Chicago Journal of Theoretical Computer Science

The MIT Press

Volume 1996, Article 4

13 November 1996

ISSN 1073–0486. MIT Press Journals, 55 Hayward St., Cambridge, MA 02142 USA; (617)253-2889; *journals-orders@mit.edu, journals-info@mit.edu*. Published one article at a time in LaTeX source form on the Internet. Pagination varies from copy to copy. For more information and other articles see:

- *http://www-mitpress.mit.edu/jrnls-catalog/chicago.html*
- *http://www.cs.uchicago.edu/publications/cjtcs/*
- *gopher.mit.edu*
- *gopher.cs.uchicago.edu*
- anonymous *ftp* at *mitpress.mit.edu*
- anonymous *ftp* at *cs.uchicago.edu*

The *Chicago Journal of Theoretical Computer Science* is abstracted or indexed in *Research Alert,*® *SciSearch,*® *Current Contents*® */Engineering Computing & Technology*, and *CompuMath Citation Index.*®

©1996 The Massachusetts Institute of Technology. Subscribers are licensed to use journal articles in a variety of ways, limited only as required to insure fair attribution to authors and the journal, and to prohibit use in a competing commercial product. See the journal's World Wide Web site for further details. Address inquiries to the Subsidiary Rights Manager, MIT Press Journals; (617)253-2864; *journals-rights@mit.edu*.

The *Chicago Journal of Theoretical Computer Science* is a peer-reviewed scholarly journal in theoretical computer science. The journal is committed to providing a forum for significant results on theoretical aspects of all topics in computer science.

Editor in chief: Janos Simon

Consulting editors: Joseph Halpern, Stuart A. Kurtz, Raimund Seidel

Editors:	Martin Abadi	Greg Frederickson	John Mitchell
	Pankaj Agarwal	Andrew Goldberg	Ketan Mulmuley
	Eric Allender	Georg Gottlob	Gil Neiger
	Tetsuo Asano	Vassos Hadzilacos	David Peleg
	Laszló Babai	Juris Hartmanis	Andrew Pitts
	Eric Bach	Maurice Herlihy	James Royer
	Stephen Brookes	Stephen Homer	Alan Selman
	Jin-Yi Cai	Neil Immerman	Nir Shavit
	Anne Condon	†Paris Kanellakis	Eva Tardos
	Cynthia Dwork	Howard Karloff	Sam Toueg
	David Eppstein	Philip Klein	Moshe Vardi
	Ronald Fagin	Phokion Kolaitis	Jennifer Welch
	Lance Fortnow	Stephen Mahaney	Pierre Wolper
	Steven Fortune	Michael Merritt	

Managing editor: Michael J. O'Donnell

Electronic mail: *chicago-journal@cs.uchicago.edu*

Weakly Growing Context-Sensitive Grammars

Gerhard Buntrock   Gundula Niemann

13 November, 1996

Abstract

Abstract-1 This paper introduces *weakly growing context-sensitive grammars*. Such grammars generalize the class of growing context-sensitive grammars (studied by several authors), in that these grammars have rules that "grow" according to a position valuation.

Abstract-2 If a position valuation coincides with the initial part of an exponential function, it is called a *steady* position valuation. All others are called *unsteady*. The complexity of the language generated by a grammar depends crucially on whether the position valuation is steady or not. More precisely, for every unsteady position valuation, the class of languages generated by *WGCSG*s with this valuation coincides with the class *CSL* of context-sensitive languages. On the other hand, for every steady position valuation, the class of languages generated corresponds to a level of the hierarchy of exponential time-bounded languages in *CSL*. We show that the following three conditions are equivalent:

- The hierarchy of exponential time-bounded languages in *CSL* collapses.
- There exists a class defined by an unsteady position valuation such that there is also a normal form of order 2 (e.g., Cremers or Kuroda normal form) for that class.
- There exists a class defined by a steady position valuation that is closed under inverse homomorphisms.

Some of these results were presented at LATIN'95 at Valparaíso, Chile.

1 Introduction

1-1 A beautiful theory always has simple concepts and remarkable results. Here we investigate a widely unknown but nice part of the class of context-sensitive languages (*CSL*).

1-2 Our simple concept consists of a valuation of strings, and concentration on the context-sensitive grammars that are growing under such a valuation. As a result, we get characterizations of known double-bounded complexity classes by this quite different concept. Namely, we view the exponential time-bounded languages in *CSL*, where *CSL* is the class of languages that can be recognized by a Turing machine in linear space. Little is known about such dual-bounded classes; see, e.g., [Coo79], [Pip79], [Bör89], and [Rei90]. Wolfgang Paul asks in general which kind of speedup theorem holds for space-bounded computations (see [Pau78]). We concentrate on this question for the case of linear-bounded automata. It is shown that this problem can be reformulated using our concept, which comes from formal language theory: We ask whether a particular class of languages is characterized by a class of normal form grammars of order 2 (i.e., left and right sides of rules have length at most 2), or, equivalently, whether another class is closed under inverse homomorphism.

1-3 Noam Chomsky has already observed in his famous work that *CSL* is characterized by monotone grammars, that is, 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 as a definition.

1-4 By replacing "at least as long as" in the definition of the class *CSG* of context-sensitive grammars by "longer than," we obtain growing context-sensitive grammars (*GCSG*) which define the class of growing context-sensitive languages (*GCSL*). Elias Dahlhaus and Manfred Warmuth investigated the complexity of *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 the class P of languages recognizable deterministically in polynomial time. 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 grammar (quasi-*GCSG*, or *QGCSG* for short) characterizes *GCSL* ([BL92], [BL94], [Bun96]). Unfortunately, the class *GCSL* is very weak. Not even the language $COPY \stackrel{\text{def}}{=} \{\, ww : w \in \{a,b\}^* \,\}$ is contained in *GCSL* ([Bun93], see also [BO95]).

1-5 Weakly growing context-sensitive grammars (*WGCSG*) are a generalization of *QGCSG*. In a *WGCSG* we valuate the positions as well as the symbols 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 weakly 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-hand side is lower than that of the right-hand side. (This is defined precisely in Definition 2.) The weakly growing context-sensitive languages related to constant position valuations characterize *GCSL*. We will show that this definition is robust under some natural changes.

1-6 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 every nonconstant 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 each position valuation s. The question as to whether or not the corresponding language classes are strictly contained in *CSL* will be viewed later.

1-7 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 (the notion of a basis was originally introduced by Raymond Smullyan in [Smu61]). With standard arguments the following can be proved: For every position valuation s, the class $WGCSL_s$ is closed under ε-free homomorphism, union, ε-free regular substitution, concatenation, and intersection with regular sets. Additionally it can be shown that $WGCSL_s$ is closed under transposition for every position valuation s.

1-8 What is missing, e.g., for $WGCSL_s$ to be an abstract family of languages (or AFL, see any good introduction to formal language theory, such as [Sal73] or [Har78]), is the closure under inverse homomorphism and also under k-bounded homomorphism. Here from [GGH69] it is known that if the one holds, then so does the other and vice versa, because of the closure properties we already know. The closure under inverse homomorphism here can neither be proved nor disproved in general; a main result in this context is the equivalence of this problem to the problem of 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 each nonconstant position valuation, the closure under inverse homomorphism of $WGCSL_s$ equals CSL.

1-9 To prove these results we distinguish *steady* and *unsteady* position valuations: A position valuation is steady if it coincides with the initial part 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.

1-10 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 weakly 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 (see [Gla64]) to $WGCSL$s. On the other hand, for every language that can be recognized by a linear space-bounded automaton there exists a steady position valuation s such that this language belongs to $WGCSL_s$. To get a very tight relation here, we introduce a new type of counters: grammars that count.

1-11 Furthermore, the weakly growing context-sensitive language classes corresponding to different growth factors build up a linearly inclusion-ordered hierarchy that characterizes the exponential time hierarchy for CSL in terms of necessary and sufficient conditions for the hierarchy to collapse.

1-12 For unsteady position valuations, there does not exist an infinite extension such that the value of a sentential form increases if and only if a weakly growing context-sensitive rule is applied. To the contrary: We present a collection of rules which when applied to a given sentential form lead to a decrement of the value of that sentential form. By this we will show that arbitrarily long derivations are possible. We interpret this to be the reason that for every unsteady position valuation the corresponding class of weakly growing context-sensitive grammars characterizes CSL.

1-13 Another important difference between the steady and unsteady cases consists in the existence of normal forms of order 2. Note that every s-weakly growing context-sensitive grammar in a normal form of order 2 is one related to a steady position valuation. Here we know that for every steady position valuation s and every weakly growing context-sensitive grammar related to it there exists an equivalent *WGCSG* of the same type in normal form. This can be proved using an obvious transformation. For this we will use Cremers normal form, which has the advantage of having a minimal number of different types of rules. Additionally, it turns out that the growth factor of a steady position valuation characterizes a weakly 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 obvious transformation to reduce the order of a rule, and similar transformations, do not work. Thus we obtain the equivalence of the following problems.

- Does the exponential time hierarchy 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 weakly growing context-sensitive languages coincides with *CSL*?
- Does there exist a steady position valuation such that the corresponding class of weakly growing context-sensitive languages is closed under inverse homomorphism?
- Does there exist an unsteady position valuation such that the following holds: Every weakly growing context-sensitive grammar related to this position valuation can be transformed into an equivalent grammar in a normal form of order 2 that is weakly growing context-sensitive related to some position valuation?

2 Preliminaries

2-1 We will denote the set of positive and negative integers by $\mathbb{Z}$, the set of natural numbers by $\mathbb{N}$, the set $\mathbb{N} \setminus \{0\}$ by $\mathbb{N}^+$, the set of rational numbers by $\mathbb{Q}$, and the set of positive rational numbers by $\mathbb{Q}^+$. Denote modified subtraction by $\dot{-}$.

2-2 We will denote a linear-bounded automaton by $M = (\Sigma, Q, \Gamma, q_0, \delta, F)$, where $\Sigma \subseteq \Gamma$ is the input alphabet, Q is the set of states, Γ is the working alphabet, q_0 is the start state, $\delta\colon (Q \times \Gamma) \to (Q \times \Gamma \times \{L, R\})$ is the transition function, and F is the set of accepting states. By uqv we denote a configuration of M, where uv is the content of the working tape, q is the actual state, and the first symbol of v is being read. By $K_1 \overset{*}{\underset{M}{\vdash}} K_2$ we denote that configuration K_2 is reachable from configuration K_1 by a computation of M (see, e.g., [HU79] for a detailed definition). The subscript M is sometimes omitted when it is obvious from the context. By $M = (\Sigma, Q, q_0, \delta, F)$ we denote a finite automaton, where the signature of δ is appropriately changed. Additionally, we use $\delta^*\colon Q \times \Sigma^* \to Q$ to denote the extension of δ to $Q \times \Sigma^*$.

2-3 A *context-sensitive grammar* (*CSG*) is a quadruple $G = \langle N, T, S, P\rangle$, where N and T are finite disjoint alphabets of nonterminal and terminal symbols, respectively, $S \in N$ is the 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-hand side of any rule. The corresponding language class is denoted by *CSL*.

2-4 If $u, v \in (N \cup T)^*$ are sentential forms, we say *u derives v in* $G = \langle N, T, S, P\rangle$, 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 $\overset{*}{\Longrightarrow}$ denote the reflexive and transitive closure of $\Longrightarrow$. Thus the *language* $L(G)$ generated by a grammar G is described by $L(G) \overset{\text{def}}{=} \left\{ w \in T^* : S \overset{*}{\underset{G}{\Longrightarrow}} w \right\}$. For a language $L \subseteq \Sigma^*$, the transposition L^T of L is defined by $L^T \overset{\text{def}}{=} \left\{ w : w^T \in L \right\}$, where w^T denotes the transposition of the string $w \in \Sigma^*$ inductively defined as follows: Define $\varepsilon^T \overset{\text{def}}{=} \varepsilon$, and for each $v \in \Sigma^*$ and each $a \in \Sigma$ define $(av)^T$ to be $v^T a$. Similarly, $L_1 \cdot L_2 \overset{\text{def}}{=} \{\, w \cdot v : w \in L_1,\ v \in L_2 \,\}$, where $w \cdot v = wv$ is w concatenated with v.

2-5 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*. Grammars with $\alpha \in N$ and $\beta \in N \cdot T \cup T$ for each rule $(\alpha \to \beta) \in P \setminus \{(S \to \varepsilon)\}$ are called *regular*. The corresponding language class is denoted by *REG*.

2-6 By replacing the inequality in the definition of context-sensitive grammars by a strict inequality, 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]).

2-7 Instead of just counting the symbols appearing in a rule (as we did above), we can assign each symbol 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 $(\alpha_1 \dots \alpha_n \to \beta_1 \dots \beta_m) \in P$:

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

The corresponding set of grammars is denoted by $QGCSG$. The function f is called a *symbol valuation for* G. By $f|_N$ we denote the restriction of f to the set N.

2-8 If in the definition of $QGCSG$ we omit the requirement of context-sensitivity, but keep the requirement that if $(S \to \varepsilon) \in P$, S does not appear on the right-hand side of any rule, we obtain the *quasi-growing grammars*.

2-9 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 positions inside a rule. A position valuation is a function that is defined for an initial segment of the natural numbers. It must valuate at least one position. In the following we will refer to $\max\{|\alpha|, |\beta|\}$ as *the order of the rule* $\alpha \to \beta$, and to $\mathrm{ord}(G) \stackrel{\text{def}}{=} \max\{\max\{|\alpha|, |\beta|\} : (\alpha \to \beta) \in P\}$ as *the order of* G. Note that in the case of a context-sensitive grammar the order of G is equal to $\max\{|\beta| : (\alpha \to \beta) \in P\}$. By "a normal form of order 2," we mean a normal form such that every grammar in this normal form has order at most 2.

Definition 1 *A* position valuation *is a function* $s\colon \mathbb{N}^+ \to \mathbb{N}^+$ *with:*

> *If s is defined for a $j \in \mathbb{N}^+$, then 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.*

2-10 It is also possible to allow zero points for position valuations, but with the techniques used to show our main results we get for each position valuation with zero points a characterization of CSL (see Theorem 8). We restrict ourselves to the most interesting position valuations which have at least two valuated positions.

Definition 2 *Let s be a position valuation. A grammar $G = \langle N, T, S, P\rangle$ is called* weakly growing context-sensitive related to the position valuation *s or also s-*weakly growing context-sensitive*, if the following hold:*

(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 $(\alpha_1 \ldots \alpha_n \to \beta_1 \ldots \beta_m) \in P \setminus \{(S \to \varepsilon)\}$ it holds that

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

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

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

2-11 As we now valuate not only the symbols but also their positions inside a rule, it is possible to interchange two symbols by a rule. This can be used to show that the language $COPY \stackrel{\text{def}}{=} \{\, ww : w \in \{a,b\}^* \,\}$ is weakly growing context-sensitive related to every nonconstant position valuation. To give an intuition of weakly growing context-sensitive grammars, we look at the following example closely.

Example 1-1 **Example 1** Consider the language $COPY \stackrel{\text{def}}{=} \{\, ww : w \in \{a,b\}^* \,\}$. Let s be any position valuation with at least two valuated positions and $s(1) < s(2)$.

Example 1-2 Define a grammar $G = \langle N, \{a,b\}, S, P \rangle$ as follows:

$$N \stackrel{\text{def}}{=} \left\{ S, \tilde{S}, A, A', B, B', (X,a), (X,b) \right\}$$

and P contains the following rules:

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

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

$$
\begin{aligned}
f(\tilde{S}) &\stackrel{\text{def}}{=} 1\\
f(S) &\stackrel{\text{def}}{=} f(A) \stackrel{\text{def}}{=} f(B) \stackrel{\text{def}}{=} f((X,a)) \stackrel{\text{def}}{=} f((X,b)) \stackrel{\text{def}}{=} 2\\
f(A') &\stackrel{\text{def}}{=} f(B') \stackrel{\text{def}}{=} 3\\
f(a) &\stackrel{\text{def}}{=} f(b) \stackrel{\text{def}}{=} 3
\end{aligned}
$$

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

Example 1 □

2-12 As $COPY \notin GCSL$ [Bun93], we have $WGCSL_s \not\subseteq GCSL$ for every 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.)

2-13 Note that if we use classical context-sensitive grammars in the definition of weakly growing context-sensitive grammars, we get a subclass of *GCSG* [Nie94]. Here the interesting question arises as to whether this class 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 every position valuation s.

2-14 We now look briefly at some robustness properties of the definition of weakly growing context-sensitive languages. As can easily be seen, the standard algorithm to delete chain rules (see, e.g., [Har78]) can be adapted for weakly growing context-sensitive grammars. The same applies to the standard algorithm to delete terminal symbols from the left-hand side of a rule. Thus we obtain:

Proposition 1-1 **Proposition 1** *For every WGCSG related to a position valuation s, there exists an equivalent $WGCSG_s$ without chain rules, even if there is only a symbol valuation known such that some chain rules are not strictly growing, i.e., cycling with chain rules can be deleted in such cases.*

Proposition 1-2 *Furthermore, in a $WGCSG_s$, terminals can be forbidden on the left-hand side of rules without affecting the grammar's inherent power.*

2-15 In an inequality, the multiplication of both sides with the same positive value does not affect its validity, and the following holds.

Proposition 2 *For all $k \in \mathbb{Q}^+$, the classes $WGCSL_{k \cdot s}$ are the same.*

2-16 Thus, for each constant position valuation C, it holds that $WGCSG_C = QGCSG$ and consequently, $WGCSL_C = GCSL$.

2-17 The values on both sides of an inequality can not only be stretched, but can also be shifted without affecting its validity, which leads to the following result.

Lemma 1 *In Definition 2 (iii) it is sufficient to demand that there exists a function $f\colon N \cup T \to \mathbb{Q}$, i.e., with values in $\mathbb{Q}$, that satisfies the inequality (*).*

Proof of Lemma 1-1 **Proof of Lemma 1** 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 $(\alpha_1 \dots \alpha_n \to \beta_1 \dots \beta_m) \in P$ not equal to $(S \to \varepsilon)$, it holds that

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

Let $\{a_1, \dots, a_k\}$ be the set of symbols $N \cup T$ 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 $q \stackrel{\text{def}}{=} \mathrm{lcm}(q_1, \dots, q_k)$ be the common denominator of all values and $z \stackrel{\text{def}}{=} q \cdot |\min\{0, p_1, \dots, p_k\}|$ be the product of this common denominator with the absolute value of the smallest negative enumerator. Then by a linear transformation we obtain natural numbers: $f'(a_i) \stackrel{\text{def}}{=} q \cdot f(a_i) + z$. The inequalities still apply, as the grammar has no length-decreasing rules.

Proof of Lemma 1 □

2-18 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.

2-19 As we mentioned above, in the case of quasi-growing context-sensitive grammars, if we omit the requirement of context-sensitivity, we obtain grammars that characterize the same language class [BL92]. In the case of weakly growing context-sensitive grammars, the effect is different: If in Definition 2 we leave out the requirement of context-sensitivity, we obtain the weakly growing grammars, which already characterize the languages generated by nonrestricted grammars [Nie96], i.e., the class of recursively enumerable sets.

2-20 To lead to our main results we now introduce the two cases of steady and unsteady position valuations.

Definition 3-1 **Definition 3** *A position valuation* $s\colon \mathbb{N}^+ \to \mathbb{N}^+$ *is called* steady *if it coincides with an exponential function on its domain, that is, if it can be written as*

$$s(i) = s(1) \cdot \mathrm{w}(s)^{i-1}$$

where $\mathrm{w}(s) \stackrel{def}{=} \frac{s(2)}{s(1)}$*, so* $\mathrm{w}(s) \in \mathbb{Q}^+$*. This fraction,* $\mathrm{w}(s)$*, is called the* growth factor of s*.*

Definition 3-2 *In the other case,* s *is* unsteady*. Note that this is the case if and only if* s *has at least three 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*.*

2-21 Note that $\mathrm{w}(s) < 1$ means s is monotone decreasing, and $\mathrm{w}(s) = 1$ means s is constant.

Definition 4 *Let* s *be a position valuation. Let* $G = \langle N, T, S, P\rangle \in WGCSG_s$*, and let* f *be an appropriate symbol valuation for* G*. The* growth rate of a rule $(\alpha_1 \ldots \alpha_n \to \beta_1 \ldots \beta_m) \in P$ *with respect to* f *is the difference* $\sum_{i=1}^{m} s(i) \cdot f(\beta_i) - \sum_{i=1}^{n} s(i) \cdot f(\alpha_i)$ *between the values on the right-hand and the left-hand sides of the rule. The* growth rate of a derivation step $u_1 \ldots u_n \underset{G}{\Longrightarrow} v_1 \ldots v_m$ *with respect to* f *is the difference* $\sum_{i=1}^{m} s(i) \cdot f(v_i) - \sum_{i=1}^{n} s(i) \cdot f(u_i)$ *between the values of* $v_1 \ldots v_m$ *and* $u_1 \ldots u_n$*.*

2-22 The relationship between the growth rate of a rule, the growth rate of a derivation step, and the growth factor (in case of a steady position valuation) is investigated in Lemma 3 and Lemma 5. Nevertheless, we first look at general properties of weakly growing context-sensitive languages.

3 Closure Properties and Normal Form

3-1 The most important closure properties are the AFL properties. It turns out that steadiness of a position valuation has no effect on these closure properties except for one (see Section 3.1). To the contrary, unsteadiness does affect the transformability of a weakly growing context-sensitive grammar into an equivalent grammar of the same type in normal form of order 2 (see Section 3.2). In Section 6 (Theorem 10), we will prove that the existence of one missing closure property (inverse homomorphism) for the weakly growing context-sensitive languages with steady position valuation is equivalent to the existence of a transformation into a normal form of order 2 for weakly growing context-sensitive grammars with unsteady position valuation.

3.1 Closure Properties

3.1-1 To get 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 every position valuation s, i.e., $WGCSL_s$ is a basis for the class of r.e. sets (originally introduced by Raymond Smullyan in [Smu61]; see also [Bun96]). With standard arguments, we show the following.

Theorem 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.*

Proof of Theorem 1

Proof of Theorem 1-1 **For ε-free homomorphism:** Let $G = \langle N, T, S, P\rangle \in WGCSL_s$, and let f be a symbol valuation for G. Let $h\colon T \to \Delta$ be an ε-free homomorphism.

Proof of Theorem 1-2 A grammar $G' = \langle N', \Delta, S, P'\rangle$ for $h(L(G))$ can be constructed by defining $N' \stackrel{\text{def}}{=} N \cup T$ and adding to P the rules $(a \to h(a))$ for every $a \in T$ to obtain P'. An appropriate symbol valuation can be found by extending f to Δ with $f(u) = \max\{\, f(a) : a \in T\,\} + 1$ for $u \in \Delta$.

Proof of Theorem 1-3 **For union:** Let $G_1 = \langle N_1, T_1, S_1, P_1\rangle$ and $G_2 = \langle N_2, T_2, S_2, P_2\rangle$ be in $WGCSL_s$. Let f_1 and f_2 be appropriate symbol valuations for G_1 and G_2, respectively. Wlog we can assume $(N_1 \cup T_1) \cap N_2 = \emptyset$ and $N_1 \cap (N_2 \cup T_2) = \emptyset$.

A grammar $G = \langle N, T_1 \cup T_2, S, P \rangle$ for $L(G_1) \cup L(G_2)$ can be constructed by $N = N_1 \cup N_2 \cup \{S\}$ and $P = P_1 \cup P_2 \cup \{(S \to S_1), (S \to S_2)\}$. (Handling the case $\varepsilon \in L(G_1) \cup L(G_2)$ is standard.) An appropriate symbol valuation f is defined by $f|_{N_1 \cup T_1} \stackrel{\text{def}}{=} 2 \cdot f_1$, $f|_{N_2 \cup T_2} \stackrel{\text{def}}{=} 2 \cdot f_2$, $f(S) \stackrel{\text{def}}{=} 1$.

Proof of Theorem 1-4

For ε-free regular substitution: In fact, we can show that $WGCSL_s$ is closed under ε-free $WGCSL_s$ substitution. Then, closure under ε-free regular substitution follows, because every regular grammar is also weakly growing context-sensitive. Just use the symbol valuation $f\colon N \cup T \to \mathbb{N}^+$ defined by $f(A) \stackrel{\text{def}}{=} 1$ for $A \in N$, $f(a) \stackrel{\text{def}}{=} 2$ for $a \in T$.

Proof of Theorem 1-5

To show closure under ε-free $WGCSL_s$ substitution, we consider $G = \langle N, T, S, P \rangle \in WGCSG_s$, and for every $a \in T$ let $G_a = \langle N_a, T_a, S_a, P_a \rangle \in WGCSG_s$, where we can assume wlog that for every a and b in T with $a \neq b$, it holds that $(N_a \cup T_a) \cap N_b = \emptyset$, and that terminals only appear on the right-hand side of rules (Proposition 1). Let f and f_a for $a \in T$ be appropriate symbol valuations, respectively. Let $L_a \stackrel{\text{def}}{=} L(G_a)$.

Proof of Theorem 1-6

Now we construct a grammar $G' = \langle N', T', S, P' \rangle$ for the language

$$L = \{\, w : \text{ there exists } v = a_1 \ldots a_n \in L \text{ with } w \in L_{a_1} \cdot \ldots \cdot L_{a_n} \,\}$$

Let

$$\begin{aligned} N' &\stackrel{\text{def}}{=} N \cup \bigcup_{a \in T} N_a \\ T' &\stackrel{\text{def}}{=} \bigcup_{a \in T} T_a \\ P' &\stackrel{\text{def}}{=} \ (P \setminus \{\, (A \to a) : A \in N, a \in T \,\}) \cup \\ &\qquad \{\, A \to S_a : (A \to a) \in P \,\} \cup \bigcup_{a \in T} P_a \end{aligned}$$

What happens is that every original terminal is taken as a start symbol for the grammar of the substituting language. The function

$$f' \colon N \cup \bigcup_{a \in T} (N_a \cup T_a) \to \mathbb{N}^+$$

defined by

$$\begin{aligned} f'|_N &\stackrel{\text{def}}{=} f \\ f'|_{N_a \cup T_a} &\stackrel{\text{def}}{=} \max\{\, f(b) : b \in T \,\} \cdot f_a \text{ for } a \in T \end{aligned}$$

is an appropriate symbol valuation for G'.

Proof of Theorem 1-7

For concatenation: Let $G_1 = \langle N_1, T_1, S_1, P_1 \rangle$ and $G_2 = \langle N_2, T_2, S_2, P_2 \rangle$ be in $WGCSL_s$. Let f_1 and f_2 be appropriate symbol valuations for G_1 and G_2, respectively.

Proof of Theorem 1-8

Wlog we can assume $(N_1 \cup T_1) \cap N_2 = \emptyset$ and $N_1 \cap (N_2 \cup T_2) = \emptyset$, and that terminals only appear on the right-hand side of rules (Proposition 1). A grammar $G = \langle N, T_1 \cup T_2, S, P \rangle$ for $L(G_1) \cdot L(G_2)$ can be constructed by defining $N \stackrel{\text{def}}{=} N_1 \cup N_2 \cup \{S\}$ and $P \stackrel{\text{def}}{=} P_1 \cup P_2 \cup \{(S \to S_1 S_2)\}$. (Handling the cases $\varepsilon \in L(G_1)$ and $\varepsilon \in L(G_2)$is standard.) An appropriate symbol valuation f is found by joining f_1 and f_2 and defining $f(S) \stackrel{\text{def}}{=} 1$.

Proof of Theorem 1-9

For intersection with regular sets: Let $G = \langle N, T, S, P \rangle \in WGCSG_s$, let L be a regular set, and let $M = (Q, \Sigma, q_0, \delta, F)$ be a deterministic finite automaton that accepts L.

Proof of Theorem 1-10

By Proposition 1, we can assume that there appear only nonterminals in the left-hand sides of the rules in G. A grammar $G' = \langle N', T \cap \Sigma, S', P' \rangle$ for $L \cap L(G)$ is constructed as follows: $N' \stackrel{\text{def}}{=} \{\, (p, A, q) : p, q \in Q, A \in N \,\} \cup \{S'\}$ (so the nonterminals are triples), and P contains the following rules (for every $A_1, \ldots, A_n,\ B_1, \ldots, B_m \in N,\ u_0, \ldots, u_{m+1}, w \in (\Sigma \cap T)^*$, and $q_1, \ldots, q_{n+1}, p_1, \ldots, p_m, p'_1, \ldots, p'_m \in Q$):

$$
\begin{aligned}
&S' \to w \quad \text{if } (S \to w) \in P \text{ and } w \in L \\
&S' \to u_0(p_1, B_1, p'_1)u_1(p_2, B_2, p'_2) \ldots u_{m-1}(p_m, B_m, p'_m)u_m \\
&\qquad\qquad \text{if } (S' \to u_0 B_1 u_1 B_2 \ldots u_{m-1} B_m u_m) \in P \text{ and} \\
&\qquad\qquad\quad \delta^*(q_0, u_0) = p_1 \text{ and} \\
&\qquad\qquad\quad \delta^*(p'_i, u_i) = p_{i+1} \text{ for } i = 1, \ldots, m-1 \text{ and} \\
&\qquad\qquad\quad \delta^*(p'_m, u_m) \in F \\
&(q_1, A_1, q_2)(q_2, A_2, q_3) \ldots (q_n, A_n, q_{n+1}) \to \\
&\qquad u_1(p_1, B_1, p'_1)u_2(p_2, B_2, p'_2) \ldots u_m(p_m, B_m, p'_m)u_{m+1} \\
&\qquad\qquad \text{if } (A_1 A_2 \ldots A_n \to u_1 B_1 u_2 B_2 \ldots u_m B_{m-1} u_{m+1}) \in P \text{ and} \\
&\qquad\qquad\quad \delta^*(q_1, u_1) = p_1 \text{ and} \\
&\qquad\qquad\quad \delta^*(p'_i,\ u_{i+1}) = p_{i+1} \text{ for } i = 1, \ldots, m-1 \text{ and} \\
&\qquad\qquad\quad \delta^*(p'_m, u_{m+1}) = q_{n+1} \\
&(q_1, A_1, q_2)(q_2, A_2, q_3) \ldots (q_n, A_n, q_{n+1}) \to w \\
&\qquad\qquad \text{if } (A_1 A_2 \ldots A_n \to w) \in P \text{ and } \delta^*(q_1, w) = q_{n+1}
\end{aligned}
$$

(This construction is due to [GGH69]. Matthias Jantzen uses this to show the same result for *GCSL* [Jan79].)

Proof of Theorem 1-11 The function $f\colon N \to \mathbb{N}^+$ defined by

$$\begin{aligned} f'((p,A,q)) &\stackrel{\text{def}}{=} f(A) && \text{for } p,q \in Q,\ A \in N \\ f'(S') &\stackrel{\text{def}}{=} f(S) \\ f'(a) &\stackrel{\text{def}}{=} f(a) && \text{for } a \in T \end{aligned}$$

is an appropriate symbol valuation for G'.

Proof of Theorem 1 □

3.1-2 We have proved the AFL properties (except the closure under inverse homomorphism). Now we show the closure under transposition. We will use this result in the proof of Theorem 4, where we show an important property concerning steady position valuations.

Theorem 2 *Let s be a nonconstant position valuation. Then $WGCSL_s$ is closed under transposition.*

Proof of Theorem 2-1 **Proof of Theorem 2** We sketch the idea for the case $s(1) < s(2)$.

Proof of Theorem 2-2 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:

$$\tilde{N} \stackrel{\text{def}}{=} N \cup \{\tilde{S}\} \cup \{\, a' : a \in T \,\} \cup \left\{\, \check{X} : X \in N \,\right\} \cup \{\, \check{a} : a \in T \,\}$$

where the rules in $\tilde{P}$ accomplish the following:

> First, for a word $w = w_1 \ldots w_n \in L$, the sentential form $\check{w}_1 w_2' \ldots w_n'$ is generated (for this, appropriate copies of the rules in P are used). Then this sentential form is transformed with rules of the form
>
> $$\begin{aligned} \check{a}b' &\to \check{b}a \\ ab' &\to b'a \\ \check{a} &\to a \end{aligned}$$
>
> into the word $w_n \ldots w_1$.

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

Proof of Theorem 2-3 Define $\tilde{f}\colon \tilde{N} \to \mathbb{N}^+$ by

$$\begin{array}{llll} \tilde{f}(X) \stackrel{\text{def}}{=} f(X) & \text{for } X \in N & \tilde{f}(\check{X}) \stackrel{\text{def}}{=} f(X) & \text{for } X \in N \\ \tilde{f}(a') \stackrel{\text{def}}{=} f(a) & \text{for } a \in T & \tilde{f}(\check{a}) \stackrel{\text{def}}{=} f(a) & \text{for } a \in T \\ \tilde{f}(\tilde{S}) \stackrel{\text{def}}{=} f(S) & & \tilde{f}(a) \stackrel{\text{def}}{=} \max\{\, f(a) : a \in T \,\} + 1 & \end{array}$$

Proof of Theorem 2-4 For a symbol valuation s with $s(1) > s(2)$, the proof works analogously. For the case of a nonconstant position valuation with $s(1) = s(2)$, this idea can be adapted by shifting the "mirror point" $\check{X}$ and treating the borders appropriately.

Proof of Theorem 2 □

3.1-3 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 1, and from [GGH69] we know that both closure properties are equivalent under the assumption that the closure properties named in Theorem 1 apply).

3.1-4 For steady position valuations, this problem is equivalent to the problem of whether all linear space-bounded automata can be simulated by linear space-bounded automata with an additional universal exponential time bound (see Theorem 10). The reason for this equivalence lies in the fact that for every position valuation, the closure under inverse homomorphisms of $WGCSL_s$ equals CSL (see Theorems 5 and 9).

3.2 Normal Form

3.2-1 In Section 2 we introduced the two cases of steady and unsteady position valuations. Now we work out what we can say about the transformability of corresponding weakly growing context-sensitive grammars into a normal form of order 2.

3.2-2 For every steady position valuation s, each grammar in $WGCSG_s$ can be transformed into a normal form of order 2. We will use Cremers normal form [Cre73]. Compared with all other normal forms of order 2, the advantage of Cremers normal form lies in its 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 2.

Definition 5 ([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* $AB \to CD$, $A \to CD$, *or* $A \to a$, *where* $A, B, C, D \in N$, $a \in T$.

3.2-3 At first, we adapt the obvious algorithm to reduce the order of a grammar.

Lemma 2 *Let* s *be a steady position valuation. Then for every grammar* $G \in WGCSG_s$ *of order* $l \geq 3$, *there exists an equivalent grammar* $G' \in WGCSG_s$ *of order* $l - 1$.

Proof of Lemma 2-1 **Proof of Lemma 2** Let $s\colon \mathbb{N}^+ \to \mathbb{N}^+$ be a steady position valuation with at least three valuated positions (in the other case there is nothing to show). Define $s'\colon \mathbb{N}^+ \to \mathbb{N}^+$ by $s'(i) \stackrel{\text{def}}{=} s(i+1)$ for all $i \in \mathbb{N}^+$, that is, if $s(i+1)$ is undefined, so is $s'(i)$. Let $G = \langle N, T, S, P\rangle \in WGCSG_s$ be of order $l \geq 3$, let $f\colon N \cup T \to \mathbb{N}$ be a symbol valuation for G.

Proof of Lemma 2-2 We now construct a grammar $G' = \langle N', T, S', P'\rangle$ and a symbol valuation $f'\colon N' \cup T \to \mathbb{Q}$ by the following algorithm:

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

It holds that $L(G') = L(G)$. As G is context-sensitive, so is G'. By calculation and Lemma 1 it can be shown that $G' \in WGCSG_{s'}$. By Proposition 2 this implies $G' \in WGCSG_s$ (since $s(i) = \mathrm{w}(s) \cdot s'(i)$ for every $i \in \mathbb{N}^+$).

Proof of Lemma 2 □

3.2-4 For unsteady position valuations, this algorithm and similar ones do not work. This can be checked looking at the following interesting example.

Example 2 Let $s = \mathrm{id}$, 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 $f(\alpha_1) = 1$, $f(\alpha_2) = 1$, $f(\alpha_3) = 3$, $f(\beta_1) = 1$, $f(\beta_2) = 5$, and $f(\beta_3) = 1$. Then the rule is id-weakly 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 cannot be valuated appropriately.

Example 2 □

3.2-5 Lemma 2 can be applied step by step; thus together with Proposition 1 we obtain the following.

Theorem 3 *Let s be a steady position valuation. For every grammar $G \in WGCSG_s$, there exists an equivalent grammar $G' \in WGCSG_s$ in Cremers normal form.*

3.2-6 Thus for a steady position valuation s, the class $WGCSL_s$ is characterized by the growth factor $\mathrm{w}(s)$ (this follows from Theorem 3 and Proposition 2). We introduce the following notation:

$$WGCSL_{\mathrm{w}(s)} \stackrel{\mathrm{def}}{=} WGCSL_s$$

3.2-7 When we treat steady position valuations, we can assume wlog 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 4 *Let s be a steady position valuation. Then*

$$WGCSL_{\mathrm{w}(s)} = WGCSL_{\frac{1}{\mathrm{w}(s)}}$$

That is, having a position valuation s, we can assume wlog it is monotone increasing.

Proof of Theorem 4-1 **Proof of Theorem 4** Let s be a steady position valuation, let $G = \langle N, T, S, P \rangle \in WGCSL_s = WGCSL_{\frac{s(2)}{s(1)}}$ be in Cremers normal form (Theorem 3), and let f be a symbol valuation for G. We define a grammar $G' = \langle N, T, S, P' \rangle$, where P' contains the transposed version of every rule in P. Clearly, G' is context-sensitive and $L(G') = L(G)^T$. Define a position valuation s' by $s'(1) \stackrel{\text{def}}{=} s(2)$, $s'(2) \stackrel{\text{def}}{=} s(1)$. If $G' \in WGCSG_{s'} = WGCSG_{\frac{s(1)}{s(2)}}$, then the claim follows from Theorem 2.

Proof of Theorem 4-2 The transposed versions of the length-preserving rules clearly are s'-weakly growing with respect to f. Now we look at the expanding rules and define an appropriate symbol valuation f' for G'.

Proof of Theorem 4-3 Consider a rule $(A \to BC) \in P$. Then $s(1) \cdot f(A) < s(1) \cdot f(B) + s(2) \cdot f(C)$. If $s(2) \cdot f'(A) < s(2) \cdot f'(C) + s(1) \cdot f'(B)$ holds, the rule $A \to CB$ is s'-weakly growing with respect to f'. We attach an additional weight piece to B in order to fulfill this inequality, that is, we define $f'(B) \stackrel{\text{def}}{=} f(B) + b$ where $b \stackrel{\text{def}}{=} \frac{s(2)-s(1)}{s(1)} \cdot m$ with $m \geq f(A)$. Attaching this same weight piece to A and C does not affect the validity of the given inequality.

Proof of Theorem 4-4 We define $m \stackrel{\text{def}}{=} \max\{\, f(A) : A \in N \,\}$, and define $f' \colon N \cup T \to \mathbb{Q}$ by

$$f'(X) \stackrel{\text{def}}{=} f(X) + \frac{s(2) - s(1)}{s(1)} \cdot m \text{ for } X \in N \cup T$$

Then every transposed version of an expanding rule in P is s'-weakly growing with respect to f'. As f' differs from f only by a constant addend, the transposed versions of the length-preserving rules in P are s'-weakly growing with respect to f' also. Thus $G' \in WGCSG_{s'}$ follows from Lemma 1.

Proof of Theorem 4 □

4 The Unsteady Position Valuation

4-1 In this section it is shown that $WGCSL_s = CSL$ for every unsteady position valuation s. The only inclusion we have to show is $CSL \subseteq WGCSL_s$.

4-2 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

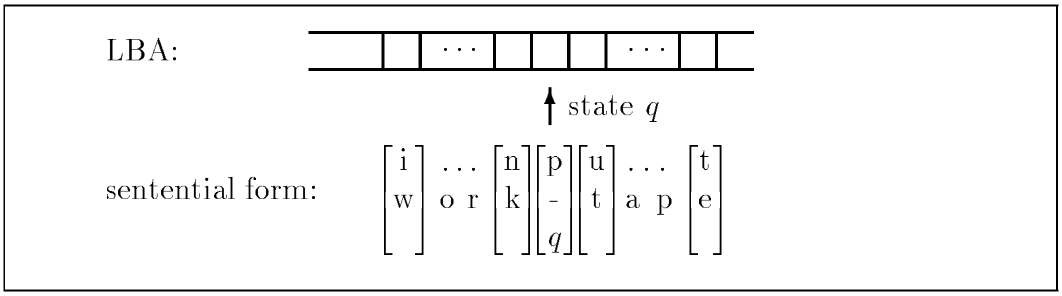

Figure 1: Tape and sentential form (sketch)

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., [Tur36], [Har78], [HU79]). Each configuration is represented by a certain sentential form (see Figure 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. This sentential form is generated with the initial configuration with an arbitrary 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.

4-3 To simulate 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.

4-4 As we mentioned earlier, for every unsteady position valuation s, $WGCSL_s = CSL$. To prove this, we use that every unsteady position valuation s has a blip (see Definition 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 an initial part up to the position $j+1$, and

(iv) s is monotone decreasing on an initial 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 construct-

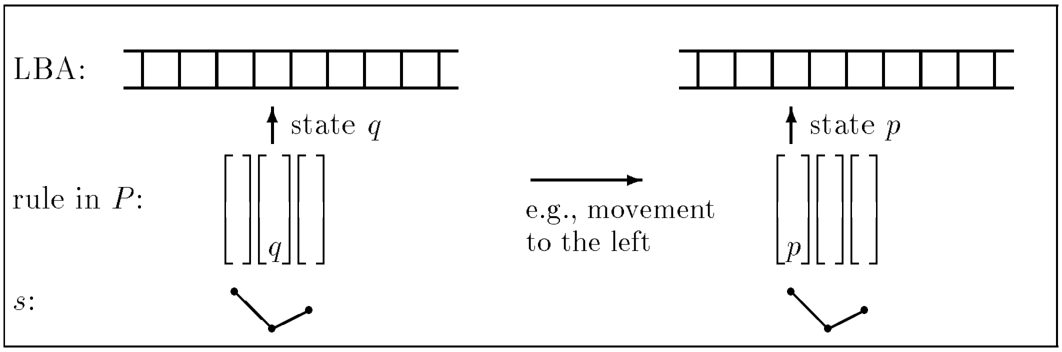

Figure 2: Simulation in the first case (sketch)

ing an s-weakly growing context-sensitive grammar following the algorithm mentioned above.

4-5 **For the first case (i):** Given a linear bounded automaton $M = (\Sigma, Q, \Gamma, q_0, \delta, F)$, we construct the simulating grammar $G = \langle N, T, S, P\rangle$ as follows. N consists of triples, where the first component contains an element of Σ, the second an element of Γ, and the third is empty or contains a state. Additionally, N contains marked copies of those symbols for the right-hand and the left-hand borders. The idea is to valuate (by an appropriately defined symbol valuation) all triples without state with the same value, and the triples containing a state with a greater value.

4-6 The "valley blip" in the position valuation is then used as follows. In a rule that simulates a step of M, the state "moves" following the move of the automaton's head. Thus, the additional weight induced to a nonterminal by the state also "moves," and by an appropriate left-hand context (that is, by $j-2$ preceding nonterminals for moving to the left, $j-1$ for moving to the right, where the position j is the "valley blip"), the rule is constructed so that on the left-hand side the state is positioned exactly on that blip. In Figure 2 this is illustrated. Thus, for $(p, y, R) \in \delta(q, x)$, the rules

$$\alpha \begin{bmatrix} a \\ x \\ q \end{bmatrix} \begin{bmatrix} b \\ z \\ \end{bmatrix} \to \alpha \begin{bmatrix} a \\ y \\ \end{bmatrix} \begin{bmatrix} b \\ z \\ p \end{bmatrix}$$

for $\alpha \in N^{j-1}$ (nonstate symbols only), $a, b \in \Sigma$, $z \in \Gamma$ are in P; for $(p, y, L) \in \delta(q, x)$ the rules

$$\alpha \begin{bmatrix} a \\ z \\ \ \end{bmatrix} \begin{bmatrix} b \\ x \\ q \end{bmatrix} \rightarrow \alpha \begin{bmatrix} a \\ z \\ p \end{bmatrix} \begin{bmatrix} b \\ y \\ \ \end{bmatrix}$$

for $\alpha \in N^{j-2}$ (nonstate symbols only), $a, b \in \Sigma$, $z \in \Gamma$ are in P.

4-7 Since the positions to the right and to the left of the "valley blip" j are both valuated higher than the blip itself, these rules become s-weakly growing with a symbol valuation according to the conditions we stated above. To continue working, that is, to apply such a rule on a sentential form, there must exist at least $j-1$ symbols in the sentential form to the left of the state.

4-8 Therefore, at the left-hand border we condense several steps: For $w, w' \in \Sigma^{j-1}$, $w = w_1 \dots w_{j-1}$, $w' = w'_1 \dots w'_{j-1}$, $x, y \in \Sigma$,$q, p \in Q$ with $wqx \underset{M}{\overset{*}{\vdash}} w'yp$ the rules

$$\begin{bmatrix} a_1 \\ w_1 \\ \heartsuit \end{bmatrix} \begin{bmatrix} a_2 \\ w_2 \\ \ \end{bmatrix} \dots \begin{bmatrix} a_{j-1} \\ w_{j-1} \\ \ \end{bmatrix} \begin{bmatrix} a \\ x \\ q \end{bmatrix} \begin{bmatrix} b \\ z \\ \ \end{bmatrix} \rightarrow \begin{bmatrix} a_1 \\ w'_1 \\ \heartsuit \end{bmatrix} \begin{bmatrix} a_2 \\ w'_2 \\ \ \end{bmatrix} \dots \begin{bmatrix} a_{j-1} \\ w'_{j-1} \\ \ \end{bmatrix} \begin{bmatrix} a \\ y \\ \ \end{bmatrix} \begin{bmatrix} b \\ z \\ p \end{bmatrix}$$

for $a_1 \dots a_{j-1}, a, b \in \Sigma$, $z \in \Gamma$ are in P. (The $\heartsuit$ here serves as a mark for the left-hand border.)

4-9 To build an initial sentential form for the input $a_1 \dots a_j \dots a_n$, we use an analogous condensation of steps: For $q_0 a_1 \dots a_{j-1} \underset{M}{\overset{*}{\vdash}} a'_1 \dots a'_{j-1} p_0$ we build

$$\begin{bmatrix} a_1 \\ a'_1 \\ \heartsuit \end{bmatrix} \begin{bmatrix} a_2 \\ a'_2 \\ \ \end{bmatrix} \dots \begin{bmatrix} a_{j-1} \\ a'_{j-1} \\ \ \end{bmatrix} \begin{bmatrix} a_j \\ a_j \\ p_0 \end{bmatrix} \begin{bmatrix} a_{j+1} \\ a_{j+1} \\ \ \end{bmatrix} \dots \begin{bmatrix} a_n \\ a_n \\ \diamondsuit \end{bmatrix}$$

(where $\heartsuit$ serves to mark the left-hand, and $\diamondsuit$ to mark the right-hand borders).

4-10 Triples containing an accepting state are transformed into their first component, which then become terminal symbols. Every triple near a terminal is transformed in this way, too. For a detailed and explicit construction of the grammar see [Nie94].

4-11 **For the second case (ii):** In the case of a "peak blip" we can use the same algorithm as we used in the first case. In fact, we can use the same grammar. We just have to redefine the symbol valuation so that the nonterminals representing simple tape cells are valuated higher than the nonterminals containing a state.

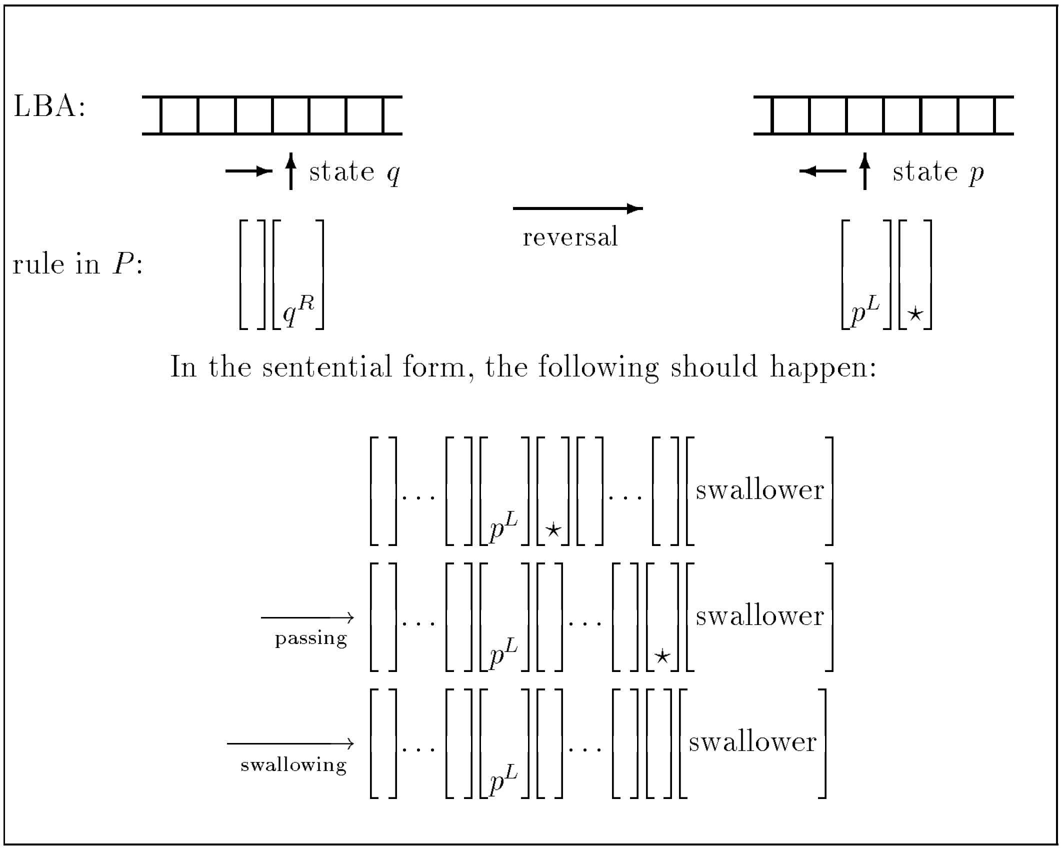

Figure 3: Simulation with swallower

4-12 **For 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 to the right and with a lower value for moving to the left. To do this, we mark the state with the direction it moved the last time. With a reversal (change of direction of the move) this value must be changed. It is no problem in a weakly growing grammar to change from a lower to a greater value. But going 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 beside the point of reversal. From there it is passed through to the right to be swallowed. This is illustrated in Figure 3. It is no problem to let the "surplus weight" (encoded in the star $\star$) pass through to the right.

4-13 We now introduce a swallower. Informally, a swallower consists of a string θ together with some rules that can be integrated into a given s-weakly growing context-sensitive grammar G so that every 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 6 *Let s be an unsteady position valuation which is monotone increasing on an initial part up to a position after its first blip, and let $G = \langle N, T, S, P\rangle \in WGCSG_s$. Define a swallower for s and G (also called an s-*swallower for *G) to be a quintuple $\mathcal{S}_{s,G} = (N_1, N_2, N_{swallow}, \theta, P_{swallow})$ such that:*

- *The components of the quintuple are*
  - *N_1 and N_2 are nonempty subsets of N,*
  - *$N_{swallow}$ is a set of nonterminals (not necessarily a subset of N),*
  - *$\theta \in N^*_{swallow}$ is a string, and*
  - *$P_{swallow}$ is a set of rules of the form $\alpha \to \beta$ where $\alpha, \beta \in (N \cup N_{swallow} \cup T)^*$. Every rule in $P_{swallow}$ is context-sensitive.*
- *There exists a symbol valuation $f\colon N \cup N_{swallow} \cup T \to \mathbb{N}$ with*
  - *f is a symbol valuation for G, i.e., every rule in P is s-weakly growing with f,*
  - *every rule in $P_{swallow}$ is s-weakly growing with f, and*
  - *for every $X \in N_1, Y \in N_2$, it holds that $f(X) < f(Y)$.*
- *There exists an $l \in \mathbb{N}$ such that for every $w \in (N \cup T)^*$ with $|w| \geq l$ and for every $X \in N_1, Y \in N_2$, it holds that $wY\theta \underset{P_{swallow}}{\overset{*}{\Longrightarrow}} wX\theta$.*

If $N_{swallow} \subseteq N$ and $P_{swallow} \subseteq P$, we also call $\mathcal{S}_{s,G}$ an s-swallower in *G. If s and G are clear from the context, the subscripts are dropped.*

4-14 To illustrate how a swallower can be built, we give an example first.

Example 3-1 **Example 3** Take the unsteady position valuation id defined by $\mathrm{id}(i) \overset{\text{def}}{=} i$ for every $i \in \mathbb{N}^+$.

Example 3-2 Let $G = \langle N, T, S, P\rangle \in WGCSG_{\mathrm{id}}$ with $X, Y \in N$, and let f be a symbol valuation for G with $f(X) = 1$, $f(Y) = 2$.

Example 3-3 Define $\mathcal{S} = (N_1, N_2, N_{\text{swallow}}, \theta, P_{\text{swallow}})$ with $N_1 \stackrel{\text{def}}{=} \{X\}$, $N_2 \stackrel{\text{def}}{=} \{Y\}$, $N_{\text{swallow}} \stackrel{\text{def}}{=} \{M, [\star], A, B, V\}$, $\theta \stackrel{\text{def}}{=} MAV$, and a set of rules P_{swallow} comprising

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

Using these 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:

$$\begin{array}{lll} f(M) \stackrel{\text{def}}{=} 1 & f([\star]) \stackrel{\text{def}}{=} 2 & \\ f(A) \stackrel{\text{def}}{=} 1 & f(B) \stackrel{\text{def}}{=} 8 & f(V) \stackrel{\text{def}}{=} 5 \end{array}$$

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

$$\begin{array}{lrcl} \text{for (R1):} & 1 \cdot 2 + 2 \cdot 1 = & 4 < 5 & = 1 \cdot 1 + 2 \cdot 2 \\ \text{for (R2):} & 1 \cdot 2 + 2 \cdot 1 + 3 \cdot 5 = & 19 < 20 & = 1 \cdot 1 + 2 \cdot 8 + 3 \cdot 1 \\ \text{for (R3):} & 1 \cdot 8 + 2 \cdot 1 = & 10 < 11 & = 1 \cdot 1 + 2 \cdot 5 \end{array}$$

Thus, these rules are id-weakly growing with f.

Example 3-4 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 must 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 is $\left\lceil \frac{1}{2} \cdot 1 \right\rceil + \frac{3}{2} \cdot 4 = 7 = f(B) - f(A)$ and happens in (R3)) the situation of the beginning is restored.

Example 3-5 In this way, the sentential form MAV together with the set of rules P_{swallow} swallows a value encoded in the star, and $\mathcal{S}$ is an id-swallower in G.

Example 3 □

4-15 Now we turn to the construction of a swallower for each unsteady position valuation that is monotone increasing at least up to a position after its first blip.

4-16 **Lemma 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.*

4-17 Note that this lemma also means that concerning an unsteady position valuation, a positive growth rate of a rule does not necessarily mean a positive growth rate of the corresponding derivation step.

Proof of Lemma 3-1 **Proof of Lemma 3** 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.

Proof of Lemma 3-2 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 $X \in N_1, Y \in N_2$. Define $N_{\text{swallow}} \stackrel{\text{def}}{=} \{M, [\star], A, B, V\}$, $\theta \stackrel{\text{def}}{=} MAV$, let

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

and define P_{swallow} as follows:

$$\begin{aligned} R_1 &\stackrel{\text{def}}{=} \left\{ \alpha Y M \to \alpha X[\star] : \alpha \in (N \cup T)^l,\, Y \in N_2,,\, X \in N_1 \right\} \\ R_2 &\stackrel{\text{def}}{=} \left\{ \alpha[\star]AV \to \alpha MBA : \alpha \in (N \cup T)^{j-2} \right\} \\ R_3 &\stackrel{\text{def}}{=} \left\{ \alpha BA \to \alpha AV : \alpha \in (N \cup T)^{j \dot{-} 3} \cdot \{M, \varepsilon\},\ |\alpha| = j-2 \right\} \\ P_{\text{swallow}} &\stackrel{\text{def}}{=} R_1 \cup R_2 \cup R_3 \end{aligned}$$

Note that the lengths of α in R_2 and R_3 are different, and that M appears in R_3 iff $j > 2$.

Proof of Lemma 3-3 Now we extend the symbol valuation f to N_{swallow} with values in $\mathbb{Q}$ as follows, where $\mu \stackrel{\text{def}}{=} \max\{ f(Y) - f(X) : X \in N_1,\, Y \in N_2 \}$:

$$f(M) \stackrel{\text{def}}{=} 1 \qquad f([\star]) \stackrel{\text{def}}{=} 1 + \mu \qquad f(A) \stackrel{\text{def}}{=} 1$$

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

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

Proof of Lemma 3-4 It can be checked arithmetically that the rules in P_{swallow} all are s-weakly growing with f. (To check R_2 and R_3, it is helpful first to compute $s(j)\cdot$

$f(B) - s(j+1) \cdot f(V)$ and $s(j) \cdot f(V) - s(j-1) \cdot f(B)$, respectively.) As we have seen in Lemma 1, f can be transformed into a symbol valuation with values in $\mathbb{N}^+$.

Proof of Lemma 3-5 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:

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

Proof of Lemma 3 □

4-18 So we can integrate a swallower into an s-weakly 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).

4-19 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 disappears. If s on a beginning part is constant, we have the same conditions for border-handling during the simulation as we had in the first and the second cases.

4-20 **For 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. This means the "surplus weight" is generated in a left-right reversal, and is passed through to the left, where it is swallowed. The swallower is also 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-hand context of the rules in R_3, and the left-hand 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.

4-21 **Conclusion:** Since in all cases (i) through (iv) we can construct an appropriate weakly growing context-sensitive grammar, we have the following:

Theorem 5 *For every unsteady position valuation s, it holds that*

$$WGCSL_s = CSL$$

4-22 For a more detailed description, see [Nie94].

5 The Steady Position Valuation

5-1 We intend to find a characterization of a weakly growing context-sensitive language class related to a steady position valuation by linear bounded automata with a certain time bound. We approach this goal in three steps: First, we estimate the length of a derivation in a given weakly growing context-sensitive grammar related to a steady position valuation, and realize an efficient simulation (Section 5.1). Second, we introduce an instrument that will help in the simulation of linear bounded automata by weakly growing context-sensitive grammars related to a steady position valuation. This instrument is a new kind of counter: grammars that count, where the counted items in fact are weight pieces (Section 5.2). With these preparations the simulation will be done easily (Section 5.3). Note, however, that with our methods it is not possible to show a one-to-one correspondence, due to the counter's capacity.

5.1 Efficient Simulation of a Grammar

5.1.1 Connected Grammars

5.1.1-1 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 7), while in a nonconnected grammar, in every step it must 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 7 *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, and let $(\alpha_i \to \beta_i) \in P$ be 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.*

5.1.1-2 Gladkii showed that every grammar (of both general and context-sensitive types) can be transformed into an equivalent grammar 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 will show that the same is true for s-weakly growing context-sensitive grammars, for every steady position valuation s.

5.1.1-3 Book's proof carries over easily when we use Cremers normal form (Theorem 3), and we can assume that the position valuation is monotone increasing (see Theorem 4).

Lemma 4 *Let s be a steady, strictly monotone increasing position valuation. Then for each grammar $G \in WGCSG_s$ in Cremers normal form, there 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 length in G.*

Proof of Lemma 4-1 **Proof of Lemma 4** Let $G = \langle N, T, S, P\rangle \in WGCSG_s$ in Cremers normal form. Let f be a symbol valuation for G.

Proof of Lemma 4-2 Define a grammar $G' = \langle N', T, \hat{S}, P'\rangle$ by $N' \stackrel{\text{def}}{=} N \cup \left\{ \hat{A} : A \in N \right\}$, that is, the old set of nonterminals together with a set of new nonterminals in one-to-one correspondence with the old, and the following rules:

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

Following an argument given in [Boo69] we see that $L(G') = L(G)$, and G' is a connected context-sensitive grammar. Now define a symbol valuation $f' \colon N' \cup T \to \mathbb{N}$ for G' by:

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

Then every production in P' is s-weakly growing with f' as shown below.

Proof of Lemma 4-3 Each rule from P grows at least by one with respect to f. The factor $s(2)^2$ in the definition of f' causes this growth rate to rise 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 take a closer look at the second rule in P_1. Because $AB \to CD$ is s-weakly growing with f, the following applies:

$$s(1) \cdot f(A) + s(2) \cdot f(B) + 1 \leq s(1) \cdot f(C) + s(2) \cdot f(D)$$

so

$$s(1) \cdot s(2)^2 \cdot f(A) + s(2)^3 \cdot f(B) + s(2)^2 \leq s(1) \cdot s(2)^2 \cdot f(C) + s(2)^3 \cdot f(D)$$

Thus, for the new rule $A\hat{B} \to \hat{C}D$, it holds that

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

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

Proof of Lemma 4 □

5.1.1-4 It is possible to use the same idea for other position valuations. We even have connectivity if there is no known transformation into Cremers normal form.

5.1.2 Efficient Simulation

5.1.2-1 When we simulate a derivation with a weakly growing context-sensitive grammar related to a steady position valuation by a linear bounded automaton, we use the algorithm of Section 4 and Lemma 4. To estimate the time bound for the linear bounded automaton, we estimate the length of the derivation in the weakly 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 us to valuate arbitrarily long sentential forms. With every application of a rule, the value of the sentential form increases (at least by one).

Lemma 5 *Let s be an infinite steady monotone increasing position valuation. Let $G = \langle N, T, S, P \rangle$ be an s-weakly 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 $u \Longrightarrow v$, and let $f \colon N \cup T \to \mathbb{N}^+$ be a symbol valuation for G. Then the following applies:*

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

That is, concerning steady position valuations, the positive growth rate of a rule induces a positive growth rate of the corresponding derivation step.

Proof of Lemma 5 Let $(\alpha \to \beta) \in P$ be the rule used in the derivation step $u \Longrightarrow v$. The growth rate of that rule is at least 1; we denote it by g. Let us now valuate the whole sentential forms $u = u_1 \alpha u_2$ and $v = u_1 \beta u_2$, and look at the growth rate of the derivation step $u \underset{G}{\Longrightarrow} v$. We obtain the value $g \cdot \mathrm{w}(s)^{|u_1|}$ where $\mathrm{w}(s)^{|u_1|} \geq 1$, if r is length-preserving, and $g \cdot \mathrm{w}(s)^{|u_1|} + b$ where, additionally, $b > 0$, if it is length-increasing. As there are no length-decreasing rules, this implies the claim.

Proof of Lemma 5 □

5.1.2-2 As we saw in Section 4 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 weakly growing rule (see Lemma 3). On the other hand, 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.

5.1.2-3 We conclude from Lemma 5 and Definition 3:

Lemma 6 *Let s be an infinite steady strictly monotone increasing position valuation. Let $G = \langle N, T, S, P \rangle \in WGCSG_s$, let f be an appropriate symbol valuation for G. Let $c = \max\{ f(A) : A \in N \cup T \}$. Let $v \in L(G)$. Then the length of a derivation $S \overset{*}{\Longrightarrow} v$ is not greater than*

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

5.1.2-4 Now we can estimate the computation time a linear bounded automaton needs to simulate a weakly growing context-sensitive grammar related to a steady position valuation using Theorem 4, Lemma 4, and Lemma 6. (Here, $T\text{-}NSPACE\text{-}TIME(s,t)$ 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 6 *Let s be a steady nonconstant position valuation. Let* $\mathrm{w} = \max\{\mathrm{w}(s), \frac{1}{\mathrm{w}(s)}\}$. *Then:*

$$WGCSL_s \subseteq T\text{-}NSPACE\text{-}TIME(n, O(\mathrm{w}^n))$$

5.2 Counters

5.2-1 We will introduce a counter as an instrument to lay down encoded weight pieces, consisting of a string and some rules. We will use it in a simulation of a linear bounded automaton by a weakly growing context-sensitive grammar. The capacity of the counter used will cause a bound on the number of steps that can be simulated.

Definition 8-1 **Definition 8** *Let s be a steady strictly monotone increasing position valuation, and let $G = \langle N, T, S, P\rangle \in WGCSG_s$. Define a* counter for G related to the position valuation s, *also called an* s-counter for G, *to be a quintuple* $\mathcal{Z}_{s,G} = (N_1, N_2, N_{count}, (z_n)_{n\in\mathbb{N}^+}, P_{count})$, *such that:*

- *The components of the quintuple are*
  - N_1 *and* N_2 *are nonempty 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* $z_n \in N^n_{count}$ *for every* $n \in \mathbb{N}^+$, *and*
  - P_{count} *is a finite subset of* $(N_2\cdot N^*_{count}\times N_1\cdot N^*_{count})\cup(N^*_{count}\times N^*_{count})$; *every rule in* P_{count} *is length-preserving.*
- *There exists a symbol valuation* $f\colon N\cup N_{count}\cup T \to \mathbb{N}$ *where*
  - f *is a symbol valuation for* G, *i.e., every rule in* P *is* s*-weakly growing with* f,
  - *every rule in* P_{count} *is* s*-weakly growing with* f, *and*

– *for every* $X \in N_1, Y \in N_2$, *it holds that* $f(X) < f(Y)$.

We say that $f|_{N_1 \cup N_2 \cup N_{count}}$ *is a* symbol valuation for $\mathcal{Z}$.

Definition 8-2 *If* $N_{count} \subseteq N$ *and* $P_{count} \subseteq P$, *we also call* $\mathcal{Z}_{s,G}$ *an s-counter* in G. *If* s *and* G *are clear from the context, the subscripts are dropped.*

5.2-2 For every $X \in N_1$ and $Y \in N_2$, the counter counts the weight piece $f(Y) - f(X)$ by applying an appropriate rule out of $N_2 \cdot N^*_{\text{count}} \times N_1 \cdot N^*_{\text{count}}$ (and after that possibly 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 as mentioned above on Yz'_n, and so on. To use the counter to count several amounts of weight, we apply corresponding increment rules (which are not in P_{count}, but will be added to P in order to use the counter in G).

Definition 9-1 **Definition 9** *Let* s *be a steady strictly monotone increasing position valuation, let* $G\langle N, T, S, P\rangle \in WGCSG_s$. *Let* $\mathcal{Z}$ *be an s-counter for* G, *and let* f *be a symbol valuation for* G *and for* $\mathcal{Z}$.

Definition 9-2 *The set* $\rho(\mathcal{Z}, f)$ *of* increment rules for $\mathcal{Z}$ with f *is defined by:*

$$\rho(\mathcal{Z}, f) \stackrel{def}{=} \{ (X \to Y) : X \in N_1,\ Y \in N_1 \cup N_2, \text{ and } f(X) < f(Y) \}$$

5.2-3 Consider a derivation 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. We are interested in the number of times an increment rule can be used (and the weight piece generated by it can be handled) in such a derivation. So, we define the *capacity* $K_{\mathcal{Z}}(n)$ to be the maximum number of times that it is possible to have an occurrence of an increment rule in a derivation starting with Xz_n and ending with $X'\tilde{z}$, where $X' \in N_1$ and $|\tilde{z}| = |z_n|$. Note that there are two kinds of increment rules. The first kind (which we call a "collecting weights" rule) leaves the first symbol in N_1. The second kind of increment rule (which replaces the first symbol from one in N_1 to one in N_2) is the kind of increment rule that directly influences the counter.

Definition 10 *Let* s *be a steady strictly monotone increasing position valuation, let* $G = \langle N, T, S, P\rangle \in WGCSG_s$, *let* $\mathcal{Z}$ *be an s-counter for* G, *let* z_n *be the start string with index* n, *and let* $V_{\mathcal{Z},G}$ *be the set of all symbol valuations* *for $\mathcal{Z}$ and G. The capacity function $K_{\mathcal{Z}}: \mathbb{N}^+ \to \mathbb{N}^+$ of $\mathcal{Z}$ is defined as follows:*

$$K_{\mathcal{Z}}(n) \stackrel{def}{=} \max \left\{ |\{ j : r_j \in \rho(\mathcal{Z}, f) \}| : \begin{array}{l} f \in V_{\mathcal{Z},G},\ X, X' \in N_1,\ \tilde{z} \in N^n_{count},\ k \in \mathbb{N}, \\ (r_1, \ldots, r_k) \in (P_{count} \cup \rho(\mathcal{Z}, f))^k, \\ X z_n \underset{r_1}{\Longrightarrow} \ldots \underset{r_k}{\Longrightarrow} X' \tilde{z} \end{array} \right\}$$

Because the set N is finite and each of the rules is length-preserving, the capacity of a counter certainly is well defined. Because of the possibility of collecting weights (in fact, we only need to extend N_1 to raise the capacity of a counter by a constant factor), there is for every counter $\mathcal{Z}$ and every $c \in \mathbb{N}^+$ a counter $\mathcal{Z}'$ with $K_{\mathcal{Z}'} = c \cdot K_{\mathcal{Z}}$.

5.2-4 Let us now have a look at a specific counter to see how it works.

Example 4-1 **Example 4** Let s be a steady strictly monotone increasing position valuation. Let $G = \langle N, T, S, P \rangle \in WGCSG_s$ with $\{\#, M\} \subseteq N$. Define $\mathcal{Z} = (N_1, N_2, N_{\text{count}}, (z_n)_{n \in \mathbb{N}^+}, P_{\text{count}})$ by: $N_1 \stackrel{\text{def}}{=} \{M\}$, $N_2 \stackrel{\text{def}}{=} \{\#\}$, $N_{\text{count}} \stackrel{\text{def}}{=} \{[0], [1], \ldots, [k]\}$ for a $k \in \mathbb{N}^+$, $z_n \stackrel{\text{def}}{=} [0]^n$ for each $n \in \mathbb{N}^+$, and P_{count} consisting of the rules:

$$\begin{array}{lrl} \text{(R1)} & \#[i] \to M[i+1] & \\ \text{(R2)} & [i+1][i] \to [i][i+1] & \end{array} \qquad \text{for } i = 0, \ldots, k-1$$

The symbol valuation f defined by

$$f(M) \stackrel{\text{def}}{=} 1,\ f(\#) \stackrel{\text{def}}{=} 2,\ f([i]) \stackrel{\text{def}}{=} i+1 \text{ for } i = 0, \ldots, k$$

shows that $\mathcal{Z}$ is an s-counter for G. $M \to \#$ is the only possible increment rule, thus the capacity is $K_{\mathcal{Z}}(n) = k \cdot n$.

Example 4-2 If we change rule (R2) to

$$\text{(R2}'\text{)} \quad [i+1][i] \to [0][i+1] \qquad \text{for } i = 0, \ldots, k-1$$

we obtain a counter with polynomial capacity. (In the symbol valuation define $f([i]) \stackrel{\text{def}}{=} \mathrm{w}(s)^i$ for $i = 0, \ldots, k$.)

Example 4 □

5.2-5 We have seen counters with linear and with polynomial capacities. What about exponential capacity? As we reach for an inversion of Theorem 6, this is our goal.

5.2-6 Let us 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.

Example 5-1 **Example 5** Let s_2 be the position valuation defined by $s_2(i) \stackrel{\text{def}}{=} 2^{i-1}$ for every $i \in \mathbb{N}^+$. Let $G = \langle N, T, S, P\rangle \in WGCSG_{s_2}$ with $\{\#, M\} \subseteq N$. Define $\mathcal{Z} = (N_1, N_2, N_{\text{count}}, (z_n)_{n\in\mathbb{N}^+}, P_{\text{count}})$ by $N_1 \stackrel{\text{def}}{=} \{M\}$, $N_2 \stackrel{\text{def}}{=} \{\#\}$, $N_{\text{count}} \stackrel{\text{def}}{=} \{[0], [1], [U]\}$, $z_n \stackrel{\text{def}}{=} [0]^n$ for each $n \in \mathbb{N}^+$ and P_{count} consisting of the rules:

$$\begin{array}{llll} \text{(R1)} & \#[0] \to M[1] & \text{(R3)} & [U][0] \to [0][1] \\ \text{(R2)} & \#[1] \to M[U] & \text{(R4)} & [U][1] \to [0][U] \end{array}$$

Using rules (R1)–(R4) above, together with the rule $M \to \#$, for every start string z_n the following derivation steps are possible in $\mathcal{Z}$:

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

At first glance, this looks like an s_2-counter with 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 $f\colon N_1 \cup N_2 \cup N_{\text{count}} \to \mathbb{N}$ be a symbol valuation. Then we have:

For (R3):

$$\begin{array}{c} 1 \cdot f([U]) + 2 \cdot f([0]) < 1 \cdot f([0]) + 2 \cdot f([1]) \\ \text{which implies } f([U]) - f([1]) < f([1]) - f([0]) \end{array}$$

For (R4):

$$\begin{array}{c} 1 \cdot f([U]) + 2 \cdot f([1]) < 1 \cdot f([0]) + 2 \cdot f([U]) \\ \text{which implies } f([1]) - f([0]) < f([U]) - f([1]) \end{array}$$

Example 5-2 Both of these inequalities cannot be fulfilled at the same time. Thus there exists no symbol valuation with which all rules of P_{count} are s_2-weakly growing. This implies that $\mathcal{Z}$ is not an s_2-counter, for every $G \in WGCSG_{s_2}$.

Example 5 $\square$

5.2-7 The reason is, that in every rule we transport a portion of the weight piece plus an increment for making the rule weakly 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 piece, because this sum is to be handled in the same way, arbitrarily often.

5.2-8 We will now look at the upper bound for the capacity of a counter $\mathcal{Z}_{s,G}$ which works after the following mechanism: It treats the start string as a number (where each digit of that number is represented by several symbols (bits) of the string) and increments this encoded number arithmetically. We fix the number of symbols used to represent a digit by $l \in \mathbb{N}$. How many different digits $[0], [1], \ldots, [i_{\max}]$ can now be represented by l bits of the string without violating the property of $\mathcal{Z}_{s,G}$ to be an s-counter? We have seen in Example 5 that, e.g., for $l = 1$ to choose $i_{\max} = 1$ (that is, choosing the base of the number representation to be 2) violates this property.

5.2-9 This number of digits representable by l bits while respecting the s-counter property determines the base of the number representation that we use to count weight pieces. Thus, it also determines the maximal capacity reachable by such a counter.

5.2-10 In the following lemma, $[0], [1], \ldots, [i_{\max}]$, and $[U]$ are metasymbols, each standing for l letters (bits) that encode the digits $0, 1, \ldots, i_{\max}$ of the number representation, and the l letters of $[U]$ represent the overflow, which contains 0 and a carry for the next position.

Lemma 7-1 **Lemma 7** *Let s be an infinite steady strictly monotone increasing position valuation. Let $G = \langle N, T, S, P\rangle \in WGCSG_s$ with $\{\#, M\} \subseteq N$, and let $\mathcal{Z} = (N_1, N_2, N_{count}, (z_n)_{n\in\mathbb{N}^+}, P_{count})$ be a counter with $M \in N_1$, $\# \in N_2$. Let $l \in \mathbb{N}^+$.*

Lemma 7-2 *If $\mathcal{Z}_{s,G}$ works after the following scheme (where $[0], [1], \ldots, [i_{\max}], [U] \in N^l_{count}$, $i_{\max} \in \mathbb{N}$, $i = 0, \ldots, i_{\max} - 1$):*

$$\begin{array}{llll} \text{(R1)} & \#[i] \to M[i+1] & \text{(R3)} & [U][i] \to [0][i+1] \\ \text{(R2)} & \#[i_{\max}] \to M[U] & \text{(R4)} & [U][i_{\max}] \to [0][U] \end{array}$$

and if $\mathcal{Z}_{s,G}$ is an s-counter, then the capacity of the counter $\mathcal{Z}_{s,G}$ is bounded by

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

where c is a constant depending on $\mathcal{Z}_{s,G}$.

5.2-11 Note that for $\mathrm{w}(s)^l \leq 2$ the capacity is at most linear, but for $\mathrm{w}(s)^l > 2$ it is at most exponential. As we stated above, the goal of this lemma is to estimate $i_{\max}$.

Proof of Lemma 7-1 **Proof of Lemma 7** Let f be a symbol valuation for $\mathcal{Z}$. Define a function $g\colon N_{\text{count}}^l \to \mathbb{N}^+$ by $g(a_1 \dots a_l) \stackrel{\text{def}}{=} \sum_{i=1}^{l} s(i) \cdot f(a_i)$, where $a_1, \dots, a_l \in N_{\text{count}}$. That is, g valuates the metasymbols $[0], [1], \dots, [i_{\max}], [U]$, each represented by l letters out of N_{count}, by condensing the symbol and the position valuation. Let w denote the growth factor $\mathrm{w}(s)$. Then we have:

Proof of Lemma 7-2 **For (R3):**

$$g([U]) - g([0]) < \mathrm{w}^l \cdot (g([i+1]) - g([i])) \text{ for } i = 0, \dots, i_{\max} - 1$$

Proof of Lemma 7-3 **For (R4):**

$$g([U]) - g([0]) < \mathrm{w}^l \cdot (g([U]) - g([i_{\max}]))$$

Let $\mu = g([U]) - g([0])$. This implies

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

For (R1) and (R2), $\mu > 0$, so $i_{\max} + 1 < \mathrm{w}^l$, from which it follows that $i_{\max} = \lceil \mathrm{w}^l - 2 \rceil$ (which means $i_{\max} = 0$ for $\mathrm{w}^l \leq 2$). Thus, we have a number representation to the base $\lceil \mathrm{w}^l - 1 \rceil$.

Proof of Lemma 7-4 Starting with $M[0]^{\lfloor \frac{n}{l} \rfloor} \alpha$, where $\alpha \in N_{\text{count}}^*$ with $|\alpha| + l \cdot \lfloor \frac{n}{l} \rfloor = n$, the string $M[i_{\max}]^{\lfloor \frac{n}{l} \rfloor} \alpha$ is derivable following the scheme (R1) through (R4) plus the increment rule ($M \to \#$). In such a derivation the increment rule is applied

$$\sum_{i=1}^{\lfloor \frac{n}{l} \rfloor} i_{\max} \cdot (i_{\max} + 1)^{i-1}$$

times. Then, with (R4), or (R2) if necessary, we can change the rightmost digit (i.e., metasymbol) to $[U]$: in doing so, the other digits change to [0]. Again we can add up (i.e., apply $(M \to \#)$ and the rules from the scheme) and thus derive $M[i_{\max}]^{\lfloor \frac{n}{l} \rfloor - 1}[U]\alpha$. Using this mechanism several times, we can fill up the string with overflows $[U]$. Thus, we derive $M[U]^{\lfloor \frac{n}{l} \rfloor}\alpha$.

Proof of Lemma 7-5 Unless P_{count} contains some rules to fill up α, now no more rules apply. The increment rule $(M \to \#)$ was applied

$$\sum_{i=1}^{\lfloor \frac{n}{l} \rfloor} (i_{\max} + 1) \cdot (i_{\max} + 1)^{i-1} = (i_{\max} + 1) \cdot \sum_{i=0}^{\lfloor \frac{n}{l} \rfloor - 1} (i_{\max} + 1)^i$$

times. If P_{count} contains some rules to fill up α, that is, to do derivation steps $[U]\alpha_j \Longrightarrow [0]\alpha_{j+1}$, where $\alpha_0 = \alpha$, $\alpha_j \in N^*_{\text{count}}$ for $j \in \mathbb{N}$, the number of these steps is bounded by a constant c, independent of n.

Proof of Lemma 7-6 To derive $M[U]^{\lfloor \frac{n}{l} \rfloor}\alpha_c$, the increment rule is used

$$(i_{\max} + 1) \cdot \sum_{i=0}^{\lfloor \frac{n}{l} \rfloor - 1} (i_{\max} + 1)^i + c \cdot (i_{\max} + 1)^{\lfloor \frac{n}{l} \rfloor}$$

times. Thus, the capacity of the counter $\mathcal{Z}_{s,G}$ can be estimated by:

$$\begin{aligned}
K_{\mathcal{Z}}(n) &\leq c \cdot \left\lceil \mathrm{w}^l - 1 \right\rceil^{\lfloor \frac{n}{l} \rfloor} + \left\lceil \mathrm{w}^l - 1 \right\rceil \cdot \sum_{i=0}^{\lfloor \frac{n}{l} \rfloor - 1} \left\lceil \mathrm{w}^l - 1 \right\rceil^i \\
&= c \cdot \left\lceil \mathrm{w}^l - 1 \right\rceil^{\lfloor \frac{n}{l} \rfloor} + \left\lceil \mathrm{w}^l - 1 \right\rceil \cdot \frac{\left\lceil \mathrm{w}^l - 1 \right\rceil^{\lfloor \frac{n}{l} \rfloor} - 1}{\lceil \mathrm{w}^l - 1 \rceil - 1} \\
&= c \cdot \left\lceil \mathrm{w}^l - 1 \right\rceil^{\lfloor \frac{n}{l} \rfloor} + \left\lceil \mathrm{w}^l - 1 \right\rceil^{\lfloor \frac{n}{l} \rfloor} - 1 + \frac{\left\lceil \mathrm{w}^l - 1 \right\rceil^{\lfloor \frac{n}{l} \rfloor} - 1}{\lceil \mathrm{w}^l - 1 \rceil - 1} \\
&\leq (c + 2) \cdot \left\lceil \mathrm{w}^l - 1 \right\rceil^{\lfloor \frac{n}{l} \rfloor}
\end{aligned}$$

Proof of Lemma 7 □

5.2-12 The upper bound stated in the lemma is optimal. That is, we can really find a counter working according to this adding mechanism and reaching the stated capacity. We put Lemma 8 with the exponent rounded up instead of rounded down, only because of technical reasons in the use of the lemma.

Lemma 8 *Let s be an infinite steady strictly monotone increasing position valuation and $G \in WGCSG_s$ with at least two different nonterminals, and let $c \in \mathbb{N}^+$. Let $l \in \mathbb{N}$. Then there exists an s-counter $\mathcal{Z}$, that has a capacity with*

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

Proof of Lemma 8-1

Proof of Lemma 8 First we consider $c = 1$. We define an s-counter

$$\mathcal{Z} = (N_1, N_2, N_{\text{count}}, (z_n)_{n \in \mathbb{N}^+}, P_{\text{count}})$$

following the scheme in Lemma 7. We will use the symbols of the string that are not used in a digit, which only occurs if n is not divisible by l, to "cheat" by using them to encode an additional digit. Thus we define:

$$N_1 \stackrel{\text{def}}{=} \{M\} \qquad N_2 \stackrel{\text{def}}{=} \{\#\}$$
$$N_{\text{count}} \stackrel{\text{def}}{=} \{D, [0], [1], \ldots, [i_{\max}], [U]\} \cup \{\langle 0\rangle, \langle 1\rangle, \ldots, \langle i_{\max}\rangle, \langle U\rangle\}$$
$$\text{where } i_{\max} = \left\lceil \mathrm{w}(s)^l - 2 \right\rceil$$

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 $n \in \mathbb{N}$. The set P_{count} consists of the following rules (where $i = 0, \ldots, i_{\max} - 1$):

$$\begin{array}{llll}
\text{(R1)} & \#D^{l-1}[i] \to MD^{l-1}[i+1] & \text{(R3)} & [U]D^{l-1}[i] \to [0]D^{l-1}[i+1] \\
\text{(R2)} & \#D^{l-1}[i_{\max}] \to MD^{l-1}[U] & \text{(R4)} & [U]D^{l-1}[i_{\max}] \to [0]D^{l-1}[U]
\end{array}$$

$$\begin{array}{ll}
\text{(R5)} & [U]\langle i\rangle \to [0]\langle i+1\rangle \\
\text{(R6)} & [U]\langle i_{\max}\rangle \to [0]\langle U\rangle
\end{array}$$

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

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

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

Proof of Lemma 8-2 The rules (R5) and (R6) give us the possibility to "cheat" by using an additional digit, if l does not divide n. The capacity now is calculated just as in the proof of Lemma 7, thus applies $K_{\mathcal{Z}}(n) \geq \left\lceil \mathrm{w}(s)^l - 1 \right\rceil^{\lceil \frac{n}{l} \rceil}$. By collecting weights we get a counter $\mathcal{Z}'$ with $K_{\mathcal{Z}'} = c \cdot K_{\mathcal{Z}}$ for every $c \in \mathbb{N}^+$.

Proof of Lemma 8 □

5.3 Simulation of Time-Bounded Linear Automata

5.3-1 The counters introduced in Lemma 8 now will be used in a simulation of a time-bounded linear-bounded automaton instead of the swallower we used in Section 4. The time bounds for linear bounded automata that can be managed with this counter are determined by the counter's capacity.

Theorem 7 *Let* $\mathrm{w} \in \mathbb{N}^+$*, let* $q \in \mathbb{Q}$ *with* $0 < q < 1$*. Then:*

$$\textit{T-NSPACE-TIME}\,(n, O(\mathrm{w}^{q \cdot n})) \subseteq \textit{WGCSL}_{\mathrm{w}}$$

Proof of Theorem 7-1 **Proof of Theorem 7** We only describe the intuitive operation of the grammar simulating a given linear-bounded automaton on a given $\mathrm{w} \in \mathbb{N}^+$. The construction of the grammar itself then is straightforward.

Proof of Theorem 7-2 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 $c \in \mathbb{N}^+$, and let s be an infinite steady position valuation with $\mathrm{w}(s) = \mathrm{w}$.

Proof of Theorem 7-3 Similar to Section 4, case (iii), we divide the sentential form into left-hand and right-hand parts: In the left-hand part the step-by-step simulation of the automaton takes place in the same manner as in the simulation mentioned above, except that we use tape compression; in the right-hand part, a counter is situated (see Figure 4).

Proof of Theorem 7-4 We choose an appropriate compression factor k so that $q < \frac{k-1}{k}$, and for this k we choose the width l of a digit so that $\left\lceil \mathrm{w}^l \right\rceil \geq \mathrm{w}^{l \cdot q \cdot \frac{k}{k-1}} + 1$. Note that this expression is equivalent to $\left\lceil \mathrm{w}^l - 1 \right\rceil^{\frac{k-1}{k} \cdot \frac{1}{l}} \geq \mathrm{w}^q$.

Proof of Theorem 7-5 Following Lemma 8 we can find an s-counter $\mathcal{Z}$ with the capacity $K_{\mathcal{Z}}(m) \geq c \cdot \left\lceil (\mathrm{w})^l - 1 \right\rceil^{\lceil \frac{m}{l} \rceil}$ where $m \stackrel{\mathrm{def}}{=} \left\lceil \frac{k-1}{k} \cdot n \right\rceil$ is the length of the part of the sentential form where the counter operates in the simulation for an input of length n. This implies

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

$$\underbrace{\overbrace{\begin{bmatrix}\text{LBA simulation}\\ \text{compressed}\end{bmatrix}}^{\frac{1}{k}\cdot n}\overbrace{\underbrace{D^{l-1}[i_1]}_{\text{lowest digit}}\ \underbrace{D^{l-1}[i_2]}_{\text{2. digit}}\ldots\underbrace{D^{l-1}[i_{\max}]}_{\text{highest digit}}}^{\frac{k-1}{k}\cdot n}}_{\text{length of the derived word (i.e., the original input)}}$$

Figure 4: Simulation with a steady position valuation: sentential form

This means that 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 disappears.

Proof of Theorem 7-6 This can be expressed by a *WGCSG* with position valuation s in the following way. First we state the length of the sentential form is exactly n, i.e., the length of the terminal word to be produced. The reason is that the length of the part of the sentential form where the counter operates is $m = \left\lceil \frac{k-1}{k} \cdot n \right\rceil$, as stated above, and the length of the part where the simulation takes place can be organized such that it equals $\left\lfloor \frac{1}{k} \cdot n \right\rfloor$, e.g., by compressing k symbols of the input for the LBA (the later terminal word) in every simulation symbol, except for the rightmost one: Here we encode $k' := k + n \bmod k$ symbols.

Proof of Theorem 7-7 Now the decompression works in the following way: First, the rightmost simulation symbol is transformed and decompressed into k' terminal symbols, where $k' - 1$ of the (leftmost) counter symbols disappear. Then the terminal symbols are passed through to the right. By valuating the terminal symbols greater than the maximum value of all other symbols, these rules are weakly growing related to s. The same procedure is done with the now-rightmost symbol, and then step by step with all the simulation symbols, where in this and the following steps $k' := k$.

Proof of Theorem 7-8 For a detailed and explicit construction of the grammar, see [Nie94].

Proof of Theorem 7 □

5.3-2 Note that the requirement $q < \frac{k-1}{k}$ is for a trick only: In order to exceed the time bound with the capacity of the counter, we strengthen it by the choice of l, the number of letters representing a digit (see Lemma 8, and regard additionally the limited space for the counter in the simulating sentential form). But such an l, fulfilling $\left\lceil \mathrm{w}^l - 1 \right\rceil^{\frac{k-1}{k}\cdot n\cdot\frac{1}{l}} \geq \mathrm{w}^{q\cdot n}$, does not exist for $\frac{k-1}{k} \leq q$, so there is a gap between Theorem 6 and Theorem 7. As the construction technique used to build a counter with exponential capacity does not lead further (see Lemma 7), closing this gap is a nontrivial problem.

6 Characterizing the Exponential Time Hierarchy of *CSL*

6-1 In the previous sections we looked at several position valuations, structuring and investigating them. Now we will gain an overview of the different classes of weakly growing context-sensitive languages related to different position valuations.

6-2 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 2). For every unsteady position valuation, we obtain the language class *CSL*, as we saw in Section 4. Allowing zero points for position valuations, we also obtain a characterization of *CSL*, as we mentioned before (see Section 2).

6-3 Note that in Definition 3 we gave an equivalent characterization for the property "unsteady," which does not hold for position valuations with zero points: A position valuation s without zero points is unsteady if and only if s has at least three valuated positions and s has a blip, i.e., a position j with $s(j)^2 \neq s(j-1)\cdot s(j+1)$. In Section 4, in fact, we used this equivalent characterization (e.g., in the values of the symbols of a swallower). Essentially, there is still something to show in cases where s has only two valuated positions or where s has no blip according to the definition given above, which is the case if s has only one nonzero point, and this is the first or the last valuated position, or if s has nonzero points at the first *and* the last position, separated at least by two zero points. On the other hand, the proof technique we use for these cases works for each position valuation with zero points. Therefore we formulate the next theorem and give a proof here.

Theorem 8 *Let s be a position valuation with zero points. Then:*

$$WGCSL_s = CSL$$

Proof of Theorem 8-1 **Proof of Theorem 8** It is sufficient to show $CSL \subseteq WGCSL_s$. And it is sufficient to show this claim for finite position valuations s. So let s be a finite position valuation with zero points.

Proof of Theorem 8-2 As the definition of $WGCSL_s$ implies that s is not constantly 0, Theorem 2 applies, and thus $WGCSL_s$ is closed under transposition. Theorem 4 can be adapted in the following sense: Let m be the greatest position valuated by s. Define $\overline{s}\colon \mathbb{N}^+ \to \mathbb{N}$ by: $\overline{s}(i) \stackrel{\text{def}}{=} s(m-i+1)$ for $i = 1, 2, \ldots, m$. Then $WGCSL_s = WGCSL_{\overline{s}}$.

Proof of Theorem 8-3 This especially means we can assume that there exists a zero point j with $s(j+1) = c \neq 0$. Regarding these two positions, the "growth factor" in fact is not defined, but it can be seen as arbitrarily large. Thus, we can construct a counter similar to the one in Section 5.2 that counts in an arbitrarily given number representation. (For $j \neq 1$ we add a left-hand context of length $j-1$ to the rules in the proof of Lemma 8, 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 7, where we choose $q = \frac{1}{2}$, $k = 2$, and b^2 instead of $\lceil \mathrm{w}(s) - 1 \rceil$ as the base of the number representation the counter uses. Thus, the linear bounded automaton can be simulated by an s-weakly growing context-sensitive grammar.

Proof of Theorem 8 □

6-4 Note that here we do not have a gap between the base of the time bounding function of the linear bounded automaton and the base of the number representation the counter uses (in contrary to Theorem 7). Therefore, here we do not need the trick of representing a digit by several symbols, and the requirement $q < \frac{k-1}{k}$ can be dropped, as the base of the number representation the counter uses can be chosen freely to ensure its capacity exceeds the time bound.

6-5 For every steady position valuation s, we saw in Section 3.2 that the corresponding class of weakly growing context-sensitive languages is characterized by the growth factor $\mathrm{w}(s)$, and in Theorems 6 and 7 the following was shown.

Proposition 3 *Let s be a steady nonconstant position valuation. Let $\mathrm{w} = \max\left\{\mathrm{w}(s), \frac{1}{\mathrm{w}(s)}\right\}$. For every $q \in \mathbb{Q}^+$, $0 < q < 1$, it holds that*

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

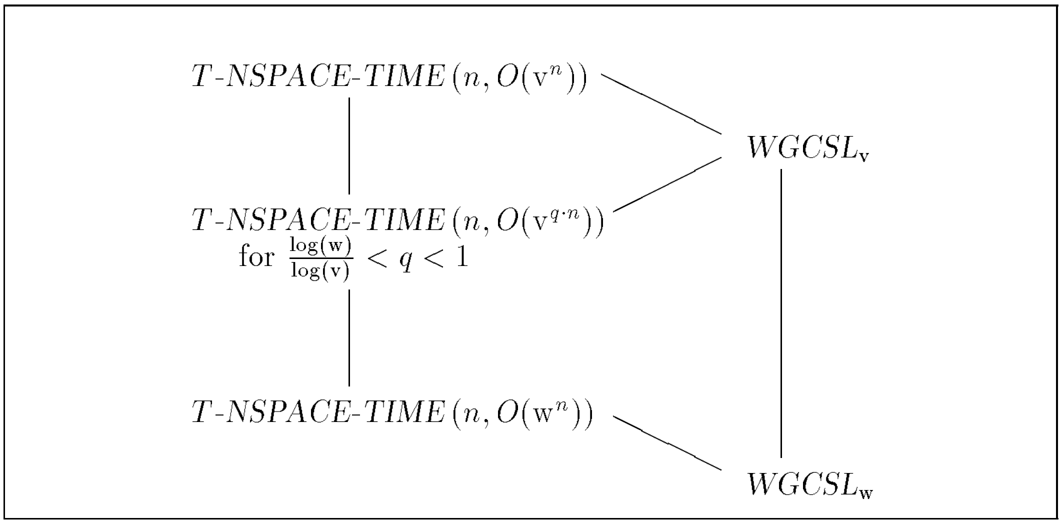

Figure 5: Relations for growth factors $\mathrm{v} > \mathrm{w} > 1$

Corollary 1 *For* $\mathrm{v}, \mathrm{w} \in \mathbb{N}^+$ *with* $\mathrm{w} < \mathrm{v}$ *it holds that* $WGCSL_{\mathrm{w}} \subseteq WGCSL_{\mathrm{v}}$.

6-6 The resulting relations for different growth factors are depicted in Figure 5.

6-7 As every polynomial time bound can be exceeded by every exponential time bound, the following can be concluded. To denote classes of sets accepted in polynomial time bounds, we use the notation pol for the set of all polynomials in a single variable n.

Corollary 2 *For every steady nonconstant position valuation s, it holds that*

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

6-8 We therefore have all the relations depicted in Figure 6, where we concentrated on some specific classes because of clarity. Thus the weakly 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 5.2 and the idea of construction in the proof of Theorem 7, the following can be shown.

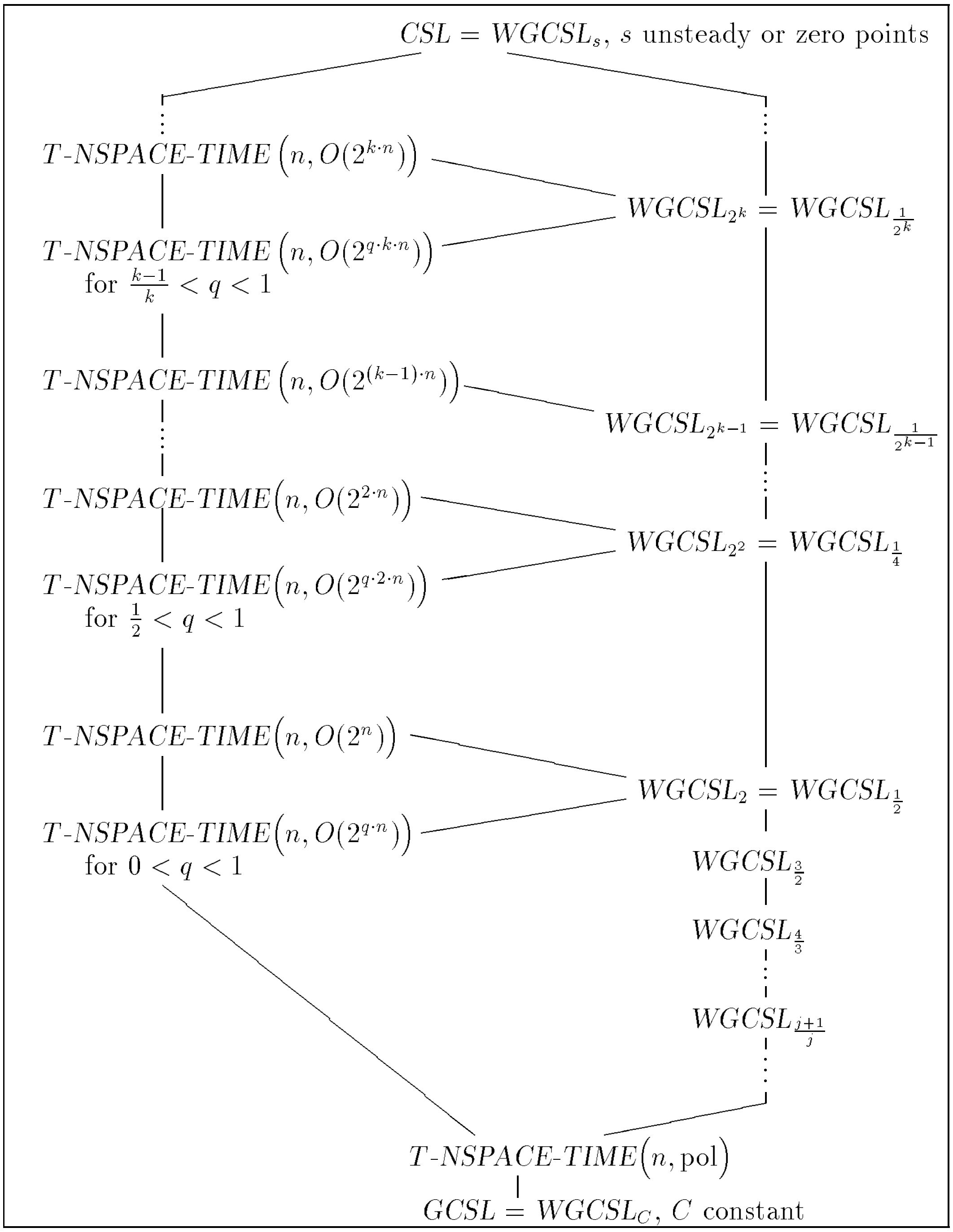

Figure 6: Exponential-time *CSL* and *WGCSL* hierarchies

Theorem 9 *Let s be a steady nonconstant position valuation. The closure of $WGCSL_s$ under inverse homomorphism is CSL. As well, the closure of $WGCSL_s$ under k-bounded homomorphism is CSL.*

Proof of Theorem 9-1 **Proof of Theorem 9** 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.

Proof of Theorem 9-2 Analogously to the grammar in the proof of Theorem 7, we construct an s-weakly growing context-sensitive grammar G that simulates M (without tape compression) and uses a counter just like the one we used in the proof mentioned above.

Proof of Theorem 9-3 For an input word of length n, we choose a start string of length m for the counter, where $m \geq \frac{l \cdot c}{\log(\mathrm{w}(s)-1)} \cdot n$ and $l \in \mathbb{N}$. Thus we reach at least the capacity

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

$$\geq t \cdot 2^{\log(\mathrm{w}(s)^l - 1) \cdot \left\lceil \frac{c}{\log(\mathrm{w}(s)^l - 1)} \cdot n \right\rceil} \geq t \cdot 2^{c \cdot n}$$

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

Proof of Theorem 9-4 Now we can expand each symbol $a \in \Sigma$ to $a\diamondsuit^{\left\lceil \frac{l \cdot c}{\log(\mathrm{w}(s)^l - 1)} \right\rceil}$, where $\diamondsuit \notin \Sigma$ is a new symbol. Together with the expansion, the counter disappears.

Proof of Theorem 9-5 We define a homomorphism $h\colon \Sigma^* \to (\Sigma \cup \{\diamondsuit\})^*$ by $h(a) \stackrel{\text{def}}{=} a\diamondsuit^{\left\lceil \frac{l \cdot c}{\log(\mathrm{w}(s)^l - 1)} \right\rceil}$ for every $a \in \Sigma$. Then $h(L) = L(G)$.

Proof of Theorem 9-6 To prove the claim for k-bounded homomorphism, define $h\colon (\Sigma \cup \{\diamondsuit\})^* \to \Sigma^*$ by $h(a) \stackrel{\text{def}}{=} a$ for every $a \in \Sigma$, $h(\diamondsuit) \stackrel{\text{def}}{=} \varepsilon$. Then h is k-bounded for $L(G)$ with $k \stackrel{\text{def}}{=} \left\lceil \frac{l \cdot c}{\log(\mathrm{w}(s)^l - 1)} \right\rceil$, and $h(L(G)) = L$.

Proof of Theorem 9 □

6-9 We point out that s is nonconstant, 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]).

6-10 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.

Theorem 10 *Let* $\mathrm{w} > 1$*, and 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 $\tilde{s}$*, for every grammar* $G \in WGCSG_{\tilde{s}}$*, there exists a grammar* $G' \in WGCSG_s$ *in Cremers normal form with* $L(G') = L(G)$*.*

d) There exists an unsteady position valuation $\tilde{s}$ *satisfying the condition of c) above.*

Additionally, a) implies:

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

and from this it can be concluded:

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

7 Conclusion

7-1 We have seen a characterization of *CSL* by weakly growing context-sensitive grammars related to an unsteady position valuation (Section 4) and a characterization of the exponential time hierarchy of *CSL* by classes of weakly growing context-sensitive languages related to steady position valuations (Section 6). Equally interesting, we consider the question of whether the exponential time hierarchy for linear bounded automata collapses.

7-2 Theorem 10 proves that the following problems are equivalent:

- For every weakly growing context-sensitive grammar related to an unsteady position valuation $\tilde{s}$, does there exist an equivalent grammar in a normal form of bounded order that is weakly growing context-sensitive related to a steady position valuation s?
- Is $WGCSL_s$ closed under inverse homomorphism for a steady position valuation s?

- Does the hierarchy of weakly growing context-sensitive language classes corresponding to different growth factors collapse down to a certain level? That is, do *CSL* and $WGCSL_s$ coincide for a steady position valuation s?
- Does the exponential time hierarchy for *CSL* collapse up to *CSL*? That is, do *CSL* and $T\text{-}NSPACE\text{-}TIME(n, O(\mathrm{w}^n))$ coincide for a certain $\mathrm{w} \in \mathbb{N}^+$?

If in the first question we allow an arbitrary position valuation instead of a steady one, we also must restrict ourselves to normal forms of order 2 (and thus end up with a steady position valuation again). The reason lies in the fact that for each context-sensitive grammar there exists an equivalent grammar of order 3 that is in $WGCSG_s$, 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 associate an additional encoded weight piece with every rule such that all the rules become weakly growing related to s with an appropriately defined symbol valuation f (this idea is similar to the one in the proof of Theorem 2). Then we add rules to pass those weights through to the right and a swallower (compare Section 4, case (iii)). Thus every step of the original derivation can be simulated by using the corresponding rule of the new grammar, passing the encoded weight through to the right, and swallowing it.

7-3 Because of Theorem 3 we know that asking for a normal form of bounded order in the case of a steady position valuation is equivalent to asking for a normal form of order 2. Remember that our notation of a characterization of one hierarchy by another means that the one collapses if and only if the other does. More precisely, we have a chain of inclusions between language classes where the members of that chain alternate between the two hierarchies.

7-4 In this context, it would be interesting to close the gap between Theorems 6 and 7, that is, to answer the question whether

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

8 Acknowledgments

8-1 We thank Hans-Jörg Burtschick for pointing out a former version of Example 2, and Steffen Reith for the idea of how to simplify the proof of Lemma 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, Theorem 1, and Theorem 3, also, the idea to build weakly growing grammars), and of her prethesis paper [Sch92], also extended (the separation of $WGCSG_s$ and CSG in Section 2). Further, we are very grateful to an anonymous LATIN'95 referee and the anonymous CJTCS referees of our paper. All of them did an extraordinary job. So did the managing editor and the copy editor of the *Chicago Journal of Theoretical Computer Science* who worked on our paper, Michael J. O'Donnell and Lisa Clark. Last, but not least, we thank Eric Allender for his constructive help.

Acknowledgement of support: Gerhard Buntrock's work was partly supported by a grant from the DFG (DFG-Projekt Wa 847/1-1).

References

[BL92] Gerhard Buntrock and Krzysztof Loryś. On growing context-sensitive languages. In Werner Kuich, editor, *Proceedings of the 19th International Colloquium on Automata, Languages and Programming*, volume 623 of *Lecture Notes in Computer Science*, pages 77–88, Berlin, Heidelberg, New York, 1992. Springer-Verlag.

[BL94] Gerhard Buntrock and Krzysztof Loryś. The variable membership problem: succinctness versus complexity. In Patrice Enjalbert, Ernst W. Mayr, and Klaus W. Wagner, editors, *Proceedings of the 11th Symposium on Theoretical Aspects of Computer Science*, volume 775 of *Lecture Notes in Computer Science*, pages 595–606, Berlin, Heidelberg, New York, 1994. Springer-Verlag.

[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 in Computer Science*, pages 313–324. Springer-Verlag, 1995. To appear in *Information and Computation*.

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

[Bör85] Egon Börger. *Berechenbarkeit, Komplexität, Logik*. Friedrich Vieweg & Sohn, Braunschweig, Wiesbaden, 1985. A third edition appeared 1995. English edition: [Bör89].

[Bör89] Egon Börger. *Computability, Complexity, Logic*, volume 128 of *Studies in Logic and the Foundations of Mathematics*. North Holland, Amsterdam, 1989. The German original appeared as: [Bör85].

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

[Bun96] Gerhard Buntrock. *Wachsende kontextsensitive Sprachen*. Habilitation thesis, Fakultät für Mathematik und Informatik, Universität Würzburg, July 1996. English title: Growing context-sensitive languages.

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

[Coo79] Stephen A. Cook. Deterministic CFL's are accepted simultaneously in polynomial time and LOG squared space. In M. Fischer, editor, *Eleventh Annual ACM Symposium on Theory of Computing*, pages 338–345. ACM, 1979. The results of this paper are contained in [vBCMV83].

[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 Sciences*, 33(3):456–472, 1986.

[GGH69] Seymour Ginsburg, Sheila Greibach, and John E. Hopcroft. Studies in abstract families of languages. *Memoirs of the American Mathematical Society*, 87:1–51, 1969.

[Gla64] Aleksey Vsevolodovich Gladkii. О сложности вывода в грамматиках непосредственно составляющих. АЛГЕБРА и ЛОГИКА, Цеминар, 3(5-6):29–44, 1964. English title: On the complexity of derivations in context-sensitive grammars; journal: *Algebra and Logics, Seminar*, Academy of Sciences of USSR, Novosibirsk.

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

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

[Jan79] Matthias Jantzen. A remark on increasing grammars. Unpublished note: the content of this note can be read in [Bun96], 1979.

[Kur64] S.-Y. Kuroda. Classes of languages and linear-bounded automata. *Information and Control*, 7(2):207–223, 1964.

[Nie92] Gundula Niemann. Die Länge wachsender Grammatiken, die aus quasi wachsenden erzeugt werden. Studienarbeit, Fakultät für Mathematik und Informatik, Universität Würzburg, December 1992. English title: The length of growing grammars that are produced out of quasi-growing ones.

[Nie94] Gundula Niemann. Charakterisierungen mit schwach wachsend kontextsensitiven Grammatiken. Diplomarbeit, Fakultät für Mathematik und Informatik, Universität Würzburg, March 1994. English title: Characterizations with weakly growing context-sensitive grammars.

[Nie96] Gundula Niemann. On weakly growing grammars. Technical Report 155, Fakultät für Mathematik und Informatik, Universität Würzburg, October 1996.

[Pau78] Wolfgang J. Paul. *Komplexitätstheorie.* Teubner Studienbücher Informatik. B. G. Teubner, Stuttgart, 1978. English title: *Complexity Theory.*

[Pip79] Nicholas Pippenger. On simultaneous resource bounds. In *Proceedings of the 20th Annual Symposium on Foundations of Computer Science*, Washington, D. C., 1979. Institute of Electrical and Electronics Engineers, IEEE Computer Society Press.

[Rei90] K. Rüdiger Reischuk. *Einführung in die Komplexitätstheorie.* Leitfäden und Monographien der Informatik. B. G. Teubner, Stuttgart, 1990.

[Ruz80] Walter L. Ruzzo. Tree-size bounded alternation. *Journal of Computer and System Sciences*, 21(2):218–235, 1980.

[Sal73] Arto Salomaa. *Formal Languages.* Academic Press, New York, 1973. German translation: [Sal78].

[Sal78] Arto Salomaa. *Formale Sprachen.* Springer-Verlag, Berlin, Heidelberg, New York, 1978. The English original appeared as [Sal73].

[Sch92] Carmen Schwenkel. Einige komplexitätstheoretische Untersuchungen kontextsensitiver Grammatiken mit Bewertungsfunktionen. Studienarbeit, Fakultät für Mathematik und Informatik, Universität Würzburg, June 1992. English title: Some complexity theoretical investigations of context-sensitive grammars with valuation functions.

[Sch93] Carmen Schwenkel. Eigenschaften schwach wachsend kontextsensitiver Sprachen. Diplomarbeit, Fakultät für Mathematik und Informatik, Universität Würzburg, August 1993. English title: Properties of weakly growing context-sensitive languages.

[Smu61] Raymond M. Smullyan. *Theory of Formal Systems*, volume 47 of *Annals of Mathematical Studies*. Princeton University Press, Princeton, New Jersey, 1961.

[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. *Proceedings of the London Mathematical Society*, 42(2144):230–265, 1936. A correction to this paper appeared in the same periodical, series 2, 43(2198):544–546, 1937.

[vBCMV83] Burchard von Braunmühl, Stephen Cook, Kurt Mehlhorn, and Rutger Verbeek. The recognition of deterministic CFLs in small time and space. *Information and Control*, 56(1/2):34–51, January/February 1983.

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