RAIRO. INFORMATIQUE THÉORIQUE

A. EHRENFEUCHT
G. ROZENBERG

On some context free languages that are not deterministic ETOL languages

RAIRO. Informatique théorique, tome 11, nº 4 (1977), p. 273-291
<http://www.numdam.org/item?id=ITA_1977__11_4_273_0>

R.A.I.R.O. Informatique théorique/Theorical Computer Science
(vol. 11, n° 4, 1977, p. 273 à 291)

ON SOME CONTEXT FREE LANGUAGES THAT ARE NOT DETERMINISTIC ETOL LANGUAGES (*) (1)

by A. EHRENFEUCHT (2) and G. ROZENBERG (3)

Communicated by W. Brauer

Abstract. — *It is shown that there exist context free languages which are not deterministic ETOL languages. The proof is based on an analysis of the structure of derivations in deterministic ETOL systems.*

I. INTRODUCTION

The theory of L systems (*see*, e. g., [8, 11 and 10]) is now a fashionable area of formal language theory. It brought along a lot of new problems (and techniques for solving them) and at the same time it put a lot of classical problems and notions in a new perspective. In particular through the theory of L systems we have gained a lot of insight into essential differences between sequential and parallel rewriting systems (*see*, e. g. [13]).

Among various families of L languages, ETOL languages occupy a central place (*see*, e. g., [2, 3, 9 and 10]). Its subfamily, the class of EDTOL languages, has a quite nice mathematical structure and at the same time plays an important role in investigating ETOL languages (*see*, e. g., [4, 5, 6 and 10]). Thus it is quite natural to compare the language generating power of context free grammars (which occupy a special place in the Chomsky hierarchy) with this of EDTOL systems.

Although one trivially establishes the existence of EDTOL languages which are not context free (the language $\{ a^{2^n} : n \geqq 0 \}$ is one of them) it was an open problem for quite a time, whether or not there exist context free languages that are not EDTOL.

In this paper we prove the existence of context free languages that are not EDTOL. Except for (as we have it already indicated) shedding some light on

(*) Received March 8, 1977.
(1) This work supported by NSF Grant # GJ-660,
(2) Department of Computer Science, University of Colorado, Boulder, U.S.A.
(3) Department of Mathematics, University of Antwerp, U.I.A., Wilrijk, Belgium and nstitute of Mathematics Utrecht University, Utrecht, De Uithof Holland.

the difference between sequential and parellel rewriting, this result seems to be also of technical importance. Thus, e. g. :

1) in [6] it is used to show that there exist indexed languages that are not ETOL;

2) in [7] it is used to show that there exist top-down deterministic tree transformation languages that are not indexed;

3) it can be used (J. Engelfriet, private communication) to show that there are context free languages that are not checking automata languages.

Troughout the paper we shall use the standard formal language theoretic terminology and notation. Also we use:

$\mu(x)$ to denote the smallest positive integer n such that any two disjoint subwords of x of length n are different;

$\#_a x$ to denote the number of occurrences of the letter a in the word x, and $\| m \|$ to denote the absolute value of an integer m.

II. EDTOL SYSTEMS AND LANGUAGES

In this section we recall (*see*, e. g., [8]) the definition of an EDTOL system (and language). We also recall from [4] some basic notions pertinent to the analysis of derivations in EDTOL systems.

DEFINITION 1 : An *extended deterministic table L system without interactions*, abbreviated as an EDTOL *system*, is defined as a construct $G = \langle V, \mathscr{P}, \omega, \Sigma \rangle$ such that.

1) V is a finite set (called the *alphabet of* G).

2) $\mathscr{P}$ is a finite set (called the *set of tables of* G), each element of which is a finite subset of $V \times V^*$. Eeach P in $\mathscr{P}$ satisfies the following conditions: for each a in V there exists *exactly one* α in V^* such that $\langle a, \alpha \rangle$ is in P.

3) $\omega \in V^+$ (called the *axiom of* G).

(We assume that V, Σ and each P in $\mathscr{P}$ are nonempty sets.)

We call G *propagating*, abbreviated as an EPDTOL *system* if each P in $\mathscr{P}$ is a subset of $V \times V^+$.

DEFINITION 2: Let $G = \langle V, \mathscr{P}, \omega, \Sigma \rangle$ be an EDTOL system. Let $x \in V^+$ $x = a_1, \ldots, a_k$, where each a_j, $1 \leqq j \leqq k$, is an element of V, and let $y \in V^*$. We say that x *directly derives* y *in* G (denoted as $x \underset{G}{\Rightarrow} y$) if and only if there exist P in $\mathscr{P}$ and $p_1, \ldots, p_k$ in P such that $p_1 = \langle a_1, \alpha_1 \rangle, \ldots, p_k = \langle a_k, \alpha_k \rangle$ and $y = \alpha_1 \ldots \alpha_k$. We say that x derives y in G (denoted as $x \overset{*}{\underset{G}{\Rightarrow}} y$) if and only if either (i) there exists a sequence of words $x_0, x_1, \ldots, x_n$ in V^* $(n \geqq 1)$ such that $x_0 = x$, $x_n = y$ and $x_0 \underset{G}{\Rightarrow} x_1 \underset{G}{\Rightarrow} \ldots \underset{G}{\Rightarrow} x_n$, or (ii) $x = y$.

DEFINITION 3: Let $G = \langle V, \mathscr{P}, \omega, \Sigma \rangle$ be an EDTOL system. The *language of* G, denoted as $L(G)$, is defined as $L(G) = \{ x \in \Sigma^* : \omega \overset{*}{\underset{G}{\Rightarrow}} x \}$.

NOTATION: Let $G = \langle V, \mathscr{P}, \omega, \Sigma \rangle$ be an EDTOL system.

1) If $\langle a, \alpha \rangle$ is an element of some P in $\mathscr{P}$ then we call it a *production* and write $a \rightarrow \alpha$ is in P or $a \underset{P}{\rightarrow} \alpha$.

2) If $x \underset{G}{\Rightarrow} y$ using table P from $\mathscr{P}$, then we also write $x \underset{P}{\Rightarrow} y$.

3) In fact each table P from $\mathscr{P}$ is a finite substitution. Hence we can use a "functional" notation and write P^m for an m-folded composition of P, $P_1 P_{2-1} \ldots P_m$ for a composition of tables $P_1, \ldots, P_m$ (first P_1, then $P_2, \ldots, P_m$), etc. In this sense $P_1 \ldots P_m(x)$ denotes the (unique) word y which is obtained by rewriting x by the sequence of tables $P_1, P_2, \ldots, P_m$.

Here are two examples of EDTOL systems and languages.

Example 1: Let $G_1 = \langle V, \mathscr{P}, \omega, \Sigma \rangle$ where

$$V = \{ A, B, a \}, \Sigma = \{ a \}, \omega = AB \text{ and } \mathscr{P} = \{ P_1, P_2 \},$$

where:

$$P_1 = \{ A \rightarrow A^2, B \rightarrow B^3, a \rightarrow a \}, \qquad P_2 = \{ A \rightarrow a, B \rightarrow a, a \rightarrow a \}.$$

G_1 is an EPDTOL system where $L(G_1) = \{ a^{2^n + 3^n}; n > 0 \}$.

Example 2: Let $G_2 = \langle \{ a, b, A, B, C, D, F \}, \mathscr{P}, CD, \{ a, b \} \rangle$, where $\mathscr{P} = \{ P_1, P_2, P_3 \}$ and

$$P_1 = \{ a \rightarrow F, b \rightarrow F, A \rightarrow A, B \rightarrow B, C \rightarrow ACB, D \rightarrow DA \},$$

$$P_2 = \{ a \rightarrow F, b \rightarrow F, A \rightarrow A, B \rightarrow B, C \rightarrow CB, D \rightarrow D \},$$

$$P_3 = \{ a \rightarrow F, b \rightarrow F, A \rightarrow a, B \rightarrow b, C \rightarrow \Lambda, D \rightarrow \Lambda \},$$

G_2 is an EDTOL system which is not propagating, and

$$L(G_2) = \{ a^n b^m a^n : n \geqq 0, m \geqq n \}.$$

Now we will recall from [4] various notions and theorems concerning derivations in EDTOL systems. They will be very essentially used in the sequel of this paper.

DEFINITION 4: Let $G = \langle V, \mathscr{P}, \omega, \Sigma \rangle$ be an EDTOL system. A *derivation* (*of* y *from* x) *in* G is a construct $D = ((x_0, \ldots, x_k), (T_0, \ldots, T_{k-1}))$ where $k \geqq 2$ and

1) $x_0, \ldots, x_k$ are in V^*;

2) $T_0, \ldots, T_{k-1}$ are in $\mathscr{P}$;

3) $x_0 = x$ and $x_k = y$ and $x_i \underset{T_i}{\Rightarrow} x_{i+1}$ for $0 \leqq i < k$.

If $x = \omega$ then we simply say that D is a *derivation* (*of* y) *in* G.

DEFINITION 5: Let $G = \langle V, \mathscr{P}, \omega, \Sigma \rangle$ be an EDTOL system and let $D = ((x_0, \ldots, x_k), (T_0, \ldots, T_{k-1}))$ be a derivation in G. For each occurrence a in x_j, $1 \leqq j \leqq k$, by a *contribution of a in D*, denoted as $\mathrm{Contr}_D(a)$, we mean the whole subword of x_k which is derived from a. (Then if x is an occurrence of a word in x_j, $\mathrm{Contr}_D(x)$ has the obvious meaning.) Also, for each T_j, $1 \leqq j \leqq k-1$, $T_j(\alpha)$ denotes both the word β such that $\alpha \underset{T_j}{\Rightarrow} \beta$ and the contribution to x_{j+1} by an occurrence (of a word) α in x_j, but this should not lead to confusion.

DEFINITION 6: Let $G = \langle V, \mathscr{P}, \omega, \Sigma \rangle$ be an EDTOL system and let $D = ((x_0, \ldots, x_k), (T_0, \ldots, T_{k-1}))$ be a derivation in G. A *subderivation* of D is a construct $\bar{D} = ((x_{i_0}, \ldots, x_{i_q}), (P_{i_0}, \ldots, P_{i_{q-1}}))$ where:

1) $0 \leqq i_0 \leqq i_1 < \ldots < i_q \leqq k-1$, and
2) for each j in $\{0, \ldots, q-1\}$, $P_{i_j} = T_{i_j} T_{i_j+1} \ldots T_{i_{j+1}-1}$.

Remark: Although a subderivation of a derivation in G does not have to be a derivation in G we shall use for subderivations the same terminology as for derivations and this should not lead to confusion. (For example we talk about tables used in a subderivation.) It is clear that to determine a subderivation $\bar{D}$ of a given derivation D it suffices to indicate which words of D form the sequence of words of $\bar{D}$. We will also talk about a subderivation $\bar{\bar{D}}$ of a subderivation $\bar{D}$ of D meaning a subderivation of D the words of which are chosen from the words of $\bar{D}$. (In this sense we have that a subderivation of a subderivation of a derivation D is a subderivation of the derivation D.) Given a subderivation $\bar{D}$ of D and an occurrence a in a word of $\bar{D}$ we talk about $\mathrm{Contr}_{\bar{D}}(a)$ in an obvious sense.

DEFINITION 7: Let $G = \langle V, \mathscr{P}, \omega, \Sigma \rangle$ be an EDTOL system and let f be a function from $\mathscr{R}_{\text{pos}}$ into $\mathscr{R}_{\text{pos}}$. Let D be a derivation in G and let $\bar{D} = ((x_0, \ldots, x_k), (T_0, \ldots, T_{k-1}))$ be a subderivation of D. Let a be an occurrence (of A from V) in x_t for some t in $\{0, \ldots, k\}$, where $|x_k| = n$.

1) a is called *(f, D)-big (in x_t)*, if $|\mathrm{Contr}_D(a)| > f(n)$;
2) a is called *(f, D)-small (in x_t)*, if $|\mathrm{Contr}_D(a)| \leqq f(n)$;
3) a is called *unique (in x_t)* if a is the only occurrence of A in x_t;
4) a is called *multiple (in x_t)* if a is not unique (in x_t);
5) a is called *$\bar{D}$-recursive (in x_t)* if $T_t(a)$ contains an occurrence of A;
6) a is called *$\bar{D}$-nonrecursive (in x_t)* if a is not $\bar{D}$-recursive (in x_t).

Remark: 1) Note that in an EDTOL system each occurrence of the same letter in a word is rewritten in the same way during a derivation process.

Hence we can talk about (f, D)-big (in x_t), (f, D)-small (in x_t), unique (in x_t), multiple (in x_t), $\bar{D}$-recursive (in x_t) and $\bar{D}$-nonrecursive (in x_t) *letters*.

2) Whenever f or D or $\bar{D}$ is fixed in considerations we will simplify the terminology in the obvious way (for example we can talk about big letters (in x_t) or about recursive letters (in x_t)).

Using standard methods (*see*, e. g., [9 or 10]) one can easily prove that for each EDTOL system G there exists an EPDTOL system H such that $L(G) \cup \{ \Lambda \} = L(H) \cup \{ \Lambda \}$. Hence, for the purpose of this paper, it suffices to analyze derivations in EPDTOL systems only.

Given a derivation in an EPDTOL system, all occurrences of the same letter on a given level are rewritten in the same way. However the behaviour of the same letter on *different* levels can be "drastically" different, which is due to the use of (possibly) different tables on different levels of the derivation. For example, the same letter can be big on some levels and small on others.

For this reason it is difficult to analyze an abritrary derivation, and so we try to find out a subderivation such that the "behaviour" of a letter does not depend on the level on which it occurs. We call such subderivations "neat". What is precisely meant by saying that a letter behaves in the same way on all levels of a subderivation is stated by conditions (2) through (7) of the following definition. It is also required, see condition (1), that in a neat subderivation the sets of letters occuring on each level are the same.

DEFINITION 8: Let $G = \langle V, \mathscr{P}, \omega, \Sigma \rangle$ be EPDTOL system and let f be a function from $\mathscr{R}_{\text{pos}}$ into $\mathscr{R}_{\text{pos}}$. Let D a be derivation in G and let $\bar{D} = ((x_0, \ldots, x_k), (T_0, \ldots, T_{k-1}))$ be a subderivation of D. We say that $\bar{D}$ is *neat* (*which respect to D and f*) if the following holds:

1) $\text{Min}(x_0) = \text{Min}(x_1) = \ldots = \text{Min}(x_k)$. (For a word x, $\text{Min}(x)$ denotes the set of all letters that occur in x).

2) If j is in $\{ 0, \ldots, k \}$ and A is a letter from $\text{Min}(x_j)$, then A is big (small, unique, multiple, recursive, nonrecursive) in x_j if and only if A is big (small, unique, multiple, recursive or nonrecursive respectively) in x_t for every t in $\{ 0, \ldots, k \}$.

3) For every j in $\{ 0, \ldots, k \}$, $\text{Min}(x_j)$ contains a big recursive letter.

4) For every j in $\{ 0, \ldots, k \}$ and every A in $\text{Min}(x_j)$, if A is big then A is unique.

5) For every j in $\{ 0, \ldots, k-1 \}$.

5.1. T_j contains a production of the form $A \to \alpha$ where A is a big letter and α contains small letters, and

5.2. If $B \to \beta$ is in T_j, then:
if B is small recursive, then $\beta = B$ and
if B is nonrecursive then β consists of small recursive letters only.

6) For every i, j in $\{0, \ldots, k\}$ and every A in V, if a is a small occurrence of A in x_i and b is a small occurrence of A in x_j then $|\mathrm{Contr}_D(a)| = |\mathrm{Contr}_D(b)|$.

7) For every big recursive letter Z and for every i, j in $\{0, \ldots, k-1\}$, if $Z \underset{T_i}{\rightarrow} \alpha$ and $Z \underset{T_j}{\rightarrow} \beta$ then α and β have the same set of big letters (and in fact none of them except for Z is recursive).

Throughout this paper we shall often use phrases like "(sufficiently) long word x with a property P" or a "(sufficiently) long (sub) derivation with a property P". Intuitively, this will have the following meaning (for a more formal definition, *see* [4]).

1) By a "(sufficiently) long word x with a property P" we mean a word x with property P which is longer than some constant C the computation of which does not depend on x itself.

2) By a "(sufficiently) long (sub)derivation with a property P" we mean a (sub) derivation D satisfying P of a word x which is longer than $|x|^C$ where C is a constant independant of either x or D.

The following result (proved in [4]) will be used to get long subderivations from other long subderivations. Before we formulate it we need another definition.

DEFINITION 9: Let f be a function from $\mathscr{R}_{\text{pos}}$ into $\mathscr{R}_{\text{pos}}$. We say that f is *slow* if

$$(\forall \alpha)_{\mathscr{R}\text{pos}} (\exists n_\alpha)_{\mathscr{R}\text{pos}} (\forall x)_{\mathscr{R}\text{pos}}, \quad [\text{if } x > n_\alpha \text{ then } f(x) < x^\alpha].$$

Thus a constant function, $(\log x)^k$ and $(\log x)^{\log \log x}$ are examples of slow functions, whereas $(\log x)^{\log x}$, x^2, $\sqrt{x}$ are examples of functions which are not slow.

Let G be an EPDTOL system and let g be a slow function. Let $\bar{D}$ be a long subderivation of a derivation D of x in G. Let us divide the words in $\bar{D}$ into classes in such a way that the number of classes is not larger than $g(|x|)$.

LEMMA 1: *There exists a long subderivation of D consisting of all the words which belong to one class of the above division into classes.*

The following notion appears to be very useful in dealing with the structure of derivations in EPDTOL systems.

DEFINITION 10: Let Σ be a finite alphabet and let f be a function from $\mathscr{R}_{\text{pos}}$ into $\mathscr{R}_{\text{pos}}$. Let w be in Σ^*. We say that w is an *f-random word* (*over* Σ) if

$$(\forall w_1, u_1, w_2, u_2, w_3)_{\Sigma^*}$$

$$[\text{if } w = w_1 u_1 w_2 u_2 w_3 \text{ and } |u_1| > f(|w|) \text{ then } u_1 \neq u_2].$$

Thus, informally speaking, we call a word w f-random if every two disjoint subwords of w which are longer than $f(|w|)$ are different.

The following result was proved in [4].

THEOREM 1: *For every EPDTOL system G and every slow function f there exist r in* $\mathscr{R}_{\text{pos}}$ *and s in* $\mathbf{N}$ *such that, for every w in L (G), if* $|w| > s$ *and w is f-random, then every derivation of w in G contains a neat subderivation longer than* $|w|^r$.

The number of f-random words for a function f which is not "too slow" over an alphabet consisting of at least two letters is "rather large" which is stated in the following theorem proved in [5].

THEORFM 2: *Let* Σ *be a finite alphabet such that* $\# \Sigma = m \geqq 2$. *Let f be a function from* $\mathscr{R}_{\text{pos}}$ *into* $\mathscr{R}_{\text{pos}}$ *such that, for every x in* $\mathscr{R}_{\text{pos}}$, $f(x) \geqq 4 \log_2 x$. *Then, for every positive integer n,*

$$\frac{\# \{ w \in \Sigma^* : |w| = n \text{ and } w \text{ is } f\text{-random} \}}{m^n} \geqq 1 - \frac{1}{n}.$$

BINARY BRACKETED LANGUAGES

In this section we introduce binary bracketed languages which are context free languages but which will be proved in the next section not to be EDTOL languages.

DEFINITION 11: Let i be a positive integer. A *binary i-bracketed language*, denoted as $\mathscr{B}_i$, is the language generated by the context free grammar $H(\mathscr{B}_i) = \langle \{ S \} \{ \underset{1}{[}, \underset{2}{[}, \ldots, \underset{i}{[}, \underset{i}{]}, \ldots, \underset{2}{]}, \underset{1}{]} \}, P, S \rangle$ where P consists of the following productions.

$$S \to \underset{i}{[} SS \underset{i}{]}, \ldots, S \to \underset{2}{[} SS \underset{2}{]}, S \to \underset{1}{[} SS \underset{1}{]},$$

$$S \to \underset{i}{[} S \underset{i}{]}, \ldots, S \to \underset{2}{[} S \underset{2}{]}, S \to \underset{1}{[} S \underset{1}{]},$$

$$S \to \Lambda.$$

In fact we will first prove that B_1 is not an EDTOL language and then we will cobclude that no $\mathscr{B}_i$, $i \geqq 1$, is an EDTOL language. Thus all our "technical" definitions concern $\mathscr{B}_1$. (To simplify notation we write "[" for "$\underset{1}{[}$" and "]" for "$\underset{1}{]}$".)

DEFINITION 12: Let $x \in \mathscr{B}_1$. The *depth of* x, denoted as Depth (x), is the depth of the longest nesting of brackets in x. More formally, Depth (x) is defined inductively as follows:

(i) Depth (Λ) $= 0$;

(ii) For $x \neq \Lambda$ let $\bar{x}$ denote the word obtained from x by erasing subwords [ ] in x. If Depth ($\bar{x}$) $= k$ then Depth (x) $= k+1$.

DEFINITION 13: Let $x \in \{ [\, , \,] \}^*$. The *score of* x, denoted as Score (x), is defined by Score $(x) = \#_{[}(x) - \#_{]}(x)$.

Now we shall prove two properties concerning scores of words in $\mathscr{B}_1$ and their depths. These properties will turn out to be very useful later on.

LEMMA 2: *Let w be in $\mathscr{B}_1$ where for som w_1, w_2, w_3 in $\{ [,] \}^*$, $w = w_1 \, w_2 \, w_3$. Then $\| \text{Score}(w_2) \| \leqq \text{Depth}(w)$.*

Proof: We prove this result by induction on the depth of w.

(i) If Depth $(w) = 0$ then $w = \Lambda$ and the lemma trivially holds.

(ii) Let us assume that the lemma holds for all w such that Depth $(w) \leqq k$.

(iii) Let Depth $(w) = k+1$ and let $w = w_1 \, w_2 \, w_3$.

Let $\bar{w}$ be the word obtained from w by erasing subwords of the form $[\;]$ from w, and let $\bar{w} = \bar{w}_1 \, \bar{w}_2 \, \bar{w}_3$ where $\bar{w}_1$, $\bar{w}_2$ and $\bar{w}_3$ correspond in this manner to w_1, w_2 and w_3 respectively. Thus Depth $(\bar{w}) = k$ and so by the inductive assumption $\| \text{Score}(\bar{w}_2) \| \leqq k$.

Let us observe that w_2 must have one of the following four forms:

1. it begins with] and ends with [;
2. it begins with [and ends with [;
3. it begins with] and ends with];
4. it begins with [and ends with].

It is easy to see that in all these cases $\| \text{Score}(\bar{w}_2) - \text{Score}(w_2) \| \leqq 1$ and so the lemma holds.

LEMMA 3: $(\forall \, n)_N \, (\exists \, m)_N \, (\forall \, w)_{\mathscr{B}_1}$

$$[\text{if } w = w_1 \, w_2 \, w_3 \text{ and } | w_2 | \geqq m \text{ then } w_2 = u_1 \, u_2 \, u_3 \text{ with } \| \text{Score}(u_2) \| \geqq n].$$

Proof: Let $n \in N$ and let $m = 2^{2n+2}$. Let w be a word in $\mathscr{B}_1$ such that $| w | \geqq m$ and let w_1, w_2, w_3 be such that $w = w_1 \, w_2 \, w_3$ and $| w_2 | \geqq m$.

Let us consider a derivation tree T for w in $H(\mathscr{B}_1)$. Let $\bar{T}$ be a subtree of T obtained by removing from T all the nodes (and edges leading to them) that do not "contribute" to w_2.

Now, if $\| \text{Score}(w_2) \| \geqq n$ then we set $u_2 = w_2$ and the lemma holds.

If not then we proceed as follows.

We divide nodes in $\bar{T}$ into three categories:

type 0, neither a node labeled with] nor a node labeled with [is a among direct descendants of such a node;

unary, among the direct descendants of such a node is either a node labeled with] or a node labeled with [, but not both;

binary, among the direct descendants of such a node are both, a node labeled with [and a node labeled with].

Please notice that unary nodes can occur only at the left or right edge of $\overline{T}$ and all the unary nodes occurring at the same edge form a path with all of them directly contributing *the same* terminal symbol (either] or [) to w_2.

Hence if $\overline{T}$ contains such a path of unary nodes not shorter than n, then it suffices to take as u_2 the subword of w_2 which is the contribution of this path to w_2.

If $\overline{T}$ does not contain such a "long enough" path, then it must contain at least one path with at least $(2n+2)$ nodes that are binary. Let p be such a path in $\overline{T}$. Thus there are at least $(2n+2)/2 = n+1$ branchings to the one side (say the left one) of p. Let X be the binary node on p closest to the root of $\overline{T}$ and let u_2 be the contribution to w_2 of all the nodes on p starting with X to the left of p.

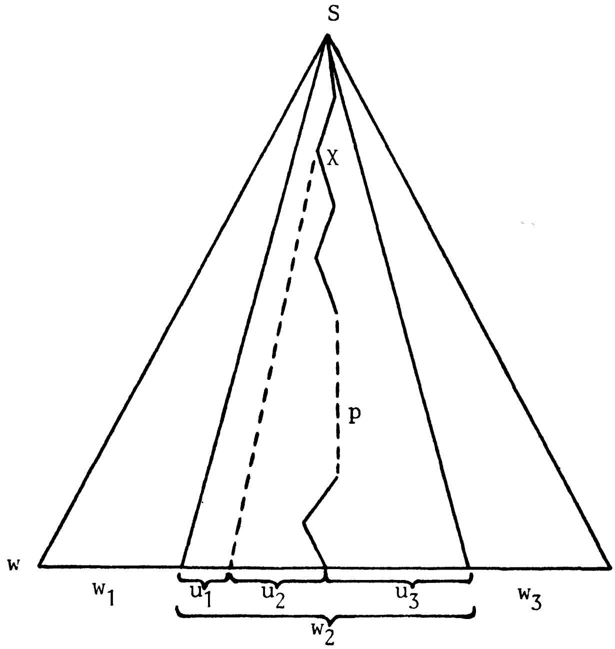

Figure 1.

Clearly Score $(u_2) \geqq n$, and so the lemma holds.

MAIN RESULTS

In this section we will prove that, for all $i \geqq 1$, i-bracketed languages are not EDTOL languages. Also as a corollary we obtain that Dyck languages are not EDTOL languages.

Firts we shall prove that for $f(g) = 32 \log_2^2 g$ we have arbitrarily long words in $\mathscr{B}_1$ which are f-random but of a "small" depth.

THEOREM 3:

$(\forall n)_N (\exists y)_{\mathscr{B}_1} [|y| > n \text{ and Depth } (y) < 2\log_2 |y| \text{ and } \mu(y) < 32\log_2^2 |y|]$.

Proof: Let x be a word in $\mathscr{B}_1$ such that its derivation tree in $H(\mathscr{B}_1)$ is of the form and is hat height n for some $n > 1$.

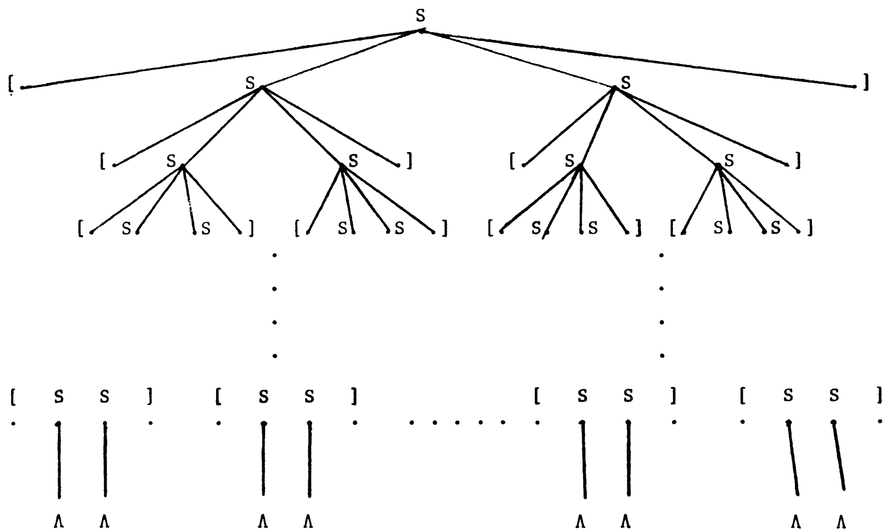

Figure 2.

(In other words after erasing in this tree all nodes not labeled by S and erasing all connections leading to them one gets a full binary tree.)

Let $\Sigma = \{B_1, B_2\}$. Let h be a homomorphism from $\{B_1, B_2\}^*$ into $\{[\,,\,]\}^*$ defined by $h(B_1) = [\ \ ]$ and $h(B_2) = [[\ \ ]]$. Let w be an arbitrary word over $\{B, B_2\}$ such that the length of w equals the number of occurrence of the word $[\ \ ]$ in x. Say $w = b_1 b_2 \ldots b_j$ with $b_1, \ldots, b_j$ in $\{B_1, B_2\}$. Let $\mu(w) \leqq k$ for some k in N.

Let $x(w)$ be the word (over $\{[\,,\,]\}$) which is obtained from x by replacing the i' th (from the left) occurrence of $[\ \ ]$ in x by $[h(b_i)]$. (For example if

$$x = [[[\ \ ][\ \ ]][[\ \ ][\ \ ]]]$$

and

$$w = B_2 B_1 B_1 B_2 \text{ then } x(w)$$

$$= [[[[[\ \ ]]][[\ \ ]]][[[\ \ ]][[[\ \ ]]]]]).$$

Let us assume that $n > 5$.

1) Note that $|x| = 2^{n+1}-2$ and $|x(w)| \geqq 2^{n+1}-2+2^n > 2^{n+1}$. Thus $n \leqq \log_2 |x(w)|$.

2) As Depth $(x) = n$ and, for i in $\{1, 2\}$,

$$\text{Depth}\,(h(B_i)) \leqq 2, \text{ Depth}\,(x(w)) \leqq n+2.$$

Thus Depth $(x(w)) \leqq n+n = 2\,n \leqq 2 \log_2 |x(w)|$.

3) Let us note that the longest subword of x which does not contain $[\ \ ]$ as its subword is no longer that $2\,n-2$. This implies that the longest subword of $x(w)$ which does not contain as a subword $[h(B_i)]\,Z$, where $i \in \{1, 2\}$ and Z does not contain $[\ \ ]$ as a subword, is no longer that $2\,n-2+5 = 2\,n+4$.

4) If $x(w)$ contains a subword α which contains as a subword

$$[h(b_{i_1})]\,Z_{i_1}\,[h(b_{i_2})]\,Z_{i_2} \ldots [h(b_{i_k})]\,Z_{i_k} \ldots \quad (\bigstar)$$

for some $i_1, \ldots, i_k$ in $\{1, \ldots, j\}$, where none of $Z_{i_1}, \ldots, Z_{i_k}$ contains $[\ \ ]$ as a subword, then no subword of $x(w)$ disjoint with α is identical to α.

This follows because if $x(w)$ would contain two disjoint occurrences of a word α of the form $(\bigstar)$ then w would contain two disjoint occurrences of an identical subword of length k. This however contradicts the assumption that $\mu(w) \leqq k$.

5) From 3 and 4 it follows that

$$\mu(x(w)) \leqq k.(2\,n+4) \leqq 2\,k.(n+2) \leqq 2\,k.4\,n \leqq 2\,k.4\log_2 |x(w)|.$$

From Theorem 2 we know that if f is a slow function such that $f(s) \geqq 4 \log_2 s$ then almost all long enough words over Σ are f-random. Hence choosing n large enough and choosing an f-random w we could assume that $k \leqq 4 \log_2 |w|$.

Thus

$$2\,k.4\log_2 |x(w)| \leqq 2.4\log_2 |w|.4\log_2 |x(w)|$$
$$\leqq 32.\log_2^2 |x(w)|. \text{So } \mu(x(w)) \leqq 32.\log_2^2 |x(w)|.$$

Consequently if we set $y = x(w)$, the theorem follows.

Next we prove that in an EDTOL language L which is a subset of $\mathscr{B}_1$ if w is a long enough f-random word in L, for every slow function f, then the depth of w is rather large.

THEOREM 4: *Let L be an EDTOL language such that $L \subseteq \mathscr{B}_1$. Then for every slow function f there exist a positive integer constant s and a positive real constant r such that if w is an f-random word from L longer than s then Depth $(w) > |w|^r$.*

Proof: Let L be an EDTOL language such that $L \subseteq \mathscr{B}_1$ and let f be a slow function. We can assume that $G = \langle V, \mathscr{P}, \omega, \Sigma \rangle$ is an EPDTOL system such that $L(G) = L$. (*See* Theorem 4 in [4].) Clearly we can also assume that $L(G)$ contains infinitely many f-random words, as otherwise the theorem is trivially true.

Let w be an f-random word long enough so that each derivation of w in G contains a long enough neat subderivation (*see* Theorem 1). Thus let

$$D = ((x_0, \ldots, x_k), \qquad (T_0, \ldots, T_{k-1}))$$

be a derivation of w in G and let

$$D_1 = ((x_{i_0}, \ldots, x_{i_q}), \qquad (\overline{T}_{i_0}, \ldots, \overline{T}_{i_{q-1}}))$$

be a sufficiently long neat subderivation of D.

In fact we assume that

1) If A is a small letter in D_1, then:

$$\text{Score}(\text{Contr}_D(\overline{T}_i(A))) = \text{Score}(\text{Contr}_D(\overline{T}_j(A))),$$

for every i, j in $\{ i_0, \ldots, i_{q-1} \}$, and

2) There exists a big recursive letter R in D_1, such that either, for every j in

$$\{ i_0, \ldots, i_{q-1} \}, \qquad \overline{T}_j(R) = \alpha_R^{(j)} R \beta_R^{(j)} \qquad \text{with} \quad \alpha_R^{(j)} \neq \Lambda,$$

or, for every j in

$$\{ i_0, \ldots, i_{q-1} \}, \qquad \overline{T}_j(R) = \alpha_R^{(j)} R \beta_R^{(j)} \qquad \text{with} \quad \beta_R^{(j)} \neq \Lambda.$$

(We will assume, without the loss of generality, that for every j in

$$\{ i_0, \ldots, i_{q-1} \}, \ \overline{T}_j(R) = \alpha_R^{(j)} R \beta_R^{(j)} \text{ with } \alpha_R^{(j)} \neq \Lambda.)$$

3) For every big recursive letter B in D_1, and for every i, j in

$$\{ i_0, \ldots, i_{q-1} \}, \qquad \text{if } B \underset{\overline{T}_i}{\rightarrow} u_1 B u_2 \qquad \text{and} \qquad B \underset{\overline{T}_j}{\rightarrow} v_1 B v_2,$$

then u_1 and v_1 contain the same set of big letters and u_2 and v_2 contain the same set of big letters.

We can assume the above conditions because if they would not hold in D_1, we could apply Lemma 1 and obtain from D_1 a sufficiently long subderivation of D satisfying these conditions. (Note that Score $(\text{Contr}_D (T_i (\Lambda))) \leq | \text{Contr}_D (\overline{T}_i (A)) | \leq f(| w |)$ if A is a small letter, and to have the conditions 2 and 3 satisfied one has to divide the words in D_1 into a constant, dependent on $\# V$ only, number of classes.)

LEMMA 4: *For every j* in $\{ i_0. \ldots, i_{q-1} \}$,

$$\| \operatorname{Score}(\operatorname{Contr}_D(\overline{T}_j(\alpha_R^{(j)}))) \| > 0.$$

Proof of Lemma 4: Let us assume, to the contrary, that

$$\operatorname{Score}(\operatorname{Contr}_D(\overline{T}_j(\alpha_R^{(j)}))) = 0.$$

Note that $\overline{T}_j(\alpha_R^{(j)})$ contanis small recursive letters only and so (by changing D in such a way that after applying $\overline{T}_j$ we iterate $\overline{T}_j$ an arbitrary number of times before applying the next table from D_1 and continuing in the manner tables were used in D) for every $n \geqq 0$ there is a word in $L(G)$ which contains $(\operatorname{Contr}_D(\overline{T}_j(\alpha_R^{(j)})))^n$ as a subword. But (with our assumption that $\operatorname{Score}(\operatorname{Contr}_D(\overline{T}_j(\alpha_R^{(j)}))) = 0$) if γ is a subword of $(\operatorname{Contr}_D(\overline{T}_j(\alpha_R^{(j)})))^n$ then $\operatorname{Score}(\gamma) \leqq 2 \mid \operatorname{Contr}_D(\overline{T}_j(\alpha_R^{(j)})) \mid$. This however implies that $L(G)$ would contain words with arbitrarily long subwords the score of which is bounded by $2 \mid \operatorname{Contr}_D(\overline{T}_j(\alpha_R^{(j)})) \mid$ which contradicts Lemma 3.
Thus Lemma 4 holds.

LEMMA 5: *For every i, j in* $\{ i_0, \ldots, i_{q-1} \}$,

$$\operatorname{sign}(\operatorname{Score}(\operatorname{Contr}_D(\overline{T}_i(\alpha_R^{(i)})))) = \operatorname{sign}(\operatorname{Score}(\operatorname{Contr}_D(\overline{T}_j(\alpha_R^{(j)})))).$$

Proof of Lemma 5: Let us assume, to the contrary, that

$$\operatorname{sign}(\operatorname{Score}(\operatorname{Contr}_D(\overline{T}_i(\alpha_R^{(i)})))) \neq \operatorname{sign}(\operatorname{Score}(\operatorname{Contr}_D(\overline{T}_j(\alpha_R^{(j)})))),$$

for example that

$$\operatorname{sign}(\operatorname{Score}(\operatorname{Contr}_D(\overline{T}_i(\alpha_R^{(i)})))) > 0$$

and

$$\operatorname{sign}(\operatorname{Score}(\operatorname{Contr}_D(\overline{T}_j(\alpha_R^{(j)})))) < 0.$$

We will describe now (an infinite) sequence $\tau_0, \tau_1, \ldots$ of compositions of tables. Each of these compositions τ_j may be used to change D into $D(j)$ in such a way that after applying $\overline{T}_i$ we apply τ before continuing applying tables in the manner they are used in D. (To better *see* what follows, recall that $\overline{T}(\alpha_R^{(i)})$, $\overline{T}_i(\alpha_R^{(j)})$, $\overline{T}_j(\alpha_R^{(i)})$ and $\overline{T}_j(\alpha_R^{(j)})$ consist of small recursive letters only).

0) $\tau_0 = \overline{T}_i$.

$$\tau_0(\alpha_R^{(i)} R) = \overline{T}_i(\alpha_R^{(i)}) \alpha_R^{(i)} R \delta_0, \quad \text{for some } \delta_0 \in V^*.$$

1) $\tau_1 = \overline{T}_j \overline{T}_i$.

$$\tau_1(\alpha_R^{(i)} R) = \overline{T}_i(\alpha_R^{(i)}) \overline{T}_j(\alpha_R^{(i)}) \alpha_R^{(j)} R \delta_1, \quad \text{for some } \delta_1 \text{ in } V^*.$$

2) $\tau_2 = \overline{T}_j \overline{T}_j \overline{T}_i$.

$$\tau_2(\alpha_R^{(i)} R) = \overline{T}_i(\alpha_R^{(i)}) \overline{T}_j(\alpha_R^{(i)}) \overline{T}_j(\alpha_R^{(j)}) \alpha_R^{(j)} R \delta_2,$$

for some δ_2 in V^*.

p_1) $\tau_{p_1} = (\overline{T}_j)^{p_1} \overline{T}_i$.

$$\tau_{p_1}(\alpha_R^{(i)} R) = \overline{T}_i(\alpha_R^{(i)}) \overline{T}_j(\alpha_R^{(i)}) \overline{T}_j(\alpha_R^{(j)}) \dots \overline{T}_j(\alpha_R^{(j)}) \alpha_R^{(j)} R \delta_{p_1},$$

for some δ_{p_1} in V^*, where p_1 is the smallest positive integer such that

$$\text{sign}(\text{Score}(\text{Contr}_{D(p_1)}(\overline{T}_i(\alpha_R^{(i)}) \overline{T}_j(\alpha_R^{(i)}) \dots \overline{T}_j(\alpha_R^{(j)})))) < 0.$$

p_1+1) $\tau_{p_1+1} = \overline{T}_i(\overline{T}_j)^{p_1} \overline{T}_i$.

$$\tau_{p_1+1}(\alpha_R^{(i)} R) = \overline{T}_i(\alpha_R^{(i)}) \overline{T}_j(\alpha_R^{(i)}) \dots \overline{T}_j(\alpha_R^{(j)}) \overline{T}_i(\alpha_R^{(j)}) \alpha_R^{(i)} R \delta_{p_1+1},$$

for some δ_{p_1+1} in V^*.

p_1+2) $\tau_{p_1+2} = \overline{T}_i \overline{T}_i(\overline{T}_j)^{p_1} \overline{T}_i$,

$$\tau_{p_1+2}(\alpha_R^{(i)} R) = \overline{T}_i(\alpha_R^{(i)}) \overline{T}_j(\alpha_R^{(i)}) \dots \overline{T}_j(\alpha_R^{(j)}) \overline{T}_i(\alpha_R^{(j)}) \overline{T}_i(\alpha_R^{(i)}) \alpha_R^{(i)} R \delta_{p_1+2},$$

for some δ_{p_1+2} in V^*.

⋮

p_1+p_2) $\tau_{p_1+p_2} = (\overline{T}_i)^{p_2}(\overline{T}_j)^{p_1} \overline{T}_i$.

$$\tau_{p_1+p_2}(\alpha_R^{(i)} R) = \overline{T}_i(\alpha_R^{(i)}) \overline{T}_j(\alpha_R^{(i)}) \dots \overline{T}_j(\alpha_R^{(j)}) \overline{T}_i(\alpha_R^{(j)}) \dots \overline{T}_i(\alpha_R^{(i)}) \alpha_R^{(i)} R \delta_{p_1+p_2},$$

for some $\delta_{p_1+p_2}$ in V^*, where p_2 is the smallest positive integer such that

$$\text{sign}(\text{Score}(\text{Contr}_{D(p_1+p_2)}(\overline{T}_i(\alpha_R^{(i)}) \dots \overline{T}_j(\alpha_R^{(j)}) \dots \overline{T}_i(\alpha_R^{(i)}))))) > 0$$

$p_1+p_2+p_3$) $\tau_{p_1+p_2+p_3} = (\overline{T}_j)^{p_3}(\overline{T}_i)^{p_2}(\overline{T}_j)^{p_1} \overline{T}_i$.

$$\tau_{p_1+p_2+p_3}(\alpha_R^{(i)} R) = \overline{T}_i(\alpha_R^{(i)}) \overline{T}_j(\alpha_R^{(i)}) \dots \overline{T}_j(\alpha_R^{(j)}) \overline{T}_i(\alpha_R^{(j)}) \dots \overline{T}_i(\alpha_R^{(i)}) \overline{T}_j(\alpha_R^{(i)}) \dots \alpha_R^{(j)} R \delta_{p_1+p_2+p_3},$$

for some $\delta_{p_1+p_2+p_3}$ in V^*, where p_3 is the smallest positive integer such that

$$\text{sign}(\text{Score}(\text{Contr}_{D(p_1+p_2+p_3)}(\overline{T}_i(\alpha_R^{(i)}) \overline{T}_j(\alpha_R^{(i)}) \dots \overline{T}_j(\alpha_R^{(j)}) \overline{T}_i(\alpha_R^{(j)}) \dots \overline{T}_i(\alpha_R^{(i)}) \overline{T}_j(\alpha_R^{(i)}) \dots \overline{T}_j(\alpha_R^{(j)})))) < 0,$$

and so on.

Thus what we are doing is alternating sequences of applications of $\bar{T}_i$ and $\bar{T}_j$ in such a way that the signs of scores of contributions of corresponding substrings (consisting of small recursive letters) of strings derived from $\alpha_R^{(i)} R$ alternate.

But in this way $L(G)$ contains strings with arbitrarily long substrings the scores of which are limited by

$$4.\max\{|\bar{T}_i(\alpha_R^{(i)})|, |\bar{T}_j(\alpha_R^{(i)})|, |\bar{T}_i(\alpha_R^{(j)})|, |\bar{T}_j(\alpha_R^{(j)})|\}.$$

This however contradicts Lemma 3.

Thus Lemma 5 holds.

To avoid notational troubles with double indices, for the rest of this proof we change a denotation for the subderivation D_1.

Thus:

$$D_1 = ((y_0, \ldots, y_q), (P_0, \ldots, P_{q-1}))$$

where in fact

$$y_0 = x_{i_0}, \ldots, \qquad y_q = x_{i_q}, \qquad P_0 = \bar{T}_{i_0}, \ldots, \qquad P_{q-1} = \bar{T}_{i_{q-1}}.$$

Thus we have now, for each i in

$$\{0, \ldots, q-1\}, \qquad P_i(R) = \alpha_R^{(i)} R \beta_R^{(i)} \qquad \text{with} \quad \alpha_R^{(i)} \neq \Lambda.$$

Note that the word x derived in the derivation D has the word:

$$P_1(\alpha_R^{(0)}) P_2(\alpha_R^{(2)}) \ldots P_{q-1}(\alpha_R^{(q-2)})$$

as a subword.

Let:

$$\theta_1 = \text{Score}(\text{Contr}_D(P_1(\alpha_R^{(0)}) P_2(\alpha_R^{(1)}) \ldots P_{q-1}(\alpha_R^{(q-2)})).$$

Let Δ be a sequence of tables which form the "tail" of D in the sense that $\Delta = T_{i_q} T_{i_{q-1}} \ldots T_{k-1}$.

Let:

$$\theta_2 = \sum_{j=1}^{q-2} \text{Score}(\Delta(P_j(\alpha_R^{(j)}))).$$

Let us estimate $\theta_1 - \theta_2$. (Note that θ_1 represents the score of a subword of a word in $L(G)$, whereas θ_2 was chosen just for "computational" reasons.)

Let for a word Z over the alphabet of letters which occur in words of D_1, Big (Z) denote the word obtained from Z by erasing all small letters from Z and Small (Z) denote the word obtained from Z by erasing all big letters from Z.

Thus

$$\theta_1 = \sum_{j=1}^{q-1} \text{Score}(\text{Contr}_D(P_j(\alpha_R^{(j-1)}))$$
$$= \sum_{j=1}^{q-1} \text{Score}(\text{Contr}_D(P_j(\text{Big}(\alpha_R^{(j-1)}))))$$
$$+ \sum_{j=1}^{q-1} \text{Score}(\text{Contr}_D(P_j(\text{Small}(\alpha_R^{(j-1)})))),$$

and

$$\theta_2 = \sum_{j=1}^{q-2} \text{Score}(\Delta(P_j(\text{Big}(\alpha_R^{(j)}))))$$
$$+ \sum_{j=1}^{q-2} \text{Score}(\Delta(P_j(\text{Small}(\alpha_R^{(j)}))))$$
$$= \sum_{j=1}^{q-2} \text{Score}(\Delta(P_j(\text{Big}(\alpha_R^{(j)}))))$$
$$+ \sum_{j=2}^{q-1} \text{Score}(\Delta(P_j(\text{Small}(\alpha_R^{(j-1)}))))$$

(because of the Condition 1 satisfied by D_1).

Thus:

$$\theta_1 - \theta_2 = \text{Score}(\text{Contr}_D(P_{q-1}(\text{Big}(\alpha_R^{(q-2)}))))$$
$$+ \text{Score}(\text{Contr}_D(P_1(\text{Small}(\alpha_R^{(0)})))).$$

Now let

$$\alpha_R^{(0)} = Z_1 B_1 Z_2 B_2 \ldots Z_l B_l Z_{l+1}, \qquad \text{where} \quad Z_1, \ldots, Z_{l+1}$$

do not contain big letters and $B_1, \ldots, B_l$ are big letters. (Note that $l < \# V$.)

Then:

$$\theta_1 - \theta_2 = \text{Score}(\text{Contr}_D(P_{q-1}(\text{Big}(\alpha_R^{(q-2)}))))$$
$$+ \sum_{i=1}^{l+1} \text{Score}(\text{Contr}_D(P_1(Z_i))).$$

Let

$$\alpha_R^{(q-2)} = u_1 C_1 u_2 C_2 \ldots u_t C_t u_{t+1}, \qquad \text{where} \quad u_1, \ldots, u_{t+1}$$

do not contain big letters and $C_1, \ldots, C_t$ are big letters. (Note that $t < \# V$.)

Then:

$$\theta_1 - \theta_2 = \sum_{i=1}^{t} \text{Score}(\text{Contr}_D(P_{q-1}(C_i)))$$
$$+ \sum_{i=1}^{l+1} \text{Score}(\text{Contr}_D(P_1(Z_i))).$$

Thus:

$$\theta_1 - \left(\sum_{i=1}^{t} \text{Score}(\text{Contr}_D(P_{q-1}(C_i))) + \sum_{i=1}^{l+1} \text{Score}(\text{Contr}_D(P_1(Z_i))) \right) = \theta_2 .$$

But, for some positive real constant $\bar{r}$, the length of D_1 is larger than $| w |^{\bar{r}}$ and each component in the formula:

$$\sum_{j=1}^{q-2} \text{Score}(\Delta(P_j(\alpha_R^{(j)})))$$

is different from 0 and is of the same sign (Lemmas 4 and 5). Thus:

$$\| \theta_2 \| > | w |^{\bar{r}-1}.$$

Consequently, the absolute value of one of the following: θ_1,

$$\text{Score}(\text{Contr}_D(P_{q-1}(C_i))), \quad \text{for} \quad 1 \leqq i \leqq t,$$

$$\text{Score}(\text{Contr}_D(P_1(Z_i))), \quad \text{for} \quad 1 \leqq i \leqq l+1$$

must be larger than $| w |^{\bar{r}-1}/2 (\# V)$.

This together with Lemma 2 yields us Theorem 4.

Now we can prove the following result.

THEOREM 5: *If L is an EDTOL language such that $L \subseteq \mathscr{B}_1$ then $L \neq \mathscr{B}_1$.*

Proof: Theorem 3 says that $\mathscr{B}_1$ contains arbitrarily long f-random words (for a slow $f(|y|) = 32 \log_2^2 |y|$) of a rather small depth (Depth $(y) < 2 \log |y|$). But Theorem 4 says that in every EDTOL language L which is included in $\mathscr{B}_1$ if an f-random word y (for every slow f) is long enough then Depth (y) is rather large (Depth $(y) > |y|^r$ for a positive real constant r). Thus L cannot contain all the words from $\mathscr{B}_1$ and Theorem 5 holds.

We leave to the reader the easy standard proofs of the following two results.

THEOREM 6: *If L is an EDTOL language and h is a homomorphism, then $h(L)$ is an EDTOL language.*

THEOREM 7: *Every regular language is an EDTOL language. If L is an EDTOL language and R is a regular language then $L \cap R$ is an EDTOL language.*

Now we can prove three main results of this paper.

THEOREM 8: *For every $i > 1$, $\mathscr{B}_i$ is not an EDTOL language.*

Proof: As a direct corollary from Theorem 5 we have that $\mathscr{B}_1$ is not an EDTOL language. But then from Theorem 6 it follows that, for every $i \geqq 0$, $\mathscr{B}_1$ is not an EDTOL language.

Let us now recall the notion of a Dyck language (*see*, e. g., Salomaa [12], p. 68). Let, for

$$i \geqq 1, \qquad V_i = \{a_1, a'_1, a_2, a'_2, \ldots, a_n, a'_n\}.$$

The context free language D_i generated by the context free grammar

$$\langle \{S\}, V_i, \{S \to \Lambda, S \to SS, S \to a_1 S a'_1, \ldots, S \to a_i S a'_i\}, S \rangle$$

is termed the *Dyck language* over the alpтabet V_i.

THEOREM 9: *For every $i \geqq 8$, D_i is not an EDTOL language.*

Proof: Let us first recall the following well-know result (*see*, e. g., Salomaa [12], Theorem 7.5): for an alphabet Σ of m letters there exists an alphabet V_i of $i = 2m+4$ letters and a homomorphism h from V_i^* onto Σ^* such that, for every context free language L over Σ, there is a regular language R over V_i with the property $L = h(D_i \cap R)$.

But $\mathscr{B}_1$ is a context free language over an alphabet Σ consisting of $m = 2$ letters and by Theorem 8, $\mathscr{B}_1$ is not an EDTOL language. Thus from the above and Theorem 7 it follows that D_8 is not an ETOL language. Hence by Theorem 6 it follows that, for no $i \geqq 8$, D_i is an EDTOL language which proves the theorem.

As a corollary from either Theorem 8 or Theorem 9 we have the following result.

THEOREM 10: *There exist context free languages that are not EDTOL languages.*

DISCUSSION

We have shown that there exist context free languages which are not EDTOL languages. This result is directly used in [6] to show the existence of indexed languages (*see* [1]) that are not ETOL languages and in [7] to show the existence of top-down deterministic tree transformation languages that are not indexed.

In fact our results have further implications.

1) They settle a controversy on the existence of context free languages that are not parellel context free languages (*see* [14] and [15]). Because the class of parellel context free languages is clearly contained in the class of EDTOL languages we have provided an alternative proof to this of [15] that, almost all, Dyck languages are not parallel context free languages.

2) Following Salomaa [13], our Theorem 10 implies that (we use here Salomaa's notation from [13]):

The pairs (CF, IP), (ED, PPDA), (ED, ETOL) are incomparable, IP is properly contained in RP, ER is not contained in ETOL and ED is not contained in RP.

As the most important open problem in connection with results presented in this paper we consider the problem of giving a characterization of context free languages which are not EDTOL languages.

REFERENCES

1. A. AHO, *Indexed Grammars, an Extension of Context Free Grammars*, Journal of the A.C.M., Vol. 15, 1968, pp. 647-671.
2. P. A. CHRISTENSEN, *Hyper AFL's and ETOL Systems*, in [11], 1974.
3. P. DOWNEY, *OL Systems, Developmental Systems and Recursion Schemes*, Proceedings of the I.E.E.E. Conference on Biologically Motivated Automata, Theory, McLean, Virginia, 1974, pp. 54-58.
4. A. EHRENFEUCHT and G. ROZENBERG, *On the Structure of Derivations in Deterministic ETOL Systems*, to appear in Journal of Computer and Systems Sciences.
5. A. EHRENFEUCHT and G. ROZENBERG, *A Pumping Theorem for Deterministic ETOL Languages*, Revue Française d'Automatique, Informatique et Recherche Opérationnelle, R-2,9, 1975, pp. 13-23.
6. A. EHRENFEUCHT, G. ROZENBERG and S. SKYUM, *A Relationship between ETOL and EDTOL Languages*, Theoretical Computer Science, Vol. 1, 1976, pp. 325-330.
7. J. ENGELFRIET and S. SKYUM, *Copying Theorems*, Information Processing Letters, Vol. 4, 1976, pp. 157-161.
8. G. T. HERMAN and G. ROZENBERG, *Developmental systems and languages*, North-Holland Publishing Company, Amsterdam, 1975.
9. G. ROZENBERG, *Extension of tabled OL systems and languages*, International Journal of Computer and Information Sciences, vol. 2, 1973, pp. 311-334.
10. G. ROZENBERG and A. SALOMAA, *The mathematical theory of L systems*, in J. T. TOU, Ed., Advances in Information Systems Science, Vol. 6, 1976, pp. 161-206.
11. G. ROZENBERG and A. SALOMAA, Eds., *L systems*, Lecture Notes in Computer Science, Springer Verlag, Heidelberg, Vol. 15, 1974.
12. A. SALOMAA, *Formal languages*, Academic Press, London, 1973.
13. A. SALOMAA, *Parallelism in rewriting systems*, Lecture Notes in Computer Science, Springer Verlag, Heidelberg, Vol. 14, 1974, pp. 523-533.
14. R. SIROMONEY and K. KRITHIVASAN, *Parallel context free languages*, Information and Control, Vol. 24, 1974, pp. 155-162.
15. S. SKYUM, *Parallel context free languages*, Information and Control, Vol. 26, 1974, pp. 280-285.