The Variable Membership Problem: Succinctness versus Complexity

Gerhard Buntrock[1] and Krzysztof Loryś[*2]

[1] Institut für Informatik, Universität Würzburg, D-97072 Würzburg, Germany.
[2] Instytut Informatyki, Uniwersytet Wrocławski, 51-151 Wrocław, Poland.

A rule in a grammar is quasi growing if it is growing with respect to given weights of the symbols. Using this definition several types of grammars similar to context sensitive grammars are defined. In this paper we examine the variable membership problem for different types of grammars, namely context-sensitive grammars (CSG), quasi context-sensitive grammars (QCSG), growing context-sensitive grammars (GCSG), quasi growing context-sensitive grammars (QGCSG), and quasi growing grammars (QGG). We show the completeness of these problems in appropriate complexity classes (PSPACE, NEXPTIME). Interestingly the complexity of the variable membership problem differs even when the grammars define the same language class.

1 Introduction

Usually for automata the universal problem is defined as follows: Given an automaton and an input string — does the automaton accept the input?

Sometimes the device solves the universal problem itself. In this case this problem is called complete for the problem class defined by this type of device. Correspondingly in formal language theory the universal problem is called the variable membership problem: Given a grammar and a string — does the grammar produce the string?

Of course the most investigated problem for language classes defined by grammars is the membership problem. The complexity of this problem represents the relationship between language classes defined by grammars and complexity classes.

The complexity of the membership problem yields a lower bound for the variable membership problem. On the other hand for different types of machines or grammars this problem may be of different complexity. If the complexity is of the same complexity as the membership problem, it makes sense to use a universal machine, say an interpreter. If it is higher we should use a compiler since the interpreter would consume higher resources.

The difference in complexity of these problems may be caused by several reasons. The descriptions of the different devices may be of different length. In this case we speak of a difference in the succinctness of the language representation. The number of derivation steps may differ as well.

* This research was partly supported by Humboldt Foundation.

Goldschlager gave an example of two different types of grammars, both characterizing CFL, which are of different variable membership complexity [Gol81]. Precisely, in [JL77] it is proved that the variable membership problem is P-complete. On the other hand for the characterization of CFL by ε-free grammars the variable membership problem is of the same complexity as the membership problem, namely it is in AC^1 [Ruz80].

For regular languages we have different variable membership problems for deterministic, nondeterministic and two-way automata. For example, in the case of deterministic automata this problem is complete for deterministic logarithmic space and in the case of nondeterministic automata it becomes complete for nondeterministic logarithmic space. For related results see [DHL92].

In 1986 Elias Dahlhaus and Manfred Warmuth introduced growing context-sensitive grammars and proved that the membership problem is efficiently solvable (i.e. contained in LOGCFL [DW86]). In [Bun93] it is shown that GCSL is contained in the closure of CFL under one-way logarithmic space bounded reductions. Also important is the automaton characterization of GCSL in [Bun93]. The complexity of the variable membership problem for growing context-sensitive grammars is shown to be NP-complete in [NK87, CH90, BL92].

In [BL92] we introduced the notion of a quasi growing grammar, i.e. a grammar for which weights exist for the terminal and non-terminal symbols such that the rules are growing only with respect to the weights. We proved that this type of grammars characterizes the growing context-sensitive languages. We were convinced that these grammars are somewhat more easy to handle, since it is often more complicate and troublesome to construct a growing context-sensitive grammar. Perhaps one reason for this may be that in growing context-sensitive grammars we cannot bound the length of the rules. For quasi growing grammars we have the Kuroda normal form [BL92]. Here we will prove that the variable membership for these types of grammars varies between NP and NEXPTIME.

In Theorem 9 we show that the variable membership problem for quasi growing context-sensitive grammars is PSPACE-complete under a logarithmic space bounded reduction. The first complete problems for this class were found in automata theory (*Linear Bounded Automaton Acceptance* [Kar72]) and logic (*Quantified Boolean Formulas* and *First Order Theory of Equality*) [SM73]).

In Theorem 10 we strengthen our completeness result and show that this problem is CSL-complete under a reduction with the same space bound and an additional linear bound on its output.

For quasi growing grammars, i.e. grammars that have not to be context-sensitive, we show that the variable membership problem is NEXPTIME-complete (see Theorem 11). Even this type of grammars characterizes the class of growing context-sensitive languages [BL92].

We interpret our results as a way to get information about the succinctness of language representation. The most succinct representation will have certainly the highest complexity.

In Corollary 12 we get that for the class of context-sensitive languages the characterizing quasi context-sensitive grammars have a NEXPTIME-hard variable membership problem. This again gives us a hint that this new type of grammars may be more succinct than the context-sensitive ones.

2 Grammars with weight functions

We assume the reader to be familiar with elementary formal language theory and computational complexity theory, and use the standard notations according to [HU79, BDG88, BDG90].

Sometimes there exists confusion on definitions, e. g. of exponential time-bounded complexity classes. Therefore we recall some important notations. By NEXPTIME we denote the class of languages acceptable in nondeterministic time $\mathrm{O}(2^{n^{\mathrm{O}(1)}})$. A grammar is a quadruple $\langle N, T, S, P\rangle$, where N and T are finite, disjoint alphabets of nonterminal and terminal symbols, respectively; $S \in N$ is a start symbol and P is a finite set of productions of the form $\alpha \to \beta$, with $\alpha, \beta \in (N \cup T)^*$, where α contains at least one nonterminal symbol. The grammars with $|\alpha| \leq |\beta|$ for all productions from P are called *context-sensitive grammars* (CSG). 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 in any rule and*

$$\forall(\alpha \to \beta) \in P : \alpha \neq S \Longrightarrow |\alpha| < |\beta|.$$

We denote the growing grammars by GCSG *and the class of languages generated by such grammars by* GCSL*.*

We distinguish two types of grammars which are growing with respect to a weight function, namely those, which are in addition context-sensitive, and those, which are not. We define a weight function to be a homomorphism from words with concatenation to the positive integers ($\mathbb{N}_+$) with addition:

Definition 2 *We call a function $f : \Sigma^* \to \mathbb{N}_+$ a* weight function *if $\forall w, v \in \Sigma^* : f(w) + f(v) = f(wv)$.*

It has been proven that weight functions for grammars are a good tool to restrict grammars such that they characterize CSL or GCSL. Such grammars are often more easy to construct than context sensitive or growing ones. Here we give the definitions:

Definition 3 *Let $G = \langle N, T, S, P\rangle$ be a grammar and $f : (N \cup T)^* \to \mathbb{N}$ a weight function.*

1. *We call G* quasi growing *if $f(\alpha) < f(\beta)$ for all productions $(\alpha \to \beta) \in P$. We denote the set of quasi growing grammars by* QGG*.*
2. *We call G* quasi growing context-sensitive *if it is quasi growing and a context-sensitive grammar. We denote the set of quasi growing context-sensitive grammars by* QGCSG*.*

3. *We call G* quasi context-sensitive *if $f(\alpha) \leq f(\beta)$ for all productions $(\alpha \to \beta) \in P$. We denote the set of quasi context-sensitive grammars by* QCSG.

There exist normal forms for context-sensitive grammars as well as for the last two types of weighted grammars (Def. $3(2), 3(3)$). A context-sensitive grammar is in *Cremers' normal form* if all productions are of the forms

$$\begin{aligned} AB &\to CD \\ A &\to CD \\ A &\to a \end{aligned}$$

where A, B, C, D are nonterminals and a is a terminal [Cre73]. Obviously such a normal form is not growing. But it is not difficult to show that every QGG or QGCSG can be transformed into an equivalent one in Cremers' normal form. We will need the following proposition:

Proposition 4 *For every context-sensitive (quasi context-sensitive, quasi growing, quasi growing context-sensitive) grammar an equivalent grammar in Cremers' normal form can be constructed in polynomial time [Cre73, BL92].*

We will extend Cremers' normal form in such a way that we allow all rules of the form $(\alpha \to \beta)$ with $|\alpha|, |\beta| \leq 2$. Then we can conclude:

Proposition 5 *For every grammar $G = \langle N, T, S, P\rangle \in$ QGG with weight function f there exists a grammar in Cremers' normal form with weight function f' which fulfills*

$$\forall x \in N \cup T : f'(x) \leq \max\{|\alpha| + |\beta| - 2 \mid (\alpha \to \beta) \in P\} \cdot \max\{f(x) \mid x \in N \cup T\}$$

For a proof see [BL92].

To get an upper bound for the complexity of variable membership problems sometimes we need to estimate the values of the weight function. For this we define minimal weight functions and minimal grammars for minimal weight functions.

Definition 6 *A weight function f is called* minimal *for a quasi growing grammar G if for all weight functions f' for G we have*

$$\sum_{x \in N \cup T} f(x) \leq \sum_{x \in N \cup T} f'(x)$$

We are interested in a minimal weight function for a grammar. At first we will extract the set of rules which are responsible for the highest size of the values of a minimal weight function. This extraction leads to a notion of *minimal grammar* for a minimal weight function.

Definition 7 *Let $G = \langle N, T, S, P\rangle$ be a quasi growing grammar and f a minimal weight function for G. We call a quasi growing grammar $G' = \langle N', T', S, P'\rangle \neq G$* smaller *for (f, G) if the following conditions are fulfilled:*

– $N' \subseteq N, T' \subseteq T$
– *f is a minimal weight function for G'*
– *G' is free of repetitions; i. e. in no rule a symbol occurs on both sides.*
– *A maximal element will remain in G':*

$$\max\{f(x) \mid x \in N \cup T\} \;=\; \max\{f(x) \mid x \in N' \cup T'\}$$

– $\forall (A'_1 A'_2 \cdots A'_{q'_1} \rightarrow B'_1 B'_2 \cdots B'_{q'_2}) \in P'$:
$\exists (A_1 A_2 \cdots A_{q_1} \rightarrow B_1 B_2 \cdots B_{q_2}) \in P$ *and a mapping*
$\sigma_i : \{1, \ldots, q_i\} \rightarrow \{0, 1, \ldots, q_i\}, \sigma_i(j) \in \{0, j\}$*:*

$$A'_1 A'_2 \cdots A'_{q'_1} = A_{\sigma_1(1)} A_{\sigma_1(2)} \cdots A_{\sigma_1(q_1)}$$
$$B'_1 B'_2 \cdots B'_{q'_2} = B_{\sigma_2(1)} B_{\sigma_2(2)} \cdots B_{\sigma_1(q_2)}$$

where A_0 and B_0 denote the empty string ε. In other words σ is deleting some symbols in a rule of the grammar.

If there is no smaller grammar for (f, G'), then G' is called minimal *for (f, G). Note that a construction of a smaller grammar may modify the language.*

A grammar which is minimal for (f, G), where G is in Cremers' normal form, is called in minimal normal form*. Note that such a minimal grammar is in Cremers' normal form.*

Proposition 8 *For every quasi growing grammar $G = \langle N, T, S, P \rangle$ there is a polynomial p and a minimal weight function f such that for all $x \in N \cup T$ the following holds*

$$f(x) \leq 2^{p(|N|+|T|)}.$$

Proof. Let $G' = \langle N', T', S, P' \rangle$ a grammar in minimal normal form for f. Let $N' \cup T' = \{x_1, x_2, \ldots, x_m\}$ such that $f(x_i) \leq f(x_{i+1})$ for $i = 1, \ldots, m-1$. Now we partition the rules P' into the sets $P_0, P_1, \ldots, P_m$ such that $P_0 = \emptyset$ and for $i > 0$:

$$P_i = \{(\alpha \rightarrow \beta) \in P' \;\mid\; \forall j < i : (\alpha \rightarrow \beta) \notin P_j, \;\; \alpha, \beta \in \{x_1, x_2, \ldots, x_i\}^*\}$$

Let j be the smallest index of all nonempty sets P_i. From the minimal condition of G' we conclude that $f(x_j)$ is bounded by $2 \cdot j$. In general we have in the worst case in P_{i+1} a rule of the form $x_i x_i \rightarrow x_{i+1}$. It is easy to see that in this case the weight for x_{i+1} can be bounded by $2f(x_i) + 1$. In this way we can conclude that the maximal weight is bounded by $2^{(|N'|+|T'|)} - 1$. Since the construction of G' is polynomial time-bounded (see Proposition 4) the number of elements in N' is also polynomial bounded. Let p be the polynomial bound of $|N'|$, then we have a bound of $2^{p(|N|+|T|)}$ for the maximal weight for G'. Note that the maximal weight of a minimal weight function for G is bounded by the maximal weight of every minimal weight function for G'. □

3 Completeness results

In this section the main results of this paper will be presented. In Theorem 9 we show that for QGCSG the variable membership problem is PSPACE-complete. For the next result, Theorem 10, we can use the proof of Theorem 9 to show that the same problem is also complete for CSL under a stronger reduction type. Finally, in Theorem 11, we show in a similar manner that the same problem for a more general type of grammars becomes NEXPTIME-complete. These results may somehow surprise. They tell us that for GCSL we have three different types of grammars which characterize the same language class and all these grammar types are of different variable membership complexity (unless NP = PSPACE or PSPACE = NEXPTIME).

Clearly, derivations in GCSG are bounded by the length of the derived words. It is easy to see that derivations in QGCSG and QGG have a length linearly depending on the length of derived words. An appropriate factor for this linear bound can be obtained from the maximum value of the minimal weight function. This fact will be important to get the upper complexity bound of the variable membership problem of QGG.

In the first proof the idea is to construct a grammar for a deterministic linearly space-bounded Turing machine such that the grammar will produce a certain very simple tally string iff the machine will accept its input. It will be shown that this construction can be done within logarithmic space. A similar method will work for the third theorem. But instead of the space-bounded machines we have nondeterministic exponentially time-bounded machines.

Theorem 9 *The variable membership problem for* QGCSG *is* PSPACE-*complete.*

Proof. The upper bound of this problem is the same as for context-sensitive grammars. Here we have Kuroda's classical result: context-sensitive languages are characterized by linear automata [Kur64]. This result is proven by a transformation which can be done in polynomial space. This result carries over to the variable membership problem, i. e. the variable membership problem for all types of context-sensitive grammars belongs to PSPACE.

The hardness will be shown by a generic reduction from the class DCSL. This is sufficient since the closure under any reasonable reduction (e. g. polynomial time-bounded, logarithmically space-bounded) of DCSL yields the class PSPACE. Without loss of generality we can assume that each language L from DCSL will be given by a deterministic single tape Turing machine M such that each word w of L will be accepted in space $|w|$ and time $t := 2^{c|w|}$ for some constant c.

We construct for every pair (M, w) a pair (G, w') such that M accepts w iff we can derive w' from G. Here $M = \langle Q, \Sigma, \Gamma, \delta, q_0, \not{b}, F\rangle$ is the space-bounded machine as mentioned above. We use Q for the states, Σ and Γ are the input and work tape alphabets, respectively, δ is the transition function, q_0 is the initial state, by $\not{b}$ we denote the blank symbol, F contains the accepting states. We will assume that $\Sigma \subseteq \Gamma$. $G = \langle N, T, P, S \rangle$ will be a QGCSG with weight function f. The set given by G will be a tally language over the single letter alphabet $T := \{0\}$.

Our transformation will produce the following fact: Each phrase in a derivation of the grammar that we will present here, will encode a configuration of the computation of the machine M on input $w = w_1 w_2 \cdots w_{|w|}$, whereat every nonterminal describing a part of the configuration (see N_1) encodes one symbol on the tape, and possibly the state, if the considered symbol is scanned by the head of the machine. With the rules we will simulate the computation. To be growing with respect to a weight function in every step some value will occur, which will be moved to the left, where it will be collected in a kind of a counter. This counting we do with the nonterminals from N_2.

We define the set of nonterminals N which are partitioned into $N_0 \cup N_1 \cup N_2$:

$$\begin{aligned}
N_0 &:= \{S, \#\} \\
N_1 &:= (Q \times \Gamma \times \{1,2\}) \ \cup \ (\{\cdot\} \times \Gamma \times \{1, 2, \ldots, |w| - 1\}) \\
N_2 &:= \{0, 1, 2\} \ \times \ \{3^0, 3^1, \ldots, 3^{c|w|}\}
\end{aligned}$$

The last component of a nonterminal in N_1 encodes the weight of this nonterminal. With the symbols from N_2 we encode the position and the digit in a ternary number that will represent a weight collector.

$$\begin{aligned}
f((r, b, j)) &:= j \\
f((k, 3^i)) &:= (k+1) \cdot 3^i \cdot |w|
\end{aligned}$$

for $j \in \{1, \ldots, |w| - 1\}$, $i \in \{0, \ldots, |w|\}$, $k \in \{0, 1, 2\}$, $r \in \{\cdot\} \cup Q$, $b \in \Gamma$. For the nonterminals S and $\#$ we define $f(S) := f(\#) := 1$. And for the only terminal symbol 0 we define $f(0) \ := \ |w| \cdot 3^{c|w|+1} + 1$, such that it is the heaviest symbol in our grammar.

The set of productions P is partitioned into the set consisting of the starting rule (P_1), the simulation rules (P_2), the weight moving rules (P_3), the weight collecting rules (P_4), and the terminal rules (P_5):

$$P \ := \ P_1 \cup P_2 \cup P_3 \cup P_4 \cup P_5$$

With the starting rule we produce the initial configuration and the symbol $\#$ initializes the counter:

$$P_1 \ := \ \{S \longrightarrow \#(q_0, w_1, 1)(\cdot, w_2, 1) \ldots (\cdot, w_{|w|}, 1)\}$$

Without loss of generality we assume that in the calculation of M only right and left moves occur, i.e.

$$\delta : Q \times \Gamma \longrightarrow Q \times \Gamma \times \{L, R\}$$

For every right move $\delta(q, a) = (q', a', R)$ and all $b \in \Gamma$ we put the rule

$$(q, a, 1)(\cdot, b, 1) \rightarrow (\cdot, a', 2)(q', b, 1)$$

into P_2. For every left move $\delta(q,a)=(q',a',L)$ and all $b\in\Gamma$ we put the rule

$$(\cdot,b,1)(q,a,1)\to(q',b,2)(\cdot,a',1)$$

into P_2.

The set P_3 consists of the rules which transport the weights to the left. For all $q\in Q$, $a,b\in\Gamma$, and $i=2,3,\ldots,|w|-2$, we put the rules

$$\begin{aligned}(\cdot,a,1)(q,b,2)&\to(\cdot,a,3)(q,b,1)\\(\cdot,a,1)(\cdot,b,i)&\to(\cdot,a,i+1)(\cdot,b,1)\end{aligned}$$

into P_3. Note that since the length of all configurations is $|w|$, the weights encoded in these nonterminals cannot exceed $|w|-1$.

The next part consists of the rules which will in some sense collect the weights produced by P_2 and transported by P_3. For all $(r,j)\in(Q\times\{2\})\cup(\{\cdot\}\times\{2,3,\ldots,|w|-1\})$, $a\in\Gamma$, $3^i\in\{3^0,3^1,\ldots,3^{c|w|-1}\}$ the following rules belong to P_4:

$$\begin{aligned}\#(r,a,j)&\to\#(1,3^0)(r,a,1)\\(0,3^0)(r,a,j)&\to(1,3^0)(r,a,1)\\(1,3^0)(r,a,j)&\to(2,3^0)(r,a,1)\\\#(2,3^i)&\to\#(1,3^{i+1})(0,3^i)\\(0,3^{i+1})(2,3^i)&\to(1,3^{i+1})(0,3^i)\\(1,3^{i+1})(2,3^i)&\to(2,3^{i+1})(0,3^i)\end{aligned}$$

A computation of M will halt and accept if a configuration will contain a final state. In the last part P_5 of the rules the nonterminal that encodes a final state will give the possibility to apply the terminal rules: For all $q\in F$, $a\in\Gamma$ the rules

$$(q,a,1)\to 0$$

belong to P_5. Moreover, for all nonterminals $X\in N$ we have in P_5 :

$$\begin{aligned}X0&\to 00\\0X&\to 00\end{aligned}$$

To check whether the grammar is growing with respect to the weight function we check only the rules in P_4, since in all other cases it is obvious:

$$\begin{gathered}
\#(r,a,j) \rightarrow \#(1,3^0)(r,a,1) \\
1+j < 1+|w| < 2|w|+2 = 2\cdot 3^0 \cdot |w| + 2
\end{gathered}$$

$$\begin{gathered}
(0,3^0)(r,a,j) \rightarrow (1,3^0)(r,a,1) \\
3^0 \cdot |w| + j < 2\cdot 3^0 \cdot |w| + 1
\end{gathered}$$

$$\begin{gathered}
(1,3^0)(r,a,j) \rightarrow (2,3^0)(r,a,1) \\
2\cdot 3^0 \cdot |w| + j < 3\cdot 3^0 \cdot |w| + 1
\end{gathered}$$

$$\begin{gathered}
\#(2,3^i) \rightarrow \#(1,3^{i+1})(0,3^i) \\
1 + 3\cdot 3^i \cdot |w| < 1 + 2\cdot 3^{i+1} \cdot |w| + 3^i \cdot |w|
\end{gathered}$$

$$\begin{gathered}
(0,3^{i+1})(2,3^i) \rightarrow (1,3^{i+1})(0,3^i) \\
3^{i+1}\cdot |w| + 3\cdot 3^i \cdot |w| < 2\cdot 3^{i+1} \cdot |w| + 3^i \cdot |w|
\end{gathered}$$

$$\begin{gathered}
(1,3^{i+1})(2,3^i) \rightarrow (2,3^{i+1})(0,3^i) \\
2\cdot 3^{i+1}\cdot |w| + 3\cdot 3^i \cdot |w| < 3\cdot 3^{i+1} \cdot |w| + 3^i \cdot |w|
\end{gathered}$$

To get the polynomial time bound, we estimate the length of the grammar G. At first we count the number of nonterminals:

$$\begin{aligned}
|N| &= |\Gamma \times (\{\cdot\} \times \{1,2,\ldots,|w|-2\} \cup Q \times \{1,2\})| \\
&\quad +3\cdot |\{3^0,3^1,\ldots,3^{c|w|}\}| + |\{S,\#\}| \\
&= |\Gamma|\cdot(|w|-2+2\cdot|Q|) \;+\; 3\cdot(c|w|+1) \;+\; 2 \\
&< d\cdot|\Gamma|\cdot(|w|+|Q|) \quad \text{for some constant } d.
\end{aligned}$$

We count the number of rules:

$$\begin{aligned}
|P| &= |P_1| + |P_2| + |P_3| && +|P_4| && +|P_5| \\
&\le \;\; 1 \;+\; |\delta| \;+(|Q|+|w|)\cdot|\Gamma|^2 && && \\
& && +3(|\Gamma|\cdot(|w|-1)+c|w|) && \\
& && && +|F|\cdot|\Gamma|+2|N| \\
&< |\delta| + d'|\Gamma|^2\cdot(|w|+|Q|) \quad \text{for some constant } d'.
\end{aligned}$$

This number is clearly polynomially bounded in the length of (M, w).

It is easy to see that G will produce a tally string of polynomial length iff the machine M will accept the input w. From this and the easyness of the construction we can conclude that in the construction of the grammar and the tally string, only logarithmic space is needed. Moreover, to simplify the computation of the length of the tally string we add the rule

$$0 \rightarrow 00$$

In this way it is sufficient to compute an upper bound of this number. □

Meyer and Stockmeyer introduced the stronger reduction type of logarithmically space-bounded reductions: reductions with the same space bound but additionally with linearly length-bounded output [MS72]. Note that the class CSL is closed under this type of reductions.

Theorem 10 *The variable membership problem for* QGCSG *is* CSL-*complete under logarithmically space-bounded and linearly length-bounded reductions.*

Proof. We can use the proof of Theorem 9. First we note that the changes needed for nondeterministic machines can be easily done. Now, without loss of generality we can assume that all Turing machines work on the same alphabet. For this case we have to enlarge the space bound such that M can use $c' \cdot |w|$ tape cells for some constant c'. One can easily verify that the length of the output of the reduction in this way is linearly bounded. □

Theorem 11 *The variable membership problem for* QGG *is* NEXPTIME-*complete.*

Proof. An upper bound of the complexity can be obtained in the following way. For a given quasi growing grammar with the minimal weight function f and a string w we have to decide if w is derivable by the grammar. Let $\&_x$ for every symbol x in the grammar be a new symbol. We substitute all symbols x in the grammar by the string $x\&_x^{f(x)}$. In this way the grammar becomes growing and membership can nondeterministically be checked in polynomial time. Since the blow-up by this substitution is bounded exponentially in the length of the grammar (see Proposition 8), it is bounded exponentially also in the length of the input of the variable membership problem for QGG. Therefore we can conclude that the latter problem can be solved nondeterministically in exponential time.

The hardness can be shown by a similar construction as in Theorem 9. The main changes consist in how the counter is organized. □

From this result we get the corresponding theorem for context-sensitive languages.

Corollary 12 *The variable membership problem for* QCSG *is* NEXPTIME-*hard.*

Proof. Obviously the reduction from Theorem 11 works for this type of grammars too. □

If we try to find an upper bound for QCSG with the method from Theorem 11 we will find exponential space. It may be the case that this problem is exponential-space complete.

Wagner investigated the complexity of combinatorial problems under different input languages. He shows that in several cases more succinctness of the input languages causes a blow-up of the complexities [Wag86]. Our results give us a hint that the grammars from QGG, QGCSG, and QCSG are somewhat more succinct than the corresponding grammars from GCSG and from CSG, respectively.

Acknowledgement

Thanks go to Ronald Book, Diana Rooß, Ulrich Hertrampf, Markus Holzer, Clemens Lautemann, Gundula Niemann, Carmen Schwenkel, Heribert Vollmer, and Klaus Wagner for several discussions and hints.

References

[BDG88] José L. Balcázar, Josep Díaz, and Joaquim Gabarró. *Structural Complexity Theory I.* Springer, 1988.

[BDG90] José L. Balcázar, Josep Díaz, and Joaquim Gabarró. *Structural Complexity Theory II.* Springer, 1990.

[BL92] Gerhard Buntrock and Krzysztof Loryś. On growing context-sensitive languages. In *Proc. of 19th ICALP*, volume 623 of *LNCS*, pages 77–88. Springer, 1992.

[Bun93] Gerhard Buntrock. Growing context-sensitive languages and automata. Technical Report 69, Universität Würzburg, Institut für Informatik, Am Exerzierplatz 3, D-97072 Würzburg, November 1993.

[CH90] Sang Cho and Dung T. Huynh. The complexity of membership for deterministic growing context-sensitive grammars. *International Journal of Computer Mathematics*, 37:185–188, 1990.

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

[DHL92] Carsten Damm, Markus Holzer, and Klaus-Jörn Lange. The parallel complexity of iterated morphisms and the arithmetic of small numbers. In *Proc. of 17th MFCS*, volume 629 of *LNCS*, pages 227–235. Springer, 1992.

[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.

[Gol81] Leslie M. Goldschlager. ε-productions in context-free grammars. *Acta Informatica*, 16(3):303–308, 1981.

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

[JL77] Neil D. Jones and William T. Laaser. Complete problems for deterministic polynomial time. *Theoretical Computer Science*, 3:105–117, 1977.

[Kar72] Richard M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations.* Plenum Press, New York, 1972.

[Kur64] S.-Y. Kuroda. Classes of languages and linear-bounded automata. *Information and Control*, 7:207–223, 1964.

[MS72] Albert R. Meyer and Larry J. Stockmeyer. The equivalence problem for regular expressions with squaring requires exponential space. In *Proc. of the 13th Annual IEEE Symposium on Switching and Automata Theory*, pages 125–129, 1972.

[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 Automaton Research, University of Maryland, College Park, MD 20742, March 1987.

[Ruz80] Walter L. Ruzzo. Tree-size bounded alternation. *Journal of Computer and System Sciences*, 21:218–235, 1980.

[SM73] Larry J. Stockmeyer and Albert R. Meyer. Word problems requiring exponential time: preliminary report. In *Proc. of 5th STOC*, pages 1–9, 1973.

[Wag86] Klaus W. Wagner. The complexity of combinatorial problems with succinct input representation. *Acta Informatica*, 23:325–356, 1986.