IA

stichting
mathematisch
centrum

MC

AFDELING INFORMATICA IW 25/74 NOVEMBER

P.M.B. VITANYI AND A. WALKER

STABLE STRING LANGUAGES OF LINDENMAYER SYSTEMS

Prepublication

2e boerhaavestraat 49 amsterdam

Printed at the Mathematical Centre, 49, 2e Boerhaavestraat, Amsterdam.

The Mathematical Centre, founded the 11-th of February 1946, is a non-profit institution aiming at the promotion of pure mathematics and its applications. It is sponsored by the Netherlands Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O), by the Municipality of Amsterdam, by the University of Amsterdam, by the Free University at Amsterdam, and by industries.

AMS (MOS) subject classification scheme (1970): 68A30 68A25 94A30 92A05

ACM -Computer Review- category: 5.22 5.23

STABLE STRING LANGUAGES OF LINDENMAYER SYSTEMS

by

Paul M.B. Vitányi and Adrian Walker.

ABSTRACT

The stable string operation selects from the strings produced by a rewriting system those strings which are invariant under the rewriting rules. Stable string languages of Lindenmayer systems are investigated. (Lindenmayer systems are a class of parallel rewriting systems originally introduced to model the growth and development of filamentous organisms.) For families of Lindenmayer systems the set of languages obtained by the stable string operation are shown to coincide with the sets of languages obtained from these systems by intersecting the languages they produce with a terminal alphabet, except in the case of Lindenmayer systems without interactions. The equivalence of a biologically highly relevant notion, i.e. that of equilibrium oriented behavior in models of morphogenesis, and the formal language concept of intersection with a terminal alphabet, establishes a new link between formal language theory and theoretical biology. Relevance to these two fields is briefly discussed.

KEYWORDS & PHRASES: formal language theory, Lindenmayer systems, dynamically stable strings, nonterminals, language families.

1. INTRODUCTION

Lindenmayer systems, L systems for short, are parallel rewriting systems introduced by Lindenmayer [4] to model the growth and development of filamentous biological organisms. An L system consists of an initial string of letters, symbolizing an initial one dimensional array of cells (a filament), and the subsequent strings (stages of development) are obtained by rewriting all letters of a string simultaneously at each time step. When the rewriting of a letter may depend on the m letters to its left and the n letters to its right we talk about an (m, n) L system. If $m = n = 0$ the L system is said to be context independent or without interactions, if $m + n > 0$ the L system is said to be context dependent or with interactions. Various restrictions and modifications of the original systems have been proposed, with or without biological motivation, and subsequently investigated, see e.g. [2]. The languages produced by L systems consist of all strings derivable from the initial string and thus correspond to the set of all morphological stages the organism may attain in its development. Herman and Walker [3], however, consider the language consisting of all strings produced by the L system which are necessarily rewritten as themselves. Such a language is taken to correspond to the set of adult stages the organism modeled by the L system might reach.

From the formal language point of view the usual way of obtaining languages from rewriting systems, be they serial (e.g. grammars) or parallel (e.g. L systems), is by intersection with a terminal alphabet, i.e. by selecting from all strings that are produced those over a terminal alphabet. The method proposed by Herman and Walker, the stable string operation, consists of selecting from all strings produced by a rewriting system those strings that are invariant under the rewriting rules. A language obtained in this manner is called the stable string language of the system (or, with biological connotations, the adult language). We shall investigate the relation between the two approaches for the various families of L systems. In [3] it is proven that the generating power of context independent L systems with respect to the stable string operation is equal to the generating power of context free grammars with respect to intersection with a terminal alphabet (i.e. the context free languages). This rather unexpected result links the study of stable string languages of L systems with the main body of formal language theory. Since the context free languages are strictly contained in the set of languages obtained from context independent L systems by intersection with a terminal alphabet (see e.g. [2]), the stable string operation yields strictly less than the operation of intersection with a terminal alphabet in this case. However, we shall prove that the set of stable string languages of a family of context dependent L systems coincides with the set of languages obtained from this family by intersection with a terminal alphabet. Moreover, analogous results hold for families of L systems using more than one set of production rules (i.e. table L systems), both context dependent and context independent. By making use of existing results on the intersections of L languages with terminal alphabets we are then able to derive many results concerning stable string languages of L systems, some of which were previously established in Walker [14] by different methods. For a more extensive discussion of the biological motivation concerning L systems in general we refer to [4, 5, 2], and of stable string languages in particular to [3, 14] or to the last section of this paper.

2. STABLE STRING LANGUAGES OF CONTEXT DEPENDENT L SYSTEMS

We assume that the reader is familiar with the usual terminology of formal language theory as e.g. in [7]. Except when indicated otherwise we shall customarily use, with or without indices, i, j, k, h, ℓ, m, n to range over the set of natural numbers $N = \{0, 1, 2, \ldots\}$; a, b, c, d to range over an alphabet Σ; $v, z, w, \alpha, \beta, \omega$ to range over Σ^* i.e. the set of all words over Σ including the *empty word* λ. $\#Z$ denotes the cardinality of a set Z; $lg(z)$ denotes the *length* of a word z and $lg(\lambda) = 0$.

An *(m, n) L system* is a triple $G = \langle \Sigma, P, \omega \rangle$ where Σ is a finite nonempty *alphabet*; P is a finite set of *production rules* of the form $(v_1, a, v_2) \to \alpha$ such that $(v_1, a, v_2) \in \bigcup_{i=0}^{m} \Sigma^i \times \Sigma \times \bigcup_{j=0}^{n} \Sigma^j$, $\alpha \in \Sigma^*$, and for each element (v_1, a, v_2) of $\bigcup_{i=0}^{m} \Sigma^i \times \Sigma \times \bigcup_{j=0}^{n} \Sigma^j$ there is at least one such rule in P; $\omega \in \Sigma\Sigma^*$ is called the *axiom*. P induces a relation $\underset{G}{\Rightarrow}$ on Σ^* as follows. $v \underset{G}{\Rightarrow} v'$ or v *directly produces* v' in G iff $v = a_1 a_2 \ldots a_k$, $v' = \alpha_1 \alpha_2 \ldots \alpha_k$, and for all i, $i = 1, 2, \ldots, k$,

$$(a_{i-m} a_{i-m+1} \ldots a_{i-1}, a_i, a_{i+1} a_{i+2} \ldots a_{i+n}) \to \alpha_i$$

is a rule in P, where we take $a_j = \lambda$ for $j < 1$ or $j > k$.

By definition $\lambda \underset{G}{\Rightarrow} \lambda$. As usual $\overset{*}{\underset{G}{\Rightarrow}}$ is the reflexive and transitive closure of $\underset{G}{\Rightarrow}$ and we say v *produces* v' in G if $v \overset{*}{\underset{G}{\Rightarrow}} v'$. We dispense with the subscripts on the relations when G is understood. The L language produced by G is defined by $L(G) = \{w \mid \omega \overset{*}{\underset{G}{\Rightarrow}} w\}$. At this stage we would like to point out that although our definition of an L system varies from the usual one (see e.g. [2]), in that it dispenses with the environmental letter g, it is exactly equivalent to the previous definitions. With regard to the amount of context used the following terminology is standard throughout the literature: a (0,0)L system is called a 0L system or a context independent L system (without interactions); a (0,1)L system or (1,0) L system is called an 1L system (one directional); a (1,1)L system is called a 2L system (two directional); a (m,n)L system such that $m + n > 0$ is called an IL system or context dependent L system (with interactions).

An L system $G = \langle\Sigma, P, \omega\rangle$ is called propagating if no rule in P is of the form $(v_1,a,v_2) \to \lambda$; it is called deterministic if for each element of $\bigcup_{i=0}^{m} \Sigma^i \times \Sigma \times \bigcup_{j=0}^{n} \Sigma^j$ there is exactly one rule in P. These properties are indicated by prefixing the appropriate capitals to the type of L system, e.g.,PD2L system, PIL system, D(1,2)L system etc. A language L is obtained from L(G) by intersection with a terminal alphabet if $L = L(G) \cap V_T^*$ where V_T is a subset of the alphabet of G. The stable string language of an L system $G = \langle\Sigma, P, \omega\rangle$ is defined by

$A(G) = \{w \in \Sigma^* \mid w \in L(G)$ and $w \Rightarrow z$ implies $z = w\}$.

Our investigations shall be concerned with the following families of languages. Let X be any type of L system. The family of L languages produced by the XL systems is denoted by $\mathcal{L}(XL)$; the family of languages obtained from $\mathcal{L}(XL)$ by intersection with a terminal alphabet is denoted by $\mathcal{E}(XL)$; the family of stable string languages of XL systems is denoted by $\mathcal{A}(XL)$. We denote the families of regular, context free, indexed, context sensitive and recursively enumerable languages by $\mathcal{L}(REG)$, $\mathcal{L}(CF)$, $\mathcal{L}(INDEX)$, $\mathcal{L}(CS)$ and $\mathcal{L}(RE)$, respectively. We immediately note the following. For any L system G

(i) $A(G) \subseteq L(G)$.

(ii) $\#A(G) \geq 0$ but $\#L(G) > 0$.

(iii) If G is deterministic then $\#A(G) \in \{0, 1\}$.

Furthermore,

(iv) $\mathcal{L}(XL) \subseteq \mathcal{E}(XL)$.

<u>Example</u>. $G = \langle\{a,b\}, \{(\lambda,a,\lambda) \to a, (\lambda,a,\lambda) \to aa, (\lambda,a,\lambda) \to b, (\lambda,b,\lambda) \to b\}, a\rangle$; i.e. G is a 0L system. $L(G) = \{a,b\}\{a,b\}^*$. $A(G) = \{b\}\{b\}^*$.

In the sequel the lemmas are our main results. They serve as technical tools to derive theorems and corollaries concerning the inclusion relations between the above families of languages.

Lemma 1. Let $G = \langle\Sigma, P, \omega\rangle$ be any type of (m,n)L system such that $m + n > 0$ and let V_T be a subset of Σ. There exists an algorithm which, given G and V_T, produces a (m,n)L system $G' = \langle\Sigma', P', \omega'\rangle$ of the same type (but for determinism and the cardinality of the alphabet), a subset V'_T of Σ' and an isomorphism h from V_T^* onto $V_T'^*$ such that $h(L(G) \cap V_T^*) = A(G')$.

Proof. We shall prove the Lemma in three stages:

(i) $L(G') \cap V_T^* = L(G) \cap V_T^*$,

(ii) $L(G') \cap V_T'^* = h(L(G') \cap V_T^*)$,

(iii) $L(G') \cap V_T'^* = A(G')$.

Consider the system $G' = \langle\Sigma', P', \omega'\rangle$ which is constructed as follows.

$$\Sigma' = \Sigma \cup V'_T \cup \{F,s\},$$

where Σ, V'_T and $\{F,s\}$ are disjoint, $\#V'_T = \#V_T$ and h is any isomorphism from V_T^* onto $V_T'^*$. $\omega' = s$ and the set of production rules P' is defined by

(1)	$(v_1,s,v_2) \to \omega$		for all v_1, v_2 in Σ'^*.
(2)		$\to h(\omega)$	if $\omega \in V_T^*$.
(3)	$(v_1,a,v_2) \to \alpha$		if $(v_1,a,v_2) \to \alpha \in P$.
(4)		$\to h(\alpha)$	if $(v_1,a,v_2) \to \alpha \in P$ and $\alpha \in V_T^*$.
(5)		$\to FF$	for all $v_1av_2 \notin V'_TV_T'^*$.
(6)		$\to a$	for all $v_1av_2 \in V'_TV_T'^*$.

(i) Since $P \subseteq P'$ and $P' - P$ does not produce words over V_T (except possibly ω) we have

$$L(G') \cap V_T^* = L(G) \cap V_T^*.$$

(ii) Suppose $s \overset{*}{\Rightarrow} z \Rightarrow v$ and $v \in V_T^*$. By (2) and (4) we then have also $s \overset{*}{\Rightarrow} z \Rightarrow h(v)$. Therefore

$$h(L(G') \cap V_T^*) \subseteq L(G') \cap V_T'^*.$$

Suppose $s \overset{*}{\Rightarrow} z \Rightarrow v$ and $v \in V_T'^*$.

Case 1. $z = s$. $z \Rightarrow h^{-1}(v) = \omega$ by (2) and (1).

Case 2. $z \neq s$. and $z \neq v$. By (4) and (3) $z \Rightarrow h^{-1}(v)$.

Case 3. $z \neq s$ and $z = v$.

Since cases 1-3 exhaust all possibilities of producing words over $V_T'^*$ we have

$$L(G') \cap V_T'^* \subseteq h(L(G') \cap V_T^*),$$

and therefore

$$L(G') \cap V_T'^* = h(L(G') \cap V_T^*).$$

(iii) Let $v \in V_T'^*$ and $v \Rightarrow z$. The only rules applicable to v are those of (6) and therefore $z = v$ and

$$L(G') \cap V_T'^* \subseteq A(G').$$

Suppose $v \Rightarrow v$ and $v \notin V_T'^*$. By (5) then also $v \Rightarrow v_1 FF v_2$ for some words v_1, v_2 in Σ'^* so $v \notin A(G')$. Therefore

$$A(G') \subseteq L(G') \cap V_T'^*.$$

Hence

$$A(G') = L(G') \cap V_T'^*. \blacksquare$$

Lemma 2. Let $G = \langle \Sigma, P, \omega \rangle$ be a (deterministic) P(m,n)L system. There is an algorithm which, given G, produces a (deterministic) P(m,n)L system $G' = \langle \Sigma', P', \omega' \rangle$, a subset V_T of Σ' and an isomorphism h from V_T^* onto Σ^* such that

$$h(L(G') \cap V_T^*) = A(G).$$

Proof. Construct $G' = <\Sigma', P', \omega'>$ as follows. $\Sigma' = \Sigma \times \{0,1\}$; $\omega' = (a_1,0)(a_2,0)\ldots(a_k,0)$ for $\omega = a_1a_2\ldots a_k$. Let g be a letter to letter homomorphism from Σ'^* onto Σ^* defined by $g((a,i)) = a$ for $i \in \{0,1\}$, and define P', $i\in\{0,1\}$, by

(1) $(v_1, (a,i), v_2) \to (a_1,0)(a_2,0)\ldots(a_\ell,0)$ if $(g(v_1),a,g(v_2)) \to a_1a_2\ldots a_\ell \in P$ and there is a rule $(g(v_1),a,g(v_2)) \to \alpha$ in P such that $\alpha \neq a$.

(2) $\to (a,1)$ otherwise.

Let $V_T = \{(a,1) \mid a \in \Sigma\}$ and define $h\colon V_T^* \to \Sigma^*$ by $h((a,1)) = a$.

Suppose $v \in A(G)$; i.e. if $\omega \overset{*}{\underset{G}{\Rightarrow}} v \underset{G}{\Rightarrow} z$ then $z = v$. Since G is propagating every letter in v must necessarily produce itself and for $v = a_1a_2\ldots a_\ell$ we therefore have $\omega' \overset{*}{\underset{G'}{\Rightarrow}} (a_1,i_1)(a_2,i_2)\ldots(a_\ell,i_\ell) \underset{G'}{\Rightarrow} (a_1,1)(a_2,1)\ldots(a_\ell,1)$ where $i_j \in \{0,1\}$, $1 \leq j \leq \ell$. Since $(a_1,1)(a_2,1)\ldots(a_\ell,1) \in V_T^*$ we have

$$A(G) \subseteq h(L(G') \cap V_T^*).$$

Suppose $v \in V_T^*$ and $\omega' \overset{*}{\underset{G'}{\Rightarrow}} z \underset{G'}{\Rightarrow} v$. Then also $\omega \overset{*}{\underset{G}{\Rightarrow}} g(z) \underset{G}{\Rightarrow} g(v)$ and because of (2) $g(z) = g(v)$ and $g(z) \underset{G}{\not\Rightarrow} x$ for $x \neq g(v)$. Therefore

$$h(L(G') \cap V_T^*) \subseteq A(G)$$

and the lemma follows. ∎

Theorem 1. (i) Let m,n be nonnegative integers such that $m + n > 0$ and let X be any property of L systems which is preserved under the construction in the proof of Lemma 1 (e.g. propagating). Then $\mathcal{E}(X(m,n)L) \subseteq \mathcal{A}(X(m,n)L)$.

(ii) Let m,n be nonnegative integers and let X be any property of L systems which is preserved under the construction in the proof of Lemma 2 (e.g. determinism, lengths of right hand sides of production rules). Then $\mathcal{A}(XP(m,n)L) \subseteq \mathcal{E}(XP(m,n)L)$.

Proof. (i) Let G be an X(m,n)L system and let V_T be a subset of the alphabet of G. By Lemma 1 there is an algorithm which, given G and V_T, produces an X(m,n)L system G' such that A(G') is isomorphic with $L(G) \cap V_T^*$. Since families of languages are invariant under isomorphism (i) holds.

(ii) Let G be a propagating X(m,n)L system. By Lemma 2 there is an algorithm which, given G, produces a propagating X(m,n)L system G' and a subset V_T of the alphabet of G' such that $L(G') \cap V_T^*$ is isomorphic with A(G). Since families of languages are invariant under isomorphism (ii) holds. ■

Corollary 1. $\mathcal{A}(P(m,n)L) = \mathcal{E}(P(m,n)L)$ for $m + n > 0$.

Since it follows from van Dalen [1] that $E(1L) = L(RE)$ we have by Theorem 1(i)

Corollary 2. $A(1L) = E(1L) = L(RE) = E(IL) = A(IL)$.

Another result of van Dalen [1] is that $E(P2L) = L(CS)$. Since it is easy to give a linear bounded automaton construction (see e.g. [7]) to show that each intersection of a $P(m,n)L$ language with a terminal alphabet is a context sensitive language we have by Corollary 1:

Corollary 3. $A(P2L) = E(P2L) = L(CS) = E(PIL) = A(PIL)$.

Furthermore,

Corollary 4. $A(P1L) = E(P1L) \subseteq L(CS)$.

We might observe that if G is deterministic then $A(G)$ consists of either one word or the empty set. It follows from the argument used in Vitányi [10] to show the undecidability of the question whether or not the lengths of strings in PD1L systems grow unboundedly, that the following theorem holds.

Theorem 2. It is undecidable for an arbitrary PD1L system G whether or not $A(G) = \emptyset$.

Although it is obviously not the case that $A(PD1L) = E(PD1L)$ we obtain from Theorem 1(ii) and Theorem 2 the additional result:

Corollary 5. It is undecidable for an arbitrary PD1L system G and a subset V_T of the alphabet of G whether or not $L(G) \cap V_T^* = \emptyset$.

For stable string languages of D0L systems, however, the emptyness problem is solvable. In Vitányi [11] it is proven that for a D0L system $G = \langle \Sigma, P, \omega \rangle$ it is decidable whether or not $L(G)$ is finite, and that if $L(G)$ is finite then $\#L(G) \leq f(G)$ where the value of f for each G is easily computed. Therefore $A(G) \neq \emptyset$ iff $L(G)$ is not infinite and $\omega \Rightarrow \omega_0 \Rightarrow \omega_1 \Rightarrow \ldots \Rightarrow \omega_{f(G)-1} \Rightarrow \omega_{f(G)} = \omega_{f(G)-1}$. In fact, for our current concerns, $\omega \Rightarrow \omega_0 \Rightarrow \omega_1 \Rightarrow \ldots \Rightarrow \omega_{\#\Sigma-1} \Rightarrow \omega_{\#\Sigma} = \omega_{\#\Sigma-1}$ suffices according to [11].

3. STABLE STRING LANGUAGES OF L SYSTEMS USING TABLES

A X(m,n)L system using tables, XT(m,n)L system, is like a X(m,n)L system except that the set of production rules is replaced by a finite set of such sets: a set of tables. Table L systems were introduced by Rozenberg [6] where also a biological motivation can be found.

A <u>XT(m,n)L system</u> is a triple $G = \langle \Sigma, P, \omega \rangle$ where $P = \{P_1, P_2, \ldots, P_k\}$ such that $G_i = \langle \Sigma, P_i, \omega \rangle$ is an X(m,n)L system for $i = 1, 2, \ldots, k$. P induces an equivalence relation $\underset{G}{\Rightarrow}$ on Σ^* defined by $v \underset{G}{\Rightarrow} v'$ if $v \underset{G_i}{\Rightarrow} v'$ for some i, $1 \leq i \leq k$. For $v \underset{G_i}{\Rightarrow} v'$ we also write $v \underset{P_i}{\Rightarrow} v'$, $i \in \{1, 2, \ldots, k\}$. As usual $\overset{*}{\underset{G}{\Rightarrow}}$ is the reflexive and transitive closure of $\underset{G}{\Rightarrow}$. We dispense with the subscripts on the relation if G is understood. The <u>language</u> produced by a table L system $G = \langle \Sigma, P, \omega \rangle$ is defined by $L(G) = \{w \mid \omega \overset{*}{\Rightarrow} w\}$. The <u>stable string language</u> of G is $A(G) = \{w \in \Sigma^* \mid w \in L(G)$ and $w \Rightarrow z$ implies $z = w\}$. The constructions in Lemmas 1 and 2 show immediately that the analog of Theorem 1 holds for table L systems in general and for table L systems using k tables (i.e. <u>T_kL systems)</u> in particular. Hence we have the following additional corollaries from Theorem 1.

Corollary 6. $A(PT_k(m,n)L) = E(PT_k(m,n)L)$ for all nonnegative integers m, n, k such that $m + n > 0$ and $k > 0$.

By the usual linear bounded automaton argument, c.f. section 2, it is easy to show that the intersections of propagating TIL languages with a terminal alphabet are context sensitive. Therefore we obtain by corollaries 3 and 6

Corollary 7. $A(PT_1 2L) = A(P2L) = L(CS) = A(PTIL)$.

Moreover, we have from Theorem 1

Corollary 8. $A(PT_k 1L) = E(PT_k 1L) \subseteq L(CS)$, for all $k > 0$.

Corollary 9. (i) $A(PT_k 0L) \subsetneq E(PT_k 0L)$, for all $k > 0$.

(ii) $A(PDT_k(m,n)L) \subsetneq E(PDT_k(m,n)L)$, for all $k > 0$.

Corollary 10. $A(T_1 1L) = A(1L) = L(RE) = A(TIL)$.

Lemma 3. Let $G = \langle\Sigma, P, \omega\rangle$ be any T0L system. There exists an algorithm which, given G, produces a T0L system $G' = \langle\Sigma', P', \omega'\rangle$ and a subset V_T of Σ' such that $A(G) = L(G') \cap V_T^*$.

Proof. It is easy to see that

$$A(G) = \bigcap_{i=1}^{k} \{w \in \Sigma^* \mid w \underset{P_i}{\Rightarrow} z \text{ implies } z = w\} \cap L(G).$$

From Herman and Walker [3, lemma 3] it follows that there exists an algorithm which, given $<\Sigma, P_i>$, $i = 1,2,\ldots,k$, produces a finite set $W_i \subseteq \Sigma^*$ such that $W_i^* = \{w \in \Sigma^* \mid w \underset{P_i}{\Rightarrow} z \text{ implies } z = w\}$. Therefore, $A(G) = \bigcap_{i=1}^{k} W_i^* \cap L(G)$. From Herman and Rozenberg [2, Theorem 9.3 (iv)] it follows that there exists an algorithm which, given a T0L system G and a regular expression R, produces a T0L system $G' = <\Sigma', P', \omega'>$ and a subset V_T of Σ' such that $L(G') \cap V_T^* = L(G) \cap R$. ■

Lemma 4. Let $G = <\Sigma, P, \omega>$ be any type of T0L system, e.g. propagating, deterministic or both, such that $\#P > 1$. There exists an algorithm which, given G and a subset V_T of Σ, produces a T0L system $G' = <\Sigma', P', \omega'>$, of the same type, $\#P' = \#P$, such that

(i) $L(G) \cap V_T^* = L(G') \cap V_T^*$

(ii) $A(G') = L(G') \cap V_T^*$.

Proof. Let $G = <\Sigma, P, \omega>$ where $P = \{P_1, P_2, \ldots, P_k\}$. Construct $G' = <\Sigma', P', \omega'>$ as follows.

$\Sigma' = V_T \cup (\Sigma \times \{1, 2, \ldots, k\} \times \{0, 1\}) \cup \{F, s\}$

where $F, s \notin \Sigma$. $\omega' = s$.

$P' = \{P_1', P_2', \ldots, P_k'\}$ where P_i', $1 \leq i \leq k$, is defined by

(1) $s \to (a_1,1,1)(a_2,1,1)\ldots(a_n,1,1)$ if $\omega = a_1a_2\ldots a_n$.

(2) $(a,j,0) \to (a_1,i,1)(a_2,i,1)\ldots(a_n,i,1)$ for all $j \in \{1,\ldots,k\}$ and $a \to a_1a_2\ldots a_n \in P_i$.

(3) $(a,i,1) \to (a,i,0)$ for all $a \in \Sigma$.

(4) $(a,j,1) \to a$ for all $a \in V_T$ and all $j \neq i$.

(5) $(a,j,1) \to FF$ for all $a \in \Sigma - V_T$ and all $j \neq i$.

(6) $F \to FF$

(7) $a \to a$ for all $a \in V_T$.

(i) Suppose $\omega \overset{*}{\underset{G}{\Rightarrow}} v$ and $v \in V_T^*$. Then there are words $v_0 = \omega, v_1, v_2, \ldots, v_h = v$ in Σ^* and tables $P_{i_1}, P_{i_2}, \ldots, P_{i_h}$ in P such that $v_0 \underset{P_{i_1}}{\Rightarrow} v_1 \underset{P_{i_2}}{\Rightarrow} v_2 \underset{P_{i_3}}{\Rightarrow} \cdots \underset{P_{i_h}}{\Rightarrow} v_h$. Let

$v_i = a_{i1}a_{i2}\ldots a_{in_i}$ for $i = 0, 1, \ldots, h$. Then

$$s \underset{P_1'}{\Rightarrow} (a_{01},1,1)(a_{02},1,1)\ldots(a_{0n_0},1,1) \underset{P_1'}{\Rightarrow} (a_{01},1,0)(a_{02},1,0)\ldots(a_{0n_0},1,0)$$

$$\underset{P_{i_1}'}{\Rightarrow} (a_{11},i_1,1)(a_{12},i_1,1)\ldots(a_{1n_1},i_1,1) \underset{P_{i_1}'}{\Rightarrow} (a_{11},i_1,0)(a_{12},i_1,0)\ldots(a_{1n_1},i_1,0)$$

$$\ldots$$

$$\underset{P_{i_h}'}{\Rightarrow} (a_{h1},i_h,1)(a_{h2},i_h,1)\ldots(a_{hn_h},i_h,1) \underset{P_j'}{\Rightarrow} a_{h1}a_{h2}\ldots a_{hn_h} = v, \; j \neq i_h.$$

Hence $v \in L(G') \cap V_T^*$ and therefore

$$L(G) \cap V_T^* \subseteq L(G') \cap V_T^*.$$

Suppose $s \overset{*}{\underset{G'}{\Rightarrow}} v$ and $v \in V_T^*$. Since $s \notin V_T^*$ we have (for $\omega = a_1 a_2 \ldots a_n$)

$$s \underset{G'}{\Rightarrow} (a_1,1,1)(a_2,1,1)\ldots(a_n,1,1) \overset{*}{\underset{G'}{\Rightarrow}} z \underset{G'}{\Rightarrow} v = b_1 b_2 \ldots b_m.$$

If $z \in V_T^*$ then by (7) $v = z$. Assume $z \notin V_T^*$. It is easy to check by inspecting the production rules that no symbol of V_T occurs in z. By (4) $z = (b_1,i_j,1)(b_2,i_j,1)\ldots(b_m,i_j,1)$ for some $i_j \in \{1, 2, \ldots, k\}$, and $z \underset{P_h}{\Rightarrow} v$ for $h \neq i_j$.
Hence there are tables $P_{i_1}, P_{i_2}, \ldots, P_{i_j}$ in P and words

$v_0 = \omega,\ v_1,\ v_2,\ \ldots,\ v_j = v$ in Σ^*, $v_i = a_{i1} a_{i2} \ldots a_{in_i}$,
$0 \leq i \leq j$, such that

$$s \underset{P_1}{\Rightarrow} (a_{01},1,1)(a_{02},1,1)\ldots(a_{0n_0},1,1) \underset{P_1}{\Rightarrow} (a_{01},1,0)(a_{02},1,0)\ldots(a_{0n_0},1,0)$$

$$\underset{P_{i_1}}{\Rightarrow} (a_{11},i_1,1)(a_{12},i_1,1)\ldots(a_{1n_1},i_1,1) \underset{P_{i_1}}{\Rightarrow} (a_{11},i_1,0)(a_{12},i_1,0)\ldots$$

$$\ldots (a_{1n_1},i_1,0)$$

$$\ldots$$

$$\underset{P_{i_j}}{\Rightarrow} (a_{j1},i_j,1)(a_{j2},i_j,1)\ldots(a_{jn_j},i_j,1) \underset{P_h}{\Rightarrow} a_{j1} a_{j2} \ldots a_{jn_j} = v,\ h \neq i_j$$

But then also

$$\omega = a_{01} a_{02} \ldots a_{0n_0} \underset{P_{i_1}}{\Rightarrow} a_{11} a_{12} \ldots a_{1n_1} \underset{P_{i_2}}{\Rightarrow} \ldots \underset{P_{i_j}}{\Rightarrow} a_{j1} a_{j2} \ldots a_{jn_j} = v,$$

i.e. $\omega \overset{*}{\underset{G}{\Rightarrow}} v$ and therefore

$$L(G') \cap V_T^* \subseteq L(G) \cap V_T^*.$$

Hence

$$L(G') \cap V_T^* = L(G) \cap V_T^*.$$

(ii) Suppose $s \overset{*}{\underset{G'}{\Rightarrow}} v$ and $v \in V_T^*$. By (7) $v \in A(G')$ and therefore

$$L(G') \cap V_T^* \subseteq A(G').$$

Suppose $s \overset{*}{\underset{G'}{\Rightarrow}} v$ and $v \notin V_T^*$. By the inherent synchronism of the production rules in P' we have, for $v \neq \lambda$,

$v \in \{s\} \cup ((V_T \cup \{F\})^* - V_T^*) \cup (\Sigma \times \{1,2,\ldots,k\} \times \{0\})^* \cup$
$(\Sigma \times \{1,2,\ldots,k\} \times \{1\})^*$.

It is easily seen that for each of the possibilities $v \notin A(G')$ and therefore

$$A(G') \subseteq L(G') \cap V_T^*.$$

Hence

$$A(G') = L(G') \cap V_T^*. \blacksquare$$

<u>Theorem 3</u>. Let G be an XT_k0L system, $X = \{\lambda, P, PD\}$ $k > 1$. There exists algorithms which given G and a subset V_T of the alphabet of G, produce XT_k0L systems G', G" and a subset V_T' of the alphabet of G' such that

(i) $A(G) = L(G') \cap V_T'^*$,

(ii) $A(G'') = L(G) \cap V_T^*$.

<u>Proof</u>. (i) The construction in Lemma 2 leaves the propagating and deterministic property intact and goes through analogously for T0L systems without changing the number of tables (c.f. Corollary 9 (i) and (ii)). The general case is covered by Lemma 3 and adds one table. Since from Herman and Rozenberg [2] it follows that there is an algorithm which, given a T_k0L system G''' and a subset V_T''' of the alphabet of G''', produces a T_20L system G' and a subset V_T' of the alphabet

of G' such that $L(G') \cap V_T'^* = L(G''') \cap V_T'''^*$, this proves (i).

(ii) By Lemma 4. ■

<u>Corollary 11</u>.

(i) $\mathcal{A}(T_k 0L) = \mathcal{E}(T_k 0L)$ $, k > 1$.

(ii) $\mathcal{A}(PT_k 0L) = \mathcal{E}(PT_k 0L)$ $, k > 1$.

(iii) $\mathcal{A}(PDT_k 0L) = \mathcal{E}(PDT_k 0L), k > 1$.

Since the construction in the proof of Lemma 4 also leaves determinism intact in the general case we have furthermore,

<u>Corollary 12</u>. $\mathcal{E}(DT_k 0L) \subseteq \mathcal{A}(DT_k 0L)$ for $k > 1$.

We now need the following results, (see e.g. [2, chapters 7 and 10]), to round off the picture.

<u>Theorem 4</u>.

(i) If $L \in \mathcal{E}(0L)$ then $L - \{\lambda\} \in \mathcal{E}(P0L)$

(ii) If $L \in \mathcal{E}(T_k 0L)$ then $L - \{\lambda\} \in \mathcal{E}(PT_{k+1} 0L)$

(iii) $\mathcal{E}(T_2 0L) = \mathcal{E}(T0L)$

(iv) $\mathcal{E}(PT_2 0L) = \mathcal{E}(PT0L)$

(v) $\mathcal{L}(CF) \subsetneq \mathcal{E}(T_1 0L) \subsetneq \mathcal{E}(T_2 0L) \subseteq \mathcal{L}(INDEX)$

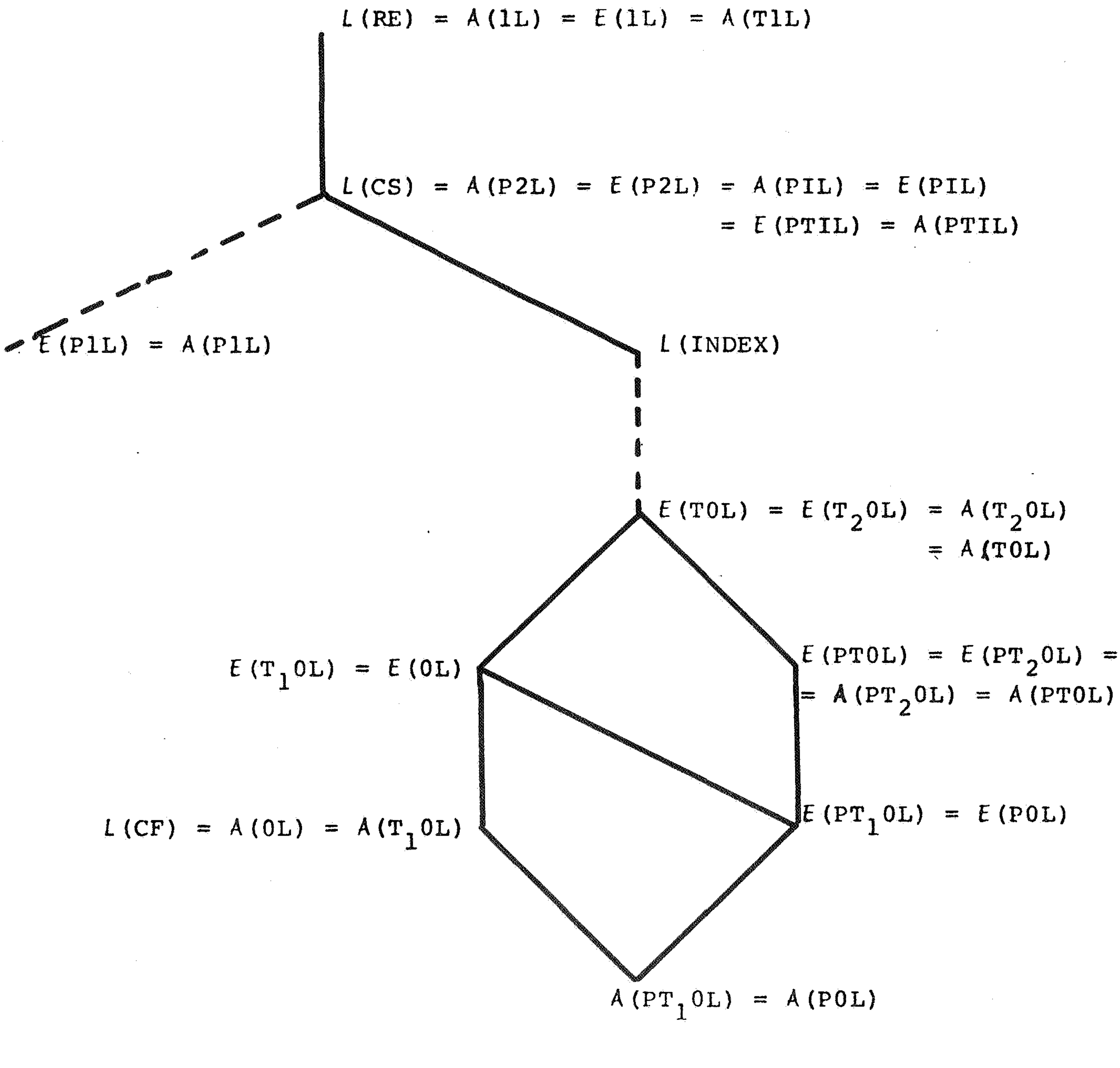
And from Herman and Walker [3]

<u>Theorem 5</u>. $\mathcal{A}(0L) = \mathcal{A}(T_1 0L) = \mathcal{L}(CF)$.

Let us summarize the results so far. We have established the following relations between the families of languages we have discussed.

1. $L(RE) = A(1L) = E(1L) = A(T1L)$: Corollaries 2,10.
2. $L(CS) = A(P2L) = E(P2L) = A(PIL) = E(PIL) = A(PT2L) = A(PTIL)$: Corollaries 3,7.
3. $L(CS) \subsetneq L(RE)$ is well known, see e.g. Salomaa [7].
4. $E(P1L) = A(P1L)$: Corollary 4.
5. $E(P1L) \subseteq E(P2L)$: by definition.
6. $L(INDEX) \subsetneq L(CS)$ is well known, see e.g. Salomaa [7].
7. $E(T0L) = E(T_2 0L) = A(T_2 0L) = A(T0L)$: Theorem 4 (iii) and Corollary 11(i).
8. $A(T_1 0L) = A(0L) = L(CF)$: Theorem 5.
9. $L(CF) \subsetneq E(0L) = E(T_1 0L) \subsetneq E(T0L) \in L(INDEX)$: Theorem 4 (v).
10. $E(PT0L) = E(PT_2 0L) = A(PT_2 0L) = A(PT0L)$: Corollary 11 (ii) and Theorem 4 (iv).
11. $E(PT0L) = \{\bar{L} \mid \bar{L} = L - \{\lambda\}$ and $L \in E(T0L)\}$.
    Theorem 4 (ii)-(iv).
    Hence $E(PT0L) \subsetneq E(T0L)$.
12. $E(PT_1 0L) = E(P0L) \subsetneq E(PT0L)$: Theorem 4.
13. $E(P0L) = \{\bar{L} \mid \bar{L} = L - \{\lambda\}$ and $L \in E(0L)\}$,
    Theorem 4 (i).
    Hence $E(P0L) \subsetneq E(0L)$.
14. $A(P0L) \subseteq A(0L)$ by definition. Strict inclusion since $\{\lambda\} \in L(CF) - A(P0L)$.
15. $A(P0L) \subseteq E(P0L)$ by Theorem 1 (ii) and strict inclusion follows since e.g. $\{a^{2^n} \mid n \geq 1\} \in E(P0L) - L(CF)$.

The results are shown diagramatically in Figure 1.



L(RE) = A(1L) = E(1L) = A(T1L)
L(CS) = A(P2L) = E(P2L) = A(PIL) = E(PIL)
= E(PTIL) = A(PTIL)
E(P1L) = A(P1L)
L(INDEX)
E(T0L) = E(T20L) = A(T20L)
= A(T0L)
E(T10L) = E(0L)
E(PT0L) = E(PT20L) =
= A(PT20L) = A(PT0L)
L(CF) = A(0L) = A(T10L)
E(PT10L) = E(P0L)
A(PT10L) = A(P0L)

Fig. 1

In the Figure, when two families are connected by a solid line the lower family is strictly included in the upper one; when they are connected by a dotted line the lower family is included in the upper one but it is not known yet whether inclusion is strict; no connection means that neither language family is included in the other, i.e. the two families are incomparable.

The incomparabilities between the families in the lower right hand side of Figure 1 follow from Theorem 4 and the fact that languages obtained from propagating L systems do not contain λ. The relation between $\mathcal{E}$(P1L) and families of languages obtained from propagating table L systems is unknown.

4. STABLE STRINGS OF DETERMINISTIC L SYSTEMS USING TABLES

The concept of languages produced by monogenic rewriting systems is altogether foreign to the usual generative grammar approach since there these languages would either be empty or contain but one element. The same holds for stable string languages of the ordinary deterministic L systems. However, stable string languages of deterministic L systems using more than one table, or deterministic L languages and their intersection with a terminal alphabet are proper language families. We shall now assess the implications of our previous results for the stable string languages of deterministic L systems using more than one table.

(4.1) $\mathcal{A}(PDT_k0L) = \mathcal{E}(PDT_k0L)$ for $k > 1$. (Corollary 11(iii)).

(4.2) $\mathcal{E}(DT_k0L) \subsetneq \mathcal{A}(DT_k0L)$ for $k > 1$. (Corollary 12).

Since the proof technique of Lemma 4 works also in the case of deterministic context dependent L systems using tables we have:

(4.3) $\mathcal{E}(DT_k(m,n)L) \subsetneq \mathcal{A}(DT_k(m,n)L)$ for $k > 1$.

(4.4) $\mathcal{E}(PDT_k(m,n)L) \subseteq \mathcal{A}(PDT_k(m,n)L)$ for $k > 1$.

(4.4) together with Corollary 9 (ii) gives us:

<u>Corollary 13</u>. $\mathcal{A}(PDT_k(m,n)L) = \mathcal{E}(PDT_k(m,n)L) \subseteq \mathcal{L}(CS)$ for $k > 1$. (The latter inclusion follows by the usual linear bounded automaton argument.)

In [12] it is proven that:

(4.5) $E(D2L) = L(RE)$,

(4.6) $E(D1L) \subsetneq L(RE)$,

and,

(4.7) the closure of $E(D1L)$ under letter to letter homomorphism is equal to $L(RE)$.

Using one table to achieve the letter to letter homomorphism it is easy to show that:

(4.8) $E(DT_2 1L) = L(RE)$.

Together with (4.5), (4.6) and (4.7) therefore:

Corollary 14. $E(D1L) \subsetneq E(DT_2 1L) = L(RE) = A(DT_2 1L) = E(D2L) = A(1L)$.

(4.1) (4.2) and Corollary 13 give rise to infinite chains of deterministically produced table L languages where strict inclusion with respect to the number of tables or the amount of context used is unknown as yet. These families of languages tie in with Fig. 1 according to the definitions.

Finally, we would like to point out that much more is proven than claimed by means of corollaries. The lemmas and theorems hold for any family of L systems which is preserved under the construction. If e.g. in Lemma 4 we change the production $F \to FF$ into $F \to F'$ and $F' \to F$ then the growth ranges stay identical i.e.

$\{i \in N \mid i = lg(v)$ and $v \in L(G) \cap V_T^*\} = \{i \in N \mid i = lg(v)$ and $v \in A(G')\}$.

Also in Lemma 2:

$\{i \in N \mid i = lg(v)$ and $v \in A(G)\} = \{i \in N \mid i = lg(v)$ and $v \in L(G') \cap V_T^*\}$.

5. RELEVANCE TO THEORETICAL BIOLOGY AND FORMAL LANGUAGE THEORY

The problem of equilibrium oriented behavior in biological morphogenesis has attracted considerable attention. For instance Turing [9] has analyzed the way in which patterns may form in a ring of cells which is initially in chemical equilibrium but is displaced from it by a small amount. Waddington [13] has given a model, called the epigenetic landscape, for the way in which development is influenced both by the genetic material and by external disturbances. Thom [8] has shown how a topological approach may be used to identify regions of sudden and drastic spontaneous change in a system. These investigations have been concerned with continuous space-time, except in the case of Turing, who has considered discrete space. As is well known, the discretization of space and time can yield considerable advantages, i.e. problems become amenable to solution which could not be tackled before. In fact, for the problem of biological development it seems natural to discretize space (in cells) and time (in discrete time observations) as has been forcefully argued by Lindenmayer [5]. Stable string languages of Lindenmayer systems seem a fruitful approach in the context of equilibrium oriented behavior in biological morphogenesis, although obviously some grave simplifications take place.

We would like to think of Turing's approach as the most detailed, Waddington's epigenetic landscape a more general concept, and Thom's theory as the most abstract of the three. In this scheme we would tentatively place the present paper as a new approach, by discretization of space-time, at an intermediate level. We have shown that, by allowing different kinds of rules for cellular behavior, we obtain different classes of stable multicellular patterns.

From the formal language point of view we have investigated the generating power of the stable string operation for Lindenmayer systems, and we have shown that it is equal to the generating power of the operation of intersection with a terminal alphabet, except in the case of context independent L systems. Furthermore, our results show that several of the langauge families in the Chomsky hierarchy can be characterized by classes of highly parallel rewriting systems together with an unusual operation for obtaining languages. Thus we have given a characterization which is structurally completely different from that by the generative grammars.

Acknowledgements. The authors would like to thank the organizers of the January 1974 Workshop on L Systems held at Aarhus, Denmark, and in particular Professor A. Salomaa, for providing a stimulating environment in which this research was begun. We would also like to thank Professors G. T. Herman and J. van Leeuwen for helpful discussions.

REFERENCES

1. D. van Dalen, A Note on some systems of Lindenmayer, <u>Mathematical Systems Theory</u>, 1971, v. 5, 128-140.

2. G. T. Herman and G. Rozenberg, <u>Developmental Systems and Languages</u>, to be published by North-Holland, Amsterdam.

3. G. T. Herman and A. D. Walker, Context Free Languages in Biological Systems, <u>Int. Jour. Computer Mathematics</u>, to appear.

4. A. Lindenmayer, Mathematical Models for Cellular Interactions in Development, Parts I and II, <u>Jour. Theor. Biology</u>, 1968, v. 18, 280-315.

5. A. Lindenmayer, Cellular Automata, Formal Languages and Developmental Systems, in <u>Logic, Methodology and Philosophy of Science IV</u>, edited by P. Suppes et al, North-Holland, Amsterdam, 1973.

6. G. Rozenberg, T0L Systems and Languages, <u>Information and Control</u>, 1973, v. 23, 357-381.

7. A. Salomaa, <u>Formal Languages</u>, Academic Press, 1973.

8. R. Thom, <u>Stabilité Structurelle et Morphogénèse</u>, Benjamin, Reading, Mass., 1972.

9. A. M. Turing, The Chemical Basis of Morphogenesis, <u>Phil. Trans. Royal Society</u>, v. 237, Aug. 1952, 37-72.

10. P. M. B. Vitányi, Growth of Strings in Context Dependent L Systems, in <u>Topics in L Systems</u>, G. Rozenberg and A. Salomaa, eds., Springer Verlag, Berlin, to appear.

11. P. M. B. Vitányi, On the Size of DOL Languages, in Topics in L Systems, G. Rozenberg and A. Salomaa, eds., Springer Verlag, Berlin, to appear.

12. P. M. B. Vitányi, Deterministic Lindenmayer Languages, in preparation.

13. C. H. Waddington, The Strategy of the Genes, Allen and Unwin, London, 1957.

14. A. Walker, Formal Grammars and the Stability of Biological Organisms, Ph.D. Thesis, Department of Computer Science, State University of New York at Buffalo, 1974.

15. A. Walker, Adult Languages of L Systems and the Chomsky Hierarchy, in Topics in L Systems, G. Rozenberg and A. Salomaa, eds., Springer Verlag, Berlin, to appear.