*International Journal of Computer and Information Sciences, Vol. 2, No. 4, 1973*

# Extension of Tabled 0*L*-Systems and Languages

Grzegorz Rozenberg[1,2]

*Revised June 1973*

This paper introduces a new family of languages which originated from a study of some mathematical models for the development of biological organisms. Various properties of this family are established and in particular it is proved that it forms a full abstract family of languages. It is compared with some other families of languages which have already been studied and which either originated from the study of models for biological development or belong to the now standard Chomsky hierarchy. A characterization theorem for context-free languages is also established.

## 1. INTRODUCTION

In 1968 Lindenmayer(18) proposed a theory for the development of filamentous organisms. During the last four years this theory (now called the theory of developmental or *L*-systems) turned out to be useful and interesting from both the biological and formal points of view.

From the biological point of view *L*-systems have provided a useful theoretical framework within which the nature of cellular behavior in development can be discussed, computed, and compared. Their study has also provided a number of biologically interesting results.(1,9,10,12,18–21,36)

Although *L*-systems were originally described in terms of linear arrays of automata, they were later reformulated in a more linguistic way using a grammarlike concept. In this way the theory of *L*-systems has moved closer to formal language theory and in fact has been found very interesting from

This research has been supported by NSF Grant GJ 998.

[1] Department of Computer Science, State University of New York at Buffalo, Amherst, New York.

[2] Mathematical Institute, Utrecht University, Utrecht–Uithof, Netherlands.

a formal language theory point of view. It has provided us with an alternative to the now standard Chomsky framework (see, e.g., Refs. 6 and 17) for defining languages. As a result of the different approach, the families of languages defined by $L$-systems are rather different from the more traditional families that have been studied (see, e.g., Refs. 11, 15, 19–21, 24, 25, 27, and 32–34). The novelty of this approach is also reflected by the fact that most standard proof techniques in formal language theory were found inapplicable to $L$-languages (languages defined by $L$-systems) and a set of new techniques had to be devised. Also, within the framework of $L$-systems one can discuss sequences as well as sets (languages) of strings, in contrast to the emphasis on languages within the Chomsky framework (for examples of such research see Refs. 5, 13, 16, 21, 22, 28, 29, and 35). Apart from the work mentioned above, $L$-systems have also been investigated in Refs. 3, 4, 8, 15, 22, 26, 30, 31, and 36.

An important subclass of $L$-systems are the so-called $0L$-systems (see, e.g., Refs. 19 and 32), which are models for multicellular organisms in which there is no communication among contemporaneous cells. In such an organism each cell is an autonomous unit which behaves according to uniform rules, applicable to all cells in the organism. Each cell may be present in one of finite number of states, and whether it divides, dies, or change its state in a given time interval is determined solely by its current state.

Thus a $0L$-system has the following components:

(a) A finite set of symbols $\Sigma$ the *alphabet*.

(b) A starting string $\omega$, the *axiom*.

(c) A finite set of *productions* which tell us by what strings in $\Sigma^*$ a symbol may be replaced. The set of productions that may be applied to a certain symbol depends on the symbol only. In every step of a derivation *all* symbols in the string must be simultaneously replaced according to the production rules.

The language of a system is defined as the set of *all* strings which can be derived from the axiom, including the axiom itself.

Even within organisms where there is essentially no communication between cells one may observe synchronous behavior of different cells positioned in different places in the organism. One such example is the behavior of the organism in the presence of the variable environment. An example of such behavior is described in Ref. 36. In fact, in Ref. 36 empirical results concerning effects of light and darkness on some of the filamentous fungi are presented, and then a conclusion reached that for those two different external conditions one needs two different sets of developmental rules which *cannot* be mixed up.

To allow the discussion of such phenomena within the framework of $L$-systems, the idea of a $0L$-system was generalized in Ref. 24 to allow a finite number of sets of productions, rather than one, to be present in a system (each such set is called a table). The resulting construct is called a tabled $0L$-system (or $T0L$ system for short). In every step of a derivation *all* symbols in the string must be simultaneously replaced according to the production rules chosen from one single, arbitrarily chosen table of the system. Again, the language of a system is the set of all strings which can be derived from the axiom including the axiom itself.

In this paper we generalize the notion of a $T0L$ system by adding an extra component to the definition of a $T0L$ system. This additional component (called the target alphabet) is the subset of a total alphabet of a system. The resulting construct is called an extended $T0L$ system (or $ET0L$ system for short). The only difference in defining languages by $T0L$ and $ET0L$ systems is that the language of a given $ET0L$ system is defined as the set of all strings *over the target alphabet* which can be derived from the axiom.

$ET0L$ systems and languages (languages defined by $ET0L$ systems) form a natural extension of $T0L$ systems and languages from three points of view.

First, the use of intersection of the set of all strings generated from an axiom with a "terminal" set to define a language is a standard device in formal language theory.

Second, comparing the original systems of Lindenmayer with grammars within the Chomsky framework (for example, comparing $0L$-systems with context-free grammars), we notice that the systems of Lindenmayer are different mainly in that they do not use "nonterminals," and at each step of a derivation in a given system one must rewrite all (rather than only one) symbols of a given string. Introducing symbols in the definition of a system which do not appear in the language of the system reduces the above-mentioned difference to one point only. One obtains systems in which our parallel rewriting process (all letters rewritten at each step) can be directly compared with the serial rewriting (one letter rewritten at each step) which is used in the grammars of Chomsky hierarchy.

Third, $ET0L$-languages are also useful from a biological point of view. This comes about as follows. When we make our observations of a particular organism and wish to describe it by strings of symbols we first of all associate a symbol to each particular cell. We divide the cells into a number of types, and we associate the same symbol with all the cells of the same type. It is possible that the development of the organism can be described by a $T0L$-system but the $T0L$-system which describes it uses a finer subdivision into types than we have decided upon. (This is often experimentally unavoidable.) In this case the language of the organism using our subdivision into types is a homomorphic image of a $T0L$-language. It will be shown that the homomorphic image of any $T0L$-language is an $ET0L$-language.

This paper is divided into six sections.

In the next section we define all necessary concepts and give some examples of $ET0L$-systems and languages.

In Section 3 we investigate some basic properties of $ET0L$-systems and languages. Among the properties under consideration in this section are the role of erasing productions in $ET0L$-systems and the effect of codings on $ET0L$-languages.

In Section 4 we are looking for subclasses of the class of $ET0L$-systems which are powerful enough to define the whole class of $ET0L$-languages, using, for example, a minimal number of tables or productions per letter. We also prove that for each $ET0L$-language there is an $ET0L$-system which generates the language in a way such that each string in the language derives itself and does not derive any other string.

In Section 5 we consider closure properties of the class of $ET0L$-languages. In particular we prove that the family of $ET0L$-languages forms a full abstract family of languages (see Ref. 7). This yields a number of interesting consequences and also provides a better link between this family of languages and some other families studied so far in formal language theory.

In Section 6 we compare different subfamilies of the family of $ET0L$ languages. Also, a characterization theorem for the class of context-free languages is given.

We assume the reader to be familiar with the basic theory of formal languages (e.g., in the scope of Ref. 17) and in Section 6 we assume some familiarity with the theory of abstract families of languages (e.g., in the scope of Ref. 7).

## 2. DEFINITIONS

In this section we shall give formal definitions and examples of the systems and languages investigated in this paper.

For the basic terms of formal language theory we shall use the notation and terminology from Ref. 17 (in Section 5 we shall also use some terminology from Ref. 7) with the following two additions.

(i) If $x$ is a word over some alphabet $V$, then $\mathrm{Min}(x)$ is defined as the set of all the letters from $V$ which occur in $x$.

(ii) Let $V$ and $\Delta$ be finite alphabets. A (partial) function from $V$ into $\Delta$ is called a *coding* (from $V$ into $\Delta$), and it is extended to words and languages over $V$ as follows:

(1) $f(\Lambda) = \Lambda$.

(2) If $x \in V^+$, $x = a_1 \cdots a_n$ for $a_1, \ldots, a_n \in V$, where $f(a_j)$ is defined for every $j \in \{1, \ldots, n\}$, then $f(x) = f(a_1) f(a_2) \cdots f(a_n)$.

(3) If $x \in V^+$, $x = a_1 \cdots a_n$ for $a_1, \ldots, a_n \in V$, where for some $j$ in $\{1, \ldots, n\}$, $f(a_j)$ is not defined, then $f(x)$ is not defined.

(4) If $L \subseteq V^*$, then $f(L) = \bigcup_{x \in L} f(L)$.

*Definition 1.* An *extended table L-system without interactions*, abbreviated as an *ET0L-system*, is defined as a four-tuple $G = \langle V, \mathscr{P}, \omega, \Sigma \rangle$ 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 $G$), $\mathscr{P} = \{P_1, \ldots, P_f\}$ for some $f \geqslant 1$, each element of which is a finite subset of $V \times V^*$. $\mathscr{P}$ satisfies the following (*completeness*) condition:

$$(\forall P)_{\mathscr{P}} (\forall a)_V (\exists \alpha)_{V^*} (\langle a, \alpha \rangle \in P)$$

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

(4) $\Sigma \subseteq V$ (called the *target alphabet* of $G$).

We assume that $V$, $\Sigma$, and each $P$ in $\mathscr{P}$ are nonempty sets. The elements of $V - \Sigma$ will be called *auxiliary symbols*.

*Definition 2.* An $ET0L$-system $G = \langle V, \mathscr{P}, \omega, \Sigma \rangle$ is called:

(1) *Propagating* if for each $P$ in $\mathscr{P}$ we have $P \subset V \times V^+$.

(2) *Deterministic* if for each $P$ in $\mathscr{P}$ and each $a$ in $V$ there exists exactly one $\alpha$ in $V^*$ such that $\langle a, \alpha \rangle \in P$.

(3) A *T0L-system* if $V = \Sigma$.

(4) An *E0L-system* if $\# \mathscr{P} = 1$.

(5) A *0L-system* if $V = \Sigma$ and $\# \mathscr{P} = 1$.

We shall use letters $P$ and $D$ to denote the propagating or deterministic restrictions respectively. Thus, for example, a $PET0L$-system will be an abbreviation for a "propagating $ET0L$ system."

*Definition 3.* Let $G = \langle V, \mathscr{P}, \omega, \Sigma \rangle$ be an $ET0L$-system. Let $x \in V^+$, $x = a_1 \cdots a_k$, 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 $G(x \Rightarrow_G y)$ if, and only if, there exist $P$ in $\mathscr{P}$ and $p_1, \ldots, p_k$ in $P$ such that $p_1 = \langle a_1, \alpha_1 \rangle$, $p_2 = \langle a_2, \alpha_2 \rangle, \ldots, p_k = \langle a_k, \alpha_k \rangle$ (for some $\alpha_1, \ldots, \alpha_k \in V^*$), and $y = \alpha_1 \cdots \alpha_k$. We say that $x$ *derives* $y$ in $G(x \overset{*}{\Rightarrow}_G y)$ 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 $x_0 = x$, $x_n = y$, and $x_0 \Rightarrow_G x_1 \Rightarrow_G \cdots \Rightarrow_G x_n$; or (ii) $x = y$.

Here $\Rightarrow_G^*$ simply denotes the transitive and reflexive closure of the relation $\Rightarrow_G$. $\Rightarrow_G^+$ shall denote the transitive closure of $\Rightarrow_G$.

*Definition 4.* Let $G = \langle V, \mathscr{P}, \omega, \Sigma \rangle$ be an $ET0L$-system. The *language of* $G$ [denoted as $\mathscr{L}(G)$] is defined as $\mathscr{L}(G) = \{x \in \Sigma^*: \omega \Rightarrow_G^* x\}$.

*Definition 5.* Let $\Sigma$ be a finite alphabet and $L \subseteq \Sigma^*$. $L$ is called an $ET0L$ ($T0L$, $0L$, $E0L$) language if, and only if, there exists an $ET0L$ ($T0L$, $0L$, $E0L$) system $G$ such that $\mathscr{L}(G) = L$.

For the purpose of this paper we shall call two $ET0L$ systems *equivalent* if their languages are identical.

*Notation.* Let $G = \langle V, \mathscr{P}, \omega, \Sigma \rangle$ be an $ET0L$-system. If $\langle a, \alpha \rangle$ is an element of some $P$ in $\mathscr{P}$, then we call it *a production* (for $a$ in $P$) and write $a \rightarrow \alpha \in P$, or $a \rightarrow_P \alpha$. $a \rightarrow a$ is called an *identity production* and $a \rightarrow \Lambda$ is called an *erasing production*. If $x \Rightarrow_G y$ "using" an element $P$ of $\mathscr{P}$, then sometimes we shall write $x \Rightarrow_P y$. By $\mathscr{D}(G)$ we shall denote the set of all derivations in $G$. This means the set of all sequences $(x_0, x_1, ..., x_n)$, $n \geqslant 1$, such that $x_0 = \omega$ and $x_j \Rightarrow_G x_{j+1}$, for $0 \leqslant j \leqslant n - 1$. Sometimes by a derivation we shall mean a sequence $(x_0, x_1, ..., x_n)$ together with the precise set of productions used in each derivation step but this will always be clear from the context and should not lead to confusion. If $\mathscr{P} = \{P_1, ..., P_f\}$, then we shall write sometimes $G = \langle V; P_1, ..., P_f; \omega, \Sigma \rangle$ or in the case $f = 1$, $G = \langle V, P_1, \omega, \Sigma \rangle$.

We end this section with some examples of $ET0L$-systems and languages.

*Example 1.*

$$\begin{aligned} G_1 = \langle &\{a, b, C, D\}; \{a \rightarrow a, b \rightarrow b, C \rightarrow aCb, D \rightarrow Da\}, \\ &\{a \rightarrow a, b \rightarrow b, C \rightarrow Cb, D \rightarrow D\}, \{a \rightarrow a, b \rightarrow b, C \rightarrow \Lambda, D \rightarrow \Lambda\}; \\ &CD, \{a, b\} \rangle \end{aligned}$$

is an $ET0L$-system which is not propagating.

$$\mathscr{L}(G_1) = \{a^n b^m a^n: \quad n \geqslant 0, \quad m \geqslant n\}$$

*Example 2.*

$$G_2 = \langle \{a, b\}; \{a \rightarrow a^2, b \rightarrow b\}, \{a \rightarrow a, a \rightarrow ab, a \rightarrow ba, b \rightarrow b\}; a, \{a, b\} \rangle$$

is a $PT0L$-system.

$$\begin{aligned} \mathscr{L}(G_2) = \{&x \in \{a, b\}^+: \\ &\text{the number of } a\text{'s in } x \text{ is } 2^n \text{ for some } n \geqslant 0\} \end{aligned}$$

*Example 3.*

$G_3 = \langle\{a, S, A, F\}, \{S \to a^3, S \to A, A \to A^2, A \to a, F \to F, a \to F\}, S, \{a\}\rangle$

is a $PE0L$-system.

$$\mathscr{L}(G_3) = \{a^3\} \cup \{a^{2^n}:\quad n \geqslant 0\}$$

*Example 4.*

$$G_4 = \langle\{a, b\}, \{a \to (ab)^2, b \to \Lambda\}, ab, \{a, b\}\rangle$$

is a $0L$-system which is not propagating.

$$\mathscr{L}(G_4) = \{(ab)^{2^n}:\quad n \geqslant 1\}$$

## 3. BASIC PROPERTIES OF *ET0L*-SYSTEMS

In this section we shall investigate some basic properties of $ET0L$ systems and languages. Among the properties under consideration are the role of erasing productions in $ET0L$-systems and effects of codings on $ET0L$-languages.

First we shall state two useful results, the simple proofs of which we leave to the reader.

*Lemma 1.* If $G = \langle V, \mathscr{P}, \omega, \Sigma\rangle$ is an $ET0L$ ($E0L$) system, then $\mathscr{L}(G) = \mathscr{L}(H) \cap \Sigma^*$, where the $T0L$ ($0L$) system $H$ is defined as $H = \langle V, \mathscr{P}, \omega, V\rangle$.

*Lemma 2.* There exists an algorithm which for every $ET0L$ ($E0L$) system $G$ constructs an equivalent $ET0L$ ($E0L$) system $H = \langle V, \mathscr{P}, \omega, \Sigma\rangle$ such that $\omega \in V - \Sigma$.

As a consequence of Lemma 2 we shall assume in the sequel (unless otherwise stated) that $ET0L$-systems *we shall deal with have axioms which are auxiliary symbols.*

Now we shall prove that it is decidable whether an arbitrary $ET0L$ system can generate the empty word.

*Lemma 3.* There exists an algorithm which for every $ET0L$-system $G$ decides whether or not $\Lambda \in \mathscr{L}(G)$.

*Proof.* Let $G = \langle V, \mathscr{P}, S, \Sigma\rangle$ be an $ET0L$-system. Let $\# V = p$ and $u = 2^p$. Let $\mathscr{A}_1, \mathscr{A}_2, \ldots$ be a sequence of families of subsets of $V$, defined recursively as follows:

(1) $W \in \mathscr{A}_1$ if, and only if, there exists a table $P$ in $\mathscr{P}$ such that $W = \{a\colon a \to_P \Lambda\}$.

(2) For $i \geqslant 1$, $W \in \mathscr{A}_{i+1}$ if, and only if, $W \in \mathscr{A}_i$ or there exist $P$ in $\mathscr{P}$ and $Z$ in $\mathscr{A}_i$ such that $W = \{a: a \rightarrow_P \alpha \text{ for some } \alpha \in Z^+\}$.

It is clear that for each $i \geqslant 1$, $\mathscr{A}_i$ is finite and its construction is effective.

Note the following properties of the sequence $\mathscr{A}_1, \mathscr{A}_2, \ldots$:

(i) $\mathscr{A}_i \subset \mathscr{A}_{i+1}$ and for every $j \geqslant 2^p$, $\mathscr{A}_j = \mathscr{A}_{j+1}$. In particular, $\mathscr{A}_i \subseteq \mathscr{A}_u$ for every $i \geqslant 1$.

We leave the easy proof of (i) to the reader.

(ii) If $W \in \mathscr{A}_i$, for some $i \geqslant 1$, then for every $x \in W^+$, $x \Rightarrow^k \Lambda$ for some $k \leqslant i$.

*Proof of* (ii). We shall prove (ii) by induction on $i$.

(ii.1) The result is obvious for $i = 1$.

(ii.2) Let us assume that the result holds for all $j \leqslant i$.

(ii.3) Let $W \in \mathscr{A}_{i+1}$ and $x \in W^+$. We have to consider two cases: (ii.3.1) If $W \in \mathscr{A}_i$, then the result holds by the inductive assumption. (ii.3.2) If $W \notin \mathscr{A}_i$, then there exist $P$ in $\mathscr{P}$ and $Z$ in $\mathscr{A}_i$ such that $W = \{a: a \rightarrow_P \alpha \text{ for some } \alpha \in Z^+\}$. Thus if $x \in W^+$, then $x \Rightarrow_P y$ for some $y \in Z^+$. But by the inductive assumption $y \Rightarrow^l \Lambda$ for some $l \leqslant i$, and so $x \Rightarrow^{l+1} \Lambda$, where $l + 1 \leqslant i + 1$. This completes the induction on $i$. Hence (ii) holds.

(iii) If $x \in V^+$ and $x \Rightarrow^k \Lambda$, then there exists $W$ in $\mathscr{A}_k$ such that $\text{Min}(x) \subseteq W$.

*Proof of* (iii). We shall prove (iii) by induction on $k$.

(iii.1) The result is obvious for $k = 1$.

(iii.2) Let us assume that the result holds for all $j \leqslant k$.

(iii.3) Let $\text{Min}(x) = B$ and $x \Rightarrow^{k+1} \Lambda$. Thus $x \Rightarrow_P y \Rightarrow^k \Lambda$ for some $P$ in $\mathscr{P}$ and $y \in V^+$. Let $\text{Min}(y) = \bar{B}$. By the inductive assumption there exists $C$ in $\mathscr{A}_k$ such that $\bar{B} \subseteq C$. Let $W = \{a: a \rightarrow_P \alpha \text{ for some } \alpha \text{ in } C^+\}$. Then $B \subseteq W$ and, by definition, $W \in \mathscr{A}_{k+1}$. This completes the induction on $k$. Hence (iii) holds.

(iv) For any $x$ in $V^+$, $x \Rightarrow^+ \Lambda$ if, and only if, there exists $W$ in $\mathscr{A}_u$ such that $\text{Min}(x) \subseteq W$.

*Proof of* (iv). This result follows directly from (i)–(iii).

Now, Lemma 3 can be proved as follows: In order to determine whether $\Lambda \in \mathscr{L}(G)$, construct the sequence $\mathscr{A}_1, \ldots, \mathscr{A}_u$. From (iv) it follows that $\Lambda \in \mathscr{L}(G)$ if, and only if, there exists $W$ in $\mathscr{A}_u$ such that $S \in W$.

Thus Lemma 3 holds.

Next, we shall prove that if $L$ is an $ET0L$-language, then so is $L \cup \{\Lambda\}$.

*Lemma 4.* There exists an algorithm which for every $ET0L$ ($E0L$) system $G$ constructs an $ET0L$ ($E0L$) system $H$ such that $\mathscr{L}(H) = \mathscr{L}(G) \cup \{\Lambda\}$.

*Proof.* Let $G = \langle V, \mathscr{P}, S, \Sigma \rangle$ be an $ET0L$ ($E0L$) system. If $\Lambda \in \mathscr{L}(G)$, then $H = G$ satisfies the condition of the Lemma.

Otherwise let $H = \langle V \cup \{\bar{S}\}, \bar{\mathscr{P}}, \bar{S}, \Sigma \rangle$ be an $ET0L$ ($E0L$) system such that $\bar{S} \notin V$ and $\bar{\mathscr{P}} = \{\bar{P}: P \in \mathscr{P}\}$, where for $P \in \mathscr{P}$, $\bar{P}$ is defined as $P \cup \{\bar{S} \to \Lambda, \bar{S} \to S\}$. It is obvious that $\mathscr{L}(H) = \mathscr{L}(G) \cup \{\Lambda\}$.

Thus Lemma 4 holds.

*Remark.* Note that the above result is not true if one considers $0L$ systems only. For example, one may easily prove that $\{a^{2^n}: n \geqslant 0\}$ is a $0L$ language, whereas $\{a^{2^n}: n \geqslant 0\} \cup \{\Lambda\}$ is not.

We now investigate the effects of (partial) codings on $ET0L$ languages.

*Lemma 5.* There exists an algorithm which for every $ET0L$ ($E0L$) system $G$ and every coding $h$ produces an $ET0L$ ($E0L$) system $H$ such that $\mathscr{L}(H) = h(\mathscr{L}(G))$.

*Proof.* We shall consider separately cases of $ET0L$- and $E0L$-systems.

(i) Let $G = \langle V, \mathscr{P}, S, \Sigma \rangle$ be an $ET0L$-system and $h$ be a coding from $\Sigma$ into some alphabet $\Delta$ (we may obviously assume that $V \cap \Delta = \varnothing$).

Let $H = \langle V \cup \Delta \cup \{F\}, \bar{\mathscr{P}}, S, \Delta \rangle$ be an $ET0L$-system such that $F \notin V \cup \Delta$ and $\bar{\mathscr{P}} = P_0 \cup \{\bar{P}: P \in \mathscr{P}\}$, where we have the following:

(1) $P_0 = \{a \to h(a):\ a \in \Sigma \text{ and } h(a) \text{ is defined}\}$

$\cup \{a \to F: (a \in \Sigma \text{ and } h(a) \text{ is not defined}) \text{ or } (a \notin \Sigma)\}$

(2) For $P \in \mathscr{P}$, $\bar{P} = P \cup \{a \to F: a \notin V\}$.

We leave to the reader the easy proof of the fact that $\mathscr{L}(H) = h(\mathscr{L}(G))$.

(ii) Let $G = \langle V, P, S, \Sigma \rangle$ be an $E0L$-system and $h$ be a coding from $\Sigma$ into some alphabet $\Delta$.

We shall use the following useful result, which was proved in Ref. 14, Theorem 7: There exists an algorithm which for every $E0L$-system $G_1$ will produce a $PE0L$-system $G_2$ such that $\mathscr{L}(G_2) = \mathscr{L}(G_1) - \{\Lambda\}$.

Thus we may construct first a $PE0L$-system $\bar{G} = \langle \bar{V}, \bar{P}, \bar{S}, \Sigma \rangle$ (we assume that $\Delta \cap \bar{V} = \varnothing$) such that $\mathscr{L}(\bar{G}) = \mathscr{L}(G) - \{\Lambda\}$.

Then we define an $E0L$-system $H = \langle V_1, P_1, S_1, \Sigma \rangle$ as follows:

(1) $V_1 = \bar{V} \cup \Delta \cup \{F, S_1\}$, where $F_1 S_1 \notin \bar{V} \cup \Delta$.

(2) $P_1 = P \cup \{a \to h(a):\ a \in \Sigma \text{ and } h(a) \text{ is defined}\}$

$\cup \{a \to F:\ a \in \Delta \cup \{F\}\} \cup \{S_1 \to \bar{S}\} \cup X$

where

$$X = \begin{cases} \varnothing & \text{if } \Lambda \notin \mathscr{L}(G) \\ \{S_1 \to \Lambda\} & \text{otherwise} \end{cases}$$

(Note that from Lemma 3 it follows that the construction of $P_1$ is effective.)

We leave to the reader the easy proof of the fact that $\mathscr{L}(H) = h(\mathscr{L}(G))$. Lemma 5 follows from (i) and (ii).

*Lemma 6.* There exists an algorithm which, given an $ET0L$ ($E0L$) system $G$, will produce a $T0L$ ($0L$) system $H$ and a coding $h$ such that $\mathscr{L}(G) = h(\mathscr{L}(H))$.

*Proof.* Let $G = \langle V, \mathscr{P}, S, \Sigma\rangle$ be an $ET0L$ ($E0L$) system.
Let $h$ be a coding from $V$ into $\Sigma$ such that for $a \in V$

$$h(a) = \begin{cases} a & \text{if } a \in \Sigma \\ \text{not defined} & \text{otherwise} \end{cases}$$

We leave to the reader the easy proof of the fact that if we set $H$ to be a $T0L$ ($0L$) system such that $H = \langle V, \mathscr{P}, S, V\rangle$, then $\mathscr{L}(G) = h(\mathscr{L}(H))$. Thus Lemma 6 holds.

We leave to the reader the easy proof of our next result.

*Theorem 1.* The following statements are equivalent:

(1) $L$ is an $ET0L$ ($E0L$) language.

(2) There exists a $T0L$ ($0L$) language $K$ and an alphabet such that $L = K \cap \Sigma^*$.

(3) There exists a $T0L$ ($0L$) language $K$ and a coding $h$ such that $L = h(K)$.

*Remark.* Note that from Lemmas 1, 5, and 6 (and their proofs) it follows that Theorem 1 is effective in the usual sense, meaning that, for example, given a $T0L$-system $G$ and coding $h$, one may effectively construct an $ET0L$-system $H$ such that $L(H) = h(\mathscr{L}(G))$.

## 4. UNIVERSAL SUBCLASSES OF THE CLASS OF *ET0L*-SYSTEMS

In this section we look for subclasses of the class of $ET0L$-systems powerful enough to generate the class of $ET0L$-languages.

First we shall prove that for each $ET0L$-system one may produce an equivalent one in which an erasing production (if any) can be applied in the first step of a derivation only.

*Theorem 2.* There exists an algorithm which for every $ET0L$ ($E0L$) system $G$ produces an equivalent system $H$ such that (1) If $\Lambda \notin \mathscr{L}(G)$, then $H$ is propagating. (2) If $\Lambda \in \mathscr{L}(G)$, then there exists only one table in $H$ containing an erasing production, the table contains exactly one erasing production, and this unique erasing production is the form $S \to \Lambda$, where $S$ is the axiom of $H$ and $S$ does not appear in the right-hand side of any production in any table of $H$.

*Proof.* (i) If $G$ is an $E0L$-system, then we can use Theorem 7 from Ref. 14 (which was quoted already in the proof of Lemma 5) to produce an $E0L$ system $G_1 = \langle V_1, P_1, S_1, \Sigma \rangle$ such that $\mathscr{L}(G_1) = \mathscr{L}(G) - \{\Lambda\}$.

Then we construct an $E0L$-system $H = \langle V_1 \cup \{S_2\}, P_2, S_2, \Sigma \rangle$ such that $S_2 \notin V_1$ and $P_2 = P_1 \cup \{S_2 \to S_1\} \cup X$, where

$$X = \begin{cases} \varnothing & \text{if } \Lambda \notin \mathscr{L}(G) \\ \{S_2 \to \Lambda\} & \text{otherwise} \end{cases}$$

The effectiveness of this construction follows from Lemma 3. But it is obvious that $\mathscr{L}(H) = \mathscr{L}(G)$ and $H$ satisfies conditions of Theorem 2, hence Theorem 2 holds in the case of an $E0L$-system.

(ii) Let $G = \langle V, \mathscr{P}, S, \Sigma \rangle$ be an $ET0L$-system.

In Ref. 24, Theorem 5 it was proved that there exists an algorithm which, given a $T0L$-system $\bar{G}$, will produce a $PT0L$-system $\bar{\bar{G}}$ and a coding $h$ such that $\mathscr{L}(\bar{G}) - \{\Lambda\} = h(\mathscr{L}(\bar{\bar{G}}))$.

Thus if $G_1 = \langle V, \mathscr{P}, S, V \rangle$, we may construct a $PT0L$-system $G_2$ and a coding $h$ such that $\mathscr{L}(G_1) - \{\Lambda\} = h(\mathscr{L}(G_2))$.

But then from Lemma 5 it follows that we can construct an $ET0L$-system $G_3 = \langle V_3, \mathscr{P}_3, S_3, \Sigma \rangle$ such that $\mathscr{L}(G_3) = h(\mathscr{L}(G_2))$.

Now let $H$ be an $ET0L$ system such that

$$H = \begin{cases} G_3 & \text{if } \Lambda \notin \mathscr{L}(G) \\ \langle V_3 \cup \{S_4\}, \mathscr{P}_4, S_4, \Sigma \rangle & \text{otherwise} \end{cases}$$

where $S_4 \notin V_3$ and $\mathscr{P}_4 = \{\bar{P} : P \in \mathscr{P}_3 - \{P_0\}\} \cup \{\bar{P}_0\}$, where $P_0$ is an arbitrary but fixed table from $\mathscr{P}_3$ and (1) for $P \in \mathscr{P} - \{P_0\}$, $\bar{P} = P \cup \{S_4 \to S_3\}$, and (2) $\bar{P}_0 = P_0 \cup \{S_4 \to \Lambda, S_4 \to S_3\}$.

Note that the effectiveness of the construction of $H$ follows from Lemma 3.

We leave to the reader the easy proof of the fact that

$$\mathscr{L}(H) = \begin{cases} \mathscr{L}(G_3) & \text{if } \Lambda \notin \mathscr{L}(G) \\ \mathscr{L}(G_3) \cup \{\Lambda\} & \text{otherwise} \end{cases}$$

But from the construction of the systems $G_1$ through $G_3$ and from Lemma 1 it follows that $\mathscr{L}(G) = \mathscr{L}(H)$.

Obviously $H$ satisfies the conditions of Theorem 2 and so this theorem holds also in the case of an $ET0L$-system.

Theorem 2 follows from (i) and (iii).

We shall prove now the existence of a normal form for $ET0L$ systems which will be of particular importance later.

*Definition 6.* An $ET0L$ system $G = \langle V, \mathscr{P}, \omega, \Sigma\rangle$ is said to be in *normal form* if:

(1) $\omega \in V - \Sigma$ and if $\omega$ appears in some table at the right-hand side of some production then this is an identity production.

(2) There exists a unique table $P_I$ in $\mathscr{P}$ (called the *initial table*) such that $\omega \rightarrow_{P_I} \alpha$ for some $\alpha \in (V - \Sigma)^*$, $\alpha \neq \omega$, and if $a \neq \omega$, then the only production for $a$ in $P_I$ is $a \rightarrow a$.

3) There exists a unique table $P_t$ in $\mathscr{P}$ (called the *terminal table*) different from $P_I$, such that if $a \in V - \Sigma - \{\omega\}$ and $a \rightarrow_{P_t} \alpha$, then $\alpha \in \Sigma \cup \{F\}$, where $F$ is a distinguished symbol in $V - \Sigma$ such that $F \rightarrow F$ is the only production for $F$ in every table of $G$. ($F$ is called the *rejection symbol.*)

(4) If $a \rightarrow_P \alpha$ for $a \in \Sigma$, $P \in \mathscr{P}$, then $\alpha = a$, and if $a \rightarrow_P \alpha$ for $a \in V - \Sigma$, $P \in \mathscr{P} - \{P_I, P_t\}$, then $\alpha \in (V - \Sigma)^+$.

*Theorem 3.* (Normal form theorem for $ET0L$-systems). There exists an algorithm which for every $ET0L$-system $G$ produces an equivalent $ET0L$-system $H$ such that $H$ is in normal form.

*Proof.* Let $G = \langle V, \mathscr{P}, \omega, \Sigma\rangle$ be an $ET0L$-system. By Theorem 2 we may assume that either $G$ is propagating or there exists exactly one table in $\mathscr{P}$ which contains an erasing production which is of the form $\omega \rightarrow \Lambda$.

Let $G' = \langle V \cup \{S\}, \mathscr{P}', S, \Sigma\rangle$ be an $ET0L$-system such that $S \notin V$, $\mathscr{P}' = \{P_0\} \cup \{P' \colon P \in \mathscr{P}\}$, where

$$P_0 = \begin{cases} \{S \rightarrow \omega, S \rightarrow \Lambda\} & \text{if } \Lambda \in \mathscr{L}(G) \\ \{S \rightarrow \omega\} & \text{otherwise} \end{cases}$$

and for $P \in \mathscr{P}$

$$P' = \begin{cases} \{S \rightarrow S\} \cup P & \text{if } P \text{ is propagating} \\ \{S \rightarrow S\} \cup (P - \{\omega \rightarrow \Lambda\}) & \text{otherwise} \end{cases}$$

It is obvious that $\mathscr{L}(G') = \mathscr{L}(G)$, and by Lemma 2 the construction of $G'$ is effective.

Let $V_1 = \{\bar{a}: a \in V \cup \{S\}\} \cup \{F\}$, where $F \notin V \cup \{S\}$. ($\bar{\Sigma}$ will denote the set $\{\bar{a}: a \in \Sigma\}$.)

Let $P_I = \{\bar{S} \to \bar{\alpha}: S \to_{P_0} \alpha\} \cup \{a \to a: aV_1 \cup \Sigma\}$, where if $\alpha = a_1 \cdots a_n$ for some $n \geqslant 1$, $a_1, \ldots, a_n \in V$, then $\bar{\alpha} = \bar{a}_1 \cdots \bar{\alpha}_n$ and if $\alpha = \Lambda$, then $\bar{\alpha} = \Lambda$.

If $P \in \mathscr{P}'$, then

$$\bar{P} = \{\bar{a} \to \bar{\alpha}: \quad a \to_P \alpha \quad \text{and} \quad \alpha \neq \Lambda\} \cup \{a \to a: \quad a \in \Sigma \cup \{F\}\}$$

Let

$$P_t = \{\bar{a} \to a: \quad a \in \Sigma\} \cup \{a \to a: \quad a \in \Sigma\} \cup \{a \to F: \quad a \notin \bar{\Sigma} \cup \Sigma\}$$

Finally, let $H = \langle V_1 \cup \Sigma, \bar{\mathscr{P}}, \bar{S}, \Sigma\rangle$ be an $ET0L$-system, where $\bar{\mathscr{P}} = \{P_I, P_t\} \cup \{\bar{P}: P \in \mathscr{P}'\}$.

It is obvious that $H$ satisfies all five conditions of Theorem 3 with $P_I$ being the initial table, $P_t$ the terminal table, and $F$ the rejection symbol.

We leave to the reader the straightforward proof of the fact that $\mathscr{L}(H) = \mathscr{L}(G')$. Thus $\mathscr{L}(H) = \mathscr{L}(G)$ and Theorem 3 holds.

$ET0L$-systems in normal form are important mainly because they satisfy the following result, the easy proof of which we leave to the reader.

*Lemma 7.* Let $G = \langle V, \mathscr{P}, S, \Sigma\rangle$ be an $ET0L$-system in normal form.

(1) If $x \in \mathscr{L}(G)$, then in each derivation of $x$ in $G$ the first table used is the initial table and the last table used is the terminal table. Furthermore, there exists a derivation of $x$ in $V$ such that both the initial and the terminal tables are used in this derivation exactly once.

(2) Let $x \in V^+$ and $(S, x_1, \ldots, x_n = x)$ be a derivation of $x$ in $V$. Let $i_0$ be the minimal element from $\{1, \ldots, n\}$ (if such exist) such that $x_{i_0}$ contains an occurrence of a symbol from $\Sigma$. Then for every $j \geqslant i_0$, $x_j = x_{i_0}$ and either $x_{i_0} \in \Sigma^+$ or $x_{i_0} \in (\Sigma \cup \{F\}^*\{F\}(\Sigma \cup \{F\})^*$ where $F$ is the rejection symbol of $G$.

Next we generalize to $ET0L$-systems the notion of a synchronization introduced in Ref. 11 for $E0L$-systems.

*Definition 7.* If $G = \langle V, \mathscr{P}, \omega, \Sigma\rangle$ is an $ET0L$-system, then it is called *synchronized* if for every $x$, $y$ such that $x \in V^*\Sigma V^*$, $y \in V^+$, and $x \Rightarrow_G^+ y$ we have $y \in V^*(V - \Sigma)\, V^*$.

*Theorem 4.* For every $ET0L$-system there exists an equivalent synchronized $ET0L$-system.

*Proof.* (outline). Let $G = \langle V, \mathscr{P}, S, \Sigma\rangle$ be an $ET0L$-system. By Theorem 3 we may assume that $G$ is in normal form. It is obvious that if we change each table in $G$ in such a way that each production $a \to a$, where

$a$ is in $\Sigma$, is replaced by a production $a \to F$, where $F$ is the rejecting symbol from $G$, then the $E0L$-system obtained in this way is synchronized.

Hence Theorem 4 follows now from Theorem 3.

Note the duality between $ET0L$-systems in normal form and synchronized $ET0L$-systems: If an $ET0L$-system $G = \langle V, \mathscr{P}, S, \Sigma\rangle$ is synchronized and $D = (x_1, ..., x_n)$ is a derivation in $G$ such that $x_{i_0} \in \Sigma^+$ for some $i_0 \in \{1,..., n\}$, then for every $j > i_0$, $x_j \in V^*(V - \Sigma)\, V^* \cup \{\Lambda\}$.

On the other hand, if an $ET0L$-system $G = \langle V, \mathscr{P}, \omega, \Sigma\rangle$ is in normal form and $D = (x_1, ..., x_n)$ is a derivation in $G$ such that $x_{i_0} \in \Sigma^+$ for some $i_0 \in \{1,..., n\}$, then for every $j > i_0$, $x_j \in \Sigma^+$.

It was proved in Ref. 24 that the minimal number of tables needed to define a language by a $T0L$-system is a good measure of complexity, giving rise to an infinite hierarchy of $T0L$-languages.

We shall prove now that this result does not carry over to $ET0L$-systems and languages. In fact, every $ET0L$-language may be defined by an $ET0L$-system having at most two tables.

*Theorem 5.* There exists an algorithm which for every $ET0L$-system $G$ constructs an equivalent $ET0L$-system $H = \langle V, \mathscr{P}, \omega, \Sigma\rangle$ such that $\#\mathscr{P} \leqslant 2$.

*Proof.* Let $G = \langle V, \mathscr{P}, S, \Sigma\rangle$ be an $ET0L$-system. Let $G_1$ be the $T0L$-system defined as $G_1 = \langle V, \mathscr{P}, S, V\rangle$.

It was proved in Ref. 24 that there exists an algorithm which for every $T0L$-system $\bar{G}$ constructs a $T0L$-system $\bar{\bar{G}}$ and a (total) coding $h$ such that the number of tables in $\bar{\bar{G}}$ is not greater than two and $\mathscr{L}(\bar{G}) = h(\mathscr{L}(\bar{\bar{G}}))$.

Hence we can construct a $T0L$-system $G_2$ and a coding $h$ such that $\mathscr{L}(G_1) = h(\mathscr{L}(G_2))$.

The rest of the proof follows now from Lemma 1, Lemma 5 and its proof, Theorem 1, and the remark following it.

In fact, the above result is the best one in the sense that one cannot reduce the number of tables needed to define an arbitrary $ET0L$ language to one. This is proved as our next result.

*Theorem 6.* There exists an $ET0L$-language such that every $ET0L$-system generating it contains at least two tables.

*Proof.* Let $G_2$ be the $T0L$-system defined in Example 2.

It was proved in Ref. 11, Theorem 4 that $\mathscr{L}(G_2)$ cannot be defined by an $E0L$-system. Thus $\mathscr{L}(G_2)$ is an example of an $ET0L$-language satisfying the conditions of Theorem 6.

Another natural measure of complexity of a $T0L$-language (in the framework of defining $T0L$-languages by $T0L$-systems) is the least number of productions for some symbol which must appear in some table in any $T0L$-

system generating the given language. It was proved in Ref. 24 that this is a good measure of complexity giving rise to an infinite hierarchy of $T0L$-languages.

We shall prove now that the situation is quite different for $ET0L$-languages and systems. Every $ET0L$-language may be defined by an $ET0L$-system in which for every letter no table specifies more than two productions.

First let us introduce some auxiliary notation. If $G = \langle V, \mathscr{P}, S, \Sigma \rangle$ is an $ET0L$-system, then (1) for $a \in V$, $P \in \mathscr{P}$, $s(a, P) = \#\{\alpha: a \rightarrow_P \alpha\}$, (2) for $a \in V$, $t(a) = \max_{P \in \mathscr{P}}\{s(a, P)\}$, (3) $\operatorname{Det} G = \max_{a \in V}\{t(a)\}$.

*Theorem 7.* There exists an algorithm which for every $ET0L$-system $G$ constructs an equivalent $ET0L$-system $H = \langle V, \mathscr{P}, \omega, \Sigma \rangle$ such that $\operatorname{Det} H \leqslant 2$.

*Proof.* Let $G = \langle V, \mathscr{P}, S, \Sigma \rangle$ be an $ET0L$-system. Let $G_1$ be the $T0L$-system defined as $G_1 = \langle V, \mathscr{P}, S, V \rangle$.

It was proved in Ref. 24, Theorem 5 that there exists an algorithm which for every $T0L$-system $\bar{G}$ constructs a $T0L$-system $\bar{\bar{G}}$ and a (total) coding $h$ such that $\operatorname{Det} \bar{\bar{G}} \leqslant 2$. Hence we can construct a $T0L$-system $G_2$ and a coding $h$ such that $\mathscr{L}(G_1) = h(\mathscr{L}(G_2))$.

The rest of the proof follows now from Lemma 1, Lemma 5 and its proof, Theorem 1, and the remark following it.

We leave as an important open problem to investigate whether every $ET0L$-language can be generated by a $DET0L$-system. Although we conjecture that this is not the case, we have not been able to prove it.

As for the subclasses of $ET0L$-systems which we investigate in this paper, the following result clarifies the picture.

*Theorem 8.* There exists a $0L$-language which cannot be generated by a $DE0L$-system.

*Proof.* Let $L = \{c\} \cup \{b^2, b^4\} \cup \{a^n: n \geqslant 1\}$.

$L$ is a $0L$-language, since it is generated by the $0L$-system

$$\langle \{a, b, c\}, \{a \rightarrow a, a \rightarrow a^2, b \rightarrow b, c \rightarrow b^2, c \rightarrow b^4, c \rightarrow a\}, c, \{a, b, c\} \rangle$$

We shall show now that the assumption that $L$ is a $DE0L$-language leads to a contradiction.

Let us assume that there exists a $DE0L$-system $G = \langle V, \mathscr{P}, S, \Sigma \rangle$ such that $\mathscr{L}(G) = L$. Then either $b^2 \Rightarrow_G^+ b^4$ or $b^4 \Rightarrow_G^+ b^2$. But then either $\mathscr{L}(G)$ then either $\mathscr{L}(G)$ contains infinitely many strings from $\{b\}^+$ or $\mathscr{L}(G)$ contains $b$, both cases leading to a contradiction.

Thus $L$ is not a $DE0L$-language and Theorem 8 holds.

*Remark.* Note that $L$ can be generated by the $DET0L$-system

$\langle\{a, b, c, A\}: \{a \rightarrow a, b \rightarrow b, c \rightarrow b^2, A \rightarrow Aa\}, \{a \rightarrow a, b \rightarrow b, c \rightarrow b^4, A \rightarrow \Lambda\}, \{a \rightarrow a, b \rightarrow b, c \rightarrow Aa, A \rightarrow A\}; c, \{a, b, c\}\rangle$

## 5. CLOSURE PROPERTIES OF THE CLASS OF *ET0L*-LANGUAGES

In this section we consider a number of operations on languages and we show that the class of $ET0L$-languages is closed with respect to all of them.

In particular we shall prove that the class of $ET0L$-languages forms a full abstract family of languages.[7] This result is important since it distinguishes the class of $ET0L$-languages as the first subfamily of developmental languages studied so far which forms an abstract family of languages. This result also gives a stronger link between developmental languages and other families of languages considered in formal language theory.

*Theorem 9.* There exists an algorithm which, given two (arbitrary) $ET0L$-systems $G_1$, $G_2$, will produce an $ET0L$-system $H$ such that $\mathscr{L}(H) = \mathscr{L}(G_1) \cup \mathscr{L}(G_2)$.

*Proof.* Let $G_1 = \langle V_1, \mathscr{P}_1, S_1, \Sigma_1\rangle$ and $G_2 = \langle V_2, \mathscr{P}_2, S_2, \Sigma_2\rangle$ be $ET0L$-systems. According to Theorem 4, we may assume that both $G_1$ and $G_2$ are in normal form. Also, without loss of generality, we may assume that $(V_1 - (\Sigma_1 \cap \Sigma_2)) \cap (V_2 - (\Sigma_1 \cap \Sigma_2)) = \varnothing$.

Let $S$ be a new symbol, $S \notin V_1 \cup V_2$. Let

$$P_0 = \{a \rightarrow a: \quad \text{for every } a \in V_1 \cup V_2\} \cup \{S \rightarrow S_1, S \rightarrow S_2\}$$

If $P \in \mathscr{P}_1$, then let $\bar{P} = P \cup \{a \rightarrow a\text{: for every } a \in V_2 \cup \{S\}\}$. If $P \in \mathscr{P}_2$, then let $\bar{P} = P \cup \{a \rightarrow a\text{: for every } a \in V_1 \cup \{S\}\}$.

Finally, let $H = \langle V_1 \cup V_2 \cup \{S\}, \mathscr{P}_3, S, \Sigma_1 \cup \Sigma_2\rangle$ be an $ET0L$-system such that $\mathscr{P}_3 = \{P_0\} \cup \{\bar{P}: P \in \mathscr{P}\}$.

We leave to the reader the easy proof of the fact that $\mathscr{L}(H) = \mathscr{L}(G_1) \cup \mathscr{L}(G_2)$. This completes the proof of Theorem 9.

*Theorem 10.* There exists an algorithm which, given an arbitrary $ET0L$-system $G$ and an arbitrary substitution $f$ into $ET0L$-languages, will produce an $ET0L$-system $H$ such that $\mathscr{L}(H) = f(\mathscr{L}(G))$.

*Proof.* Let $G = \langle V, \mathscr{P}, S, \Sigma\rangle$ be an $ET0L$-system such that $\mathscr{L}(G) = L$, and let $f$ be a substitution from $\Sigma$ into some alphabet $\Delta$, such that for each

$a$ in $\Sigma$, $f(a) = L_a$, where $L_a$ is an $ET0L$-language generated by an $ET0L$-system $G_a = \langle V_a, \mathscr{P}_a, S_a, \Sigma_a \rangle$. We assume that

$$V \cap \Big(\bigcup_{a\in\Sigma} V_a\Big) = \varnothing \qquad \text{and} \qquad \Big(V_a - \bigcup_{c\in\Sigma} \Sigma_c\Big) \cap \Big(V_b - \bigcup_{c\in\Sigma} \Sigma_c\Big) = \varnothing$$

for every $a$, $b$ in $\Sigma$ such that $a \neq b$. We also assume that the $ET0L$-systems $G$, $G_a$ for $a$ in $\Sigma$ are in normal form.

If $P$ is a table in $G$, then $\bar{P}$ is defined as follows: $a \to \alpha$ is in $\bar{P}$ if, and only if, (i) $a \in \bigcup_{b\in\Sigma} V_b$ and $\alpha = a$, or (ii) $a \in V$, $\alpha \in (V - \Sigma)^*$, and $a \to \alpha$ is in $P$, or (iii) $a \in V$, $\alpha = S_b$ for some $b$ in $\Sigma$, and $a \to b$ is in $P$.

If $P$ is a table in $G_b$ for some $b$ in $\Sigma$, then $\bar{P}$ is defined as follows: $a \to \alpha$ is in $\bar{P}$ if, and only if, (i) $a = \alpha = S_b$, or (ii) $a \to \alpha$ is in $P$, or (iii) $\alpha = a$ and $a \in V \cup \bigcup_{x\in\Sigma-\{b\}} V_x$.

Finally, let $H = \langle \bar{V}, \bar{\mathscr{P}}, \bar{S}, \bar{\Sigma} \rangle$ be an $ET0L$-system such that (1) $\bar{V} = V \cup \bigcup_{a\in\Sigma} V_a$, (2) $\bar{S} = S$, (3) $\bar{\Sigma} = \bigcup_{a\in\Sigma} \Sigma_a$, (4) $\bar{\mathscr{P}} = \{\bar{P} \in \mathscr{P} \cup \bigcup_{a\in\Sigma} \mathscr{P}_a\}$.

We shall outline now the proof that $\mathscr{L}(H) = f(\mathscr{L}(G))$.

(i) $f(\mathscr{L}(G)) \subseteq \mathscr{L}(H)$.

If $x \in f(\mathscr{L}(G))$, then for some $z$ in $\mathscr{L}(G)$, $x = f(z)$. Let $z = a_1 \cdots a_n$ for some $n \geqslant 1$, $a_1, \ldots, a_n \in \Sigma$ [if $z = \Lambda$, then $S \to_P \Lambda$ for some $P$ in $\mathscr{P}$, hence, by construction of $H$, $S \to_{\bar{P}} \Lambda$ for some $\bar{P}$ in $\bar{\mathscr{P}}$ and consequently $x = f(\Lambda) = \Lambda \in \mathscr{L}(H)$].

Since $G$ is in normal form, there exists (see Lemma 8) a derivation $(z_1, \ldots, z_m)$ of $z$ in $G$ where $z_1 = S$ and the terminal table of $G$ is applied only once (in deriving $z = z_m$ from $z_{m-1}$). Hence by the construction of $\bar{\mathscr{P}}$ there exists a derivation $(\bar{z}_1, \ldots, \bar{z}_m)$ in $H$ such that $\bar{z}_1 = S$ and $\bar{z}_m = S_{a_1} \cdots S_{a_n}$.

But by the construction of $H$, (1) each table in $\bar{\mathscr{P}}$ contains a production $S_a \to S_a$ for every $a$ in $\Sigma$, (2) if $a \in \bar{\Sigma}$ and $\bar{P} \in \bar{\mathscr{P}}$, then, if $a \to_{\bar{P}} \alpha$, then $\alpha = a$, (3) for every $a$ in $\Sigma$ and $P$ in $\mathscr{P}_a$ there exists a table $\bar{P}$ in $\bar{\mathscr{P}}$ such that $P \subseteq \bar{P}$.

Thus if $x = \alpha_1 \cdots \alpha_n$ for $\alpha_1 \in \mathscr{L}(G_{a_1}), \ldots, \alpha_n \in \mathscr{L}(G_{a_n})$, then

$$\begin{aligned} \bar{z}_m = S_{a_1} \cdots S_{a_n} &\overset{+}{\underset{H}{\Rightarrow}} \alpha_1 S_{a_2} \cdots S_{a_n} \\ &\overset{+}{\underset{H}{\Rightarrow}} \alpha_1 \alpha_2 S_{a_3} \cdots S_{a_n} \overset{+}{\underset{H}{\Rightarrow}} \cdots \overset{+}{\underset{H}{\Rightarrow}} \alpha_1 \cdots \alpha_n \end{aligned}$$

and consequently $\alpha_1 \cdots \alpha_n \in \mathscr{L}(H)$.

Hence $f(\mathscr{L}(G)) \subseteq \mathscr{L}(H)$.

(ii) $\mathscr{L}(H) \subseteq f(\mathscr{L}(G))$.

If $S \to_{\bar{P}} \Lambda$ for some $\bar{P}$ in $\bar{\mathscr{P}}$ and $x = \Lambda$, then from the construction of $H$ it follows that $S \to_P \Lambda$ for some $P$ in $\mathscr{P}$ and consequently $f(x) = \Lambda \in f(\mathscr{L}(G))$.

Thus let us assume that $S \to \Lambda$ is not in $\bar{P}$ for any $\bar{P}$ in $\bar{\mathscr{P}}$.

From the fact that the $ET0L$-systems $G$ and $G_a$ for $a \in \Sigma$ are in normal forms and from the construction of $H$ [see similar arguments in part (i) of this proof] it follows that there exists a derivation (of $x$ in $G$)

$$(x_1, \ldots, x_{m_0}, \ldots, x_{m_1}, \ldots, x_{m_2}, \ldots, x_{m_p})$$

where $1 < m_0 < \cdots < m_p$, $p \geqslant 1$, such that (1) $x_{m_0} = S_{a_1} \cdots S_{a_p}$ for some $a_1, \ldots, a_p$ in $\Sigma$ such that $a_1 \cdots a_p \in \mathscr{L}(G)$, and (2) for $1 \leqslant i \leqslant p$, $x_{m_i} = \alpha_1 \cdots \alpha_i S_{a_{i+1}} \cdots S_{a_p}$, where $\alpha_j \in \mathscr{L}(G_{a_j}) = f(a_j)$ for $1 \leqslant j \leqslant i$ and all the productions used in each step of the derivation $(x_{m_{i-1}}, \ldots, x_{m_i})$ are from the set $\mathscr{P}_{a_1} \cup \{S_a \to S_a : a \in \Sigma\}$.

Thus it easily follows that $x = f(a_1) \cdots f(a_p)$ for some $a_1 \cdots a_p \in \mathscr{L}(G)$ and so $x \in f(\mathscr{L}(G))$.

Hence $\mathscr{L}(H) \subseteq f(\mathscr{L}(G))$.

From (i) and (ii) it follows that $\mathscr{L}(H) = f(\mathscr{L}(G))$ and this completes the proof of Theorem 10.

*Theorem 11.* There exists an algorithm which, given arbitrary $ET0L$-systems $G_1$ and $G_2$, will produce an $ET0L$-system $H$ such that $\mathscr{L}(H) = \mathscr{L}(G_1)\,\mathscr{L}(G_2)$.

*Proof.* Let $G_1 = \langle V_1, \mathscr{P}_1, S_1, \Sigma_1 \rangle$ and $G_2 = \langle V_2, \mathscr{P}_2, S_2, \Sigma_2 \rangle$ be $ET0L$-systems. We assume that $(V_1 - (\Sigma_1 \cap \Sigma_2)) \cap (V_2 - (\Sigma_1 \cap \Sigma_2)) = \varnothing$ and $G_1$ and $G_2$ are in the normal form.

Let $P_0 = \{S \to S_1S_2\}\{a \to a$: for every $a \in V_1 \cup V_2\}$, where $S$ is a new symbol, $S \notin V_1 \cup V_2$. If $P \in \mathscr{P}_1$, then let $\bar{P} = P \cup \{a \to a$: for every $a \in V_2 \cup \{S\}\}$. If $P \in \mathscr{P}_2$, then let $\bar{P} = P \cup \{a \to a$: for every $a \in V_1 \cup \{S\}\}$.

Finally, let $G = \langle V, \mathscr{P}, S, \Sigma \rangle$ be an $ET0L$-system such that (1) $V = V_1 \cup V_2 \cup \{S\}$, (2) $\Sigma = \Sigma_1 \cup \Sigma_2$, and (3) $\mathscr{P} = \{P_0\} \cup \{\bar{P}: P \in \mathscr{P}_1 \cup \mathscr{P}_2\}$.

We leave to the reader the easy proof of the fact that $\mathscr{L}(G) = \mathscr{L}(G_1)\,\mathscr{L}(G_2)$. Thus Theorem 11 holds.

*Theorem 12.* There exists an algorithm which, given an arbitrary $ET0L$-system $G$, will produce an $ET0L$-system $H$ such that $\mathscr{L}(H) = (\mathscr{L}(G))^+$.

*Proof.* Let $G = \langle V, \mathscr{P}, S, \Sigma \rangle$ be an $ET0L$-system. We assume that $G$ is in normal form. Let $P_0 = \{\bar{S} \to \bar{S},\ \bar{S} \to \bar{S}\bar{S},\ \bar{S} \to S\} \cup \{a \to a: a \in V\}$, where $\bar{S}$ is a new symbol, $\bar{S} \notin V$. For every $P$ in $\mathscr{P}$ let $\bar{P} = \{\bar{S} \to \bar{S}\} \cup P$.

Finally, let $H = \langle V \cup \{\bar{S}\}, \bar{\mathscr{P}}, \bar{S}, \Sigma\rangle$ be an $ET0L$-system, where $\bar{\mathscr{P}} = \{P_0\} \cup \{\bar{P}: P \in \mathscr{P}\}$.

We leave to the reader the easy proof of the fact that $\mathscr{L}(H) = (\mathscr{L}(G))^+$. Thus Theorem 12 holds.

*Theorem 13.* There exists an algorithm which, given an $ET0L$-system $G$, will produce an $ET0L$-system $H$ such that $\mathscr{L}(H) = (\mathscr{L}(G))^*$.

*Proof.* This result follows directly from Theorem 12 and Lemma 3.

*Theorem 14.* There exists an algorithm which, given an $ET0L$-system $G$ and a finite automaton $A$, will produce an $ET0L$-system $H$ such that $\mathscr{L}(H) = \mathscr{L}(G) \cap \mathscr{L}(A)$.

*Proof.* Let $G = \langle V, \mathscr{P}, S, \Sigma\rangle$ be an $ET0L$-system and $A = \langle Q, U, \delta, q_0, F\rangle$ be a finite automaton. We shall assume that $G$ is in normal form. Let $V_1 = \{[q, a, \bar{q}]: q, \bar{q} \in Q, a \in V\}$ be a new alphabet and $D \notin V_1 \cup \Sigma \cup \{S\}$.

If $P_0$ is the initial table of $G$, then we define $T_0$ as follows:

$$\begin{aligned} T_0 = \{&a \to a : a \in V_1 \cup \Sigma \cup \{D\}\} \\ &\cup \{S \to X : X = \Lambda \text{ if } \Lambda \in \mathscr{L}(A) \text{ and } X = S \text{ otherwise}\} \\ &\cup \{S \to [q_0, a_1, q_{i_1}][q_{i_1}, a_2, q_{i_2}] \cdots [q_{i_{n-2}}, a_{n-1}, q_{i_{n-1}}][q_{i_{n-1}}, a_n, q]: \\ &\qquad S \xrightarrow[P_0]{} a_1 \cdots a_n, q \in F, q_{i_1}, \ldots, q_{i_{n-1}} \in Q\} \end{aligned}$$

Let

$$\begin{aligned} T_f = \{&a \to a: a \in \{D, S\} \cup \Sigma\} \cup \{[q, a, \bar{q}] \to a: \\ &a \in \Sigma \quad \text{and} \quad \bar{q} \in \delta(q, a)\} \cup \{[q, a, \bar{q}] \to D: \bar{q} \notin \delta(q, a)\} \end{aligned}$$

If $P$ is a table in $\mathscr{P}$ such that $P$ is neither initial nor terminal, then we define $\bar{P}$ as follows:

$$\begin{aligned} \bar{P} = \{&a \to a : a \in \{S, D\} \cup \Sigma\} \\ &\cup \{[q, a, \bar{q}] \to [q, a_1, q_{i_1}][q_{i_1}, a_2, q_{i_2}] \cdots [q_{i_{n-1}}, a_n, \bar{q}]: \\ &\qquad a \xrightarrow[P]{} a_1 \cdots a_n, q_{i_1}, \ldots, q_{i_{n-1}} \in Q\} \end{aligned}$$

Finally, let $H = \langle V_2, \bar{\mathscr{P}}, S, \Sigma\rangle$ be an $ET0L$-system such that (1) $V_2 = V_1 \cup \Sigma \cup \{D, S\}$ and (2) $\bar{\mathscr{P}} = \{T_0, T_f\} \cup \{\bar{P}: P \text{ is neither initial nor terminal table in } \mathscr{P}\}$.

Since the presented construction is rather standard in formal language theory (see, e.g., Ref. 6, Theorem 3.2.1), we leave to the reader the formal proof of the fact that $\mathscr{L}(H) = \mathscr{L}(G) \cap \mathscr{L}(A)$.

Thus Theorem 14 holds.

*Theorem 15.* There exists an algorithm which, given an $ET0L$-system $G$ and a generalized sequential machine $A$, will produce an $ET0L$-system $H$ such that $\mathscr{L}(H) = A(\mathscr{L}(G))$.

*Proof.* Theorem 15 follows from Theorem 10, Theorem 14, and a well-known result (see, e.g., Lemma 9.3 and its proof in Ref. 17) which says that if $C$ is the class of languages which is effectively closed under finite substitution and intersection with a regular set, then $C$ is effectively closed under gsm mappings.

*Theorem 16.* There is an algorithm which, given an $ET0L$-system $G$ and a generalized sequential machine $A$, will produce an $ET0L$-system $H$ such that $\mathscr{L}(H) = A^{-1}(\mathscr{L}(G))$.

*Proof.* Theorem 16 follows from Theorems 9, 10, 14, and 15 and a well-known result (see, e.g., Lemma 9.4 and its proof in Ref. 17) which says that if $C$ is the class of languages which is effectively closed under union, $\Lambda$-free substitution, $k$-limited erasing, and intersection with regular sets, then $C$ is effectively closed under inverse gsm mappings.

As a straightforward corollary from Theorems 9–16 we have the following results.

*Theorem 17.* The class of $ET0L$-languages is closed with respect to the following operations: (i) union, (ii) substitution, (iii) product, (iv) the cross operator, (v) the star operator, (vi) intersection with a regular set, (vii) gsm mapping, (viii) inverse gsm mapping.

An important corollary of Theorem 17 is the following result.

*Theorem 18.* The family of $ET0L$-languages forms a full abstract family of languages. This result is quite important for the following reasons.

(1) It puts the family of $ET0L$-languages in a better perspective.

(2) It distinguishes the family of $ET0L$-languages as the first subfamily of developmental languages which has been studied which is a full AFL (in fact none of the families of developmental languages studies so far was even a pre-AFL).

(3) It gives a stronger link between developmental languages and other families of languages studied in formal language theory.

(4) As a corollary of this result, we get the closure of the family of $ET0L$-languages with respect to quite a number of other operations.

As an illustration of point 4, we shall state some simple results which are corollaries from Theorem 18 and appropriate results from the theory of abstract families of languages.

*Corollary 1.* (follows from Theorem 18 and Theorem 2.1 in Ref. 7). The family of $ET0L$-languages is closed under arbitrary $a$-transducers.

*Corollary 2.* (follows from Theorem 18 and Corollary 2 in Ref. 7). If $L$ is an $ET0L$-language and $R$ is a regular language, then

$$L/R = \{w: \quad wy \in L \quad \text{for some } y \text{ in } R\}$$

and

$$R\backslash L = \{w: \quad yw \in L \quad \text{for some } y \text{ in } R\}$$

are both $ET0L$-languages.

*Corollary 3.* (from Theorem 18 and Corollary 3 in Ref. 7). If $L$ is an $ET0L$-language, then

$$\text{Init}(L) = \{w \neq \Lambda: \quad wy \in L \quad \text{for some } y\}$$
$$\text{Fin}(L) = \{w \neq \Lambda: \quad yw \in L \quad \text{for some } y\}$$

and

$$\text{Sub}(L) = \{w \neq \Lambda: \quad uwv \in L \quad \text{for some } u, v\}$$

are $ET0L$-languages.

## 6. INTERRELATIONS AMONG SOME FAMILIES OF LANGUAGES

In this section we shall compare the generative power of different subfamilies of the family of $ET0L$-languages. We shall also compare these families with the languages in the Chomsky hierarchy and with the context-free programmed languages of Rosenkrantz.[23]

Also, at the end of this section we give a necessary and sufficient condition for an $ET0L$-system to generate a context-free language.

Let us first introduce notation for different classes of languages to be considered.

| | |
|---|---|
| $\mathscr{R}EN$ | denotes the class of recursively enumerable languages |
| $\mathscr{C}S$ | denotes the class of context-sensitive languages |
| $\mathscr{P}$ | denotes the class of context-free programmed languages |
| $\bar{\mathscr{P}}$ | denotes the class of $\Lambda$-free context-free programmed languages |
| $\mathscr{C}F$ | denotes the class of context-free languages |
| $\mathscr{R}EG$ | denotes the class of regular languages |

| | |
|---|---|
| $\mathscr{E}T0L$ | denotes the class of $ET0L$ languages |
| $\mathscr{T}0L$ | denotes the class of $T0L$ languages |
| $\mathscr{E}0L$ | denotes the class of $E0L$ languages |
| $\mathscr{O}L$ | denotes the class of $0L$ languages |

For the notion of a ($\Lambda$-free) context-free programmed grammar and language we refer the reader to Ref. 23.

*Theorem 19.* The following diagram holds:

$$\begin{tikzcd}
 & REN=P & & \\
 & CS \arrow[u] & & \\
 & \overline{P} \arrow[u] & & \\
 & ET0L \arrow[u] & & \\
T0L \arrow[ur] & & E0L \arrow[ul] & \\
 & 0L \arrow[ul] \arrow[ur] & & CF \arrow[ul] \\
 & & & REG \arrow[u]
\end{tikzcd}$$

where a solid line denotes the strict inclusion (in the direction indicated) and when two classes are not connected by a directed path in this diagram it means that they are incomparable but not disjoint.

*Proof.*

(i) It was proved in Ref. 23, Theorems 4 and 6 that $\mathscr{P} = \mathscr{R}EN$ and $\overline{\mathscr{P}} \subset \mathscr{C}S$.

(ii) It was proved in Ref. 24, Theorem 6 that $\mathscr{T}0L \subseteq \overline{\mathscr{P}}$. In fact one can apply almost the same proof to show that $\mathscr{E}T0L \subseteq \overline{\mathscr{P}}$.

(iii) Using a standard method in formal language theory, one may easily prove (we leave this proof to the reader) that from $\mathscr{P} = \mathscr{R}EN$ and $\overline{\mathscr{P}} \subset \mathscr{C}S$ it follows that the class of $\Lambda$-free context-free programmed languages is not closed with respect to (erasing) homomorphic mappings.

Thus from (ii) and the fact that the class of $ET0L$-languages is closed with respect to an arbitrary homomorphism it follows that $\mathscr{E}T0L \subset \overline{\mathscr{P}}$.

(iv) By definition $\mathscr{O}L \subseteq \mathscr{T}0L \subseteq \mathscr{E}T0L$ and $\mathscr{O}L \subseteq \mathscr{E}0L \subseteq \mathscr{E}T0L$.

(v) It was proved in Ref. 24, Theorem 2 that $\{a^3\} \cup \{a^{2^n}: n \geqslant 0\}$ is not a $T0L$-language, whereas (see Example 3) it is an $E0L$-language.

(vi) It was proved in Ref. 11, Theorem 4 that the language $\{x \in \{a, b\}^+$: the number of $a$'s in $x$ is $2^n$ for some $n \geqslant 0\}$ is not an $E0L$-language, whereas (see Example 2) it is a $T0L$-language.

(vii) It is known (see, e.g., Ref. 32, Corollary 4.5) that $\mathscr{C}F \subset \mathscr{E}0L$ and it was proved in Ref. 24, Section 2(iv) that there exist regular languages which are not in $\mathscr{T}0L$.

(viii) The well-known fact that $\mathscr{R}EG \subset \mathscr{C}F$ completes the proof of Theorem 19.

Now we shall investigate the effect of identity productions in $ET0L$-systems.

*Lemma 8.* There exists an algorithm which, given an arbitrary context-free grammar $G$, will produce an $E0L$-system $H = \langle V, P, S, \Sigma\rangle$ such that $\mathscr{L}(H) = \mathscr{L}(G)$ and $a \rightarrow_P a$ for every $a$ in $V$.

*Proof.* This result follows from Theorem 1 and Theorem 4.2 in Ref. 24 which, together with its proof, says that there exists an algorithm which, given an arbitrary context-free grammar $G = \langle V_1, T_1, P_1, S_1\rangle$, will produce a $0L$-system $H = \langle V_2, P_2, S_2, V_2\rangle$ such that $a \rightarrow_{P_2} a$ for every $a$ in $V_2$ and $\mathscr{L}(G) = \mathscr{L}(H) \cap T_1{}^*$.

*Lemma 9.* There exists an algorithm which, given an $ET0L$-system $G = \langle V, \mathscr{P}, S, \Sigma\rangle$ such that $a \rightarrow_P a$ for every $a$ in $V$ and every $P$ in $\mathscr{P}$, wil produce a context-free grammar $H$ such that $\mathscr{L}(G) = \mathscr{L}(H)$.

*Proof.* Let $G = \langle V, \mathscr{P}, S, \Sigma\rangle$ be an $ET0L$-system such that $a \rightarrow_P a$ for every $a$ in $V$ and every $P$ in $\mathscr{P}$. Let $V_1 = \{\bar{a}: a \in V\}$ and if $\alpha \in V^+$, $\alpha = a_1 \cdots a_n$ for $a_i \in V$ for $1 \leqslant i \leqslant n$, then $\bar{\alpha} = \bar{a}_1 \cdots \bar{a}_n$ (also $\bar{\Lambda} = \Lambda$). Let $H = \langle V_1, \Sigma, R, S\rangle$ be a context-free grammar such that

$$R = \{\bar{a} \rightarrow \bar{\alpha}: a \rightarrow_P \alpha\} \cup \{\bar{a} \rightarrow a: a \in \Sigma\}.$$

The proof of the fact that $\mathscr{L}(H) = \mathscr{L}(G)$ may be done similarly to the proof of Theorem 4.2 in Ref. 24, and so we leave it to the reader.

Thus Lemma 9 holds.

*Lemma 10.* If $G = \langle V, P, S, \Sigma\rangle$ is an $E0L$-system such that $a \rightarrow_P a$ for every $a$ in $\Sigma$, and $H = \langle V, \bar{P}, S, \Sigma\rangle$ is an $ET0L$-system such that $\bar{P} = P \cup \{a \rightarrow a: a \in V - \Sigma\}$, then $\mathscr{L}(G) = \mathscr{L}(H)$.

*Proof.* Let $G$ and $H$ satisfy conditions of Lemma 10.

Obviously $\mathscr{L}(G) \subseteq \mathscr{L}(H)$ and so it is enough to prove that $\mathscr{L}(H) \subseteq \mathscr{L}(G)$.

To this aim, we shall prove the following claim: For $a \in V$, $\alpha \in \Sigma^*$, $k \geqslant 1$, if $a \Rightarrow_H^k \alpha$, then $a \Rightarrow_G^k \alpha$.

The proof goes by induction on $k$ as follows.

(1) $k = 1$. If $a \Rightarrow_H \alpha$ and $a \in \Sigma^*$, then obviously $a \rightarrow_P \alpha$ and so $a \Rightarrow_G \alpha$.

(2) Let us assume that the claim holds for all $l \leqslant k$.

(3) If $a \Rightarrow_H^{k+1} \alpha$, then $a \Rightarrow_H \beta \Rightarrow_H^k \alpha$ for some $\beta \in V^+$.

If $a \rightarrow \beta$ is in $P$, then $a \Rightarrow_G \beta$, which together with the inductive hypothesis implies that $a \Rightarrow_G \beta \Rightarrow_G^k \alpha$.

If $a \rightarrow \beta$ is not in $P$, then (by the construction of $H$) $a \in V - \Sigma$ and $\beta = a$. Hence $a \Rightarrow_H a \Rightarrow_H^k \alpha$. But then by the inductive hypothesis $a \Rightarrow_G^k \alpha$ and (because $b \rightarrow_P b$ for every $b$ in $\Sigma$) $\alpha \Rightarrow_G \alpha$. Thus $a \Rightarrow_G^k \alpha \Rightarrow_G \alpha$ and so $a \Rightarrow_G^{k+1} \alpha$.

This completes the proof of our claim.

In particular, from the claim it follows that for every $\alpha$ in $\Sigma^*$ if $S \Rightarrow_H^+ \alpha$, then $S \Rightarrow_G^+ \alpha$ and so $\mathscr{L}(H) \subseteq \mathscr{L}(G)$, which completes the proof of Lemma 10.

*Remark.* It is interesting to note that the above result is not true in the case when one considers *ET0L*- rather than *E0L*-systems. Thus, for example,

$$G = \langle \{A, a\}; \{A \rightarrow A^2, a \rightarrow a\}, \{A \rightarrow a, a \rightarrow a\}; A, \{a\} \rangle$$

is an *ET0L*-system containing two tables only. If we augment either of the tables of $G$ by the production $A \rightarrow A$, then the new *ET0L*-system generates the language $\{a\}^+$ whereas $\mathscr{L}(G) = \{a^{2^n} : n \geqslant 0\}$.

Finally we have the following characterization of context-free languages.

*Theorem 20.* A language is context-free if, and only if, it can be generated by an *E0L*-system $G$ where $a \rightarrow a$ is a production for every letter $a$ in the target alphabet of $G$.

*Proof.* This result follows directly from Lemmas 8–10.

*Remark.* Note that the above result does not hold for *ET0L*-systems. To the contrary, every *ET0L*-language can be generated by an *ET0L*-system in normal form which is of such a nature that a production $a \rightarrow a$ is included in every table for every target symbol $a$. Still (see Theorem 19) the class of context-free languages is strictly included in the class of *ET0L*-languages.

## ACKNOWLEDGMENTS

The author is grateful to Dr. G. T. Herman and Messrs. K. P. Lee and A. Walker for corrections to the original manuscript.

## REFERENCES

1. R. Baker and G. T. Herman, "Simulation of organisms using a developmental model, Parts I and II," *Int. J. Bio-Med. Comp.*, to appear.
2. E. F. Codd, *Cellular Automata* (Academic Press, New York, 1968).
3. D. van Dalen, "A note on some systems of Lindenmayer," *Math. Systems Theory* **5**:128–140 (1971).
4. P. Doucet, "On the membership question in some Lindenmayer systems," *Indag. Math.* **34**:45–52 (1972).
5. H. Feliciangeli and G. T. Herman, "Algorithms for producing grammars from sample derivations," *J. Comp. Syst. Sci.*, to appear.
6. S. Ginsburg, *The Mathematical Theory of Context-Free Languages* (McGraw-Hill, New York, 1966).
7. S. Ginsburg and S. Greibach, "Abstract families of languages," *Mem. Am. Math. Soc.* **87**:1–32 (1969).
8. G. T. Herman, "The computing ability of a developmental model for filamentous organisms," *J. Theoret. Biol.* **25**:421–435 (1969).
9. G. T. Herman, "The role of environment in developmental models," *J. Theoret. Biol.* **29**:329–341 (1970.
10. G. T. Herman, "Models for cellular interactions in development without polarity of individual cells, Parts I and II." *Int. J. Systems Sci.* **2**:271–289 (1971); **3**:149–175 (1972).
11. G. T. Herman, "Closure properties of families of languages associated with biological systems," submitted to a technical journal, abstract in *Proc. 5th Annual Princeton Conf. Inf. Sciences Syst.*, 1971.
12. G. T. Herman, "Polar organisms with apolar individual cells," in *Proc. Int. Congr. on Logic, Math. and Phil. of Science*, 1971.
13. G. T. Herman, "The syntactic inference problem as applied to biological systems," to appear in *Machine Intelligence* 7.
14. G. T. Herman, "A biologically motivated extension of Algol-like languages," to appear in *Information and Control.*
15. G. T. Herman, K. P. Lee, J. van Leeuwen, and G. Rozenberg, "Characterization of unary developmental languages," to appear in *Discrete Mathematics.*
16. G. T. Herman, A. Lindenmayer, and G. Rozenberg, "Description of developmental systems using recurrence systems," submitted to a technical journal.
17. J. E. Hopcroft and J. D. Ullman, *Formal Languages and Their Relation to Automata* (Addison-Wesley, Reading, Mass., 1969).
18. A. Lindenmayer, "Mathematical models for cellular interactions in development, Parts I and II," *J. Theoret. Biol.* **18**:280–315 (1968).
19. A. Lindenmayer, "Developmental systems without cellular interactions, their languages and grammars," *J. Theoret. Biol.* **30**:455–484 (1971).
20. A. Lindenmayer, "Cellular automata, formal languages and developmental systems," in *Proc. Int. Congr. on Logic, Math. and Phil. of Science*, 1971.
21. A. Lindenmayer and G. Rozenberg, "Developmental systems and languages," in *Proc. 4th ACM Symp. Theory Comp.* (1972), pp. 214–221.

22. A. Paz and A. Salomaa, "Integral sequential word functions and growth equivalence of Lindenmayer systems," submitted to a technical journal.
23. D. J. Rosenkrantz, "Programmed grammars and classes of formal languages," *J. Assoc. Comp. Mach.* **16**:107–131 (1969).
24. G. Rozenberg, "*T0L* systems and languages," to appear in *Information and Control.*
25. G. Rozenberg, "On 0*L* languages with restricted use of productions," to appear in *J. Comp. Syst. Sci.*
26. G. Rozenberg, "The equivalence problem for deterministic *T0L*-systems is undecidable," *Inf. Processing Letters,* **1**:201–204 (1972).
27. G. Rozenberg, "*L*-systems with interactions," to appear in *J. Comp. Syst. Sci.*
28. G. Rozenberg, "*D0L* sequences," submitted to a technical journal.
29. G. Rozenberg, "Circularities in *D0L* sequences," to appear in *Revue Roum. de Math. Pures et Appl.*
30. G. Rozenberg, "Direct proofs of the unsolvability of the equivalence problem for sentential forms of context-free grammars and the equivalence problem for 0*L* systems," to appear in *Inf. Processing Letters.*
31. G. Rozenberg, "On a machine model for *L*-systems without interactions," submitted to a technical journal.
32. G. Rozenberg and P. Doucet, "On 0*L* languages," *Information and Control* **19**:302–318 (1971).
33. G. Rozenberg and K. P. Lee, "Some properties of the class of *L*-languages with interactions," submitted to a technical journal.
34. G. Rozenberg and K. P. Lee, "Developmental systems with finite axiom sets, Parts I and II," submitted to a technical journal.
35. G. Rozenberg and A. Lindenmayer, "Developmental systems with locally catenative formulas," submitted to a technical journal.
36. V. Surapipith and A. Lindenmayer, "Thioquanine-dependent light sensitivity of Perithecial initiation in Sordia fimicola," *J. Gen. Microb.* **57**:227–237 (1969).

*Printed in Belgium*