REVUE FRANÇAISE D'AUTOMATIQUE INFORMATIQUE
RECHERCHE OPÉRATIONNELLE. INFORMATIQUE THÉORIQUE

A. EHRENFEUCHT
G. ROZENBERG

A pumping theorem for deterministic ETOL languages

Revue française d'automatique informatique recherche opérationnelle. Informatique théorique, tome 9, n° R2 (1975), p. 13-23

<http://www.numdam.org/item?id=ITA_1975__9_2_13_0>

R.A.I.R.O.
(9e année, août 1975, R-2, p. 13-23)

A PUMPING THEOREM FOR DETERMINISTIC ETOL LANGUAGES

by A. EHRENFEUCHT [1] and G. ROZENBERG [2]

Communicated by W. BRAUER

Abstract. — *This paper is concerned with deterministic ETOL languages. A theorem is proved which, roughly speaking, says that if a deterministic ETOL language contains a word with a special property then it must contain an infinite set of words obtained from the given one by « synchronously pumping » a number of subwords of the given word. This theorem has a number of applications for proving that certain languages are not deterministic ETOL languages.*

I. INTRODUCTION

The theory of L systems which originated from the works of Lindenmayer (see Lindenmayer [6]) turned out to be useful and interesting from both the biological and formal points of view (see, e.g., Herman and Rozenberg [5] and Rozenberg and Salomaa [8]).

In fact the theory of L systems forms today one of the most vigorously investigated topics in formal language theory. It shed new light on basic problems in formal language theory and it introduced the whole range of new problems and techniques for solving them.

One of the research areas in the theory of L systems is an investigation of the (combinatorial) structure of L languages (as opposed to the structure of various classes of L languages). We consider this to be one of the central areas in the theory. For example, unless we learn about a « structure of

(1) Department of Computer Science, University of Colorado at Boulder, Colorado U.S.A.
(2) Department of Mathematics, Utrecht University, the Netherlands and Department of Mathematics, University of Antwerp, U.I.A., Wilrijk, Belgium.

a single L language » there is a little chance that we will be able to have a feedback from the theory of L systems into the area where all this research originated (theoretical biology) or into the areas where undoubtedly L languages have some advantages over traditional Chomsky languages (for example linguistics or theoretical computer science).

This paper concentrates on the so called deterministic ETOL languages, one of the central families of languages in the L systems theory (see, e.g., Downey [1], Ehrenfeucht and Rozenberg [4], Rozenberg [7] and Salomaa [9]).

In trying to discover a result on L languages which would be analogous to the famous « pumping lemma for context free languages » (see, e.g., Salomaa [10], p. 56), which is probably the most useful known result on the structure of a context free language, the basic difficulty met can be described as follows.

In context free grammars in long enough derivations one can always find a self-embedding nonterminal and then iterate its rewritings an arbitrary number of times *with the rest of the string remaining unchanged.* This is due to a totally sequential way of rewritings in context free grammars (one occurrence of a symbol is rewritten in a single step). This « trick » does not work in L systems because in a single derivation step all occurrences of all symbols in the string under consideration must be rewritten. In fact such a single iteration can not take place because even the simplest classes of L languages contain languages such that the sets of lengths of their strings do not have to contain an arithmetic progression.

We have resolved the difficulty in this way that

(1) we have used a classification of symbols much finer than that of dividing them in self-embedding and non-self embedding categories only (such a classification was introduced in Ehrenfeucht and Rozenberg [4]), and

(2) we have considered only special words in the given language, the so called « f-random words ».

This is presented in Section III of this paper.

Section IV provides the proof of our main result and Section V provides some of its applications for a rather difficult task of proving that certain languages are not deterministic ETOL languages.

Throughout this paper we use standard formal-language theoretic notation and terminology.

II. EDTOL SYSTEMS AND LANGUAGES

In this section we recall the definitions of deterministic ETOL systems and languages (see Rozenberg [7]) 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 $G = \langle V, \mathfrak{P}, \omega, \Sigma \rangle$ such that

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

2) $\mathfrak{P}$ 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 $\mathfrak{P}$ satisfies the following condition : 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 $\mathfrak{P}$ are nonempty sets.)

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

Definition 2. Let $G = \langle V, \mathfrak{P}, \omega, \Sigma \rangle$ be an EDTOL system. Let $x \in V^+$, $x = a_1 \ldots a_k$, where each a_j, $1 \leqslant j \leqslant k$, 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 $\mathfrak{P}$ 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^* ($n \geqslant 1$) 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 $G = \langle V, \mathfrak{P}, \omega, \Sigma \rangle$ be an EDTOL system. The language of G, denoted as $L(G)$, is defined as $L(G) = \{ x \in \Sigma^* : w \overset{*}{\underset{G}{\Rightarrow}} x \}$.

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

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

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

3) In fact each table P from $\mathfrak{P}$ 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, ..., finally P_m), etc. In this sense $P_m \ldots P_1(x)$ denotes the (unique) word y which is obtained by rewriting x by the sequence of tables $P_1, P_2, \ldots, P_m$.

We end this section with two examples of EDTOL systems and languages.

EXAMPLE 1. Let $G_1 = \langle V, \mathfrak{T}, \omega, \Sigma \rangle$ where $V = \{ A, B, a \}$, $\Sigma = \{ a \}$, $\omega = AB$ and $\mathfrak{T} = \{ P_1, P_2 \}$, where

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

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

EXAMPLE 2. Let $G_2 = \langle \{ a, b, A, B, C, D, F \}, \mathfrak{T}, CD, \{ a, b \} \rangle$, where $\mathfrak{T} = \{ P_1, P_2, P_3 \}$ and

$$P_1 = \{ a \to F, b \to F, A \to A, B \to B, C \to ACB, D \to DA \},$$
$$P_2 = \{ a \to F, b \to F, A \to A, B \to B, C \to CB, \quad D \to D \},$$
$$P_3 = \{ a \to F, b \to F, A \to a, \ B \to b, \ C \to \Lambda, \quad D \to \Lambda \}.$$

G_2 is an EDTOL system which is not propagating, and

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

III. DERIVATIONS IN EDTOL SYSTEMS

A central notion in investigating the structure of an EDTOL language is « a derivation in an EDTOL system ».

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

1) $x_0, ..., x_k$ are in V^*.

2) $T_0, ..., T_{k-1}$ are in $\mathfrak{T}$,

3) $x_0 = x$, $x_k = y$ and $T_i(x_i) = x_{i+1}$ for $0 \leqslant i \leqslant k - 1$.

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

Definition 5. Let $G = \langle V, \mathfrak{T}, \omega, \Sigma \rangle$ be an EDTOL system and let $D = ((x_0, ..., x_k), T_0, ..., T_{k-1}))$ be a derivation in G. For each occurrence a in x_j, $1 \leqslant j \leqslant k$, 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.

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

1) $0 \leqslant i_0 < i_1 < ... < i_q \leqslant k - 1$,

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

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.) Given a subderivation $\bar{D}$ of N and an occurrence a in a word of $\bar{D}$ we talk about $\mathrm{Contr}_D(a)$ in an obvious sense.

Definition 7. Let $G = \langle V, \mathfrak{T}, \omega, \Sigma \rangle$ be an EPDTOL system and let f be a function from $\mathfrak{R}_{\mathrm{pos}}$ into $\mathfrak{R}_{\mathrm{pos}}$. Let D be a derivation in G and let $\bar{D} = ((x_0, ..., x_k), (T_0, ..., T_{k-1}))$ be a subderivation of D. Let a be an occurrence (of A from V) in x_t for some t in $\{ 0, ..., k \}$.

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

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

1) $\mathrm{Min}\,(x_0) = \mathrm{Min}\,(x_1) = ... = \mathrm{Min}\,(x_k)$, $\mathrm{Min}\,(x)$ denotes the set of all letters occurring in x.

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

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

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

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

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 $B \to \alpha$ is in T_j, then
- if B is small recursive, then $\alpha = B$, and
- if B is nonrecursive then α consists of small recursive letters only.

6) For every i, j in $\{ 0, ..., k \}$ 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 $|\mathrm{Contr}_D(a)| = |\mathrm{Contr}_D(b)|$.

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

Definition 9. Let f be a function from $\mathcal{R}_{\mathrm{pos}}$ into $\mathcal{R}_{\mathrm{pos}}$. We say that f is *slow* if

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

Thus, for example, a constant function and $(\log x)^k$ are examples of slow functions.

Definition 10. Let Σ be a finite alphabet and let f be a function from $\mathcal{R}_{\mathrm{pos}}$ into $\mathcal{R}_{\mathrm{pos}}$. 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 proved in Ehrenfeucht and Rozenberg [4], is the central result for proving our pumping theorem for EDTOL languages.

Theorem 1. For every EPDTOL system G and every slow function f there exist r in $\mathcal{R}_{\mathrm{pos}}$ 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 $|w|^r$.

IV. A PUMPING THEOREM FOR EDTOL LANGUAGES

In this section we prove the main result of this paper.

Theorem 2. For every EDTOL language K and for every slow function f there exists a constant s such that for every f-random word x in K longer than s there exists a positive integer constant t and words $x_0, ..., x_t, \sigma_1, ..., \sigma_t$ with $\sigma_1\sigma_2 ... \sigma_t \neq \Lambda$ such that $x = x_0x_1 ... x_t$ and for every non-négative integer n, $x_0\sigma_1^n x_1\sigma_2^n ... x_t\sigma_t^n$ is in L.

Proof

Let K be an EDTOL language and let f be a slow function. According to Theorem 4 in Ehrenfeucht and Rozenberg [4] we can assume that $K - \{ \Lambda \}$ is generated by an EPDTOL system $G = \langle V, \mathfrak{T}, \omega, \Sigma \rangle$. We also assume that K contains infinitely many f-random words, because otherwise Theorem 2 is trivially true.

Now by Theorem 1 we can assume that there exists a constant s such that if w is an f-random word in $L(G)$ longer than s then every derivation of w in G contains a neat subderivation containing at least three words.

Thus let x be an f-random word in $L(G)$ such that $|x| > s$. Let $D = ((y_0, \ldots, y_p), (T_0, \ldots, T_{p-1}))$ be a derivation of x in G and let $\bar{D} = ((y_{i_0}, \ldots, y_{i_q}), (P_{i_0}, \ldots, P_{i_{q-1}}))$ be a neat subderivation of D where $q \geqslant 2$ and $0 \leqslant i_0 < \ldots < i_q \leqslant p - 1$.

For j in $\{ 0, \ldots, q - 1 \}$ let us call a big recursive letter A in y_{i_j} *expansive* if $A \xrightarrow[P_j]{} \alpha A \beta$ where $\alpha\beta \neq \Lambda$. Note that by the definition of a neat subderivation (see points 3, 5 and 7 in Definition 8) y_{i_0} contains an expansive big recursive letter.

We can write y_{i_0} as

$$y_{i_0} = \gamma_0 B_1 \gamma_1 \ldots B_k \gamma_k$$

where $B_1, \ldots, B_k$ are big recursive letters and none of the words $\gamma_0, \ldots, \gamma_k$ contains a big recursive letter. Note that by the definition of a neat subderivation (see point 4 in Definition 8), $k \leqslant \# V$ and so $1 \leqslant k \leqslant \# V$.

Let, for i in $\{ 1, \ldots, k \}$, $P_0(B_i) = \alpha_{0i} B_i \beta_{0i}$.

Hence,

$$y_{i_1} = P_0(\gamma_0)\alpha_{01} B_1 \beta_{01} P_0(\gamma_1) \ldots \alpha_{0k} B_k \beta_{0k} P_0(\gamma_k)$$

and if we set $R = T_{i_1} T_{i_1+1} \ldots T_{p-1}$ we have

$$x = y_p = RP_0(\gamma_0)R(\alpha_{01})R(B_1)R(\beta_{01})RP_0(\gamma_1) \ldots R(\alpha_{0k})R(B_k)R(\beta_{0k})RP_0(\gamma_k).$$

But we can change the derivation D in such a way that, for an arbitrary $n \geqslant 1$, we can apply P_0 n times to y_{i_0} and then apply R (let us denote the so obtained word by $x^{(n)}$). In this way we have

$$\begin{aligned}
P_0(y_{i_1}) &= P_0^2(\gamma_0)P_0(\alpha_{01})\alpha_{01} B_1 \beta_{01} P_0(\beta_{01}) \ldots P_0(\alpha_{0k})\alpha_{0k} B_k \beta_{0k} P_0(\beta_{0k}) P_0^2(\gamma_k), \\
P_0^2(y_{i_1}) &= P_0^3(\gamma_0)P_0^2(\alpha_{01})P_0(\alpha_{01})\alpha_{01} B_1 \beta_{01} P_0(\beta_{01}) P_0^2(\beta_{01}) \ldots \\
&\quad \ldots P_0^2(\alpha_{0k})P_0(\alpha_{0k})\alpha_{0k} B_k \beta_{0k} P_0(\beta_{0k}) P_0^2(\beta_{0k}) P_0^3(\gamma_k), \\
&\quad \vdots \qquad \vdots \qquad \vdots \qquad \vdots \qquad \vdots \qquad \vdots \\
P_0^n(y_{i_1}) &= P_0^{n+1}(\gamma_0)P_0^n(\alpha_{01}) \ldots P_0(\alpha_{01})\alpha_{01} B_1 \beta_{01} P_0(\beta_{01}) \ldots P_0^n(\beta_{01}) \ldots \\
&\quad \ldots P_0^n(\alpha_{0k}) \ldots P_0(\alpha_{0k})\alpha_{0k} B_k \beta_{0k} P_0(\beta_{0k}) \ldots P_0^n(\beta_{0k}) P_0^{n+1}(\gamma_k)
\end{aligned}$$

and finally

$$x^{(n)} = RP_0^n(y_{i_1}) = RP_0^{n+1}(\gamma_0)RP_0^n(\alpha_{01})\dots RP_0(\alpha_{01})R(\alpha_{01})R(B_1)\dots$$
$$\dots R(B_k)R(\beta_{0k})RP_0(\beta_{0k})\dots RP_0^n(\beta_{0k})RP_0^{n+1}(\gamma_k).$$

However $P_0(\gamma_0), P_0(\gamma_1), \dots, P_0(\gamma_k), P_0(\alpha_{01}), \dots, P_0(\alpha_{0k}), P_0(\beta_{01}), \dots, P_0(\beta_{0k})$ are words over small recursive letters only.

Consequently, for every $m \geqslant 1$,

$$P_0^m(\gamma_0) = P_0(\gamma_0), \dots, P_0^m(\gamma_k) = P_0(\gamma_k),$$
$$P_0^m(\alpha_{01}) = P_0(\alpha_{01}), \dots, P_0^m(\beta_{0k}) = P_0(\beta_{0k}).$$

Thus

$$x^{(n)} = RP_0(\gamma_0)(RP_0(\alpha_{01}))^nR(\alpha_{01})R(B_1)R(\beta_{01})(R(\beta_{01}))^nRP_0(\gamma_1)\dots$$
$$\dots(RP_0(\alpha_{0k}))^nR(\alpha_{0k})R(B_k)R(\beta_{0k})(RP_0(\beta_{0k}))^nRP_0(\gamma_k).$$

Let us notice that (because y_{i_0} contains an expansive big recursive letter) at least one from $RP_0(\alpha_{01}), RP_0(\beta_{01}), \dots, RP_0(\alpha_{0k}), RP_0(\beta_{0k})$ is a nonempty word. Also, for every $n \geqslant 0$, $x^{(n)}$ is clearly in $L(G)$. Thus if we set

$$\begin{aligned}
x_0 &= RP_0(\gamma_0),\\
\sigma_1 &= RP_0(\alpha_{01}),\ x_1 = R(\alpha_{01})R(B_1)R(\beta_{01}),\\
\sigma_2 &= RP_0(\beta_{01}),\ x_2 = RP_0(\gamma_1),\\
&\ \ \vdots \qquad\quad \vdots \qquad\quad \vdots \qquad\quad \vdots\\
\sigma_t &= RP_0(\beta_{0k}),\ x_t = RP_0(\gamma_k),
\end{aligned}$$

then we see that Theorem 2 holds. (Note that $t \leqslant 2(\#\, V)$.

V. APPLICATIONS

In this section we indicate a number of applications of Theorem 2.

First we need a definition.

Definition 11. Let K be a language. The *length set of* K, denoted as *Length*(K), is defined by *Length*$(K) = \{ n :$ there exists a word x in K, such that $|x| = n \}$.

As a direct consequence of Theorem 2 we get the following result.

Theorem 3. If f is a slow function and K is an EDTOL language which contains infinitely many f-random words, then Length(K) includes an arithmetic progression.

Here is a rather strange, but instructive, example of an application of Theorem 3 to prove that a particular language is not an EDTOL language.

Let $\Sigma = \{ 0, 1, \$ \}$. Let for each positive integer k, α_k denote an arbitrary, but fixed, word of the form $x_1\$x_2\$ \ldots \$x_{2^k}\$$ where $x_1, \ldots, x_{2^k}$ exhaust the set of all different words of length k over the alphabet $\{ 0, 1 \}$.

Let $M = \{ \alpha_k : k \geqslant 1 \}$.

Corollary 1. *M is not an EDTOL language.*

Proof

This follows directly from Theorem 3 once we notice that if f is the function defined by $f(y) = 2 \log y$ then each word in M is f-random. (Notice that α_k has no identical disjoint subwords of length larger than $2k$. But $\log |\alpha_k| = \log 2^k(k+1) = k + \log(k+1) > k$, and so $2k < 2 \log |\alpha_k|$).

Now we would like to point out that restricting ourselves to f-random words only still leaves us (in general) with a considerable number of words providing that f is not « too slow ». This is shown as follows.

Lemma 1. Let Σ be a finite alphabet such that $\# \Sigma = m \geqslant 2$. Let f be a function from $\mathcal{R}_{pos}$ into $\mathcal{R}_{pos}$ such that, for every x in $\mathcal{R}_{pos}$, $f(x) \geqslant 4 \log_2 x$.

Then, for every positive integer n,

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

Proof

Let Σ and f satisfy the statement of the theorem.

First let us find an upper bound on the number of words in Σ^* of length n which are not f-random.

1) If a word w is not f-random, then it can be written in the form $a_1 \ldots a_{n_1} \alpha a_{n_1+|\alpha|+1} \ldots a_{n_2} \alpha a_{n_2+|\alpha|+1} \ldots a_n$ for some $a_1, \ldots, a_{n_1}, a_{n_1+|\alpha|+1}, \ldots, a_n$ in Σ and α in Σ^+ where $|\alpha| > f(n)$.

2) With the fixed values of n_1, n_2 and $|\alpha|$ we may have at most $m^{|\alpha|} \cdot m^{n-2|\alpha|} = m^{n-|\alpha|}$ words which are not f-random. But $|\alpha| > f(n)$ and so $m^{n-|\alpha|} < m^{n-f(n)}$.

3) The number of choices for n_1, n_2 and α is not larger than n^3.

4) Thus the number of words of length n which are not f-random is smaller than $n^3 \cdot m^{n-f(n)}$.

Consequently,

$$\frac{\# \{ w \in \Sigma^* : |w| = n \text{ and } w \text{ is not } f\text{-random} \}}{m^n} < \frac{n^3 \cdot m^{n-f(n)}}{m^n} = \frac{n^3}{m^{f(n)}}.$$

But $f(n) \geqslant 4 \log_2 n$ and so

$$\frac{n^3}{m^{f(n)}} \leqslant \frac{n^3}{m^{4 \log_2 n}} = \frac{n^3}{2^{\log_2 m \cdot 4 \log n}} = \frac{n^3}{2^{\log_2 n \cdot 4 \log_2 m}} = \frac{n^3}{n^{4 \log_2 m}} \leqslant \frac{1}{n}.$$

Thus

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

which proves the lemma.

As a direct corollary from Theorem 3 and Lemma 1 we have the following result.

Theorem 4. Let K be an EDTOL language over an alphabet Σ, where $\# \Sigma = m \geqslant 2$. If length$(K)$ does not contain an arithmetic progression then

$$\lim_{n \to \infty} \frac{\#\{ w \in K^* : |w| = n \}}{m^n} = 0.$$

Using this result we can show several interesting examples of languages which are not EDTOL languages.

Corollary 2. Let Σ be a finite alphabet with $\# \Sigma \geqslant 2$. Let k be a positive integer larger than 1. Then

1) $\{ w \in \Sigma^* : |w| = k^n \text{ for some } n \geqslant 0 \}$ is not an EDTOL language.
2) $\{ w \in \Sigma^* : |w| = n^k \text{ for some } n \geqslant 0 \}$ is not an EDTOL language.

Let us finally remark that finding examples of languages which are not EDTOL languages is very useful for finding examples of languages which are not ETOL languages. In fact by Theorems 1 and 2 from Ehrenfeucht, Rozenberg and Skyum [3] each example of a language which is not an EDTOL language may be used to provide infinitely many examples of languages which are not ETOL languages.

REFERENCES

[1] P. J. DOWNEY, Developmental systems and recursion schemes, *Proceedings of the Conference on Biologically Motivated Theory*, McLean, Virgina, 1974, p. 54-58.

[2] A. EHRENFEUCHT and G. ROZENBERG, The equality of EOL languages and codings of OL languages, *International Journal of Computer Mathematics*, *4* (1974), 95-104.

[3] A. EHRENFEUCHT, G. ROZENBERG and S. SKYUM, A relationship between ETOL and EDTOL languages, submitted for publications; also available as a technical report PB-40/74 of the Computer Science Department, Aarhus University, Aarhus, Denmark, 1974.

[4] A. EHRENFEUCHT and G. ROZENBERG, On structure of derivations in EDTOL systems, University of Colorado at Boulder, Department of Computer Science Technical Report, # CU-CS-046-74, 1974.

[5] G. T. HERMAN and G. ROZENBERG, *Developmental systems and languages*, North Holland Publishing Company, 1975.

[6] A. LINDENMAYER, Mathematical models for cellular interactions in development, Parts I and II, *Journal of Theoretical Biology*, vol. *18*, (1968), 280-315.

[7] G. ROZENBERG, Extension of tabled OL systems and languages, *International Journal of Computer and Information Sciences*, *2*, (1973), 311-336.

[8] G. ROZENBERG and A. SALOMAA, L *systems*, *Lecture Notes in Computer Science Number 15*, Springer-Verlag, 1974.

[9] A. SALOMAA, Recent results on L systems, *Proceedings of the Conference on Biologically Motivated Automata Theory*, McLean, Virginia, 1974, p. 38-45.

[10] A. SALOMAA, *Formal languages*, Academic Press, 1973.