Weak Growing Context-Sensitive Grammars

Gerhard Buntrock, Gundula Niemann

Report No. 112 Mai 1995

Universität Würzburg, Technical Report

Weak Growing Context-Sensitive Grammars *

Gerhard Buntrock and Gundula Niemann

Institut für Informatik
Universität Würzburg
D-97072 Würzburg, Germany;
E-mail: `[bunt, niemann]@informatik.uni-wuerzburg.de`

Mai 1995

Abstract. Well investigated are growing context-sensitive languages (GCSL), i. e. languages defined by grammars that have only growing rules, where the left side of a growing rule is shorter than its right side. *GCSL* is characterized by quasi growing context-sensitive grammars (QGCSG), i. e. the rules therein are growing only with respect to a valuation of all the strings by a homomorphism to natural numbers with addition.

In this paper we introduce *weak growing context-sensitive grammars* (WGCSG), i. e. grammars that are growing with a weighted valuation of all strings, where the weights are the values of a position valuation. Here we evaluate a string by summing up every product of a symbol value with its position value.

For every position valuation which does not agree with the beginning of any exponential function, the corresponding class of WGCSGs characterizes *CSL*, shown by a somehow tricky proof. Such a valuation function is called *unsteady*. On the other hand all WGCSG related to steady position valuations are linearly inclusion-ordered and characterize the hierarchy of exponential time bounded languages in *CSL*. This hierarchy collapses iff

- any class defined by an unsteady position valuation has a normal form of order 2 (e. g. Cremers' or Kuroda's normal form)
- any class defined by a steady position valuation is closed under inverse homomorphisms.

Introduction

A beautiful theory has always simple concepts and remarkable results. Here we investigate a widely unknown but nice part of the class of context-sensitive languages (*CSL*).

Our simple concept consists in a valuation of strings and concentration on context-sensitive grammars that are growing under such valuation, and the remarkable results are in form

*Some of these results were presented at LATIN'95 at Valparaíso, Chile of characterizations of known double bounded complexity classes by this quite different concept. Paul asks in general which kind of speed up theorem does exist for space bounded computations ([Pau78]). Only little is known about such dual bounded classes, see e. g. [Pip79]. Here we characterize the open problem, whether all linear space-bounded automata can be simulated by linear space-bounded automata with an additional universal exponential time bound in the concepts of formal language theory. There it is questioned, whether a class of languages has a normal form of bounded order or whether another class is closed under inverse homomorphisms.

Noam Chomsky already has observed in his famous work that *CSL* is characterized by monotone grammars, that are grammars in which in every rule a string is replaced by a string that is at least as long as the first one ([Cho59]. We will use this characterization for definition.

Dahlhaus and Warmuth investigated the complexity of growing context-sensitive languages (*GCSL*) and found out that *GCSL* is contained in *LOGCFL* ([DW86]), the class of such languages that can be transformed into a context-free language (*CFL*) using logarithmic space ([Sud78], [Ruz80]). Note, that *LOGCFL* is a subclass of *P*. *GCSL* is defined with growing context-sensitive grammars (*GCSG*), that is done by replacing "as least as long as" in the definition of *CSG* by "longer than" [1]. If we valuate the symbols of a CSG with a homomorphic mapping into the natural numbers with addition and demand that in every rule the sum of the values of the symbols increases, this type of grammars (quasi GCSG, QGCSG for short) characterizes *GCSL* ([BL92], [BL94]).

Unfortunately, the class *GCSL* is very weak. Not even the language $COPY := \{ww \mid w \in \{a,b\}^*\}$ is contained in *GCSL* ([Bun93], see also [BO95]).

Weak growing context-sensitive grammars (WGCSG) are a generalization of QGCSG. In a WGCSG we valuate the symbols as well as the positions inside a rule: Every position inside a rule is related to a certain value. The valuation of a side of a rule is obtained by the sum of the products of the symbol valuation and the position valuation for every symbol. A context-sensitive grammar is called weak growing context-sensitive related to a certain position valuation ($WGCSG_s$, where s stands for the position valuation), if there exists a symbol valuation such that for every rule the valuation of the left side is lower than the one of the right side. (This is defined precisely in Def. 1.2.) The weak growing context-sensitive languages related to constant position valuations characterize *GCSL*. We will show that this definition is robust under some natural changes.

The valuation of positions gives the possibility to interchange two symbols by a rule. This can be used to prove $COPY \in WGCSL_s$ for any non-constant position valuation s. On the other hand it can easily be seen that it is not possible to interchange two symbols back and forth in the same grammar. Thus we know that $WGCSG_s$ is strictly contained in *CSG* for any position valuation s.

For important language classes we expect some essential closure properties; clearly, the closure under non-ε-free homomorphisms of $WGCSL_s$ yields the class of all recursively enumerable sets, i.e. $WGCSL_s$ is a basis for recursively enumerable sets. With standard arguments the following can be proved: for any position valuation s, the class $WGCSL_s$ is closed under ε-free homomorphism, union, ε-free regular substitution, concatenation and

[1] Note we differ between the name of the class *CSG* and a member CSG

intersection with regular sets. Additionally it can be shown that $WGCSL_s$ is closed under transposition for any position valuation s.

What is missing (e. g. to be an abstract family of languages) is the closure under inverse homomorphism and the one under k-bounded homomorphism, where from [GGH69] it is known that, if the one holds, then also the other and vice versa because of the closure properties we already know. The closure under inverse homomorphism here can neither be shown nor negated; a main result in this context is the equivalence of this problem to the problem whether all linear space-bounded automata can be simulated by linear space-bounded automata with an additional universal exponential time bound. The reason for this equivalence lies in the fact that for any non-constant position valuation the closure under inverse homomorphism of $WGCSL_s$ equals CSL.

To prove these results we distinguish steady and unsteady position valuations: A position valuation is steady, if it agrees with the beginning of an exponential function. We refer to the base of that corresponding function as the growth factor of the position valuation. In the other case, the position valuation is unsteady.

It turns out that steady position valuations are uniquely extendible to infinite position valuations such that all strings can be valuated. Moreover, the number of derivation steps can be bounded by the valuation of the derived string. This gives the result that for a steady position valuation every weak growing context-sensitive grammar related to it can produce only sets acceptable by linear space-bounded automata in a certain exponential time, where the base of the bounding exponential function is the growth factor of the position valuation (if it is monotone increasing). To get this result so tight, we transfer Gladkii's Connectivity Theorem to WGCSLs. On the other hand for every language that can be recognized by a linear space-bounded automaton exists a steady position valuation s such that this language belongs to $WGCSL_s$. To get here a very tight relation, we introduce a new type of counters: grammars that count.

Furthermore, the weak growing context-sensitive languages corresponding to different growth factors build up a linearly inclusion-ordered hierarchy that characterizes the exponential time hierarchy for CSL.

For unsteady position valuations an infinite extension in a way that the value of a sentential form increases if and only if a weak growing context-sensitive rule is applied does not exist. Here we will show that arbitrary long derivations are possible. We interpret this to be the reason for the outcome that for every unsteady position valuation the corresponding class of weak growing context-sensitive grammars characterizes CSL.

Another important difference for the steady and unsteady cases consists in the existence of normal forms of order 2. Note, every s-weak growing context-sensitive grammar in a normal form of order 2 is a weak growing context-sensitive grammar related to a steady position valuation. Here we know that for every steady position valuation and every weak growing context-sensitive grammar related to it there exists a normal form. For this we will use Cremers' normal form, which has the advantage that it has a minimal number of different types of rules. Additionally it turns out that the growth factor of a steady position valuation characterizes a weak growing context-sensitive language class and that the language classes corresponding to a growth factor and its reciprocal value coincide. For unsteady position valuations the natural algorithm to reduce the order of a rule and similar ones do not work.

Thus we obtain the equivalence of the following problems

- Does the exponential time hierarchie for *CSL* collapse? That is: Can all linear space bounded automata be simulated by linear space bounded automata with an additional universal exponential time bound?
- Does there exist a steady position valuation such that the corresponding class of weak growing context-sensitive languages coincides with *CSL*?
- Does there exist a steady position valuation such that the corresponding class of weak growing context-sensitive languages is closed under inverse homomorphism?
- Does there exist an unsteady position valuation such that the following holds: Any weak growing context-sensitive grammar related to this position valuation can be transformed into an equivalent one in an order-2-normal form that is weak growing context-sensitive related to any position valuation?

1 Preliminaries

A *context-sensitive grammar* (CSG) 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 rules (productions) of the form $\alpha \to \beta$ with $\alpha, \beta \in (N \cup T)^*$, where α contains at least one nonterminal symbol, and $|\alpha| \le |\beta|$ or $(\alpha \to \beta) = (S \to \varepsilon)$, and if $(S \to \varepsilon) \in P$, S does not appear on the right side of any rule. The corresponding language class is denoted by *CSL*.

If $u, v \in (N \cup T)^*$ are sentential forms, we say *u derives v in G*, denoted by $u \Longrightarrow v$, if there exist $x, y, \alpha, \beta \in (N \cup T)^*$ such that $u = x\alpha y, v = x\beta y$ and $(\alpha \to \beta) \in P$. Let $\stackrel{*}{\Longrightarrow}$ denote the reflexive and transitive closure of $\Longrightarrow$. Thus the *language* $L(G)$ generated by a grammar $G = \langle N, T, S, P\rangle$ is described by: $L(G) = \{w \in T^* \mid S \stackrel{*}{\underset{G}{\Longrightarrow}} w\}$.

Grammars with $\alpha \in N$ for each rule $(\alpha \to \beta) \in P$ are called context-free. The corresponding language class is denoted by *CFL*.

By replacing the inequality in the definition of context-sensitive grammars by a sharp one, we obtain growing context-sensitive grammars (*GCSG*), that generate the growing context-sensitive languages (*GCSL*) (see [DW86], also [BL92], [Nie92]). The class *GCSL* is placed properly between *CFL* and *CSL* ([DW86], also see [Nie92]).

Instead of just counting the symbols appearing in a rule like we did above we can associate them with a certain value (or weight).

A grammar $G = \langle N, T, S, P\rangle$ is called *quasi growing context-sensitive*, if it is context-sensitive and there exists a function $f\colon N \cup T \to \mathbb{N}$ such that for each rule $(\alpha_1 \dots \alpha_n \to \beta_1 \dots \beta_m) \in P$ holds:

$$\sum_{i=1}^{n} f(\alpha_i) < \sum_{i=1}^{m} f(\beta_i) \quad \text{or} \quad \alpha = S,\, \beta = \varepsilon$$

The corresponding set of grammars is denoted by *QGCSG*. The function f is called *symbol valuation for G*.

If in the definition above we do without the context-sensitivity, but continue demanding that if $(S \to \varepsilon) \in P$, S does not appear on the right side of any rule, we obtain the *quasi growing grammars*, and the set of all such grammars is denoted by QGG.

In [BL92] and [BL94] it was shown that both quasi growing grammars and quasi growing context-sensitive grammars characterize the class of growing context-sensitive languages (also see [Nie92]).

Instead of valuating only the symbols we can also valuate their position inside a rule. A position valuation must valuate at least one position and it must be unbroken, because otherwise the valuation of a rule could not be defined naturally. In the following we will refer to $\max\{\max\{|\alpha|, |\beta|\} \mid (\alpha \to \beta) \in P\}$ as *the order of* G.

Definition 1.1 *A* position valuation *is a function* $s\colon \mathbb{N}^+ \to \mathbb{N}^+$ *with:*

> *If s is defined for a $j \in \mathbb{N}^+$, s is also defined for every $i \in \mathbb{N}^+$ with $i < j$.*

If s is defined for all $i \in \mathbb{N}^+$, we say s is an infinite *position valuation — in the other case, s is a* finite *position valuation. We call each $i \in \mathbb{N}^+$ where $s(i)$ is defined a* valuated position of s.

It is also possible to allow zero points for position valuations, but with the techniques used to show our main results we get for any position valuation with zero points a characterization of *CSL*.

We restrict ourselves to the most interesting position valuations which have at least two valuated positions.

Definition 1.2 *Let s be a position valuation.*
A grammar $G = \langle N, T, S, P\rangle$ is called weak growing context-sensitive related to the position valuation s *or also* s-weak growing context-sensitive, *if the following holds:*

(i) *G is context-sensitive*

(ii) *Let ℓ be the order of G. Then s has at least ℓ valuated positions.*

(iii) *There exists a function $f\colon N \cup T \to \mathbb{N}$ such that for every rule $(\alpha_1 \dots \alpha_n \to \beta_1 \dots \beta_m) \in P$ holds:*

$$(*) \qquad \sum_{i=1}^{n} s(i) \cdot f(\alpha_i) < \sum_{i=1}^{m} s(i) \cdot f(\beta_i)$$

Such a function f is called symbol valuation for G. *If the inequality applies for a rule $(\alpha \to \beta) \in P$, we say it is* s-weak growing with f.

The sets of corresponding grammars and languages are denoted by $WGCSG_s$ and $WGCSL_s$, respectively.

As we now valuate not only the symbols but also their position inside a rule, it is possible to interchange two symbols by a rule. This can be used to show that the language $COPY := \{ww \mid w \in \{a,b\}^*\}$ is weak growing context-sensitive related to any non-constant position valuation. To give an intuition of weak growing context-sensitive grammars, we look at this example closely:

Example 1.3 *We look at the language $COPY := \{ww \mid w \in \{a,b\}^*\}$.*
Let s be any position valuation with at least two valuated positions and $s(1) < s(2)$.
Define a grammar $G = \langle N, \{a,b\}, S, P\rangle$ as follows:

$$N = \{S, A, A', B, B', (X,a), (X,b)\}$$

and P consisting of the following rules:

$$\begin{array}{llll} S \to AA'S & A'A \to AA' & A'(X,a) \to (X,a)a & (X,a) \to a \\ S \to BB'S & A'B \to BA' & A'(X,b) \to (X,a)b & (X,b) \to b \\ S \to A(X,a) & B'A \to AB' & B'(X,a) \to (X,b)a & A \to a \\ S \to B(X,b) & B'B \to BB' & B'(X,b) \to (X,b)b & B \to b \end{array}$$

This grammar is context-sensitive, of the order 2, and it generates the language COPY. Now we define $f\colon N \cup T \to \mathbb{N}$ by:

$$\begin{array}{l} f(S) = f(A) = f(B) = f((X,a)) = f((X,b)) = 1 \\ f(A') = f(B') = 2, \qquad f(a) = f(b) = 2 \end{array}$$

With a simple calculation it can be seen that every rule of P is s-weak growing with respect to this symbol valuation. Thus the grammar G is s-weak growing context-sensitive.

As $COPY \notin GCSL$ is true ([Bun93]), we have $WGCSL_s \nsubseteq GCSL$ for any non-constant position valuation s. (The example above can be adapted easily for position valuations s with $s(1) > s(2)$ and for those with a constant beginning part.)

Note, if we use classical context-sensitive grammars in the definition of WGCSGs, we get a subclass of GCSG ([Nie94a]). Here the interesting question arises whether this type of grammars already characterizes the class *GCSL*.

On the other hand it can easily be seen that it is not possible to interchange two symbols back and forth in the same grammar, thus we know $WGCSG_s \subset CSG$ for any position valuation s.

We now look briefly at some robustness properties of the definition of weak growing context-sensitive languages:

As can easily be seen the standard algorithm to delete chain rules (see e.g. [Har78]) can easily be adapted for weak growing context-sensitive grammars. The same applies for the standard algorithm to delete terminal symbols from the left side of a rule. Thus we obtain:

Proposition 1.4 *In a WGCSG related to a position valuation s we can delete chain rules, even if there is only a symbol valuation such that some chain rules are not strictly growing, i.e. cycling with chain rules can be deleted in such cases.*

Furthermore terminals can be forbidden for the left side of rules.

As in an inequality the multiplication of both sides with the same positive value does not effect its validity, the following holds:

Proposition 1.5 *For all $k \in \mathbb{N}^+$ the classes $WGCSL_{k \cdot s}$ are the same. The same holds for $k \in \mathbb{Q}^+$.*

Thus for any constant position valuation C holds $WGCSG_C = QGCSG$ and consequently $WGCSL_C = GCSL$.

The values can not only be stretched but also moved, which leads to the following result:

Proposition 1.6 *In Definition 1.2(iii) it is sufficient to demand that there exists a function with values in $\mathbb{Q}$ that fulfills the property $(*)$.*

PROOF: Let s be a position valuation and let $G = \langle N, T, S, P\rangle$ be a context-sensitive grammar where s has at least $\mathrm{ord}(G)$ valuated positions. Let $f\colon N \cup T \to \mathbb{Q}$ be a symbol valuation such that for every rule $(\alpha_1 \dots \alpha_n \to \beta_1 \dots \beta_m) \in P$ holds

$$\sum_{i=1}^{n} s(i) \cdot f(\alpha_i) < \sum_{i=1}^{m} s(i) \cdot f(\beta_i)$$

Let $N \cup T =: \{a_1, \dots, a_k\}$ and $f(a_i) = \frac{p_i}{q_i}$ with $p_i \in \mathbb{Z}$ and $q_i \in \mathbb{N}$ for $i = 1, \dots, k$. With an easy transformation we obtain a symbol valuation with symbols in $\mathbb{N}$: Let $q := \mathrm{lcm}(q_1, \dots, q_k)$ the common denominator of all values and $z := q \cdot |\min\{0, q_1, \dots, q_k\}|$ the product of this common denominator with the absolute value of the smallest negative enumerator. Then by a linear transformation we obtain natural numbers: $f'(a_i) := q \cdot f(a_i) + z$. The inequalities still apply, as the grammar has no length-decreasing rules. ♣

In a similar way it can be seen that allowing values out of $\mathbb{Q}^+$ for position valuations does not lead to any new language classes.

As we mentioned above, in the case of quasi growing context-sensitive grammars if we do without context-sensitivity we obtain grammars that characterize the same language class ([BL92]). In the case of weak growing context-sensitive grammars the effect is different: If in Definition 1.2 we leave out the demanding of context-sensitivity, we obtain the quasi weak growing grammars, which already characterize the languages generated by general grammars ([Nie94b]), i. e. the class of recursively enumerable sets.

2 Closure and Normal Form

2.1 Closure Properties

To come to know a new language class more closely, it is useful to learn something about its closure properties. Clearly, the closure under non-ε-free homomorphism of $WGCSL_s$ yields the class of all recursively enumerable (r. e.) sets for any position valuation s, i. e. $WGCSL_s$ is a basis for r. e. sets.

With standard arguments it can be shown:

Theorem 2.1 *Let s be a position valuation with at least two valuated positions. $WGCSL_s$ is closed under ε-free homomorphism, union, ε-free regular substitution, concatenation and intersection with regular sets.*

Additionally it can be shown:

Theorem 2.2 *Let s be a non-constant position valuation. Then $WGCSL_s$ is closed under transposition.*

PROOF: We sketch the idea for the case $s(1) < s(2)$.

For a grammar $G = \langle N, T, S, P\rangle \in WGCSG_s$ a grammar $\tilde{G} = \langle \tilde{N}, T, \tilde{S}, \tilde{P}\rangle$ is constructed as follows:

$$\tilde{N} = N \cup \{\tilde{S}\} \cup \{a' \mid a \in T\} \cup \{(M, X) \mid X \in N\} \cup \{(M, a') \mid a \in T\}$$

and the rules in $\tilde{P}$ are to realize the following:

> First for a word $w = w_1 \dots w_n \in L$ the sentential form $(M, w_1')w_2' \dots w_n'$ is generated (for this appropriate copies of the rules in P are used). Then this sentential form is transformed with rules of the form
>
> $$\begin{array}{l} (M, a')b' \to (M, b')a \\ ab' \to b'a \\ (M, a') \to a \end{array}$$
>
> into the word $w_n \dots w_1$.

Clearly $L(\tilde{G}) = L(G)^T$. Out of a symbol valuation for G an appropriate one for $\tilde{G}$ can be constructed.

For a symbol valuation s with $s(1) > s(2)$ the proof works analogously.

For the case of a non-constant position valuation with $s(1) = s(2)$ this idea can be adapted by shifting the “mirror point” (M, X) and treating the borders appropriately. ♣

In order to show that $WGCSL_s$ is an AFL for a certain position valuation s we would have to show that $WGCSL_s$ is closed under inverse homomorphism (or under k-bounded homomorphism because of Theorem 2.1 and [GGH69]).

It turns out that for steady position valuations this problem is equivalent to the problem whether all linear space-bounded automata can be simulated by linear space-bounded automata with an additional universal exponential time bound (see Theorem 5.5). The reason for this equivalence lies in the fact that for any position valuation the closure under inverse homomorphisms of $WGCSL_s$ equals CSL (see Theorems 3.4 and 5.4).

2.2 Normal Form

To get our main results we now introduce the two cases of steady and unsteady position valuations, and first we work out what we can say about the transformability of corresponding weak growing context-sensitive grammars into an order-2-normal form.

Definition 2.3 *A position valuation $s: \mathbb{N}^+ \to \mathbb{N}^+$ is called* steady*, if it agrees with the beginning part of an exponential function, that is, if it can be written as*

$$s(i) = s(1) \cdot \mathrm{w}(s)^{i-1} \quad \textit{where } \mathrm{w}(s) \in \mathbb{Q}^+, \quad \textit{and } \mathrm{w}(s) = \frac{s(2)}{s(1)} \ ;$$

$\mathrm{w}(s)$ *is called the* growth factor *of s.*

In the other case, s is unsteady*. Note, that this is the case if and only if s has at least 3 valuated positions and if there exists a valuated position $i \in \mathbb{N}^+$ of s, for which $i-1$ and $i+1$ are valuated, too, with:*

$$s(i)^2 \neq s(i-1) \cdot s(i+1)$$

In this case we call such an i a blip *of s, and the smallest such i we call the* first blip *of s.*

For every steady position valuation s any grammar in $WGCSG_s$ can be transformed into a normal form of order 2. We will use the Cremers' normal form ([Cre73]). Compared with all other normal forms of order 2, the advantage of Cremers' normal form consists in the fact that it has a minimal number of different types of rules. Moreover, with the probably more popular normal form proposed by Kuroda we can only describe classical context-sensitive grammars, which leads here to a language class contained in $GCSL$, as we mentioned in Section 1.

Definition 2.4 *[Cre73] A context-sensitive grammar $G = \langle N, T, S, P\rangle$ is in* Cremers normal form*, if every rule in P is of one of the forms $AB \to CD$, $A \to CD$, $A \to a$, where $A, B, C, D \in N$, $a \in T$.*

At first we adapt the natural algorithm to reduce the order of a grammar.

Theorem 2.5 *Let s be a steady position valuation. Then for every grammar $G \in WGCSG_s$ of the order $\ell \geq 3$ exists an equivalent grammar $G' \in WGCSG_s$ of the order $\ell - 1$.*

PROOF: Let $s: \mathbb{N}^+ \to \mathbb{N}^+$ be a position valuation with at least three valuated positions (in the other case there is nothing to show). Define $s': \mathbb{N}^+ \to \mathbb{N}^+$ by $s'(i) := s(i+1)$ for all $i \in \mathbb{N}^+$, that is, if $s(i+1)$ is undefined, so is $s'(i)$. Let $G = \langle N, T, S, P\rangle \in WGCSL_s$ of the order $\ell \geq 3$, let $f: N \cup T \to \mathbb{N}$ be a symbol valuation for G.

We now construct a grammar $G' = \langle N', T, S', P'\rangle$ and a symbol valuation $f': N' \cup T \to \mathbb{Q}$ by the following algorithm:

1. $N' := N$, $P' := \emptyset$
2. $f'(A) = f(A)$ for every $A \in N$, $\quad f'(a) = f(a)$ for every $a \in T$
3. For every rule $(\alpha_1 \ldots \alpha_n \to \beta_1 \ldots \beta_m) \in P$ repeat the following:
    4. If $m = 2$, let $P' := P' \cup \{(\alpha_1 \ldots \alpha_n \to \beta_1 \ldots \beta_m)\}$
    5. If $3 \leq m \leq \ell$, let X be a new nonterminal symbol, $X \notin N'$.
        6. Define the rules
        $$\begin{array}{ll} \text{(R1)} & \alpha_1\alpha_2 \to \beta_1 X \\ \text{(R2)} & X\alpha_3 \ldots \alpha_n \to \beta_2 \ldots \beta_m \end{array}$$
        Here $n = 2$, that means $\alpha_3 \ldots \alpha_n = \varepsilon$, or $n = 1$, that means additionally $\alpha_2 = \varepsilon$, is possible.
        7. Let $P' := P' \cup \{(R1), (R2)\}$
        8. Let $f'(X) := \frac{s(1)}{s(2)} \cdot f(\alpha_1) + f(\alpha_2) - \frac{s(1)}{s(2)} \cdot f(\beta_1) + \frac{1}{2 \cdot s(2)}$

It holds $L(G') = L(G)$. By calculation and Proposition 1.6 can be shown $G' \in \mathit{WGCSG}_{s'}$. By Proposition 1.5 this implies $G' \in \mathit{WGCSG}_s$ (since $s(i) = \mathrm{w}(s) \cdot s'(i)$ for every $i \in \mathbb{N}^+$).
♣

For unsteady position valuations this algorithm and similar ones do not work. This can be checked looking at the following interesting example.

Example 2.6 *Let $s = id$, i. e. $s(i) = i$ for every $i \in \mathbb{N}^+$, let $\alpha_1\alpha_2\alpha_3 \to \beta_1\beta_2\beta_3$ be a rule and let*

$$f(\alpha_1) = 1, \quad f(\alpha_2) = 1, \quad f(\alpha_3) = 3, \qquad f(\beta_1) = 1, \quad f(\beta_2) = 5, \quad f(\beta_3) = 1$$

Then the rule is id-weak growing with f, and in the rules $\alpha_1\alpha_2 \to \beta_1 X$ and $X\alpha_3 \to \beta_2\beta_3$ the symbol X can not be valuated appropriately.

Theorem 2.5 can be applied step by step:

Theorem 2.7 *Let s be a steady position valuation. For every grammar $G \in \mathit{WGCSG}_s$ exists an equivalent grammar $G' \in \mathit{WGCSG}_s$ in Cremers normal form.*

Remark: Thus for a steady position valuation s the class WGCSL_s is characterized by the growth factor $\mathrm{w}(s)$ (this follows from Theorem 2.7 and Proposition 1.5). So we introduce the following notion:

$$\mathit{WGCSL}_{\mathrm{w}(s)} := \mathit{WGCSL}_s$$

So when we treat steady position valuations we can assume without loss of generality that s is an exponential function, or, if more convenient, that s is a position valuation with only two valuated positions. Additionally we can assume that it is monotone increasing:

Theorem 2.8 *Let s be a steady position valuation. Then*

$$\mathit{WGCSL}_{\mathrm{w}(s)} = \mathit{WGCSL}_{\frac{1}{\mathrm{w}(s)}}$$

Proof: Let s be a steady position valuation and $G = \langle N, T, S, P \rangle \in WGCSL_s = WGCSL_{\frac{s(2)}{s(1)}}$ in Cremers normal form (Theorem 2.7), and let f be a symbol valuation for G. We just describe the construction of a grammar $G' = \langle N', T, S, P' \rangle \in WGCSG_{\frac{s(1)}{s(2)}}$ that generates $L(G)^T$. Then the claims of the theorem can be concluded easily with Theorem 2.2.

The length preserving rules are just transposed. Now we look at the expanding rules. In a derivation generating a word w there can (at most) $|w|$ expanding rules be used. With every use of an expanding rule a new symbol is generated. Thus, in the transposed version of an expanding rule we append an encoded weight to the new symbol such that the resulting rule becomes s-weak growing with respect to an appropriate extension of $f|_N$. In further rules this appended encoded weight just continues to be glued (at the same position inside a rule). That means, corresponding copies of all other rules are put into P', too. In the terminal rules those appended encoded weights are swallowed (by defining the symbol values of the terminals appropriately). ♣

3 The Unsteady Position Valuation

In this section it turns out that $WGCSL_s = CSL$ for any unsteady position valuation s. The only inclusion we have to show is $CSL \subseteq WGCSL_s$.

It is well known that linear bounded automata characterize the context-sensitive languages ([Kur64]). This will be used extensively when we show the claim mentioned above. There the following technique for simulating a linear bounded automaton by a context-sensitive grammar is adapted (see e. g. [Har78]): We use a linear bounded automaton that has a single tape, writes only on the tape cells marked by the input, and makes no null moves. This can be done without loss of generality (see e. g. [Har78], [HU79], [Tur36]). Each configuration is represented by a certain sentential form (see Fig. 1). We work on three tracks as follows: on the first track we hold the input, on the second track we always have the actual contents of the working tape, and on the third track we have the actual state, put down exactly below the symbol the linear bounded automaton is actually reading.

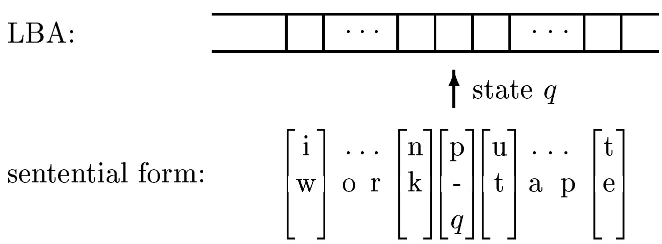

Figure 1: tape and sentential form (sketch)

This sentential form is generated with the initial configuration with any input, then M is simulated step by step, and if M accepts, the triples are transformed into the contents of their first component, which then become terminal symbols.

In the other way, simulating a context-sensitive grammar by a linear bounded automaton we just follow a derivation step by step and compare the derived word with the input afterwards.

As we mentioned before, for any unsteady position valuation holds $WGCSL_s = CSL$. To prove this, we use that every unsteady position valuation s has a blip (see Definition 2.3).

We distinguish four cases for an unsteady position valuation s, where j is the first blip:

(i) $s(j-1) > s(j)$ and $s(j) < s(j+1)$, that is, j is a "valley blip"

(ii) $s(j-1) < s(j)$ and $s(j) > s(j+1)$, that is, j is a "peak blip"

(iii) s is monotone increasing on a beginning part up to the position $j+1$

(iv) s is monotone decreasing on a beginning part up to the position $j+1$

Every unsteady position valuation corresponds to one of the cases above. In each case we introduce for a linear bounded automaton a way of construction for an s-weak growing context-sensitive grammar analogously to the algorithm mentioned above.

To the first case (i):

The "valley blip" in the position valuation is used like follows:
All nonterminal symbols that represent simple tape cells are valuated (by an appropriately defined symbol valuation) with the same value, and the nonterminal symbols to which a state is added are valuated with a greater value. In a rule that simulates a step of the linear bounded automaton, the state "moves" following the move of the automaton's head. Thus, the additional weight (corresponding to the state) also "moves", and by an appropriate left context the rule is constructed such that on the left side the state is positioned exactly on that blip. Since the positions to the right and to the left of this one are both valuated higher, the rule becomes s-weak growing with the symbol valuation mentioned above.

With Figure 2 this is illustrated.

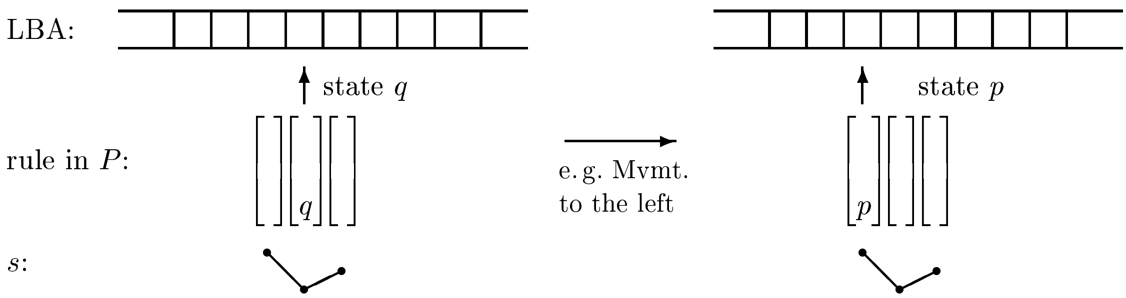

Figure 2: Simulation in the first case (sketch)

To continue working there must exist several symbols in the sentential form on both sides of the state: to the right at least one, to the left at least $j-1$. Therefore at the borders several steps must be condensed, because the state must not be placed in a border symbol in the sentential form since no s-weak growing rule can be built that moves the state out of such a position.

To the second case (ii):

In the case of a "peak blip" we can use the same algorithm as we used in the first case. We just have to redefine the symbol valuation such that the nonterminals representing simple tape cells are valuated higher than the ones containing a state.

To the third case (iii):

Concerning a monotone increasing position valuation (up to the position $j+1$) we relate the state with a greater value (than the simple tape cells) for moving right and with a lower one for moving left. To do this we mark the state with the direction it moved to the last time. With a reversal (change of direction of the move) this value has to be changed. It is no problem in a weak growing grammar to change from a lower to a greater value. But in the other way we will use a trick to erase the "surplus weight". First it is encoded in a certain symbol (we will use a star $\star$) and laid down besides the point of reversal. From there it is hiking to the right to be swallowed.

This is illustrated in Figure 3.

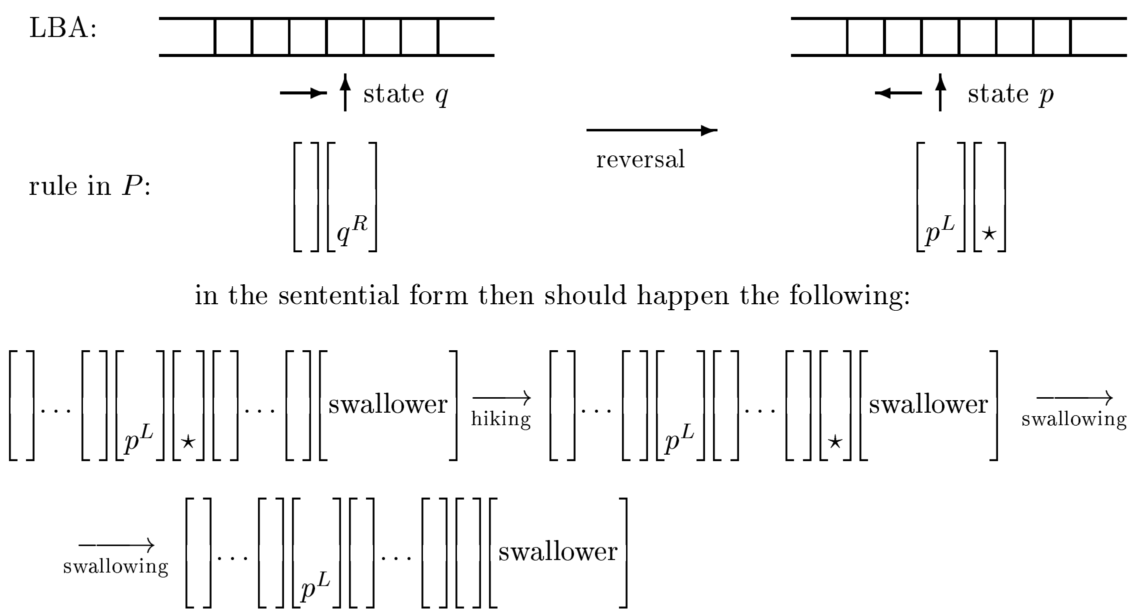

Figure 3: Simulation with swallower

It is no problem to let the "surplus weight" (encoded in the star $\star$) hike to the right.

We now introduce a swallower, that is a string θ together with some rules that can be integrated into a given s-weak growing context-sensitive grammar G such that any sentential form $wY\theta$ can be transformed into $wX\theta$, where the nonterminal X is valuated lower than the nonterminal Y by a symbol valuation for G.

Definition 3.1 *Let s be an unsteady position valuation which is monotone increasing on a beginning part up to its first blip, let $G = \langle N, T, S, P\rangle \in WGCSG_s$. Define a quintuple $\mathcal{S} = (N_1, N_2, N_{\text{swallow}}, \theta, P_{\text{swallow}})$ where N_1 and N_2 are nonempty subsets of N, N_{swallow} is a set of nonterminals (not necessarily a subset of N), $\theta \in N^*_{\text{swallow}}$ is a string and P_{swallow} is a set of rules of the form $\alpha \to \beta$ where $\alpha, \beta \in (N \cup N_{\text{swallow}} \cup T)^*$. $\mathcal{S}_s$ is a* swallower for s and G, *or shorter an s-*swallower for G, *if the following holds:*

- *there exists a symbol valuation $f\colon N \cup N_{\text{swallow}} \cup T \to \mathbb{N}$ with*
  - *f is a symbol valuation for G, i. e. every rule in P is s-weak growing with f*
  - *every rule in P_{swallow} is s-weak growing with f*
  - *for every $X \in N_1, Y \in N_2$ holds $f(X) < f(Y)$*
- *there exists an $l \in \mathbb{N}$ such that for every $w \in (N \cup T)^*$ with $|w| \geq l$ and for every $X \in N_1, Y \in N_2$ holds: $wY\theta \overset{*}{\underset{P_{\text{swallow}}}{\Longrightarrow}} wX\theta$.*

If $N_{\text{swallow}} \subseteq N$ and $P_{\text{swallow}} \subseteq P$, we also call $\mathcal{S}_s$ an s-swallower in G.

To illustrate how a swallower can be built, we give an example first.

Example 3.2 *Take the unsteady position valuation id defined by $id(i) = i$ for every $i \in \mathbb{N}^+$.*
Let $G = \langle N, T, S, P\rangle \in WGCSG_{id}$ with $\{X, Y\} \in N$ and let f be a symbol valuation for G with $f(X) = 1$, $f(Y) = 2$.

Define $\mathcal{S} = (N_1, N_2, N_{\text{swallow}}, \theta, P_{\text{swallow}})$ with $N_1 = \{X\}$, $N_2 = \{Y\}$, $N_{\text{swallow}} = \{X, Y, M, [\star], A, B, V\}$, $\theta = MAV$, and a set of rules $P_{\text{swallow}} = \{(R1), (R2), (R3)\}$ where:

$$(R1)\quad YM \to X[\star] \qquad (R2)\quad [\star]AV \to MBA \qquad (R3)\quad BA \to AV$$

Using this rules out of P_{swallow} the following derivation steps are possible:

$$YMAV \underset{R1}{\Longrightarrow} X[\star]AV \underset{R2}{\Longrightarrow} XMBA \underset{R3}{\Longrightarrow} XMAV$$

Now extend the symbol valuation f to N_{swallow} as follows:

$$\begin{aligned} &f(M) = 1, \quad f([\star]) = 2, \\ &f(A) = 1, \quad f(B) = 8, \quad f(V) = 5 \end{aligned}$$

In this way we obtain the following valuations for the rules in P_{swallow}:

for $(R1)$ **:** $1 \cdot 2 + 2 \cdot 1 = 4 < 1 \cdot 1 + 2 \cdot 2 = 5$

for $(R2)$ **:** $1 \cdot 2 + 2 \cdot 1 + 3 \cdot 5 = 19 < 1 \cdot 1 + 2 \cdot 8 + 3 \cdot 1 = 20$

for $(R3)$ **:** $1 \cdot 8 + 2 \cdot 1 = 10 < 1 \cdot 1 + 2 \cdot 5 = 11$

Thus, these rule are id-weak growing with f.

In the construction of these rules we exploit that with the position valuation id a value moving from the first to the second position can be divided in half (this happens in $(R2)$ with the "value" 1 of the star), and moving from the third to the second position it has to be multiplied only with one and a half (this happens in $(R2)$ with the "value" $f(V)-f(A) = 4$). Here the values are chosen in a way that dividing the value collected in B by 2 (that's $\left\lceil \frac{1}{2} \cdot 1 \right\rceil + \frac{3}{2} \cdot 4 = 7 = f(B) - f(A)$ and happens in $(R3)$) the situation of the beginning is restored.

In this way, the sentential form MAV together with the set of rules P_{swallow} swallows a value encoded in the star, and $\mathcal{S}_s$ is an s-swallower in G.

Now we turn to the construction of a swallower for any unsteady position valuation that is monotone increasing at least up to a position after the first blip.

Theorem 3.3 *Let s be an unsteady position valuation which is monotone increasing on a beginning part up to a position after the first blip, and let $G = \langle N, T, S, P\rangle \in WGCSG_s$. Let f be a symbol valuation for G that valuates at least two symbols out of N differently.*

Then there exists an s-swallower for G.

PROOF: We just introduce the construction of the swallower and how the symbol valuation f can be extended to become an appropriate symbol valuation for the swallower $\mathcal{S} = (N_1, N_2, N_{\text{swallow}}, \theta, P_{\text{swallow}})$ as well.

Let j be the first blip of s.

Let N_1 and N_2 be nonempty subsets of N with $f(X) < f(Y)$ for every $X \in A_1, Y \in A_2$. Define $N_{\text{swallow}} = \{M, [\star], A, B, V\}$, $\theta = MAV$, let

$$l = \begin{cases} j-1 & \text{if } s(j-1) = s(j) \\ j-2 & \text{if } s(j-1) < s(j) \end{cases}$$

and define P_{swallow} as follows:

$$\begin{aligned}
R_1 &= \left\{ \alpha Y M \to \alpha X[\star] \mid \alpha \in (N \cup T)^l,\ Y \in A_2,\ X \in A_1 \right\} \\
R_2 &= \{ \alpha[\star]AV \to \alpha MBA \mid \alpha \in (N \cup T)^{j-2} \} \\
R_3 &= \{ \alpha MBA \to \alpha MAV \mid \alpha \in (N \cup T)^{j-3} \} \\
P_{\text{swallow}} &= R_1 \cup R_2 \cup R_3
\end{aligned}$$

Note that the length of α in R_2 and R_3 is different.

Now we extend the symbol valuation f to N_{swallow} with values in $\mathbb{Q}$ as follows, where $\mu := \max\{f(Y) - f(X) \mid X \in N_1, Y \in N_2\}$:

$$f(M) := 1, \qquad f([\star]) := 1 + \mu, \qquad f(A) := 1$$

$$f(B) := \frac{1}{s(j)^2 - s(j-1) \cdot s(j+1)} \cdot (s(j-1) \cdot s(j) \cdot \mu + s(j) + s(j+1)) + 1$$

$$f(V) := \frac{1}{s(j)^2 - s(j-1) \cdot s(j+1)} \cdot ((s(j-1))^2 \cdot \mu + s(j-1) + s(j)) + 1$$

It can be checked arithmetically that the rules in P_{swallow} all are s-weak growing with f. (In order to check R_2 and R_3 it is helpful to compute first $s(j) \cdot f'(B) - s(j+1) \cdot f'(V)$ and $s(j) \cdot f'(V) - s(j-1) \cdot f'(B)$, respectively.) As we have seen in Proposition 1.6, f can be transformed into a symbol valuation with values in $\mathbb{N}^+$.

Thus f is now a symbol valuation for the swallower as well. Let $w \in (N \cup T)^*$ with $|w| \geq l$, and let $X \in N_1, Y \in N_2$. With $\theta = MAV$ the following derivation steps are possible:

$$wY\theta = wYMAV \underset{R_1}{\Longrightarrow} wX[\star]AV \underset{R_2}{\Longrightarrow} wXMBA \underset{R_3}{\Longrightarrow} wXMAV = wX\theta$$

♣

So we can integrate a swallower into an s-weak growing context-sensitive grammar, where s is unsteady and monotone increasing on a beginning part up to a position after the first blip j, that simulates a linear bounded automaton in the way we mentioned at the beginning of this section. To get a sentential form that is not too long, we condense every two cells of the linear bounded automaton (see linear tape compression in the literature).

When the linear bounded automaton accepts, the sentential form is transformed into terminals and expanded at the same time. Together with the expansion the swallower is swallowed. If s on a beginning part is constant, we have the same conditions for border-handling as in the first and the second case.

To the fourth case (iv):

Concerning an unsteady position valuation that is monotone decreasing on a beginning part up to a position after the first blip j, the simulation of a linear bounded automaton works quite analogously: For a move to the right the state is related to a lower value, for a move to the left the state is related to a higher value than the simple tape cells. That means, the "surplus weight" is generated in a left-right-reversal and hikes to the left, where it is swallowed. The swallower also is constructed in a similar way; the string and the parts of the rules that change are transposed, a dummy symbol is added to the left context of the rules in R_3, and the left context of every rule is replaced by dummy symbols. Such a swallower can be integrated in an analogous way to the one introduced above, and thus the claim can be shown for this case, too.

Since in all cases (i) – (iv) we can construct an appropriate weak growing context-sensitive grammar we have the following:

Theorem 3.4 *For every unsteady position valuation s holds:*

$$WGCSL_s = CSL$$

For a more detailed description see [Nie94a].

4 The Steady Position Valuation

4.1 Efficient Simulation of a Grammar

4.1.1 Connected Grammars

In a simulation of a grammar with a linear bounded automaton, the automaton can follow the derivation step by step, if it is connected (see Definition 4.1), while in a non-connected grammar in every step it has to move to the position of replacement first. To gain an effective simulation we will make use of the property to be connected (see [Gla64], [Boo69], [Bun93]).

Definition 4.1 *Let $G = \langle N, T, S, P\rangle$ be a grammar.*
Let $w_0 = S \Longrightarrow w_1 \Longrightarrow \ldots \Longrightarrow w_t$ be a derivation in G, let $(\alpha_i \to \beta_i) \in P$ the rule applied in the step $w_i \Longrightarrow w_{i+1}$. The derivation is connected*, if for each $i = 1, 2, \ldots, t-1$ the substrings α_i and β_{i-1} of w_i have a nonempty overlap. The grammar G is connected, if each derivation $w_0 \overset{*}{\Longrightarrow} w_t$ with $w_0 = S$ is connected.*

Gladkii showed, that every grammar (of both types: general and context-sensitive) can be transformed into an equivalent one of the same type that is connected ([Gla64]). Using Book's proof in [Boo69], in [Bun93] it was shown that this also is true for quasi growing context-sensitive grammars. Here we want to show the same is true for weak growing context-sensitive grammars.

Book's proof carries over easily when we use that we have Cremers' normal form and can assume, the position valuation is monotone increasing (see Theorems 2.7 and 2.8).

Theorem 4.2 *Let s be a steady strictly monotone increasing position valuation. Then for each grammar $G \in WGCSG_s$ in Cremers' normal form exists a connected grammar $G' \in WGCSG_s$ with $L(G') = L(G)$. Additionally the length of a derivation for a word w in G' is at most doubled compared to the one in G.*

Proof: Let $G = \langle N, T, S, P\rangle \in WGCSG_s$ in Cremers' normal form. Let f be a symbol valuation for G.

Now define a grammar $G' = \langle N', T, \hat{S}, P'\rangle$ by $N' = N \cup \{\hat{A} \mid A \in N\}$, that is the old set of nonterminals together with a set of new nonterminals in 1-to-1 correspondence with the old, and the following rules:

$$P_1 = \left\{ \hat{A}B \to \hat{C}D,\ A\hat{B} \to \hat{C}D,\ A\hat{B} \to C\hat{D} \,\middle|\, A, B, C, D \in N,\ (AB \to CD) \in P \right\}$$
$$P_2 = \{\hat{A} \to \hat{C}D,\ \hat{A} \to C\hat{D} \mid A, C, D \in N,\ (A \to CD) \in P\}$$
$$P_3 = \{\hat{A} \to a,\ \hat{A}X \to a\hat{X} \mid A, X \in N,\ a \in T,\ (A \to a) \in P\}$$
$$P_4 = \{\hat{A}B \to A\hat{B} \mid A, B \in N\}$$
$$P' = P_1 \cup P_2 \cup P_3 \cup P_4$$

Following [Boo69] applies $L(G') = L(G)$, and G' is a connected grammar. Now define a symbol valuation $f'\colon N' \cup T \to \mathbb{N}$ for G' by:

$$\begin{array}{lll} f'(A) = s(2)^2 \cdot f(A) & & \text{for every } A \in N \\ f'(\hat{A}) = s(2)^2 \cdot f(A) + s(2) & & \text{for every } A \in N \\ f'(a) \ = s(2)^2 \cdot f(a) & & \text{for every } a \in T \end{array}$$

Then every production in P' is s-weak growing with f':
Each rule from P grows at least by 1 with respect to f. The factor $s(2)^2$ in the definition of f' causes that this "growth ratio" rises to $s(2)^2$. As $0 < s(2) \leq s(1) \cdot s(2) < s(2)^2$ (because $s(1) \geq 1$ and s is strictly monotone increasing), the claim can be concluded for every rule of P'. To illustrate we have a closer look at the third rule in P_1: Because $AB \to CD$ is s-weak growing with f, applies

$$\begin{array}{rrcl} & s(1) \cdot f(A) + s(2) \cdot f(B) + 1 & \leq & s(1) \cdot f(C) + s(2) \cdot f(D) \\ \Rightarrow & s(1) \cdot s(2)^2 \cdot f(A) + s(2)^3 \cdot f(B) + s(2)^2 & \leq & s(1) \cdot s(2)^2 \cdot f(C) + s(2)^3 \cdot f(D) \end{array}$$

Thus for the left side of the new rule $A\hat{B} \to \hat{C}D$ holds:

$$\begin{aligned} s(1) \cdot f'(A) + s(2) \cdot f'(\hat{B}) &= s(1) \cdot s(2)^2 \cdot f(A) + s(2) \cdot s(2)^2 \cdot f(B) + s(2) \cdot s(2) \\ &\leq s(1) \cdot s(2)^2 \cdot f(C) + s(2) \cdot s(2)^2 \cdot f(D) \\ &< s(1) \cdot s(2)^2 \cdot f(C) + s(1) \cdot s(2) + s(2) \cdot s(2)^2 \cdot f(D) \\ &= s(1) \cdot f'(\hat{C}) + s(2) \cdot f'(D) \end{aligned}$$

Thus applies $G' \in WGCSG_s$. The second claim is true following [Boo69]. ♣

It is possible to use the same idea for other position valuations. We even have connectivity if there is no transformation into Cremers' normal form known.

4.1.2 Efficient Simulation

When we simulate a derivation with a weak growing context-sensitive grammar related to a steady position valuation by a linear bounded automaton, we use the algorithm of Section 3 and Theorem 4.2. In order to estimate the time-bound for the linear bounded automaton, we estimate the length of the derivation in the weak growing context-sensitive grammar related to the steady monotone increasing position valuation by the value of the derived word. This is done by the natural infinite extension of the position valuation. Such an extension allows to valuate arbitrary long sentential forms. With every application of a rule the value of the sentential form increases (at least by one).

Theorem 4.3 *Let s be a steady monotone increasing position valuation. Let $G = \langle N, T, S, P\rangle$ be an s-weak growing context-sensitive grammar, let $u = u_1 \dots u_r$, $v = v_1 \dots v_s \in (N \cup T)^*$ be sentential forms with $u \Longrightarrow v$, let $f\colon N \cup T \to \mathbb{N}^+$ be a symbol valuation for G. Then applies:*

$$\sum_{i=1}^{r} s(i) \cdot f(u_i) + 1 \leq \sum_{i=1}^{s} s(i) \cdot f(v_i)$$

PROOF: Let $r \in P$ be the rule used in the derivation step $u \Longrightarrow v$. The growth ratio in that rule is at least 1, we denote it by g. Let us now valuate the whole sentential forms u and v and look at the growth ratio of the derivation step $u \underset{G}{\Longrightarrow} v$. We obtain a value that differs from g by a factor at least 1, if r is length preserving, and additionally a summand greater 0, if it is length increasing. As there are no length decreasing rules, this implies the estimation in the theorem. ♣

As we saw in Section 3 in the construction of swallowers, in the case of an unsteady position valuation it is not possible to build a natural extension such that the value of a sentential form increases with every application of a weak growing rule. And in the other way because of this effect there cannot exist a swallower for a steady position valuation.

Thus we can estimate the length of a derivation generating a word w in a grammar G by the valuation of the word w itself. Therefor we need an estimation for symbol valuations.

Proposition 4.4 *Let s be a steady strictly monotone increasing position valuation. Let $G = \langle N, T, S, P\rangle$ be an s-weak growing context-sensitive grammar of the order ℓ. Then exists a symbol valuation f for G with* $\max\{f(X) \mid X \in N \cup T\} \leq (\mathrm{w}(s))^{\ell \cdot |N \cup T|}$.

Now we conclude from Theorem 4.3, Definition 2.3 and Proposition 4.4:

Theorem 4.5 *Let s be a steady strictly monotone increasing position valuation. Let $G = \langle N, T, S, P\rangle$ be an s-weak growing context-sensitive grammar of the order ℓ, let $v \in L(G)$. Then the length of a derivation $S \stackrel{*}{\Longrightarrow} v$ is not greater than*

$$\sum_{i=1}^{|v|} s(i) \cdot \mathrm{w}(s)^{\ell \cdot |N \cup T|} = \mathrm{w}(s)^{\ell \cdot |N \cup T|} \cdot \sum_{i=1}^{|v|} s(1) \cdot \mathrm{w}(s)^{i-1} \leq (\mathrm{w}(s))^{|v|} \cdot \frac{(\mathrm{w}(s))^{\ell \cdot |N \cup T|}}{\mathrm{w}(s) - 1} \cdot s(1)$$

Now we can estimate the computation time a linear bounded automaton needs to simulate a weak growing context-sensitive grammar related to a steady position valuation using Theorems 2.8 and 4.5. (Here *T–NSPACE-TIME*(s,t) denotes the class of languages that are accepted by a one-tape nondeterministic Turing machine with the given space bound s and the given time bound t (see [WW86]).)

Theorem 4.6 *Let s be a steady non-constant position valuation.*
Let $\mathrm{w} = \max\{\mathrm{w}(s), \frac{1}{\mathrm{w}(s)}\}$. *Then:*

$$\mathit{WGCSL}_s \subseteq \mathit{T\text{–}NSPACE\text{-}TIME}\,(n, \mathrm{O}(\mathrm{w}^n))$$

4.2 Counters

We will introduce a counter as an instrument to lay down encoded weights, consisting of a string and some rules. We will use it in a simulation of a linear bounded automaton by a weak growing context-sensitive grammar. The capacity of the counter used will cause a bound of the number of steps that can be simulated.

Definition 4.7 *Let s be a steady strictly monotone increasing position valuation, let $G = \langle N, T, S, P \rangle \in WGCSG_s$. Define a quintuple $\mathcal{Z} = \big(N_1, N_2, N_{\text{count}}, (z_n)_{n \in \mathbb{N}^+}, P_{\text{count}}\big)$ where[2] N_1 and N_2 are finite subsets of N, N_{count} is a set of nonterminals (not necessarily a subset of N), $(z_n)_{n\in\mathbb{N}^+}$ is a sequence of start strings with $z_n \in N_{\text{count}}^n$ for every $n \in \mathbb{N}^+$, and P_{count} is a finite subset of $(N_2 \cdot N_{\text{count}}^* \times N_1 \cdot N_{\text{count}}^*) \cup (N_{\text{count}}^* \times N_{\text{count}}^*)$.*

$\mathcal{Z}$ is a counter for G related to the position valuation s *or shorter* s-counter for G, *if the following is true:*

- *every rule in P_{count} is length-preserving*
- *there exists a symbol valuation $f: N \cup N_{\text{count}} \cup T \to \mathbb{N}$ with:*
  - *f is a symbol valuation for G, i. e. every rule in P is s-weak growing with f*
  - *every rule in P_{count} is s-weak growing with f*
  - *for every $X \in N_1, Y \in N_2$ holds $f(X) < f(Y)$*

  $f\mid_{N_1 \cup N_2 \cup N_{\text{count}}}$ is a symbol valuation for $\mathcal{Z}$.

If $N_{\text{count}} \subseteq N$ and $P_{\text{count}} \subseteq P$, we also call $\mathcal{Z}_s$ an s-counter in G.

So the counter counts the weight $f(Y) - f(X)$ by applying an appropriate rule out of $N_2 \cdot N_{\text{count}}^* \times N_1 \cdot N_{\text{count}}^*$ (and after that probably some rules out of $N_{\text{count}}^* \times N_{\text{count}}^*$) on the string Yz_n, where z_n is one of the start strings of $\mathcal{Z}$. This results in a string Xz'_n of the same length. Another weight can be counted applying rules like mentioned above on Yz'_n, and so on. So to use the counter to count several amounts of weight, we apply corresponding increment rules (which are not in $P_{counter}$, but will be added to P in order to use the counter in G): The set $\rho(\mathcal{Z}, f)$ of increment rules for $\mathcal{Z}$ with f *is defined by: $\rho(\mathcal{Z}, f) = \{(X \to Y) \mid X \in N_1,\ Y \in N_2\}$.*

As capacity $K_{\mathcal{Z}}: \mathbb{N}^+ \to \mathbb{N}^+$ *of $\mathcal{Z}$ we declare the function that relates to every index n, how often an increment rule can be used (and the weight "generated" by it can be handled) starting with a string of the form Xz_n, where $X \in N_1$ and z_n is the start string with index (and length) n, where "collecting" of weights is allowed (i. e. using several increment rules directly after another). Formally:*

$$K_{\mathcal{Z}}(n) = \max_{f \text{ symb. val. for } \mathcal{Z}} \left\{ i \in \mathbb{N} \;\middle|\; \begin{array}{l} \exists M \in N_1 \;\; \exists \tilde{z} \in N_{\text{count}}^n \;\; \exists k \in \mathbb{N} \text{ such} \\ \text{that } (r_1, \ldots, r_k) \in (P_{\text{count}} \cup \rho(\mathcal{Z}, f))^k \text{ is a se-} \\ \text{quence of rules with } Mz_n \underset{r_1}{\Longrightarrow} \ldots \underset{r_k}{\Longrightarrow} M\tilde{z} \text{ and} \\ \big|\{j \in \{1, \ldots, k\} \mid r_j \in \rho(\mathcal{Z}, f)\}\big| = i \end{array} \right\}$$

Because the set N is finite and each of the rules is length preserving, the capacity of a counter certainly is well-defined. For every counter $\mathcal{Z}$ and every $c \in \mathbb{N}^+$ there is (for the possibility of "collecting" weights) a counter $\mathcal{Z}'$ with $K_{\mathcal{Z}'} = c \cdot K_{\mathcal{Z}}$.

We now check the possible capacity for a counter that follows the algorithm: treat the string as a number to a certain base and add up arithmetically. We look at an example first.

[2] We preferred the symbol "$\mathcal{Z}$" for counters that abbreviates the German word "Zähler", as the abbreviation "$\mathcal{C}$" often is used for classes and "C" for constants or constant functions.

Example 4.8 *Let s_2 be the position valuation defined by $s_2(i) = 2^{i-1}$ for every $i \in \mathbb{N}^+$. Define $\mathcal{Z} = \left(N_1, N_2, N_{\text{count}}, (z_n)_{n\in\mathbb{N}^+}, P_{\text{count}}\right)$ by: $N_1 = \{M\}$, $N_2 = \{\#\}$, $N_{\text{count}} = \{[0],[1],[U]\}$, $z_n = [0]^n$ for each $n \in \mathbb{N}^+$ and R consisting of the rules:*

$$\begin{array}{llll} (1) & \#[0] \to M[1] & (3) & [U][0] \to [0][1] \\ (2) & \#[1] \to M[U] & (4) & [U][1] \to [0][U] \end{array}$$

Together with the rule $M \to \#$, for every start string z_n the following derivation steps are possible in $\mathcal{Z}$:

$$\begin{aligned} Mz_n &= M[0]\ldots[0] \overset{*}{\Longrightarrow} M[1][0]\ldots[0] \overset{*}{\Longrightarrow} M[U][0]\ldots[0] \\ &\overset{*}{\Longrightarrow} M[0][1][0]\ldots[0] \overset{*}{\Longrightarrow} M[1][1][0]\ldots[0] \overset{*}{\Longrightarrow} M[U][1][0]\ldots[0] \overset{*}{\Longrightarrow} M[0][U][0]\ldots[0] \\ &\overset{*}{\Longrightarrow} M[0][0][1][0]\ldots[0] \\ &\ldots \overset{*}{\Longrightarrow} M[U][U]\ldots[U] \end{aligned}$$

That seems to be a nice counter with the capacity

$$2 \cdot \sum_{i=1}^{n} 2^{i-1} = \sum_{i=1}^{n} 2^i = 2^{n+1} - 2 \geq 2^n$$

Now let $f : N_1 \cup N_2 \cup N_{\text{count}} \to \mathbb{N}$ be a symbol valuation. Then we have:

$$\begin{array}{lll} \textbf{out of rule } (3) & : & 1\cdot f([U]) + 2\cdot f([0]) < 1\cdot f([0]) + 2\cdot f([1]) \\ & & \Rightarrow \quad f([U]) - f([1]) < f([1]) - f([0]) \\ \textbf{and out of rule } (4) & : & 1\cdot f([U]) + 2\cdot f([1]) < 1\cdot f([0]) + 2\cdot f([U]) \\ & & \Rightarrow \quad f([1]) - f([0]) < f([U]) - f([1]) \end{array}$$

Both of these inequalities cannot be fulfilled at the same time. Thus there exists no symbol valuation all rules of P_{count} are s_2-weak growing with.

This implies that $\mathcal{Z}$ is no s_2-counter.

The reason is, that in every rule we transport a portion of the weight plus an increment for making the rule weak growing to the right. When all these portions and increments come together, they add up to a value that must not exceed the original weight.

Theorem 4.9 *Let s be an infinite steady strictly monotone increasing position valuation, let $l \in \mathbb{N}$. Let $\mathcal{Z} = \left(N_1, N_2, N_{\text{count}}, (z_n)_{n\in\mathbb{N}^+}, P_{\text{count}}\right)$ be an s-counter working after the following scheme (where $[i]$ denotes a digit represented by l letters from N_{count}, and $M \in N_1, \# \in N_2$);*

$$\begin{array}{lll} (R1) & \#[0] \to M[1], & [U][0] \to [0][1] \\ (R2) & \#[i] \to M[i+1], & [U][i] \to [0][i+1] \quad \textit{for } i = 1,\ldots,i_{max}-1 \\ (R3) & \#[i_{max}] \to M[U], & [U][i_{max}] \to [0][U] \end{array}$$

that is, $\mathcal{Z}$ treats the start string like a number to a certain base (here $i_{max}+1$) and increments arithmetically (without collecting weights), where $[U]$ denotes the overflow (representing $[0]$ and a carry at the same time) and $i_{max} \in \mathbb{N}^+$.

⟶

Then the counter's capacity can be estimated by

$$K_{\mathcal{Z}}(n) \leq 2 \cdot c \cdot \left(\left\lceil \mathrm{w}(s)^l - 1 \right\rceil\right)^{\left\lfloor \frac{n}{l} \right\rfloor}$$

where l *is such that* $\mathrm{w}(s)^l \geq 3$*, and* c *is a constant defined by* $\mathcal{Z}$*.*
That is, i_{max} *is bounded, and we estimate the bound.*

Proof: Let f be a symbol valuation for $\mathcal{Z}$. Define a function $g\colon N^l \to \mathbb{N}^+$ by $g(a_1 \ldots a_l) = \sum_{i=1}^{l} s(i) \cdot f(a_i)$, where $a_1, \ldots, a_l \in N$, that is, g valuates the digits, each represented by l symbols, by condensing the symbol and the position valuation. Let w denote the growth factor $\mathrm{w}(s)$. Then we have

out of rules $(R1)$**,** $(R2)$ **:** $g([U]) - g([0]) < \mathrm{w}^l \cdot (g([i+1]) - g([i]))$ for $i = 0, \ldots, i_{max} - 1$
and out of rule $(R3)$ **:** $\underbrace{g([U]) - g([0])}_{=:\mu} < \mathrm{w}^l \cdot (g([U]) - g([i_{max}]))$

This implies

$$\begin{aligned} \mu = g([U]) - g([0]) &= g([U]) - g([i_{max}] + g([i_{max}] - + \ldots + g([1]) - g([0]) \\ &> \underbrace{\tfrac{1}{\mathrm{w}^l} \cdot \mu + \ldots + \tfrac{1}{\mathrm{w}^l} \cdot \mu}_{i_{max}+1 \text{times}} \\ &= \tfrac{i_{max}+1}{\mathrm{w}^l} \cdot \mu \end{aligned}$$

And this implies $i_{max} + 1 < \mathrm{w}^l \quad \Rightarrow \quad i_{max} = \left\lceil \mathrm{w}^l - 2 \right\rceil$ (which implies $\mathrm{w}^l \geq 3$).

Thus we have a number representation to the base $\left\lceil \mathrm{w}^l - 1 \right\rceil$. We can fill up the string with overflows "$[U]$", and then no rule is applicable any more. Since in a string of the length n we have $\left\lfloor \frac{n}{l} \right\rfloor$ digits, and we can use the symbols that might not be used in a digit (this occurs if n is not divisible by l) only to collect a constant number of weights, the capacity of the counter can be estimated by:

$$\begin{aligned} K_{\mathcal{Z}}(n) &= c \cdot \left\lceil \mathrm{w}^l - 1 \right\rceil \cdot \sum_{i=1}^{\left\lfloor \frac{n}{l} \right\rfloor} \lceil \mathrm{w}^l - 1 \rceil^{i-1} = c \cdot \frac{\lceil \mathrm{w}^l - 1 \rceil^{\left\lfloor \frac{n}{l} \right\rfloor} - 1}{\lceil \mathrm{w}^l - 1 \rceil - 1} \cdot \left\lceil \mathrm{w}^l - 1 \right\rceil \\ &= c \cdot \left(\left\lceil \mathrm{w}^l - 1 \right\rceil^{\left\lfloor \frac{n}{l} \right\rfloor} - 1 + \frac{\lceil \mathrm{w}^l - 1 \rceil^{\left\lfloor \frac{n}{l} \right\rfloor} - 1}{\lceil \mathrm{w}^l - 1 \rceil - 1} \right) \leq 2 \cdot c \cdot \left\lceil \mathrm{w}^l - 1 \right\rceil^{\left\lfloor \frac{n}{l} \right\rfloor} \end{aligned}$$

♣

The optimum mentioned in the theorem above can really be reached. We put Theorem 4.10 with the exponent rounded up instead of rounded down because of technical reasons in the use of the theorem.

Theorem 4.10 *Let* s *be an infinite steady strictly monotone increasing position valuation, let* $l \in \mathbb{N}$ *such that* $\mathrm{w}(s)^l \geq 3$*, let* $c \in \mathbb{N}^+$*. Then there exists an* s*-counter* $\mathcal{Z}$*, that has a capacity with*

$$K_{\mathcal{Z}}(n) \geq c \cdot \left\lceil \mathrm{w}(s)^l - 1 \right\rceil^{\left\lceil \frac{n}{l} \right\rceil} \quad \textit{for } n \in \mathbb{N}^+,\ n \geq l$$

Proof: We define an s-counter $\mathcal{Z} = \left(N_1, N_2, N_{\text{count}}, (z_n)_{n \in I\!N^+}, P_{\text{count}}\right)$ following the scheme in Theorem 4.9. We will use the symbols that are not used in a digit, which only occurs if n is not divisible by l, to use a "cheated" additional digit in this case.
Thus we define: $N_1 = \{M\}$, $N_2 = \{\#\}$, $N_{\text{count}} = \{D, [0], [1], \ldots, [i_{max}], [U]\} \cup \{\langle 0\rangle, \langle 1\rangle, \ldots, \langle i_{max}\rangle, \langle U\rangle\}$ where $i_{max} = \left\lceil \mathrm{w}(s)^l - 2\right\rceil$. Each digit "$i$" is represented by $D^{l-1}[i]$, and the start strings are $z_n = \left(D^{l-1}[0]\right)^{\left\lfloor \frac{n}{l}\right\rfloor} \langle 0\rangle^{\,n \bmod l}$ for every $n \in I\!N$. The set P_{count} consists of the following rules (for $i = 1, \ldots, i_{max} - 1$):

$(R1)$	$\#D^{l-1}[0] \to MD^{l-1}[1]$,	$[U]D^{l-1}[0] \to [0]D^{l-1}[1]$,	$[U]\langle 0\rangle \to [0]\langle 1\rangle$
$(R2)$	$\#D^{l-1}[i] \to MD^{l-1}[i+1]$,	$[U]D^{l-1}[i] \to [0]D^{l-1}[i+1]$,	$[U]\langle i\rangle \to [0]\langle i+1\rangle$
$(R3)$	$\#D^{l-1}[i_{max}] \to MD^{l-1}[U]$,	$[U]D^{l-1}[i_{max}] \to [0]D^{l-1}[U]$,	$[U]\langle i_{max}\rangle \to [0]\langle U\rangle$

Now we define a symbol valuation $f\colon N_1 \cup N_2 \cup N_{\text{count}} \to I\!N^+$ by (where $i = 1, \ldots i_{max}$):

$$\begin{array}{lll} f(M) = 1 & f(D) = 1 & f(\#) = \left\lceil \mathrm{w}(s)^l \right\rceil \\ f([0]) = 1 & f([i]) = i + 1 & f([U]) = \left\lceil \mathrm{w}(s)^l \right\rceil \\ f(\langle 0\rangle) = 1 & f(\langle i\rangle) = i \cdot \left\lceil \mathrm{w}(s)^l \right\rceil & f(\langle U\rangle) = \left(\left\lceil \mathrm{w}(s)^l \right\rceil - 1\right) \cdot \left\lceil \mathrm{w}(s)^l \right\rceil \end{array}$$

It can be checked easily that the given rules all are s-weak growing with f.

The rules in the third column give us the possibility to use a "cheated" additional digit, if l does not divide n. The capacity now is calculated just as in the proof of Theorem 4.9, thus applies $K_{\mathcal{Z}}(n) \geq \left\lceil \mathrm{w}(s)^l - 1\right\rceil^{\left\lceil \frac{n}{l} \right\rceil}$. By collecting weights we get a counter $\mathcal{Z}'$ with $K_{\mathcal{Z}'} = c \cdot K_{\mathcal{Z}}$ for any $c \in I\!N^+$. ♣

4.3 Simulation of Time-Bounded Linear Automata

The counters introduced in Theorem 4.10 now will be used in a simulation of a time-bounded linear bounded automaton instead of the swallower we used in Section 3. Out of the capacity of the counter used, the time-bounds for linear bounded automata that can be managed with this counter result.

Theorem 4.11 *Let* $\mathrm{w} \in I\!N^+$, *let* $q \in \mathbb{Q}$ *with* $0 < q < 1$.
Then:

$$T\text{–}NSPACE\text{-}TIME\left(n, \mathrm{O}\left(\mathrm{w}^{q \cdot n}\right)\right) \subseteq WGCSL_{\mathrm{w}}$$

Proof: We just describe the algorithm how the grammar is constructed for a given $\mathrm{w} \in I\!N^+$ and a given linear bounded automaton. The construction itself then is straight forward.

Let M be any linear bounded automaton that recognizes its language $L(M)$ in the time $c \cdot \mathrm{w}^{q \cdot n}$ with an appropriate $c \in I\!N^+$, and let s be a steady position valuation with $\mathrm{w}(s) = \mathrm{w}$.

Similar to the third case (iii) in the proof of Theorem 3.4 we divide the sentential form into two parts: In the left one the step-by-step simulation of the automaton takes place in

$$\text{sentential form (scheme):}\quad \underbrace{\overbrace{\left[\begin{array}{c}\text{LBA - Sim}\\ \text{compressed}\end{array}\right]}^{\frac{1}{k}\cdot n}\ \overbrace{\underbrace{D^{l-1}[i_1]}_{\text{lowest digit}}\ \underbrace{D^{l-1}[i_2]}_{\text{2. digit}}\ \cdots\ \underbrace{D^{l-1}[i_{max}]}_{\text{highest digit}}}^{\frac{k-1}{k}\cdot n}}_{\text{length of the derived word (i. e. the original input)}}$$

Figure 4: Simulation with a steady position valuation (sketch)

the same manner as in the simulation in the proof mentioned above, except that we use tape compression; in the right one a counter is situated.

We choose an appropriate compression factor k such that $\frac{k}{k-1} \cdot q < 1$, and to this k we choose the width l of a digit such that $\left\lceil \mathrm{w}(s)^l \right\rceil \geq \mathrm{w}(s)^{l \cdot q \cdot \frac{k}{k-1}} + 1$ and at the same time $\mathrm{w}(s)^l \geq 3$. Note that the first expression is equivalent to $\left\lceil \mathrm{w}(s)^l - 1 \right\rceil^{\frac{k-1}{k} \cdot \frac{1}{l}} \geq \mathrm{w}(s)^q$.

Following Theorem 4.10 we can find an s-counter $\mathcal{Z}$ with the capacity $K_{\mathcal{Z}}(m) \geq c \cdot \left\lceil (\mathrm{w}(s))^l - 1 \right\rceil^{\left\lceil \frac{m}{l} \right\rceil}$ where $m = \left\lceil \frac{k-1}{k} \right\rceil \cdot n$ is the width of the counter used in the simulation for an input of the length n. This implies

$$K(m) \geq c \cdot (\left\lceil \mathrm{w}(s)^l - 1 \right\rceil)^{\left\lceil \frac{k-1}{k} \cdot \frac{n}{l} \right\rceil} \geq c \cdot \mathrm{w}(s)^{q \cdot n}$$

That means, if M accepts an input word, the simulation can be completed. Then the k-tuples of the compressed LBA-simulation are transformed into terminals and expanded at the same time. Together with the expansion the counter is swallowed. ♣

This means a gap between Theorem 4.6 and Theorem 4.11.

As the construction technique used to build a counter with exponential capacity does not lead further (see Theorem 4.9), to close this gap is a nontrivial problem.

5 Characterizing the Exponential Time Hierarchy of *CSL*

In the previous sections we had a look on several position valuations, structured and investigated them. Now we want to gain an overview about the different classes of weak growing context-sensitive languages related to different position valuations.

For a constant position valuation we obtain the same grammars as if we had no position valuation at all, namely the quasi growing context-sensitive grammars, which characterize *GCSL* ([BL92], compare Section 1).

For any unsteady position valuation we obtain the language class *CSL*, as we saw in Section 3, and the same holds for position valuations with zero points, as we mentioned before (see Section 1). We give an outline of the proof here:

Theorem 5.1 *Let s be a position valuation with zero points. Then:*

$$\mathit{WGCSL}_s = \mathit{CSL}$$

PROOF: (sketch) It is sufficient to show $CSL \in WGCSL_s$.
We can assume that there exists a zero point j with $s(j+1) = c \neq 0$, as the definition of $WGCSG_s$ implies that s is not constantly 0, $WGCSL_s$ is closed under transposition (Theorem 2.2 holds) and Theorem 2.8 can be adapted. Regarding these two positions, the "growth factor" in fact is not defined, but it can be seen as arbitrary large. Thus, we can construct a counter similar to the one in Section 4.2 that counts in an arbitrarily given number representation. (For $j \neq 1$ we add a left context of length $j-1$ to the rules in the proof of Theorem 4.10, we can choose $l = 1$, and the symbol valuation is constructed in the same way as it was there.) Now given a linear bounded automaton with the exponential time bound $b^n = (b^2)^{\frac{1}{2}\cdot n}$, $b \geq 2$, we construct a grammar similar to the one in the proof of Theorem 4.11, where we choose $q = \frac{1}{2}, k = 2$, and b^2 instead of $\lceil \mathrm{w}(s) - 1 \rceil$ is the base of the number representation the counter uses. Thus, the linear bounded automaton can be simulated by an s-weak growing context-sensitive grammar. ♣

For every steady position valuation s we saw in Section 2.2 that the corresponding class of weak growing context-sensitive languages is characterized by the growth factor $\mathrm{w}(s)$, and in the Theorems 4.6 and 4.11 was shown:

Theorem 5.2 *Let s be a steady non-constant position valuation. Define* $\mathrm{w} := \max\left\{\mathrm{w}(s), \frac{1}{\mathrm{w}(s)}\right\}$. *For every $q \in \mathbb{Q}^+$, $0 < q < 1$ holds:*

$$T\text{–}NSPACE\text{-}TIME\,(n, \mathrm{O}\,(\mathrm{w}^{q\cdot n})) \subseteq WGCSL_s = WGCSL_{\mathrm{w}} \subseteq T\text{–}NSPACE\text{-}TIME\,(n, \mathrm{O}\,(\mathrm{w}^{n}))$$

This implies $WGCSL_{\mathrm{w}} \subsetneq WGCSL_{\mathrm{v}}$ for $\mathrm{v} \in \mathbb{N}^+$, $\mathrm{v} > \mathrm{w}$.

The resulting relations for different growth factors are depicted in Figure 5.

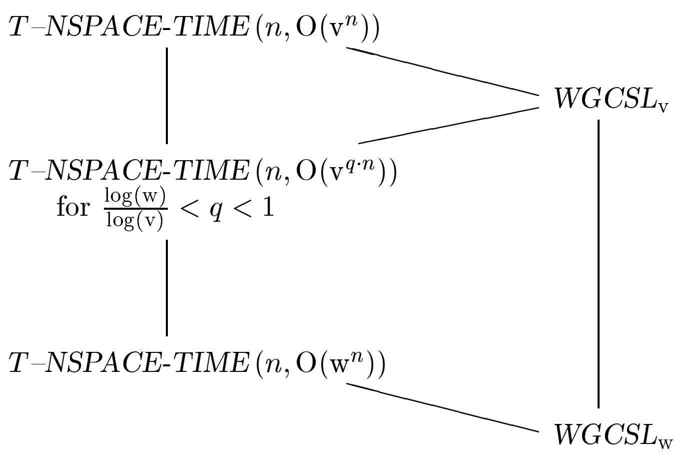

Figure 5: Relations for growth factors $\mathrm{v} > \mathrm{w} > 1$

As every polynomial time bound can be exceeded by any exponential time bound, it can be concluded:

Theorem 5.3 *For every steady non-constant position valuation s holds:*

$$T\text{–}NSPACE\text{-}TIME\,(n, pol) \subseteq WGCSL_s$$

Thus we have all the relations depicted in Figure 6 (p. 27), where we concentrated on some specific classes because of clarity.

Thus the weak growing context-sensitive languages related to steady position valuations build a linearly inclusion-ordered hierarchy that characterizes the exponential time hierarchy for context-sensitive languages in the sense that the one collapses to a certain level if and only if the other collapses to a corresponding level.

Additionally, with the counters we introduced in Section 4.2 and the idea of construction in the proof of Theorem 4.11 it can be shown:

Theorem 5.4 *Let s be a steady non-constant position valuation. The closure of $WGCSL_s$ under inverse homomorphism [k-bounded homomorphism] is CSL.*

Proof: Let $L \in CSL$, and let M be a linear bounded automaton that accepts L in the time $t \cdot 2^{c \cdot n}$. Let Σ be the input alphabet.

Analogously to the grammar in the proof of Theorem 4.11 we construct an s-weak growing context-sensitive grammar G that simulates M and uses a counter just like the one we used in the proof we mentioned above.

For an input word of length n we choose a start string of the length m for the counter, where $m \geq \frac{l \cdot c}{\log(\mathrm{w}(s) - 1)} \cdot n$ and l is chosen such that $\mathrm{w}(s)^l \geq 3$. Thus we reach at least the capacity

$$\begin{aligned} K_{\mathcal{Z}}(m) &\geq t \cdot \left(\mathrm{w}(s)^l - 1\right)^{\left\lceil \frac{m}{l} \right\rceil} \geq t \cdot \left(\mathrm{w}(s)^l - 1\right)^{\left\lceil \frac{l \cdot c}{l \cdot \log(\mathrm{w}(s)^l - 1)} \cdot n \right\rceil} \\ &\geq t \cdot 2^{\log(\mathrm{w}(s)^l - 1) \cdot \left\lceil \frac{c}{\log(\mathrm{w}(s)^l - 1)} \cdot n \right\rceil} \geq t \cdot 2^{c \cdot n} \end{aligned}$$

that means, if M accepts an input word, its work is simulated completely.

Now we can expand each symbol $a \in \Sigma$ to $a\Diamond^{\left\lceil \frac{l \cdot c}{\log(\mathrm{w}(s)^l - 1)} \right\rceil}$. Together with the expansion the counter is swallowed.

We define a homomorphism $h \colon \Sigma^* \to (\Sigma \cup \{\Diamond\})^*$ by $h(a) = a\Diamond^{\left\lceil \frac{l \cdot c}{\log(\mathrm{w}(s)^l - 1)} \right\rceil}$ for every $a \in \Sigma$.
[a homomorphism $h \colon (\Sigma \cup \{\Diamond\})^* \to \Sigma^*$ by $h(a) = a$ for every $a \in \Sigma$, $h(\Diamond) = \varepsilon$.]
Then $h(L) = L(G)$.
[Then h is k-bounded for $L(G)$ with $k = \left\lceil \frac{l \cdot c}{\log(\mathrm{w}(s)^l - 1)} \right\rceil$, and $h(L(G)) = L$.] ♣

We point out that s is non-constant, not only because the counter is not defined otherwise but also because *GCSL* is closed under inverse homomorphism [BL92] (and thus also under k-bounded homomorphism, [GGH69]).

We obtain the following connections between the closure of $WGCSL_s$ under inverse homomorphism for steady position valuations s, the transformability of a grammar from $WGCSG_s$ into Cremers normal form for unsteady position valuations s and the exponential time hierarchy of *CSL*.

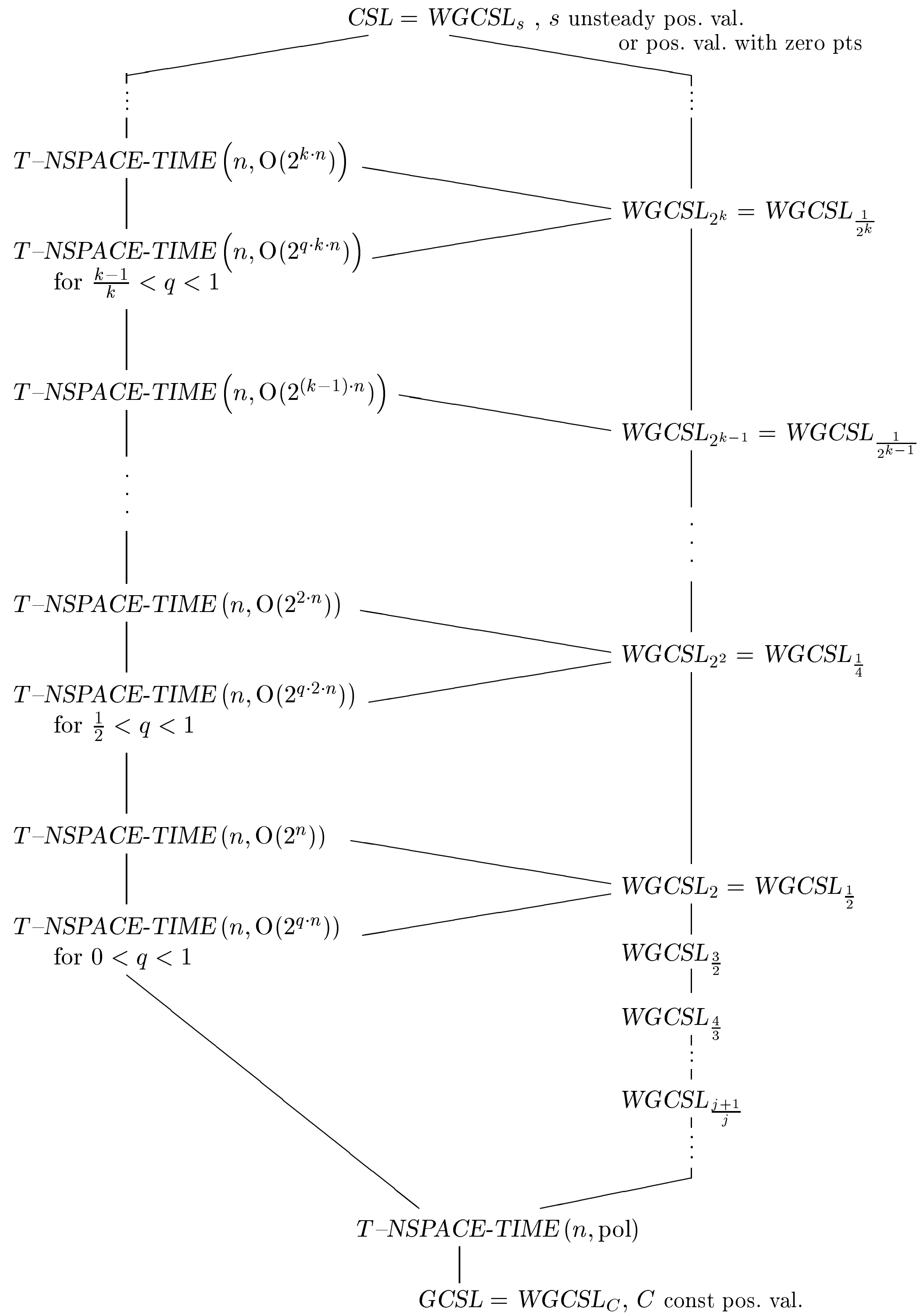

Figure 6: The exponential time hierarchy of *CSL*
and the hierarchy of weak growing context-sensitive languages

Theorem 5.5 *Let* $\mathrm{w} > 1$*, let* s *be a steady position valuation with* $\mathrm{w}(s) = \mathrm{w}$*. The following statements are equivalent:*

a) $WGCSL_{\mathrm{w}} = CSL$

b) $WGCSL_{\mathrm{w}}$ *is closed under inverse homomorphism*

c) For every unsteady position valuation $\tilde{s}$ *holds:*

  $(*)$ *For every grammar* $G \in WGCSG_{\tilde{s}}$ *exists a grammar* $G' \in WGCSG_s$ *in Cremers' normal form with* $L(G') = L(G)$

d) For a certain unsteady position valuation $\tilde{s}$ *holds* $(*)$*.*

Additionally a) implies:

e) $T\text{–}NSPACE\text{-}TIME(n, \mathrm{O}(\mathrm{w}^n)) = T\text{–}NSPACE(n)$

and from this can be concluded:

f) $WGCSL_{\mathrm{v}} = CSL$ *for every* $\mathrm{v} > \mathrm{w}$

6 Conclusion

We have seen a characterization of *CSL* by weak growing context-sensitive grammars related to an unsteady position valuation (Section 3) and a characterization of the exponential time hierarchy of *CSL* by classes of weak growing context-sensitive languages related to steady position valuations (Section 5).

As interesting we consider the question, whether the exponential time hierarchy for linear bounded automata collapses.

Theorem 5.5 means the equivalence of the following problems for a steady position valuation s:

- Does for any $\tilde{s}$-weak growing context-sensitive grammar related to an unsteady position valuation $\tilde{s}$ exist an equivalent one in a normal form of bounded order that is weak growing context-sensitive related to the steady position valuation s?
- Is $WGCSL_s$ is closed under inverse homomorphism?
- Does the hierarchy of weak growing context-sensitive language classes corresponding to the different growth factors collapse down to $WGCSL_{\mathrm{w}(s)}$?
  That is: Do *CSL* and $WGCSL_s$ coincide?
- Does the exponential time hierarchy for *CSL* collapse down to the level of $\mathrm{w}(s)^n$?
  That is: Do *CSL* and $T\text{–}NSPACE\text{-}TIME(n, \mathrm{O}(\mathrm{w}(s)^n))$ coincide?

If in the first question we restrict ourselves on an arbitrary position valuation instead of a steady one, we also have to restrict ourselves on normal forms of order 2 (and thus result in a steady position valuation again). The reason lies in the fact that for any context-sensitive grammar exists an equivalent one of the order 3 that is in $WGCSL_s$, where s is an arbitrary unsteady position valuation with three valuated positions. This can be shown in the following way:

> We restrict ourselves to the case $s(1) < s(2)$. The others work analogously. Given a context-sensitive grammar G. First G is transformed into an equivalent grammar G' in Cremers' normal form. Then we glue an encoded weight to every rule such that all the rules become weak growing related to s with an appropriately defined symbol valuation f (this idea is similar to the one in the proof of Theorem 2.2). Then we add rules to let those weights hike to the right and a swallower (compare the third case (iii) in the proof of Theorem 3.4). Thus every step of the original derivation can be simulated by using the corresponding rule of the new grammar, moving the encoded weight to the right and swallowing it.

Because of Theorem 2.7 we know that asking for a normal form of bounded order in the case of a steady position valuation is equivalent to asking for an order-2-normal form.

Remember that our notion of a characterization of a hierarchy by another only means that the one collapses if and only if the other does.

In this context it would be interesting to close the gap between Theorem 4.6 and Theorem 4.11, that is to answer the question:

> Is $WGCSL_{\mathrm{w}} = T\text{–}NSPACE\text{-}TIME\,(n, \mathrm{O}(\mathrm{w}^n))$?

Acknowledgements

We thank Hans-Jörg Burtschick for pointing out a former version of Example 2.6. Also we are grateful to Carmen Schwenkel for allowing us to include some results of her thesis paper ([Sch93]), extending them for arbitrary steady position valuations (Example 1.3, Theorem 2.1, Theorem 2.7 and Theorem 4.4, also the idea to build quasi weak growing grammars), and of her pre-thesis paper [Sch92], also extended (the separation of $WGCSG_s$ and CSG in Section 1).

References

[BL92] Gerhard Buntrock and Krzysztof Loryś. On growing context-sensitive languages. In *Proceedings of the 19th International Colloquium on Automata, Languages and Programmaing*, volume 623 of *Lecture Notes of Computer Science*, pages 77–88. Springer, 1992.

[BL94] Gerhard Buntrock and Krzysztof Loryś. The variable membership problem: Succinctness versus complexity. In *Proceedings of the 11th Symposium on Theoretical Aspects of Computer Science*, volume 775 of *Lecture Notes of Computer Science*, pages 595–606. Springer, 1994.

[BO95] Gerhard Buntrock and Friedrich Otto. Growing context-sensitive languages and Church-Rosser languages. In *Proceedings of the 12th Symposium on Theoretical Aspects of Computer Science*, volume 900 of *Lecture Notes of Computer Science*, pages 313–324. Springer, 1995.

[Boo69] Ronald V. Book. *Grammars with Time Functions*. Dissertation, Harvard University, Cambridge, Massachusetts, February 1969.

[Bun93] Gerhard Buntrock. Growing context-sensitive languages and automata. Technical Report 69, Universität Würzburg, September 1993.

[Cho59] Noam Chomsky. On certain formal properties of grammars. *Information and Control*, 2:137–167, 1959.

[Cre73] Armin 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 Science*, 33:456–472, 1986.

[GGH69] Seymour Ginsburg, Sheila Greibach, and John E. Hopcroft. Studies in abstract families of languages. *American Mathematical Society*, 87, 1969.

[Gla64] Aleksey Vsevolodovich Gladkii. On the complexity of derivations in context-sensitive grammars. *Algebri i Logika Sem.*, 3:29–44, 1964. In Russian.

[Har78] Michael A. Harrison. *Introduction to Formal Language Theory*. Series in Computer Science. Addison-Wesley, 1978.

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

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

[Nie92] Gundula Niemann. The length of growing grammars that were produced out of quasi growing ones. Studienarbeit, Universität Würzburg, December 1992. In German.

[Nie94a] Gundula Niemann. Characterizations with weak growing context-sensitive grammars. Diplomarbeit, Universität Würzburg, March 1994. In German.

[Nie94b] Gundula Niemann. Quasi weak growing grammars characterize the recursive enumerable sets. Manuscript, in German, January 1994.

[Pau78] Wolfgang J. Paul. *Komplexitätstheorie.* B. G. Teubner, Stuttgart, 1978.

[Pip79] Nicholas Pippenger. On simultaneous resource bounds. In *Proc. of 20th Annual Symposium on Foundations of Computer Science*, 1979.

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

[Sch92] Carmen Schwenkel. Some complexity theoretical investigations of context-sensitive grammars with valuation functions. Studienarbeit, Universität Würzburg, 1992. In German.

[Sch93] Carmen Schwenkel. Properties of weak growing context-sensitive languages. Diplomarbeit, Universität Würzburg, 1993. In German.

[Sud78] Ivan H. Sudborough. On the tape complexity of deterministic context–free languages. *Journal of the ACM*, 25(3):405–414, 1978.

[Tur36] Alan Mathison Turing. On computable numbers, with an application to the entscheidungsproblem. *Proc. London Mathematical Society*, 2:230–265, 1936. A correction to this paper has appeared in the same periodical, 43:544–546 (1937).

[WW86] Klaus Wagner and Gerd Wechsung. *Computational Complexity.* VEB Deutscher Verlag der Wissenschaften and Reidel Verlag, Dordrecht, 1986.

Preprint-Reihe
Institut für Informatik
Universität Würzburg

Verantwortlich: Die Vorstände des Institutes für Informatik.

[1] K. Wagner. *Bounded query classes.* Februar 1989.

[2] P. Tran-Gia. *Application of the discrete transforms in performance modeling and analysis.* Februar 1989.

[3] U. Hertrampf. *Relations among mod-classes.* Februar 1989.

[4] K. W. Wagner. *Number-of-query hierarchies.* Februar 1989.

[5] E. W. Allender. *A note on the power of threshold circuits.* Juli 1989.

[6] P. Tran-Gia und Th. Stock. *Approximate performance analysis of the DQDB access protocol.* August 1989.

[7] M. Kowaluk und K. W. Wagner. *Die Vektor-Sprache: Einfachste Mittel zur kompakten Beschreibung endlicher Objekte.* August 1989.

[8] M. Kowaluk und K. W. Wagner. *Vektor-Reduzierbarkeit.* August 1989.

[9] K. W. Wagner (Herausgeber). *9. Workshop über Komplexitätstheorie, effiziente Algorithmen und Datenstrukturen.* November 1989.

[10] R. Gutbrod. *A transformation system for chain code picture languages: Properties and algorithms.* Januar 1990.

[11] Th. Stock und P. Tran-Gia. *A discrete-time analysis of the DQDB access protocol with general input traffic.* Februar 1990.

[12] E. W. Allender und U. Hertrampf. *On the power of uniform families of constant depth threshold circuits.* Februar 1990.

[13] G. Buntrock, L. A. Hemachandra und D. Siefkes. *Using inductive counting to simulate nondeterministic computation.* April 1990.

[14] F. Hübner. *Analysis of a finite capacity a synchronous multiplexer with periodic sources.* Juli 1990.

[15] G. Buntrock, C. Damm, U. Hertrampf und C. Meinel. *Structure and importance of logspace-MOD-classes.* Juli 1990.

[16] H. Gold und P. Tran-Gia. *Performance analysis of a batch service queue arising out of manufacturing systems modeling.* Juli 1990.

[17] F. Hübner und P. Tran-Gia. *Quasi-stationary analysis of a finite capacity asynchronous multiplexer with modulated deterministic input.* Juli 1990.

[18] U. Huckenbeck. *Complexity and approximation theoretical properties of rational functions which map two intervals into two other ones.* August 1990.

[19] P. Tran-Gia. *Analysis of polling systems with general input process and finite capacity.* August 1990.

[20] C. Friedewald, A. Hieronymus und B. Menzel. *WUMPS Würzburger message passing system.* Oktober 1990.

[21] R. V. Book. *On random oracle separations.* November 1990.

[22] Th. Stock. *Influences of multiple priorities on DQDB protocol performance.* November 1990.

[23] P. Tran-Gia und R. Dittmann. *Performance analysis of the CRMA-protocol in high-speed networks.* Dezember 1990.

[24] C. Wrathall. *Confluence of one-rule Thue systems.*

[25] O. Gihr und P. Tran-Gia. *A layered description of ATM cell traffic streams and correlation analysis.* Januar 1991.

[26] H. Gold und F. Hübner. *Multi server batch service systems in push and pull operating mode — a performance comparison.* Juni 1991.

[27] H. Gold und H. Grob. *Performance analysis of a batch service system operating in pull mode.* Juli 1991.

[28] U. Hertrampf. *Locally definable acceptance types—the three valued case.* Juli 1991.

[29] U. Hertrampf. *Locally definable acceptance types for polynomial time machines.* Juli 1991.

[30] Th. Fritsch und W. Mandel. *Communication network routing using neural nets – numerical aspects and alternative approaches.* Juli 1991.

[31] H. Vollmer und K. W. Wagner. *Classes of counting functions and complexity theoretic operators.* August 1991.

[32] R. V. Book, J. H. Lutz und K. W. Wagner. *On complexity classes and algorithmically random languages.* August 1991.

[33] F. Hübner. *Queueing analysis of resource dispatching and scheduling in multi-media systems.* September 1991.

[34] H. Gold und G. Bleckert. *Analysis of a batch service system with two heterogeneous servers.* September 1991.

[35] H. Vollmer und K. W. Wagner. *Complexity of functions versus complexity of sets.* Oktober 1991.

[36] F. Hübner. *Discrete-time analysis of the output process of an ATM multiplexer with periodic input.* November 1991.

[37] P. Tran-Gia und O. Gropp. *Structure and performance of neural nets in broadband system admission control.* November 1991.

[38] G. Buntrock und K. Loryś. *On growing context-sensitive languages.* Januar 1992.

[39] K. W. Wagner. *Alternating machines using partially defined "AND" and "OR".* Januar 1992.

[40] F. Hübner und P. Tran-Gia. *An analysis of multi-service systems with trunk reservation mechanisms.* April 1992.

[41] U. Huckenbeck. *On a generalization of the bellman-ford-algorithm for acyclic graphs.* Mai 1992.

[42] U. Huckenbeck. *Cost-bounded paths in networks of pipes with valves.* Mai 1992.

[43] F. Hübner. *Autocorrelation and power density spectrum of ATM multiplexer output processes.* September 1992.

[44] F. Hübner und M. Ritter. *Multi-service broadband systems with CBR and VBR input traffic.* Oktober 1992.

[45] M. Mittler und P. Tran-Gia. *Performance of a neural net scheduler used in packet switching interconnection networks.* Oktober 1992.

[46] M. Kowaluk und K. W. Wagner. *Vector language: Simple description of hard instances.* Oktober 1992.

[47] B. Menzel und J. Wolff von Gudenberg. *Kommentierte Syntaxdiagramme für C++.* November 1992.

[48] D. Emme. *A kernel for funtion definable classes and its relations to lowness.* November 1992.

[49] S. Öhring. *On dynamic and modular embeddings into hyper de Bruijn networks.* November 1992.

[50] K. Poeck und M. Tins. *An intelligent tutoring system for classification problem solving.* November 1992.

[51] K. Poeck und F. Puppe. *COKE: Efficient solving of complex assignment problems with the propose-and-exchange method.* November 1992.

[52] Th. Fritsch, M. Mittler und P. Tran-Gia. *Artificial neural net applications in telecommunication systems.* Dezember 1992.

[53] H. Vollmer und K. W. Wagner. *The complexity of finding middle elements.* Januar 1993.

[54] O. Gihr, H. Gold und S. Heilmann. *Analysis of machine breakdown models.* Januar 1993.

[55] S. Öhring. *Optimal dynamic embeddings of arbitrary trees in de Bruijn networks.* Februar 1993.

[56] M. Mittler. *Analysis of two finite queues coupled by a triggering scheduler.* März 1993.

[57] J. Albert, F. Duckstein, M. Lautner und B. Menzel. *Message-passing auf transputer-systemen.* März 1993.

[58] Th. Stock und P. Tran-Gia. *Basic concepts and performance of high-speed protocols.* März 1993.

[59] F. Hübner. *Dimensioning of a peak cell rate monitor algorithm using discrete-time analysis.* März 1993.

[60] G. Buntrock und K. Loryś. *The variable membership problem: Succinctness versus complexity.* April 1993.

[61] H. Gold und B. Frötschl. *Performance analysis of a batch service system working with a combined push/pull control.* April 1993.

[62] H. Vollmer. *On different reducibility notions for function classes.* April 1993.

[63] S. Öhring und S. K. Das. *Folded Petersen Cube Networks: New Competitors for the Hyepercubes.* Mai 1993.

[64] S. Öhring und S. K. Das. *Incomplete Hypercubes: Embeddings of Tree-Related Networks.* Mai 1993.

[65] S. Öhring und S. K. Das. *Mapping Dynamic Data and Algorithm Structures on Product Networks.* Mai 1993.

[66] F. Hübner und P. Tran-Gia. *A Discrete-Time Analysis of Cell Spacing in ATM Systems.* Juni 1993.

[67] R. Dittmann und F. Hübner. *Discrete-Time Analysis of a Cyclic Service System with Gated Limited Service.* Juni 1993.

[68] M. Frisch und K. Jucht. *Pascalli-P.* August 1993.

[69] G. Buntrock. *Growing Context-Sensitive Languages and Automata.* September 1993.

[70] S. Öhring und S. K. Das. *Embeddings of Tree-Related Topologies in Hyper Petersen Networks.* Oktober 1993.

[71] S. Öhring und S. K. Das. *Optimal Communication Primitives on the Folded Petersen Networks.* Oktober 1993.

[72] O. Rose und M. R. Frater. *A Comparison of Models for VBR Video Traffic Sources in B-ISDN.* Oktober 1993.

[73] M. Mittler und N. Gerlich. *Reducing the Variance of Sojourn Times in Queueing Networks with Overtaking.* November 1993.

[74] P. Tran-Gia. *Discrete-Time Analysis Technique and Application to Usage Parameter Control Modelling in ATM Systems.* November 1993.

[75] F. Hübner. *Output Process Analysis of the Peak Cell Rate Monitor Algorithm.* Januar 1994.

[76] K. Cronauer. *A Criterion to Separate Complexity Classes by Oracles.* Januar 1994.

[77] M. Ritter. *Analysis of the Generic Cell Rate Algorithm Monitoring ON/OFF-Traffic.* Januar 1994.

[78] K. Poeck, D. Fensel, D. Landes und J. Angele. *Combining KARL and Configurable Role Limiting Methods for Configuring Elevator Systems.* Januar 1994.

[79] O. Rose. *Approximate Analysis of an ATM Multiplexer with MPEG Video Input.* Januar 1994.

[80] A. Schömig. *Using Kanban in a Semiconductor Fabrication Environment — a Simulation Study.* März 1994.

[81] M. Ritter, S. Kornprobst und F. Hübner. *Performance Analysis of Source Policing Architectures in ATM Systems.* April 1994.

[82] U. Hertrampf, H. Vollmer und K. W. Wagner. *On Balanced vs. Unbalanced Computation Trees.* Mai 1994.

[83] M. Mittler und A. Schömig. *Entwicklung von „Due–Date"–Warteschlangendisziplinen zur Optimierung von Produktionssystemen.* Mai 1994.

[84] U. Hertrampf. *Complexity Classes Defined via k-valued Functions.* Juli 1994.

[85] U. Hertrampf. *Locally Definable Acceptance: Closure Properties, Associativity, Finiteness.* Juli 1994.

[86] O. Rose und M. R. Frater. *Delivery of MPEG Video Services over ATM.* August 1994.

[87] B. Reinhardt. *Kritik von Symptomerkennung in einem Hypertext-Dokument.* August 1994.

[88] U. Rothaug, E. Yanenko und K. Leibnitz. *Artificial Neural Networks Used for Way Optimization in Multi-Head Systems in Application to Electrical Flying Probe Testers.* September 1994.

[89] U. Hertrampf. *Finite Acceptance Type Classes.* Oktober 1994.

[90] U. Hertrampf. *On Simple Closure Properties of #P.* Oktober 1994.

[91] H. Vollmer und K. W. Wagner. *Recursion Theoretic Characterizations of Complexity Classes of Counting Functions.* November 1994.

[92] U. Hinsberger und R. Kolla. *Optimal Technology Mapping for Single Output Cells.* November 1994.

[93] W. Nöth und R. Kolla. *Optimal Synthesis of Fanoutfree Functions.* November 1994.

[94] M. Mittler und R. Müller. *Sojourn Time Distribution of the Asymmetric $M/M/1//N$ – System with LCFS-PR Service.* November 1994.

[95] M. Ritter. *Performance Analysis of the Dual Cell Spacer in ATM Systems.* November 1994.

[96] M. Beaudry. *Recognition of Nonregular Languages by Finite Groupoids.* Dezember 1994.

[97] O. Rose und M. Ritter. *A New Approach for the Dimensioning of Policing Functions for MPEG-Video Sources in ATM-Systems.* Januar 1995.

[98] T. Dabs und J. Schoof. *A Graphical User Interface For Genetic Algorithms.* Februar 1995.

[99] M. R. Frater und O. Rose. *Cell Loss Analysis of Broadband Switching Systems Carrying VBR Video.* Februar 1995.

[100] U. Hertrampf, H. Vollmer und K. W. Wagner. *On the Power of Number-Theoretic Operations with Respect to Counting.* Januar 1995.

[101] O. Rose. *Statistical Properties of MPEG Video Traffic and their Impact on Traffic Modeling in ATM Systems.* Februar 1995.

[102] M. Mittler und R. Müller. *Moment Approximation in Product Form Queueing Networks.* Februar 1995.

[103] D. Rooß und K. W. Wagner. *On the Power of Bio-Computers.* Februar 1995.

[104] N. Gerlich und M. Tangemann. *Towards a Channel Allocation Scheme for SDMA-based Mobile Communication Systems.* Februar 1995.

[105] A. Schömig und M. Kahnt. *Vergleich zweier Analysemethoden zur Leistungsbewertung von Kanban Systemen.* Februar 1995.

[106] M. Mittler, M. Purm und O. Gihr. *Set Management: Synchronization of Prefabricated Parts before Assembly.* März 1995.

[107] A. Schömig und M. Mittler. *Autocorrelation of Cycle Times in Semiconductor Manufacturing Systems.* März 1995.

[108] A. Schömig und M. Kahnt. *Performance Modelling of Pull Manufacturing Systems with Batch Servers and Assembly-like Structure.* März 1995.

[109] M. Mittler, N. Gerlich und A. Schömig. *Reducing the Variance of Cycle Times in Semiconductor Manufacturing Systems.* April 1995.

[110] A. Schömig und M. Kahnt. *A note on the Application of Marie's Method for Queueing Networks with Batch Servers.* April 1995.

[111] F. Puppe, M. Daniel und G. Seidel. *Qualifizierende Arbeitsgestaltung mit tutoriellen Expertensystemen für technische Diagnoseaufgaben.* April 1995.

[112] G. Buntrock, und G. Niemann. *Weak Growing Context-Sensitive Grammars.* Mai 1995.

[113] J. García and M. Ritter. *Determination of Traffic Parameters for VPs Carrying Delay-Sensitive Traffic.* Mai 1995.

[114] M. Ritter. *Steady-State Analysis of the Rate-Based Congestion Control Mechanism for ABR Services in ATM Networks.* Mai 1995.

[115] H. Graefe. *Konzepte für ein zuverlässiges Message-Passing-System auf der Basis von UDP.* Mai 1995.

[116] A. Schömig und H. Rau. *A Petri Net Approach for the Performance Analysis of Business Processes.* Mai 1995.

[117] K. Verbarg. *Approximate Center Points in Dense Point Sets.* Mai 1995.

[118] K. Tutschku. *Recurrent Multilayer Perceptrons for Identification and Control: The Road to Applications.* Juni 1995.

[119] U. Rhein-Desel. *Eine „Übersicht“ über medizinische Informationssysteme: Krankenhausinformationssysteme, Patientenaktensysteme und Kritiksysteme.* Juli 1995.

[120] O. Rose. *Simple and Efficient Models for Variable Bit Rate MPEG Video Traffic.* Juli 1995.

[121] A. Schömig. *On Transfer Blocking and Minimal Blocking in Serial Manufacturing Systems — The Impact of Buffer Allocation.* Juli 1995.