CONTINUOUS GRAMMARS

by

A. Ehrenfeucht*
H. Maurer**
G. Rozenberg***

CU-CS-162-79 August, 1979

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

**H. Maurer, Institut für Informationsverarbeitung, Technische Universität Graz, A-8010 Graz, AUSTRIA

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

<u>All correspondence to G. Rozenberg</u>.

CONTINUOUS GRAMMARS

by

A. Ehrenfeucht

Department of Computer Science
University of Colorado-Boulder
Boulder, Colorado 80309 USA

H. Maurer

Institut für Informationsverarbeitung
Technische Universität Graz
A-8010 Graz
Austria

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

All correspondence to G. Rozenberg.

CONTINUOUS GRAMMARS

by

A. Ehrenfeucht

Department of Computer Science
University of Colorado-Boulder
Boulder, Colorado 80309 USA

H. Maurer

Institut für Informationsverarbeitung
Technische Universität Graz
A-8010 Graz
Austria

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

All correspondence to G. Rozenberg.

ABSTRACT

In this paper we continue the study of selective substitution grammars which form a framework for most of the rewriting systems studied in the literature. The continuous grammars we study in this paper generalize the basic rewriting principle of context-free grammars and EOL systems. The paper studies the language generating power of continuous grammars.

INTRODUCTION

Selective substitution grammars were introduced in [R] and further studied in [RW]. They form a framework for most of the rewriting systems studied in the literature. A selective substitution grammar is of the form $G = (\Sigma,h,\omega,K,\Delta)$ where (Σ,h,ω,Δ) is an EOL system(1) and K is a selector set, $K \subseteq (\Sigma \cup \overline{\Sigma})^*$ where $\overline{\Sigma} = \{\overline{a} : a \in \Sigma\}$. To rewrite a word α from Σ^+ in G one looks in K for a word $\tilde{\alpha}$ that differs from α only in that some occurrences of letters in α "get barred" (that is they become elements of $\overline{\Sigma}$) and then one rewrites <u>all</u> occurrences from α that are barred in $\tilde{\alpha}$. Hence G becomes a context-free grammar if $K = \Sigma^* \overline{(\Sigma \setminus \Delta)}\, \Sigma^*$ and it becomes an EOL system if $K = \overline{\Sigma}^+$. It is demonstrated in [R] how by a simple choice of the selector set one can characterize a multitude of rewriting systems. In this way one gets a unifying framework for studying rewriting systems.

This paper continues the study of selective substitution grammars by concentrating on a particular kind of selector set; we require that $K = \bigcup_{i=1}^{n} X_i^* \overline{Y}_i^+ Z_i^*$ where $X_i,\ Y_i,\ Z_i \subseteq \Sigma$ for $1 \leq i \leq n$. In this way each time a word α is rewritten it is really divided into three parts $\alpha = xyz$ where for some $1 \leq i \leq n$, $x \in X_i^*$, $y \in Y_i^+$ and $z \in Z_i^*$ and then the subword y is rewritten according to productions from h. In other words each time a word α is rewritten its <u>continuous</u> setment is rewritten. Context-free grammars are like this with the restriction that such a segment consists of one occurrence of one letter only; EOL systems are like this with the restriction that such a segment is the whole word. Hence context-free grammars and EOL systems form two extreme cases of continuous rewriting. In this way continuous grammars form a natural generalization of both context-free grammars and EOL systems.

The paper is organized as follows.

In Section I continuous grammars are introduced and illustrated by examples.

In Section II two important normal form theorems for continuous grammars are established.

In Section III we depart to study the structure of ETOL languages. The results of this section, interesting on its own, are used in Section IV.

In Section IV we compare the language generating power of ETOL systems and continuous grammars.

In Section V we consider a variation of continuous grammars. The language generating power of those grammars is considered.

We assume the reader to be familiar with rudiments of formal language theory (see, e.g., [S]) including the theory of L systems (see, e.g., [RS] - our notation concerning EOL and ETOL systems is from there).

We use mostly standard language - theoretic notation and terminology. Perhaps only the following requires an additional explanation.

(1). For a word α and a letter b, $\#_b\alpha$ denotes the number of occurrences of b in α; $|\alpha|$ denotes the length of α and Λ denotes the empty word.

(2). For a class W of rewriting systems, $\mathcal{L}(W)$ denotes the class of languages generated by systems from W.

(3). Throughout the paper, given an alphabet Σ, $\overline{\Sigma} = \{\overline{a} : a \in \Sigma\}$. Then $\overline{\Lambda} = \Lambda$ and for $\alpha = a_1 \ldots a_k$, $k \geq 1$, $a_1, \ldots, a_k \in \Sigma$, $\overline{\alpha} = \overline{a}_1 \ldots \overline{a}_k$.

I. CONTINUOUS GRAMMARS AND LANGUAGES

In this section continuous grammars are introduced and illustrated by examples.

Definition. Let n be a positive integer.

(1). An n-continuous grammar (nC grammar, for short) is a construct $G = (\Sigma,h,\omega,K,\Delta)$ where (Σ,h,ω,Δ) is an EOL system (called an underlying EOL system of G and denoted U(G)) and $K = X_1^*\bar{Y}_1^+Z_1^* \cup \ldots \cup X_n^*\bar{Y}_n^+Z_n^*$ with $X_i, Y_i, Z_i \subseteq \Sigma$ for $1 \leq i \leq n$; K is called the selector set and each $X_i^*\bar{Y}_i^+Z_i^*$ is referred to as a selector.

(2). Let $\alpha \in \Sigma^+$ and $\beta \in \Sigma^*$. We say that α directly derives β in G, written as $\alpha \underset{G}{\Rightarrow} \beta$, if there exists an i, $1 \leq i \leq n$, such that $\alpha = xyz$ for some $x \in X_i^*$, $y \in Y_i^+$, $z \in Z_i^*$ with $y = a_1 \ldots a_m$, $m \geq 1$, $a_1,\ldots,a_m \in \Sigma$ and $\beta = xy_1y_2\ldots y_mz$ where $y_j \in h(a_j)$ for $1 \leq j \leq m$.

(3). The relation $\overset{*}{\underset{G}{\Rightarrow}}$ is the transitive and the reflexive closure of the $\underset{G}{\Rightarrow}$ relation; if $x \overset{*}{\underset{G}{\Rightarrow}} y$ then we say that x derives y in G.

(4). The language of G, written as L(G), is defined by $L(G) = \{\alpha \in \Delta^* : \omega \overset{*}{\underset{G}{\Rightarrow}} \alpha\}$; L(G) is referred to as a nC language.

(5). A grammar G(language L) is continuous if it is n-continuous for some $n \geq 1$; we say that G is a C grammar (L is a C language). □

Thus to rewrite a word α in G one has to choose a selector $X_i^*\overline{Y}_i^+Z_i^*$ such that α can be written in the form $\alpha = xyz$ and the chosen selector contains the word $x\overline{y}z$, then one rewrites all letters from y according to productions in U(G), while x and z remain intact.

Definition. Let $G = (\Sigma,h,\omega,K,\Delta)$ be a n-continuous grammar where $K = \bigcup_{i=1}^{n} X_i^*\overline{Y}_i^+Z_i^*$.

(1). We say that G is n-left-continuous (nLC for short) if $Z_i = \emptyset$ for all $1 \leq i \leq n$. We also say that G is a left-continuous grammar, written as LC grammar.

(2). We say that G is n-right-continuous (nRC for short) if $X_i = \emptyset$ for all $1 \leq i \leq n$. We also say that G is a right-continuous grammar, written as RC grammar. □

We give now several examples of continuous grammars and languages.

Example 1. Let $G = (\{A,a,b\}, h, A, K, \{a,b\})$ where $h(A) = \{A^2,a\}$, $h(a) = \{a,ab,ba\}$, $h(b) = \{b\}$ and $K = X_1^*\overline{Y}_1^+Z_1^* \cup X_2^*\overline{Y}_2^+Z_2^*$ where $X_1 = Z_1 = \emptyset$, $Y_1 = \{A\}$ and $X_2 = Y_2 = Z_2 = \{a,b\}$. G is a 2-continuous grammar. It is easy to see that $L(G) = \{\alpha \in \{a,b\}^+ : \#_a\alpha = 2^n \text{ for some } n \geq 0\}$. Note that G is neither left-continuous nor right-continuous grammar. □

Example 2. Let $G = (\{a,b\}, h, bab, K, \{a,b\})$ where $h(a) = \{babab\}$, $h(b) = \{\Lambda,b,b^2\}$ and $K = X_1^*\overline{Y}_1^+Z_1^* \cup X_2^*\overline{Y}_2^+Z_2^*$, where $X_1 = Z_1 = \emptyset$, $Y_1 = X_2 = Z_2 = \{a,b\}$ and $Y_2 = \{b\}$. G is a 2-continuous grammar. It is easy to see that $L(G) = \{\alpha \in \{a,b\}^+ : \#_a\alpha = 2^n \text{ for some } n \geq 0\}$, hence we get the language from Example 1. However, this time our grammar does not have nonterminals (that is $\Sigma = \Delta$). Note that G is neither a left-continuous nor a right-continuous grammar. □

Example 3. Let $G = (\Sigma,h,\omega,K,\Delta)$ be the 3C grammar where

$\Sigma = \{S,1,\not{c},\not{S},a,b,*\}$, $\omega = S$, $\Delta = \{a,b,*\}$, $h(S) = \{aS1\not{S},*\}$, $h(*) = \{\not{c}\}$, $h(1) = \{\Lambda\}$, $h(a) = \{a,b\}$, $h(\not{c}) = \{\not{c},*\}$, $h(\not{S}) = \{\Lambda\}$ and $h(b) = \{b\}$, $K = \bigcup_{i=1}^{3} X_i^* \overline{Y_i}^+ Z_i^*$ with $X_1 = X_2 = \{a\}$, $X_3 = \emptyset$, $Y_1 = \{S\}$, $Y_2 = \{*,1\}$, $Y_3 = \{a,b,\not{c},\not{S}\}$ and $Z_1 = Z_2 = Z_3 = \{1,\not{S}\}$. It is very instructive for the reader to check that $L(G) = \{(ab^m)^n * : m \geq n \geq 1\}$.

Note that G is neither a left-continuous nor a right-continuous grammar. □

Example 4. Let $G = (\{a,b,c\}, h, cba, K, \{a,b,c\})$ be the right-continuous grammar where $K = \{a,b,c\}^* \overline{\{a,b,c\}}^+$ and the finite substitution is defined by $h(a) = \{a^2\}$, $h(b) = \{b^2\}$ and $h(c) = \{c^2\}$. It is not difficult to see that $L(G) = \{c^n b^m a^\ell : n \geq 1$, $\underline{spt}(n)$ and $\underline{spt}(m) \leq \ell\}$, where $\underline{spt}(x)$ is the smallest positive integer y of the form 2^i for some $i \geq 0$ such that $y \geq x$. □

II. NORMAL FORM THEOREMS

In this section we establish two normal form theorems for continuous grammars: one says that it sufficies to consider left- and right-continuous grammars only, the second one says that it suffices to consider continuous grammars with two selectors only.

Theorem 1. For every continuous grammar there exists an equivalent left-continuous and an equivalent right-continuous grammar.

Proof.

We will demonstrate the existence of an equivalent left-continuous grammar; the existence of an equivalent right-continuous grammar is shown analogously.

Let $G = (\Sigma, h, \omega, K, \Delta)$ be a continuous grammar with $K = K_1 \cup \ldots \cup K_n$ where $K_i = X_i^* \overline{Y}_i^+ Z_i^*$ for $1 \leq i \leq n$.

Let, for $x \in \{\ell, r, \ell\ell, \ell r\}$, $\Sigma_x = \{[a,x] : a \in \Sigma\}$ and $\Sigma_{\ell\ell,1} = \{[a,\ell\ell,1] : a \in \Sigma\}$.

Let $\hat{\Sigma} = \bigcup_{x \in \{\ell, r, \ell\ell, \ell r\}} \Sigma_x \cup \Sigma_{\ell\ell,1} \cup \Sigma.$

Let $\hat{h}$ be the finite substitution on $\hat{\Sigma}^*$ defined by:

for $a \in \Sigma$,

$\hat{h}(a) = \{[a,\ell], [a,r]\}$,

$\hat{h}([a,\ell]) = \{[a,\ell\ell], [a,\ell r]\}$,

$\hat{h}([a,\ell\ell]) = \{[a,\ell\ell,1]\}$,

$\hat{h}([a,\ell\ell,1]) = \hat{h}([a,r]) = \{a\}$, and

$\hat{h}([a,\ell r]) = \{\alpha : \alpha \in h(a)\}$.

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

$\Theta_i = \{[a,\ell\ell,1] : a \in X_i\} \cup \{[a,\ell r] : a \in Y_i\} \cup \{[a,r] : a \in Z_i\}$.

Let $M_b = \overline{\Sigma}^+$, $M_\ell = \overline{\Sigma}_\ell^+ \Sigma_p^*$, $M_{\ell\ell} = \overline{\Sigma}_{\ell\ell}^+(\Sigma_{\ell r} \cup \Sigma_r)^*$ and, for $1 \leq i \leq n$, $M_i = \overline{\Theta}_i^+$. Then let $\hat{K} = M_b \cup M_\ell \cup M_{\ell\ell} \cup \bigcup_{i=1}^{n} M_i$.

Let $H = (\hat{\Sigma}, \hat{h}, \omega, \hat{K}, \Delta)$.

M simulates G as follows.

Given a string α over Σ (the axiom ω is such a string) every letter a in it gets rewritten as $[a,\ell]$ or $[a,r]$ using selector M_b. Since M_ℓ is the only selector that can be used next, all $[a,\ell]$-letters must appear to the left of all $[a,r]$-letters. Selector M_ℓ rewrites every letter $[a,\ell]$ into $[a,\ell\ell]$ or into $[a,\ell r]$. Since $M_{\ell\ell}$ is the only selector that can be used next, all $[a,\ell\ell]$-letters must appear to the left of all $[a,\ell r]$-letters. Then if we want to simulate the application of selector K_i to α in G, $1 \leq i \leq n$, we apply selector M_i in H.

Based on the above description one can easily construct a formal proof that indeed $L(H) = L(G)$. Since H is left-continuous the theorem holds. □

<u>Theorem 2</u>. For every continuous grammar G there exists an equivalent 2-continuous grammar H. Moreover, if G is left-continuous or right-continuous then so is H.

<u>Proof</u>.

Let $G = (\Sigma, h, \omega, K, \Delta)$ be a continuous grammar with $K = K_1 \cup \ldots \cup K_n$ where $K_i = X_i^* \overline{Y}_i^+ Z_i^*$ for $1 \leq i \leq n$ with $X_i, Y_i, Z_i \subseteq \Sigma$. (If $\Theta \subseteq \Sigma$, then $\Theta_{(i)} = \{a_{(i)} : a \in \Theta\}$ and if $\alpha \in \Sigma^+$ then $\alpha_{(i)}$ denotes the word resulting from α by replacing every letter a in α by $a_{(i)}$; $\Lambda_{(i)} = \Lambda$).

Let, for $1 \leq i \leq n$, $\Sigma_{(i)} = \{a_{(i)} : a \in \Sigma\}$, $\dot{\Sigma}_{(i)} = \{\dot{a}_{(i)} : a \in \Sigma\}$ and $\hat{\Sigma} = \Sigma \cup \bigcup_{i=1}^{n} \Sigma_{(i)} \cup \bigcup_{i=1}^{n} \dot{\Sigma}_{(i)}$.

Let $\hat{h}$ be the finite substitution on $\hat{\Sigma}^*$ defined by:

for $a \in \Sigma$ and $1 \leq i \leq n-1$,

$\hat{h}(a_{(i)}) = \{a_{(i+1)}, \dot{a}_{(i+1)}, a\}$, $\hat{h}(a_{(n)}) = \{a_{(1)}, \dot{a}_{(1)}, a\}$, and

$\hat{h}(\dot{a}_{(i)}) = \{\alpha_{(i)} : \alpha \in h(a)\}$.

Let $M = \{M_1, M_2\}$, where $M_1 = \left(\overline{\bigcup_{i=1}^{n} \Sigma_{(i)}}\right)^+$ and

$$M_2 = \left(\bigcup_{i=1}^{n} (X_i)_{(i)}\right)^* \overline{\left(\bigcup_{i=1}^{n} (\dot{Y}_i)_{(i)}\right)}^+ \left(\bigcup_{i=1}^{n} (Z_i)_{(i)}\right)^*.$$

Then let $H = (\hat{\Sigma}, \hat{h}, \omega_{(1)}, M, \Delta)$. Clearly if G is a left-continuous or a right-continuous grammar then so is H. It is not difficult to see that indeed L(G) = L(H). The key observation here is that the selector M_1 rotates subscripts (i) in a word in a cyclic way so as to make it possible to simulate an arbitrary K_i. This is actually done by M_2; moreover if any letter from $\hat{\Sigma} \setminus \Sigma$ in a word α is replaced by a terminal letter (from Δ) then all letter from $\hat{\Sigma} \setminus \Sigma$ in α must be replaced by elements from Δ (and this is done by using selector M_1); otherwise the obtained word cannot be rewritten any more and one obtains a "useless setential form." □

III. AN EXCURSION IN ETOL LANGUAGES

In this section we investigate the structure of derivations in ETOL systems (Theorem 3 and Corollary 1) and then provide a new result on the combinatorial structure of ETOL languages (Theorem 4) which enables one to prove that certain languages are not ETOL languages (Corollary 2). Our Corollary 2 will be very crucial in the next section when we compare L(ETOL) with L(C). We are also convinced that the results of this section shed new light on L(ETOL).

To investigate the structure of derivations in (E)TOL systems we need the following notions.

Definition. Let $G = (\Sigma, H, \omega)$ be a TOL system.

(1). For a word $\alpha \in L(G)$ let $\underline{sh}\alpha$ denote the length of the shortest derivation of α in G.

(2). For a letter $b \in \Sigma$ let $M(b)$ denote the set of all words $\alpha \in L(G)$ satisfying the following two properties:

(i). if $\beta \in L(G)$ and $\underline{sh}\beta < \underline{sh}\alpha$ then $\#_b\beta < \#_b\alpha$, and

(ii). if $\beta \in L(G)$ and $\underline{sh}_\beta = \underline{sh}_\alpha$ then $\#_b\beta \leq \#_b\alpha$. □

Theorem 3. Let $G = (\Sigma, H, \omega)$ be a TOL system, let $\#\Sigma = m$ and let $b \in \Sigma$. If $\alpha \in M(b)$ then $\underline{sh}\alpha \leq \#_b\alpha \cdot m^m$.

Proof.

Let $\overline{G} = (\Sigma, \overline{H}, \omega)$ be the combinatorially complete version of G, that is $\overline{G}$ is a DTOL system such that

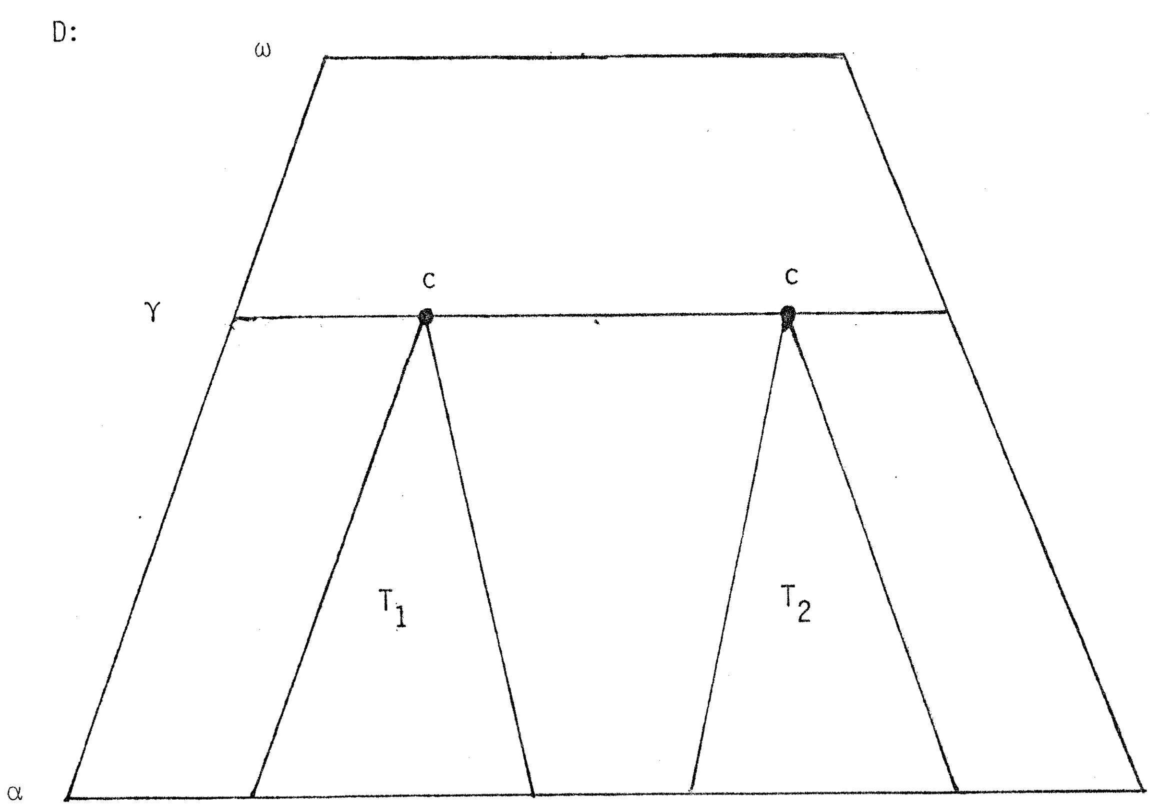
$\overline{H} = \{\overline{h} : \overline{h}$ is a homomorphism on Σ^* and for some h in H, $\overline{h} \subseteq h\}$.

(i). For every $b \in \Sigma$ and for every $\alpha \in M(b)$ there exists a $\beta \in M(b) \cap L(\overline{G})$ such that $\underline{sh}\alpha = \underline{sh}\beta$ and $\#_b\alpha = \#_b\beta$.

Proof of (i):

Assume that $\alpha \in M(b)$ and consider a shortest derivation D of α in G (that is a derivation of α of the length $\underline{sh}\alpha$). If in $\underline{trace}D$ (that is in the sequence of words occurring in D except for α) every word is such that if it contains at least two occurrences of the same letter then all those occurrences contribute equal subwords to α, then clearly $\alpha \in M(b) \cap L(\overline{G})$ and (i) trivially holds.

Hence let us assume that $\underline{trace}D$ contains a word γ such that γ contains two occurrences 0_1 and 0_2 of a letter (say c) which contribute different subwords to α; let T_1 and T_2 be subtrees of the derivation forest T_D of the derivation D rooted at 0_1 and 0_2 respectively. The situation is represented by the following picture (we assume that 0_1 is to the left of 0_2):



Let us modify D in such a way that we replace T_2 by T_1. Clearly in this way we obtain a derivation $\hat{D}$ of a word $\hat{\alpha}$ in G. Note that both D and $\hat{D}$ are of the same length.

Notice that $\#_b\alpha = \#_b\hat{\alpha}$. Otherwise one of T_1, T_2 (say T_1) contributes more occurrences of b in α (in D) than the other. Then $\#_b\hat{\alpha} > \#_b\alpha$ which means that $\underline{sh}\hat{\alpha} \leq \underline{sh}\alpha$ and $\#_b\hat{\alpha} > \#_b\alpha$ contradicting the fact that $\alpha \in M(b)$. (Symmetrically if T_2 contributes more occurrences of b in α (in D) than T_1 then if $\hat{\alpha}$ is the word obtained by replacing T_1 by T_2, the above reasoning holds again).

This implies that $\underline{sh}\hat{\alpha} = \underline{sh}\alpha$.

If we iterate the above process going (say top-down) through all the words in <u>trace</u> D then we arrive at a derivation tree $\overline{D}$ of a word β such that $\underline{sh}\beta = \underline{sh}\alpha$ and $\#_b\beta = \#_b\alpha$ where $\overline{D}$ is also a derivation tree in $\overline{G}$. Hence (i) holds. □

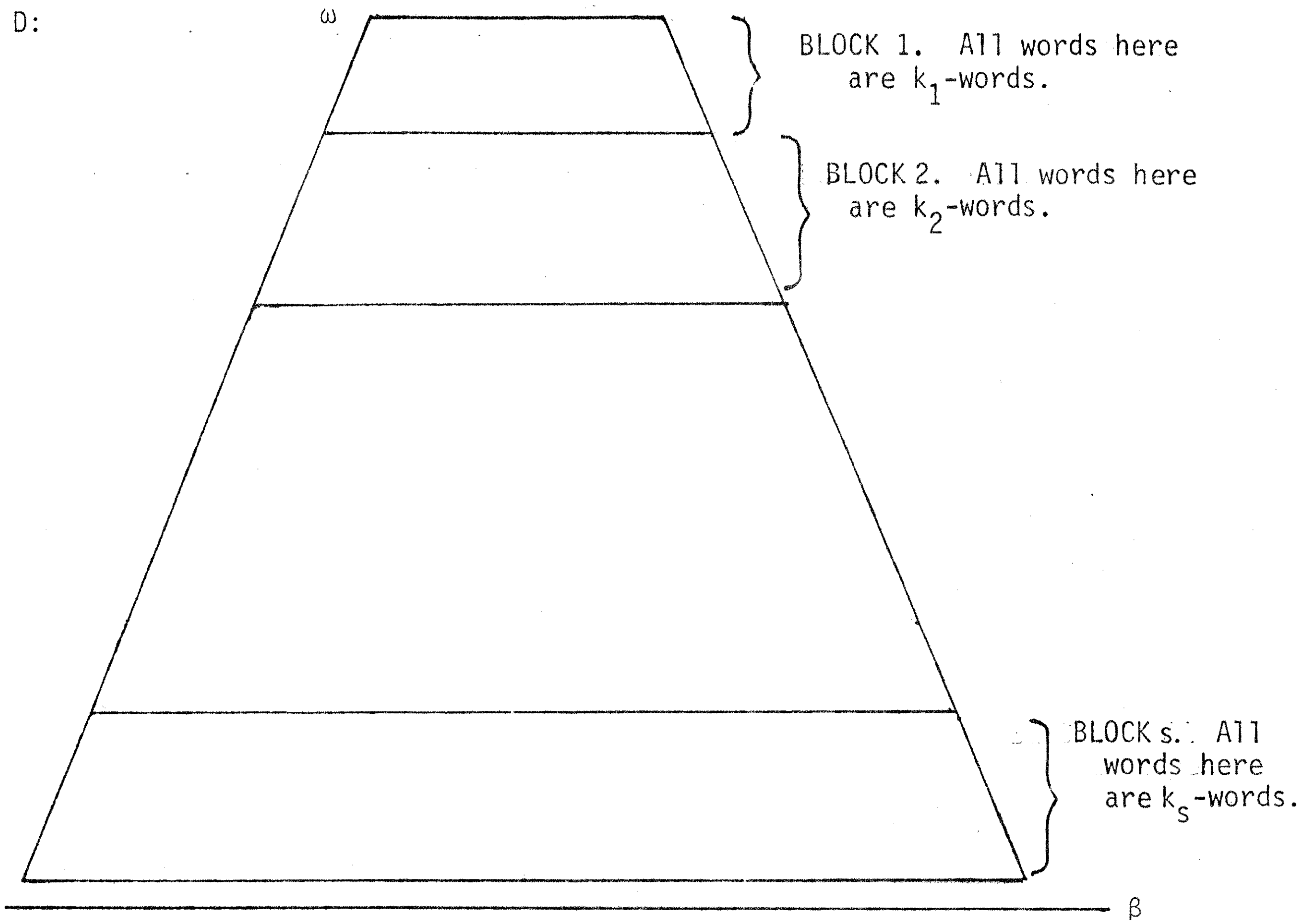
(ii). Let $b \in \Sigma$ and let $\beta \in M(b) \cap L(\overline{G})$. Then $\underline{sh}_\beta \leq \#_b\beta \cdot m^m$.

<u>Proof of (ii)</u>.

Let D be a shortest derivation of β in $\overline{G}$. For a word γ in <u>trace</u> D and a nonnegative integer k we say that γ is a k-<u>word</u> if γ contains exactly k occurrences each of which contributes in D at least one occurrence of b in β.

Now we can divide words in <u>trace</u> D in blocks grouping together all consecutive words that are k-words for some k. In this way we get the following situation:

D:



Since an ancestor of an occurrence that contributes an occurrence of b to β also contributes an occurrence of b to β, we get $k_1 < k_2 < \ldots < k_s$.

Let us consider now a block, say BLOCK i for $1 \le i \le s$. Let it consist of ℓ_i words. We claim that $\ell_i \le m^m$. This is seen as follows. If $\ell_i = 1$ then clearly $\ell_i \le m^m$. Otherwise we proceed as follows.

Let γ_j be the j'th word in BLOCK i, $1 \le j \le \ell_i$ and let $X_{j,1},\ldots,X_{j,k_i}$ be the sequence of all occurrences in γ_j (occurring in γ_j in this order) that contribute at least one occurrence of a to β. In this way for each $2 \le j \le \ell_i$ we get a ("being an ancestor of") function g_j mapping $\{X_{1,1},\ldots,X_{1,k_i}\}$ onto $\{X_{j,1},\ldots,X_{j,k_i}\}$ such that

$g_j(X_{1,t}) = X_{j,t}$ for $1 \le t \le k_i$. Since $\overline{G}$ is a DTOL system we can consider g to be a function from a subset of Σ into a subset of Σ.

Now to prove that $\ell_i \le m^m$ we assume otherwise, i.e., $\ell_i > m^m$. Then clearly either g_r is the identity mapping (on the set of letters corresponding to $X_{1,1},\ldots,X_{1,k_i}$) or there exist r, s, $2 \le r < s \le \ell_i$ such that $g_r = g_s$. Let us consider the first possibility first. Then we can modify D to $\overline{D}$ in such a way that we do not change D up to γ_r and then we apply the sequence of tables leading from γ_s to β. Note that in this way we get a word $\overline{\beta}$ (the result of $\overline{D}$) such that $\underline{sh}\overline{\beta} < \underline{sh}\beta$ and $\#_b\overline{\beta} \ge \#_b\beta$. This contradicts the fact that $\beta \in M(b)$ and so it must be that $\ell_i \le m^m$.

On the other hand it is clear that $s \le \#_b\beta$ and so $\underline{sh}\beta \le \#_b\beta \cdot m^m$ which proves (ii). □

Clearly (i) and (ii) imply the theorem. □

Corollary 1. Let $G = (\Sigma, H, \omega)$ be a TOL system, where $\#\Sigma = m$, the maximal length of the right-hand side of a production in G equals n, $|\omega| = q$, and let $b \in \Sigma$. If $\alpha \in M(b)$ and $\#_b\alpha = r$ then $|\alpha| \le qn^{rm^m}$.

Proof.

Directly from Theorem 3. □

Now using the above result we can prove a result on the combinatorial structure of languages in L(ETOL).

Theorem 4. Let K be an ETOL language over an alphabet Σ and let $b \in \Sigma$ be such that $\{\#_b\alpha : \alpha \in K\}$ is infinite. There exist a positive integer constant Q, an infinite strictly growing sequence $\{n_i\}_{i \in \mathbf{N}^+}$ of positive integers and an infinite sequence $\{\alpha_i\}_{i \in \mathbf{N}^+}$ of words from K such that $\#_b\alpha_i = n_i$ and $|\alpha_i| < Q^{n_i}$ for each $i \in \mathbf{N}^+$.

Proof.

Let K and b satisfy the assumption of the theorem.

It is well-known (see [ER3]) that each ETOL language is a coding of a TOL language. Thus let $G = (\Theta, H, \omega)$ be a TOL system and let f be a coding such that $K = f(L(G))$.

Since $\{\#_b\alpha : \alpha \in K\}$ is infinite, there exists a letter $d \in \Sigma$ such that $f(d) = b$, $\{\#_d\alpha : \alpha \in L(G)\}$ is infinite and consequently M(d) is infinite. Let $\beta \in M(d)$. By Corollary 1 there exists a positive integer constant Q such that $|\beta| \leq Q^{\#_d\beta}$. Since $|f(\beta)| = |\beta|$ and $\#_b f(\beta) \geq \#_d\beta$ we get

$$|f(\beta)| \leq Q^{\#_b f(\beta)} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (*).$$

Let $W = \{f(\beta) : \beta \in M(d)\}$. Then clearly $Z = \{\#_b\alpha : \alpha \in W\}$ is infinite. Let us choose from W an infinite sequence $\{\alpha_i\}_{i \in \mathbf{N}^+}$ of nonempty words such that $\#_b\alpha_i < \#_b\alpha_{i+1}$ for $j \in N^+$ and let $\{n_i\}_{i \in \mathbf{N}^+}$ be the corresponding sequence of lengths (that is $n_i = |\alpha_i|$ for $i \in \mathbf{N}^+$). Then (*) implies that the theorem holds. □

Corollary 2. Let K be a language such that $K \subseteq \{b^n a^{2^m} : n \geq 0,\ m \geq 2^n\}$ and $\{\#_b\alpha : \alpha \in K\}$ is infinite. Then $K \notin \mathcal{L}(ETOL)$.

Proof.

Assume to the contrary that K is an ETOL language. Then K satisfies the assumptions of Theorem 3. So let Q, $\{n_i\}_{i \in \mathbf{N}^+}$ and $\{\alpha_i\}_{i \in \mathbf{N}^+}$ satisfy the conclusion of Theorem 3. Then however we get that, for each $i \in \mathbf{N}^+$, $2^{2^{n_i}} \leq |\alpha_i| < Q^{n_i}$; a contradiction.

Hence the corollary holds. □

IV. COMPARISON WITH L(ETOL)

In this section we locate the position of L(EOL) and L(ETOL) within the framework of continuous grammars.

First of all we can characterize L(EOL) and L(ETOL) by requiring some natural restrictions on the form of selectors in a continuous grammar.

Definition. Let $G = (\Sigma, h, \omega, K, \Delta)$ be a n continuous grammar with $K = \bigcup_{i=1}^{n} X_i^* \bar{Y}_i^+ Z_i^*$. We say that G is a simple n-continuous grammar (abbreviated as nSC grammar) if $X_i = Z_i = \emptyset$ for each $1 \le i \le n$. We also say that G is a simple continuous grammar abbreviated as SC grammar. □

Theorem 5. Let L be a language. $L \in L(\mathrm{EOL})$ if and only if $L \in L(1\mathrm{SC})$.

Proof.

Obvious. □

Theorem 6. $L(\mathrm{ETOL}) = L(\mathrm{SC})$.

Proof.

(i). Let $G = (\Sigma, H, \omega, \Delta)$ be an ETOL system where $H = \{h_1, \ldots, h_n\}$ for some $n \ge 1$. Let, for $1 \le i \le n$, $\Sigma_{(i)} = \{a_{(i)} : a \in \Sigma\}$ and let $\Theta = \bigcup_{i=1}^{n} \Sigma_{(i)} \cup \Sigma$. (For a word $\alpha \in \Sigma^+$, $\alpha_{(i)}$ denotes the word obtained from α by replacing in it every occurrence of every letter a by $a_{(i)}$.)

Let $\hat{G} = (\Theta, h, \omega_{(1)}, K)$ be a SC grammar, where $K = \bigcup_{i=1}^{n} \bar{\Sigma}_{(i)}^+$ and h is defined as follows:

for $a \in \Sigma$, $1 \le i \le n-1$,

$h(a_{(i)}) = \{a_{(i+1)}, a\} \cup \{\alpha_{(i)} : \alpha \in h_i(a)\}$

$h(a_{(n)}) = \{a_{(1)}, a\} \cup \{\alpha_{(n)} : \alpha \in h_i(a)\}$,

$h(a) = \{a\}$.

It is easy to see that $L(\hat{G}) = L(G)$, consequently $\mathcal{L}(\text{ETOL}) \subseteq \mathcal{L}(\text{SC})$.

(ii). Let $G = (\Sigma, h, \omega, K, \Delta)$ be a simple continuous grammar with $K = \bigcup_{i=1}^{n} \overline{Y}_i^{+}$. Let for $1 \le i \le n+1$, $\Sigma_{(i)} = \{a_i : 1 \le i \le n\}$, F be a new symbol, $\Theta = \bigcup_{i=1}^{n} \Sigma_{(i)} \cup \Sigma \cup \{F\}$. Let, for $1 \le i \le n+1$, h_i be the finite substitution on Θ^* defined by:

for $a \in \Sigma$, $1 \le j \le n$, $1 \le i \le n$,

$$h_i(a_{(j)}) = \begin{cases} \{\alpha_{(i)} : \alpha \in h(a)\} & \text{if } i = j \text{ and } a \in Y_i, \\ \{F\}, & \text{otherwise,} \end{cases}$$

$$h_{n+1}(a_{(j)}) = \begin{cases} \{a_{(j+1)}, a\} & \text{if } j < n, \\ \{a_{(1)}, a\} & \text{if } j = n, \end{cases}$$

$h_i(a) = h_i(F) = h_{n+1}(a) = h_{n+1}(F) = \{F\}$.

Let $\hat{G} = (\Theta, H, \omega_{(1)}, \Delta)$ be the ETOL system where $H = \{h_1, \ldots, h_{n+1}\}$. It is easy to see that $L(\hat{G}) = L(G)$, hence $\mathcal{L}(\text{SC}) \subseteq \mathcal{L}(\text{ETOL})$. □

Corollary 3. $\mathcal{L}(\text{SC}) = \mathcal{L}(\text{2SC})$.

Proof.

It follows from the observation that the algorithm from the proof of Theorem 2 produces a SC grammar if the original grammar is SC. It also follows from the proof of the previous theorem and a well-known fact that every ETOL language can be generated by an ETOL system with two tables only. □

We are going now to compare $L(\mathrm{EOL})$ and $L(\mathrm{ETOL})$ with $L(\mathrm{C})$. First of all let us notice that Example 1 (and so also Example 2) provides a 2C language that is well-known not to be an EOL language (see, e.g., [ER1]). Then Example 3 provides an example of a 3C language (hence, by Theorem 2, a 2C language) that is well-known not to be an ETOL language (see [ER2]). We are going to sharpen those observations now.

By Theorem 5 we know that $L(\mathrm{EOL}) \subseteq L(1\mathrm{SC})$. In looking for a candidate for a language $L \in L(1\mathrm{SC}) \setminus L(\mathrm{EOL})$ one may be inclined (as we were!) to think that the RC language $\{c^n b^m a^\ell : n \geq 1, \underline{spt}(n) \leq m$ and $\underline{spt}(m) \leq \ell\}$ from Example 4 may do the job. It has the "right to left orientation" which seems to be impossible to achieve in EOL languages. However, it turns out that one can generate this language by an EOL system! Our next example is showing how it is done. We find this quite instructive for the reader to go through this example as, in our opinion, it is one of the very few "concrete" nontrivial examples of EOL systems.

Example 5. Let $G_1 = (\Sigma_1, h_1, S, \{a,b,c\})$, $G_2 = (\Sigma_2, h_2, CB^2A^2, \{a,b,c\})$ and $G_3 = (\Sigma_3, h_3, CBA, \{a,b,c\})$ be EOL systems such that

$\Sigma_1 = \{S,T,R,A,B,C,a,b,c,F\}$,

$\Sigma_2 = \{A,B,C,a,b,c,F\}$,

$\Sigma_3 = \Sigma_2$,

$h_1(S) = \{TA\}$, $h_1(T) = \{RB,T\}$, $h_1(R) = \{C,R\}$, $h_1(A) = \{A^2,a,a^2\}$,

$h_1(B) = \{B^2,b,b^2\}$, $h_1(C) = \{C^2,c,c^2\}$, $h_1(a) = h_1(b) = h_1(c) = h_1(F) = \{F\}$,

$h_2(A) = \{A^2,a,a^2\}$, $h_2(B) = \{B^2,b\}$, $h_2(C) = \{C^2,c,c^2\}$,

$h_2(a) = h_2(b) = h_2(c) = h_2(F) = \{F\}$,

$h_3(A) = \{A^2,a,a^2\}$, $h_3(B) = \{B^2,b\}$, $h_3(C) = \{C^2,c\}$

$h_3(a) = h_3(b) = h_3(c) = h_3(F) = \{F\}$.

Let $L = \{c^n b^m a^\ell : n \geq 1$, $\underline{spt}(n) \leq m$ and $\underline{spt}(m) \leq \ell\}$.

We claim that $L = L(G_1) \cup L(G_2) \cup L(G_3)$.

To see this let us first divide L into five sublanguages as follows. Note that each word α in L can be written in the form

$$\alpha = c^n b^m a^\ell = c^{2^{n_1}+n_2} b^{2^{m_1}+m_2} a^{2^{\ell_1}+\ell_2},$$

where $\underline{spt}(n) \leq m$, $\underline{spt}(m) \leq \ell$, $2^{n_1} \leq n$, $2^{n_1+1} > n$, $2^{m_1} \leq m$, $2^{m_1+1} > m$, $2^{\ell_1} \leq \ell$ and $2^{\ell_1+1} > \ell$.

With the form of α in L as above we define now

$L_1 = \{\alpha \in L : n_1 < m_1 < \ell_1\}$,

$L_2 = \{\alpha \in L ; n_1 = m_1 < \ell_1\}$,

$L_3 = \{\alpha \in L : n_1 + 1 < m_1 = \ell_1\}$,

$L_4 = \{\alpha \in L : n_1 + 1 = m_1 = \ell_1\}$, and

$L_5 = \{\alpha \in L : n_1 = m_1 = \ell_1\}$.

Note that as the consequence of the above definitions we get that

- if $\alpha \in L_2$ then $n_2 = 0$,
- if $\alpha \in L_3$ then $m_2 = 0$,
- if $\alpha \in L_4$ then $m_2 = 0$, and
- if $\alpha \in L_5$ then $n_2 = 0$ and $m_2 = 0$.

Clearly $L(G_1) \subseteq L$, $L(G_2) \subseteq L$ and $L(G_3) \subseteq L$.

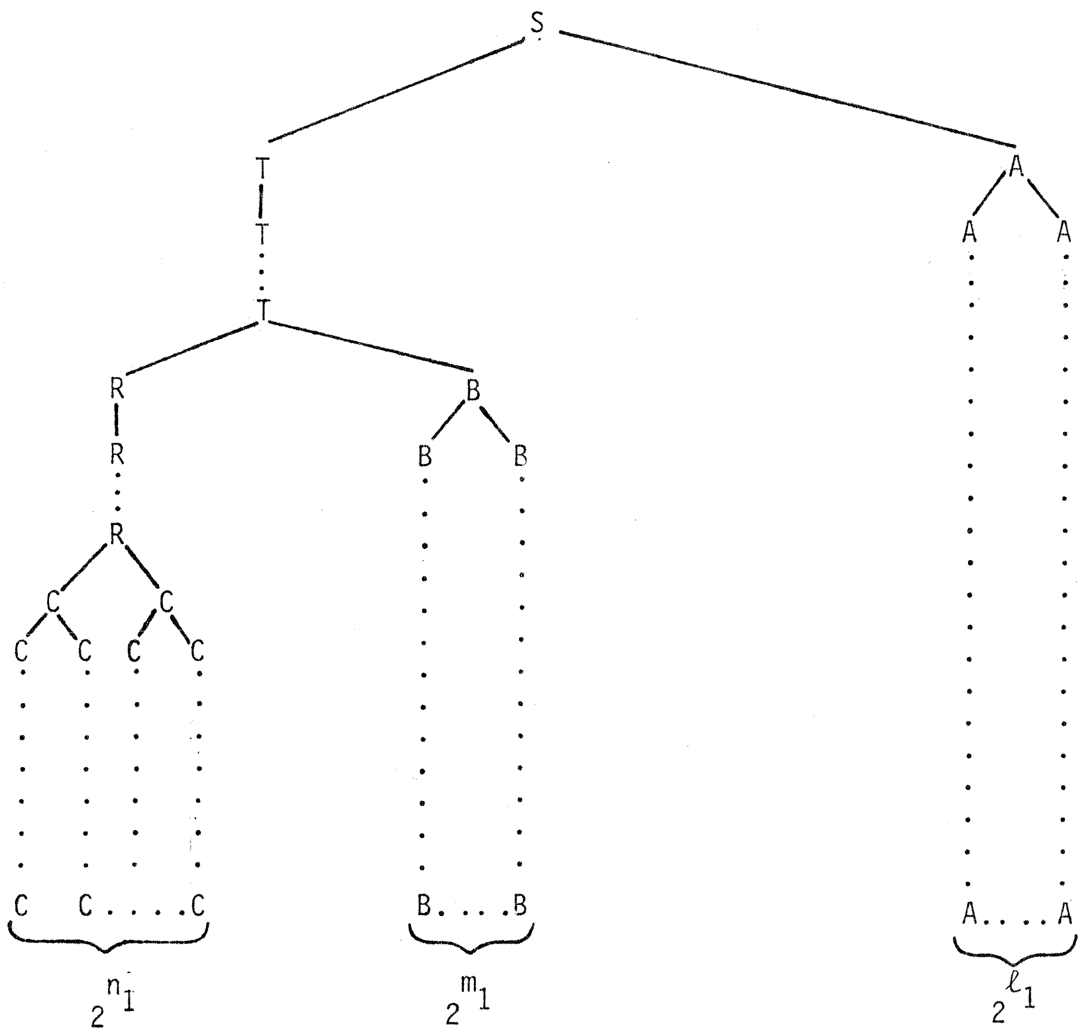
On the other hand we have the following.

(i). $L_1 \cup L_2 \cup L_3 \subseteq L(G_1)$.

This is proved as follows.

(i.1). Let $\alpha \in L_1$, $\alpha = c^{2^{n_1}+n_2} b^{2^{m_1}+m_2} a^{2^{\ell_1}+\ell_2}$.

We first derive $C^{2^{n_1}} B^{2^{m_1}} A^{2^{\ell_1}}$ in the way that corresponds to the following derivation tree:



We can do this because $n_1 < m_1 < \ell_1$. Now to n_2 occurrences of C we apply the production $C \to c^2$, to m_2 occurrences of B we apply the production $B \to b^2$ and to ℓ_2 occurrences of A we apply the production $A \to a^2$; all other occurrences are rewritten using productions $C \to c$, $B \to b$ and $A \to a$. In this way we get α in $L(G_1)$.

(i.2). Let $\alpha \in L_2$, $\alpha = c^{2^{n_1}} b^{2^{m_1}+m_2} a^{2^{\ell_1}+\ell_2}$.

Analogously to the way described in (i.1) we first derive $C^{2^{n_1-1}} B^{2^{m_1}} A^{2^{\ell_1}}$. We can do this because $n_1 - 1 < m_1 < \ell_1$. Then we rewrite all occurrences of C using production $C \to c^2$, m_2 occurrences of B using production $B \to b^2$ and ℓ_2 occurrences of A using production $A \to a^2$; all other occurrences are rewritten using productions $B \to b$ and $A \to a$. In this way we generate α in G_1.

(i.3). Let $\alpha \in L_3$, $\alpha = c^{2^{n_1}+n_2} b^{2^{m_1}} a^{2^{\ell_1}+\ell_2}$.

Analogously to the way described in (i.1) we first derive $C^{2^{n_1}} B^{2^{m_1-1}} A^{2^{\ell_1}}$. We can do this because $n_1 < m_1-1 < \ell_1$. Then we rewrite n_2 occurrences of C using production $C \to c^2$, all occurrences of B using production $B \to b^2$ and ℓ_2 occurrences of A using production $A \to a^2$; all other occurrences are rewritten using productions $C \to c$ and $A \to a$. In this way we get α in $L(G_1)$.

(ii). $L_4 \subseteq L(G_2)$.

This is proved as follows. Let $\alpha = c^{2^{n_1}+n_2} b^{2^{m_1}} a^{2^{m_1}+\ell_2}$.

We first derive in n_1 steps the word $C^{2^{n_1}} B^{2^{n_1+1}} A^{2^{n_1+1}}$. Then we rewrite n_2 occurrences of C using production $C \to c^2$, all occurrences of B using production $B \to b$ (this is the only "finishing production" for B in G) and ℓ_2 occurrences of A using production $A \to a^2$; all other occurrences are rewritten using productions $C \to c$ and $A \to a$. In this way we get α in $L(G_2)$.

(iii). $L_5 \subseteq L(G_3)$.

This is seen as follows. Let $\alpha = c^{2^{n_1}} b^{2^{n_1}} a^{2^{n_1}+\ell_2}$. We first derive in n_1 steps the word $C^{2^{n_1}} B^{2^{n_1}} A^{2^{n_1}}$. Then we rewrite ℓ_2 occurrences of A using production $A \to a^2$; all other occurrences are rewritten using productions $C \to c$, $B \to b$ and $A \to a$. In this way we get α in $L(G_3)$.

Since a finite union of EOL languages is an EOL language we have demonstrated that L is an EOL language. □

Hence we are still left with the task of finding an example of a 1C language that is not an EOL language. It turns out that there exist 1C languages that are not ETOL languages, which will be demonstrated now.

Let $G_0 = (\Sigma, h, \omega, K, \Delta)$ be the 1LC system where

$\Sigma = \Sigma_1 \cup \Sigma_2$, $\Sigma_1 = \{a,b,c,d\}$, $\Sigma_2 = \{A,B,C,D,X\}$,

$\omega = AX$,

$\Delta = \{a,b,c,d,X\}$,

$K = \Sigma_1^* \bar{\Sigma}_2^+$, and

$h(A) = \{ABC, aBC\}$, $h(B) = \{BC, bC\}$, $h(C) = \{C, cD\}$, $h(X) = \{X^2\}$ and $h(D) = \{d\}$.

Let f be the weak identity on Δ^* defined by $f(X) = X$, $f(b) = b$, $f(a) = f(c) = f(d) = \Lambda$.

<u>Lemma 1</u>. $L(G_0) \notin \mathcal{L}(ETOL)$.

<u>Proof</u>.

Let us consider a "typical" derivation D of a word in G_0. It can be pictured as follows:

part 1

$$
\left\{
\begin{array}{l}
A\,X \\
A\,B\,C\,X^{2} \\
A\,B\,C\,B\,C^{2}\,X^{2^{2}} \\
\quad\vdots \\
A\,B\,C\,B\,C^{2}\,\ldots\,B\,C^{n}\,X^{2^{n}}
\end{array}
\right.
$$

part 2

$$
\left\{
\begin{array}{lll}
& z = & a\,B\,C\,B\,C^{2}\ldots B\,C^{n+1}\,X^{2^{n+1}} \\
& & \vdots \\
r_1 \geq 0 & & a\,B\,C^{r_1+1}\,B\,C^{r_1+2}\ldots B\,C^{n+1+r_1}\,X^{2^{n+1+r_1}} \\
& y_1 = & a\,b\,C\,C^{r_1+1}\,B\,C^{r_1+3}\ldots B\,C^{n+1+r_1+1}\,X^{2^{n+1+r_1+1}} \\
& & a\,b\,c\,D\,C^{r_1+1}\,B\,C^{r_1+4}\ldots B\,C^{n+1+r_1+2}\,X^{2^{n+1+r_1+2}} \\
& & abcdcDc^{r_1}\,B\,C^{r_1+5}\ldots B\,C^{n+1+r_1+3}\,X^{2^{n+1+r_1+3}} \\
& & \vdots \\
& & ab(cd)^{r_1+2}\,B\,C^{2(r_1+2)}\ldots B\,C^{n+1+r_1+r_1+2}\,X^{2^{n_1+1+r_1+r_1+2}} \\
& & \vdots \\
r_2 \geq 0 & & ab(cd)^{r_1+2}\,B\,C^{2(r_1+2)+r_2}\ldots B\,C^{n+2r_1+3+r_2}\,X^{2^{n_1+2r_1+3+r_2}} \\
& y_2 = & ab(cd)^{r_1+2}\,b\,C^{2(r_1+2)+r_2+1}\ldots B\,C^{n+2r_1+3+r_2}\,X^{2^{n_1+2r_1+3+r_2+1}} \\
& & \vdots \\
& & ab(cd)^{\ell_1}\;b(cd)^{\ell_2}\;\ldots\;b(cd)^{\ell_{n+1}}\;X^{2^{q}}
\end{array}
\right.
$$

Note that because $K = \Sigma_1^* \bar{\Sigma}_2$, in every word occurring in D all occurrences of all letters from Σ_2 are to the right of all occurrences of all letters from Σ_1. We can drive D in two parts, part 1 consisting of all words containing an occurrence of A and part 2 containing all words containing an occurrence of a. Let "block" be a subword (of a word in D) beginning with an occurrence of B or b and ending on an occurrence just before an occurrence of B, b or X. Note that the first word (z) in part 2 determines the number of blocks (n+1) in any subsequent word in D. Because of the form of K blocks in words in D have the left-to-right priority order, that is first the leftmost block must be rewritten into a terminal (sub)word, then the second from the left block must be rewritten into a terminal (sub)word, etc. So we can talk about the first, second, third, etc. block in D. Let "the representation of block i" be the subword corresponding to this block of the form $bC^{\rho(i)}$ appearing for the first time. Thus in D above bC^{r_1+2} is the representation of block 1 (it appears in y_1) and $bC^{2(r_1+2)+r_2+1}$ is the representation of block 2.

Clearly

$\rho(1) \geq 2$, and

$\rho(i+1) \geq 2\rho(i)$, for $i \geq 1$, . . . (**).

It is also clear that

$\ell_i = \rho(i)$, for $i \geq 1$, . . . (***).

Moreover, because at each step of a derivation the production $X \to X^2$ is applied, $q > 2^{\ell_{n+1}}$.

Now (**) implies that $\rho(n+1) \geq 2^{n+1}$ and so (***) implies that $q \geq 2^{2^{n+1}}$.

On the basis of the above analysis it is not difficult to show that $f(L(G_0))$ satisfies the assumptions of Corollary 2 (set b = b and a = X). Consequently $f(L(G_0)) \notin \mathcal{L}(ETOL)$. Since it is well-known (see, e.g., [RS]) that $\mathcal{L}(ETOL)$ is closed with respect to homomorphic mappings, $L(G_0) \notin \mathcal{L}(ETOL)$. □

Consequently we get the following results.

Theorem 7. $\mathcal{L}(1LC) \setminus \mathcal{L}(ETOL) \neq \emptyset$.

Proof.

Directly from Lemma 1. □

Corollary 4. (i). $\mathcal{L}(EOL) \subsetneq \mathcal{L}(1C)$.

(ii). $\mathcal{L}(ETOL) \subsetneq \mathcal{L}(2C)$.

Proof.

(i) follows from Theorem 5 and Theorem 7, and (ii) follows from Theorem 6, Corollary 3 and Theorem 7. □

V. MAXIMALLY CONTINUOUS GRAMMARS

Although, clearly, $\mathcal{L}(C) \subseteq \mathcal{L}(RE)$ we do not know whether or not $\mathcal{L}(RE) \subseteq \mathcal{L}(C)$. In this section we introduce a very natural variation of continuous grammars, called maximally continuous grammars, and demonstrate that they generate precisely the class of $\mathcal{L}(RE)$.

Definition. (1). Let $G = (\Sigma, h, \omega, K, \Delta)$ be a nC grammar where $K = \bigcup_{i=1}^{n} X_i^* \overline{Y}_i^+ Z_i^*$. The direct maximal derivation relation $\underset{G;max}{\Longrightarrow}$ is defined as follows.

For $\alpha \in \Sigma^+$, $\beta \in \Sigma^*$, $\alpha \underset{G;max}{\Longrightarrow} \beta$ if there exists an i, $1 \leq i \leq n$, such that $\alpha = xyz$ for some $x \in X_i^*$, $y \in Y_i^*$, $z \in Z_i^*$ with $y = a_1 \ldots a_m$, $m \geq 1$, $a_1, \ldots, a_m \in \Sigma$, $\beta = x y_1 \ldots y_m z$ where $y_j \in h(a_j)$ for $1 \leq j \leq m$ and the following condition is satisfied:

if $x = x_1 a$ for $a \in \Sigma$ then $a \notin Y_i$ and if $z = a z_1$ for $a \in \Sigma$ then $a \notin Y_i$.

Then $\overset{*}{\underset{max;G}{\Longrightarrow}}$ denotes the reflexive and the transitive closure of $\underset{max;G}{\Longrightarrow}$.

(2). The maximally n-continuous grammar (or nMC grammar for short) is a nC grammar $G = (\Sigma, h, \omega, K, \Delta)$, where the direct derivation relation $\underset{G}{\Longrightarrow}$ is replaced by direct maximal derivation relation $\underset{max;G}{\Longrightarrow}$. Hence the language of G is defined by $L(G) = \{\alpha \in \Delta^* : \omega \overset{*}{\underset{max;G}{\Longrightarrow}} \alpha\}$; L(G) is referred to as a nMC language.

(3). A grammar G (language L) is continuous if it is n-continuous for some $n \geq 1$; we say that G is a MC grammar (L is a MC language). □

Example 5. Let $G = (\{a,b\}, h, bab, K, \{a,b\})$ be a 1MC grammar where $K = \{a,b\}^* \overline{\{a\}}^+ \{a,b\}^*$, $h(a) = \{a^2, aba\}$ and $h(b) = \{b\}$. Then $L(G) = \{b a^{2^{n_1}} b a^{2^{n_2}} b \ldots b a^{2^{n_m}} b : m \geq 1, n_1, \ldots, n_m \geq 0\}$. Note that if we consider G to be a 1C grammar then

$L(G) = \{ba^{n_1}ba^{n_2}b\ldots ba^{n_m}b : m \geq 1, n_1,\ldots,n_m \geq 1\}$. □

First of all let us notice that the proofs of Theorem 1 and Theorem 2 carry over to maximally continuous grammars (with the notions of maximally left-continuous, written LMC, and maximally right-continuous written RMC, grammars, defined in the obvious way).

Theorem 8. For every MC grammar there exists an equivalent LMC grammar and an equivalent RMC grammar. □

Theorem 9. For every MC grammar G there exists an equivalent 2MC grammar H. Moreover, if G is RMC or LMC then so is H. □

We investigate now the language generating power of MC grammars. We term a MC grammar G propagating, denoted PMC grammar if $G = (\Sigma, h, \omega, K, \Delta)$ and, for every $a \in \Sigma$, $\Lambda \notin h(a)$.

Theorem 10. (1). $L(PMC) = L(CS)$, (2). $L(MC) = L(RE)$.

Proof.

We will prove (1) and then (2) follows from the well-known fact that adding erasing productions to context-sensitive productions yields the class of grammars generating $L(RE)$.

Since it is straightforward to construct a linear bounded automaton to accept the language of a given PMC grammar, $L(PMC) \subseteq L(CS)$.

To show that $L(CS) \subseteq L(PMC)$ we proceed as follows.

Let $G = (\Sigma, P, S, \Delta)$ be a context-sensitive grammar where Σ is the total alphabet of G, Δ is its terminal alphabet, P its set of productions, Δ its terminal alphabet and S its axiom. We can assume that G is in the Penttonen normal form (see [P]), that is $P = P_1 \cup P_2 \cup P_3$ where all productions in P_1 are of the form $A \to a$, $A \in \Sigma \backslash \Delta$, $a \in \Delta$, all productions in P_2 are of the form $A \to BC$ with

$A,B,C \in \Sigma\backslash\Delta$ and all productions in P_3 are of the form $AB \rightarrow AC$ with $A,B,C \in \Sigma\backslash\Delta$. Let $\#P = r$ and we assume that productions in P are ordered, hence, for $1 \leq j \leq r$, we can talk about the j'th production of P denoted π_j; then $\underline{\ell hs}\pi_j$ denotes the left-hand side of π_j and $\underline{rhs}\pi_j$ denotes the right-hand side of π_j.

We assume that G is organized in such a way that the rightmost symbol of any sentential form α in G ($\alpha \notin L(G)$) is always marked in a special way, hence if G rewrites such a symbol then it knows that this is the rightmost symbol. Clearly every context-sensitive language can be generated by a context-sensitive grammar G satisfying the above assumption.

Let $\Sigma_b = \{[A,i] : A \in \Sigma,\ 1 \leq i \leq 3\}$,

for $s \in \{0,1,4,5\}$

$\Sigma_s = \{[A,i,j,s] : A \in \Sigma,\ 1 \leq i \leq 3 \text{ and } 1 \leq j \leq r\}$, and

for $s \in \{2,3\}$

$\Sigma_s = \{[A,i,-,s] : A \in \Sigma,\ 1 \leq i \leq 3\}$.

Then let $\Theta = \Sigma_b \cup \bigcup_{s=0}^{5} \Sigma_s \cup \{F\}$ where F is a new symbol, and let

$\Theta(2) = \{[A,i,-,2] : A \in \Sigma \text{ and } 1 \leq i \leq 3\}$,

$\Theta(3) = \{[A,i,-,3] : A \in \Sigma \text{ and } 1 \leq i \leq 3\}$,

for $1 \leq i \leq 3$

$\Gamma_i = \{[A,i] : \text{for some } \pi \in P_1,\ \underline{\ell hs}\pi = A\}$ and

$\Sigma_{b,i} = \{[A,i] : A \in \Sigma\}$.

Let h be the finite substitution on Θ^* defined by[2]:

for $[A,i] \in \Sigma_b$,

$h([A,i]) = \{A\} \cup \{[A,i,j,0] : \pi_j \in P_1 \text{ and } A = \underline{\ell hs}\pi_j\} \cup$
$\cup \{[A,i,j,1] : \pi_j \in P_2 \text{ and } A = \underline{\ell hs}\pi_j\} \cup \{[A,i,-,2]\} \cup$
$\cup \{[A,i,j,4] : \pi_j \in P_3,\ AB = \underline{\ell hs}\pi_j \text{ for some } B \in \Sigma\backslash\Delta\} \cup$
$\cup \{[A,i,j,5] : \pi_j \in P_3,\ BA = \underline{\ell hs}\pi_j \text{ for some } B \in \Sigma\backslash\Delta\},$

for $A \in \Sigma\backslash\Delta$, $1 \le i \le 3$, $1 \le j \le r$,

$$h([A,i,j,0]) = \begin{cases} \{[B,i]\} & \text{if } \pi_j = A \to B, \\ \{F\} & \text{otherwise,} \end{cases}$$

$$h([A,i,j,1]) = \begin{cases} \{[B,i][C,i+1(\text{mod } 3)]\} & \text{if } \pi_j = A \to BC, \\ \{F\} & \text{otherwise,} \end{cases}$$

$h([A,i,-,2]) = \{[A,i],\ [A,i+1(\text{mod } 3),-,3]\},$

$h([A,i,-,3]) = \{[A,i,-,2]\},$

$$h([A,i,j,4]) = \begin{cases} \{[A,i]\} & \text{if } \underline{\ell hs}\pi_j = AB \text{ for some } B \in \Sigma\backslash\Delta, \\ \{F\} & \text{otherwise,} \end{cases}$$

$$h([A,i,j,5]) = \begin{cases} \{[C,i]\} & \text{if } \pi_j = BA \to BC \text{ for some } B \in \Sigma\backslash\Delta, \\ \{F\} & \text{otherwise,} \end{cases}$$

for all other $x \in \Theta$,

$h(x) = \{F\}.$

Let, for $1 \le i \le 3$, $1 \le j \le r$,

$$K_{cf,c,b,i} = \Sigma_b^* \overline{\{[A,i]\}}^+ \Sigma_b^* ,$$

$$K_{cf,c,i,j} = \Sigma_b^* \overline{\{[A,i,j,0]\}}^+ \Sigma_b^*,$$

$$K_{cf,t,b} = \overline{\Sigma}_b^+$$

$$K_{cf,t,i,j,1} = (\{[A,i,j,1]\} \cup \Theta(2))^* \overline{(\Theta(2))}^+ \text{ where } A = \underline{\ell hs}\pi_j,$$

$$K_{cf,t,i,j,2} = (\Theta(2))^* \overline{\{[A,i,j,1]\}}^+ (\Theta(3))^* \text{ where } A = \underline{\ell hs}\pi_j,$$

$K_{cf,t,3} = (\Theta(2))^*(\overline{\Theta(3)})^+$,

$K_{cf,t,e} = (\overline{\Theta(2)})^+$,

$K_{cs,b,i} = \Sigma_b^*\{\overline{\Sigma}_{b,i} \cup \overline{\Sigma}_{b,i+1(\mathrm{mod}\,3)}\}^+\Sigma_b^*$,

$K_{cs,i,j,1} = \Sigma_b^*\{\overline{[A,i,j,4]}\}^+(\Sigma_b \cup [B,i+1(\mathrm{mod}\,3),j,5])^*$

where $\pi_j = AB \to AC$ for some $C \in \Sigma\backslash\Delta$,

$K_{cs,i,j,e} = \Sigma_b^*\{\overline{[B,i,j,5]}\}^+\Sigma_b^*$ where $\pi_j = AB \to AC$ for

some $A,C \in \Sigma\backslash\Delta$.

Then let K be the union of all the above languages.

Let $H = (\Theta, h, [S,1], \Delta, K)$.

H simulates G as follows. Assume that we have a sentential form α in H such that $\alpha = [A_1,i_1][A_2,i_2] \ldots [A_n,i_n]$, where $i_{u+1} = i_u + 1(\mathrm{mod}\,3)$ for $1 \leq u \leq n-1$ (note that [S,1] is such a word).

To simulate an application of a production $\pi_j \in P_1$, $\pi_j = (A \to a)$ H applies selector $K_{cf,c,b,i}$ for some $i \in \{1,2,3\}$ yielding the replacement of one occurrence of the symbol [A,i] in α by [A,i,j,0]. Then H applies $K_{cf,c,i,j}$ to the resulting string replacing the unique occurrence of [A,i,j,0] by [B,i]. As a result, in two steps we have replaced an occurrence of [A,i] in α by [B,i] obtaining a new string over Σ_b satisfying the same condition on the second components of letters from Σ_b as α did.

The main problem in simulating production $\pi_j \in P_2$, $\pi_j = (A \to BC)$, is that "its straightforward application" to an occurrence [A,i] in α would yield a string that does not satisfy "the periodicity condition" that α satisfies ($i_{u+1} = i_u + 1(\mathrm{mod}\,3)$). This condition is absolutely necessary to maintain, so that the MC grammar H in simulating G can rewrite one occurrence of a symbol in a word-precisely as G does. To overcome this difficulty H proceeds in several steps as follows. First it rewrites α using selector $K_{cf,t,b}$ which results in a string α_1 having occurrences of symbols from $\Theta(2) \cup \{[A,i,j,1]\}$ only. The only selector that can be applied now is $K_{cf,t,i,j,1}$ (except for $K_{cf,t,e}$ the application of which would lead to an "idle rewriting"). As a result a suffix of α_1 consisting of letters from $\Theta(2)$ only got rewritten into letters from $\Theta(3)$ only yielding α_2. Now (providing that α_1 contained at least one occurrence of the symbol [A,i,j,1]) the only selector that can be applied is $K_{cf,t,i,j,2}$. However this implies that α_3 contains precisely one occurrence of [A,i,j,1] and so α was rewritten in α_1 in such a way that precisely one occurrence of [A,i,j,1] was introduced. Note that when $K_{cf,t,i,j,1}$ was applied to α_1 yielding α_2 the suffix of α_1 consisting of letters from $\Theta(2)$ was changed into a suffix of α_2 consisting of letters from $\Theta(3)$ in such a way that the second components of letters were increased by 1 (modulo 3). Since the unique occurrence of [A,i,j,1] in α_2 must be the occurrence neighboring this suffix (otherwise $K_{cf,t,i,j,2}$ cannot be applied) there is a "gap" between the second component of [A,i,j,1] and the leftmost element of the suffix considered (if it is not empty). So we have the situation

$\alpha_2 = \ldots [A,i,j,1][D,i+2(\mathrm{mod}\,3),-,3] \ldots$

(or $\alpha_2 = \ldots[A,i,j,1]$).

Hence the application of $K_{cf,t,i,j,2}$ which uses production $[A,i,j,1] \to [B,i][C,i+1(\mathrm{mod}\,3)]$ yields the string α_3 satisfying the periodicity condition. Then the application of $K_{cf,t,3}$ rewrites symbols from $\Theta(3)$ into the corresponding symbols in $\Theta(2)$ and the application of selector $K_{cf,t,e}$ (using the substitution $h([A,i,-,2]) = [A,i]!!$) yields then the string β, where the transition from α to β simulates the application of production $A \to BC$ in G.

To simulate an application of a production $\pi_j \in P_3$, $\pi_j = AB \to AC$ for $A,B,C \in \Sigma \backslash \Delta$, H applies selector $K_{cs,b,i}$ first (note that it does not depend on j). As a result of this we get from α a string α_1 such that it contains exactly one occurrence of a letter from Σ_4, say x, and exactly one occurrence of a letter from Σ_5, say y; moreover those two occurrences are next to each other with x to the left of y. Now to simulate the specific context-sensitive production π_j, selector $K_{cs,i,j,1}$ must be applied. However $K_{cs,i,j,1}$ can be applied only if α_1 is of the form $\alpha_1 = \ldots xy \ldots$, where $x = [A,i,j,4]$ and $y = [B,i,j,5]$. Then the application of $K_{cs,i,j,1}$ yields the word α_2 differing from α_1 only by the replacement of the unique occurrence of $[A,i,j,4]$ by $[A,i]$. Consequently α_2 contains the unique occurrence of $[B,i,j,5]$ with all other letters in α_2 being from Σ_b. Now $K_{4,i,j}$ can (and must) be applied replacing the unique occurrence of $[B,i,j,5]$ by $[C,i]$ yielding β. Thus what happened on the transition from α to β is that a subword of the form $[A,i][B,i+1]$ in α was replaced by $[A,i][C,i+1]$. Hence the simulation of π_j is successfully completed. Note that obviously β again satisfies the periodicity condition and so the simulation of an application of an arbitrary production from P can be started again.

The rewriting process in H ends by applying selector $K_{cf,t,b}$ substituting A for $[A,i]$, $1 \le i \le 3$, $A \in \Sigma$. If any occurrence in α is replaced in this way then all occurrences in α must be replaced in this way, because no selector in H can be applied to a string containing an element of Δ.

Based on the above intuitive comments one can construct a formal proof of the fact that indeed $L(H) = L(G)$.

Hence the theorem holds. $\square$

VI. DISCUSSION

There are at least two methodological advantages to the study of continuous grammars.

(1). They form a special case of selective substitution grammars. They demonstrate that selective substitution grammars not only generalize a multitude of rewriting systems discussed in the literature, they also provide interesting new classes of rewriting systems.

(2). Since continuous grammars constitute a natural case of a rewriting system lying midway between two extreme cases of rewriting - totally sequential (as in context-free grammars) and totally parallel (as in EOL systems) - their study contributes to our understanding of the difference between sequential and parallel rewriting.

In this paper we have concentrated on the study of the language generating power of continuous grammars. In order to fully understand this topic one should attempt now to answer the following questions (which we are not yet able to answer).

(i). What are the precise relationships among the seven classes of languages emerging from our study: $\mathcal{L}(1C)$, $\mathcal{L}(1LC)$, $\mathcal{L}(1RC)$, $\mathcal{L}(2C)$, $\mathcal{L}(1MC)$, $\mathcal{L}(1LMC)$, $\mathcal{L}(1RMC)$?

(ii). What is the relationship between $\mathcal{L}(2C)$ and $\mathcal{L}(2MC)$? We know that $\mathcal{L}(2C) \subseteq \mathcal{L}(2MC)$. Is it the case that $\mathcal{L}(2C) \subsetneq \mathcal{L}(2MC)$?

(iii). Does there exist a language in $\mathcal{L}(ETOL) \setminus \mathcal{L}(1C)$?

In addition to answering those questions one should also attempt to study the different classes of continuous grammars introduced here with respect to the usual language - theoretic properties. For example, results on the combinatorial structure of languages in these language classes and results on the closure properties of these language classes would be natural next topics to consider.

ACKNOWLEDGMENT

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

[ER1] Ehrenfeucht, A. and Rozenberg, G., The number of occurrences of letters versus their distribution in EOL languages, Information and Control 26, 256-271, 1975.

[ER2] Ehrenfeucht, A. and Rozenberg, G., On proving that certain languages are not ETOL, Acta Informatica, 6, 407-415, 1976.

[ER3] Ehrenfeucht, A. and Rozenberg, G., Nonterminals versus homomorphisms in defining languages for some classes of rewriting systems, Acta Informatica, 4, 87-106, 1974.

[P] Penttonen, M., One-sided and two-sided context in formal grammars, Information and Control, 25, 371-392, 1974.

[R] Rozenberg, G., Selective substitution grammars (towards a framework for rewriting systems). Part I: definitions and examples. EIK, 1977.

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

[RW] Rozenberg, G. and Wood, D., Selective substitution grammars. Part II: the role of selection, McMaster University, Dept. of Computer Science, Tech. Rept. No. 77-CS-10.

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

FOOTNOTES

(1). We choose here an EOL system rather than a context-free grammar because the latter has a limitation of not being able to rewrite terminal symbols.

(2). In what follows, when we count modulo 3 then it is counted on positive integers, hence, e.g., $4 = 1 (\mathrm{mod}\ 3)$.