KYBERNETIKA — VOLUME 26 (1990), NUMBER 4

REGULATED GALIUKSCHOV SEMICONTEXTUAL GRAMMARS

MONICA MARCUS, GHEORGHE PĂUN

We consider matrix, programmed, random context and regular control semicontextual grammars in Galiukschov sense ([4]), with and without appearance checking. The generative capacity of such grammars is investigated, compared with non-restricted semicontextual grammars and with Chomsky grammars.

1. INTRODUCTION

The semicontextual grammars were introduced in [4] under linguistical motivation and they were investigated in [1], [6], [7], [8] from formal language theory point of view. These grammars are interesting counterparts of context sensitive Chomsky grammars (based on the process of rewriting nonterminals in given contexts) and of Marcus contextual grammars ([5]) (in which contexts are adjoined to strings), as they combine modified versions of such grammars: in Galiukschov grammars *strings* are *adjoined* in given *contexts*. The formal study of these grammars is therefore quite natural and the present paper relates the semicontextual grammars to an extensively investigated topic of formal language theory, namely the regulated rewriting (see [3] for a survey of the domain). Four of the most known regulating mechanisms are considered here, the matrix, the programmed, the regular control and the random context restrictions. Part of the results are the expected ones: all these mechanisms increase the generative capacity, whereas when adding the appearance checking feature the power is more increased. However, the four mechanisms do not lead to equivalent classes of regulated semicontextual grammars (as in the case of Chomsky context-free grammars; the same conclusion has been obtained for pure grammars in [2]). Some extensions of previous results about semicontextual grammars and languages are obtained too: infinite hierarchies introduced by grammars degree, inclusions into context-free languages family of random context semicontextual languages family of degree 1 (this extends a Galiukschov theorem).

2. SEMICONTEXTUAL GRAMMARS

A *semicontextual grammar* is a system $G = (V, B, P)$, where V is a (finite and nonempty) vocabulary, B is a finite language over V, and P is a finite set of rewriting rules of the form $xy \to xzy$, with $x, y, z \in V^*, x, y \neq \lambda$ (V^* is the free monoid generated by V under the operation of concatenation and the null element λ).

If $w, w' \in V^*$, $w = uxyv$, $w' = uxzyv$, $u, v \in V^*$, and $xy \to xzy$ is a rule in P, then we write $w \Rightarrow w'$. Denote by $\Rightarrow^*$ the reflexive transitive closure of this relation. The language generated by G is

$$L(G) = \{w \in V^* \mid \text{there is } u \in B \text{ such that } u \Rightarrow^* w\}$$

We denote

$$deg\,(G) = \max\{|x| \mid \text{there is } y \in V^* \text{ such that } xy \to xzy \in P \text{ or } yx \to yzx \in P\}$$

where $|x|$ denotes the length of the string x. This is the *degree* of the grammar G. Then we define

$$\mathcal{S}_k = \{L \mid L = L(G), deg(G) \leqq k\}, \quad k \geqq 1,$$
$$\mathcal{S}_\infty = \bigcup_{k \geqq 1} \mathcal{S}_k$$

(the family of languages generated by grammars of degree at most k and the family of all semicontextual languages, respectively).

The following results are known from [4], [6], [7], [8]:

$$\mathcal{S}_1 \subset \mathcal{S}_2 \subset \ldots \subset \mathcal{S}_\infty, \quad \text{strict inclusions,}$$
$$\mathcal{S}_1 \subset \mathcal{L}_2, \quad \text{strict inclusion},$$
$$\mathcal{S}_2 - \mathcal{L}_2 \neq \emptyset,$$
$$\{a^n b a^n \mid n \geqq 1\} \notin \mathcal{S}_\infty$$

($\mathcal{L}_i$, $i = 0, 1, 2, 3$, denote the families in Chomsky hierarchy).

3. REGULATED SEMICONTEXTUAL GRAMMARS

A *matrix semicontextual grammar* is a system $G = (V, B, M, F)$, where V is a vocabulary, B is a finite language over V, M is a finite set of sequences of the form

$$(x_1 y_1 \to x_1 z_1 y_1, \ldots, x_k y_k \to x_k z_k y_k), \quad k \geqq 1,$$

of semicontextual rules, and F is a set of occurrences of rules in M. For $w, w' \in V^*$ we write $w \Rightarrow w'$ if there are $w_1, \ldots, w_{k+1}$ in V^* and $(x_1 y_1 \to x_1 z_1 y_1, \ldots, x_k y_k \to x_k z_k y_k)$ in M such that $w = w_1$, $w_{k+1} = w'$ and for each i, $1 \leqq i \leqq k$, either

$w_i = u_ix_iy_iv_i$, $w_{i+1} = u_ix_iz_iy_iv_i$, or $w_i = w_{i+1}$, w_i does not contain the substring x_iy_i and $x_iy_i \to x_iz_iy_i$ appears in F.

The language generated by G is defined in the natural way.

A *programmed semicontextual grammar* is a system $G = (V, B, P)$, where V, B are as above and P is a finite set of quadruples of the form

$$(b\colon xy \to xzy,\ E(b),\ F(b))$$

where b is the label of this rule and $E(b)$, $F(b)$ are sets of labels of rules in P. A derivation in G has the form $(w_1, b_1) \Rightarrow (w_2, b_2) \Rightarrow (w_3, b_3) \Rightarrow \ldots \Rightarrow (w_n, b_n)$, where $w_1 \in B$, and for each i, $1 \leqq i \leqq n$, $w_i \in V^*$, b_i are labels of rules in P and $(w_i, b_i) \Rightarrow$ $\Rightarrow (w_{i+1}, b_{i+1})$ holds if either $w_i = u_ix_iy_iv_i$, $w_{i+1} = u_ix_iz_iy_iv_i$, for $(b_i\colon x_iy_i \to$ $\to x_iz_iy_i, E(b_i), F(b_i))$ in P and $b_{i+1} \in E(b_i)$, or $w_i = w_{i+1}$, w_i does not contain the substring x_iy_i for $(b_i\colon x_iy_i \to x_iz_iy_i, E(b_i), F(b_i))$ in P and $b_{i+1} \in F(b_i)$.

A *regular control semicontextual grammar* is a system $G = (V, B, P, C, F)$, where V, B are as above, P is a set of labelled semicontextual rules, C is a regular language over $Lab(P)$ (the set of labels for rules in P), and F is a subset of P. A derivation in G has the form

$$w_1 \Rightarrow^{r_1} w_2 \Rightarrow^{r_2} \ldots \Rightarrow^{r_n} w_{n+1}$$

where $r_1, \ldots, r_n$ are labels of rules in P, $r_1r_2 \ldots r_n \in C$ and $w_i \Rightarrow^{r_i} w_{i+1}$ if either $w_i =$ $= u_ix_iy_iv_i$, $w_{i+1} = u_ix_iz_iy_iv_i$, $r_i\colon x_iy_i \to x_iz_iy_i$, or $w_i = w_{i+1}$, w_i does not contain the substring x_iy_i and the rule $r_i\colon x_iy_i \to x_iz_iy_i$ appears in F.

A *random context semicontextual grammar* is a system $G = (V, B, P)$, where V, P are as above and P is a finite set of random context rules, that is triples of the form

$$(xy \to xzy,\ E,\ F)$$

with E, $F \subseteq V$. Such a rule can be applied to a string w in order to obtain a string w' if $w = uxyv$, $w' = uxzyv$, all symbols of E appear in w, but no symbol of F appears in w. (Note that we check the "random contexts" E, F in the whole string w, not in uv, outside the lefthand member of the applied rules, as usual [3].)

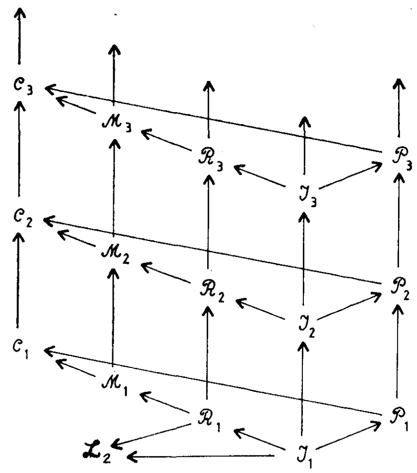
In all the four cases, the starting set B of strings is included in $L(G)$ — by definition or by convention.

For all these classes of grammars we can define the degree as for non-regulated semicontextual grammars. The families of languages generated by matrix, programmed, regular control and random context semicontextual grammars of degree at most k, $k \geqq 1$, are denoted by $\mathcal{M}_k(ac)$, $\mathcal{P}_k(ac)$, $\mathcal{C}_k(ac)$, $\mathcal{R}_k(ac)$, respectively; $\mathcal{M}_\infty(ac)$, $\mathcal{P}_\infty(ac)$, $\mathcal{C}_\infty(ac)$, $\mathcal{R}_\infty(ac)$ denote the families of languages as above, of arbitrary degree. When no appearance checking feature is present (that is $F = \emptyset$ in matrix and regular control grammars, $F = \emptyset$ in rules of random context grammars and $F(b) = \emptyset$ in rules of programmed grammars), then we write $\mathcal{M}_k$, $\mathcal{P}_k$, $\mathcal{C}_k$, $\mathcal{R}_k$, $\mathcal{M}_\infty$, $\mathcal{P}_\infty$, $\mathcal{C}_\infty$, $\mathcal{R}_\infty$, respectively, for the corresponding families.

4. THE GENERATIVE CAPACITY OF NON-APPEARANCE CHECKING CASE

The results of this section are summarized in the next theorem, proved in a series of lemmas.

Theorem 1. The diagram in Figure 1 holds, where each arrow denotes a strict inclusion and the unrelated families are incomparable.



$$\begin{tikzcd}
\, & & & & \\
\mathcal{C}_3 \arrow[u] & & & & \\
& \mathcal{M}_3 \arrow[ul] \arrow[uu] & & & \\
& & \mathcal{R}_3 \arrow[ul] \arrow[uuu] & & \mathcal{P}_3 \arrow[uu, llll] \arrow[uuu] \\
& & & \mathcal{J}_3 \arrow[ul] \arrow[ur] \arrow[uuuu] & \\
\mathcal{C}_2 \arrow[uuuu] & & & & \\
& \mathcal{M}_2 \arrow[ul] \arrow[uuuu] & & & \\
& & \mathcal{R}_2 \arrow[ul] \arrow[uuuu] & & \mathcal{P}_2 \arrow[uu, llll] \arrow[uuuu] \\
& & & \mathcal{J}_2 \arrow[ul] \arrow[ur] \arrow[uuuu] & \\
\mathcal{C}_1 \arrow[uuuu] & & & & \\
& \mathcal{M}_1 \arrow[ul] \arrow[uuuu] & & & \\
& & \mathcal{R}_1 \arrow[ul] \arrow[uuuu] \arrow[dll] & & \mathcal{P}_1 \arrow[uu, llll] \arrow[uuuu] \\
\mathcal{L}_2 & & & \mathcal{J}_1 \arrow[ul] \arrow[ur] \arrow[uuuu] \arrow[lll] &
\end{tikzcd}$$

Fig. 1.

Lemma 1. (i) $\mathcal{J}_k \subseteq \mathcal{X}_k$, $k \geqq 1$, for all $\mathcal{X} \in \{\mathcal{M}, \mathcal{P}, \mathcal{C}, \mathcal{R}\}$; (ii) $\mathcal{X}_k \subseteq \mathcal{X}_k(ac)$, $k \geqq 1$. for all $\mathcal{X} \in \{\mathcal{M}, \mathcal{P}, \mathcal{C}, \mathcal{R}\}$.

Proof. Obvious, from definitions. □

Lemma 2. $\mathcal{R}_1 - \mathcal{P}_k \neq \emptyset$, $k \geqq 1$.

Proof. Let us consider the language

$$L_k = \{ab^nc^n \mid n \geqq k\} \cup \{ab^nf^me^mc^n \mid n \geqq k, m \geqq 1\}$$
$$\cup \{db^nc^n \mid n \geqq k\} \cup \{db^ne^mf^mc^n \mid n \geqq k, m \geqq 1\}$$

The grammar

$$G = (\{a, b, c, d, e, f\}, \{ab^kc^k, db^kc^k\},$$
$$\{(bc \to bbcc, \emptyset), (bc \to bfec, \{a\}), (fe \to ffee, \emptyset),$$
$$(bc \to befc, \{d\}), (ef \to eeff, \emptyset)\})$$

generates L_k, hence $L_k \in \mathcal{R}_1$.

Suppose that $L_k = L(G')$, for some programmed grammar $G' = (V, B, P)$. As B is finite, only for finitely many n's we have $ab^nf^me^mc^n$ in B, $db^ne^mf^mc^n$ in B. Therefore we have to perform derivations of the form $ab^rc^r \Rightarrow^* ab^nc^n$, $db^rc^r \Rightarrow^* db^nc^n$, respectively, hence we need rules for introducing symbols e, f in strings not containing them. As $B \subseteq L_k$, each string in B contains the substrings b^k, c^k. There are infinitely many derivations of the form

$$D\colon (ab^{i_1}c^{i_1}, t_1) \Rightarrow^* (ab^{i_2}c^{i_2}, t_2) \Rightarrow (ab^{i_3}f^{j_1}e^{j_1}c^{i_3}, t_3) \Rightarrow^* (ab^{i_4}f^{j_2}e^{j_2}c^{i_4}, t_4)$$

and of the form

$$D'\colon (db^{i_1'}c^{i_1'}, t_1') \Rightarrow^* (db^{i_2'}c^{i_2'}, t_2') \Rightarrow (db^{i_3'}e^{j_1'}f^{j_1'}c^{i_3'}, t_3')$$
$$\Rightarrow^* (db^{i_4'}e^{j_2'}f^{j_2'}c^{i_4'}, t_4')$$

Assume $i_1' \geqq i_1$; the case $i_1 \geqq i_1'$ is similar. The rules in D can be applied in the same order to $db^{i_1'}c^{i_1'}$, thus obtaining a string of the form $db^if^je^jc^i$, which is not in L_k, contradiction. □

Corollary. $\mathcal{R}_1 - \mathcal{S}_k \neq \emptyset$ for all $k \geqq 1$.

Lemma 3. $\mathcal{M}_1 - \mathcal{P}_\infty \neq \emptyset$.

Proof. Consider the language

$$L = \{a^nba^nca^n \mid n \geqq 1\}$$

It can be generated by the matrix grammar of degree 1

$$G = (\{a, b, c\}, \{abaca\}, \{(ab \to aab, ba \to baa, ca \to caa)\})$$

Suppose $L = L(G')$, $G' = (V, B, P)$ being a programmed grammar and consider an effective derivation $w \Rightarrow^* a^nba^nca^n \Rightarrow a^mba^pca^q$, $m + p + q > 3n$. The used rule is of the form $xy \to xzy$, hence $z = a^i$, $i \geqq 1$, hence only one of m, p, q is different from n, that is $m \neq p$, or $p \neq q$; the obtained string is not in L, contradiction. □

Lemma 4. $\mathcal{R}_k \subset \mathcal{M}_k$, $k \geqq 1$, strict inclusion.

Proof. Let $G = (V, B, P)$ be a random context grammar of degree k and construct the matrix grammar

$$G' = (V, B, \{(\alpha_1 \to \alpha_1, \ldots, \alpha_m \to \alpha_m, xy \to xzy) \mid (xy \to xzy, \{a_1, \ldots, a_m\}) \in$$
$$\in P, \alpha_i \in \{a_ib, ba_i \mid b \in V\}, 1 \leqq i \leqq m\})$$

Clearly, $L(G) = L(G')$, hence $\mathcal{R}_k \subseteq \mathcal{M}_k$.

The language $L = \{a^nba^nca^n \mid n \geqq 1\}$ considered in the above lemma is not in $\mathcal{R}_\infty$. For, if $L = L(G'')$, G'' a random context grammar, then, as each string in L contains all occurrences of symbols in V, the grammar G'' can be considered a usual unrestricted semicontextual one. However, as we have proved, $L \notin \mathcal{P}_\infty$, contradiction. □

Corollary. $\mathcal{M}_1 - \mathcal{R}_\infty \neq \emptyset$.

Lemma 5. $\mathcal{P}_1 - \mathcal{M}_\infty \neq \emptyset$.

Proof. Consider the language

$$L = \{a^nb^n \mid n \geqq 1\} \cup \{a^nb \mid n \geqq 1\}$$

generated by the programmed semicontextual grammar

$$G = (\{a, b\}, \{ab\}, \{(1: ab \to a^2b^2, \{1\}), (2: ab \to a^2b, \{2\})\})$$

Suppose that $L = L(G')$, for a matrix grammar $G' = (V, B, M)$. A matrix which can be applied to a string a^nb (leading to a string a^mb, $m > n$) can also be applied to a string a^nb^n, thus obtaining a string a^mb^n, $m > n > 1$, which is not in L, contradiction. □

Corollary. $\mathscr{P}_1 - \mathscr{R}_\infty \neq \emptyset$.

Lemma 6. $\mathscr{M}_k \subset \mathscr{C}_k$, $\mathscr{P}_k \subset \mathscr{C}_k$, $k \geqq 1$, strict inclusions.

Proof. In view of Lemmas 3 and 5, it is enough to prove the inclusions, and this can be done in the standard way followed for Chomsky grammars (see [3], Chapters 2, 3). □

Lemma 7. $\mathscr{S}_{k+1} - \mathscr{C}_k \neq \emptyset$ for all $k \geqq 1$.

Proof. Consider the language

$$L_k = \{ba^{2k+1}\} \cup \{a^{2nk+1} \mid n \geqq 1\}$$

generated by the semicontextual grammar of degree $k + 1$

$$G = (\{a, b\}, \{ba^{2k+1}, a^{2k+1}, a^{4k+1}\},$$
$$\{a^{k+1}a^{k+1} \to a^{k+1}a^{2k}a^{k+1}\})$$

Suppose that $L_k = L(G')$, $G' = (V, B, P, C)$ being a semicontextual grammar of degree k with control language C. Clearly, the string ba^{2k+1} must be in B: each rule $xy \to xzy$ has $|x| \geqq 1$, $|y| \geqq 1$, hence we cannot add the symbol b in the left hand side of a string a^r. On the other hand, in order to obtain strings a^{2nk+1} with arbitrary n, we need derivations of the form $a^{2mk+1} \Rightarrow^* a^{2nk+1}$, for $a^{2mk+1} \in B$ and using rules $xy \to xzy$ with $x = a^i$, $y = a^j$, $1 \leqq i, j \leqq k$. The corresponding rules used in the order imposed by a string in C can be used also starting from ba^{2k+1}, thus obtaining string of the form ba^{2rk+1}, contradiction. □

Corollary. All inclusions $\mathscr{R}_k \subset \mathscr{R}_{k+1}$, $\mathscr{M}_k \subset \mathscr{M}_{k+1}$, $\mathscr{P}_k \subset \mathscr{P}_{k+1}$, $\mathscr{C}_k \subset \mathscr{C}_{k+1}$ are proper, $k \geqq 1$.

A problem of interest in this context is also the relation with $\mathscr{L}_2$, the family of context-free languages. In [7], [8] it is proved that $\mathscr{S}_2 - \mathscr{L}_2 \neq \emptyset$. The language $\{a^nba^nca^n \mid n \geqq 1\}$ in Lemma 3 is not context-free, hence $\mathscr{M}_1 - \mathscr{L}_2 \neq \emptyset$. A similar relation holds also for programmed semicontextual languages: the grammar $(\{a, b, c\}$, $\{abaca\}$, $\{(1: ab \to aab, \{2\}), (2: ba \to baa, \{3\}), (3: ca \to caa, \{1\})\})$ generates the non-context-free language

$$\{a^nba^nca^n, a^{n+1}ba^nca^n, a^nba^{n+1}ca^n, a^nba^nca^{n+1},$$
$$a^{n+1}ba^{n+1}ca^n, a^{n+1}ba^nca^{n+1}, a^nba^{n+1}ca^{n+1} \mid n \geqq 1\}$$

hence $\mathscr{P}_1 - \mathscr{L}_2 \neq \emptyset$. As we shall see in the next section, $\mathscr{R}_1 \subset \mathscr{L}_2$ (in fact, $\mathscr{R}_1(ac) \subset \subset \mathscr{L}_2$ too).

On the other hand, the linear language $\{a^nba^nba^m \mid n, m \geqq 1\}$ is not in $\mathscr{C}_\infty$: we need rules $xy \to xa^iy$ for modifying the suffix a^m; as $|x|$ is bounded, there are such rules with $x = a^rba^s$, or $x = a^r$, which can be applied in such a way to modify the second substring a^n in a string $a^nba^nba^m$ with arbitrarily large n and m, thus obtaining $a^nba^rba^m$, $n \neq r$.

5. THE GENERATIVE CAPACITY IN THE APPEARANCE CHECKING CASE

The relationships between the considered families of regulated semicontextual languages generated with or without appearance checking are summarized in the next theorem.

Theorem 2. The diagram in Figure 2 holds for all $k \geqq 1$ (the dotted lines point out to open problems).

$$\begin{tikzcd}
\mathscr{C}_k(ac) \arrow[ddrr, dashed, no head] \arrow[ddrrrr, dashed, no head] & & & & \\
& \mathscr{M}_k(ac) \arrow[ul] \arrow[dr, dashed, no head] & & & \\
& & \mathscr{R}_k(ac) & & \mathscr{P}_k(ac) \\
\mathscr{C}_k \arrow[uuu] & & & & \\
& \mathscr{M}_k \arrow[uuu] \arrow[ul] & & & \\
& & \mathscr{R}_k \arrow[uuu] \arrow[ul] & & \mathscr{P}_k \arrow[uuu] \arrow[uullll] \\
& & & \mathscr{I}_k \arrow[ul] \arrow[ur] &
\end{tikzcd}$$

Fig. 2.

We prove these relations (sometimes, stronger results) in a series of lemmas.

Lemma 8. $\mathscr{R}_1(ac) - \mathscr{M}_k(ac) \neq \emptyset$ for all $k \geqq 1$.

Proof. The language

$$L_k = \{(a^{2k}b^{2k})^n \mid n \geqq 1\} \cup \{b^n \mid n \geqq 2\}$$

is generated by the random context grammar of degree 1

$$G = (\{a, b\}, \{a^{2k}b^{2k}, b^2\}, \{(ab \to ab^{2k}a^{2k}b, \emptyset, \emptyset), (bb \to bbb, \emptyset, \{a\})\})$$

but $L_k \notin \mathscr{M}_k(ac)$: each matrix $(r_1, \ldots, r_t)$ which is used in a derivation $b^i \Rightarrow^* b^j$ and introduces at least a new symbol b can be also used for rewriting a string of the form $(a^{2k}b^{2k})^{t+1}$ into a string containing at least one substring b^{2k} and at least one substring b^s, $s \neq 2k$. $\square$

Corollary. $\mathscr{R}_k \subset \mathscr{R}_k(ac)$, $k \geqq 1$, proper inclusion.

Lemma 9. $\mathscr{R}_1 - \mathscr{P}_k(ac) \neq \emptyset$, $k \geqq 1$.

Proof. We consider again the language L_k in the proof of Lemma 2. Following the same argument and the fact that the derivation in a programmed grammar may begin by any rule, we start a derivation from ab^kc^k using a rule which introduces a substring ef; a string not in L_k is obtained, hence $L_k \notin \mathscr{P}_k(ac)$. □

Lemma 10. $\mathscr{M}_1 - (\mathscr{P}_\infty(ac) \cup \mathscr{R}_\infty(ac)) \neq \emptyset$.

Proof. The language $\{a^nba^nca^n \mid n \geqq 1\}$ is in $\mathscr{M}_1$, but it is not in $\mathscr{P}_\infty(ac) \cup \mathscr{R}_\infty(ac)$ (the same arguments as in the proof of Lemmas 3, 4). □

Lamma 11. $\mathscr{P}_1 - \mathscr{M}_k(ac) \cup \mathscr{R}_\infty(ac)) \neq \emptyset$, $k \geqq 1$.

Proof. The language

$$L_k = \{a^{n+1}b^{n+k} \mid n \geqq 1\} \cup \{a^nb^k \mid n \geqq 1\}$$

is in $\mathscr{P}_1$ (it is generated by $(\{a, b\}, \{ab^k\}, \{(1\colon ab \to a^2b^2, \{1\}), (2\colon ab \to a^2b, \{2\})\})$ but it is not in $\mathscr{R}_\infty(ac)$ (the random context restriction is of no use, as each string contains all symbols), nor in $\mathscr{M}_k(ac)$ (each matrix used for rewriting $a^{m+1}b^{m+k}$ can be used also for rewriting a^nb^k, thus obtaining parasitic strings. □

Corollary. $\mathscr{M}_k(ac) \subset \mathscr{C}_k(ac)$, $k \geqq 1$, strict inclusion.

Proof. The inclusion is obtained in the standard way and the above lemma proves that it is proper. □

Lemma 12. $(\mathscr{M}_1(ac) \cap \mathscr{R}_1(ac)) - \mathscr{C}_k \neq \emptyset$, $k \geqq 1$.

Proof. The language

$$L_k = \{a^{kn}b^{kn}c^{2k} \mid n \geqq 2\} \cup \{b^{kn}c^{2k+1} \mid n \geqq 2\}$$

can be generated by the matrix grammar

$$G = (\{a, b, c\}, \{a^{2k}b^{2k}c^{2k}, b^{2k}c^{2k+1}\},$$
$$\{(aa \to aa^ka, bb \to bb^kb)\}, \{aa \to aa^ka\})$$

as well as by the random context grammar

$$G' = (\{a, b, c\}, \{a^{2k}b^{2k}c^{2k}, b^{2k}c^{2k+1}\},$$
$$\{(ab \to aa^kb^kb, \emptyset, \emptyset), (bc \to bb^kc, \emptyset, \{a\})\}$$

The proof that $L_k \notin \mathscr{C}_k$ is similar to the proof of Lemma 7. □

Corollary. $\mathscr{M}_k \subset \mathscr{M}_k(ac)$, $\mathscr{C}_k \subset \mathscr{C}_k(ac)$, $k \geqq 1$, strict inclusions.

Lemma 13. $\mathscr{P}_1(ac) - \mathscr{C}_k \neq \emptyset$ for all $k \geqq 1$.

Proof. Consider the language

$$L_k = \{ab^kc^k, ab^{2k}c^{2k}\} \cup \{b^{nk}c^{nk} \mid n \geqq 1\}$$

generated by the programmed grammar

$$G = (\{a, b, c\}, \{ab^kc^k, b^kc^k\}, \{(1\colon bc \to bb^kc^kc, \{2\}, \emptyset),$$
$$(2\colon ab \to ab, \emptyset, \{1\})\})$$

Suppose $L_k = L(G')$ for a regular control grammar $G' = (V, B, P, C)$ without appearance checking. The strings ab^kc^k, b^kc^k must be in B. Each derivation $b^{rk}c^{rk} \Rightarrow^* \Rightarrow^* b^{nk}c^{nk}$, $b^{rk}c^{rk} \in B$, $n > r$, must use rules of the form $xy \to xzy$, $|x| \leqq k$, $|y| \leqq k$, and all of them are effectively used, hence these rules can be applied also to ab^kc^k, in the same order, thus obtaining strings not in L_k, contradiction. □

Corollary. $\mathscr{P}_k \subset \mathscr{P}_k(ac)$ is a strict inclusion for all $k \geqq 1$.

Lemma 14. All inclusions $\mathscr{R}_k(ac) \subset \mathscr{R}_{k+1}(ac)$, $\mathscr{M}_k(ac) \subset \mathscr{M}_{k+1}(ac)$, $\mathscr{P}_k(ac) \subset \subset \mathscr{P}_{k+1}(ac)$, $k \geqq 1$, are proper.

Proof. The language L_k in the proof of Lemma 7 is not in $\mathscr{P}_k(ac)$, the language L_k in the proof of Lemma 8 is not in $\mathscr{M}_k(ac)$, whereas the language $\{ba^{2k+1}\} \cup \cup \{a^{2nk+1}b \mid n \geqq s\}$ is not in $\mathscr{R}_k(ac)$. All these languages are in $\mathscr{S}_{k+1}$, hence the relations in Lemma follow. □

Open problems. Which are the relations between $\mathscr{P}_k(ac)$, $\mathscr{R}_k(ac)$ and $\mathscr{C}_k(ac)$? (We know only that $\mathscr{C}_k(ac) - (\mathscr{P}_k(ac) \cup \mathscr{R}_k(ac)) \neq \emptyset$.) Are the inclusions $\mathscr{C}_k(ac) \subset \subset \mathscr{C}_{k+1}(ac)$ proper?

The last problem may seem surprising, but note that the language L_k in the proof of Lemma 7 can be generated by a regular control grammar of degree 2: $(\{a, b\}, \{ba^{2k+1}, a^{2k+1}\}, \{(r_1: baa \to baba), (r_2: aa \to aa^{2k}a)\}, \{r_1^{2k}r_2^*r_2\}, \{r_1\})$ (each derivation starts by applying $2k$ times the rule r_1, in the appearance checking mode, and ends by using at least one time the rule r_2; if we start from ba^{2k+1}, after $2k$ applications of r_1, the rule r_2 cannot be applied, therefore the derivation is blocked).

Consider now the relation with families in Chomsky hierarchy. In [4] it is proved that $\mathscr{S}_1 \subset \mathscr{L}_2$; a stronger result is true.

Theorem 3. $\mathscr{R}_1(ac) \subset \mathscr{L}_2$, strict inclusion.

Proof. The language L_k in the proof of Lemma 11 is linear, hence it is enough to prove the inclusion. For, consider a random context grammar $G = (V, B, P)$, of degree 1. Define the sets

$$B' = \{a_1(a_1, a_2)(a_2, a_3) \dots (a_{k-1}, a_k) \mid a_1a_2 \dots a_k \in B, a_i \in V, 1 \leqq i \leqq k\},$$
$$P' = \{(a, b) \to (a, a_1)(a_1, a_2) \dots (a_{k-1}, a_k)(a_k, b) \mid (ab \to aa_1a_2 \dots a_kb, E, F) \in P, a_i \in V, 1 \leqq i \leqq k\}$$

For a string $x \in (V \cup V \times V)^*$, denote by $alph(x)$ the set of symbols in V appearing in x (possibly in couples of $V \times V$). For a subset $T \subseteq P'$, take the pure grammars $G_T(a, b) = (V \times V, (a, b), T)$, $(a, b) \in V \times V$, and define the substitution s_T: $(V \times V)^* \to 2^{(V \times V)^*}$, $s_T((a, b)) = L((G_T(a, b))$, $(a, b) \in V \times V$ (the languages $L(G_T(a, b))$ are context-free). Moreover, for a rewriting rule r': $(a, b) \to z$ in P' associated to r: $(ab \to awb, E, F)$ in P, we consider the gsm g_r, which replaces one occurrence of (a, b) by z, leaves unchanged the other symbols and checks whether all the symbols in E are present, but no symbol in F appears in the processed string (g_r simulates the application of the rule r).

Consider now the subsets $B_1^{(0)}, B_2^{(0)}, \ldots, B_{n_0}^{(0)}$ of B' such that:

(i) $B' = \bigcup_{i=1}^{n_0} B_i^{(0)}$

(ii) $B_i^{(0)} \cap B_j^{(0)} = \emptyset$ for $i \neq j, \quad 1 \leqq i, \quad j \leqq n_0$,

(iii) $alph(x) = alph(y)$ for $x, y \in B_i^{(0)}, 1 \leqq i \leqq n_0$,

(iv) if $x \in B_i^{(0)}, y \in B_j^{(0)}, i \neq j, 1 \leqq i, j \leqq n_0$, then $alph(x) \neq alph(y)$.

For a set C of strings, denote $alph(C) = \bigcup_{x \in C} alph(x)$.

Starting from a family $B_1^{(i)}, \ldots, B_{n_i}^{(i)}$ of sets of strings in $(V \cup V \times V)^*$ (initially we have $i = 0$ and the sets $B_j^{(0)}$, $1 \leqq j \leqq n_0$, constructed as above), we consider the following *procedure*:

For each $B_j^{(i)}$, $1 \leqq j \leqq n_i$, define

$$T_j^{(i)} = \{r' \in P' \mid r'\colon (a, b) \to z \text{ is obtained from } r\colon (ab \to awb, E, F) \text{ in } P \text{ and } alph(z) \subseteq (B_j^{(i)}),\ E \subseteq alph(B_j^{(i)}),\ F \cap alph(B_j^{(i)}) = \emptyset\}$$

and, for each rule $r'\colon (a, b) \to z$ in P' such that $alph(z) - alph(B_j^{(i)}) \neq \emptyset$, consider the set $g_r(s_{T_j^{(i)}}(B_j^{(i)}))$. The family of these sets is finite (the set P' is finite and the family of sets $B_j^{(i)}$ is finite); denote these sets, in a given order, by $B_1^{(i+1)}, \ldots, B_{n_{i+1}}^{(i+1)}$.

Some remarks about this procedure are worth mentioning:

1. $alph(B_j^{(i)}) = alph(s_{T_j^{(i)}}(B_j^{(i)})) \subset alph(g_r(s_{T_j^{(i)}}(B_j^{(i)}))) \subseteq V$, for all $r \in P$ as above.
2. As a consequence of the above point, we find that the procedure stops after a finite number of steps, as no further rules $r'\colon (a, b) \to z$ can be found such that $alph(z) - alph(B_j^{(i)}) \neq \emptyset$ for some $B_j^{(i)}$. Let t be this moment of procedure halting.
3. All sets $B_j^{(i)}$ are context-free languages: we start from finite language $B_j^{(0)}$ and use finitely many context-free substitutions and *gsm* mappings (both preserve the context-free-ness).
4. Each substitution $s_{T_j^{(i)}}$ corresponds to a derivation in G which does not introduce new symbols in the current string; moreover the $T_j^{(i)}$ definition ensures the fact that the random context restrictions are observed. Similarly, each use of a *gsm* corresponds to using the random context rule r for one step rewriting (this rule enlarges the set $alph(x)$ of the current string).

Consequently, the language $C = \bigcup_{\substack{0 \leqq i \leqq t \\ 1 \leqq j \leqq n_i}} B_j^{(i)}$ is context-free and $h(C) = L(G)$, where $h\colon (V \cup V \times V)^* \to V^*$ is the homomorphism defined by $h(a) = a, a \in V, h((a, b)) = b, (a, b) \in V \times V$. In conclusion, $L(G)$ is a context-free language and the proof is terminated. $\square$

(Received April 4, 1989.)

REFERENCES

[1] G. Ciucar: On the syntactic complexity of Galiukschov semicontextual languages. Rev. Roumaine Lingv. CLTA *25* (1988), 1, 23—28.

[2] J. Dassow: Pure grammars with regulated rewriting. Rev. Roumaine Math. Pures Appl. *31* (1986), 8, 657—666.

[3] J. Dassow and Gh. Păun: Regulated Rewriting in Formal Language Theory. Akademie Verlag, Berlin 1989.

[4] B. S. Galiukschov: Semicontextual grammars (in Russian). Mat. logica i mat. lingv. Kalinin Univ. 1981, 38—50.

[5] S. Marcus: Contextual grammars. Rev. Roumaine Math. Pures Appl. *14* (1969), 10, 1525 to 1534.

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

[7] Gh. Păun: Two theorems about Galiukschov semicontextual languages. Kybernetika *21* (1985), 5, 360—365.

[8] C. C. Squier: Semicontextual grammars: an example. Bull. Math. Sci. Math. R. S. Roumanie (N. S.) *32* (80) (1988), 2, 167—170.

Monica Marcus, Dr. Gheorghe Păun, The Computer Centre of the University of Bucharest, Str. Academiei 14, Bucureşti 70109. Roumania.