A parallel context-free derivation hierarchy ⋆

Klaus Reinhardt

Wilhelm-Schickhard Institut für Informatik, Universität Tübingen
Sand 13, D-72076 Tübingen, Germany
e-mail: reinhard@informatik.uni-tuebingen.de

Abstract. We consider the number of parallel derivation steps as complexity measure for context-free languages and show that a strict and dense hierarchy is obtained between logarithmic and linear (arbitrary) tree height. Furthermore we give a non-regular language with logarithmic tree height disproving a conjecture of Culik and Maurer. As a new method we use counter-representations, where the successor relation can be handled as the complement of context-free languages.

1 Introduction

Like in complexity theory, where different measures for costs for recognizing languages are used, Book [Boo71] considered the costs of derivations of words in languages by grammars. Correspondences between the number of derivation steps and Turing machine computations were established by Igarashi in [Iga77].

The height of a derivation-tree corresponds to the minimal number of parallel derivation steps using a context-free grammar, where we allow an arbitrary number of variables to be replaced simultaneously, and also to the space needed on the store of a push-down automaton. Heights of derivation trees generated by context-free grammars with regular parallel control languages were considered in [KPU79]. Brandenburg used the height of syntactical graphs[1] as complexity measure and showed equivalences to complexity classes [Bra81].

One reason, why context-free languages with sub-linear derivation tree height are interesting, is that their recognition can be parallelized efficiently as we will describe in Section 2. Unfortunately most context-free languages do not have this property. For some of them like $\{a^n b^n \mid n \in \mathcal{N}\}$ this was shown by Culik and Maurer [CM78] who also showed that regular languages have logarithmic tree height. Furthermore, it was conjectured [Bra81,CM78] that that context-free languages with logarithmic tree height are regular. We will disprove this conjecture in Section 5.

We will give a quite general criterion for languages not to have sub-linear derivation tree height by Theorem 2 in Section 4 using bounded languages.

On the other hand there are non-regular context-free languages which do not fit into this criterion [Boa97]: Consider the infinite word $w = baba^2ba^3b.....a^nb...$,

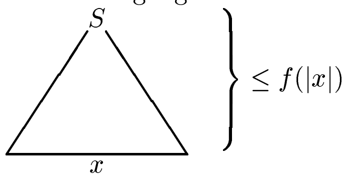
⋆ This research has been partially supported by the DFG Project La 618/3-2 KOMET.

[1] Generalizations of derivation trees for arbitrary grammars

which is a sequence of unary encodings of increasing numbers. Let the language L be the set of finite words which are not prefixes of w. L is context-free (the idea is that there exists a block of letters a with an exponent different from the number of letters b sitting at the left of the block). According to [Gol72] (see also [Gol76]) any bounded language in the full AFL generated by L is regular. Indeed this language has $\sqrt{n}$ tree height as we show in Section 5.

In section 6 we generalize this method to various kinds of representations of counters and define languages as sets of words which are not prefix of an infinite word being the sequence of such counter-representations (because of [AFG87] all these languages have to be inherently ambiguous). Furthermore we give a separation of tree height classes which means for every reasonable function f between logarithmic and linear, we can construct a context free language with $f(n)$ but no less tree height. Thus we have a strict and dense hierarchy.

We consider consider context-free languages, which can be generated by a context-free grammar such that every word in the language has a derivation tree of height $f(|x|)$:



A main difference to E0L and similar systems is that in a parallel derivation step not all variables have to be replaced (this explains why we need not write $\overset{\leq f(|x|)}{\underset{\|G}{\Longrightarrow}}$ instead of $\overset{f(|x|)}{\underset{\|G}{\Longrightarrow}}$ in the following definition[2]).

Definition 1. *Let $G = (V, \Sigma, P, S)$ be a context-free grammar. A parallel derivation step is defined by $\alpha_1 A_1 \alpha_2 A_2 ... \alpha_k A_k \alpha_{k+1} \underset{\|G}{\Longrightarrow} \alpha_1 r_1 \alpha_2 r_2 ... \alpha_k r_k \alpha_{k+1}$ with $A_i \to r_i \in P$ and $\alpha_i \in (V \cup \Sigma)^*$ for all $i \leq k \geq 0$. CFLth$(f(n)) := \{L \subseteq \Sigma^* | \exists G, L = L(G), \forall x \in L \;\; S \overset{f(|x|)}{\underset{\|G}{\Longrightarrow}} x\}$.*

Remark: It is easy to see that height can be compressed by a constant factor c by using $P' := \{A \to r \mid A \in V, A \overset{\leq c}{\underset{\|G}{\Longrightarrow}} r\}$. For that reason throughout the paper we need not care about additive and multiplicative constants. Observe that this destroys Chomsky normal form, which makes the O-notation for multiplicative constants necessary in this case.

Proposition 1. *[CM78] REG$\subseteq$CFLth$(\log n)$.*

Clearly only finite languages can be generated with sub-logarithmic tree height.

---

[2] $\overset{f(|x|)}{\underset{\|G}{\Longrightarrow}}$ denotes $f(|x|)$ derivation steps with $\underset{\|G}{\Longrightarrow}$ in sequence

2 The connection to parallel recognition

With the method in [GR88], context-free languages can be recognized by a CRCW-PRAM in $O(log(n))$ steps. But this method needs n^6 processors, which makes it very inefficient. On the other hand the CYK-algorithm [Kas65] allows easily a parallelization on a CRCW-PRAM in linear time with n^3 processors[3]. This idea can be used to recognize languages in CFLth$(f(n))$ in time $O(f(n))$ with n^3 processors[4]. This means that the derivation tree height corresponds to the running time of the following parallel algorithm. For every production $A \to BC$ and every infix uv in the input word, a processor writes a 1 to the position for $A \stackrel{*}{\Longrightarrow} uv$ if the positions for $B \stackrel{*}{\Longrightarrow} u$ and $C \stackrel{*}{\Longrightarrow} v$ have a 1. If $w \in L \in$CFLth$(f(n))$, then $S \stackrel{*}{\Longrightarrow} w$ will have a 1 after $O(f(n))$ steps.

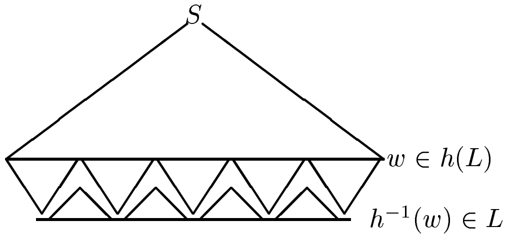
3 Closure properties

Theorem 1. *CFLth(f) is an AFL and closed under reversal and substitution by other languages in CFLth(f) not containing λ.*

Proof. For the closure properties union, product, $*$, intersection with regular languages, reversal and substitution by other languages in CFLth(f) not containing λ we can just take the standard construction in [Ber79] for context-free languages and observe that the tree-height is only increased by an constant factor.

To build the grammar for $h(L)$ for $L \in$CFLth(f) and a non-erasing homomorphism h, we simply replace every terminal letter a by $h(a)$. (An erasing homomorphism h could make the word significantly shorter.)

For the inverse homomorphism, a slight modification to the proof in [Ber79][page 31] is necessary. The problem is that there may be letters z with $h(z) = \lambda$ occurring in $h^{-1}(w)$, which are produced in [Ber79][page 31] by a linear part of the grammar, which could increase the tree height to linear. To prevent this, we need to insert the production $\omega \to \omega\omega$ to P' in [Ber79][page 31] in order to be able to produce letters z with $h(z) = \lambda$ in binary trees which consume only logarithmic tree height. Furthermore we observe that the erasing homomorphism used for the construction of the inverse homomorphism in [Ber79][page 31] can decrease the length of the word only by a constant factor.



Open problem: Is CFLth(f) closed under erasing homomorphisms?

[3] A more clever algorithm works on a CROW-PRAM in linear time with n^2 processors.

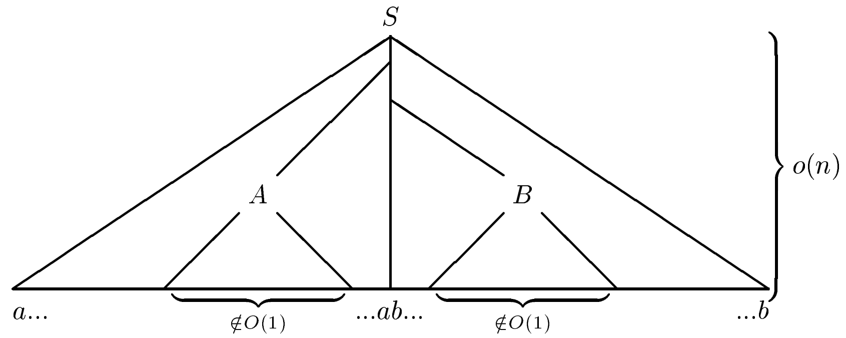
[4] In most cases an improvement to n^2 processors is possible.

4 Bounded languages have linear derivation tree height

Definition 2. *[Gin66] A language L is* bounded *if $L \subseteq w_1^* w_2^* ... w_m^*$ for $w_1, ..., w_m \in \Sigma^*$, $m \geq 1$.*

Theorem 2. *No bounded non regular language is in CFLth$(o(n))$.*

Proof. We first start to proof the theorem for 2-bounded[5] languages. Since CFLth$(o(n))$ is an AFL we may assume w.l.o.g. $L \subset a^* b^*$. Let G be a context-free grammar for L. Consider all variables $A, B, ...$ which are produced on the left or right side in the path from S to the border between a's and b's. The words producible by such a variable are in a sub-language $\subseteq a^* \cup b^*$ which is regular. Let s be the product of of all lengths of minimal loops of the minimal deterministic automaton for all this languages. If L is not regular then there must be $k, k' \leq s$ such that $L' = L \cap a^k (a^s)^* b^{k'} (b^s)^*$ is not regular. Thus the semi-linear Parik-image of L' must have a period incrementing a's and b's simultaneously. If L and thus $L' \in$CFLth$(o(n))$, there must be infinite sub-languages on both sides. Then there is a word $w \in L'$ with an infinite sub-language on both sides, which means $(a^s)^* w (b^s)^* \subseteq L'$, thus L' is regular in contradiction to the assumption.



Now consider an m-bounded language $L \subset a_1^* a_2^* ... a_m^*$ and construct L' analogously. If the semi-linear Parik-image of L' has a period incrementing all a_i's simultaneously and if L and thus $L' \in$CFLth$(o(n))$, there must be a word in L' with an infinite sub-language in every block, which means $(a_1^s)^* a_1^{c_1} (a_2^s)^* a_2^{c_2} ... (a_m^s)^* a_m^{c_m} \subseteq L'$. Otherwise we have $(a_1^s)^* a_1^{c_1} (a_2^s)^* a_2^{c_2} ... (a_m^s)^* a_m^{c_m} \cap L' = \emptyset$. In both cases the intersection with one of the languages $a_1^* ... a_i^c ... a_m^*$ must have the non-regular part, which means that the AFL generated by L and thus L' contains a $m-1$-bounded non-regular language, which leads to a contradiction by induction.

Corollary 1. *Every context-free language L generating an AFL containing a bounded non regular language is not in CFLth$(o(n))$.*

If we would know that every AFL generated by a non-regular unambiguous context-free language contains a non-regular bounded language, we could proof the following:

[5] $L \subseteq w_1^* w_2^*$ for $w_1, w_2 \in \Sigma^*$

Conjecture Every non-regular unambiguous context-free non-regular language L is not in CFLth$(o(n))$.

For a more general separation result as Theorem 2 we need the following definition:

Definition 3. *Let Σ be an alphabet and $a \neq b$ two symbols. A language L is $f(n)$* tail-bounded *if for every $w \in \Sigma^*$ and $n \in \mathcal{N}$ with $wab^n \notin L$ it holds $n = f(|wa|)$.*

Lemma 1. *No $f(n)$ tail-bounded non regular language is in CFLth$(o(f(n)))$*

Proof. Assume in the contrary G to be a context-free grammar w.l.o.g. for a language $L \subseteq \Sigma^* ab^*$. Consider all variables which are produced on the right side in the path from S to a in the derivation tree. The words producible by such a variable form regular sub-languages in b^*. Let s be the product of of all lengths of loops of the minimal deterministic automaton for all this languages. If L is not regular then there must be a $k \leq s$ such that $L' = L \cap \Sigma^* ab^k(b^s)^*$ is not regular. If L and thus $L' \in$CFLth$(o(n))$, there must be an infinite sub-language of b's occur on every but finitely many paths. Thus there are only finitely many $wab^n \notin L'$ (having $wab^{n-s} \in L'$ produced by one of the finitely many paths). Thus L is co-finite in contradiction to the assumption.

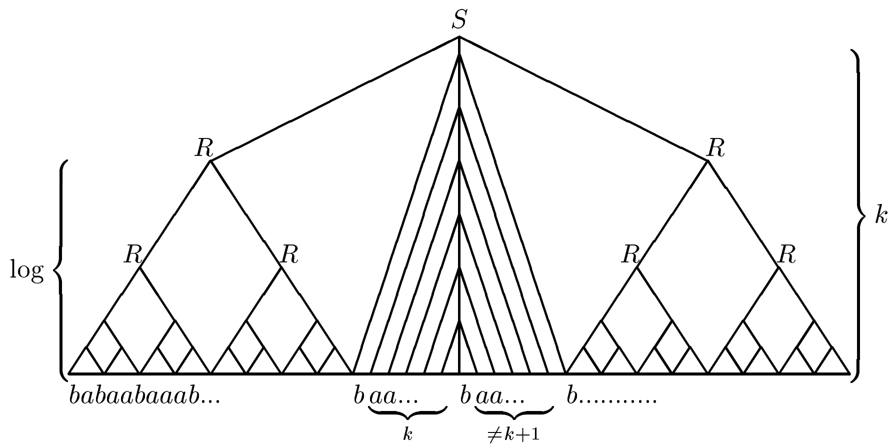
5 Parallelizable non regular languages

There are non-regular languages in CFLth$(o(n))$:

Example [Boa97]: Let L be the set of non-prefixes of the infinite word $w = baba^2ba^3b.....a^nb....$ Then $L \in$CFLth$(\sqrt{n})$ by the following grammar:

$\{S \to RABabR|RABab|RbBAaR|baAR|bbR|babbR|babaaAR|AR,$
$A \to AA|a, R \to RR|a|b, B \to aBa|b\}$

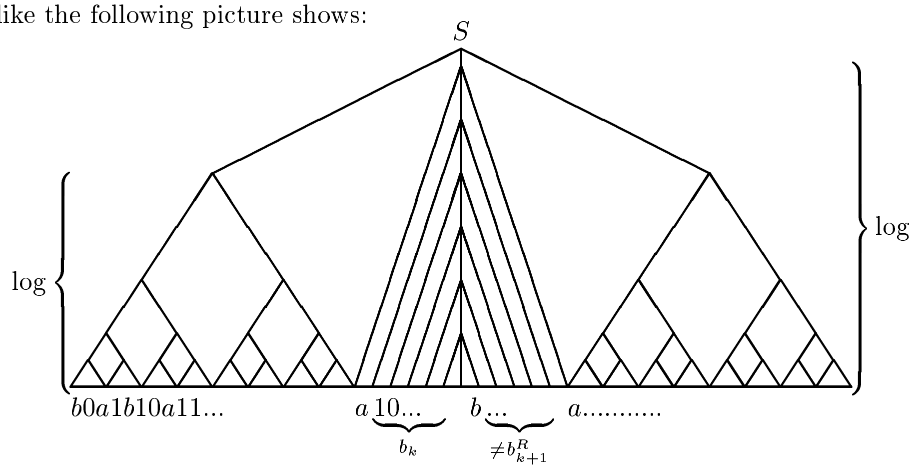
The variable R produces any string with logarithmic tree height. The variable B produces $\{a^nba^n \mid n \in \mathcal{N}\}$ thus $ABab$ produces $\{a^nba^mb \mid n \geq m \in \mathcal{N}\}$, which can not occur in w and $bBAa$ produces $\{ba^nba^m \mid m > n+1 \in \mathcal{N}\}$, which also can not occur in w. Thus a word in L can be produced making use of the first 'error' in respect to w, which is illustrated in the following picture:



For a word in L we consider the derivation using the first position where a block has length k and the following block has not the length $k+1$. Thus we can estimate the length n of the word by $n \geq \sum_{i=1}^{k} i$ which means $k \in O(\sqrt{n})$.

We now improve this by constructing an example with the smallest possible derivation tree height and hereby disprove the conjecture in [Bra81,CM78]. There are non-regular languages in CFLth(log):

Example: Let L be the set of non-prefixes of the infinite word $w = b0a1b10a11b100a101b110a111b...a111100001b100010000a...bb_kab_{k+1}^Rb...$, where $b_k \in \{0,1\}^*$ is the binary representation of k. It holds $L \in$CFLth(log(n)) by a grammar which produces words containing an 'error'-part showing that the word is not prefix of w; this may be some part ab_kbv^Ra or bb_k^Ravb with $v \neq b_{k+1}$ or some other syntactic 'error'. Since the length of the binary representations grow only logarithmically until an 'error' occurs, the tree-height is also logarithmic like the following picture shows:



6 Strictness and denseness of the hierarchy

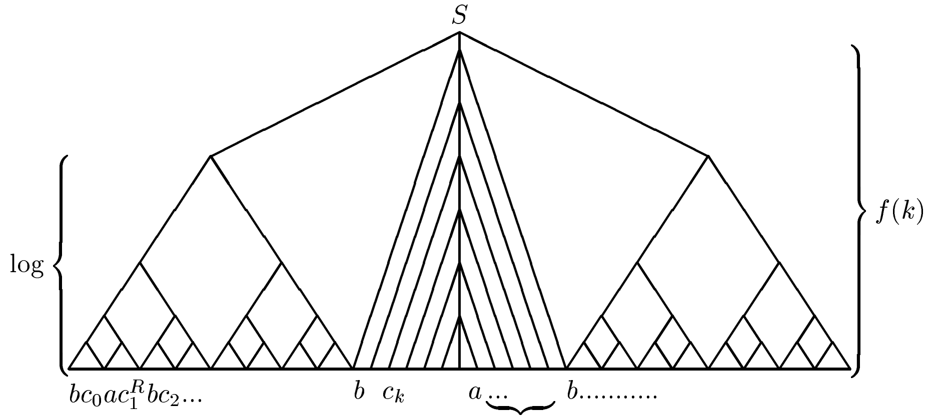
As an important tool we use a generalization of the representation of a counter to define context-free complement constructible functions.

Definition 4. *A function $f : \mathcal{N} \to \mathcal{N}$ is called* context-free complement constructible *(ccc), if there is an infinite sequence $c_0, c_1, ...$ of words over an alphabet Σ_f with $|c_k| = f(k)$, such that the complement of $\{c_k\$c_{k+1}^R | k \in \mathcal{N}\}$ is context-free.*

The following property of ccc functions will help us to establish the strictness and denseness of the hierarchy:

Theorem 3. *For all functions $g : \mathcal{N} \to \mathcal{N}$ with* log $\leq g \leq$ *lin such that there is a ccc function $f : \mathcal{N} \to \mathcal{N}$ with $g(\sum_{i=1}^{k} f(i)) = f(k)$ for all k, there are languages in CFLth($g(n)$)$\setminus$CFLth($o(g(n))$).*

Proof. Let L be the set of non-prefixes of the infinite word $u = bc_0ac_1^Rbc_2ac_3^R...ac_{n-1}^Rbc_na...$, with (c_i) being the context-free complement construction for f with $a, b \notin \Sigma_f$. Then $L \in$CFLth$(g(n))$ by a grammar which works like the picture shows:



The grammar produces at the beginning and at the end some rest with a binary tree and in the middle it produces either $b\{c_kaw^R | k \in \mathcal{N}, c_{k+1} \neq w \in \Sigma^*\}b$ according to Definition 4 or $a\{c_k^Rbw | k \in \mathcal{N}, c_{k+1} \neq w \in \Sigma^*\}a$ analogously. We may assume w.l.o.g. that the first part of the word until c_k is a prefix of u since we could choose an earlier c_k otherwise. Therefor the length of the word is $n \geq \sum\limits_{i=1}^{k} f(i)$ thus the height $g(n) = f(k)$ is sufficient.

Let $L' := Lb^* \cup \overline{(\Sigma_f \cup \{a,b\})^*ab^*} \cup (\Sigma_f \cup \{a,b\})^*b\{vab^n \mid v \in \Sigma_f^*, n \neq |v|\}$. Every word $wab^n \in (\Sigma_f \cup \{a,b\})^*ab^*$ is in Lb^* unless wa is a prefix of u; in this case wab^n is in L' iff the length of the last counter representation in w is not equal to n, which can be tested within tree-height $g(|wa|)$ thus $L' \in$CFLth$(g(n))$. The only possibility for wab^n not to be in L' is $n = g'(|wa|)$ for a $g' \in \Theta(g)$, which means L' is $g'(n)$ tail-bounded. Because of Lemma 1, L' can not be in CFLth$(o(g'(n)))$ =CFLth$(o(g(n)))$.

Lemma 2. *1, 2^n and* log *are ccc. If f and g are ccc, then also $f+g$ and $f*g$ and the functions e, d with $e(n) = \sum_{i=1}^n f(i)$ and $d^{-1}(n) = \sum_{i=1}^n f^{-1}(i)$.*

Proof. Let $c_{1,i} = 1$, $c_{2^n,i} = 1^{2^n}$ and $c_{\log,i} = bin(i)$. Assume w.l.o.g. $\Sigma_f \cap \Sigma_g = \emptyset$ and $d \notin \Sigma_f \cup \Sigma_g$.

Let $\Sigma_{f+g} = \Sigma_f \cup \Sigma_g$, $c_{f+g,i} := c_{f,i}c_{g,i}$ and the complement of $\{c_{f+g,k}\$c_{f+g,k+1}^R | n \in \mathcal{N}\}$ is $\{c_{f+g,k}aw^R | k \in \mathcal{N}, c_{f+g,k+1} \neq w \in \Sigma^*\}b$ and can be generated by generating at least one counter wrong.

Let $\Sigma_{f*g} := \Sigma_f \times \Sigma_g$ with the canonical projections π_1 and π_2 and $c_{f*g,i} := w$ with $\pi_1(w) = c_{f,i}^{|c_{g,i}|}$ and $\pi_2(w) = h(c_{g,i})$ with $h(a) := a^{|c_{f,i}|}$ and the complement of $\{c_{f*g,k}\$c_{f*g,k+1}^R | n \in \mathcal{N}\}$ can be generated analogously.

Let $\Sigma_e = \Sigma_d = \Sigma_f \cup \{d\}$ $c_{e,i} := d^{\sum_{j=1}^{i-1} |c_{f,j}|}c_{f,i}$ and the complement of $\{c_{e,k}\$c_{e,k+1}^R | n \in \mathcal{N}\}$ can be generated by either generating a wrong $c_{f,i+1}$ or not adding $|c_{f,i}|$ correctly to the number of d's.

The last transformation enables building roots of functions (log is fix-point of the transformation); the idea to construct the counter is to keep a size $d(i)$ and simulate the counter $c_{f,j}$ for f inside by $c_{d,i} = d^k c_{f,j}$, where the rest is filled with d's. This is done until the size is not anymore sufficient, then the space is incremented an the simulated counter for f is started from 0 again. Formally this is $c_{d,i+1} := \begin{cases} d^{|c_{d,i}|+1-|c_{f,0}|}c_{f,0} \text{ if } f(j+1) > d(i) \\ d^{k+|c_{f,j}|-|c_{f,j+1}|}c_{f,j+1} \text{ else} \end{cases}$.

Lemma 2 shows that all polynomials with rational coefficient and its multiplications with poly-logarithms are ccc. For f ranging from logarithmic to very big polynomials, the function g with $g(\sum_{i=1}^k f(i)) = f(k)$ ranges from logarithmic towards linear. If $g(n)$ is a polynomial $n^{p/q}$ with $q > p$ then a ccc function $f(n) \in \Theta(n^{p/(q-p)})$ fulfills the condition $g(\sum_{i=1}^k f(i)) = f(k)$ and Theorem 3 can be applied. If $g(n)$ is a polylogarithmic function $\log^{p/q} n$, we do not know how to find an appropriate function f to apply Theorem 3 directly. However, if we set $f(n) = \log^{p/q} n$, the obtained function $g(n)$ can be estimated by $\log^{(p/q)-\varepsilon} n < g(n) < \log^{p/q} n$ for every $\varepsilon > 0$ thus Theorem 3 can separate any CFLth($\log^r$) from CFLth($\log^{r'}$) with $r < r'$. Since the ccc functions f are dense, also the obtained functions g are dense between between log and lin; therefor we get:

Conclusion: The parallel context-free derivation hierarchy is strict and dense between CFLth(log) and CFLth(lin) =CFL.

Remark: It is easy to see that any nontrivial decision problem, where it has to be decided for a given grammar for a language L in CFLth($g'(n)$) whether L is in CFLth($g(n)$), is undecidable (provided corresponding ccc function f, f' exist) since counter representations could also contain configurations of Turing machines and the behavior of the length can change if an accepting configuration is reached.

Acknowledgment: We thank H. Fernau and K.-J. Lange for helpful remarks.

References

[AFG87] J.-M. Autebert, P. Flajolet, and J. Gabarró. Prefixes of infinite words and ambiguous context-free languages. *Information Processing Letters*, 25:211–216, 1987.

[Ber79] J. Berstel. *Transductions and context-free languages.* Teubner Studienbücher, Stuttgart, 1979.

[Boa97] Luc Boasson. personal communication. 1997.

[Boo71] Ronald V. Book. Time-bounded grammars and their languages. *Journal of Computer and System Sciences*, 5(4):397–429, August 1971.

[Bra81] F. J. Brandenburg. On the height of syntactical graphs. In Peter Deussen, editor, *Proceedings of the 5th GI-Conference on Theoretical Computer Science*, volume 104 of *LNCS*, pages 13–21, Karlsruhe, FRG, March 1981. Springer.

[CM78] K. Culik II and H.A. Maurer. On the height of derivation trees. *Forschungsbericht Nr. 18, Inst. für Informationsverarbeitung TU Graz*, 1978.

[Gin66] S. Ginsburg. *The Mathematical Theory of Context-Free Languages.* McGraw-Hill, New York, 1966.

[Gol72] Jonathan Goldstine. Substitution and bounded languages. *Journal of Computer and System Sciences*, 6(1):9–29, February 1972.

[Gol76] J. Goldstine. Bounded AFLs. *Journal of Computer and System Sciences*, 12(3):399–419, June 1976.

[GR88] A. Gibbons and W. Rytter. *Efficient Parallel Algorithms.* Cambridge University Press, 1988.

[Iga77] Y. Igarashi. General properties of derivational complexity. *Acta Inf.*, 8(3):267–283, 1977.

[Kas65] T. Kasami. An efficient recognition and syntax algorithm for context-free languages. Scientific Report AFCRL-65-758, Air Force Cambridge Research Laboratory, Bedford MA, 1965.

[KPU79] Werner Kuich, Helmut Prodinger, and Friedrich J. Urbanek. On the height of derivation trees. In Hermann A. Maurer, editor, *Automata, Languages and Programming, 6th Colloquium*, volume 71 of *Lecture Notes in Computer Science*, pages 370–384, Graz, Austria, 16–20 July 1979. Springer-Verlag.