GENERATIVELY DETERMINISTIC L LANGUAGES. SUBWORD POINT OF VIEW

A. EHRENFEUCHT
Department of Computer Science, University of Colorado, at Boulder,
K.P. LEE [1)]
Department of Computer Science, S.U.N.Y. at Buffalo,
and
G. ROZENBERG
Institute of Mathematics, Utrecht University, and
Department of Mathematics, Antwerp University, UIA

INTRODUCTION

The notion of a "deterministic machine" or a "deterministic language" (as opposed to their nondeterministic counterparts) is one of the oldest and most investigated in the theory of computation and in formal language theory. One can however observe that whereas the notion of a deterministic machine is usually the natural one (in every situation there is at most one possible "move" the machine can make), the notion of a deterministic language is often not natural at all. In fact a deterministic language is almost always defined as a language which can be recognized by a deterministic machine, although in many cases the languages themselves are being defined by grammars rather than by machines. The typical situation is of the following kind: first a class of languages $\mathcal{L}$ is defined by a class of grammars $\mathcal{G}$, then one finds an "equivalent" class of machines $\mathcal{M}$, and then by considering the deterministic subclass $\mathcal{M}_D$ of the class $\mathcal{M}$ one obtains the deterministic subclass $\mathcal{L}_D$ of the class $\mathcal{L}$. What subclass of $\mathcal{G}$ generates $\mathcal{L}_D$ is mostly not understood at all, or, in the best case, it is the "translation" of $\mathcal{M}_D$ into the subclass of $\mathcal{G}$, which could neither be called natural nor give any insight into the nature of the deterministic restriction. The basic difficulty lies in the fact that the notion of a deterministic language is defined via recognizers whereas the languages themselves are often defined in terms of generative devices.

In this paper we want to point out several classes of languages for which the notion of "generative determinism" (deterministic restriction defined in terms of grammars rather than recognizers) is not only a very natural one but it also lends itself to mathamatical treatment.

The theory of L systems and languages originated with the work of

1) This paper is based on part of this author's Ph.D. thesis.

Lindenmayer [59], [60]. Its purpose was to model the growth of filamentous organisms. From the formal language theory point of view, L systems are string rewriting systems. They have provided us with an alternative to the now standard Chomsky framework for defining languages. Basically L systems differ from Chomsky grammars in the lack of nondeterminals and in the totally parallel manner of rewriting (meaning that in a single derivation step one must rewrite all occurrences of all the symbols in the string being rewritten). For more detailed discussion see, for example, [45] or [66]. In the theory of L systems the deterministic restriction arose for a number of natural and "practical" reasons. Its investigation has led to novel fields like growth functions (see [75] and its references) and to new research on rather established topics like the deterministic simulation of one kind of system by another (see, for example, [12]). This paper continues the study of the role determinism plays in various classes of L systems.

A possible division line in the theory of L systems is the distinction between systems without interactions and systems with interactions. Accordingly the present paper is divided into two parts. In the first part we treat systems without interactions, while the second part is concerned with systems with interactions.

PART I
L systems without interactions

In an L system without interactions, the rewriting of a letter in a string does not depend on the context in which the letter occurs (in other words, each occurrence of the same letter may be rewritten in the same way).

I.1 TOL systems and languages.

TOL systems and languages were divised to model special cases of development in which no cell interaction takes place but there is a finite number of possible environments. In different environments, the behaviour of the same cell may be different. TOL systems were introduced in [81]. Their formal definitions and basic properties can be found there. (TOL systems, or languages, abbreviates "table L systems, or languages, without interactions".)

A TOL system has the following components[2]:

(i) A finite set of symbols Σ, the alphabet.

(ii) A finite set $\mathcal{P}$ of tables of productions. Each production in a table is usually written in the form $a \to \alpha$, where $a \in \Sigma$ and $\alpha \in \Sigma^*$. The meaning of $a \to \alpha$ is that an occurrence of the letter a in a string may be replaced by α (where each replacement is "context-free"). In general, a table may contain several productions for each symbol. In every step of a derivation, all symbols in a string must be simultaneously replaced according to the production rules of one arbitrarily chosen table.

(iii) A starting string, σ, the axiom.

Thus a TOL system G is usually specified as $G = <\Sigma,\mathcal{P},\sigma>$. The language generated by G, denoted as L(G), consists of σ and all strings which can be derived from σ in a finite number of steps. A language L is called a TOL language if there exists a TOL system G such that L = L(G).

Example 1. Let $G = <\{a,b\}, \{\{a \to a^2,\ b \to b^2\}, \{a \to a^3,\ b \to b^3\}\}, ab>$. Then

$$L(G) = \{a^{2^n 3^m} b^{2^n 3^m} \mid n,m \geq 0\}.$$

I.2. Deterministic TOL languages. A limit theorem.

If we view TOL systems as models of development, then each table of the system represents a particular environment. A TOL system is called deterministic if in each environment there is only one choice for the next developmental step. This means that each next string in a derivation starting from the axiom is uniquely determined by the previous one and the table applied.

Formally the deterministic restriction is defined as follows.

Definition 1. A TOL system $G = <\Sigma,\mathcal{P},\sigma>$ is called deterministic if, for each P in $\mathcal{P}$ and each a in Σ, there exists exactly one α in Σ^* such that $a \to \alpha$ is in P. A language L is called a deterministic TOL language if there exists a deterministic TOL system G such that L = L(G).

It is not difficult to construct examples of languages which can

---

2) Throughout this paper we shall use standard formal language notation, as for example in (Hopcroft & Ullman, Formal Languages and their Relation to Automata, Addison-Wesley, 1969). We use $|x|$ for the length of a string x and $\#A$ for the cardinality of a set A. The empty string is denoted by the symbol Λ. If we write that L is a language over an alphabet Σ, or just $L \subseteq \Sigma^*$, then we also mean that each letter of Σ occurs in a word of L.

be generated by a nondeterministic TOL system but cannot be generated by deterministic TOL systems. One would like however to find a non-trivial (and hopefully interesting) property which would be inherent to the class of deterministic TOL languages.

Investigating the set of words generated by a particular grammar is one of the most basic activities in formal language theory. It is however often interesting and well motivated physically to investigate the set of all subwords (subpatterns) generated by a particular grammar. Quite often one is interested in just the number of different subwords of a particular length encountered in a given language.

It turns out that the ability to generate an arbitrary number of subwords of an arbitrary length is a property of a TOL system which disappears when the deterministic restriction is introduced. More precisely, if L is a deterministic TOL language over an alphabet containing at least two letters, then the ratio of the number of different subwords of a given length k occurring in the words of L to the number of all possible words of length k tends to zero as k increases. Formally this is stated as follows.

Theorem 1. Let Σ be a finite alphabet such that $\#\Sigma = n \geq 2$. If L is a deterministic TOL language, $L \subseteq \Sigma^*$, then

$$\lim_{k\to\infty} \frac{\pi_k(L)}{n^k} = 0,$$

where $\pi_k(L)$ denotes the number of all subwords of length k occurring in the words of L.

The proof of Theorem 1 appears in [17].
All other results presented in this paper have not yet been published before.

Note that this result is not true if $\#\Sigma = 1$ (the language $\{a^{2^n} \mid n \geq 1\}$ is a deterministic TOL language). Neither is it true for nondeterministic TOL languages (Σ^* is a TOL language for every alphabet Σ).

We believe that the above results is a fundamental one for characterizing deterministic TOL languages. It can be used, for example, in both intuitive and formal proofs that some languages are not deterministic TOL languages (an example of such an application is a proof that if $\Sigma = \{a,b\}$ and F is a finite language over Σ, then Σ^*-F is not in the class of deterministic TOL languages).

One has however to be careful in understanding this result. Note for example that if $\Sigma = \{a_1,\ldots,a_n\}$ for some $n \geq 2$, then the deterministic TOL system $G = < \Sigma,\{P_1,\ldots,P_{n-1}\},a_n >$, where

$P_i = \{a_n \to a_n a_i\} \cup \{a_j \to a_j \mid 1 \leqslant j \leqslant n-1\}$ for $1 \leqslant i \leqslant n-1$ is such that, for each $k \geqslant 1$, $\pi_k(L(G)) \geqslant (n-1)^k$. The ramifications of Theorem 1 will be discussed in more detail below.

I.3. Some subclasses of the class of TOL languages.

We will explore now further the subword point of view of the deterministic restriction in TOL systems. In particular we will see that this way of viewing deterministic TOL systems and languages possesses one very pleasant and desirable feature. It is "very sensitive" to various structural changes imposed on the class of TOL systems. In fact we will be able to classify a number of subclasses of the class of TOL systems according to their subword generating efficiency.

First we need some definitions.

Definition 2. A TOL system $G = < \Sigma, \mathcal{P}, \sigma >$ is called:

1) a 0L system if $\#\mathcal{P} = 1$,

2) propagating, if for every P in $\mathcal{P}$, $P \subset \Sigma \times \Sigma^+$,

3) everywhere growing, if for every P in $\mathcal{P}$ and every α in Σ^*, whenever $a \to \alpha$ is in P (for arbitrary a in Σ), then $|\alpha| > 1$,

4) uniform, if there exists an integer $t > 1$ such that, for every P in $\mathcal{P}$ and every α in Σ^*, if $a \to \alpha$ is in P (for arbitrary a in Σ), then $|\alpha| = t$.

Definition 3. A TOL language L is called propagating, everywhere growing, uniform or a 0L language if $L = L(G)$ for a propagating TOL system, everywhere growing TOL system, uniform TOL system or a 0L system, respectively.

We will use the letters P, G and U to denote the propagating, everywhere growing, and uniform restrictions respectively. Thus, for example, a UTOL system means a uniform TOL system and a deterministic GOL system means a deterministic everywhere growing 0L system. It should be obvious to the reader that a GTOL system (language) is also a PTOL system (language) and that each UTOL system (language) is also a GTOL system (language).

Example 2.

1) $G = < \{a\}, \{\{a \to a,\ a \to aa\}\}, a >$ is a POL system. It is not deterministic. Thus $\{a\}^+$ is a POL language.

2) The TOL system from Example 1 is a deterministic GTOL system. Thus $\{a^{2^{n}3^{m}} b^{2^{n}3^{m}} \mid n,m \geqslant 0\}$ is a deterministic GTOL language.

We would like to point out that the restrictions which have been defined in this section (propagating, 0L, etc.) were not introduced for the purpose of this paper; they have already been studied earlier in the theory of L systems.

I.4. Everywhere growing deterministic TOL languages.

As has been indicated at the end of section I.1, even deterministic PTOL systems can generate "a lot" of subwords (say, for each $k \geqslant 0$, at least $(n-1)^k$ out of the total number n^k of possible subwords of length k in an alphabet of size $n \geqslant 2$). The situation is however quite different for deterministic GTOL languages.

Theorem 2.

1) If L is a deterministic GTOL language, then there exist positive constants α and β, such that, for every $k > 0$, $\pi_k(L) \leqslant \alpha k^\beta$.

2) For every positive number ℓ, there exists a deterministic UTOL language L such that if α, β are positive constants such that, for every $k > 0$, $\pi_k(L) \leqslant \alpha k^\beta$, then $\beta > \ell$.

I.5. Deterministic 0L languages.

The class of deterministic 0L systems is one of the most important and most intensively studied classes of L systems (see, e.g., [45], [75], [82] and [95]). In this section we shall investigate the "subword complexity" of deterministic 0L languages as well as how various structural restrictions on the class of deterministic 0L systems influence the subword complexity of the corresponding classes of languages.

As to the whole class of deterministic 0L languages we have the following result.

Theorem 3.

1) For every deterministic 0L language L there exists a constant α_L such that, for every $k > 0$, $\pi_k(L) \leqslant \alpha_L k^2$.

2) For every positive number ℓ there exists a deterministic POL language L such that $\pi_k(L) \geqslant \ell . k^2$ for infinitely many positive integers k.

If we restrict ourselves to languages generated by deterministic 0L systems in which each letter is rewritten as a word of length at least 2, then we get the following subword complexity class.

Theorem 4.

1) For every deterministic GOL language L there exists a positive constant α_L such that, for every $k > 0$, $\pi_k(L) \leqslant \alpha_L.k.\log k$.

2) For every positive number ℓ there exists a deterministic GOL language L such that $\pi_k(L) \geqslant \ell.k.\log k$ for infinitely many positive integers k.

Further restriction to deterministic uniform 0L systems yields us a class of generative devices with very limited ability of subword generation.

Theorem 5.

1) For every deterministic UOL language L there exists a positive constant α_L such that, for every $k > 0$, $\pi_k(L) \leqslant \alpha_L.k$.

2) For every positive number ℓ, there exists a deterministic UOL language L such that $\pi_k(L) \geqslant \ell.k$ for infinitely many positive integers k.

PART II
L systems with interactions

In an L system with interaction, the rewriting of a letter in a string depends on the context in which the letter occurs (in other words, two occurrences of the same letter may have to be rewritten in different ways if they are in different contexts).

This part of the paper will be organized in more or less the same way as Part I so that the reader can more easily compare and contrast the results for L systems without interactions and those for L systems with interactions.

II.1. TIL systems and languages.

Whereas TOL systems attempt to model growth in different environments but with no cell interactions, TIL systems also allow interaction among cells to take place in addition to environmental changes. They were introduced in (Lee & Rozenberg)[3], where the relevant formal definitions and basic properties can be found. (TIL systems, or languages, abbreviates "table L systems, or languages, with interactions".)

A TIL system G has four components, $G = < \Sigma, \rho, \sigma, g >$, where

(i) Σ is the alphabet,

3) Lee & Rozenberg: TIL systems and languages, submitted for publication.

(ii) σ is the axiom, as in the TOL case.
(iii) The symbol g is a new symbol, called the environment symbol. It represents the environment and its usage will be clear from the following description of productions in G.
(iv) $\mathcal{P}$ is a finite set of tables of productions. Each production is of the form $< \alpha,a,\beta > \rightarrow \gamma$, where $a \in \Sigma, \alpha \in g^*\Sigma^*, \beta \in \Sigma^*g^*, \gamma \in \Sigma^*$. For each particular system G, there are numbers $k,\ell \geq 0$ such that $|\alpha| = k$ and $|\beta| = \ell$ for all productions in G. The meaning of $< \alpha,a,\beta > \rightarrow \gamma$ is that an occurrence of the letter a in a word, with the string of letters α immediately to its left and the string of letters β immediately to its right, may be replaced by the string γ . α and β are thus the left and right contexts for a, respectively. Productions for letters at the edges of a string will have an appropriate number of environment symbols g in the context. A string x is said to derive a string y if the letters of x in $g^k x g^\ell$ can be rewritten in the above way to produce the string y, where all productions are from an arbitrarily chosen table.

The language generated by a TIL system G, denoted as L(G), consists of σ and all strings which can be derived from σ in a finite number of steps. A Language L is called a TIL language if there exists a TIL system G such that L = L(G).

Example 3. Let $G = < \{a\}, \{\{ < g,a,\Lambda > \rightarrow a^3, < a,a,\Lambda > \rightarrow a^2 \}, \{ < g,a,\Lambda > \rightarrow a^5, < a,a,\Lambda > \rightarrow a^3 \}\}, a^5, g >$. Then $L(G) = \{a^{2^n 3^m - 1} | n,m \geq 1\}$. Here the amount of left context is $k = 1$ and the amount of right context is $\ell = 0$.

It should be noted that TOL systems can be identified with those TIL systems whose productions are of the form $< \Lambda,a,\Lambda > \rightarrow \alpha$.

II.2. Deterministic TIL systems.

A TIL system is called deterministic if for each particular environment, a letter in a given context can be replaced by only one string. Formally, the deterministic restriction is defined for TIL systems as follows.

Definition 4. A TIL system $G = < \Sigma,\mathcal{P},\sigma,g >$ is called deterministic if the following condition holds: For each $P \in \mathcal{P}$, each $a \in \Sigma$, if $< \alpha,a,\beta > \rightarrow \gamma_1$ and $< \alpha,a,\beta > \rightarrow \gamma_2$ are productions for a in P in the context of α and β, then $\gamma_1 = \gamma_2$. A language L is called a

deterministic TIL language if there exists a deterministic TIL system G such that L = L(G).

In the rest of Part II we shall look at the role determinism plays in TIL systems from the subword point of view.

First we may remark that the analogue of Theorem 1 for deterministic TIL languages does not hold. It is an easy exercise to construct, for any alphabet Σ (with $\#\Sigma = n$), a deterministic TIL language L such that $\pi_k(L) = n^k$ for every $k \geq 0$; hence for this L,

$$\lim_{k\to\infty} \frac{\pi_k(L)}{n^k} = 1.$$

II.3. Some subclasses of the class of TIL languages.

Analogous to the TOL case, we have the following definition.

Definition 5. A TIL system $G = \langle \Sigma, \mathcal{P}, \sigma, g \rangle$ is called

1) an IL system if $\#\mathcal{P} = 1$.

2) propagating if for every $P \in \mathcal{P}$, $P \subset g^*\Sigma^* \times \Sigma \times \Sigma^* g^* \times \Sigma^+$.

3) everywhere growing if for every $P \in \mathcal{P}$ and every $\gamma \in \Sigma$, whenever $\langle \alpha, a, \beta \rangle \to \gamma$ is in P (for some $a \in \Sigma$, $\alpha \in g^*\Sigma^*$, $\beta \in \Sigma^*\beta^*$), then $|\gamma| > 1$.

4) uniform if there exists an integer $t > 1$ such that for every $P \in \mathcal{P}$ and $\gamma \in \Sigma^*$, if $\langle \alpha, a, \beta \rangle \to \gamma$ is in P (for some $a \in \Sigma$, $\alpha \in g^*\Sigma^*$, $\beta \in \Sigma^*\beta^*$) then $|\gamma| = t$.

Definition 6. A TIL language L is called propagating, everywhere growing, uniform or an IL language if L = L(G) for a propagating TIL system, everywhere growing TIL system, uniform TIL system or an IL system G, respectively.

We shall also use the letters P, G, and U to denote the propagating, everywhere growing and uniform restrictions respectively, as explained for the TOL case.

Example 4.

1) The TIL system G from Example 3 is a deterministic GTIL system. Thus the language $\{a^{2^n 3^m - 1} \mid n,m \geq 1\}$ is a deterministic GTIL language.

2) Let $G = \langle \{a\}, \{\{ \langle \Lambda, a, g \rangle \to a^2, \langle \Lambda, a, a \rangle \to a\}\}, a, g \rangle$. Then G is a deterministic PIL system and so $\{a\}^+$ is a deterministic PIL language.

II.4. Everywhere growing deterministic TIL languages.

At the end of section II.2, we have remarked that given any alphabet Σ a deterministic TIL system can be found generating all possible subwords over Σ. The addition of the everywhere growing restriction reduces this subword generating ability as in the case of L systems without interactions. In fact the analogue of Theorem 2 (concerning deterministic GTOL languages) for deterministic GTIL languages holds.

Theorem 6.

1) If L is a deterministic GTIL language, then there exist positive constants α and β, such that, for every $k > 0$, $\pi_k(L) \leqslant \alpha k^{\beta}$.

2) For every positive number ℓ, there exists a deterministic UTIL language L such that if α,β and positive constants such that, for every $k > 0$, $\pi_k(L) \leqslant \alpha k^{\beta}$, then $\beta > \ell$.

II.5. Deterministic IL languages.

Theorem 3 states that for a deterministic OL language L, the number of subwords of length k is proportional to k^2. Thus the subword generating ability of a OL system is reduced from n^k (where n is the cardinality of the alphabet) to k^2 by the addition of the deterministic restriction. The situation is different concerning IL languages. Vitanyi (personal communication) has a construction which, for any alphabet Σ, produces a DIL system G with alphabet $\Sigma \cup \{a,b\}$ (where a,b are new symbols) which generates all possible subwords over Σ. Thus the following theorem is true.

Theorem 7. Given any integer $n > 2$, there exists a DIL language L such that, for any $k \geqslant 0$, $\pi_k(L) \geqslant (n-2)^k$.

The above theorem says that the addition of determinism to IL systems (in general) reduces only slightly their subword generating ability. Despite this, we find that deterministic GIL and deterministic GOL systems, as well as deterministic UIL and deterministic UOL systems, have the same subword generating power. This can be seen from the following two theorems and Theorems 4 and 5.

Theorem 8.

1) For every deterministic GIL language L there exists a positive constant α_L such that, for every $k > 0$, $\pi_k(L) \leqslant \alpha_L.k.\log k$.

2) For every positive number ℓ there exists a deterministic GIL language such that $\pi_k(L) \geqslant \ell.k.\log k$ for infinitely many positive integers k.

Theorem 9.

1) For every deterministic UIL language L there exists a positive constant α_L such that, for every $k > 0$, $\pi_k(L) \leqslant \alpha_L.k$.

2) For every positive number ℓ, there exists a deterministic UIL language L such that $\pi_k(L) \geqslant \ell.k$ for infinitely many positive integers k.