On Growing Context-Sensitive Languages

Gerhard Buntrock* Krzysztof Loryś*†

Report No. 38 January, 1992

Abstract

Growing context-sensitive grammars (GCSG) are investigated. The variable membership problem for GCSG is shown to be NP-complete. This solves a problem posed in [DW86]. It is also shown that the languages generated by GCSG form an abstract family of languages and several implications are presented.

*Institut für Informatik, Universität Würzburg, D-8700 Würzburg, Germany.

†Instytut Informatyki, Uniwersytet Wrocławski, 51-151 Wrocław, Poland (permanent address). This research was supported by the Humboldt Foundation.

1 Introduction

It is well known that the class of languages generated by context-sensitive grammars is equal to NSPACE(n) and that, even for fixed grammars, the membership problem can be PSPACE-complete. On the other hand the context-free grammars are known to have, for many applications, too weak derivative power. While many modifications extending context-free grammars have been studied, only a few papers concern some restricted versions of context-sensitive grammars. Book ([Boo73]) investigated context-sensitive grammars with certain restrictions on the allowed context or on the use of rewriting rules and showed that some of them cause the grammars to generate only context-free languages. On the other hand when we put the restriction only on the length of derivations to be linear then the context-sensitive grammars are still very powerful, namely they can generate NP-complete languages ([Boo78]).

In [DW86] Dahlhaus and Warmuth considered grammars with only growing rules, i.e. rules whose right side is longer than the left side. This restriction naturally bounds the length of derivations by the length of derivated words. They showed that such growing grammars have the membership problem solvable in polynomial time and even contained in the class of problems logspace-reducible to context-free languages. We denote this class with LOGCFL and refer to [Sud78, Ven91]. With a careful examination of their proof one can get a stronger result: the membership problem for growing grammars is contained in 1-LOGCFL, the class of problems reducible to context-free languages via logspace reductions in which the input is read in one-way mode. The class 1-LOGCFL was introduced by Lautemann ([Lau88]) and by its automata characterization it is known to be a proper subclass of LOGCFL ([Bra77, Lau88]).

Dahlhaus and Warmuth have posed a problem to determine the complexity of the "variable" membership problem for growing grammars, i.e. the problem in which not only the tested word but also a growing grammar is a parameter. We solve this problem by proving (via generic reduction) its NP-completeness. The reduction is of low complexity, namely one-way log space. For such one-way reductions see [HIM78, All88].

Having a very elegant and simple definition growing grammars appear at the first look to generate not an interesting class of languages. In particular, this class (we denote it by GCSL) seemed to be not closed under such basic operations as intersection with regular sets or inverse homomorphism. We refute this opinion and show that GCSL forms an abstract family of languages. This allows us to give a characterization of GCSL by means of weighted grammars. With each such a grammar there is an associated function determining weights of all terminal and nonterminal symbols, and by a natural extension also weights of strings. If we restrict grammars to have only productions with the weight of the right side greater than the weight of the left side then they derive exactly GCSL. Since such weighted grammars allow derivations to have a linear length, it is often much easier to check a membership of a language in the class GCSL via constructing a weighted grammar than via constructing a growing one.

We believe that GCSL is an important class of languages and deserves to be thoroughly studied. With our paper we hope to increase an interest in it. In the last section we give some questions which seem to be worth of considerations.

2 Preliminaries

We assume the reader to be familiar with elementary formal language theory and computational complexity theory, and use the notations standard according to [HU79]. Here we recall only the notions which are basic for our purposes.

A grammar is a quadruple $\langle N, T, S, P\rangle$, where N and T are finite, disjoint alphabets of nonterminal and terminal (respectively) symbols, $S \in N$ is a start symbol and P is a finite set of productions of the form $\alpha \to \beta$ with $\alpha, \beta \in (N \cup T)^*$. The grammars with $|\alpha| \leq |\beta|$ for all productions from P are called *context-sensitive*.

By replacing the above inequality by a sharp one we obtain growing context-sensitive grammars.

Definition 1 *A context-sensitive grammar $G = \langle N, T, S, P\rangle$ is* growing *if S does not appear on the right side and $|\alpha| < |\beta|$ for any $(\alpha \to \beta), \alpha \neq S$ from P. We denote the growing grammars by* GCSG *and the class of languages generated by such grammars by* GCSL.

By a *growth ratio* of a production $\alpha \to \beta$ we mean the ratio $\frac{|\alpha|}{|\beta|}$. It is often convenient to deal with GCSG in the following form, which productions, excepting those with S on the left side, have a normalized growth ratio.

Definition 2 *We call a growing context-sensitive grammar $G = \langle N, T, P, S\rangle$ k-*normal *if all productions have one of the following form:*

(1) $S \to \beta$, where $\beta \in T^*$ and $|\beta| \leq k$

(2) $S \to \beta$, where $\beta \in (T \cup N)^k$

(3) $\alpha \to \beta$, where $\alpha, \beta \in (T \cup N)^*$, $|\alpha| = k$ and $|\beta| = k + 1$

Proposition 3 *For each language $L \in$ GCSL there exists a number $k_\circ$ such that for all $k \geq k_\circ$ there exists a k-normal growing context-sensitive grammar G_k such that $L = L(G_k)$.*

Proof. Let $G = (N, T, P, S)$ be a growing context-sensitive grammar. We set $k_\circ$ to be equal $\max\{|\alpha| : (\alpha \to \beta) \in P \text{ for some } \beta\}$.

The productions of G_k are divided into two subsets P_0 and P_1. To P_0 we put all productions $S \to \beta$ of the form 1 and 2 (from Definition 2 such that β can be derived from S in the grammar G. The subset P_1 we create in two stages. At the beginning we put into it all productions $\alpha \to \beta$ from P with $\alpha \neq S$, and then we normalize the length of the left and the right side of each of them.

- "Normalization of α": each $\alpha \to \beta$ in P_1 we replace by productions $\alpha_1\alpha\alpha_2 \to \alpha_1\beta\alpha_2$ for each $\alpha_1\alpha_2 \in (T \cup N)^*$ such that $|\alpha_1\alpha\alpha_2| = k$.

- "Normalization of β": each production $a_1 a_2 ... a_k \to b_1 b_2 ... b_k b_{k+1} ... b_{k+c}$ we replace by productions

$$\begin{array}{lcll} a_1 a_2 ... a_k & \to & b_1 B_2 ... B_{k+1}, & \\ B_i B_{i+1} ... B_{k+i-1} & \to & b_i B_{i+1} ... B_{k+i} & \text{for } i = 2, ..., c-1, \\ B_c ... B_{k+c-1} & \to & b_c ... b_{k+c}, & \end{array}$$

  where B_i (for $i = 2, ..., k + c - 1$) are unique new nonterminals.

It is clear that G_k and G are equivalent. □

The restriction imposed on the growth ratio to be greater than one for all productions in /GCSG/ implies that all derivations in these grammars are not longer than the derivated word. This requirement is relaxed in quasi-growing grammars.

Definition 4 *A grammar $G = \langle N, T, S, P\rangle$ is* quasi-growing *if there exists a function $f : (N \cup T) \rightarrow \mathsf{N}$ (here N stands for the set of positive integers) such that for all productions $(\alpha \rightarrow \beta) \in P$ it holds $f(\alpha) < f(\beta)$, where $f(a_1...a_n) = \sum_{i=1}^{n} f(a_i)$ for $a_1...a_n \in (N \cup T)^*$. The set of quasi growing grammars we denote by* QGG.

A similar concept of weight functions was already used in string rewriting ([Jan88]). It is easy to see that derivations in QGG have length only linearly depending on the length of derivated words. However, in Theorem 19 we show that quasi-growing grammars derive exactly GCSL.

For context-sensitive grammars there are some well known simple normal forms. The following form was proposed by Cremers ([Cre73]):

Definition 5 *A context-sensitive grammar is in* Cremers' normal form *if all productions are of the forms*

$$\begin{array}{rcl} AB & \longrightarrow & CD \\ A & \longrightarrow & CD \\ A & \longrightarrow & a \end{array}$$

where A, B, C, D are nonterminals and a is a terminal.

Obviously such a normal form is not growing.

From Theorem 19 it follows that for every QGG G there exists an equivalent grammar G' in Cremers' normal form; however, the size of G' can be much bigger (even exponentially) than the size of G. The next proposition shows that for a QGG with only length-nondecreasing productions such a normal form can be efficiently obtained.

Proposition 6 *For every* QGG *an equivalent quasi-growing grammar in Cremers' normal form can be constructed in polynomial time.*

Proof. Let $G = \langle N, T, S, P\rangle$ be a quasi-growing grammar with only length-nondecreasing productions and let f be a corresponding weight function. We can assume that G does not contain productions of the form $\alpha \rightarrow \beta$ with $|\alpha| = |\beta| = 1$. A grammar in Cremers' normal form we obtain by replacing all productions from P which are not in a good form by an equivalent sequence of productions.

Let suppose that $a_1...a_k \rightarrow b_1...b_l$ is a production from P, $k \geq 2$. We differentiate between two cases:

(1) $l \geq k + 2$

(2) $l = k$ or $l = k + 1$

In the first case we take the following sequence of productions:

$$\begin{array}{lcll} a_1a_2 & \rightarrow & dA_2 & \\ A_ia_{i+1} & \rightarrow & dA_{i+1} & \text{for } i = 2, ..., k-1 \\ A_k & \rightarrow & B_kB_{k+1} & \\ dB_i & \rightarrow & B_{i-1}b_i & \text{for } i = 2, ..., k-1 \\ dB_2 & \rightarrow & b_1b_2 & \\ B_i & \rightarrow & b_iB_{i+1} & \text{for } i = k+1, ..., l-2 \\ B_{l-1} & \rightarrow & b_{l-1}b_l & \end{array}$$

The symbols d, A_i (for $i = 2, ..., k$) and B_i (for $i = 2, ..., l-1$) are new nonterminals with the following weights:

$$\begin{array}{rcl} f(d) & = & 1 \\ \text{for } i = 2, ..., k: & & \\ f(A_i) & = & f(a_1 ... a_i) \\ f(B_i) & = & f(b_1 ... b_i) - 2(i-1) \\ \text{and for } i = k+1, ..., l-1: & & \\ f(B_i) & = & f(b_i ... b_l) - l + i \end{array}$$

Note that, without loss of generality, we can assume that $f(\alpha) \leq f(\beta) - (|\alpha| + |\beta| - 2)$ for each production $\alpha \to \beta$ from P. If f does not fulfill this condition then instead of f one can take the function $f' = c \cdot f$, where $c = max\{|\alpha| + |\beta| - 2 : \alpha \to \beta \in P\}$.

Now it is easy to check that all above productions are growing with respect to the function f.

For the second case a corresponding sequence of productions can be obtained in an analogous way, so we omit this construction as well as an easy proof of the equivalence of such obtained grammar and grammar G.

From this construction follows that the number of rules will be linear bounded in the length of the encoded grammar G. Hence we see that easily a polynomial time-bounded algorithm can be obtained. □

At the end of this section we recall the notion of *abstract family of languages*.

Definition 7 *A* family of languages *is a collection of languages containing at least one nonempty language. An* abstract family of languages *(*AFL*) is a family of languages closed under union, concatenation, iteration, intersection with regular languages, ε-free homomorphism and inverse homomorphism.*

It is well known that the "AFL operations" are not independent. In Section 4 we will make use of the following proposition (see [GGH69]).

Proposition 8 *Let $\mathcal{F}$ be a family of languages. If $\mathcal{F}$ is ε-free and closed under concatenation, ε-free homomorphism and inverse homomorphism, then $\mathcal{F}$ is closed under intersection with regular sets.*

3 The complexity of the membership problem for variable grammars

The variable membership problem consists in determining, for given a word w and a grammar G, whether $w \in L(G)$. It is known that for context-sensitive grammars this problem is PSPACE-complete. The open problem posed in [DW86] was to determine its complexity for growing context-sensitive grammars; more precisely, Dahlhaus and Warmuth asked whether this problem can be solved in polynomial time or whether it is NP-complete. The polynomial time algorithm, given by them, for the membership problem with fixed grammars, adapted for the problem with variable grammars, would work in superpolynomial time because the degree of the polynomial bounding its complexity is dependent on the grammar and can be arbitrary large. We give here a strong argument that this problem presumably is intractable, showing that it is NP-complete under a weak type of reducibility, namely one-way logspace (see [HIM78, All88] for more information about this type of reducibility).

Let $\bar{G}$ denote a word coding a grammar G. Our problem consists in determining the complexity of the language

$$L_{GCSL} = \{\langle \bar{G}, w\rangle : G \text{ is a growing context-sensitive grammar and } w \in L(G)\}.$$

As a function coding grammars can be taken any reasonable one, we will not specify it. The same concerns also a function coding configurations of Turing machines that will be used later in the proof of NP-hardness of the language L_{GCSL}.

Proposition 9 $L_{GCSL} \in$ NP.

Proof. Let $\langle \bar{G}, w\rangle \in L_{GCSL}$ and let c be the maximal length of the code of a single symbol from G. Clearly, $c < |\bar{G}|$. Since w is derived in no more than $|w|$ steps and the length of the longest phrase in any derivation of w is bounded by $c|w| < |\bar{G}| \cdot |w|$, it is easy to construct a nondeterministic algorithm which simulates derivations in G in time $|w|^2 \cdot |\bar{G}|^2 < |\langle \bar{G}, w\rangle|^4$. □

Now we show how, for each polynomial-time bounded nondeterministic Turing machine M, one can construct growing context-sensitive grammars $G_{w,M}$ such that each $L(G_{w,M})$ is over one-letter alphabet and it is nonempty if and only if M accepts w. Moreover, in this case the length of the shortest word in $L(G_{w,M})$ is bounded by a polynomial in $|w|$.

Let T_1 be an alphabet in which configurations of Turing machines are coded. We assume that $\#, [,] \notin T_1$. For simplicity, from now on, we do not distinguish between objects like configurations and productions and their codes. We define

$$\begin{aligned} Seq_{w,M} = \{ \; & [l_1\#k_2][l_2^R\#k_3^R][l_3\#k_4][l_4^R\#k_5]...[l_{p-1}^R\#k_{p-1}^R][l_{p-1}\#k_p] : \\ & \forall_{2\le i\le p-1}\ k_i \text{ and } l_i \text{ are } M\text{'s configurations such that } l_i \text{ is a successor of } k_i, \\ & l_1 \text{ is the initial configuration of } M \text{ when } w \text{ is an input, } k_p \text{ is a predeces-} \\ & \text{sor of an accepting configuration and } p \text{ is an even natural number } \}.^1 \end{aligned}$$

The following proposition can be easily seen.

Proposition 10 *For any Turing machine M and a word w the language $Seq_{w,M}$ can be generated by a context-free grammar $F_{w,M} = \langle N_1, T_1 \cup \{\#, [,]\}, S, P_{w,M}\rangle$. Moreover, for any fixed M there exists a GSM [2] which having w as an input produces a code of $F_{w,M}$.*

A key part of the grammar $G_{w,M}$ is a subset H_k ($k = |w|$) of its productions which, roughly speaking, allows checking whether corresponding subwords are equal. More precisely, starting from a word $[l_1\#k_2][l_2^R\#k_3^R][l_3\#k_4][l_4^R\#k_5]...[l_{p-1}^R\#k_{p-1}^R][l_{p-1}\#k_p] \in Seq_{w,M}$ and using productions from H_k one can derive a terminal word if and only if for all i with $1 \le i < p$ it holds $l_i = k_{i+1}$. Note that H_k is common for all grammars $G_{w,M}$ with $|w| = k$.

Let $N_2 = T_1 \cup \bar{T_1} \cup \{[,], \#, C\}$ be an alphabet of nonterminal symbols such that $\bar{T_1} = \{\bar{a} : a \in T_1\}$ and let T_2, a terminal alphabet, consist of only one letter d. For every

[1] By l^R we denote the reversal of the string l.

[2] General sequential machine, i.e. a deterministic finite one-way automaton with output, see[HU79] page 272 for an exact definition

$a, b \in T_1$, to the system H_k belong the following productions:

$$
\begin{array}{lrll}
(1) & [a \rightarrow & [C\bar{a} & \\
(2) & C\bar{a}d^ib \rightarrow & d^{i+1}bC\bar{a}, & \forall_{0\leq i<k} \\
(3) & C\bar{a}\#d^{4i}a \rightarrow & \#d^{4(i+1)}, & \forall_{0\leq i<k} \\
(4) & [d^i \rightarrow & d^{i+1}[, & \forall_{0\leq i<k} \\
(5) & [\# \rightarrow & dd[& \\
(6) & [\,] \rightarrow & ddd & \\
(7) & d \rightarrow & dd. &
\end{array}
$$

Proposition 11 *H_k consists of $k(r^2+r+1)+r+3$ productions, where $r = |T_1|$.*

Proposition 12 *Let $u = [u_1\#v_1][u_2\#v_2]\ ...\ [u_s\#v_s]$, $u_i, v_i \in T_1^*$ for $i = 1, ..., s$. If a terminal word can be derived from u using production from H_k then for all i with $1 \leq i \leq s$ it holds $u_i = v_i$.*

Proof.(sketch) The only way of removing # and [from a phrase, being derived, is to apply the productions (5) and (6), respectively. Before this can be done, each letter of u_i has to be changed via production (1) into its counterpart from T_2 and then transported via production (2) to #, where it is compared with the corresponding letter of v_i. Since there is no production changing the order of transported letters it is clear that $u_i = v_i$. □

Proposition 13 *From each word $[w\#w]$, ($w \in T_1^*, |w| \leq k$), system H_k derives d^l, for any $l \geq \frac{(|w|+9)(|w|+2)}{2}$.*

Proof. Let $w = a_1a_2...a_n$, $n \leq k$. One can easily show by induction that for all i, $1 \leq i \leq n$

$$[w\#w] \stackrel{*}{\Longrightarrow} d^{\frac{i^2+i-2}{2}}[a_id^{i-1}a_{i+1}d^{i-1}...d^{i-1}a_n\#d^{4(i-1)}a_ia_{i+1}...a_n].$$

Thus finally

$$[w\#w] \stackrel{*}{\Longrightarrow} d^{\frac{n^2+3n}{2}}[\#d^{4n}] \stackrel{*}{\Longrightarrow} d^{\frac{n^2+3n}{2}+2}[d^{4n}] \stackrel{*}{\Longrightarrow} d^{\frac{(n+9)(n+2)}{2}}.$$

Now, for $l > \frac{(n+9)(n+2)}{2}$, the words d^l can be derived by use of production (7). □

Theorem 14 *The variable membership problem for growing context-sensitive Grammars i.e. the language L_{GCSL}, is* NP*-complete under one-way logspace reducibility.*

Proof. By Proposition 9 it is sufficient to show that L_{GCSL} is one-way logspace hard for NP.

Let M be a nondeterministic Turing machine working in time $p(n)$. Without loss of generality we can assume that there exists a polynomial $p'(n)$ which bounds the length of codes of the configurations entered by M within $p(n)$ steps. Let N_1, S and $P_{w,M}$ be from $F_{w,M}$ (see Proposition 10) and let N_2 and $H_p(|w|)$ be as described above. We set

$$G_{w,M} = \langle N_1 \cup N_2, \{d\}, S, P_{w,M} \cup H_{p(|w|)}\rangle$$

and

$$t_M(w) = \langle G_{w,M}, d^{(p'(|w|)+9)(p'(|w|)+2)p(|w|)}\rangle.$$

All productions from $H_{p(|w|)}$ are polynomially bounded in length and have very simple form. This, together with Proposition 10, gives that t_M can be constructed to be a one-way logspace transducer.

Since all productions from $H_{p(|w|)}$ contain no symbol from N_1 and $P_{w,M}$ consists of context-free rules, each derivation in $G_{w,M}$ can be transformed into an equivalent one in which no production from $H_{p(|w|)}$ is applied before any production from $P_{w,M}$. From this and Proposition 13 it is easy to see that $w \in L(M) \Leftrightarrow L(G_{w,M})$ is nonempty $\Leftrightarrow d^{(p'(|w|)+9)(p'(|w|)+2)p(|w|)} \in L(G_{w,M})$. $\square$

Let us note that the languages $G_{w,M}$ can be changed so they would be either empty or would contain at most one word from $\{d\}^*$. Thus the variable membership problem for GCSG remains NP-complete even for some restricted subclasses of GCSL.

4 Closure properties

In this section we show that the class GCSL is closed under all six AFL operations. A nontrivial part of the proof (Proposition 17) concerns the closure under inverse homomorphism. As a consequence we obtain a characterization of GCSL by quasi-growing grammars (Theorem 19).

Proposition 15 *The class* GCSL *is closed under union, concatenation, iteration, substitution and ε-free homomorphism.*

Proof. By standard methods using the fact that for any language $L \in$ GCSL there exists a growing context-sensitive grammar G generating L such that terminal symbols appear only on the right side of productions. $\square$

To prove that GCSL is an AFL it is now sufficient, by Propositions 15 and 8, to show that it is closed under inverse homomorphism.

We start with defining some notions which will be useful in the proof of this fact. Following [Loe70, DW86] with each derivation in $G = \langle N, T, S, P\rangle$ we associate a planar acyclic graph called a *derivation graph*. The vertices in such a graph are labeled by corresponding symbols and productions used in the derivation. The label of a vertex v we denote by $l(v)$. For a derivation $\alpha_1 \overset{*}{\underset{G}{\Longrightarrow}} \alpha_2 \overset{*}{\underset{G}{\Longrightarrow}} \dots \overset{*}{\underset{G}{\Longrightarrow}} \alpha_k$ the derivation graph $D_k = (V_k, E_k)$ is defined inductively. At each step the sources of V_i correspond to α_1 and sinks correspond to α_i.

(1) if $\alpha_1 = a_1a_2...a_r$ then $D_1 = (\{v_j : j = 1, ..., r \text{ and } l(v_j) = a_j\}, \emptyset)$.

(2) if $\alpha_{i-1} = w\alpha w' \Longrightarrow w\beta w' = \alpha_i$ and $\alpha = a_1a_2...a_r$, $\beta = b_1b_2...b_s$ then

- $V_i = V_{i-1} \cup \{v^i, v^i_1, ..., v^i_s\}$, where v^i and v^i_j are new vertices, v^i is labeled by the production $(\alpha \to \beta)$ and $l(v^i_j) = b_j$ for $j = 1, ..., s$,
- $E_i = E_{i-1} \cup \{(v^i, v^i_j) : j = 1, ..., s\} \cup \{(u_j, v^i)\colon j = 1, ..., s\}$ where $u_1, u_2, ..., u_r \in V_{i-1}$ correspond to α.

Let D be a derivation graph corresponding to $z \overset{*}{\underset{G}{\Longrightarrow}} u$. We say that a subgraph $F \subseteq D$ is *autonomous* if:

(1) considered as an undirected graph F is connected,

(2) all predecessors of F's vertices also belong to F,

(3) all sinks of F are labeled by productions.

Let $v_1, ..., v_s$ be the sequence of all sources of F and let $v'_1, ..., v'_{s'}$ be the sequence of all successors of F's sinks (remind that there is a natural order among the sources and among the sinks due to planarity of D). The word $l(v_1)...l(v_s)$ we call *a base* of F and the word $l(v'_1)...l(v'_{s'})$ we call *a derivative* of F. It is easy to see that the base is a subword of z and that $D \setminus F$ is a derivation graph of some derivation $z' \overset{*}{\underset{G}{\Longrightarrow}} u$, where z' is created by replacing the base with the derivative.

If v is a node labeled by a production then by $I(v)$ we denote the subgraph of D induced by v and the set of its ancestors. It is easy to see that $I(v)$ is autonomous for any v.

By *a size of autonomous subgraph* we mean the number of its vertices labeled by productions.

Proposition 16 *Let D be a derivation graph for some derivation in k-normal growing context-sensitive grammar.*

(1) For any vertex $v \in D$ the subgraph $I(v)$ has the base of length not greater that $(2h-1) \cdot k - 2(h-1)$, where h is the depth of v.

(2) One of the following conditions is fulfilled:

- *(a) there exists an autonomous subgraph F in D of the size at least k and the base not longer than $4k(k-1)$*
- *(b) all autonomous subgraphs of D have size smaller than k.*

Proof. We omit an easy proof of the part *(1)*.

To prove *(2)* let us suppose that there exists a subgraph $F \subseteq D$ of the size at least k and the base longer than $4k(k-1)$. If F contains a vertex v of the depth k then $I(v)$ satisfies the conditions of the point 1. So let us suppose that all vertices of F have depth not greater than $k-1$. We say that a sink of F is maximal if none of its successors belong to F. Let $v_1, ..., v_r$ be succeeding maximal sinks of F. Let now $F'_i = \bigcup_{j=1}^{i} I(v_j)$. From (i) it follows that the first F'_i of the size at least k has the base not longer than $4k(k-1)$. □

Proposition 17 *The class* GCSL *is closed under inverse homomorphism.*

Proof. Let $L \subseteq \Sigma^*$ be a growing context-sensitive language, let $h : \Delta^* \to \Sigma^*$ and let $l = \max\{|h(a)|: a \in \Sigma\}$.
Since GCSL is closed under substitution it is sufficient to prove the theorem for ε-free h. Namely, if $\Delta_1 = \{a \in \Delta : h(a) = \varepsilon\}$ and $\Delta_2 = \Delta \setminus \Delta_1$ are not empty, then $h_1 = h|_{\Delta_2}$ is ε-free and $h^{-1}(L) = S(h_1^{-1}(L))$, where S is a substitution such that $S(a) = \Delta_1^* a \Delta_1^*$ for each $a \in \Delta_2$.

By Proposition 3 there exists a k-normal grammar $G = \langle N, \Sigma, P, S\rangle$ generating L for some $k > 2l$. Now we define a new growing context-sensitive grammar $G' = \langle N', \Delta, P', S'\rangle$ and then we show that $L(G') = h^{-1}(L)$.

We set $N' = N_1 \cup N_2$, where $N_1 = (N \cup \Sigma)^k$ and $N_2 = \bigcup_{i=1}^{l-1} \bigcup_{j=1}^{k-1} \Sigma^i \times (N \cup \Sigma)^j$. The set P' of productions is divided into four subsets P_i for $i = 0, .., 3$. In the description of these subsets we use the following natural projection from N' onto N^*: $C(\langle a_1, a_2, ..., a_j\rangle) = a_1 a_2 ... a_j$ for each $\langle a_1, a_2, ..., a_j\rangle \in N'$.

Now we define

$$
\begin{array}{ll}
P_0 = & \{S' \to w : w \in \Delta^*, |h(w)| \leq 4k(k-1) \text{ and } S \overset{*}{\underset{G'}{\Longrightarrow}} h(w)\}, \\
P_1 = & \{S' \to \beta : \beta \in (N')^{4(k-1)} \text{ and } S \overset{*}{\underset{G'}{\Longrightarrow}} C(\beta)\}, \\
P_2 = & \{\alpha \to \beta : \alpha \in (N')^{4(k-1)}, \beta \in (N')^{4(k-1)+1} \text{ and } C(\alpha) \overset{*}{\underset{G'}{\Longrightarrow}} C(\beta)\} \\
\text{and} & \\
P_3 = & \{t_1\alpha \to wt_2 : t_1, t_2 \in N_2 \cup \{\varepsilon\}, \alpha \in (N')^m, m \leq 4(k-1), w \in \Delta^* \\
& \text{and } h(w)C(t_2) \text{ is derivable in } G \text{ from } C(t_1\alpha) \text{ in less than } k \\
& \text{steps. }\}
\end{array}
$$

Let us note that $|wt_2| > |t_1\alpha|$ for all $(t_1\alpha \to wt_2) \in P_3$, so G' is growing.

One can easily prove by induction on the length of derivation that all phrases derived from S' have form $\alpha_1 w_1 \alpha_2 w_2 ... w_{s-1}\alpha_s$, where all α's are from $(N')^*$, all w's are from Δ^* and $S \overset{*}{\underset{G'}{\Longrightarrow}} C(\alpha_1)h(w_1)C(\alpha_2)h(w_2)...h(w_{s-1})C(\alpha_s)$.

Therefore $w \in L(G') \Rightarrow \exists_{u \in L} h(w) = u$.

Thus it remains to prove that $h^{-1}(L) \subseteq L(G')$.

Let $w \in h^{-1}(L)$ and let $u \in L$ be such that $h(w) = u$. If $|u| \leq 4k(k-1)$ then w is derived by a production from P_0. So let assume that $|u| > 4k(k-1)$ and let u_1 be such that $S \overset{*}{\underset{G}{\Longrightarrow}} u_1 \overset{*}{\underset{G}{\Longrightarrow}} u$ and $|u_1| = 4k(k-1)$. Of course, there exists $z_1 \in (N')^{4(k-1)}$ such that $C(z_1) = u_1$, so $S' \to z_1$ is in P_1. Let F be a derivation graph for $u_1 \overset{*}{\underset{G}{\Longrightarrow}} u$. If F contains an autonomous subgraph of the size at least k (by Proposition 16 we can assume that its base is not longer than $4k(k-1)$), then repeating an appropriate number of times an operation of removing a maximal sink from the subgraph we can obtain a subgraph $F' \subseteq F$ of the size k. Clearly, F' corresponds to a sequence of k productions which applied to the base of F derive from the word u_1 a word u_2 such that $u_2 \overset{*}{\underset{G}{\Longrightarrow}} u$. By the definition of the set P_2 we have that $z_1 \underset{G'}{\Longrightarrow} z_2$, where $C(z_2) = u_2$. Applying this consideration recursively to u_2 and to the graph $F \setminus F'$ we can finally show that there exist words $u_x \in (N \cup \Sigma)^*$ and $z_x \in (N')^*$ such that

(1) $S' \overset{*}{\underset{G'}{\Longrightarrow}} z_x$,

(2) $C(z_x) = u_x$,

(3) $u_x \overset{*}{\underset{G}{\Longrightarrow}} u$ and all autonomous subgraphs of the corresponding derivation graph H are of size $< k$.

Since the autonomous subgraphs of H have basis not longer than $k + (k-1)(k-2)$, the words u_x and u can be divided into subwords $u_x^1, ..., u_x^r$ and $u_1, ..., u_r$, respectively, such that $\forall_{1 \leq i \leq r} k \leq |u_x^i| \leq 2k + (k-1)(k-2)$ and u_i is derived from u_x^i in less than k steps. Now for each $i = 1, ..., r$ we define $w_i, s_i, z_{j_i}, z'_{j_i}, z''_{j_i}, t_i$ as follows:

- w_i is the maximal prefix of w such that $h(w_i)$ is a prefix of $u_1 u_2 ... u_i$,
- s_i is the suffix of $u_1 u_2 ... u_i$ such that $u_1 u_2 ... u_i = h(w_i)s_i$,
- if $z_x = z_1 z_2 ... z_y$, where $z_1, ..., z_y \in N'$, then $C(z_{j_i}) = z'_{j_i} z''_{j_i}$ and $u_x^1 ... u_x^i = C(z_1 ... z_{j_i - 1})z'_{j_i}$.
- $t_i \in N_2 \cup \{\varepsilon\}$ is such that $C(t_i) = s_i z''_{j_i}$ (note that $t_r = \varepsilon$).

One can observe that the following derivation is possible due to productions from P_3:

$$z_x \underset{G'}{\Longrightarrow} w_1 t_1 z_{j_1+1} ... z_y \underset{G'}{\Longrightarrow} w_1 w_2 t_2 z_{j_2+1} ... z_y \overset{*}{\underset{G'}{\Longrightarrow}} w_1 w_2 ... w_r = w$$

□

Now we are ready to state the main theorems of this section:

Theorem 18 *The class* GCSL *forms an* AFL.

Proof. By the propositions 8, 15, and 17 GCSL is closed under all six AFL operations. □

Theorem 19 *The languages producible by* QGG*s and* GCSG*s are the same.*

Proof. Each GCSG can be seen as a QGG simply with a constant weight function.

Let L be generated by a quasi-growing grammar $G = \langle N, T, S, P\rangle$ with weight function f. Without loss of generality we can assume that S does not appear on the right side of any production and that $f(S) = 1$.

Let $c \notin N \cup T$ and let h be a homomorphism such that $h(a) = ac^{f(a)-1}$ for each $a \in N \cup T$. Then $G' = \langle N \cup \{c\}, T, S, P'\rangle$, where $P' = \{h(\alpha) \to h(\beta) : (\alpha \to \beta) \in P\}$ is a growing context-sensitive grammar and $L(G) = h^{-1}(L(G'))$. □

5 Conclusions and open problems

A disadvantage of the class of growing context-sensitive languages is that, being a subset of linear context-sensitive languages, it does not contain some very simple languages like $\tilde{L} := \{ww^Rw : w \in \Sigma^*\}$ ([Gla64]). It would be interesting to find a natural extension of context-sensitive grammars which on the one hand would allow to derive such languages but on the other hand would not lead the membership problem outside LOGCFL.

Another interesting problem concerns the intersection hierarchy for GCSL, i.e. the hierarchy of classes $\mathrm{GCSL}^i = \{L : \exists_{L_1, L_2, \ldots, L_i \in \mathrm{GCSL}}\ L = \bigcap_{j=1}^{i} L_j\}$, for $i \geq 1$. It is easy to see that the language $\tilde{L}$, defined above, separates the first two levels of the hierarchy and that $\bigcup_{j=1}^{\infty} \mathrm{GCSL}_j \subseteq \mathrm{LOGCFL}$. An analogous hierarchy for context-free languages was shown to be infinite ([LW73]). We conjecture that it is also infinite for GCSL.

The complexity of GCSL is still not well examined. In contrary to the context-free languages GCSL contain nonregular tally languages. What is their complexity? Can one show that tally languages from GCSL are included in NSPACE($\log n$)? Also the derivative power of GCSG is not exactly established. Even some very low complexity classes are not known to be included in GCSL. An example of such a class can be the class of languages recognized by one-way deterministic Turing machines in real-time and logarithmic space.

Acknowledgements

Thanks are due to Hans-Jörg Burtschick, Jaime Caro, Diana Emme, Ulrich Hertrampf, Albrecht Hoene, Klaus-Jörn Lange, Heribert Vollmer, and Klaus Wagner for some hints and fruitful discussions on the subject of this paper.

We are gratefull also to Manfred Warmuth who brought to our attention that the variable membership problem was already solved in two technical reports [NK87, CH89].

References

[All88] Eric W. Allender. Isomorphisms and 1-L reductions. *Journal of Computer and System Sciences*, 36:336–350, 1988.

[Boo73] Ronald V. Book. On the structure of context-sensitive grammar. *International Journal of Computer and Information Sciences*, 2:129–139, 1973.

[Boo78] Ronald V. Book. On the complexity of formal grammars. *Acta Informatica*, 9:171–181, 1978.

[Bra77] Franz-Josef Brandenburg. On one-way auxiliary pushdown automata. In *3rd GI Conference on theoretical computer science*, volume 48 of LNCS, pages 132–144. Springer, 1977.

[CH89] Sang Cho and Dung T. Huynh. Uniform membership for deterministic growing context-sensitive grammars is P-complete. Technical Report UTDCS-5-89, Computer Science Department, University of Texas at Dallas, Richardson, Texas 75083, February 1989.

[Cre73] A.B. Cremers. Normal forms for context-sensitive grammars. *Acta Informatica*, 3:59–73, 1973.

[DW86] Elias Dahlhaus and Manfred K. Warmuth. Membership for growing context-sensitive grammars is polynomial. *Journal of Computer and System Sciences*, 33:456–472, 1986.

[GGH69] Seymour Ginsburg, Sheila Greibach, and John Hopcroft. *Studies in Abstract Families of Languages.* Memoir #87, American Mathematical Society, 1969.

[Gla64] A.W. Gladkij. On the complexity of phrase structure grammars. *Algebra i Logika Sem.*, 3:29–44, 1964. (in Russian).

[HIM78] Juris Hartmanis, Neil Immerman, and Stephen R. Mahaney. One-way log-tape reductions. In *19th* Annual Symposium on Foundations of Computer Science (FOCS), pages 65–71, 1978.

[HU79] John E. Hopcroft and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages and Computation.* Addison-Wesley Series in Computer Science. Addison-Wesley, 1979.

[Jan88] Matthias Jantzen. *Confluent string rewriting.* Springer, 1988.

[Lau88] Clemens Lautemann. One pushdown and a small tape. In Klaus W. Wagner, editor, *Dirk Siefkes zum 50. Geburtstag*, pages 42–47. Technische Universität Berlin and Universität Augsburg, 1988.

[Loe70] Jacques Loecks. The parsing of general phrase-structure grammars. *Information and Control*, 16:443–464, 1970.

[LW73] Leonard Y. Liu and Peter Weiner. An infinite hierarchy of intersections of context-free languages. *Mathematical Systems Theory*, 7:185–192, 1973.

[NK87] Paliath Narendran and Kamela Krithivasan. On the membership problem for some grammars. COINS Technical Report CAR-TR-267 & CS-TR-1787 & AFOSR-86-0092, Center for Automation Research, University of Maryland, College Park, MD 20742, March 1987.

[Sud78] Ivan H. Sudborough. On the tape complexity of deterministic context-free languages. *Journal of the Association for Computing Machinery*, 25:405–414, 1978.

[Ven91] H. Venkateswaran. Properties that characterize LOGCFL. *Journal of Computer and System Sciences*, 43:380–404, 1991.