KYBERNETIKA — VOLUME *30* (1994), NUMBER 2, PAGES 177 – 186

SYNTACTIC COMPLEXITY OF REGULATED REWRITING

Alexander Meduna, Cynthia J. Crooks and Milan Šárek

The syntactic complexity of regulated grammars with respect to the number of nonterminals is investigated. Several characterizations of the family of recursively enumerable languages are established; most importantly, it is proved that this family is defined by programmed grammars with only seven nonterminals.

1. OVERVIEW

Recently, the language theory has systematically investigated the syntactic complexity of various grammars with respect to the number of nonterminals (see Chapter 4 in [3] and references therein). In particular, several characterizations of the family of recursive enumerable languages by regulated grammars with a reduced number of nonterminals have been established.

For context free grammars, the parameter of the number of nonterminals leads to an infinite hierarchy of languages (see [5]). On the other hand, two (three) nonterminals are sufficient for the generation of all linear (metalinear) languages by matrix grammars; moreover, there are non-context-free languages that can be generated by matrix grammars with only one nonterminal (see [2] and [4]). Furthermore, any recursively enumerable language can be generated by matrix grammars with six nonterminals and by programmed grammars with eight nonterminals (see [6]). For further results, the reader is referred to [2–6].

In this paper, we contribute to this vivid area of the language theory by presenting several characterizations of the family of recursively enumerable languages based upon regulated grammars, including matrix and programmed grammars, with a reduced number of nonterminals. In particular, we characterize the family of recursively enumerable languages by seven-nonterminal programmed grammars.

2. PRELIMINARIES

We assume that the reader is familiar with the basic concepts and notation used in formal languages (Chapter 1 in [3]). Next, we give informal descriptions of the grammars discussed in this paper.

For a grammar, G, the following notation is used:

(1) N and T are the alphabets of nonterminals and terminals, respectively; $V = N \cup T$;

(2) $S \in N$ denotes the axiom of G;

(3) P denotes the set of productions of G;

(4) F_G (or simply F if G is understood) denotes the set of labels assigned to the productions in P.

Unless stated otherwise, the language generated by G, $L(G)$, is defined as $L(G) = \{w : w \in T^*,\ S \Rightarrow^* w\}$ where $\Rightarrow^*$ denotes the reflexive and transitive closure of the direct derivation $\Rightarrow$. The notation $x \Rightarrow y\ [p]$ is used to express $x \Rightarrow y$ according to production p; $x \Rightarrow^* y\ [p]$ indicates that p was the last production applied in $x \Rightarrow^* y$.

The productions of a programmed grammar, $G = (N, T, P, S)$, are of the form

$$(\ell : A \to \alpha, \Gamma_S, \Gamma_F)$$

where $\ell \in F, A \in N, \alpha \in V^*, \Gamma_S \subseteq F$, and $\Gamma_F \subseteq F$. A production of this form is used so A is replaced with α, and, in the next direct derivation, a rule with a label in Γ_S is used. If A does not occur in the current sentential form, this production allows us to use a production with a label from Γ_F in the next step. An unconditional transfer programmed grammar is a programmed grammar in which each production of the above form has $\Gamma_S = \Gamma_F$.

A matrix grammar has the form $G = (N, T, M, S, R)$ where M, called the set of matrices, is a finite set of sequences of productions, and R is a subset of F_G. All of the productions of a matrix must be used sequentially; if a label, ℓ, appears in R and the production labeled with ℓ is not applicable to the current sentential form, then this production is skipped.

A regularly controlled grammar has the form $G = (N, T, P, S, C, K)$, where productions in P are of the form $\ell : A \to w$ with $A \in N, w \in V^*, \ell \in F$; C is a regular language over F, and $K \subseteq F$. If $i_0\, i_1 \cdots i_n \in C (n > 0)$, $t_0, \ldots, t_n \in V^*$ so $t_0 = S$, $t_n \in T^*$, and for every $j, 0 \leq j \leq n-1$, either $t_j = u_j A_j v_j$, $t_{j+1} = u_j w_j v_j$, $i_j : A_j \to w_j \in P$ (for some u_j, $v_j \in V^*$) or $t_j = t_{j+1}$, $i_j : A_j \to w_j \in P$, A_j does not occur in t_j and $i_j \in K$, then t_n is a word in the language generated by G. The generated language consists of all words obtained in this way.

The classes of matrix, programmed, unconditional transfer programmed, regularly controlled, and type-0 grammars are denoted by M, PR, $UTPR$, RC, and RE, respectively. Let X be a class of grammars, $X \in \{M, PR, UTPR, RC, RE\}$; then, $\mathcal{L}(X)$ denotes the class of languages defined by X, that is, $\mathcal{L}(X) = \{L(G) : G \in X\}$. Let $G \in X$, N is the set of nonterminals in G, and $L \in \mathcal{L}(X)$. $\mathcal{NONTER}(G)$ denotes the cardinality of N. We set

$$\mathcal{N}(X, L) = \min\{\mathcal{NONTER}(G) : G \in X, L(G) = L\}.$$

3. NONTERMINAL COMPLEXITY OF REGULATED REWRITING

This section establishes several results concerning the nonterminal complexity of grammars defined the previous section.

Theorem 1. For every $L \in \mathcal{L}(RE) : \mathcal{N}(RC, L) \leq \mathcal{N}(M, L)$.

Proof. Let $G = (N, T, M, S, R)$, $G \in M$, $L(G) = L$, and $\mathcal{NONTER}(G) = \mathcal{N}(M, L)$. Then, this theorem follows immediately from the first part of the proof of Theorem V.6.1 in [7] because this construction introduces no new nonterminal. □

Theorem 2. For every $L \in \mathcal{L}(RE) : \mathcal{N}(PR, L) \leq \mathcal{N}(RC, L) + 1$.

Proof. Intuitively, we define a controlled language of a regularly controlled grammar, G, by a finite automaton, A. Then, we construct a programmed grammar G that synchronously simulates both the derivations in G and the computations in A.

Formally, let $G \in RC$, $L = L(G)$, and $\mathcal{NONTER}(G) = \mathcal{N}(RC, L)$. Assume that G is of the form $G = (N, T, P, S, L(A), K)$ where $A = (Q, \Sigma, \delta, q_0, Q_f)$ is the completely specified finite automaton – see [1] (in this paper, the set of final states is denoted by Q_f).

The new set of labels, $F_{G'}$, is defined as:

$$F_{G'} = \{\langle \gamma, \ell \rangle : \gamma \in Q, \ell \in F_G\} \cup \{\langle FIRST \rangle, \langle LAST \rangle\}.$$

Let $\delta(\gamma, \ell) = q$, $q \in Q_F$, and $\ell : A \to \alpha \in P$. Then, we define a new programmed production, p, as follows:

(1) if $\ell \notin K$, then $p = (\langle \gamma, \ell \rangle : A \to \alpha, \{\langle q, k \rangle : k \in F_G\} \cup \{\langle LAST \rangle\}, 0)$;

(2) if $\ell \in K$, then $p = (\langle \gamma, \ell \rangle : A \to \alpha,\ \{\langle q, k \rangle : k \in F_G\} \cup \{\langle LAST \rangle\},$ $\{\langle q, k \rangle : k \in F_G\} \cup \{\langle LAST \rangle\})$.

Let P' be the set of all productions obtained in this way.

Now, let $\delta(\gamma, \ell) = q$, $q \notin Q_F$, $\ell : A \to \alpha \in P$. Then, we add to P' a new programmed production according to the following two conditions:

(1) if $\ell \notin K$, then $p = (\langle \gamma, \ell \rangle : A \to \alpha,\ \{\langle q, k \rangle : k \in F_G\}, 0)$;

(2) if $\ell \in K$, then $p = (\langle \gamma, \ell \rangle : A \to \alpha,\ \{\langle q, k \rangle : k \in F_G\}, \{\langle q, k \rangle : k \in F_G\})$.

Let $X \notin N \cup T$. Consider the programmed grammar $G' = (N \cup \{X\}, T, \{(\langle FIRST \rangle : X \to SX, \{\langle q_0, \ell \rangle : \ell \in F_G\}, 0), (\langle LAST \rangle : X \to \lambda, 0, 0\} \cup P', X)$ — observe that X is the only new nonterminal.

Next, we prove that $L(G) = L(G')$.

Claim 1. $S \Rightarrow^n w\ [p]$ in G iff $X \Rightarrow^m wX\ [\langle \gamma, p \rangle]$ in G' for some $n \geq 1$, $m \geq 2$, $w \in (N \cup T)^*$, $p \in F_G$, $\langle \gamma, p \rangle \in F_{G'}$ (p and $\langle \gamma, p \rangle$ are the labels of the last productions applied in these derivation).

Only if: If $S \Rightarrow^n w\ [p]$ in G, then $X \Rightarrow^* wX\ [\langle \gamma, p \rangle]$ in G' for some $n \geq 1$.

Base Case: Let $n = 1$. If S in G is to derive any w, then there surely exists a production, p, such that $p : S \to \alpha$ for some $\alpha \in (N \cup T)^*$. From the above construction, we see that the following two productions are in G':

$$\langle FIRST \rangle : S \to SX$$

and

$$\langle q_0, p\rangle : S \to \alpha.$$

Therefore, $S \Rightarrow^1 \alpha[p]$ in G and $S \Rightarrow SX\ [\langle FIRST\rangle] \Rightarrow \alpha\, X\ [\langle q_0, p\rangle]$ in G'.

Induction Hypothesis: Assume that if $S \Rightarrow^n w\ [p]$ in G, then $X \Rightarrow^* wX[\langle \gamma, p\rangle]$ in G' for all $n \geq 1$.

If $S \Rightarrow^n w[p] \Rightarrow y$ is a valid derivation in G, then the following two conditions hold:

a. A production, $p_1 : A \to \alpha \in P$, was applied to w to derive y in G: $S \Rightarrow^n w[p] \Rightarrow y[p_1]$.

b. The state of A is some $\gamma\prime \in Q$ and $\delta(\gamma\prime, p_1) = q$ for some $q \in Q$.

Case 1: $w = y$.

a. Production p_1 is not applicable to w, that is, $A \notin alph(w)$.

b. $p_1 \in K$; therefore, it can be ignored in G.

c. At least one of the following productions is in P':
   1. $(\langle \gamma\prime, p_1\rangle : A \to \alpha, \{\langle q, k\rangle : k \in F_G\} \cup \{\langle LAST\rangle\}, \{\langle q, k\rangle : k \in F_G\} \cup \{\langle LAST\rangle\})$ if $q \in Q_F$.
   2. $(\langle \gamma\prime, p_1\rangle : A \to \alpha, \{\langle q, k\rangle : k \in F_G\}, \{\langle q, k\rangle : k \in F_G\})$ if $q \notin Q_F$.

d. By the induction assumption, $X \Rightarrow^* wX[\langle \gamma, p\rangle] \Rightarrow yX[\langle \gamma\prime, p_1\rangle]$.

e. Processing can continue in G' because Γ_F of $\langle \gamma\prime, p_1\rangle$ is nonempty.

Case 2: $w \neq y$.

a. Production p_1 is applicable to w in G, that is, $A \in alph(w)$.

b. At least one of the productions presented in Case 1.c or of the following productions appear in P':
   1. $(\langle \gamma\prime, p_1\rangle : A \to \alpha, \{\langle q, k\rangle : k \in F_G\} \cup \{\langle LAST\rangle\}, 0)$, if $q \in Q_F$.
   2. $(\langle \gamma\prime, p_1\rangle : A \to \alpha, \{\langle q, k\rangle : k \in F_G\}, 0)$, if $q \notin Q_F$.

c. By the induction assumption, $X \Rightarrow^* wX[\langle \gamma, p\rangle] \Rightarrow yX[\langle \gamma\prime, p_1\rangle]$ in G'.

Hence, if $S \Rightarrow^{n+1} y\ [p_1]$ in G, then $X \Rightarrow^* yX[\langle \gamma\prime, p_1\rangle]$ in G'.

Therefore if $S \Rightarrow^n w\ [p]$ in G, then $X \Rightarrow^* wX[\gamma, p]$ in G' for all $n \geq 1$ and $w \in (N \cup T)^*$.

If: If $X \Rightarrow^m wX\ [\langle \gamma, p_\alpha\rangle]$ in G', then $S \Rightarrow^* w\ [p_\alpha]$ in G for some $m \geq 2$.

Base Case: $m = 2$. For any production $\langle q_0, p\rangle : S \to \alpha \in P'$ in G', there exists a production $p : S \to \alpha \in P$ in G. Therefore, $S \Rightarrow SX\ [\langle FIRST\rangle] \Rightarrow \alpha X\ [\langle q_0, p\rangle]$ in G' and $S \Rightarrow \alpha\ [p]$ in G.

Induction Hypothesis: Assume that for some $m \geq 2$, we have if $X \Rightarrow^n wX\ [\langle \gamma, p_\alpha\rangle]$ in G', then $S \Rightarrow^* w\ [p_\alpha]$ in G for all $n = 2, \ldots, m$.

Induction Step: If $X \Rightarrow^m wX[\langle \gamma, p_\alpha\rangle] \Rightarrow yX$ is a valid derivation in G', the following conditions (a) through (d) hold:

a. A production of the form $\langle \gamma\prime, p_{\alpha+1} \rangle : A \to \alpha \in P'$ was applied to w to derive y in G':

$$X \Rightarrow^m wX[\langle \gamma, p_\alpha \rangle] \Rightarrow yX[\langle \gamma\prime, p_{\alpha+1} \rangle].$$

b. $p_{\alpha+1} : A \to \alpha \in P$ in G.

c. $\delta(\gamma\prime, p_{\alpha+1}) = q, q \in Q$.

d. $p_1\, p_2 \cdots p_\alpha\, p_{\alpha+1}$ is a prefix of a string $s \in L(A)$.

Case 1: $w = y$.

a. Production $\langle \gamma\prime, p_{\alpha+1} \rangle$ is not applicable to w in G', that is, $A \notin alph(w)$.

b. Γ_F of production $\langle \gamma\prime, p_{\alpha+1} \rangle$ is nonempty if processing is to continue.

c. $p_{\alpha+1} \in K$.

d. By the induction assumption, $S \Rightarrow^* w[p_\alpha] \Rightarrow y[p_{\alpha+1}]$ in G.

Case 2: $w \neq y$.

a. Production $\langle \gamma\prime, p_{\alpha+1} \rangle$ is applicable to w in G, that is, $A \in alph(w)$.

b. By the induction assumption, $S \Rightarrow^* w[p_\alpha] \Rightarrow y[p_{\alpha+1}]$ in G.

Hence, if $X \Rightarrow^{m+1} yX\ [\langle \gamma\prime, p_{\alpha+1} \rangle]$ in G', then $S \Rightarrow^* y\ [p_{\alpha+1}]$ in G.

Therefore, if $X \Rightarrow^m wX\ [\langle \gamma, p_\alpha \rangle]$ in G', then, by induction, $S \Rightarrow^* w\ [p]$ in G for all $m \geq 2$ and $w \in (N \cup T)^*$.

Claim 2: $S \Rightarrow^n w[p]$ in G if and only if $X \Rightarrow^m wX[\langle \gamma, p \rangle] \Rightarrow w[\langle LAST \rangle]$ in G' for some $n \geq 1$, $m \geq 2$, $w \in T^*$, $p \in F_G$, $\langle \gamma, p \rangle \in F_{G'}$, $\gamma \in Q_F$ (p and $\langle \gamma, p \rangle$ are the labels of the last productions applied in these derivation).

Since $\gamma \in Q_F$, if $\langle \gamma, p \rangle$ was applicable, $\langle LAST \rangle \in \Gamma_{S\gamma}$; otherwise, $\langle LAST \rangle \in \Gamma_{F\gamma}$. That is, the production $\langle LAST \rangle$ may be applied to the word wX in G'. Observe that this is the only way to continue the derivation, so, $S \Rightarrow^n w[p]$ iff $X \Rightarrow^m wX[\langle \gamma, p \rangle] \Rightarrow w[\langle LAST \rangle]$.

Hence, $L(G) = L(G')$. □

The construction of the proof of Theorem 2 implies the following two corollaries.

Corollary 1 (Normal Form of Programmed Grammars). For every $L \in \mathcal{L}(RE)$, there exists $G^{\in}PR$, $G = (N, T, P, S)$, such that

(i) $L = L(G)$;

(ii) if $(\ell : A \to \alpha, \Gamma_S, \Gamma_F) \in P$, then either $\Gamma_S = \Gamma_F$ or $\Gamma_F = 0$.

Corollary 2. Let $G = (N, T, P, S, C, K)$, $G^{\in}RC$ and $F_G = K$. Then $L(G) \in \mathcal{L}(UTPR)$.

By Lemma 5 in [4], $N(PR, L) \leq N(M, L) + 2$ (for every $L \in \mathcal{L}(RE)$). Next, we improve this relation.

Theorem 3. For every $L \in \mathcal{L}(RE) : N(PR, L) \leq N(M, L) + 1$.

Proof. It follows from Theorems 1 and 2. □

Theorems 1 and 3 together with Theorem 2 in [6] imply the following two results.

Corollary 3. For every $L \in \mathcal{L}(RE)$:

(i) $\mathcal{N}(RC, L) \leq 6$;

(ii) $\mathcal{N}(PR, L) \leq 7$.

Notice that (ii) improves the relation $\mathcal{N}(PR, L) \leq 8$ (for every $L \in \mathcal{L}(RE)$) in [2].

Theorem 4. For every $L \in \mathcal{L}(RE)$, $\mathcal{N}(RC, L) \leq \mathcal{N}(PR, L) + 1$

Proof. Informally, given a programmed grammar, G, we construct a regularly controlled grammar, G', whose controlled language is defined by a finite automaton, A. G and A synchronously simulate derivations of G.

Formally, consider $G^{\in} PR$, $L = L(G)$, $G = (N, T, P, S)$, and $\mathcal{NONTER}(G) = \mathcal{N}(PR, L)$.

Let $G' \in RG$, $G' = (N \cup \{\mathcal{B}\}, T, P', S, C, K)$, be defined as follows:

Let $\mathcal{B}$ be a new symbol, $\mathcal{B} \notin N \cup T$. P' and $F_{G'}$ are defined by:

if $(\ell : A \to \alpha, \Gamma_S, \Gamma_F) \in P$, then add $\{[\ell, 1], [\ell, 2]\}$ into F_G, and $\{[\ell, 1]{:}A \to \alpha,\ [\ell, 2]{:}A \to \mathcal{B}\}$ into P'.

Now, a nondeterministic finite automaton (see [1]), A, is constructed as follows:

$$A = (Q, F_{G'}, \delta, q_0, Q - \{q_0\})$$

where $Q = \{q_0, q_{\text{empty}}\} \cup F_G$ (we assume that q_0, $q_{\text{empty}} \notin F_G$), and δ is defined as follows:

for each $(\ell : A \to \alpha, \Gamma_S, \Gamma_F) \in P$, we define:

(1) $\delta(\ell, [\ell, 1]) = \{k : k \in \Gamma_s\}$ if $\Gamma_S \neq 0$,

(2) $\delta(\ell, [\ell, 1]) = \{q_{\text{empty}}\}$ if $\Gamma_S = 0$,

(3) $\delta(\ell, [1, 2]) = \{k : k \in \Gamma_F\}$ if $\Gamma_F \neq 0$,

(4) $\delta(\ell, [\ell, 2]) = \{q_{\text{empty}}\}$ if $\Gamma_F = 0$,

(5) $\delta(q_0, [\ell, 1]) = \{k : k \in \Gamma_S\}$ if $A = S$ and $\Gamma_S \neq 0$,

(6) $\delta(q_0, [\ell, 1]) = \{q_{\text{empty}}\}$ if $\Gamma = 0$.

Let $C = L(A)$ and $K = \{[\ell, 2] : 1 \in F_G^{\}}$.

The proof of the equivalence of G and G' follows next:

Claim 1: $S \Rightarrow^n w[p_1]$ in G where $w \Rightarrow^* t_1$ in G with $t_1 \in T^*$ iff $S \Rightarrow^m w[p_1, \ell_1]$ in G' where $w \Rightarrow^* t_2$ in G' with $t_2 \in T^*$, $\ell_1 \in \{1, 2\}$, for some $n, m \geq 1$, $w \in (N \cup T)^*$ ($p_1 \in P$, $[p_1, \ell_1] \in P'$ are the last productions applied in these derivations).

Only if: If $S \Rightarrow^n w[p_1]$ in G where $w \Rightarrow^* t_1$ in G with $t_1 \in T^*$, then $S \Rightarrow^* w[p_1, \ell_1]$ in G', $n \geq 1$, $w \in (N \cup T)^*$, $w \Rightarrow^* t_2$ in G' with $t_2 \in T^*$, and $\ell_1 \in \{1, 2\}$.

Base Case: Let $n = 1$. There must exist at least one production in G of the form $p : S \to \alpha$ where $\alpha \in (N \cup T)^*$, which implies that productions $[p, 1] : S \to \alpha$ and $[p, 2] : S \to \mathcal{B}$ are in G'. As production $[p, 2]$ would block a complete derivation in G', $[p, 1]$ is surely used. Therefore, $S \Rightarrow \alpha\ [p]$ in G and $S \Rightarrow \alpha[p, 1]$ in G'. The rest of the base case is left to the reader.

Induction Hypothesis: Assume that if $S \Rightarrow^n w\ [p_1]$ in G where $w \Rightarrow^* t_1 \in T^*$, then $S \Rightarrow^* w[p_1, \ell_1]$ in G' where $w \Rightarrow^* t_2$ in G' with $t_2 \in T^*$,$\ell_1 \in \{1, 2\}$, for all $n \geq 1$.

If $S \Rightarrow^n w\ [p_1] \Rightarrow y$ is a valid derivation in G, then the following conditions hold:

a. A production $p_2 \in P$ was applied to w to derive y in G, that is $S \Rightarrow^n w\ [p_1] \Rightarrow y\ [p_2]$.

b. p_1 and p_2 are labels of productions of the form:

$$(p_1 : A \to \alpha, \Gamma_{S1}, \Gamma_{F1}),$$
$$(p_2 : B \to \beta, \Gamma_{S2}, \Gamma_{F2}),$$

$\alpha, \beta \in (N \cup T)^*,\ A, B \in N.$

c. By the definition of a programmed grammar, $\Gamma_{S1} \cup \Gamma_{F1} \neq 0$.

d. $p_2 \in \Gamma_{S1} \cup \Gamma_{F1}$.

If $S \Rightarrow^* w[p_1, \ell_1]$ is a valid derivation in G', then the following conditions hold:

a. The sequence of labels $[p_0, \ell_0] \ldots [p_1, \ell_1]$ is a valid prefix of a word $s \in L(A)$, and $[p_1, \ell_1]$ is the label of the last production applied in the derivation in G'.

b. If $p_2 \in \Gamma_{S1}$ in G, then $p_2 \in \delta(p_1, [p_1, 1])$ in G'. If $p_2 \in \Gamma_{F1}$ in G, the $p_2 \in \delta(p_1, [p_1, 2])$ in G'.

Case 1: $w = y$.

a. B is not a substring of w, that is, $B \notin alph(w)$.

b. Production p_2 is not applicable to w in G.

c. If the derivation is to continue, then Γ_{F2} of p_2 in G is nonempty,

d. If Γ_{F2} of p_2 in G is nonempty, then $\delta(p_2, [p_2, 2])$ in G' is nonempty and $q_{\text{empty}} \notin \delta(p_2, [p_2, 2])$.

e. Since $\delta(p_2, [p_2, 2])$ is defined, $[p_0, l_0] \ldots [p_1, l_1][p_2, 2]$ is a valid prefix of $s \in L(A)$ in G'.

f. Because $[p_2, 2] \in K$, processing can continue in G'.

g. By the induction assumption, $S \Rightarrow^* w[p_1, \ell_1] \Rightarrow y[p_2, 2]$ in G'.

Case 2: $w \neq y$.

a. B is a substring of w, that is, $B \in alph(w)$.

b. Production p_2 is applicable to w in G.

c. If the derivation is to continue, then Γ_{S2} of p_2 is nonempty.

d. If Γ_{S2} of p_2 in G is nonempty, then $\delta(p_2, [p_2, 1])$ is nonempty and does not contain q_{empty}.

e. As $\delta(p_2, [p_2, 1])$ is defined, $[p_0, \ell_0] \dots [p_1, \ell_1][p_2, 1]$ is a valid prefix of $s \in L(A)$ in G'.

f. By the induction assumption, $S \Rightarrow^* w[p_1, l_1] \Rightarrow y[p_2, 1]$ in G'.

Hence, if $S \Rightarrow^{n+1} y[p]$ in G, where $y \Rightarrow^* t_1$ in G with $t_1 \in T^*$, then $S \Rightarrow^* y[p, \ell]$ in G', where $y \Rightarrow^* t_2$ in G with $t_2 \in T^*, \ell \in \{1, 2\}$. Therefore, we have completed the only if part of the induction.

If: If $S \Rightarrow^m w\, [p_1, \ell_1]$ in G' where $w \Rightarrow^* t_2$ in G' with $t_2 \in T^*$, then $S \Rightarrow^* w\, [p_1]$ where $w \Rightarrow^* t_1$ in G with $t_1 \in T^*$ in G, $m \geq 1$, $w \in (N \cup T)^*$, $\ell_1 \in \{1, 2\}$ ($p_1 \in P$, $[p_1, l_1] \in P'$ are the last productions applied in the derivations).

Base Case: Let $m = 1$. For any derivation to yield a word $t_2 \in T^*$ in G', there surely exists a production, $[p, 1] : S \rightarrow \alpha$, where $\alpha \in (N \cup T)^*$ in G'. Then, by the construction, there exists a production $p : S \rightarrow \alpha$ in G. Hence $S \Rightarrow \alpha[p, 1]$ in G', and $S \Rightarrow \alpha[p]$ in G. The rest of the base case is left to the reader.

Induction Hypothesis: Assume that if $S \Rightarrow^i w\, [p_1, \ell_1]$ in G', where $w \Rightarrow^* t_2 \in T^*$, $l_1 \in \{1, 2\}$, then $S \Rightarrow^* w\, [p_1]$ in G where $w \Rightarrow^* t_1 \in T^*$ for any i satisfying $i \leq m$ for some $m \geq 1$.

If $S \Rightarrow^m w\, [p_1, \ell_1] \Rightarrow y$ is a valid derivation in G', then the following holds:

a. a production $[p_2, \ell_2]$ was applied to w to derive y in G', that is, $S \Rightarrow^m w[p_1, \ell_1] \Rightarrow y[p_2, \ell_2]$.

b. productions $[p_1, \ell_1]$ and $[p_2, \ell_2]$ are of the form:

$$[p_1, 1] : A \rightarrow \alpha,$$
$$[p_1, 2] : A \rightarrow \mathcal{B},$$
$$[p_2, 1] : B \rightarrow \beta,$$
$$[p_2, 2] : B \rightarrow \mathcal{B}.$$

$A, B \in N$, $\alpha, \beta \in (N \cup T)^*$.

c. The production sequence $[p_0, \ell_0] \dots [p_1, \ell_1][p_2, \ell_2]$ is a valid prefix of a control word $s \in L(A)$ in G'.

d. By the construction, the following productions belong to P in G:

$$(p_1 : A \rightarrow \alpha, \Gamma_{S_1}, \Gamma_{F_1}),$$
$$(p_2 : B \rightarrow \beta, \Gamma_{S_2}, \Gamma_{F_2}).$$

e. $\Gamma_{S_1} \cup \Gamma_{F_1} \neq 0$.

Case 1: $w = y$.

a. B is not a substring of w, that is, $B \notin alph(w)$.

b. Production $[p_2, \ell_2]$ is not applicable to w.

c. If the derivation is to continue in G', then:

1. $[p_2, \ell_2] \in K$ and $\ell_2 = 2$.
2. The current state of A in G' is not q_{empty}.

d. $p_2 \in \Gamma_{S1}$ and/or $p_2 \in \Gamma_{F1}$ in G; therefore, p_2 can be applied at this point in the derivation.

e. Γ_{F2} in G is nonempty and processing can continue.

f. By the induction assumption, $S \Rightarrow^* w[p_1] \Rightarrow y[p_2]$ in G.

Case 2: $w \neq y$.

a. B is a substring of w, that is, $B \in alph(w)$.

b. Production $[p_2, \ell_2]$ is applicable to w in G'.

c. If $\ell_2 = 2$, then B would be replaced by $\mathcal{B}$ and y does not derive $t_2 \in T^*$ in G'; therefore, $\ell_2 = 1$ for processing to continue.

d. If the derivation is to continue in G', then the current state of A in G' cannot be q_{empty}.

e. $p_2 \in \Gamma_{S1}$ and/or $p_2 \in \Gamma_{F1}$ in G and p_2 can be used at this point in the derivation.

f. $\Gamma_{S2} \neq 0$ in G, so y can continue to be processed in G.

g. By the induction assumption, $S \Rightarrow^* w[p_1] \Rightarrow y[p_2]$ in G.

Hence, if $S \Rightarrow^{m+1} y[p_2, \ell_2]$ in G', $S \Rightarrow^* y[p_2]$ in G. Therefore, the if part of the induction holds.

Claim 2: $S \Rightarrow^* t$, $t \in T^*$ in G if and only if $S \Rightarrow t, t \in T^*$ in G'.

Claim 2 follows from Claim 1. Consider the case when $S \Rightarrow^* w[p]$ in G where $w \in T^*$ (p is the label of the last production applied in the derivation). From Claim 1, we know that $S \Rightarrow^* w[p, \ell]$ in G', $w \in T^*$ and $[p, \ell]$ is the label of the last production applied. Thus, $w \in L(G)$. Since all states, excluding q_0, of A are final states the control word $[p_0, \ell_0] \dots [p, \ell] \in L(A)$ and $w \in L(G')$.

Hence, $L(G) = L(G')$. □

Corollary 4 (Normal Form of Regularly Controlled Grammars). For each $L \in \mathcal{L}(RE)$, there exists $G \in RC$, $G = (N, T, P, S, C, K)$, such that:

(i) $L = L(G)$;

(ii) if $x \in C$, then every (nonempty) prefix of x is also from C.

Corollary 5. $L = L(G)$ for some $G \in UTPR$ if and only if $L = L(G')$ for some $G' \in RC$, and $K = F_{G'}$.

Proof. *If:* See Corollary 2.

Only if: This can be established by analogy with the method of the proof of Theorem 4 (we only take $[\ell, 1]$–$[\ell, 2]$ for all $\ell \in F_G$ and omit from P' every production for which the right side is equal to $\mathcal{B}$). □

The following corollary follows from Theorems 3 and 4.

Corollary 6. For every $L \in \mathcal{L}(RE) : N(RC, L) \leq N(M, L) + 2$.

(Received October 17, 1989.)

REFERENCES

[1] A.V. Aho and J.D. Ullman: The Theory of Parsing, Translation, and Computing, Vol. 1: Parsing. Prentice-Hall, N.J. 1972.

[2] J. Dassow and G. Păun: Further remarks on the complexity of regulated rewriting. Kybernetika *21* (1985), 213–227.

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

[4] J. Dassow: Remarks on the Complexity of Regulated Rewriting. Fund. Inform. *7* (1984), 83–103.

[5] J. Gruska: Descriptional Complexity of Context-Free Languages. Proc. 2^{nd} MFCS, 71–83, 1973.

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

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

Alexander Meduna, Department of Computer Science, University of Missouri – Columbia, Columbia, MO65211, USA and Computing Center, Technical University of Brno, Údolní 19, 60 200 Brno. Czech Republic.

Cynthia J. Crooks, Department of Computer Science, University of Missouri – Columbia, Columbia, MO 65211. USA.

Milan Šárek, Institute of Computer Science, Masaryk University, Burešova 20, 60 200 Brno. Czech Republic.