DOS SYSTEMS AND LANGUAGES

by

A. Ehrenfeucht[*]

and

G. Rozenberg[**]

CU-CS-160-79 August, 1979

[*] A. Ehrenfeucht, Dept. of Computer Science, University of Colorado,
Boulder, Colorado 80309 USA

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

All correspondence to G. Rozenberg.

DOS SYSTEMS AND LANGUAGES

by

A. Ehrenfeucht
Department of Computer Science
University of Colorado at Boulder
Boulder, Colorado 80309
U.S.A.

and

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

All correspondence to G. Rozenberg

DOS SYSTEMS AND LANGUAGES

by

A. Ehrenfeucht
Department of Computer Science
University of Colorado at Boulder
Boulder, Colorado 80309
U.S.A.

and

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

All correspondence to G. Rozenberg

ABSTRACT

A new type of grammar called a DOS system is introduced and investigated. Essentially it formalizes the notion of a context free grammar without variables that is generatively deterministic.

INTRODUCTION

There are several approaches to a systematic build-up of formal language theory or various fragments of it. An example of such an approach is the mathematical theory of L systems (see, e.g., [R] and [RS]). Its basic component is a DOL system which is essentially an iterated homomorphism on a free monoid. A DOL system can be generalized to a OL system by allowing an iteration of a finite substitution rather than the iteration of a homomorphism. Then to either a DOL system or to a OL system nonterminals can be added giving rise to EDOL and EOL systems respectively. These four classes of systems (DOL, OL, EDOL and EOL) form the basic framework for the systematic development of the theory of L systems.

In an attempt to build-up a systematic theory of context free languages one can look for an analogue of the above situation in the framework of context free grammars. Obviously, context free grammars correspond to EOL systems, and, roughly speaking, context free grammars without nonterminals correspond to OL systems. In recent years such "classical" grammars without nonterminals were investigated (see, e.g., [BPR], [HP], [MSW] and [S1]).

What is missing at this moment is the sequential analogue of DOL systems, which, in the above outlined approach, forms the very essential element of the theory. In this paper we introduce such a sequential analogue of a DOL system, called a DOS system, and we believe it will play the same role in the theory of context free languages as DOL systems play in the theory of EOL languages. One of the essential advantages of DOS systems (in our opinion) is the fact that they allow for the first time to formalize the notion of "grammatical determinism" in the framework of "context-free-like" sequential grammars.

The paper is organized as follows.

In the first section DOS systems and languages are introduced and illustrated by examples. Also a graph representation of a DOS system is presented.

In the second section some very basic problems, including the role of nonterminal symbols, the role of erasing and the relationship between DOS and DOL systems, are investigated.

In Section III we provide a result on the combinatorial structure of DOS languages that allows one to provide various examples of languages that are not DOS languages. Also closure properties of the class of DOS languages are investigated in this section.

In the last section we establish a representation theorem analogous to the Chomsky-Schützenberger theorem except that it uses DOS languages rather than Dyck languages (Dyck languages are not DOS languages).

We assume the reader to be familiar with the rudiments of the theory of context free languages. We use mostly the standard notation and terminology. Perhaps only the following points require an additional explanation.

(1). For a word α, $|\alpha|$ denotes its length, and, for $1 \leq i \leq |\alpha|$, $\alpha[i]$ denotes the letter that occurs in α as the i'th element from the left.

(2). For a class of grammars X, $L(X)$ denotes the class of languages generated by grammars in X.

(3). As usual, throughout this paper we apply the convention that, for a language K, $K = K \cup \{\Lambda\}$.

(4). Given a labelled graph G, ℓ_G denotes its labeling function; if G is understood we write ℓ, rather than ℓ_G.

I. DOS SYSTEMS AND LANGUAGES

In this section DOS systems and languages are introduced and illustrated by examples. Also a "forest representation" of (all derivations in) a DOS system is presented.

Our first notion is that of a sequential homomorphism, which is like a homomorphism except that it is applied "sequentially", that is one occurrence in a string is replaced in one application of the sequential homomorphism.

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 letters 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 $n \geq 0\}$, and referred to as a DOS *language*. If for no $a \in \Sigma$, $h(a) = \Lambda$ then we call G *propagating* and refer to it as a PDOS *system* (in this case L(G) is called 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). Then we use the notation $x \underset{G}{\Rightarrow} y$ and $x \overset{*}{\underset{G}{\Rightarrow}} y$ in the usual sense (with omitting the reference to G whenever G is clear from the context).

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

A special kind of a labelled ordered forest is naturally associated with a DOS system. It plays the role of a derivation tree in a context free grammar except that now this one forest represents *all* derivations in a given DOS system. It is defined as follows.

Definition. Let Σ be a finite alphabet. A D-*forest* (*over* Σ) is an infinite ordered labeled forest T such that:

(1). there exists a positive integer k such that for every node u of T the out-degree of u is not bigger than k,

(2). for every node u of T the subtree of T rooted at u (denoted as T_u) is infinite,

(3). every node of T is labelled either by an element of Σ or by the empty word,

(4). if a node u is labelled by Λ then every node in T_u is labelled by Λ, (5). if nodes u and v have the same labels, then T_u and T_v are isomorphic (with the isomorphism on labels being the identity mapping). □

The ordered sequence of roots of the trees of T (in the order they occur in T) is referred to as the *origin* of T. (Hence, if T is a tree then the origin of T is the root of T).

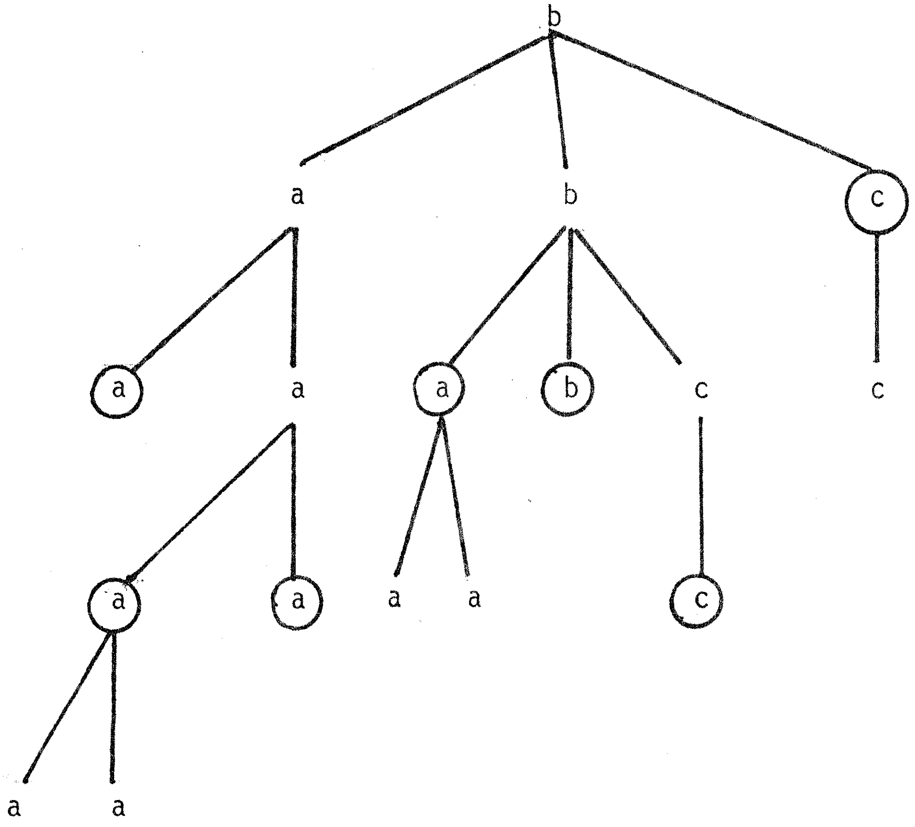
With every D-forest over Σ we can associate a language over Σ as follows.

Definition. Let T be a D-forest. A *cut of* T is a sequence τ of nodes from T such that on each infinite path in T starting in the origin of T there is exactly one node from τ and the order of nodes in τ is their (left to right) order in T. We use *cut* T to denote the set of all cuts of T. □

Definition. Let T be a D-forest over an alphabet Σ. The *cut language of* T, denoted L_{cut} (T), is defined by

$$L_{cut}(T) = \{\alpha \in \Sigma^* : \alpha = \ell(u_1)\ldots\ell(u_n) \text{ where } u_1,\ldots,u_n \text{ are nodes of T and } u_1\ldots u_n \in \textit{cut}\ T\}. \quad \square$$

Example 2. Let T be a D-forest represented by the following fragment of it:



The sequence of encircled nodes in their left-to-right order represents a cut of T; the word corresponding to this cut is a^4bc^2. The cut language of T is $\{a^mbc^n : m \geq n \geq 0\}$. □

Given a DOS system $G = (\Sigma, h, \omega)$ one can construct its D-forest T_G "originating in ω" in much the same way as a derivation tree is constructed in a context free grammar except that:

(1). T_G is infinite (and, as usual, if u is a node labeled by $b \in \Sigma$ then the word obtained by catenating, from left to right, the labels of direct successors of u equals α if and only if $h(b) = \alpha$), and

(2). if a node is labeled by the empty word (corresponding to a production $b \to \Lambda$ in G) then it has precisely one direct successor also labelled by the empty word.

Example 3. The D-forest T from Example 2 corresponds to the D-forest T_G of the DOS system G from Example 1. □

Clearly, given a DOS system G, to each word in L(G) there corresponds a (not necessarily unique) cut in T_G and the sequence of labels corresponding to a cut in T_G yields a word in L(G). As a matter of fact we get the following easy to prove result.

Theorem 1.

(1). Let G be a DOS system. Then $L(G) = L_{cut}(T_G)$.

(2). Let K be a language. K is a DOS language if and only if there exists a D-forest T such that $L_{cut}(T) = K$. □

II. SOME BASIC PROPERTIES

In this section we investigate briefly the role of erasing in DOS systems, investigate the affect of adding nonterminals to DOS systems and look at some natural relationships between DOL and DOS systems.

It turns out that there are DOS languages that cannot be defined by PDOS systems.

Theorem 2. There exists a finite language which is in $\mathcal{L}\mathrm{DOS} \setminus \mathcal{L}(\mathrm{PDOS})$.

Proof.

Consider $K = \{ab, b\}$. K is a DOS language because it is generated by the DOS system $(\{a,b\}, h, ab)$ where $h(a) = \Lambda$ and $h(b) = b$. However if we assume that $K = L(G)$ for a PDOS system G then b must be the axiom of G and consequently (since $b \overset{+}{\Rightarrow} ab$) $L(G)$ must be infinite; a contradiction. □

A standard language-theoretic method to increase the language generating power of a class X of language generating systems is to equip the elements of X with the mechanism of nonterminal symbols. Surprisingly enough, adding nonterminals to DOS systems does not alter the class of languages generated.

Definition. An EDOS *system* is a construct $G = (\Sigma, h, \omega, \Delta)$ where $U(G) = (\Sigma, h, \omega)$ is a DOS system and $\Delta \subseteq \Sigma$ (elements of Δ are called *terminal symbols* and elements of $\Sigma \setminus \Delta$ are called *nonterminal symbols*). The *language of* G is defined by $L(G) = L(U(G)) \cap \Delta^*$. □

Theorem 3. $\mathcal{L}(\mathrm{DOS}) = \mathcal{L}(\mathrm{EDOS})$.

Proof.

(i). Obviously $\mathcal{L}(\mathrm{DOS}) \subseteq \mathcal{L}(\mathrm{EDOS})$.

(ii). Let us consider an EDOS system $G = (\Sigma, \omega, h, \Delta)$. Let $b \in \Sigma$, let u be a node in T_G labelled by b and let us consider T_u (the subtree of T_G rooted at u). We say that b is *blocking* if there is an infinite path τ in T_u originating in u such that all nodes appearing in τ, with the possible exception of u, are labelled by the elements of $\Sigma \backslash \Delta$. Otherwise b is called *nonblocking*.

Let $\overline{h}$ be the homomorphism on Σ^* defined as follows.

If b is blocking, then $\overline{h}(b) = b$.

If b is nonblocking, then on every infinite path ξ in T_u originating in u we choose the node on ξ which is closest to (but different from) u and labelled by an element of $\Delta \cup \{\Lambda\}$. By Konig's lemma there is a finite number of such nodes (on all infinite paths originating in u); let their (left to right) order in T_u be $v_1,\ldots,v_{m_b}$ and let $c_1,\ldots,c_{m_b}$ be their corresponding labels (all of them are elements of $\Delta \cup \{\Lambda\}$). Then let $\overline{h}(b) = c_1 \ldots c_{m_b}$.

Let $\overline{G} = (\Sigma, \overline{h}, \overline{\omega})$ be the DOS system where $\overline{\omega} = \overline{h}(\omega)$.

It is easy to see that $L(\overline{G}) = L(G)$; the key observation here is that the sequence $v_1,\ldots,v_{m_b}$ used in the definition of $\overline{h}(b)$ for a nonblocking b is a cut of T_G and in every successful derivation in G if an occurrence of b is rewritten then at some stage it must yield an occurrence of $\overline{h}(b)$.

Hence $L(\text{EDOS}) \subseteq L(\text{DOS})$. □

In the rest of this section we will contrast DOL systems with DOS systems. One of the basic properties of a DOL language is that the number of different subwords of the length k it can have is bounded by the quadratic function $C \cdot k^2$ where C is a constant (see [RS]). DOS languages are not subject to such a restriction on the number of subwords they can generate.

Theorem 4. There exists a DOS language K over a two-letter alphabet Σ, such that for each $n \geq 0$ all words over Σ of length n appear as subwords in K.

Proof.

Consider the DOS system $G = (\{a,b\}, h, a)$ where $h(a) = ba$ and $h(b) = ab$. Clearly for every word α in $\{a,b\}^*$ there exists a word β in $L(G)$ such that $\beta = \alpha\,\gamma$ for some word γ in $\{a,b\}^*$. Thus $K = L(G)$ satisfies the statement of the theorem. □

An instructive way of investigating the relationship between parallel and sequential rewriting systems is to consider a DOL system as a DOS system (that is to apply the homomorphism involved sequentially) and, the other way around, to consider a DOS system as a DOL system (that is to apply the s-homomorphism involved in the parallel fashion). This topic is investigated now.

Definition. Let $G = (\Sigma, h, \omega)$ be a DOL system and $\overline{G} = (\overline{\Sigma}, h, \overline{\omega})$ be a DOS system. We say that G and $\overline{G}$ are *twins* if $\Sigma = \overline{\Sigma}$, $h = \overline{h}$ and $\omega = \overline{\omega}$ (and we write $G = \mathit{twin}\ \overline{G}$ and $\overline{G} = \mathit{twin}\ G$).(1) □

Theorem 5.

(1). There exist DOL systems G_1, G_2 such that $L(G_1) = L(G_2)$ but $L(\mathit{twin}\ G_1) \neq L(\mathit{twin}\ G_2)$.

(2). There exist DOS systems G_1, G_2 such that $L(G_1) = L(G_2)$ but $L(\mathit{twin}\ G_1) \neq L(\mathit{twin}\ G_2)$.

Proof.

(1). Consider DOL systems $G_1 = (\{a,b,c\}, h_1, abc)$ and $G_2 = (\{a,b,c\}, h_2, abc)$ where $h_1(a) = ab^2$, $h_1(b) = b$, $h_1(c) = c$

and $h_2(a) = ab$, $h_2(b) = b$ and $h_2(c) = bc$. Then
$L(G_1) = L(G_2) = \{a\, b^{2n+1}\, c : n \geq 0\}$, while
$L(\mathit{twin}\ G_1) = \{a\, b^{2n+1}\, c : n \geq 0\} \neq L(\mathit{twin}\ G_2) = \{ab^n c : n \geq 1\}$.

(2). Consider DOS systems $G_1 = (\{a,b,c\}, h_1, abc)$ and $G_2 = (\{a,b,c\}, h_2, abc)$ where $h_1(a) = ab^2$, $h_1(b) = b$, $h_1(c) = c$ and $h_2(a) = a$, $h_2(b) = b^3$, $h_2(c) = c$. Then
$L(G_1) = L(G_2) = \{a\, b^{2n+1}\, c : n \geq 0\}$, while
$L(\mathit{twin}\ G_1) = \{a\, b^{2n+1}\, c : n \geq 0\} \neq L(\mathit{twin}\ G_2) = \{a\, b^{3^n} c : n \geq 0\}$. □

We will discuss now a situation in which knowing a property of a DOL system G we can infer a property of the language $L(\mathit{twin}\ G)$. We start by recalling the notion of rank of a DOL system (see [ER]).

Definition. Let $G = (\Sigma, h, \omega)$ be a DOL system.

(1). The *rank of a letter* b *in* G, denoted as $\mathit{rank}_G\ b$ is defined inductively as follows.

(i). If $\{\alpha : b \overset{*}{\underset{G}{\Rightarrow}} \alpha\}$ is finite then $\mathit{rank}_G\ b = 0$.

(ii). For $n \geq 1$ let h_n denote the restriction of h to $\Sigma_n = \Sigma \backslash \{a : \mathit{rank}_G\ a \leq n\}$ and let $G_n = (\Sigma, h_n, \omega)$. If $\{\alpha : b \overset{*}{\underset{G_n}{\Rightarrow}} \alpha\}$ is finite then $\mathit{rank}_G\ b = n$.

(2). We say that G is a DOL *system with rank* if every useful letter in G (that is a letter in ω or a letter reachable from a letter in ω) has a rank. In that case the *rank of* G, denoted as *rank* G, is defined as the highest of the ranks of useful letters in G.

Theorem 6. Let G be a DOL system with rank. Then $L(\mathit{twin}\ G)$ is a context free language of finite index.

Proof.

Let $G = (\Sigma, h, \omega)$. Let $\overline{\Sigma} = \{\overline{a} : a \in \Sigma\}$ and let for α in Σ^*, $\overline{\alpha}$

be the word produced by replacing all occurrences in α by their barred counterparts from $\overline{\Sigma}$; also $\overline{\Lambda} = \Lambda$. Then let $H = (V_N, V_T, P, S)$ be the context free grammar where $V_N = \{S\} \cup \overline{\Sigma}$ with $S \notin \Sigma \cup \overline{\Sigma}$, $V_T = \Sigma$ and $P = \{S \to \omega\} \cup \{\overline{a} \to \overline{\alpha} : h(a) = \alpha\} \cup \{\overline{a} \to a : a \in \Sigma\}$.
Clearly $L(H) = L(\mathit{twin}G)$.

Let T be a derivation tree of a word α in H. We give now a method for obtaining from T a derivation of α in H such that the number of nonterminals in every word of it is smaller than some constant Q dependent on H only.

(1). First rewrite the axiom S using the production $S \to \omega$.

(2). Let β be a sentential form already obtained in the derivation process.

(2.1). If β contains an occurrence o of a letter $\overline{b} \in \overline{\Sigma}$ which in T corresponds to a node replaced by the production $\overline{b} \to b$ then this occurrence o must be replaced by the production $\overline{b} \to b$.

(2.2). If step (2.1) cannot be applied to β, then productions different from the productions of the form $\overline{b} \to b$, $\overline{b} \in \overline{\Sigma}$, can be applied to β, subject to the following restriction: an occurrence of a letter of rank m in β, $m \geq 0$, can be rewritten only if β contains no occurrence of a letter of rank smaller than m.

Observe that if we consider only derivations in H obtained according to the above method then we derive all elements of L(H) (because we get every derivation tree in H). However if D is a derivation obtained as above then

(i). if we apply a production of the form $\overline{b} \to b$, $\overline{b} \in \overline{\Sigma}$ to a sentential form β, then the number of occurrences of (barred) letters of any given rank (if a in G has rank m, then we say that $\overline{a}$ has rank m) does not increase,

(ii). if we rewrite a letter $\overline{b}$ of rank m, then we do not introduce any letters of rank bigger than m, and

(iii). if we rewrite a letter of rank $m \geq 1$ then if we introduce letters of rank smaller than m then in a single rewriting we cannot introduce more of them than the maximal length of the right-hand side of a production in P.

Thus, clearly, no sentential form in H obtained as above has more than Q nonterminals, where Q is a constant dependent on H only. Consequently, H is a context free grammar of finite index. □

We end this section by demonstrating that the emptiness of the intersection of a DOS language with a DOL language problem is undecidable.

Theorem 7. If is undecidable whether or not $L(G_1) \cap L(G_2) = \emptyset$ where G_1 is a DOS system and G_2 is a DOL system.

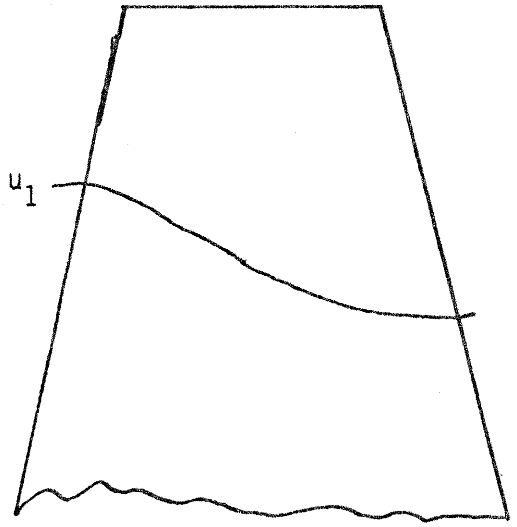
Proof.

Let $G_1 = (\Sigma, h, \omega_1)$ and $\overline{G}_2 = (\Sigma, h, \omega_2)$ be two arbitrary cofunctional DOS systems, that is DOS systems with the same s-homomorphism. Let $G_2 = \textit{twin}\ \overline{G}_2$.

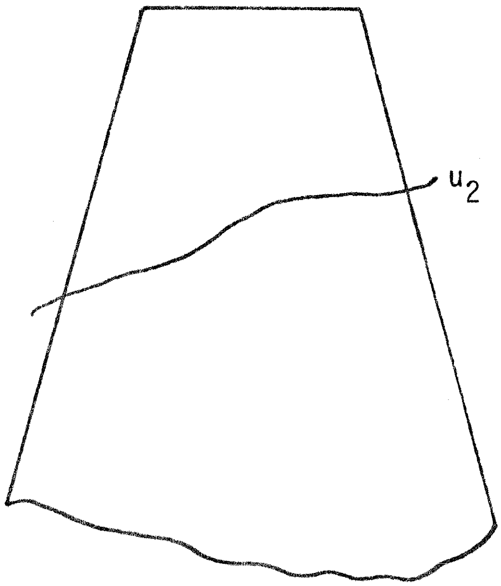
(i). Assume that $L(G_1) \cap L(G_2) \neq \emptyset$. Since, $L(G_2) \subseteq L(G_1)$, this implies that $L(G_1) \cap L(\overline{G}_1) \neq \emptyset$.

(ii). Assume that $L(G_1) \cap L(G_2) \neq \emptyset$. This means that there exists a word α which is both in $L(G_1)$ and in $L(G_2)$. If $\alpha \in L(G_2)$ then $L(G_1) \cap L(G_2) \neq \emptyset$. Otherwise, let us consider a cut u_1 in T_G yielding α in $L(G_1)$ and a cut u_2 in T_{G_2} yielding α in $L(\overline{G}_2)$; the situation is illustrated by the following picture:

T_{G_1} :

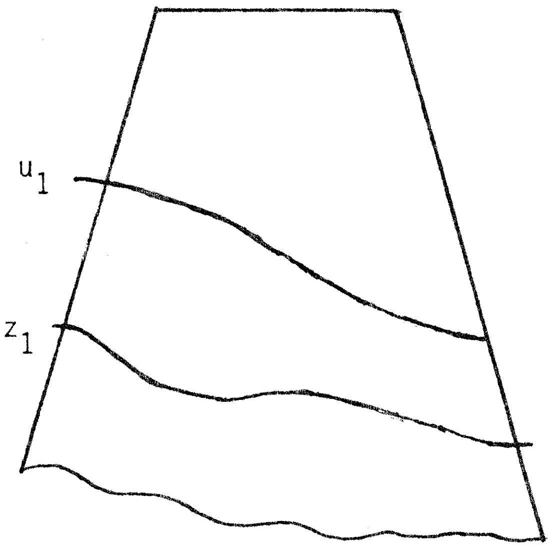


$T_{\overline{G}_2}$:

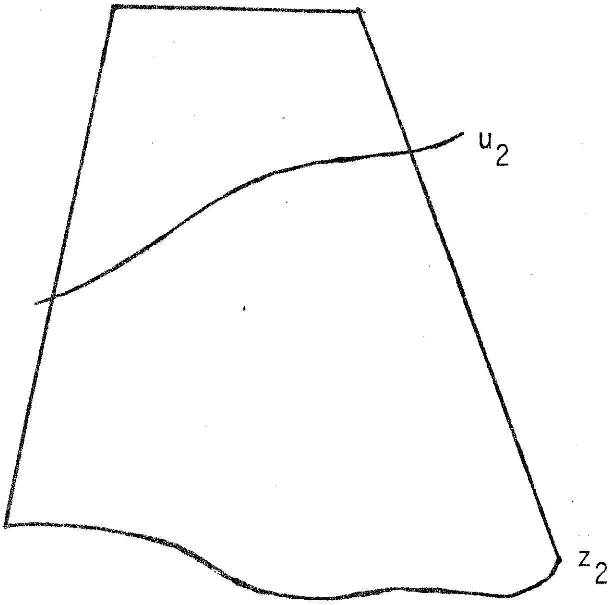


We can certainly rewrite nodes in u_2 in such a way as to get a "horizontal cut" z_2 in $T_{\overline{G}_2}$, that is a cut consisting of nodes all of which are of the same distance from the origin of $T_{\overline{G}_2}$. Since G_1 has the same s-homomorphism as G_2 and both u_1 and u_2 yield the same word (α) we can certainly rewrite nodes of u_1 in T_{G_1} in the same way as the corresponding nodes of u_2 are rewritten in $T_{\overline{G}_2}$, in this way we get a cut z_1 (note that z_1 does not have to be a parallel cut in T_{G_1}). Hence we get the following situation:

T_{G_i}:



$T_{\overline{G}_2}$:



Obviously both z_1 and z_2 will yield the same word, say β. Hence $\beta \in L(G_1) \cap L(G_2)$ and consequently $L(G_1) \cap L(G_2) \neq \emptyset$.

(iii). From (i) and (ii) it follows that $L(G_1) \cap L(\overline{G}_2) \neq \emptyset$ if and only if $L(G_1) \cap L(G_2) \neq \emptyset$. Since it was proved in [ER2] that it is undecidable whether or not $L(G_1) \cap L(\overline{G}_2) = \emptyset$ for arbitrary DOS systems G_1, $\overline{G}_2$, this implies that the theorem holds. □

III. ON THE STRUCTURE OF DOS LANGUAGES

In this section we provide a result on the combinatorial structure of DOS languages and investigate some of its consequences. We also look at the closure properties of L(DOS).

First of all we need the following terminology: if (α, β) is a pair of words such that either $|\alpha| = 1$ or $|\beta| = 1$ then (α,β) is called *unary*.

Theorem 8. Let K be a DOS language. For every $\alpha,\beta \in$ K there exists a positive integer and words $\alpha_1,\dots,\alpha_n$, $\beta_1,\dots,\beta_n$ such that $\alpha = \alpha_1\dots\alpha_n$, $\beta = \beta_1\dots\beta_n$, (α_i, β_i) is unary for $1 \leq i \leq n$ and $\gamma_1\dots\gamma_n \in$ K for all words $\gamma_1,\dots,\gamma_n$ such that, for every $1 \geq i \geq n$, either $\gamma_i = \alpha_i$ or $\gamma_i = \beta_i$.

Proof.

Let K $\in L$(DOS) and let $\alpha,\beta \in$ K. Let G = (Σ, h, ω) be a DOS system generating K, let cut_α and cut_β be two cuts of T_G corresponding to α and β respectively, and let CUT(α,β) be the set of all nodes appearing either in cut_α or in cut_β. Let u $\in$ CUT(α,β) and let τ be an infinite path in T_G to which u belongs. We say that: u is *equal on* τ if u is the only node from CUT(α,β) on τ; u is *higher on* τ if there are two nodes from CUT(α,β) on τ and out of these two u is closer to the origin of T_G; u is *lower on* τ if there are two nodes from CUT(α,β) on τ and out of these two u is further from the origin of T_G.

Note that the three cases above exhaust all possibilities for u on τ and moreover if ξ is an infinite path in T_G such that u belongs to ξ then:

u is equal (higher, lower respectively) on τ

if and only if

u is equal (higher, lower respectively) on ξ.

Consequently we can partition elements of $CUT(\alpha,\beta)$ into three classes as follows:

$E = \{u \in CUT(\alpha,\beta)$: u is equal on τ for every infinite
path τ in T_G to which u belongs$\}$,

$H = \{u \in CUT(\alpha,\beta)$: u is higher on τ for every infinite
path τ in T_G to which u belongs$\}$, and

$L = \{u \in CUT(\alpha,\beta)$: u is lower on τ for every infinite
path τ in T_G to which u belongs$\}$.

Elements of E, H and L are referred to as *equal*, *higher* and *lower* nodes respectively.

Let $t_1,\dots,t_q$ be the sequence of all nodes from $E \cup H$ in the (left to right) order that they appear in T_G. Let h be the mapping from $E \cup H$ into the ordered pairs of words over Σ defined as follows. Let $u \in E \cup H$; then

(1). $h(u) = (\ell(u), \ell(u))$ if $u \in E$,

(2). $h(u) = (\ell(u), \ell(\overline{u}_1)\dots\ell(\overline{u}_{m_u}))$ if $u \in H$ and $u \in cut_\alpha$, where $\overline{u}_1,\dots,\overline{u}_{m_u}$ are all nodes from $CUT(\alpha,\beta)$ that are descendants of u in T_G and they appear in this, left to right, order in T_G,

(3). $h(u) = (\ell(\overline{u}_1)\dots\ell(\overline{u}_{m_u}), \ell(u))$ if $u \in H$ and $u \in cut_B$, where $\overline{u}_1,\dots,\overline{u}_{m_u}$ are all nodes from $CUT(\alpha,\beta)$ that are descendants of u in T_G and they appear in this, left to right, order in T_G.

Clearly

- if $u \in H$ and either $h(u) = (a,\alpha)$ where $u,(a,\alpha)$ satisfy (2) above, or $h(u) = (\alpha,a)$ where u, (α,a) satisfy (3) above, then $a \overset{*}{\underset{G}{\Rightarrow}} \alpha$. ...($).

Let $\pi = h(t_1)\ h(t_2)\ \ldots\ h(t_q)$ and let $(\alpha_1, \beta_1)\ (\alpha_2, \beta_2)\ \ldots\ (\alpha_n, \beta_n)$ result from π by erasing from it all pairs of the form (Λ, Λ). Now it is easy to see that the fact that $t_1,\ldots,t_q$ is a cut of T_G, observation ($) and Theorem 1 imply that $n, \alpha_1,\ldots,\alpha_n, \beta_1,\ldots,\beta_n$ satisfy the statement of the theorem. □

The following result demonstrates that the above theorem cannot be strengthened into the "if and only if" result.

Theorem 9. There exists a nonrecursive language K satisfying the conclusion of Theorem 8.

Proof.

Let M be a nonrecurisve set of positive integers and let $\Sigma = \{a, b\}$. Then, obviously, $L = \{a\ b^n\ a : n \in M\}^*$ is nonrecursive. Let $\Sigma = \overline{\Sigma} \cup \{A,B,C\}$, where $\{A,B,C\} \cap \overline{\Sigma} = \emptyset$, and let $K = A\ \{BLC\}^*$. That K satisfies the conclusion of Theorem 8 is seen as follows.

Let $\alpha, \beta \in K$.

If either $|\alpha| = 1$ or $|\beta| = 1$ then the conclusion of Theorem 8 obviously holds.

Hence assume that

$\alpha = AB\pi_r CB\pi_{r-1}C\ldots B\pi_1 C,$

$\beta = AB\xi_s CB\xi_{s-1}C\ldots B\xi_1 C$

for some $s \geq r \geq 1$ and $\pi_1,\ldots,\pi_r,\xi_1,\ldots,\xi_s$ in L.

Set $n = 2\ r+1$, and

- if $s = r$ then set

$\alpha_1 = A,\ \beta_1 = A,$

$\alpha_2 = B,\ \beta_2 = \beta\xi_s,$

$\alpha_3 = \pi_r C,\ \beta_3 = C,$

$$\vdots \qquad \vdots$$

$\alpha_{2r} = B,\ \beta_{2r} = B\xi_1,$

$\alpha_{2r+1} = \pi_1 C,\ \beta_{2r+1} = C;$

- if $s > r$ then set

$\alpha_1 = A,\ \beta_1 = AB\xi_s CB\xi_{s-1}C\ldots B\xi_{r+1}C,$

$\alpha_2 = B,\ \beta_2 = \beta\xi_r,$

$\alpha_3 = \pi_r C,\ \beta_3 = C,$

$$\vdots \qquad \vdots$$

$\alpha_{2r} = B,\ \beta_{2r} = B\xi_1,$

$\alpha_{2r+1} = \pi_1 C,\ \beta_{2r+1} = C.$

Then, obviously, the conclusion of Theorem 8 holds for this choice of n, $\alpha_1,\ldots,\alpha_n$, $\beta_1,\ldots,\beta_n$. □

Even if we consider only languages K "quite close" to DOS languages, the condition of Theorem 8 would not suffice for the characterization of the class of DOS languages as illustrated by the following. Let us call a unary pair of words (α,β) from K *strong* if whenever $\gamma \in \{\alpha,\beta\}$ and $|\gamma| = 1$ then $\delta_1\gamma\delta_2 \in K$ implies that $\delta_1\beta\delta_2 \in K$ if $\gamma = \alpha$ and $\delta_1\alpha\delta_2 \in K$ if $\gamma = \beta$. Let us refer to the conclusion of Theorem 8, where we replace the word "unary" by "strong and unary" as the "modified conclusion of Theorem 8." Clearly our proof of Theorem 8 implies that its modified conclusion holds. A OS *system* is a nondeterministic version of a DOS system, that is the s-homomorphism in a DOS system is replaced by a finite substitution. OS systems generate OS *languages*.

Now we can state our second "negative" result about the possibility of turning Theorem 8 into an "if and only if" theorem. (Note that the modified conclusion of Theorem 8 is stronger than the conclusion of Theorem 8).

Theorem 10. There exists a OS language K satisfying the modified statement of Theorem 8 such that K is not a DOS language.

Proof.

Consider a (propagating) OS system $G = (\{a\},h,a^2)$ where the finite substitution h is defined by $h(a) = \{a_3,a_4\}$. Then, obviously, $L(G) = \{a^2\} \cup \{a^n : n \geq 4\}$. Let us consider a pair of words (α,β) from K.

(1). If $\alpha = \beta$ then clearly the modified statement of Theorem 8 holds.

(2). If either $\alpha = a^2$ or $\beta = a^2$ then obviously the modified statement of Theorem 8 holds.

(3). If $\alpha = a^4$, $\beta = a^5$ then $\alpha_1 = a^3$, $\alpha_2 = a$, $\beta_1 = a$, $\beta_2 = a^4$ satisfy the modified statement of Theorem 8.

(4). If $\alpha = a^4$, $\beta = a^n$, $n > 5$ then we can write $\alpha = \alpha_1\alpha_2$, $\beta = \beta_1\beta_2$ with $\alpha_1 = \alpha_2 = a^2$, $\beta_1 = a^2$ and $\beta_2 = a^{n-2}$ where $n - 2 \geq 4$. Then it follows from (2) that (α,β) satisfies the modified statement of Theorem 8.

(5). Since it is clear that if all pairs of the form (a^4,a^n) satisfy the modified statement of Theorem 8, then also all pairs of the form (a^k,a^n) , $k > 4$ satisfy this statement, the theorem holds. □

The next result following directly from Theorem 8 allows one to provide numerous examples of languages that are not in $L(\mathrm{DOS})$.

Corollary 1. Let $K \in L(\mathrm{DOS})$.

(1). Let α,β in K be such that $|\alpha| \geq 2$ and $|\beta| \geq 2$. Then there exist words α_1, α_2, β_1, β_2 such that $\alpha = \alpha_1\alpha_2$, $\beta = \beta_1\beta_2$, $\alpha_1\beta_1 \neq \Lambda$, $\alpha_2\beta_2 \neq \Lambda$, $\alpha_1\beta_2 \in K$ and $\beta_1\alpha_2 \in K$.

(2). Let $K \subseteq \Sigma^*$ and let (Σ_1,Σ_2) be a partition of Σ. If there exist α,β in K such that $|\alpha| \geq 2$, $|\beta| \geq 2$, $\alpha \in \Sigma_1^+$, $\beta \in \Sigma_2^+$ then there exists a word γ in K such that $\gamma \in \Sigma_1^* \cup \Sigma_2^*$.

Proof.

(1) follows easily from Theorem 8 and (2) follows directly from (1). □

We conclude this section by establishing the closure properties of L(DOS).

Theorem 11. For each of the following operations

(i). union,

(ii). intersection,

(iii). concatenation,

(iv). the star operation,

(v). intersection with a regular set,

(vi). Λ-free homomorphism,

(vii). inverse homomorphism,

there exists a finite DOS language, or finite DOS languages if the operation is binary, such that the application of the given operation to the given language or languages produces a language which is not a DOS language.

Proof.

Let $K_0 = \{a^2,b^2\}$. It follows directly from Corollary 1.(2) that $K_0 \notin L$(DOS).

Let $K_1 = \{ab, a^3, b^3, a^2b^2\}$. That $K_1 \notin L(DOS)$ is seen as follows. If we assume that $G = (\{a,b\}, h, \omega)$ is a DOS system such that $L(G) = K_1$ then it follows immediately from the form of K_1 that is $\alpha \in h(x)$ for $x \in \{a,b\}$ then $|\alpha| = 1$. Then however it must be that $\omega = a^2b^2$ and none of the words ab, a^3, b^3 is in $L(G)$; a contradiction.

Let $K_2 = \{ab, a^3\}^*$. That $K_2 \notin L(DOS)$ is seen as follows. If we assume that $G = (\{a,b\}, h, \omega)$ is a DOS system such that $L(G) = K_2$ then it follows immediately from the form of K_2 that if $\alpha \in h(x)$ for $x \in \{a,b\}$ then $|\alpha| \geq 1$. Consequently $ab \underset{G}{\Rightarrow} a^3$ which implies that $h(b) = a^2$. Since either $ab \underset{G}{\Rightarrow} abab$ or $a^3 \underset{G}{\Rightarrow} abab$, and $a^3 \underset{G}{\Rightarrow} abab$ is impossible, it must be that $h(a) = aba$. Then however $(aba)^3 \in h^3(a^3)$ while $(aba)^3 \notin K_2$; a contradiction.

Let $K_3 = \{ab, ba, b^3\}$. It follows directly from Corollary 1.(1) that $K_3 \notin L(DOS)$ (take $\alpha = ab$ and $\beta = ba$).

Now we prove the theorem as follows.

(i). Both $\{a^2\}$ and $\{b^2\}$ are DOS languages, however $\{a^2\} \cup \{b^2\} = K_0$.

(ii). Let $G = (\{a,b\}, h, a^2)$ and $H = (\{a,b,c\}, \bar{h}, a^2)$ where $h(a) = b$, $h(b) = b$ and $\bar{h}(a) = c$, $\bar{h}(b) = b$, $\bar{h}(c) = b$. Then $L(G) \cap L(\bar{G}) = K_0$.

(iii). Both $\{a, b^2\}$ and $\{b, a^2\}$ are obviously DOS languages, however $\{a, b^2\}\,\{b, a^2\} = K_1$.

(iv). $\{ab, a^3\}$ is obviously a DOS language, however $\{ab, a^3\}^* = K_2$.

(v). Let $G = (\{a,b\}, h, a^2)$ where $h(a) = b$ and $h(b) = b$. Then $L(G) \cap \{a^2, b^2\} = \{a^2, b^2\}$.

(vi). $\{a,b\}$ is obviously a DOS language, however $h(\{a,b\}) = K_0$ where h is the Λ-free homomorphism defined by $h(a) = a^2$ and $h(b) = b^2$.

(vii). Let h be the homomorphism from $\{a,b\}^*$ into $\{A\}^*$ defined by $h(b) = A$ and $h(a) = A^2$. Then $h^{-1}(\{A^3\}) = K_3$, where $\{A^3\}$ is a DOS language. □

IV. A REPRESENTATION THEOREM FOR CONTEXT FREE LANGUAGES

In this section we establish a representation theorem for the class of context free languages that is analogous to the well-known Chomsky-Schützenberger Theorem except that rather than Dyck languages it uses DOS analogues of Dyck languages. To put this result in proper perspective we observe first that Dyck languages using more than one kind of parenthesis are not DOS languages.

Example. Let $n \geq 2$ and let D_n be the Dyck language over n letters (so the alphabet of D_Σ is $\{[_1,\dots,[_n,{}_1],\dots,{}_n]\}$). Then $D_n \notin$ DOS.

Proof.

Take $\alpha = [_1\, {}_1]$ and $\beta = [_2\, {}_2]$. Then the conclusion of Theorem 8 does not apply and so $D_n \notin L(\mathrm{DOS})$. □

Theorem 12. For each context free language K there is a PDOS language L, a regular language R and a weak identity h such that $K = h(L \cap R)$.

Proof.

Let K be a context free language.

By the Chomsky-Schützenberger Theorem (see, e.g., [S2]) there exists an integer n a regular language M and a weak identity g such that $K = g(D_n \cap M)$ where D_n is the Dyck language on n letters, assume that $\Sigma = \{a_1,\dots,a_n\}$ and $\bar{\Sigma} = \{\bar{a}_1,\dots,\bar{a}_n\}$ are letters (left and right "parenthesis") of D_n. Let $\Delta = \{b_1,\dots,b_n\}$, $\Delta \cap (\Sigma \cup \bar{\Sigma}) = \emptyset$. Let $\alpha = b_1 \dots b_n$, $\Theta = \Sigma \cup \bar{\Sigma}$ and let τ_Δ be the mapping of Θ^* defined as follows:

$\tau_\Delta(\Lambda) = \alpha$, and

for $\beta = x_1 \dots x_k$, $k \geq 1$, $x_1,\dots, x_k \in \Theta$,

$\tau_\Delta(\beta) = \alpha\, x_1\, \alpha\, x_2 \ldots \alpha\, x_k\, \alpha$.

For $K \subseteq \Theta^*$, $\tau_\Delta(K) = \bigcup_{\beta \in K} \tau_\Delta(\beta)$.

Let $R = \tau_\Delta(M)$ and let h be the homomorphism on $(\Theta \cup \Delta)^*$ defined by

$$h(x) = \begin{cases} g(x) & \text{if } x \in \Sigma \cup \overline{\Sigma}, \\ \Lambda & \text{if } x \in \Delta. \end{cases}$$

Clearly $K = g(D_n \cap M) = h(\tau_\Delta(D_n) \cap \tau_\Delta(M)) = h(\tau_\Delta(D_n) \cap R)$.

Hence to complete the proof of the theorem if suffices to show that $\tau_\Delta(D_n) \in L(DOS)$. To this aim, let $G = (\Theta, f, \alpha)$ be the DOS system where f is defined by:

- for $x \in \Sigma \cup \overline{\Sigma}$, $f(x) = x$, and
- for $1 \le i \le n$, $f(b_i) = b_i b_{i+1} \ldots b_n a_i\, \alpha\, \overline{a}_i b_1 b_2 \ldots b_i$.

It is not difficult to see that indeed $L(G) = \tau_\Delta(D_n)$. Rather than provide a rather tedious proof of this fact we give now the basic intuition underlying the equality $L(G) = \tau_\Delta(D_n)$.

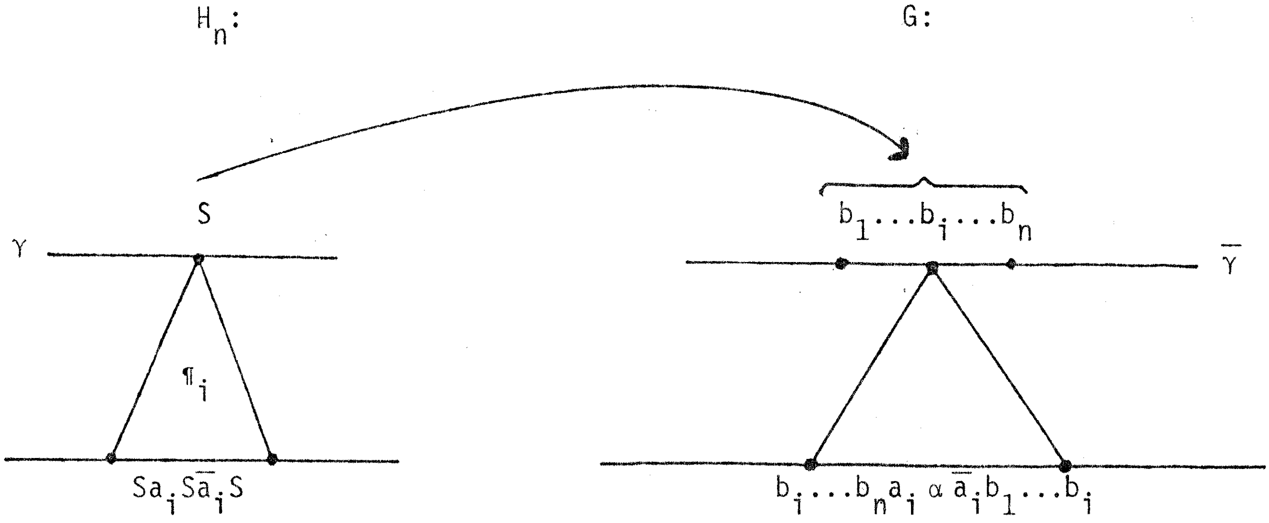
It is well known (see, e.g., [S2]) that D_n is generated by the context free grammar H_n with one nonterminal only, say S, and the following productions:

- $S \to \Lambda$, and
- for every $i \in \{1,\ldots,n\}$,

$\pi_i = (S \to Sa_i\; S\overline{a}_i S)$ is a production.

Our DOS system G does nothing else but simulates H_n in such a way that S is replaced in every sentential form by α . Then whenever an occurrence of S in a sentential form γ of H_n is replaced by π_i, in the corresponding occurrence of $\alpha = b_1 \ldots b_n$ in the corresponding sentential form (here the element of L(G)) $\overline{\gamma}$ of G the unique occurrence of b_i in (the given occurrence of α) is replaced using the production $b_i \to b_i b_{i+1} \ldots b_n a_i \alpha \overline{a}_i b_1 \ldots b_i$.

It is best illustrated by the following diagram.


H_n:
G:
S
γ
$\P_i$
$Sa_iS\bar{a}_iS$
$b_1 \ldots b_i \ldots b_n$
$\bar{\gamma}$
$b_i \ldots b_n a_i \alpha \bar{a}_i b_1 \ldots b_i$

Since G is organized in such a way that between any two consecutive occurrences of elements from $\Sigma \cup \bar{\Sigma}$ in any element of L(G) there is an occurrence of α, indeed it is intuitively clear that $L(G) = \tau_\Delta(D_n)$.

Consequently the theorem holds. □

ACKNOWLEDGMENTS

The authors are indebted to 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] Buttelmann, H.W., Pyster, A. and Reeker, L.M., Grammars without syntactic variables, University of Oregon, Dept. of Computer Science, Technical Report 74-1, 1974.

[ER1] Ehrenfeucht, A. and Rozenberg, G., On the structure of polynomially bounded DOL systems, Fundamenta Informatica, to appear.

[ER2] Ehrenfeucht, A. and Rozenberg, G., On the emptiness of the intersection of two DOS languages problem, Dept. of Computer Science, University of Colorado-Boulder, Technical Report No. CU-CS-159-79.

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

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

[R] Rozenberg, G., A systematic approach to formal language theory through parallel rewriting, Lecture Notes in Computer Science, v. , Springer-Verlag, Berlin, Heidelberg, 1979.

[RS] Rozenberg, G. and Salomaa, A., The mathematical theory of L systems, Academic Press, New York-London, to appear.

[S1] Salomaa, A., On sentential forms of context free grammars, Acta Informatica, 2, 40-49, 1973.

[S2] Salomaa, A., Formal languages, Academic Press, New York-London, 1973.

FOOTNOTES

[1]. As usual to simplify the notation we consider a finite substitution on Σ^* yielding a singleton image for each element of Σ to be a homomorphism on Σ^*.