INFORMATION AND CONTROL 25, 10–19 (1974)

On the Complexity of Regulated Context-Free Rewriting

A. B. CREMERS,* O. MAYER AND K. WEISS

Department of Computer Science, University of Karlsruhe

Some complexity measures which are well-known for context-free languages are generalized in order to classify matrix languages and programmed languages. It is shown that the complexity of some context-free languages decreases if they are generated by matrix grammars or programmed grammars. An arithmetic characterization is given for infinite languages generated by two matrices. The number of matrices (as a complexity measure) is shown to be independent from any other complexity measure regarded in this paper.

1. INTRODUCTION

Matrix grammars and programmed grammars are well-known generalizations of context-free grammars; they are defined as context-free grammars with certain restrictions on the use of productions. Such grammars are called regulated context-free rewriting devices.

In Gruska (1969), several complexity criteria for context-free grammars have been investigated. Some of these criteria are generalized for matrix grammars and programmed grammars (Section 2). It is proved that certain context-free languages can be generated by such regulated rewriting devices with less variables or less productions or with a lower index as compared to their generation by ordinary context-free grammars (Section 3).

In the rest of the paper, especially matrix grammars are considered. According to matrix grammars, rewriting is only by the application of entire matrices (strings of productions). The number of matrices is introduced as a complexity measure for matrix languages (Section 4). For infinite languages generated by two matrices, an arithmetic characterization is established.

Finally, we show that there are languages which cannot be generated by

* Currently associated with University of Southern California, Computer Science Program, Los Angeles, CA, 90007.

matrix grammars which are minimal both according to the number of matrices and according to any other complexity measure considered in this paper (Section 5).

2. Complexity Measures

In this section the definitions of matrix grammars (Abraham, 1965) and programmed grammars (Rosenkrantz, 1969) are given and then three criteria of grammatical complexity are introduced. These criteria are generalizations of well-known complexity measures for context-free grammars.

For the basic notions and results of the theory of context-free languages, the reader is referred to Ginsburg (1966).

Definition. Let $G = (N, T, R, S)$ be a context-free grammar where N is the finite nonterminal alphabet, T is the finite terminal alphabet, R is a finite set of context-free productions, and S in N the start symbol.

(a) Let M be a finite set of finite strings

$$r_{i_1} r_{i_2} \cdots r_{i_{n_i}}, \qquad n_i \geqslant 1,$$

of labels of productions r_{i_j} in R. These sequences are called matrices and the pair

$$G_m = (G, M)$$

is called a context-free matrix grammar (mg). Derivations in mg's are defined as follows:

The application of a matrix $f = r_1 \cdots r_n$ to a word w in $(N \cup T)^+$, denoted by $w \overset{*}{\underset{f}{\Rightarrow}} \bar{w}$, is defined as a context-free derivation

$$w = w_0 \underset{r_1}{\Rightarrow} w_1 \underset{r_2}{\Rightarrow} \cdots \underset{r_n}{\Rightarrow} w_n = \bar{w}$$

where $w_{i-1} \Rightarrow w_i$ is realized by applying the production with label r_i. (For an alphabet T, $T^+ = T^* - \{\epsilon\}$, where ϵ denotes the empty word.) The language generated by G_m exactly contains those words of $L(G)$ which can be obtained by a successive application of entire matrices and is denoted by $L(G_m) \cdot \mathscr{M}^{\epsilon}$ denotes the family of languages generated by arbitrary mg's.

(b) A programmed grammar (with empty failure fields in the sense of Rosenkrantz), shortly pg, is a pair

$$G_p = (G, \phi),$$

where $G = (N, T, R, S)$ is a context-free grammar and ϕ is a mapping of the set F of production labels of G into the set of subsets of F.

The language generated by G_p, denoted by $L(G_p)$, exactly contains those words of $L(G)$ which possess a context-free derivation where for each pair of succeeding steps

$$w_{i-1} \underset{r_i}{\Rightarrow} w_i \underset{r_{i+1}}{\Rightarrow} w_{i+1},$$

the label r_{i+1} is in the set $\phi(r_i)$.

The family of all languages generated by pg's (with empty failure fields) is denoted by $\mathscr{P}^\epsilon$.

By Salomaa (1970), $\mathscr{M}^\epsilon = \mathscr{P}^\epsilon$. Clearly, each family properly includes the family of context-free languages.

The generation of a language by such grammars is called a generation by regulated context-free rewriting.

Definition. For an mg G_m and a pg G_p, we define

(a) $\mathrm{Var}_m(G_m)$ and $\mathrm{Var}_p(G_p)$ as the number of nonterminals of G_m and G_p, respectively;

(b) $\mathrm{Prod}_m(G_m)$ and $\mathrm{Prod}_p(G_p)$ as the number of productions of G_m and G_p, respectively.

For a context-free grammar G, the measures $\mathrm{Var}(G)$ and $\mathrm{Prod}(G)$ are analogously defined.

Definition. Let F be a terminal derivation according to an mg $G_m = (G, M) = ((N, T, R, S), M)$:

$$F\colon S = w_0 \underset{f_{i_1}}{\overset{*}{\Rightarrow}} w_1 \underset{f_{i_2}}{\overset{*}{\Rightarrow}} w_2 \overset{*}{\Rightarrow} \cdots \underset{f_{i_n}}{\overset{*}{\Rightarrow}} w_n = w, \qquad w \text{ in } T^*,$$

where f_{i_j} in M for $1 \leqslant j \leqslant n$.

We define

$$\mathrm{Ind}_m(F) = \max\{l(d(w_i) \mid 0 \leqslant i \leqslant n\},$$

where $d(w)$ is the word obtained from w by deleting all terminals, and for a word w, $l(w)$ denotes the length of w;

$$\mathrm{Ind}_m(w) = \min\{\mathrm{Ind}_m(F) \mid F \text{ is a derivation of } w \text{ according to } G_m\};$$

$$\mathrm{Ind}_m(G_m) = \max\{\mathrm{Ind}_m(w) \mid w \text{ in } L(G_m)\}.$$

If κ_γ is a complexity measure related to a class γ of grammars and L a language which can be generated by a grammar in γ, we define

$$\kappa_\gamma(L) = \min\{\kappa_\gamma(G) \mid G \text{ in } \gamma, L = L(G)\}.$$

The classes of context-free grammars, matrix grammars, and programmed grammars are denoted by c, m, and p, respectively.

3. Context-Free Languages Generated by Regulated Rewriting

In this section we show that for each of the three previously defined measures of complexity there exists a context-free language L such that the description of L by a context-free grammar is more complex than by a programmed grammar or a matrix grammar.

Theorem 1. *There is a context-free language L such that*

$$\mathrm{Var}_m(L) < \mathrm{Var}_c(L).$$

Proof. Consider the language

$$L = \{a^m b^n c^n, b^n a^m c^n, b^n c^n a^m, a^m c^n b^n, c^n a^m b^n, c^n b^n a^m \mid m, n \geqslant 1\}.$$

L is generated by an mg with

$$M = \{(S \to AAA, A \to bB, A \to cC, A \to aA), (B \to bB, C \to cC), \\ (B \to \epsilon, C \to \epsilon), (A \to aA), (A \to \epsilon)\};$$

thence, $\mathrm{Var}_m(L) \leqslant 4$.

It is easily verified that $\mathrm{Var}_c(G) > 4$ for each context-free grammar G generating L.

Lemma 1. $\mathrm{Prod}_m(L) \leqslant 2\,\mathrm{Var}_m(L) + \#(T) + 1$, *for each matrix language L.*

Proof. Let $G_m = ((N, T, R, S), M)$ be an mg generating L which is minimal according to Var_m. If each production $A \to w_1 \cdots w_l$ occurring in a matrix f of M is replaced by the following sequence of productions

$$A \to X, X \to w_1X, X \to w_2X,..., X \to w_lX, X \to \epsilon,$$

we obtain an equivalent mg $\bar{G}_m$ generating L so that

$$\mathrm{Prod}_m(\bar{G}_m) \leqslant 2\,\mathrm{Var}_m(L) + \#(T) + 1.$$

A similar argument yields

LEMMA 2. $\mathrm{Prod}_p(L) \leqslant 2\,\mathrm{Var}_p(L) + \#(T) + 1$ *for each programmed language* L *over* T.

THEOREM 2. *Let* $T = \{a\}$. *There is a finite language* L *over* T *such that*

$$\mathrm{Prod}_m(L) < \mathrm{Prod}_c(L),$$

$$\mathrm{Prod}_p(L) < \mathrm{Prod}_c(L).$$

Proof. Let $L = \{a^{2^{2^i}} \mid 0 \leqslant i \leqslant 4\}$; by Gruska (1969), $\mathrm{Prod}_c(L) = 5$; by Lemmas 1 and 2, $\mathrm{Prod}_m(L) \leqslant 4$ and $\mathrm{Prod}_p(L) \leqslant 4$.

Remark. For arbitrary context-free languages L, the differences $\mathrm{Prod}_c(L) - \mathrm{Prod}_m(L)$ and $\mathrm{Prod}_c(L) - \mathrm{Prod}_p(L)$ are not bounded.

Next, we study the classification of languages according to the measure Var_γ.

THEOREM 3. *There is a linear language* L *such that*

$$\mathrm{Var}_p(L) < \mathrm{Var}_c(L).$$

Proof. By Gruska (1969), there exists for each $n \geqslant 1$ a regular language L_n with $\mathrm{Var}_c(L_n) = n$; on the other hand, $\mathrm{Var}_p(L) = 1$ for each linear language L.

Remark. For each matrix grammar, an equivalent programmed grammar can be effectively constructed without increasing the number of variables. Thence, for each matrix language L, $\mathrm{Var}_p(L) \leqslant \mathrm{Var}_m(L)$. Clearly, the linear language L used in the proof of Theorem 1 cannot be generated by a matrix grammar with only one variable. Consequently, there are languages L with $\mathrm{Var}_p(L) < \mathrm{Var}_m(L)$.

We now establish similar results for the complexity measure Ind_m.

THEOREM 4. $\mathrm{Ind}_m(L) \leqslant \mathrm{Ind}_c(L)$ *for each context-free language* L. *There are languages with*

$$\mathrm{Ind}_m(L) < \mathrm{Ind}_c(L).$$

Proof. The first assertion is obvious; to prove the second, let us consider the Dyck-language L generated by the grammar with the productions $S \to aSb$, $S \to SS$, $S \to \epsilon$.

By Salomaa (1969), $\text{Ind}_c(L) = \infty$. But L is generated by the matrix grammar G_m with the matrices

$$(S \to AB), (A \to aA, B \to bB), (A \to a, B \to bS), (A \to a, B \to Sb),$$
$$(A \to aS, B \to b), (A \to Sa, B \to b), (A \to a, B \to b), (A \to \epsilon, B \to \epsilon),$$
$$(S \to \epsilon).$$

Clearly, $L(G_m) = L$ and $\text{Ind}_m(G_m) = 2$.

Remark. The proofs of Theorems 3 and 4 show that the differences $\text{Var}_c(L) - \text{Var}_p(L)$ and $\text{Ind}_c(L) - \text{Ind}_m(L)$ are not bounded for arbitrary context-free languages L. Similar results can be obtained for the differences $\text{Ind}_c(L) - \text{Ind}_p(L)$ for a suitable defined measure Ind_p.

4. The Complexity Measure Mat

Since in derivations according to matrix grammars entire matrices have to be applied, it makes sense to consider as a complexity measure not only the number of productions but also the number of matrices.

Definition. For an mg $G_m = (G, M)$, we define $\text{Mat}(G_m)$ as the number of matrices in M.

We now introduce a system of linear diophantine equations controlling the nonterminal balance in derivations according to matrix grammars.

Definition. Let $G_m = (G, M)$ be an mg. Let $G = (N, T, R, A_1)$ with $N = \{A_1, ..., A_n\}$ and $M = \{f_1, ..., f_m\}$, where $f_i = r_{i_1} \cdots r_{i_{n_i}}$ and $r_{i_j} : A_{i_j} \to w_{i_j}$ for $1 \leqslant i \leqslant m$, $1 \leqslant j \leqslant n_i$.

For each matrix f_i and each variable A_j, we define

$$k_{ji} = l_{A_j}(w_{i_1} w_{i_2} \cdots w_{i_{n_i}}) - l_{A_j}(A_{i_1} A_{i_2} \cdots A_{i_{n_i}}),$$

where the number of occurrences of a symbol A in a word w is denoted by $l_A(w)$.

Let K denote the matrix (k_{ji}) associated to G_m.

Obviously, k_{ji} is the number of occurrences of the variable A_j "introduced" by the application of the matrix f_i; $k_{ji} < 0$ means that the number of occurrences of A_j has decreased.

THEOREM 5. *Let w in $L(G_m)$ have the derivation τ, let x_i be the number of applications of f_i in τ, then $x = (x_1, \ldots, x_m)^T$ is a solution of the system of linear equations*

$$(+) \qquad Kx = -e_1, \quad \textit{where} \quad e_1 = (1, 0, 0, \ldots, 0)^T.$$

In connection with matrix grammars, we are only interested in nonnegative integer solutions of (+). Obviously, to each nonnegative integer solution $x = (x_1, \ldots, x_m)^T$ of (+) corresponds a finite subset $L(x)$ of $L(G_m)$, where

$$L(x) = \{w \mid w \text{ in } L(G_m), \text{ a derivation of } w \text{ contains } x_i \text{ applications of } f_i, 1 \leqslant i \leqslant m\}.$$

Note that $L(x)$ may be empty for a nonnegative integer solution x of (+).

Let k_l denote the lth row of K, i.e., $k_l = (k_{l1}, k_{l2}, \ldots, k_{lm})$.

LEMMA 3. *Let $G_m = (G, M)$ be an* mg *with $L(G_m) \neq \varnothing$. Let $k_1, k_2, \ldots, k_n$ be the set of the rows of the matrix K associated to G_m. Then k_1 is linearly independent from any subset of rows not containing k_1.*

Proof. Assume that there are rows $k_2, \ldots, k_l$ such that $k_1 = \sum_{j=2}^{l} k_j p_j$, where p_j are rational numbers and $(p_2, \ldots, p_l) \neq (0, 0, \ldots, 0)$.

Since $L(G_m)$ is not empty, there is a solution $x = (x_1, \ldots, x_m)^T$ of (+). Thence, $\sum_{i=1}^{m} k_{1i} x_i = -1$; on the other hand,

$$\sum_{i=1}^{m} k_{1i} x_i = \sum_{i=1}^{m} \sum_{j=2}^{l} k_{ji} p_j x_i = \sum_{j=2}^{l} p_j \sum_{i=1}^{m} k_{ji} x_i = 0.$$

This is a contradiction.

THEOREM 6. *Let $G_m = (G, M)$ be an* mg *and r the rank of the associated matrix K; then*

(1) Mat$(G_m) < r$ *implies* $L(G_m) = \varnothing$,

(2) Mat$(G_m) = r$ *implies* $L(G_m)$ *is finite.*

Proof. (1) If Mat$(G_m) < r$, then the system (+) is overdetermined and therefore no solution exists.

(2) If Mat$(G_m) = r$, then there is at most one integer solution.

Now, we discuss the conditions under which $L(G_m)$ is infinite provided that Mat$(G_m) = 2$.

THEOREM 7. *Let* G_m *be an* mg *with* $\mathrm{Mat}(G_m) = 2$; *let* $K = (k_{ji})$ *be the matrix associated to* G_m. *Then* $L(G_m)$ *is infinite iff* $L(G_m)$ *contains two words of different length and* $k_{11} \cdot k_{12} \leqslant 0$.

Proof. (1) If $L(G_m)$ is infinite, then by Theorem 6 the rank of K equals 1 and the first row of K is independent from the second. The existence of infinitely many solutions of $(+)$ $k_{11}x_1 + k_{12}x_2 = -1$ implies that $k_{11} \cdot k_{12} \leqslant 0$.

(2) Let $G_m = ((\{A_1, A_2, ..., A_n\}, T, R, A_1, \{f_1, f_2\})$.

$L(G_m) \neq \varnothing$ implies that for at least one matrix, say f_2, $k_{j2} \leqslant 0$ for all j, $1 \leqslant j \leqslant n$. It is impossible that also $k_{j1} \leqslant 0$ for all j, $1 \leqslant j \leqslant n$; otherwise two words of different length can only be derived in G_m if $k_{11} = -1$ and $k_{12} = -1$ which is a contradiction to $k_{11} \cdot k_{12} \leqslant 0$.

Now let w' and w'' be two words of different length in $L(G_m)$ with derivations τ' and τ'', which can be chosen in such a way that no application of f_1 is proceeded by an application of f_2.

Different word lengths of w' and w'' can only be obtained by a different number of applications of f_1 in the corresponding derivations τ' and τ''. Let n' and n'' be these numbers and $n' < n''$. Clearly, for each $p \geqslant 1$, there is a terminal derivation in G_m starting with $n' + p(n'' - n')$ applications of f_1 and followed by a suitable number of applications of f_2. These derivations generate words of increasing length.

Remark. It has been shown by Maurer (1973) that there are noncontext-free languages generated by mg's with two matrices. An example is the language generated by the mg with the matrices

$$(S \to SSS,\ S \to Sa,\ S \to Sb,\ S \to Sc) \qquad \text{and} \qquad (S \to d),$$

where a, b, c and d are terminals and S is the start symbol.

5. THE INDEPENDENCE OF THE COMPLEXITY MEASURE MAT

In this section we show that the complexity measure Mat is independent of Prod_m, Var_m, and Ind_m in the sense of the following

DEFINITION. Two complexity measures κ_1 and κ_2 for a family of languages Γ are said to be independent iff there is a language L in Γ which cannot be generated by a grammar which is minimal both according to κ_1 and κ_2.

Notation. $\kappa^{-1}(L)$ denotes the set of grammars which generate L and are minimal according to κ.

THEOREM 8. *The complexity measures*

(a) Mat *and* Prod_m ,

(b) Mat *and* Var_m ,

(c) Mat *and* Ind_m

are independent.

Proof. The finite language $L = \{\epsilon, a, aa, aaa\}$ is basic to all the parts of the proof.

(a) Obviously, $\text{Prod}_m(L) > 1$. $\text{Prod}_m(L) = 2$ because L can be generated by the following grammar $G_m = (G, M)$ with the matrices

$$(S \to \epsilon), (S \to aS, S \to \epsilon), (S \to aS, S \to aS, S \to \epsilon),$$
$$(S \to aS, S \to aS, S \to aS, S \to \epsilon).$$

Now we show that $\text{Mat}(G_m) = 4$ for each G_m generating L with $\text{Prod}_m(G_m) = 2$.

Let G_m have the productions

$$(1) \quad S \to w,$$
$$(2) \quad X \to v.$$

It is easily seen that these productions must be of the form

$$(1) \quad S \to \epsilon,$$
$$(2) \quad S \to S^k a S^j \qquad \text{with} \qquad k + j \geqslant 1.$$

The finiteness of L implies that each $w \neq S$ which is obtained from S by the application of entire matrices does not contain S. Therefore, $\text{Mat}(G_m) \geqslant \text{card}(L) = 4$.

On the other hand, L can be generated by a grammar with the following three matrices $(S \to AAA)$, $(A \to \epsilon)$, $(A \to a)$. Thence $\text{Mat}(L) \leqslant 3$; thus $\text{Prod}_m^{-1}(L) \cap \text{Mat}^{-1}(L)$ is empty.

(b) Trivially, $\text{Var}_m(L) = 1$. For each grammar G_m in $\text{Var}_m^{-1}(L)$, we can conclude $\text{Mat}(G_m) = 4$ by a similar argument as in the proof of part (a). Therefore, $\text{Var}_m^{-1}(L) \cap \text{Mat}^{-1}(L)$ is empty.

(c) Let $G_m = (G, M)$ be in $\text{Mat}^{-1}(L)$. Each sentential form $w \neq S$, which is obtained by the application of entire matrices, does not contain S; otherwise we have a contradiction either to the finiteness of L or to the assumption that G_m is in $\text{Mat}^{-1}(L)$. Obviously, $\text{Mat}(G_m) > 1$. Assume

$M = \{f_1, f_2\}$, and let f_1 contain a production $S \to w$. Since S is not a subword of u, where u is obtained from S by the application of f_1, u contains another nonterminal symbol A. Then all words of L are derived by applications of f_2 to u. But this is impossible because of the different word lengths. Thence, together with part (a), $\text{Mat}(L) = 3$.

It remains to show that for each grammar G_m with $\text{Mat}(G_m) = 3$ holds $\text{Ind}_m(G_m) > \text{Ind}_m(L) = 1$.

But this follows easily from the discussion of the two cases

(i) $(S \to \epsilon)$ in M,

(ii) $(A \to \epsilon)$ in M, $A \neq S$.

In both cases, the assumption $\text{Ind}_m(G_m) = 1$ implies that there exists one matrix in M, by the application of which at least two words of different lengths can be obtained from S or A.

Thus, $\text{Ind}_m(G) > 1$; therefore, $\text{Ind}_m^{-1}(L) \cap \text{Mat}^{-1}(L)$ is empty.

RECEIVED: March 8, 1973

REFERENCES

ABRAHAM, S. (1965), Some questions of phrase structure grammars, *Computational Linguistics* **4**, 61–70.

GINSBURG, S. (1966), "The Mathematical Theory of Context-Free Languages," McGraw–Hill, New York.

GRUSKA, J. (1969), Some classifications of context-free languages, *Information and Control* **14**, 152–179.

MAURER, H. (1973), private communication.

ROSENKRANTZ, D. (1969), Programmed grammars and classes of formal languages, *J. Assoc. Comput. Mach.* **16**, 107–131.

SALOMAA, A. (1969), On the index of context-free grammar and language, *Information and Control* **14**, 474–477.

SALOMAA, A. (1970), Periodically time-variant context-free grammars, *Information and Control* **17**, 294–311.