The Contextsensitivity Bounds of Contextsensitive Grammars and Languages

Franz-Josef Brandenburg

Institut für Informatik der Universität Bonn

Wegelerstr. 6, 53oo Bonn, West Germany

ABSTRACT:

In this paper we study the derivational complexity of contextsensitive grammars and languages by placing bounds on their contextsensitivity. The contextsensitivity of a grammar is defined on its derivations, and it is determined by the maximal length of the strings of ancestors of any symbol occurring at any position of the derived strings. A total recursive function f bounds the (right-) contextsensitivity function of grammar G, if for every terminal string x of length n generated by G there is a (right-canonical) derivation from S to x in G whose contextsensitivity is less than or equal to $f(n)$.

We investigate lower and upper bounding functions for the right-contextsensitivity functions of contextsensitive grammars and languages and study the families of context-sensitive languages with right-contextsensitivity functions bounded by some particular sublinear functions f.

INTRODUCTION:

In automata theory and formal language theory much recent activity is concerned with the computational complexity of formal languages measured by grammars or machines with a bounded amount of properties, in particular with time and tape bounds.

Time and tape are complexity measures which bound the behaviour of machines or grammars as a whole and do not give insight into the intrinsic structures of computations and derivations. They tell nothing about the efforts a machine or a grammar must do at any single position of the input string or the string generated.

In terms of machines, the crossing sequences founded by Hennie (1965) and the active reversals studied by Wechsung (1975) and Chytil (1976) bound the activities of a Turing machine on any square of the input tape. For context-free grammars, the complexity of derivations may be measured by the number of occurrences of nonterminals at any step of the derivations, and we obtain nonterminal bounded and derivation bounded grammars and languages. See Ginsburg, Spanier (1968). Finally, in the case of contextsensitive grammars the contextsensitivity will shed light on the derivational complexity of grammars and languages. For any occurrence of a symbol b in a terminal string x generated by grammar G, the contextsensitivity states an upper bound of the number of ancestors of that position at any step of a derivation from S to x, and thus it bounds the length of the "global context" on which the distinguished b depends.

In particular, we show that every context-free grammar has a (right-) contextsensitivity which is bounded by 1, and that the linear functions bound the (right-) contextsensitivity functions of contextsensitive grammars and languages. Furthermore, log log n is a strict lower bounding function for the right-contextsensitivity of grammars in order to generate non-context-free languages, and $\{w_1 c w_2 c \overleftarrow{w}_1 c \overleftarrow{w}_2 / w_1, w_2 \in \{a,b\}^*, |w_1| = |w_2|\}$ is the "hardest" contextsensitive language in the sense that its right-contextsensitivity is at least a linear function. These proofs are based on similar results for f(n)-tape bounded one-way auxiliary pushdown automata which simulate grammars with f(n)-bounded right-contextsensitivity.

PRELIMINARIES:

Notice that throughout this paper we restrict ourselves to contextsensitive grammars. The case of type 0 grammars and languages will be considered in a forthcoming paper.

Our subsequent studies of the interactions of individual applications of productions in a local manner and the global consequences thereof require a detailed and unambiguous description of the positions of substrings in a string and of the rewriting processes in derivations.

Suppose that x is a nonempty substring of a string w. Now, the substring property alone does not completely specify the relations between x and w, since x can occur in w several times. If w is a nonempty string and k,l are integers such that $o \leq k < l \leq |w|$, where $|w|$ denotes the length of w, then w(k,l) or $w^{k,l}$ denote the nonempty substring of w which begins with the k+1-st symbol of w and ends with the l-th symbol. Thus $w = w(o,k)\ w(k,l)\ w(l,|w|)$ with $|w(o,k)| = k$ and $|w(o,k)\ w(k,l)| = l$. If $k \geq l$, then $w(k,l) = e$, the empty string. If w(i,j) and w(k,l) are nonempty substrings of w and $j \leq k$, then w(i,j) is said to occur to the left of w(j,k), and w(j,k) is said to occur to the right of w(i,j), and w(i,j) and w(k,l) overlap, if $k < j$ and $i < l$.

Definition 1:

If G = (N,T,S,P) is a grammar, then a derivation from Q_1 to Q_m in G is a sequence of m triples $D = [(Q_i, \alpha_i \to \beta_i, t_i), i=1,\dots,m]$, where $Q_1, Q_2, \dots, Q_m$ is the sequence of strings derived by D. For each i, $1 \leq i \leq m-1$, $\alpha_i \to \beta_i \in P$ is the production applied in the i-th step of D to Q_i to obtain Q_{i+1}, and t_i determines the position of the application of $\alpha_i \to \beta_i$ such that $\alpha_i = Q_i(t_i, t_i+|\alpha_i|)$ and $\beta_i = Q_{i+1}(t_i, t_i+|\beta_i|)$. $\alpha_m \to \beta_m$ and t_m are left unspecified.

Definition 2:

Let $D = [(Q_i, \alpha_i \to \beta_i, t_i), i=1,\ldots,m]$ be a derivation in grammar G. For each i, $1 \leqq i \leqq m-2$, step i occurs to the left of step i+1, if $t_i+|\beta_i| \leqq t_{i+1}$. Step i occurs to the right of step i+1, if $t_{i+1}+|\alpha_{i+1}| \leqq t_i$, and step i is connected to step i+1, if $t_i < t_{i+1}+|\alpha_{i+1}|$ and $t_{i+1} < t_i+|\beta_i|$.

It can be easily seen that these three cases are mutually exclusive and exhaust all possibilities.

Definition 3:

D is a right-canonical derivation, if each step is either connected to the consecutive step or occurs to the right of it.

It can be shown that for any derivation D there is a uniquely determined right-canonical derivation which is obtained from D by trivial rearrangements of the productions applied in the steps of D. The rearrangements are based on the possibility of interchanging the order of the applications of productions which do not interfere with each other, i.e. which are applied to the left or to the right of each other. Two derivations are called similar, if they can be obtained from each other by rearrangements of applications of productions. In this sense, the right-canonical derivations are the representatives of the classes of similar derivations. For details and proofs see Griffiths (1968), Book (1969) or Buttelmann (1975).

ANCESTORS AND CONTEXTSENSITIVITY:

We now define the strings of ancestors of substrings occurring in the strings generated by a contextsensitive grammar G. The notion of the strings of ancestors of single symbols is fundamental for the definition of the contextsensitivity of derivations, grammars, and languages, which is the main notion of this paper and is introduced in this section. Notice that our definition of ancestors differs basically from that of Book (1969).

Definition 4:

If $D = [(Q_i, \alpha_i \to \beta_i, t_i), i=1,\ldots,m]$ is a derivation from Q_1 to Q_m in grammar G, then for any nonempty substring $Q_m(k,l)$ of Q_m and for each i, $1 \leqq i \leqq m$, we define the strings of ancestors of $Q_m(k,l)$, $A_i(k,l)$ by:

$A_m(k,l) = Q_m(k_m,l_m)$, where $k_m = k$ and $l_m = l$.

If $i < m$, then $A_i(k,l) = Q_i(k_i,l_i)$, where

$$k_i = \begin{cases} k_{i+1} & \text{if } k_{i+1} \leqq t_i \\ k_{i+1}-|\beta_i|+|\alpha_i|, & \text{if } t_i+|\beta_i| \leqq k_{i+1} \\ t_i, & \text{if } t_i < k_{i+1} < t_i+|\beta_i|, \end{cases}$$

$$\text{and} \quad l_i = \begin{cases} l_{i+1} & \text{if } l_{i+1} \leq t_i \\ l_{i+1}-|\beta_i|+|\alpha_i|, & \text{if } t_i+|\beta_i| \leq l_{i+1} \\ t_i+|\alpha_i|, & \text{if } t_i < l_{i+1} < t_i+|\beta_i|. \end{cases}$$

It can be easily seen that these cases are mutually exclusive and exhaust all possibilities.

Informally, if $A_{i+1}(k,l)$ and the right side of the production $\alpha_i \to \beta_i$ which is applied in the i-th step of D do not overlap, then the string of ancestors itself remains unchanged although its boundaries may be altered; otherwise $A_i(k,l)$ is obtained from $A_{i+1}(k,l)$ by substituting β_i or that part of β_i which overlaps with $A_{i+1}(k,l)$ by α_i. Hence, $A_i(k,l)$ only depends on those applications of productions of D which overlap with the strings of ancestors $A_j(k,l)$ für $j = i+1,\ldots,m$.

If G is a context-free grammar, then in the tree representing a derivation from S to x in G, the strings of ancestors of a single symbol occurring in x are the labels of the nodes on the path from the root S to the leaf which corresponds to the designated symbol, and these labels may occur with a certain multiplicity, determined by the derivation.

The importance of the strings of ancestors for the structure of derivations is made clear now.

Theorem 1:

If $D = [(Q_i,\ \alpha_i \to \beta_i,\ t_i),\ i=1,\ldots,m]$ is a derivation from Q_1 to Q_m in grammar G, then for any nonempty substring $Q_m(k,l)$ of Q_m, the sequence of strings of ancestors $A_1(k,l) = Q_1(k_1,l_1),\ldots,\ A_m(k,l) = Q_m(k_m,l_m) = Q_m(k,l)$ induces a splitting of D into three subderivations D_{kl}, D'_{kl} and D''_{kl}. D_{kl} is a derivation from $A_1(k,l)$ to $\gamma\ A_m(k,l)\ \delta$ for some uniquely determined strings γ,δ, which is built along the sequence of strings of ancestors. D'_{kl} is a derivation from $Q_1(o,k_1)\gamma$ to $Q_m(o,k_m)$, which occurs to the left of the strings of ancestors and D''_{kl} is a derivation from $\delta\ Q_1(l_1,|Q_1|)$ to $Q_m(l_m,|Q_m|)$, which occurs to the right of the strings of ancestors.

Proof:

Notice that for each Q_{i+1}, $1 \leq i \leq m-1$, there are two distinguished substrings of Q_{i+1} namely the string of ancestors $A_{i+1}(k,l) = Q_{i+1}(k_{i+1},l_{i+1})$ and the right side of the production applied in the i-th step of D, β_i, which is determined by $Q_{i+1}(t_i,\ t_i+|\beta_i|)$.

Now, there is a uniquely determined partition of the set $\{1,\ldots,m-1\}$ into three strictly ordered sets of indices $F = \{f_1,f_2,\ldots,f_n\}$, $H = \{h_1,\ldots,h_r\}$ and $J = \{j_1,\ldots,j_s\}$, such that for each $f_i \in F$, β_{f_i} and $A_{f_i+1}(k,l)$ overlap, and for each $h_i \in H$ and each $j_i \in J$, β_{h_i} occurs to the left of $A_{h_i+1}(k,l)$ and β_{j_i} occurs to the right of $A_{j_i+1}(k,l)$, respectively.

Additionally, let $f_{n+1} = h_{r+1} = j_{s+1} = m$.

By the definition of the strings of ancestors, for each $f_i \in F$, $k_{f_i} \leq t_{f_i} \leq t_{f_i}+|\alpha_{f_i}| \leq l_{f_i}$. Hence α_{f_i} is a substring of $A_{f_i}(k,l) =: A_{f_i}^{k,l}$ which is determined by $\alpha_{f_i} = A_{f_i}^{k,l}\ (t_{f_i}-k_{f_i},\ t_{f_i}-k_{f_i}+|\alpha_{f_i}|)$, and so the production

$\alpha_{f_i} \to \beta_{f_i}$ can be applied to $A_{f_i}(k,l)$ in a similar way as to Q_{f_i}. The application of $\alpha_{f_i} \to \beta_{f_i}$ to $A_{f_i}(k,l)$ at the position $t_{f_i} - k_{f_i}$ yields the string $\beta'_{f_i} A_{f_i+1}(k,l) \beta''_{f_i}$, where β'_{f_i} and β''_{f_i} occur as the "left and right overflow" of $A_{f_{i+1}}(k,l)$ and are specified as substrings of Q_{f_i+1} by $\beta'_{f_i} = Q_{f_i+1}(k_{f_i}, k_{f_i+1})$ and $\beta''_{f_i} = Q_{f_i+1}(l_{f_i+1}, |\beta_{f_i}| - |\alpha_{f_i}| + l_{f_i})$.

Now define $\beta'_m = \beta''_m = e$, $\gamma = \beta'_{f_1} \ldots \beta'_{f_n}$ and $\delta = \beta''_{f_n} \ldots \beta''_{f_1}$.

Then $D_{kl} = [(\beta'_{f_1} \ldots \beta'_{f_{i-1}} A_{f_i}(k,l) \beta''_{f_{i-1}} \ldots \beta''_{f_1}, \ \alpha_{f_i} \to \beta_{f_i}, |\beta'_{f_1} \ldots \beta'_{f_{i-1}}| + t_{f_i} - k_{f_i}),$
$i = 1,\ldots,n+1]$.

$D'_{kl} = [Q_{h_i}(o,k_{h_i}) \beta'_{f_\nu} \ldots \beta'_{f_n}, \ \alpha_{h_i} \to \beta_{h_i}, t_{h_i}), \ i=1,\ldots,r+1]$
where $\nu, \nu \leqq n+1$, is the least integer such that $h_i \leqq f_\nu$, and

$D''_{kl} = [\beta''_{f_n} \ldots \beta''_{f_\mu} Q_{j_i}(l_{j_i}, |Q_{j_i}|), \ \alpha_{j_i} \to \beta_{j_i}, \ t_{j_i} - l_{j_i} + |\beta''_{f_n} \ldots \beta''_{f_\mu}|), \ i=1,\ldots,s+1]$,
where $\mu, \mu \leqq n+1$ is the least integer such that $j_i \leqq f_\mu$.

By assumption, for each $h_i \in H$, $t_{h_i} + |\alpha_{h_i}| \leqq k_{h_i}$. Thus $\alpha_{h_i} = Q_{h_i}^{o,k_{h_i}}(t_{h_i}, t_{h_i} + |\alpha_{h_i}|) = Q_{h_i}(t_{h_i}, t_{h_i} + |\alpha_{h_i}|)$.

Furthermore, for each $j_i \in J$, $t_{j_i} \geqq l_{j_i}$. Thus $Q_{j_i}^{l_{j_i}, |Q_{j_i}|}(t_{j_i} - l_{j_i}, t_{j_i} + |\alpha_{j_i}| - l_{j_i}) = \alpha_{j_i} = Q_{j_i}(t_{j_i}, t_{j_i} + |\alpha_{j_i}|)$.

So the applications of $\alpha_{h_i} \to \beta_{h_i}$ and $\alpha_{j_i} \to \beta_{j_i}$ in D'_{kl} and D''_{kl} can be carried out in a similar way as in D. Thus, D'_{kl} and D''_{kl} are subderivations of D.

The subderivations D_{kl}, D'_{kl} and D''_{kl} can now be combined to the derivations $\tilde{D}$ and $\hat{D}$, which are similar to D and defined by

$$\tilde{D} = [Q_m(o,k)\, Q_m(k,l) \times D''_{kl}] \cdot [D'_{kl} \times Q_m(k,l)\, \delta\, Q_1(l_1, |Q_1|)] \cdot [Q_1(o,k_1) \times D_{kl} \times Q_1(l_1, |Q_1|)] \text{ and}$$

$$\hat{D} = [D'_{kl} \times Q_m(k,l)\, Q_m(l, |Q_m|)] \cdot [Q_1(o,k_1)\, \gamma\, Q_m(k,l) \times D''_{kl}] \cdot [Q_1(o,k_1) \times D_{kl} \times Q_1(l, |Q_1|)],$$

where for some strings u,v and $D = [(Q_i, \alpha_i \to \beta_i, t_i), i=1,\ldots,m]$ the (u,v) - <u>extension</u> of D is defined by $[u \times D \times v] = [(u Q_i v, \alpha_i \to \beta_i, t_i + |u|), i=1,\ldots,m]$, and the <u>composition</u> of two derivations D from Q_1 to Q_m and D' from Q'_1 to Q'_n is the derivation $D' \cdot D$ from Q_1 to Q'_n, which is the serial product of D and D', provided $Q_m = Q'_1$.

Thus, $\tilde{D}$ and $\hat{D}$ are obtained from the derivation D by rearranging the applications of productions, such that in the first phase the steps of D_{kl} are carried out and in the second and third phases those of D'_{kl} and D''_{kl}, or reverse. Furthermore, within the single phases the order of applications of productions coincides with the order of applications of the corresponding productions in D.

Notice that the steps carried out in the second and third phases of $\tilde{D}$ or $\hat{D}$ are separated by the strings of ancestors of $Q_m(k,l)$. So the strings of ancestors play the role of barriers for the transmission of informations from the left sides of the strings of ancestors to the right sides, or reverse.

Thus, the subderivations D'_{kl} and D''_{kl} cannot influence each other; they are completely independent of each other as it is due to independent subderivations in context-free grammars.

Exploring, how the use of context allows a contextsensitive grammar to generate a non-context-free language, Book (1973) claims that the capacity of contextsensitive grammars to store and to transmit messages along a string in derivations is responsible for their ability to generate such languages.
This intuitive explanation of the mechanism of how context works originates from our experience of how context is used in many examples of the generation of non-context-free languages, and it is supported by several results where the capacity of sending messages is restricted by the creation of barriers, and then only context-free languages can be generated. For further details see Book (1973).

From this point of view our approach shows, that any occurrence of a symbol in the last string of a derivation creates barriers by its strings of ancestors, and so any derivation contains several barriers for the transmission of messages along the strings in derivations.
The restriction of the generative power of derivations and grammars imposed by these barriers is determined by the number of barriers and thus by the distance between barriers or the distance over which informations can be sent. This quantity is measured by the length of the strings of ancestors of symbols, which we'll consider now.

Definition 5:

Let $G = (N,T,S,P)$ be a contextsensitive grammar and $D = [(Q_i, \alpha_i \to \beta_i, t_i), i=1,\dots,m]$ a derivation from Q_1 to Q_m in G.
Furthermore, for each k,l with $o \leq k < l \leq Q_m$ and for each i, $1 \leq i \leq m$ let $A_i(k,l) = Q_i(k_i,l_i)$ be the strings of ancestors of $Q_m(k,l)$.

The <u>contextsensitivity of the l-th position</u> in Q_m with respect to D is defined by the maximal length of the strings of ancestors of $Q_m(l-1,l)$. Formally,

$$cs_G(D,Q_m,l) = \max\{|A_i(l-1,l)| \,/\, i=1,\dots,m\} = \max\{l_i-(l-1)_i \,/\, i = 1,\dots,m\}.$$

The <u>contextsensitivity of the string Q_m</u>, generated by D is

$$cs_G(D,Q_m) = \max\{cs_G(D,Q_m,l) \,/\, l=1,\dots,|Q_m|\}.$$

The <u>contextsensitivity of a sentential form Q</u> of G is

$$cs_G(Q) = \min\{cs_G(D,Q) \,/\, D \text{ is a derivation from } S \text{ to } Q \text{ in } G\}.$$

Finally, the <u>right-contextsensitivity of a right-sentential form</u> Q of G is obtained by considering right-canonical derivations only. Formally,

$$rcs_G(Q) = \min\{cs_G(D,Q) \,/\, D \text{ is a right-canonical derivation from } S \text{ to } Q\}.$$

Obviously, the contextsensitivity or right-contextsensitivity of any (right-) sentential form Q of a contextsensitive grammar G is greater or equal to 1, $cs_G(Q)$ and $rcs_G(Q)$ are bounded from above by the length of Q, and $cs_G(Q) \leq rcs_G(Q)$.
Furthermore, if a non-context-free production is applied in D, then $cs_G(D,Q) > 1$, and $rcs_G(D,Q) > 1$, respectively.
Finally, for every contextsensitive grammar G, $cs_G(D,Q,l)$, $cs_G(D,Q)$, $cs_G(Q)$ and $rcs_G(Q)$ can effectively be computed, where the computability of $cs_G(Q)$ or

$rcs_G(Q)$ is due to the fact, that there is only a finite number of derivations D from S to Q in G, such that if there is a loop in D, i.e. if $D = [(Q_i, \alpha_i \to \beta_i, t_i),\ i=1,\dots,m]$ and for p,q with $1 \leq p < q \leq m$, $Q_p = Q_q$, then the "distribution" of the strings of ancestors of the single positions of Q_m in Q_p and Q_q must be different, i.e. there exist l, $1 \leq l \leq |Q_m|$ such that $A_p(l-1,l) = Q_p(k_p,l_p)$, $A_q(l-1,l) = Q_q(k_q,l_q)$, and $(k_p,l_p) \neq (k_q,l_q)$

Definition 6:

For every contextsensitive grammar G, we define the total recursive contextsensitivity and right-contextsensitivity functions from $\mathbb{N}$ to $\mathbb{N} \cup \{o\}$ by

$$CS_G(n) = \begin{cases} \max\{cs_G(x) / x \in L(G) \text{ and } |x| = n\}, & \text{if } G \text{ generates terminal strings of length } n. \\ 0 & \text{otherwise.} \end{cases}$$

$$rCS_G(n) = \begin{cases} \max\{rcs_G(x) / x \in L(G) \text{ and } |x| = n\}, & \text{if } G \text{ generates terminal strings of length } n. \\ 0 & \text{otherwise.} \end{cases}$$

Notice that $CS_G(n) = o$ iff $rCS_G(n) = o$ iff G does not generate terminal strings of length n, and $CS_G(n) \leq rCS_G(n) \leq n$ for all n.

The contextsensitivity and right-contextsensitivity functions measure the derivational complexity of grammars, and the derivational complexity of languages may be measured by assigning to a language L the "smallest" contextsensitivity or right-contextsensitivity functions of all contextsensitive grammars that generate L.

Definition 7:

A total recursive function f bounds the contextsensitivity or right-contextsensitivity function of grammar G, if for every nonnegative integer n, $CS_G(n) \leq f(n)$ or $rCS_G(n) \leq f(n)$, respectively. Then G is termed a f-CS or f-rCS bounded grammar or a grammar with f(n)-bounded contextsensitivity or right-contextsensitivity.

The families of contextsensitive languages with f(n)-bounded contextsensitivity or right-contextsensitivity are defined by

$$CSL(f) = \bigcup_{c>o} \{L(G) / G \text{ is a contextsensitive grammar with } c \cdot f(n)\text{-bounded contextsensitivity}\}$$

$$CSL_r(f) = \bigcup_{c>o} \{L(G) / G \text{ is a contextsensitive grammar with } c \cdot f(n)\text{-bounded right-contextsensitivity}\}.$$

These are the families of languages we shall study.

LOWER AND UPPER BOUNDING FUNCTIONS

For every grammar G, if G generates terminal strings of length n, then $CS_G(n) \geq 1$ and $rCS_G(n) \geq 1$. Furthermore, if G is context-free, then $CS_G(n) > o$ implies $rCS_G(n) = CS_G(n) = 1$.

Hence, the function $f(n) = 1$ for all n is the "smallest" bounding function for the contextsensitivity and right-contextsensitivity functions of nontrivial grammars and languages, and f dominates the context-free grammars and languages.
Thus, the context-free grammars and languages are almost "free of cost" of contexsensitivity and right-contextsensitivity.

On the other hand, since $CS_G(n) \leq n$ and $rCS_G(n) \leq n$, $id(n) = n$ is an upper bounding function for the contextsensitivity and right-contextsensitivity functions of contextsensitive grammars and languages, and $CSL_r(id) = CSL(id) = CSL$, where CSL denotes the family of contextsensitive languages.

Finally, for every bounding function f, $CSL_r(f) \subseteq CSL(f)$,
and if $f(n) \leq g(n)$ for all n, then $CSL(f) \leq CSL(g)$ and $CSL_r(f) \leq CSL_r(g)$.

We will show now, that grammars with bounded right-contextsensitivity can easily generate integers of the form 2^{2^n}, however it is very hard to compare two binary numbers or to make a copy of a binary number, or something else.

Theorem 2:

There is a log log n - rCS bounded contextsensitive grammar G that generates the non-context-free language $L_1 = \{a^{2^{2^n}} / n \geq 1\}$.
Proof:
Suppose that the derivations of G start with applications of the productions $S \rightarrow ¢\ 1\ A\ \$$, $A \rightarrow 0A$, $A \rightarrow 0$, and terminate by the use of $¢10\$ \rightarrow aaaa$.
Furthermore, there are two phases in the derivations of G, which are carried out one after another. In the first phase, the binary number $¢b_i\$$ is diminished by 1, where ¢, \$ are markers and b_i is the binary encoding of the integer i, and in the second phase, a copy of $¢b_i\$$ is made. Then $L(G) = \{a^{2^{2^n}} / n \geq 1\}$ and for any single a occurring at any position of the terminal string $a^{2^{2^n}}$, $n \geq 2$, the string of ancestors of maximal length is $¢\ b_i\ \$\ ¢\ b_i\ \$$ with $i = 2^n-1$, whose length is about $\log\log(2^{2^n})$. Now, an appropriate speed up precisely gives log log n.

There is a new type of automata, called one-way auxiliary pushdown automata, which fits to grammars with bounded right-contextsensitivity functions. These automata are introduced in Brandenburg (1977), and are the one-way model of the well-known (two-way) auxiliary pushdown automata, studied by Cook (1971).

Definition 8:

A one-way auxiliary pushdown automaton M, 1-APDA for short, is a Turing machine with a one-way, read-only input tape, several read-write work tapes and a pushdown tape. M is f(n)-tape bounded, if its work tapes are $f(n)$-tape bounded, i.e. if for every $x \in T(M)$ there is an accepting computation such that M uses at most $f(|x|)$ squares on any of its work tapes.

The subsequent results characterize the power of 1-APDA. For proofs see Brandenburg (1977).

Proposition 1:

i) If $\lim\sup f(n)/\log\log n = o$, then $f(n)$-tape bounded 1-APDA accept only context-free languages.

ii) The language $L = \{w_1 c w_2 c \overleftarrow{w}_1 c \overleftarrow{w}_2 / w_1, w_2 \in \{o,1\}^*, |w_1| = |w_2|\}$ cannot be accepted by $f(n)$-tape bounded 1-APDA, unless $f(n)$ is at least a linear function.

In the following we show that $f(n)$-tape bounded 1-APDA can simulate right-canonical derivations, whose contextsensitivity is bounded by $f(n)$.

Lemma 1:

Let $G = (N,T,S,P)$ be a separated contextsensitive grammar, that is G only has productions of the form $\alpha \to \beta$ with $\alpha,\beta \in N^+$ or $A \to a$ with $A \in N$ and $a \in T$. If D is a right-canonical derivation from Q_1 to Q_m, $Q_m \in T^*$ and $cs_G(D,Q_m) = k$ for some integer k, then at each step of the derivation the rewriting takes place at most k symbols to the left of the rightmost nonterminal occurring in the current string Q_i.

Proof:

Consider the i-th step of D. Then $Q_i = \gamma_i\alpha_i\delta_i$ with $\delta_i = \delta_i' y_i$, such that $\gamma_i, \alpha_i, \delta_i' \in N^*$, $y_i \in T^*$ and y_i is a final substring of Q_m. Now consider the position in Q_m immediately to the left of y_i. Then $\alpha_i\delta_i'$ is a final substring of the string of ancestors of $Q_m(l-1,l)$ with $l = |Q_m| - |y_i|$. Thus $|\alpha_i\delta_i| \leq k$.

Notice, that the standard construction of an equivalent separated grammar G' from a contextsensitive grammar G preserves the contextsensitivity functions, i.e. $CS_{G'} = CS_G$ and $rCS_{G'} = rCS_G$.

Theorem 3:

If G is a f-rCS bounded contextsensitive grammar, then one can effectively find an $f(n)$-tape bounded 1-APDA M with $L(G) = T(M)$. Furthermore, the sum of the lengths of the work tapes and the pushdown tape does not exceed the length of the input string.

Proof:

Assume that G is a separated grammar. Then M acts as a "bottum-up analyser" which simulates the right-canonical derivations of G in reverse.

If $D = [(Q_i, \alpha_i \to \beta_i, t_i), i=1,..,m]$ is a right-canonical derivation from S to x in G with $x \in L(G)$ and $cs_G(D,x) \leqq f(n)$, then for each i, $1 \leqq i \leqq m-1$, $Q_{i+1} = \gamma_i \beta_i \delta_i y_i$, where $\gamma_i \in N^*$ and $|\gamma_i| = t_i$, $y_i \in T^*$ is a final substring of x, $\delta_i \in N^*$ and $\beta_i \in N^*$ or $\beta_i \in T$, and if $\beta_i \in T$, then $\delta_i = e$ and $\beta_i y_i$ is a final substring of x. Furthermore, $|\beta_i \delta_i| = |\alpha_i \delta_i| - |\alpha_i| + |\beta_i| \leqq f(n) - |\alpha_i| + |\beta_i|$.

Now, Q_{i+1} is stored by M such that the pushdown tape contains γ_i with the rightmost symbol of γ_i on the top, the work tape contains $\beta_i \delta_i$, which is possibly compressed to a string of length less or equal to $f(n)$, and the input head scans the first symbol of y_i. If $\beta_i \in N^*$, then M rewrites β_i by α_i on its work tape, and if $\beta_i \in T$, then if the input head reads β_i, M writes α_i on its work tape which was empty before, and the input head moves one cell to the right.

Now, M guesses β_{i-1} and its position and shifts some symbols from the work tape on the pushdown tape, or reverse, such that β_{i-1} occurs at the left end of the work tape. M is initiated with an empty work tape and an empty pushdown tape while the input head scans the first symbol of the input, and M accepts, if the total input is processed, the pushdown tape is empty and the work tape contains S.

Theorem 4:

i) If $\lim\sup f(n)/\log\log n = o$, then grammars with $f(n)$-bounded right-context-sensitivity generate only context-free languages.

ii) The language $L_2 = \{w_1 c w_2 c \overleftarrow{w}_1 c \overleftarrow{w}_2 / \ w_1, w_2 \in \{o,1\}^*, \ |w_1| = |w_2|\}$ cannot be generated by f-rCS bounded grammars, unless $f(n) \geqq c \cdot n$ for some $c > o$.

Thus, if the "cost of right-contextsensitivity" is considered, then $\log\log n$ is a strict lower bounding function for the generation of non-context-free languages and $L_1 = \{a^{2^{2^n}} / \ n \geqq 1\}$ is the "simplest" non-context-free language.

Furthermore, the linear functions are strict upper bounding functions for the right-contextsensitivity functions of contextsensitive grammars and languages and $L_2 = \{w_1 c w_2 c \overleftarrow{w}_1 c \overleftarrow{w}_2 / \ w_1, w_2 \in \{o,1\}^*, \ |w_1| = |w_2|\}$ is the "hardest" contextsensitive language.

The simplest and the hardest languages L_1 and L_2 are now used to pad every contextsensitive language to a "simple" language and to establish dense chains of families of contextsensitive languages with $f(n)$-bounded right-contextsensitivity.

Theorem 5:

If $L \subseteq \Sigma^*$ is a contextsensitive language and $a \notin \Sigma$, then $L_s = \{xy / \ x \in L \text{ and } y = a^{2^{2^{|x|}}}\} \in CSL_r(\log\log n)$.

Proof:

The grammar G_s generating L_s first simulates the grammar G with $L(G) = L$, and if G generates a terminal string, then G_s completes the tail of $2^{2^{2^n}}$ a's in the way described in theorem 2.

Theorem 6:

If $f(n)$ and $g(n)$ are tape constructable functions with $n \leq f(n), g(n) \leq 2^{2^n}$, and $\limsup f(n)/g(n) = o$, then there exist a language L generated by a $g(\lceil \log \log n \rceil)$-rCS bounded grammar G, but not generated by any $f(\lceil \log \log n \rceil)$-rCS bounded grammar.

$\lceil k \rceil$ denotes the least integer greater or equal to k.

Proof:

Let $L = L(G) = \{w_1 c w_2 c \overleftarrow{w}_1 c \overleftarrow{w}_2 a^{2^{2^m}} / \; w_1, w_2 \in \{o,1\}^*, \; |w_1| = |w_2| = g(m)\}$.
G first generates strings of the form $B^{g(m)} A^m$, $m \geq 1$, and then transforms $B^{g(m)}$ into $w_1 c w_2 c \overleftarrow{w}_1 c \overleftarrow{w}_2$, with $w_1, w_2 \in \{o,1\}^*$, $|w_1| = |w_2| = g(m)$, and A^m into $a^{2^{2^m}}$.
It should be obvious that $rCS_G(n) \leq c_1 \cdot g(\lceil m \rceil)$ with $m = \log \log n$.
Thus, $L \in CSL_r(g(\lceil \log \log n \rceil))$.
Now, assume that there is a $f(\lceil \log \log n \rceil)$-rCS bounded grammar G' generating L.
Then there is a $f(n)$-rCS bounded grammar G'', which is constructed from G', such that $L(G'') = \{w_1 c w_2 c \overleftarrow{w}_1 c \overleftarrow{w}_2 / \; w_1, w_2 \in \{o,1\}^*, \; |w_1| = |w_2|\}$.
G'' imitiates G', however G'' suppresses the generation of the tails of a's of the strings of $L(G')$.
Since $g(n) \leq 2^{2^n}$ and $\limsup f(n)/g(n) = o$, this contradicts theorem 4, ii).

The main result of Cook (1971) is the equivalence of $f(n)$-tape bounded (two-way) auxiliary pushdown automata with $f(n) \geq \log n$ and deterministic multitape Turing machines which operate in time $2^{c \cdot f(n)}$. Since (two-way) auxiliary pushdown automata simulate 1-APDA, and $f(n)$-tape bounded 1-APDA are stronger than grammars with $f(n)$-bounded right-contextsensitivity, we can improve the best-known lower time bound for the recognition of contextsensitive languages, which is exponential time, for some particular classes of contextsensitive language, and e.g. we obtain that the large class of contextsensitive languages with right-contextsensitivity functions bounded by $\log n$ are in P, the family of sets which are accepted by deterministic Turing machines which operate in polynomial time.
However, the "hardest" language" L_2 can be recognized in linear time on a two-way pushdown automaton. Thus, our bounds cannot be tight.

Theorem 7:

If L is a f-rCS bounded context-sensitive language and $f(n) \geq \log n$, then there is a deterministic multitape Turing machine operating in time $2^{c \cdot f(n)}$ that accepts L.

CLOSURE PROPERTIES AND DECIDABILITY QUESTIONS:

If $\lim\sup f(n)/\log\log n = o$, then $CSL_r(f)$ coincides with the family of e-free context-free languages. (See theorem 3). Thus, $CSL_r(f)$ has the same closure properties and decidability questions as the context-free languages. (See Hopcroft, Ullman 1969).

However, it turns out that the families of contextsensitive languages with $f(n)$-bounded right-contextsensitivity, where $f(n) \geq \log\log n$, have the same closure and decidability properties as the contextsensitive languages.
In this sense, the bounding function $\log\log n$ is a border line for the families of contextsensitive languages with bounded right-contextsensitivity between similarity to the context-free languages and to the contextsensitive languages.

From theorem 5 and the fact that every recursively enumerable set can be expressed as the homomorphic image of a contextsensitive language, we obtain:

Theorem 8:

For every contextsensitive language L there is a homomorphism h and a $\log\log n$ - rCS bounded contextsensitive grammar G such that $L = L(G)$. h is at most a doubly exponentially erasing homomorphism.
Furthermore, every recursively enumerable set can be expressed as the homomorphic image of a $\log\log n$ - rCS bounded language.

Corollary 1:

If $f(n) \geq \log\log n$, then $CSL_r(f)$ and $CSL(f)$ are not closed under erasing homomorphisms.

Since the context-free languages are closed under homomorphisms, the opposite result holds for $CSL_r(f)$ with $\lim\sup f(n)/\log\log n = o$, provided we join the empty word.

Theorem 9:

For any bounding function f, $CSL_r(f)$ and $CSL(f)$ are closed under the operations of union,product, Kleene plus, intersection with regular sets, and substitution.

The proofs for the closure properties easily come out by combining the constructions which are used for the context-free and the contextsensitive languages.

We do not know yet, whether these families of languages are AFL's, since we do not know, whether they are closed under inverse homomorphisms or under k-restricted homomorphisms. This problem is connected with the solution of the linear speed up theorem, which is one of our open problems.

Theorem 1o:

If lim sup $f(n)/n = o$, then $CSL_r(f)$ does not contain the intersections of all context-free languages.

Hence, $CSL_r(f)$ is not closed under intersection.

Furthermore, $CSL_r(f)$ is not closed under complementation.

Proof:

The "hardest" language $L_2 = \{w_1cw_2c\overleftarrow{w}_1c\overleftarrow{w}_2 / w_1,w_2 \in \{o,1\}^*, \quad |w_1| = |w_2|\}$ can be expressed as the intersection of the context-free languages $\{w_1cxc\overleftarrow{w}_1cy/x,y,w_1 \in \{o,1\}^*\}$, $\{xcw_2cyc\overleftarrow{w}_2/x,y,w_2 \in \{o,1\}^*\}$ and $\{w_1cw_2cxcy/|w_1| = |w_2|, \quad x,y,w_1,w_2 \in \{o,1\}^*\}$, and the complement of L_2 is a context-free language.

Notice that $CSL_r(id) = CSL$ is closed under intersection, and it is the well-known 2. LBA problem, whether the contextsensitive languages are closed under complementation.

Theorem 11:

For any bounding function f with $f(n) \geqq \log\log n$, and any f-rCS or f-CS bounded contextsensitive grammar G, the emptiness and the admissability problem are undecidable, where the admissability problem is the question, whether for any string u there are strings x and y such that there is a derivation from S to xuy in G.

Furthermore, it is undecidable, whether for any u,v there exists a derivation in G such that v is a string of ancestors of an occurrence of u.

The proofs of the first and the second problem are based on theorem 8 and can be given as for contextsensitive grammars, and the third claim can be deduced from the undecidability of the admissability problem.

OPEN PROBLEMS:

Finally, we state some open problems, which immediately result from the afore-said.

i) Is $CSL_r(f) = CSL(f)$ for any f?

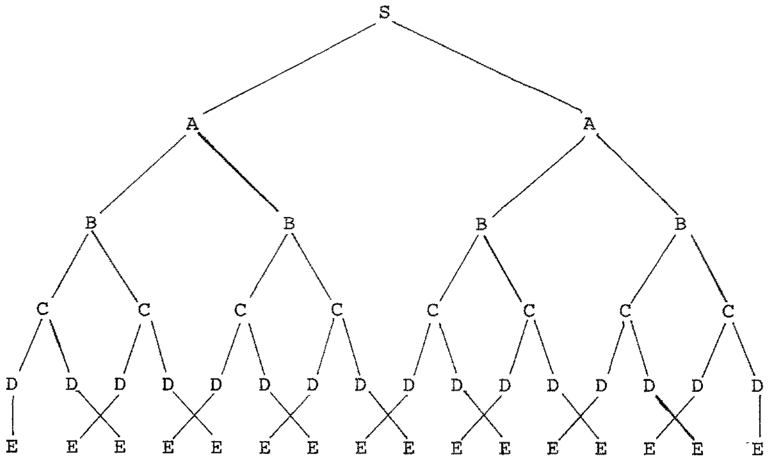
Our attempt to prove this equality by applying the usual tactic of rearranging the applications of productions, such that a given derivation is transformed to a similar right-canonical derivation, failed, since this method does not necessarily preserve the contextsensitivity of strings.

This is illustrated by the example $S \to AB$, $AC \to aA$, $B \to CB$, $AB \to bb$. If $x = a^nbb$ is generated, then a derivation with a zigzag application of $B \to CB$ and $AC \to aA$ yields $cs_G(x) = 3$, however $rcs_G(x) = |x|$.

In connection with this question it might be of interest whether $CSL_r(f)$ is closed under reversal.

ii) Is there a speed-up theorem for grammars and languages with f(n)-bounded context-sensitivity or right-contextsensitivity?

Difficulties occur, if the derivations are "balanced trees" where the test of balance is accomplished in the final part of the derivation, i.e. if the graphical representation of the derivations is as follows



Now, the usual "speed up" construction does not work, if the nodes themselves cannot be speeded up sufficiently.
It seems to me, that a speed up may destroy the intrinsic structures of derivations.

REFERENCES:

R.V. Book (1969), Grammars with time functions,
"Mathematical Linguistics and Automatic Translation, NSF 23", Harvard University Cambridge, Mass., 1969.

R.V. Book (1971), Time-bounded grammars and their languages,
J. Comput. System Sci. 5 (1971), 397 - 429.

R.V. Book (1973), On the structure of context-sensitive grammars,
Internat. J. Comput. Information Sci. 2 (1973), 129 - 139.

F.J. Brandenburg (1977), On one-way auxiliary pushdown automata,
Proceedings 3rd GI Conference Theoretical Computer Science, in:
Lecture Notes in Computer Science, 48 (1977), 132 - 144.

H.W. Buttelmann (1975), On the syntactic structures of unrestricted grammars,
I. Generative grammars and phrase structure grammars.
Information and Control 29 (1975), 29 - 8o.

M.P. Chytil (1976), Analysis of the non-context-free component of formal languages,
Proceedings 5th Symposium Mathematical Foundations of Computer Science 1976, in:
Lecture Notes in Computer Science 45 (1976), 23o - 236.

S.A. Cook (1971), Characterization of pushdown machines in terms of time-bounded computers. J. Assoc. Comput. Mach. 18 (1971), 4-18 .

S. Ginsburg, E.H. Spanier (1968), Derivation-bounded languages, J. Comput. System Sci. 2 (1968), 218 - 25o.

T.V. Griffiths (1968), Some remarks on derivations in general rewriting systems, Information and Control 12 (1968), 27 - 54.

F.C. Hennie (1965), One-tape, off-line Turing machine computations, Information and Control 8 (1965), 533 - 546.

J.E. Hopcroft, J.D. Ullman (1969), Formal languages and their relation to automata, Addison-Wesley Publishing Company (1969).

G. Wechsung (1975), Characterization of some classes of context-free languages in terms of complexity classes.
Proceedings 4th Symposium Mathematical Foundations of Computer Science 1975, in: Lecture Notes in Computer Science 32 (1975), 457 - 461.