On Some Context Free Languages That are not Deterministic ETOL Languages*

by

A. Ehrenfeucht**

and

G. Rozenberg***

Report #CU-CS-048-74 July 1974

* This work supported by NSF Grant #GJ-660

** Department of Computer Science
University of Colorado
Boulder, Colorado 80302 U.S.A.

**** Institute of Mathematics
Utrecht University
Utrecht-De Uithof HOLLAND
and
Department of Mathematics
University of Antwerp, U.I.A.
Wilrijk BELGIUM

All correspondence to: G. Rozenberg
Institute of Mathematics
Utrecht-De Uithof
Holland

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

L systems and languages become now a fashionable area of formal language theory (the reader is referred to Herman and Rozenberg [7] and to Rozenberg and Salomaa [9] for a more tutorial and a more research oriented texts respectively).

Among various families of L languages one of the central families is this of ETOL languages (see, e.g., Christensen [2], Downey [3], Rozenberg [8] and Salomaa [11]). On the other hand the family of deterministic ETOL languages turned out to be the very central subfamily of the family of ETOL languages (see, e.g., Ehrenfeucht and Rozenberg [4] and Ehrenfeucht and Rozenberg [6]).

The question of existence of context free languages which are not deterministic ETOL languages became recently quite vigorously investigated (see, e.g., Salomaa [12], Skyum [14] and Siromoney and Krithivasan [13]). There are at least two reasons for this:

1) The answer to this question puts a difference between sequential grammars of Chomsky and very parallel in nature L systems in a better light, and

2) The existence of context free languages which are not deterministic ETOL languages would imply (see Ehrenfeucht and Rozenberg [4]) the existence of indexed languages (see Aho [1]) which are not ETOL languages. This in turn would solve a quite important open problem (see, e.g., Salomaa [11]).

In this paper we prove the existence of context free languages which are not deterministic ETOL languages. (Among these languages are, almost all, Dyck languages.)

Throughout 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 introduce the class of EDTOL systems (and languages) and provide some examples of them.

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

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

2) $\mathcal{P}$ is a finite set (called the set of tables of G), each element of which is a finite subset of $V \times V^*$. Each P in $\mathcal{P}$ 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) $\omega \varepsilon V^+$ (called the axiom of G).

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

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

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

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

Notation. Let $G = \langle V, \mathcal{P}, \omega, \Sigma \rangle$ be an EDTOL system.

1) If $\langle a,\alpha \rangle$ is an element of some P in $\mathcal{P}$ then we call it a production and write

$a \rightarrow \alpha$ is in P or $a \underset{P}{\rightarrow} \alpha$.

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

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

We end this section with some examples of ETOL systems and languages.

Example 1. Let $G_1 = \langle V, \mathcal{P}, \omega, \Sigma \rangle$ where $V = \{A, B, a\}$, $\Sigma = \{a\}$, $\omega = AB$ and $\mathcal{P} = \{P_1, P_2\}$, where

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

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

Example 2. Let $G_2 = \langle \{a, b, A, B, C, D, F\}, \mathcal{P}, CD, \{a, b\} \rangle$, where $\mathcal{P} = \{P_1, P_2, P_3\}$ 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 $L(G_2) = \{a^n b^m a^n : n \geq 0, m \geq n\}$.

DERIVATIONS IN EDTOL SYSTEMS

In this section various notions and theorems concerning derivations in EDTOL systems are introduced. They will be very essentially used in the sequel of this paper.

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

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

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

3) $\mathcal{O}$ is an unambiguous description which tells us, for each j in $\{0, \ldots, k-1\}$, how each occurrence in x_j is rewritten using T_j to obtain x_{j+1},

4) $x_0 = x$ and $x_k = y$.

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

Definition 5. Let $G = \langle V, P, \omega, \Sigma \rangle$ be an EDTOL system and let $D = ((x_0, \ldots, x_k), (T_0, \ldots, T_{k-1}), \mathcal{O})$ be a derivation in G. For each occurrence a in x_j, $1 \leq j \leq k$, by a contribution of a in D, denoted as $Contr_D(a)$, 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, $Contr_D(x)$ has the obvious meaning.) Also, for each T_j, $1 \leq j \leq k-1$, $T_j(\alpha)$ 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 a confusion.

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

1) $0 \leq i_0 < i_1 < \ldots < i_q \leq k-1$,

2) for each j in $\{0, \ldots, q-1\}$, $P_{i_j} = T_{i_j} T_{i_{j+1}} \ldots T_{i_{j+1}-1}$,

3) $\mathcal{D}$ is an unambiguous description which tells us, for each j in {0, . . ., q-1}, how each occurrence in x_{i_j} is rewritten by P_{i_j} to obtain $x_{i_{j+1}}$.

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

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 $|\mathrm{Contr}_D(a)| \leq f(n)$,

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)).

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

1) $Min(x_0) = Min(x_1) = \ldots = Min(x_k)$.

2) If j is in $\{0, \ldots, k\}$ and A is a letter from $Min(x_j)$, 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 $\{0, \ldots, k\}$, $Min(x_j)$ contains a big recursive letter.

4) For every j in $\{0, \ldots, k\}$ and every A in $Min(x_j)$, 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 $A \to \alpha$ is in T_j, then

if A is small recursive, then $\alpha = A$, and

if A in nonrecursive then α 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_2 and b is a small occurrence of A in x_j then $|Contr_D(a)| = |Contr_D(b)|$.

7) For every big recursive letter A and for every i, j in $\{0, \ldots, k-1\}$,

if $Z \xrightarrow[T_i]{} \alpha$ and $Z \xrightarrow[T_j]{} \beta$ 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". This will have the following meaning.

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 independent of either x or D.

The following result (proved in Ehrenfeucht and Rozenberg [5]) 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 $\mathcal{R}_{pos}$ into $\mathcal{R}_{pos}$. We say that f is slow if

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

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 EDTOL system and let g be a slow function. Let $\overline{D}$ be a long subderivation of a derivation D of x in G. Let us divide the words in $\overline{D}$ into classes in such a way that a 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 EDT0L systems.

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

$$(\forall w_1, u_1, w_2, u_2, w_3)_{\Sigma^*} \; [\text{if } w = w_1u_1w_2u_2w_3 \text{ and } |u_1| > f(|w|) \text{ and } |u_2| > f(|w|), \text{ then } u_1 \neq u_2]$$

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 Ehrenfeucht and Rozenberg [5].

Theorem 1. For every EPDT0L system G and every slow function f there exist r in $\mathbb{R}_{pos}$ and s in $\mathbb{N}$ 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 Ehrenfeucht and Rozenberg [5].

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

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

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 to be not EDTOL languages.

Definition 11. Let i be a positive integer. A binary i-bracketed language, $\mathcal{B}_i$, is the language generated by the context free grammar

$$H(\mathcal{B}_i) = \langle \{S\}, \{[_1, [_2, \ldots, [_i,]_i, \ldots,]_2,]_1\}, \{S \to [_1 SS]_1, \ldots, S \to [_i SS]_i, S \to [_1\,]_1, \ldots, S \to [_i\,]_i\}, S \rangle .$$

In fact we will prove that $\mathcal{B}_1$ is not an EDTOL language and then using a very simple fact we will conclude that no $\mathcal{B}_i$, $i \geq 1$, is an EDTOL language. Thus all our "technical" definitions concern $\mathcal{B}_1$. (To simplify notation we write "[" for "$[_1$" and "]" for "$]_1$".)

Definition 12. Let $x \varepsilon \mathcal{B}_1$. 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) $\mathrm{Depth}(\Lambda) = 0$

(ii) For $x \neq \Lambda$ let $\bar{x}$ denote the word obtained from x by erasing subwords ( ) in x.

If $\mathrm{Depth}(\bar{x}) = k$ then $\mathrm{Depth}(x) = k+1$.

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

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

Lemma 2. Let w be in $\mathcal{B}_1$ where for some w_1, w_2, w_3 in $\{[,]\}^*$, $w = w_1 w_2 w_3$. Then $||\mathrm{Score}(w_2)|| \leqslant \mathrm{Depth}(w)$.

Proof

1) Let us note that if u_1, u_2 $\varepsilon\{[,]\}^*$ with $u_1 \neq \Lambda$, $u_2 \neq \Lambda$ and u_1u_2 in $\mathcal{B}_1$, then Score(u_1) > 0 and Score(u_2) < 0.

This follows from the fact that Score(u_1) + Score(u_2) = 0 and that in every prefix v of a word in $\mathcal{B}_1$ it must be that $\#_[(v) \geqslant \#_](v)$ whereas in every suffix $\bar{v}$ of a word in $\mathcal{B}$, it must be that $\#_](\bar{v}) \geqslant \#_[(\bar{v})$.

2) Now let us prove the lemma by induction on Depth(w).

(i) For Depth(w) = 1 the lemma obviously holds

(ii) Let us assume that the lemma holds for all w in $\mathcal{B}_1$ such that Depth(w) $\leqslant$ k.

(iii) Let $w \varepsilon \mathcal{B}_1$ and let Depth(w) = k + 1, for some k $\geqslant$ 1. Hence one can derive w in H($\mathcal{B}_1$) in k + 1 steps. Consequently one can derive (in H($\mathcal{B}_1$)) w in k steps either from [SS] or from [S].

Let $\bar{w}$ be a subword of w such that $||$Score($\bar{w}$)$||$ is at least as big as $||$Score(α)$||$ for any subword α of w.

Thus we have three casses

(iii.1) $\bar{w}$ is a subword of a word derived in k steps from S. Hence by the inductive assumption $||$Score($\bar{w}$)$||$ < Depth(w) .

(iii.2) $\bar{w}$ is a prefix of a word derived in k steps from [S (or symmetrically, $\bar{w}$ is a suffix of a word derived in k steps from S]). Then by inductive assumption $||$Score($\bar{w}$)$||$ $\leqslant$ k + 1.

(iii.3) $\bar{w}$ is the catenation of a word derived in k steps from [S with the prefix of a word derived in k steps from S. But if x is a word derived from S then Score(x) = 0. Thus by inductive assumption $||$Score($\bar{w}$)$||$ $\leqslant$ k + 1.

(Note that $\bar{w}$ cannot be the catenation of a prefix of a word derived from S in k steps with a suffix of a word derived from S in k steps, because then it could not be that $||$Score($\bar{w}$)$||$ is not smaller that $||$Score(α)$||$ for any subword α of w.

Lemma 3.

$(\forall n)_N (\exists m)_N (\forall w)_{\mathcal{B}_1}$ [if $w = w_1 w_2 w_3$ and $|w_2| \geq m$ then $w_2 = u_1 u_2 u_3$ with $||\text{Score}(u_2)|| \geq n$].

Proof

Let $n \varepsilon N$. Let $m = 2^{2n+2}$.

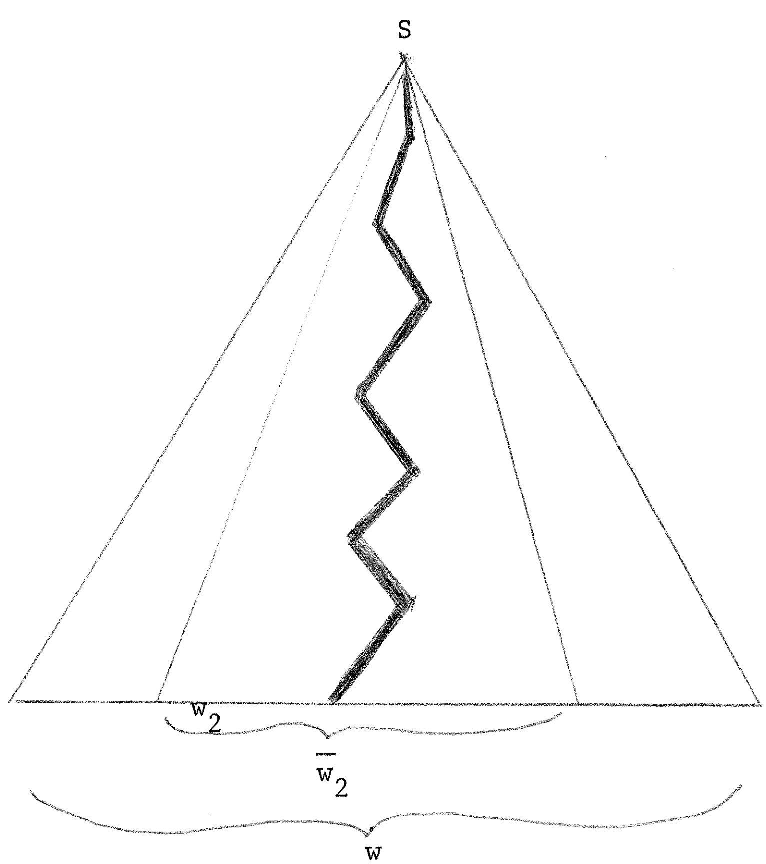
Let w be a word in $\mathcal{B}_1$ such that $|w| \geq m$.

Let $\overline{w}_1$, $\overline{w}_2$, $\overline{w}_3$ be such that $\overline{w}_1\overline{w}_2\overline{w}_3 = w$ and $|\overline{w}_2| \geq m$.

Let us consider a derivation tree T for w in $H(\mathcal{B}_1)$.

Let us then consider a subtree $\overline{T}$ of T obtained by removing from $\overline{T}$ all nodes (and edges leading to them) that do not "contribute" to $\overline{w}_2$.

Note that $\overline{T}$ is at most binary tree "producing" $\overline{w}_2$ (where $|\overline{w}_2| \geq 2^{2n+2}$) and so it contains at least one path with at least (2n + 2) nodes that are binary. Consequently from such a path, let us call one of them p, there is at least 2n+2/2 branchings to the one side (say the left one) of p. Let us denote the part of $\overline{w}_2$ contributed by these branchings by w_2.



Thus we have that $||Score(w_2)|| \geq (2n+2)/2 - 1 = n$, which proves the Lemma.

MAIN RESULTS

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

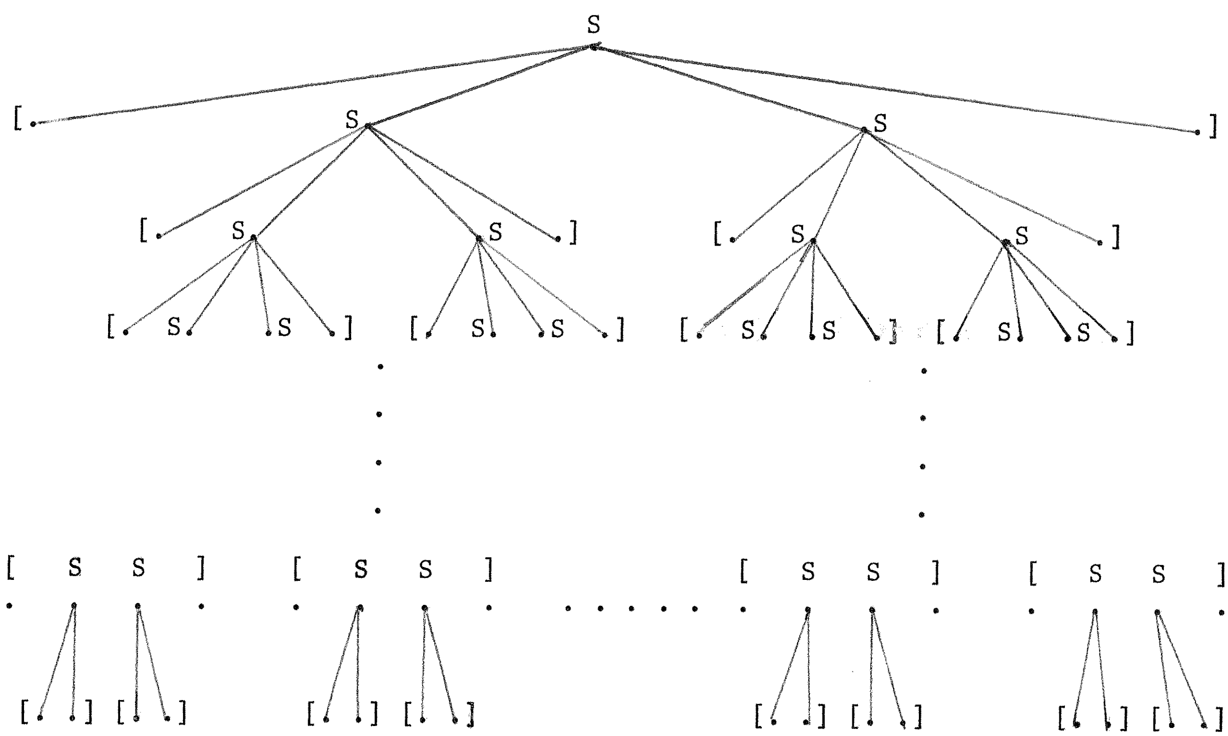
First we shall prove that for $f(g) = 32 \log_2^2 g$ we have arbitrarily long words in $\mathcal{B}_1$ which are f-random but of a "small" depth.

Theorem 3.

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

Proof

Let x be a word in $\mathcal{B}_1$ such that its derivation tree in $M(\mathcal{B}_1)$ is of the form



and it has height n for some $n > 1$.

(In other words after erasing in this tree all nodes not labeled by S and erasing all connections leading to them one gets 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 $(B_2) = [[\ ]]$. Let w be an arbitrary word over $\{B_1, B_2\}$ such that the length of w equals the number of occurrences of the word [ ] in x. Say $w = b_1 b_2 . . . b_j$ with $b_1, . . ., b_j$ in $\{B_1, B_2\}$. Let $\mu(w) \leqslant k$ 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 $w = B_2 B_1 B_1 B_2$ then x(w) = [[[[[ ]]] [[ ]]] [[[ ]] [[[ ]]]]]).

Let us assume that $n > 5$.

1) Note that $|x(w)| \geqslant |x| \cdot 2 \geqslant 2^n$, because $|x| = 2^{n-1}$. Thus $n \leqslant \log_2 |x(w)|$.

2) As $\text{Depth}(x) \leqslant n-1$ and, for i in {1,2}, $\text{Depth}(h(B_i)) \leqslant 2$, $\text{Depth}(x(w)) \leqslant n + 1$. Thus $\text{Depth}(x(w)) \leqslant n + n = 2n = 2 \log_2 |x(w)|$.

3) Let us note that the longest subword of x which does not contain [ ] as its subword is shorter than $2n + 1$. This implies that the longest subword of x(w) which does not contain as a subword $[h(B_i)]Z$, where $i \varepsilon \{1,2\}$ and Z does not contain [ ] as a subword, is shorter than $2 \cdot (2n + 1 + 8)$.

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

$$[h(b_{i_1})]Z_{i_1} [h(b_{i_2})]Z_{i_2} \ . \ . \ . \ [h(b_{i_k})]Z_{i_k} \ . \ . \ . \ . \qquad (*)$$

for some $i_1, . . ., i_k$ in {1, . . ., j}, where none of $Z_{i_1}, . . ., Z_{i_k}$ 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 (*) then w would contain two disjoint occurrences of an identical subword of length k. This however contradicts the assumption that $\mu(w) \leqslant k$.

5) From 3 and 4 it follows that $\mu(x(w)) \leqslant k \cdot 2(2n + 9) \leqslant 2 \cdot k \cdot 2(n + 5) < 2k \cdot 4n \leqslant 2k \cdot 4 \log_2|x(w)|$. From Theorem 2 we know that if f is a slow function such that $f(s) \geqslant 4 \log_2 s$ 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 $k \leqslant 4 \log_2|w|$.

Thus $2k \cdot 4 \log_2|x(w)| \leqslant 2 \cdot 4 \log_2|w| \cdot 4 \log_2|x(w)| \leqslant 32 \cdot \log_2^2|x(w)|$. So $\mu(x(w)) \leqslant 32 \cdot \log_2^2|x(w)|$.

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

Next we prove that in an EDTOL language L which is a subset of $\mathcal{B}_1$ if w is 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 $L \subseteq \mathcal{B}_1$. 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 $L \subseteq \mathcal{B}_1$ and let f be a slow function. Let $G = \langle V, \mathcal{P}, \omega, \Sigma \rangle$ be an EPDTOL system such that $L(G) = L$. (See Theorem 4 in Ehrenfeucht and Rozenberg [5].) Clearly we can 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

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

be a derivation of w in G and let

$D_1 = ((x_{i_0}, \ldots, x_{i_q}), (\overline{T}_{i_0}, \ldots, \overline{T}_{i_{q-1}}), \mathcal{O}_1)$

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 eixsts a big recursive letter R in D_1, such that either, for every j in $\{i_0, \ldots, i_{q-1}\}$, $\overline{T}_j(R) = \alpha_R^{(j)} R \beta_R^{(j)}$ with $\alpha_R^{(j)} \neq \Lambda$,
or, for every j in $\{i_0, \ldots, i_{q-1}\}$, $\overline{T}_j(R) = \alpha_R^{(j)} R \beta_R^{(j)}$ with $\beta_R^{(j)} \neq \Lambda$.

(We will assume, without the loss of generality, that for every j in $\{i_0, \ldots, i_{q-1}\}$, $\overline{T}_j(R) = \alpha_R^{(j)} R \beta_R^{(j)}$ with $\alpha_R^{(j)} \neq \Lambda$.)

3) For every big recursive letter B in D_1, and for every i, j in $\{i_0, \ldots, i_{q-1}\}$, if $B \underset{\overline{T}_i}{\Longrightarrow} u_1 B u_2$ and $B \underset{\overline{T}_j}{\Longrightarrow} v_1 B v_2$ 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 $\text{Score}(\text{Contr}_D(\overline{T}_i(A))) \leq |\text{Contr}_D(\overline{T}_i(A))| \leq f(|w|)$ 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.)

<u>Lemma 4.</u> For every j in $\{i_0, \ldots, i_{q-1}\}$,
$||\text{Score}(\text{Contr}_D(\overline{T}_j(\alpha_R^{(j)})))|| > 0$.

<u>Proof of Lemma 4.</u>

Let us assume, to the contrary, that $\text{Score}(\text{Contr}_D(\overline{T}_j(\alpha_R^{(j)}))) = 0$.

Note that $\overline{T}_j(\alpha_R^{(j)})$ contains 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 $n \geq 0$ there is a word in L(G) which contains $(\text{Contr}_D(T_j(\alpha_R^{(j)})))^n$ as a subword. But (with our assumption that $\text{Score}(\text{Contr}_D(\overline{T}_j(\alpha_R^{(j)}))) = 0$) if γ is a subword of $(\text{Contr}_D(\overline{T}_j(\alpha_R^{(j)})))^n$ then $\text{Score}(\gamma) \leqslant 2|\text{Contr}_D(\overline{T}_j(\alpha_R^{(j)}))|$. This however implies that L(G) would contain words with arbitrarily long subwords the score of which is bounded by $2|\text{Contr}_D(\overline{T}_j(\alpha_R^{(j)}))|$ which contradicts Lemma 3.

Thus Lemma 4 holds.

Lemma 5. For every i, j in $\{i_0, \ldots, i_{q-1}\}$,
$\text{sign}(\text{Score}(\text{Contr}_D(\overline{T}_i(\alpha_R^{(i)})))) = \text{sign}(\text{Score}(\text{Contr}_D(\overline{T}_j(\alpha_R^{(j)}))))$.

Proof of Lemma 5.

Let us assume, to the contrary, that
$\text{sign}(\text{Score}(\text{Contr}_D(\overline{T}_i(\alpha_R^{(i)})))) = \text{sign}(\text{Score}(\text{Contr}_D(\overline{T}_j(\alpha_R^{(j)}))))$
for example that
$\text{sign}(\text{Score}(\text{Contr}_D(\overline{T}_i(\alpha_R^{(i)})))) > 0$ and
$\text{sign}(\text{Score}(\text{Contr}_D(\overline{T}_j(\alpha_R^{(j)})))) < 0$.

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 $\overline{T}_i(\alpha_R^{(i)})$, $\overline{T}_i(\alpha_R^{(j)})$, $\overline{T}_j(\alpha_R^{(i)})$ and $\overline{T}_j(\alpha_R^{(j)}$ consist of small recursive letters only.)

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

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

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

$\tau_1(\alpha_R^{(i)}R) = \overline{T}_i(\alpha_R^{(i)})\overline{T}_j(\alpha_R^{(i)})\alpha_R^{(j)}R\delta_1$, for some δ_1 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) $\tau_{p_1} = (\overline{T}_j)^{p_1}\overline{T}_i$.

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

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

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

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

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

for some δ_{p_1+1} in V*.

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

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

for some δ_{p_1+1} in V*.

.
.
.

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

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

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

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

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

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

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

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

and so on.

Thus what we are doing is alternating sequences of applications of $\overline{T}_i$ and $\overline{T}_j$ 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 $4 \cdot \max\{|\overline{T}_i(\alpha_R^{(i)})|, |\overline{T}_j(\alpha_R^{(i)})|, |\overline{T}_i(\alpha_R^{(j)})|, |\overline{T}_j(\alpha_R^{(j)})|\}$. 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

$D_1 = ((y_0, \ldots, y_q), (P_0, \ldots, P_{q-1}), \mathscr{T}_1)$

where in fact

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

Thus we have now, for each i in $\{0, \ldots, q-1\}$, $P_i(R) = \alpha_R^{(i)}R\beta_R^{(i)}$ with $\alpha_R^{(i)} \neq \Lambda$.

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

$\Theta_1 = \mathrm{Score}(\mathrm{Contr}_D(P_1(\alpha_R^{(0)})P_2(\alpha_R^{(1)}) \ldots P_{q-1}(\alpha_R^{(q-2)})).$

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

$$\Theta_2 = \sum_{j=1}^{q-2} \mathrm{Score}(\Delta(P_j(\alpha_R^{(j)}))).$$

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

$$\Theta_1 = \sum_{j=1}^{q-1} \mathrm{Score}(\mathrm{Contr}_D(P_j(\alpha_R^{(j-1)})) =$$

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

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

and

$$\Theta_2 = \sum_{j=1}^{q-2} \mathrm{Score}(\Delta(P_j(\mathrm{Big}(\alpha_R^{(j)})))) +$$

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

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

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

(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 $\alpha_R^{(0)} = Z_1B_1Z_2B_2 \; . \; . \; . \; Z_\ell B_\ell Z_{\ell+1}$, where $Z_1, \; . \; . \; ., \; Z_{\ell+1}$ do not contain big letters and $B_1, \; . \; . \; ., \; B_\ell$ are big letters. (Note that $\ell < \#V$.)

Then

$$\Theta_1 - \Theta_2 = \text{Score}(\text{Contr}_D(P_{q-1}(\text{Big}(\alpha_R^{(q-2)})))) + \sum_{i=1}^{\ell+1} \text{Score}(\text{Contr}_D(P_1(Z_i)))$$

Let $\alpha_R^{(q-2)} = u_1C_1u_2C_2 \; . \; . \; . \; u_tC_tu_{t+1}$, where $u_1, \; . \; . \; ., \; u_{t+1}$ do not contain big letters and $C_1, \; . \; . \; ., \; C_t$ are big letters. (Note that $t < \#V$.)

Then

$$\Theta_1 - \Theta_2 = \sum_{i=1}^{t} \text{Score}(\text{Contr}_D(P_{q-1}(C_i))) + \sum_{i=1}^{\ell+1} \text{Score}(\text{Contr}_D(P_1(Z_i))).$$

Thus

$$\Theta_1 - \left(\sum_{i=1}^{t} \text{Score}(\text{Contr}_D(P_{q-1}(C_i))) + \sum_{i=1}^{\ell+1} \text{Score}(\text{Contr}_D(P_1(Z_i))) \right) = \Theta_2.$$

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

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

Θ_1,

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

$\text{Score}(\text{Contr}_D(P_1(Z_i)))$ for $1 \leqslant i \leqslant \ell + 1$

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

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 $L \subseteq \mathcal{B}_1$ then $L \neq \mathcal{B}_1$.

Proof.

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

We leave to the reader the easy standard proofs of our next 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 $i > 1$, $\mathcal{B}_i$ is not an EDTOL language.

Proof

As a direct corollary from Theorem 5 we have that $\mathcal{B}_1$ is not an EDTOL language. But then from THeorem 6 it follows that for every $i \geqslant 0$ $\mathcal{B}_i$ is not an EDTOL language.

Let us now recall the notion of a Dyck language (see, e.g., Salomaa [10], p.68). Let, for $i \geq 1$, $V_i = \{a_1, a'_1, a_2, a'_2, \ldots, a_n, a'_n\}$. The context free language D_i generated by the context free grammar

$\langle \{S\}, V_i, \{S \to \Lambda, S \to SS, S \to a_1Sa'_1, \ldots, S \to a_iSa'_i\}, S \rangle$

is termed the Dyck language over the alphabet V_i.

Theorem 9. For every $i \geqslant 8$, D_i is not an EDTOL language.

Proof

Let us first recall the following well-known result (see, e.g., Salomaa [10], 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 $\mathcal{B}_1$ is a context free language over an alphabet Σ consisting of $m = 2$ letters and by Theorem 8, $\mathcal{B}_1$ 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 $i \geqslant 8$ 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 eixst context free languages which are not EDT0L languages. This result is directly used in Ehrenfeucht and Rozenberg [4] to show existence of indexed languages (see Aho [1]) that are not ET0L languages.

In fact our results have further implications.

1) They settle a controversy on the existence of context free languages that are not parallel context free languages (see Siromoney and Krithivasan [13] and Skyum [14]). Because the class of parallel context free languages is clearly contained in the class of EDT0L languages we have provided an alternative proof to this of Skyum [14] that, almost all, Dyck languages are not parallel context free languages.

2) Following Salomaa [12], our Theorem 10 implies that (we use here the Salomaa's notation four [12]):
the pairs (CF, IP), (ED, PPDA), (ED, ET0L) are incomparable, IP is properly contained in RP, ER is not contained in ET0L 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 EDT0L languages.

REFERENCES

1. A. Aho, Indexed grammars - an extension of context-free grammars, Journal of the ACM, v. 15, 647-671, 1968.

2. P.A. Christensen, Hyper AFL's and ETOL systems, in [9], 1974.

3. P. Downey, Developmental systems and recursion schemes, Proceedings of the Conference on Biologically Motivated Automata Theory, McLean, Virginia, 1974.

4. A. Ehrengeucht and G. Rozenberg, On decomposing some ETOL languages into deterministic ETOL languages, University of Colorado, Boulder, Department of Computer Science, Technical Report Number CU-CS-045-74.

5. A. Ehrenfeucht and G. Rozenberg, On the structure of derivations in EDTOL systems, University of Colorado, Boulder, Department of Computer Science, Technical Report Number CU-CS-046-74.

6. A. Ehrenfeucht and G. Rozenberg, A pumping theorem for EDTOL languages, University of Colorado, Boulder, Department of Computer Science, Technical Report Number CU-CS-047-74.

7. G.T. Herman and G. Rozenberg, Developmental Systems and Languages, North-Holland Publishing Company, 1974.

8. G. Rozenberg, Extension of tabled OL systems and languages, International Journal of Computer and Information Sciences, v. 2, 311-334, 1973.

9. G. Rozenberg and A. Salomaa, L systems, Lecture Notes in Computer Science Number 15, Springer-Verlag, 1974.

10. A. Salomaa, Formal languages, Academic Press, 1973.

11. A. Salomaa, Recent results on L systems, Proceedings of the Conference on Biologically Motivated Automata Theory, McLean, Virginia, 1974.

12. A. Salomaa, Parallelism in rewriting systems, Proceedings of the Second Colloquium on Automata, Languages and Programming, Saarbrucken, July, 1974.

13. R. Siromoney and K. Krithivasan, Parallel context free languages, Information and Control, v. 24, 155-162, 1974.

14. S. Skyum, Parallel context free languages, Technical Report of the Department of Computer Science, University of Aarkus, Denmark, 1974.