REPRESENTATION THEOREMS USING DOS LANGUAGES

by

A. Ehrenfeucht*

and

G. Rozenberg**

CU-CS-161-79 August, 1979

* A. Ehrenfeucht, Department of Computer Science, University of Colorado at Boulder, 80309 USA

** G. Rozenberg, Institute of Applied Mathematics and Computer Science, University of Leiden, 2300 RA Leiden, Holland

All correspondence to G. Rozenberg

ABSTRACT

It is demonstrated that every context-free language is a homomorphic image of the intersection of two DOS languages and that every recursively enumerable language is the homomorphic image of the intersection of three DOS languages. It is also proved that by increasing the number of components in the intersections of DOS languages one gets an infinite hierarchy of classes of languages within the class of context-sensitive languages.

INTRODUCTION

Recently there have appeared a number of papers investigating sentential forms of grammars in the classical Chomsky hierarchy (see, e.g., [BPR], [HP], [MSW] and [S 2]). Clearly such an investigation is needed if one is to fully understand language theory from the "grammatical point of view" (as, for example, opposed to the "machine point of view"). Moreover such a research provides a chance for a systematic build-up of the theory of, e.g., context-free languages. An example of a systematic build-up of a theory is, in our opinion, provided by the mathematical theory of L systems (see, e.g., [RS]), the core of which fits into a very basic mathematical framework. The essential construct of the theory of L systems is a DOL system which is really an iterative homomorphism on a free monoid. In [ER1] a sequential analogue of a DOL system, called a DOS system, was introduced and investigated. We believe that DOS systems can play the same essential role in the theory of context-free languages, that DOL systems play in the theory of L systems. This paper supports our belief. We demonstrate the ability of DOS languages to represent arbitrary context-free languages and arbitrary recursively enumerable languages. Since intersections of DOS languages are essential in those representations, they are also investigated in this paper. It is shown that increasing the number of components in the intersections of DOS languages gives rise to an infinite hierarchy of classes of languages.

We assume the reader to be familiar with the rudiments of formal language theory. We use mostly standard terminology and notation. Perhaps only the following requires an additional explanation.

(1). A *weak identity* is a homomorphism that maps each letter either into itself or into the empty word.

(2). For a word α, *alph* α denotes the set of letters occurring in α and *mir* α denotes the mirror image of α; for a language K, *mir* K = $\{mir\ \alpha : \alpha \in K\}$.

(3). Throughout this paper we consider two languages identical if they differ by the empty word only.

(4). If X is a class of grammars than $L(X)$ denotes the class of all languages generated by grammars in X.

I. DOS SYSTEMS AND LANGUAGES

In this section we recall from [ER1] the definitions of a DOS system and a DOS language.

Definition. Let Σ be a finite alphabet.
A *sequential homomorphism* (abbreviated *s-homomorphism*) *on* Σ^* is a mapping h from Σ^* into 2^{Σ^*} defined inductively as follows:

(1). $h(\Lambda) = \{\Lambda\}$,

(2). for each $b \in \Sigma$ there exists a $\beta \in \Sigma^*$ such that $h(b) = \{\beta\}$,

(3). for each $\alpha \in \Sigma^+$,

$h(\alpha) = \{\alpha_1\beta\alpha_2 : \alpha = \alpha_1 b \alpha_2$ for some $b \in \Sigma$, $\alpha_1, \alpha_2 \in \Sigma^*$ and $h(b) = \{\beta\}\}$.

The s-homomorphism h is extended to 2^{Σ^*} by letting
$h(K) = \bigcup_{\alpha \in K} h(\alpha)$ for each $K \subseteq \Sigma^*$. □

As usual, we assume that an s-homomorphism on Σ^* is given by providing its values for all elements from Σ. To simplify the notation, in the sequel we will often identify a singleton $\{x\}$ with its element x.

Definition. A DOS *system* is a construct $G = (\Sigma, h, \omega)$ where Σ is a finite nonempty alphabet, $\omega \in \Sigma^*$ and h is an s-homomorphism on Σ^*. The *language of* G, denoted L(G), is defined by
$L(G) = \{x : x \in h^n(\omega)$ for some $n \geq 0\}$. L(G) is referred to as a DOS *language*. If G is such that for no $a \in \Sigma$, $h(a) = \Lambda$ then we call G *propagating* and refer to it as a PDOS *system* (and we refer to L(G) as a PDOS *language*). □

Remark.

(1). As customary in language theory, whenever $h(a) = \alpha$ for $a \in \Sigma$ then we refer to (a, α) as a *production* of G and write it in the form $a \to \alpha$.

Also, if for $x,y \in \Sigma^*$ and $n \geq 0$, we have $y \in h^n(x)$, then we say that x *derives* y (*in* G) *in* n *steps*.

(2). Clearly, each DOS language is generated by a *reduced* DOS system, that is by a DOS system $G = (\Sigma, h, \omega)$ such that each letter from Σ appears in at least one word of L(G). In the sequel we will consider reduced DOS systems only. □

Example. Let $G = (\{a,b,c\},h,a)$ be the DOS system where $h(a) = bc$, $h(b) = b^2$ and $h(c) = cb$. Then bc derives b^3cb^2 and $L(G) = \{a\} \cup \{b^m cb^n : m \geq 1, n \geq 0\}$. □

Example. In [ER1] a theorem is given (Theorem 8) allowing one to provide various examples of languages that are not DOS languages. Thus, for example:

(1). There exist finite languages that are not DOS languages; $\{a^2, b^2\}$ is an example of such a language.

(2). Dyck languages over more than one sort of parenthesis are not DOS languages. □

II. A COMBINATORIAL RESULT

In this section we present a combinatorial result that will be very essential in the proof of the representation theorem for context-free languages presented in the next section. The proof of the combinatorial result presented in this section is based on the following construction.

CONSTRUCTION 1.

Let π be a permutation on the set $\{1,\dots,n\}$, $n \geq 2$, and let $\tau = \tau_1\tau_2\dots\tau_n$ be a sequence of all pairs from $\{1,\dots,n\} \times \{1,\dots,n\}$ describing π.

Let us consider the set $Z = \{X_i, \bar{X}_i, Y_i, \bar{Y}_i : 1 \leq i \leq n\}$ and let h be the mapping from $Z \times \{\tau_1,\dots,\tau_n\}$ into the set of $\{0,1\}$- sequences of length four defined as follows:

for $1 \leq i \leq n$, $1 \leq k \leq n$,

$$h(X_i,\tau_k) = \begin{cases} 1100 & \text{if } \tau_k = (i,j) \text{ for some } 1 \leq j \leq n, \\ 0000 & \text{otherwise,} \end{cases}$$

$$h(\bar{X}_i,\tau_k) = \begin{cases} 1010 & \text{if } \tau_k = (i,j) \text{ for some } 1 \leq j \leq n, \\ 0000 & \text{otherwise,} \end{cases}$$

$$h(Y_i,\tau_k) = \begin{cases} 0011 & \text{if } \tau_k = (j,i) \text{ for some } 1 \leq j \leq n, \\ 0000 & \text{otherwise,} \end{cases}$$

$$h(\bar{Y}_i,\tau_k) = \begin{cases} 0101 & \text{if } \tau_k = (j,i) \text{ for some } 1 \leq j \leq n, \\ 0000 & \text{otherwise.} \end{cases}$$

Let h_τ be the function from Z into the set of sequences of length 4n over $\{0,1\}$ defined as follows:

for $a \in Z$,

$h_\tau(a) = h(a,\tau_1)h(a,\tau_2)\ldots h(a,\tau_n)$.

The following property of the above construction is very essential for our applications.

Lemma 1. Let $r > 2$. For each $a \in Z$ consider $h_\tau(a)$ as a number written in base r with the rightmost character of $h_\tau(a)$ being the least significant digit of $h_\tau(a)$. Then the following holds:

(1). for $i_1, i_2, j_1, j_2 \in \{1,\ldots,n\}$,

$h_\tau(X_{i_1}) + h_\tau(Y_{j_1}) = h_\tau(\bar{X}_{i_2}) + h_\tau(\bar{Y}_{j_2})$

if and only if

$i_1 = i_2$, $j_1 = j_2$ and $j_1 = \pi(i_1)$, and

(2). for $i \in \{1,\ldots,n\}$, $h_\tau(\bar{X}_i) \neq h_\tau(X_i)$ and $h_\tau(Y_i) \neq h_\tau(\bar{Y}_i)$.

Proof.

(i). Assume that $h_\tau(X_{i_1}) + h_\tau(Y_{j_1}) = h_\tau(\bar{X}_{i_2}) + h_\tau(\bar{Y}_{j_2})$. Consider $p = h_\tau(X_{i_1}) + h_\tau(Y_{j_1})$. Assume that $s \in \{1,\ldots,n\}$ is such that $\tau_s = (i_1,\pi(i_1))$. Then the $4(s-1) + 1$ element of p (counted from the left) is 1. Since $p = q$, where $q = h_\tau(\bar{X}_{i_2}) + h_\tau(\bar{Y}_{j_2})$, $i_2 = i_1$. But also the $4(s-1) + 2$ element of p is 1 and so $j_2 = \pi(i_1)$. Then however, for every $1 \leq t \leq 4$, the $4(s-1) + t$ element in q is 1 and so (because $p = q$) $j_1 = \pi(i_1)$. Thus $j_1 = j_2$ and so $X_{i_1} + Y_{j_1} = \bar{X}_{i_2} + \bar{Y}_{j_2}$ implies that $i_1 = i_2$, $j_1 = j_2$ and $j_2 = \pi(i_1)$.

(ii). It follows directly from the construction that $i_1 = i_2$, $j_1 = j_2$ and $j_1 = \pi(i_1)$ implies that $h_\tau(X_{i_1}) + h_\tau(X_{j_1}) = h_\tau(\bar{X}_{i_2}) + h_\tau(\bar{Y}_{j_2})$.

Note that (2) follows immediately from the construction used, hence the lemma holds. □

Let $\bar{h}_\tau$ be the function from Z into the set of sequences of length $4n+1$ over the alphabet $\{0,1,\ldots,n\}$ defined as follows:

for $1 \le i \le n$ and $a \in \{X,\bar{X},Y,\bar{Y}\}$,

$\bar{h}_\tau(a_i) = ih_\tau(a_i)$.

The $\bar{h}_\tau$ function satisfies the following property.

Lemma 2. Let $r > n$. For each $a \in Z$ consider $\bar{h}_\tau(a)$ as a number written in base r with the rightmost element of $\bar{h}_\tau(a)$ being the least significant digit of $\bar{h}_\tau(a)$. Then the following holds:

(1). Each of the sequences

$\bar{h}_\tau(X_1),\ \bar{h}_\tau(X_2),\ \ldots,\ \bar{h}_\tau(X_n)$;

$\bar{h}_\tau(\bar{X}_1),\ \bar{h}_\tau(\bar{X}_2),\ \ldots,\ \bar{h}_\tau(\bar{X}_n)$;

$\bar{h}_\tau(Y_1),\ \bar{h}_\tau(Y_2),\ \ldots,\ \bar{h}_\tau(Y_n)$; and

$\bar{h}_\tau(\bar{Y}_1),\ \bar{h}_\tau(\bar{Y}_2),\ \ldots,\ \bar{h}_\tau(\bar{Y}_n)$

is strictly growing,

(2). for $i_1,\ i_2,\ j_1,\ j_2 \in \{1,\ldots,n\}$,

$\bar{h}_\tau(X_{i_1}) + \bar{h}_\tau(Y_{j_1}) = \bar{h}_\tau(\bar{X}_{i_2}) + \bar{h}_\tau(\bar{Y}_{j_2})$

if and only if

$i_1 = i_2$, $j_1 = j_2$ and $j_1 = \pi(i_1)$, and

(3). for $i \in \{1,\ldots,n\}$,

$\bar{h}_\tau(X_i) \ne \bar{h}_\tau(\bar{X}_i)$ and $\bar{h}_\tau(Y_i) \ne \bar{h}_\tau(\bar{Y}_i)$.

Proof.

Directly from the definition of $\bar{h}_\tau$ and Lemma 1. □

Now we get our basic combinatorial result on sequences of positive integers "satisfying a given permutation".

Theorem 1. For every $n \geq 2$ and every permutation π on $\{1,...,n\}$ there exist four strictly decreasing sequences of positive integers $X_1, ..., X_n$; $Y_1, ..., Y_n$, $\bar{X}_1, ..., \bar{X}_n$ and $\bar{Y}_1, ..., \bar{Y}_n$

such that:

(1). for $i_1, i_2, j_1, j_2 \in \{1,...,n\}$,

$X_{i_1} + Y_{j_1} = \bar{X}_{i_2} + \bar{Y}_{j_2}$ if and only if

$i_1 = i_2$, $j_1 = j_2$ and $\pi(i_1) = j_1$, and

(2). for $i \in \{1,...,n\}$,

$X_i \neq \bar{X}_i$ and $Y_i \neq \bar{Y}_i$.

Proof.

Directly from Lemma 2. □

III. REPRESENTING CONTEXT-FREE LANGUAGES

In this section we demonstrate that every context-free language K is of the form $K = \phi(M_1 \cap M_2)$ where ϕ is a weak identity and M_1, M_2 are DOS languages.

Theorem 1 from the last section will be an essential tool in the proof of the above mentioned result. We will use it for the following permutation π_n on the set $\{1,\ldots,n\}$ (where $n \geq 2$):

for $1 \leq i \leq n$,

$$\pi_n(i) = \begin{cases} 1 & \text{if } i = 2, \\ i-2 & \text{if } i \text{ is even and } i \neq 2, \\ i+2 & \text{if } i \text{ is odd, } i+2 \leq n \\ n-1 & \text{if } i = n, \ n \text{ odd}, \\ n & \text{if } i = n-1, \ n \text{ even}. \end{cases}$$

Then let $X_1, \ldots, X_n$; $Y_1, \ldots, Y_n$; $\bar{X}_1, \ldots, \bar{X}_n$ and $\bar{Y}_1, \ldots, \bar{Y}_n$ be four fixed strictly decreasing sequences of positive integers (associated with π_n) satisfying the statement of Theorem 1.

Using those sequences we will define now the basic tool for proving the main result of this section: a *blocking pair of* DOS *systems*.

CONSTRUCTION 2

Let $m \geq 1$ and let $n = 2m+1$. Let $A_b, A_1, \ldots, A_m, A_e$ be distinct letters and let $\omega = A_bA_1\ldots A_mA_e$. Let

$\Sigma_\omega = \Sigma_{\omega,1} \cup \Sigma_{\omega,2} \cup \Sigma_{\omega,3} \cup \Sigma_{\omega,4}$ where

$\Sigma_{\omega,1} = \{A_b, A_1, \ldots, A_m, A_e\}$,

$\Sigma_{\omega,2} = \{B_{i,j} : 1 \leq i \leq m, 1 \leq j \leq n\} \cup \{B_{b,j} : 2 \leq j \leq n-1, j \text{ even}\} \cup$

$\cup \{B_{e,j} : 3 \leq j \leq n, j \text{ odd}\}$,

$\Sigma_{\omega,3} = \{C_i : 1 \le i \le m\}$,

$\Sigma_{\omega,4} = \{¢\}$

and $\Sigma_{\omega,i}$, $1 \le i \le 4$, are mutually disjoint.

Let h_ω be the s-homomorphism on Σ_ω^* defined as follows:

$h_\omega(¢) = ¢$,

$h_\omega(A_i) = ¢^{Y_n} B_{i,n} ¢^{X_n}$ for $1 \le i \le m$,

$h_\omega(A_b) = B_{b,n-1} ¢^{X_{n-1}}$,

$h_\omega(A_e) = ¢^{Y_n} B_{e,n}$,

$h_\omega(B_{b,2}) = B_{b,2}$,

$h_\omega(B_{b,j}) = B_{b,j-2} ¢^{X_{j-2}-X_j}$ for $j \in \{4,\ldots,n-1\}$,

$h_\omega(B_{e,3}) = B_{e,3}$,

$h_\omega(B_{e,j}) = ¢^{Y_{j-2}-Y_j} B_{e,j-2}$ for $j \in \{5,\ldots,n\}$,

$h_\omega(B_{i,j}) = ¢^{Y_{j-1}-Y_j} B_{i,j-1} ¢^{X_{j-1}-X_j}$ for $j \in \{2,\ldots,n\}$, $i \in \{1,\ldots,m\}$,

$h_\omega(B_{i,2}) = ¢^{Y_1-Y_2} B_{i,1} ¢^{X_1-X_2} A_{i+1} A_{i+2} \ldots A_m A_e C_i A_b A_1 B_2 \ldots$

$\ldots A_{i-1} ¢^{Y_1} B_{i,1} ¢^{X_1-X_2}$ for $i \in \{1,\ldots,m\}$,

$h_\omega(B_{i,1}) = B_{i,1}$ for $i \in \{1,\ldots,m\}$, and

$h_\omega(C_i) = C_i$ for $i \in \{1,\ldots,m\}$.

Let $\bar{h}_\omega$ be the homomorphism on Σ_ω^* defined in the same way as h_ω, except that everywhere X_i is replaced by $\bar{X}_i$ and Y_i is replaced by $\bar{Y}_i$ for $1 \leq i \leq n$.

Let $G_\omega = (\Sigma_\omega, h_\omega, \omega)$ and $\bar{G}_\omega = (\Sigma_\omega, \bar{h}_\omega, \omega)$.

Let us consider $L(G_\omega) \cap L(\bar{G}_\omega)$.

(i). Obviously $\omega \in L(G_\omega) \cap L(\bar{G}_\omega)$.

(ii). If a word $\alpha \in L(G_\omega) \cap L(\bar{G}_\omega)$ and $\alpha \neq \omega$ then α does not contain any occurrence of a letter from $\Sigma_{\omega,1}$.

This is so, because then α must be either of the form
$\alpha_1 A_j ¢^{U_i} B_{r,s} \alpha_2 = \alpha_1 A_j ¢^{\bar{U}_i} B_{r,s} \alpha_2$ or of the form
$\alpha_1 B_{r,s} ¢^{U_i} A_j \alpha_2 = \alpha_1 B_{r,s} ¢^{\bar{U}_i} A_j \alpha_2$ for some words α_1, α_2 and $j,r \in \{b,e,1,\ldots,m\}$ $i,s \in \{1,\ldots,n\}$, $U \in \{X,Y\}$, which is impossible because for every $i \in \{1,\ldots,n\}$, $X_i \neq \bar{X}_i$ and $Y_i \neq \bar{Y}_i$.

(iii). If a word $\alpha \in L(G_\omega) \cap L(\bar{G}_\omega)$ and $\alpha \neq \omega$ then α is of the form $\gamma_i C_i \gamma_i$ for some $i \in \{1,\ldots,m\}$, where

$$\gamma_i = B_{b,r} ¢^{X_r + Y_{r-2}} B_{1,r-2} \ldots B_{i-2,4} ¢^{X_4 + Y_2} B_{i-1,2} ¢^{X_2 + Y_1} B_{i,1} ¢^{X_1 + Y_3}$$
$$B_{i+1,3} ¢^{X_3 + Y_5} B_{i+2,5} \ldots ¢^{X_{s-2} + Y_s} B_{e,s}$$

where $r,s \in \{2,\ldots,n\}$, r is even and s is odd.

This is seen as follows.

From (ii) it follows that α does not contain occurrences from $\Sigma_{\omega,1}$. Let us inspect α from left to right. It must be of the form

$$\alpha = B_{b,j_0} ¢^{X_{j_0}+Y_{j_1}} B_{1,j_1} \alpha_1 =$$

$$= B_{b,j_0} ¢^{\bar{X}_{j_0}+\bar{Y}_{j_1}} B_{1,j_1} \alpha_1 ,$$

for $j_0, j_1 \in \{1,..,n\}$, j_0 even and α_1 a word.

Consequently $j_1 = \pi_n(j_0)$. Thus we can write

$$\alpha = B_{b,j_0} ¢^{X_{j_0}+Y_{j_1}} B_{1,j_1} ¢^{X_{j_1}+Y_{j_2}} B_{2,j_2} \alpha_2 =$$

$$= B_{b,j_0} ¢^{\bar{X}_{j_0}+\bar{Y}_{j_1}} B_{1,j_1} ¢^{\bar{X}_{j_1}+\bar{Y}_{j_2}} B_{2,j_2} \alpha_2$$

for $j_2 \in \{1,...,n\}$ and α_2 a word.

Consequently $j_2 = \pi_n(j_1)$.

And so on . . . .

Thus $j_1 = \pi_n(j_0)$, $j_2 = \pi_n(j_1)$, $j_3 = \pi_n(j_2)$, ...

Hence according to the definition of the permutation π_n, the sequence j_1, j_2, j_3, ... is a sequence of positive integers descending (according to π_n) until we get into an i such that $j_i = 1$. This must happen because $n = 2m+1$. But that means that the production for $B_{i,2}$ must have been used, so that α must have the alleged form. Note that for no other $\bar{i} \neq i$ the production for $B_{\bar{i},2}$ could be used, because the form of the permutation π_n implies that the consecutive second indices of letters from $\Sigma_{\omega,2}$ in α to the right of $B_{i,1}$ ascend through odd numbers 3, 5, 7, ... and so until we meet C_i no element of the form $B_{\bar{i},1}$ for $\bar{i} \neq i$ can occur. However the form of the production for $B_{i,2}$ (and the form of π_n) implies that if $\alpha = \gamma C_i \delta$ then $\gamma = \delta$ and so (iii) holds.

(iv). If $\alpha \in L(G_\omega) \cap L(\bar{G}_\omega)$, $\alpha \neq \omega$ and an occurrence of a letter from $\Sigma_\omega \backslash (\Sigma_{\omega,3} \cup \Sigma_{\omega,4} \cup \{B_{b,1}, B_{e,3}\})$ in α is rewritten, then the resulting word is not in $L(G_\omega) \cap L(\bar{G}_\omega)$.

This follows from the form of productions in G_ω and $\bar{G}_\omega$ and from the observation made in the proof of (iii): if δ is a subword of a word in $L(G_\omega) \cap L(\bar{G}_\omega)$ and δ does not contain an element from $\Sigma_{\omega,3}$ then it can have at most one occurrence of a letter b from $\Sigma_{\omega,2}$ such that $b = B_{i,j}$ for $j = 1$.

It is because of properties (iii) and (iv) above that we call the pair $(G_\omega, \bar{G}_\omega)$ the *blocking pair*. If we want to get a word in $L(G_\omega) \cap L(\bar{G}_\omega)$ then in both G_ω and $\bar{G}_\omega$ only one but arbitrary letter of type B (that is a letter from $\Sigma_{\omega,2}$) which is not a B_b- or B_e- type can be "completely rewritten" (the same letter in G_ω and $\bar{G}_\omega$) yielding a letter of type C (that is a letter from $\Sigma_{\omega,3}$). All other elements of ω are prevented (blocked) from being completely rewritten.

We are ready now to prove the main result of this section.

Theorem 2. For every context-free language K there exist a weak identity ϕ and DOS languages M_1, M_2 such that $K = \phi(M_1 \cap M_2)$.

Proof.

Let K be a context-free language.

Let $G = (V_N, V_T, P, S)$ be a context-free grammar with $V = V_N \cup V_T$ such that $L(G) = K$.

Let for each nonterminal a in V_N, $\rho_a = \mu_{a,1}, \mu_{a,2}, \ldots, \mu_{a,m_a}$ be an ordered sequence of all right-hand sides of productions for a in G.

Let for each a in V_N, $\omega_a = A_b^{(a)} A_1^{(a)} \ldots A_{m_a}^{(a)} A_e^{(a)}$.

Let f be the homomorphism on V^* defined as follows:

for $a \in V$,

$$f(a) = \begin{cases} a & \text{if } a \in V_T, \\ \omega_a & \text{if } a \in V_N, \end{cases}$$

and let $\bar{f}$ be the homomorphism on V^* defined as follows:

for $a \in V$,

$$\bar{f}(a) = \begin{cases} a & \text{if } a \in V_T, \\ \bar{\omega}_a & \text{if } a \in V_N, \end{cases}$$

where $\bar{\omega}_a = \bar{A}_b^{(a)}\bar{A}_1^{(a)}\ldots\bar{A}_{m_a}^{(a)}\bar{A}_e^{(a)}$ with

$\{A_b^{(a)},A_1^{(a)},\ldots,A_e^{(a)}\} \cap \{\bar{A}_b^{(a)},\bar{A}_1^{(a)},\ldots,\bar{A}_e^{(a)}\} = \emptyset$.

Then let $[G_{\omega_a}]$ and $[\bar{G}_{\omega_a}]$ be the DOS systems constructed in the same way that G_{ω_a} and $\bar{G}_{\omega_a}$ are constructed in CONSTRUCTION 2 except that only the following changes are made:

(1). ω_a is the axiom of $[G_{\omega_a}]$ and $\bar{\omega}_a$ is the axiom of $[\bar{G}_{\omega_a}]$,

(2). $[h_\omega](C_i) = f(\mu_{a,i})$, $[\bar{h}_\omega](C_i) = \bar{f}(\mu_{a,i})$,

(3). both $[h_\omega]$ and $[\bar{h}_\omega]$ have only identity productions for symbols from V_T,

(4) $[h_\omega](A_x^{(a)}) = h_\omega(A_x^{(a)})$,

$[\bar{h}_\omega](\bar{A}_x^{(a)}) = \bar{h}_\omega(A_x^{(a)})$ for $x \in \{1,\ldots,m_a,b,e\}$,

where $[h_\omega]$ is the s-homomorphism of $[G_{\omega_a}]$ and $[\bar{h}_\omega]$ is the s-homomorphism of $[\bar{G}_{\omega_a}]$.

Now we consider all pairs $([G_{\omega_a}],[\bar{G}_{\omega_a}])$, $a \in V_N$, and we take care that in two different pairs the alphabets involved are disjoint, except for symbols from V_T.

Then let $H_1 = (\Theta_1, g_1, \zeta_1)$ $H_2 = (\Theta_2, g_2, \zeta_2)$ where $\zeta_1 = f(S)$, $\zeta_2 = \bar{f}(S)$,

Θ_1 is the union of alphabets of all $[G_{\omega_a}]$, $a \in V_N$,

Θ_2 is the union of alphabets of all $[\bar{G}_{\omega_a}]$, $a \in V_N$,

g_1 is the union of all s-homomorphisms $[h_{\omega_a}]$, $a \in V_N$, and

g_2 is the union of all s-homomorphisms $[\bar{h}_{\omega_a}]$, $a \in V_N$.

Let ϕ be the weak identity on $(\Theta_1 \cup \Theta_2)^*$ that erases all letters except for letters from V_T.

Rather than to provide a formal and rather tedious proof that $L(G) = \phi(L(H_1) \cap L(H_2))$ we give some intuition of how a derivation step in G is simulated by $L(H_1) \cap L(H_2)$.

First of all every nonterminal a is coded as a block $\omega_a = A_b^{(a)} A_1^{(a)} \ldots A_{m_a}^{(a)} A_e^{(a)}$ in H_1 and as $\bar{\omega}_a = \bar{A}_b^{(a)} \bar{A}_1^{(a)} \ldots \bar{A}_{m_a}^{(a)} \bar{A}_e^{(a)}$ in H_2 where m_a is the number of different productions for a in G. The intention is that if one rewrites an occurrence of a in a sentential form α in G by its i-th production, then the corresponding occurrence of ω_a in the corresponding sentential form β in H_1 and the corresponding occurrence of $\bar{\omega}_a$ in the corresponding sentential form $\bar{\beta}$ in H_2 are rewritten in such a way that in a number of steps it leads to the subword $\gamma_i C_i \gamma_i$ in H_1 and in H_2, where γ_i is of the form described under (iii) in CONSTRUCTION 2. Then in H_1 this occurrence of C_i is rewritten by $(f(\mu_{a,i})$ and in H_2 this occurrence of C_i is rewritten by $\bar{f}(\mu_{a,i})$, where $\mu_{a,i}$ is the right-hand side of the i'th production for a.

Hence the single rewriting step of an occurrence of a into $\mu_{a,i}$ was simulated in a number of steps, by the pair of DOS systems $[G_{\omega_a}]$ and $[\bar{G}_{\omega_a}]$ acting on the corresponding occurrence of ω_a in the corresponding sentential form in H_1 and the corresponding occurrence of $\bar{\omega}_a$ in the corresponding sentential form in H_2, respectively. As the result of this simulation the given occurrence of ω_a and the given occurrence of $\bar{\omega}_a$, respectively, give rise to blocks $f(\mu_{a,i})$ and $\bar{f}(\mu_{a,i})$ respectively interspersed by subwords consisting of symbols ¢ and $B_{i,j}$ only. Those symbols are distributed in such a way as to prevent the rewriting of any of them subsequently (otherwise one will never get a word which is in $L(H_1) \cap L(H_2)$).

In this way, although one symbol, say A, in V_N is coded by a block of symbols in H_1 and in H_2, care is taken that only one symbol of this block leads to a rewrite that codes a rewrite of A in G.

Finally, ϕ takes care of erasing all those auxiliary symbols (that is symbols different from terminal symbols of G).

Hence, if we set $M_1 = L(H_1)$ and $M_2 = L(H_2)$ the theorem holds. □

Remark. Notice that the DOS systems resulting from the construction of the proof of Theorem 2 are propagating so that M_1 and M_2 in the statement of Theorem 2 can be taken to be PDOS languages. □

IV. REPRESENTING RECURSIVELY ENUMERABLE LANGUAGES

In this section we demonstrate that every recursively enumerable language K is of the form $K = \mu(M_1 \cap M_2 \cap M_3)$ where μ is a weak identity and M_1, M_2, M_3 are DOS languages.

Theorem 3. For every recursively enumerable language L there exist DOS languages M_1, M_2, M_3 and a weak identity μ such that $L = \mu(M_1 \cap M_2 \cap M_3)$.

Proof.

It is well known (see, e.g., [S1]) that for every recursively enumerable language L there exist a weak identity ψ and context-free languages L_1, L_2 such that $L = \psi(L_1 \cap L_2)$.

Let Δ_1 be the alphabet of L_1, Δ_2 be the alphabet of L_2 and let $\Delta_2' = \{a' : a \in \Delta_2\}$ where $(\Delta_1 \cup \Delta_2) \cap \Delta_2' = \emptyset$. Let L_2' be the language resulting from L_2 by replacing every occurrence of a letter a from Δ_2 in L_2 by a' from Δ_2'.

Let $K = L_1 \mathit{mir} L_2'$. Clearly K is a context-free language. Let $G = (V_N, V_T, P, S)$ be a context-free grammar generating K. Then let us use the construction from the proof of Theorem 2 which yields DOS systems $H_1 = (\Theta_1, g_1, \zeta_1)$, $H_2 = (\Theta_2, g_2, \zeta_2)$ and a weak identity ϕ such that $K = \phi(L(H_1) \cap L(H_2))$. Let $\Theta = (\Theta_1 \cup \Theta_2) \setminus V_T$, $\Theta = \{b_1, \ldots, b_r\}$, $\Theta_1 = \mathit{alph}(L_1 \cap L_2) = \{c_1, \ldots, c_s\}$, $\Theta_1' = \{c_1', \ldots, c_s'\}$ and $m = r + s$. Let Δ be a new alphabet, $\Delta = \{F_1, \ldots, F_m\}$ and let $\delta = F_1 \ldots F_m$.

Let $H = (\Theta \cup \Delta, g, \delta)$ be the DOS system where the s-homomorphism g is defined by:

$$g(F_i) = F_i F_{i+1} \ldots F_m b_i F_1 F_2 \ldots F_i \qquad \text{for } 1 \le i \le r,$$

$$g(F_{r+j}) = F_{r+j} F_{r+j+1} \ldots F_m c_j \, \delta \, c_j' F_1 F_2 \ldots F_{r+j} \qquad \text{for } 1 \le j \le s,$$

$g(\alpha) = \alpha$ for $\alpha \in \Theta$.

Let τ_δ be the mapping on $(\Theta_1 \cup \Theta_2)^*$ defined by:

$\tau_\delta(\Lambda) = \delta$, and

$\tau_\delta(a_1 a_2 \ldots a_m) = \delta a_1 \delta a_2 \delta \ldots \delta a_m \delta$ for $m \geq 1$, $a_1, \ldots, a_m \in \Theta_1 \cup \Theta_2$.

Then for $K \subseteq (\Theta_1 \cup \Theta_2)^*$ let

$$\tau_\delta(K) = \bigcup_{\alpha \in K} \tau_\delta(\alpha).$$

Obviously $\tau_\delta(L(H_1))$ and $\tau_\delta(L(H_2))$ are DOS languages.

Let $\hat{\mu}$ be the weak identity on $(\Theta \cup \Delta)^*$ defined by

$$\hat{\mu}(a) = \begin{cases} a & \text{if } a \in V_T \cup V_T' \text{ where } V_T' = \{a' : a \in V_T\} \\ \Lambda & \text{otherwise.} \end{cases}$$

It is rather easy to see that

$\hat{K} = \hat{\mu}(\tau_\delta(L(H_1)) \cap \tau_\delta(L(H_2)) \cap L(H)) = \{x\, mir\, x' : x \in L_1 \cap L_2\}$.

The key observation here is that if a word α in $L(H)$ is also in $\tau_\delta(L(H_1)) \cap \tau_\delta(L(H_2))$ then in its derivation in $L(H)$ each production introducing an element from Θ_1 (and so also its primed companion from Θ_1') is used in such a way that never to the right of an occurrence of an element from Θ_1' is there an element from Θ_1 and never to the left of an element from Θ_1 is there an element from Θ_1'. Consequently $\hat{\mu}(\alpha) = \beta\, mir\, \beta'$ where $\beta \in L_1 \cap L_2$.

Now let μ be the weak identity obtained from $\hat{\mu}$ by changing $\hat{\mu}$ in such a way that it also erases letters from V_T' and erases the letters that ψ erases.

Then obviously

$L = \mu(K) = \mu(\tau_\delta(L(H_1)) \cap \tau_\delta(L(H_2)) \cap L(H))$

and so if we set $M_1 = \tau_\delta(L(H_1))$, $M_2 = \tau_\delta(L(H_2))$ and $M_3 = L(H)$, the theorem holds. □

Remark.

Note that from the proof of Theorem 3 it follows that DOS languages M_1, M_2, M_3 from the statement of Theorem 3 can be taken to be PDOS languages. □

Coming back to Theorem 2 we notice that the class of languages of the form $\phi(M_1 \cap M_2)$ where ϕ is a weak identity and M_1, M_2 are DOS languages is larger than the class of context free languages, as shown by the following example.

Example. Let $G_1 = (\Sigma, h_1, \omega)$, $G_2 = (\Sigma, h_2, \omega)$ be DOS systems where $\Sigma = \{a,b,c,A,B\}$, $h_1(a) = a$, $h_1(b) = b$, $h_1(c) = c$, $h_1(A) = aAb$, $h_1(B) = Bc$, $h_2(a) = a$, $h_2(b) = b$, $h_2(c) = c$, $h_2(A) = aA$, $h_2(B) = bBc$ and $\omega = aAbBc$.
Then $L(G_1) \cap L(G_2) = \{a^nAb^nBc^n : n \geq 1\}$ - a well known example of a language that is not context-free. □

We do not know whether the class of languages of the form $\phi(M_1 \cap M_2)$, where ϕ is a weak identity and M_1, M_2 are DOS languages, forms a subclass of the class of recursive languages. However we can show that if it is the case then such an inclusion cannot be effective in the following sense.

Theorem 4. Let C be an effective enumeration of a recursive subclass of the class of recursive languages. There does not exist a total recursive function f such that, given a weak identity ϕ and DOS systems G_1, G_2, $f(\phi, G_1, G_2) = n$ where n is the index of $\phi(L(G_1) \cap L(G_2))$ in C.

Proof.

Let $\bar{G}_1$, $\bar{G}_2$ be two arbitrary DOS systems. We catenate to their axioms a new letter ¢ at the right end and then augment productions in $\bar{G}_1$ and $\bar{G}_2$ by $¢ \to ¢$. Let G_1, G_2 be systems obtained (effectively) in this way.

Let ϕ be the weak identity on the intersection of alphabets of G_1 and G_2 which erases all letters except for ¢.

Then clearly we have

$¢ \in \phi(L(G_1) \cap L(G_2))$ if and only if $L(\bar{G}_1) \cap L(\bar{G}_2) \neq \emptyset$.

Since it was proved in [ER2] that it is undecidable whether or not $L(\bar{G}_1) \cap L(\bar{G}_2) = \emptyset$ for arbitrary DOS systems $\bar{G}_1$, $\bar{G}_2$, the above property implies the theorem. □

IV. ON INTERSECTIONS OF DOS LANGUAGES

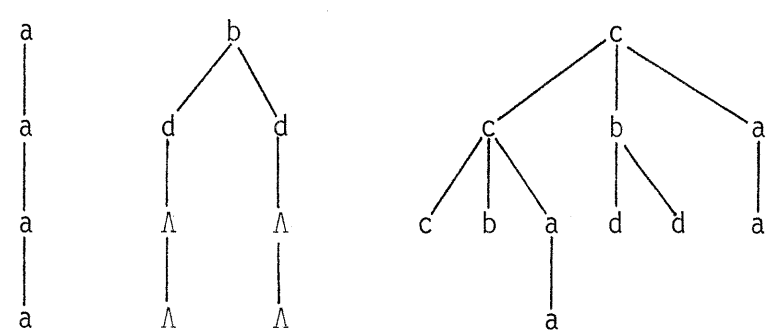
The results of the last two sections indicate that the class of languages consisting of intersections of (several) DOS languages is worth investigating. In particular a natural question arises: is the class of interesections of n DOS languages, denoted by $\bigcap_n \mathcal{L}(\mathrm{DOS})$, larger than the class of intersections of (n-1) DOS languages for every $n \geq 2$. In this section we will show that the answer to the above question is affirmative, and moreover for every $n \geq 2$ there exists a finite language in the difference $\bigcap_n \mathcal{L}(\mathrm{DOS}) \setminus \bigcap_{n-1} \mathcal{L}(\mathrm{DOS})$.

First we need some notions concerning DOS systems.

We start by recalling from [ER1] the notion of the derivation forest T_G of a DOS system G. For the purpose of this section it is best explained informally by an example.

Let $G = (\{a,b,c,d\},h,abc)$ be the DOS system with $h(a) = a$, $h(b) = d^2$, $h(c) = cba$ and $h(d) = \Lambda$. Then T_G is an infinite forest with the following being an initial subforest of it:

the origin of T_G →



A *path* in T_G is an infinite path starting in one of the nodes of the origin of T_G. A *cut* in T_G is a sequence τ of nodes of T_G such that on each path of T_G there is precisely one node from τ. It is easily seen that x is a word in L(G) if and only if it corresponds to (the sequence of labels of) a cut in T_G.

Also we call a letter a in a DOS system $G = (\Sigma,h,\omega)$ *propagating* if for no positive integer r, $h^r(a) = \Lambda$; otherwise a is called *erasing*. We use *pr* G and *er* G to denote the set of propagating letters in G and the set of erasing letters in G respectively.

The following lemma will be useful in our proofs of the following two theorems.

Lemma 3. Let $G = (\Sigma,h,\omega)$ be a DOS system. If

(1). for every $a \in \Sigma$, there exists a positive integer s such that $a^s \in L(G)$, and

(2). there exists a letter $a \in \Sigma$ such that a is propagating,

then every letter in Σ is propagating.

Proof.

From (2) it follows that, for every $\alpha \in L(G)$, we have $(\mathit{alph}\ \alpha) \cap (\mathit{pr}\ G) \neq \emptyset$. Then (1) implies that every letter in Σ is propagating. □

We show now that increasing the number of components in the intersections of DOS languages leads to an infinite hierarchy of classes of languages.

Theorem 5. For every $n \geq 2$ there exists a finite language K_n such that $K_n \in \bigcap_n L(DOS)$ and $K_n \notin \bigcap_{n-1} L(DOS)$.

Proof.

Let $n \geq 2$, $\Sigma_n = \{A_1,\ldots,A_n\}$ and let

$K_n = \{a_1 \ldots a_r : r \geq 0, a_1,\ldots,a_r \in \Sigma_n$ and $a_i \neq a_j$ for $i \neq j$, $1 \leq i, j \leq n\}$.

(i). We will demonstrate now that $K_n \in \bigcap_n \mathcal{L}(DOS)$.

To this aim let

$G_1 = (\Sigma_n, h_1, \omega_1)$ where $\omega_1 = A_2^{n-1} A_1 A_2^{n-1}$ and

$h_1(A_1) = \Lambda, h_1(A_2) = A_3, h_1(A_3) = A_4, \dots, h_1(A_{n-1}) = A_n, h_1(A_n) = \Lambda$,

$G_n = (\Sigma_n, h_n, \omega_n)$ where $\omega_n = A_1^{n-1} A_n A_1^{n-1}$ and

$h_n(A_1) = A_2, h_n(A_2) = A_3, \dots, h_n(A_{n-1}) = \Lambda, h_n(A_n) = \Lambda$,

for $2 \leq i \leq n-1$,

$G_i = (\Sigma_n, h_i, \omega_i)$ where $\omega_i = A_{i+1}^{n-1} A_i A_{i+1}^{n-1}$ and

$h_i(A_1) = A_2, h_i(A_2) = A_3, \dots, h_i(A_{i-2}) = A_{i-1}, h_i(A_{i-1}) = \Lambda, h_i(A_i) = \Lambda$,

$h_i(A_{i+1}) = A_{i+2}, h_i(A_{i+2}) = A_{i+3}, \dots, h_i(A_n) = A_1$.

Note that, for $i \in \{1, \dots, n\}$,

$L(G_i) = \{a_1 \dots a_k : a_1, \dots, a_k \in \Sigma_n, k \leq 2n-1$ and A_i occurs at most once in $a_1 \dots a_k\}$.

It is easy to see that $K_n \subseteq \bigcap_{i=1}^{n} L(G_i)$. On the other hand, for each $i \in \{1, \dots, n\}$, A_i has at most one occurrence in each word of $L(G_i)$. Consequently $\bigcap_{i=1}^{n} L(G_i) \subseteq K_n$. Thus $K_n = \bigcap_{i=1}^{n} L(G_i)$ and so $K_n \in \bigcap_n \mathcal{L}(DOS)$.

(ii). We will demonstrate now that if $K_n \in \bigcap_m \mathcal{L}(DOS)$ then $m \geq n$.

This is shown by the following sequence of observations.

(1). If $K_n = L(G_1) \cap L(G_2) \cap \dots \cap L(G_m)$ then we can assume that the alphabet of each G_i, $1 \leq i \leq m$, equals Σ_n.

Proof of (1):

Clearly the alphabet of every G_i, $1 \le i \le m$, must contain Σ_n. If for some $j \in \{1,\dots,m\}$ the alphabet of G_j contains some letters not in Σ_n, then we can consider Σ_n as the terminal alphabet of G_j, and all other letters in G_j can be considered as nonterminal letters of G_j. In this way we can view G_j as a DOS systems with nonterminals (called an EDOS system). It is proved in [ER1] that for every EDOS system there exists a DOS system generating the same language. Hence (1) holds. □

(2). Let $K_{n,2} = \{A_i : 1 \le i \le n\} \cup \{A_iA_j : 1 \le i \ne j \le n\}$ and let G be a DOS system with the alphabet Σ_n. If

(I). $K_{n,2} \subseteq L(G)$, and

(II). for some $j \in \{1,\dots,n\}$, A_j is propagating,

then $L(G) = \Sigma_n^+$.

Proof of (2):

(I) and (II) together with Lemma 3 imply that every letter in $G = (\Sigma_n, h, \omega)$ is propagating. Then, because all one letter words over Σ_n are in $K_{n,2}$, one can order elements of Σ_n into a chain $a_1,\dots,a_n$ such that $h^{t_1}(a_1) = a_2$, $h^{t_2}(a_2) = a_3, \dots, h^{t_{n-1}}(a_{n-1}) = a_n$, for some $t_1, t_2, \dots, t_{n-1} \ge 1$. Since $K_{n,2} \subseteq L(G)$ it must be that $h^t(a_n) = a_1a_1$ for some $t \ge 1$ and so $L(G) = \Sigma_n^+$. □

(3). Assume that $K_n = L(G_1) \cap L(G_2) \cap \ldots \cap L(G_m)$ for some $m \geq 1$. A letter a in Σ_n is called *multiple* if it appears in at least two different paths of T_{G_j} for each $j \in \{1,\ldots,n\}$. There exist no multiple letters in Σ_n.

Proof of (3):

Assume that a is a multiple letter and consider an arbitrary G_j, $1 \leq j \leq m$. By (1) we can assume that the alphabet of G_j equals Σ_n. We have two cases to consider.

(a). All letters in G_j are erasing.

Then obviously $a^2 \in L(G_j)$.

(b). G_j contains a propagating letter.

Since $K_{n,2} \subseteq K_n$, $K_{n,2} \subseteq L(G_j)$ and so (2) implies that $L(G_j) = \Sigma_n^+$.

Hence $a^2 \in L(G_j)$.

Thus $a^2 \in L(G_j)$ and, since j was arbitrary, $a^2 \in L(G_1) \cap \ldots \cap L(G_j)$; a contradiction.

Hence Σ_n contains no multiple letters. □

(4). Assume that $K_n = L(G_1) \cap L(G_2) \cap \ldots \cap L(G_m)$ for some $m \geq 1$. Let f be a function from Σ_n into $\{1,\ldots,m\}$ defined by: for $a \in \Sigma_n$, f(a) equals the minimal index j from $\{1,..,m\}$ such that in T_{G_j} a appears on one path only (by (3) f is a well defined function). Then f is injective.

Proof of (4):

Assume to the contrary that Σ_n contains a,b with $a \neq b$ such that $f(a) = f(b)$. We have two cases to consider.

Case 1. In $T_{G_{f(a)}}$ both a and b appear on the same path. Then no word in $L(G_{f(a)})$ contains both an occurrence of a and an occurrence of b.

Consequently no word in $L(G_1) \cap \ldots \cap L(G_m)$ contains both an occurrence of a and an occurrence of b; a contradiction.

Case 2. In $T_{G_{f(a)}}$ a appears on a different path than b. Without loss of generality assume that a appears on a path that is to the left of the path on which b appears. Then in each word of $L(G_{f(a)})$, and hence in each word of $L(G_1) \cap \ldots \cap L(G_n)$, the unique occurrence of a is always to the left of the unique occurrence of b; a contradiction.

Thus f must be injective. □

(5). From (4) it follows that whenever $K_n \in \bigcap_m L(\mathrm{DOS})$ then $m \geq n$.

Now the theorem follows from (i) and (ii). □

To put the previous result in a proper perspective we show now that the class of languages obtained by the intersections of DOS languages is properly contained in the class of context sensitive languages.

First of all we have the following result.

Lemma 4. There exists a finite language K such that

$$K \notin \bigcup_{n=1}^{\infty} \left(\bigcap_n L(\mathrm{DOS})\right).$$

Proof.

Let $K = \{a, b, ab, ba, a^3\}$.

(1). If G is a DOS system with the alphabet {a,b} such that G contains a propagating letter and $K \subseteq L(G)$ then $a^2 \in L(G)$.

Proof of (1).

This follows directly from Lemma 3. □

(2). If G is a DOS system with the alphabet {a,b} such that G does not contain a propagating letter and $K \subseteq L(G)$ then $a^2 \in L(G)$.

Proof.

Obvious (because then $a^3 \Rightarrow a^2$). □

(3). Now we complete the proof of the lemma as follows.

Assume that $G_1,\ldots,G_n$ are DOS systems such that $L(G_1) \cap \ldots \cap L(G_n) = K$. We can assume that the alphabet of each G_j, $1 \le j \le n$, equals $\{a,b\}$ (see the reasoning under (ii).1 in the proof of Theorem 5). From (1) and (2) it follows then that $L(G_1) \cap \ldots \cap L(G_n)$ must contain a^2; a contradiction.

Consequently $K \notin \bigcup_{i=1}^{\infty}(\bigcap_n L(DOS))$. □

Theorem 6. $\bigcup_{n=1}^{\infty}(\bigcap_n L(DOS))$ is strictly included in the class of context sensitive languages.

Proof.

Since for every $n \ge 1$ and every DOS systems $G_1,\ldots,G_n$ one easily constructs a linear bounded automaton accepting $L(G_1) \cap \ldots \cap L(G_n)$, the weak inclusion is obvious. The strict inclusion follows from Lemma 4. □

ACKNOWLEDGMENTS

The authors are indebted to H.C.M. Kleijn, R. Verraedt and P. Zeiger for comments concerning the first version of this paper. The authors gratefully acknowledge the financial support of NSF grant number MCS 79-03838.

REFERENCES

[BPR] H.W. Buttelmann, A. Pyster and L.H. Reeker, Grammars without syntactic variables, University of Oregon, Dept. of Computer Science, Technical Report 74-1, 1974.

[ER1] A. Ehrenfeucht and G. Rozenberg, On basic properties of DOS systems and languages, *Information and Control*, 47, 137-153, 1980.

[ER2] A. Ehrenfeucht and G. Rozenberg, On the emptiness of the intersection of two DOS languages problem, *Information Processing Letters*, 10, 223-225, 1979.

[HP] T. Harju and M. Pentonen, Some decidability problems of sentential forms, *International Journal of Computer Mathematics*, 7, 95-108, 1979.

[MSW] H.A. Maurer, A. Salomaa and D. Wood, Pure Grammars, Computer Science Technical Report No. 79-CS-7, Dept. of Applied Mathematics, McMaster University, 1979.

[S1] A. Salomaa, *Formal languages*, Academic Press, New York-London, 1973.

[S2] A. Salomaa, On sentential forms of context-free grammars, *Acta Informatica* 2, 1973, 40-49.