JOURNAL OF COMPUTER AND SYSTEM SCIENCES 8, 158–166 (1974)

On Vector Languages

A. B. CREMERS* AND O. MAYER

Computer Science Department, University of Karlsruhe, Karlsruhe, Germany

Received January 2, 1973; revised July 3, 1973

The family of vector languages properly contains all context-free languages. For vector languages the emptiness and finiteness problems are proved to be decidable. The Parikh mapping of a vector language is shown to be semilinear.

INTRODUCTION

In this paper we investigate the family of so-called vector languages which properly contains all context-free languages. Vector languages are generated by context-free grammars with certain matrixlike restrictions.

It is shown that the emptiness and finiteness problems are solvable for vector languages. Furthermore, the Parikh mapping is proved to be semilinear for each language in the family considered. These results are achieved by an arithmetic approach involving systems of linear diophantine equations.

1. PRELIMINARIES

The reader is assumed to be familiar with the basics of formal language theory. 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 the finite set of context-free productions, i.e., productions of the form $A \to w$, A in N, w in $(N \cup T)^*$, and S in N the starting symbol. $L(G)$ denotes the language generated by G.

Let $F = \{r_1, ..., r_m\}$ be an alphabet of labels. To each production $A \to w$ we associate a label r, such that different productions are denoted by different labels. The application of a production $r: A \to w$ to a word x is denoted by

$$x \Rightarrow_r y.$$

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

If $\tau: S = w_0 \Rightarrow_{r_{i_1}} w_1 \Rightarrow \cdots \Rightarrow_{r_{i_n}} w_n$ is a derivation according to G, then $r_{i_1} \cdots r_{i_n}$ in F^* is called a control word of the derivation τ.

Matrix grammars are context-free grammars with restrictions on the use of the productions (cf. [1]).

DEFINITION. A matrix grammar (mg) is a pair

$$G_m = (G, M), \qquad \text{where} \qquad G = (N, T, R, S)$$

denotes a context-free grammar, and M is a finite set of finite strings

$$f_i = r_{i_1} \cdots r_{i_{n_i}} \qquad \text{with} \qquad r_{i_j} \text{ in } F \qquad \text{for} \quad 1 \leqslant j \leqslant n_i .$$

These strings are called matrices.

The language $L(G_m)$ generated by $G_m = (G, M)$ is the set of all words in $L(G)$ which have a derivation with a control word in M^*. The family of languages generated by arbitrary matrix grammars is denoted as usual by $\mathscr{M}^\epsilon$.

We define a generalization of matrix grammars.

DEFINITION. A (generalized unordered) vector grammar (guvg) is a pair $G_v = (G, V)$ where $G = (N, T, R, S)$ is a context-free grammar and V is 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 in R. The elements of V are called vectors. Let

$$V^\pi = \{ p_1 \cdots p_n \mid p_1, \ldots, p_n \text{ in } F \text{ and some permutation of } p_1 \cdots p_n \text{ is in } V^* \}.$$

The language $L(G_v)$ generated by $G_v = (G, V)$ is the set of all words in $L(G)$ which have a derivation with a control word in V^π. $L(G_v)$ is called a vector language.

As in [2] the family of vector languages is denoted by $\mathscr{UV}_\infty^\epsilon$.

The family $\mathscr{UV}_\infty^\epsilon$ properly contains all context-free languages. The well-known non-context-free standard language

$$\{a^n b^n c^n \mid n \geqslant 1\}$$

is generated by the grammar (G, V) with

$$G = (\{S, A, B\}, \{a, b, c\}, \{S \to AB, A \to aAb, A \to ab, B \to cB, B \to c\}, S)$$

and

$$V = \{(S \to AB), (A \to aAb, B \to cB), (A \to ab, B \to c)\}.$$

THEOREM 1. $\mathscr{UV}_\infty^\epsilon \subset \mathscr{M}^\epsilon$.

The proof of Theorem 1 can be found in [2] where some other matrixlike restrictions

of context-free grammars are also considered and further inclusion results are established. In a corollary of Section 4 it will be shown that the inclusion in Theorem 1 is proper.

2. An Arithmetic Approach to Matrixlike Languages

A system of linear diophantine equations is introduced controlling the nonterminal balance in derivations according to matrixlike grammars and vector grammars.

Let $G_v = (G, V)$ be a grammar as described in Section 1. Let $G = (N, T, R, A_1)$ with $N = \{A_1, ..., A_n\}$ and $V = \{v_1, ..., v_m\}$ where $v_i = r_{i_1} \cdots r_{i_{n_i}}$ 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 vector v_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 $l_A(w)$.

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

Proposition 1. *Let w in $L(G_v)$ have the derivation τ, let x_i be the number of applications of v_i in τ, then $x = (x_1, ..., x_m)^T$ is a solution of the system of linear equations*

$$\sum_{i=1}^{m} k_{ji} x_i = \begin{cases} -1 & \text{for } j = 1, \\ 0 & \text{for } 2 \leqslant j \leqslant n. \end{cases} \tag{1}$$

In connection with vector grammars (or matrix grammars), we are only interested in nonnegative integer solutions of (1). Obviously, to each nonnegative integer solution $x = (x_1, ..., x_m)^T$ of (1) corresponds a finite subset $L(x)$ of $L(G_v) : L(x) = \{w \mid w \text{ in } L(G_v)$, a derivation of w contains x_i applications of v_i, $1 \leqslant i \leqslant m\}$. Note that $L(x)$ may be empty for a nonnegative integer solution x of (1).

In the following, we develop a finite representation of the set of nonnegative solutions of the diophantine problem (1).

Let $K = (k_{ji})$, $1 \leqslant j \leqslant n$, $1 \leqslant i \leqslant m$;

$X = \{x \mid Kx = (-1, 0, ..., 0)^T, x \text{ in } \mathbb{N}^m\}$ where $\mathbb{N} = \{0, 1, 2, ...\}$;

$H = \{x \mid Kx = (0, 0, ..., 0)^T, x \text{ in } \mathbb{N}^m\}$;

Z be a maximal subset of pairwise incomparable[1] elements of H where z in Z implies that there is no x in $H - (0, ..., 0)^T$ such that $x < z$;

[1] For $x = (x_1, ..., x_m)$ and $y = (y_1, ..., y_m)$ in $\mathbb{N}^m$, let $x \leqslant y$ if $x_i \leqslant y_i$ for all i, $1 \leqslant i \leqslant m$; let $x < y$ if $x \leqslant y$ and there exists at least one index i, $1 \leqslant i \leqslant m$, such that $x_i < y_i$; x and y are said to be incomparable if neither $x \leqslant y$ nor $y \leqslant x$.

Y be a maximal subset of X such that for all y in Y and z in Z the vector $y - z$ is not in $\mathbb{N}^m$.

It follows from well-known results that Z and Y are finite subsets of $\mathbb{N}^m$ (cf. [3, 6]).

Let $Z = \{z_1, ..., z_s\}$, $Y = \{y_1, ..., y_r\}$.

PROPOSITION 2. *The sets Z and Y can be effectively constructed* (cf. [6]).

By this construction we get the following finite representation of the set X of solutions:

$$X = \bigcup_{l=1}^{r} \left\{ x \mid x = y_l + \sum_{i=1}^{s} k_i z_i , k_i \text{ in } \mathbb{N} \right\}.$$

X is a finite union of linear sets, i.e., a semilinear set in the sense of [4].

3. DECIDABILITY RESULTS

The generalized unordered vector grammars have in common with context-free grammars, that the emptiness and finiteness of the generated languages are decidable. The proof is based on certain relations between the language generated by a *guvg* and its matching system of equations (1). For each pair (G, V) and matching system (1), let X, Y, Z be the sets as defined in Section 2.

Notation. For $u = (u_1, u_2, ..., u_m)^T$ in $\mathbb{N}^m$, let R_u denote the collection of the productions the labels of which form the word $v_1^{u_1} v_2^{u_2} \cdots v_m^{u_m}$; i.e., R_u contains for $1 \leqslant i \leqslant m$ all productions of each vector v_i exactly u_i times.

We say a collection P of productions $A_1 \to w_1, ..., A_s \to w_s$ is balanced, if $l_A(A_1 \cdots A_s) = l_A(w_1 \cdots w_s)$ for each A in $\{A_1, ..., A_s\}$.

A set D of derivations is called a complete application of the collection P, if the collection of all productions applied in D forms P.

A derivation $w_0 \Rightarrow_{r_1} \cdots \Rightarrow_{r_n} w_n$ is said to be a derivation in P, if any permutation of $r_1, ..., r_n$ is obtained by deleting suitable production labels in $v_1^{u_1} v_2^{u_2} \cdots v_m^{u_m}$.

LEMMA 1. *If P is a balanced collection of productions, then there exist nonterminals $B_1, ..., B_l$ and words $\alpha_1, \beta_1, ..., \alpha_l, \beta_l$ in T^* such that for each i, $1 \leqslant i \leqslant l$, there is a derivation*

$$(+) \qquad \tau_i : B_i \overset{*}{\Rightarrow} \alpha_i B_i \beta_i \qquad \text{in } P$$

where the collection of productions applied in τ_i is balanced for each i, $1 \leqslant i \leqslant l$, and $D = \{\tau_1, ..., \tau_l\}$ is a complete application of P.

Proof. Let $N = \{A_1, ..., A_n\}$ be the set of all nonterminals occurring in the productions of P. Clearly, the collection of productions applied in a derivation $(+)$ in P is balanced.

Now we show the existence of at least one B in N for which there is a derivation $(+)$: We consider a derivation $B \overset{*}{\Rightarrow} u$ in P. Then three cases have to be distinguished:

Case 1. u contains a variable $A \neq B$. Then the balance of P implies that the derivation $B \overset{*}{\Rightarrow} u$ can be extended to a derivation $B \overset{*}{\Rightarrow} u \overset{*}{\Rightarrow} \bar{u}$ in P and $\bar{u}$ does not contain any variable $A \neq B$.

Case 2. u contains only the variable B. Hence there exists a derivation $(+)$.

Case 3. u is a terminal word. Then the balance of P implies the existence of a production $A \to w$ occurring in P, such that $w = xBy$ and $A \Rightarrow xBy \Rightarrow^+ xuy$ is a derivation in P.

In cases 1 and 3 we extend the considered derivation in P. Either we get a derivation τ in P of the desired form $(+)$ after a finite number of extension steps, or we have a contradiction to the finiteness or balance of P.

Let P_1 be the subcollection of P, obtained by decreasing the number of occurrences of a production p in P by the number of its applications in τ. P_1 is balanced; the above arguments hold true also for P_1. Then the lemma results from the finiteness of P.

LEMMA 2. *Let (G, V) be a guvg. For elements x in X and z in Z, $L(x + z) \neq \varnothing$ if and only if $L(x + 2z) \neq \varnothing$.*

Proof. Assume $L(x + z) \neq \varnothing$. In a derivation τ of a word w' in $L(x + z)$, all productions of R_z are applied. Then all derivations $(+)$ of Lemma 1 which form a complete application of R_z can be embedded in τ. Thereby, we obtain a derivation of a word w in $L(x + 2z)$.

Let $L(x + z) = \varnothing$. We consider a derivation τ: $S \overset{*}{\Rightarrow} t$ in R_{x+z} of maximal length; let P be the collection of all productions of R_{x+z} not applied in τ. Clearly, t is a terminal word and P is balanced. Then by Lemma 1 there exist variables $B_1, ..., B_s$ and derivations $(+)$ $B_i \overset{*}{\Rightarrow} \alpha_i B_i \beta_i$, α_i and β_i terminal words, $1 \leqslant i \leqslant s$, forming a complete application of P. Furthermore, the maximal length of τ implies that no B_i, $1 \leqslant i \leqslant s$, occurs in τ. For convenience, let $A \rhd_R B$ denote the fact that there is a production $A \to uBv$ in a collection R of productions. The transitive closure of relation $\rhd_R$ is denoted by $\rhd_R^+$.

In order to prove that $L(x + 2z)$ is empty, it is sufficient to show that there is at least one production $B_i \to w_i$, $1 \leqslant i \leqslant s$, in P, such that $S \rhd_{R_{x+z}}^+ B_i$ does not hold. The proof will be by contradiction:

Assume $S \rhd_{R_{x+z}}^+ B_i$ holds for each i, $1 \leqslant i \leqslant s$.

Consider for each i, $1 \leqslant i \leqslant s$, a sequence of variables $S = A_{i_1}, A_{i_2}, A_{i_3}, \ldots, A_{i_n} = B_i$ with

$$S = A_{i_1} \underset{R_{x+z}}{\rhd} A_{i_2} \underset{R_{x+z}}{\rhd} A_{i_3} \underset{R_{x+z}}{\rhd} \cdots \rhd A_{i_{n_i-1}} \underset{R_{x+z}}{\rhd} A_{in_i} = B_i .$$

The rightmost symbol in this sequence which occurs in τ is denoted by A_i. Then $A_i \rhd_P^+ B_i$; hence, by the balance, of P the variable A_i must also occur in one of the derivations $(+)$ above. Now for A_i there must be a B_{i_1}, $1 \leqslant i_1 \leqslant s$, such that $B_{i_1} \rhd_P^+ A_i$.

Now take a fixed A_i, $1 \leqslant i \leqslant s$, and track it back by means of the relation $\rhd_P^+$ first to B_{i_1}, then this B_{i_1} to A_{i_1}, then A_{i_1} to a B_{i_2} and so on. By $2s$ such backtracking steps we get a chain

$$A_{i_s} \overset{+}{\underset{P}{\rhd}} B_{i_s} \overset{+}{\underset{P}{\rhd}} A_{i_{s-1}} \overset{+}{\underset{P}{\rhd}} B_{i_{s-1}} \overset{+}{\underset{P}{\rhd}} \cdots \overset{+}{\underset{P}{\rhd}} A_{i_2} \overset{+}{\underset{P}{\rhd}} B_{i_2} \overset{+}{\underset{P}{\rhd}} A_{i_1} \overset{+}{\underset{P}{\rhd}} B_{i_1} \overset{+}{\underset{P}{\rhd}} A_{i_0} = A_i .$$

In this chain, at least one A_j, $1 \leqslant j \leqslant s$, occurs twice. Therefore, there is a subchain

$$A_j = A_{i_{j_1}} \overset{+}{\underset{P}{\rhd}} B_{i_{j_1}} \overset{+}{\underset{P}{\rhd}} \cdots \overset{+}{\underset{P}{\rhd}} B_{i_{j_0}} \overset{+}{\underset{P}{\rhd}} A_{i_{j_0}} = A_j , \qquad 0 \leqslant j_0 \leqslant j_1 \leqslant s.$$

This implies that there are $C_{j_0}, \ldots, C_{j_{k+1}}$ such that

$$A_j = C_{j_0} \underset{P}{\rhd} C_{j_1} \underset{P}{\rhd} C_{j_2} \underset{P}{\rhd} \cdots \underset{P}{\rhd} C_{j_{k+1}} = A_j ,$$

where $C_{j_i} \neq C_{j_l}$ for $1 \leqslant i < l \leqslant k$.

Hence there is a derivation $A_j \overset{*}{\Rightarrow} uA_jv$ in P in which no production is applied twice. The balance of P implies, that this derivation can be elongated to a derivation $A_j \overset{*}{\Rightarrow} u_tA_jv_t$ in P where u_t and v_t are terminal words. This is a contradiction to the maximal length of τ.

THEOREM 2. *It is decidable whether the language generated by a guvg is* (a) *empty, and* (b) *finite.*

Proof. Let G_v be a *guvg*.

(a) Lemma 2 implies that the language generated by G_v is empty if and only if $L(y + \sum_{i=1}^{s} k_i z_i) = \varnothing$ for each y in Y and each $(k_1, \ldots, k_s)$ in $\{0, 1\}^s$.

(b) Lemma 2 implies that the language generated by G_v is finite if and only if for each y in Y, each $(k_1, \ldots, k_s)$ in $\{0, 1\}^s$ either $L(y + \sum_{i=1}^{s} k_i z_i) = \varnothing$ for $(k_1, \ldots, k_s) \neq (0, \ldots, 0)$, or $L(y + \sum_{i=1}^{s} k_i z_i) \neq \varnothing$ implies

$$L\left(y + \sum_{i=1}^{s} k_i' z_i\right) = L\left(y + \sum_{i=1}^{s} k_i z_i\right) \qquad \text{for any} \quad (k_1', \ldots, k_s')$$

which is obtained from $(k_1, \ldots, k_s)$ by increasing arbitrary nonzero elements k_i by 1. The conditions in (a) and (b) can be checked in a finite number of steps.

COROLLARY 1. *Given guvgs G_v and $\bar{G}_v$ there is no recursive procedure for obtaining a guvg generating the intersection of $L(G_v)$ and $L(\bar{G}_v)$ or a guvg generating the complement of $L(G_v)$.*

This results from the fact that the finiteness is undecidable for the intersection and the complement of context-free languages.

4. THE PARIKH MAPPING OF VECTOR LANGUAGES

In this section we consider the Parikh mapping of unordered vector languages.

DEFINITION. Let $T = \{a_1, a_2, \ldots, a_n\}$ be an alphabet. The Parikh mapping is a mapping $\psi: T^* \to \mathbb{N}^n$ defined as follows:

$$\begin{aligned}
\psi(\epsilon) &= (0, 0, \ldots, 0) \\
\psi(a_1) &= (1, 0, \ldots, 0) \\
\psi(a_2) &= (0, 1, \ldots, 0) \\
&\vdots \\
\psi(a_n) &= (0, 0, \ldots, 1) \\
\psi(xy) &= \psi(x) + \psi(y) \qquad (x, y \text{ in } T^*) \\
\psi(L) &= \bigcup_{x \text{ in } L} \psi(x) \qquad (L \subset T^*).
\end{aligned}$$

$\psi(x)$ gives the number of occurrences of each terminal symbol a_i in each word x in T^*. In [4], it is shown that $\psi(L)$ is a semilinear set for each context-free language L. The following theorem is an extension of this result.

THEOREM 3. *Let L be a vector language. Then $\psi(L)$ is a semilinear set.*

Proof. Let L be the language generated by a *guvg* (G, V). By a remark in Section 2 the set X of all nonnegative integer solutions of the matching system of equations is semilinear. By a straightforward argument using Lemma 2 the set

$$M = \{x \text{ in } X \mid L(x) \neq \varnothing\}$$

is also semilinear; M can be effectively constructed. Let w in L, then there is an

$x = (x_1, \ldots, x_m)^T$ in M, such that x_i gives the number of applications of the v_i in V in a derivation of w. Conversely, for each x in M there is such a word w in L. As each application of a vector v_i in V yields a fixed number of terminal symbols the semilinearity of L immediately results from that of M.

COROLLARY 2. *Each infinite vector language L contains a subset L' such that the lengths of the words in L' form an arithmetic progression.*

COROLLARY 3. *The family $\mathcal{UV}_\infty^\epsilon$ of vector languages is a proper subset of the family $\mathcal{M}^\epsilon$ of matrix languages.*

Proof. Following [8], a matrix grammar is given which generates a language whose Parikh mapping is not semilinear.

Let $G = (N, T, R, S)$ where $N = \{S, A, B, C, D, E\}$, $T = \{a, b, c\}$.

Let M consist of the following matrices.

$$(S \to aAE),\ (E \to DD,\ A \to A),$$
$$(A \to aB),\ (D \to EE,\ B \to B),$$
$$(B \to aA),\ (A \to C),$$
$$(B \to C),\ (C \to c),$$
$$(E \to b,\ C \to C),\ (D \to b,\ C \to C),$$
$$(S \to cb).$$

Obviously, this matrix grammar generates the language

$$L = \{a^n c b^m \mid n \geqslant 0,\ 1 \leqslant m \leqslant 2^n\}.$$

Note added in proof. The results of this paper were presented at the Symposium and Summer School on "Mathematical Foundations of Computer Science," September 3-8, 1973, High Tatras, Czechoslovakia.

REFERENCES

1. S. ABRAHAM, Some questions of phrase structure grammars, *Computational Linguistics* **4** (1965), 61–70.
2. A. B. CREMERS AND O. MAYER, On matrix languages, *Information and Control* **23** (1973), 86–96.
3. S. GINSBURG, "The Mathematical Theory of Context-Free Languages," McGraw–Hill, New York, 1966.
4. R. J. PARIKH, On context-free languages, *J. Assoc. Comput. Mach.* **13**, **4** (1966), 570–581.

5. D. J. ROSENKRANTZ, Programmed grammars and classes of formal languages, *J. Assoc. Comput. Mach.* **16**, 1 (1969), 107–131.
6. T. L. SAATY, "Optimization in Integers and Related Extremal Problems," McGraw–Hill, New York, 1970.
7. A. SALOMAA, Periodically time-variant context-free grammars, *Information and Control* **17**, 3 (1970), 294–311.
8. E. D. STOTSKIJ, Context-free grammars with restricted rewriting, *VINITI*, Moscow (in Russian), 1972.
9. K. WEISS, "Zur Erweiterung der Theorie der Erzeugungskomplexität auf nicht-kontext-freie Sprachen," Diploma-thesis, University of Karlsruhe, 1972.