# On state grammars

A. MEDUNA and GY. HORVÁTH

In this paper we study some properties of state grammars. Among others, it is shown that for every recursively enumerable language there exists a state grammar with erasing rules that generates it. Some problems concerning the descriptive complexity of state grammars are discussed.

## 1. Introduction

In the last years several types of grammars have been defined which have context-free rules and an additional mechanism which regulates the derivation process (see e.g. [1]—[6]), since such grammars can describe such special aspects of programming languages that cannot be covered by context-free grammars. One of the typical grammars of this kind is the state grammar (see [3]) — the subject under investigation in the present paper.

Intuitively, a state grammar is a context-free grammar with an additional mechanism which consists of the following: at each derivation step the grammar is in a state which influences the choice of the next production to be applied, and the next state is determined by this production. Moreover, rewriting is done in a leftmost fashion.

## 2. Preliminaries

In this section we briefly review some of the basic notions and notations of formal language theory. Items not defined explicitly are standard in language theory, see e.g. [5].

For a finite set $A$ we use $|A|$ to denote its cardinality. If $w$ is a word over an alphabet $Z$, then $|w|$ denotes the length of $w$, alph $(w)$ denotes the set of letters occuring in $w$ and $N_Y(w)$ denotes the number of occurrences of letters from $Y(\subseteq Z)$ in $w$. The empty word is denoted by $\lambda$. For a grammar $G$, $\underset{G}{\Rightarrow}$ denotes its direct derivation relation and $\underset{G}{\Rightarrow}^*$ denotes the derivation relation of $G$. We also write $\Rightarrow$ and $\Rightarrow^*$ rather than $\underset{G}{\Rightarrow}$ and $\underset{G}{\Rightarrow}^*$ if no confusion arises. A production $(p, q)$ of a grammar $G$ will also be called a rule and written as $p \to q$. For context-free grammars we use

the following notation: $G=(\Sigma, \Delta, x_0, P)$, where $\Sigma$ is the total alphabet of $G$, $\Delta \subseteq \Sigma$ is the terminal alphabet of $G$, $x_0 \in \Sigma \setminus \Delta$ is the initial letter and $P$ is the set of productions of $G$.

Now we recall the definition of a queue grammar defined in [4].

A queue grammar is a 7-tuple $Q=(\Sigma, \Delta, S, F, x_0, s_0, P)$ where $\Sigma$ and $S$ are finite sets, the total alphabet and the state set of $Q$, respectively, $\Delta \subseteq \Sigma$ is the terminal alphabet, $F \subseteq S$ is the final state set, $x_0 \in \Sigma \setminus \Delta$ is the initial letter, $s_0 \in S$ is the initial state and finally, $P \subseteq \Sigma \times (S \setminus F) \times \Sigma^* \times S$ is a finite set, the production set of $Q$. A production $(x, s, u, s')$ in $P$ is also written $(x, s) \to (u, s')$.

For any $(u, s), (v, s') \in \Sigma^* \times S$ the direct derivation $(u, s) \underset{Q}{\Rightarrow} (v, s')$ holds iff $u=xu_1$ for some $x \in \Sigma$ and $u_1 \in \Sigma^*$, and there is a production $(x, s) \to (v_1, s')$ in $P$ such that $v=u_1 v_1$. The derivation relation $\underset{G}{\Rightarrow}^*$ is the reflexive transitive closure of $\underset{Q}{\Rightarrow}$.

The language $L(Q)$ generated by $Q$ is defined by $L(Q)=\{w \in \Delta^*: (x_0, s_0) \Rightarrow^* (w, s)$ for some $s \in F\}$.

The central notion of this paper is that of a state grammar that we recall now.

A state grammar is a construct $G=(\Sigma, \Delta, S, x_0, s_0, P)$ where $\Sigma$ and $S$ are finite sets, the total alphabet and the state set, respectively, $\Delta \subseteq \Sigma$ is the terminal alphabet, $x_0 \in \Sigma \setminus \Delta$ is the initial letter, $s_0 \in S$ is the initial state and $P$ is a finite set of productions of the form $(x, s) \to (u, s')$ where $x \in \Sigma \setminus \Delta$, $s, s' \in S$ and $u \in \Sigma^*$. The direct derivation $(u,s) \underset{G}{\Rightarrow} (v, s')$ holds for $u, v \in \Sigma^*$ and $s, s' \in S$ iff there exist decompositon $u=u_1 x u_2$, $v=u_1 v_1 u_2$ and a production $(x, s) \to (v_1, s')$ in $P$ such that for every nonterminal $y \in \Sigma \setminus \Delta$ occuring in the word $u_1$ there is no production with left side $(y, s)$ in $P$.

The language $L(G)$ generated by $G$ is defined by $L(G)=\{w \in \Delta^*: (x_0, s_0) \underset{G}{\Rightarrow}^* (w, s)$ for some $s \in S\}$, where $\underset{G}{\Rightarrow}^*$ is the reflexive transitive closure of $\underset{G}{\Rightarrow}$.

A state grammar $G$ is called propagating if for every production $(x, s) \to (u, s')$ of $G$ we have $u \neq \lambda$.

We point out that the above definition differs from Kasai's original definition in [3] since Kasai considers only propagating state grammars.

The family of languages generated by context-free, contex-sensitive, type 0, queue and state grammars is denoted by $\mathscr{L}(\mathrm{CF})$, $\mathscr{L}(\mathrm{CS})$, $\mathscr{L}(\mathrm{RE})$, $\mathscr{L}(\mathrm{Q})$ and $\mathscr{L}(\mathrm{S})$, respectively.

## 3. On the generative power of state grammars with erasing productions

In his original definition of state grammars, Kasai did not consider erasing productions. It is however a natural generalization (see e.g. [6]). But then, the first question we have to ask is: what is the generative power of such grammars? We are going to show in this section that $\mathscr{L}(\mathrm{S})$ and $\mathscr{L}(\mathrm{RE})$ coincide. Although the equality $\mathscr{L}(\mathrm{S})=\mathscr{L}(\mathrm{RE})$ can be obtained as a direct consequence of a theorem in section 4, we prove it here in order to demonstrate the close connection between queue and state grammars, moreover, this connection will be used in a subsequent section.

A state grammar $G=(\Sigma, \Delta, S, \#, s_0, P)$ is called front-end state grammar if the nonterminal letter $\#$ is an endmarker, which means that every production containing $\#$ is of the two forms $(\#, s) \to (u\#, s')$ or $(\#, s) \to (u, s')$, where $\#$ does

not occur in the word $u$, moreover, for every state $s$, if there is a production with left side $(x, s)$ for some $x \neq \#$, then for each nonterminal letter $x' \neq \#$ there is a production with left side $(x', s)$.

One can see that in any derivation step according to a front-end state grammar, the rewritten non-terminal letter is either the first nonterminal in the word or the endmarker.

**Lemma 1.** For any queue grammar $Q$ one may construct an equivalent front-end state grammar $G$.

*Proof.* Let $Q=(\Sigma, \Delta, S, F, x_0, s_0, P)$ be an arbitrary queue grammar. We construct a state grammar $G=(\Sigma', \Delta, S, \#, q_0, P)$. For each terminal letter $a \in \Delta$ we introduce a new nonterminal letter $x_a$. We set

$$\Sigma' = \Sigma \cup \{x_a : a \in \Delta\} \cup \{\#\}$$

$$S' = \{q_0, q_1\} \cup S \cup (\Sigma \setminus \Delta \cup \{x_a : a \in \Delta\}) \times S.$$

We define a homomorphism $h$ of $\Sigma$ into $\Sigma'$ by $h(x)=x$ for $x \in \Sigma \setminus \Delta$ and $h(a)=x_a$ for $a \in \Delta$. The homomorphism $g$ of $\Sigma \cup h(\Delta)$ into $\Sigma$ is defined by $g(x)=x$ for $x \in \Sigma$ and $g(x_a)=a$ for $a \in \Delta$. Clearly, $g(h(x))=x$ for $x \in \Sigma$ and $h(g(x))=x$ for $x \in \Sigma \cup h(\Delta)$. Now the production set $P'=P'_0 \cup P'_1 \cup P'_2 \cup P'_3 \cup P'_4$ is constructed in the following way:

$$P'_0 = \{(\#, q_0) \to (x_0 \#, s_0)\},$$

$$P'_1 = \{(x, s) \to (\lambda, [x, s]) : s \in S \setminus F, x \in h(\Sigma)\},$$

$$P'_2 = \{(\#, [x, s]) \to (h(u) \#, s') : (g(x), s) \to (u, s') \in P\},$$

$$P'_3 = \{(\#, s) \to (\lambda, q_1) : s \in F\},$$

$$P'_4 = \{(x, q_1) \to (g(x), q_1) : x \in h(\Sigma)\}.$$

It is evident that the state grammar $G$ constructed above satisfies the front-end requirement. We are going to prove that a derivation

$$(x_0, s_0) \underset{Q}{\Rightarrow}^* (w, s) \quad \text{holds iff}$$

$$(x_0 \#, s_0) \underset{G}{\Rightarrow}^* (h(w) \#, s) \quad \text{holds.}$$

Indeed, for any production $(x, s) \to (u, s')$ in $Q$, the productions $(h(x), s) \to (\lambda, [h(x), s])$ and $(\#, [h(x), s]) \to (h(u) \#, s')$ are present in $G$, moreover, for every word $v \in h(\Sigma)^*$ the derivation

$$(h(x)v \#, s) \underset{G}{\Rightarrow} (v \#, [h(x), s]) \underset{G}{\Rightarrow} (vh(u) \#, s')$$

holds. Conversely, a derivation $(x_0 \#, s_0) \underset{G}{\Rightarrow}^* (w \#, s)$ can only be carried out by using productions from $P'_1$ and from $P'_2$. If a production $(h(x), s) \to (\lambda, [h(x), s])$, $x \in \Sigma$, is used in a derivation step, then after it a suitable production $(\#, [h(x), s]) \to (h(u) \#, s')$ must be applied, where $(x, s) \to (u, s')$ is in $P$. By the front-end property of $G$ we obtain that $(x_0, s_0) \underset{Q}{\Rightarrow}^* (w, s)$ holds. Assume that $w \in L(Q)$. Then

$(x_0, s_0) \overset{}{\underset{Q}{\Rightarrow}}^*(w, s)$ holds for some $s \in F$, consequently we obtain the derivation

$$(\#, q_0) \overset{}{\underset{G}{\Rightarrow}} (x_0\#, s_0) \overset{}{\underset{G}{\Rightarrow}}^* (h(w)\#, s) \overset{}{\underset{G}{\Rightarrow}} (h(w), q_1).$$

The iterated application of productions from $P_4'$ gives the derivation

$$(\#, q_0) \overset{}{\underset{G}{\Rightarrow}} (h(w), q_1) \overset{}{\underset{G}{\Rightarrow}}^* (g(h(w)), q_1) = (w, q_1).$$

Therefore $w \in L(G)$.

To establish the converse inclusion $L(G) \subseteq L(Q)$, assume that $w \in L(G)$. Thus $(\#, q_0) \overset{}{\underset{G}{\Rightarrow}}^* (w, q_1)$ must hold since every production of $G$ not containing nonterminal letter on its right side is a member of $P_3'$ or $P_4'$. The nonterminal letter $\#$ can only be eliminated by a production from $P_3'$, thus we obtain the derivation

$$(\#, q_0) \overset{}{\underset{G}{\Rightarrow}} (x_0\#, s_0) \overset{}{\underset{G}{\Rightarrow}}^* (v\#, s) \overset{}{\underset{G}{\Rightarrow}} (v, q_1) \overset{}{\underset{G}{\Rightarrow}}^* (w, q_1)$$

for some $s \in F$ and $v \in h(\Sigma)^*$. Furthermore, $w = g(v)$ by virtue of definition of $P_4'$, thus $h(w) = h(g(v)) = v$. We obtain the derivation $(x_0\#, s_0) \overset{}{\underset{G}{\Rightarrow}}^* (h(w), s)$, which implies $(x_0, s_0) \overset{}{\underset{Q}{\Rightarrow}}^* (w, s)$. Therefore $w \in L(Q)$. □

**Lemma 2.** Every recursively enumerable language can be generated by a front-end state grammar.

*Proof.* From Theorem 2.1 Chapter 2 in [4] we know that $\mathscr{L}(Q) = \mathscr{L}(RE)$, hence the lemma holds by Lemma 1. □

The following theorem is now an immediate consequence of Lemma 2.

**Theorem 1.** $\mathscr{L}(S) = \mathscr{L}(RE)$.

## 4. On the complexity of state grammars

It is a natural question whether or not the representation of languages by grammars of a certain type is "better" than by grammars of another type. In this section we study complexity measures of state grammars in terms of [2] as for example the number of nonterminals and the number of states sufficient to generate any language of a given type.

First we investigate the complexity of state grammars with regard to the measure of states.

It is an immediate consequence of the definition of the state grammar that any language $L$ is context-free iff $L$ can be generated by a state grammar with a single state.

**Theorem 2.** Let $L \in \mathscr{L}(RE)$. Then there exists a state grammar $G = (\Sigma, \Delta, S, x_0, s_0, P)$ such that $L = L(G)$ and $|S| = 3$.

*Proof.* Every type-0 language $L$ can be obtained in the form $L = h(L_1 \cap L_2)$ where $h$ is a homomorphism and $L_1$ and $L_2$ are context-free languages (Theorem 11.1, Part one in [6]). Assume that the languages $L_1$ and $L_2$ are generated by the context-free grammars $G_1 = (\Sigma_1, \Delta', x_0^1, P_1)$ and $G_2 = (\Sigma_2, \Delta', x_0^2, P_2)$, such that

$\Sigma_1 \cap \Sigma_2 = \emptyset$, $\Delta \cap \Delta' = \emptyset$. Let $h: \Delta' \to \Delta^*$ be the required homomorphism where $L$ is a language over the alphabet $\Delta$. Assume that $\Delta' = \{a_1, \ldots, a_n\}$ for some $n \geqq 1$ and consider an auxiliarly alphabet $\Delta'' = \{b_1, \ldots, b_n\}$. We define a function $\varphi: \Sigma_2 \to \to \Sigma_2 \setminus \Delta' \cup \Delta''$ by $\varphi(x) = x$ for $x \in \Sigma_2 \setminus \Delta$ and $\varphi(a_i) = b_i$ for $a_i \in \Delta'$. Furthermore, let $\psi: \Delta'' \to \Delta'$ be defined by $\psi(b_i) = a_i$ for $b_i \in \Delta''$.
Now we construct the state grammar $G$ in the following way:

$$\Sigma = \{x_0, \#, \bot\} \cup \Sigma_1 \cup \Sigma_2 \cup \Delta'' \cup \{a_i^j, b_i^j \colon 1 \leqq j \leqq i \leqq n\} \cup \Delta,$$

$$S = \{s_0, s_1, s_2\},$$

$$P = P^0 \cup P^1 \cup P^2 \cup P^3 \cup P^4.$$

$$P^0 = \{(x_0, x_0) \to (x_0^1 x_0^2 \#, s_0), (\#, s_0) \to (\lambda, s_1)\},$$

$$P^1 = \{(x, s_0) \to (p, s_0) \colon x \to p \in P_1\} \cup \{(y, s_0) \to (\varphi(q), s_0) \colon y \to q \in P_2\},$$

$$P^2 = \{(a_i, s_1) \to (a_i^1, s_2), (a_i^j, s_1) \to (a_i^{j+1}, s_2) \colon 1 \leqq j < i \leqq n\},$$

$$P^3 = \{(b_i, s_2) \to (b_i^1, s_1), (b_i^j, s_2) \to (b_i^{j+1}, s_1) \colon 1 \leqq j < i \leqq n\},$$

$$P^4 = \{(a_i^i, s_1) \to (h(a_i), s_0), (b_i^i, s_0) \to (\lambda, s_1), (b_i^i, s_2) \to (\bot, s_2) \colon 1 \leqq i \leqq n\}.$$

We prove first that $L \subseteq L(G)$. Let $w = h(v)$ for some $v \in L_1 \cap L_2$. Then there exist (leftmost) derivations $x_0^1 \underset{G_1}{\Rightarrow}{}^* v$ and $x_0^2 \underset{G_2}{\Rightarrow}{}^* v$. By the construction of the production set $P^1$ we obtain the derivation

$$(x_0, s_0) \underset{G}{\Rightarrow} (x_0^1 x_0^2 \#, s_0) \underset{G}{\Rightarrow}{}^* (v x_0^2 \#, s_0) \underset{G}{\Rightarrow}{}^* (v\varphi(v) \#, s_0) \underset{G}{\Rightarrow} (v\varphi(v), s_1).$$

Using productions from $P^2$, $P^3$ and $P^4$ we have

$$(x_0, s_0) \underset{G}{\Rightarrow}{}^* (v\varphi(v), s_1) \underset{G}{\Rightarrow}{}^* (h(v), s_1).$$

Therefore $w \in L(G)$.

To establish the converse inclusion $L(G) \subseteq L$ consider a derivation $(x_0, s_0) \underset{G}{\Rightarrow}{}^* \underset{G}{\Rightarrow}{}^* (w, s)$ for some $w \in \Delta^*$ and $s \in S$. This derivation must start with the use of the production $(\#, s_0) \to (x_0 x_0^2 \#, s_0)$, and the nonterminal letter $\#$ can be eliminated by the production $(\#, s_0) \to (\lambda, s_1)$ only if

$$(x_0^1 x_0^2 \#, s_0) \underset{G}{\Rightarrow}{}^* (v_1 \varphi(v_2) \#, s_0) \underset{G}{\Rightarrow} (v_1 \varphi(v_2), s_1) \underset{G}{\Rightarrow}{}^* (w, s)$$

holds for some words $v_1, v_2 \in \Delta'^*$ such that $x_0^1 \underset{G_1}{\Rightarrow}{}^* v_1$ and $x_0^2 \underset{G_2}{\Rightarrow}{}^* v_2$. Suppose that $v_1 = a_{i_1} \ldots a_{i_k}$, $a_{i_1}, \ldots, a_{i_k} \in \Delta'$, and $\varphi(v_2) = b_{j_1} \ldots b_{j_l}$, $b_{j_1}, \ldots, b_{j_l} \in \Delta''$. Considering the production sets $P^2$ and $P^3$ one can see that the nonterminal $a_{i_1}$ can be derived to a terminal word in the derivation $(v_1 \varphi(v_2), s_1) \underset{G}{\Rightarrow}{}^* (w, s)$ only if the subderivation

$$(a_{i_1} \ldots a_{i_k} b_{j_1} \ldots b_{j_l}, s_1) \underset{G}{\Rightarrow}{}^* (a_{i_1}^{i_1} \ldots a_{i_k} b_{j_1}^{i_1} \ldots b_{j_l}, s_1) \underset{G}{\Rightarrow}$$

$$\underset{G}{\Rightarrow} (h(a_{i_1}) a_{i_2} \ldots a_{i_k} b_{j_1} \ldots b_{j_l}, s_0)$$

holds. Moreover, this subderivation can be continued only if $i_1 = j_1$ and then the production $(b_{i_1}^{i_1}, s_0) \to (\lambda, s_1)$ must be used. Repeated application of the above proc-

ess yields the equalities $k=l$, $i_t=j_t$ $(t=1, \ldots, k)$, i.e. $\varphi(v_1)=\varphi(v_2)$ and $w=h(v_1)$. Since $v_1=\psi(\varphi(v_1))=\psi(\varphi(v_2)=v_2$ we conclude that $w\in h(L_1\cap L_2)=L$. □

Now we will study the complexity of state grammars with regard to the measure of nonterminals. First we shall give a uniform definition of the (uncontrolled) finite restriction for grammars.

We say that a grammar $G$ is of index $k$ (for some positive integer $k$) if, for every word in the language $L(G)$ there exists a derivation such that no word in this derivation contains more than $k$ occurrences of nonterminal symbols. We say that $G$ is of uncontrolled index $k$ if every word in each derivation of a word in $L(G)$ contains no more than $k$ occurrences of nonterminal symbols. Finally, we say that $G$ is of (uncontrolled) finite index if it is of (uncontrolled) index $k$ for some positive integer $k$.

**Lemma 3.** Let $G=(\Sigma, \Delta, S, x_0, s_0, P)$ be a state grammar such that $\Sigma\setminus\Delta=\{x_0\}$. Then there exists an equivalent context-free grammar $G'$, moreover, if $G$ is of uncontrolled index $k$ then so is $G'$.

*Proof.* We construct the context-free grammar $G'=(\Sigma', \Delta, y_0, P')$

$$\Sigma' = \{y_0\}\cup S\times S\cup\Delta,$$

$$P' = \{y_0 \to [s_0, s]\colon s\in S\}\cup$$

$$\{[s, s_{k+1}] \to u_1[s_1, s_2]u_2[s_2, s_3]\ldots u_k[s_k, s_{k+1}]u_{k+1}\colon k\geqq 1$$

$$s, s_1, \ldots, s_{k+1}\in S,\ u_1, \ldots, u_{k+1}\in\Delta^*,$$

$$(x_0, s) \to (u_1x_0u_2\ldots u_kx_0u_{k+1}, s_1)\in P\}\cup$$

$$\{[s, s'] \to u\colon (x_0, s) \to (u, s')\in P,\ u\in\Delta^*\}.$$

We prove by induction on the length of derivation that for any $s, s'\in S$ and $w\in\Delta^*$, $(x_0, s)\underset{G}{\Rightarrow}{}^*(w, s')$ holds iff $[s, s']\underset{G'}{\Rightarrow}{}^*w$ holds. If $(x_0, s)\underset{G}{\Rightarrow}(w, s')$ is a direct derivation then $[s, s']\underset{G'}{\Rightarrow}w$ holds by the definition of $P'$. Assume that for every derivation $(x_0, s)\underset{G}{\Rightarrow}{}^*(w, s')$ of length less than a given integer $l$ $(\geqq 2)$ the derivation $[s, s']\underset{G'}{\Rightarrow}{}^*w$ holds. Let

$$(x_0, s)\underset{G}{\Rightarrow}(u_1x_0u_2\ldots u_kx_0u_{k+1}, s_1)\underset{G}{\Rightarrow}{}^*(u_1v_1u_2\ldots u_kv_ku_{k+1}, s') = (w, s')$$

be a derivation of length $l$, where $k\geqq 1$ and $v_i\in\Delta^*$ $(i=1, \ldots, k+1)$. Since $x_0$ is the only nonterminal letter in $G$ there are derivations

$$(x_0, s_1)\underset{G}{\Rightarrow}{}^*(v_1, s_2), \ldots, (x_0, s_k)\underset{G}{\Rightarrow}{}^*(v_k, s_{k+1})$$

for some states $s_2, \ldots, s_k\in S$ and $s_{k+1}=s'$ such that the length of each derivation is less than $l$. By the induction hypothesis we obtain derivations $[s_1, s_2]\underset{G'}{\Rightarrow}{}^*v_1, \ldots, \ldots, [s_k, s']\underset{G'}{\Rightarrow}{}^*v_k$. Furthermore, the rule $[s, s']\to u_1[s_1, s_2]u_2\ldots u_k[s_k, s']u_{k+1}$ is in $P'$, thus $[s, s']\underset{G'}{\Rightarrow}{}^*w$ holds. The reverse implication can be proved similarly. The second statement of the Lemma follows immediately. □

**Theorem 3.** Any language $L$ can be generated by a context-free grammar of uncontrolled index $k$ iff there exists a state grammar $G$ of uncontrolled index $k$ such that the number of nonterminals of $G$ is one and $L=L(G)$.

*Proof.* Let

$$G' = (\Sigma, \Delta, x_0, P')$$

be a context-free grammar of uncontrolled index $k$ for some $k \geqq 1$. Obviously, for any $w \in L(G')$ there is a leftmost derivation such that no word in this derivation contains more than $k$ occurrences of nonterminals. Consider a state grammar

$$G = (\Delta \cup \{y_0\}, \Delta, S, y_0, s_0, P)$$

where $y_0$ is a new nonterminal,

$$S = \{[\alpha]: \alpha \in (\Sigma \setminus \Delta)^*, 0 \leqq |\alpha| \leqq k\},$$

$$s_0 = [x_0],$$

and the set of productions $P$ is defined as follows:

i) if $A \to u_0 B_1 u_1 \ldots u_{n-1} B_n u_n \in P'$

where $A, B_1, \ldots, B_n \in \Sigma \setminus \Delta$, $u_0, u_1, \ldots, u_n \in \Delta^*$ for some $n \geqq 1$

then $(y_0, [A\alpha]) \to (u_0 y_0 u_1 \ldots u_{n-1} y_0 u_n, [B_1 \ldots B_n \alpha]) \in P$ for every $[A\alpha] \in S$;

ii) if $A \to u \in P'$ where $u \in \Delta^*$

then $(y_0, [A\alpha]) \to (u, [\alpha]) \in P$ for every $[A\alpha] \in S$;

iii) each element of $P$ is obtained by i) or ii).

It follows immediately that $L(G')=L(G)$ and that $G$ is of uncontrolled index $k$. The reverse implication is true by Lemma 3. □

For the definition of metalinear languages we refer to [5].

**Corollary 1.** The family of metalinear languages is properly contained in the family of languages generated by state grammars of uncontrolled finite index with one nonterminal.

*Proof.* The statement follows from Theorem 3 and Theorem 3.14 in [6]. □

Now we recall the definition of forbidding context grammars.

A forbidding context grammar is a construct $G=(\Sigma, \Delta, x_0, P)$ which is very much like a context-free grammar except that each production $p$ of $P$ is of the form $A \to \alpha, F$ where $F$ is a subset of the nonterminal alphabet $\Sigma \setminus \Delta$, called the forbidding field of $p$. Such a production $p$ can be used to derive a word $w$ in context-free fashion only if $F \cap \text{alph}(w) = \emptyset$. For detailed information we refer e.g. to Chapter 3 in [4].

**Theorem 4.** For any forbidding context grammar of uncontrolled index $k$ one may construct an equivalent state grammar of uncontrolled index $k$ with two nonterminals.

*Proof.* Let

$$G' = (\Sigma', \Delta, x_0, P)$$

be an arbitrary forbidding context grammar of uncontrolled index $k$ for some $k \geqq 1$. We construct a state grammar

$$G = (\Sigma, \Delta, S, y_0, [\succ x_0], P)$$

in the following way:

$$\Sigma = \{y_0, y_0'\} \cup \Delta,$$

$$S = \{[\alpha \succ \beta], [\alpha \prec \beta] : \alpha, \beta \in (\Sigma' \setminus \Delta)^*, \quad 0 \leqq |\alpha\beta| \leqq k, \succ \text{ and } \prec \text{ are new symbols}\},$$

$$P = P_1 \cup P_2 \cup P_3 \cup P_4, \text{ where}$$

$$P_1 = \{(y_0, [\alpha_1 \succ A\alpha_2]) \to (y_0', [\alpha_1 A \succ \alpha_2]),$$
$$(y_0', [\alpha_1 A \prec \alpha_2]) \to (y_0, [\alpha_1 \prec A\alpha_2]) : A \in \Sigma' \setminus \Delta,$$
$$\alpha_1, \alpha_2 \in (\Sigma' \setminus \Delta)^* \text{ and } |\alpha_1\alpha_2| \prec k\},$$

$$P_2 = \{(y_0, [\prec \alpha]) \to (y_0, [\succ \alpha]) : \alpha \in (\Sigma' \setminus \Delta)^*, |\alpha| \leqq k\},$$

$$P_3 = \{(y_0, [\alpha_1 \succ A\alpha_2]) \to (u_0 y_0 u_1 \ldots u_{m-1} y_0 u_m, [\alpha_1 \prec B_1 \ldots B_m \alpha_2])$$
$$: m \geqq 1, \quad A \to u_0 B_1 u_1 \ldots u_{m-1} B_m u_m, F \in P',$$
$$A, B_1, \ldots, B_m \in \Sigma' \setminus \Delta, u_0, u_1, \ldots, u_m \in \Delta^*,$$
$$\alpha_1, \alpha_2 \in (\Sigma' \setminus \Delta)^* \text{ and } |\alpha_1\alpha_2| \prec k, \ F \cap \text{alph}(\alpha_1 A \alpha_2) = \emptyset\},$$

$$P_4 = \{(y_0, [\alpha_1 \succ A\alpha_2]) \to (u, [\alpha_1 \prec \alpha_2]) : A \to u, F \in P',$$
$$F \cap \text{alph}(\alpha_1 A \alpha_2) = \emptyset, \ u \in \Delta^*, \ \alpha_1, \alpha_2 \in (\Sigma' \setminus \Delta)^*, \ |\alpha_1\alpha_2| \prec k\}.$$

It is easy to verify that $L(G') = L(G)$ and that $G$ is of uncontrolled index $k$. □

**Corollary 2.** Let $L$ be a language generated by a state grammar of finite index. Then $L$ can be generated by a state grammar of finite index with (at most) two nonterminals.

*Proof.* Immediately follows from Theorem 4. and Theorem 3.22 in [6]. □

**Theorem 5.** Any recursively enumerable language can be generated by a state grammar with at most three nonterminals.

*Proof.* Let $L \in \mathscr{L}(\text{RE})$. We may assume by Lemma 2 that $L$ is generated by front-end state grammar $G = (\Sigma, \Delta, S, \#, s_0, P)$. Let us denote by $X$ the set of nonterminals of $G$ excluding $\#$. Assume that $X = \{x_1, \ldots, x_n\}$ for some $n \geqq 1$. We define a coding function $\varphi : X \to \{0, 1\}^*$ by

$$\varphi(x_i) = 0^i 1 \quad \text{for } i = 1, \ldots, n.$$

Extend $\varphi$ to a homomorphism of $\Sigma^*$ into $\{0, 1, \#\}^*$ by $\varphi(y) = y$ for $y \in \Delta \cup \{\#\}$.

From the front-end property of the state grammar $G$ it follows that the state set $S$ can be partitioned into subsets $S_1$ and $S_2$. Let $S_1$ be the set of all states $s \in S$ such that for every $x \in X$ there is a production in $P$ with left side $(x, s)$. Now the state grammar

$$G' = (\Sigma', \Delta, S', \#, [\lambda, s_0], P')$$

is constructed in the following way.

$$\Sigma' = \{0, 1, \#\} \cup \Delta,$$

$$S' = \{0^i : 0 \leqq i \leqq n\} \times S,$$

$$P' = P_1' \cup P_2' \cup P_3', \quad \text{where}$$

$$P_1' = \{(0, [0^i, s]) \to (\lambda, [0^{i+1}, s]) : s \in S_1, 0 \leqq i < n\},$$

$$P_2' = \{(1, [0^i, s]) \to (\varphi(u), [\lambda, s']) : (x_i, s) \to (u, s') \in P\},$$

$$P_3' = \{(\#, [\lambda, s]) \to (\varphi(u), [\lambda, s']) : (\#, s) \to (u, s') \in P\}.$$

One can see that for every $s \in S$ and every $w \in \Sigma^*$ a derivation $(\#, s_0) \underset{G}{\Rightarrow}^* (w, s)$ holds iff $(\#, [\lambda, s_0]) \underset{G'}{\Rightarrow}^* (\varphi(w), [\lambda, s])$ holds. Since $\varphi(w) = w$ if $w \in \Delta^*$, we obtain the desired equality $L(G) = L(G')$. □

To conclude this section let us remark that it remains an open question whether the number of nonterminals (three) in Theorem 5 is minimal.

COMPUTING CENTRE
TECHNICAL UNIVERSITY
OBRÁNCŮ MIRU 21
BRNO
CZECHOSLOVAKIA

A. JÓZSEF UNIVERSITY
BOLYAI INSTITUTE
ARADI VÉRTANÚK TERE 1.
6720 SZEGED
HUNGARY

## References

[1] DASSOW, J., Comparison of some types of regulated rewriting. Technological University Magdeburg, Department of Mathematics and Physics, Technical Report SMA 58/83 (1983).
[2] DASSOW, J. and PAUN, G., Further remarks on the complexity of regulated rewriting. Technological University Magdeburg, Department of Mathematics and Physics, Technical Report SMA 70/83 (1983).
[3] KASAI, T., A hierarchy between context-free and context sensitive languages. Journal of Computer and System Sciences 4 (1970), 492—508.
[4] KLEIJN, H. C. M., Selective substitution grammars based on context-free production. Ph. D. Thesis, University Of Leiden (1983).
[5] SALOMAA, A., Formal languages. Academic Press, New York 1973.
[6] VERMIER, D., On structural restrictions of ET0L Systems. Ph. D. Thesis, University of Antwerpen (1978).

*(Received July 2, 1987)*