Language Classes Generated by Tree Controlled Grammars with Bounded Nonterminal Complexity

Sherzod Turaev[a,1], Jürgen Dassow[b], Florin Manea[c,2], Mohd Hasan Selamat[a,1]

[a] *Faculty of Computer Science and Information Technology, University Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia*
[b] *Otto-von-Guericke-Universität Magdeburg, Fakultät für Informatik PSF 4120, D-39016 Magdeburg, Germany*
[c] *Institut für Informatik, Christian-Albrechts-Universität zu Kiel, Christian-Albrechts-Platz 4, D-24098 Kiel, Germany*

Abstract

A tree controlled grammar is specified as a pair (G, G') where G is a context-free grammar and G' is a regular grammar. Its language consists of all terminal words with a derivation in G such that all levels of the corresponding derivation tree – except the last level – belong to $L(G')$. We define the nonterminal complexity $\mathrm{Var}(H)$ of $H = (G, G')$ as the sum of the numbers of nonterminals of G and G'. In [23] it is shown that tree controlled grammars H with $\mathrm{Var}(H) \leq 9$ are sufficient to generate all recursively enumerable languages. In this paper, we improve the bound to seven. Moreover, we show that all linear and regular simple matrix languages can be generated by tree controlled grammars with a nonterminal complexity bounded by three, and we prove that this bound is optimal for the mentioned language families. Furthermore, we show that any context-free language can be generated by a tree controlled grammar (G, G') where the number of nonterminals of G and G' is at most four.

Keywords: tree controlled grammars, nonterminal complexity, bounds for linear, context-free, and regular simple matrix languages

Email addresses: sherzod@fsktm.upm.edu.my (Sherzod Turaev), dassow@iws.cs.uni-magdeburg.de (Jürgen Dassow), flmanea@informatik.uni-kiel.de (Florin Manea), hasan@fsktm.upm.edu.my (Mohd Hasan Selamat)

[1]The works of the first and fourth authors were supported by University Putra Malaysia via RUGS 05-01-10-0896RU.

[2]Florin Manea is on leave of absence from the University of Bucharest, Faculty of Mathematics and Computer Science. His work is supported by the *DFG* grant 582014. Part of this paper was writen when Florin Manea was a Research Fellow of the *Alexander von Humboldt Foundation* at the Otto-von-Guericke-Universität Magdeburg, Fakultät für Informatik, whose support is also acknowledged.

1. Introduction

Besides the efficiency of algorithms and devices for the acceptance of languages with respect to time and space a very important topic of theoretical computer science is the study of succinct descriptions of algorithms and languages. For instance, algorithms are described by programs whose size is measured by the number of commands (or lines of codes). If languages are described by (finite) automata, then the number of states is one of the possible measures of descriptional complexity; and the minimization of finite automata is very early result in the theory of automata. With respect to the generation of languages by (different types of) grammars, the number of nonterminals, or the number of productions, or the total number symbols in rules are well-known measures of size.

The study of the descriptional complexity with respect to regulated grammars started in [1, 4, 5, 6, 21]. In recent years several interesting results on this topic have been obtained. There are results which compare the conciseness of minimal descriptions of languages by different types of regulated grammars as well as statements that grammars with a bounded size suffice to generate all languages of certain language classes. For instance, the nonterminal complexity of programmed and matrix grammars is studied in [9], where it is shown that three nonterminals for programmed grammars with appearance checking, and four nonterminals for matrix grammars with appearance checking are enough to generate every recursively enumerable language. A more detailed investigation with respect to the appearance checking is given in [10]. There are several papers which present analogous results for scattered context grammars [2, 11, 12, 17, 24], semi-conditional grammars [18, 19, 21, 24], and multi-parallel grammars [16].

In this paper we study the nonterminal complexity of tree controlled grammars. A tree controlled grammar is specified as a pair (G, G') where G is a context-free grammar and G' is a regular grammar. Its language consists of all terminal words with a derivation in G such that all levels of the corresponding derivation tree – except the last level – belong to $L(G')$. We define the nonterminal complexity $\mathrm{Var}(H)$ of $H = (G, G')$ as the sum of the numbers of nonterminals of G and G'. In contrast to most of the papers cited above, we do not only take the number of nonterminals of G, we also add the number of nonterminals of G', i.e., we also measure the complexity of the control device (however, we note that, for the matrix, programmed and scattered context grammars, it is not clear how one can measure the complexity of the matrices and the success field and failure field in terms of nonterminals). In [23], it is shown that there is an infinite hierarchy with respect to the nonterminal complexity, if we consider tree controlled grammars with non-erasing rules only. It is worth to note that the proof uses regular languages. On the other side, the allowance of erasing rules leads to the result that every recursively enumerable language can be generated by a tree controlled grammar with not more than nine nonterminals in G and G'. In this paper we continue the research by showing that some known language classes can be generated by tree controlled grammars with three, four, or seven nonterminals.

The paper is organized as follows. In Section 2, we recall the necessary concepts and notations. In Section 3, we improve the bound for recursively enumerable languages from nine to seven. In Section 4, we show that all linear and regular simple matrix languages can be generated by tree controlled grammars with a nonterminal complexity bounded by three, and we prove that this bound is optimal for the mentioned language families. In Section 5, we show that tree controlled grammars with the nonterminal complexity bounded by four are sufficient to generate all context-free languages. Finally, we add some concluding remarks which summarize the results and mention some open problems and directions for further research.

2. Definitions

We assume that the reader is familiar with formal language theory (see [7, 22]).

Let T^* denote the set of all words over an alphabet T. The empty word is denoted by ε. The cardinality of a finite set X is denoted by $|X|$.

A (phrase structure) grammar is specified as a quadruple $G = (N, T, P, S)$ where N and T are the disjoint alphabets of nonterminals and terminals, respectively, P is a finite set of productions (of the form $\alpha \to \beta$, where $\alpha \in (N \cup T)^* N (N \cup T)^*$, and $S \in N$.

A grammar is called context-free if all rules have the form $A \longrightarrow w$ where $A \in N$ and $w \in (N \cup T)^*$).

A context-free grammar is called regular, if all production are of the form $A \longrightarrow wB$ or $A \longrightarrow w$ with $A, B \in N$ and $w \in T^*$.

A context-free grammar is called linear, if all production are of the form $A \longrightarrow wBv$ or $A \longrightarrow w$ with $A, B \in N$ and $w, v \in T^*$.

By $\mathcal{L}(REG)$, $\mathcal{L}(LIN)$, $\mathcal{L}(CF)$, and $\mathcal{L}(RE)$ we denote the families of all regular, linear, context-free, and recursively enumerable languages, respectively.

With each derivation in a context-free grammar G, one associates a derivation tree. The *level* associated with a node is the number of edges in the path from the root to the node. The *height* of the tree is the largest level number of any node. With a derivation tree t of height k and each number $0 \leq i \leq k$, we associate the *word of level* i which is given by all nodes of level i read from left to right, and we associate the *sentential form of level* i which consists of all nodes of level i and all leaves of level less than i read from left to right. Obviously, if u and v are sentential forms of two successive levels, then $u \Longrightarrow^* v$ holds and this derivation is obtained by a parallel replacement of all nonterminals occurring in the sentential form u.

In [13], it was shown that every recursively enumerable language is generated by a grammar

$$G = (\{S, A, B, C\}, T, P \cup \{ABC \to \varepsilon\}, S)$$

in the *Geffert normal form* where P contains only context-free rules of the form

$S \to uSa$ where $u \in \{A, AB\}^*$, $a \in T$,
$S \to uSv$ where $u \in \{A, AB\}^*$, $v \in \{BC, C\}^*$,
$S \to uv$ where $u \in \{A, AB\}^*$, $v \in \{BC, C\}^*$.

In addition, any terminal derivation in G is of the form

- $S \Longrightarrow^* w'Sw$ by productions of the form $S \to uSa$, where $w' \in \{A, AB\}^*$ and $w \in T^+$,
- $w'Sw \Longrightarrow^* w_1w_2w$ by productions of the form $S \to uSv$ and $S \to uv$, where $w_1 \in \{A, AB\}^*$ and $w_2 \in \{BC, C\}^*$, or
- $w_1w_2w \Longrightarrow^* w$ by $ABC \to \varepsilon$

In order to distinguish the phases in a terminal derivation, we use a new nonterminal and slightly modify the rules of the grammar. A grammar G is in the *modified Geffert normal form* if

$$G = (\{S, S', A, B, C\}, T, P \cup \{ABC \to \varepsilon\}, S)$$

where P contains only context-free rules of the form

(a) $S \to uSa$ where $u \in \{A, AB\}^*$, $a \in T$,

(b) $S \to S'$,

(c) $S' \to uS'v$ where $u \in \{A, AB\}^*$, $v \in \{BC, C\}^*$,

(d) $S' \to \varepsilon$.

In addition, any terminal derivation in G is of the form

(α) $S \Longrightarrow^* w'Sw \Longrightarrow w'S'w$ by productions of the form $S \to uSa$ and $S \to S'$, where $w' \in \{A, AB\}^*$ and $w \in T^*+$,

(β) $w'S'w \Longrightarrow^* w_1S'w_2w \Longrightarrow w_1w_2w$ by productions of the form $S' \to uS'v$ and $S' \to \varepsilon$, where $w_1 \in \{A, AB\}^*$ and $w_2 \in \{BC, C\}^*$,

(γ) $w_1w_2w \Longrightarrow^* w$ by $ABC \to \varepsilon$.

For the sake of completeness, we also recall definitions concerning regular simple matrix grammars and tree controlled grammars.

A *regular simple matrix grammar of degree* n, $n \geq 1$, is an $(n+3)$-tuple $G = (V_1, V_2, \ldots, V_n, T, M, S)$, where $V_1, V_2, \ldots, V_n$ are pairwise disjoint alphabets of nonterminals, T is an alphabet of terminals, S is a nonterminal which is not in $\bigcup_{i=1}^n V_i$, and M is a set of matrices of the following forms:

1. $(S \to x)$ with $x \in T^*$,
2. $(S \to A_1 A_2 \cdots A_n)$ with $A_i \in V_i$ for $1 \leq i \leq n$,
3. $(A_1 \to x_1 B_1, A_2 \to x_2 B_2, \ldots, A_n \to x_n B_n)$ with $A_i, B_i \in V_i$ and $x_i \in T^*$ for $1 \leq i \leq n$,
4. $(A_1 \to x_1, A_2 \to x_2, \ldots, A_n \to x_n)$ with $A_i \in V_i$ and $x_i \in T^*$ for $1 \leq i \leq n$.

We say that G is a *regular simple matrix grammar*, if it is a regular simple matrix grammar of some degree n.

A direct derivation step in a regular simple matrix grammar G is defined by

- $S \Longrightarrow z$ if and only if there is a matrix $(S \to z) \in M$,
- $z_1 A_1 z_2 A_2 \cdots z_n A_n \Longrightarrow z_1 x_1 B_1 z_2 x_2 B_2 \cdots z_n x_n B_n$ if and only if there exists a matrix $(A_1 \to x_1 B_1, \ldots, A_n \to x_n B_n) \in M$,
- $z_1 A_1 z_2 A_2 \cdots z_n A_n \Longrightarrow z_1 x_1 z_2 x_2 \cdots z_n x_n$ if and only if there exists a matrix $(A_1 \to x_1, A_2 \to x_2, \ldots, A_n \to x_n) \in M$.

The language $L(G)$ generated by a regular simple matrix grammar is defined as $L(G) = \{z \mid z \in T^*,\ S \Longrightarrow^* z\}$ where $\Longrightarrow^*$ is the reflexive and transitive closure of $\Longrightarrow$.

Simple matrix grammar and languages have been introduced by O. Ibarra in [15]. A summary of results on them can be found in Section 5.1 of [7].

Intuitively, a regular matrix grammar of degree n performs in parallel the derivations of n regular grammars. Moreover, in the corresponding derivation tree, the word of any level t is obtained by a concatenation of words of level t of the derivation trees from the regular grammars.

We now show that the rules of type 1 can be omitted without a decreasing of the generative power.

Lemma 1 *For any regular simple matrix grammar $G = (V_1, V_2, \ldots, V_n, T, M, S)$ there is a regular simple matrix grammar $G' = (V_1', V_2', \ldots, V_n', T, M', S)$ such that M' only contains matrices of the forms 2, 3, and 4 and $L(G') = L(G)$ holds.*

Proof. Let $G = (V_1, V_2, \ldots, V_n, T, M, S)$ be a regular simple matrix grammar. If M does not contain matrices of type 1, we choose $G' = G$. Otherwise, let M'' be the set of matrices of type 1. Furthermore, let $B_1, B_2, \ldots, B_n$ be new pairwise different nonterminals not contained in $V_1 \cup V_2 \cup \cdots \cup V_n$. Then we consider the regular simple matrix grammar

$$G' = (V_1 \cup \{B_1\}, V_2 \cup \{B_2\}, \ldots, V_n \cup \{B_n\}, T, (M \setminus M'') \cup Q, S)$$

where Q consists of all rules of the following forms

$(S \longrightarrow B_1 B_2 \ldots B_n)$
$(B_1 \longrightarrow x, B_2 \longrightarrow \varepsilon, B_3 \longrightarrow \varepsilon, \ldots, B_n \longrightarrow \varepsilon)$ with $(S \longrightarrow x) \in M''$.

Obviously, the application of $(S \longrightarrow x)$ in G is simulated by the application of $(S \longrightarrow B_1B_2\dots B_n)$ followed by an application of $(B_1 \longrightarrow x, B_2 \longrightarrow \varepsilon, \dots, B_n \longrightarrow \varepsilon)$. Therefore it is easy to see that $L(G) = L(G')$. Moreover, in the set $(M \setminus M'') \cup Q$ is no matrix of type 1. Thus G' satisfies all requirements. □

We mention the normal form given in Lemma 1 does not necessarily hold for regular simple matrix grammars without erasing rules since the construction in the proof of Lemma 1 introduces erasing rules and the elimination of erasing rules (see Theorem 1.5.3 and Lemma 1.5.7 in [7]) introduces rules of form (1).

By $\mathcal{L}(RSM)$ we denote the family of all languages generated by regular simple matrix grammars.

A *tree controlled grammar* is a quintuple $H = (N, T, P, S, R)$ where $G = (N, T, P, S)$ is a context-free grammar and $R \subseteq (N \cup T)^*$ is a regular set. The language $L(H)$ consists of all words w generated by the underlying grammar G such that there is a derivation tree t of w with respect to G, where the words of all levels (except the last one) are in R.

Since $R = L(G')$ for some regular grammar $G' = (N', T', P', S')$, a tree controlled grammar H can be given as a pair $H = (G, G')$.

For a context-free grammar $G = (N, T, P, S)$, by $\mathrm{Var}(G)$, we denote the number of the nonterminals of a grammar , i. e., $\mathrm{Var}(G) = |N|$.

Let the tree controlled grammar H be given as a pair $H = (G, G')$ where G is the underlying context-free grammar and G' generates the control language. Then we set

$$\mathrm{Var}(H) = \mathrm{Var}(G) + \mathrm{Var}(G').$$

By this measure we take into consideration the size of the underlying grammar G as well as the size of control grammar G'.

For a tree controlled language L, we define

$$\mathrm{Var}(L) = \min\{\mathrm{Var}(H) \mid H = (G, G'),\ G \text{ is a context-free grammar } G' \text{ is a regular grammar and } L(H) = L\}.$$

Note that, by definition, $\mathrm{Var}(H) \geq 2$ for each $H = (G, G')$ since G as well as G' have at least one nonterminal.

Moreover, we set

$$\mathcal{L}_n(TC) = \{L(H) \mid H \text{ is a tree controlled grammar and } \mathrm{Var}(H) \leq n\}$$

and

$$\mathcal{L}(TC) = \bigcup_{n \geq 2} \mathcal{L}_n(TC).$$

By definition and [23], we have the following statements.

Lemma 2 *i) For any $n \geq 2$, $\mathcal{L}_n(TC) \subseteq \mathcal{L}_{n+1}(TC)$.*
ii) $\mathcal{L}_9(TC) = \mathcal{L}(TC) = \mathcal{L}(RE)$.

3. A Bound for Recursively Enumerable Languages

In this section we show that the bound for recursively enumerable languages established in [23] can be improved from nine to seven.

Theorem 3 $\mathcal{L}(RE) \subseteq \mathcal{L}_7(TC)$.

Proof. Let $L \subseteq T^*$ be a recursively enumerable language generated by the grammar

$$G = (\{S, S', A, B, C\}, T, P \cup \{ABC \to \varepsilon\}, S)$$

in the modified Geffert normal form. We define the morphism $\phi : \{A, B, C\}^* \to \{0, \$\}^*$ by setting

$$\phi(A) = 0\$,\ \phi(B) = 0^2\$,\ \phi(C) = 0^3\$,$$

and construct a tree controlled grammar $H' = (N', T, P_\phi \cup P'', S, R')$ where

$$\begin{aligned}
N' =& \{S, S', 0, 1, \$, \#\},\\
P_\phi =& \{S \to \phi(u)Sa \mid S \to uSa \in P, u \in \{A, AB\}^*, a \in T\}\\
&\cup \{S \to S'\}\\
&\cup \{S' \to \phi(u)S'\phi(v) \mid S' \to uS'v \in P, u \in \{A, AB\}^*, v \in \{BC, C\}^*\}\\
&\cup \{S' \to \varepsilon\},\\
P'' =& \{0 \to 0, 0 \to 1, \$ \to \$, \$ \to \#, 1 \to \varepsilon, \# \to \varepsilon\},\\
R' =& (\{S, S', 0, \$, 1\#1^2\#1^3\#\} \cup T)^*.
\end{aligned}$$

First we show that any terminal derivation in G can be simulated by a derivation in H. It is clear that the first and second phases of the derivation for $w \in T^*$ in the grammar G

$$S \Longrightarrow^* w'Sw \Longrightarrow w'S'w \Longrightarrow^* w_1S'w_2w \Longrightarrow w_1w_2w,$$

$w', w_1 \in \{A, AB\}^*$, $w_2 \in \{BC, C\}^*$, $w \in T^*$, can be simulated in H using the corresponding rules of P_ϕ and chain rules $0 \to 0$, $\$ \to \$$, which result in the sentential form

$$S \Longrightarrow^* \phi(w')Sw \Longrightarrow \phi(w')S'w \Longrightarrow^* \phi(w_1)S'\phi(w_2)w \Longrightarrow \phi(w_1)\phi(w_2)w.$$

Since the rules of P_ϕ generate words from $(\{S, S', 0, \$\} \cup T)^*$, every control word of R in these phases of the derivation is also in $(\{S, S', 0, \$\} \cup T)^*$.

Let

$$z = uABCvw, u \in \{A, AB\}^*, v \in \{BC, C\}^*, w \in T^*,$$

be a sentential form in the third phase of the derivation in G. Then

$$z' = \phi(u)0\$0^2\$0^3\$\phi(v)w, \phi(u) \in \{0, \$\}^*, \phi(v) \in \{0, \$\}^*, w \in T^*,$$

is the corresponding sentential form in the derivation in H, and z' is continued as follows:

$$\phi(u)0\$0^2\$0^3\$\phi(v)w \xRightarrow{(0\to 1)^6(\$\to\#)^3(0\to 0)^*(\$\to\$)^*} \phi(u)1\#1^2\#1^3\#\phi(v)w$$
$$\xRightarrow{(1\to\varepsilon)^6(\#\to\varepsilon)^3(0\to 0)^*(\$\to\$)^*} \phi(u)\phi(v)w,$$

which simulates the elimination of the substring ABC in z.

Now we show that $L(H) \subseteq L(G)$ also holds.

Let $D : S \Longrightarrow^* w = x_1x_2\cdots x_n \in T^*$, $x_1, x_2, \ldots, x_n \in T$, be a derivation in the grammar H.

Since $x_1x_2\cdots x_n$ can be generated only by rules $S \to \phi(u)Sa \in P''$,

$$S \Longrightarrow^* w'Sx_1x_2\cdots x_n \Longrightarrow^* w''S'x_1x_2\cdots x_n,\ w', w'' \in \{0, 1, \$, \#\}^*, \quad (1)$$

is a phase of the derivation D.

If w', w'' have occurrences of 1 or #, then they must have the subword $1\#1^2\#1^3\#$ by the construction of R. Since rules of the form $S \to \phi(u)Sa$ can generate at most subwords $0\$0^2\$$, i. e., $0^3\$$ cannot be generated. Therefore w', w'' cannot contain the subword $1\#1^2\#1^3\#$. Thus, in this phase, rules of the form $S \to \phi(u)Sa$ and chain rules $0 \to 0$, $\$ \to \$$ are applied. It follows that

$$w' = w'' = \phi(u_n)\cdots\phi(u_2)\phi(u_1)$$

for some $\phi(u_n), \ldots, \phi(u_2), \phi(u_1) \in \{0, \$\}^*$. Then

$$S \Longrightarrow^* u_n\cdots u_2u_1Sx_1x_2\cdots x_n \Longrightarrow u_n\cdots u_2u_1S'x_1x_2\cdots x_n$$

is the first phase of a derivation in G, which simulates (1).

Let from S' some sentential form $w_1S'w_2$ with $w_1w_2 \in \{0, 1, \$, \#\}^*$ be generated, i. e., in H we have the derivation

$$S \Longrightarrow^* w'S'w \Longrightarrow^* w'w_1S'w_2w. \quad (2)$$

Though the subwords $0\$$, $0^2\$$ and $0^3\$$ can be generated in the first part of this phase, w_1w_2 cannot contain a subword $0\$0^2\$0^3\$$, as S' separates subwords $0\$0^2\$$ and $0^3\$$ or $0\$$ and $0^2\$0^3\$$, i. e., $0\$0^2\$S'0^3\$$ and $0\$S'0^2\$0^3\$$ can be possible subwords. Thus a subword $1\#1^2\#1^3\#$ cannot be generated, and in $S' \Longrightarrow^* w_1S'w_2$, only rules of the form $S' \to \phi(u)S'\phi(v)$, $\phi(u), \phi(v) \in \{0, \$\}^*$ and the chain rules $0 \to 0$, $\$ \to \$$ are applied. It follows that

$$w_1 = \phi(u'_m)\cdots\phi(u'_2)\phi(u'_1) \text{ and } w_2 = \phi(v'_1)\phi(v'_2)\cdots\phi(v'_m)$$

for some $\phi(u'_1), \phi(u'_2), \ldots, \phi(u'_m), \phi(v'_1), \phi(v'_2), \ldots, \phi(v'_m) \in \{0, \$\}^*$.

Then

$$u_n\cdots u_2u_1S'x_1x_2\cdots x_n \Longrightarrow^* u_n\cdots u_2u_1u'_m\cdots u'_2u'_1S'v'_1v'_2\cdots v'_mx_1x_2\cdots x_n$$

is the second phase of a derivation in G, which simulates the second phase of (2).

Let us now consider the sentential form

$$w'w_1S'w_2w. \tag{3}$$

As it is stated above, $0\$0^2\$S'0^3\$$ and $0\$S'0^2\$0^3\$$ are possible subwords containing nonterminals S', 0 and \$, (3) can be in the form

$$w_1'0\$S'0^2\$0^3\$w_2'w, \text{ where } w_1'0\$ = w'w_1, 0^2\$0^3\$w_2' = w_2$$

or

$$w_1'0\$0^2\$S'0^3\$w_2'w, \text{ where } w_1'0\$0^2\$ = w'w_1, 0^3\$w_2' = w_2.$$

By eliminating S', we obtain the sentential form

$$w'w_1w_2w$$

by rules $S' \to \varepsilon$ and $0 \to 0$, $\$ \to \$$ or the sentential form

$$w_1'1\#1^2\#1^3\#w_2'w$$

by rules $S' \to \varepsilon$, $0 \to 0$, $\$ \to \$$, and $0 \to 1$, $\$ \to \#$.

Further, the subword $1\#1^2\#1^3\#$ is erased by $1 \to \varepsilon$ and $\# \to \varepsilon$, resulting in $w_1'w_2'w$.

In the former case,

$$w'w_1S'w_2w \Longrightarrow^* w'w_1w_2w$$

is simulated by

$$uS'vw \Longrightarrow uvw, \phi(u) = w'w_1, \phi(v) = w_2,$$

which is obtained by $S' \to \varepsilon$.

In the latter case,

$$w'w_1S'w_2w = \left\{ \begin{array}{l} w_1'0\$S'0^2\$0^3\$w_2'w \\ w_1'0\$0^2S'\$0^3\$w_2'w \end{array} \right\} \Longrightarrow^* w_1'w_2'w$$

is simulated by

$$uS'vw \Longrightarrow u'ABCv'w \Longrightarrow^* u'v'w,$$

$\phi(u) = w'w_1, \phi(v) = w_2, \phi(u') = w_1', \phi(v') = w_2'$, which is obtained by $S' \to \varepsilon$ and $ABC \to \varepsilon$.

Any sentential form $z \in \{0, 1, \$, \#\}^*$ of D associated with some level (except the last one) and containing occurrences of 1 and #, has to be of the form

$$z = x1\#1^2\#1^3\#yw \text{ for some } x, y \in \{0, \$\}^*$$

by the definition of R'.

Then the possible sentential forms z^- and z^+ associated with the previous and next levels of the derivation tree are

$$\begin{aligned} z^- \in \{ & x0\$0^2\$0^3\$yw,\ x0\$S'0^2\$0^3\$yw,\ x0\$0^2\$S'0^3\$yw, \\ & x0\$1\#1^2\#1^3\#0^2\$0^3\$yw,\ x0\$0^2\$1\#1^2\#1^3\#0^3\$yw\} \end{aligned}$$

and

$$z^+ \in \{xyw,\ x'0\$1\#1^2\#1^3\#0^2\$0^3\$y'w,\ x'0\$0^2\$1\#1^2\#1^3\#0^3\$y'w\},$$

respectively, where $x', y' \in \{0, \$\}^*$.

Without loss of generality we can assume that

$$\left.\begin{array}{r} x0\$0^2\$0^3\$yw \\ x0\$S'0^2\$0^3\$yw \\ x0\$1\#1^2\#1^3\#0^2\$0^3\$yw \end{array}\right\} \Longrightarrow^* z \Longrightarrow^* \left\{\begin{array}{l} xyw \\ x'0\$1\#1^2\#1^3\#0^2\$0^3\$y'w, \end{array}\right.$$

Since the application of rules $0 \to 1$ and $\$ \to \#$ can be delayed without changing z and still generating words of R', we replace

$$x0\$S'0^2\$0^3\$yw \xRightarrow{(S'\to\varepsilon)(0\to1)^6(\$\to\#)^3(0\to0)^*(\$\to\$)^*} x1\#1^2\#1^3\#yw$$

with

$$\begin{aligned} x0\$S'0^2\$0^3\$yw & \xRightarrow{(S'\to\varepsilon)(0\to0)^*(\$\to\$)^*} x0\$0^2\$0^3\$yw \\ & \xRightarrow{(0\to1)^6(\$\to\#)^3(0\to0)^*(\$\to\$)^*} x1\#1^2\#1^3\#yw. \end{aligned}$$

The same changes can be done with the derivation

$$\begin{aligned} x0\$1\#1^2\#1^3\#0^2\$0^3\$yw & \xRightarrow{(1\to\varepsilon)^6(\#\to\varepsilon)^3} \\ & \xRightarrow{(0\to1)^6(\$\to\#)^3(0\to0)^*(\$\to\$)^*} x1\#1^2\#1^3\#yw, \end{aligned}$$

which is replaced with

$$\begin{aligned} x0\$1\#1^2\#1^3\#0^2\$0^3\$yw & \xRightarrow{(1\to\varepsilon)^6(\#\to\varepsilon)^3(0\to0)^*(\$\to\$)^*} x0\$0^2\$0^3\$yw \\ & \xRightarrow{(0\to1)^6(\$\to\#)^3(0\to0)^*(\$\to\$)^*} x1\#1^2\#1^3\#yw. \end{aligned}$$

We also do similar changes with the derivation

$$x1\#1^2\#1^3\#yw \xRightarrow{(1\to\varepsilon)^6(\#\to\varepsilon)^3(0\to1)^6(\$\to\#)^3(0\to0)^*(\$\to\$)^*} x'0\$1\#1^2\#1^3\#0^2\$0^3\$y'w,$$

i. e.,

$$\begin{aligned} x1\#1^2\#1^3\#yw & \xRightarrow{(1\to\varepsilon)^6(\#\to\varepsilon)^3(0\to0)^*(\$\to\$)^*} x'0\$0^2\$0^3\$y'w \\ & \xRightarrow{(0\to1)^6(\$\to\#)^3(0\to0)^*(\$\to\$)^*} x'1\#1^2\#1^3\#y'w. \end{aligned}$$

Now, from all cases above, we can see that $z = x1\#1^2\#1^3\#yw$ is generated from $x0\$0^2\$0^3\$yw$, and results in xyw, i. e.,

$$x0\$0^2\$0^3\$yw \Longrightarrow^* x1\#1^2\#1^3\#yw \Longrightarrow^* xyw.$$

This phase of the derivation D can be simulated by

$$uABCvw \Longrightarrow^* uvw, \phi(u) = x, \phi(v) = y,$$

in G by using $ABC \to \varepsilon$.

Thus, for every derivation D in H, we can construct a derivation in G simulating D, i. e., $L(H) \subseteq L(G)$.

Since R' can be generated by the regular grammar $G' = (\{S''\}, T'', P'', S'')$ where

$$T'' = \{S, S', 0, 1, \$, \#\} \cup T,$$
$$P'' = \{S'' \to xS'' : x \in \{S, S', 0, \$, 1\#1^2\#1^3\#\} \cup T\} \cup \{S'' \to \varepsilon\},$$

we have $\text{Var}(H) = 7$ and, consequently, $\text{Var}_{TC}(L) \leq 7$.

Thus every recursively language is generated by a tree controlled grammar with at most seven nonterminals. □

4. A Bound for Linear and Regular Simple Matrix Languages

In this section, for regular, linear and simple matrix languages, we improve the bound seven given in the preceding section to three.

Theorem 4 *$\mathcal{L}(REG) \subseteq \mathcal{L}_3(TC)$.*

Proof. Let L be a regular language and $G = (N, T, P, S)$ a regular grammar which generates L. Let $N = \{A_1, A_2, \ldots, A_n\}$ and $S = A_1$. We now construct the tree controlled grammar $H = (\{A, B\}, T, P', A, R)$ with

$$\begin{aligned} P' =& \{A \to BwA^i \mid A_j \to wA_i \in P \text{ for some } 1 \leq i, j \leq n\} \\ & \cup \{A \to Bw \mid A_j \to w \in P \text{ for some } 1 \leq j \leq n\} \\ & \cup \{A \to B,\ B \to \varepsilon\}, \\ R =& \{A\} \cup \{B^j wA^i \mid A_j \to wA_i \in P\} \cup \{B^j w \mid A_j \to w \in P\}. \end{aligned}$$

Any derivation in H has the form

$$\begin{aligned} A \Longrightarrow & Bw_1A^{i_1} \Longrightarrow^* w_1B^{i_1}w_2A^{i_2} \Longrightarrow^* w_1w_2B^{i_2}w_3A^{i_3} \\ \Longrightarrow^* & w_1w_2 \cdots w_{n-2}B^{i_{n-2}}w_{n-1}A^{i_{n-1}} \Longrightarrow^* w_1w_2 \cdots w_{n-2}w_{n-1}B^{i_{n-1}}w_n \\ \Longrightarrow^* & w_1w_2 \cdots w_{n-2}w_{n-1}w_n \end{aligned} \quad (4)$$

(by the structure of R, in the sentential form $w_1w_2 \ldots w_{r-1}B^{i_{r-1}}w_rA^{i_r}$, we have to replace the first $i_r - 1$ occurrences of A by B's and the last occurrence of

A by $Bw_{r+1}A^{i_{r+1}}$ or by Bw_n for $r = n-1$) and the words at the levels of the corresponding derivation tree are

$$A,\ Bw_1A^{i_1},\ B^{i_1}w_2A^{i_2}, \ldots,\ B^{i_{n-2}}w_{n-1}A^{i_{n-1}},\ B^{i_{n-1}}w_n. \tag{5}$$

According to R, we have the rules

$$\begin{aligned} S =& A_1 \to w_1A_{i_1},\ A_{i_1} \to w_2A_{i_2},\ A_{i_2} \to w_3A_{i_3}, \ldots, \\ & A_{i_{n-2}} \to w_{n-1}A_{i_{n-1}},\ A_{i_{n-1}} \to w_n \end{aligned} \tag{6}$$

in P. Hence we have the derivation

$$\begin{aligned} S = A_1 &\Longrightarrow w_1A_{i_1} \Longrightarrow w_1w_2A_{i_2} \Longrightarrow w_1w_2w_3A_{i_3} \Longrightarrow \cdots \\ &\Longrightarrow w_1w_2\cdots w_{n-2}A_{i_{n-2}} \Longrightarrow w_1w_2\cdots w_{n-2}w_{n-1}A_{i_{n-1}} \\ &\Longrightarrow w_1w_2\cdots w_{n-2}w_{n-1}w_n \end{aligned} \tag{7}$$

in G. Therefore, $L(H) \subseteq L(G)$.

Conversely, it is easy to see that, for any derivation (7) in G, where the rules (6) are applied, there is a derivation (4) with the words given in (5) in the levels. Hence we have $L(G) \subseteq L(H)$.

Since R is a finite set, it can be generated by a regular grammar with one nonterminal (the nonterminal generates all words in one step by a rule). Therefore we have $\mathrm{Var}(H) = 3$. □

We note that the existence of an upper bound for the number of nonterminals comes from the control since there are regular languages L_n, $n \geq 0$, which require n nonterminals for the generation by context-free grammars (see [14]).

We now generalize the proof to linear languages.

Theorem 5 $\mathcal{L}(LIN) \subseteq \mathcal{L}_3(TC)$.

Proof. Let L be a linear grammar. It is well-known that L can be generated by a linear grammar $G = (N, T, P, S)$, where all rules are of the form $A \to wB$ or $A \to Bw$ or $A \to w$ with $A, B \in N$ and $w \in T^*$. Moreover, let $N = \{A_1, A_2, \ldots, A_n\}$ and $S = A_1$. Starting from G, we now modify the construction of $H = (\{A, B\}, T, P', A, R)$ in the proof of Theorem 4 by defining the set of productions and the control set as follows:

$$\begin{aligned} P' =& \{A \to BwA^i \mid A_j \to wA_i \in P \text{ for some } 1 \leq i, j \leq n\} \\ & \cup \{A \to A^iwB \mid A_j \to A_iw \in P \text{ for some } 1 \leq i, j \leq n\} \\ & \cup \{A \to wB \mid A_j \to w \in P \text{ for some } 1 \leq j \leq n\} \\ & \cup \{A \to B,\ B \to \varepsilon\}, \\ R =& \{A\} \cup \{B^jwA^i \mid A_j \to wA_i \in P\} \\ & \cup \{A^iwB^j \mid A_j \to A_iw \in P\} \cup \{wB^j \mid A_j \to w \in P\}. \end{aligned}$$

If we have a sentential form zA_iz', then we have the level A^ivB^r or B^rvA^i for some r in the corresponding derivation tree. If we apply a rule $A_i \longrightarrow wA_j$ or $A_i \longrightarrow A_jw$ to zA_iz', we erase the r occurrences of B, replace the first $i-1$ occurrences of A by B and the last of occurrence of A by BwA^j or the first occurrence of A by A^jwB and the remaining occurrences of A by B, respectively. Then we get the sentential forms zB^iwA^jz' or zA^jwB^iz' and the corresponding levels B^iwA^j or A^jwB^i, respectively.

Now we can follow the arguments given in the proof of Theorem 4 to show that $L(H) = L$. Since R is finite, again, we obtain $\mathrm{Var}(H) = 3$. □

We can transform the proof to regular simple matrix grammars, too.

Theorem 6 $\mathcal{L}(RSM) \subseteq \mathcal{L}_3(TC)$.

Proof. Let $G = (V_1, V_2, \ldots, V_n, T, M, S)$ be a regular simple matrix grammar. By Lemma 1, without loss of generality we assume that M does not contain rules of type 1. Let

$$V_1 \cup V_2 \cup \cdots \cup V_n = \{A_2, A_3, \ldots, A_m\}.$$

Then we construct the tree controlled grammar $H = (\{A, B\}, T, P, A, R)$ with

$$\begin{aligned} P =& \{A \longrightarrow BA^{i_1}BA^{i_2}\ldots BA^{i_n} \mid (S \longrightarrow A_{i_1}A_{i_2}\ldots A_{i_n}) \in M\} \\ &\cup \{A \longrightarrow Bw_rA^{i_r} \mid \\ &\qquad (A_{j_1} \longrightarrow w_1A_{i_1}, \ldots, A_{j_r} \longrightarrow w_rA_{i_r}, \ldots, A_{j_n} \longrightarrow w_nA_{i_n}) \in M\} \\ &\cup \{A \longrightarrow Bw_r \mid (A_{j_1} \longrightarrow w_1, \ldots, A_{j_r} \longrightarrow w_r, \ldots, A_{j_n} \longrightarrow w_n) \in M\} \\ &\cup \{A \longrightarrow B, B \longrightarrow \varepsilon\} \end{aligned}$$

and

$$\begin{aligned} R =& \{A\} \cup \{BA^{i_1}BA^{i_2}\ldots BA^{i_n} \mid (S \longrightarrow A_{i_1}A_{i_2}\ldots A_{i_n}) \in M\} \\ &\cup \{B^{j_1}w_1A^{i_1}B^{j_2}w_2A^{i_2}\ldots B^{j_n}w_nA^{i_n} \mid \\ &\qquad (A_{j_1} \longrightarrow w_1A_{i_1}, A_{j_2} \longrightarrow w_2A_{i_2}, \ldots, A_{j_n} \longrightarrow w_nA_{i_n}) \in M\} \\ &\cup \{B^{j_1}w_1B^{j_2}w_2\ldots B^{j_n}w_n \mid \\ &\qquad (A_{j_1} \longrightarrow w_1, A_{j_2} \longrightarrow w_2, \ldots, A_{j_n} \longrightarrow w_n) \in M\}. \end{aligned}$$

It is easy to see (by arguments as given in the proof of Theorem 4) that

$$q_1v_1A_{j_1}q_2v_2A_{j_2}\ldots q_nv_nA_{j_n} \Longrightarrow q_1v_1w_1A_{i_1}q_2v_2w_2A_{i_2}\ldots q_nv_nw_nA_{i_n}$$

holds in G if and only if

$$\begin{aligned} &q_1B^{k_1}v_1A^{j_1}q_2B^{k_2}v_2A^{j_2}\ldots q_nB^{k_n}v_nA^{j_n} \\ &\qquad \Longrightarrow q_1v_1B^{j_1}w_1A^{i_1}q_2v_2B^{j_2}w_2A^{i_2}\ldots q_nv_nB^{j_n}w_nA^{i_n} \end{aligned}$$

holds in H and analogous relations hold for the initial and terminating derivation steps. Thus we get $L(G) = L(H)$. By construction $\mathrm{Var}(H) = 3$ since R is finite. □

We now prove the optimality of the bounds given in the Theorems 4, 5, and 6.

Lemma 7 *The regular language* $L = \{a^r\#a^s\#a^t \mid r, s, t \geq 0\}$ *is not in* $\mathcal{L}_2(TC]$.

Proof. Assume that this language is in $\mathcal{L}_2(TC)$. Then there is a tree controlled grammar $H = (G, G')$, where G is a context-free grammar and G' is a regular grammar, such that $\mathrm{Var}(H) = 2$. Thus any of these grammars has exactly one nonterminal. Let S be the unique nonterminal of G. Clearly, if $S \longrightarrow x$ is a production such that $x \in \{a, \#\}^*$, then x belongs to $L(H)$ and contains exactly two symbols #. Also, the maximum number of nonterminals that may appear in a level of a sentential form of G, according to the control language, is 1 (otherwise one would derive words that can have more than two #s by a termination of all occurrences of S). Finally, the only productions used in a derivation that is accepted by the control language and introduce a symbol # are those that end the derivation. So, in a word derived by H, the two symbols # have at most distance d, where d is the maximum length of the righthand side of a production. This is a contradiction. □

Corollary 8 $\mathcal{L}_2(TC)$ *is properly included in* $\mathcal{L}_3(TC)$.

Proof. The language L of the proof of Lemma 7 is in $\mathcal{L}(REG)$. By Theorem 4, it is in $\mathcal{L}_3(TC)$. Now the proper inclusion follows from Lemma 7. □

We mention that, conversely, $\mathcal{L}_2(TC)$ contains the languages $\{a^{2^n} \mid n \geq 0\}$ (the tree controlled grammar $(\{S\}, \{a\}, \{S \longrightarrow SS, S \longrightarrow a\}, S, \{S\}^*)$ generates it, see [7], Example 2.3.2) which does not belong to $\mathcal{L}(CF)$ and $\mathcal{L}(RSM)$ (see [7], Corollary 2 of Section 1.5).

5. A Bound for Context-Free Languages

In this section we prove that four nonterminals are sufficient to generate context-free languages by tree controlled grammars.

Theorem 9 $\mathcal{L}(CF) \subseteq \mathcal{L}_4(TC)$.

Proof. Let $G = (N, T, P, A_1)$ be a context-free grammar in Chomsky Normal Form. Also, assume that the starting symbol A_1 does not appear in the right-hand side of any production of G; the only allowed λ-production is $A_1 \longrightarrow \lambda$. Let $N = \{A_1, A_2, \ldots, A_n\}$ for some $n \geq 1$.

Let $A, B, \#$ be three symbols not contained in N. We define the context-free grammar $G' = (N', T, P', B)$ having the set of non-terminals $N' = \{A, B, \#\}$

and the productions set $P' = M_1 \cup M_2 \cup M_3 \cup M_4$, where

$$
\begin{aligned}
M_1 = & \{B \longrightarrow \#Aa \mid A_1 \longrightarrow a \in P, a \in T \cup \{\lambda\}\} \\
& \cup \{B \longrightarrow \#AB^jAB^kA \mid A_1 \longrightarrow A_jA_k \in P, 2 \leq j, k \leq n\} \\
M_2 = & \{B \longrightarrow A\} \\
M_3 = & \{B \longrightarrow AB^jAB^kA \mid A_i \longrightarrow A_jA_k \in P, 2 \leq i, j, k \leq n\} \\
& \cup \{B \longrightarrow AaA \mid A_i \longrightarrow a \in P, a \in T, 2 \leq i \leq n\} \\
M_4 = & \{A \longrightarrow \#, \# \longrightarrow \lambda\}.
\end{aligned}
$$

We also define the regular language $R = R_1^* R_2$, where

$$
\begin{aligned}
R_1 = & \{\#\} \cup \{\#A^iB^jAB^kA \mid A_i \longrightarrow A_jA_k \in P, i, j, k \geq 2\} \\
& \cup \{\#A^iaA \mid A_i \longrightarrow a \in P, i \geq 2\} \\
R_2 = & \{\lambda\} \cup \{\#AB^jAB^kA \mid A_1 \longrightarrow A_jA_k \in P, 1 \leq j, k \leq n\} \\
& \cup \{\#Aa \mid A_1 \longrightarrow a \in P\} \cup \{B\}.
\end{aligned}
$$

Note that the words of the control language, by its definition, consist of the catenation of t words from R_1, where $t \geq 0$, and exactly one word from R_2; however, this last word can be λ, so the control language R contains all the words from R_1^*. Nevertheless, all the words from R_2 are in R, as the prefix of a word from R consisting in the catenation of t words from R_1 can actually be empty, for $t = 0$.

In the following, we describe the derivations of the tree controlled grammar $H = (N', T, P', B, R)$ and show that it generates the same language as G.

The first step in a derivation of H always consists in rewriting B according to one of the rules from M_1. That is, a derivation in H starts only with a rule $B \longrightarrow \#AB^jAB^kA$ with $A_1 \longrightarrow A_jA_k \in P$ or with a rule $B \longrightarrow \#Aa$ for $A_1 \longrightarrow a \in P$. In both these cases, the words found on the second level of the derivation tree are from R_2 and, consequently, from R. No other rule that rewrites B can be applied, as we would obtain a non-empty word that contains no symbol $\#$ on the second level of the tree; but such a word would not be contained in R.

In the case when we have $\#Aa$ on the second level of the tree, the derivation continues in only possible way. In the first step, $\#$ is rewritten into λ and A is rewritten into $\#$, to obtain $\# \in R_1$ on the third level. In the second and final step, the symbol $\#$ is rewritten into λ and the derivation ends. The generated string was a, and this belonged to $L(G)$ as $A_1 \longrightarrow a \in P$.

In the case when the second level of the tree contains a word $\#AB^jAB^kA$ with $2 \leq j, k \leq n$ the derivation is continued as follows. The symbol $\#$ is rewritten into λ and the symbols A are all rewritten into $\#$, as there are no other choice. Hence, we will have on the third level of the tree a word $\#x\#y\#$, where x is derived in one step from B^j and y is derived from B^k. In a correct derivation (with respect to the control language) we should have $\#x\#y\# \in R$. As this word ends with $\#$ it means that its suffix from R_2 is the empty word.

Consequently, no word from R_2 can appear as a factor of $\#x\#y\#$, so no rule from M_1 can be applied at this derivation step. It follows that the symbols B from the first group can only be rewritten into A, AaA or AB^sAB^tA, with $a \in T$ and $2 \leq s,t \leq n$. But this means that no other $\#$ symbols appears in x or y, and that $\#x\#y\# \in R_1^*$. The only way for this to hold is to have $\#x, \#y \in R_1$. Moreover, the only possibility to have this is to rewrite the first $j-1$ symbols B into A and the last symbol B into AaA or AB^sAB^tA, with $a \in T$ and $2 \leq s,t \leq n$. In the first case, $\#x = \#A^jaA$ will be in R_1 if an only if $A_j \longrightarrow a \in P$, while in the second case, $\#x = \#A^jB^sAB^tA$ will be in R_1 if and only if $A_j \longrightarrow A_sA_t \in P$. In a similar fashion, one can show that $\#y = \#A^kaA$ with $A_k \longrightarrow a \in P$ or $\#x = \#A^kB^sAB^tA$ with $A_k \longrightarrow A_sA_t \in P$.

Further, we show by induction that the words that may appear on the level r of a derivation tree of H, for $r \geq 3$, have the from

$$\#^m(\#^{t_1}\#x_1)\dots(\#^{t_p}\#x_p)\#^s$$

where $p,m \geq 0$, $s > 0$, $x_i \in \{A^\ell B^jAB^kA \mid A_\ell \longrightarrow A_jA_k \in P, \ell, j,k \geq 2\} \cup \{A^jaA \mid A_j \longrightarrow a \in P, a \in V\}$ and $t_i \geq 0$ for all $1 \leq i \leq p$, and there is a derivation tree of G that has on the r^{th} level the word $y_1 \dots y_p$ such that $y_i = A_jA_k$ if $x_i = A^\ell B^jAB^kA$ and $y_i = a$ if $x_i = A^ja$.

The property holds for $r = 3$, by the explanations above. Let us assume that it holds for some $r \geq 3$, and we show that it also holds for $r+1$. Let w be the word appearing on level r of some derivation tree of H. All the $\#$ appearing in this word will be rewritten into λ and all the symbols A will be rewritten into $\#$, as these are the only rules that can be applied to $\#$ and A, respectively. If w contains no B or terminal symbol, the conclusion follows: the next level will contain only symbols $\#$. Let us assume now that w contains at least one symbol B. Therefore, w contains at least one factor of the form $\#A^sB^tAB^pA$. Take the leftmost such factor that occurs in w; it will be followed only by symbols $\#$ and A; anyway, as the last A of that factor is rewritten into $\#$, it is clear that the word on the next level will end with $\#$. The same reasoning holds for case when w contains terminal symbols, and we obtain that the word on the next level will end with $\#$. We continue by looking at the way the factors $\#x_\ell$ are rewritten. First, a factor $\#x_\ell = \#A^jaA$ is transformed into $\#^{j+1}$. Further, let us analyse how a factor $\#x_\ell = \#A^iB^jAB^kA$ of w is rewritten in a valid derivation step. This word becomes $\lambda\#^ix\#y\#$, where B^j is rewritten into x and B^k to y, and $i > 1$. By arguments similar to the ones used in the description of the derivation step transforming the second level of a tree in its third level, we obtain that the only possibility to rewrite the first group of symbols B is the following. We rewrite the first $j-1$ symbols B into A and the last symbol B into AaA or AB^sAB^tA, for some $a \in T$ and $2 \leq s,t \leq n$. Similarly, the only possibility to rewrite the second group of symbols B is to rewrite the first $k-1$ symbols B into A and the last symbol B into AaA or AB^sAB^tA, for some $a \in T$ and $2 \leq s,t \leq n$. We obtain, once more, that $\#x = \#A^jaA$ for some $A_j \longrightarrow a \in P$ or $\#x = \#A^jB^sAB^tA$ for some $A_j \longrightarrow A_sA_t \in P$; also, $\#y = \#A^kaA$ with $A_k \longrightarrow a \in P$ or $\#x = \#A^kB^sAB^tA$ with $A_k \longrightarrow A_sA_t \in P$. But this proves that our statement is true.

In other words, we showed that there is a bijection between the derivations in the grammar H and those of grammar G. Now, it follows easily that the language generated by G' with respect to the control language R, thus, $L(H)$, equals $L(G)$.

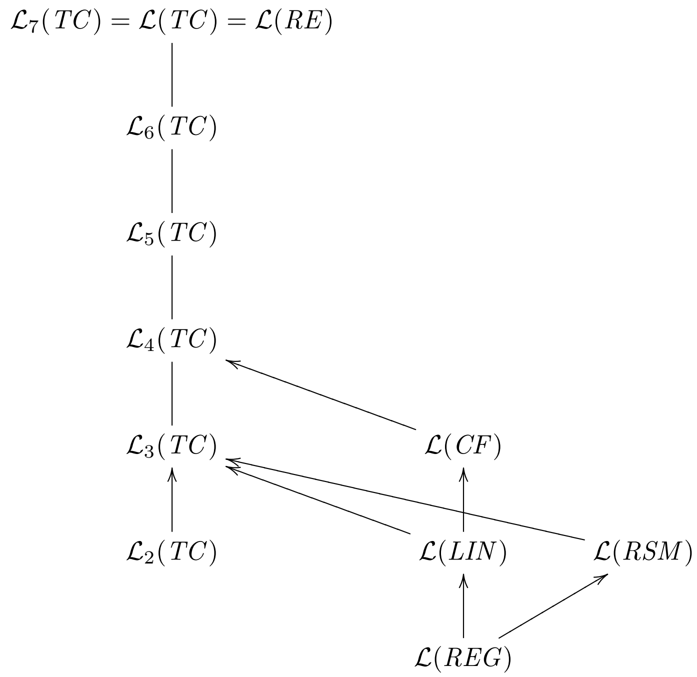
Since G' has three nonterminals and R is generated by the grammar

$$G'' = (\{S\}, \{A, B, \#\} \cup T, \{S \longrightarrow wS \mid w \in R_1\} \cup \{S \longrightarrow w \mid w \in R_2\}, S)$$

with only one nonterminal S, we get that $L(G)$ can be generated by a tree controlled grammar given by (G', G''), with nonterminal complexity 4. □

6. Conclusions

First we summarize our results in the diagram shown in Figure 6, where (upward) lines and arrows denote inclusion and proper inclusion, respectively, and families are incomparable if they are not connected.



It is an open problem whether the inclusions $\mathcal{L}_n(TC) \subseteq \mathcal{L}_{n+1}(TC)$ are proper for $3 \leq n \leq 7$.

We know that $\mathcal{L}_2(TC)$ does not contain all regular sets (Lemma 7), i.e., $\mathcal{L}(REG)$, $\mathcal{L}(LIN)$ and $\mathcal{L}(RSM)$ are not included in $\mathcal{L}_2(TC)$. However, we do not know whether or not $\mathcal{L}(CF)$ is included in $\mathcal{L}_3(TC)$.

Moreover, we do not know good bounds for matrix or ET0L languages which can be obtained by special choices of control languages (see [8]).

The aim of the control is to check that the levels of the derivation tree have a special form described by a regular language. That means that one has to check whether the levels belong to some given regular language. Such a check can easily be done by a finite automata but hardly by a regular grammar. Therefore it is of interest to study a complexity measure which – besides the number of nonterminals of the underlying context-free grammar – takes into consideration the complexity of the finite automaton (for instance, its number of states). Using this approach, we get much higher bounds since we need more states to accept the considered regular languages than nonterminals to generate them. An improvement of such bounds remains to be done.

References

[1] A.B. Cremers, O. Mayer, K. Weiss, On the complexity of regulated rewriting, Information and Control 33 (1974) 10–19.

[2] E. Csuhaj-Varjú, Gy. Vaszil, Scattered context grammars generated any recursively enumerable language with two nonterminals, Information Processing Letters 110 (2010) 902–907.

[3] K. Čulik II, H. Maurer, Tree controlled grammars, Computing 19 (1977) 129–139.

[4] J. Dassow, Remarks on the complexity of regulated rewriting, Fundamenta Informaticae 7 (1984) 83–103.

[5] J. Dassow, Gh. Păun, Further remarks on the complexity of regulated rewriting, Kybernetika 21 (1985) 213–227.

[6] J. Dassow, Gh. Păun, Some notes on the complexity of regulated rewriting, Bull. Math. Soc. Sci. Math. R.S. Roumanie 30 (1986) 203–212.

[7] J. Dassow, Gh. Păun, Regulated Rewriting in Formal Language Theory. Springer-Verlag, Berlin, 1989.

[8] J. Dassow, R. Stiebe, B. Truthe, Generative capacity of subregularly tree controlled grammars, International Journal of Foundations of Computer Science 21 (2010) 723–740.

[9] H. Fernau, Nonterminal complexity of programmed grammars, Theoretical Computer Science 296 (2003) 225–251.

[10] H. Fernau, R. Freund, M. Oswald, K. Reinhardt, Refining the nonterminal complexity of graph-controlled, programmed, and matrix grammars, Journal of Automata, Languages and Combinatorics 12 (2007) 117–138.

[11] H. Fernau, A. Meduna, On the degree of scattered context-sensitivity, Theoretical Computer Science 290 (2003) 2121-2124.

[12] H. Fernau, A. Meduna, A simultaneous reduction of several measures of descriptional complexity in scattered context grammars, Information Processing Letters 86 (2003) 235–240.

[13] V. Geffert, Context-free-like forms for phrase-structure grammars, in: M. Chytil, L. Janiga, V. Koubek (eds.), Mathematical Foundations of Computer Science, Lecture Notes in Computer Science 324, Springer, 1988, 309–317.

[14] J. Gruska, On a classification of context-free languages, Kybernetika 3 (1967) 22–29.

[15] O. Ibarra, Simple matrix grammars, Information and Control 17 (1970) 359–394.

[16] T. Masopust, Descriptional complexity of multi-parallel grammars, Information Processing Letters 108 (2008) 68–70.

[17] A. Meduna, Generative power of three-nonterminal scattered context grammars, Theoretical Computer Science 246 (2000) 279–284.

[18] A. Meduna, A. Gopalaratnam, On semi-conditional grammars with productions having either forbidding or permitting conditions, Acta Cybernetica 11 (1994) 307–323.

[19] A. Meduna, M. Švec, Reduction of simple semi-conditional grammars with respect to the number of conditional productions, Acta Cybernetica 15 (2002) 353–360.

[20] Gh. Păun, On the generative capacity of tree controlled grammars, Computing 21 (1979) 213–220.

[21] Gh. Păun, Six nonterminals are enough for generating each RE language by a matrix grammar, International Journal of Computional Mathematics 15 (1984) 23–37.

[22] G. Rozenberg, A. Salomaa (eds.), Handbook of Formal Languages, Vol. I–III, Springer, Berlin, 1997.

[23] S. Turaev, J. Dassow, M. Selamat, Nonterminal complexity of tree controlled grammars, Theoretical Computer Science 412 (2011) 5789–5709.

[24] Gy. Vaszil, On the descriptional complexity of some rewriting mechanisms regulated by context conditions, Theoretical Computer Science 330 (2005) 361–373.