KYBERNETIKA — VOLUME 21 (1985), NUMBER 3

FURTHER REMARKS ON THE COMPLEXITY OF REGULATED REWRITING*

JÜRGEN DASSOW, GHEORGHE PĂUN

The paper continues the investigations of [1], [8] about the number of variables necessary/sufficient in order to generate certain languages by various grammars with restrictions in derivation. Here, we deal with matrix, programmed, and random context grammars (with appearance checking and erasing rules).

1. INTRODUCTION

The succinctness of descriptions of a language by some given devices themselves is a very important problem, and many researchers have devoted to this subject (especially in the context free case, see [3], [4], [9]). In particular, the natural parameter of the cardinality of the nonterminal alphabet is of central interest. For context-free grammars this parameter leads to an infinite hierarchy of languages (even if we take into account only regular languages, [3]). For regulated rewriting the situation is quite different: 6 nonterminals are sufficient in order to generate recursively enumerable language by a matrix grammar (with appearance checking and erasing rules), [8]. Further it has been shown that two nonterminals are necessary (sufficient) for the generation of some (all) linear languages.

The aim of the present paper is to say more on families of languages which can be generated by matrix grammars with 1, 2, or 3 nonterminals and to obtain similar results for programmed grammars (which are related — from the complexity point of view — in a very precise way to matrix grammars, [1]).

We prove that three nonterminals are necessary (sufficient) in order to generate some (all) metalinear languages by a matrix grammar, that three nonterminals are sufficient for the generation of non-semilinear languages, and that one nonterminal

* Research done during a stay of the second author at the Technological University Magdeburg.

is enough for the generation of context-free languages of arbitrary complexity and of non-context free languages but not for generating non-semilinear languages.

Similar results are proved for programmed grammars. By these results it follows that there is no function bounding the increase of the number of necessary non-terminals when going from matrix (programmed) grammars to context-free grammars (see also [1]). In this paper we show that also for random context grammars no such function exists.

2. NOTATIONS AND TERMINOLOGY

The reader is assumed to be familiar with basic notions and results of formal language theory (see e.g. [2], [10]). We specify here only some notations and recall informally some definitions.

Let $V = \{x_1, x_2, \ldots, x_n\}$ be an alphabet. V^* denotes the set of all strings (words) over V (including the empty string λ). By $|w|$ and $|w|_x$ we denote the length of the string $w \in V^*$ and the number of occurrences of the letter $x \in V$ in $w \in V^*$, respectively. The Parikh vector $p_V(w)$ is defined by

$$p_V(w) = (|w|_{x_1}, |w|_{x_2}, \ldots, |w|_{x_n}) .$$

For a language $L \subseteq V^*$ we put

$$p_V(L) = \{p_V(w) : w \in L\} .$$

A language $L \subseteq V^*$ is called *semilinear* if there are n-dimensional row-vectors $p_i, q_{i1}, q_{i2}, \ldots, q_{ir_i}$, $1 \leqq i \leqq m$, such that

$$p_V(L) = \bigcup_{i=1}^{n} \{p_i + \alpha_1 q_{i1} + \alpha_2 q_{i2} + \ldots + \alpha_{r_i} q_{ir_i} : \alpha_i \in \mathbb{N}\} .$$

In all following grammars, V_N is the set of nonterminals, V_T denotes the set of terminals, and S is the axiom, $S \in V_N$.

A context-free grammar $G = (V_N, V_T, P, S)$ is called *linear* iff all productions of P have the form

(1) $$A \to uBv \quad \text{or} \quad A \to u \quad \text{where} \quad u, v \in V_T^* .$$

It is called *metalinear* iff the following three conditions are satisfied:

i) for all productions $A \to w$ in P, S does not occur in w,

ii) for $A \in V_N$, $A \neq S$ all productions are of the form (1),

iii) all rules with left-hand side S are of the form

$$S \to w_1 A_1 w_2 A_2 \ldots w_s A_s w_{s+1} \quad \text{where} \quad w_j \in V_T^* \quad \text{and} \quad s \geqq 0 .$$

We now recall informally the definitions of three types of regulated rewritting. For detailed information on matrix and programmed grammars see [7] and [10],

Chapter 5; concerning random context grammars we refer to [6] and [11]. A *matrix grammar* is a quadruple $G = (V_N, V_T, M, S, F)$ where

$$M = \{m_1, m_2, \ldots, m_n\}$$

is a finite set of sequences of productions,

$$m_i = [A_{i_1} \to w_{i_1}, A_{i_2} \to w_{i_2}, \ldots, A_{i_{s(i)}} \to w_{i_{s(i)}}] \quad \text{for} \quad i = 1, 2, \ldots, n\,,$$

and F is a subset of the set of all productions occurring in M. The application of a matrix m_i to w is defined as follows: there are words $w = w_1, w_2, \ldots, w_{s(i)}$ such that, for $1 \leqq j < s(i)$

i) $w_j = w_{j+1}$ if A_{i_j} does not occur in w_j and $A_{i_j} \to w_{i_j} \in F$

or

ii) $w_j = v_j A_{i_j} u_j$ and $w_{j+1} = v_j w_{i_j} u_j$.

Only those words are in generated language $L(G)$ which are obtained by applications of whole matrices.

The productions of a *random context grammar* $G = (V_N, V_T, P, S)$ are of the form

$$(A \to w, R, Q)$$

where R and Q are subsets of V_N. Such a production is only applicable to $x = x_1 A x_2$ if $x_1 x_2$ contains no letter of R and $x_1 x_2$ contains all letters of Q.

The productions of a *programmed grammar* $G = (V_N, V_T, P, S)$ are of the form

$$(l, A \to w, E, F)$$

where l is the label of the production, F and E are sets of labels. If $A \to w$ is applicable to x, then the next production has to be a rule with a label in the success field E. If $A \to w$ is not applicable, then the next production has to have a label contained in the failure field F.

Please note that throughout this paper we consider only grammars with context free core productions and that we allow erasing rules $A \to \lambda$.

By *LIN*, *MLIN*, *CF*, *M*, *PR*, *RC*, *CS*, *RE* we denote the classes of linear, metalinear, context free, matrix, programmed, random context, context sensitive, and type – 0 Chomsky grammars, respectively. For a class X of grammars, let $\mathscr{L}(X)$ be the family of languages $L(G)$ generated by grammars G in X. (*RE* stands for "recursively enumerable".) It is known (see [7], [10], [6]) that

$$\mathscr{L}(M) = \mathscr{L}(PR) = \mathscr{L}(RC) = \mathscr{L}(RE)\,,$$

i.e. the generative capacities of all these types of grammars coincide. In order to illustrate differences between these regulation mechanisms the descriptional complexity of regulated rewriting has been studied in [1]. Here we continue this research. We consider the following complexity measure.

For a grammar $G = (V_N, V_T, S, P)$ $(G = (V_N, V_T, S, M, F))$, we put

$$\mathrm{Var}(G) = \mathrm{card}(V_N)$$

and, for a language $L \in \mathscr{L}(X)$, we define

$$\mathrm{Var}_X(L) = \inf\{\mathrm{Var}(G) : G \in X,\ L(G) = L\}\,.$$

3. ON MATRIX GRAMMARS WITH ONE, TWO, OR THREE NONTERMINALS

In [8] it is proved that.

1) $\mathrm{Var}_M(L) \leqq 6$ for all $L \in \mathscr{L}(RE)$,
2) $\mathrm{Var}_M(L) \leqq 2$ for all $L \in \mathscr{L}(LIN)$ and there is a regular language L' such that $\mathrm{Var}_M(L') = 2$.

It is an open problem whether the bound 6 in 1) is optimal. Further one is interested in uniform estimations similar to 2) for some other subfamilies of $\mathscr{L}(CF)$ $(\mathscr{L}(M))$. In this chapter we contribute to the theory of matrix grammars with at most three nonterminals.

First we consider matrix grammars with only one nonterminal symbol.

Theorem 1. i) Every language L with $\mathrm{Var}_M(L) = 1$ is semilinear ii) There are non-context-free languages L such that $\mathrm{Var}_M(L) = 1$.

Proof. i) $G = (V_N, V_T, S, M, F)$ be a matrix grammar with $V_N = \{S\}$. Since each nonterminal sentential form generated by G contains an occurrence of S it follows that, possibly excepting the last rules of the last matrix used in some derivations, all rules are effectively used.

Let $|x|_a$ be the number of occurrences of the letter a in the word x. For a matrix $m = (S \to x_1, S \to x_2, \ldots, S \to x_t) \in M$ we define

$$E(m) = |x_1 x_2 \ldots x_t|_S\,,$$

and we put

$$E(M) = \max\{E(m) : m \in M\}\,.$$

Moreover, let

$$p = \max\{t : (S \to x_1, S \to x_2, \ldots, S \to x_t) \in M\}\,,$$

$$q = 2p + E(M)\,.$$

Clearly, any matrix of M can be applied to each string which contains at least p occurrence of S, and each matrix in M replaces at most p occurrences of S by terminal strings. Therefore, given a derivation D with respect to G producing a string w, we can apply the matrices in D in such an order that we obtain a derivation D'

which produces a string w' letter-equivalent to w, and all sentential forms occurring in the derivation D' have at most q occurrences of S. We take some initial steps of D, and if we obtain more than $2p$ occurrences of S then we use one or more matrices of D which decrease the number of occurrences of S in order to obtain a sentential form with less than $2p$ occurrences of S, we continue by a matrix increasing the number of S-occurrences; the only restriction is to use the last matrix of D also as the last matrix of D'.

Consequently a sublanguage $L \subseteq L(G)$ exists which is letter-equivalent to $L(G)$ and has only derivations with a bounded number of nonterminals in the sentential forms (of "finite index"). Now we can continue as in the proof of [7], Lemma 3.13 in order to show that $L(G)$ is letter-equivalent to a regular language, and hence $L(G)$ is semilinear by [10], Theorem 7.1.

ii) We consider the matrix grammar

$$G = (\{S\}, \{a, b, c, d\}, S, M, \emptyset)$$

with

$$M = \{(S \to SS), (S \to aSb, S \to cSd), (S \to ab, S \to cd)\} .$$

Clearly

$$L(G) \cap a^*b^*c^*d^* = \{a^n b^n c^n d^n : n \geqq 1\} .$$

Since $\{a^n b^n c^n d^n : n \geqq 1\}$ is not context-free, $L(G)$ is also not context-free.

Corollary. Each non-regular $L \subseteq \{a\}^*$ satisfies $\mathrm{Var}_M(L) \geqq 2$, and $\mathrm{Var}_M(L) \geqq 2$ holds for each non-context-free language $L \subseteq \{a\}^* \{b\}^*$.

Indeed, the semilinear languages in a^* are regular, and the semilinear languages in a^*b^* are context-free, [2].

The one-nonterminal matrix grammars can generate context-free languages of arbitrary context-free complexity (with respect to the measure Var). More precisely, we have

Theorem 2. For each $n \geqq 1$, there are context-free languages L_n such that $\mathrm{Var}_{CF}(L_n) > n$ and $\mathrm{Var}_M(L_n) = 1$.

Proof. Let $n \geqq 2$. We consider the matrix grammar

$$G_n = (\{S\}, \{a, b, a_1, a_2, \ldots, a_n\}, S, M, \emptyset)$$

with

$$M = \{m_0, m_0'\} \cup \{m_i : 1 \leqq i \leqq n\} ,$$

$$m_0 = (S \to SS) , \quad m_0' = (S \to aSb) ,$$

$$m_i = (\underbrace{S \to a_i, S \to a_i, \ldots, S \to a_i}_{(i+1)\text{-times}}) , \quad 1 \leqq i \leqq n .$$

The language $L_n = L(G_n)$ is context-free. Indeed, it consists of strings of the

DYCK language over $\{a, b\}$ shuffled in any possible way with strings in the language

$$L_0 = \{w : w \in \{a_1, a_2, \ldots, a_n\}^*,\ |w|_{a_i} = k_i(i+1) \quad \text{for some} \quad k_i \geqq 0,\ 1 \leqq i \leqq n\}\,.$$

More specifically, a derivation D in G_n proceeds as follows: after using once or several times the matrices m_0, m_0' (thus producing a sentential form of the Dyck language) we replace $i+1$ occurrences of S by a_i using a matrix m_i for some i, $1 \leqq i \leqq n$; then again we use the matrices m_0, m_0' and so on. Therefore each such derivation can be rearranged in such a way that we use first all rules $S \to SS$, $S \to aSb$ from D keeping unchanged all symbols S which are replaced by terminals at earlier steps of D, and then we use only matrices m_i, $1 \leqq i \leqq n$. Since L_0 is regular and $\mathscr{L}(CF)$ is closed under shuffles with regular sets, L_n is context-free (see [7], p. 90).

Now let us consider a context-free grammar $G = (V_N, V_T, S, P)$ for the language L_n. We regard all the strings of L_n of the forms

$$(*) \qquad a_i^i a^m a_i b^m\,, \quad m \geqq 1\,,$$

$$(**) \qquad a_j^i a^m a_j b^m\,, \quad m \geqq 1\,,$$

for some $i \neq j$, $1 \leqq i, j \leqq n$. In order to generate strings of the form $(*)$ we need derivations

$$S \Rightarrow^* a_i^i a^r A_i b^s \Rightarrow^* a_i^i a^r a^k A_i a^k b^s \Rightarrow^* a_i^i a^m a_i b^m$$

with $r, s \geqq 0$, $k > 0$, and similarly for the strings of the form $(**)$ involving a nonterminal A_j in a self-embedding derivation $A_j \Rightarrow^* a^t A_j b^t$, $t > 0$. If $A_i = A_j$ then the derivation

$$S \Rightarrow^* a_i^i a^r A_i b^s = a_i^i a^r A_j b^s \Rightarrow^* a_i^i a^r a^u a_j b^v b^s$$

is possible and produces a string not in L_n.

Moreover, $S \neq A_i$ because S can produce a string a_j^{i+1} which cannot be replaced for A_i in $a_i^i a^r A_i b^s$ without contradicting the structure of L_n. Hence at least $n+1$ nonterminals are necessary for the generation of L_n. Thus $\mathrm{Var}_{CF}(L_n) \geqq n+1$.

Remark 1. In [3] it is proved that each language B_m, $m \geqq 1$, defined by

$$B_1 = \{a\}\,,$$

$$B_2 = \{a\}^* \cup \{b\}\,,$$

$$B_m = \bigcup_{i=1}^{m-1} \{a_i\}^* \quad \text{for} \quad m \geqq 3\,,$$

satisfies $\mathrm{Var}_{CF}(B_m) = m$. Since these languages are regular, $\mathrm{Var}_M(B_m) \leqq 2$ for each m. Hence we found regular languages L_n with $\mathrm{Var}_M(L_n) \leqq 2$ and $\mathrm{Var}_{CF}(L_n) = n$. Note that in Theorem 2 we have $\mathrm{Var}_{CF}(L_n) > n$; we do not know languages L_n for which $\mathrm{Var}_M(L_n) = 1$ and $\mathrm{Var}_{CF}(L_n) = n$ holds (this would say that Var_{CF} is a connected complexity measure – in the sense of [4] – on the family $\{L : L \in \mathscr{L}(CF)$ $\mathrm{Var}_M(L) = 1\}$).

Now let us jump to matrix grammars with three nonterminals. First we give a counterpart of Theorem 1.

Theorem 3. There are non-semilinear languages L with $\mathrm{Var}_M(L) = 3$.

Proof. We consider the matrix grammar

$$G = (\{S, A, B\}, \{a\}, S, M, F)$$

with

$$\begin{aligned}
F &= \{S \to S^4,, A \to S^4, B \to S^4\} \\
M &= \{A \to S^4,\ B \to S^4,\ S \to SA)\,, \\
&\quad (S \to \lambda,\ S \to S^4,\ A \to SBB)\,, \\
&\quad (S \to \lambda,\ S \to S^4,\ A \to S^4,\ B \to SSA)\,, \\
&\quad (S \to \lambda,\ S \to \lambda,\ S \to S^4,\ B \to SSA)\,, \\
&\quad (S \to \lambda,\ S \to \lambda,\ S \to S^4,\ B \to S^4,\ A \to SSB)\,, \\
&\quad (S \to \lambda,\ S \to \lambda,\ S \to S^4,\ B \to S^4,\ A \to SSSa)\,, \\
&\quad (S \to \lambda,\ S \to \lambda,\ S \to \lambda,\ S \to S^4,\ A \to SSSa)\,, \\
&\quad (S \to \lambda,\ S \to \lambda,\ S \to \lambda,\ S \to S^4,\ A \to a)\}\,.
\end{aligned}$$

If there is exactly one symbol S in the current string, then each occurrence of A is replaced by two occurrences of the letter B; in the presence of exactly two symbols S each B is transformed in A (and these operations can be iterated); and if exactly three symbols S occur, then the derivation enters the final stage and terminates; if (at least) four occurrences of S are produced, then the derivation cannot terminate. Consequently,

$$L(G) = \{a^{2^n} : n \geqq 1\}\,,$$

and this is not a semilinear language.

Remark 2. It is an open problem whether the matrix grammars with two nonterminals can generate non-semilinear languages. However, such grammars can generate languages, which are not simple matrix languages in the sense of [5] which form a strong extension of contex-free languages. An example for such a language is generated by the grammar

$G = (\{S, A\}, \{a, b, c, d\}, S, M, \{A \to d\})$ with

$M = \{(A \to d, S \to AS), (A \to aAb, S \to cS), (A \to ab, S \to cASS, (A \to ab, S \to c)\}$.

$L(G)$ is not a simple matrix language because the family of simple matrix languages is closed under intersections with regular sets and $L(G) \cap \{a, b, c\}^* = \{a^n b^n c^n : n \geqq 1$ is closed under intersections with regular sets and $L(G) \cap \{a, b, c\}^* = \{a^n b^n c^n : n \geqq 1\}^*$ is not a simple matrix language (see [7]).

Now we give counterparts of the result quoted in Section 3 under the number 2) for metalinear languages.

Theorem 4. For each metalinear language L,

$$\mathrm{Var}_M(L) \leqq 3 .$$

Proof. Let $G = (V_N, V_T, S, P)$ be a metalinear grammar containing the nonlinear rules $r_1, r_2, \ldots, r_p$:

$$r_i : S \to w_{i,1}A_{i,1}w_{i,2}A_{i,2} \ldots w_{i,k_i}A_{i,k_i}w_{i,k_i+1} ,$$

$1 \leqq i \leqq p$, $w_{i,j} \in V_T^*$, $A_{i,j} \in V_N \setminus \{S\}$ for $1 \leqq j \leqq k_i$. We construct a matrix grammar $G' = (\{S, A, B\}, V_T, S, M, F)$ in the following way. For every tripel (i, j, X), $1 \leqq \leqq i \leqq p$, $1 \leqq j \leqq k_i$, and $X \in V_N$, let $q(i, j, X)$ be a positive natural number such that $q(i, j, X) \neq q(k, t, Y)$ for $(i, j, X) \neq (k, t, Y)$. Further let

$$Q = 1 + \max \{q(i, j, X) : 1 \leqq i \leqq p, 1 \leqq j \leqq k_i, X \in V_N\} .$$

We put

$$F = \{A \to A^Q\}$$

and introduce in M the following matrices:

1) $(S \to w)$ for $S \to w \in P$, $w \in V_T^*$,

2) $(A \to A^Q, S \to w_{i,1}A^{q(i,1,A_{i,1})}BS)$, $1 \leqq i \leqq p$,

3) $([A \to \lambda]^{q(i,j,X)}, A \to A^Q, B \to uA^{q(i,j,Y)}Bv)$ for $1 \leqq i \leqq p$, $1 \leqq j \leqq k_i$, $X \to uYv \in P$, $X, Y \in V_N$, $u, v \in V_T^*$

($[A \to \lambda]^t$ is a short writing for t occurrences of the rule $A \to \lambda$),

4) $([A \to \lambda]^{a(i,j,X)}, A \to A^Q, B \to z, S \to w_{i,j+1}A^{q(i,j+1,A_{i,j+1})}BS)$ for $1 \leqq i \leqq p$, $1 \leqq j < k_i$, $X \to z \in P$, $X \in V_N$, $z \in V_T^*$,

5) $([A \to \lambda]^{q(i,k_i,X)}, A \to A^Q, B \to z, S \to w_{i,k_i+1})$ for $1 \leqq i \leqq p$, $X \to z \in P$, $X \in V_N$, $z \in V_T^*$.

Each derivation in G starts with a rule r_i, $1 \leqq i \leqq p$, and then we can apply only linear rules in order to obtain the derivation $A_{i,j} \Rightarrow^* z_{i,j}$. Therefore without loss of generality we have only the following terminating derivations in G:

$$\begin{aligned}
S &\Rightarrow^* w_{i,1}A_{i,1}w_{i,2}A_{i,2} \ldots w_{i,k_i}A_{i,k_i}w_{i,k_i+1} \Rightarrow \\
&\Rightarrow w_{i,1}u_{i,1}B_{i,1}v_{i,1}w_{i,2}A_{i,2} \ldots w_{i,k_i}A_{i,k_i}w_{i,k_i+1} \Rightarrow^* \\
&\Rightarrow^* w_{i,1}z_{i,1}w_{i,2}A_{i,2} \ldots w_{i,k_i}A_{i,k_i}w_{i,k_i+1} \Rightarrow^* \\
&\Rightarrow^* w_{i,1}z_{i,1}w_{i,2}z_{i,2}w_{i,3}A_{i,3} \ldots w_{i,k_i}A_{i,k_i}w_{i,k_i+1} \Rightarrow^* \\
&\quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
&\Rightarrow^* w_{i,1}z_{i,1}w_{i,2}z_{i,2} \ldots w_{i,k_i}z_{i,k_i}w_{i,k_i+1} .
\end{aligned}$$

We now prove that the only possible derivations in G' are the same ones, and thus $L(G) = L(G')$.

Since the rules $A \to \lambda$ have to be effectively used a derivation in G' has to start with a matrix of type (1) or type (2). In the former case a derivation in G is simulated

directly. In the latter case we have obtained

$$S \Rightarrow w_{i,1} A^{q(i,1,A_{i,1})} BS = v_1 .$$

We cannot apply a matrix of type (2) since it introduces at least Q occurrences of A, all matrices do not decrease the number of occurrences of A in the sentential forms and hence no termination is possible.

Since the matrices of type (3) do not change the number of B's and the application of (4) or (5) requires the occurrence of S in the sentential form, (1) is also not applicable to v_1.

Further the application of a matrix of type (3) or (4) or (5) is only possible if it starts with $([A \to \lambda]^{q(i,1,A_{i,1})}, A \to A^Q, \ldots)$, otherwise we cannot effectively use all productions $A \to \lambda$ or we can effective apply $A \to A^Q$.

Hence we obtain

$$S \Rightarrow v_1 \Rightarrow w_{i,1} u_{i,1} A^{q(i,1,B_{i,1})} B v_{i,1} S = v_2 .$$

by application of a rule of type (3). Now we can iterate this process and obtain

$$S \Rightarrow v_1 \Rightarrow v_2 \Rightarrow^* w_{i,1} z_{i,1} w_{i,2} A^{q(i,2,A_{i,2})} BS = v_3$$

by application of a matrix of type (4). Again we iterate and get

$$S \Rightarrow v_1 \Rightarrow v_2 \Rightarrow^* v_3 \Rightarrow^* w_{i,1} z_{i,1} w_{i,2} z_{i,2} \cdots w_{i,k_i} z_{i,k_i} w_{i,k_i+1}$$

terminating with a matrix of type (5). Thus we have simulated a derivation in G. Since $\mathrm{Var}(G') = 3$, we obtain $\mathrm{Var}_M(L(G)) \leqq 3$ and the theorem is proved. □

Theorem 5. There are metalinear languages L such that $\mathrm{Var}_M(L) = 3$.

Proof. Let us consider the language

$$L = \{a^n b^n c^m d^m e^p f^p : n, m, p \geqq 1\}$$

and suppose that there is a grammar $G = (\{S, A\}, \{a, b, c, d, e, f\}, S, M, F)$ which generates it.

Now we discuss all possible derivations in G with respect to the number of occurrences of A and/or S in the sentential forms. In order to do this we introduce the following notions. Let

$$D : S = u_0 \Rightarrow u_1 \Rightarrow u_2 \Rightarrow \ldots \Rightarrow u_n \in \{a, b, c, d, e, f\}^*$$

be a derivation in G such that u_{i+1} is obtained from u_i by application of a matrix of M. Then

$$u_i \Rightarrow u_{i+1} \Rightarrow \ldots \Rightarrow u_j , \quad 0 \leqq i < j \leqq n$$

is called a subderivation of D. It is called linear if each u_k, $i \leqq k \leqq j$, contains at most one nonterminal (i.e. the matrix acts like a linear rule (1) in a context-free grammar). If $u_i = u_i' B u_i''$ where u_i' and u_i'' are terminal words we obtain $u_j = u_i' u_j' u_i''$

in a linear subderivation, and then we call $B \Rightarrow^* u'_j$ also a linear subderivation of D.

Consider a linear derivation with respect to G. If there is a linear subderivation $S \Rightarrow^* xSy$, $A \Rightarrow^* xAy$, then we have to have $x = \alpha^i$, $y = \beta^j$ for some $\alpha, \beta \in \{a, b, c, d, e, f\rangle$. If $i > 0$ and $j > 0$, then (α, β) has to be a pair in $\{(a, b), (c, d), (e, f)\}$, because otherwise we contradict the relations between the number of occurrences of the letters in the strings of L. If $i = 0$, then $j = 0$; and if $j = 0$, then $i = 0$, too. Consequently, each starting linear subderivation of a derivation of a string in L has at most two steps. Thus, for an infinite part of L, we need nonlinear derivation steps.

If in some non-linear derivation

$$D : S \Rightarrow^* x_1 X x_2 Y x_3 \Rightarrow^* x'_1 w_2 x'_2 w_1 x'_3 = z ,$$

$z \in V_T^*$, x'_i being obtained from x_i, $i = 1, 2, 3$, we have $X = Y$ and w_1 contains a symbol γ, w_2 contains a letter δ such that $\gamma + \delta$, then we can produce the derivation

$$D' : S \Rightarrow^* x_1 X x_2 Y x_3 = x_1 Y x_2 X x_3 \Rightarrow^* x'_1 w_1 x'_2 w_2 x'_3 ,$$

and its terminal word contains "illegal" substrings.

Therefore, if the first non-linear step of a derivation contains two or more occurrences of both S and A, then the derivation can introduce an arbitrary number of at most two symbols (in the non-linear phase and only a bounded number of symbols in the starting linear phase). If the first non-linear step contains occurrences of only one symbol from $\{A, S\}$ (and then for at least two times), then the non-linear phase introduces an arbitrary number of at most one symbol (and the starting linear phase only a bounded number of symbols).

Thus, for infinitely many strings in L, we need derivations whose first non-linear step contains both S and A and one of them only for one time.

If such a derivation is of the form

$$S \Rightarrow^* w \Rightarrow^* x_1 S x_2 \Rightarrow^* z$$

where w contains both S and A, x_1, x_2, z are terminal strings, then we have to have $x_1 = x_2 = \lambda$, because otherwise the symbol S in $x_1 S x_2$ can be replaced by an arbitraty string $y = a^n b^n c^n d^n e^n f^n$ of L and $x_1 y x_2$ cannot be in L.

If such a derivation is of the form

$$S \Rightarrow^* w \Rightarrow^* x_1 A x_2 \Rightarrow^* z$$

with w, x_1, x_2, z as above, then the phase $x_1 A x_2 \Rightarrow^* z$ can contain a string $x_1 y_1 A y_2 x_2$ with $y_1 y_2 \neq \lambda$ only if $y_1 = \varepsilon^i$, $y_2 = \eta^j$, (ε, η) is a pair in $\{(a, b), (c, d), (e, f)\}$. Thus at most two symbols can be introduced for arbitrary many times by derivations involving the above step $A \Rightarrow^* y_1 A y_2$.

The above discussion implies that there are infinitely many strings in L such that in their derivations we can find subderivations $w_1 \Rightarrow^* w_2$ of arbitrary length such that w_1, w_2, and all intermediate steps contain occurrences of S and A, and it introduces an arbitrary number of occurrences of at least four symbols of $\{a, b, c, d, e, f\}$.

If all intermediate steps in such a derivation $w_1 \Rightarrow^* w_2$ contain exactly one S and one A, then each direct derivation in it can introduce only pairs of symbols (otherwise we avoid them and obtain incorrect strings). At most two such pairs p_1, p_2 can be introduced, and moreover, the above considered derivation $A \Rightarrow^* y_1 A y_2$ has to introduce also one of these pairs p_1, p_2. Hence there are infinitely many derivations $w_1' \Rightarrow^* w_2$ involving strings with more occurrence of one of S and A in w_1' (possibly ignore initial parts of derivation $w_1 \Rightarrow^* w_2$). The symbol occurring two or more times can introduce occurrences of at most one terminal. Thus the other symbol has to introduce at least three terminal letters. Assume that there are infinitely many derivations $w_1' \Rightarrow^* w_2$ as above such that A occurs one time in all its sentential forms (hence S occurs at least two times). The same arguments hold if there are infinitely many derivations with at least two A's.

If at some stage A is erased and a new A is introduced from an S, then this new occurrence of A cannot introduce any symbol different from that which is produced by A (if any). Therefore there are infinitely many derivations $z_1 \Rightarrow^* z_2$ such that

1) all steps contain one A and at least two S's,
2) each direct derivation uses only nonterminal linear rules for A (may be, two or more rules in the same matrix have the left-hand-side A; hence are applied at the same point).
3) the rules involving A introduce at least three different terminals,
4) all rules of all matrices in these derivations are used effectively (and not overpassed in the appearance checking sense, because all nonterminals symbols are present).

If the number of occurrence of S in these derivations is bounded, then, with exception of a finite number of derivations, the derivations have the form

$$z_1 \Rightarrow^* z_3 \Rightarrow^* z_4 \Rightarrow^* z_2 \quad \text{where} \quad |z_3|_S = |z_4|_S .$$

Then the subderivation $z_3 \Rightarrow^* z_4$ can be iterated, and hence A can introduce only a pair of terminal symbols. Therefore there are infinitely many derivations $z_1 \Rightarrow^* z_2$ as above involving sentential forms with an arbitrary number of occurrences of S.

We consider such a derivation. In it we have to use two matrices m_1, m_2 where m_1 rewrites A by some $\varrho^i A \sigma^j$ and m_2 replaces A by some $\varphi^k A \psi^r$ and either $i > 0$, $k > 0$, $\varrho \neq \psi$ or $j > 0$, $r > 0$, $\sigma \neq \psi$ (otherwise A cannot introduce three different terminals). Since all matrices are used effectively and since there is a sufficient number of occurrences of S the application of each of m_1, m_2 can be delayed up to the application of the other matrix. Interchanging the matrices we obtain incorrect pairs of symbols. This completes the proof. □

Remark 3. Note that it is essential that L is built by three pairs of six different letters since languages $\{a^n b^n c^m d^m : n, m \geqq 1\}$, $\{a^n b^n c^m d^m a^p b^p : n, m, p \geqq 1\}$ can be generated by matrix grammars with two nonterminals.

4. THE MEASURE Var FOR PROGRAMMED GRAMMARS

In [1] it has been proved that, for each $L \in \mathscr{L}(RE)$,

3) $\mathrm{Var}_M(L) \leqq \mathrm{Var}_{PR}(L) + 2$,

4) $\mathrm{Var}_{PR}(L) \leqq \mathrm{Var}_M(L) + 2$.

First we note that relation 3) can be improved to

3') $\mathrm{Var}_M(L) \leqq \mathrm{Var}_{PR}(L) + 1$

(taking the initial matrices $(C \to C^{p+1}, S \to C^r S)$, $1 \leqq r \leqq p$, instead of $(S' \to C^r S)$ in the proof of Lemma 6 in [1]. The rule $C \to C^{p+1}$ forbids the use of this new initial matrix in a later step of a derivation ($p + 1$ occurrences of C block the derivation). Thus the symbol S' becomes useless.)

By the results on the complexity of matrix grammars therefore we obtain

$$\mathrm{Var}_{PR}(L) \leqq 8 \quad \text{for all} \quad L \in \mathscr{L}(RE),$$
$$\mathrm{Var}_{PR}(L) \leqq 4 \quad \text{for all} \quad L \in \mathscr{L}(LIN),$$
$$\mathrm{Var}_{PR}(L) \leqq 5 \quad \text{for all} \quad L \in \mathscr{L}(MLIN).$$

In this chapter we improve the latter two relations.

Theorem 6. i) For any $L \in \mathscr{L}(LIN)$, $\mathrm{Var}_{PR}(L) \leqq 2$.

ii) For any $L \in \mathscr{L}(MLIN)$, $\mathrm{Var}_{PR}(L) \leqq 3$.

Proof. i) Let $G = (V_N, V_T, S, P)$ be a linear context-free grammar. We construct the programmed grammar $G' = (\{S, A\}, V_T, S, P')$ where

$$\begin{aligned} P' = {} & \{(*, S \to A, \{[S]\}, \emptyset\} \cup \\ & \cup \{([X], A \to uAv, \{[Y]\}, \emptyset) : X \to uYv \in P,\ u, v \in V_T^*,\ X, Y \in V_N\} \cup \\ & \cup \{([X], A \to z, \emptyset, \emptyset) : X \to z \in P,\ z \in V_T^*,\ X \in V_N\}. \end{aligned}$$

Clearly, the label $[X]$ of a rule identifies the left-hand member of the corresponding rules in P. Hence the labels control the derivation in the same way as the nonterminals in V_N. Consequently $L(G) = L(G')$ and $\mathrm{Var}_{PR}(L(G)) \leqq 2$.

ii) Now consider a metalinear grammar $G = (V_N, V_T, S, P)$ containing the nonlinear rules r_i, $1 \leqq i \leqq p$, as in the proof of Theorem 4. We construct the programmed grammar

$$\begin{aligned} G' = {} & (\{S, A, B\}, V_T, S, P') \quad \text{with} \quad B \notin V_N, \\ P' = {} & \{(*_w, S \to w, \emptyset, \emptyset) : S \to w \in P,\ w \in V_T^*\} \cup \\ & \cup \{(*_i, S \to w_{i,1}AB, \{[i, 1, A_{i,1}]\}, \emptyset) : 1 \subseteq i \leqq p\} \cup \\ & \cup \{([i, j, X], A \to uAv, \{[i, j, Y]\}, \emptyset) : 1 \subseteq i \leqq p,\ 1 \leqq j \leqq k_i, \\ & \quad X \to uYv \in P,\ u, v \in V_T^*,\ X, Y \in V_N\} \\ & \cup \{([i, j, X], A \to x, \{[i, j+1, B]\}, \emptyset) : 1 \leqq i \leqq p,\ 1 \leqq j < k_i, \end{aligned}$$

$$X \to x \in P,\ x \in V_T^*,\ X \in V_N\} \cup$$
$$\cup \{([i, j, B], B \to w_{i,j}AB, \{[i, j, A_{i,j}]\}, \emptyset) : 1 \leqq i \leqq p,\ 2 \leqq j \leqq k_i\} \cup$$
$$\cup \{([i, k_i + 1, B], B \to w_{i,k_i+1}, \emptyset, \emptyset) : 1 \leqq i \leqq p\} .$$

Again, the labels $[i, j, X]$ identify the non-linear rule, the nonterminal position in it and the nonterminal in this position. Hence $L(G) = L(G')$, $\mathrm{Var}_{PR}(L(G)) \leqq 3$.

Theorem 7. There exists a linear language L with $\mathrm{Var}_{PR}(L) = 2$.

Proof. We consider the language

$$L = \{ab^n c^n d : n \geqq 1\}$$

and suppose that $L = L(G)$ for a programmed grammar

$$G = (\{S\}, \{a, b, c, d\}, S, P) .$$

For any derivation

$$S \Rightarrow^{r_1} w_1 \Rightarrow^{r_2} w_2 \Rightarrow^{r_3} \ldots \Rightarrow^{r_m} w_m = ab^n c^n d ,$$

the rule r_m is used effectively (each w_i contains at least one S) and thus the derivation

$$S \Rightarrow^{r_m} w'$$

is correct. Hence r_m introduces the terminal symbols a and d. Since n is an arbitrary number, r_m cannot introduce a and d in w_m. This contradiction proves $\mathrm{Var}_{PR}(L) \geqq 2$. According to the previous theorem, we obtain $\mathrm{Var}_{PR}(L) = 2$.

Also Theorem 2 can be reproduced for programmed grammars, even in a sharper form.

Theorem 8. For any $n \geqq 1$ there is a language L_n such that

$$\mathrm{Var}_{PR}(L_n) = 1 , \quad \mathrm{Var}_{CF}(L_n) = n .$$

Proof. We consider the languages B_n, $n \geqq 1$, of Remark 1. It is known, [3], that $\mathrm{Var}_{CF}(B_n) = n$. Further $\mathrm{Var}_{PR}(B_n) = 1$ is proved in [1] for $n \geqq 3$. The proof for $n = 1, 2$ is left to the reader. □

5. THE MEASURE Var FOR RANDOM CONTEXT GRAMMARS

By Theorem 2 and Theorem 8 there are no functions f and g such that

$$\mathrm{Var}_{CF}(L) \leqq f(\mathrm{Var}_M(L)) \quad \text{and} \quad \mathrm{Var}_{CF}(L) \leqq g(\mathrm{Var}_{PR}(L))$$

for all $L \in \mathscr{L}(CF)$. In this chapter we prove the analogous statement for random context grammars. This improves the relation between context-free and random context grammars given in [1].

Theorem 9. For any $n \geqq 1$, there is a context-free language L_n such that

$\mathrm{Var}_{RC}(L_n) \leqq 8$ and $\mathrm{Var}_{CF}(L_n) = n$ (hence there is no mapping such that $\mathrm{Var}_{CF}(L) \leqq f(\mathrm{Var}_{RC}(L))$ for all $L \in \mathscr{L}(CF)$).

Proof. We consider the following languages

$$L_n = \bigcup_{i=1}^{n-1} \{(a^i b)^m b : m \geqq 1\} \quad \text{for any} \quad n \geqq 3$$

Clearly, if we consider a context-free grammar for L_n, then each i, $1 \leqq i \leqq n$, requires a derivation $A_i \Rightarrow^* (a^i b)^j A_i (a^i b)^k$, $j + k > 0$, and thus we have to have $A_j \neq A_i$ for $i \neq j$ and $A_i \neq S$ for $1 \leqq i \leqq n$. Consequently, $\mathrm{Var}_{CF}(L_n) = n$ because it is easy to construct a context free grammar with n nonterminals which generates L_n.

However, the following (λ-free) random context grammar generates the language L_{n+1}:

$$G = (\{S, A, B, C, D, E, X, Y\}, \{a, b\}, S, P)$$

where P consists of the following rules:

1) $(S \to A^i bB, \emptyset, \emptyset)$, $1 \leqq i \leqq n$,
2) $(B \to X, \emptyset, \emptyset)$,
3) $(A \to C, \{C\}, \{X\})$,
4) $(X \to ED, \emptyset, \{C\})$,
5) $(C \to a, \emptyset, \{D\})$,
6) $(D \to X, \{C\}, \emptyset)$,
7) $(X \to Y, \{A, C\}, \emptyset)$,
8) $(E \to A, \emptyset, \{Y\})$,
9) $(Y \to bB, \{E\}, \emptyset)$,
10) $(B \to b, \emptyset, \emptyset)$,
11) $(A \to a, \{B, X, Y, D\}, \emptyset)$.

A string $(a^i b)^m A^i bB$, $m \geqq 0$ (at the first step we have $m = 0$) can be rewritten only by rule 2) or by rule 10). After rule 10) only rule 11) can be used, and we obtain a terminal string $(a^i b)^{m+1} b$.

Using rule 2), we obtain the string $(a^i b)^m A^i bX$, which can be rewritten only by rule 3), and we get a string $(a^i b)^m A^k C A^j bX$, $k + j = i - 1$. Now, only rule 4) can be applied and then we have to use rule 5) passing to $(a^i b)^m A^k a A^j bED$. This string can be rewritten only by rule 6) which leads to $(a^i b)^m A^k a A^j bEX$. Now we have to use again rule 3) and so until we obtain $(a^i b)^m a^i bE^i X$. Then rule 7) can be used in order to obtain $(a^i b)^{m+1} E^i Y$, and in the presence of Y, rule 8) can replace each E by A, and then rule 9) leads to $(a^i b)^{m+1} A^i bB$, which has the same form as the word in the beginning of this proof. Consequently, $L(G) = L_n$, $\mathrm{Var}_{RC}(L_n) \leqq 8$.

Thus the proof is complete for $n \geqq 3$. The proof for $n = 1, 2$ is left to the reader. □

Remark 4. The relation $\mathrm{Var}_M(L) \leqq \mathrm{Var}_{RC}(L) + 2$ (Lemma 4 in [1]) can be improved to

$$\mathrm{Var}_M(L) \leqq \mathrm{Var}_{RC}(L) + 1 \quad \text{for all} \quad L \in \mathscr{L}(RE).$$

Indeed, the symbol C in the proof of [1], Lemma 4 can be removed, replacing it by the trap symbol N in rules which check the presence of the permitting symbols in the current string and replacing the appearance checking rules $B \to N$ (which verify the absence of the forbidden symbols) by $B \to NN$.

ACKNOWLEDGEMENT

The authors are grateful to the referee for his comments which considerably improved the paper.

(Received August 23, 1983.)

REFERENCES

[1] J. Dassow: Remarks on the complexity of regulated rewriting. Fund. Inform. *7* (1984), 83—103.

[2] S. Ginsburg: The Mathematical Theory of Context-free Languages. McGraw Hill, New York 1966.

[3] J. Gruska: On a classification of context-free languages. Kybernetika *3* (1967), 22—29.

[4] J. Gruska: Descriptional complexity of context-free languages. In: Proc. Symp. Math. Foundations of Computer Science '73, 71—83.

[5] O. Ibarra: Simple matrix languages. Inform. and Control *17* (1970), 359—394.

[6] O. Mayer: Some restrictive devices for context-free grammars. Inform. and Control *20* (1972), 69—92.

[7] Gh. Păun: Matrix Grammars (in Romanian). The Scientific and Encyclopaedic Publ. House, Bucharest 1981.

[8] Gh. Păun: Six nonterminals are enough for generating a recursively enumerable language by a matrix grammar. Internat. J. Comput. Math. *15* (1984).

[9] A. Pirická-Kelemenová: Doctoral dissertation, Bratislava 1980.

[10] A. Salomaa: Formal Languages. Academic Press, New York 1973.

[11] A. P. J. van der Walt: Random context languages. Information Processing 1971, North-Holland, Amsterdam 1972, 66—68.

Dr. J. Dassow, Technological University Magdeburg, Department of Mathematics and Physics, DDR-3010 Magdeburg, PSF 124. German Democratic Republic.

Dr. Gh. Păun, University of Bucharest, Faculty of Mathematics, R-70109 Bucuresti, Str. Academiei 14. Romania.