMICONTEXTUAL LANGUAGES

GHEORGHE PĂUN

We solve an open problem formulated in [1] (there are semicontextual grammars of degree two which generate non-context-free languages) and we extend a result in [1], concerning the closure properties of semicontextual languages families (all of them are *anti-AFL*'s).

1. DEFINITIONS AND TERMINOLOGY

We assume the reader familiar with the basic notions of formal language theory (from [3], for example) and we specify only some notions about the semicontextual grammars introduced in [1], under linguistic motivations.

A *semicontextual grammar* is a triple $G = (V, B, P)$, where V is a nonempty finite alphabet, B is a finite language over V and P is a finite set of rewriting rules of the form $xy \to xzy$, x, y, z being non-null strings over V. If $w = uxyv$ and $w' = uxzyv$ are two strings in V^* (V^* is the free monoid generated by V under the concatenation operation and the null element λ) and $xy \to xzy$ is a rule in P, then we write $w \Rightarrow w'$. We denote by $\Rightarrow^*$ the reflexive transitive closure of the relation $\Rightarrow$ and define the language generated by G as

$$L(G) = \{x \in V^* \mid z \Rightarrow^* x \text{ for some } z \text{ in } B\}.$$

Remark. In [1], instead of the set B, a semicontextual grammar contains a start symbol I and a finite set of rules of the form $I \to x$, x in V^*, which begin each derivation. Clearly, our modification is quite non-essential. Moreover, in [1] one defines some different variants of semicontextual grammars, but we do not consider them here.

A semicontextual grammar G as above is said to be *of degree m* if

$$m = \max\{|x| \mid xy \to xzy \text{ or } yx \to yzx \text{ is a rule in } P\}$$

($|x|$ is the length of the string x). We denote by $\mathcal{S}_i$, $i \geqq 1$, the family of languages generated by semicontextual grammars of degree not greater than i.

In [1] it is proved that $\mathcal{S}_1$ is a proper subset of the family of context-free languages and that $\mathcal{S}_1$ is an *anti-AFL* (it is not closed under none of the six *AFL* operations: union, concatenation, Kleene closure, λ-free homomorphisms, intersection with regular sets and inverse homomorphisms) and one asks whether $\mathcal{S}_2$ contain non-context-free languages.

In [2] it is proved that $\mathcal{S}_4$ contains non-context-free languages and the same result has been obtained in the meantime by B. S. Galiukschov for $\mathcal{S}_3$ (personal communication). Here we settle the question by finding a non-context-free language in $\mathcal{S}_2$ and also we prove that each family $\mathcal{S}_i$, $i \geqq 1$, is an *anti-AFL* (in fact, we find even a non-semilinear language in $\mathcal{S}_2$, that is a language having a non-semilinear Parikh image).

2. RESULTS

Theorem 1. The family $\mathcal{S}_2$ contains non-context-free languages.

Proof. We consider the following semicontextual grammar of degree two:

$$G = (\{a, b, c, d, f, g\}, \{fabcdf\}, P)$$

with the set P containing the rules:

1) $f\,ab \to f\,ga\,ab$
$aa\,bc \to aa\,b\,bc$
$bb\,cd \to bb\,c\,cd$
$cc\,da \to cc\,d\,da$
$dd\,ab \to dd\,a\,ab$
$cc\,df \to cc\,d\,df$

(Starting from the substring fab of the current string, these rules double each occurrence of symbols a, b, c, d, step-by-step, from the left to the right. Please note that – excepting the first rule – each rule has the form $xy \to xzy$ with $x = \alpha\alpha$, $\alpha \in \{a, b, c, d\}$, and y belongs to the set $\{ab, bc, cd, da\}$ – excepting the last rule, for which $y = df$. The pairs ab, bc, cd, da are called *legal*; they are the only two-letters substrings of a string of the form $(abcd)^n$.

Clearly, starting from a string of the form $wf(abcd)^n f$ (initially we have $w = \lambda$ and $n = 1$), we can pass to a string

$$(*) \qquad wfg(aabbccdd)^m xy(abcd)^p f$$

with $m \geqq 0$, $p \geqq 0$, $m + p + 1 = n$, y is a suffix of $abcd$, $abcd = zy$ and x is obtained by doubling each symbol in z. When $m = n - 1$ and $y = \lambda$, then we obtain the string $wfg(aabbccdd)^n f$, hence the length of the string obtained beetwen g and f is equal to $8n$, two times the length of the initial string $(abcd)^n$.)

2) $g\,aa \to g\,c\,aa$
$ca\,a \to ca\,c\,a$

$ca\ bb \to ca\ d\ bb$
$db\ b \to db\ d\ b$
$db\ cc \to db\ a\ cc$
$ac\ c \to ac\ a\ c$
$ac\ dd \to ac\ b\ dd$
$bd\ d \to bd\ b\ d$
$bd\ aa \to bd\ c\ aa$

(Starting from the substring gaa, hence from the symbol g introduced by the rules of group 1, these rules replaces each substring $\alpha\alpha$, $\alpha \in \{a, b, c, d\}$, by $\beta\alpha\ \beta\alpha$, $\beta \in \in \{a, b, c, d\}$, in such a way that all pairs $\beta\alpha$, $\alpha\beta$ are not legal. In view of the fact that – excepting the first rule – all the rules in group 2 are of the form $xy \to xzy$ with x a non-legal pair, it follows that these rules can be applied only in a step-by-step manner, from the left to the right. As each rule $xy \to xzy$ as above contains pair $\alpha\alpha$, $\alpha \in \{a, b, c, d\}$, in the string xy, it follows that they can be applied only after the rules of group 1 have been applied. Consequently, from a string of the form $(*)$, using the rules of group 2, we can pass to a string of the form

$$(**) \qquad wfg(cacadbdbacacbdbd)^r uv(aabbccdd)^s xy(abcd)^p f$$

with $0 \leqq r \leqq m$, $r + s + 1 = m$, v is a suffix of $aabbccdd$ and u is obtained by "translating" the string z for which $zv = aabbccdd$ by means of the rules in group 2, or to a string of the form

$$wfg(cacadbdbacacbdbd)^m x'y(abcd)^p f$$

where x' is obtained from a prefix of x by "translating" it using the above rules.

Let us note that the rules of group 2 also double the number of the symbols in the substring they "translate", therefore, when the string $(*)$ is of the form $wfg(aabbccdd)^n f$, then we can obtain a string $wfg(cacadbdbacacbdbd)^n f$, that is with the substring bounded by g and f of length $16n$, two times the length of $(aabbccdd)^n$ and four times the length of the initial string $(abcd)^n$.)

3) $b\ df \to b\ c\ df$
$d\ bc \to d\ a\ bc$
$b\ da \to b\ c\ da$
$c\ bc \to c\ a\ bc$
$c\ ab \to c\ d\ ab$
$a\ cd \to a\ b\ cd$
$a\ da \to a\ c\ da$
$b\ ab \to b\ d\ ab$
$d\ cd \to d\ b\ cd$
$gc\ ab \to gc\ f\ ab$

(All the above rules are of the form $xy \to xzy$ with y a legal pair, or $y = df$ in the first rule. Moreover, excepting the last rule, each rule has $y = y_1y_2$, $y_1, y_2 \in \{a, b, c, d\}$,

$x \in \{a, b, c, d\}$, and xy_1 is a non-legal pair. Each rule introduces a symbol z between x and y in such a way that xy_1 is a legal pair. Consequently, the rules of group 3 can be applied only in the step-by-step manner, from the right to the left, starting either from the rightmost symbol f – by the first rule – or from the rightmost position where the rules of group 2 have been applied; indeed, only in that position appears a three-letters substring xy_1y_2 as above, with xy_1 a non-legal pair and y_1y_2 a legal pair. Using the above rules we obtain only legal pairs, therefore we pass to a string containing substrings *abcd*.

As both groups of rules 1 and 2 need substrings $\alpha\alpha$, $\alpha \in \{a, b, c, d\}$, in order to can be used, it follows that the rules of group 1 can be applied only after "legalizing" all pairs of symbols, hence only after using the last rule of group 3, which introduces a new occurrence of the symbol f and the first rule of group 1 can be applied.

The application of rules in group 3 again doubles the length of the "translated" string. Consequently, a string of the form (**) is transformed by rules in group 3 into

$$wfgcf(abcd)^{8r}u'v(aabbccdd)^s xy(abcd)^p f,$$

where u' is obtained from u in the above manner. When the string $wfg(aabbccdd)^n f$ has been transformed into $wfg(cacadbdbacacbdbd)^n f$ by means of rules in group 2, then the above group of rules provides the string $wfgcf(abcd)^{8n} f$.

Clearly, after using the rules of group 3 as many times as possibly, the derivation can be reiterated, using again the rules of group 1.)

The above grammar generates a non-context-free language. In fact, the language $L(G)$ is even non-semilinear.

Indeed, the following *assertion* is obvious. For each semilinear set $E \subseteq N^n$ and for each i, j, $1 \leqq i < j \leqq n$, either there is a constant $k_{i,j}$ such that $u_j/u_i \leqq k_{i,j}$, or there exist n-uples $(u_1, \ldots, u_{i-1}, u, u_{i+1}, \ldots, u_n)$ in E with given u and arbitrarily many u_j (and arbitrary u_k, $k \neq i$, $k \neq j$).

Let us consider the Parikh mapping Ψ_V associated to the alphabet $V = \{g, a, b, c, d, f\}$ (please note the order). The above assertion is not true for the set $\Psi_V(L(G))$. Indeed, let us consider the positions 1, 2 (corresponding to symbols g, a) of 6-tuples in $\Psi_V(L(G))$. From the above explanations, one can see that the rules in groups 1, 2, 3 can be applied only in this order; at each such step one introduces one symbol g and some symbols a such that from a string x one passes to a string y with at most 8 times more occurrences of the symbol a. Consequently, each 6-tuple $(u_1, u_2, u_3, u_4, u_5, u_6) \in \Psi_V(L(G))$ has $u_1 \leqq u_2 \leqq 8^{u_1}$. As the ratio $8^{u_1}/u_1$ can be arbitrarily large, but for each given u_1 the component u_2 cannot have arbitrarily large values, it follows that the mentioned assertion is not fulfilled, hence $\Psi_V(L(G))$ is not semilinear, and, in conclusion, $L(G)$ is not a context-free language. □

Corollary. Each family $\mathscr{S}_1$, $i \geqq 2$, is incomparable with each of the families of regular, linear and context-free languages.

The result follows from the above theorem, the inclusions $\mathscr{S}_i \subseteq \mathscr{S}_{i+1}$, $i \geqq 1$, and the fact that for each $\mathscr{S}_i$, $i \geqq 1$, there is a regular language L_i such that $L_i \notin \mathscr{S}_i$ [2] (such a regular language appears also in the proof of the next theorem).

Theorem 2. Each family $\mathscr{S}_i$, $i \geqq 1$, is an *anti-AFL*.

Proof. *Union.* Let us consider the languages

$$L_0 = \{a^n \mid n \geqq 1\}\,,$$
$$L_i = \{a^{2i}ba^{2i}b\}\,, \quad i \geqq 1\,.$$

The grammars $G_0 = (\{a\}, \{a, aa\}, \{aa \to aaa\})$, respectively, $G_i = (\{a, b\}, \{a^{2i}ba^{2i}b\}, \emptyset)$, generate these languages, hence $L_i \in \mathscr{S}_1$, $L_0 \in \mathscr{S}_1$. Let us consider the language $L_0 \cup L_i$ and suppose that it is generated by a semicontextual grammar of degree i, $G = (\{a, b\}, B, P)$. In order to generate the strings a^n with arbitrarily large n we need at least a rule of the form $a^k a^j \to a^k a^r a^j$, $k, j, r > 0$, $k \leqq i$, $j \leqq i$. This rule can be applied to the string $a^{2i}ba^{2i}b$, hence we obtain the string $a^{2i}ba^{2i+r}b$, which is not in $L_0 \cup L_i$, hence $L(G) \neq L_0 \cup L_i$ and $L_0 \cup L_i \notin \mathscr{S}_i$.

Concatenation. The language L_iL_0 does not belong to the family $\mathscr{S}_i$ and this assertion can be proved as previously (in order to generate strings $a^{2i}ba^{2i}ba^n$ with arbitrary large n we need rules of the form $a^k a^j \to a^k a^r a^j$, $k, j, r > 0$, $k \leqq i$, $j \leqq i$, or of the form $a^k b a^j a^p \to a^k b a^j a^r a^p$, $k + 1 + k > 0$, $p > 0$, $k + 1 + j \leqq i$, $p \leqq i$, $r > 0$, and each such rule can be used in a derivation $a^{2i}ba^{2i}ba \Rightarrow a^{2i}ba^{2i+r}ba$.

Kleene closure. The language

$$M_i = \{ba^k ba^{2i} \mid k \geqq 1\}$$

belongs to $\mathscr{S}_1$ (it is generated by the grammar having $B = \{baba^{2i}\}$ and the rule $ab \to aab$), but the language $M_i^* - \{\lambda\}$ does not belong to $\mathscr{S}_i$ (in order to generate arbitrarily long substrings a^k we need at least a rule of the form $xy \to xa^r y$, $x, y \in \{a, b\}^*$, $r > 0$, $|x| \leqq i$, $|y| \leqq i$; each such rule can be applied to substrings of the form $ba^j ba^{2i} ba^k ba^{2i}$ in order to introduce further occurrences of the symbol a in the subword $ba^{2i}b$ and in this way we obtain strings which are not in M_i^*).

Intersection with regular sets. Obvious, because there are regular languages which are not in $\mathscr{S}_i$, for each $i \geqq 1$ (see the previous points of the proof), but V^* is in $\mathscr{S}_1$ for any alphabet V.

λ-free homomorphisms. Let us consider the language

$$R_i = \{a^{2i}b^{2i}b\} \cup \{c^n \mid n \geqq 1\}$$

and the homomorphism h defined by $h(a) = h(c) = a$, $h(b) = b$. The grammar with $B = \{c, cc, a^{2i}ba^{2i}b\}$ and $P = \{cc \to ccc\}$, generates the language R_i, hence $R_i \in \mathscr{S}_1$. As $h(R_i) = L_i \cup L_0$ and this language is not in $\mathscr{S}_i$, it follows that $\mathscr{S}_i$ is not closed under λ-free homomorphisms.

Inverse homomorphisms. We take the language

$$L = \{(ab)^n (ba)^n \mid n \geqq 1\} \cup \{(ab)^n aa(ba)^n \mid n \geqq 1\}\,.$$

The grammar $G = (\{a, b\}, \{abba\}, \{bb \to babbab, bb \to baab\})$ generates the language L, hence $L \in \mathscr{S}_1$. We consider also the homomorphism h defined by $h(a) = ab$, $h(b) = a$. Clearly, $h^{-1}(L) = \{a^n b a^n b \mid n \geqq 1\}$ and this language is not in $\mathscr{S}_i$ for any i. Indeed, each string in $h^{-1}(L)$ contains two occurrences of the symbol b, hence each rule $xy \to xzy$ of a semicontextual grammar generating $h^{-1}(L)$ must have $z = a^p$, $p \geqq 1$. Using such a rule we can produce strings of the form $a^n b a^m b$ with $n \neq m$, which is a contradiction. The proof is over. □

Open problem. Is each regular language contained in $\bigcup_{i=1}^{\infty} \mathscr{S}_i$? (In [2] it is proved that each regular language is the homomorphic image of a language in $\mathscr{S}_1$.)

(Received April 5, 1983.)

REFERENCES

[1] B. S. Galiukschov: Semicontextual grammars. Mat. logica i mat. lingv., Kalinin Univ., 1981, 38—50. In Russian.

[2] Gh. Păun: On semicontextual grammars. Bull. Math. Soc. Sci. Math. R. S. Roumanie *28 (76)* (1984), 1, 63—68.

[3] A. Salomaa: Formal Languages. Academic Press, New York—London 1973.

Dr. Gheorghe Păun, Institute of Mathematics, Str. Academiei 14, 70109 Bucharest. Romania.