Nonterminal complexity of tree controlled grammars

S. Turaev\textsuperscript{a,}\textsuperscript{*}, J. Dassow\textsuperscript{b,}\textsuperscript{**}, M. Selamat\textsuperscript{a,}\textsuperscript{**}

\textsuperscript{a}Faculty of Computer Science and Information Technology, University Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia
\textsuperscript{b}Faculty of Computer Science, Otto-von-Guericke University Magdeburg, D-39016 Magdeburg, Germany

\textbf{A R T I C L E I N F O}

Article history:
Received 26 February 2011
Received in revised form 1 June 2011
Accepted 22 June 2011
Communicated by M. Ito

Keywords:
Formal languages
Regulated rewriting
Tree controlled grammars
Descriptive complexity

\textbf{A B S T R A C T}

This paper studies the nonterminal complexity of tree controlled grammars. It is proved that the number of nonterminals in tree controlled grammars without erasing rules leads to an infinite hierarchy of families of tree controlled languages, while every recursively enumerable language can be generated by a tree controlled grammar with erasing rules and at most nine nonterminals.

\textcopyright 2011 Elsevier B.V. All rights reserved.

1. Introduction

A tree controlled grammar (see \cite{21}) is a context-free grammar accompanied by some regular language. The structure of the derivation trees of the context-free grammar is restricted by the requirement that all words belonging to all levels (except the last one) of the derivation tree have to belong to the regular language. Tree controlled grammars generate all context-sensitive languages if erasing rules are not allowed (see Theorem 4 in \cite{17}), and all recursively enumerable languages if erasing rules are allowed (see Theorem 3.6 in \cite{21}).

Since "economical" representation of formal languages has been always important, it is interesting to investigate their grammars from the point of view of descriptive complexity measures such as the number of nonterminals and the number of production rules.

The study of the descriptive complexity with respect to regulated grammars was started in \cite{1,3–5,18}. In recent years several interesting results on this topic have been obtained. For instance, \cite{13} demonstrates that four-nonterminal matrix grammars with leftmost derivations characterize the family of recursively enumerable languages. The nonterminal complexity of programmed and matrix grammars is studied in \cite{7}, 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 \cite{8}. There are several papers which study the descriptive complexity of scattered context grammars \cite{2,9,10,14,20}, semi-conditional grammars \cite{15,16,18,20}, and multi-parallel grammars \cite{12}.

This paper is devoted to the investigation of the nonterminal complexity of tree controlled grammars, which has not been studied at all until now. We prove that every recursively enumerable language is generated by a tree controlled grammar with no more than nine nonterminals.

\textsuperscript{*} Corresponding author. Tel.: +60 389466517; fax: +60 389466576.
\textsuperscript{**} Corresponding author.

E-mail addresses: sherzod@fsktm.upm.edu.my, sturaev@gmail.com (S. Turaev), dassow@iws.cs.uni-magdeburg.de (J. Dassow), hasan@fsktm.upm.edu.my (M. Selamat).

0304-3975/$– see front matter \textcopyright 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2011.06.033
2. Definitions and an example

We assume that the reader is familiar with formal language theory (see [6,19]). Let \( T^* \) denote the set of all finite words over an alphabet \( T \). The empty word is denoted by \( \lambda \). The cardinality of a set \( X \) is denoted by \(|X|\).

Let \( \mathcal{G} \) be a family of grammars and \( \mathcal{L}(\mathcal{G}) \) be the family of languages generated by grammars of \( \mathcal{G} \). The family of arbitrary phrase structure grammars is denoted by \( \mathcal{R} \).

A context-free grammar is specified as a quadruple \( G = (N, T, P, S) \) where \( N \) and \( T \) are the disjoint alphabets of nonterminals and terminals, respectively. A grammar is called regular if all its rules are of the form \( A \to wB \) or \( A \to w \) with \( A, B \in N \) and \( w \in T^* \). By \( \text{Var}(G) \) we denote the number of the nonterminals of a grammar \( G = (N, T, P, S) \), i.e., \( \text{Var}(G) = |N| \).

By the definition, the following lemma is obvious.

**Lemma 1.** For \( \mathcal{G} \in \{ \mathcal{C}, \mathcal{C} - \lambda \} \),
\[
\mathcal{L}_2(\mathcal{G}) \subseteq \mathcal{L}_3(\mathcal{G}) \subseteq \cdots \subseteq \mathcal{L}_n(\mathcal{G}) \subseteq \cdots .
\]

**Example 1.** Let \( L_1 = \{a^{2^n} : n \geq 0\} \). Then \( L_1 \) is generated by the tree controlled grammar \( H_1 = (\{S\}, \{a\}, \{S \to a, S \to SS\}, S, S^*) \) (for details, see Example 2.3.2 in [6]).

For the core grammar \( G_1 = (\{S\}, \{a\}, \{S \to a, S \to SS\}, S) \) and for the regular grammar \( G'_1 = (\{S'\}, \{S\}, \{S' \to SS', S' \to S\}, \{S\}) \), which generates \( S^* \),
\[
\text{Var}(G_1) = 1 \quad \text{and} \quad \text{Var}(G'_1) = 1.
\]

Thus,
\[
\text{Var}_{\mathcal{C} - \lambda}(L_1) \leq \text{Var}(H_1) = 2.
\]

On the other hand, for any tree controlled grammar \( L = \mathcal{L}(G, G') \), \( \text{Var}(L) \geq 2 \), where \( \mathcal{G} \in \{ \mathcal{C}, \mathcal{C} - \lambda \} \), since \( \text{Var}(G) \geq 1 \) and \( \text{Var}(G') \geq 1 \). Therefore, \( \text{Var}_{\mathcal{C} - \lambda}(L_1) = 2 \). \( \square \)

3. An infinite hierarchy

In this section, we investigate the hierarchy problem of the families of languages generated by tree controlled grammars without erasing rules; we show that the inclusions in (1) are strict in this case.

**Lemma 2.** For \( n \geq 1 \), let
\[
L_n = \bigcup_{i=1}^{n} \{a^i : j \geq 1\}.
\]

Then
\[
\text{Var}_{\mathcal{C} - \lambda}(L_n) = n + 1.
\]
Proof. Let $n \geq 1$. We consider the tree controlled grammar

$$H_n = ([A_1, A_2, \ldots, A_n], \{a_1, a_2, \ldots, a_n\}, P_n, A_1, R_n)$$

with

$$P_n = \bigcup_{i=2}^{n} \{ A_1 \rightarrow A_i \} \cup \bigcup_{i=1}^{n} \{ A_i \rightarrow a_iA_iA_i, A_i \rightarrow a_i, A_i \rightarrow a_ia_i \}$$

and

$$R_n = \{ A_i : 1 \leq i \leq n \} \cup \{ a_k^iA_iA_i : 1 \leq k \leq 2, 1 \leq i \leq n \}.$$ 

Starting from the axiom $A_1$, we have the following derivations

$$A_1 \Rightarrow a_1, \quad A_1 \Rightarrow a_2^2, \quad A_1 \Rightarrow a_1A_1A_1, \quad A_1 \Rightarrow A_i \text{ for } 2 \leq i \leq n. \quad (3)$$

The derived word also form the first level of the derivation tree, and all these words belong to $R_n$ or are terminal ones. We now discuss the possible continuation, where we assume without loss of generality that all nonterminals of the word of some level are replaced before a nonterminal in a later level is replaced, i.e., we go from one level of the derivation tree to the next one.

We first consider the continuation from $a_1A_1A_1$. In order to get the next level we have to replace both occurrences of $A_1$. Let us assume that we apply the rules $A_1 \rightarrow u$ and $A_1 \rightarrow v$. Then the word of the second level is $uw$ and the derived word is $a_1uw$. If $u = A_i$ for some $i, 2 \leq i \leq n$, or $u = a_1A_1A_1$, then $uw$ is not in $R_n$ because $v$ is not the empty word. If $u = a_1a_1$, then $v$ has to be terminal, too, since otherwise we do not get a word of $R_n$. Therefore we get the words $a_1^2$ and $a_1^3$ in the last level with the sentential forms $a_1^4$ and $a_1^5$, which belong to $L(H)$. Finally, let $u = a_1A_1$. If $v$ is a terminal word, too, then we get the words $a_1^2$ or $a_1^3$ as words of the last level and the sentential forms words $a_1^4$ and $a_1^5$ in $L(H)$. The remaining case is $u = a_1$ and $v = a_1A_1A_1$, which gives the word $uw = a_1^3A_1A_1$ in the second level and the sentential form $a_1^3A_1A_1$. Now, as above in the second level, we can show that in the third level, we can only obtain either $a_1^3A_1A_1$ with the corresponding sentential form $a_1^3A_1A_1$, or a terminal word $a_1^5$ or $a_1^6$ in $L(H)$. In general, if we have the word $a_1^3A_1A_1$ in level $k$, then the corresponding sentential form is $a_1^{2k-1}A_1A_1$ and in the $(k + 1)$th level we have the word $a_1^{2k-1}A_1A_1A_1$, again, with the sentential form $a_1^{2k-1}A_1A_1$ or terminal words with the sentential form

$$a_1^{2k-1}A_1A_1 \Rightarrow \begin{cases} a_1^{2k+1}, & \text{by two applications of } A_1 \rightarrow a_1, \\ a_1^{2k+2}, & \text{by applying } A_1 \rightarrow a_1A_1A_1, \\ a_1^{2k+3}, & \text{by two applications of } A_1 \rightarrow a_1a_1. \end{cases}$$

Thus, from $a_1A_1A_1$, the grammar $H_n$ generates all and only powers of $a_1$, that is $a_1^j, j \geq 1$.

We now discuss the continuation from $A_i, 2 \leq i \leq n$, which is the second nonterminal word obtained in (3). Then we get termination or $a_iA_1A_1$ in the second word in the level of the sentential form. As above, we can show that from $A_i$, the grammar $H_n$ generates all and only powers of $a_i$, that is $a_i^j, 1 \leq i \leq n, j \geq 1$.

Thus we have $L(H_n) = L_n$.

Since $R_n$ for each $n \geq 1$ is a finite language, one nonterminal is enough to generate $R_n$, i.e., $R_n$ can be generated by the grammar

$$G_n = ([S], \{ A_i : 1 \leq i \leq n \} \cup \{ a_k^i : 1 \leq k \leq 2, 1 \leq i \leq n \}, P_n', S)$$

where $P_n' = \{ S \rightarrow A_i : 1 \leq i \leq n \} \cup \{ S \rightarrow a_k^iA_iA_i : 1 \leq k \leq 2, 1 \leq i \leq n \}$. Then $Var(H_n) = n + 1$ and consequently,

$$\text{Var}_{\rho_{e-1}}(L_n) \leq n + 1.$$

Now we show that $\text{Var}(L_n) \geq n + 1$. Let $L_0$ be generated by a tree controlled grammar $H' = (N', T', P', S', R')$. Let $q = \max(||w| : A \rightarrow w \in P')$ and $l$ be a natural number with $l \geq q + 1$. Since $H'$ does not have erasing rules, for each terminal $a_i, 1 \leq i \leq n$, it has to have a terminating rule of the form $A_i \rightarrow a_i^{(r,i)}$ for some positive integer $(r, i)$.

For each $i$ with $1 \leq i \leq n$, we consider an allowed derivation

$$S' \Rightarrow w_{i,1} \Rightarrow w_{i,2} \Rightarrow \cdots \Rightarrow w_{i,m_i} \Rightarrow a_i^{(k,i)},$$

where $w_{i,m_i} = a_i^{(k,i)}A_i^{(k',i)}, (k, i), (k', i) \in \mathbb{N}$, with the word $a_i^{(s,i)}A_i^{(s',i)}$ at the corresponding level of the derivation tree where $0 \leq (s, i) \leq (k, i)$ and $0 \leq (s', i) \leq (k', i)$. Then there is a rule $A_i \rightarrow a_i^{(r,i)}$ in $P'$ such that $(k, i) + (k', i) + (r, i) = l$.

Assume that for some $1 \leq i < j \leq n, A_i = A_j$, i.e., for two different terminals $a_i$ and $a_j$, there are two terminating rules with the same left-hand side. Let $A_i \rightarrow a_i^{(r,i)} \in P'$ and $A_i \rightarrow a_i^{(r'',j)} \in P'$ for some positive integers $(r, i)$ and $(r'', j)$. Then

$$S' \Rightarrow w_{i,1} \Rightarrow w_{i,2} \Rightarrow \cdots \Rightarrow w_{i,m_i} = a_i^{(k,i)}A_i^{(k',i)} \Rightarrow \begin{cases} a_i^{(k,i)}a_j^{(r',j)}a_i^{(k',i)}, \\ a_i^{(k,i)}a_j^{(r'',j)}a_i^{(k',i)}. \end{cases}$$

are also the allowed derivations but the word $a_i^{(k,i)}a_j^{(r'',j)}a_i^{(k',i)}$ is not in $L_n$. 

S. Turaev et al. / Theoretical Computer Science 412 (2011) 5789–5795 5791
Therefore, to generate $L_n$ we need at least $n$ nonterminals in the grammar $(N', T', P', S')$, and at least one nonterminal in the grammar which generates $R'$. It follows that

$$\text{Var}_{T_C - \lambda}(L_n) \geq n + 1. \quad \Box$$

Since for the languages $L_n$, $n \geq 2$, in Lemma 2,

$$L_n \in \mathcal{L}_{n+1}(T_C - \lambda) - \mathcal{L}_n(T_C - \lambda),$$

we have the strict inclusions in Lemma 1 for the case $T_C - \lambda$, i.e.,

**Theorem 3.**

$$\mathcal{L}_2(T_C - \lambda) \subset \mathcal{L}_3(T_C - \lambda) \subset \cdots \subset \mathcal{L}_n(T_C - \lambda) \subset \cdots .$$

(4)

The following example shows that not more than four nonterminals are enough to generate the languages $L_n$, $n \geq 1$, in Lemma 2 by tree controlled grammars with erasing rules.

**Example 2.** For $n \geq 1$, let $H''_n = (\{S, A, C\}, \{a_1, a_2, \ldots, a_n\}, P''_n, S, R''_n)$ be a tree controlled grammar where

$$P''_n = \{ S \rightarrow A'a_A : 1 \leq i \leq n \} \cup \{ A \rightarrow a_A : 1 \leq i \leq n \} \cup \{ A \rightarrow A, A \rightarrow C, C \rightarrow \lambda \}.$$

$$R''_n = \{ S, C \} \cup \{ A'a_A : 1 \leq i \leq n \} \cup \{ C'A : 1 \leq i \leq n \}.$$

The word $A'a_A$, $1 \leq i \leq n$, at a level of the derivation tree requires that if the rule $A \rightarrow a_A$ is applied then it has to be applied for the last occurrence of $A$ and the chain rule for the rest occurrences of $A$ in a sentential form. If $A \rightarrow C$ is applied then all occurrences of $A$ in the sentential form except the last one have to be replaced with $C$'s. Further all occurrences of $C$ are erased by $C \rightarrow \lambda$ and $A$ is replaced with $C$ which is also erased in the last step of the derivation, i.e.,

$$S \Rightarrow A'a_A \Rightarrow^* A'd'_A \Rightarrow^* C'd'_A \Rightarrow^* d'_C \Rightarrow d'_i.$$

Therefore, $L(H''_n) = L_n$ and $\text{Var}(H''_n) = 4$ for all $n \geq 1. \quad \Box$

This example shows that the use of erasing rules may lead to the decrease of the number of nonterminals in tree controlled grammars.

**4. Nine nonterminals are sufficient for $\mathcal{L}(T_C)$**

In this section, we prove that a fixed number of nonterminals is sufficient to generate all recursively enumerable languages.

In [11], it was proven that every recursively enumerable language is generated by a grammar $G = (\{S, S', A, B\}, T, P \cup \{ABBB \rightarrow \lambda\}, S')$, where $P$ consists of context-free productions of the forms

(1') $S' \rightarrow uS'a$, where $u \in \{AB, ABB\}^\ast, a \in T$,

(2') $S' \rightarrow S$,

(3') $S \rightarrow uSv$, where $u \in \{AB, ABB\}^\ast, v \in \{BA, BBA\}^\ast$,

(4') $S \rightarrow uv$, where $u \in \{AB, ABB\}^\ast, v \in \{BA, BBA\}^\ast$.

With respect to the rules above, the derivation of a word $w \in T^\ast$ can be divided into the following phases:

(a) $S' \Rightarrow^* w'S'w \Rightarrow w'Sw$, where $w' \in \{AB, ABB\}^\ast$ and $w \in T^\ast$, by rules of the form $S' \rightarrow uS'a$ and $S' \rightarrow S$, where $u \in \{AB, ABB\}^\ast$ and $a \in T$.

(b) $w'Sw \Rightarrow^* w_1w_2w$, where $w_1 \in \{AB, ABB\}^\ast$ and $w_2 \in \{BA, BBA\}^\ast$, by rules of the form $S \rightarrow uSv$ and $S \rightarrow uv$, where $u \in \{AB, ABB\}^\ast, v \in \{BA, BBA\}^\ast$.

(c) $w_1w_2w \Rightarrow^* w$ by $ABBB \rightarrow \lambda$.

**Theorem 4.** Every recursively enumerable language is generated by a tree controlled grammar with no more than nine nonterminals.

**Proof.** Let $L$ be a recursively enumerable language. Then, there is a grammar

$$G = (\{S, S', A, B\}, T, P \cup \{ABBB \rightarrow \lambda\}, S')$$

in the Geffert normal form such that $L(G) = L$.

We define the tree controlled grammar $G' = (N', T, P \cup P', S', R')$ where

$$N' = \{S, S', A, B, A', B'\},$$

$$P' = \{ A \rightarrow A, A \rightarrow A', A' \rightarrow \lambda, B \rightarrow B, B \rightarrow B', B' \rightarrow \lambda \},$$

$$R' = (\{S, S', A, B\} \cup T)^\ast \cup \{A'B'B'A', A, B\}^\ast.$$
We first prove that $L(G) \subseteq L(G')$. Without loss of generality, let a word $x_1x_2\cdots x_n \in L(G)$, $x_1, x_2, \ldots, x_n \in T$, $n \geq 1$, be obtained by a derivation according to the phases (a), (b) and (c) given before this theorem. We simulate this derivation in $G'$ by the additional construction of a corresponding derivation tree where the words of all levels – except the last one – belong to $R'$.

The derivation starts with $S' \Rightarrow u_nS'x_n$ for some $u_n \in \{AB, ABB\}^*$. Therefore, $u_nS'x_n$ is the sentential form and the word of the first level.

Let us now assume that we have derived the sentential form

$$u_nu_{n-1} \cdots u_2S'x_n + 1 \cdots x_n$$

where $u_n$, $u_{n-1}$, \ldots, $u_1 \in \{AB, ABB\}^*$

and the word of the corresponding level in the constructed tree is

$$u_nu_{n-1} \cdots u_2S'x_n,$$

which belongs to $R'$. If we apply the rule $S' \rightarrow u_{n-1}S'x_{n-1}$, where $u_{n-1} \in \{AB, ABB\}^*$, and the chain rules $A \rightarrow A$ and $B \rightarrow B$ to all other occurrences of nonterminals $A$ and $B$, then we get the sentential form

$$u_nu_{n-1} \cdots u_2S'x_{n-1}x_n$$

and the word $u_nu_{n-1} \cdots u_2S'x_{n-1}x_n$ forms the next level of the derivation tree and is in $R'$, too. Phase (a) finishes by the application of the rule $S' \rightarrow S$ and the chain rules $A \rightarrow A$ and $B \rightarrow B$ to the sentential form $u_nu_{n-1} \cdots u_2S'x_1 \cdots x_{n-1}x_n$, and as a result the sentential form

$$u_nu_{n-1} \cdots u_2Sx_1 \cdots x_{n-1}x_n$$

and the word $u_nu_{n-1} \cdots u_2S \in R'$ at the associated level of the derivation tree are obtained.

Analogously, we can simulate the derivation step

$$u_nu_{n-1} \cdots u_2u'_{m-1}u'_ku'_kv_{k+1}v_{k+2} \cdots v_m \Rightarrow u_nu_{n-1} \cdots u_2u'_{m-1}u'_ku'_kv_{k+1}v_{k+2} \cdots v_m x_1x_2 \cdots x_n$$

in phase (b) in such a way that the word

$$u_nu_{n-1} \cdots u_2u'_{m-1}u'_ku'_kv_{k+1}v_{k+2} \cdots v_m \in R'$$

in the associated level of the derivation tree changes to

$$u_nu_{n-1} \cdots u_2u'_{m-1}u'_ku'_kv_{k+1}v_{k+2} \cdots v_m \in R'$$

in the next level. Phase (b) finishes by the application of the rule of the form $S \rightarrow u'_{1}v_{1}$, where $u'_{1} \in \{AB, ABB\}^*$ and $v_{1} \in \{BA, BBA\}^*$, and the chain rules $A \rightarrow A$ and $B \rightarrow B$ obtaining the sentential form $u_nu_{n-1} \cdots u_2u'_{m-1}u'_ku'_kv_{k+1}v_{k+2} \cdots v_m x_1x_2 \cdots x_n$.

and the word in the associated level of the derivation tree

$$u_nu_{n-1} \cdots u_2u'_{m-1}u'_ku'_kv_{k+1}v_{k+2} \cdots v_m \in R'.$$

If $AABBBA \rightarrow \lambda$ is applied in a derivation step of phase (c) to the sentential form $z$, then $z = uABBBAux_1x_2 \cdots x_n$ for some words $u \in \{AB, ABB\}^*$ and $v \in \{BA, BBA\}^*$, and the next derived word is $z' = uux_1x_2 \cdots x_n$. Moreover, the word in the corresponding level is $uABBBAuv \in R'$. We now proceed on $z$ in $G'$ as follows: first, we apply $A \rightarrow A'$ and $B \rightarrow B'$ to the distinguished occurrences of $A$ and $B$, and the chain rules $A \rightarrow A$ and $B \rightarrow B$ to all remaining occurrences of nonterminals $A$ and $B$, and the word $uABBBAuv \in R'$ in the next level. Next, we apply $A' \rightarrow \lambda$ and $B' \rightarrow \lambda$, and the chain rules $A \rightarrow A$ and $B \rightarrow B$, which give the sentential form $z'$ and the word $uv \in R'$ in the next level.

Thus, all the derivation phases in the grammar $G$ can also be simulated in the grammar $G'$, and we have shown that $x_1x_2 \cdots x_n$ is in $L(G')$. Analogously, we can show that $\lambda \in L(G)$ implies $\lambda \in L(G')$. Thus the desired relation $L(G) \subseteq L(G')$ holds.

In order to prove the converse inclusion, we consider a derivation of the terminal word $w = x_1x_2 \cdots x_n$, $x_i \in T$, $1 \leq i \leq n$, according to $P \cup P'$ such that the words of all levels of the corresponding derivation tree belong to $R'$. Let $w_0, w_1, \ldots, w_m$ be the words in the levels of this derivation tree. With each level $i$ we associate a sentential form $z_i$ such that, for all $i$, the sentential form $z_{i+1}$ is obtained from $z_i$ by the application of all the rules to all occurrences of nonterminals used in $i$th level of the derivation tree. Without loss of generality we can assume that $w_i \neq w_{i+1}$ for $0 \leq i \leq m - 1$.

Obviously, $w_0 = z_0 = S'$ and $w_1 = z_1 = u_nS'x_n$ for some word $u_n \in \{AB, ABB\}^*$ and some $x_n \in T$. Clearly, $S' \Rightarrow u_nS'x_n$ also holds in $G$. We distinguish some cases.

Case 1. $w_k = uS'x_k$ and $z_k = uS'x_{k+1} \cdots x_n$ with $u \in \{A, B\}^*$ (this situation holds for $k = 1$).

Since $S'$ introduces at least one occurrence of $A$ or a symbol from $T$, but no occurrence of a nonterminal from $\{S, A', B'\}$ can be introduced from $w_k$, the word $w_{k+1}$ has to belong to $((A, B, S') \cup T)^+$ by the structure of words in $R'$. Hence all other occurrences of nonterminals $A$ and $B$ in $z_k$ have to be replaced according to chain rules $A \rightarrow A$ and $B \rightarrow B$. If $S'$ is replaced by $uS'x_{k+1}$ with $u' \in \{AB, ABB\}^*$ or $S$ then we get

$$w_{k+1} = uu'S'x_{k+1} \quad \text{and} \quad z_{k+1} = uu'S'x_{k+1}x_{k+2} \cdots x_n$$

or

$s = 1, \quad w_{k+1} = uS' \quad \text{and} \quad z_{k+1} = uSx_1x_2 \cdots x_n$.  

S. Turaev et al. / Theoretical Computer Science 412 (2011) 5789–5795

5793
We mention that \( z_k \Rightarrow z_{k+1} \) is a derivation in \( G \) obtained by an application of \( S' \rightarrow u'S'x\}_{c-1} \) or \( S' \rightarrow S \).

Case 2. \( w_k = uSv \) and \( z_k = uSvx_1x_2 \cdots x_n \) with \( u, v \in \{ A, B \}^* \).

Then we can prove as above that all occurrences of \( A \) and \( B \) have to be replaced according to chain rules and we obtain

\[
 w_{k+1} = uu'Sv' \quad \text{and} \quad z_{k+1} = uu'Sv'x_1x_2 \cdots x_n
\]

by application of \( S \rightarrow u'Sv' \) or

\[
 w_{k+1} = u'u'v' \quad \text{and} \quad z_{k+1} = u'u'v'x_1x_2 \cdots x_n
\]

by application of \( S \rightarrow u'u'v' \). Again, we have that \( z_k \Rightarrow z_{k+1} \) also holds in \( G \) by an application of \( S \rightarrow u'Sv' \) or \( S \rightarrow u'u'v' \).

Case 3. \( w_k = uv \) and \( z_k = uvx_1 \cdots x_n \) with \( u, v \in \{ A, B \}^* \).

From \( uv \) we can only apply \( A \rightarrow A, B \rightarrow B, A \rightarrow A' \) or \( B \rightarrow B' \). Since \( w_{k+1} \neq w_k \), we have to apply at least one rule from \( \{ A \rightarrow A', B \rightarrow B' \} \). But in order to obtain a word \( w_{k+1} \in R' \), \( w_k \) has to be of the form \( w_k = u'ABBBAv' \) for some \( u', v' \in \{ A, B \}^* \), and the subword \( ABBBA \) in \( w_k \) has to be replaced by the subword \( A'B'B'B'A' \) using the rules \( A \rightarrow A' \) and \( B \rightarrow B' \), and all other occurrences of nonterminals by the chain rules. Thus we get

\[
 w_{k+1} = u'A'B'B'B'A'v' \quad \text{and} \quad z_{k+1} = u'A'B'B'B'A'v'x_1x_2 \cdots x_n.
\]

By analogous arguments, in order to ensure that \( w_{k+2} \) is in \( R' \), one of the following cases holds:

Case 3.1. We replace all \( A \)'s and \( B \)'s by the empty word and all other occurrences of nonterminals according to chain rules, which result in

\[
 w_{k+2} = u'u'v' \quad \text{and} \quad z_{k+2} = u'u'v'x_1x_2 \cdots x_n.
\]

In this case, \( z_k \Rightarrow z_{k+2} \) is a direct derivation in \( G \), obtained by the application of \( ABBBA \rightarrow \lambda \).

Case 3.2. We have

\[
 w_{k+1} = u''ABA'B'B'B'ABBBAv'' \quad \text{or} \quad w_{k+1} = u''ABBBA'B'B'B'BAuv''
\]

for some \( u'', v'' \in \{ A, B \}^* \), and we replace all \( A \)'s and \( B \)'s by the empty word, the distinguished occurrences of \( A \) and \( B \) above by \( A' \) and \( B' \), respectively, and all other occurrences of nonterminals according to chain rules, which give

\[
 w_{k+2} = u'A'B'B'B'v' \quad \text{and} \quad z_{k+2} = u'A'B'B'B'A'v'x_1x_2 \cdots x_n.
\]

Besides the priming of some letters, each derivation \( z_k \Rightarrow \^* z_{k+2} \) erases a subword \( ABBBA \). Moreover, if Case 3.2 holds, the type of the word and sentential form are reproduced. However, this form can only be reproduced as long as \( ABBBA \) is present in the word, i.e., after some steps Case 3.1 has to be used. Thus the derivation \( z_k \Rightarrow \^* z_l \) for some \( l \geq k + 3 \) where in the last step Case 3.1 holds, is a derivation in \( G \), too.

Thus, for any \( w \in L(G') \), we also have \( w \in L(G) \). Hence \( L(G') \subseteq L(G) \). Combining the two shown inclusions, we get \( L(G') = L(G) = L \).

Since the language \( R' \) can be generated by a regular grammar \( G'' \) with the nonterminal complexity three, i.e., \( \text{Var}(G'') = 3 \) defined by

\[
 G'' = \left( N'', T'', P'', S'' \right)
\]

where

\[
 N'' = \{ S'', S_1, S_2 \},
\]

\[
 T'' = \{ S, S', A, B, A', B' \} \cup T,
\]

\[
 P'' = \{ S'' \rightarrow S_1, S'' \rightarrow S_2 \},
\]

\[
 \cup \{ S_1 \rightarrow xS_1 : x \in \{ S, S', A, B \} \cup T \} \cup \{ S_1 \rightarrow \lambda \}
\]

\[
 \cup \{ S_2 \rightarrow xS_2 : x \in \{ A, B, A'B'B'B'A' \} \cup \{ S_2 \rightarrow \lambda \} \}.
\]

we have \( \text{Var}(G'') = 9 \) and, consequently, \( \text{Var}(L) \leq 9 \).

Since \( L_G(T^C) \subseteq L(R^C) \) is obvious, we get

**Corollary 5.** \( L(R^C) = L_G(T^C) \).

5. Conclusion

In this paper, we have studied the nonterminal complexity of tree controlled grammars. We have shown that in the case of tree controlled grammars without erasing rules we obtain an infinite hierarchy of language families with respect to the number of nonterminals (Theorem 3). On the other hand, every recursively enumerable language can be generated by a tree controlled grammar with erasing rules and no more than nine non terminals (Theorem 4), but the optimality of this number has still to be investigated.

It is also interesting to find bounds for other families of languages, e.g., context-free, E0L and ET0L languages, which may have bounds smaller than nine. We should mention that Lemma 2 already shows that there is no bound in the case without erasing rules if we restrict to regular languages.

Finally, we add some remarks on the used concept of descriptive complexity. The notation of a (context-free or phrase structure) grammar as a quadruple can be interpreted as a description by a word over a certain alphabet, and the length of
this word would be a very natural measure of desciptional complexity. In many papers, only a part of the word is taken as
the measure. One such “approximation” is given by the number of nonterminals. If we consider tree controlled grammars,
then the complete description requires the description of the underlying context-free grammar and a description of the
control device, i.e., the regular language. In this paper, we used for both parts the number of nonterminals as a complexity
measure. This differs from the notions in the papers [2,3,7,8] where only the number of nonterminals of the underlying
grammar is considered and the complexity of the control mechanism is ignored. This comes from the problem to measure
the interplay between the rules in a matrix or programmed or scattered context grammar, whereas for tree controlled
grammars any measure for the complexity of the regular control language is suitable. If we also would restrict to the number
of nonterminals of the underlying grammar, then we would get essentially the same results, but the necessary number to
generate all recursively enumerable languages would be six. Also here it remains open whether this bound is optimal.
Obviously, the problem of necessary resources can be asked for other approximations (e.g., number of productions) or
the length of the word describing tree controlled grammar as a eight-tuple (for both grammars a four-tuple). This remains
as a topic for further investigation.
We have used the number of nonterminals as a measure for both languages. Since the control has to check whether or
not a word of a level of the derivation tree belongs to a language, one can use an accepting device for the description of the
control language and not a grammar, which is a generating device. Therefore one can be interested in the state complexity
of the regular language, i.e., in the number of states of a minimal deterministic automaton which accepts the language. Then
the nonterminal/state complexity of a tree controlled is the sum of the nonterminals of the underlying context-free grammar
and the state complexity of the control language. Looking to our proofs we get the following statements: an infinite hierarchy
is obtained for non-erasing tree controlled grammars, again. On the other hand, every recursively enumerable language can
be obtained by a tree controlled grammar with nonterminal/state complexity at most fourteen.

Acknowledgements

We would like to thank the anonymous referees of an earlier version of our paper for their hints, especially for the
suggestion to use the Geffert normal form for recursively enumerable languages (which we have taken in this paper) which
led to a reduction of the number of nonterminals from fifteen to nine.

This work was supported by University Putra Malaysia via RUGS 05-01-10-0896RU.

References