Theoretical Computer Science 28 (1984) 299–314
North-Holland

COMPLEXITY OF NORMAL FORM GRAMMARS*

Alica KELEMENOVÁ
Mathematical Institute, Slovak Academy of Sciences, 814 73 Bratislava, Czechoslovakia

Communicated by A. Salomaa
Received October 1982
Revised March 1983

Abstract. Various types of grammars can be used to describe context-free languages. Such are context-free grammars and their normal form restrictions. Rewriting of a context-free grammar to an equivalent grammar in required (normal) form can cause a change of parameters of the grammar such as the number of rules, the number of nonterminals, etc. Greibach normal form grammars and position restricted grammars will be investigated from the point of view of descriptional complexity of context-free languages.

1. Introduction

Various types of grammars can be used to describe context-free languages. Such are context-free grammars and their normal form restrictions (e.g., Chomsky normal form, Greibach normal form, etc.). Also a special subclass of context-sensitive grammars—terminal bounded grammars—has that property [2].

Rewriting of a context-free grammar to an equivalent grammar in required (normal) form can cause, in general, a change of such parameters of the grammar as the number of rules, the number of nonterminals, etc. For instance, the ε-rule removal substantially increases the number of rules of context-free grammars, while the total length of the grammar can increase at most ten-times [6].

In the present paper Greibach normal form grammars and position restricted grammars will be investigated from the point of view of descriptional complexity of context-free languages. We shall prove that the number of nonterminals needed to describe a language by a grammar in Greibach normal form does not exceed twice the number of nonterminals needed to describe it by an ε-free chain-free reduced grammar. Increasing of productions and total length for grammars in Greibach normal form is at most cubic in the number of productions or in the total length for ε-free chain-free reduced grammars, respectively. We shall give an example, where increasing is exactly square. (For the results discussed above, terminals are allowed in any position on right-hand sides of rules of grammars in Greibach normal form.)

* Part of the results of this paper were previously reported in [11].

For the case of language description by position restricted grammars, the increase of values for the number of nonterminals, the number of rules, the number of grammatical levels and the height of grammatical levels cannot be bounded by any function.

For the total length of grammars we shall give optimal linear increase both for position restricted grammars of type (0, 0, 0) and for grammars in Chomsky normal form with respect to the total length of ε-free chain-free reduced grammars.

2. Preliminaries

We assume the reader to be familiar with the basic formal language theory in a scope of [8] or [12]. We briefly review some well-known notations and definitions from descriptional complexity of formal languages but refer to [7] for further details.

In this paper only context-free grammars and languages will be considered.

We shall use $G=(T, N, P, S)$ for a context-free grammar, where N and T denote the sets of nonterminals and terminals, respectively, P is the set of rules and S is the initial nonterminal.

To evaluate the complexity of the grammar $G=(N, T, P, S)$, we shall use five parameters known from literature. We shall call them by common name complexity measure. Three of them reflect in some way the size of the grammar. They are:

$$\mathrm{Var}(G)=|N| \quad \text{(the number of nonterminals).}$$

$$\mathrm{Prod}(G)=|P| \quad \text{(the number of productions).}$$

$$\mathrm{Symb}(G)=\sum(2+|\alpha|) \quad \text{(the total length of } G\text{)}$$

where the sum goes over all productions $A\to\alpha$ in P.

Two further parameters, based on the notion of grammatical level, reflect the structure of grammars. To introduce them, first we define the relation $\rhd$ on the nonterminals of the grammar $G=(N, T, P, S)$.

For A, B in N, $A \rhd B$ if there is a production $A\to xBy$ in P. The relation $\rhd^*$ is the reflexive and transitive closure of $\rhd$.

We say that nonterminals A and B are structurally equivalent and we write $A\equiv B$ if $A\rhd^* B$ and $B\rhd^* A$.

Every class of partition of N by the equivalence relation $\equiv$ is called a grammatical level of G. For two different grammatical levels Q_1 and Q_2 we write $Q_1 > Q_2$ if there are nonterminals A in Q_1 and B in Q_2 such that $A \rhd B$. By the graph of grammatical levels of G we shall understand the digraph, the nodes of which are grammatical levels of G and edges are ordered pairs (Q_1, Q_2) for $Q_1 > Q_2$.

Now we are ready to introduce complexity measures Lev and Hei.

$$\mathrm{Lev}(G)=\text{"the number of levels in } G\text{"}.$$

$$\mathrm{Hei}(G)=\max\{\mathrm{Hei}(Q): Q \text{ is a grammatical level of } G\},$$

where $\mathrm{Hei}(Q)=1$ iff $S\in Q$ and $\mathrm{Hei}(Q_i)=1+\max\{\mathrm{Hei}(Q_t)\colon Q_t>Q_i\}$, i.e., $\mathrm{Hei}(G)$ denotes the length of the longest way in the acyclic graph of grammatical levels, starting with the initial level (i.e., the level containing S).

Any complexity measure K from Var, Prod, Symb, Lev and Hei allows us to introduce the grammatical complexity of the language L generated by grammars from $\mathscr{G}$ in the following manner:

$$K_{\mathscr{G}}(L)=\min\{K(G)\colon L(G)=L,\ G\in\mathscr{G}\}.$$

For $\mathscr{G}$ being the class of all context-free grammars we shall omit the subscript $\mathscr{G}$ in $K_{\mathscr{G}}(L)$.

By the complexity of class $\mathscr{L}$ of languages we shall mean

$$K_{\mathscr{G}}(\mathscr{L})=\sup\{K_{\mathscr{G}}(L)\colon L\in\mathscr{L}\}.$$

A grammar $G=(N, T, P, S)$ is reduced if

(1) for any $A\in N$, $S\Rightarrow^* \alpha A\beta$ for some α, β in $(N\cup T)^*$, and

(2) for any $A\in N$ there is a w in T^* such that $A\Rightarrow^* w$.

We shall use $\mathscr{R}$ to denote the class of all completely reduced grammars, i.e., reduced grammars with no ε-rules (i.e., that of $A\to\varepsilon$) and no chain rules (i.e., that of type $A\to B$, A, B in N).

In this paper the problem to compare the values $K_{\mathscr{N}}(L)$ and $K_{\mathscr{R}}(L)$ will be investigated, where $\mathscr{R}$ is the class of completely reduced grammars and $\mathscr{N}$ is the class of grammars in Greibach normal form or alternatively the class of position restricted grammars of type t. We prefer to compare normal form complexity with that on $\mathscr{R}$ instead of that on the whole of context-free grammars since comparison for K and $K_{\mathscr{R}}$ gives nontrivial results [6, 7]. Also, known techniques transforming a grammar to an equivalent one in Greibach normal form or to an equivalent one in position restricted grammar (especially to a grammar in Chomsky normal form) use as input a grammar in completely reduced form or produce a completely reduced grammar as an intermediate result of the algorithm.

For our purpose it is sufficient to take into consideration ε-free languages only. We shall use the following ε-free definitions of normal form grammars.

Definition. A grammar $G=(N, T, P, S)$ is in Greibach normal form if all its productions are of type $A\to a\alpha$ for $A\in N$, $a\in T$, $\alpha\in N^*$.

The class of grammars in Greibach normal form will be denoted by $\mathscr{G}_t$.

Definition. A grammar $G=(N, T, P, S)$ is in weak Greibach normal form if all its productions are of type $A\to a\alpha$ for $A\in N$, $a\in T$ and $\alpha\in(N\cup T)^*$.

A class of grammars in weak Greibach normal form will be denoted by $\mathscr{G}_t'$.

Definition. A grammar $G=(N, T, P, S)$ is a position restricted grammar of type

$t=(m_1, m_2, \ldots, m_k, m_{k+1})$, where m_i is a nonnegative integer for $1 \leq i \leq k+1$ if every production of the grammar is either of the form

(i) $A \to w_1 A_1 w_2 A_2 \cdots w_k A_k w_{k+1}$, where $A, A_i \in N$, $1 \leq i \leq k$ and $w_j \in T^{m_j}$ for $1 \leq j \leq k+1$, or

(ii) $A \to aB$ or $A \to a$, where $A, B \in N$ and $a \in T$.

The collection of position restricted grammars of type $t=(m_1, m_2, \ldots, m_k, m_{k+1})$ will be denoted by $\mathscr{t}$ if t is understood or by $(m_1, \ldots, m_k, m_{k+1})$.

Remark. Grammars in Chomsky normal form forms the proper subclass of position restricted grammars of type (0, 0, 0). Position restricted grammars are treated in [3, 4, 10].

Algorithms for transformations of grammars to equivalent grammars in weak Greibach normal form are well known and they are described in Appendix A, together with the transformations of grammars to equivalent position restricted grammars of type (0, 0, 0) and to equivalent grammars in Chomsky normal form.

In order to formulate our results, Knuth's symbols O, Ω and Θ will be used. For $f, g: \mathbb{Z} \to \mathbb{Z}$ ($\mathbb{Z}$ is the set of integers) the meaning of $g(n)=O(f(n))$, $g(n)=\Omega(f(n))$ and $g(n)=\Theta(f(n))$ is as in [9], i.e.,

$g(n)=O(f(n))$ if there are positive numbers c and n_0 such that $|g(n)| \leq cf(n)$ for all $n \geq n_0$,

$g(n)=\Omega(f(n))$ if there are positive numbers c and n_0 such that $g(n) \geq cf(n)$ for all $n \geq n_0$,

$g(n)=\Theta(f(n))$ if there are positive numbers c, c' and n_0 such that $cf(n) \leq g(n) \leq c'f(n)$ for all $n \geq n_0$.

For a complexity measure K we shall define the set of languages $\mathscr{L}_K^{(n)} = \{L: K(L)=n\}$, where n is a natural number, and the function $\mathscr{L}_K$, which associates with n the set of languages $\mathscr{L}_K^{(n)}$, i.e., for every n, $\mathscr{L}_K(n)=\mathscr{L}_K^{(n)}$.

Lemma 2.1. *Let K be a complexity measure and $\mathscr{N}$ be a set of normal form grammars.*

If for any natural number $m \geq k_0$ there is a language L_m and a constant c such that $K_{\mathscr{N}}(L_m)=c$ and $K_{\mathscr{N}}(L_m)=m$, then $K_{\mathscr{N}}(\mathscr{L}_K(n))$ cannot be bounded by any function $\varphi(n)$.

Notation. (1) Let A be a nonterminal of a grammar $G=(N, T, P, S)$. Then

$$L(A)=\{w: w \in T^*, A \Rightarrow^* w\}.$$

(2) For a letter $a \in T$ and for a word $w=x_1x_2\cdots x_n$, $x_i \in T$ for $1 \leq i \leq n$, we shall write $a \in \mathrm{alph}(w)$ if $a=x_j$ for some j.

For a language L and for a letter a, $a \in \mathrm{alph}(L)$ if $a \in \mathrm{alph}(w)$ for some w in L.

Lemma 2.2. *Let $G=(N, T, P, S)$ be a completely reduced grammar and Z_1, Z_2 be nonterminals in the same grammatical level. Then*

$$\text{alph}(L(Z_1))=\text{alph}(L(Z_2)).$$

Proof. Z_1 and Z_2 are in the same grammatical level so there are words $\alpha_1, \alpha_2, \beta_1$ and β_2 such that $Z_2 \Rightarrow^* \alpha_1 Z_1 \beta_1 \Rightarrow^* w_1$ and $Z_1 \Rightarrow^* \alpha_2 Z_2 \beta_2 \Rightarrow^* w_2$. The words generated by Z_1 appear as subwords of words generated by Z_2 and vice-versa. □

Notation. Let $T_1 \subseteq T$. By e_{T_1} we denote the homomorphism

$$e_{T_1}(x)=\begin{cases}\varepsilon & \text{if } x \in T_1,\\ x & \text{if } x \in T-T_1.\end{cases}$$

For $L \subseteq T^*$, $e_{T_1}(L)=\{e_{T_1}(w): w \in L\}$.

Lemma 2.3. *Let $G=(N, T, P, S)$ be a reduced grammar. A language $e_{\Sigma-\{a\}}(L(G))$ is infinite iff G contains a nonterminal A such that $A \Rightarrow^* \alpha A \beta$ and $a \in \text{alph}(\alpha\beta)$.*

Proof. Obvious. □

3. Greibach normal form complexity

In this section we shall investigate the complexity of the weak Greibach normal form. An obvious correspondence between $K_{\mathcal{G}_1}$ and $K_{\mathcal{G}_1'}$ is given by the following theorem.

Theorem 3.1. *Let L be a language and $|\text{alph}(L)|=k$. Then*

$$K_{\mathcal{G}_1}(L) \leq k + K_{\mathcal{G}_1'}(L) \quad \text{for } K \in \{\text{Var}, \text{Prod}, \text{Lev}\},$$

$$\text{Symb}_{\mathcal{G}_1}(L) \leq 3k + \text{Symb}_{\mathcal{G}_1'}(L) < 4\,\text{Symb}_{\mathcal{G}_1'}(L),$$

$$\text{Hei}_{\mathcal{G}_1}(L) \leq 1 + \text{Hei}_{\mathcal{G}_1'}(L).$$

All these estimates are definitive (i.e., a sequence of languages $\{L_k\}_{k=1}^{\infty}$ can be given such that $|\text{alph}(L_k)|=k$ and in the estimates above equalities hold).

Proof. The last part of algorithm $\mathcal{A}_1$ (see Appendix A) beginning with label (L3) produces a grammar in Greibach normal form for input a grammar in weak Greibach normal form. The inequalities can be shown by a straightforward analysis of that part of algorithm $\mathcal{A}_1$.

The equalities are fulfilled, e.g., for the languages $L_k=\{a_1^2 a_2 \cdots a_k\}$. □

Theorem 3.2. *For every language L, $\text{Var}_{\mathcal{G}_1'}(L) \leq 2\,\text{Var}_{\mathcal{R}}(L)$. The coefficient 2 in the*

estimate is the best possible [i.e., for every $\delta > 0$ there is a language L_δ such that $\mathrm{Var}_{\mathscr{G}_{ii'}}(L_\delta) > (2-\delta)\,\mathrm{Var}_{\mathscr{R}}(L_\delta)$].

Proof. The inequality $\mathrm{Var}_{\mathscr{G}_{ii'}}(L) \leq 2\,\mathrm{Var}_{\mathscr{R}}(L)$ can be proved immediately by analyzing the algorithm $\mathscr{A}_{1,m}$ (see Appendix A). Using the algorithm $\mathscr{A}_{1,m}$, the number of nonterminals of the grammar can grow in the part between the labels (L1) and (L2) only. The number of nonterminals increases to the value at most twice greater than the original one.

Let L_n be the language generated by the grammar $\bar{G}_n$:

$$S \to a_1A_1b_1 \mid \cdots \mid a_nA_nb_n \mid A_1c_1 \mid \cdots \mid A_nc_n,$$

$$A_i \to a_iA_ib_i \mid A_ic_i \mid a_ib_i \mid c_i \quad \text{for } 1 \leq i \leq n.$$

We shall prove that $\mathrm{Var}_{\mathscr{R}}(L_n) = n+1$ and $\mathrm{Var}_{\mathscr{G}_{ii'}}(L_n) = 2n+1$, i.e., for any $\delta > 0$ and for $n > 1/\delta - 1$ the language L_n satisfies the inequality $\mathrm{Var}_{\mathscr{G}_{ii'}}(L_n) > (2-\delta)\,\mathrm{Var}_{\mathscr{R}}(L_n)$.

Let G_n be a grammar generating the language L_n. According to Lemma 2.3, G_n contains nonterminals A_i such that $A_i \Rightarrow^* \alpha A_i \beta$ and $c_i \in \mathrm{alph}(\alpha\beta)$. Since for any $u_0 \in L$ there is a fixed number i_0 for which

$$\{i\colon a_i \in \mathrm{alph}(u_0),\ b_i \in \mathrm{alph}(u_0),\ c_i \in \mathrm{alph}(u_0)\} = \{i_0\},$$

$A_i \neq A_j$ for $i \neq j$ and every A_i differs from the initial nonterminal of G_n. So for G_n generating L_n, $\mathrm{Var}(G_n) \geq n+1$. Because $\mathrm{Var}(\bar{G}_n) = n+1$ we have $\mathrm{Var}_{\mathscr{R}}(L_n) = n+1$.

For $\tilde{G}_n = \mathscr{A}_{1,m}(\bar{G}_n)$ being the grammar in weak Greibach normal form, we have $\mathrm{Var}_{\mathscr{G}_{ii'}}(L_n) \leq \mathrm{Var}(\tilde{G}_n) = 2n+1$.

Let G'_n be a grammar in weak Greibach normal form generating the language L_n. For $u \in L_n$, $u = u_1u_2$ and $u_1 \in a_i^*$, $u_2 \in (b_i \cup c_i)^*$, the number of occurrences of a in u_1 equals the number of occurrences of b_i in u_2. According to Lemma 2.3, G'_n contains nonterminals A_i and B_i iterating the letters a_i and c_i, respectively, i.e., $A_i \Rightarrow^* x\alpha A_i\beta$, where $x \in T$, $a_i \in \mathrm{alph}(x\alpha\beta)$ and $B_i \Rightarrow^* y\gamma B_i\delta$, where $y \in T$ and $c_i \in \mathrm{alph}(y\gamma\delta)$.

From the structure of the words in L_n we can see that $x = a_i$ and $L(\alpha) \subseteq a_i^*$. Since $c_ic_i^* \subseteq L_n$ we also have $y = c_i$. $A_i \neq A_j$ for $i \neq j$ since for u in L_n, $\{i\colon c_i \in \mathrm{alph}(u)$ or $a_i \in \mathrm{alph}(u)\} = \{i_0\}$; $A_i \neq B_i$ for $i = 1, 2, \ldots, n$ because all occurrences of a_i in $u \in L_n$ precede the occurrences of c_i's and $S \neq A_i$ and $S \neq B_i$ for $i = 1, 2, \ldots, n$.

We have proved that $\mathrm{Var}(G_n) \geq 2n+1$ for any G'_n in weak Greibach normal form generating L_n. So we conclude that $\mathrm{Var}_{\mathscr{G}_{ii'}}(L_n) = 2n+1$. □

Theorem 3.3. *For ε-free languages,*

$$\mathrm{Prod}_{\mathscr{G}_{ii'}}(\mathscr{S}_{\mathrm{Prod}_{\mathscr{R}}}(n)) = O(n^3) \quad \textit{and} \quad \mathrm{Prod}_{\mathscr{G}_{ii'}}(\mathscr{S}_{\mathrm{Prod}_{\mathscr{R}}}(n)) = \Omega(n^2).$$

Proof. Let G be a $\mathrm{Prod}_{\mathscr{R}}$-minimal grammar generating a language L and let $A = AR + b$ be its matrix expression. Assume that the number of words in the components

of R (i.e., R-words) is r and similarily the number of b-words is b. Then $\mathrm{Prod}_{\mathscr{R}}(L) = r+b = n$.

The grammar $\mathscr{A}_2(G)$ is defined by the equations

$$A = bH + b, \tag{3.1}$$

$$H = R_0H + R_0, \tag{3.2}$$

where R_0 is a matrix constructed from R by replacing the first nonterminals in R-words by corresponding words specified by (3.1).

The number of productions determined by (3.1) equals $mb+b$ for m being the total number of nonterminals in G.

Let $r_1 \leq r$ be a number of those R-words, which have nonterminals as their first symbol. Since for each nonterminal the number of A-words is at most $2b$, the number of R_0-words is at most $2br_1 + r - r_1$ and the upper bound for the number of R_0H-words is $m(2b+r-r_1)$. Thus,

$$\mathrm{Prod}_{\mathscr{G}_1}(L) \leq \mathrm{Prod}(\mathscr{A}_2(G)) \leq (m+1)(2br+r-r_1+b).$$

Since for a $\mathrm{Prod}_{\mathscr{R}}$-minimal grammar $m \leq n/2$ and since $2br+r-r_1+b \leq n^2$, there is a constant c such that $\mathrm{Prod}_{\mathscr{G}_1}(L) \leq cn^3$ for all $L \in \mathscr{L}^{(n)}_{\mathrm{Prod}}$, i.e., $\mathrm{Prod}_{\mathscr{G}_1}(\mathscr{L}_{\mathrm{Prod}_{\mathscr{R}}}(n)) = O(n^3)$.

It remains to prove that $\mathrm{Prod}_{\mathscr{G}_1}(\mathscr{L}_{\mathrm{Prod}_{\mathscr{R}}}(n)) = \Omega(n^2)$, i.e., $\sup\{\mathrm{Prod}_{\mathscr{G}_1}(L)$: $\mathrm{Prod}_{\mathscr{R}}(L) \leq n$, L is context-free$\} \geq cn^2$ for some real number c. It is enough to prove for $L_n = \{(b_1 \cup \cdots \cup b_n))^{2^i}: 1 \leq i \leq n\}$ and $n > 1$ that $\mathrm{Prod}_{\mathscr{R}}(L_n) \leq 2n$ and $\mathrm{Prod}_{\mathscr{G}_1}(L_n) \geq n^2$.

The grammar with production rules

$$S \to A^2 | \cdots | A^{2^n}, \qquad A \to b_1 | \cdots | b_n,$$

generates the language L_n, i.e., $\mathrm{Prod}_{\mathscr{R}}(L_n) \leq 2n$.

Let $G'_n = (N, T, P, S)$ be a $\mathrm{Prod}_{\mathscr{G}_1}$-minimal grammar for the language L_n. Let $P_i = \{A \to b_i\omega: \omega \in (\{b_i\} \cup N)^*\}$ and let for i_0, $|P_{i_0}| \leq |P_i|$, $i = 1, 2, \ldots, n$. The grammar $G''_n = (N, T, P, S)$ generates the language $L(G''_n) = \{b_{i_0}^{2^i}: 1 \leq i \leq n\}$. According to [5, Theorem 6.3], $\mathrm{Prod}_{\mathscr{G}_1}(L(G''_n)) = n$. Evidently, the P_i's are pairwise disjoint for $i = 1, 2, \ldots, n$ and so $\mathrm{Prod}_{\mathscr{G}_1}(L_n) = \mathrm{Prod}(G'_n) \geq n\ \mathrm{Prod}_{\mathscr{G}_1}(L(G''_n)) = n^2$. □

The results for the measure Symb given in Theorem 3.4 were previously discussed in [11] and they are also included in [8].

Theorem 3.4. *For ε-free languages,*

$$\mathrm{Symb}_{\mathscr{G}_1}(\mathscr{L}_{\mathrm{Symb}_{\mathscr{R}}}(n)) = O(n^3), \qquad \mathrm{Symb}_{\mathscr{G}_1}(\mathscr{L}_{\mathrm{Symb}_{\mathscr{R}}}(n)) = \Omega(n^2).$$

Proof. The proof proceeds in the same way as for the measure Prod (see [8, pp. 129–131] or [11, pp. 348–349]). □

Theorem 3.5. $\mathrm{Lev}_{\mathscr{G}_1}(\mathscr{L}_{\mathrm{Lev}_{\mathscr{R}}}(n))$ *cannot be bounded by any total function $\varphi(n)$.*

Proof. According to Lemma 2.1 it is sufficient to prove Lemma 3.6. □

Lemma 3.6. *For any natural number n there is a language L_n such that*

$$\mathrm{Lev}_{\mathscr{R}}(L_n)=1 \quad \textit{and} \quad \mathrm{Lev}_{\mathscr{G}'}(L_n)=n+1.$$

Proof. For $n=1$, $L_1=ab^*$ satisfies the conditions of Lemma 3.6.

For $n>1$, let L_n be a language generated by the grammar G_n with starting nonterminal A_1 and with production rules

$$A_i \to a_iA_ib_i \mid a_iA_{i+1}b_i \mid A_ic_i, \quad 1 \leq i \leq n-1,$$

$$A_n \to a_nA_nb_n \mid a_nA_1b_n \mid A_nc_n \mid a_nb_n.$$

Obviously, $\mathrm{Lev}(G_n)=\mathrm{Lev}_{\mathscr{R}}(L_n)=1$.

$\mathrm{Lev}_{\mathscr{G}'}(L_n) \leq n+1$ since a grammar $\hat{G}_n=\mathscr{A}_{1,m}(G_n)$ has $n+1$ grammatical levels.

Let G'_n be a grammar in $\mathscr{G}'$ and $L(G'_n)=L_n$. According to Lemma 2.3, for any c_i there is a Z_i in G'_n, such that $Z_i \Rightarrow^* aZ_i\beta$, $c_i \in \mathrm{alph}(\alpha\beta)$. The number of occurrences of c_i in a word in L_n does not depend on the number of occurrences of the other symbols, i.e., $\alpha\beta$ can be chosen in such a manner that $\mathrm{alph}(\alpha\beta)=\{c_i\}$, i.e., $Z_i \Rightarrow^* c_i^{k_i}Z_ic_i^{l_i}$, $k_i \geq 1$ and $l_i \geq 0$. $Z_i \neq Z_j$ for $i \neq j$ since if $ac_j\beta \in L_n$ then $ac_i\beta \notin L_n$.

An initial nonterminal S differs from all Z_i because c_i is prefix of no word in L.

To get $\mathrm{Lev}_{\mathscr{G}'}(L_n)=n+1$ it remains to prove that $Z_1, \ldots, Z_n$ and S are elements of different grammatical levels. Following Lemma 2.2, it is enough to prove that $\mathrm{alph}(L(Z_i))=\{c_i\}$.

From the structure of the language L_n it follows that:

(a) if $uc_i^tv \in L_n$ for $t \geq 1$, then $u \in \Sigma^*(c_i \cup b_i)$, $v \in (c_i \cup b_i \cup b_{i-1})\Sigma^*$ for $i \geq 2$ and $v \in \{\varepsilon\} \cup (c_1 \cup b_1 \cup b_n)\Sigma^*$ for $i=1$;

(b) if $uc_iv \in L_n$, then $v \notin \Sigma^*(a_1 \cup \cdots \cup a_n)\Sigma^*$, i.e., all letters a_i precede the letters c_i in the word in L_n;

(c) the number and positions of b_i's in a word of language L_n uniquely determine the number and position of letters a_i in the word.

According to (a), $\mathrm{alph}(L(Z_i)) \subseteq \{c_i, b_i, b_{i-1}\}$ but since (c) and (b) must be also satisfied $\mathrm{alph}(L(Z_i))=\{c_i\}$. □

4. Complexity of position restricted grammars

Theorem 4.1. *For K being* Var, Prod, Lev *or* Hei, *and t being any position restricted type, $K_t(\mathscr{S}_{K,t}(n))$ cannot be bounded by any function φ.*

Proof. The theorem will be proved by proving the following lemma.

Lemma 4.2. *Let K be* Var, Prod, Lev *or* Hei. *Let $\mathscr{t}$ be a class of position restricted grammars of type $t=(m_1, \ldots, m_k, m_{k+1})$.*

For any natural number n there is a language L_n such that $K_{\mathscr{R}}(L_n)=1$ and $K_{\mathscr{L}}(L_n)=n$.

Proof. For the language $L_n=\{a^{f(n)}\}$, where $f(n)=k^n+\sum_{j=1}^{n-1} k^j \sum_{j=1}^{k+1} m_j$, evidently $\mathrm{Var}_{\mathscr{R}}(L_n)=\mathrm{Prod}_{\mathscr{R}}(L_n)=1$.

Let G_n be a position restricted grammar of type t and let $L(G_n)=L_n$. The language L_n is finite so all nodes of any branch of a derivation tree of the word $a^{f(n)}$ in G_n are denoted by distinct nonterminals. A position restricted grammar of type t has at most k nonterminals in the right-hand side of production rules and moreover $\sum_{i=1}^{k+1} m_i$ is the total length of terminal words in it, so the derivation tree of the word $a^{f(n)}$ in G_n has a branch of length $l \geqslant n$, i.e., $\mathrm{Var}(G_n) \geqslant n$ and $\mathrm{Prod}(G_n) \geqslant n$ for any $G_n \in \mathscr{L}$.

Let G_n be a $\mathrm{Lev}_{\mathscr{L}}$-minimal grammar for the language L_n. Since L_n is finite, G_n contains single-letter levels only, i.e., $\mathrm{Lev}(G_n)=\mathrm{Var}(G_n) \geqslant n$.

Finally, $\mathrm{Hei}_{\mathscr{L}}(L_n) \geqslant n$ since G_n consists of trivial levels only (i.e., single-letter levels) and the derivation tree of the word $a^{f(n)}$ in G_n has a branch of length at least n.

Moreover, $K_{\mathscr{L}}(L_n)=n$ for K being Var, Prod, Lev and Hei since for a grammar G_n with the production rules

$$A_i \to a^{m_1} A_{i+1} a^{m_2} A_{i+1} \cdots a^{m_k} A_{i+1} a^{m_{k+1}}, \quad 1 \leqslant i \leqslant n-1,$$

$$A_n \to a,$$

we have $G_n \in \mathscr{L}$, $L(G_n)=L_n$ and $\mathrm{Var}(G_n)=\mathrm{Prod}(G_n)=\mathrm{Lev}(G_n)=\mathrm{Hei}(G_n)=n$. □

The relationships between $\mathrm{Symb}_{\mathscr{L}}$ and $\mathrm{Symb}_{\mathscr{R}}$ depend on the grammatical type t. We shall give a result only for the case $t=(0,0,0)$.

Theorem 4.3. *We have*

$$\mathrm{Symb}_{(0,0,0)}(L) \leqslant 4\,\mathrm{Symb}_{\mathscr{R}}(L)-9.$$

Also, there is a sequence of languages $\{L_n\}_{n=1}^{\infty}$ such that $\mathrm{Symb}_{(0,0,0)}(L_n)=4\,\mathrm{Symb}_{\mathscr{R}}(L_n)-9$.

Proof. Let G be a $\mathrm{Symb}_{\mathscr{R}}$-minimal grammar for a language L (i.e., G and $\mathrm{Symb}(G)=\mathrm{Symb}_{\mathscr{R}}(L)$). Let $\mathscr{A}_3$ be the algorithm, described in Appendix A, which transforms a completely reduced grammar to a grammar of type $(0,0,0)$. Suppose G has s productions (i.e., $P=\{p_i=A_i \to \alpha_i : i=1,\ldots,s\}$) and, moreover, $\{j: |\alpha_j| \geqslant 2\}=\{j: 1 \leqslant j \leqslant r,\ r \leqslant s\}$. Then

$$\mathrm{Symb}_{\mathscr{R}}(L)=\mathrm{Symb}(G)=\sum_{i=1}^{r}(|\alpha_i|+2)+3(s-r)=3s-r+\sum_{i=1}^{r}|\alpha_i|,$$

$$\mathrm{Symb}_{(0,0,0)}(L) \leqslant \mathrm{Symb}(\mathscr{A}_3(G))=\sum_{i=1}^{r} 4(|\alpha_i|-1)+3k+3(s-r),$$

where k is the number of terminals, which occur as the last symbol on the right-hand sides of production rules $p_1, \ldots, p_r$, i.e., $k \leq \min(r, |\text{alph}(L)|)$.

$$\text{Symb}_{(0,0,0)}(L) \leq 4 \sum_{i=1}^{r} |\alpha_i| - 4r + 3 \min(r, |\text{alph}(L)|) + 3(s-r)$$
$$\leq 4 \text{Symb}_{\mathscr{R}}(L) - 9s \leq 4 \text{Symb}_{\mathscr{R}}(L) - 9.$$

We shall prove the second part of the theorem for $L_n = \{a_1 a_2 \cdots a_n\}$, $n \geq 1$. Clearly, $\text{Symb}_{\mathscr{R}}(L_n) = n + 2$.

In a grammar of type $t = (0, 0, 0)$ generating L_n, all vertices of a derivation tree for $a_1 \cdots a_n$ are labelled by different nonterminals and terminals. To get n leaves of a binary tree we need at least $n-1$ additional vertices (except of leaves) and for a derivation tree of a grammar of type $(0, 0, 0)$ we need at least one more vertex to get all leaves labelled by terminals. Therefore,

$$\text{Symb}_{(0,0,0)}(L_n) \geq 4(n-1) + 3 = 4 \text{Symb}_{\mathscr{R}}(L_n) - 9.$$

For a grammar G_n with the productions

$$A_i \to a_i A_{i+1}, \quad i = 1, 2, \ldots, n-1, \qquad A_n \to a_n,$$

$L(G_n) = \{a_1 \cdots a_n\}$, G_n is in $(0, 0, 0)$ and $\text{Symb}(G_n) = 4n - 1 = 4 \text{Symb}_{\mathscr{R}}(L_n) - 9$. □

Now we shall give some remarks concerning Chomsky normal form, since it is stronger than position restricted type $(0, 0, 0)$ in a sense that production rules of a form $A \to aB$ are not allowed in Chomsky normal form.

Remark 4.4. In Theorem 4.1, position restricted grammars of type t can be replaced by grammars in Chomsky normal form. The proof of this new theorem can be done exactly in the same way as that of Theorem 4.1 (i.e., using the same language as for grammars of type $(0, 0, 0)$).

Remark 4.5. If $\mathscr{CH}$ is the class of grammars in Chomsky normal form, then $\text{Symb}_{\mathscr{CH}}(L) \leq 7 \text{Symb}_{\mathscr{R}}(L) - 15$.

There is a sequence of languages $\{L_n\}_{n=1}^{\infty}$ such that $\text{Symb}_{\mathscr{CH}}(L_n) = 7 \text{Symb}_{\mathscr{R}}(L_n) - 15$.

The proof can be done in the same way as the proof of Theorem 4.3. If r, s and $A_i \to \alpha_i$ have the same meaning as in the proof of Theorem 4.3 we have

$$\text{Symb}_{\mathscr{R}}(L) = 3s - r + \sum_{i=1}^{r} |\alpha_i|.$$

Let G be a $\text{Symb}_{\mathscr{R}}$-minimal grammar for L and let algorithm $\mathscr{A}_4$ be that from Appendix A. We have

$$\text{Symb}_{\mathscr{CH}}(L) \leq \text{Symb}(\mathscr{A}_4(G)) \leq 3(s-r) + 4 \sum_{i=1}^{r} (|\alpha_i| - 1) + 3|\text{alph}(L)|.$$

Since $s \geq 1$, $r \geq 0$ and $|\text{alph}(L)| \leq \text{Symb}_{\mathscr{R}}(L) - 2$, we now have $\text{Symb}_{\mathscr{CH}}(L) \leq 7\,\text{Symb}_{\mathscr{R}}(L) - 15$.

The optimality of the result can be proved in the same way as in Theorem 4.3. For $L_n = \{a_1 \cdots a_n\}$ one can get $\text{Symb}_{\mathscr{R}}(L_n) = n + 2$ and $\text{Symb}_{\mathscr{CH}}(L_n) = 7n - 1$.

Open problem. Let $\mathscr{L}_k$ be the class of languages such that, for $L \in \mathscr{L}_k$, $|\text{alph}(L)| = k$. Then by analyzing algorithm $\mathscr{A}_4$ for $L \in \mathscr{L}_k$ one can get $\text{Symb}_{\mathscr{CH}}(L) \leq 4\,\text{Symb}_{\mathscr{R}}(L) - 3k - 9$.

Improve the coefficient 4 in the estimation or prove its optimality.

Remark 4.6. In connection with the coefficient 4 we shall prove in Example 4.7 the optimality of that coefficient for the case of the sequence $\{L_n\}_{n=1}^{\infty}$ such that $\lim_{n \to \infty} |\text{alph}(L_n)|/\text{Symb}_{\mathscr{R}}(L_n) = 0$.

Example 4.7. Let $\Sigma_n = \{a_0, \ldots, a_{n-1}\}$ be the alphabet of L_n. Let

$$g(i, j) = (\tfrac{1}{2}i(i+1) + j) \pmod{n}$$

and

$$L_n = \{a_{g(0,i)} a_{g(1,i)} \cdots a_{g(n-1,i)} : 0 \leq i \leq n-1\}.$$

The optimality will be proved in the sense that for any $\delta > 0$ there is an n_0 such that for all $n \geq n_0$,

$$\text{Symb}_{\mathscr{CH}}(L_n) > (4 - \delta)\,\text{Symb}_{\mathscr{R}}(L_n). \tag{4.1}$$

We shall prove $\text{Symb}_{\mathscr{R}}(L_n) = n^2 + 2n$ and $\text{Symb}_{\mathscr{CH}}(L_n) = 4n^2 - n$, i.e., inequalitity (4.1) holds for $n > (4 - 2\delta)/\delta$.

For a language L_n the following property can be proved immediately:

(P) All subwords of L_n of length at least 2 differ from each other.

We shall prove that a grammar G_n with productions

$$S \to a_{g(0,j)} \cdots a_{g(n-1,j)}, \quad 0 \leq j \leq n-1,$$

is a $\text{Symb}_{\mathscr{R}}$-minimal grammar for L_n and therefore $\text{Symb}_{\mathscr{R}}(L_n) = n(n+2)$.

Suppose that G'_n contains a nonterminal $A \neq S$ such that either $L(A) = \{u\}$ and A occurs at least twice on the right-hand side of rules in G'_n or $\{u_1, u_2\} \subseteq L(A)$ for $u_1 \neq u_2$.

If $L(A) = \{u\}$, then $|u| = 1$ because of property (P) and it leads to a contradiction with $\text{Symb}_{\mathscr{R}}$-minimality of G'_n.

For $\{u_1, u_2\} \subseteq L(A)$ let us consider the derivations

$$S \Rightarrow^+ uAv \Rightarrow^+ uu_1v \quad \text{and} \quad S \Rightarrow^+ uAv \Rightarrow^+ uu_2v.$$

Since no two words in L_n have same prefix or suffix we have $u = v = \varepsilon$. So $S \Rightarrow^+ A$ in G'_n and this is a contradiction since G'_n is completely reduced.

Let G'_n be a $\text{Symb}_{\mathscr{CH}}$-minimal grammar for L_n. Because of property (P) and the finiteness of L_n a derivation tree of the word $a_{g(0,i)} \cdots a_{g(n-1,i)}$ in G'_n has $2n - 1$

vertices labelled by different nonterminals; $n-1$ of them are left-hand sides of production rules of length 4 and the last n of them are on left-hand sides of production of type $A \to a$. According to property (P), the sets of internal nonterminals used in derivations of different words from L_n are disjoint. (X is an internal nonterminal if $X \to AB$ for some A and B.) Therefore, $\text{Symb}_{\mathscr{A}}(L_n) \geq 4(n-1)n+3n$. Since for $\mathscr{A}_4(G_n)$ we have $\text{Symb}(\mathscr{A}_4(G_n) = 4n^2 - n$, and we have completed our proof.

Appendix A

We shall give the algorithms used in previous parts of the paper. For the transformation of a completely reduced grammar to an equivalent one in Greibach normal form we shall give two algorithms. The classical sequential one will be denoted by $\mathscr{A}_1$ and the matrix algorithm denoted by $\mathscr{A}_2$. For the transformation of a completely reduced grammar to an equivalent position restricted grammar of type (0, 0, 0) we shall give algorithm $\mathscr{A}_3$, and the algorithm $\mathscr{A}_4$ will transform completely reduced grammars to equivalent grammars in Chomsky normal form.

Let us assume for all algorithms that $G = (N, T, P, S)$ is a completely reduced grammar, $N = \{S = A_1, A_2, \ldots, A_m\}$, $T = \{t_1, t_2, \ldots, t_l\}$ and $P = \{p_1, p_2, \ldots, p_s\}$.

Algorithm $\mathscr{A}_1$ works in three steps. Firstly, it modifies the grammar G so that if $A_i \to A_j\alpha$ is its production rule then $j > i$ (that part of algorithm is between labels (L1) and (L2)); then the algorithm produces a grammar in weak Greibach normal form (just before label (L3)) and in the last part of algorithm (beginning with (L3)) terminals on the right-hand side of productions of grammar which are not on the first position are changed to nonterminals to obtain a grammar in Greibach normal form.

We shall fix the following notations for $\mathscr{A}_1$. For $i = 1, 2, \ldots, m$, we denote $P_i = \{A_i \to \alpha : A_i \to \alpha \in P\}$. For a grammar with no left recursion of nonterminals

$$P_{i,j} = \{A_i \to \alpha\beta : A_i \to A_j\beta \in P,\ A_j \to \alpha \in P\}$$

and

$$\bar{P}_{i,j} = \{A_i \to A_j\alpha : A_i \to A_j\alpha \in P\}$$

for $i, j = 1, 2, \ldots, m$.

Algorithm $\mathscr{A}_1$

procedure GREIBACH NORMAL FORM (P, P')
(algorithm starts with the set of productions P)
(L1) **begin** $i := 1$;

while $i \leq m$ **do** $j := 1$;

while $j \leq i-1$ **do** $P := (P \cup P_{i,j}) - \bar{P}_{i,j}$;

$j := j+1$; **end**

$R := P_i$;

$P := P - P_i \cup \{A_i \to \beta \mid \beta Z_i \colon A_i \to \beta \in R, \beta \notin A_i(N \cup T)^*\}$

$\cup \{Z_i \to \alpha \mid \alpha Z_i \colon A_i \to A_i\alpha \in R\}$;

$i := i+1$; **end**

(L2) $i := m-1$;

while $i > 0$ **do** $P := P \cup \{P_{i,j} \colon i < j \leq m\} - \{\bar{P}_{i,j} \colon i < j \leq m\}$;

$i := i-1$; **end**

$i := 1$;

while $i \leq m$ **do** $P := P \cup \{Z_i \to \beta\alpha \colon Z_i \to A_j\alpha \in P, A_j \to \beta \in P\}$

$$- \left\{ Z_i \to \alpha \colon \alpha \in N \left(T \cup N \cup \bigcup_{i=1}^{m} Z_i \right)^* \right\};$$

$i := i+1$; **end**

(L3) $P' := \{T_j \to t_j \colon 1 \leq j \leq 1\}$

$\cup \{X \to tY_1 Y_2 \cdots Y_r \colon$

$X \to tX_1 X_2 \cdots X_r \in P$, where $Y_i = X_i$ for $X_i \in \{A_i, Z_i \colon 1 \leq i \leq m\}$
and $Y_i = T_j$ for $X_i = t_j\}$;

end

(The algorithm produces the set of production rules in P')

We denote by $\mathcal{A}_{1,m}$ the initial part of the algorithm $\mathcal{A}_1$ ended just before the command labelled by (L3) with the set of productions in P.

Proposition A.1. *Let $G = (N, T, P, S)$ be a grammar in completely reduced form. Let $\mathcal{A}_1(G) = G'$ and $\mathcal{A}_{1,m}(G) = G''$.*

Then $G' \in \mathcal{G}_{\mathfrak{k}}$, $G'' \in \mathcal{G}_{\mathfrak{k}'}$ and $L(G) = L(G') = L(G'')$.

Proof. The proof can be found in [8, p. 113] or [1, pp. 156–158]. □

Algorithm $\mathcal{A}_2$

procedure GREIBACH NORMAL FORM-MATRIX (P, P')
(the algorithm starts with the set of productions in P)

begin express G by equations of form $A = AR + b$;

construct a new system of equation $A = bH + b$

$H = RH + R$;

(where H be an $m \times m$ matrix of new nonterminals)

P' := "set of productions corresponding to $A = bH + b$";

P := "set of productions corresponding to $H = RH + R$";

$P' := P' \cup \{H_{ij} \to \alpha_1 \cdots \alpha_s \in P: \alpha_1 \in T\}$

$\cup \{H_{ij} \to \beta\alpha_2 \cdots \alpha_s: \alpha_1 \in N, \alpha_1 \to \beta \in P', H_{ij} \to \alpha_1\alpha_2 \cdots \alpha_s \in P\}$;

end

(The algorithm produces set of production rules in P')

Proposition A.2. *Let $G = (N, T, P, S)$ be a completely reduced grammar and $G' = \mathscr{A}_2(G)$. Then $G' \in \mathscr{G}'$ and $L(G) = L(G')$.*

Proof. For the proof, see, e.g., [1, p. 162].

Algorithm $\mathscr{A}_3$

procedure POSITIONAL RESTRICTED-TYPE $(0, 0, 0)$ (P, P')
(the algorithm starts with the set of productions in P)

begin $P' := \emptyset$;

$i := 1$;

while $i \leq s$ **do if** p_i has the form $A \to a$ **then** $P' := P' \cup \{p_i\}$ **else** p_i has a form $A \to X_1X_2 \cdots X_n$, $n \geq 2$,

if $X_n \in T$, $X_n = a_r$ **then** $P' := \{X'_r \to a_r\} \cup P'$
$\cup \{A \to X_1A_1, A_1 \to X_1A_2, \ldots, A_{n-2} \to X_{n-1}X'_r\}$;
else $P' := P' \cup \{A \to X_1A_1, A_1 \to X_2A_2, \ldots, A_{n-2} \to X_{n-1}X_n\}$;

$i := i + 1$;

end

(The algorithm produces a set of production rules in P')

Remark. The A_i's in the algorithm $\mathscr{A}_3$ are new nonterminals which differ with each other for all productions p_i.

Proposition A.3. *Let G be a completely reduced grammar. Let $G' = \mathscr{A}_3(G)$. Then G' is a position restricted grammar of type $(0, 0, 0)$ and $L(G) = L(G')$.*

Proof. The proof is obvious. It can be done in the same way as that for Algorithm $\mathcal{A}_4$. □

Algorithm $\mathcal{A}_4$

procedure CHOMSKY NORMAL FORM (P, P')
(algorithm starts with the set of productions P)

begin $P' := \emptyset$;

$i := 1$;

while $i \leq s$ **do if** p_i has a form $A \to a$ **then** $P' := P' \cup \{p_i\}$ **else**

p_i has a form $A \to X_1 X_2 \cdots X_n$, $n \geq 2$

$P' := P' \cup \{A \to X_1' A_1, A_1 \to X_2' A_2, \ldots, A_{n-2} \to X_{n-1}' X_n'\}$,

where $X_i' = X_i$ if $X_i \in N$ and $X_i' = T_r$ if $X_i = t_r$;

$i := i + 1$;

$P' := P' \cup \{T_r \to t_r : 1 \leq k \leq l\}$

end

(The algorithm produces the set of production rules in P')

Remark. The A_i's in the algorithm $\mathcal{A}_4$ are new nonterminals which differ with each other for all productions p_i.

Proposition A.4. *Let $G = (N, T, P, S)$ be a completely reduced grammar and let $G' = \mathcal{A}_4(G)$. Then G' is in Chomsky normal form and $L(G) = L(G')$.*

Proof. For the proof, see, e.g., [8, p. 104]. □

References

[1] A.V. Aho and J.D. Ullman, *The Theory of Parsing, Translation and Compiling, Vol. 1* (Prentice-Hall, Englewood Cliffs, NJ, 1972).

[2] B.S. Baker, Non-context-free grammars generating context-free languages, *Inform. and Control* **24** (1974) 231–246.

[3] M. Blattner and S. Ginsburg, *Canonical Forms of Context-free Grammars and Position Restricted Grammar Forms*, Lecture Notes in Computer Science **56** (Springer, Berlin, 1977) pp. 49–55.

[4] M. Blattner and S. Ginsburg, Position restricted grammar forms and grammars, *Theoret. Comput. Sci.* **17** (1982) 1–27.

[5] J. Gruska, Some classifications of context-free languages, *Inform. and Control* **14** (1969) 152–179.

[6] J. Gruska, A note on ε-rules in context-free grammars, *Kybernetika* **11** (1975) 26–31.

[7] J. Gruska, *Descriptional Complexity (of Languages). A Short Survey*, Lecture Notes in Computer Science **45** (Springer, Berlin, 1976) pp. 65–80.

[8] M.A. Harrison, *Introduction to Formal Language Theory* (Addison-Wesley, Reading, MA, 1978).

[9] D.E. Knuth, Big omicron and big omega and big theta, *SIGACT News* **8** (1976) 18–24.

[10] H.A. Maurer, A. Salomaa and D. Wood, A super normal form theorem for context-free grammars, *J. ACM* **30** (1983) 95–102.

[11] A. Pirická-Kelemenová, *Greibach Normal Form Complexity*, Lecture Notes in Computer Science **32** (Springer, Berlin, 1975) pp. 344–350.

[12] A. Salomaa, *Jewels of Formal Language Theory* (Computer Science Press, Rockville, MD, 1981).