Winter 11-1-1983

An East Proof of Greibach Normal Form ; CU-CS-255-83

Andrzej Ehrenfeucht
University of Colorado Boulder

Grzegorz Rozenberg
University of Leiden

AN EASY PROOF OF GREIBACH NORMAL FORM

by

Andrzej Ehrenfeucht* and Grzegorz Rozenberg**

CU-CS-255-83 September, 1983

All correspondence to second author.

*University of Colorado, Department of Computer Science, Boulder, CO 80309

**Institute of Applied Mathematics and Computer Science, University of Leiden, Leiden, The Netherlands.

This research was supported by NSF grant number MCS 83-05245.

AN EASY PROOF OF GREIBACH NORMAL FORM

by

Andrzej Ehrenfeucht* and Grzegorz Rozenberg**

All correspondence to second author.

*University of Colorado, Department of Computer Science, Boulder, CO 80309

**Institute of Applied Mathematics and Computer Science, University of Leiden, Leiden, The Netherlands.

ABSTRACT

We present an algorithm which given an arbitrary Λ-free context-free grammar produces an equivalent context-free grammar in 2 Greibach normal form. The upper bound on the size of the resulting grammar in terms of the size of the initially given grammar is given. Our algorithm consists of an elementary construction, while the upper bound on the size of the resulting grammar is not bigger than the bounds known for other algorithms for converting context-free grammars into equivalent context-free grammars in Greibach normal form.

INTRODUCTION

One of the important directions of research in the theory of context-free grammars is searching for normal forms, see, e.g., [H], [S] and [MSW]. Among many normal forms available for context-free grammars Greibach normal form plays a very important role and so quite a number of algorithms are available which given a context-free grammar yield an equivalent one in Greibach normal form.

In this paper we present an algorithm for achieving the same aim. However our algorithm is different from other existing algorithms both in the methodology (its main part consists of rather simple manipulations of essentially right, or left, linear grammars) and in the result (it yields *directly*, independently of the form of the original grammar, a context-free grammar in Greibach Normal Form where the length of the right-hand side of any production does not exceed three). Our algorithm reduces the whole construction to the level of regular languages. Therefore, we believe that also in a basic course about formal languages one should teach Greibach Normal Form in this fashion: when the fundamentals concerning regular languages are known, our algorithm is easily understood. Essentially, the algorithm consists of a switch between right and left linear grammars.

0. PRELIMINARIES

We assume the reader to be familiar with the basic theory of context-free grammars (see, e.g, [H] and [S]).

We recall now several notational and terminological matters needed in the paper.

For a finite set Z, $\#Z$ denotes its cardinality. For sets Z and V, $Z \setminus V$ denotes their difference.

For a word x, $|x|$ denotes its length and if x is nonempty, then $first(x)$ denotes the first letter of x while $rest(x)$ denotes the word resulting from x after removing its first letter; $alph(x)$ denotes the set of letters occurring in x. Λ denotes the empty word. For a language K, $first(K) = \{first(x) : x \in K\}$.

For a context-free production $\pi = A \to \alpha$, $rhs(\pi) = \alpha$.

For a set P of context-free productions, the *size of* P, denoted $size(P)$, denoted $size(P)$, is defined by $size(P) = \sum_{A \to \alpha \in P} |A\alpha|$. If Γ is an alphabet, then

$good_\Gamma(P) = \{\pi \in P : first(rhs(\pi)) \in \Gamma\}$ and

$bad_\Gamma(P) = \{\pi \in P : first(rhs(\pi)) \notin \Gamma\}$.

A *context-free grammar (cf grammar) is specified in the form* $G = (\Sigma, \Delta, P, S)$ where Σ is the total alphabet of G, Δ its terminal alphabet, P its set of productions and S its axiom. We use Σ_G, Δ_G, P_G and S_G to denote Σ, Δ, P and S respectively; Θ_G denotes the set of nonterminals of G - that is $\Theta_G = \Sigma_G \setminus \Delta_G$. Also, $maxr(G) = \max\{|rhs(\pi)| : \pi \in P_G\}$ and the *size of* G is defined by $size(G) = size(P_G)$. We say that G is *chain-free* if it does not contain productions of the form $A \to B$ with $A, B \in \Theta_G$.

A cf grammar G is called *right (left) linear* if all productions of it are of the form $A \to \alpha B$ ($A \to B\alpha$ respectively), where $A \in \Theta_G$, $B \in \Theta_G \cup \{\Lambda\}$ and $\alpha \in \Delta_G^*$.

Let G be a cf grammar.

G is in *Greibach Normal Form* (GNF) if $first(rhs(\pi)) \in \Delta_G$ and $rest(rhs(\pi)) \in \Theta_G^*$ for each $\pi \in P_G$.

If G is in (weak) GNF and k is a positive integer such that $| rest(rhs(\pi)) | \leq k$ for each $\pi \in P_G$, then G is in k *Greibach Normal Form* abbreviated k GNF.

A *context-free scheme (cf scheme)* is a construct (Σ, Δ, P) such that for each $S \in \Sigma \backslash \Delta$, (Σ, Δ, P, S) is a cf grammar. All terminology and notation concerning cf grammars (except for matters involving the axiom) carries over to cf schemes.

As usual in formal language theory, in order to simplify notation and terminology, we will often identify a derivation in a cf grammar with its trace (which is the sequence of intermediate sentential forms). Also, sometimes we will not distinguish too carefully between letters and their occurrences in words. These notational simplifications should not lead to confusion.

We recall now two well-known (and easy to prove) results concerning transformations of cf grammars.

Let G be a cf grammar, let $A \in \Theta_G$, $A \neq S_G$ and let $A \to \gamma_1, \ldots, A \to \gamma_m$ be all productions for A in G. Assume that $A \notin alph(\gamma_1 \cdots \gamma_m)$. Let ρ_A be the transformation of G done as follows:

remove A from Σ_G, remove from P_G all productions for A and in all other productions of P_G replace all occurrences of A at their right-hand sides by all combinations of $\gamma_1, \ldots, \gamma_m$.

Let $\rho_A(G)$ be the resulting cf grammar.

Proposition 0.1. $L(G) = L(\rho_A(G))$. ▪

Let G be a cf grammar, let $A \in \Theta_G$ and let $A \to \gamma_1, \ldots, A \to \gamma_m$ be all productions for A in G. Let $\pi = Y \to \alpha A \beta \in P_G$. Let $\zeta_{\pi,A}$ be the transformation of G

done as follows: remove π from P_G and add to P_G the set of productions $Y \to \alpha\gamma_1\beta$, ..., $Y \to \alpha\gamma_m\beta$. Let $\zeta_{\pi,A}(G)$ be the resulting cf grammar. (As we have said above, in our notation we do not distinguish very carefully between letters and their occurrences - however we point out here that the subscript A in $\zeta_{\pi,A}$ refers to the particular occurrence of A in $\alpha A\beta$.)

Proposition 0.2. $L(G) = L(\zeta_{\pi,A}(G))$. ∎

We conclude this section by defining the notion that is very basic for this paper.

Let G be a cf scheme and let $A \in \Theta_G$.

Then $L_A(G) = L((\Sigma_G, \Delta_G, P_G, A))$ and $\mathbf{F}(G) = \{L_A(G) : A \in \Theta_G\}$.

If $\mathbf{L}$ is a finite family of languages, then we say that G *covers* $\mathbf{L}$ if $\mathbf{L} \subseteq \mathbf{F}(G)$.

1. THE MAIN CONSTRUCTION

In this section we present a construction which given an arbitrary Λ-free cf grammar G yields an equivalent cf grammar in 2 GNF.

Our construction consists of three steps.

Construction 1.1. Let G be a Λ-free cf grammar.

STEP 1

For each $A \in \Theta_G$ let W_A be the subset of all sentential forms obtained (starting with A) by rewriting *only the first symbol of a sentential form;* then let $T_A = W_A \cap \Delta_G \Sigma_G^*$. (The above definition of T_A is somewhat informal; in fact T_A can be defined by a left linear grammar). Then $\mathbf{T}(G) = \{T_A \mid A \in \Theta_G\}$.

STEP 2

Let H be an arbitrary right linear scheme such that

(1) H is Λ-free, chain-free and every nonterminal of H is successful (i.e., it can derive a terminal string),

(2) each production of H has the form $X \to YZ$, where $X \in \Theta_H$, $Y \in \Delta_H$ and $Z \in \Theta_H \cup \{\Lambda\}$,

(3) $\Delta_H = \Sigma_G$ and

(4) H covers $\mathbf{T}(G)$.

For each $A \in \Theta_G$, let N_A be an arbitrary but fixed nonterminal of H such that $L_{N_A}(H) = T_A$.

STEP 3

Let J be the cf grammar such that $\Sigma_J = \Sigma_H \backslash \Theta_G$, $\Delta_J = \Delta_G$, $S_J = N_{S_G}$ and $P_J = P_J^1 \cup P_J^2$, where $P_J^1 = good_{\Delta_G}(P_H)$ and P_J^2 is defined as follows:

for each production $X \to YZ$ in $bad_{\Delta_G}(P_H)$, where $Z \in \Theta_H \cup \{\Lambda\}$ and each produc-

tion $N_Y \to CT$ in P_H such that $T \in \Theta_H \cup \{\Lambda\}$, P_J^2 includes the production $X \to CTZ$; P_J^2 contains only productions obtained in this way. ■

Remark.

(1) The notation T_A used in the description of STEP 1 is somewhat ambiguous (because no index referring to G is involved). However we will use it only in situations where G is understood from the context.

(2) Since, obviously, all languages in $\mathbf{T}(G)$ are regular (and Λ-free), a right linear scheme required in STEP 2 exists (an algorithmic construction of such a scheme is discussed in the next section).

(3) One should note that, in the notation of STEP 3 above, $C \in \Delta_G$.

(4) The following two lemmas demonstrate that J is well defined; moreover they will be used in the proof of Theorem 1.1. Since the proofs follow easily from the definition of J given above, they are left to the reader.

Lemma 1.1. For each production $\pi \in P_J$, $first(rhs(\pi)) \in \Delta_G$. ■

Lemma 1.2. For each $\pi \in P_J$, $rest(rhs(\pi)) \in (\Sigma_H \backslash \Sigma_G)^*$ and moreover $| rest(rhs(\pi)) | \leq 2$. ■

We will prove now that J has the intended properties.

Theorem 1.1. J is a cf grammar in 2 *GNF* and $L(G) = L(J)$.

Proof.

That J is in 2 *GNF* follows directly from Lemma 1.1 and Lemma 1.2.

In order to prove that $L(G) = L(J)$ we introduce an auxiliary construct - the cf grammar I defined by: $\Sigma_I = \Sigma_H$, $\Delta_I = \Delta_G$, $S_I = N_{S_G}$ and $P_I = P_H \cup \{A \to N_A : A \in \Theta_G\}$.

Now the equality $L(G) = L(J)$ follows from the following two lemmas.

Lemma 1.3. $L(I) = L(J)$.

Proof of Lemma 1.3.

This follows from Proposition 0.1, Proposition 0.2 and the observation that J is obtained from I by first applying to I a finite number of times, transformations of type ρ_A and then applying transformations of type $\zeta_{\pi,A}$. ∎

Lemma 1.4. $L(G) = L(I)$.

Proof of Lemma 1.4.

(i) $L(G) \subseteq L(I)$.

Let $\tau = (S_G = z_0, z_1, \ldots, z_m)$, $m \geq 1$, be a leftmost derivation of a word in $L(G)$. We divide τ into *segments* as follows.

The *first segment* of τ is $\tau^{(1)} = (z_0, z_1, \ldots, z_{i_1})$, $i_1 \geq 1$, where i_1 is the minimal integer such that $first(z_{i_1}) \in \Delta_G$. Let X_1 be the leftmost (occurrence of a) nonterminal in z_{i_1}.

Now assume that for $j \geq 1$ the *j-th segment* of τ, $\tau^{(j)} = (z_{i_{j-1}+1}, \ldots, z_{i_j})$, is defined (for $j = 1$ we set $i_{j-1}+1 = 0$), where $i_j \neq m$. Let X_j be the leftmost (occurrence of a) nonterminal in z_{i_j}. Then the $(j+1)$*-th segment* of τ is defined as $\tau^{(j+1)} = (z_{i_j+1}, \ldots, z_{i_{j+1}})$, where i_{j+1} is the minimal integer bigger than i_j such that the leftmost (occurrence of a) letter contributed by X_j to $z_{i_{j+1}}$ belongs to Δ_G.

In this way we have partitioned τ into r segments for some $r \geq 1$. For $0 \leq j \leq r-1$, let w_j be the subword contributed by X_j to $z_{i_{j+1}}$ (we set $X_0 = S_G$). Clearly $w_j \in T_{X_j}$ .....(1)

Now z_m can be derived in I as follows.

Rewriting N_{S_G} using productions of H we derive w_0; this can be done because H covers $\mathbf{T}(G)$ and (1) holds. Hence we have simulated $\tau^{(1)}$ deriving z_{i_1} in I. If $r = 1$, then we are done, otherwise we proceed as follows.

Assume that we have derived z_{i_j} in I, where $1 \leq j < r$. Then by rewriting X_j using production $X_j \to N_{X_j}$ and then using productions of H to derive w_j from N_{X_j} we obtain $z_{i_{j+1}}$; this can be done because H covers $\mathbf{T}(G)$ and (1) holds.

Hence z_m can be derived in I and consequently $L(G) \subseteq L(I)$.

(ii) $L(I) \subseteq L(G)$.

We divide the nonterminals in I into two categories: $\mathbf{C}_1 = \Sigma_H \backslash \Sigma_G = \Theta_H$ and $\mathbf{C}_2 = \Theta_G$; thus $\Sigma_I \backslash \Delta_I = \mathbf{C}_1 \cup \mathbf{C}_2$. Clearly, nonterminals from $\mathbf{C}_1$ are rewritten using productions from P_H and nonterminals from $\mathbf{C}_2$ are rewritten using productions of the form $A \to N_A$.

A *two-phase derivation* in I is a derivation satisfying the following condition:
- if a sentential form contains a letter from $\mathbf{C}_1$, then an occurrence of a letter from $\mathbf{C}_1$ is rewritten, otherwise an occurrence of a letter from $\mathbf{C}_2$ is rewritten.

Obviously each word in $L(I)$ can be derived by a two-phase derivation. Moreover it is obvious that:
- each sentential form of a two-phase derivation contains at most one occurrence of a letter from $\mathbf{C}_1$, and
- if a sentential form of a two-phase derivation contains no occurrence of a letter from $\mathbf{C}_1$, then either this sentential form is in $L(I)$ or the next sentential form contains a letter from $\mathbf{C}_1$.

Consider now a successful two-phase derivation τ in I and let $u_1, \ldots, u_r$ be the sequence of all consecutive sentential forms in τ such that, for $1 \leq j \leq r$, u_j does not contain a letter from $\mathbf{C}_1$; u_r is in $L(I)$. Note that $u_1 \in L_{N_{S_G}}(H) = T_{S_G}$

and so u_1 can be derived from S_G in G. So if $r = 1$ then $u_r \in L(G)$. Note that $u_{j+1} = w_1 v w_2$, where $u_j = w_1 A w_2$, $A \in \Theta_G$ and $v \in L_{N_A}(H) = T_A$; hence $u_j \overset{*}{\underset{G}{\Longrightarrow}} u_{j+1}$. Consequently $u_r \in L(G)$.

Hence $L(I) \subseteq L(G)$.

Thus $L(I) = L(G)$ and Lemma 1.3 holds. ∎

From Lemma 1.3 and Lemma 1.4 it follows that $L(G) = L(J)$. Hence the theorem holds. ∎

We end this section with an example illustrating Construction 1.1.

Example 1.1. Consider the cf grammar G such that $\Sigma_G = \{A_1, A_2, A_3, 0, 1\}$, $\Delta_G = \{0, 1\}$, $S_G = A_1$ and P_G consists of the following productions:

$A_1 \to A_2 A_3$,

$A_2 \to A_1 A_2 \mid A_2 \to 1$,

$A_3 \to A_1 A_3 \mid A_3 \to 0$.

This is a grammar from [H] (p. 113).

It is easily seen that

$T_{A_1} = 1(A_3 A_2)^* A_3$,

$T_{A_2} = 1(A_3 A_2)^*$ and

$T_{A_3} = 0 + 1(A_3 A_2)^* A_3 A_3$.

Let H be the following right linear scheme.

$\Sigma_H = \{Y_1, Y_2, Y_3, Z_1, Z_2, Z_3, Z_4, U_1, U_2, U_3, U_4\} \cup \Sigma_G$,

$\Delta_H = \Sigma_G$ and P_H consists of the following productions:

$Y_1 \to 1 Y_2$

$Y_2 \to A_3 Y_3 \mid A_3$,

$Y_3 \to A_2 Y_2$,

$Z_1 \to 1Z_2 \mid 1$,

$Z_2 \to A_3 Z_3 \mid A_3 Z_4$,

$Z_3 \to A_2 Z_2$,

$Z_4 \to A_2$,

$U_1 \to 1U_2 \mid 0$,

$U_2 \to A_3 U_3 \mid A_3 U_4$,

$U_3 \to A_2 U_2$,

$U_4 \to A_3$.

It is easily seen that H satisfies the requirements from STEP 2 of Construction 1.1.

If we set now the correspondence

$N_{A_1} = Y_1$, $N_{A_2} = Z_1$ and $N_{A_3} = U_1$,

then indeed we have

$L_{Y_1}(H) = T_{A_1}$, $L_{Z_1}(H) = T_{A_2}$ and $L_{U_1}(H) = T_{A_3}$.

Furthermore

$good_{\Delta_G}(P_H) = \{Y_1 \to 1Y_2 , Z_1 \to 1Z_2 , Z_1 \to 1 , U_1 \to 1U_2 , U_1 \to 0\}$

and

$bad_{\Delta_G}(P_H) = P_H \setminus good_{\Delta_G}(P_H)$.

Finally, let J be the cf grammar such that

$\Sigma_J = \{Y_1 , Y_2 , Y_3 , Z_1 , Z_2 , Z_3 , Z_4 , U_1 , U_2 , U_3 , U_4 , 0, 1\}$,

$\Delta_J = \{0, 1\}$, $S_J = Y_1$ and $P_J = P_J^1 \cup P_J^2$, where

$P_J^1 = \{Y_1 \to 1Y_2 , Z_1 \to 1Z_2 , Z_1 \to 1 , U_1 \to 1U_2 , U_1 \to 0\}$ and

P_J^2 consists of the following productions

$Y_2 \to 1U_2Y_3 \mid 0Y_3 \mid 1U_2 \mid 0$,

$Y_3 \to 1Z_2Y_2 \mid 1Y_2$,

$Z_2 \to 1U_2Z_3 \mid 0Z_3 \mid 1U_2Z_4 \mid 0Z_4$,

$Z_3 \to 1Z_2Z_2 \mid 1Z_2$,

$Z_4 \to 1Z_2 \mid 1$,

$U_2 \to 1U_2U_3 \mid 0U_3 \mid 1U_2U_4 \mid 0U_4$,

$U_3 \to 1Z_2U_2 \mid 1U_2$,

$U_4 \to 1U_2 \mid 0$.

It is easily seen that J results from H be applying STEP 3 of Construction 1.1.

Note that J is in 2 GNF and moreover $\#\Theta_J = 11$, $\#P_J = 27$ and $size(J) = 62$.

The cf grammar I used in the proof of Theorem 1.1 looks as follows.

$\Sigma_I = \Sigma_H$, $\Delta_I = \Delta_G$, $S_I = N_{S_G} = Y_1$ and

$P_I = P_H \cup \{A_1 \to Y_1 , A_2 \to Z_1 , A_3 \to U_1\}$. ■

2. ON THE SIZE OF J

In this section we present an algorithm to implement Construction 1.1 and then estimate the size of the resulting cf grammar J (in $2GNF$) in terms of the size of the initially given Λ-free cf grammar G.

Throughout this section we will use the notation introduced in the description of Construction 1.1.

Lemma 2.1. There exists an algorithm which given an arbitrary Λ-free cf grammar G and an arbitrary nonterminal A of G yields a Λ-free, chain-free right linear grammar $G^{(A)}$ such that $L(G^{(A)}) = T_A$ and $size(G^{(A)}) \leq 2\#\Theta_G size(G)$.

Proof.

Let G be an arbitrary Λ-free cf grammar and let $A \in \Sigma_G \backslash \Delta_G$.

Let $M^{(A)}$ be the left linear grammar such that $\Sigma_{M^{(A)}} = \Sigma_G \cup \{B' : B \in \Theta_G\}$, where it is assumed that $\Sigma_G \cap \{B' : B \in \Theta_G\} = \emptyset$, $\Delta_{M^{(A)}} = \Sigma_G$, $S_{M^{(A)}} = A'$ and $P_{M^{(A)}}$ is defined as follows.

(1) For each $\pi = X \to Y_1 \ldots Y_k \in good_{\Delta_G}(P_G)$, where $k \geq 1$ and $Y_j \in \Sigma_G$ for $1 \leq j \leq k$, $P_{M^{(A)}}$ contains the production $X' \to Y_1 Y_2 \cdots Y_k$,

(2) For each $\pi = X \to Y_1 \cdots Y_k \in bad_{\Delta_G}(P_G)$, where $k \geq 1$ and $Y_j \in \Sigma_G$ for $1 \leq j \leq k$, $P_{M^{(A)}}$ contains the production $X' \to Y_1' Y_2 \cdots Y_k$,

(3) $P_{M^{(A)}}$ contains only productions resulting from (1) and (2) above.

Clearly

$L(M^{(A)}) = T_A$, $size(M^{(A)}) = size(G)$ and $\#\Theta_{M^{(A)}} = \#(\Sigma_G \backslash \Delta_G)$.....(2)

Let $F^{(A)}$ be the left linear grammar resulting from $M^{(A)}$ by removing (in the standard way) chain productions from it. Clearly $L(F^{(A)}) = L(M^{(A)})$ and $size(F^{(A)}) \leq \#\Theta_{M^{(A)}} size(M^{(A)})$. Thus from (2) it follows that

$L(F^{(A)}) = T_A$ and $size(F^{(A)}) \leq \#\Theta_G size(G)$.....(3)

Finally let $G^{(A)}$ be the right linear grammar such that $L(G^{(A)}) = L(F^{(A)})$ resulting by applying the standard algorithm for constructing an equivalent right linear grammar for a given left linear grammar (so $G^{(A)}$ simulates $F^{(A)}$ bottom-up). Clearly $size(G^{(A)}) \leq 3size(F^{(A)})$. Hence, by (3) $L(G^{(A)}) = T_A$ and $size(G^{(A)}) \leq 3\#\Theta_G\ size(G)$ and so the lemma holds. ∎

Lemma 2.2. There exists an algorithm which given an arbitrary Λ-free cf grammar G yields a Λ-free, chain-free right linear 3cheme H such that $maxr(H) \leq 2$, $\Delta_H = \Sigma_G$, H covers $\mathbf{T}(G)$ and
$size(H) \leq 9(\#\Theta_G)^2\ size(G)$.

Proof.

Let G be an arbitrary Λ-free cf grammar and let, for each $A \in \Theta_G$, $G^{(A)}$ be as in the statement of Lemma 2.1. We assume that if $A,B \in \Theta_G$ are different, then the sets of nonterminals of $G^{(A)}$ and $G^{(B)}$ are disjoint.

Let $\overline{H}$ be the right linear scheme such that $\Sigma_{\overline{H}} = \bigcup_{A \in \Theta_G} \Sigma_{G^{(A)}}$, $\Delta_{\overline{H}} = \Sigma_G$ and

$P_{\overline{H}} = \bigcup_{A \in \Theta_G} P_{G^{(A)}}$.

Clearly $\overline{H}$ is Λ-free and chain-free, $\Delta_{\overline{H}} = \Sigma_G$ and H covers $\mathbf{T}(G)$.
Also $size(\overline{H}) \leq \#\Theta_G \max \{\ size(G^{(A)}) : A \in \Theta_G\ \}$.
Hence by Lemma 2.1 , $size(\overline{H}) \leq 2(\#\Theta_G)^2\ size(G)$.

Now let H be the right linear scheme resulting from $\overline{H}$ by applying the standard algorithm for getting an equivalent right linear scheme with the length of the right-hand side of any production not exceeding two.

Clearly $size(H) \leq 3size(\overline{H})$.
we get $size(H) \leq 9(\#\Theta_G)^2 size(G)$.

Thus the lemma holds. ∎

Theorem 2.1. There exists an algorithm which implements Construction 1.1 and is such that $size(J) \leq 9^2(\#\mathbb{0}_G)^4(size(G))^2 \leq 9^2(size(G))^6$.

Proof.

This follows directly from Lemma 2.2 and from the description of STEP 3 of Construction 1.1 (obviously $size(J) \leq (size(H))^2$) . ∎

ACKNOWLEDGEMENTS

This research was supported by NSF grant number MCS 83-05245.

REFERENCES

[H] Harrison, M., *Introduction to formal language theory*, Addison-Wesley, Reading, Massachusetts, 1978.

[MSW] Maurer, H., Salomaa, A. and Wood, D., "A supernormal-form theorem for context-free grammars," *Journal of the Association for Computing Machinery*, v. 30, n. 1, 95-102, 1983.

[S] Salomaa, A., *Formal languages*, Academic Press, London-New York, 1973.