LINDENMAYER SYSTEMS: STRUCTURE, LANGUAGES, AND GROWTH FUNCTIONS

P.M.B. VITÁNYI

LINDENMAYER SYSTEMS: STRUCTURE, LANGUAGES, AND GROWTH FUNCTIONS

VRIJE UNIVERSITEIT TE AMSTERDAM

LINDENMAYER SYSTEMS: STRUCTURE, LANGUAGES, AND GROWTH FUNCTIONS

ACADEMISCH PROEFSCHRIFT

TER VERKRIJGING VAN DE GRAAD VAN
DOCTOR IN DE WISKUNDE EN NATUURWETENSCHAPPEN
AAN DE VRIJE UNIVERSITEIT TE AMSTERDAM,
OP GEZAG VAN DE RECTOR MAGNIFICUS
DR. D.M. SCHENKEVELD,
HOOGLERAAR IN DE FACULTEIT DER LETTEREN,
IN HET OPENBAAR TE VERDEDIGEN
OP VRIJDAG 28 APRIL 1978 TE 13.30 UUR
IN HET HOOFDGEBOUW DER UNIVERSITEIT,
DE BOELELAAN 1105

DOOR

PAUL MICHAEL BÉLA VITÁNYI

GEBOREN TE BUDAPEST

1978

MATHEMATISCH CENTRUM, AMSTERDAM

PROMOTOR : PROF. DR. J.W. DE BAKKER

COREFERENT: PROF. DR. A. SALOMAA

To: THE L JUNGLE
discovered by ARISTID
containing many quaint life FORMS
in the WOODS
like TARZAN
and his friendly gorilla BOLGANI
and other GROWTH.

"But", I said, "Euler showed that hexagons alone cannot enclose a volume". To which the innominate biologist retorted, "That proves the superiority of God over mathematics"....

D'Arcy Thompson as quoted by
W. McCulloch in: *Mysterium Inequitatis of Sinful Man Aspiring to the Place of God*.

PREFACE

Usually, a Ph.D. Thesis reports on some scientific results, and, after it has accomplished its purpose of being a catalyst in the transformation of its author to a Ph.D., it is carved up into one or more pieces which are presented to the scientific community in media with a wider circulation. Here, I have followed the converse course. This work wants to present a unified treatment of research, done by its author, most of which has been published previously in reports, journals and conference proceedings. Where it was necessary to my purpose I have drawn from the work of other investigators. A bibliographical comment accompanies each chapter, disclosing its sources. Whereas it has not been my contention to give a complete account of the mathematical theory of L systems, part of the field seems reasonably covered. The treatment of the subject is self-contained and, hopefully, easy to follow, but it is obvious that a rudimentary knowledge of formal language theory is more or less required from the reader. For instance, a glancing acquaintance with HOPCROFT and ULLMAN [1969], or SALOMAA [1973a], will be helpful. Thus, Section 2.1 on formal grammars is intended as a review of some elementary concepts, and to ensure uniform notation, but not as a substitute for the required background.

In my investigations in L theory I have been helped along by J.W. de Bakker, P.G. Doucet, G.T. Herman, J. van Leeuwen, A. Lindenmayer, H. and J. Lück, G. Rozenberg, A. Salomaa, and W.J. Savitch.

Views expressed on the biological applicability of L systems, and the merits of several attempts in that direction, are for the author's sole responsibility, as are the views on the mathematical and the computer science aspects of the same.

I wish to thank especially Aristid Lindenmayer for his encouragement at the outset of my scientific work; my *promotor* Jaco de Bakker who is also the head of the Computer Science Department at the Mathematical Centre; and my *coreferent* Arto Salomaa. Prof. A. van Wijngaarden, director of the Mathematical Centre, gave me the working environment in which this research could take place.

The technical realization of this monograph was made possible by the speedy and excellent typing of the manuscript by Linda Brown and Lenie Rijs, the drawing of figures by Tobias Baanders, and the printing by Dick Zwarst, Jan Schipper, Jan Suiker, Etienne Michel and Jos van der Werf; all at the

Mathematical Centre.

The front cover was designed by Tobias Baanders, using a fragment of Botticelli's allegory *La Primavera* depicting *Spring escaping from Winter*. The fragment symbolizes fertility and the beginning of growth of plants which is associated with that season.

CONTENTS

CHAPTER 1

INTRODUCTION

There have been many attempts to describe the process of biological development by mathematical models. Here we shall deal only with aspects of the mathematical models of development first advocated by LINDENMAYER [1968a,b]. These models are called L systems, after their originator. An L system is a string rewriting system, where each letter of a string symbolizes the presence in that position of a cell of a certain type or state, and the whole string symbolizes a filament of cells. Time is assumed to be discrete, and, in between two consecutive moments of time, say between t and $t+1$, each letter of a string is rewritten as a string which may be empty. This rewriting may depend on the m left and n right neighbors of the letter concerned. The resultant string at time $t+1$ consists of the concatenation of the strings resulting from the rewriting of the individual letters. By repeating this process, we obtain a sequence of strings symbolizing the developmental history of the modeled filamentous organism. Various embellishments, with or without biological interpretation, of this basic model can be contrived, as we shall see in the sequel.

The subject has caused much recent activity from the side of mathematicians and formal language theorists (see the bibliography), but has not yet been applied substantially by workers in the field of developmental biology. The fact that formal machinery developed in theoretical computer science, or in mathematics in general for that matter, is not applied to any great extent is, though regrettable, not unusual. For the case under consideration, the mathematical questions considered in the formal study of L systems often have no interest, or even interpretation, for the practicing biologist. A fundamental difficulty might be that the basic assumptions of the model do not allow adequate modeling of certain biological phenomena at all. This might be remedied by adding features or changes ad hoc, a procedure which has been followed in nearly all existing L system models for concrete biological phenomena. As a consequence, the sophisticated mathematical theory, which has been erected on the firm fundaments of the basic model, then comes apart and does not hold for the featured model. However, the descriptive convenience of L systems has already been used in several biological modeling efforts.

The most successful seems that by H. LŪCK [1975], H. LŪCK and J. LŪCK [1976], and J. LŪCK [1977], who have used PD0L systems to describe the development of (filamentous) blue-green algae. Their model enables them to determine key parameters of algae from observations on only one or two stages of development, whereas the direct experimental method requires the rather lengthy and laborous tracing of indivudual cell histories. The model, which makes extensive use of the theory of D0L growth functions (see Ch. 4), and locally catenative systems (see Section 3.1.2.2), seems to be the first example of an operational technique based on L systems. Moreover, its theoretical potential has not yet been exhausted, and the organisms studied conform relatively well to the basic assumptions of L systems. A substantial number of descriptive models using L systems have been investigated by means of the powerful simulation program CELIA designed by BAKER and HERMAN [1970] (see also HERMAN and LIU [1973]). CELIA has proved to be a practical simulation tool as is borne out by various studies: BAKER and HERMAN [1972a,b] on heterocyst formation in blue-green algae; FRIJTERS and LINDENMAYER [1974, 1976] and FRIJTERS [1976] on inflorescences of *ASTER*; STAFLEU [1973] on the branching pattern of barley root; HERMAN, LIU, ROWLAND and WALKER [1974] on patterns on shells of molluscs (see also HERMAN and ROZENBERG [1975, Chs. 16, 18]); VEEN and LINDENMAYER [1973] and HELLENDOORN and LINDENMAYER [1974] on phyllotaxis; HERMAN and SCHIFF [1975] on regeneration of *HYDRA*; and HOGEWEG and HESPER [1974] and HOGEWEG [1976] on biological pattern analysis.

As a model for biology, L systems have very appealing features, which obviously have their counterparts in reality. It has been claimed, however, that, e.g., the influences of concentrations of chemicals or enzymes can be modeled by increasing the number of states (of the basic unit), and decreasing the length of the time step (of the basic transition). This may be right in principle, but obscures what is going on. We need, e.g., 1000^{10} states to account for the influences of 10 different substances with concentrations graded 1-1000. Such things are easy to describe as difference equations in a simulation program like CELIA, but the theoretical model can be manipulated, and yields significant results, only if the (relatively small) number of cell types with stereotype behavior (corresponding more or less to the basic genetical differentiation) is treated apart from influences of substances like chemicals or enzymes which act like, e.g., synchronous inhibitors. For instance, if the cells of an organism are essentially interactionless, but we model the influences of extracellular agents like enzymes by making the model a very complicated context sensitive one, we violate the principles on which the organism operates, and lose adequacy of the model and a host of mathematical results which are applicable. For a more extensive discussion along these lines see Chapter 5.

As will be readily noticed, the approach taken to model development is by discretizing space and time. This is natural in the context of biological development: we discretize space in discrete cells and time in discrete time observations. The discretizing of time is usually justified by three reasons. Pragmatically: the mathematics of the subject becomes more accessible; for empirical reasons: in practice we can only make discrete time observations; and since we assume a finite set of states for each cell. The justification for assuming a finite set of states for each cell is that there are usually threshold values for parameters that determine the behavior of a cell. Thus with respect to each of these parameters, it suffices to specify two conditions of a cell: "below threshold" and "above threshold", although the parameters themselves may have infinitely many values. Even where such a simple minded scheme is insufficient, it is argued, it usually is possible to approximate the infinite set of values by a sufficiently large finite set of values, without any serious detriment to the accuracy of the developmental model. Although seemingly plausible, it will be clear that this reduction of continuous parameters to a finite state set will lead to serious problems in many cases, like the ones related to the modeling of the influences of chemical concentrations sketched above. Notice also, that the models we treat here apply only to organisms which consist of autonomous segments or compartments (living cells are naturally such compartments). By "autonomy" is meant primarily independence in the hereditary sense, in the sense that cells are known to carry their own genetic instructions and pass them on to their daughters, but also metabolic and functional independence. The existence of compartments enables us to describe such organisms as automata arrays, from which the formalism of L systems is easily derivable. But a consequence of this requirement is that these models cannot be directly applied to subcellular growth processes.

Another restriction of the models lies in their one-dimensional nature. This implies that at the present time they are only applicable to filamentous organisms. Furthermore, the models are based on the assumption that the relative position of cells (or compartments) cannot change during growth, and neither can the neighborhoods of the daughter cells be different from those of the mother cells. These assumptions are in agreement with development in plants but not necessarily with that in animals, where cells may slide past each other in the course of growth.

From the mathematical viewpoint L systems are more appealing than the usual sequential rewriting systems such as formal grammars of the Chomsky type. In the context free case the rewriting is a homomorphism or a finite substitution; in the context sensitive case it is a generalized sequential machine mapping. This makes the problems we usually consider in formal language theory more amenable to ordinary mathematical treatment, since the action is not localized but global over the entire string. As a result of this, new structures have been developed and new problems arisen, of which the analysis enriches formal language theory and, amongst others, AFL theory (see, e.g., SALOMAA [1974], van LEEUWEN [1974]). Research in new machine models (van LEEUWEN [1974], ENGELFRIET, SCHMIDT and van LEEUWEN [1977]) and complexity theory (JONES and SKYUM [1976; 1977a, b, c, d], van LEEUWEN [1975a, b, c; 1976], and SUDBOROUGH [1977]), have benefitted from L system theory. Although formerly the study of L systems has been almost exclusively biologically motivated, the underlying structure is today recognized in a growing variety of problems in computer science, ranging from pure formal language theory to more applied subjects, while at the same time the analysis has led to interesting mathematical problems.

The organization of this monograph is as follows. In Chapter 2 we give some formal definitions and preliminaries. Section 2.1 reviews some pertinent concepts of formal language theory in the range of HOPCROFT and ULLMAN [1969], mainly in order to review some concepts and standardize notation. Section 2.2 supplies basic definitions of L system theory. Chapter 3 treats L systems, -sequences and -languages. Starting in Section 3.1 with a structural treatment of deterministic context free L systems (D0L systems) we study, for instance, how restrictions on the rewriting rules affect the associated sequences and languages (local versus global characteristics).

We subsequently treat the connection between DOL sequences and locally catenative sequences, i.e., where a string is obtained from earlier strings in the sequence by a fibonacci-like formula. In Section 3.2 and 3.3 we obtain a rather complete picture of the power of the various types of context sensitive L systems using nonterminals, homomorphic mappings etc., and we mold the (in)famous LBA problem from automata theory in the form of whether or not a trade-off is possible between context and rewriting rules in L systems. Section 3.4 is concerned with stable string languages of L systems, i.e., languages consisting of those strings produced by a given system which are invariant under the rewriting rules. In Section 3.5 we study some aspects of certain variations of L systems in relation with problems of regeneration. In particular a form of the French Flag problem (see e.g. WOLPERT [1968]) is treated. Chapter 4 is concerned with growth functions of L systems. The growth function of a (deterministic) L system relates the length of the i-th derived word in the sequence with i. It is for this subject in L system theory that the most extensive claims for biological relevance have been made. This is not surprising, since a large part of the literature on developmental biology is concerned with the changes in size and weight of a developing organism as a function of the elapsed time. We treat some of the analytical theory of DOL growth functions in Section 4.1; relations between restrictions on the rewriting rules, and the overall form of the derived growth function for the DOL case, in Section 4.2; and the theory of context sensitive growth functions in Section 4.3. In Chapter 5 we discuss the adequacy of the theory of Chapter 4 to model biological phenomena, and, as a result, modify some basic assumptions to obtain a more realistic model. It then appears that, without undue difficulties, we are able to derive the sigmoidal growth curves, occurring regularly in developmental biology studies, which were not realizable with the theory of Chapter 4.

Chapter 3 and 4 are more or less independent, and contain the necessary definitions in so far as they are not supplied in Chapter 2. Chapter 5 presupposes Chapter 4. Section 3.1.2.1 ties in with Section 4.2. Chapter 6 consists of an epilogue, in which we evaluate the work presented in this monograph. We chose to do so in an epilogue, rather than in the introduction, because in that way some familiarity with the subject could be assumed and we were not hampered by the need to explain too many concepts or by the need to mince words. The reader who wants to have a more or less informal preview and assessment of the research covered by the coming chapters can proceed there.

Our investigations do not lead up to one or a few main results, but, more textbook-like, cover part of the field. Some results or topics presented are, for a variety of reasons, more interesting than others. The two main themes are language classification (Ch. 3) and growth functions (Ch. 4 and 5). The techniques used are mainly combinatorial.

It should be noted here, before starting with the meat of the work, that the discussion about the usefulness or relevance of L system theory for practicing biologists, or biology in general, invokes heated debates. A protagonist is LINDENMAYER [1975], and some careful criticism is contained in DOUCET [1975, 1976]. We will touch the subject somewhat in Chapter 5. It ought to be stressed that the fact, that only more superficial aspects of an extensive mathematical theory have been applied directly (as is the case with L system theory), is not an exception, but rather the converse. Theoretical computer science itself is usually applicable only in a superficial way, like when computer languages are designed which have context free (or anyway easily parsable) grammars, but the extensive mathematical theory which has been erected is not used widely even where it is superior, operative and applicable. De MILLO, LIPTON and PERLIS [1977] discuss this phenomenon as related to theoretical investigations concerning the proving of correctness of programs. Large areas in the analysis-of-algorithms theory suffer the same lack of being applicable or used in practice. Such pessimistic considerations, however, do not influence the essential mathematical beauty of the results or insights gained in the nature of mathematical structures, nor do they preclude later, possibly quite unrelated, applications of the developed theories.

CHAPTER 2.

DEFINITIONS AND PRELIMINARIES

We shortly review the more needed formal language theoretical machinery and the most generally used concepts of L system theory. For formal language theory and automata theory we use notation and so forth from HOPCROFT and ULLMAN [1969], and for L system theory we sometimes depart somewhat from the (nonuniform) notation in the literature, so as to obtain some unity in our treatment.

2.1. FORMAL GRAMMARS

Formal grammars originate from CHOMSKY [1957], who introduced them for largely linguistical reasons. In essence, a formal grammar is a string rewriting system which transforms strings into strings. The purpose is to define in a finite way (by means of the grammar) an infinite number of strings (the language), such that the particular definition of each string (the derivation) yields some structural information about it. For proofs of lemmas and theorems in this section consult any textbook on the subject, e.g., HOPCROFT and ULLMAN [1969].

We denote generally, with or without indices, *symbols* (equivalently, *letters*) by a,b,c,...; *strings* (or *words*) of letters by u,v,w,x,y,z or $\alpha,\beta,\gamma,\ldots,\omega$; sets of letters *(alphabets)* by A,B,U,V,W; sets of strings (*languages*) by L,X,Y,Z; numbers by i,j,k,ℓ,m,n,p,q,r,s,t. We will not always strictly adhere to these conventions, but then the context will allay confusion. The set of natural numbers $\{0,1,2,\ldots\}$ is denoted by $\mathbb{N}$; the set of reals by $\mathbb{R}$; and the set of positive reals by $\mathbb{R}_+$. If X is a set then $\#X$ denotes the *cardinality* of X; if x is a string then $\ell g(x)$ or $|x|$ denotes the *length* (number of occurrences of letters) of x.

The set of all strings over some finite alphabet of letters W is denoted by W^*, e.g., $W^* = \{\lambda,a,b,aa,ab,ba,bb,aaa,\ldots\}$ for $W = \{a,b\}$, and λ denotes the *empty string* (the string consisting of no letters at all).

W^* is customarily called the free monoid finitely generated by W. If X and Y are two sets then

$$
\begin{aligned}
X \cup Y &= \{x \mid x \in X \text{ or } x \in Y\},\\
X \cap Y &= \{x \mid x \in X \text{ and } x \in Y\},\\
X - Y &= \{x \mid x \in X \text{ and } x \notin Y\},\\
XY &= \{w \mid w = xy \text{ and } x \in X \text{ and } y \in Y\},\\
X^i &= \begin{cases} \{\lambda\} & \text{if } i = 0 \\ XX^{i-1} & \text{if } i > 0, \end{cases}\\
X^* &= \bigcup_{i=0}^{\infty} X^i,\\
X^+ &= \bigcup_{i=1}^{\infty} X^i,
\end{aligned}
$$

and $\emptyset$ denotes the *empty set.* The class of *regular* sets over an alphabet W is formed as follows:

(i) $\emptyset$ and the singleton sets of elements in W are regular sets.

(ii) If R_1 and R_2 are regular sets then so are $R_1 \cup R_2$ and R_1R_2.

(iii) If R is a regular set then so is R^*.

(iv) Only sets formed by application of (i) - (iii) are regular sets.

DEFINITION 2.1. A *generative* or *formal grammar* is an ordered quadruple $G = \langle V_N, V_T, S, P\rangle$ where V_N and V_T are finite nonempty alphabets, $V_N \cap V_T = \emptyset$, $S \in V_N$, and P is a finite set of ordered pairs (α,β) such that β is a word over the alphabet $V = V_N \cup V_T$ and α is a word over V containing at least one letter of V_N. The elements of V_N are called *nonterminals* and those of V_T *terminals*; S is called the *start symbol.* Elements (α,β) of P are called *rewriting rules* or *productions* and are written $\alpha \to \beta$.

P induces a relation "$\Rightarrow$" on V^* as follows. v' is *directly produced* from v: $v \Rightarrow v'$ if there are $\gamma_1,\gamma_2,\alpha,\beta \in V^*$ such that $v' = \gamma_1\beta\gamma_2$, $v = \gamma_1\alpha\gamma_2$, and there is a $\alpha\to\beta \in P$. The transitive reflexive closure of $\Rightarrow$ is $\overset{*}{\Rightarrow}$ and the transitive irreflexive closure of $\Rightarrow$ is $\overset{+}{\Rightarrow}$. If $v \overset{*}{\Rightarrow} v'$ or $v \overset{+}{\Rightarrow} v'$ we say v *produces* or *derives* v'. If $v_0 \Rightarrow v_1 \Rightarrow \dots \Rightarrow v_n$, we write $v \overset{(n)}{\Rightarrow} v'$, and say v_0 *produces* v_n *in* n *steps*. The *language* produced by G is defined by

$$L(G) = \{v \in V_T^* \mid S \overset{*}{\Rightarrow} v\}.$$

A *family* of languages is a nonempty set of languages closed under isomorphism (with respect to the operation of concatenation in this case), i.e., renaming of letters.

THEOREM 2.2. *The family of languages* $\{L \mid L = L(G)$ for some generative grammar $G\}$ *equals the class of recursively enumerable languages.*

According to Church's thesis, the class of recursively enumerable languages is the largest class of sets obtainable by effective means. By successive restrictions on the form of the production rules, we obtain successively restricted classes of grammars.

DEFINITION 2.3. A grammar $G = \langle V_N, V_T, S, P\rangle$ is of *type i* if the restrictions (i) on P, as given below, are satisfied.

(0) No restrictions.

(1) Each production in P is of the form $\alpha_1 X \alpha_2 \to \alpha_1 v \alpha_2$, $\alpha_1, \alpha_2 \in V^*$, $X \in V_N$ and v is a *nonempty* word over V, with the possible exception of the production $S \to \lambda$ whose occurrence in P implies, however, that S does not occur on the righthand side of any other production in P.

(2) Each production in P is of the form $X \to \beta$ where $X \in V_N$ and $\beta \in V^*$.

(3) Each production in P is of one of the two forms $X \to Y\alpha$ or $X \to \alpha$ where $X, Y \in V_N$ and $\alpha \in V_T^*$.

We call the grammars of types 0,1,2, and 3, *recursively enumerable, context sensitive, context free* and *regular*, respectively. We denote the corresponding families of languages by RE, CS, CF and REG.

THEOREM 2.4. REG *equals the class of regular sets.*

THEOREM 2.5. $\mathrm{REG} \subset \mathrm{CF} \subset \mathrm{CS} \subset \mathrm{RE}$ *where* "$\subset$" *denotes strict inclusion.*

These (by inclusion) nested language families make up the so-called *Chomsky hierarchy*. We call languages in the difference X-Y, Y and X in sequence as in Theorem 2.5, *strictly* X where Y is understood.

EXAMPLE 2.6.

$L_0 = \{1^{f(n)} \mid n \geq 0\} \in \mathrm{RE} - \mathrm{CS}$, if $f\colon \mathbb{N} \to \mathbb{N}$ enumerates some nonrecursive, but recursively enumerable, set.

$L_1 = \{a^n b^n c^n \mid n \geq 1\} \in \mathrm{CS} - \mathrm{CF}$.

$L_2 = \{a^n b^n \mid n \geq 1\} \in \mathrm{CF} - \mathrm{REG}$.

$L_3 = \{a^n \mid n \geq 1\} \in \mathrm{REG}$.

$L_4 = \{a^n | n \text{ is a prime number}\} \in CS - CF$, to give a feeling of the power of context sensitivity as opposed to context freeness.

Above it was shown how the four main language families of the Chomsky hierarchy are derived by classes of generating devices, viz., by suitable restrictions on the form of the production rules in grammars. They can also be characterized by accepting devices, i.e., classes of machines which accept exactly the languages generated by a class of grammars. By acceptance we we mean, that, if L is the language accepted by a machine M then M enters an accepting configuration after reading a word v iff $v \in L$. So RE is accepted by *Turing machines,* CS is accepted by *Linear Bounded Automata,* CF is accepted by *Pushdown Automata* and REG is accepted by *Finite Automata.* When, where and if, necessary we shall introduce these devices.

2.2. LINDENMAYER SYSTEMS

As we have seen in the previous section, a generative grammar is a sequential rewriting system, i.e., in each production step part of the string is rewritten. L systems are rewriting systems where we rewrite all letters in a string simultaneously in each production step. Moreover, they have no terminal symbols in the sense of formal grammars as defined above.

EXAMPLE 2.7. The production rule $a \to aa$ yields, if we start with the string a, the string sequence $a, aa, a^4, a^8, a^{16}, \ldots$ and the produced language is $\{a^n | n = 2^i,\ i \in \mathbb{N}\}$. This system is context free (each letter is rewritten independent of the context in which it occurs) and deterministic (letters can be rewritten in but one way). In a context sensitive L system the letters in a string are rewritten, by the production rules, according to the context in which they occur. In an (m,n) L system this context consists of the m left- and n right letters of the letter to be rewritten, and we rewrite letters according to the production rules which are applicable to the letter with its m letter left- and n letter right context in the string before the rewriting. Formally,

DEFINITION 2.8. An (m,n) L *system* is a triple $G = \langle W,P,w\rangle$ where W is a finite nonempty *alphabet,* $w \in W^+$ is the *initial string,* and

$$P \subseteq \Big(\bigcup_{i=0}^{m} W^i \times W \times \bigcup_{i=0}^{n} W^i\Big) \times W^*$$

is a finite set of production rules. We write an element of P also as $(u,a,v) \to \alpha$ where $u \in \bigcup_{i=0}^{m} W^i$, $a \in W$, $v \in \bigcup_{i=0}^{n} W^i$ and $\alpha \in W^*$.

We derive strings by the system as follows. $\to$ induces a relation $\Rightarrow$ on W^* defined by

$$a_1a_2\ldots a_k \Rightarrow \alpha_1\alpha_2\ldots\alpha_k \qquad a_1,a_2,\ldots,a_k \in W \text{ and } \alpha_1,\alpha_2,\ldots,\alpha_k \in W^*,$$

if

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

for all i, $1 \le i \le k$, where we take $a_j = \lambda$ whenever $j < 1$ or $j > k$. If $v_0 \Rightarrow v_1 \Rightarrow v_2 \Rightarrow \ldots \Rightarrow v_\ell$ for some $v_0,v_1,\ldots,v_\ell \in W^*$ we write $v_0 \overset{(\ell)}{\Rightarrow} v_\ell$ and say v_0 *derives* or *produces* v_ℓ in ℓ *steps*. As usual, $\overset{*}{\Rightarrow}$ and $\overset{+}{\Rightarrow}$ are the transitive reflexive closure and the transitive irreflexive closure of $\Rightarrow$, respectively. (I.e., $\overset{*}{\Rightarrow} = \bigcup_{i=0}^{\infty} \overset{(i)}{\Rightarrow}$ and $\overset{+}{\Rightarrow} = \bigcup_{i=1}^{\infty} \overset{(i)}{\Rightarrow}$.)

A *sequence* produced by G is a sequence $v_0,v_1,\ldots,v_t,\ldots$ where $v_i \Rightarrow v_{i+1}$ for all $i \ge 0$. In case G is deterministic the produced sequence is unique relative to v_0, and if, moreover, $v_0 = w$ then the produced sequence is called the *string sequence* S(G) associated with G. The *language* produced by G is:

$$L(G) = \{v \in W^* | w \overset{*}{\Rightarrow} v\}.$$

We subscript the relations $\to$, $\Rightarrow$, $\overset{*}{\Rightarrow}$, $\overset{+}{\Rightarrow}$, $\overset{(i)}{\Rightarrow}$ with the appropriate identifiers when neccessary. Similarly to the generative grammars in the previous section we obtain classes of L systems by imposing restrictions on the form of the production rules.

<u>DEFINITION 2.9</u>. Let G = <W,P,w> be an (m,n) L system.

(i) Without any restriction G is called *context sensitive*, or interacting, and the corresponding class of L systems is denoted as IL *systems*. With fixed m and n we call the corresponding class the class of (m,n)L *systems*.

(ii) If $(u,a,v) \to \alpha \in P$ implies that $u,v = \lambda$ then G is *context free*, or interactionless, and the corresponding class of systems is denoted as 0L *systems*. For ease of notation we write rules in P as $a \to \alpha$.

(iii) If for each $(u,a,v) \to \alpha \in P$ either always $v = \lambda$ or always $u = \lambda$ then G is *left*- or *right context sensitive*, and the corresponding class of systems is denoted as I_LL- or I_RL *systems*. For ease of notation we write rules in P as $(u,a) \rightarrow \alpha$ or $(a,v) \rightarrow \alpha$, respectively.

(iv) If $(u,a,v) \rightarrow \alpha$ and $(u,a,v) \rightarrow \alpha'$ imply that $\alpha = \alpha'$ then G is called *deterministic*, and we indicate this property by prefixing a "D" in the denotation of the class of systems. We also denote the set of production rules P by a function δ, and write $\delta(u,a,v) = \alpha$ for $(u,a,v) \rightarrow \alpha \in P$. We extend δ to W^* by defining $\delta(v) = v'$ if $v \Rightarrow v'$, $v,v' \in W^*$. δ^i is defined as the i-fold composition of δ: $\delta^0(v) = v$ and $\delta^i(v) = \delta(\delta^{i-1}))$, $i \geq 1$.

(v) If $(u,a,v) \rightarrow \alpha$ implies $\alpha \neq \lambda$ the system is *nonerasing* or *propagating* and we denote this property by prefixing a "P" in the denotation of the class.

Hence we have, e.g., PDIL systems, D0L systems, D(m,n)L systems, DI_LL systems etc. The following notation is standard throughout the literature and partly follows from above.

(0,0)L systems $\equiv$ 0L systems,
(1,0)L systems or (0,1)L systems $\equiv$ 1L systems,
(1,1)L systems $\equiv$ 2L systems.

The notion of L systems has been extended to the important table L systems.

DEFINITION 2.10. A *table* L *system with* q *tables*, T_qL *system*, is a triple $G = \langle W,P,w\rangle$ where $P = \{P_1,P_2,\ldots,P_q\}$, W and w are as before, and P_i, $1 \leq i \leq q$, is as P in Def. 2.8. Therefore, a table (m,n)L system is a triple $G = \langle W,P,w\rangle$ with $P = \{P_1,P_2,\ldots,P_q\}$ such that for each i, $1 \leq i \leq q$, $G_i = \langle W,P_i,w\rangle$ is an (m,n)L system.

Strings are derived in a table L system $G = \langle W,P,w\rangle$ as follows.

$$a_1a_2\ldots a_k \underset{G}{\Rightarrow} \alpha_1\alpha_2\ldots\alpha_k, \qquad \begin{array}{l} a_1,a_2,\ldots,a_k \in W \text{ and} \\ \alpha_1,\alpha_2,\ldots,\alpha_k \in W^*, \end{array}$$

if there is a *table* P_i in the set *of tables* P such that

$$a_1a_2\ldots a_k \underset{G_i}{\Rightarrow} \alpha_1\alpha_2\ldots\alpha_k.$$

$\overset{*}{\underset{G}{\Rightarrow}}$ and $\overset{+}{\underset{G}{\Rightarrow}}$ are the usual closures of $\underset{G}{\Rightarrow}$. The *language* generated by G is defined by $L(G) = \{v \mid w \overset{*}{\underset{G}{\Rightarrow}} v\}$, etc. G is a XYT_qZL system if, for $P = \{P_1,P_2,\ldots,P_q\}$ each G_i, $1 \leq i \leq q$, is an XYZL system. E.g., PDT_21L systems, T0L systems etc.

(No subscript on T means that $q \geq 1$, no T_q at all means that $q = 1$. E.g., PDT_1OL systems are PDOL systems.) The family of languages generated by XYT_qZL systems is denoted as XYT_qZL.

DEFINITION 2.11. A *semi* XYT_qZL system is a XYT_qZL system without the initial string.

We can squeeze languages out of L systems in various ways. One way is (as above) to consider all strings generated from the initial string: the *pure* L language of the system. By dividing the alphabet into a set of terminals and a set of nonterminals, we can consider the language consisting of all strings over the terminals occurring in the pure L language. Such a language is called an *extension* language, since the terminal-nonterminal mechanism extends the generating power of a class of L systems. Another device is to take a homomorphism*) of a pure L language or extension language. A third method we shall meet is to consider the *stable string* language of an L system. That is, the set of all strings, occurring in the pure L language, which are invariant under the rewriting rules. Accepting devices for families of L languages, similar to machine type characterizations of families of languages produced by classes of generative grammars like in Section 2.1, are not treated in this work. They have, however, been studied in van LEEUWEN [1974], ROZENBERG [1974] and SAVITCH [1975].

2.3. BIBLIOGRAPHICAL COMMENTS

Generative grammars were introduced by CHOMSKY [1957]. The concepts and results in Section 2.1 are treated in any textbook on the subject, like HOPCROFT and ULLMAN [1969] or SALOMAA [1973a]. Lindenmayer systems were

---

*) By homomorphisms we will mean monoid homomorphisms which are mappings between monoids in which the operation is concatenation. More precisely, the free monoid S finitely generated by a finite alphabet W is W^*, i.e., if $x,y \in S$ then so does xy. The operation of concatenation is associative; $(xy)z = x(yz)$ and the identity element of S is the *empty word* λ: the word with no letters. If W^* and V^* are two monoids and $h: W^* \to V^*$ is a homomorphism between them then $h(\lambda) = \lambda$, $h(xy) = h(x)h(y)$ for all $x,y \in W^*$. We extend the concept of homomorphisms to sets by defining $h(L) = \{h(w) \mid w \in L\}$ for each subset $L \subseteq W^*$.

proposed by LINDENMAYER [1968a,b] and a first formal language type of treatment was given by HERMAN [1969] and van DALEN [1971]. Table L systems were introduced by ROZENBERG [1973a] and the concept of stable string languages of L systems is due to WALKER [1974a,b]. Extension languages of L systems were first introduced by van LEEUWEN (unpublished) who, with an alternative interpretation of the concept, called them restriction languages. A textbook covering most of the research (done in collaboration with the authors) in L system theory up to 1972 - 1973 is HERMAN and ROZENBERG [1975]. Collections of research papers and tutorials presented at L system conferences are contained in the PROCEEDINGS of an Open House in Unusual Automata Theory [1972], PROCEEDINGS of the IEEE Conference on Biologically Motivated Automata Theory [1974], ROZENBERG and SALOMAA [1974] and LINDENMAYER and ROZENBERG [1976].

CHAPTER 3

L SYSTEMS, SEQUENCES, AND LANGUAGES

According to the definitions in Section 2.2, L systems are but a type of string rewriting systems. In the theory developed on the basis of string rewriting systems, such as formal grammars generating languages, the following problems are usually studied:

- *Classification.* What are the inclusion relations between the new classes of languages and the various known ones (e.g. those in the Chomsky hierarchy)?
- *Closure properties.* Is the class closed under union, intersection, concatenation, homomorphisms of various types, and other operations?
- *Decision problems.* Is it decidable whether a given word is generated by a given grammar? (I.e., the *membership problem.*) Is it decidable whether the languages generated by two grammars are equal? And so on.
- *Characterization.* Given a class of grammars; are there properties by which a language can be identified as (not) belonging to the corresponding class of languages?

In this chapter we investigate some of these formal language oriented aspects of L systems, but we will also encounter concepts which do not occur in conventional formal language theory. In Section 3.1, we look at structural aspects of deterministic context free L systems (DOL's). More in particular, we will study the relation between local properties (e.g., the form of the production rules) and global properties (e.g., of the produced sequences and languages). In Sections 3.2 and 3.3, we obtain a rather complete picture of the generating power (language generating capacity) of various subclasses of context sensitive L systems with and without tables. Section 3.4 is concerned with the set of those strings produced by an L system which are invariant under the rewriting rules, i.e., the strings which are necessarily rewritten as themselves. A variation on the basic model, the context variable L system, is the subject of Section 3.5. There we shall look at some simple regenerating structures and solve a form of the French Flag problem.

3.1. DOL SYSTEMS

About the simplest type of L system you can meet is the DOL system. Essentially, it consists of a homomorphism on a finitely generated free monoid. For that very reason, its mathematical theory is quite extensive and leads to interesting mathematical byways (see also Section 4.1). Formally then, a DOL system $G = \langle W,\delta,w\rangle$ consists of a finite nonempty alphabet W, a total mapping $\delta\colon W \to W^*$ and an initial string $w \in W^*$. We denote both δ and its extension to a homomorphism on W^* by δ. Clearly, $S(G) = w,\delta(w), \delta^2(w),\ldots$ and $L(G) = \bigcup_{i=0}^{\infty} \{\delta^i(w)\}$, where $\delta^0(w) = w$ and $\delta^i(w) = \delta(\delta^{i-1}(w))$ for $i \geq 1$.

<u>EXAMPLE 3.1</u>. Let $G = \langle\{a,b\},\{\delta(a)=b,\delta(b)=ab\},a\rangle$. Then $S(G) = a,b,ab,bab, abbab,\ldots$, and in general $\delta^i(a) = \delta^{i-2}(a)\delta^{i-1}(a)$, $i \geq 2$. That is, $S(G)$ is the Fibonacci string sequence of KNUTH [1969a, Exercise 1.2.8-36], and $\ell g(\delta^i(a))$ is the i-th fibonacci number.

<u>EXAMPLE 3.2</u>. Let $G = \langle\{a\},\{\delta(a)=aa\},a\rangle$. Then $L(G) = \{a^{2^n} \mid n \geq 0\}$.

The most intriguing question, asked about L systems, used to be: is it decidable whether $L(G) = L(G')$ for given DOL systems G, G'. This matter was settled affirmatively by ČULIK and FRIS [1977a,b]. A seemingly related question, about whether it is decidable that a given DOL system has the locally catenative property (a generalization of the property we met in the DOL system of Example 3.1), is still open. In Section 3.1.2 we shall consider it in more detail.

3.1.1. DOL LANGUAGES

In the first part of this section we establish, by a simple combinatorial argument, necessary and sufficient conditions (with respect to the production rules) under which the language, generated by a deterministic context free Lindenmayer system, is finite. These conditions yield sharp bounds on the size of such a language, which depends on the size of the alphabet and the interrelations of the production rules. Furthermore, a feasible decision procedure for the membership question is provided, and we solve the problems of what is the minimum sized alphabet over which there is a deterministic context free Lindenmayer language of size n and, conversely, what is the maximum sized finite deterministic context free Lindenmayer language over an alphabet of m letters. The solutions to these last two problems provide us with some number theoretic functions, interesting in their own right, which form the object of study in the second part of this section. We derive several properties, interrelations, and asymptotic approximations to these functions.

We classify the letters in W with respect to the homomorphism δ as follows.

DEFINITION 3.3. A letter $a \in W$ is *mortal* ($a \in M$) if $\delta^i(a) = \lambda$ for some i; *vital* ($a \in V$) if $a \notin M$; *recursive* ($a \in R$) if $\delta^i(a) \in W^*\{a\}W^*$ for some $i > 0$; *monorecursive* ($a \in MR$) if $\delta^i(a) \in M^*\{a\}M^*$ for some $i > 0$; *expanding* ($a \in E$) if $\delta^i(a) \in W^*\{a\}W^*\{a\}W^*$ for some i; *accessible* from a string $v \in W^*$ ($a \in U(v)$) if $\delta^i(v) \in W^*\{a\}W^*$ for some $i > 0$. We subscript to identify the DOL system concerned when necessary.

The global properties of (sequences of) strings produced by a DOL system, such as the "patterns" (characteristic substrings) occurring, are essentially due to the recursive letters and the derivational relations between them. For instance, a language like $\{a^{2^n}b^{2^n}c^{3^n} \mid n \geq 0\}$ can only be produced by the DOL system.

$$G = \langle\{a,b,c\},\{\delta(a)=a^2,\delta(b)=b^2,\delta(c)=c^3\},abc\rangle$$

From the produced patterns it can readily be deduced, that the system has to contain 3 expanding recursive letters with no derivational relations between each other at all. We shall see in the sequel (Section 3.1.2) that types of growth functions, the locally catenative property, regularity, and context freeness, depend to a very large extent on the recursive letters and the accessibility between them: properties of recursive letters govern the relation between local properties of DOL systems and global properties of the derived string sequences.

We define an equivalence relation $\sim$ on R by $a \sim b$ if $a \in U(b)$ and $b \in U(a)$. Hence $\sim$ induces a partition of R in equivalence classes $[a] = \{b \in R \mid b \sim a\}$ and

$$R/\sim \; = \{[a] \mid a \in R\}$$

LEMMA 3.4. *There is an algorithm to determine* $U(a)$ *for all* $a \in W$.

PROOF. Define for each $a \in W$ a sequence of nested sets as follows.

$$U_1(a) = \{b \mid \delta(a) \in W^*\{b\}W^*\}$$

$$U_{i+1}(a) = U_i(a) \cup \{b \mid \delta(c) \in W^*\{b\}W^* \text{ and } c \in U_i(a)\}.$$

By observing that

(i) $U_i(a) \subseteq U_{i+1}(a) \subseteq W$ for all $i \geq 1$, and

(ii) if $U_{k+1}(a) = U_k(a)$ for some k then $U_{k+j}(a) = U_k(a)$ for all $j \geq 0$

we obtain: there is a $k \leq \#W$ such that $U_k(a) = U(a)$. □

EXAMPLE 3.5. Let $S = \langle\{a,b,c,d\}, \{\delta(a) = cd, \delta(b) = a^2bc, \delta(c) = c, \delta(d) = \lambda\}\rangle$ be a semi DOL.

$U_1(a) = \{c,d\}$	$U_2(a) = U_1(a)$	$U(a) = \{c,d\}$
$U_1(b) = \{a,b,c\}$	$U_2(b) = W$	$U(b) = W$
$U_1(c) = \{c\}$	$U_2(c) = U_1(c)$	$U(c) = \{c\}$
$U_1(d) = \emptyset$		$U(d) = \emptyset$

COROLLARY 3.6. *We can determine* M, V, R, MR *and* E *as follows.*

(i) $R = \{a \in W \mid a \in U(a)\}$.

(ii) $V = \{a \in W \mid U(a) \cap R \neq \emptyset\}$

(iii) $M = W - V$.

(iv) $MR = \{a \in R \mid \forall b \in U(a)\ [\delta(b) \in M^*RM^* \cup M^*]\}$

(v) $E = \{a \in R \mid \exists b \in U(a)\ [\delta(b) \in W^*[a]W^*[a]W^*]\}$

Hence all of these sets can be obtained in $\mathcal{O}(\#W)$ steps. (Corollary 3.6(v) will be justified by Lemma 4.20).

LEMMA 3.7. *Let* $G = \langle W,\delta,w\rangle$ *be a* DOL *system. If there is an* i *and a* $b \in R-MR$ *such that* b *is a subword of* $\delta^i(w)$ *then* $L(G)$ *is infinite.*

PROOF. If $b \in R-MR$ then there is a $j \leq \#R$ and a $c \in V$ such that $\delta^j(b) = v_1bv_2cv_3$ or $\delta^j(b) = v_1cv_2bv_3$. Hence, if $\ell g_V(v)$ denotes the number of occurrences of vital letters in a word v, we have

(1) $$\ell g_V(\delta^{i+nj}(w)) \geq \ell g_V(\delta^{nj}(b)) > n,$$

and L(G) is infinite. □

LEMMA 3.8. *Let* G = <W,δ,w> *be a* DOL *system. If* $i \geq \#(V-R)$ *and* $b \in V-R$ *such that* b *is a subword of* $\delta^i(a)$ *for some* $a \in W$ *then there is a* $j < i$ *and a* $c \in R-MR$ *such that* c *is a subword of* $\delta^j(a)$.

PROOF. There is a sequence of letters $a_0, a_1, \ldots, a_i$ such that $a_0 = a$, $a_i = b$, and a_{j+1} is a subword of $\delta(a_j)$ for $0 \leq j < i$. If $b \in V-R$ then $a_j \in V$ for $0 \leq j \leq i$. Since there are at least $\#(V-R)+1$ a_j's there is one which is recursive and therefore there is a $j_1 < i$ such that $a_{j_1} \in R$. It is easy to see that for a recursive letter d always holds that $\delta^t(d)$ contains a recursive letter as a subword for each t. Therefore, $\delta^{i-j_1}(a_{j_1}) = v_1 d v_2 b v_3$ or $\delta^{i-j_1}(a_{j_1}) = v_1 b v_2 d v_3$, where $d \in R$ and $b \in V-R$. Hence $a_{j_1} \in R-MR$. By taking c equal to a_{j_1} the lemma is proved. □

LEMMA 3.9. *Let* G = <W,δ,w> *be a* DOL *system. If* $\delta^t(w) \in (M \cup MR)^*$ *for* $t = \#(V-R)$ *then* L(G) *is finite.*

PROOF. Suppose

(2) $$\delta^{\#(V-R)}(w) = v_1 a_1 v_2 a_2 \ldots v_n a_n v_{n+1},$$

where $a_1, a_2, \ldots, a_n \in MR$ and $v_1, v_2, \ldots, v_{n+1} \in M^*$. Now it is easy to see that for each $a_i \in MR$ there is a k_i $(1 \leq k_i \leq \#MR)$ and a sequence $a_{i0}, a_{i1}, \ldots, a_{ik_i}$ such that $a_{i0} = a_{ik_i} = a_i$, $a_{ij_1} \neq a_{ij_2}$ for $0 \leq j_1 < j_2 < k_i$, and $a_{i\,j+1} \in MR$ is the only vital letter in $\delta(a_{ij})$, $0 \leq j < k_i$. I.e., $\{a_{ij} \mid 0 \leq j < k_i\} = [a_i]$. Also,

(3) $$\delta^{\#M}(b) = \lambda \qquad \text{for all } b \in M.$$

Hence, for all $a_i \in MR$ and all $t, t' \geq \#M$ holds

(4a) $$\delta^t(a_i) = \delta^{t'}(a_i) \qquad \text{for } t \equiv t' \bmod k_i,$$

(4b) $$\delta^t(a_i) \neq v_1 \delta^{t'}(a_i) v_2 \qquad \text{for } t \not\equiv t' \bmod k_i, \quad \text{for all } v_1, v_2 \in W^*.$$

By (2), (3) and (4) we have that for all $t \geq \#(W-R)$ holds:

(5) $$\delta^t(w) = \alpha_{1j_1}\alpha_{2j_2}\dots\alpha_{nj_n},$$

where $\alpha_{ij_i} = \delta^{j_i}(a_i)$, $j_i \equiv t \bmod k_i$ and $\#M \leq j_i < \#M+k_i$, $1 \leq i \leq n$. By (2), (4) and (5):

(6a) $\delta^t(w) \neq \delta^{t'}(w)$ for all t,t' such that $\#(W-R) \leq t < t' < \#(W-R) + \text{l.c.m.}\ (k_1,k_2,\dots,k_n)$;

(6b) $\delta^t(w) = \delta^{t'}(w)$ for all t,t' such that $t,t' \geq \#(W-R)$ and $t \equiv t' \bmod (\text{l.c.m}\ (k_1,k_2,\dots,k_n))$.

Therefore,

(7) $$\text{l.c.m.}\ (k_1,k_2,\dots,k_n) \leq \#L(G) \leq \text{l.c.m.}\ (k_1,k_2,\dots,k_n) + \#(W-R).$$

□

We are now ready to state the main theorem of this section.

THEOREM 3.10. *Let* $G = \langle W,\delta,w\rangle$ *be a* DOL *system.* L(G) *is finite iff* $\delta^t(w) \in (M \cup MR)^*$ *for* $t = \#(V-R)$.

PROOF. "If". By Lemma 3.9.
"Only if".
Case 1. $\delta^t(w) \in W^*(R-MR)W^*$. By Lemma 3.7 L(G) is infinite.
Case 2. $\delta^t(w) \in W^*(V-R)W^*$ for $t = \#(V-R)$. By Lemma 3.8 there is a $t' < t$ such that $\delta^{t'}(w) \in W^*(R-MR)W^*$, and therefore case 1 holds and L(G) is infinite.
Hence, if $\delta^t(w) \in W^*(V-MR)W^*$ for $t=\#(V-R)$ then L(G) is infinite, i.e., if L(G) is finite then $\delta^t(w) \in (M \cup MR)^*$ for $t = \#(V-R)$. □

From the previous lemmas and the theorem we can derive some interesting corollaries.

COROLLARY 3.11. L(G) *is finite iff* $\delta^t(w) \in (M \cup MR)^*$ *for all* $t \geq \#(V-R)$.

COROLLARY 3.12. *A DOL language is finite iff all recursive letters which are accessible from the initial string (i.e., which occur in words in the language) are monorecursive.*

COROLLARY 3.13. *There is an algorithm to determine whether the language generated by a DOL system is finite or not. (Hint: determine* M,V,R *and* MR *and apply Theorem* 3.10 *or Corollary* 3.12*).*

Next we consider the *membership problem*: given a DOL system $G = \langle W,\delta,w\rangle$ and a word $v \in W^*$, decide whether or not v is in $L(G)$. (Equivalently, is there an i such that $\delta^i(w) = v$). Now assume that $L(G)$ is finite and

$$\delta^{\#(V-R)}(w) = v_1a_1v_2a_2\ldots v_na_nv_{n+1},$$

where $a_1,a_2,\ldots,a_n \in MR$ and $v_1,v_2,\ldots,v_{n+1} \in M^*$. Assume further that $v = \alpha_{1j_1}\alpha_{2j_2}\ldots\alpha_{nj_n}$ where $\alpha_{ij_i} = \delta^{j_i}(a_i)$ for some j_i such that $\#M \le j_i < \#M+k_i$, $1 \le i \le n$. By (4b) $\delta^{j_i}(a_i) \ne v_1\delta^{j_i'}(a_i)v_2$ for all j_i,j_i' such that $\#M \le j_i < j_i' < \#M+k_i$ and all $v_1,v_2 \in W^*$, $1 \le i \le n$. Furthermore, if $[a] \ne [b]$ for some $a,b \in MR$ then $\delta^t(a) \ne \delta^{t'}(b)$ for all t,t', since $\delta^t(a) \in M^*[a]M^*$, $\delta^{t'}(b) \in M^*[b]M^*$ and $[a] \cap [b] = \emptyset$. Therefore, the parse of v (if it exists) is unique, and can be executed easily from left to right given $\delta^t(a_i)$ for all t and i, $\#M \le t < \#M+k_i$, $1 \le i \le n$. Since by (4a) $\delta^t(a_i) = \delta^{t'}(a_i)$ for all $t,t' \ge \#M$ such that $t \equiv t' \bmod k_i$, the problem can now be restated as follows: is there a positive integer u such that $u \equiv (j_i-\#M) \bmod k_i$, $1 \le i \le n$. The solution is well known, and is given by the so-called Chinese remainder theorem, see e.g. KNUTH [1969b, 256]:

LEMMA 3.14. *Let $k_1,k_2,\ldots,k_n$, be positive integers and let $t_1,t_2,\ldots,t_n$, be any integers. There is exactly one integer u which satisfies the conditions*

$$0 \le u < \text{l.c.m.}\,(k_1,k_2,\ldots,k_n) \quad \textit{and} \quad u \equiv t_i \bmod k_i \quad (1 \le i \le n)$$

iff

$$t_i \equiv t_j \bmod (\text{g.c.d.}\,(k_i,k_j)) \quad (1 \le i < j \le n).$$

There is no integer $u \equiv t_i \bmod k_i$, $(1 \le i \le n)$, if not $t_i \equiv t_j \bmod (\text{g.c.d.}\,(k_i,k_j))$, $(1 \le i < j \le n)$.

Therefore, if u exists then $v = \delta^{\#(W-R)+u}(w)$ and $v \ne \delta^t(w)$ for all $t \ge \#(W-R)$ otherwise. If a parse of v as mentioned is not possible then by

(5) $v \neq \delta^t(w)$ for all $t \geq \#(W-R)$. Hence we have

THEOREM 3.15. *There is an algorithm which solves the membership problem for* DOL *languages.*

PROOF. The proof consists of giving an outline of the algorithm. Let $G = \langle W,\delta,w\rangle$ be the DOL system concerned and let v be the target word.

ALGORITHM 3.16.

(i) Determine whether or not $L(G)$ is finite, using Corollary 3.13. If $L(G)$ is infinite then generate successively $w,\delta(w),\delta^2(w)\ldots$ and compare each $\delta^i(w)$ with v. Is $\delta^i(w) \neq v$ for all $i < t_0$ and $\delta^{t_0}(w)$ contains more occurrences of vital letters than does v then $v \notin L(G)$. By (1) $t_0 \leq \#V(\ell g_V(v) - \ell g_V(w)+1)$.

(ii) $L(G)$ is finite. Generate successively $w,\delta(w),\ldots,\delta^{\#(W-R)}(w)$ and compare each such $\delta^i(w)$ with v. Is the matching still unsuccessful then try to parse v as discussed above. Is the parse successful then apply the Chinese remainder theorem. Depending on whether or not an integer u, as stated in that theorem, exists v does or does not belong to $L(G)$. If the parse is not successful then $v \notin L(G)$. □

We can speed up stage (i) of the algorithm somewhat as follows. Let h be a homomorphism which erases all mortal letters and leaves the vital letters unchanged. Let $L(G)$ be infinite. If $v = \delta^i(w)$ then $h(v) = h(\delta^i(w))$ and $v \neq \delta^j(w)$ for all $j \neq i$. Hence it suffices to ascertain whether $h(v) \in L(G')$, where $G' = \langle V,\delta',h(w)\rangle$ and $\delta'(a) = h(\delta(a))$ for all $a \in V$. If $h(v) \in L(G')$, i.e., $h(v) = \delta'^i(h(w))$ for some i then $v \in L(G)$ iff $v = \delta^i(w)$. We can determine i by solving $f_{G'}(i) = \ell g(h(v))$ where $f_{G'}$ is the *growth function* (see Ch. 4) of G'. Since G' is propagating $f_{G'}$ is monotone increasing and constant for at most $\#V$ consecutive integer argument values. Hence even by trial and error we can find a solution i for the above equation in about $\log i$ trials. Let i_{min} be the least integer solution of $f_{G'}(i) = \ell g(h(v))$ and i_{max} the largest, $i_{max} - i_{min} \leq \#V$. It suffices to generate

$$\delta^{\#M+j}(\delta'^{(i_{min}-\#M)}(h(w)))$$

for $j = 0,1,\ldots,i_{max}-i_{min}$ and compare these words with v to check whether $v \in L(G)$. To generate $\delta'^{(i_{min}-\#M)}(h(w))$ we can generate $\delta'(a),\delta'^2(a)$, $\delta'^4(a),\ldots,\delta'^{2^n}(a)$, for all $a \in V$ and $n \leq \log_2(i_{min}-\#M)$, and so compose

$\delta'^{(i_{min}-\#M)}$ in the obvious fashion.

The decision procedure for the membership problem for D0L languages we gave above is unusual under mathematical decision procedures of this sort in that it is *feasible*, i.e., gives answers to reasonable questions within a reasonable time, as testified by an ALGOL 60 implementation, VITÁNYI [1972b]. Of course, if L(G) is finite, we can test for membership by generating the whole of L(G) in stage (ii) of the algorithm. But, as will appear from the next corollary and the asymptotic approximations in Section 3.1.1.2, even for a modest alphabet of, say, a hundred letters, this may turn out to be quite unfeasible.

Clearly, the given algorithm works for ED0L systems as well. (An EXL-system G is an XL system together with a terminal alphabet V_T, and the language produced is $E(G,V_T) = L(G) \cap V_T^*$). Although we will not treat the computational complexity of problems connected with L systems in this work, we make a few remarks on the topic which bear on the above algorithm. The *general membership* problem for some XL system G and a word v is: "given G and v, decide whether G derives v and express the time/space used by the (Turing machine implementation of the) decision algorithm as a function of n, the number of symbols in the description of G and v". It follows from the above, that the general membership for EPD0L systems and the infiniteness of D0L systems can be decided deterministically in polynomial time. The algorithm, however, does not yield polynomial time for nonpropagating systems, since they involve directly simulating G's derivation for $\#V(\ell g_V(v) - \ell g_V(w) + 1)$ steps. This derivation can produce intermediate strings, whose length is exponential in W, if G has many mortal letters in W.
JONES and SKYUM [1976, 1977a,c] modified the above algorithm to the general membership problem for ED0L systems, involving a more efficient way to simulate short derivations and the construction of an auxiliary propagating system, to show that also the general membership for ED0L systems can be decided deterministically in polynomial time. Further results on computational complexity issues related to problems about L systems can be found in the above references, or the references on the subject occurring in the Introduction.

Our next corollary defines some number theoretic functions.

COROLLARY 3.17.

(i) *Let* P: $\mathbb{N} \to \mathbb{N}$ *be defined as follows.* P(m) *is the greatest natural number* n *which is the least common multiple of* $k_1, k_2, \ldots, k_q$*, for all possible partitions of* m *into* $q = 1,2,\ldots,m$ *positive integral summands, plus the number of summands equal to* 1. *By* (7) P(m) *is the maximum cardinality of a finite* DOL *language over an alphabet of* m *letters.*

(ii) *Let* S: $\mathbb{N} \to \mathbb{N}$ *be defined as follows.* S(n) *is the least natural number* m *such that there exists a partition of* m *into positive integral summands* $k_1, k_2, \ldots, k_q$, $q \le m$, *and* l.c.m. $(k_1, k_2, \ldots, k_q) + \#\{i \mid k_i = 1\} = n$. *By* (7) S(n) *is the minimum cardinality of an alphabet over which there is a* DOL *language of cardinality* n.

The following Sections 3.1.1.1 - 3.1.1.2 will be concerned with the investigation of the number theoretic functions S, P and some variants. Thus we derive lower bounds on the size of the alphabet as a function S of the size of a finite DOL language over such an alphabet, and upper bounds on the size of a finite DOL language as a function P of the size of the alphabet.

3.1.1.1. FUNCTIONS WHICH RELATE SIZE OF LANGUAGE WITH SIZE OF ALPHABET.

The number theoretic functions S and P of Corollary 3.17 have a much broader setting than just their connection with DOL systems. Imagine a process which starts by counting up to some number d and then initializes some number q of periodic counters. Then S(n) and P(m) have a natural interpretation as the smallest number of states needed to generate a prescribed number n of distinguishable configurations, and the largest number of distinguishable configurations which can be generated by using a prescribed number m of different states, respectively. (We assume that all counters used are identical finite state machines.) If we have the additional restriction $d = 0$ then we ask in effect for the maximum order of a permutation of the m-th degree. (The order of a permutation of the m-th degree is the exponent of the smallest power of a permutation on m elements which is equal to the identity permutation). Already LANDAU [1903] investigated the maximum order f(m) of a permutation of a given degree m. I.e. f: $\mathbb{N} \to \mathbb{N}$ where f(m) is defined as the maximum of the least common multiple of $k_1, k_2, \ldots, k_q$ for all possible partitions of m into $q = 1,2,\ldots,m$ positive integral summands. We shall return to this connection with Landau's work in Section 3.1.1.2.

According to Corollary 3.17,

(8) $$S(n) = \min\{\sum_{i=1}^{q} k_i + d \mid \text{l.c.m.}\ (k_1,k_2,\ldots,k_q) + d = n\},$$

(9) $$P(n) = \max\{\text{l.c.m.}\ (k_1,k_2,\ldots,k_q) + d \mid \sum_{i=1}^{q} k_i + d = n\}.$$

For the smallest values of n we find:

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14
S(n)	1	2	3	4	5	5	6	7	8	7	8	7	8	9
P(n)	1	2	3	4	6	7	12	15	20	30	31	60	61	84

For instance,

$S(14) = 2+7 = 4+3+2 = 9$ since $14 = 2*7 = 4*3 + 2$.

$P(14) = 2*2*3*7 = 4*3*7 = 84$ since $14 = 2+2+3+7 = 4+3+7$.

Hence, the corresponding representations of S(n) and P(n) in $k_1,k_2,\ldots,k_q,d$ are not unique. Clearly, in (8) and (9) the $\bar{k}_1,\bar{k}_2,\ldots,\bar{k}_{\bar{q}}$ for which the extrema are reached for a given n will be relatively prime. Suppose we can factorize a $\bar{k}_i$, $1 \le i \le \bar{q}$, into two relatively prime factors $\bar{k}_{i1}$ and $\bar{k}_{i2}$:

$$\bar{k}_i = \bar{k}_{i1}*\bar{k}_{i2}, \quad \bar{k}_{i1} > 1, \quad \bar{k}_{i2} > 1.$$

Then

$$\bar{k}_i - (\bar{k}_{i1}+\bar{k}_{i2}) = \bar{k}_{i1}*\bar{k}_{i2} - (\bar{k}_{i1}+\bar{k}_{i2}) = (\bar{k}_{i1}-1)(\bar{k}_{i2}-1) - 1 \ge 0.$$

Therefore, it suffices to look for $\bar{k}_1,\bar{k}_2,\ldots,\bar{k}_{\bar{q}}$ which are powers of distinct primes.

Hence we replace (8) and (9) by

(10) $$S(n) = \min\{\sum p^{\alpha} + d \mid \prod p^{\alpha} + d = n\},$$

(11) $$P(n) = \max\{\prod p^{\alpha} + d \mid \sum p^{\alpha} + d = n\},$$

where p denotes some prime. To obtain a canonical representation for S(n) and P(n) we take the representation with the smallest d for which the extrema are reached. By the unique factorization property of the natural numbers this representation will be unique. Additionally we define

$$(12) \qquad S'(n) = \min\{\sum p^{\alpha} + d \mid \prod p^{\alpha} + d \geq n\},$$

$$(13) \qquad P'(n) = \max\{\prod p^{\alpha} + d \mid \sum p^{\alpha} + d \leq n\}.$$

(Then $S'(n)$ is the number of letters in the smallest alphabet over which there is a finite DOL language of at least cardinality n, and $P'(n)$ is the cardinality of the largest finite DOL language over an alphabet of at most n letters.) It is convenient to introduce also

$$(14) \qquad s(n,d) = \sum p^{\alpha} + d \quad \text{such that } \prod p^{\alpha} = n-d,$$

since by the unique factorization property $s(n,d)$ is found immediately; and we see that

$$(15) \qquad S(n) = \min\{s(n,d) \mid 0 \leq d \leq n\}.$$

The first 2000 values of $S(n)$ were determined by computer, and showed a quite erratic behavior. E.g. $S(1971) = 61$, $S(1972) = 50$, $S(1973) = 51$ and $S(2000) = 39$. (ØSTERBY [1973] contains a detailed computer analysis of $S(n)$ for $1 \leq n \leq 5.10^{11}$. Furthermore, $S'(n)$ and $P(n)$ are computed for a large number of values. He considers e.g. the question in how many different ways $S(n)$ can be obtained from n.)

Now let us take a closer look at the general behavior and interrelations of our functions, It is at once apparent that, since $P(n+1) \geq P(n)+1$ for all n, P is strictly increasing and therefore $P' = P$. $S(n+1) \leq S(n)+1$ and $S(8) = S(10) = 7$ while $S(9) = 8$. Therefore, S is not monotonic. By its definition S' is monotonic increasing and $S'(n) \leq S(n)$ for all n. A crude approximation gives us (for $n > 1$):

$$(16a) \qquad P(n) < n^{n};$$

$$(16b) \qquad S(n)^{S(n)} > n;$$

$$(16c) \qquad S'(n)^{S'(n)} > n.$$

From (16b) and (16c) it follows that $S(n) \to \infty$ and $S'(n) \to \infty$ for $n \to \infty$. In

Section 3.1.1.2 we shall derive asymptotic approximations for P, S' and inf S. It will appear that these functions are intimately related to the distribution of the prime numbers. We use the notation $f(x) \sim g(x)$ for $f(x)$ is *asymptotic* to $g(x)$, i.e. $\lim_{x\to\infty} f(x)/g(x) = 1$. It is well known, that the number of primes $\pi(x)$ not exceeding x is asymptotic to $x/\log x$: $\pi(x) \sim x/\log x$. Furthermore, the i-th prime p_i is asymptotic to $i \log i$: $p_i \sim i \log i$. It then follows from (16a) that $e^{\log P(n)} \leq e^{n \log n}$ and therefore $\log P(n) \leq n \log n \sim p_n$. Since $S'(n)^{S'(n)} \geq n$, similarly $\log n \leq S'(n) \log S'(n)$. By noting that $x/\log x$ is asymptotic to the function inverse of $x \log x$, see e.g. HARDY and WRIGHT [1945, 9-10], we have that $S'(n) \geq g(n)$ for some function $g(n) \sim \frac{\log n}{\log \log n} \sim \pi(\log n)$. Therefore, $S(n) \geq g(n)$ also.

Since P is strictly increasing and $P(6) = 7$, $P(7) = 12$: $P: \mathbb{N} \to \mathbb{N}$ is an injection but no surjection; since $S(n+1) \leq S(n)+1$ and $S'(n+1) \leq S'(n)+1$ for all n, $S(n) \to \infty$ and $S'(n) \to \infty$ for $n \to \infty$, $S(5) = S'(5) = S(6) = S'(6) = 5$: $S,S': \mathbb{N} \to \mathbb{N}$ are surjections but no injections. From the definitions we would expect S and S' to be some kind of an inverse of P. Since P gives the maximum size finite language over an alphabet of n letters, and since P is strictly increasing, an alphabet of size n is the minimum size alphabet over which there is a finite language of (at least) size $P(n)$. Therefore, we obtain $S(P(n)) = S'(P(n)) = n$ for all $n \in \mathbb{N}$. Hence the restrictions of S and S' to $A = \{P(i) \mid i \geq 0\}$ are the inverse of P:

$$(17) \qquad S_{/A} = S'_{/A} = P^{-1}.$$

From the definitions we also see that, between two consecutive values of P, S' is constant (S' is monotonic, $S'(P(n)) = n$ for all n, $S'(P(n)+1) = n+1$ for all n) and therefore:

$$(18) \qquad S'(m) = P^{-1}(n) \quad \text{for all } m, \quad P(P^{-1}(n)-1) < m \leq n,$$

where $n \in A$. Since $S'(n) \leq S(n)$ for all n we have therefore by (17)

$$(19) \qquad S(n) = S'(n) = P^{-1}(n) \quad \text{and} \quad S(m) \geq P^{-1}(n),$$

for all $n \in A$ and all $m > P(P^{-1}(n)-1)$.

Therefore, S' is a stepfunction where every step of 1 takes place at a value of P. Furthermore, S' is the greatest monotonic increasing function which is a lower bound on S.

In looking at the function S and trying to distinguish its features, we readily notice that if n is a prime or the power of a prime then $S(n) = S(n-1)+1$. The way S is defined, however, does not give us a general method, to find the value of S for a certain argument, better than by trial and error. The following theorem is one of the main results of this section and provides an inductive definition of S.

THEOREM 3.18.

$$S(n) = \begin{cases} n & \text{for } n = 0,1,2,3,4,5. \\ \min\{S(n-1)+1,\ s(n,0)\} & \text{for } n > 5. \end{cases}$$

PROOF. By induction on n. The theorem holds for $n = 0,1,2,3,4,5$. Suppose the theorem is true for all $n \le m$. Since

$$S(m+1) = \min\{s(m+1,d) \mid 0 \le d \le m+1\},$$

and

$$s(m'+1,d') = s(m',d'-1)+1,$$

for all m' and all d' such that $0 < d' \le m'+1$, we have

$$S(m+1) = \min\{S(m)+1,\ s(m+1,0)\}. \qquad \square$$

The following corollary to Theorem 3.18 is also stated by ØSTERBY [1973, 1976] and gives a definition which bounds the amount of computing we have to perform to obtain S(n). By Theorem 3.18 we have for all n

$$S(n) = \min\{s(n,0),\ s(n-1,0)+1,\dots,s(1,0)+n-1,\ n\}.$$

Since for all k such that $n \ge k > S(n)$ holds $S(n) < s(n-k,0)+k$, we have:

COROLLARY 3.19.

$$S(n) = \min\{s(n,0),\ s(n-1,0)+1,\dots,s(n-S(n),0)+S(n)\}.$$

Hence we only have to compute s(n,d), i.e. the sum of the highest powers of primes in the factorization of n-d, for $d = 0,1,\dots,k_0$, where k_0 is the minimum of the previously computed values of s(n,d) + d.

The analogue of Theorem 3.18 for P is:

$$P(n) = \begin{cases} n & \text{for } n = 0,1,2,3,4 \\ \max\{P(n-1)+1, \max\{m \mid s(m,0) = n\}\} & \text{for } n > 4. \end{cases}$$

This does not help us very much, essentially because although the factorization of a natural number is unique, its partition is not. If we could assume that the following conjecture by LANDAU [1903] is true, viz. $P(\sum_{i=1}^{k} p_i) = \prod_{i=1}^{k} p_i$ for all k, then, since P is strictly increasing, we can slightly limit the number of m's which have to be investigated. It was noted by ØSTERBY [1976], however, that Landau's conjecture is false since

$$\begin{aligned} P(100) &\ge 16.9.5.7.11.13.17.19 + 3 \\ &= 232792563 \\ &> 223092870 \\ &= 2.3.5.7.11.13.17.19.23 \end{aligned}$$

where $2+3+5+7+11+13+17+19+23 = 100$.

3.1.1.2. ASYMPTOTIC APPROXIMATIONS OF S AND P

We now investigate the asymptotic behavior of our functions. LANDAU [1903] proves that for $f(n) = \max\{\Pi p^{\alpha} \mid \Sigma p^{\alpha} \le n\}$

(20) $$\log f(n) \sim \sqrt{n \log n}.$$

THEOREM 3.20.

$$\log P(n) \sim \sqrt{n \log n}.$$

PROOF. By (20) $\log f(n) \sim \sqrt{n \log n}$, i.e.,

$$\lim_{n\to\infty} \frac{\log f(n)}{\sqrt{n \log n}} = 1.$$

Also,

$$\lim_{n\to\infty} \frac{\log(f(n)+n)}{\sqrt{n \log n}} = 1 + \lim_{n\to\infty} \frac{\log(1+n/f(n))}{\sqrt{n \log n}} = 1.$$

Since by (11) and the definition of f(n) we have:

$$f(n) \le P(n) < f(n)+n, \text{ i.e., } \log f(n) \le \log P(n) < \log (f(n)+n),$$

and we proved above that

$$\log f(n) \sim \log (f(n)+n) \sim \sqrt{n \log n},$$

we have

$$\log P(n) \sim \sqrt{n \log n}. \qquad \square$$

COROLLARY 3.21. $\log P(n) = \sqrt{p_n}$ *where* p_n *is the* n-*th prime*.

THEOREM 3.22.

$$S'(n) \sim \frac{\log^2 n}{\log \log^2 n}.$$

PROOF. If $\log y = \sqrt{x \log x}$, then $\log^2 y = x \log x$ and

$$\log \log^2 y = \log x + \log \log x \sim \log x.$$

Since

$$x = \frac{\log^2 y}{\log x} \quad \text{we have} \quad x \sim \frac{\log^2 y}{\log \log^2 y}.$$

By this argument and since $\log P(m) \sim \sqrt{m \log m}$ it follows:

$$m \sim \frac{\log^2 P(m)}{\log \log^2 P(m)}$$

or

$$P^{-1}(n) \sim \frac{\log^2 n}{\log \log^2 n} \quad \text{for } n \in \{P(i) \mid i > 0\}.$$

Denote $\log^2 n/\log\log^2 n$ by $h(n)$. By (18) $S'(n) \sim h(n)$ for n in the range of P. This cannot tell us anything about the sup $S'(n)$, since the restriction of S' to special values of n do not need to yield a lower- or an upper bound. According to (18), however, we have for all pairs of consecutive values of P, say n_1, n_2:

$$S'(n_1) \le S'(m) \le S'(n_2) = S'(n_1) + 1, \qquad n_1 \le m \le n_2.$$

Since h is strictly increasing,

$$\begin{aligned}
\underline{\lim}_{m\to\infty} S'(m)/h(m) &\ge \underline{\lim}_{m\to\infty} S'(m)/h(n_2) \\
&\ge \underline{\lim}_{m\to\infty} (S'(n_2) - 1)/h(n_2) \\
&= \lim_{n_2\to\infty} (S'(n_2)/h(n_2) - 1/h(n_2)) \\
&= 1 - \lim_{n_2\to\infty} 1/h(n_2) = 1.
\end{aligned}$$

Analogous we prove that $\overline{\lim}_{m\to\infty} S'(m)/h(m) \le 1$, and therefore $S'(m) \sim h(m)$ for all $m \in \mathbb{N}$. $\square$

COROLLARY 3.23.

$$S'(n) \sim \pi(\log^2 n).$$

The greatest monotonic increasing function which is a lower bound on S is $S'(n) \sim h(n)$. Therefore

COROLLARY 3.24.

$$\inf S(n) \sim \frac{\log^2 n}{\log \log^2 n}.$$

Because of Theorem 3.18 $\inf S(n) \sim \inf s(n,0)$ and we have:

COROLLARY 3.25. *The greatest monotonic increasing function which is a lower bound on the sum of the greatest powers of primes in the factorization of* n, *i.e.* s(n,0), *is asymptotic to* h(n). *Hence:*

$$\inf s(n,0) \sim \frac{\log^2 n}{\log \log^2 n}\ .$$

As is to be expected, this lower bound is reached for the special sequence of values $n = \Pi_{i=1}^{k} p_i$, $k \in \mathbb{N}$.

LEMMA 3.26.

$$\sum_{i=1}^{k} p_i \sim \frac{\log^2 n}{\log \log^2 n}\ ,$$

where $n = \prod_{i=1}^{k} p_i$ *and* $k \in \mathbb{N}$.

PROOF. The number of factors in a factorization of a natural number n is denoted by $\omega(n)$. According to HARDY and WRIGHT [1945]

$$\omega(n) \sim \frac{\log n}{\log \log n}\ .$$

Therefore, $\Sigma_{i=1}^{k} p_i \sim \Sigma_{i=1}^{\omega(n)} i \log i$. Bounding this discrete summation on both sides by an integral we obtain:

$$\int_1^{\omega(n)} i \log i \; di \le \sum_{i=1}^{\omega(n)} i \log i \le \int_2^{\omega(n)+1} i \log i \; di,$$

$$\frac{1}{2}\left[i^2 \log i - i^2/2 \right]_1^{\omega(n)} \le \sum_{i=1}^{\omega(n)} i \log i \le \frac{1}{2}\left[i^2 \log i - i^2/2 \right]_2^{\omega(n)+1},$$

$$\frac{1}{2}(\omega(n)^2 (\log \omega(n) - \tfrac{1}{2}) + \tfrac{1}{2}) \le \sum_{i=1}^{\omega(n)} i \log i$$

$$\le \frac{1}{2}((\omega(n)+1)^2 (\log(\omega(n)+1) - \tfrac{1}{2}) - 4 \log 2 + 2).$$

Hence, if $n \to \infty$ through this particular series of values we have that

$$\sum_{i=1}^{k} p_i \sim \frac{1}{2}(\omega(n)^2 \log \omega(n) - \omega(n)^2/2)$$

$$\sim \frac{1}{2}\,\omega(n)^2 \log \omega(n)$$

$$\sim \frac{\log^2(n)(\log\log n - \log\log\log n)}{2(\log\log n)^2}$$

$$\sim \frac{\log^2 n}{2\log\log n} = \frac{\log^2 n}{\log\log^2 n}. \qquad \square$$

A numerical verification shows:

$$(2+3+5)/(\log^2(2*3*5)/\log\log^2(2*3*5)) \approx 0.47$$

$$(2+3+\ldots+17)/(\log^2(2*3*\ldots*17)/\log\log^2(2*3*\ldots*17)) \approx 0.58$$

$$(2+3+\ldots+97)/(\log^2(2*3*\ldots*97)/\log\log^2(2*3*\ldots*97)) \approx 0.75$$

$$(2+3+\ldots+173)/(\log^2(2*3*\ldots*173)/\log\log^2(2*3*\ldots*173)) \approx 0.79.$$

Summarizing the results of this section we have:

$$\log P(n) \sim \sqrt{n \log n} \sim \sqrt{p_n};$$

$$S'(n) \sim \inf S(n) \sim \inf s(n,0) \sim \frac{\log^2 n}{\log\log^2 n} \sim \pi(\log^2 n);$$

and, furthermore,

$$s(n,0) \sim \frac{\log^2 n}{\log\log^2 n},$$

for $n \to \infty$ through the particular series of values $n = \prod_{i=1}^{k} p_i$.

3.1.1.3. CLASSIFICATION AND CLOSURE PROPERTIES

As is readily proved, there are regular, strictly context free (CF-REG), and strictly context sensitive (CS-CF) languages which are DOL languages, and there are such languages which are not DOL languages. This fact shows that the DOL languages are incomparable with REG and CF. They are, however, included in CS.

With regard to closure under several operations, it can be shown that the D0L languages are not closed under intersection, union, complement, concatenation, homomorphisms, nonerasing homomorphisms, intersection with regular sets, inverse homomorphisms, etc. In short, they are not closed under any of the usual AFL operations and have therefore been called anti-AFLs, cf. SALOMAA [1973a]. The D0L languages share this characteristic with the other families of pure L languages. This resistance against closure operations is by and large a consequence of the lack of a terminal-nonterminal mechanism, although in the case of D0L or PD1L systems the addition of this mechanism does not improve the closure properties. Further details are found in, e.g., HERMAN and ROZENBERG [1975] and SALOMAA [1973a]. See also Section 3.2.

3.1.2. STRUCTURE OF D0L SYSTEMS WITH APPLICATIONS TO GROWTH FUNCTIONS, LOCAL CATENATIVENESS, AND CHARACTERIZATIONS.

Although objections may be raised against the adequacy of L systems to model phenomena occurring in actual biological development, and against the usefulness of sophisticated mathematical theorems in developmental biology (see the introductory chapter), it seems nevertheless that developmental biologists might find conceptual help from the more superficial aspects of the theoretical framework embodied by Lindenmayer's model. Some of the mathematical theorems might be useful to confirm or refute biological hypotheses - but only after careful scruteny as to whether the assumptions under which the theorems hold are reflected entirely by the biological reality in the case under consideration.

As a reference frame to think about cell-lineage, cell-differentiation, cell-potential and the like, the associated digraphs introduced in this section may be of value to developmental biologists. In this respect also theorems about growth functions, locally catenative systems etc. may prove worthwhile. With this idea in mind, we digress in Section 3.1.2.4 from mathematics into possible biological interpretations of the material covered in this section.

In the following we will construct four digraphs associated with a (semi) D0L system, which form in increasing levels of abstraction a representation of the structure of derivations between letters in the system. These are the associated digraph, the condensed associated digraph, the recursive structure, and the unlabeled recursive structure, respectively. We investigate the relations between types of growth functions (cf. also

Ch. 4) and types of recursive structures. We will derive necessary conditions on the recursive structure of a DOL system, in order that it can have the locally catenative property, and an $e^{\sqrt{n \log n}}$ worst-case lower bound on the minimal depth of a locally catenative formula for a locally catenative DOL system with an n letter alphabet. It is shown that the sequence and language equivalence problems for locally catenative DOL systems are decidable. Furthermore, it is shown that deciding whether a DOL system has the locally catenative property is equivalent to deciding whether the monoid generated by the language of a DOL system is finitely generated. At the end of this section we apply associated digraphs to some results by SALOMAA [1975b], which yields necessary (and sometimes sufficient) conditions on the recursive structure of a DOL system for the produced languages to be regular or context free.

Anticipating Chapter 4 on growth functions, we introduce part of the material here. Let $G = \langle W,\delta,w\rangle$ be a DOL system. The *growth function* of G is the function $f_G\colon \mathbb{N} \to \mathbb{N}$ defined by $f_G(t) = \ell g(\delta^t(w))$. As we will see in Chapter 4, f_G is a generalized exponential polynomial

$$f_G(t) = \sum_{i=1}^{r} p_i(t)c_i^t ,$$

where the c_i's are distinct (and possibly complex) constants, the p_i's are polynomials in t (with possibly complex coefficients) such that $\Sigma_{i=1}^{r}$ (degree $p_i + 1) \le \#W$.

A DOL system $G = \langle W,\delta,w\rangle$ has the *locally catenative* property, cf. ROZENBERG and LINDENMAYER [1973], if there exist fixed positive integers $n_0, i_1, i_2, \ldots, i_k$ such that

$$\delta^n(w) = \delta^{n-i_1}(w)\delta^{n-i_2}(w)\ldots\delta^{n-i_k}(w)$$

for all $n \ge n_0$. n_0 is called the *cut* and $\max\{i_1,i_2,\ldots,i_k\}$ the *depth* of the *locally catenative formula* $(n_0,i_1,i_2,\ldots,i_k)$.

We now construct four digraphs from a semi DOL system $S = \langle W,\delta\rangle$, which form in increasing levels of abstraction a representation of the derivational relations between letters.

I. The *associated digraph* of S (AD(S)), called the *dependence graph* in ROZENBERG and LINDENMAYER [1973], is the labeled digraph $AD(S) = (W,A)$ where W is the set of *points* and A the set of *directed arcs* defined by

$$A = \{(a,b) \mid \delta(a) = v_1 b v_2,\ a,b \in W,\ v_1,v_2 \in W^*\}.$$

Note that we identify points with their *labels*, since for all digraphs we discuss there is a one-to-one correspondence between the set of points and the set of labels. We admit digraphs with *loops*, i.e., a point can be connected to itself by an arc.

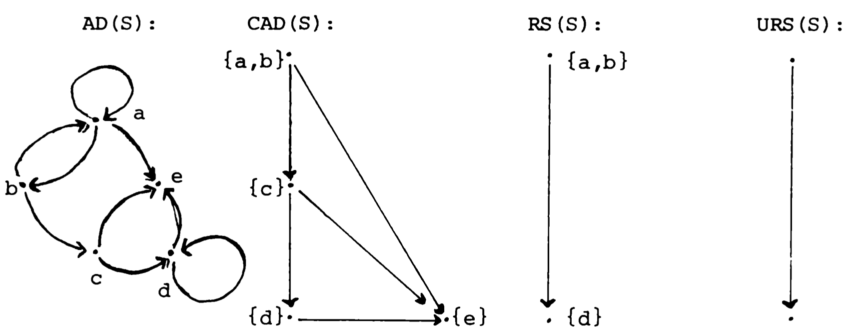
A digraph is *strong* if every two points are mutually reachable, i.e., if p,q are two points of the digraph then there is a sequence of arcs (p_1, p_2), $(p_2,p_3),\ldots,(p_{n-1},p_n)$ such that $p_1 = p$ and $p_n = q$. (We consider the graph on a single point without arcs to be a strong digraph.) A *strong component* of a digraph is a maximal strong subgraph. Let $D_1,D_2,\ldots,D_n$ be strong components of a digraph D. The *condensation* $D^{(c)}$ of D has the strong components of D as its points, with an arc from D_i to D_j $(i \neq j)$ whenever there is at least one arc in D from a point in D_i to a point in D_j. If follows from the maximality of the strong components that the condensation of a digraph has no cycles.

II) The *condensed associated digraph* of S (CAD(S)) is the condensation of AD(S). A point in CAD(S) is labeled by the set of letters labeling the points of the corresponding strong component in AD(S).

III) The *recursive structure* of S (RS(S)) is obtained from CAD(S) by deleting all points labeled by $\{a\}$ where a is not a recursive letter. Two points p,q in RS(S) are connected by an arc (p,q) if there is a sequence of arcs $(p_1,p_2),(p_2,p_3),\ldots,(p_{n-1},p_n)$ in CAD(S) such that $p_1 = p$, $p_n = q$ and $p_i \in \{\{a\} \mid a \notin R\}$ for all i, $1 < i < n$.

IV) The *unlabeled recursive structure* of S (URS(S)) is obtained from RS(S) by removing the labels.

Let $S = \langle\{a,b,c,d,e\},\{\delta(a) = abe, \delta(b) = ac, \delta(c) = de, \delta(d) = de, \delta(e) = \lambda\}\rangle$.

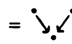


We now tie in the digraph approach with the preceding classification of letters. It is easy to see that $RS(S) = (P_3, A_3)$ is the labeled acyclic digraph such that $P_3 = R/\sim$ and $A_3 \subseteq R/\sim \times R/\sim$ is as defined in (iii). Similarly, each subset of W-R labeling a point in CAD(S) is a singleton subset of $W - R$, and conversely. A letter $a \in W$ labeling a point in AD(S) with no outgoing arcs is an element of M, etc.

For each unlabeled acyclic digraph D we can find a semi DOL system S such that URS(S) $\simeq$ D ("$\simeq$" means "is isomorphic with"). Hence the set of all homomorphisms $\delta: W^* \to W^*$, where W is a finite nonempty subset of some infinite alphabet Σ, can be divided into disjoint classes of homomorphisms having isomorphic unlabeled recursive structures. It is natural to assign to a given homomorphism its URS(S) as its complexity (*structural* complexity which should not be confused with *computational* complexity). We define a partial ordering on the thus constructed disjoint complexity classes as a partial ordering according to graph inclusion. It is of interest to see how many different URS's are possible for an alphabet of n letters. If we call the number of unlabeled acyclic digraphs on n points H(n) then this is given by $F(n) = \Sigma_{i=0}^{n} H(n)$. ROBINSON [1970] gives a method to compute H(n) for all n; in particular this yields: F(0) = 1, F(1) = 2, F(2) = 4, F(3) = 10, F(4) = 41, F(5) = 343 and F(6) = 6327. The partial ordering $\leq$ induced by "being a subgraph of" on the set of unlabeled acyclic digraphs (on i points, $0 \leq i \leq n$) has a 0 element: the empty graph; and a 1 element: the *complete* unlabeled acyclic digraph (on n points), i.e., the unlabeled acyclic digraph with the maximal number of arcs ($\frac{1}{2}n(n-1)$) which is unique up to isomorphism. In a similar way we can define complexity classes of (semi) DOL systems and a partial ordering between them with respect to the levels of abstraction I-III.

A DOL system $G = \langle W,\delta,w \rangle$ is *reduced* if all letters of W occur in L(G), or equivalently, if the axiom w contains letters from each point which is a maximal element (point without incoming arcs) of CAD($\langle W,\delta \rangle$). Considerable attention has been given to the problem which properties are possible for DOL systems with different initial strings and the same semi DOL system $S = \langle W,\delta \rangle$. This problem can often be reduced to looking at subgraphs of CAD(S), with as maximal elements points labeled by the sets of letters in the chosen initial string.

We have seen that the set of all (semi) DOL systems is partitioned in disjoint classes having isomorphic characteristic digraphs. We would like to know whether this is also the case for the corresponding classes of languages. However, there are DOL systems G_1, G_2 such that $L(G_1) = L(G_2)$ while $URS(G_1) \not\cong URS(G_2)$ as is shown by the examples

$G_1 = \langle\{a,b,c\},\{\delta(a) = a, \delta(b) = ba, \delta(c) = ac\}, bac\rangle$, $URS(G_1) =$ 

$G_2 = \langle\{a,b,c\},\{\delta(a) = a, \delta(b) = b, \delta(c) = a^2c\}, bac\rangle$, $URS(G_2) =$  .

Yet it is to be expected that DOL systems with different associated digraphs often generate different languages. For instance, the previously mentioned language $\{a^{2^n}b^{2^n}c^{3^n} \mid n \geq 0\}$ can only be produced by a DOL system having a totally disconnected URS on three points. For the class of DOL languages, such that each language in the class can be produced by exactly one DOL system, obviously the URS complexity classes are disjoint.

3.1.2.1. GROWTH FUNCTIONS

First we will consider how the CAD(S) and RS(S) of a semi DOL S can look with regard to the distribution of different types of letters over the labels in the digraphs. Let $S = \langle W,\delta\rangle$ be a semi DOL system. A letter $a \in W$ is of growth type 3 (*exponential*) if $\overline{\lim_{t\to\infty}}\, \ell g(\delta^t(a))/x^t > 0$ for some $x > 1$; *) of growth type 2 (*polynomial*) if there exist polynomials p,q such that $p(t) \leq \ell g(\delta^t(a)) \leq q(t)$ for all t; of growth type 1 (*limited*) if there is a constant c such that $1 \leq \ell g(\delta^t(a)) \leq c$ for all t; of growth type 0 (*terminating*) if $\ell g(\delta^t(a)) = 0$ for all $t \geq \#W$. Similarly we classify DOL systems $G = \langle W,\delta,w\rangle$ where we substitute w for a in the definition. (Note that $\ell g(\delta^t(a_1a_2\ldots a_n)) = \ell g(\delta^t(a_1)) + \ell g(\delta^t(a_2)) + \ldots + \ell g(\delta^t(a_n)))$. A complete investigation into growth types of letters, DOL systems and semi DOL systems will be encountered in Chapter 4. We define $GT(i) = \{a \mid a \in W$ and a is of growth type $i\}$, $i = 0,1,2,3$. Recall that a point p is reachable from a point q in a digraph D if there is a sequence $(p_1,p_2),(p_2,p_3)\ldots(p_{n-1},p_n)$ of arcs in D such that $q = p_1$ and $p = p_n$.

We can distinguish two distinct regions in CAD(S): an exponential region and a polynomial region (and of course a region consisting of points labeled by mortal letters). Clearly no point in the exponential region (labeled by subsets of GT(3)) is reachable from a point in the polynomial region (labeled by subsets of $GT(2) \cup GT(1)$). Both regions have minimal elements. For the exponential region this is the set:

*) Since under the morphism δ only mortal letters can derive λ and do so within $\#M$ steps, we have for each letter a that for some constant $x > 1$ holds: $\overline{\lim_{t\to\infty}}\, (\ell g(\delta^t(a))/x^t) > 0$ iff $\underline{\lim_{t\to\infty}}\, (\ell g(\delta^t(a))/x^t) > 0$.

$$M_E = \{[a] \in R/\sim \mid [a] \text{ is the label of a minimal exponential element}\},$$

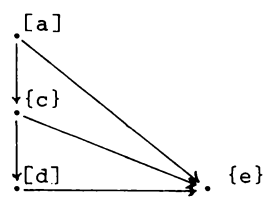
and it appears that

$M_E \subseteq \{[a] \in R/\sim \mid [a] \leq E\}$, the proof of which we leave to the reader (hint: similar to the proof of Theorem 4.22).

For the polynomial region:

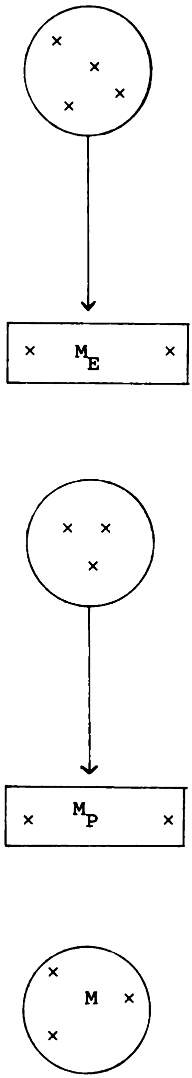
$M_P = \{[a] \in R/\sim \mid [a] \leq MR\}$, which is proved by Theorem 4.22.

For RS(S) the same holds except that only labels in $R/\sim$ occur. Letters which are vital but not recursive and which label points in CAD(S) serve as transitionary stages in the genealogical development of a cell from one recursive state to another, a development which is irreversible and corresponds to further differentiation. Cells which are in a state $b \in [a]$ have the potential to produce cells in states $U(a)$ and there is always a cell in their offspring which has the same potential. Take as an example the CAD(S) we met previously.



Here c is a transitionary state in the offspring of a cell in state a or b, with the corresponding potential, specializing or differentiating to a cell in state d with lower potential.

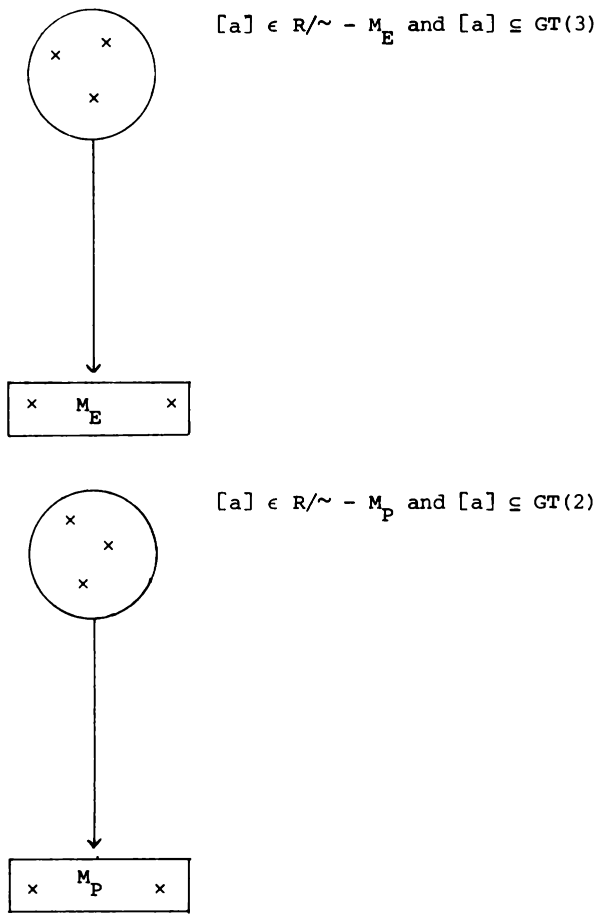
The following pictures hold, where a solid arrow implies that for each point in the upper set there is at least one point in the lower set which is reachable from it; a rectangle means that distinct points in there cannot be reached from each other. Points in lower sets may always be reachable from points in upper sets.



$$\begin{cases} \{a\} \subseteq (V-R) \cap GT(3) \\ \quad\text{or} \\ [a] \in R/{\sim} - M_E \text{ and } [a] \subseteq GT(3). \end{cases}$$

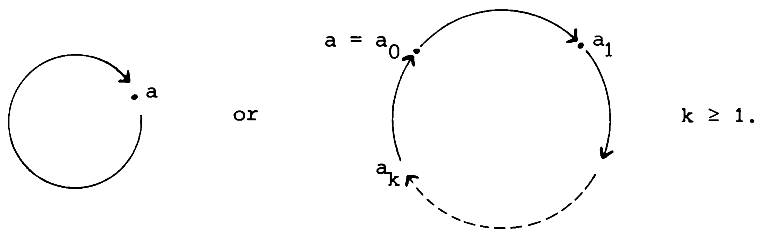
$$\begin{cases} \{a\} \subseteq (V-R) \cap GT(2) \\ \quad\text{or} \\ [a] \in R/{\sim} - M_P \text{ and } [a] \subseteq GT(2). \end{cases}$$

CAD(S)



RS(S).

The structure of the CAD and the RS are justified more or less as follows. To obtain an insight in the properties of a DOL system we often have to take the number of occurrences of each letter in the values of δ into account. We can do so by weighing the arcs in the associated digraphs or, alternatively, using multi digraphs. That is, digraphs with more than one arc between two vertices. Taking the latter course, we define the associated multi digraph (AMD) by drawing i arcs from vertex a to vertex b if $\delta(a)$ contains i occurrences of the letter b. Often it is more rewarding to consider the AMD of a semi DOL system than the AD. The recursive letters, or rather the equivalence classes of R induced by ~, correspond to the (nontrivial) strong components of the associated multi digraph. It is easy to see that for polynomially growing recursive letters a ($a \in R \cap (GT(1) \cup GT(2))$), the strong component in the AMD associated with [a] is a simple cycle (a cycle without multiple arcs):



For exponentially growing recursive letters a ($a \in R \cap GT(3)$) the strong component in the AMD associated with [a] can be any strong multi digraph (with multi loops). All such exponentially growing letters are expanding ($\in E$) except those which occur in a strong component which is a simple cycle. We can now see how the AMD (or AD with some additional information) ties in with the properties of the letters. If a letter ($\in R$) is exponential but not expanding, its associated strong component is a simple cycle, but from this strong component we can always reach another strong component which is not a simple cycle and which therefore is associated with an expanding letter. Hence $M_E \subseteq E/\sim$. If from a given strong component which is a simple cycle we cannot reach a strong component which is not, then the letters of the simple cycle induce polynomial or limited growth. Strong components which are simple cycles and from which we cannot reach nontrivial strong components induce limited growth (and they only). Hence $M_P = MR/\sim$.

As we will prove in Chapter 4 all letters belong to GT(0), GT(1), GT(2) or GT(3). When we talk about a digraph associated with a DOL system G we shall assume that G is reduced and we restrict the homomorphism involved accordingly and write CAD(G), RS(G), etc.

The following observations are readily deduced from the previous exposition.

- Since according to Section 3.1.1 L(G) is finite iff MR = R and clearly points in $MR/\sim$ cannot be reachable from each other we have: L(G) is finite iff RS(G) is totally disconnected and $\cup R/\sim = MR$ (i.e., if RS(G) consists of the bottom rectangle only).
- If RS(G) is nonempty, totally disconnected and $\cup R/\sim \neq MR$ then L(G) is infinite and f_G is exponential (i.e., if RS(G) contains at least the

upper rectangle and at most both rectangles without reachability among them.

EHRENFEUCHT and ROZENBERG [1974c] define the *rank* of a DOL system $G = \langle W,\delta,w\rangle$ as follows. (N.B. Not all DOL systems have a rank).

(i) If $\ell g(\delta^t(a)) \leq c$ for some constant c and all t then $\rho_G(a) = 1$.

(ii) Let $W_0 = W$ and $\delta_0 = \delta$. For $j \geq 1$, δ_j denotes the restriction of δ to $W_j = W - \{a \mid \rho_G(a) \leq j\}$. I.e., all rules with an argument in $W - W_j$ are deleted and in the values of the remaining rules letters in $W - W_j$ are replaced by λ. For $j \geq 1$ if $\ell g(\delta_j^t(a)) \leq c_j$ for some constant c_j and all t then $\rho_G(a) = j+1$.

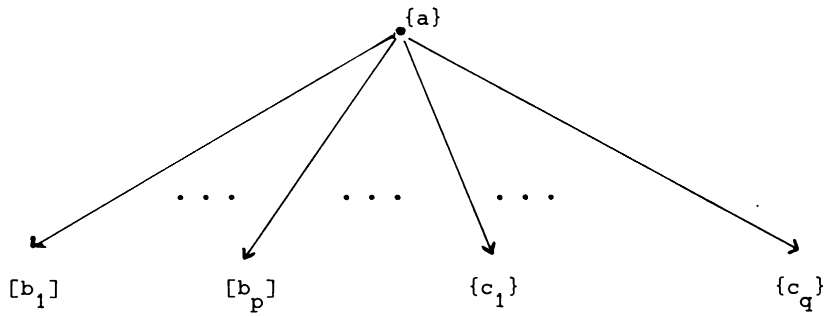
$\rho_G(a)$ is called the rank of a letter a in $\langle W,\delta\rangle$. If each letter $a \in W$ has a rank then G has a rank. The rank of G is the largest one of the ranks of all letters accessible from the initial string or, equivalently, of all letters in the initial string. According to EHRENFEUCHT and ROZENBERG [1974c] G is a DOL system with rank iff there are polynomials p,q of degree (rank (G)-1) such that $p(t) \leq f_G(t) \leq q(t)$ for all t.

THEOREM 3.27. *The rank of* G *is equal to the length of the longest path in* RS(G) *or, equivalently,* f_G *is bounded above and below by polynomials of degree one less than the length of the longest path in* RS(G) *iff* $E = \emptyset$.

PROOF. "$\Rightarrow$". If f_G is bounded by a polynomial then $E = \emptyset$ since an expanding letter would induce exponential growth.

"$\Leftarrow$". Suppose $E = \emptyset$. Then there are also no other letters inducing exponential growth, since they would derive a letter in E. Hence we have a CAD where every point is labeled by either an equivalence class [a] or a singleton set {a}, and since there are no exponential letters (inducing exponential growth) each letter of an equivalence class [a] is rewritten as a string containing one letter from [a] and with the remaining letters chosen from the sets labeling the descendants of the point in the CAD labeled by [a]. Hence we can estimate the order of growth in G from its CAD as follows. The bottom elements of the CAD are sets $\{a\} \subseteq M$ or $[a] \subseteq MR$. If $a \in M$ then f_a is vanishing and $f_a(t) \in \Theta(0)$. ($\Theta(g(n))$ denotes the set of all f(n) such that there exist positive constants c,c' and n_0 with $cg(n) \leq f(n) \leq c'g(n)$ for all $n \geq n_0$. $f(n) \in \Theta(g(n))$: f(n) is of *order of magnitude* of g(n)). If $a \in MR$ then f_a is limited and $f_a(t) \in \Theta(1)$. Assume that all growth functions associated with descendants of {a} or [a] in the CAD are not faster than polynomial, where {a} or [a] labels a nonbottom element. We have two possibilities:

(i) $a \in V - R$. Then

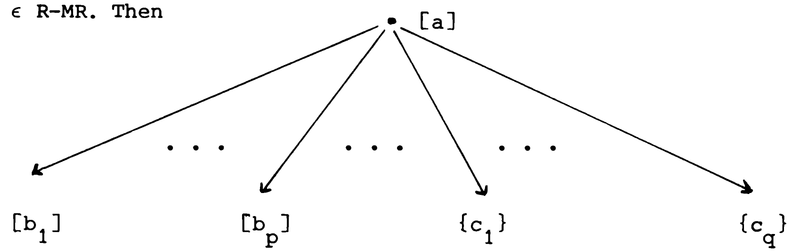


is the subgraph in the CAD consisting of $\{a\}$ plus its direct descendants (corresponding to $\delta(a) = d_1 d_2 \ldots d_\ell$ with $\ell \le \max\{\ell g(\delta(a)) \mid a \in W\}$). Since $f_a(t) \in \Theta(f_b(t))$ for a and b in the same equivalence class and since $f_a(t) \in \Theta(f_a(t-i))$, $1 \le i \le \#W$, for DOL growth functions (which follows, e.g., from the difference equation representation of DOL growth functions in Section 4.1) we have

$$f_a(t) \in \Theta\left(\sum_{i=1}^{p} f_{b_i}(t) + \sum_{i=1}^{q} f_{c_i}(t)\right)$$

Hence $f_a(t)$ is of the same order of magnitude as the fastest increasing function among the f_{b_i} s and the f_{c_i} s.

(ii) $a \in$ R-MR. Then



is the subgraph in the CAD consisting of $[a]$ plus its direct descendants. It is easy to see that $\bigcup_{i=1}^{p} [b_i] \cup \{c_i \mid 1 \le i \le q\}$ equals the set of letters not in $[a]$ which occur in some $\delta(c)$, $c \in [a]$. From the fact that the elements of $[a]$ form a simple cycle in the AD of length, say k, $1 \le k \le \#W$, and the mentioned properties of Θ for DOL growth functions it therefore follows that:

$$f_a(t) \in \Theta\left(\sum_{i=0}^{\lfloor t/k \rfloor} \left(\sum_{j=1}^{p} f_{b_j}(k*i) + \sum_{j=1}^{q} f_{c_j}(k*i)\right)\right)$$

$$= \Theta\left(t\left(\sum_{j=1}^{p} f_{b_j}(t) + \sum_{j=1}^{q} f_{c_j}(t)\right)\right)$$

and therefore $f_a(t) \in \Theta(t*f_{max}(t))$ where f_{max} is the fastest increasing among the f_{b_i} s and f_{c_i} s.

From (i) and (ii) we can conclude that, if $E = \emptyset$, the growth of a letter a is of the order of a polynomial of degree one less than the maximal number of equivalence classes of recursive letters we can encounter in a path from a point {a} or [a] in the CAD to a bottom point. From this the theorem clearly follows. □

3.1.2.2. THE LOCALLY CATENATIVE PROPERTY

We now turn our attention to locally catenative DOL systems producing infinite languages (finite DOL systems are trivially locally catenative), i.e., $k > 1$ in the locally catenative formula.

<u>THEOREM 3.28</u>. *If a* DOL *system* $G = \langle W,\delta,w \rangle$ *is locally catenative then* RS(G) *is a directed labeled rooted tree with branches of at most length* 1 *such that* $[c] = E$ *labels the root and the elements of* $R/\!\sim - \{[c]\}$ *label the leaves,* $\cup(R/\!\sim - \{[c]\}) = MR$.

<u>PROOF</u>. If $G = \langle W,\delta,w \rangle$ is locally catenative then there are fixed integers $n_0, i_1, i_2, \ldots, i_k$ such that

$$\delta^n(w) = \delta^{n-i_1}(w)\,\delta^{n-i_2}(w)\ldots\delta^{n-i_k}(w)$$

for all $n \geq n_0$. Therefore, $L(G) \subseteq \{\delta^i(w) \mid i < n_0\}^*$ and if $\delta^t(w) = v_1 a v_2 b v_3$, $a \sim b$ and $v_2 \in (W - [a])^*$ then

(1) $$\ell g(v_2) < 2 \max\{\ell g(\delta^i(w)) \mid i < n_0\}.$$

Assume that $a,b \in R$ and $a \notin U(b)$. Since G is reduced at least one letter from both [a] and [b] occurs in $\delta^{n_0}(w)$. By the locally catenative property there must be an i such that

$$\delta^i(w) \in W^*[a]W^*[b]W^*[a]W^*.$$

Then for all t holds: $\delta^{i+t}(w) = v_1cv_2dv_3$ for some $c,d \in [a]$, $v_2 \in (W-[a])^+$ and $\ell g(v_2) \geq \ell g(\delta^t(e))$ for some $e \in [b]$. By (1) it follows that $[b] \subseteq MR$. Since for all $[a],[b] \in R/\sim$, $[a] \neq [b]$, either $a \notin U(b)$ or $b \notin U(a)$ we have: either all $[b] \in R/\sim$ are contained in MR and L(G) is finite or there is exactly one $[c] \in R/\sim$ which is not contained in MR. Since the assumption that there exists a $[b] \neq [c]$ such that $b \notin U(c)$ leads to the contradiction that $[c] \subseteq (R-MR) \cap MR$ we have that

(2) $\qquad b \in U(c) \qquad$ for all $b \in R - [c] = MR$.

If $[c] \not\subseteq MR$ then L(G) is infinite by Theorem 3.10; and under the assumption that G is locally catenative; $k > 1$ in the locally catenative formula. Then, as we easily see, f_G is exponential and by Theorem 3.27 $E \neq \emptyset$. Hence $[c] = E$ and by (2) the theorem follows. $\square$

Note that Theorem 3.28 gives a necessary but not sufficient condition for a DOL system to possess the locally catenative property. For instance

$$G = \langle\{a,b\},\{\delta(a) = b,\delta(b) = ab\},ba\rangle$$

with

$$S(G) = ba,abb,babab,abbabbab,\ldots$$

is easily proven not to be locally catenative but

$$G = \langle\{a,b\},\{\delta(a) = b,\delta(b) = ab\},a\rangle$$

with

$$S(G) = a,b,ab,\ldots$$

is locally catenative.

<u>LEMMA 3.29</u>. *Let* $G = \langle W,\delta,w\rangle$ *be a* DOL *system such that there exist integers* $n_0,i_1,i_2,\ldots,i_k$ *for which* $\delta^{n_0}(w) = \delta^{n_0-i_1}(w)\delta^{n_0-i_2}(w)\ldots\delta^{n_0-i_k}(w)$ *then* G *is*

locally catenative with formula $(n_0,i_1,i_2,\dots,i_k)$.

PROOF. By induction on n, $n \geq n_0$, in $\delta^n(w)$. □

Obviously, any locally catenative DOL sequence can be characterized by infinitely many locally catenative formulas. From the above lemma we see that we can assign a unique locally catenative formula to such a sequence. E.g. given a locally catenative DOL sequence, assign to it the first formula in the lexicographical ordering of the set of formulas satisfying the sequence. We call this locally catenative formula the *canonical* locally catenative formula of the DOL system. The following two decision problems suggest themselves immediately.

(i) Decide whether or not a given DOL system is locally catenative.

(ii) Decide whether two locally catenative DOL systems produce the same sequences (languages), i.e., given two locally catenative DOL systems G, G' decide whether or not $S(G) = S(G')$ $(L(G) = L(G'))$.

In view of the preceding remark on locally catenative formulas the second question is settled easily. In fact, much more easily than for DOL systems in general which requires the method of ČULIK and FRIS [1977a,b].

THEOREM 3.30. *The sequence (language) equivalence is decidable for locally catenative* DOL *systems.*

PROOF. Let $G_i = \langle W,\delta_i,w\rangle$, $i = 1,2$, be two locally catenative DOL systems. $S(G_1) = S(G_2)$ iff both G_1, G_2 have the same canonical locally catenative formula, say $(n_0,i_1,i_2,\dots,i_k)$, and $\delta_1^i(w) = \delta_2^i(w)$ for all $i < n_0$. By NIELSEN [1974] a decision procedure for the sequence equivalence can be extended to a decision procedure for the language equivalence. □

To decide whether a DOL system is locally catenative is much more difficult and still open at the time of writing, but we shall prove some related results.

Define the functions $c,d\colon \mathbb{N} \to \mathbb{N}$ as follows:

$$c(n) = \sup\{n_0 \mid G \text{ is a loc. cat. DOL system with an } n \text{ letter alphabet and } n_0 = \inf\{m \mid m \text{ is the cut of a loc. cat. formula for } G\}\};$$

$$d(n) = \sup\{d \mid G \text{ is a loc. cat. DOL system with an } n \text{ letter alphabet and } d = \inf\{m \mid m \text{ is the depth of a loc. cat. formula for } G\}\}.$$

To decide whether or not a given DOL system is locally catenative it suffices to exhibit a total function $k\colon \mathbb{N} \to \mathbb{N}$ such that $k(n) \geq c(n)$ for all n and show that k is computable. We shall prove that if such a k exists then $\log k(n)$ is asymptotically greater or equal to $\sqrt{n \log n}$. For technical reasons k, c and d are taken to assume the value ∞ if they are undefined in some argument. Clearly, $k(n) \geq c(n) \geq d(n)$ for all n. First we prove a stronger result.

THEOREM 3.31.

$$\liminf_{n\to\infty} \frac{\log d(n)}{\sqrt{n \log n}} \geq 1.$$

PROOF. Let $G_1 = \langle W_1, \delta_1, w_1\rangle$ be a DOL system with $\#W_1 = n-1$ and $L(G_1)$ is finite. We construct a DOL system $G_2 = \langle W_2, \delta_2, w_2\rangle$ where $W_2 = W_1 \cup \{a\}$, $a \notin W_1$, $\delta_2 = \delta_1 \cup \{\delta_2(a) = aw_1a\}$, $w_2 = aw_1$.

CLAIM. For all $i > 0$, $\delta_2^i(w_2) = \delta_2^{i-1}(w_2)\delta_2^{i-2}(w_2)\ldots\delta_2^0(w_2)a\delta_2^i(w_1)$.

PROOF OF CLAIM. By induction on i.

$i = 1$. $\delta_2(aw_1) = aw_1a\delta_2(w_1) = \delta_2^0(w_2)a\delta_2(w_1)$.

$i > 1$. Suppose the claim is true for all $j \leq i$.

$$\begin{aligned}\delta_2^{i+1}(aw_1) &= \delta_2(\delta_2^i(aw_1)) = \delta_2(\delta_2^{i-1}(w_2)\delta_2^{i-2}(w_2)\ldots\delta_2^0(w_2)a\delta_2^i(w_1)) \\ &= \delta_2^i(w_2)\delta_2^{i-1}(w_2)\ldots\delta_2^1(w_2)aw_1a\delta_2^{i+1}(w_1) \\ &= \delta_2^i(w_2)\delta_2^{i-1}(w_2)\ldots\delta_2^0(w_2)a\delta_2^{i+1}(w_1),\end{aligned}$$

which proves the claim. *End of proof of Claim.*

Since $L(G_1)$ is finite there are smallest integers $t_0, u \in \mathbb{N}$ such that

(3) $\quad \delta_1^t(w_1) = \delta_1^{t'}(w_1)$ for $t, t' \geq t_0$ and $t \equiv t' \bmod u$,

(4) $\quad \delta_1^t(w_1) \neq \delta_1^{t'}(w_1)$ for $t < t_0$ and $t' \geq t_0$ or for $t, t' \geq t_0$ and $t \not\equiv t' \bmod u$.

i.e., $\#L(G_1) = t_0 + u$.

G_2 is locally catenative since for all $t \geq t_0$:

$$\delta_2^{t+u}(w_2) = \delta_2^{t+u-1}(w_2)\delta_2^{t+u-2}(w_2)\ldots\delta_2^{0}(w_2)a\delta_2^{t+u}(w_1) \quad \text{(by the claim)}$$

$$= \delta_2^{t+u-1}(w_2)\delta_2^{t+u-2}(w_2)\ldots\delta_2^{t}(w_2)\delta_2^{t-1}(w_2)\ldots\delta_2^{0}(w_2)a\delta_2^{t}(w_1)$$
$$\text{(by (3))}$$

$$= \delta_2^{t+u-1}(w_2)\delta_2^{t+u-2}(w_2)\ldots\delta_2^{t}(w_2)\delta_2^{t}(w_2) \quad \text{(by the claim)}.$$

Since for each i holds $\delta_2^i(w_2) = \ldots a\delta_2^i(w_1)$ we see from the locally catenative formula above and from (4) that if $(n_0,i_1,i_2,\ldots,i_k)$ is a locally catenative formula for G_2 then $i_k \geq u$ and

(5) $\quad$ depth(loc. cat. formula for G_2) $\geq u$.

In Section 3.1.1 the maximum cardinality of a finite DOL language over n letters was studied. Let u(n) be the maximum *period* of a finite DOL language over n letters, i.e.,

$$u(n) = \sup\{u \mid G = \langle W,\delta,w\rangle \text{ with } \#W = n \text{ is a DOL system generating a finite language with } u \text{ defined by (3) and (4)}\}.$$

Then, according to Section 3.1.1,

$$u(n) = \sup\{\text{l.c.m.}(k_1,k_2,\ldots,k_q) \mid k_1,k_2,\ldots,k_q \text{ is a partition of } n \text{ in } q \leq n \text{ positive integral summands}\}$$

and

$$\lim_{n\to\infty} \frac{\log u(n)}{\sqrt{n \log n}} = 1.$$

Hence also

$$\lim_{n\to\infty} \frac{\log u(n-1)}{\sqrt{n \log n}} = 1,$$

and by (5) $d(n) \geq u(n-1)$ for all n. Therefore,

$$\lim_{n\to\infty} \inf \frac{\log d(n)}{\sqrt{n \log n}} \geq 1. \qquad \Box$$

COROLLARY 3.32.

$$\liminf_{n\to\infty} \frac{\log k(n)}{\sqrt{n \log n}} \geq \liminf_{n\to\infty} \frac{\log c(n)}{\sqrt{n \log n}} \geq \liminf_{n\to\infty} \frac{\log d(n)}{\sqrt{n \log n}} \geq 1.$$

where we can substitute $\sqrt{p_n}$ *for* $\sqrt{n \log n}$ *in the formulas by the well-known asymptotic approximation of the* n*-th prime number* p_n.

Recently, LINNA [1977] showed that it is decidable whether a DOL string sequence has the *prefix* property, that is, whether there are integers n_0, d and a word v such that $\delta^n(w) = \delta^{n-d}(w)\delta^{n-n_0}(v)$ for all $n \geq n_0$. Although clearly a step forward, this result does not appear to generalize so as to solve the general problem of deciding whether a DOL system has the locally catenative property. Furthermore, EHRENFEUCHT and ROZENBERG [1978] have shown that if we choose a depth d then we can decide whether a given DOL system is locally catenative with a formula of depth at most d. This result does not seem to generalize either.

Finally, we provide an equivalent form of the locally catenative property, which links this property of the derived dequence with a property of the derived language.

THEOREM 3.33. *Let* G *be a* DOL *system. The following two statements are equivalent:*

(i) G *is locally catenative*

(ii) *The monoid* $L(G)^*$ *is finitely generated.*

PROOF.

(i) → (ii). Let $G = \langle W,\delta,w\rangle$ be a locally catenative DOL system with formula $(n_0,i_1,i_2,\ldots,i_k)$. Then $L(G) \subseteq \{\delta^i(w) \mid i < n_0\}^* \subseteq L(G)^*$ and therefore $L(G)^* = \{\delta^i(w) \mid i < n_0\}^*$.

(ii) → (i). Suppose $L(G)^* = \{v_1,v_2,\ldots,v_\ell\}^* \subseteq W^*$. Without loss of generality we can assume that $v_i \notin \{v_1,v_2,\ldots,v_{i-1},v_{i+1},\ldots,v_\ell\}^*$ for all i, $1 \leq i \leq \ell$. Hence $v_i \in L(G)$ for all i, $1 \leq i \leq \ell$, and there is a j, $i \leq j \leq \ell$, such that $v_j = \delta^t(w)$ for some t and for no j', $1 \leq j' \leq \ell$, $v_{j'} = \delta^{t'}(w)$ with $t' > t$. Hence there exist $j_1,j_2,\ldots$ $\ldots, j_k$ such that $\delta^{t+1}(w) = v_{j_1}v_{j_2}\ldots v_{j_k}$ and therefore there are $i_1,i_2,\ldots,i_k$ such that $\delta^{t+1}(w) = \delta^{t+1-i_1}(w)\delta^{t+1-i_2}(w)\ldots$ $\ldots\delta^{t+1-i_k}(w)$ where $\delta^{t+1-i_h}(w) = v_{j_h}$ for all h, $1 \leq h \leq k$. By Lemma 3.29 G is locally catenative. □

3.1.2.3. REGULARITY AND CONTEXT FREENESS

In SALOMAA [1975b] it is proven that the regularity and context freeness of DOL languages are decidable. Roughly, this is achieved as follows. Given a DOL system G, with at most a linear growth function, we can construct (a decomposition of G in) DOL systems $G_1, G_2, \ldots, G_k$ such that $L(G) = h(L(G_1) \cup L(G_2) \cup \ldots \cup L(G_k))$ where h is a nonerasing homomorphism. $G_1, G_2, \ldots, G_k$ satisfy restrictions like: there are no mortal letters in G_i and every letter from the alphabet of G_i occurs in each word in $L(G_i)$. Salomaa then gives a definition of the degree of a DOL system G satisfying said restrictions and proves:

<u>LEMMA 3.34</u>. (SALOMAA). *If* G *has degree* ≤ 1 *then* L(G) *is regular. If* G *is of degree* > 4 *then* L(G) *is non-context free. If* G *is of degree* 2, L(G) *is context free and possibly regular. If* G *is of degree* 3 *or* 4, L(G) *is non-regular (but possibly context free). It is decidable which of the alternatives hold in the last two sentences.*

Since a DOL system can only generate a context free language if its associated growth function is bounded by a linear polynomial we have the following. If L(G) is context free then RS(G) contains paths of at most length 1 and $E = \emptyset$. We can improve on Salomaa's results by showing that under a slightly modified definition of degree decomposition of G is not necessary.

For the vital letters of a DOL system $G = \langle W, \delta, w \rangle$ we define the *degree* as follows. (N.B. Not all vital letters have a degree).

$$\begin{array}{ll} \text{degree (a)} = 0 & \text{if } U(a) \cap (R-MR) = \emptyset;\ \mathcal{O} = \{a \mid \text{degree}(a) = 0\}, \\ \text{degree (a)} = 2 & \text{if } U(a) \cap (R-MR) = [a] \text{ and } \delta^i(a) = v_1 a v_2 \\ & \text{for some } i \leq \#W \text{ and } v_1, v_2 \in (\mathcal{O} \cup M)^* \mathcal{O} (\mathcal{O} \cup M)^*, \\ \text{degree (a)} = 1 & \text{if } U(a) \cap (R-MR) = [a] \text{ and } \delta^i(a) = v_1 a v_2 \text{ or} \\ & v_2 a v_1 \text{ for some } i \leq \#W \text{ and } v_1 \in (\mathcal{O} \cup M)^* \mathcal{O} (\mathcal{O} \cup M)^*, \\ & v_2 \in M^*. \end{array}$$

The degree of G is found by adding the degrees of all vital letters in $\delta^{\#(W-R)}(w)$ where each letter is counted as many times as it occurs. Note that f_G is linear iff all letters occurring in $\delta^{\#(W-R)}(w)$ have a degree or are mortal.

THEOREM 3.35. *Under the given definition of the degree of a* DOL *system, Lemma* 3.34 *holds for arbitrary* DOL *systems.*

INDICATION OF PROOF. The degree of a letter is invariant if we substitute δ by δ^k in the definitions, i.e., under decomposition. Furthermore, the degree of a letter is invariant under restriction of δ to the vital letters, or equivalently, if G has degree i then the PDOL G', constructed such that there is a nonerasing homomorphism h such that $hS(G') = S(G)$, has degree i. Therefore each G_i, $1 \le i \le k$, in the above decomposition of G in $G_1, G_2, \ldots$ $\ldots, G_k$ has the degree of G. □

Since each letter in $[a] \in R/\sim$ must have the same degree in G (if f_G is bounded by a linear polynomial) we say degree $[a]$ = degree (a). If degree $[a] = 1,2$ then $[a] \subseteq R - (MR \cup E)$ and $[b] < [a] \Rightarrow b \subseteq MR$. (N.B. $[b] < [a]$ if $b \in U(a)$ and $a \notin U(b)$.) By now we have obtained some good criteria to prove that a language does not belong to a given language family:

COROLLARY 3.36.

- L(G) *is finite iff* $\sum_{[a]\in R/\sim}$ *degree* $[a] = 0$.
- *If* L(G) *is regular then* $\sum_{[a]\in R/\sim}$ *degree* $[a] \le 2$.
- *If* L(G) *is context-free then* $\sum_{[a]\in R/\sim}$ *degree* $[a] \le 4$.
- *If* L(G) *is infinite and locally catenative then* $E = [b]$ *for some letter* b *and* $\sum_{[a]\in R/\sim - \{[b]\}}$ *degree* $[a] = 0$.

3.1.2.4. BIOLOGICAL INTERPRETATION

In biology we encounter the phenomenon of cell differentiation as opposed to cell potential. In higher species cells become so specialized (highly differentiated) that they lose their ability to produce cells of other types (low potential). In the embryonic stage, and to a large extent in the vegetative kingdom this seems not to be the case (low differentiation and high potential). The associated digraphs, as in I-IV, form in increasing levels of abstraction a formal representation of cell lineage and cell differentiation of an organism modeled by a DOL system. In I the AD depicts the cell lineage. The CAD in II shows us the stages of cell differentiation where the labels consisting of sets of recursive letters correspond to, as it were, meta-stable stages of cell differentiation, i.e., the descendancy of such a cell always contains a cell with the same cell potential as the original one, and each cell type of a meta-stable stage of differentiation occurs in the descendancy of each other cell type of this stage. The points labeled by singleton sets of vital nonrecursive letters correspond to transitory stages of cells between one meta-stable stage of cell differentiation and a next one. The RS shows us the lineage between the meta-stable stages which is of prime importance and the URS the same structure without labels.

EXAMPLES.

(i) If the CAD consists of the graph on one point the modeled organism is very regenerative: each cell type has the possibility of deriving any other cell type.

(ii) If the CAD consists of a directed tree we observe a type of cell differentiation similar to that in higher organisms. Cells in the leaves of the tree are completely specialized and have no regenerative capacity to produce cells of other types in their progeny, as opposed to the cells at the root which can produce all other cell types.

(iii) To be able to reproduce from a single cell, the CAD of the associated DOL system must be such, that every two points of the CAD have a common ancestral point while the unique maximal element is labeled by an equivalence class of recursive letters. The rules must be such that at any time the description of the organism (i.e., the produced string) contains a cell in the maximal point of the CAD. All living plants and animals seem always to contain some cells which are capable of division, and through that to give rise to cells from which a new similar organism can be derived.

To interpret some of the results in this Section 3.1.2:

If an organism grows under optimal conditions (and if it can be adequately modeled by a DOL system) it exhibits linear growth iff it has exactly two meta-stable levels of cell differentiation. More generally, if it exhibits polynomial growth of degree n it has exactly n + 1 meta-stable levels of cell differentiation (by this we mean that if we trace the cell lineage from a least differentiated cell to a most differentiated cell there is at least one cell lineage such that we meet n + 1 different meta-stable stages of differentiation).

If an organism has the locally catenative property, i.e., if at time t the organism is composed from the previous stages in its developmental history, as in ROZENBERG and LINDENMAYER [1973], it contains at most two meta-stable levels of differentiation and it can be grown from cells occurring in a single uppermost meta-stable stage of differentiation. The RS is a tree of at most two levels, with a meta-stable stage of cell differentiation at the top from which all other completely differentiated cell types are derived without intermediate meta-stable stages of differentiation. Another result shows that if a relatively simple organism, i.e. one having not many different cell types, is locally catenative we might have to wait a very long time to see that it is such.

In general we can think of the URS, or the genealogical relations between meta-stable stages of cell differentiation, as a measure of the complexity of the organism, see e.g. Corollary 3.36.

3.2. DETERMINISTIC CONTEXT SENSITIVE LINDENMAYER SYSTEMS WITHOUT TABLES.

From the point of view of developmental biology, the language consisting of the set of all strings generated by the system is of primary interest. Such an L language is taken to correspond to the set of all developmental stages which might be attained by the organism during its development. Here, also, homomorphic mappings (especially those in which a letter is mapped to a letter) are of considerable importance, cf. NIELSEN, ROZENBERG, SALOMAA and SKYUM [1974a,b].

More formal language oriented investigators, however, divide the set of letters used by the L system into a set of terminals and a set of nonterminals. The language obtained from the L system by the use of this mechanism consists of all the strings over the terminals in the pure L language (the set of all strings generated by the system). Such languages are called extensions of L languages. Families of extensions of L languages usually have welcome mathematical properties, such as closure under certain operations. One of the facts which have made the use of nonterminals interesting within the theory of developmental languages is that it was established in EHRENFEUCHT and ROZENBERG [1974a,b] that for basic families of 0L systems the use of nonterminals and the use of letter-to-letter homomorphisms are equivalent as far as the generating capacity is concerned. Thus, the tradeoff between the two language-defining mechanisms (i.e., nonterminals versus homomorphisms) has become a very interesting and well-motivated problem for

L systems. Continuing this train of thought, trade-offs between combinations of one- or two-sided context, restrictions where no letter is rewritten as the empty word, and the use of nonterminals and various kinds of homomorphisms are interesting. The present section is concerned with this topic, especially with respect to language classification, but we restrict our attention to the deterministic L systems.

These systems are particularly relevant in the biological setting, as would also appear to be indicated by the fact that most attempts to provide L systems modeling the development of actual biological organisms use deterministic systems; see the references in the Introduction. Furthermore, it can be noted that the study of the change in pattern, size and weight of a growing organism as a function of time constitutes a considerable portion of the literature on developmental biology. Usually, genetically identical specimens of a specific organism are investigated in a controlled environment and their changes with respect to time are described. The scientific presupposition is that identical genetical material and identical environment will result in an identical developmental history, i.e., that the experiment is repeatable. This assumes a deterministic (causal) underlying structure, and makes a good case for the biological importance of the study of deterministic L systems.

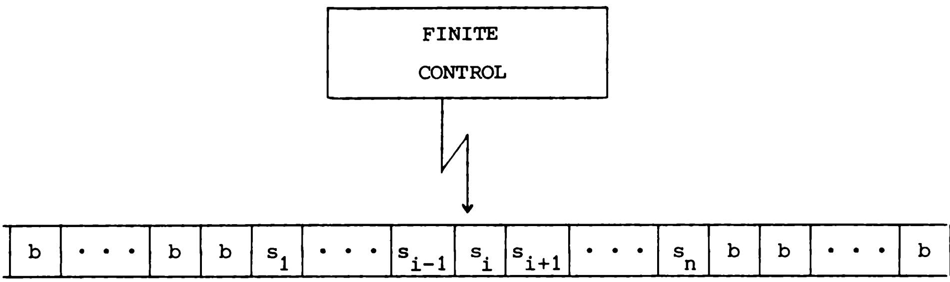
This section can be divided in three parts. In Subsection 3.2.1 we relate L systems to Turing machines as in van DALEN [1971] or HERMAN [1969]. Subsections 3.2.2 and 3.2.3 are concerned with (the classification of) pure deterministic L languages, i.e., the languages consisting of all strings generated by the systems. In Subsections 3.2.4 and 3.2.5 we deal with extensions of deterministic L languages, i.e., languages consisting of all the strings over some terminal alphabet which are generated by the system.

In Subsection 3.2.2 we are interested in Lindenmayer languages which are not recursive and we develop a simulation technique which will prove to be useful in the sequel. In 3.2.3 families of deterministic L languages are compared with the Chomsky hierarchy. Families of extensions of deterministic L languages are classified in the Chomsky hierarchy in 3.2.4, and in 3.2.5 we consider extensions and homomorphic closures of families of languages generated by deterministic L systems with the propagating property. As is well known, such a restriction (on erasing) usually drastically limits the generating power of a rewriting system. To give some examples of typical results which we shall encounter: in 3.2.4 it is shown that the amount of context needed for rewriting makes no difference for families of extensions; the only differences lie in no context, one-sided context and two-sided context. The family of extensions of D2L languages equals the family of recursively enumerable languages, as does the closure under letter-to-letter homomorphisms of the family of extensions of D1L languages. On the other hand, the family of extensions of D1L languages does not even contain all regular languages. In 3.2.5 it appears that the family of extensions of PD2L languages is equal to the family of languages accepted by deterministic linear bounded automata (to be defined in that section). The closure under nonerasing homomorphisms of the family of extensions of PD1L languages is strictly included in the family of extensions of PD2L languages. Indeed, this closure does not even contain languages like $\{a_1,a_2,\ldots,a_n\}^* - \{\lambda\}$, $n \geq 2$. (Contrast this with the result for the nonpropagating case in 3.2.4.) On the other hand, the closure of the family of PD1l languages under homomorphisms which map a letter either to itself or to the empty word is again equal to the family of recursively enumerable languages. At the end of Section 3.2.4 we consider the question whether all finite languages are generated by a given class of L systems. As is easy to prove, for each class of D(m,n) L systems there are finite languages which cannot be generated, but all finite languages are obtainable as a letter-to-letter homomorphism of the PD0L languages. With regard to the extension operation the situation is not so clear: each finite language can be obtained as an extension of a PD2L language but with one-sided or no context it will appear in Theorem 3.54 that the class of finite languages is not contained in ED0L or EPD1L, but is contained in ED1L. In Section 3.2.6 we combine the results in Section 3.2 to obtain a coherent picture of the power of parallel rewriting with respect to the set of the various additional operations or mechanisms.

The strict inclusion results obtained follow from necessary properties of the concerned language families rather than by an exhaustive analysis of a particular example. By stating the results in their strongest form, we obtain a systematic classification of the effect of the discussed mechanisms on the generating power of deterministic L systems using context. For a treatment of the effect of nonterminals, homomorphisms and letter-to-letter homomorphisms in different variations of context free L systems the reader is referred to NIELSEN, ROZENBERG, SALOMAA and SKYUM [1974a,b].

3.2.1. LINDENMAYER SYSTEMS AND TURING MACHINES

For our purposes, a *Turing machine* is an abstract device consisting of a *finite control* attached to a *read-write* head scanning a both ways infinite (or infinitely expandable) *tape* which is divided into *squares*. Each square can contain one out of a finite nonempty set of *symbols* S. There is one distinguished symbol called the *blank* symbol b.



TURING MACHINE

All squares not yet scanned by the read-write head are assumed to contain the symbol b. The finite control can be in anyone of a finite nonempty set of *states* Ψ. According to the state q of the finite control and the symbol s in the tapesquare under scan of the read-write head, the machine replaces s by a symbol s', moves the head one square left, right or not at all and enters a state q'. Hence the action of the machine is completely determined by a, possibly partial, function from $S \times \Psi$ into $S \times \{$left,right, no move$\} \times \Psi$ which can be described by the relevant finite set of quintuples. The machine halts when it enters a halting state $q \in F$ where $F \subseteq \Psi$ is the set of *halting states*. Starting in a distinguished *start state* q_0 with the head scanning the leftmost nonblank symbol, a Turing machine computes a (possibly partial) function from the input (i.e., the nonblank tape contents at time 0) to the ouput (i.e., the nonblank tape contents at the time the machine halts). Every (partial) *recursive* function can be computed in this way by a Turing machine. An *instantaneous description* (ID) is a snapshot of the machine configuration at a particular instant of time; it consists of the state of the finite control, the tape contents and the position of the read-write head on the tape. An ID is often denoted as $s_1 s_2 \ldots s_{i-1} q s_i$ $s_{i+1} \ldots s_n$ where s_1 is the leftmost nonblank symbol (or its blank left neighbor) on the tape and s_n is the rightmost nonblank symbol, or its blank right neighbor, while the finite control in state q is scanning the tape-square containing s_i by means of the head. The Turing machine as we have described it is *deterministic* since the transition function (set of quintuples) uniquely determines the ID at time t + 1 from the ID at time t. A *nondeterministic* Turing machine is defined analogously but with the feature that the partial function from $S \times \Psi$ into $S \times \{\text{left,right,no move}\} \times \Psi$ is replaced by a relation between the two sets concerned, i.e., there can be a nondeterministic choice of a next move from among a finite set of alternatives. A more extensive treatment of Turing machines, variants thereof, terminology and results can be found in MINSKY [1967] or in HOPCROFT and ULLMAN [1969]. The device was introduced by TURING [1936].

Here we need the following. A language L is said to be *accepted* by a Turing machine T if for all words in L as input T halts in an accepting state $q_f \in F \subseteq \Psi$, where F is a fixed set of accepting states. (If T is of the nondeterministic variety we only require that there is a sequence of choices of next moves for each input word taken from L which drives T in an accepting state). I.e., for an input consisting of a word not in L the Turing machine either halts not at all or halts in a nonaccepting state (under all possible choice sequences in the case of the nondeterministic variety). A language is *recursively enumerable* if it is accepted by a deterministic (equivalently, nondeterministic) Turing machine. A language is *recursive* if it is accepted by a deterministic Turing machine and its complement is accepted by a deterministic Turing machine. In the sequel of Section 3.2 we assume that all Turing machines we consider are deterministic.

It was shown by van DALEN [1971] that for a suitable standard definition of Turing machines (e.g. the quintuple version above), for every Turing machine with symbol set S and state set Ψ we can effectively construct a D2L system $G = \langle W,\delta,w\rangle$, with $W = \Psi \cup S$, which simulates it in real time, viz., the t-th instantaneous description of T is equal to $\delta^t(w)$. If we do away with the excess blank symbols on the ends of the Turing machine tape, by letting the letters corresponding to such blank symbols derive the empty word λ in the L system simulation of T, then the following statement clearly holds. Let $G = \langle W,\delta,w\rangle$ be a D2L system, let S and Ψ be disjoint subsets of W and let h be a homomorphism from $S^*\Psi S^*$ into S^* defined by $h(q) = \lambda$ for all $q \in \Psi$ and $h(a) = a$ for all $a \in S$. The set of languages of the form $h(L(G) \cap S^*\Psi S^*)$ is the family of recursively enumerable languages. Since the family of recursive languages is closed under intersection with a regular set and k-limited erasing, and since there exist recursively enumerable languages that are not recursive, there exist D2L languages which are not recursive*) ($S^*\Psi S^*$ is regular and h is 1-limited on $S^*\Psi S^*$). That all L languages are recursively enumerable follows by the usual Turing machine simulation argument.

3.2.2. NONRECURSIVE L LANGUAGES

At the end of the last section we gave the usual proof that there are nonrecursive D2L languages. By an application of a result due to RABIN and WANG [1963] we can be slightly more specific and at the same time develop a simulation technique which will be of use in the sequel. Let the *word* at any moment t in the history of a Turing machine be the string consisting of the contents of the minimum block on the tape at t that includes all the marked squares and the square scanned at the initial moment (the origin).

<u>THEOREM 3.37</u>. (RABIN and WANG). *For any fixed (finite) word at the initial moment we can find a Turing machine* T *such that the set of words* P *in its subsequent history is not recursive.*

<u>THEOREM 3.38</u>. *Let* G_T *be a* D2L *which simulates (in the sense explained in Section* 3.2.1*) a Turing machine* T *satisfying the statement of Theorem* 3.37. *Then* $L(G_T)$ *is nonrecursive.*

<u>PROOF</u>. Let h be a homomorphism on $L(G_T)$ defined by $h(a) = a$ and $h(q) = \lambda$ for all $a \in S$ and all $q \in \Psi$, where S and Ψ are the symbol set and the state set of T, respectively. Since $L(G_T) \subseteq S^*\Psi S^*$, h is 1-limited on $L(G_T)$. $h(L(G_T)) = P$ and since P is nonrecursive $L(G_T)$ is nonrecursive. □

*) A family of languages is said to be closed under *k-limited erasing* if, for any language L of the class and any homomorphism h with the property that h never maps more than k consecutive symbols of any sentence x in L to λ, h(L) is in the class. We shall furthermore be concerned with *nonerasing homomorphisms*, i.e. homomorphisms which map no letter to the empty word λ; *letter-to-letter homomorphisms* (also called *codings*), i.e. homomorphisms which map letters to letters; and homomorphisms which map a letter either to itself or to the empty word λ (these homomorphisms are a subclass of the *weak codings* where a letter is mapped either to a letter or to λ). For further details concerning homomorphisms and other operations on languages and closure under these operations see HOPCROFT and ULLMAN [1969] or SALOMAA [1973a].

We use G_T to construct a nonrecursive D(0,1)L language.*)

LEMMA 3.39. *Let* $G = \langle W,\delta,w\rangle$ *be any* D2L. *There is an algorithm which, given* G, *produces a* D(0,1)L $G' = \langle W',\delta',w'\rangle$ *such that for all* t, $\delta'^{2t}(w') = \not{c}\delta^{t}(w)$ *and*

$$\delta'^{2t+1}(w') = \not{c}'(a_1,a_2)(a_2,a_3)\dots(a_k,\lambda)$$

if $\delta^t(w) = a_1a_2\dots a_k$, *where* $\not{c}$ *and* $\not{c}'$ *are letters not in* W.

PROOF. Construct $G' = \langle W',\delta',w'\rangle$ as follows.

$$W' = W \cup (W \times (W \cup \{\lambda\})) \cup \{\not{c},\not{c}'\},$$

where $\not{c}$ and $\not{c}'$ are letters not in W.

$$\begin{aligned}
&w' = \not{c}w,\\
&\delta'(\lambda,a,c) = (a,c),\\
&\delta'(\lambda,\not{c},c) = \not{c}',\\
&\delta'(\lambda,\not{c}',\lambda) = \not{c},\\
&\delta'(\lambda,(a,b),(b,c)) = \delta(a,b,c),\\
&\delta'(\lambda,\not{c}',(a,c)) = \not{c}\delta(\lambda,a,c),\\
&\delta'(\lambda,(a,\lambda),\lambda) = \lambda,
\end{aligned}$$

for all $a,b \in W$ and all $c \in W \cup \{\lambda\}$. (The arguments for which δ' is not defined will not occur in our operation of G'.)

For all words $v = a_1a_2\dots a_k \in W^*$ we have

---

*) Similarly to the case of DOL systems, where we extended the mapping $\delta: W \to W^*$ to a homomorphism $\delta: W^* \to W^*$ we will extend the mapping $\delta: \bigcup_{i=0}^{m} W^i \times W \times \bigcup_{j=0}^{n} W^j \to W^*$ to a mapping $\bar{\delta}: W^* \to W^*$ for D(m,n)L systems as follows. $\bar{\delta}(v) = v'$ iff $v \Rightarrow v'$ and $\bar{\delta}^i(v) = v'$ iff $v \overset{(i)}{\Rightarrow} v'$. When no confusion can result we identify δ and $\bar{\delta}$ and denote both mappings by δ. If we want to set off the difference between δ and $\bar{\delta}$, as is the case in, e.g., the proof of Lemma 3.39 we use the notations δ and $\bar{\delta}$.

$$k > 1:\ \bar{\delta}'^{2}(\not{c}a_1a_2\ldots a_k) = \bar{\delta}'(\not{c}'(a_1,a_2)(a_2,a_3)\ldots(a_k,\lambda))$$

$$= \not{c}\delta(\lambda,a_1,a_2)\delta(a_1,a_2,a_3)\ldots\delta(a_{k-1},a_k,\lambda)$$

$$= \not{c}\bar{\delta}(a_1a_2\ldots a_k);$$

$$k = 1:\ \bar{\delta}'^{2}(\not{c}a_1) = \bar{\delta}'(\not{c}'(a_1,\lambda)) = \not{c}\delta(\lambda,a_1,\lambda) = \not{c}\bar{\delta}(a_1);$$

$$k = 0:\ \bar{\delta}'^{2}(\not{c}) = \bar{\delta}'(\not{c}') = \not{c} = \not{c}\bar{\delta}(\lambda).$$

Therefore, for all t, $\bar{\delta}'^{2t}(\not{c}w) = \not{c}\bar{\delta}^{t}(w)$ and

$$\bar{\delta}'^{2t+1}(\not{c}w) = \not{c}'(a_1,a_2)(a_2,a_3)\ldots(a_k,\lambda)$$

if

$$\bar{\delta}^{t}(w) = a_1a_2\ldots a_k. \qquad \square$$

From Lemma 3.39 we see that if $L \in$ D2L then there is an $L' \in$ D(0,1)L (respectively $L'' \in$ D(1,0)L) such that $\{w \mid \not{c}w \in L'\} = L$ (respectively $\{w \mid w\not{c} \in L''\} = L$).

The following two corollaries illustrate some more relations between D1L and D2L languages.

COROLLARY 3.40. *Let* $G = \langle W,\delta,w\rangle$ *be a* D2L. *There is an algorithm which, given* G, *produces a* D(0,1)L G' *(respectively a* D(1,0)L G"*) and a letter-to-letter homomorphism* h *such that* $h(L(G')) = \{\not{c}\}L(G)$ *(respectively* $h(L(G'')) = L(G)\{\not{c}\}$).

(Hint: Let h be a letter-to-letter homomorphism defined by $h(a) = a$ for all $a \in W \cup \{\not{c}\}$, $h(\not{c}') = \not{c}$, and $h((a,b)) = a$ for all $(a,b) \in W \times (W \cup \{\lambda\})$.)

COROLLARY 3.41. *Let* $G = \langle W,\delta,w\rangle$ *be any* D2L. *There is an algoritm which, given* G, *produces a* D(0,1)L G' *(respectively* D(1,0)L G"*) and a homomorphism* h, *which maps a letter either to itself or to* λ, *such that*

$$h(L(G') \cap \{\not{c}\}W^{*}) = h(L(G'') \cap W^{*}\{\not{c}\}) = L(G).$$

(Hint: h is defined by $h(a) = a$ for all $a \in W$ and $h(\not{c}) = \lambda$. h is 1-limited on $\{\not{c}\}W^*$ and $W^*\{\not{c}\}$.)

THEOREM 3.42. *We can construct* D1Ls *whose languages are not recursive.*

PROOF. Let $G_T = \langle W_T, \delta_T, w_T \rangle$ be a D2L as in Theorem 3.38. By Corollary 3.41 we can construct a D(0,1)L G' such that $h(L(G') \cap \{\not{c}\}W_T^*) = L(G_T)$. Since $\{\not{c}\}W_T^*$ is regular, h is a 1-limited homomorphism on $\{\not{c}\}W_T^*$, and $L(G_T)$ is not recursive, it follows that L(G') is not recursive. □

3.2.3. DETERMINISTIC L LANGUAGES AND THE CHOMSKY HIERARCHY

From the work of van DALEN [1971], ROZENBERG [1972a,b] and ROZENBERG and LEE [1975] on nondeterministic L systems we can readily deduce several facts about the place in the Chomsky hierarchy of the deterministic languages: e.g. the PD1L languages are strictly included in the context sensitive languages, the D1L languages are strictly included in the recursively enumerable languages. By the use of direct arguments concerning the deterministic nature of the systems under consideration we shall refine these results implicit in the above references and fix completely the place of the D(m,n)L- and PD(m,n)L languages with respect to the four main classes of the Chomsky hierarchy.

LEMMA 3.43. *There are regular languages over a one letter alphabet which are not* DIL *languages.*

PROOF. $L = \{aaa\}^*\{a,aa\}$ is such a language. To prove this we make use of the following:

CLAIM. If $G = \langle W,\delta,w \rangle$ is a *unary* D(m,n)L (i.e. $\#W = 1$) which generates an infinite language then there exist nonnegative integers t_0, p and x such that for all $t \geq t_0$ the following equation holds:

$$(1) \qquad \ell g(\bar{\delta}^{t+1}(w)) = p(\ell g(\bar{\delta}^t(w)) - m - n) + x.$$

PROOF OF CLAIM. Let $\delta(a^m,a,a^n) = a^p$ and let

$$x = \sum_{i=0}^{m-1} \ell g(\delta(a^i,a,a^n)) + \sum_{i=0}^{n-1} \ell g(\delta(a^m,a,a^i)).$$

If L(G) is infinite then there exists a t_0 such that

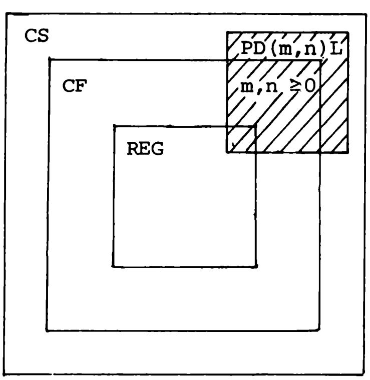
$$\ell g(\bar{\delta}^{t_0}(w)) \geq 2(m+n) + x + 1.$$

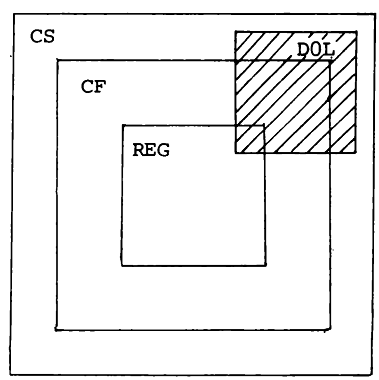
<u>Case 1</u>. $p = 0$. $\ell g(\bar{\delta}^t(w)) \leq y$ for all $t > 0$ where $y = \max\{\ell g(\bar{\delta}(a^k)) \mid k \leq m+n\}$, contrary to the assumption.

<u>Case 2</u>. $p > 0$. Clearly (1) holds. *End of proof of Claim.* By observing that $L = \{a^i \mid i \not\equiv 0 \bmod 3\}$ we see that for every positive integer k such that $k \equiv 0 \bmod 3$ holds that $a^{k-1}, a^{k+1}, a^{k+2} \in L$ and $a^k \notin L$. Hence, if $L(G) = L$ it follows that $p = 1$ in (1). But then the lengths of the subsequent words in L(G), ordered by increasing length, differ by a constant amount $x - m - n$ and hence $L(G) \neq L$. $\square$

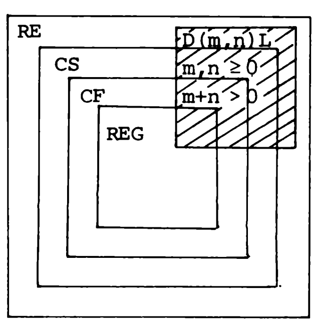
<u>THEOREM 3.44</u>. *The inclusion relations between the various classes of deterministic* L *languages and the main language classes of the Chomsky hierarchy are as follows.*

(i)





(ii)



(iii) *For all* $m,n \geq 0$, $PD(m,n)L \subset D(m,n)L$; $PDIL \subset DIL$.

PROOF. (i) and (ii). Let G_1, G_2 and G_3 be PD0L systems defined by

$$G_1 = \langle\{a\},\{\delta(a)=a\},a\rangle,$$

$$G_2 = \langle\{a,b,c\},\{\delta(a)=a,\ \delta(b)=b,\ \delta(c)=acb\},c\rangle,$$

$$G_3 = \langle\{a\},\{\delta(a)=aa\},a\rangle.$$

$L(G_1) = \{a\}$, $L(G_2) = \{a^n cb^n \mid n \geq 0\}$ and $L(G_3) = \{a^{2^n} \mid n \geq 0\}$. $L(G_1) \in$ REG; it is well known that $L(G_2) \in$ CF - REG; and $L(G_3) \in$ CS by the working space theorem or the usual linear bounded automaton argument and $L(G_3) \notin$ CF by the uvwxy-lemma. (The working space theorem is a variant of the linear bounded automaton lemma which tells us that the family of languages accepted by linear bounded automata is equal to CS. For a definition of linear bounded automata see e.g. Section 3.2.5. For a more extensive discussion or for the uvwxy-lemma see e.g. HOPCROFT and ULLMAN [1969] or SALOMAA [1973 a]). This proves that all considered families of L languages have nonempty intersections with REG, CF - REG and CS - CF. By Theorem 3.42 there are D1L languages which are not recursive and therefore not context sensitive. Hence there are D1L languages in RE - CS.

The language L from Lemma 3.43 belongs to REG but not to DIL. $L \cup L(G_2) \in$ CF - REG and it is easy to show that $L \cup L(G_2) \notin$ DIL. $L' = \{a^{2^{(2^n)^2}} \mid n \geq 0\}$ does not belong to DIL because of equation (1) but L' belongs to CS because of the working space lemma and $L' \notin$ CF because of the uvwxy lemma. Each nonrecursive but recursively enumerable language $A \subseteq \{1\}^*$ belongs to RE - CS but not to DIL in view of equation (1). (Note: languages satisfying equation (1) are recursive.) Hence there are languages in REG, CF - REG, CS - CF and RE - CS which are not in DIL.

(iii) $PD(m,n)L \subseteq D(m,n)L$ holds by definition. Assume that $m+n > 0$. By Theorem 3.42 there are nonrecursive $D(m,n)L$ languages, hence belonging to RE - CS, but all $PD(m,n)L$ languages are in CS. It is easy to give nontrivial examples of D0L languages which are not PD0L languages, thereby covering the case $m = n = 0$. Hence the above inclusion is strict. Similarly we prove $PDIL \subset DIL$. □

From equation (1) it follows immediately that $D(m,n)L \subset D(m',n')L$ for

$m < m'$ and $n = n'$, or $m = m'$ and $n < n'$, or $m < m'$ and $n < n'$. In particular, D0L $\subset$ D1L $\subset$ D2L. Analogously this holds with the propagating restriction added. For a further discussion of the inclusion relations between families of L languages using different amount of context, see ROZENBERG [1972a,b] and ROZENBERG and LEE [1975].

3.2.4. EXTENSIONS AND HOMOMORPHIC CLOSURES OF DETERMINISTIC L LANGUAGES

The favorite device in formal language theory for extracting languages from rewriting systems is the use of of nonterminals, i.e., by selecting from the set of produced words all those words which are over the terminal alphabet. (This device allows us, as it were, to get rid of the intermediate work necessary to generate the desired word over the terminals by the rewriting system, so that these intermediate strings do not show up in the related language.) This operation is called intersection with a terminal alphabet; it usually considerably contributes to the generating power of a system and is therefore called an extension. For instance, in a pure L language, if we order the words in the language according to their lengths, there is always a constant c such that the length of the i + 1-th word is less than or equal to c times the length of the i-th word. This is due to the fact that all words used to derive the i + 1-th word belong to the language. For extensions of L languages this property does not hold. The *extension* (language) produced by an XL system $G = \langle W,P,w \rangle$ with respect to a terminal alphabet V_T is defined as $E(G,V_T) = L(G) \cap V_T$. We also call the quadruple $G' = \langle W,P,w,V_T \rangle$ an EXL system. Considering nondeterministic L systems, van DALEN [1971] proved that E1L = RE and EP2L = CS. Furthermore, we can easily show (by the working space theorem) that E0L $\subseteq$ CS. For deterministic L systems it therefore follows that ED1L $\subseteq$ ED2L $\subseteq$ RE; EPD1L $\subseteq$ EPD2L $\subseteq$ CS (and in general by the working space theorem that EPDIL $\subseteq$ CS) and ED0L $\subseteq$ CS. It, furthermore, follows immediately from the definitions that XL $\subseteq$ EXL for all classes of XL systems.

THEOREM 3.45. ED2L = RE.

PROOF. Let A be a recursively enumerable language over some alphabet V_T which is enumerated by a 1: 1 recursive function $f\colon \mathbb{N} \xrightarrow{1:1} A$; n is recovered from f(n) by f^{-1}. That every infinite recursively enumerable language can be enumerated by a one-one recursive function follows from ROGERS [1967, Exercise 5.2]; for finite languages clearly an appropriate version of our proof suffices. Let T be a Turing machine with symbol set $S = V_T \cup \{a,b\}$ where $a,b \notin V_T$ and b is the blank symbol. At time $t = 0$, T is presented with a finitely inscribed tape of which the origin contains a. We assume that the tape is halfway infinite, i.e., the reading head of T never scans a square left of the origin. That this is no restriction on the power of a Turing machine is well known. T starts with erasing the finitely many marks on its tape except the symbol a at the origin, returns to the origin, writes the representation of 0 on the tape and calculates the value of f(0). Subsequently, T erases everything else except the representation of f(0), retrieves the representation of 0 from f(0) by f^{-1}, adds one to this representation and computes f(1), and so on. In particular we can do this in such a way that the specific symbol a is used only to mark the origin and is erased only to indicate f(0),f(1),...; it is printed again before we calculate $f(n+1)$ from f(n). If P is the set of all words in the history of T then $P \cap \{b\}V_T^* = \{b\}A$. Let $G_T = \langle W_T,\delta_T,w_T\rangle$ be a D2L which simulates T in the sense of Section 3.2.1. Since T uses a halfway infinite tape the strings of G_T always have a letter a at the left end except when f(n) has been computed for some n in which case the string has a letter q_{i_n} (indicating the state of the simulated Turing machine) at the left end. That is, for each $n \in \mathbb{N}$ there is a $t_n \in \mathbb{N}$ and a state $q_{i_n} \in \Psi$ (where Ψ is the state set of T) such that $\delta_T^{t_n}(w_T) = q_{i_n}af(n)$. We can construct T with two distinguished states q', q" in Ψ such that (eliminating some superfluous intermediate steps of T in the simulating G_T) for all n:

$$\delta_T^{t_n+1}(w_T) = q'f(n), \qquad \delta_T^{t_n+2}(w_T) = aq''f(n).$$

and q', q" never occur in $\delta_T^t(w_T)$ for $t_n+2 < t \le t_{n+1}$, $n \in \mathbb{N}$. Now we modify G_T to $G = \langle W_T,\delta,w_T\rangle$ where δ is defined by: if $\delta_T(\lambda,q,a) = q'$ then $\delta(\lambda,q,a) = \lambda$, $\delta(\lambda,c,d) = aq''c$ for all letters $c \in V_T$ and $d \in V_T \cup \{\lambda\}$, and $\delta(\cdot) = \delta_T(\cdot)$ for all other arguments. It is easily seen that $\delta^{t_n+1}(w_T) = f(n)$ for all n and $\delta^t(w_T) = \delta_T^t(w_T) \in W_T^*\Psi W_T^*$ for all t such that $t \ne t_n + 1$, $n \in \mathbb{N}$. Hence $L(G) \cap V_T^* = A$. (To capture the case where $\lambda \in A$ we could define $\bar{\delta}(\lambda) = aq''$.) □

<u>THEOREM 3.46</u>. *The closure of* ED(0,1)L *(or* ED(1,0)L*) under letter-to-letter homomorphisms is equal to* RE.

PROOF. We prove the theorem for D(0,1)Ls. The case for D(1,0)Ls is completely analogous. Let $G = \langle W,\delta,w\rangle$ be a D2L constructed as in Theorem 3.45. Let $G' = \langle W',\delta',w'\rangle$ be a D(0,1)L defined as follows:

$$W' = W \cup (W \times (W \cup \{0,1,\lambda\})) \cup \{\not{c}\},$$

where 0, 1, $\not{c}$ are letters not in W,

$$w' = (b_1,1)(b_2,0)\ldots(b_n,0) \quad \text{if } w = b_1b_2\ldots b_n,$$

$$\delta'(\lambda,a,b) = (b,0),$$

$$\delta'(\lambda,\not{c},a) = (a,1),$$

$$\delta'(\lambda,\not{c},\lambda) = \delta'(\lambda,a,\lambda) = \delta'(\lambda,(a,\lambda),\lambda) = \lambda,$$

$$\delta'(\lambda,(a,0),(b,0)) = (a,b),$$

$$\delta'(\lambda,(a,1),(b,0)) = \not{c}(a,b),$$

$$\delta'(\lambda,(a,0),\lambda) = (a,\lambda),$$

$$\delta'(\lambda,(a,1),\lambda) = \not{c}(a,\lambda),$$

$$\delta'(\lambda,(a,b),(b,c)) = \delta(a,b,c),$$

$$\delta'(\lambda,\not{c},(a,c)) = \not{c}\delta(\lambda,a,c),$$

for all $a,b \in W$ and all $c \in W \cup \{\lambda\}$. (The arguments for which δ' is not defined shall not occur in our operation of G'.) Assume that $\lambda \notin L(G)$.

We see that for all t holds that $h(\delta'^{3t}(w')) = \delta^t(w)$ where h is a letter-to-letter homomorphism from $(W \times \{1,0\})^*$ onto W^* defined by $h((a,0)) = h((a,1)) = a$ for all $a \in W$. Since by the synchronicity of the productions $\delta^t(w') \in \{\not{c}\}W'^*$ for all $t \not\equiv 0 \bmod 3$ we have $h(L(G') \cap (W\times\{0,1\})^*) = L(G)$ and therefore $h(L(G') \cap (V_T \times \{0,1\})^*) = L(G) \cap V_T^*$. (To capture the case where $\lambda \in L(G)$ we could define $\bar{\delta}'(\lambda) = \not{c}\bar{\delta}(\lambda)$, and the proof proceeds analogously.) □

THEOREM 3.47. *If* L $\in$ ED2L or, *equivalently,* L $\in$ RE *then* $\{\not{c}\}$L $\in$ ED(0,1)L *(similarly* L$\{\not{c}\}$ $\in$ ED(1,0)L) *where* $\not{c}$ *is a letter not occurring in a word of* L.

PROOF. The theorem follows immediately from Lemma 3.39. $\square$

We shall now prove some properties of D0L and D1L languages which give us criteria to show that certain languages cannot be D0L or D1L languages or their intersections with a terminal alphabet.

We call a language *permutation free* if no word in the language is a permutation of any other word in the language.

LEMMA 3.48. *Let* $G = \langle W,\delta,w\rangle$ *be a* D0L. *If* L(G) *is infinite then* L(G) *is permutation free.*

PROOF. Suppose L(G) is infinite, $v,v' \in L(G)$, $v \neq v'$, and v' is a permutation of v. Let $\delta^k(v) = v'$ for some $k > 0$. Since v' is a permutation of v we have for each $n > 0$: $\delta^{nk}(v)$ is a permutation of v. There are only a finite number of words in W^* which are a permutation of v and therefore there exist $n_2 > n_1 > 0$ such that $\delta^{n_1k}(v) = \delta^{n_2k}(v)$. But $v = \delta^{t_0}(w)$ for some t_0 and therefore $\delta^{t_0+n_1k}(w) = \delta^{t_0+n_2k}(w)$ so L(G) is finite: contradicting the assumption. $\square$

The converse of the lemma holds in the following sense. Let $G = \langle W,\delta, w\rangle$ be a D0L. L(G) is infinite iff for no integers i and j, $i \neq j$, holds that $\delta^i(w)$ is a permutation of $\delta^j(w)$. (We consider λ to be a permutation of λ.)

COROLLARY 3.49. *Let* $G = \langle W,\delta,w\rangle$ *be a* D0L *and* V_T *a subset of* W. *If* $E(G,V_T)$ *is infinite then* $E(G,V_T)$ *is permutation free, i.e., all infinite languages in* ED0L *are permutation free.*

We call a word v' a *prefix (postfix)* of a word v if $v = v'z$ ($v = zv'$) for some word z. We call v' a *proper prefix (proper postfix)* of a word v if v' is a prefix (postfix) of v and $v' \neq v$.

LEMMA 3.50. *Let* $G = \langle W,\delta,w\rangle$ *be a* D(1,0)L (D(0,1)L).

(i) L(G) *is finite iff* $\delta^t(w) = \delta^{t'}(w)$ *for some* t, t' *such that* $t \neq t'$.

(ii) *Let* L(G) *be infinite. If* $v,v' \in L(G)$ *and* v' *is a proper prefix (proper postfix) of* v *then, with finitely many exceptions, for each word* u *in* L(G) *there is a word* u' *in* L(G) *such that* u' *is a proper prefix (postfix) of* u.

PROOF.

(i) Obvious by the deterministic property of G.

(ii) We prove (ii) only for D(1,0)Ls and prefixes. The proof is completely analogous for D(0,1)Ls and postfixes.

Case 1. $\delta^t(w) = v'$ and $\delta^k(v') = v = v'z$ for some $t > 0$ and some $k > 0$. For each $j > 0$ there is a $z' \in W^*$ such that $\delta^{t+k+j}(w) = \delta^j(v) = \delta^j(v'z) = \delta^j(v')z' = \delta^{t+j}(w)z'$, and by (i), $z' \neq \lambda$.

Case 2. $\delta^t(w) = v = v'z$ and $\delta^k(v'z) = v'$ for some $t \geq 0$ and some $k > 0$. $\delta^k(v'z) = \delta^k(v')z' = v'$ for some $z' \in W^*$ and by (i), $z' \neq \lambda$. Therefore, $\ell g(\delta^k(v')) < \ell g(v')$. By iterating this argument $\ell g(v')+1$ times we obtain either $\ell g(\delta^{k(\ell g(v')+1)}(v')) < \ell g(v') - \ell g(v')$ which is impossible or $\delta^{k\ell g(v')}(v') = \delta^{k(\ell g(v')+1)}(v')$. In the latter case L(G) is finite; contradictory to the assumption. □

If we allow $\bar{\delta}(\lambda) \neq \lambda$ then Lemma 3.50 (ii) holds under the additional restriction: not both $\lambda \in L(G)$ and $\bar{\delta}(\lambda) \neq \lambda$.

COROLLARY 3.51. *Let* $G = \langle W,\delta,w\rangle$ *be a* D(1,0)L (D(0,1)L) *such that* $E(G,V_T)$ *is infinite for some* V_T *(and not both* $\lambda \in L(G)$ *and* $\bar{\delta}(\lambda) \neq \lambda$*). If* $v,v' \in E(G,V_T)$ *such that* v' *is a proper prefix of* v *(*v' *is a proper postfix of* v*) then, with finitely many exceptions, for each word* u *in* $E(G,V_T)$ *there is a word* u' *in* $E(G,V_T)$ *such that* $u = u'z$ $(u = zu')$ *for some* $z \in V_TV_T^*$.

Clearly, Lemma 3.50 and Corollary 3.51 hold for D(m,0)Ls with respect to prefixes and for D(0,m)Ls with respect to postfixes, $m \geq 0$.

THEOREM 3.52.

(i) *The intersections of* EPD1L *with* REG, CF - REG *and* CS - CF *are nonempty. There are languages in* REG, CF - REG *and* CS - CF *which are not in* EPD1L. EPD1L $\subset$ CS.

(ii) *The intersections of* ED1L *with* REG, CF - REG, CS - CF *and* RE - CS *are nonempty. There are languages in* REG, CF - REG, CS - CF *and* RE - CS *which are not in* ED1L. ED1L $\subset$ RE.

(iii) *The intersections of* EDOL *with* REG, CF - REG *and* CS - CF *are nonempty. There are languages in* REG, CF - REG *and* CS - CF *which are not in* EDOL. EDOL $\subset$ CS.

PROOF. Since $DXL \subseteq EDXL$, the first sentence in each statement (i) - (iii) is correct by Theorem 3.44. Let $L_1 = \{a,aa\} \cup \{b\}\{c\}^*\{b\}$, $L_2 = \{a,aa\} \cup \{a^n bc^n \mid n > 0\}$, $L_3 = \{a,aa\} \cup \{b^n c^n d^n \mid n > 0\}$ and $L_4 = \{a,aa\} \cup \{a\}A\{a\}$ where $A \subseteq \{1\}^*$ is the recursively enumerable but nonrecursive language of Theorem 3.44. By Corollary 3.51 L_1, L_2, L_3 and L_4 do not belong to ED1L. But $L_1 \in REG$; $L_2 \in CF - REG$ and $L_3 \in CS - CF$ as is well known; $L_4 \in RE - CS$. This proves the second sentence in each of the statements of (i) - (iii). The inclusion in the last sentence in the statements of (i) and (iii) follows by the usual working space theorem and strict inclusion by the foregoing. The inclusion in the last sentence of the statement of (ii) is true by the usual Turing machine simulation argument and strict inclusion follows by the foregoing.

□

Note that the existence of languages in REG, CF - REG and CS - CF which are not in ED0L could also have been proven using Corollary 3.49.

That with respect to families of extensions of L languages differences can only lie in no context, one-sided context and two-sided context, but not in the amount of context is shown by the next theorem.

THEOREM 3.53.

(i) $ED2L = EDIL$

(ii) $EPD2L = EPDIL$

(iii) $ED1L = \bigcup_{i \in \mathbb{N}} (ED(i,0)L \cup ED(0,i)L)$

(iv) $EPD1L = \bigcup_{i \in \mathbb{N}} (EPD(i,0)L \cup EPD(0,i)L)$

PROOF. We give the outline of a simulation technique to prove (i). (ii) - (iv) are completely analogous. ((i) also follows from Theorem 3.45 but the present proof is direct).

Let $G = \langle W,\delta,w\rangle$ be a D(m,n)L, $m,n > 0$, and let r be the greater one of m and n. We construct a D2L $G' = \langle W',\delta',w'\rangle$ as follows:

$$W' = W \cup \left(\bigcup_{i=0}^{m-1} W^i \times W \times \bigcup_{i=0}^{n-1} W^i \right) \quad \text{and } w' = w.$$

The production rules δ' are defined in such a way that, for each production of G, G' executes r productions. The first r - 1 of these r productions serve to gather the necessary context for each letter in the string and the r-th production produces the string produced by G.

E.g., if $\delta(a_1 a_2 \ldots a_k) = \alpha_1 \alpha_2 \ldots \alpha_k$, then (for $k \geq m,n$):

$$\delta'^{r}(a_1a_2\ldots a_k) = \delta'^{r-1}((\lambda,a_1,a_2)(a_1,a_2,a_3)\ldots(a_{k-1},a_k,\lambda))$$

$$= \delta'^{r-2}((\lambda,a_1,a_2a_3)(a_1,a_2,a_3a_4)\ldots(a_{k-2}a_{k-1},a_k,\lambda))$$

$$\vdots$$

$$= \delta'((\lambda,a_1,a_2a_3\ldots a_n)(a_1,a_2,a_3a_4\ldots a_{n+1})\ \ldots$$

$$\ldots(a_{k-m+1}a_{k-m+2}\ldots a_{k-1},a_k,\lambda))$$

$$= \alpha_1\alpha_2\ldots\alpha_k.$$

Therefore, $\delta'^{tr}(w') = \delta^{t}(w)$ for all t, and $\delta'^{t}(w') \notin W^*$ for all $t \not\equiv 0$ mod r. Hence, for each subset V_T of W, $L(G') \cap V_T^* = L(G) \cap V_T^*$.

Similarly, we can prove the analog of Theorem 3.53 for the general case of nondeterministic L systems.

Since the extensions of the deterministic L languages using one-sided or no context do not contain **all** regular languages it is a logical next step to see whether they do contain all finite languages. The next theorem tells us that one-sided context without erasing cannot give us all finite languages but one-sided context with erasing can. Let FIN denote the *family of finite languages*, where we shall make no distinction between the λ-free and non-λ-free finite languages since this would create trivial inclusion results from the sheer impossibility for propagating L systems to generate λ in their languages.

THEOREM 3.54.

(i) FIN $\not\subseteq$ EDOL

(ii) FIN $\not\subseteq$ EPD1L

(iii) FIN $\subset$ ED1L

PROOF.

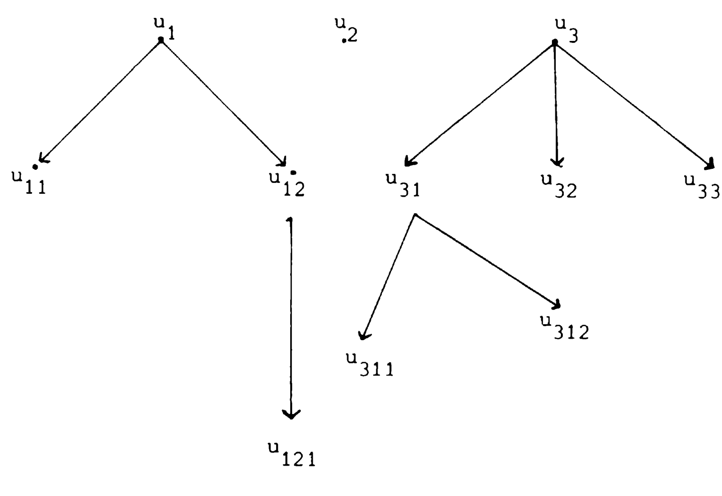
(i) Assume that $\{a,aa\} = L(G) \cap \{a\}^*$ for a DOL system $G = \langle W,\delta,w\rangle$. Since δ is a homomorphism, $\delta^i(a^k) = (\delta^i(a))^k$. Therefore, $\delta^i(aa) = \delta^i(a)\delta^i(a) \neq a$ for all i and S(G) = w,...,a,...,aa,... . Hence there is an i such that $\delta^i(a) = aa$. But then $\delta^{2i}(a) = aaaa$ also occurs in S(G) and consequently in $L(G) \cap \{a\}^*$: contradiction.

(ii) Assume that $\{a,aa,bbb\} = L(G) \cap \{a,b\}^*$ for a PD1L system $G = \langle W,\delta,w\rangle$. Since G is propagating S(G) = w,...,a,...,aa,...,bbb,... . Let G be left context sensitive. (The case that G is right context sensitive is identical since $\{a,aa,bbb\}$ is invariant under reversal.) Let j be the smallest integer such that $\delta^j(aa) = bbb$. Since $\delta^j(aa) = \delta^j(a)v$ for some $v \in W^*$ we have $\delta^j(a) =$ b,bb or bbb. Since a occurs in S(G), the case that $\delta^j(a) = b$ or bb implies that b or bb occur in S(G) and hence in $L(G) \cap \{a,b\}^*$: contradiction. The case that $\delta^j(a) = bbb$ implies that $\delta^j(a) = \delta^j(aa) = bbb$ which is impossible since G is propagating.

(iii) Since ED1L = ED(m,0)L for all $m \in \mathbb{N}$, $m > 0$, by Theorem 3.53, it suffices to show that each finite language L belongs to ED(m,0)L for some $m \in \mathbb{N}$. Let L be a finite language and choose $m = \max_{v \in L}\{\ell g(v)\}$. Order the words in $L - \{\lambda\}$ according to prefix inclusion as follows. For $u,v \in L$ we have $u \leq v$ if $v = uu'$ for some $u' \in W_T^*$ where W_T is the alphabet of L. Construct a finite labeled directed forest F reflecting the ordered set $(L,\leq)$ by defining $F = \langle P,E \rangle$ where each node p_v of P corresponds to an element v of L and (p_u,p_v) is an arc in L if $u < v$ and there is no z in L such that $u < z < v$. We now label a node p_v by v if p_v is a root of F (equivalently, there is no word z in $L - \{\lambda\}$ such that $z < v$); and by u' if p_u is a direct ancestor of p_v and $v = uu'$.

<u>EXAMPLE</u>.

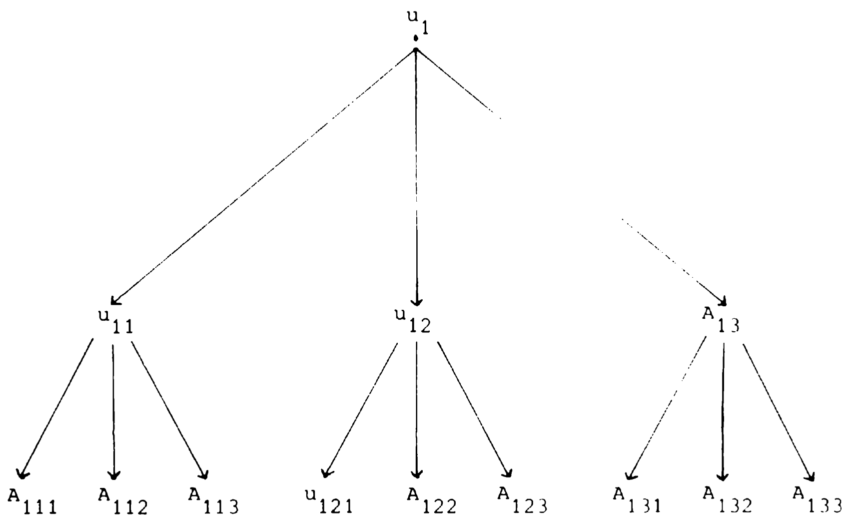
F =



$$L - \{\lambda\} = \{u_1,u_2,u_3,u_1u_{11},u_1u_{12},u_3u_{31},u_3u_{32},u_3u_{33},u_1u_{12}u_{121},$$
$$u_3u_{31}u_{311},u_3u_{31}u_{312}\}$$

Extend F to the forest F' of uniform branching degree d which is equal to the largest branching degree in F or to the number of roots in F if that is greater. The new nodes are labeled with new letters which constitute the alphabet W_N. All branches in F' are equal in length to the longest branch in F. Let ℓ be the number of nodes in such a branch.

<u>EXAMPLE CONTINUED</u>. E.g., the subtree attached to root u_1 in F is extended in F' to



We now define a D(m,0)L system $G = \langle W_T \cup W_N, \delta, w\rangle$ which generates all words which can be formed by concatenating from left to right the labels occurring in a path in F' starting from a root. Call the set of such words L'. Subscript the labels v of the forest F' as above, that is, $v_{i_1 i_2 \ldots i_r}$ at level r has descendants $v_{i_1 i_2 \ldots i_r i_{r+1}}$ at level $r+1$, $1 \le r < \ell$ and $i_1, i_2, \ldots, i_{r+1} \in \{1,2,\ldots,d\}$. Corresponding to each label $v_{i_1 i_2 \ldots i_r}$ there is a word $z_{i_1 i_2 \ldots i_r} = v_{i_1} v_{i_1 i_2} \ldots v_{i_1 i_2 \ldots i_r}$ in L', $1 \le r \le \ell$. Choose the initial string $w = z_{\underbrace{11\ldots1}_{\ell x}}$, that is, the word in L' corresponding to the left-most leaf of F'. For convenience sake we define for a word z in L' the functions *tail* and *head*. $\mathrm{tail}(z) = a$ and $\mathrm{head}(z) = z'$ for $z = z'a$, $z' \in (W_N \cup W_T)^*$ and $a \in W_N \cup W_T$.

(i) Suppose $\lambda \in L$. δ is defined inductively as follows.

$\underline{r=1}.\quad \delta(\text{head}(z_i),\text{tail}(z_i)) :=$

$\qquad \underline{\text{if}}\ i < d \quad \underline{\text{then}}\ v_{i+1}\ (= z_{i+1})$

$\qquad\qquad \underline{\text{else}}\ \lambda\ \underline{\text{fi}}$

$\underline{r=2}.\quad \delta(\text{head}(z_{i_1 i_2}),\text{tail}(z_{i_1 i_2})) :=$

$\qquad \underline{\text{if}}\ i_1 < d \quad \underline{\text{then}}\ v_{i_1+1\ i_2}$

$\qquad \underline{\text{else}}\ \underline{\text{if}}\ i_2 < d \quad \underline{\text{then}}\ v_1 v_{1\ i_2+1}$

$\qquad\qquad\qquad \underline{\text{else}}\ v_1$

$\qquad\quad \underline{\text{fi}}$

$\qquad \underline{\text{fi}}$

$\underline{r>2}.\quad \delta(\text{head}(z_{i_1 i_2 \ldots i_r}),\text{tail}(z_{i_1 i_2 \ldots i_r})) :=$

$\qquad \underline{\text{if}}\ \delta(\text{head}(z_{i_1 i_2 \ldots i_{r-1}}),\text{tail}(z_{i_1 i_2 \ldots i_{r-1}}))$

$\qquad\qquad \in \{v_{j_1 j_2 \ldots j_{r-1}}, v_{j_1 j_2 \ldots j_{r-2}}\ v_{j_1 j_2 \ldots j_{r-1}}\}$

$\qquad \underline{\text{then}}\ v_{j_1 j_2 \ldots j_{r-1} i_r}$

$\qquad \underline{\text{else}}\ (\delta(\text{head}(z_{i_1 i_2 \ldots i_{r-1}}),\text{tail}(z_{i_1 i_2 \ldots i_{r-1}})) = v_{\underbrace{11\ldots1}_{r-2\ x}})$

$\qquad\quad \underline{\text{if}}\ i_r < d \quad \underline{\text{then}}\ v_{\underbrace{11\ldots1}_{r-1\ x}}\ v_{\underbrace{11\ldots1}_{r-1\ x}\ i_r+1}$

$\qquad\quad \underline{\text{else}}\ v_{\underbrace{11\ldots1}_{r-1\ x}}$

$\qquad\quad \underline{\text{fi}}$

$\qquad \underline{\text{fi}}$

(ii) Suppose $\lambda \notin L$. Only the cases for $r = 1,2,3$ are different.

$\underline{r=1}.\quad \delta(\text{head}(z_i),\text{tail}(z_i)):=$

$\qquad \underline{\text{if}}\ i < d \quad \underline{\text{then}}\ v_{i+1}\ \underline{\text{else}}\ v_1\ \underline{\text{fi}}$

$\underline{r=2}.\quad \delta(\text{head}(z_{i_1 i_2}),\text{tail}(z_{i_1 i_2})) :=$

$\qquad \underline{\text{if}}\ i_1 < d \quad \underline{\text{then}}\ v_{i_1+1\ i_2}$

$\qquad \underline{\text{else}}\ \underline{\text{if}}\ i_2 < d \quad \underline{\text{then}}\ v_{1\ i_2+1}\ \underline{\text{else}}\ \lambda$

$\qquad\quad \underline{\text{fi}}$

$\qquad \underline{\text{fi}}$

$\underline{r=3}$. $\delta(\text{head}(z_{i_1 i_2 i_3}),\text{tail}(z_{i_1 i_2 i_3}))$:=

  $\underline{\text{if}}$ $\delta(\text{head}(z_{i_1 i_2}),\text{tail}(z_{i_1 i_2})) = v_{j_1 j_2}$ $\underline{\text{then}}$ $v_{j_1 j_2 i_3}$

  $\underline{\text{else}}$ $(\delta(\text{head}(z_{i_1 i_2}),\text{tail}(z_{i_1 i_2})) = \lambda)$

    $\underline{\text{if}}$ $i_3 < d$ $\underline{\text{then}}$ v_{11} $v_{11\ i_3+1}$

    $\underline{\text{else}}$ v_{11}

    $\underline{\text{fi}}$

  $\underline{\text{fi}}$

$\delta(\cdot) = \lambda$ for arguments not defined above. Since each letter in a string $z \in L'$ has the entire string of letters left of itself as its context, G starting with string w generates all words in L' in some order ending with λ iff $\lambda \in L$. Hence $L(G) = L'$ if $\lambda \notin L$ and $L(G) = L' \cup \{\lambda\}$ if $\lambda \in L$. It then follows from the definition of L' that $L(G) \cap W_T^* = L$.

<u>EXAMPLE CONTINUED AGAIN</u>. For the given example G is defined as follows (assuming $\lambda \in L$). $G = \langle W,\delta,w\rangle$ with $w = u_1 u_{11} A_{111}$, and δ is defined by the following list (where we assume that the rightmost letter of the argument is rewritten and the remainder of the argument is left context).

$\delta(u_1) = u_2$	$\delta(u_1 u_{11}) = A_{21}$
$\delta(u_2) = u_3$	$\delta(u_2 A_{21}) = u_{31}$
$\delta(u_3) = \lambda$	$\delta(u_3 u_{31}) = u_1 u_{12}$
	$\delta(u_1 u_{12}) = A_{22}$
	$\delta(u_2 A_{22}) = u_{32}$
	$\delta(u_3 u_{32}) = u_1 A_{13}$
	$\delta(u_1 A_{13}) = A_{23}$
	$\delta(u_2 A_{23}) = u_{33}$
	$\delta(u_3 u_{33}) = u_1$

$\delta(u_1 u_{11} A_{111}) = A_{211}$	$\delta(u_1 u_{11} A_{112}) = A_{212}$	$\delta(u_1 u_{11} A_{113}) = A_{213}$
$\delta(u_2 A_{21} A_{211}) = u_{311}$	$\delta(u_2 A_{21} A_{212}) = u_{312}$	$\delta(u_2 A_{21} A_{213}) = A_{313}$
$\delta(u_3 u_{31} u_{311}) = u_{121}$	$\delta(u_3 u_{31} u_{312}) = A_{122}$	$\delta(u_3 u_{31} A_{313}) = A_{123}$
$\delta(u_1 u_{12} u_{121}) = A_{221}$	$\delta(u_1 u_{12} A_{122}) = A_{222}$	$\delta(u_1 u_{12} A_{123}) = A_{223}$
$\delta(u_2 A_{22} A_{221}) = A_{321}$	$\delta(u_2 A_{22} A_{222}) = A_{322}$	$\delta(u_2 A_{22} A_{223}) = A_{323}$
$\delta(u_3 u_{32} A_{321}) = A_{131}$	$\delta(u_3 u_{32} A_{322}) = A_{132}$	$\delta(u_3 u_{32} A_{323}) = A_{133}$
$\delta(u_1 A_{13} A_{131}) = A_{231}$	$\delta(u_1 A_{13} A_{132}) = A_{232}$	$\delta(u_1 A_{13} A_{133}) = A_{233}$
$\delta(u_2 A_{23} A_{231}) = A_{331}$	$\delta(u_2 A_{23} A_{232}) = A_{332}$	$\delta(u_2 A_{23} A_{233}) = A_{333}$
$\delta(u_3 u_{33} A_{331}) = u_{11} A_{112}$	$\delta(u_3 u_{33} A_{332}) = u_{11} A_{113}$	$\delta(u_3 u_{33} A_{333}) = u_{11}$

Writing out S(G) reveals that $L(G) \cap W_T^\star = L$ where W_T is $W - W_N$ and W_N is the alphabet of capitals.

$u_1 u_{11} A_{111}$

$u_2 A_{21} A_{211}$

$\boxed{u_3 u_{31} u_{311}}$

λ $\boxed{u_1 u_{12} u_{121}}$

$u_2 A_{22} A_{221}$

$u_3 u_{32} A_{321}$

λ $u_1 A_{13} A_{131}$

$u_2 A_{23} A_{231}$

$u_3 u_{33} A_{331}$

λ $u_1 u_{11} A_{112}$

$u_2 A_{21} A_{212}$

$\boxed{u_3 u_{31} u_{312}}$

λ $u_1 u_{12} A_{122}$

$u_2 A_{22} A_{222}$

$u_3 u_{32} A_{322}$

λ $u_1 A_{13} A_{132}$

$u_2 A_{23} A_{232}$

$u_3 u_{33} A_{332}$

λ $u_1 u_{11} A_{113}$

$u_2 A_{21} A_{213}$

$u_3 u_{31} A_{313}$

λ $u_1 u_{12} A_{123}$

$u_2 A_{22} A_{223}$

$u_3 u_{32} A_{323}$

λ $u_1 A_{13} A_{133}$

$u_2 A_{23} A_{233}$

$u_3 u_{33} A_{333}$

λ $\boxed{u_1 u_{11}}$

$u_2 A_{21}$

$\boxed{u_3 u_{31}}$

λ $\boxed{u_1 u_{12}}$

$u_2 A_{22}$

$\boxed{u_3 u_{32}}$

λ $u_1 A_{13}$

$u_2 A_{23}$

$\boxed{u_3 u_{33}}$

λ $\boxed{u_1}$

$\boxed{u_2}$

$\boxed{u_3}$

$\boxed{\lambda}$

(N.B. the words in L are enclosed in boxes.)

□

3.2.5. EXTENSIONS AND HOMOMORPHIC CLOSURES OF PROPAGATING DETERMINISTIC L LANGUAGES.

In this section we study EPD2L and EPD1L and their closures under several types of homomorphisms. It is shown that the closure of EPD1L under nonerasing homomorphisms is strictly included in EPD2L. The proof exploits an interesting property of deterministic L systems with one-sided context. In contrast to this, it will appear that already the simplest type of erasing homomorphism which maps a letter either to itself or to λ increases PD1L to the recursively enumerable languages.

First we define the concept of a linear bounded automaton. A *linear bounded automaton* (LBA) M is a Turing machine with, say, symbol set S, state set Ψ and start state $q_0 \in \Psi$, such that M accepts a word v over a subset V_T of S using at most $c\,\ell g(v)$ tapesquares during its computation, where c is a fixed constant for M. It is well known that the family of languages accepted by linear bounded automata is equal to CS (see, e.g., HOPCROFT and ULLMAN [1969] or SALOMAA [1973a]). A *deterministic* LBA or DLBA is an LBA such that each instantaneous description has exactly one successor. We shall show that EPD2L equals the family of languages accepted by DLBAs, that is, the *deterministic context sensitive languages* denoted by DLBA. Thus the question of whether or not the inclusion of EPD2L in EP2L is strict is shown to be equivalent to the classic problem (see HOPCROFT and ULLMAN [1969] or SALOMAA [1973a]) in formal language theory of whether or not the inclusion of DLBA in CS is strict. (Recall that van DALEN [1971] showed that EP2L = CS.)

Investigating the role of one-sided and two-sided context for EPDIL systems we note immediately that EPD1L $\subset$ EPD2L since it is easy to construct a PD2L G such that $L(G) = \{a,aa\} \cup \{b\}\{c\}^*\{b\}$ which language is not in EPD1L by Corollary 3.51. (We can also use Theorem 3.54(ii) to prove this fact. Surely, {a,aa,bbb} is an EPD2L language!) Later it will be shown that already the simplest type of erasing homomorphism, which maps a letter to itself of to λ, extends PD1L to RE. However, as we shall see, not even the most powerful nonerasing homomorphisms can extend EPD1L to EPD2L = DLBA $\subseteq$ CS.

<u>THEOREM 3.55</u>. EPD2L = DLBA

<u>PROOF</u>. We give an outline since the details would be tedious. Let $G = \langle W,\delta,$ $w\rangle$ be a PD2L and V_T a subset of W. Construct a deterministic linear bounded

automaton M as follows. M uses an amount of tape equal to 4*(length of input + 1), divided in 4 sections I, II, III, IV of equal length. The input word v is written on I; section II contains the initial string w, section III is blank and section IV contains the representation of 0 in the #W-ary number system. M compares $\delta^i(w)$ with $v, i \geq 0$, and accepts v if $\delta^i(w) = v$. Otherwise, scuttling back and forth between sections II and III, M produces $\delta^{i+1}(w)$ from $\delta^i(w)$ such that $\delta^{i+1}(w)$ is written on III if $\delta^i(w)$ is written on II and vice versa. (If $\ell g(\delta^{i+1}(w)) \geq \ell g(v) + 1$ then M rejects v.) Subsequently, M increments the number written on IV by 1. If IV contains a number equal to $\#W^{\ell g(v)+1}-1$ then M rejects v. Otherwise, M compares $\delta^{i+1}(w)$ with v, and so on. Since $v \in L(G)$ iff $v = \delta^i(w)$ for some $i < \#W^{\ell g(v)+1}$ we see that $L(M) = L(G)$, where $L(M)$ is the language accepted by M. Now construct M' from M where M' is exactly like M except that M' first ascertains that $v \in V_T^*$ and rejects v if $v \notin V_T^*$. Then $L(M') = L(G) \cap V_T^*$.

Let M be a DLBA, which accepts L(M) over S, using no more than cn tapesquares for an input word of length n. Now construct a DLBA M' such that M' generates all words $v_0, v_1, \ldots$ over S in lexicographical order and accepts or rejects them by simulating M. In particular we can do it such that M', started in state q_0' on a word v_i, i 0, written from left to right from the origin with the remaining $(c-1)\ell g(v_i)$ tapesquares containing blank symbols, computes the next word v_{i+1} written from left to right from the origin with the remaining tapesquares containing blank symbols. Subsequently, M' proceeds to the origin, enters the start state q_0 of M and simulates M. After rejection or acceptance M' erases everything but v_{i+1} from the tape and starts in q_0' at the origin, i.e. scanning the leftmost letter of v_{i+1}, and so on.

Let V be the set of symbols of M', b the blank symbol, and Ψ the state set of M'. Construct $G = \langle W, \delta, w \rangle$ as follows:

$$W = V \cup (V^c \times (\Psi \cup \{\lambda\}) \times \{0,1,2,\ldots,c\}),$$

$$w = (a,b,b,\ldots,b,q_0,1),$$

where a is the first word of SS^* in the lexicographical order. G simulates M' as follows: if $\delta^t(w) = \underline{a}_1\underline{a}_2\ldots\underline{a}_n$,

$$\underline{a}_1\underline{a}_2\ldots\underline{a}_n \in (V^c \times \{\lambda\} \times \{0\})^* (V^c \times \Psi \times \{1,2,\ldots,c\}) (V^c \times \{\lambda\} \times \{0\})^*,$$

then the j-th element of $\underline{a}_i$, $1 \le j \le c$ and $1 \le i \le n$, corresponds with the $(i+(j-1)n)$-th tapesquare of M', the $(c+1)$-th element of $\underline{a}_i$ indicates the present state of M' if one of the tapesquares coded in $\underline{a}_i$ is under scan (and is λ otherwise) and the $(c+2)$-th element tells which tapesquare (and is 0 otherwise). In particular we can construct G such that if M' enters an accepting state the accepted word v_i over S is "read out" from right to left, and subsequently is restored (from left to right) to the form

$$(a_1,b,b,\ldots,b,q_0',1)(a_2,b,b,\ldots,b,\lambda,0)\ldots(a_n,b,b,\ldots,b,\lambda,0)$$

for $v_i = a_1a_2\ldots a_n$. Hence $L(G) \cap S^* = L(M)$. $\square$

We now proceed to show that the closure of EPD1L under nonerasing homomorphisms does not contain REG.

LEMMA 3.56. *Let* $G = \langle W,\delta,w\rangle$ *be a* PD(1,0)L *such that* L(G) *is infinite. Let* $r = \#W$. *For each* $t \ge r$ *there is a prefix* v *of* $\delta^t(w)$, $\ell g(v) \ge \lfloor \log_r((r-1)t+r)\rfloor$, *and a constant* k, $0 < k \le r^{\ell g(v)}$, *such that* v *is a prefix of* $\delta^{t+nk}(w)$ *for all* n. *For* PD(0,1)Ls *this holds with respect to postfixes.*

PROOF. Denote the i-th letter of a string $\delta^j(w)$, $i,j \in \mathbb{N}$, by a_{ij}. Since L(G) is infinite, the slowest rate of growth G can achieve is by generating all words over W in lexicographical order, i.e. $\ell g(\delta^t(w)) \ge \lfloor \log_r((r-1)t+r)\rfloor$. Therefore, a_{ij} is indeed a letter in W for all j such that $j \ge \Sigma_{\ell=1}^{i-1} r^{\ell}$. Since there are only r different letters in W, there are natural numbers j_1 and k_1, $j_1,k_1 \le r$ and $k_1 > 0$, such that $a_{1j_1} = a_{1j_1+k_1}$. Since G is a PD(1,0)L, $a_{1j_1+nk_1} = a_{1j_1}$ for all n. Therefore, a letter in the second position has a_{1j_1} as its left neighbor at all times j_1+nk_1, $n \in \mathbb{N}$. There is surely a letter in the second position for all times $t \ge r$. Therefore, there are positive natural numbers j_2 and k_2, $j_2 \ge r$, $k_2 \le r^2$ and $j_2+k_2 \le r+r^2$, such that $j_2 = j_1+n_1k_1$, $j_2+k_2 = j_1+n_2k_1$ for some $n_1,n_2 \in \mathbb{N}$ and $a_{2j_2} = a_{2j_2+k_2}$. By iteration of this argument, for each $s = 1,2,\ldots$ there are positive natural numbers j_s and k_s, $j_s \ge \Sigma_{i=1}^{s-1} r^i$, $k_s \le r^s$ and $j_s+k_s \le \Sigma_{i=1}^{s} r^i$, such that

$$a_{1j_1}a_{2j_2}\ldots a_{sj_s} = a_{1j_s+nk_s}\,a_{2j_s+nk_s}\ldots a_{sj_s+nk_s},$$

for all n. Since G is a PD(1,0)L,

$$a_{1j_s+t}a_{2j_s+t}\cdots a_{sj_s+t} = a_{1j_s+t+nk_s}a_{2j_s+t+nk_s}\cdots a_{sj_s+t+nk_s}$$

for all t and n. Therefore, for all s and all t such that

$$\sum_{i=1}^{s} r^i > t \geq j_s \geq \sum_{i=1}^{s-1} r^i,$$

there is a prefix v of $\delta^t(w)$, $\ell g(v) \geq \lfloor \log_r((r-1)t+r) \rfloor = s$, and a positive constant $k_s \leq r^s$ such that v is a prefix of $\delta^{t+nk_s}(w)$ for all n. Hence the lemma. □

Contrasting Lemma 3.56 and Lemma 3.50 gives a nice insight in the influence of the propagating restriction with respect to the necessary behavior of the pre- and postfixes of the sequences of words generated by D1L systems.

THEOREM 3.57. *Let* V *be any alphabet containing at least two letters. No language containing* VV^* *belongs to the closure under nonerasing homomorphisms of* EPD1L.

PROOF. Assume that $\{a,b\} \subseteq V$, and consider the subset $L = \{(a^n b^n)^{f(n)} \mid n \geq 1\}$ of V^*. Suppose that $L \subseteq h(L(G) \cap V_T^*)$ for some PD(1,0)L $G = \langle W,\delta,w \rangle$, an alphabet V_T and a nonerasing homomorphism h from V_T^* into V^*. Define t_n by

$$t_n = \min\{i \in \mathbb{N} \mid \delta^i(w) \in V_T^* \text{ and } h(\delta^i(w)) = (a^n b^n)^{f(n)}\}.$$

As is easily seen, $\ell g(\delta^t(w)) \leq m^t \ell g(w)$ where m is the maximum length of a value of δ. Therefore, $2n\, f(n) \leq m^{t_n} \ell g(w)c$ where $c = \max\{\ell g(h(a)) \mid a \in V_T\}$. Or, $t_n \geq \log_m(f(n)(2n/(\ell g(w)c))) > \log_m f(n)$ for all $n \geq n_0$ where n_0 is some fixed natural number. For each $n \geq n_0$, $\delta^{t_n}(w)$ has a prefix v_n such that, for $f(n) > m^{(r^{n+1})}$,

$$\ell g(v_n) \geq \lfloor \log_r(t_n(r-1)+r) \rfloor > n, \quad r = \#W,$$

and v_n occurs infinitely often with a constant period k_n by Lemma 3.56. Since for each n, prefix v_n of $\delta^{t_n}(w)$ is mapped under h to $a^n bz$, $z \in \{a,b\}^*$, v_n cannot be a prefix of $\delta^{t_{n'}}(w)$ for $n \neq n'$ and $n,n' \geq n_0$. We now derive a contradiction by showing that then $k_n = k_{n_0}$ for all $n \geq n_0$. Since G is propagating and the prefix v_n $(n \geq n_0)$ occurs with a constant period k_n there is a j_n such that $\delta^{j_n}(v_{n_0}) = v_n z$ for some $z \in W^*$. But then

$$\delta^{t_{n_0}+pk_{n_0}+j_n}(w) = \delta^{j_n}(v_{n_0} z_p) = \delta^{j_n}(v_{n_0})z'_p = v_n zz'_p$$

for all p and some $z, z_p, z'_p \in W^*$. I.e. from time $t_{n_0} + j_n$ the prefix v_n occurs with period k_{n_0} and $k_n = k_{n_0}$ (or k_n divides k_{n_0}) for all $n \geq n_0$. Hence

$$\#(h(L(G) \cap V_T^*) \cap \{(a^n b^n)^{f(n)} \mid n \geq n_0\}) \leq k_{n_0}$$

and

$$VV^* \not\subseteq h(L(G) \cap V_T^*).$$

(Since $VV^* = (VV^*)^R$, i.e. the language consisting of all words from VV^* reversed, the above proof holds also for PD(0,1)Ls.) □

From the above proof we see that any language which contains a language like $\{(a^n b^n)^{f(n)} \mid n \geq 1\}$ cannot be the image under nonerasing homomorphism of a language in EPD1L. Hence also e.g. $(\{a\}^+\{b\}^+)^+$. The idea behind the proof is roughly the following. If a language L contains a large enough subset L' such that each pair of words in L', say u and v, are distinguishable by their respective prefixes (postfixes) u' and v' for which hold that $\ell g(u') = \mathcal{O}(\log\log u)$ and $\ell g(v') = \mathcal{O}(\log\log(v))$ then L cannot be in the closure under nonerasing homomorphisms of EPD(1,0)L and EPD(0,1)L, respectively. For example, $\{b\}^+\{a\}^*\{b\}^+$ contains $\{b^n(a^n)^{f(n)}b^n \mid n \geq 0\}$ for each $f\colon \mathbb{N} \to \mathbb{N}$ and therefore is not contained in a nonerasing homomorphic image of a language in EPD1L.

Denote the closures of a language family X under nonerasing homomorphisms by $h_{\lambda\text{-free}}$ X and the closure of X under letter-to-letter homomorphisms by $h_{1:1}$ X. Gathering the results up to now about EPD1L and EPD2L we have:

THEOREM 3.58.

(i) EPD1L $\subset h_{1:1}$ EPD1L $\subseteq h_{\lambda\text{-free}}$ EPD1L $\subset$ EPD2L = DLBA.

(ii) *For each* $x \in \{\lambda, h_{1:1}, h_{\lambda\text{-free}}\}$ *the language family* xEPD1L *has nonempty intersections with* REG, CF − REG, CS − CF; *there are languages in*

REG, CF - REG *and* CS - CF *which are not in* xEPD1L; $h_{\lambda\text{-free}}$ EPD1L $\subset$ DLBA.

PROOF.

(i) Let

$$G = \langle\{a_1,a_2,a_3,b,c\},\{\delta(\lambda,a_1,\lambda) = a_2a_3,\delta(\lambda,a_2,\lambda)$$

$$= \delta(a_2,a_3,\lambda) = \delta(\lambda,b,\lambda) = b,\delta(b,b,\lambda) = \delta(c,b,\lambda) = cb,$$

$$\delta(b,c,\lambda) = \delta(c,c,\lambda) = c\},a_1\rangle$$

be a PD(1,0)L. Let h be a letter-to-letter homomorphism defined by $h(a_i) = a$ for $i = 1,2,3$, and $h(b) = b, h(c) = c. h(L(G)) = \{a,aa\} \cup \{b\}\{c\}^*\{b\}$ and by Corollary 3.51 $h(L(G)) \notin$ EPD1L. Therefore, EPD1L $\subset h_{1:1}$ EPD1L. $h_{1:1}$ EPD1L $\subseteq$ $h_{\lambda\text{-free}}$ EPD1L by definition. It is easy to show that DLBA = $h_{\lambda\text{-free}}$ DLBA; together with Theorem 3.55 this gives EPD2L = $h_{\lambda\text{-free}}$ EPD2L = DLBA. Since EPD1L $\subseteq$ EPD2L we have $h_{\lambda\text{-free}}$ EPD1L $\subseteq$ EPD2L. CF $\subset$ DLBA (see Exercise 8.3 in HOPCROFT and ULLMAN [1969] for CF $\subseteq$ DLBA and $\{a^nb^nc^n \mid n \geq 1\} \in$ DLBA - CF) and therefore $\{a,b\}^+ \in$ EPD2L and by Theorem 3.57 we have that $\{a,b\}^+ \notin$ $h_{\lambda\text{-free}}$ EPD1L. Hence $h_{\lambda\text{-free}}$ EPD1L $\subset$ EPD2L.

(ii) Since PD1L $\subseteq$ xEPD1L, the first sentence follows from Theorem 3.44. The second sentence follows by taking languages from REG, CF - REG and CS - CF, forming their union with $\{a,b\}^+$, where a,b are new letters, and applying Theorem 3.57. The last sentence follows from (i). □

We have seen above that the generating power of deterministic propagating L systems with one-sided context, together with the nonterminal mechanisms and nonerasing homomorphisms stays within the range of DLBA languages and does not encompass the regular languages. We conclude the main results in this subsection by proving that the closure of the family of PD1L languages under homomorphisms that map a letter either to itself or to the empty string equals the family of recursively enumerable languages.

The proof method was suggested by a proof of EHRENFEUCHT and ROZENBERG [1974b] for the equality of RE and the closure of D2L under weak codings. (A weak coding is a homomorphism that maps a letter either to a letter or to the empty string.) The weak coding allows us to get rid of the intermediate work done by the L system in computing the subsequent words of the desired r.e. language L. The difficulty lies in the fact that we have to "read out" the complete word in L from a word over an alphabet disjoint from the alphabet of L in one production, since otherwise also subwords of the desired words appear under the homomorphism. The solution makes essential use of the parallelism in L systems by a firing squad synchronization. The *Firing Squad Synchronization* problem, see e.g. MINSKY [1967], can be stated as follows. Suppose we want to synchronize an arbitrary long finite chain of interacting identical finite state automata. All finite state automata are initially in the same state m and stay in that state if both neighbors are in state m. The automata on the ends of the chain are allowed to be different since they sense that they lack one neighbor. Synchronization is achieved if and when all automata enter the firing state f at the same time and no automaton in the chain is in state f before that time. In the terminology of L systems a firing squad is a PD2L system $F = \langle W_F, \delta_F, m^k \rangle$ such that $\delta_F(m,m,m) = \delta_F(m,m,\lambda) = m$. F satisfies the following requirement: there is a function $t\colon \mathbb{N} \to \mathbb{N}$ such that for each $k \in \mathbb{N}$ it holds that for all i, $1 \le i < t(k)$,

$$\delta_F^{t(k)}(m^k) = f^k \quad \text{and} \quad \delta_F^{i}(m^k) \notin W_F^*\{f\}W_F^* .$$

BALZER [1967] proved that there is such an F with $\#W_F = 8$ and $t(k) = 2k-2$. After these preliminaries we state the theorem.

THEOREM 3.59. *The closure of* PD1L *under homomorphisms, which map a letter either to itself or to* λ, *is equal to* RE.

PROOF. Since by now these kinds of proofs are familiar we give only an outline. Let A be an infinite recursively enumerable language enumerated by a 1:1 recursive function $f\colon \mathbb{N} \xrightarrow{1:1} A$; n is recovered from f(n) by f^{-1}. (The case where A is finite follows by a similar method.) Let T be a Turing machine which starts with the representation of 0 on its tape, say $a_1a_2\ldots$ $\ldots a_{n_0}$, computes f(0), replaces everything except f(0) on its tape by the blank symbol b and returns to the leftmost symbol of f(0). Subsequently T retrieves 0 from f(0) by f^{-1}, increments 0 with 1, and computes f(1), and so on. In particular we can do this in such a way that after the computation of f(n) the instantaneous description of T is $b^{\ell}q'f(n)b^r$ for some ℓ, $r \in \mathbb{N}$ and a distinguished state q' of T. The next instantaneous description of T is $b^{\ell}q''f(n)b^r$ for another distinguished state q" of T. Scanning the leftmost symbol of f(n), T starts retrieving n from f(n) by f^{-1} in state q". We simulate T by a PD2L $G = \langle W,\delta,w \rangle$; hence the blank symbols will not disappear. G is defined as follows:

$$W = (\Psi \times S \cup S) \times W_F \cup S,$$

where Ψ is the state set of T, S is the symbol set of T and b is the blank symbol, and W_F is the alphabet of the firing squad F.

$$w = (q_0, a_1, m)(a_2, m)\ldots(a_{n_0}, m),$$

where q_0 is the start state of T, $a_1 a_2 \ldots a_{n_0}$ is the representation of 0 and m is the initial state of the firing squad F. G simulates T until the situation

$$\delta^{t_0}(w) = b^{\ell}(q', c_1, m)(c_2, m)\ldots(c_{\ell_0} m)b^r$$

occurs where $c_1 c_2 \ldots c_{\ell_0}$ is f(0). Subsequently, the substring between the b's executes a firing squad and, when the squad fires, maps itself to f(0). I.e.

$$\delta^{t_0+2\ell_0-2}(w) = b^{\ell}(q', c_1, f)(c_2, f)\ldots(c_{\ell_0}, f)b^r,$$

$$\delta^{t_0+2\ell_0-1}(w) = b^{\ell} c_1 c_2 \ldots c_{\ell_0} b^r = b^{\ell} f(0) b^r.$$

δ is constructed such that a letter $c \in S - \{b\}$ is rewritten as (c,m), except when it has b of λ as left neighbor in which case it is rewritten as (q",c,m). Therefore,

$$\delta^{t_0+2\ell_0}(w) = b^{\ell}(q'', c_1, m)(c_2, m)\ldots(c_{\ell_0}, m)b^r$$

and G continues simulating T, retrieves 0, adds 1 and computes the representation of f(1), and so on. Hence $h(L(G)) = A$ where h is a homomorphism defined by $h(a) = a$ if $a \in S - \{b\}$ and $h(a) = \lambda$ otherwise.

We now simulate G by a PD1L $G' = \langle W', \delta', w' \rangle$ which is defined exactly as the D(0,1)L in Lemma 3.39 except that $\delta'(\lambda, (a, \lambda), \lambda) = b$ for all $a \in W$. Then $h'(L(G')) = A$ where h' is a homomorphism defined by $h'(a) = a$ if $a \in S - \{b\}$ and $h'(a) = \lambda$ otherwise. □

The last result we prove in this section tells that any class of deterministic context sensitive L systems, but for the nonerasing one-sided context types, is capable of generating a large subclass of the DLBA languages if we use as additional mechanism a nonerasing homomorphic mapping. Let the *exponential* DLBA *languages* ($DLBA_{exp}$) be the subclass of DLBA languages for which the following property holds: for each infinite $L \in DLBA_{exp}$ there is a constant c such that for each word $v \in L$ there is a word $v' \in L$ such that $\ell g(v) < \ell g(v') \leq c\ell g(v)$. Furthermore, $FIN \subset DLBA_{exp}$. It is easy to see that $CF \subset DLBA_{exp} \subset DLBA$.

THEOREM 3.60.

(i) $DLBA_{exp} = h_{1:1}\ PD2L = h_{\lambda\text{-free}}\ PDIL$

(ii) $DLBA_{exp} \subset h_{1:1}\ D1L$

(iii) $h_{\lambda\text{-free}}\ PDI_L L \subset DLBA_{exp}$

PROOF.

(i) The proof is similar to that of Theorem 3.55, where we proved that EPD2L = DLBA, but with the additional use of a firing squad synchronization to read out complete words in one production like in the proof of Theorem 3.59. Let $L = \{w_1, w_2, \ldots, w_i, \ldots\}$ be a $DLBA_{exp}$ language of which the words are ordered according to increasing lengths and lexicographically within a set of the same length. Let $G = \langle W, \delta, w_1 \rangle$ be a PD2L system. Suppose we have generated, starting from w_1, the word $w_i = a_1 a_2 \ldots a_n$. For the subsequent computation up to w_{i+1} we use strings of length n always taking care that each letter in position j in such a string is marked with the letter a_j. Like in the proof of Theorem 3.55 we generate all possible candidate strings of length $\leq c * \ell g(w_i)$ and check for inclusion in L by simulating a DLBA M for which $L(M) = L$. Since c is a fixed constant for L and all said candidate strings are of length $\leq c * \ell g(w_i)$ they can be coded in a word of $\ell g(w_i)$. The computing activities involved are completely analogous to those in the proof of Theorem 3.55 and therefore they can be performed on a string of length $\ell g(w_i)$. After having found the candidate word w_{i+1}, which is the next word in the ordered set L, we compute the substring the jth letter of the string of length $\ell g(w_i)$ must derive (to derive w_{i+1}) for each letter of the string. Subsequently we simulate a firing squad which then, in one production, generates from the string of length $\ell g(w_i)$ the new word w_{i+1}. The letter-to-letter homomorphism h is defined such that it maps each letter marked with a letter a from the alphabet of L to a. Since we have marked

all letters in position j, $1 \leq j \leq n$, of the intermediate strings in the computation of w_{i+1} from w_i with the j-th letter a_j of w_i, all these strings derive w_i under h. If L happened to be finite, the L system goes into a loop after checking all possible word candidates of length $\leq c * \ell g(w_\ell)$ where w_ℓ is the last word of the ordered set L. Hence $L = h(L(G))$ and therefore $DLBA_{exp} \subseteq h_{1:1}$ PD2L. From Theorem 3.55 we have that all PD2L languages are DLBA languages, and since $h_{1:1}$ DLBA = DLBA it follows that $h_{1:1}$ PD2L $\subseteq$ DLBA. Since an L system can increase the length of a string by at most a constant multiple in one production it follows that $h_{1:1}$ PD2L $\subseteq DLBA_{exp}$. Together with the previous inclusion this proves that $h_{1:1}$ PD2L $= DLBA_{exp}$. Since $DLBA_{exp}$ is clearly invariant under nonerasing homomorphisms it follows that also $h_{\lambda\text{-free}}$ PD2L $= DLBA_{exp}$. Similarly it follows that $h_{\lambda\text{-free}}$ PDIL $= DLBA_{exp}$.
(ii) Using the method of Lemma 3.39 to simulate a D2L system by a D1L system it is easy to show that we can simulate a PD2L system G by a D1L system G' such that $h(L(G')) = L(G)$ for a letter-to-letter homomorphism h. (Hint; since no letter is rewritten as λ by G we can code the end marker in the last letter of a string in G'.) Hence PD2L $\subseteq h_{1:1}$ D1L and therefore $h_{1:1}$ PD2L $\subseteq h_{1:1}$ D1L. Strict inclusion follows since $h_{1:1}$ PD2L $\subset$ DLBA and D1L contains nonrecursive languages according to Theorem 3.42. Hence, since $h_{1:1}$ PD2L $= DLBA_{exp}$ according to (i) we have that $DLBA_{exp} \subset h_{1:1}$ D1L.
(iii) Follows from (i) and the observation that $h_{\lambda\text{-free}}$ $PDI_L L$ does not contain REG by Theorem 3.57. □

REMARK. Similarly to the above we can prove that all nonerasing homomorphisms of families of pure L languages are contained in RE_{exp}, where RE_{exp} is the recursively enumerable languages analogon of $DLBA_{exp}$. (Cf. Chapter 6).

3.2.6. COMBINING THE RESULTS OF SECTION 3.2.

We combine and interrelate the above results on the generating power of various types of deterministic context sensitive L systems, with and without additional mechanisms, in Table 3.1. Horizontally we list the type of L system, that is, the combinations between one- and two-sided context and erasing or nonerasing production rules. Vertically we list the additional mechanisms: none (pure), letter-to-letter homomorphism ($h_{1:1}$), nonerasing homomorphism ($h_{\lambda\text{-free}}$), letter-to-itself-or-letter-to-λ homomorphism (h_w), extension (E) and the combinations between the above homomorphisms and extension: $h_{1:1}E$, $h_{\lambda\text{-free}}E$ and h_wE. In the box corres- poding to the XL systems and the additional mechanism x we classify the generative power by listing above the smallest family Y_s which contains the language family xXL and listing below the largest family Y_ℓ which is contained in xXL. We choose these least upper bounds Y_s and greatest lower bounds Y_ℓ from the better understood and neatly nested families of the Chomsky hierarchy: RE, DLBA, $DLBA_{exp}$, CF, REG, FIN, $\emptyset$ which strictly include each other in this order. ($Y_\ell = \emptyset$ means that not even FIN is included in xXL.) If there are families Z $\in$ {DLBA,$DLBA_{exp}$,CF,REG,FIN} in between Y_ℓ and Y_s then this means that xXL is incomparable with Z, i.e., xXL has a nonempty intersection with Z but neither contains the other. (Clearly, then xXL contains also languages from the differences between two consecutive language families of the listed hierarchy in between Y_s and Y_ℓ.)

		PD1L	D1L	PD2L	D2L
(i)	pure	$DLBA_{exp}$ - - - $\emptyset$	RE - - - $\emptyset$	$DLBA_{exp}$ - - - $\emptyset$	RE - - - $\emptyset$
(ii)	$h_{1:1}$	$DLBA_{exp}$ - - - FIN	RE - - - $DLBA_{exp}$	$DLBA_{exp}$ - - - $DLBA_{exp}$	RE - - - $DLBA_{exp}$
(iii)	$h_{\lambda\text{-free}}$	$DLBA_{exp}$ - - - FIN	RE - - - $DLBA_{exp}$	$DLBA_{exp}$ - - - $DLBA_{exp}$	RE - - - $DLBA_{exp}$
(iv)	h_w	RE - - - RE	RE - - - RE	RE - - - RE	RE - - - RE
(v)	E	DLBA - - - $\emptyset$	RE - - - FIN	DLBA - - - DLBA	RE - - - RE
(vi)	$h_{1:1}E$	DLBA - - - FIN	RE - - - RE	DLBA - - - DLBA	RE - - - RE
(vii)	$h_{\lambda\text{-free}}E$	DLBA - - - FIN	RE - - - RE	DLBA - - - DLBA	RE - - - RE
(viii)	h_wE	RE - - - RE	RE - - - RE	RE - - - RE	RE - - - RE

Table 3.1 Comparing L language families with families from the Chomsky hierarchy.

Table 3.1 gives us a good overview of the generating power of context sensitive parallel rewriting (in the L system sense), as compared to the better understood generating power of sequential rewriting (in the Chomsky-type grammar sense). It also tells us how powerful intersection with a terminal alphabet, various types of homomorphisms, or combinations of these are with respect to one- or two-sided context and erasing or nonerasing production rules for deterministic L systems. Most noticeable is the extreme power of erasing homomorphisms (h_w) which gives us RE in all cases and the extreme resistance of PD1L against all other operations. The tendency for all families is that the more and more powerful operations are used, the better they nest in the Chomsky hierarchy. Extension (intersection with a terminal alphabet) is more powerful for two-sided context than nonerasing homomorphism, but nonerasing homomorphism is more powerful than extension for one-sided context. Extension together with nonerasing homomorphism makes everything a neat Chomsky family but for PD1L.

We discuss how the table is derived row by row.

(i) The *upper bounds* on the language families follow from Theorem 3.44 (where we can replace CS by $DLBA_{exp}$ in view of Theorem 3.55 and Theorem 3.60). The *lower bounds* follow from the fact that not even D2L systems can generate all finite languages as their pure L languages. For instance, if we choose a finite language over a singleton alphabet we have to use a D2L system with a singleton alphabet and hence all letters in a string are rewritten the same but for the two end letters. Therefore, e.g., $\{a^4,a^5,a^7\}$ cannot be a D2L language. Analogously we prove the same thing for pure D(m,n)L languages for given $m,n \geq 0$, but clearly FIN $\subset$ DIL in general. Similarly, FIN $\subset$ PDIL but FIN $\not\subseteq$ PDI_LL, PDI_RL (by Theorem 3.53(iv) and Theorem 3.54(ii)). None of these pure L language families contains REG by Lemma 3.43.

(ii) The *upper bounds* follow from (i) since the language families serving as a least upper bound there are invariant under $h_{1:1}$. The *lower bounds* are argued as follows. FIN $\subset h_{1:1}$ PD1L since for each finite (λ-free) language L we can choose an arbitrary large alphabet W and generate words (which are mapped to the words in L by a letter-to-letter homomorphism) in increasing length even by a PD0L system. Hence we have FIN $\subset h_{1:1}$ PD0L from which FIN $\subset h_{1:1}$ PD1L follows. Furthermore, FIN is the greatest lower bound for $h_{1:1}$ PD1L since REG $\not\subseteq h_{1:1}$ PD1L by Theorem 3.57.

The remainder of the lower bounds follows from the fact that $DLBA_{exp} \subseteq h_{1:1}$ D1L, $h_{1:1}$ PD2L, $h_{1:1}$ D2L by Theorem 3.60, and by the word length argument.
(iii) Is proved similarly to (ii).
(iv) Follows from Theorem 3.59 which states that h_w PD1L = RE.
(v) The *upper bounds* for ED1L and EPD1L follow from Theorem 3.52(i),(ii) by replacing CS by DLBA for the EPD1L case, since EPD2L = DLBA by Theorem 3.55 which then also gives the upper bound for EPD1L. ED2L = RE by Theorem 3.45. It remains to settle the greatest *lower bound* for ED1L and EPD1L. Now FIN $\subset$ ED1L by Theorem 3.54(iii) and REG $\not\subseteq$ ED1L by Theorem 3.52(ii). FIN $\not\subseteq$ EPD1L by Theorem 3.54(ii).
(vi) The *upper bounds* follow from those in (v) by noting that the language families concerned are invariant under $h_{1:1}$, and hence also the *lower bounds* for EPD2L and ED2L are the same as in (v). The lower bounds for $h_{1:1}$ EPD1L and $h_{1:1}$ ED1L are derived as follows. FIN $\subseteq h_{1:1}$ EPD1L follows from (ii) since there it was shown that FIN $\subseteq h_{1:1}$ PD1L. REG $\not\subseteq h_{1:1}$ EPD1L follows from Theorem 3.57. Hence FIN is the greatest lower bound on $h_{1:1}$ EPD1L. The greatest lower bound RE for $h_{1:1}$ ED1L follows from Theorem 3.46 where it was proven that $h_{1:1}$ ED1L = RE.
(vii) Similarly to (vi).
(viii) Follows from (iv).

In row (i) - (iii) the least upper bounds RE can be replaced by RE_{exp}.

In Figure 3.1, we summarize the inclusion relations between the most important L language families, their extensions and homomorphic closures, and the languages in the Chomsky hierarchy as treated in Section 3.2. Connection by a solid arrow means that the upper language family strictly includes the lower one; connection by a dotted arrow means that the upper language family contains the lower one and it is not known yet whether the inclusion is strict; if two language families are not connected at all this means that their intersection is nonempty but neither contains the other: they are *incomparable*.

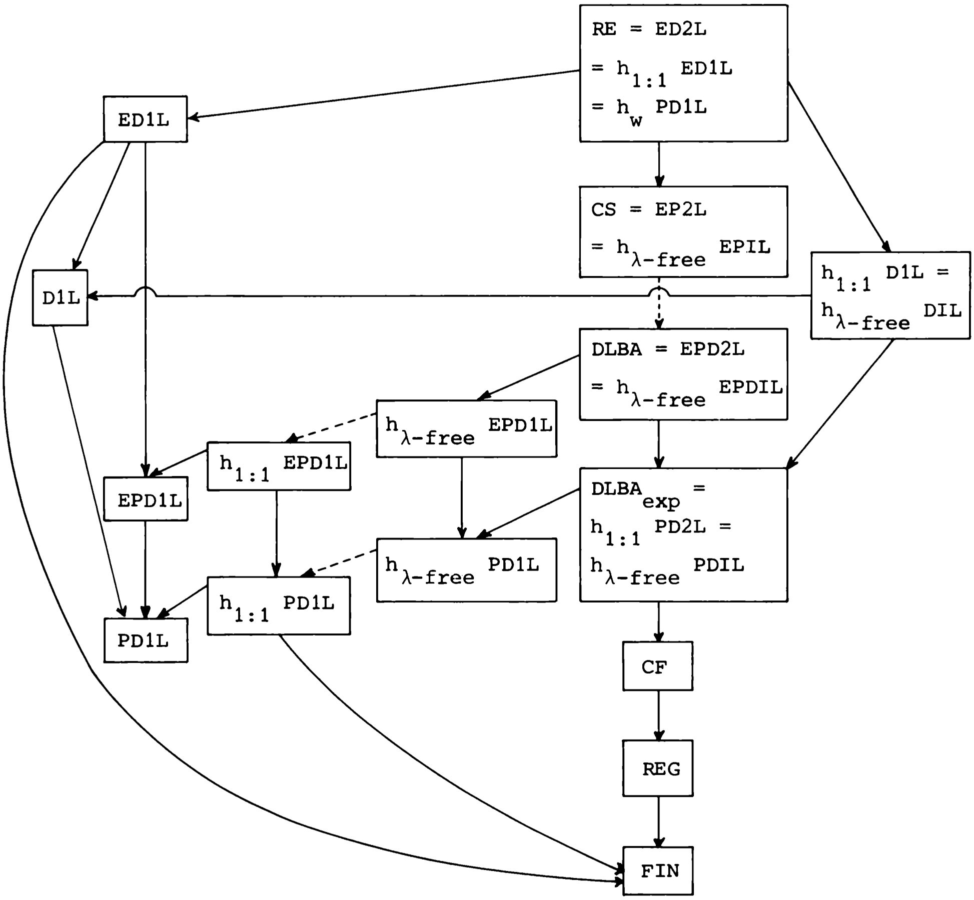


Figure 3.1. Classification of families of deterministic context sensitive L languages, extensions and their homomorphic closures.

In the diagram of Figure 3.1 and in Table 3.1 we have mainly considered language families obtained from (propagating or nonpropagating) D(m,n)L systems with $m+n = 1$ or $m = n = 1$. The reason for this is, that with regard to extensions the amount of context does not matter: the only differences lie in no context, one-sided or two-sided context as was shown in Theorem 3.53. The same thing holds for the closure of pure L language families under homomorphic mappings like $h_{1:1}$, $h_{\lambda\text{-free}}$ or h_w:

$$h_{1:1}\ \mathrm{D2L} = h_{1:1}\ \mathrm{DIL}$$

$$h_{1:1}\ \mathrm{PD2L} = h_{1:1}\ \mathrm{PDIL}$$

$$h_{1:1}\ \mathrm{D1L} = \bigcup_{i=0}^{\infty} (h_{1:1}\ \mathrm{D(i,0)L} \cup h_{1:1}\ \mathrm{D(0,i)L})$$

etc., as is easily proved in a way similar to Theorem 3.53. Hence the only thing not covered by Figure 3.1 is the hierarchy of pure deterministic context sensitive L language families according to amount of context and propagating restriction in between DIL and PD1L. This hierarchy was covered by ROZENBERG [1972a,b] and ROZENBERG and LEE [1975] for the nondeterministic case but holds analogously also for the deterministic case. It ties in with Figure 3.1 in the obvious way. With respect to the language families of the Chomsky hierarchy, we have seen in the discussion of Table 3.1 row (i) that FIN $\subset$ PDIL $\subset$ DIL but that FIN $\not\subseteq$ $\mathrm{PDI_LL}$,$\mathrm{PDI_RL}$. The question of whether or not FIN $\subset$ $\mathrm{DI_LL}$,$\mathrm{DI_RL}$ is still open. The inclusion relations depicted in Figure 3.1 follow largely by the results in Table 3.1, by the various inclusions by definition, or by other results in Section 3.2. We leave the verification as exercises for the reader but for a few cases.

(i) ED1L is incomparable with $h_{1:1}$ EPD1L and $h_{\lambda\text{-free}}$ EPD1L.

Let $L = \{a,aa\} \cup \{b\}\{c\}^*\{b\}$. $L \in h_{1:1}$ EPD1L $\subseteq h_{\lambda\text{-free}}$ EPD1L by the proof of Theorem 3.58(i), and $L \notin$ ED1L by the proof of Theorem 3.52. Therefore

(a) $\qquad h_{1:1}$ EPD1L, $h_{\lambda\text{-free}}$ EPD1L $\not\subseteq$ ED1L.

Since ED1L contains languages in RE − CS by Theorem 3.52(ii) and $h_{1:1}$ EPD1L $\subseteq$ $h_{\lambda\text{-free}}$ EPD1L $\subset$ CS by Theorem 3.58(i) we have:

(b) $\qquad$ ED1L $\not\subseteq h_{1:1}$ EPD1L,$h_{\lambda\text{-free}}$ EPD1L.

Furthermore, by definition PD1L $\subseteq$ ED1L, $h_{1:1}$ EPD1L, $h_{\lambda\text{-free}}$ EPD1L which together with (a) and (b) proves (i).

(ii) D1L incomparable with $h_{1:1}$ EPD1L and $h_{\lambda\text{-free}}$ EPD1L is proven as (i).

(iii) ED1L incomparable with $h_{1:1}$ PD1L and $h_{\lambda\text{-free}}$ PD1L is proven as (i) by noting that $L \in h_{1:1}$ PD1L.

(iv) D1L incomparable with $h_{1:1}$ PD1L and $h_{\lambda\text{-free}}$ PD1L is proven as (iii).

3.3. CONTEXT SENSITIVE TABLE LINDENMAYER SYSTEMS AND A TRADE-OFF EQUIVALENT TO THE LBA PROBLEM

While in the previous Section 3.2 we were almost exclusively concerned with languages derived from deterministic L systems, we will now consider nondeterministic and table Lindenmayer languages as defined in Section 2.2. Table Lindenmayer systems were introduced by ROZENBERG [1973a] and consist of L systems, with several sets (tables) of rewriting rules, where at each moment all letters in a string are rewritten simultaneously according to the production rules chosen from a *single* table. Whereas in the sequential rewriting of generative grammars this would not constitute any difference, because of the parallel nature of L systems the use of tables can result in an increase of generating power. The use of tables can be taken to correspond with the impact of external conditions on the developmental growth of an organism, e.g., with changes of light and dark or temperature, each environmental condition corresponding with the use of a particular set of production rules. (See also Chapter 5.) Context free table L systems have been studied extensively, see, e.g., HERMAN and ROZENBERG [1975]. It has been shown that, for instance,

$$\text{CF} \subset \text{E0L} \subset \text{ET0L} \subset \text{INDEX} \subset \text{DLBA}$$

where INDEX stands for the family of *indexed* context free languages. Furthermore, ET0L has extraordinary closure properties: it is a full AFL, SALOMAA [1974], and it is about the smallest language family for which it is known that the membership question is NP-complete, van LEEUWEN [1975c]. I.e., the question of whether a word belongs to an ET0L language L(G) for some ET0L system G can be solved by a nondeterministic Turing machine in polynomial time, and each problem solvable by a nondeterministic Turing machine in polynomial time is deterministic polynomial time reducible to the membership problem for ET0L languages. Further time and storage complexity results for context free (table) L languages can be found in the references concerning complexity theory cited in the Introduction.

In this section we will treat all families of languages generated by context sensitive L systems with tables using nonterminals according to the effects of restrictions like: λ-freeness of production rules, determinism of production rules, number of tables, one- or two-sided context, and closures of these families under various types of homomorphisms. Because of the great generating power of already deterministic context sensitive L systems using the terminal-nonterminal mechanism, the partial ordering according to set inclusion of the considered language families basically collapses to the recursively enumerable languages, context sensitive languages and deterministic context sensitive (DLBA) languages. Hence the classification yields an interesting equivalence of the classic LBA problem (is the family of DLBA languages equal to the family of context sensitive languages?) in terms of L systems. In the previous Section 3.2 it was proven that the family of DLBA languages coincides with the family of languages generated by λ-free deterministic context sensitive L systems (with one table) using nonterminals. Van DALEN [1971] showed that the family of context sensitive languages equals the family of languages generated by λ-free context sensitive L systems (with one table) using nonterminals. Hence the LBA problem can be stated in terms of determinism versus nondeterminism in L systems. By arguments similar to those used in Theorem 3.55 WOOD [1976] proved that the family of languages generated by λ-free deterministic context sensitive L systems with two tables using nonterminals is equal to the family of context sensitive languages. Here the LBA problem was stated in the form of whether or not two tables can be reduced to one in the case under consideration. We will demonstrate that the family of context sensitive languages equals the family of languages generated by λ-free deterministic left context-sensitive L systems with two tables using nonterminals, thereby molding the LBA problem in the form of whether or not a trade-off is possible between one-sided context with two tables and two-sided context with one table for λ-free deterministic L systems using nonterminals. From the results it will appear that any further restriction on one of the two participants in the trade-off reduces the generating power to below the DLBA languages. If we relax the restriction of λ-freeness we obtain in both cases the recursively enumerable languages: then the trade-off is possible. We should stress, however, that although it seems that the trade-off corresponding to the LBA problem is between two deterministic rewriting systems, nondeterminism creeps in whenever we use more than one table for L systems since the choice of the next table to be used is nondeterministic. For a survey of the LBA problem and its reduction to other problems see HARTMANIS and HUNT [1974].

THEOREM 3.61. *The families of languages generated by the various subclasses of* ETIL *systems and their closures under several types of homomorphisms are classified by the diagram of Figure* 3.2. *Solid arrows imply proper set inclusion of the lower family in the upper one. Broken arrows imply inclusion where strictness is not known. If two of the displayed families are not connected by (a sequence of) arrows this means that these families are incomparable, i.e., their intersection contains nontrivial languages and neither family contains the other.* $X \equiv Y \bmod \lambda$ *means* $L \in X$ *iff* $L-\{\lambda\} \in Y$.

Note that all families of context sensitive table L languages obtained with the use of nonterminals are classified by the displayed diagram since the results are stated in their strongest form and cannot be improved (except for the broken arrow which corresponds to the LBA problem). But for EPD1L and ED1L all families are closed under nonerasing homomorphisms.

The proof of the theorem proceeds by a number of lemmas, but first we introduce a concept needed in the proof of Lemma 3.63. To express restrictions on the choice of tables to be used in a derivation we need the notion of a control word. Let $G = \langle W, \{P_1, P_2, \ldots, P_q\}, w\rangle$ be a table L system. A *control word* u for a derivation in G is an element of $\{1,2,\ldots,q\}^*$, and

$$v \overset{u}{\underset{G}{\Rightarrow}} v'$$

$v, v' \in W^*$, $u = i_1 i_2 \ldots i_k$ with $i_1, i_2, \ldots, i_k \in \{1,2,\ldots,q\}$, means that

$$v = v_0 \underset{G_{i_1}}{\Longrightarrow} v_1 \underset{G_{i_2}}{\Longrightarrow} v_2 \underset{G_{i_3}}{\Longrightarrow} \cdots \underset{G_{i_k}}{\Longrightarrow} v_k = v'$$

For some $v_1, v_2, \ldots, v_{k-1}$ in W^* and $G_{i_j} = \langle W, P_{i_j}, w\rangle$, $1 \le j \le k$.

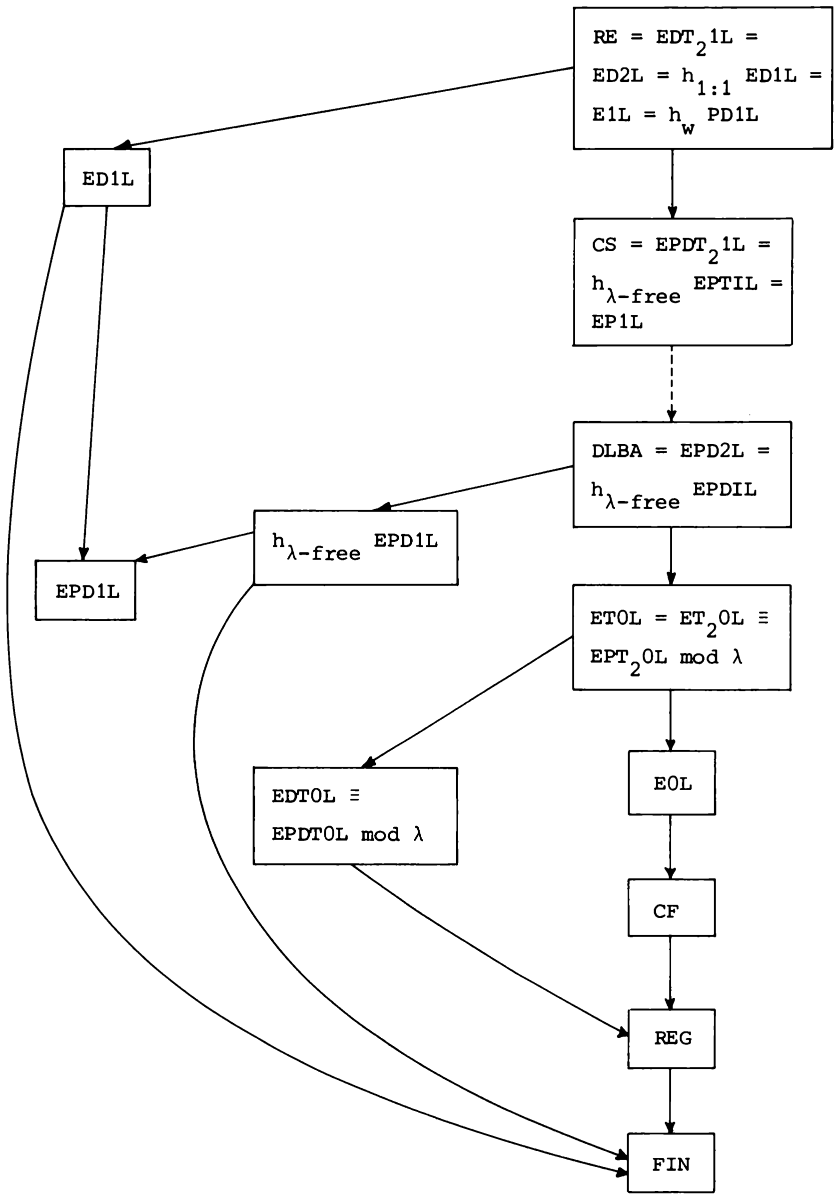


Figure 3.2. Classification of families of context sensitive table L extension languages and their homomorphic closures.

LEMMA 3.62. $EDT_2 1L = RE$.

PROOF. By Figure 3.1 $h_{1:1}$ ED1L = RE. Let $G = \langle W,P,w,V_T\rangle$ be an ED1L system and $h: V_T^* \to V^*$ a letter-to-letter homomorphism. Assume without loss of generality that $W \cap V = \emptyset$. Construct the $EDT_2 1L$ system $G' = \langle W',\{P_1,P_2\},w,V\rangle$ as follows (G and G' are left context sensitive).

$$W' = W \cup V \cup \{F\} \quad \text{with} \quad F \notin W \cup V;$$

$$P_1 = P \cup \{(x,a) \to F \mid (x,a) \notin (W \cup \{\lambda\}) \times W\}$$

$$P_2 = \{(x,a) \to h(a) \mid (x,a) \in (V_T \cup \{\lambda\}) \times V_T\}$$

$$\cup \{(x,a) \to F \mid (x,a) \notin (V_T \cup \{\lambda\}) \times V_T\}.$$

The reader can satisfy himself easily that $E(G') = h(E(G))$. □

From Lemma 3.62 and Figure 3.1 it follows that RE = ED2L = $EDT_2 1L$ = $h_{1:1}$ ED1L = h_w PD1L. E1L = RE follows from Lemma 3.62 by constructing an E1L system from the $EDT_2 1L$ system by lumping the two tables together to one table and preventing the simultaneous use in a given string of production rules from both tables by having the letter in the resultant string which senses that its left neighbor resulted from an application of a production rule from the other table derive the F symbol. In van DALEN [1971] it is proved that EP2L = CS. By the working space theorem, SALOMAA [1973a], or by the usual LBA simulation argument, it follows that EPTIL = CS. CS is closed under $h_{\lambda\text{-free}}$. WOOD [1976] proved that $EPDT_2 2L$ = CS. We now come to the main result of this section.

LEMMA 3.63. $EPDT_2 1L = CS$.

PROOF. According to PENTTONEN [1975], left context sensitive grammars (or more restrictedly, generative grammars with production rules of the form $AB \to A\beta$ or $B \to \beta$ where A and B are nonterminals and β is a nonempty string over the terminals and the nonterminals) suffice to generate all context sensitive languages.

CLAIM. EP1L = CS

Proof of Claim. Since EP2L = CS we only have to prove CS $\subseteq$ EP1L. Let

$G = \langle V_N, V_T, P, S\rangle$ be a grammar with nonterminals V_N, terminals V_T, the production rules in P of the form $AB \to A\beta$ or $B \to \beta$ where $A,B \in V_N$ and $\beta \in (V_N \cup V_T)^+$, and starting symbol $S \in V_N$. Construct an EP1L system $G' = \langle W', P', w', V_T\rangle$ as follows (G' is left context sensitive):

> $W' = V_N \cup \bar{V}_N \cup V_T \cup \{F\}$, $\bar{V}_N = \{\bar{A} \mid A \in V_N\}$ and V_N, $\bar{V}_N$, V_T and $\{F\}$ are pairwise disjoint. $w' = S$ and P' is defined by:
>
> $$\left.\begin{array}{ll}(1) & (x,A) \to A \\ (2) & \to \bar{A}\end{array}\right\} \text{ for all } A \in V_N \text{ and all } x \in W' \cup \{\lambda\},$$
>
> (3) $(A,\bar{B}) \to \beta$ if $AB \to A\beta \in P$ and $A \in V_N$, $\bar{B} \in \bar{V}_N$,
>
> (4) $(x,\bar{B}) \to \beta$ if $B \to \beta \in P$ and $x \in (W' \cup \{\lambda\}) - \bar{V}_N$,
>
> (5) $(\bar{A},\bar{B}) \to F$ for all $\bar{A},\bar{B} \in \bar{V}_N$,
>
> (6) $(x,F) \to F$ for all $x \in W' \cup \{\lambda\}$,
>
> (7) $(x,a) \to a$ for all $a \in V_T$ and $x \in W' \cup \{\lambda\}$.

(i) Clearly, if $S \overset{*}{\underset{G}{\Rightarrow}} v$ and $v \in V_T^*$ then there is a twice as long derivation $S \overset{*}{\underset{G'}{\Rightarrow}} v$. Therefore $L(G) \subseteq E(G')$.

(ii) Suppose $S \overset{*}{\underset{G'}{\Rightarrow}} v$ and $v \in V_T^*$. Because of (6) at no step of the derivation (5) was used: no adjacent barred nonterminals occurred in a word of the derivation.
Therefore, if $S = v_0 \underset{G'}{\Rightarrow} v_1 \underset{G'}{\Rightarrow} v_2 \underset{G'}{\Rightarrow} \cdots \underset{G'}{\Rightarrow} v_k = v$ then for each derivation step $v_i \underset{G'}{\Rightarrow} v_{i+1}$, $0 \le i < k$, there are $u_{i_1}, u_{i_2}, \ldots, u_{i_\ell} \in (V_N \cup V_T)^*$ such that either $\ell = 1$ or $u_{i_1} \underset{G}{\Rightarrow} u_{i_2} \underset{G}{\Rightarrow} \cdots \underset{G}{\Rightarrow} u_{i_\ell}$ where u_{i_1} and u_{i_ℓ} are equal to v_i and v_{i+1} with all bars removed from the nonterminals, respectively. Hence $S \overset{*}{\underset{G}{\Rightarrow}} v$ and $E(G') \subseteq L(G)$.

By (i) and (ii) $E(G') = L(G)$, and in view of the cited result by PENTTONEN [1975] this proves the claim. *End of proof of Claim.*

Above we noted that EPTIL $\subseteq$ CS and by the claim it therefore suffices to prove EP1L $\subseteq$ $EPDT_2$1L to prove the lemma. Let $G = \langle W,P,w,V_T\rangle$ be an EP1L system with $W = \{a_1,a_2,\ldots,a_n\}$ and P defined by:

$$\begin{array}{llll} (\lambda,a_j) & \to \alpha_{0j0} & (a_i,a_j) & \to \alpha_{ij0} \\ & \to \alpha_{0j1} & & \to \alpha_{ij1} \\ & \vdots & & \vdots \\ & \to \alpha_{0jn_{0j}} & & \to \alpha_{ijn_{ij}} \end{array}$$

for $1 \le i,\ j \le n$.

Construct an $EPDT_2 1L$ system $G' = \langle W', \{P_1,P_2\}, w, V_T \rangle$ defined by:

$$W' = W \cup W_\lambda \times W \cup (\bar{W}_\lambda \times \bar{W} \cup \bar{\bar{W}}_\lambda \times \bar{\bar{W}} \cup \bar{\bar{\bar{W}}}_\lambda \times \bar{\bar{\bar{W}}}) \times \{0,1,\ldots,k\} \cup \{F\}$$

where $k = \max\{n_{ij} \mid 0 \le i \le n \text{ and } 0 < j \le n\}$; $X_\lambda = X \cup \{\lambda\}$ and $\bar{X} = \{\bar{a} \mid a \in X\}$ for $X \in \{W, W_\lambda, \bar{W}, \bar{W}_\lambda, \bar{\bar{W}}, \bar{\bar{W}}_\lambda\}$; F is a new letter.

$$\begin{aligned} P_1:\ & (y,(a_{i_1},a_{i_2})) \to (a_{i_1},a_{i_2}) \\ & (y,(\bar{a}_{i_1},\bar{a}_{i_2},i)) \to (\bar{\bar{a}}_{i_1},\bar{\bar{a}}_{i_2},i) \\ & (y,(\bar{\bar{a}}_{i_1},\bar{\bar{a}}_{i_2},i)) \to (\bar{\bar{\bar{a}}}_{i_1},\bar{\bar{\bar{a}}}_{i_2},i) \\ & (y,(\bar{\bar{\bar{a}}}_{i_1},\bar{\bar{\bar{a}}}_{i_2},i)) \to (\bar{\bar{a}}_{i_1}\bar{\bar{a}}_{i_2},i) \end{aligned}$$

For all $y \in W' \cup \{\lambda\}$, $a_{i_1} \in W_\lambda$, $a_{i_2} \in W$ and i such that $0 \le i \le k$.
$(.,.) \to F$ if $(.,.)$ is not in the above list.

$$\begin{aligned} P_2:\ & (a_{i_1},a_{i_2}) \to (a_{i_1},a_{i_2}) \\ & (\lambda,(a_{i_1},a_{i_2})) \to (\bar{a}_{i_1},\bar{a}_{i_2},0) \\ & ((\bar{\bar{\bar{a}}}_{i_1},\bar{\bar{\bar{a}}}_{i_2},i),(a_{i_3},a_{i_4})) \to (\bar{a}_{i_3},\bar{a}_{i_4},0) \\ & (x,(a_{i_1},a_{i_2})) \to (a_{i_1},a_{i_2}) \\ & (y,(\bar{a}_{i_1},\bar{a}_{i_2},i)) \to (\bar{a}_{i_1},\bar{a}_{i_2},\ \text{remainder}\ ((i+1)/(n_{i_1 i_2}+1))) \\ & (y,(\bar{\bar{\bar{a}}}_{i_1},\bar{\bar{\bar{a}}}_{i_2},i)) \to (\bar{\bar{\bar{a}}}_{i_1},\bar{\bar{\bar{a}}}_{i_2},i) \end{aligned}$$

$$(z,(\bar{\bar{a}}_{i_1},\bar{\bar{a}}_{i_2},i)) \to \alpha_{i_1 i_2 i}$$

for all $a_{i_1} \in W_\lambda$, $a_{i_2}, a_{i_3}, a_{i_4} \in W$, i such that $0 \le i \le k$,

$x \in W_\lambda \times W \cup (\bar{W}_\lambda \times \bar{W} \cup \bar{\bar{W}}_\lambda \times \bar{\bar{W}}) \times \{0,1,\ldots,k\}$,

$y \in \bar{\bar{\bar{W}}}_\lambda \times \bar{\bar{\bar{W}}} \times \{0,1,\ldots,k\} \cup \{\lambda\}$, $z \in \bar{\bar{W}}_\lambda \times \bar{\bar{W}} \times \{0,1,\ldots,k\} \cup \{\lambda\}$.

$(.,.) \to F$ if $(.,.)$ is not in the above list.

Suppose

$$a_{i_1} a_{i_2} \cdots a_{i_n} \underset{G}{\Rightarrow} \alpha_{0 i_1 j_1} \alpha_{i_1 i_2 j_2} \cdots \alpha_{i_{n-1} i_n j_n}.$$

Then

$$a_{i_1} a_{i_2} \cdots a_{i_n} \overset{u}{\underset{G'}{\Rightarrow}} \alpha_{0 i_1 j_1} \alpha_{i_1 i_2 j_2} \cdots \alpha_{i_{n-1} i_n j_n}$$

under the control word

$$u = 2\ 2^{j_1+1}\ 11\ 2^{j_2+1}\ 11 \ldots 11\ 2^{j_n+1}\ 12.$$

Hence

$$E(G) \subseteq E(G').$$

Now suppose that $v \overset{u}{\underset{G'}{\Rightarrow}} z$ and $v,z \in W^*$ and no intermediate word in the derivation belongs to W^*. According to the productions the last table applied must have been P_2 and the word v' it was applied to belongs to $(\bar{\bar{W}}_\lambda \times \bar{\bar{W}} \times \{0,1,2,\ldots,k\})^*$ since otherwise F would occur in z. But the only way to derive such a v' by application of tables P_1 and P_2 under the given assumptions yields a v' such that if $v' \overset{2}{\underset{G'}{\Rightarrow}} z$ then $v \underset{G}{\Rightarrow} z$ as careful scrutiny of the production rules shows. [In fact if $u = u'2$ then under the assumptions

$$u' \in 2\ 1^*(2^+(11)^+)^{lg(v)-1}\ 2^+1(12^*1)^* \ ]$$

Hence $E(G') \subseteq E(G)$, which together with the previous implication shows that $E(G') = E(G)$. $\square$

The inclusion relations between RE, CS, DLBA, CF, REG, F1N, ED1L, EPD1L and $h_{\lambda\text{-free}}$ EPD1L are stated already in Figure 3.1. The connected parts of the diagram of Figure 3.2 from ET0L downwards follow by various combinations of Lemma 3.2 and Theorem 6.4 - 6.7 from NIELSEN, ROZENBERG, SALOMAA and SKYUM [1974a] and EHRENFEUCHT, ROZENBERG and SKYUM [1976]. ET0L has deterministic tape complexity $\mathcal{O}(n)$ and therefore ET0L $\subseteq$ DLBA; since moreover ET0L is a full AFL and DLBA is not it follows that the inclusion is strict, van LEEUWEN [1976]. The only thing remaining to be shown is:

LEMMA 3.64. X *and* Y *are incomparable for all* X *and* Y *such that* X $\in$ {ED1L, EPD1L, $h_{\lambda\text{-free}}$ EPD1L} *and* Y $\in$ {ET0L, EDT0L}.

PROOF. REG $\not\subseteq$ X by Figure 3.1, but, according to the established part of the diagram of Figure 3.2, REG $\subset$ Y. By definition EPD0L $\subseteq$ X $\cap$ Y (EPD0L is not displayed in Figure 3.2). Since the homomorphic closure of X is equal to RE (by the fact that h_w PD1L = RE) and the homomorphic closure of Y is contained in ET0L (by definition and the fact that ET0L is a full AFL) there are languages in X which are not in Y. Hence X and Y have a nonempty intersection and neither contains the other. □

3.4. STABLE STRING LANGUAGES OF L SYSTEMS

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 [1975, 1976], 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.

As we saw before, the usual way in formal language theory for obtaining languages from rewriting systems (be they sequential, e.g., grammars, or parallel like L systems) is by intersection with a terminal alphabet. That is, by selecting from all strings that are produced those over a terminal alphabet. The method proposed by Herman and Walker, viz. the stable string operation, consists of selecting from all strings produced by the 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 in this section the relation between the above two approaches for the various families of L systems. In HERMAN and WALKER [1975] it is proved that the generating power of context free 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 free L systems by intersection with a terminal alphabet, see e.g. HERMAN and ROZENBERG [1975], 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 class of context sensitive L systems generally coincides with the set of languages obtained from this class by intersection with a terminal alphabet. Moreover, analogous results hold for classes of L systems using more than one set of production rules, i.e., the table L systems, both context free and context sensitive. By making use of the previous results concerning extensions of L languages in Sections 3.2 and 3.3 we are able to derive many results concerning stable string languages of L systems, some of which are also established in WALKER [1974a,b, c] by different methods.

The *stable string language* of an L system $G = \langle W,P,w\rangle$ is defined by

$$A(G) = \{v \in W^* \mid v \in L(G) \text{ and } \check{v} \Rightarrow z \text{ implies } z = v\}.$$

The family of stable string languages of XL systems is denoted by AXL. We immediately note the following. $A(G) \subseteq L(G)$; although $A(G)$ may be empty this is not the case for $L(G)$; if the G is deterministic (and with but one table) then $\#A(G)$ is either 0 or 1.

<u>EXAMPLE</u>. Let G be the 0L system $\langle\{a,b\},\{a \to a,\ a \to aa,\ a \to b,\ b \to b\},a\rangle$. Then $L(G) = \{a,b\}^+$ and $A(G) = \{b\}^+$.

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

3.4.1. STABLE STRING LANGUAGES OF L SYSTEMS WITHOUT TABLES

LEMMA 3.65. *Let* $G = \langle W,P,w\rangle$ *be any type of* (m,n)L *system such that* $m+n > 0$ *and let* V_T *be a subset of* W. *We can effectively derive from* G *and* V_T *an* (m,n)L *system* $G' = \langle W',P',w'\rangle$ *of the same type as* G *(but for determinism and the cardinality of the alphabet), a subset* V_T' *of* W' *and an isomorphism* h *from* V_T^* *onto* $V_T'^*$ *such that* $h(L(G) \cap V_T^*) = A(G')$.

PROOF. We 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 L system $G' = \langle W',P',w'\rangle$ which is constructed as follows.

$$W' = W \cup V_T' \cup \{F,s\}$$

where W, V_T' and $\{F,s\}$ are disjoint, $\#V_T' = \#V_T$ and h is an isomorphism from V_T^* onto $V_T'^*$; $w' = s$ and the set of production rules P' is defined by

(1)	$(\lambda ,s, \lambda) \to w$		.
(2)		$\to h(w)$	if $w \in V_T^*$.
(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 $v_1av_2 \notin V_T'^+$.
(6)		$\to a$	for all $v_1av_2 \in V_T'^+$.

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

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

(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 $v \in V_T'^*$.

<u>Case 1</u>. $z = s$. Then $z \Rightarrow h^{-1}(v) = w$ by (2) and (1).

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

<u>Case 3</u>. $z \neq s$ and $z = v$. By (6) and (5) we can reduce this to cases 1 and 2.

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 which can have been applied 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 $v \Rightarrow v_1FFv_2$ for some words v_1,v_2 in W'^* and therefore $v \notin A(G')$. Hence

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

which together with the previous inclusion shows that

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

<u>LEMMA 3.66</u>. *Let* $G = \langle W,P,w\rangle$ *be a (deterministic)* $P(m,n)L$ *system. Given* G, *we can effectively produce a (deterministic)* $P(m,n)L$ *system* $G' = \langle W',P',w'\rangle$, *a subset* V_T *of* W' *and an isomorphism* h *from* V_T^* *onto* W^* *such that*

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

<u>PROOF</u>. Construct $G' = \langle W',P',w'\rangle$ as follows: $W' = W \times \{0,1\}$; and the inittial string $w' = (a_1,0)(a_2,0)\ldots(a_k,0)$ for $w = a_1a_2\ldots a_k$. Let g be a letter-to-letter homomorphism from W'^* onto W^* defined by $g((a,i)) = a$ for $i \in \{0,1\}$; and define P', for $i = 0,1$, by:

(1) $\quad (v_1,(a,i),v_2) \rightarrow (a_1,0)(a_2,0)\ldots(a_\ell,0) \quad$ if

$(g(v_1),a,g(v_2)) \rightarrow a_1a_2\ldots a_\ell \in P$ and

there is a rule $(g(v_1),a,g(v_2)) \rightarrow \alpha$

in P such that $\alpha \neq a$.

(2) $\quad \rightarrow (a,1)$ otherwise.

Let $V_T = \{(a,1) \mid a \in W\}$ and define $h: V_T^* \rightarrow W^*$ by $h((a,1)) = a$.

Suppose $v \in A(G)$; i.e., if $w \overset{*}{\underset{G}{\Rightarrow}} v \underset{G}{\Rightarrow} z$ then $z = v$. Since G is propagating every letter in v must necessarily produce itself and for $v = a_1a_2\ldots$ $\ldots a_\ell$ we therefore have

$$w' \overset{*}{\underset{G'}{\Rightarrow}} (a_1,i_1)(a_2,i_2)\ldots(a_\ell,i_\ell) \underset{G'}{\Rightarrow} (a_1,1)(a_2,1)\ldots(a_\ell,1).$$

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 that

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

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

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

and the lemma follows. $\square$

THEOREM 3.67.

(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* 3.65 *(e.g. the propagating property). Then* $EX(m,n)L \subseteq AX(m,n)L$.

(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* 3.66 *(e.g., determinism, lengths of the right hand sides of the production rules). Then* $AXP(m,n)L \subseteq EXP(m,n)L$.

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 3.65 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) Follows similarly to (i) from Lemma 3.66. □

<u>COROLLARY 3.68</u>. AP(m,n)L = EP(m,n)L *for* m+n > 0.

Since E1L = RE by the diagram of Figure 3.2 it follows from Theorem 3.67(i) that:

<u>COROLLARY 3.69</u>. A1L = E1L = RE = AIL = EIL.

It similarly follows that:

<u>COROLLARY 3.70</u>. AP1L = EP1L = CS = EPIL = APIL.

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 we will use in Chapter 4 to show the undecidability of whether or not the lengths of strings in PD1L systems grow unboundedly that the following holds:

<u>THEOREM 3.71</u>. *The emptiness of the stable string languages of* PD1L *systems is undecidable.*

Although it is obviously not the case that APD1L = EPD1L we obtain in a similar way the additional result:

<u>THEOREM 3.72</u>. *The emptiness of* EPD1L *languages is undecidable.*

For stable string languages of DOL systems, however, the emptiness problem is solvable. In Section 3.1.1 it was proven that for a DOL system G = <W,P,w> it is decidable whether or not L(G) is finite and that if L(G) is finite then $\#L(G) \preccurlyeq e^{\sqrt{\#W \log \#W}}$. Therefore we can determine whether $A(G) \neq \emptyset$ by, e.g., checking whether $w \Rightarrow w_1 \Rightarrow w_2 \Rightarrow \ldots \Rightarrow w_{i-1} \Rightarrow w_i \Rightarrow w_i$ for some $i \leq \#L(G)$ if L(G) is finite. In fact, according to the theory developed in Section 3.1.1, $i \leq \#W$ suffices.

3.4.2. STABLE STRING LANGUAGES OF TABLE L SYSTEMS.

Let G = <W,$\mathcal{P}$,w> be a table L system. Similarly to the above we define the stable string language of G as

$$A(G) = \{v \in L(G) \mid v \Rightarrow z \text{ implies } z = v\}.$$

The constructions in Lemmas 3.65 and 3.66 show immediately that the analog of Theorem 3.67 holds for table L systems in general and for table L systems using k tables (i.e., T_kL *systems*) in particular. Hence we have the following additional corollaries from Theorem 3.67.

COROLLARY 3.73. $APT_k(m,n)L = EPT_k(m,n)L$ *for all nonnegative integers* m,n,k *such that* m+n > 0 *and* k > 0.

Since by Figure 3.2 EPTIL = CS we have that

COROLLARY 3.74. $APT_1 1L = AP1L = CS = APTIL$.

It furthermore follows from Theorem 3.67(ii) that:

COROLLARY 3.75.
(i) $APT_k 0L \subseteq EPT_k 0L$ *for all* k > 0.
(ii) $APDT_k(m,n)L \subseteq EPDT_k(m,n)L$ *for all* k > 0 *and* m,n ≥ 0.

and more in particular from Theorem 3.67(i) that:

COROLLARY 3.76. $AT_1 1L = A1L = RE = ATIL$.

LEMMA 3.77. *Let* $G = \langle W,P,w\rangle$ *be a* T0L *system. There is an algorithm which, given* G, *produces a* T0L *system* $G' = \langle W',P',w'\rangle$ *and a subset* V_T *of* W' *such that* $L(G') \cap V_T^* = A(G)$.

PROOF. It is easy to see that, for $P = \{P_1,P_2,\ldots,P_k\}$ and $G_i = \langle W,P_i,w\rangle$, $1 \le i \le k$,

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

From HERMAN and WALKER [1975, Lemma 3] it follows that there exists an algorithm which, given $\langle W,P_i\rangle$, i = 1,2,...,k, produces a finite set of strings $W_i \subseteq \Sigma^*$ such that $W_i^* = \{v \in W^* \mid v \underset{G_i}{\Rightarrow} z \text{ implies } z = v\}$. Therefore, $A(G) = \bigcap_{i=1}^{k} W_i^* \cap L(G)$. From HERMAN and ROZENBERG [1975, 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 $G' = \langle W',P',w'\rangle$ and a subset V_T of W' such that $L(G') \cap V_T^* = L(G) \cap L(R)$, where L(R) is the language denoted by R. □

LEMMA 3.78. *Let* $G = \langle W,\mathcal{P},w\rangle$ *be any type of* TOL *system, e.g., propagating, deterministic or both such that* $\#\mathcal{P} > 1$. *There is an algorithm which, given* G *and a subset* V_T *of* W, *produces a* TOL *system* $G' = \langle W',\mathcal{P}',w'\rangle$ *of the same type with* $\#\mathcal{P}' = \#\mathcal{P}$ *such that*

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

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

PROOF. Let $G = \langle W,\mathcal{P},w\rangle$ where $\mathcal{P} = \{P_1,P_2,\ldots,P_k\}$. Construct $G' = \langle W',\mathcal{P}',w'\rangle$ as follows.

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

where $F,s \notin W$. The initial string $w' = s$. $\mathcal{P}' = \{P_1',P_2',\ldots,P_k'\}$ with P_i', $1 \le i \le k$, defined by

(1)	$s \to (a_1,1,1)(a_2,1,1)\ldots(a_n,1,1)$	if $w = a_1a_2\ldots a_n$.
(2)	$(a,j,0) \to (a_1,i,1)(a_2,i,1)\ldots(a_n,i,1)$	for all $j \in \{1,2,\ldots,k\}$ and $a \to a_1a_2\ldots a_n \in P_i$.
(3)	$(a,i,1) \to (a,i,0)$	for all $a \in W$.
(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 $a \in W-V_T$ and all $j \neq i$.
(6)	$F \to FF$	
(7)	$a \to a$	for all $a \in V_T$.

(i) Recall the notion of a control word from the previous Section 3.3 for the notation $v \overset{u}{\underset{G}{\Rightarrow}} v'$, that is, v derives v' by the successive application of tables $P_{i_1},P_{i_2},\ldots,P_{i_n}$ from the table L system $G = \langle W,\mathcal{P},w\rangle$ if $u = i_1i_2\ldots i_n$.

Now suppose $w \overset{*}{\underset{G}{\Rightarrow}} v$ and $v \in V_T^*$. Then there are words $v_0 = w$, $v_1,v_2,\ldots,$ $v_h = v$ in W^* and tables $P_{i_1},P_{i_2},\ldots,P_{i_h}$ in $\mathcal{P}$ such that $v_0 \overset{i_1}{\underset{G}{\Rightarrow}} v_1 \overset{i_2}{\underset{G}{\Rightarrow}} v_2 \overset{i_3}{\underset{G}{\Rightarrow}} \ldots$ $\ldots \overset{i_h}{\underset{G}{\Rightarrow}} v_h$. Let $v_i = a_{i1}a_{i2}\ldots a_{in_i}$ for $i = 0,1,\ldots,h$. Then

$$
\begin{aligned}
s &\overset{1}{\underset{G'}{\Rightarrow}} (a_{01},1,1)(a_{02},1,1)\ldots(a_{0n_0},1,1)\\
&\overset{1}{\underset{G'}{\Rightarrow}} (a_{01},1,0)(a_{02},1,0)\ldots(a_{0n_0},1,0)\\
&\overset{i_1}{\underset{G'}{\Rightarrow}} (a_{11},i_1,1)(a_{12},i_1,1)\ldots(a_{1n_1},i_1,1)\\
&\overset{i_1}{\underset{G'}{\Rightarrow}} (a_{11},i_1,0)(a_{12},i_1,0)\ldots(a_{1n_1},i_1,0)\\
&\vdots\\
&\overset{i_h}{\underset{G'}{\Rightarrow}} (a_{h1},i_h,1)(a_{h2},i_h,1)\ldots(a_{hn_h},i_h,1)\\
&\overset{j}{\underset{G'}{\Rightarrow}} a_{h1}a_{h2}\ldots a_{hn_h} = v \quad \text{for each } j \neq i_h .
\end{aligned}
$$

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

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

Now suppose that $s \overset{*}{\underset{G'}{\Rightarrow}} v$ with $v \in V_T^*$. Since $s \notin V_T^*$ we have, for $w = a_1a_2\ldots a_n$, that

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

with the first production using a rule of type (1). If $z \in V_T^*$ then the only applicable productions are of type (7) and therefore $v = z$. Assume therefore, without loss of generality, that there occurs no word over V_T but v in the above derivation. Scrutiny of the production rules shows that only rules of type (1) - (4) have been used in the above derivation: a rule of type (1) as the first one and a rule of type (4) as the last one. All other rules used must be of type (2) and (3). Therefore the above derivation has to look as follows.

$$
\begin{aligned}
s &\overset{i}{\underset{G'}{\Rightarrow}} (a_{01},1,1)(a_{02},1,1)\ldots(a_{0n_0},1,1)\\
&\overset{1}{\underset{G'}{\Rightarrow}} (a_{01},1,0)(a_{02},1,0)\ldots(a_{0n_0},1,0)\\
&\overset{i_1}{\underset{G'}{\Rightarrow}} (a_{11},i_1,1)(a_{12},i_1,1)\ldots(a_{1n_1},i_1,1)
\end{aligned}
$$

$$\overset{i_1}{\underset{G'}{\Rightarrow}} (a_{11},i_1,0)(a_{12},i_1,0)\dots(a_{1n_1},i_1,0)$$

$$\vdots$$

$$\overset{i_h}{\underset{G'}{\Rightarrow}} (a_{h1},i_h,1)(a_{h2},i_h,1)\dots(a_{hn_h},i_h,1)$$

$$\overset{i_{h+1}}{\underset{G'}{\Rightarrow}} a_{h1}a_{h2}\dots a_{hn_h} = v, \qquad i_{h+1} \neq i_h.$$

But then also

$$w = a_{01}a_{02}\dots a_{0n_0} \overset{i_1}{\underset{G}{\Rightarrow}} a_{11}a_{12}\dots a_{1n_1} \overset{i_2}{\underset{G}{\Rightarrow}} \dots \overset{i_h}{\underset{G}{\Rightarrow}} a_{h1}a_{h2}\dots a_{hn_h} = v,$$

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

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

which together with the previous inclusion gives us that

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

(ii) Suppose $s \overset{*}{\underset{G'}{\Rightarrow}} v$ and $v \in V_T^*$.
Since rules of type (7) are the only ones applicable on $v \in V_T^*$ we have $v \in A(G')$ 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 $v \neq \lambda$, that

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

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

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

THEOREM 3.79. *Let* G *be an* XT_k0L *system,* $X \in \{\lambda, P, PD\}$ *and* $k > 1$. *There exist 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) $$A(G") = L(G) \cap V_T^*.$$

PROOF.

(i) The construction in Lemma 3.66 leaves the propagating and deterministic-property intact and goes through analogously for T0L systems without changing the number of tables (cf. Corollary 3.75(i) and (ii)). This proves (i) for X = P or PD. The case of $X = \lambda$, i.e., G is a T_k0L system is covered by Lemma 3.77 and adds one table. Since from HERMAN and ROZENBERG [1975] 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) Follows by Lemma 3.78. $\square$

COROLLARY 3.80.

(i) $AT_k0L = ET_k0L, \quad k > 1;$

(ii) $APT_k0L = EPT_k0L, \quad k > 1;$

(iii) $APDT_k0L = EPDT_k0L, \quad k > 1.$

Since the construction in the proof of Lemma 3.78 also leaves determinism intact for the nonpropagating case we have that

COROLLARY 3.81. $EDT_k0L \subseteq ADT_k0L$ *for* $k > 1$.

We now need the following results from HERMAN and ROZENBERG [1975, Chs. 7-10] to round off the picture.

LEMMA 3.82.

(i) $$E0L \equiv EP0L \bmod \lambda;$$

(ii) $ET_2OL = ETOL$;

(iii) $EPT_2OL = EPTOL$;

(iv) $ETOL \equiv EPTOL \bmod \lambda$;

(v) $CF \subset EOL \subset ETOL \subset CS$.

And from HERMAN and WALKER [1975],

LEMMA 3.83. $AOL = AT_1OL = CF \equiv APOL \bmod \lambda$.

THEOREM 3.84. *The inclusion relations between the various language families of stable string languages are summarized in the diagram of Figure 3.3. The arrows between the boxes have the usual interpretation as meaning strict inclusion.*

PROOF. By the results of Sections 3.4.1 and 3.4.2 together with the fact that $CF \subset CS \subset RE$, cf. Section 2.1. □

Hence we see that whenever the L systems have some context sensitivity, by having the rewriting of a letter depend on a neighboring letter or by being able to choose production rules by the judicious use of tables, the devices of obtaining languages from the systems by considering only the stable strings or the strings over a terminal alphabet are in general exactly as powerful. This equality in power of the two operations for obtaining languages from L systems breaks down only at the bottom of the scale where there is no context sensitivity whatsoever as in the case of OL systems. There it appears that the use of nonterminals is strictly stronger than the use of stable strings. Of course, the above does not hold completely true for deterministic L systems (with or without tables) where it is clear that e.g. the stable string languages of untabled deterministic L systems can never consist of more than one word while the extensions can. We will look at the deterministic case in more detail below.

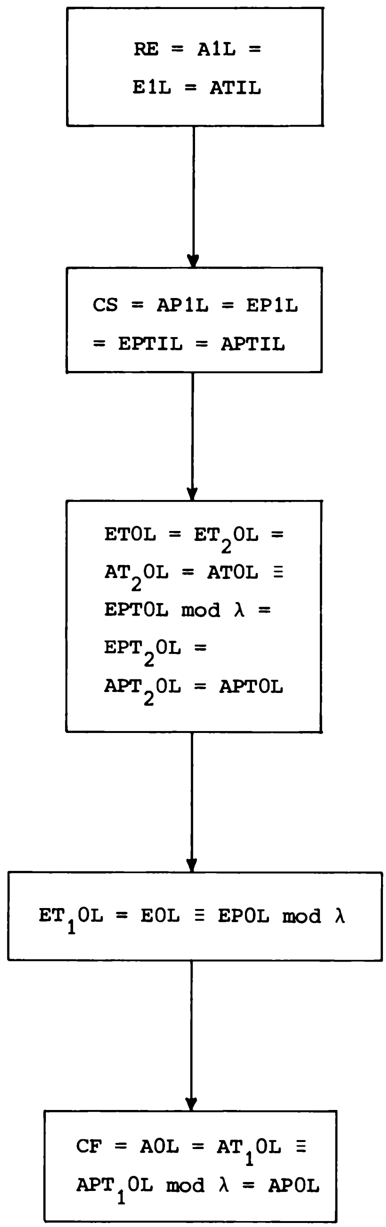


Figure 3.3. Classification of stable string language families of non-deterministic L systems with and without tables.

3.4.3. STABLE STRING LANGUAGES OF DETERMINISTIC TABLE L SYSTEMS

The concept of languages produced by deterministic (or 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 deterministic L systems (i.e., those with one table). However, stable string languages of deterministic L systems using more than one table, or deterministic L languages and their intersections 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.

(1) $\mathrm{APDT}_k\mathrm{0L} = \mathrm{EPDT}_k\mathrm{0L}$ for $k > 1$ by Corollary 3.80(iii).

(2) $\mathrm{EDT}_k\mathrm{0L} \subseteq \mathrm{ADT}_k\mathrm{0L}$ for $k > 1$ by Corollary 3.81.

Since the proof technique of Lemma 3.78 works also in the case of deterministic context sensitive table L systems we have that

(3) $\mathrm{EDT}_k(m,n)\mathrm{L} \subseteq \mathrm{ADT}_k(m,n)\mathrm{L}$ for $k > 1$

(4) $\mathrm{EPDT}_k(m,n)\mathrm{L} \subseteq \mathrm{APDT}_k(m,n)\mathrm{L}$ for $k > 1$.

(4) together with Corollary 3.75(ii) gives us

COROLLARY 3.85. $\mathrm{APDT}_k(m,n)\mathrm{L} = \mathrm{EPDT}_k(m,n)\mathrm{L}$ for $k > 1$.

Together with the results of the previous Sections 3.2 and 3.3 we can now collapse substantial parts of the hierarchies of stable string languages of deterministic table L systems according to the amount of context and/ or number of tables. Let SING denote the family of languages consisting of singleton languages and $\emptyset$. (Where we do not make difference between SING and SING-$\{\lambda\}$.)

THEOREM 3.86. *The inclusion relations between the various language families concerned are given in the diagram of Figure 3.4.*

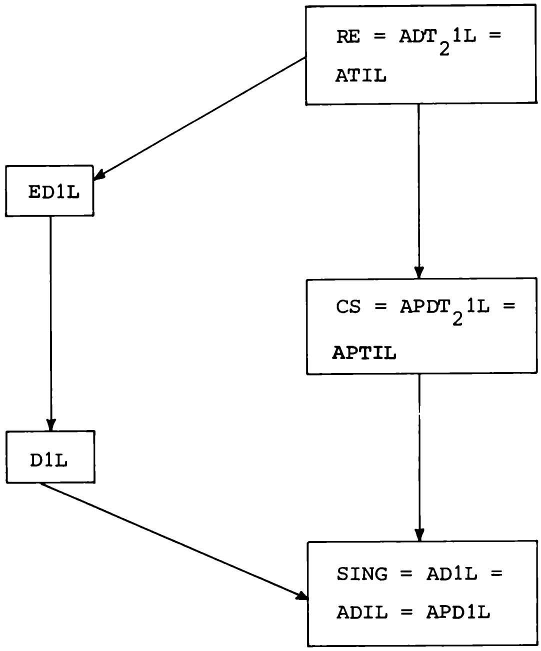


Figure 3.4. Classification of stable string languages of deterministic context sensitive table L systems.

PROOF.

RE = $ADT_2 1L$ = ATIL: By (3) above $EDT_2 1L \subseteq ADT_2 1L$ and, since $EDT_2 1L$ = RE by Theorem 3.61, we have that RE = $ADT_2 1L$ = ATIL. (Every class of languages effectively obtainable which includes RE equals RE.)

CS = $APDT_2 1L$ = APTIL: By Theorem 3.61 we have that CS = $EPDT_2 1L$ and, by Corollary 3.85, $EPDT_2 1L = APDT_2 1L$. Hence CS = $APDT_2 1L$. CS = APTIL by Theorem 3.84.

D1L $\subset$ ED1L $\subset$ RE: By Figure 3.1. The remainder of the theorem is trivial. □

In HERMAN and ROZENBERG [1975] it is shown that two tables suffice to generate all ET0L languages: ET0L = $ET_2 0L \equiv$ EPT0L mod λ and EPT0L = $EPT_2 0L$. The same method, of clocking the use of more tables with one table and fixating derived strings with the other table, works also for the deterministic variants. Hence EDT0L = $EDT_2 0L$ and EPDT0L = $EPDT_2 0L$.

Therefore, it follows from (1) that

$$(5) \qquad APDT_2OL = EPDT_2OL = EPDTOL = APDTOL;$$

and from (2) and since it can be proved that EDTOL $\equiv$ EPDTOL mod λ (Hint: similarly to the proof of ETOL $\equiv$ EPTOL mod λ) we have that:

$$(6) \qquad EDTOL = EDT_2OL = ADT_2OL = ADTOL \equiv APDTOL \bmod \lambda.$$

The family EDTOL resulting from (5) and (6) ties in with Figure 3.4. according to Figure 3.2.

Finally we would like to point out that much more is proven than claimed by means of corollaries etc. in this Section 3.4. The lemmas and theorems hold for any family of L systems which is preserved under the constructions. If e.g. in Lemma 3.78 we change the production $F \to FF$ to $F \to F'$ and $F' \to F$ then the *growth ranges* stay identical. I.e.,

$$\{i \in \mathbb{N} \mid i = \ell g(v) \text{ and } v \in L(G) \cap V_T^*\} =$$

$$= \{i \in \mathbb{N} \mid i = \ell g(v) \text{ and } v \in A(G')\}$$

Also in Lemma 3.66:

$$\{i \in \mathbb{N} \mid i = \ell g(v) \text{ and } v \in A(G)\} =$$

$$= \{i \in \mathbb{N} \mid i = \ell g(v) \text{ and } v \in L(G') \cap V_T^*\}.$$

3.4.4. RELEVANCE TO THEORETICAL BIOLOGY AND FORMAL LANGUAGE THEORY

The problem of equilibrium oriented behavior in biological morphogenis has attracted considerable attention. For instance, TURING [1952] 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 [1957] 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. 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.

Stable string languages of Lindenmayer systems may be 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 and Waddington's epigenetic landscape as a more general concept. In this scheme we would tentatively place the present approach, viz. by discretization of space-time, at an intermediate level. It appeared above 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, the generating power of the stable string operation was investigated with respect to Lindenmayer systems, and it was shown to be equal to the generating power of the operation of intersection with a terminal alphabet except in the case of context free L systems and deterministic L systems (with one table). Furthermore, the results show that several of the language families of the Chomsky hierarchy can be characterized by classes of highly parallel rewriting systems together with a universal operation for obtaining languages. Thus we have given a characterization for these language families which is structurally completely different from that by generative grammars.

3.5. CONTEXT VARIABLE L SYSTEMS AND SOME SIMPLE REGENERATING STRUCTURES

This section does not treat a well-entrenched part of L theory; its aim is to present some tentative ideas, illuminated by examples and rash interpretations, rather than to exhibit a piece of mathematical theory.

As we have seen in the Introduction (Chapter 1), the models treated in this monograph are based on the assumption that the relative position of cells (or compartments) cannot change during growth, and neither can the neighborhoods of daughter cells be different from those of the mother cells. Here we explore some aspects of context variable L systems, where a letter in a string is rewritten according to a selection of letters from that string. An interpretation of this variant can be that the string represents an enumeration of the cells making up the bulky organism, and a selection of letters from the string, determining the way a letter is going to be rewritten, represents the influence of a cell's neighbors in the bulky organism on its behavior. We might go about this as follows. Given the organism at time t, enumerate all cells in it to a linear string and attach to each cell in the string the place numbers of the cells which are going to influence the rewriting of that particular cell. These influential cells could be e.g. those in the physical neighborhood of the cell concerned in the bulky organism at time t. Subsequently, we rewrite the string, each cell according to the cells (scattered throughout the string) which influence it according to its attached place numbers. The resultant string at time t+1 represents an enumeration of the bulky organism at time t+1. If we can also attach the place numbers in the string, of the neighboring cells of each cell in the bulky organism at time t+1, to each cell in the string at time t+1 during the rewriting, we might be able to model the physical constraints on growth in bulky organisms such as cells sliding past each other, cells being of different sizes and forms and pushing against each other in the course of growth, etc. This attaching of neighborhood places to cells resulting from rewriting can be done by giving superscripts to the letters in the right hand sides of production rules. For an attempt towards a general framework in this direction see VITÁNYI [1971]. The context variable L systems we meet in this section form a modest approach towards the goal sketched above. These systems will appear to be especially suited to model certain properties of fully grown organisms and regeneration. The accompanying languages we may call context variable languages.

The main feature that distinguishes context variable L systems from the ordinary ones is that in a context variable L system the relative place of the context of a letter may vary in time and place. This feature makes the concept difficult to handle, but we shall give some simple examples below. In these examples the systems seem to strive at attaining a certain fully grown size and structure which, however, is not terminal in the usual sense. Cells, i.e., letters, are changing state, dividing and dying all the time. When we chop off a piece we observe a certain regenerative behavior. (Note that this phenomenon of dynamically stable strings is similar to the stable strings encountered in the previous section).

A *context variable* L *system* or CV L system is a triple $G = \langle W,\delta,w\rangle$ such that W is a finite nonempty alphabet of letters; the transition function δ maps elements of $W^* \times W$ to strings consisting of letters in W, each letter superscripted with an element $\tau \in \mathbb{Z}^*$ ($\mathbb{Z}$ is the set of *integers* $\{0,\pm1,\pm2,\dots\}$). I.e.,

$$\delta(a_1a_2\ldots a_{n-1},a_n) = b_1^{\tau_1}b_2^{\tau_2}\ldots b_m^{\tau_m}.$$

with

$$\tau_j = \rho_{j1}\rho_{j2}\ldots\rho_{jn_j}, \qquad \rho_{jh} \in \mathbb{Z}$$

for $1 \le j \le m$ and $0 \le h \le n_j$; the initial string w is a string over W with each letter superscripted with a string over $\mathbb{Z}$, i.e.,

$$w = a_1^{\tau_1}a_2^{\tau_2}\ldots a_n^{\tau_n},$$

with the τ 's as above.

The superscript $\tau_j = \rho_{j1}\rho_{j2}\ldots\rho_{jn_j}$ selects in a string $b_1^{\tau_1}b_2^{\tau_2}\ldots b_j^{\tau_j}\ldots$ $\ldots b_n^{\tau_n}$ the *context* $h(b_j^{\tau_j})$ according to which the letter b_j is going to be rewritten:

$$h(b_j^{\tau_j}) = b_{j+\rho_{j1}}b_{j+\rho_{j2}}\ldots b_{j+\rho_{jn_j}}.$$

If $j+\rho_{ji} < 1$ or $>n$ we substitute the empty word λ for $b_{j+\rho_{ji}}$ in $h(b_j)$. The CV L system G generates words as follows.

Let $x = a_1^{\tau_1}a_2^{\tau_2}\ldots a_n^{\tau_n}$ be a string. Then x *generates* y *directly,* written as $x \Rightarrow y$ if

$$x = a_1^{\tau_1}a_2^{\tau_2}\ldots a_n^{\tau_n},$$

$$y = \alpha_1\alpha_2\ldots\alpha_n,$$

and for every j, $1 \le j \le n$,

$$\alpha_j = \delta(a_{j+\rho_{j1}}a_{j+\rho_{j2}}\ldots a_{j+\rho_{jn_j}},a_j)$$

with

$$\rho_{j1}\rho_{j2}\ldots\rho_{jn_j} = \tau_j.$$

$\overset{*}{\Rightarrow}$ denotes the reflexive and transitive closure of $\Rightarrow$; and $x \overset{(k)}{\Rightarrow} y$ denotes a chain of length k:

$$x = x_0 \Rightarrow x_1 \Rightarrow \ldots \Rightarrow x_k = y.$$

If $x \overset{*}{\Rightarrow} y$ we say x *derives* y and if $x \overset{(k)}{\Rightarrow} y$, we say x *derives* y *in* k *steps*. A string $x = a_1^{\tau_1} a_2^{\tau_2} \ldots a_n^{\tau_n}$ is called a *description*, and an element of x is called a *cell*. Let G be as above. The CV L *language* produced by G is the set $L(G) \subseteq W^*$ defined by

$$L(G) = \{a_1 a_2 \ldots a_n \mid w^{\bullet} \overset{*}{\Rightarrow} a_1^{\bullet} a_2^{\bullet} \ldots a_n^{\bullet}\}$$

For ease of notation we write $a_1 \ldots a_n b \to y$ for $\delta(a_1 a_2 \ldots a_n, b) = y$.

<u>EXAMPLE</u>. $G = \langle\{a\},\{a \to a^{-1}a^{+1},\ aa \to \lambda\},a\rangle$. Then $a \Rightarrow a^{-1}a^{+1} \Rightarrow a^{-1}a^{+1}a^{-1}a^{+1} \Rightarrow a^{-1}a^{+1}a^{-1}a^{+1} \Rightarrow \ldots$.

We notice that when the description has reached a certain fully grown size it does not change any more although the individual letters certainly are not terminal or static, i.e. letters are dividing and dying all the time but the structure, complete with context relations, stays unaltered. The language generated by this example is

$$L(G) = \{a,aa,aaaa\}.$$

Let $G(k) = \langle\{a\},\{a \to a^{-k}a^{+k}, aa \to \lambda\},a\rangle$. The language produced by G(k) will be called $L^a(k)$. Then $L^a(1) = \{a,aa,aaaa\}$. In a similar way we obtain

$L^a(2) = \{a,aa,aaaa\}$

$L^a(3) = \{a,aa,aaaa,aaaaaaaa\}$

$L^a(4) = L^a(3)$

$L^a(5) = \{a^n \mid n = 1,2,4,8,12\}$

$L^a(6) = L^a(5)$

etc.

$L^a(0) = \{\lambda,a,aa\}$

$L^a(-1) = \{\lambda,a,aa\}$

$L^a(-2) = \{a,aa,aaaa\}$

$L^a(-3) = L^a(-2)$

etc.

The general form of such an $L^a(k)$ language is described by:

LEMMA 3.87. *Let* $G(k)$ *and* $L^a(k)$ *be as above.*

(i) $k > 0$ *and* k *is even:*

$$L^a(k) = \{a^{2^t} \mid t \in \mathbb{N} \text{ and } 0 \leq t \leq \log_2(k) + 1\} \cup \{a^{2k}\}.$$

$k > 0$ *and* k *is odd*: $L^a(k) = L^a(k+1)$.

(ii) $k < -1$: $L^a(k) = L^a(-k-1)$

(iii) $L^a(0) = \{\lambda, a, aa\}$; $L^a(-1) = \{\lambda, a, aa\}$.

PROOF. By $\delta^t(a)$ we mean $\dot{a}_1\dot{a}_2\ldots\dot{a}_n$ if $a \overset{(t)}{\Rightarrow} \dot{a}_1\dot{a}_2\ldots\dot{a}_n$.

(i) $k > 0$.

(a) $t \leq \log_2 k$. Then $|\delta^t(a)| = 2^t \leq k$ and there are no cells in $\delta^t(a)$ for which production rule $aa \to \lambda$ is applicable. Hence all cells divide and $|\delta^{t+1}(a)| = 2^{t+1}$.

(b) $\log_2 k < t \leq \log_2(k) + 1$. Let $\delta^t(a) = \dot{a}_1\dot{a}_2\ldots\dot{a}_n$. For all cells a^{+k}_{2i} and $a^{-k}_{2^t-2i+1}$ $(i > 0)$, such that $2i + k \leq 2^t$, production rule $aa \to \lambda$ will be applied. Let $j = \max_{2i+k\leq 2^t}(i)$; then there are $2j$ cells in $\delta^t(a)$ such that $aa \to \lambda$ will be the applied production rule. For k is even: $2j + k = 2^t$ or $2^t - 2j = k$. $2j$ cells disappear and k cells divide in the next production, so $|\delta^{t+1}(a)| = 2k$. For k is odd $2j + k = 2^t - 1$ or $2^t - 2j = k + 1$. $2j$ cells disappear and $k + 1$ cells divide in the next production, so $|\delta^{t+1}(a)| = 2k + 2$.

(c) $t > \log_2(k) + 1$. The last production gave us $|\delta^t(a)| = 2k$ (k even), so half of the cells divide and the other half disappears in the next production: $|\delta^{t+1}(a)| = 2k$. For k is odd we get $|\delta^{t+1}(a)| = 2k + 2$.

(ii) Similar to (i).

(iii) Follows from the productions. □

It follows from the above that

$$\bigcup_{k\in\mathbb{Z}} L^a(k) = \{a^{4n} \mid n \geq 0\} \cup \{a, aa\}.$$

The CV L systems we have been considering all start from a single cell, and, according to the predetermined genetical instructions (i.e., δ and the specification of k) they grow at an exponential rate until the fully grown size is reached but for one production step. Next the CV L system grows on the remainder and stays at the same size and structure, although in each production step individual cells disappear and divide. Note that there is a limited interaction all the time between the cells to achieve this goal.

We can investigate regenerative processes in these systems, by removing part of the (fully grown or growing) description. The missing part then is regrown again. When we divide a description into several parts, all of these will eventually reach a fully grown stage. This is reminiscent of the remarkable regenerative properties of flatworms. The discussed CV L systems are very simple, i.e., there is no differentiation of cells. It would be interesting to investigate similar regenerative processes in more complex CV L systems with, e.g., more cellular states. By noting that a CV L system, or indeed a D2L system, can simulate a Turing machine we can simulate totally regenerative structures. That is, a structure which, when we chop off any piece of it, regrows the missing piece. Here, however, we would have to increase the number of states rather drastically. We may qualify questions about regeneration by distinguishing between several types of regeneration: (i) Starting with one cell in a special state, i.e. reproduction. (ii) Starting from arbitrary parts of a fully grown description. (iii) starting from arbitrary parts of a description at some stage of the growth process. (iv) Starting from selected parts removed from the fully grown description, etc. Note that above there is a difference between cases where we remove an end part of a fully grown description, and cases where we remove a middle part. We illustrate this with the following example (k = 2): The fully grown description is: $a^{-2}a^{+2}a^{-2}a^{+2}$. Regeneration with the left-end (skin) cell removed: $a^{+2}a^{-2}a^{+2} \Rightarrow a^{-2}a^{+2}a^{-2}a^{+2}$. The two cells right have divided, while the new leftmost cell has disappeared in the production. Regeneration with the third (middle) cell removed: $a^{-2}a^{+2}a^{+2} \Rightarrow a^{-2}a^{+2}a^{-2}a^{+2}$ $a^{-2}a^{+2} \Rightarrow a^{-2}a^{+2}a^{-2}a^{+2}$. All three cells divide in the first production. In the second production only the two outermost cells divide and the others disappear: the fully grown size is reached.

We observe that the removal of different parts of the fully grown description may yield different courses for the regenerative process.

3.5.1. THE EXTENDED FRENCH FLAG PROBLEM

Usually the *French Flag problem* is stated as follows: suppose we have a string of cells all of which are in an identical state but because of some disturbance produce the pattern of a French Flag, i.e. one third red, one third white and one third blue. Moreover, when we cut off any piece of it which is large enough it produces this pattern again. The above is supposed to be (e.g. HERMAN [1972]) a meaningful statement of problems of biological regeneration. However, as we have stated before, what seems more meaningful is the design of structures which, starting from a single cell, attain a certain fully grown stage, no cell staying static, and furthermore, when we chop off a piece of this structure, regrow the missing piece until the fully grown stage has been reached again. When we discuss the French Flag in this context what we want is that:

(i) One cell divides and gives rise to a fully grown French Flag of a certain size which retains the same pattern and structure while individual cells are disappearing all the time.

(ii) When we chop off a piece of a fully grown French Flag it regrows the missing piece.

We will present a CV L system which does (i) and (ii). Since the system has to reach a certain fully grown size, clearly the production rules depend on this size. When we want a different fully grown size we will have to find a new set of productions.

Furthermore, in the discussed system the a's serve as some kind of "head" of the structure, i.e., the front part always regenerates a new end part but an end part does not always regenerate a new front part. When part of the head is contained in it, however, it does. The biological interpretation of this phenomenon is so obvious (lizards!) that this type of partial regeneration need not be justified further. We may point out that higher organisms, which are more differentiated, mostly lose regenerative properties to a certain extent which seems to be the price to be paid for a more complex structure. We exhibit an example of a context variable Lindenmayer system with maximal a two neighbor context, which, starting from a single cell, attains the following fully grown description, viz. the French Flag

$$a^{\bullet}a^{\bullet}a^{\bullet}a^{\bullet}b^{\bullet}b^{\bullet}b^{\bullet}b^{\bullet}c^{\bullet}c^{\bullet}c^{\bullet}c^{\bullet}.$$

When this French Flag is cut, the left part always regenerates completely;

the right part mostly not, depending on where the cut was placed. We will call $a^{\bullet}a^{\bullet}a^{\bullet}a^{\bullet}$ the *head*, $b^{\bullet}b^{\bullet}b^{\bullet}b^{\bullet}$ the *trunk* and $c^{\bullet}c^{\bullet}c^{\bullet}c^{\bullet}$ the *tail* of the French Flag.

$W = \{a,b,c\}$. The transition function is specified by the following rules (we only write those we need and leave the others open):

$a \to a^{-1+1}b^{-1+1}$	$aab \to a^{-1+1}a^{-1+1}$
$b \to b^{-1+1}c^{+1-1}$	$bbc \to b^{-1+1}b^{-1+1}$
$c \to c^{+1-1}c^{+1-1}$	$aba \to a^{-1+1}a^{-2+2}$
$aa \to \lambda$	$bcb \to b^{-1+1}b^{-2+2}$
$bb \to \lambda$	$cbc \to c^{+1-1}c^{+1-1}$
$cc \to c^{+1-1}c^{+1-1}$	$bac \to b^{+2+1}b^{-1+1}$
$ab \to a^{+2+1}a^{-1+1}$	$ccb \to c^{+1-1}c^{+1-1}$
$bc \to b^{+2+1}b^{-1+1}$	$bab \to \lambda$
$ba \to b^{-1+1}c^{+1-1}$	$bcc \to \lambda$
$cb \to c^{+1-1}c^{+1-1}$	$cbb \to \lambda$
$aaa \to a^{-1+1}a^{-2+2}$	$cba \to \lambda$
$bbb \to b^{-1+1}b^{-2+2}$	$acb \to \lambda$
$ccc \to \lambda$	

Starting from initial string a we obtain the following production:

(1) $$a \Rightarrow a^{-1+1}b^{-1+1} \Rightarrow a^{+2+1}a^{-1+1}b^{-1+1}c^{+1-1}$$

$$\Rightarrow a^{-1+1}a^{-2+2}a^{-1+1}a^{-1+1}b^{+2+1}b^{-1+1}c^{+1-1}c^{+1-1}$$

$$\Rightarrow a^{-1+1}a^{-2+2}a^{-1+1}a^{-1+1}b^{-1+1}b^{-2+2}b^{-1+1}b^{-1+1}c^{+1-1}c^{+1-1}c^{+1-1}c^{+1-1}$$

$$\Rightarrow a^{-1+1}a^{-2+2}a^{-1+1}a^{-1+1}b^{-1+1}b^{-2+2}b^{-1+1}b^{-1+1}c^{+1-1}c^{+1-1}c^{+1-1}c^{+1-1}$$

$\Rightarrow$ idem.

We call this fully grown description FF, and observe that FF is the desired French Flag; it stays at this structure although the individual cells are dividing and dying off continuously. Note that the head grows fastest and is completed first. Next we investigate the regenerative properties. There are eleven places at which FF can be cut. When we look at the left part resulting from such a cut we see: (N.B. We will sometimes omit superscripts when no confusion can result, e.g. a^4b^{-1+1} for $a^{-1+1}a^{-2+2}a^{-1+1}a^{-1+1}b^{-1+1}$.)

(2.1) $a^{-1+1} \Rightarrow a^{-1+1}b^{-1+1} \overset{*}{\Rightarrow}$ FF by (1)

(2.2) $a^{-1+1}a^{-2+2} \Rightarrow a^{-1+1}b^{-1+1} \overset{*}{\Rightarrow}$ FF by (1)

(2.3) $a^{-1+1}a^{-2+2}a^{-1+1} \Rightarrow a^{-1+1}b^{-1+1} \overset{*}{\Rightarrow}$ FF by (1)

(2.4) $a^{-1+1}a^{-2+2}a^{-1+1}a^{-1+1} \Rightarrow a^{-1+1}a^{-2+2} \overset{*}{\Rightarrow}$ FF by (2.2)

(2.5) $a^4b^{-1+1} \Rightarrow a^4b^{-1+1}c^{+1-1} \Rightarrow a^4b^{+2+1}b^{-1+1}c^{+1-1}c^{+1-1} \Rightarrow$ FF

(2.6) $a^4b^{-1+1}b^{-2+2} \Rightarrow a^4b^{-1+1}c^{+1-1} \overset{*}{\Rightarrow}$ FF by (2.5)

(2.7) $a^4b^{-1+1}b^{-2+2}b^{-1+1} \Rightarrow a^4b^{-1+1}c^{+1-1} \overset{*}{\Rightarrow}$ FF by (2.5)

(2.8) $a^4b^{-1+1}b^{-2+2}b^{-1+1}b^{-1+1} \Rightarrow a^4b^{-1+1}b^{-2+2} \overset{*}{\Rightarrow}$ FF by (2.6)

(2.9) $a^4b^4c^{+1-1} \Rightarrow a^4b^4c^{+1-1}c^{+1-1} \Rightarrow$ FF

(2.10) $a^4b^4c^{+1-1}c^{+1-1} \Rightarrow$ FF

(2.11) $a^4b^4c^{+1-1}c^{+1-1}c^{+1-1} \Rightarrow$ FF.

Hence all left parts regenerate completely. The reader may verify that the fully grown descriptions reached by the right parts are according to (3.1)-(3.11) (when the cuts are placed as in (2.1)-(2.11)).

(3.1) $a^3b^4c^4 \Rightarrow$ FF

(3.2) $a^2b^4c^4 \Rightarrow a^2b^4c^4$

(3.3) $a\,b^4c^4 \overset{*}{\Rightarrow}$ FF

(3.4) $b^4c^4 \Rightarrow b^4c^4$

(3.5) $b^3c^4 \Rightarrow b^4c^4$

(3.6) $b^2c^4 \Rightarrow b^2c^4$

(3.7) $b\,c^4 \overset{*}{\Rightarrow} b^4c^4$

(3.8) $c^4 \Rightarrow c^4$

(3.9) $c^3 \Rightarrow c^4$

(3.10) $c^2 \Rightarrow c^4$

(3.11) $c \overset{*}{\Rightarrow} c^4$.

We may also cut a piece out of the middle of FF. It may be verified that

(4.1) Every part of FF containing cells of the head regenerates completely to FF except parts of the form $a^{-1+1}a^{-1+1}\eta$

(i) $a^{-1+1}a^{-1+1} \Rightarrow \lambda$

(ii) $a^{-1+1}a^{-1+1}\eta \overset{*}{\Rightarrow} a^2b^4c^4$ for $\eta \neq \lambda$.

(4.2) Every part of FF containing cells of the trunk but no head cells grows to a fully grown description b^4c^4 except parts of the form $b^{-1+1}b^{-1+1}\eta$

(i) $b^{-1+1}b^{-1+1} \Rightarrow \lambda$

(ii) $b^{-1+1}b^{-1+1}\eta \overset{*}{\Rightarrow} b^2c^4$ for $\eta \neq \lambda$.

(4.3) Every part of FF consisting of tail cells grows to a full tail c^4, i.e. a fully grown description.

Of course, the CV L systems as defined do not have a greater generating power than do ordinary L systems, since it is easy to define for each CV L system an equivalent ordinary L system. They do, however, have a structural simplicity which exhibits the phenomenon to be modeled more transparently. Also, the general idea presented in the beginning of this section permits more powerful devices as in VITÁNYI [1971]. A further development of the theory of CV L systems can be found in RUOHONEN [1974] where also the connection with ordinary L system theory is investigated.

3.6. BIBLIOGRAPHICAL COMMENTS

Section 3.1.1 is based on VITÁNYI [1972b, 1974a] and Section 3.1.2 on VITÁNYI [1976b]. Section 3.2 is based on VITÁNYI [1976a] and contains new material, e.g., all results connected with FIN and $DLBA_{exp}$. The proof of Theorem 3.54(iii) is based on an idea of P.G. DOUCET. Section 3.3 follows VITÁNYI [1977a]. For a classification of families of pure context sensitive L languages with respect to each other see ROZENBERG [1972a,b] and ROZENBERG and LEE [1975]. For a similar classification of the table variants see LEE and ROZENBERG [1974]. Facts about context free table L systems can be found in HERMAN and ROZENBERG [1975], and about context free L systems in HERMAN and ROZENBERG [1975], ROZENBERG and DOUCET [1971], and NIELSEN, ROZENBERG, SALOMAA and SKYUM [1974a,b]. The material presented in Section 3.4 stems from VITÁNYI and WALKER [1978]. Further information about stable string languages of L systems can be found in WALKER [1974a,b,c; 1975] and HERMAN and WALKER [1975, 1976]. Section 3.5 is based on an idea in VITÁNYI [1971] and follows VITÁNYI [1972a]. CV L systems were further investigated by RUOHONEN [1974] and regeneration in symmetric DIL systems in RUOHONEN [1976]. All throughout Chapter 3 we have drawn somewhat on general knowledge, e.g., in the range of HERMAN and ROZENBERG [1975]. In particular in Sections 3.2.1 - 3.2.2 we used a simulation technique originating from van DALEN [1971].

CHAPTER 4.

GROWTH FUNCTIONS

The study of the changes in size and weight of a growing organism as a function of elapsed time constitutes a considerable part of the literature on developmental biology. Usually, genetically identical specimens of a specific organism are investigated in controlled environments and their changes in size and weight in time are described. The scientific presupposition is that identical genetic material and identical environments will result in identical growth rates, i.e., that the experiment is repeatable. This assumes a deterministic (i.e. causal) underlying structure and makes a good case for the biological relevance of the study of growth functions of deterministic L systems, where we assume that the production rules reflect the simultaneous influence of the inherited genetic factors and a specific environment on the developmental behavior of cells. Thus, when an organism is growing under optimal conditions it may be assumed that its growth rate, and that of its parts, is governed by internal inherited factors. One of the easiest things to observe about a filamentous organism is the number of cells it has. Suppose, having observed the development of a particular organism, we generalize our observations by giving a function f such that f(t) is the number of cells in the organism after t steps. The problem then arises to produce a developmental system whose growth function is f.

One of the restrictions on the models we consider lies in their one-dimensional nature. This implies, as noted before, that at the present time they are only applicable to filamentous organisms or to one-dimensional aspects of the growth of bulky organisms (such as length measurements). Multi-dimensional models similar to L systems have recently been introduced by various workers (e.g. graph L systems), their growth functions have been investigated e.g. by ČULIK [1975] and were found to be simple extensions of the one-dimensional case. Thus the restriction to one-dimensional growth descriptions can be viewed as a temporary and not very essential one.

One of the problems we shall consider in detail below is the following. Clearly, any growth function which can be achieved by a D0L system can also be achieved by a D1L system, simply by giving production rules for the D1L system which for all practical purposes ignore the state of the neighbor. The question arises whether the converse is also true. It will appear that it is not: if a D0L system keeps growing at all it must be growing "fast" as opposed to systems with interactions which are capable of "slow" but nevertheless unbounded growth. Thus, interactions between cells provides organisms with the capability of controlling the rate of their growth in an orderly manner. When this interaction breaks down, tumors containing cells which do not interact with their neighbors may begin to grow at an exponential rate. For this reason, some early workers in the field of growth functions referred to such exponential growth as "malignant".

DEFINITION 4.1. Let $G = \langle W,\delta,w\rangle$ be a DIL system. Then the function f_G from the nonnegative integers into the nonnegative integers defined by $f_G(t) = \ell g(\delta^t(w))$ for all t, is said to be the *growth function* of G.

EXAMPLE 4.2. Let $G = \langle\{a,b\},\{\delta(a) = b,\delta(b) = ab\},a\rangle$ be a D0L system. Then,

$$f_G(0) = f_G(1) = 1,$$

and for all t such that $t \geq 0$,

$$f_G(t+2) = f_G(t+1) + f_G(t).$$

Thus, $f_G(t)$ is the t-th element of the well-known Fibonacci sequence 1,1,2, 3,5,8,13,21,... .

EXAMPLE 4.3. Let $G = \langle\{a,b,c\},\{\delta(a) = abc^2,\delta(b) = bc^2,\delta(c) = c\},a\rangle$ be a D0L system. Then,

$$f_G(0) = \ell g(a) = 1,$$

$$f_G(1) = \ell g(abc^2) = 4,$$

$$f_G(2) = \ell g(abc^2bc^4) = 9,$$

$$f_G(3) = \ell g(abc^2bc^4bc^6) = 16.$$

In fact for all $t > 0$,

$$f_G(t) = f_G(t-1) + 2t + 1.$$

By induction it follows that $f_G(t) = (t+1)^2$.

In investigating growth functions, one of the first questions we ask is what rates of growth are possible. That the rate of growth of a DxL system is at most exponential follows from the next lemma which is immediate from the definitions. (In the sequel of this chapter we will mean by a DxL system a D0L-, D1L- or D2L system).

LEMMA 4.4. *For a* DxL *system* $G = \langle W,\delta,w\rangle$, $x \in \{0,1,2\}$, *we have that* $f_G(t) \leq \ell g(w)m^t$, *where* m *is the maximal length of a value* δ *may have. (I.e.,* $m = \max\{\ell g(\alpha) \mid \alpha$ *is in the range of the set of production rules* $\delta\}$.*)*

The problems which have been investigated with respect to growth functions fall roughly into the following six catagories.

(i) *Analysis problems.* Given a DxL system, describe its growth function in some fixed predetermined formalism.

(ii) *Synthesis problems.* Given a function f in some fixed predetermined formalism and an $x \in \{0,1,2\}$, find a DxL system whose growth function is f. Related to this is the problem: which functions can be growth functions of DxL *systems*?

(iii) *Growth equivalence problems.* Given two DxL systems, decide whether or not they have the same growth function.

(iv) *Classification problems.* Given a DxL system decide what is its growth type. (E.g., is there a polynomial or even a constant which bounds its growth function. Growth types will be rigorously defined in Definition 4.13.)

(v) *Structural problems.* What properties of production rules induce what types of growth?

(vi) *Hierarchy problems.* Is the set of growth functions of DxL systems a proper subset of the set of growth functions of D(x+1)L systems, $x \in \{0,1\}$, and similar problems.

In the first five cases we would like to solve our problems effectively. That is, we would like to be able to write computer programs (algorithms) which, in the case of the analysis problem, say, provide us with an explicit description of the growth function whenever they are given the description of a DxL system.

4.1. DOL GROWTH FUNCTIONS: ANALYTICAL APPROACH

I: *Analysis.* For DOL growth functions we can derive a closed form solution for f_G as follows. (PAZ and SALOMAA [1973], SALOMAA [1973b].)

Associate with each element v of W^* its *Parikh vector* $\bar{v}$, i.e., the row vector $(i_1,i_2,\ldots,i_n)$ where i_j denotes the number of occurrences of a_j in v, $1 \le j \le n$, for $W = \{a_1,a_2,\ldots,a_n\}$. The *growth matrix* M_G of $G = \langle W,\delta,w\rangle$ is the n×n matrix of which the j-th row consists of $\overline{\delta(a_j)}$. It is easy to see that $\overline{\delta^t(w)} = \bar{w}\, M_G^t$ and that

$$(1) \qquad f_G(t) = \bar{w}\, M_G^t\, \eta$$

where $\eta = (1,1,\ldots,1)^T$: the n-dimensional column-vector with all entries equal 1. (T denotes transposition). Now $f_G(t)$ is the number of cells constituting the organism at time t. If we want $f_G(t)$ to denote the length/weight of the organism at time t, and if different cell types have different lengths/weights, we only have to choose η in $\mathbb{R}_+^n$ (where $\mathbb{R}_+^n$ denotes the n dimensional real space coordinates), such that the j-th element of η is the length/weight of a cell a_j.

According to the Cayley-Hamilton theorem, M_G must satisfy its own characteristic equation $p(x) = \det(M_G - Ix) = 0$, where I is the n×n identity matrix. Hence $p(M_G) = 0$, where 0 stands for the n×n matrix with all of its entries 0. Then for each $t \ge n$, after multiplication of $p(M_G)$ with M_G^{t-n}, left multiplication with $\bar{w}$ and right multiplication with η, the following homogenous linear difference equation with constant coefficients holds:

$$(2) \qquad f_G(t) = \sum_{i=1}^{n} b_i\, f_G(t-i), \qquad t \ge n,$$

such that $p(x) = \sum_{i=0}^{n} b_i x^{n-i} = 0$, $b_0 = -1$, is the characteristic equation of M_G. It is well known that such difference equations have solutions of the form appearing as the closed form solution for DOL growth functions in the following:

<ins>THEOREM 4.5.</ins> *Let* $G = \langle W,\delta,w\rangle$ *be a* DOL *system. Then*

(3) $$f_G(t) = \sum_{i=1}^{r} p_i(t)c_i^t, \qquad t \geq \#W = n,$$

where the c_i*'s are the* r *distinct roots of the characteristic equation of* M_G*, and* $p_i(t)$ *is a polynomial in* t *of degree* 1 *less than the multiplicity of the root* c_i*,* $1 \leq i \leq r$*. The constant coefficients in the terms of the polynomials* $p_1(t), p_2(t), \ldots, p_r(t)$ *are determined from* $f_G(s), f_G(s+1), \ldots, f_G(n-1)$ *where* s *is the multiplicity of the zero root in* $p(x) = 0$*, the characteristic equation of* M_G*. (Remember that* $f_G(t) = \ell g(\delta^t(w))$ *gives us the initial values of* f_G*.)*

EXAMPLE 4.6. Take the DOL system of Example 4.2. Then

$$M_G = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix},$$

$$\det(M_G - Ix) = x^2 - x - 1.$$

Hence the roots are: $x_{1,2} = \frac{1}{2}(1 \pm \sqrt{5})$ and

$$f_G(t) = a_1((1+\sqrt{5})/2)^t + a_2((1-\sqrt{5})/2)^t.$$

Since $f_G(0) = 1$ and $f_G(1) = 1$ we have:

$$f_G(t) = \frac{1+\sqrt{5}}{2\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^t - \frac{1-\sqrt{5}}{2\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^t$$

EXAMPLE 4.7. Let $G = \langle\{a,b,c\},\{\delta(a) = a^2, \delta(b) = a^5b, \delta(c) = b^3c\}, a^m b^n c^p\rangle$ be a DOL system. The characteristic equation $x^3 - 4x^2 + 5x - 2 = 0$ of the growth matrix

$$M_G = \begin{pmatrix} 2 & 0 & 0 \\ 5 & 1 & 0 \\ 0 & 3 & 1 \end{pmatrix}$$

has roots $x_1 = x_2 = 1$ and $x_3 = 2$. (Note that M_G is independent of the initial string.) Since the initial string has n occurrences of a, m occurrences of b and p occurrences of c, we obtain as the growth function of G:

$$f_G(t) = a_1 + a_2t + a_3 2^t,$$

where

$$f_G(0) = a_1 + a_3 = m + n + p,$$

$$f_G(1) = a_1 + a_2 + 2a_3 = 2m + 6n + 4p,$$

$$f_G(2) = a_1 + 2a_2 + 4a_3 = 4m + 16n + 22p.$$

Consequently,

$$f_G(t) = (m + 5n + 15p)2^t - 12pt - 4n - 14p.$$

This shows immediately, that G has an exponentially increasing growth function for all initial strings unequal to λ.

An alternative approach for solving the analysis problem, which is also of use for the growth equivalence problem and the synthesis problem, is an application of the theory of generating functions.

DEFINITION 4.8. With any function f from the nonnegative integers into the nonnegative integers we associate its *generating function* $F(x)$ which is defined to be the formal infinite power series $\Sigma_{t=0}^{\infty} f(t)x^t$. We also say that $F(x)$ *generates* f.

The reason for such a definition is that very often the function $F(x)$ can be represented in a simple way. For example, if $f(t) = 2^t$ then $F(x) = 1/(1-2x) = 1 + 2x + 4x^2 + 8x^3 + \ldots$.

The following lemmas are well known and easily proven mathematical facts; ($p(x)/q(x)$ denotes the quotient, $p(x)q(x)$ the product of the polynomials p and q).

LEMMA 4.9.

(i) *If* $p(x)$ *and* $q(x)$ *are two polynomials with integer coefficients such that* $q(0) = 1$, *then* $p(x)/q(x)$ *uniquely determines an infinite power series with integer coefficients, i.e.* $p(x)/q(x) = \Sigma_{t=0}^{\infty} f(t)x^t$, *where* $f(t)$ *is an integer for all* t. *Thus* $p(x)/q(x)$ *generates the function* f. *Furthermore, given* $p(x)$ *and* $q(x)$, $f(t)$ *is effectively computable for every nonnegative integer* t.

(ii) *Let* $p(x)$, $q(x)$, $p'(x)$ *and* $q'(x)$ *be polynomials with integer* *coefficients such that* $q(0) = q'(0) = 1$, *and let* f *and* f' *be functions generated by* $p(x)/q(x)$ *and* $p'(x)/q'(x)$, *respectively. Then* $f(t) = f'(t)$ *for all* t *if and only if* $p(x)q'(x) = p'(x)q(x)$, *for all* x. *Thus it is effectively decidable whether or not* $p(x)/q(x)$ *and* $p'(x)/q'(x)$ *generate the same function.*

LEMMA 4.10. *Let* n *be any integer and let* A *be an* $n \times n$ *matrix whose entries are polynomials in* x *with integer coefficients. Let* $q(x) = \det(A)$. *If there exists a value of* x *such that* $q(x) \neq 0$, *then* A *is invertible, i.e. there exists an* $n \times n$ *matrix* A^{-1} *such that* $AA^{-1} = A^{-1}A = I$, *where* I *denotes the* $n \times n$ *identity matrix. Furthermore, given* A, A^{-1} *can be effectively obtained, and each entry of* A^{-1} *will be of the form* $p_{i,j}(x)/q(x)$, *where* $p_{i,j}(x)$ *is a polynomial with integer coefficients.*

These lemmas lead us to the following theorem.

THEOREM 4.11. *There is an algorithm which, for any* DOL *system* G, *effectively computes two polynomials* $p(x)$ *and* $q(x)$ *with integer coefficients where* $q(0) = 1$, *such that* $p(x)/q(x)$ *generates the growth function* f_G *of* G.

PROOF. Let G be the given DOL system and let $\bar{w}$, M_G and η be as usual. Suppose the alphabet of G contains n elements. Let $M_G x$ be the $n \times n$ matrix obtained by multiplying each entry of M_G by the variable x. Let I denote the $n \times n$ identity matrix. Then $I - M_G x$ is an $n \times n$ matrix whose entries are polynomials with integer coefficients. Let $q(x) = \det(I - M_G x)$. Since $q(0) = 1$ we see that $I - M_G x$ is an invertible matrix. According to Lemma 4.10 we can effectively produce an $n \times n$ matrix $(I - M_G x)^{-1}$ whose entries are all of the form $p_{i,j}(x)/q(x)$, where $p_{i,j}(x)$ and $q(x)$ are polynomials with integer coefficients. Clearly, $\bar{w}(I - M_G x)^{-1}\eta$ is of the form $p(x)/q(x)$ where $p(x)$ is a polynomial with integer coefficients and can be effectively computed. All we need to complete the proof of the theorem is to show that $p(x)/q(x)$ generates the growth function f_G of G.

For $1 \le i \le n$, $1 \le j \le n$, let $f_{i,j}$ be the function generated by $p_{i,j}(x)/q(x)$, i.e. $p_{i,j}(x)/q(x) = \Sigma_{t=0}^{\infty} f_{i,j}(t)x^t$. (That such an $f_{i,j}$ exists and is unique follows from Lemma 4.9 (i).) For $t \ge 0$, let F_t be the $n \times n$ matrix whose typical entry is $f_{i,j}(t)$. Then we have that

$$I = (I - M_G x)(I - M_G x)^{-1}$$

$$= (I - M_G x)\left(\sum_{t=0}^{\infty} F_t x^t\right)$$

$$= \sum_{t=0}^{\infty} (F_t x^t) - \sum_{t=0}^{\infty} (M_G F_t x^{t+1})$$

$$= \sum_{t=0}^{\infty} (F_t x^t) - \sum_{t=1}^{\infty} (M_G F_{t-1} x^t).$$

Identifying coefficients of powers of x we get that $F_0 = I$, and, for $t \geq 1$, $F_t = M_G F_{t-1}$. From this it follows that, for $t \geq 0$, $F_t = M_G^t$. Hence

$$p(x)/q(x) = \bar{w}(I - M_G x)^{-1}\eta$$

$$= \bar{w}\left(\sum_{t=0}^{\infty} F_t x^t\right)\eta$$

$$= \bar{w}\left(\sum_{t=0}^{\infty} M_G^t x^t\right)\eta$$

$$= \sum_{t=0}^{\infty} (\bar{w} M_G^t \eta) x^t$$

$$= \sum_{t=0}^{\infty} f_G(t) x^t.$$

Thus, $p(x)/q(x)$ is the generating function of f_G. □

This theorem can certainly be considered as a solution to the analysis problem for DOL systems, since given a DOL system the algorithm provides us with a description of its growth function in the form of a rational generating function.

<u>EXAMPLE 4.12</u>. Consider the DOL system $G = \langle\{a,b,c\},\delta,a\rangle$, where $\delta(a) = abc^2$, $\delta(b) = bc^2$, $\delta(c) = c$.

$$\bar{a} = (1,0,0), \qquad M_G = \begin{pmatrix} 1 & 1 & 2 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{pmatrix}, \qquad \eta = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}.$$

Using Cramer's rule we see that

$$\bar{a}\,(I - M_G x)^{-1}\eta = \frac{\det\begin{pmatrix} 1 & -x & -2x \\ 1 & 1-x & -2x \\ 1 & 0 & 1-x \end{pmatrix}}{\det\begin{pmatrix} 1-x & -x & -2x \\ 0 & 1-x & -2x \\ 0 & 0 & 1-x \end{pmatrix}}$$

$$= \frac{1+x}{(1-x)^3} = 1 + 4x + 9x^2 + 16x^3 + \ldots .$$

II: *Growth classification.*

DEFINITION 4.13. The growth of a DxL system G is called:

(i) *Exponential* or *type* 3 if there exists a real number $x > 1$ such that $\overline{\lim}_{t\to\infty} f_G(t)/x^t > 0$.

(ii) *Subexponential* or *type* 2½ if the growth is not exponential and there does not exist a polynomial p such that $f_G(t) \le p(t)$ for all t.

(iii) *Polynomial* or *type* 2 if f_G is unbounded (i.e., $\overline{\lim}_{t\to\infty} f_G(t) > c$ for all constants c) and there exist polynomials p, q such that $p(t) \le f_G(t) \le q(t)$ for all t.

(iv) *Subpolynomial* or *type* 1½ if f_G is unbounded and for each unbounded polynomial p holds that $\overline{\lim}_{t\to\infty} f_G(t)/p(t) = 0$.

(v) *Limited* or *type* 1 if there exists an integer m such that $0 < f_G(t) < m$ for all t.

(vi) *Terminating* or *type* 0 if there exists an integer t_0 such that $f_G(t) = 0$ for all $t \ge t_0$.

As we remarked before, we can allow any valuation of η in $\mathbb{R}^n_+$. Then the growth function f_G is a total mapping from the nonnegative integers in the nonnegative reals defined by $f_G(t) = \bar{w}\, M_G^t \eta$, $t \ge 0$, where $\bar{w}$ and M_G are as before. The matrix valued analytic mapping $(I - xM_G)^{-1}$ can be represented by the power series.

$$(I - M_G x)^{-1} = \sum_{t=0}^{\infty} x^t M_G^t$$

with a positive radius of convergence. By Cramer's rule there are polynomials p(x) and q(x) such that for the generating function $F(x) = \sum_{t=0}^{\infty} f_G(t) x^t$

holds that

$$F(x) = \sum_{t=0}^{\infty} f_G(t)x^t = \sum_{t=0}^{\infty} \bar{w}\, M_G^t \,\eta\, x^t$$

$$= \bar{w}(I - M_G x)^{-1}\eta = p(x)/q(x)$$

where $p(x)$, $q(x)$ are in least terms and $q(0) = 1$. Similarly, the analytic expression in Theorem 4.5 goes through.

From the expressions in Theorems 4.5 and 4.11 it seems clear that the growth function of a DOL system can only be of types 3, 2, 1 or 0. Since according to Lemma 4.4 a DxL system has as fastest growth the growth type 3 and, moreover, the growth types in Definition 4.13 form a continuous spectrum, we only have to prove

THEOREM 4.14. *There are no* DOL *growth functions of type* $1\frac{1}{2}$ *or type* $2\frac{1}{2}$.

We prove Theorem 4.14 in the next Section 4.2 by combinatorial arguments.

It is apparent from Theorem 4.5 that the sizes of the characteristic values of M_G determine the growth type of G and, similarly, from Theorem 4.11 that the distribution of the poles of the generating function determines the growth type. Actually, of course, the characteristic values of M_G and the roots of the denominator polynomial of the generating function are related as follows. Let the characteristic polynomial of M_G be $p(x) = \det(M_G - Ix)$ and let the denominator polynomial of the generating function $F(x)$ be $q(x) = \det(I - M_G x)$. Then

$$p(x) = \det(M_G - Ix)$$

$$= -x^n \det(I - M_G \frac{1}{x})$$

$$= -x^n q(\frac{1}{x})$$

Hence the roots of the characteristic polynomial of M_G and the poles of the generating function F are inversely related. The following theorem of POLLUL and SCHÜTT [1975] makes this explicit.

THEOREM 4.15. *Let* f_G *be the growth function of a* DOL *system* G, $p(x)$ *and* $q(x)$ *be polynomials in least terms and* $q(0) = 1$ *such that* $p(x)/q(x) = \sum_{t=0}^{\infty} f_G(t)x^t$, *then:*

(i) f_G *is of type* 0 *iff* $q(x)$ *is a constant.*

(ii) f_G *is of type* 1 *or* 2 *iff* $q(x)$ *has a root and all roots of* $q(x)$ *have absolute value* ≥ 1.

(iii) f_G *is of type* 3 *iff there is a root of* $q(x)$ *of absolute value* <1.

This holds for any valuation of $\eta \in \mathbb{R}_+^n$.

PROOF. By the Cauchy-Hadamard formula, it holds for the radius R of convergence of the series $\sum_{t=0}^{\infty} f_G(t)x^t$ that:

$$1/R = \overline{\lim_{t\to\infty}} \sqrt[t]{f_G(t)} \qquad (\tfrac{1}{0} \overset{\text{def}}{=\!=} \infty,\ \tfrac{1}{\infty} \overset{\text{def}}{=\!=} 0)\ .$$

Since $p(x)$ and $q(x)$ are in least terms we have

$$R = \begin{cases} \min\{|x| \mid q(x) = 0\} & \text{if } q(x) \text{ has a root} \\ \infty & \text{if } q(x) = \text{constant} \end{cases}$$

Therefore the theorem is equivalent to

(i') type 0 iff $R = \infty$

(ii') type 1 or 2 iff $1 \leq R < \infty$

(iii') type 3 iff $R < 1$.

Clearly, f_G is terminating (type 0) iff $p(x)/q(x)$ is a polynomial, i.e., $R = \infty$. If f_G is limited (type 1) or polynomial (type 2) then there are $m, t_0 \in \mathbb{N}$ such that $f_G(t) \leq t^m$ for $t \geq t_0$ and therefore

$$1/R = \overline{\lim_{t\to\infty}} \sqrt[t]{f_G(t)} \leq \overline{\lim_{t\to\infty}} \sqrt[t]{t^m} = 1.$$

If f_G is exponential (type 3) then for some $x > 1$

$$1/R = \overline{\lim_{t\to\infty}} \sqrt[t]{f_G(t)} \geq \lim_{t\to\infty} \sqrt[t]{x^t} = x > 1. \qquad \square$$

Similar considerations connected with arbitrary choices of initial strings can be found in POLLUL and SCHÜTT [1975].

The growth type of a semi DOL system, i.e., what growth types occur if the initial string varies over all possible choices, will be studied in Section 4.2.

III. Synthesis. Here the problem concerns:

(i) Characterizing which functions can be DOL (PDOL) growth functions.

(ii) Given such a function, realize a DOL (PDOL) system of which it is the growth function.

Below we shall sketch briefly how this problem has been solved.

It was already shown by SZILARD [1971] that any positive, nondecreasing ultimately polynomial function is the growth function of a PDOL system. The proof of this fact provided an algorithm which for any such function produces the required PDOL system. The method uses many results in the nature of polynomial functions. On the way to proving the main theorem Szilard shows, for example, that if the generating functions $F(x)$ and $F'(x)$ generate growth functions of DOL systems, then so do $F(x) + F'(x)$, $1 + xF(x)$ and $F(x)/(1-x)$. His proofs were effective: given the DOL systems whose growth functions are generated by $F(x)$ and $F'(x)$, it was shown how one can obtain $F(x) + F'(x)$, $1 + xF(x)$ and $F(x)/(1-x)$. Thus, if we know how to obtain DOL systems whose growth functions are generated by basic generating functions, results such as this provide us with the ability to construct DOL systems whose growth functions are generated by more and more complicated generating functions put together from the basic ones by the operations described above.

SOITTALA [1976] gave a complete characterization and solved the DOL synthesis problem as follows. A sequence (r_n) is called $\mathbb{Z}$*-rational* ($\mathbb{N}$-*rational*) if it can be given a representation $r_n = PM^nQ$ where P is a row vector, M a square matrix, Q is a column vector and all the entries of P, M and Q are integers (respectively, natural numbers zero included). Hence we see that the sequence $(f_G(t))$, where f_G is a DOL growth function, is $\mathbb{N}$-rational. For the DOL synthesis problem one has to decide for a given $\mathbb{Z}$-rational sequence whether or not it is a DOL sequence, and in the affirmative case to construct a corresponding DOL system. By giving necessary and sufficient conditions for $\mathbb{Z}$-rational sequences to be $\mathbb{N}$-rational, DOL- or PDOL sequences the DOL synthesis problem was solved; in fact we are able to find a DOL system with minimal alphabet whose growth sequence coincides with a given DOL sequence.

Hence, let the $\mathbb{Z}$-rational sequence (r_n) be given in matrix representation or by a generating function. Then the algorithm to decide whether or not (r_n) is a D0L or PD0L growth sequence works as follows. By results of BERSTEL and MIGNOTTE [1975] and BERSTEL [1971], concerning conditions about poles of generating functions of $\mathbb{Z}$-rational sequences characterizing the $\mathbb{N}$-rational sequences, we can decide whether (r_n) is $\mathbb{N}$-rational. Having decided that (r_n) is $\mathbb{N}$-rational and nonterminating, we decide whether (r_n) is a D0L sequence primarily by again examining the poles of certain generating functions related with (r_n). Once we have decided whether a given $\mathbb{Z}$-rational sequence is a D0L sequence we can construct a D0L system with a minimal alphabet realizing it, by Soittala's method. Hence we have:

<u>THEOREM 4.16</u>. (SOITTALA). *The* D0L *synthesis problem is solvable.*

Using similar methods Soittala was able to prove that if $f_{G_1}(t)$ and $f_{G_2}(t)$ are D0L growth functions, $f_{G_2}(t) \neq 0$ for all t, and $f(t) = f_{G_1}(t)/f_{G_2}(t)$ is integer valued then $f(t)$ is a D0L growth function. The following interesting characterization of PD0L growth sequences was also given:

<u>THEOREM 4.17</u>. (SOITTALA). *An integer sequence* (r_n) *is a* PD0L *growth sequence iff* $r_0 > 0$ *and* $(s_n) = (r_{n+1} - r_n)$ *is* $\mathbb{N}$*-rational.*

The formal power series methods, hinted at above, have been applied to many questions concerning $\mathbb{Z}$-rational, $\mathbb{N}$-rational, D0L growth sequences and variations thereof, cf. SALOMAA [1976a,b] and SALOMAA and SOITTALA [1978]. By an appeal to the unsolvability of Hilbert's tenth problem several questions concerning D0L growth functions were shown to be undecidable too. These subjects are vigorously studied by a school of Finnish mathematicians (Karhumäki, Ruohonen, Salomaa and Soittala) and a school of French mathematicians (Berstel, Mignotte) but fall outside the scope of the present monograph.

IV. *Growth equivalence*. It is a consequence of Theorem 4.11 that the growth equivalence problem for D0L systems is solvable.

<u>THEOREM 4.18</u>. *Given two* D0L *systems* G_1, G_2 *we can decide whether or not* $f_{G_1} = f_{G_2}$.

PROOF. Let $p_i(x)/q_i(x)$ be the generating function of f_{G_i}, $i = 1,2$. Then $f_{G_1} = f_{G_2}$ iff $p_1(x)q_2(x) = p_2(x)q_1(x)$. □

V. *Classification*. This is in general more easily done by structural (combinatorial) means, as in Section 4.2, than by analytic means.

VI. *Hierarchy*. Clearly, the class of PD0L growth functions is strictly contained in the class of D0L growth functions. In Section 4.3 it will appear, e.g., that the D0L growth functions are a proper subset of the D1L growth functions.

4.2. D0L GROWTH FUNCTIONS: COMBINATORIAL APPROACH

In this section we will also assume familiarity with Sections 3.1.1 and 3.1.2. We are interested here in what combinatorial (structural) properties of the homomorphism of a (semi) D0L system cause differences in the overall growth of a D0L string sequence.

As we have seen in Section 3.1.2 it is easy to determine the condensed digraph CAD(S) of a semi D0L system $S = \langle W,\delta\rangle$ from which various conclusions concerning the growth behavior of individual letters in S can be drawn. Obviously, the growth function of a D0L system $G = \langle W,\delta,w\rangle$ equals the sum of the growth functions of $G_i = \langle W,\delta,a_i\rangle$ for $w = a_1a_2\ldots a_n$ and $1 \le i \le n$. That is, $f_G(t) = \sum_{i=1}^{n} f_{G_i}(t)$. Furthermore, f_G is of type 0 iff all letters in w are mortal; f_G is of type 1 iff the only recursive letters accessible from w are monorecursive. Since f_G is of type 0 iff $\delta^i(w) = \lambda$ for some i; and f_G is of type 1 iff $\delta^i(w) \neq \lambda$ for all i and $\delta^t(w) = \delta^{t'}(w)$ for some $t \neq t'$ the above follows from the definitions and Corollary 3.12. It was observed in Theorem 4.14 without proof that there are no D0L systems with growth of type 1½ or 2½. We will now give a proof of that statement by combinatorial means.

PROOF OF THEOREM 4.14.

By the above remarks, if f_G is not of type 0 or 1 there are recursive letters accessible from the initial string which are not monorecursive. Let $a \in W$ be such a letter, then there is a $p > 0$ such that $\delta^p(a) = v_1av_2$ and v_1v_2 contains at least one vital letter. Hence $\delta^{kp}(a) = \delta^{(k-1)p}(v_1)\delta^{(k-2)p}(v_1)\ldots$ $\ldots\delta^p(v_1)v_1av_2\delta^p(v_2)\ldots\delta^{(k-1)p}(v_2)$ and $\ell g(\delta^{kp}(a)) - \ell g(v_1av_2) > k$ for all k.

Therefore, if $\delta^u(w) = w_1 a w_2$ then $f_G(u+kp) \geq 1+k$ and hence $f_G(t) \geq \lfloor (t-u)/p \rfloor$ for $t > u \geq 0$ and $p > 0$. That is, f_G is of type 2, 2½ or 3. Hence the only thing left to prove is that there do not exist DOL systems of growth type 2½.

LEMMA 4.19. (SALOMAA [1973b]). $G = \langle W,\delta,w\rangle$ *is a* DOL *system with exponential growth iff* G *contains an expanding letter*

PROOF. Suppose G contains an expanding letter. Then $\delta^u(w) = w_1 a w_2$ and $\delta^p(a) = w_3 a w_4 a w_5$ for some u and p. Hence $f_G(t) \geq 2^{\lfloor (t-u)/p \rfloor^1}$ for all $t > u$: f_G is of growth type 3.

Suppose G contains no expanding letter ($E = \emptyset$). Then, according to Theorem 3.27 f_G is bounded above by a polynomial, and hence f_G is not of type 3 or 2½. □

Theorem 4.14 follows from the observation that all DOL growth which is faster than type 1 is of type 2, 2½ or 3 and the observation in the proof of Lemma 4.19 that all DOL growth which is slower than type 3 is of type 2, 1½, 1 or 0. END OF PROOF OF THEOREM 4.14.

We will now show what types of letters are responsible for the different growth types, and how to determine them.

LEMMA 4.20. *Let* $S = \langle W,\delta\rangle$ *be a semi* DOL *system and* $a \in W$. $a \in E$ *iff* $\delta^i(a) = W^*[a]W^*[a]W^*$ *for some* i. *(Where* $[a] = \{b \mid b \sim a\}$ *as in Section* 3.1.1.)

PROOF. "only if". Since there is a j_1 such that $\delta^{j_1}(a) \in W^*\{a\}W^*[a]W^* \cup W^*[a]W^*\{a\}W^*$ there is also a j_2 such that $\delta^{j_2}(a)$ contains 3 occurrences of letters from [a]. By the same argument there is a j_3 such that $\delta^{j_3}(a)$ contains (at least) $k+1$ occurrences of letters from [a] for $k = \#[a]$ and hence two occurrences of the same letter $b \in [a]$. Then there also exists a j_4 such that $\delta^{j_4}(a) \in W^*\{a\}W^*\{a\}W^*$ since each occurrence of b will derive an occurrence of a in a certain number of productions. The "if" part is trivial. □

By the above lemma we have an easy algorithm to determine for a letter $a \in W$ whether or not $a \in E$.

(i) Determine $R/\sim$. (E.g. with help of Lemma 3.4.)

(ii) Replace in the production rules all $b \notin [a]$ by λ

(iii) If there is a production rule $c \to v$ left with $\ell g(v) \geq 2$ then $[a] \subseteq E$ and $[a] \not\subseteq E$ otherwise.

The other types of letters R, M, V, MR are easily determined according to Corollary 3.6. To determine in general what type of growth a letter $a \in W$ induces under a homomorphism δ we proceed as follows.

(a) Determine M, R and MR.

(b) $RM \overset{def}{=} R - MR$ and $RME \overset{def}{=} \{a \mid a \in RM \,\&\, U(a) \cap E = \emptyset\}$.

From the foregoing it should be clear that $a \in W$ is of growth type 3 iff $U(a) \cap E \neq \emptyset$; of growth type 2 iff $U(a) \cap E = \emptyset$ and $U(a) \cap RM \neq \emptyset$; of growth type 1 iff $U(a) \cap (E \cup RME) = \emptyset$ and $U(a) \cap MR \neq \emptyset$; of growth type 0 iff $a \in M$ or, equivalently, if $U(a) \cap R = \emptyset$. We see that the growth type of a letter depends on the accessible recursive letters. The growth type of a DOL system is equal to the highest growth type among the letters constituting its initial string. When we look at what types of growth are possible for strings over W under a homomorphism δ we ask in effect for the growth type of a semi DOL system. The growth type of a semi DOL system $S = \langle W,\delta \rangle$ then is $\chi_3\chi_2\chi_1\chi_0$ where $\chi_i = i$ if $G = \langle W,\delta,a \rangle$ is of growth type i for some $a \in W$ and $\chi_i = \lambda$ otherwise.

Examples of semi DOL growth types.

type 321	$S_1 = \langle\{a,b,c\},\{a \to a^2b,\ b \to bc,\ c \to c\}\rangle$
type 31	$S_2 = \langle\{a,b\},\{a \to a^2b,\ b \to b\}\rangle$
type 3	$S_3 = \langle\{a,b\},\{a \to b,\ b \to ab\}\rangle$
type 21	$S_4 = \langle\{a,b\},\{a \to ab,\ b \to b\}\rangle$
type 1	$S_5 = \langle\{a,b\},\{a \to b,\ b \to b\}\rangle$
type 0	$S_6 = \langle\{d\},\{d \to \lambda\}\rangle$

We form the types 3210, 310, 30, 210 and 10 by adding d and $d \to \lambda$ to the alphabets and production rules of $S_1 - S_5$, respectively. The other possible combinations, i.e., 320, 32, 20 and 2 will be excluded below.

Since we saw that the growth type of a letter depends on the kinds of accessible recursive letters as indicated above, we have that a semi DOL S is of growth type $\chi_3(E)\chi_2(RME)\chi_1(MR)\chi_0(M)$ where $\chi_i(\cdot) = i$ if $\cdot \neq \emptyset$ and $\chi(\cdot) = \lambda$ otherwise. Hence:

THEOREM 4.21. *There is an algorithm to determine the growth type of a given semi* DOL *system,* DOL *system or letter under a homomorphism (on basis of structural properties alone).*

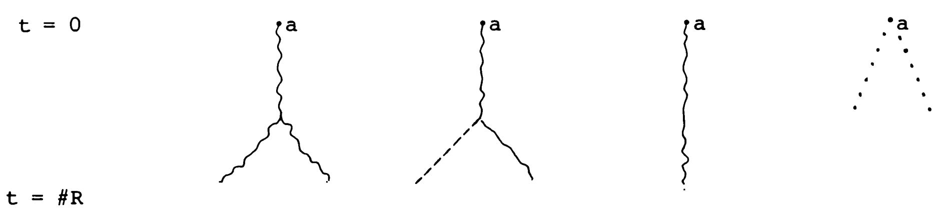
To see that growth types 320, 32, 20 and 2 are impossible for semi DOL systems we prove:

THEOREM 4.22. *If* $G = \langle W,\delta,a\rangle$ *and* $a \in$ RME *then there is a letter* $a' \in U(a)$ *which is monorecursive.*

PROOF. Suppose $a \in$ RME and there is no $a' \in U(a) \cap$ MR. There is a $j_1 \le \#R$ and a $b \in V$ such that $\delta^{j_1}(a) = v_1\, a\, v_2\, b\, v_3$ (or $v_1\, b\, v_2\, a\, v_3$). Since every vital letter produces a recursive letter within $\#(V - R)$ steps there is a $j_2 \le \#V$, a letter $c \in R$ and a letter $d \in [a]$ such that $\delta^{j_2}(a)$ contains c and d. Because of the assumption $c,d \in$ RME. By iteration of the argument we have that $\ell g(\delta^{k\#V}(a)) \ge 2^k$ for all k. But then $f_G(t) \ge 2^{\lfloor t/\#V \rfloor}$ which contradicts $a \in$ RME. □

COROLLARY 4.23. *If* RME $\neq \emptyset$ *then* MR $\neq \emptyset$ *and hence there do not exist semi* DOL *systems of growth types* 320, 32, 20 *and* 2.

A conceptually simple characterization of the necessary and sufficient conditions that determine the growth type of a letter can be obtained by depicting necessary and sufficient subtrees of the production trees (similar to the production trees of context free grammars) of letters of classes E, RME, MR and M.



$a \in$ E: leading to *exponential* growth.	$a \in$ RME: leading to *polynomial* growth.	$a \in$ MR: leading to *limited* growth.	$a \in$ M: leading to *terminal* growth.

Solid, broken and dotted lines represent chains of descendants b_i of a such that $b_i \in [a]$, $b_i \in V-[a]$ & $U(b_i) \cap E = \emptyset$, $b_i \in M$, respectively. From this characterization it is easy to derive expressions for the slowest growth possible in each of the discussed growth types.

Let $G_i = \langle W, \delta_i, a\rangle$, $i = 0,1,2,3$ with $W = \{a_1, a_2, \ldots, a_n\}$ and $a = a_1$.

$$\delta_3 = \{a_i \to a_{i+1} \mid 1 \le i < n\} \cup \{a_n \to a_1 a_1\} \text{ yields } f_{G_3}(t) = 2^{\lfloor t/n \rfloor};$$

$$\delta_2 = \{a_i \to a_{i+1} \mid 1 \le i < n-1\} \cup \{a_{n-1} \to a_1 a_n,\ a_n \to a_n\}$$
$$\text{yields } f_{G_2}(t) = \lfloor 1 + t/(n-1) \rfloor;$$

$$\delta_1 = \{a_i \to a_{i+1} \mid 1 \le i < n\} \cup \{a_n \to a_1\} \text{ yields } f_{G_1}(t) \equiv 1;$$

$$\delta_0 = \{a_i \to \lambda \mid 1 \le i \le n\} \text{ yields } f_{G_0}(0) = 1 \text{ and } f_{G_0}(t) = 0$$
$$\text{for all } t > 0;$$

which are, respectively, the slowest possible growths of types 3, 2, 1 and 0.

Returning to the associated digraphs of Section 3.1.2 we notice that if we have determined $RS(\langle W,\delta\rangle)$ we can see what the growth type of the semi DOL system is; and for each letter which grows polynomially the degree of polynomial growth can be determined according to Theorem 3.27 and the discussion preceding it.

4.3. GROWTH FUNCTIONS OF CONTEXT SENSITIVE L SYSTEMS

In the previous Sections 4.1 and 4.2 we have studied growth functions of DOL systems, and almost all questions posed have been solved affirmatively by algebraic or analytic means (Section 4.1) and some by combinatorial means (Section 4.2). Here we study the growth of strings in deterministic context sensitive L systems. By quite elementary techniques, viz. reduction to the halting problem, we show that it is recursively unsolvable to determine the growth type of a DIL system, or even of a PD1L system. (I.e., the growth type is undecidable in these cases). Furthermore, growth equivalence is undecidable for these systems and as a by product it is shown that the language equivalence for PD1L languages is undecidable and that a problem proposed by Varshavsky has a negative solution. Apart from these undecidability results, we derive bounds on the fastest and slowest growth in such systems; a method is given for obtaining growth functions of systems with smaller context from systems with a larger context; it is shown that all bounded growth functions of context sensitive L systems are within the realm of context free growth functions whereas for each type of unbounded context sensitive growth functions there are growth functions which are not; similarly, all growth functions of context sensitive L systems using a one letter alphabet are growth functions of context free L systems whereas this is not the case for growth functions of the simplest context sensitive L systems using a two letter alphabet; we give an application of the firing squad synchronization problem to growth functions etc. The section is divided in two parts. In Sections 4.3.1 - 4.3.3 we develop outlines for a theory of context sensitive growth functions and give some theorems and illuminating examples. In Section 4.3.4 we prove the undecidability of several problems in the area.

We start by giving an example of a semi PD1L system where growth type 2 occurs without being accompanied by growth type 1, which is impossible for DOL systems by Corollary 4.23.

EXAMPLE 4.24. Let $S = \langle\{a\},\{\delta(\lambda,a,\lambda) = a^2,\ \delta(\lambda,a,a) = a\}\rangle$ be a semi PD(0,1)L system. It is easily verified that for every initial string a^k, $k > 0$, S yields the growth function $f(t) = k + t$. (In each time step the letter on the right end of the string generates aa while the remaining letters generate a.) Therefore, even for PD1L systems using a one letter alphabet, growth type 2 can occur without growth type 1 and all combinations of growth types 0, 1, 2 and 3 are possible for semi PD1L systems.

In the previous section we have seen that the growth types 3, 2, 1, 0 exhaust all possibilities in the DOL case. However, as will appear in the sequel, this is not the case for DIL systems. There are also subexponential (type $2\frac{1}{2}$) and subpolynomial (type $1\frac{1}{2}$) DIL growth functions. Again, for DOL systems, the following problems have been solved effectively: (i) Analysis problem; (ii) Synthesis problem; (iii) Growth equivalence problem; (iv) Classification problem and (v) Structural problems. In Section 4.3.4 we show that for already PD1Ls the problems (i) - (v) are recursively unsolvable.

4.3.1. BOUNDS ON UNBOUNDED GROWTH

Since it is difficult to derive explicit formulas for growth functions of the more involved examples of DIL systems, and according to Section 4.3.4 impossible in general, we avail ourselves of the following notational devices.

- $\lfloor f(t) \rfloor$ is the *lower entier* of $f(t)$, i.e. for each t, $\lfloor f(t) \rfloor$ is the largest integer not greater than $f(t)$.
- $f(t) \sim g(t)$: $f(t)$ is *asymptotic* to $g(t)$, i.e. $\lim_{t\to\infty} f(t)/g(t) = 1$.
- $f(t) \gtrsim\!\!> g(t)$: $f(t)$ *slides onto* $g(t)$ (terminology provided by G. Rozenberg) if for each maximum argument interval $[t',t'']$ on which $g(t)$ has a constant value holds that there is a t''', $t' \leq t''' \leq t''$, such that for all t, $t''' \leq t \leq t''$, holds $f(t) = g(t)$.

As in the DOL case, for each DIL system $G = \langle W,\delta,w \rangle$ holds that $f_G(t) \leq \ell g(w) \cdot m^t$ where $m = \max\{\ell g(\delta(v_1,a,v_2)) \mid v_1,v_2 \in W^* \text{ and } a \in W\}$. Hence the fastest growth is exponential, and furthermore for each DIL system there is a DOL system which grows faster. We shall now investigate what is the slowest unbounded growth which can occur. Remember that a function f is unbounded if for each n_0 there is a t_0 such that $f(t) > n_0$ for $t > t_0$.

<u>THEOREM 4.25</u>.

(i) *For any* PDIL *system* $G = \langle W,\delta,w \rangle$ *such that* f_G *is unbounded holds:*

$$\lim_{t\to\infty} f_G(t)/\log_r t \geq 1, \qquad \text{where } r = \#W > 1.$$

(ii) *For any* DIL *system* $G = \langle W,\delta,w \rangle$ *such that* f_G *is unbounded holds:*

$$\lim_{t\to\infty} \sum_{i=0}^{t} f_G(t) / \sum_{i=0}^{t} \lfloor \log_r((r-1)i+r) \rfloor \geq 1, \qquad \text{where } r = \#W > 1.$$

<u>PROOF</u>.

(i) Order all strings in WW^* according to increasing length. The number of strings of length less than k is given by $t = \Sigma_{i=1}^{k-1} r^i$, $r = \#W$. Hence $t = \frac{r^k - r}{r-1}$ and therefore $k = \log_r((r-1)t+r)$. If we define $f(t)$ as the length of the t-th string in WW^* then, clearly, $f(t) = \lfloor \log_r((r-1)t+r) \rfloor$ and $\lim_{t\to\infty} f(t)/\log_r t = 1$. The most any PDIL system with an unbounded growth function can do is to generate all strings of WW^* in order of increasing length and without repetitions. Therefore $\lim_{t\to\infty} f_G(t)/\log_r t \geq 1$.

(ii) The most any DIL system with an unbounded growth function can do is to generate all strings of WW^* in some order and without repetitions. Therefore, $\lim_{t\to\infty} \Sigma_{i=0}^{t} f_G(t)/\Sigma_{i=0}^{t} f(i) \geq 1$. $\square$

In the sequel of this section we shall show that Theorem 4.25 is optimal.

<u>EXAMPLE 4.26</u>. Let $G_1 = \langle W,\delta,w\rangle$ be a PD(0,1)L system such that $W = \{0,1,2,\ldots\ldots,r-1,¢,s\}$ $(r > 1)$; $\delta(\lambda,¢,i) = ¢$ for $0 \leq i \leq r-1$, $\delta(\lambda,¢,s) = ¢0$, $\delta(\lambda,i,\lambda) = \delta(\lambda,i,s) = i+1$ for $0 \leq i < r-1$, $\delta(\lambda,s,\lambda) = 1$, $\delta(\lambda,s,0) = \delta(\lambda,s,1) = 0$, $\delta(\lambda,r-1,\lambda) = \delta(\lambda,r-1,s) = s$, $\delta(\lambda,i,j) = i$ for $0 \leq i, j \leq r-1$; $w = ¢0$.

The starting sequence is: $¢0, ¢1, \ldots, ¢r-1, ¢s, ¢01, \ldots, ¢0r-1, ¢0s, ¢11, \ldots,$ $¢\underbrace{r-1\ldots r-1}_{k\ \times}, ¢\underbrace{r-1\ldots r-1}_{k-1\ \times}s, ¢\underbrace{r-1\ldots r-1}_{k-2\ \times}s1, \ldots, ¢s00\ldots, ¢0000\ldots, \ldots$

Observe that G counts all strings over an alphabet of r letters. When an increment of the length k is due on the left side it needs k extra steps. Furthermore, there is an additional letter ¢ on the left. Therefore,

$$f_{G_1}(t) = \lfloor \log_r((r-1)t+r - \lfloor \log_r((r-1)t/r+1)\rfloor)\rfloor + 1$$

$$\rightsquigarrow \lfloor \log_r((r-1)t+r)\rfloor + 1.$$

Hence $f_{G_1}(t) \sim \log_r t$. Therefore, with a PD1L system using r+2 letters we can reach the slowest unbounded growth possible for a PD1L system using r letters.

Some variations of Example 4.26 are the following:

<u>EXAMPLE 4.27</u>. Let G_2 be a PD(0,1)L system defined as G_1 but with $\delta(\lambda,¢,s) = ¢1$. Then, essentially, G_2 counts on a number base r and

$$f_{G_2}(t) = 2, \qquad 0 \leq t < r$$

$$f_{G_2}(t) = \lfloor \log_r(t - \lfloor \log_r t/r\rfloor)\rfloor + 2$$

$$\rightsquigarrow \lfloor \log_r t\rfloor + 2, \qquad t \geq r.$$

<u>EXAMPLE 4.28</u>. Let $G_3 = \langle\{0,1,2,\ldots,r-1\} \times \{0,¢,s\},\delta_3,(0,¢)\rangle$ be such that the action is as in G_1 but with ¢ and s coded in the appropriate letters. Then,

$$f_{G_3}(t) = \lfloor \log_r((r-1)t+r - \lfloor \log_r((r-1)t/r+1)\rfloor)\rfloor$$

$$\Rightarrow \lfloor \log_r((r-1)t+r)\rfloor$$

EXAMPLE 4.29. Let G_4 be as G_2 with the modifications of G_3. Then

$$f_{G_4}(t) = 1, \qquad 0 \le t < r$$

$$f_{G_4}(t) = \lfloor \log_r(t-\lfloor \log_r t/r\rfloor)\rfloor + 1$$

$$\Leftrightarrow \lfloor \log_r t\rfloor + 1, \qquad t \ge r.$$

Examples 4.26 - 4.29 all corroborate the fact that for any PDIL system with an unbounded growth function there is a PD1L system with an unbounded growth function which grows slower, although not slower than logarithmic. That Theorem 4.25 (ii) cannot be improved upon follows from the following lemma, (see Section 3.2.1).

LEMMA 4.30. *For a suitable standard formulation of Turing machines, e.g. the quintuple version, holds that for any deterministic Turing machine* T *with symbol set* S *and state set* ψ *we can effectively construct a* D2L *system* $G_5 = \langle W_5, \delta_5, w_5\rangle$ *which simulates it in real time. I.e. the t-th instantaneous description of* T *is equal to* $\delta_5^t(w_5)$. *There is a required* G_5 *with* $W_5 = S \cup \psi$ *and a required propagating* G_5 *with* $W_5 = \psi \cup (S \times \psi)$.

Since T can expand its tape with at most one tape square per move we have that $f_{G_5}(t+1) \le f_{G_5}(t) + 1$, and $f_G(t) = \mathcal{O}(t)$.

As we have seen before, a Turing machine can compute every recursively enumerable set $A = \{1^{f(t)} \mid f(t)$ is a 1:1 total recursive function$\}$. We can do this in such a way that for each t, when f(t) has been computed, the Turing machine erases everything else on its tape. Subsequently, it recovers t from f(t) by f^{-1}, adds 1 and computes f(t+1). In particular, the simulating D2L system G_5 can, instead of replacing all symbols except the representation of f(t) by blank symbols, replace all the superfluous blank letters by the empty word λ. Suppose that A is nonrecursive. Then, clearly, it is not the case that for each n_0 we can find a t_0 such that $f_{G_5}(t) > n_0$ for all $t > t_0$, although such a t_0 exists for each n_0. Hence Theorem 4.25 (ii) is optimal for D2L systems, and as will appear from the next lemma also for D1L systems.

LEMMA 4.31.

(i) *Let* $G = \langle W,\delta,w\rangle$ *be any* D2L *system. We can effectively find a* D1L *system* $G' = \langle W',\delta',w'\rangle$ *with* $¢ \in W' - W$ *such that for all* t *holds:* $\delta'^{2t}(w') = ¢\delta^{t}(w)$.

(ii) *Let* $G = \langle W,\delta,w\rangle$ *be any* PD2L *system. We can effectively find a* PD1L *system* $G'' = \langle W'',\delta'',w''\rangle$ *with* $¢,\$ \in W' - W$ *such that for all* t *holds:* $\delta''^{2t}(w'') = ¢\delta^{t}(w)\$^{t}$.

PROOF.

(i) Cf. Lemma 3.39.

(ii) Let $G = \langle W,\delta,w\rangle$ be any PD2L system. Define a PD1L system $G'' = \langle W'',\delta'',w''\rangle$ as follows:

$$W'' = W \cup (W \times (W \cup \{\lambda\})) \cup \{¢,\$\}, \quad ¢,\$ \notin W;\ w'' = ¢w;$$

$$\delta''(\lambda,a,c) = (a,c), \qquad \delta''(\lambda,(a,b),(b,c)) = \delta(a,b,c),$$

$$\delta''(\lambda,a,\$) = (a,\lambda), \qquad \delta''(\lambda,¢,(a,c)) = ¢\delta(\lambda,a,c),$$

$$\delta''(\lambda,\$,\$) = \$, \qquad \delta''(\lambda,(a,\lambda),\lambda) = \delta''(\lambda,(a,\lambda),\$) = \$,$$

$$\delta''(\lambda,¢,d) = ¢, \qquad \delta''(\lambda,\$,\lambda) = \$,$$

for all $a,b \in W$, all $c \in W \cup \{\lambda\}$ and all $d \in W \cup \{\lambda,\$\}$. Analogous with the above we prove that if $\delta^{t}(w) \neq \lambda$ for all t then $\delta''^{2t}(w'') = ¢\delta^{t}(w)\$^{t}$. □

THEOREM 4.32.

(i) *If* $f(t)$ *is a* D2L *growth function then* $g(t) = f(\lfloor t/2\rfloor) + 1$ *is a* D1L *growth function.*

(ii) *If* $f(t)$ *is a* PD2L *growth function then* $g(t) = f(\lfloor t/2\rfloor) + \lfloor t/2\rfloor + 1$ *is a* PD1L *growth function.*

(iii) *If* $f(t)$ *is a* PD2L *growth function then* $g(t) = f(\lfloor t/2\rfloor)$ *is a* D1L *growth function.*

(iv) *If* $f(t)$ *is a* PD2L *growth function then* $g(t) = f(\lfloor t/2\rfloor) + \lfloor t/2\rfloor$ *is a* PD1L *growth function.*

PROOF. (i) and (ii) follow from Lemma 4.31 and its proof. (iii) and (iv) follow from Lemma 4.31 and its proof by the observation that we can encode the left end marker ¢ in the leftmost letter of a string and keep it there in the propagating case. □

Note that by Lemma 4.31 the transition in Theorem 4.32 is effective, i.e. given a D2L system G, of which f is the growth function, we can construct a required D1L system G' such that $f_{G'} = g$.

4.3.2. SYNTHESIS OF CONTEXT SENSITIVE GROWTH FUNCTIONS

In the last section we saw that if f(t) is the growth function of a D2L system G then $g(t) = f(\lfloor t/2 \rfloor) + 1$ is the growth function of a D1L system G' and there is a uniform method to construct G' given G. In this sense we shall treat some methods for obtaining growth functions. We consider operations under which families of growth functions are closed. An important tool here is an application of the *Firing Squad Synchronization problem,* cf. Section 3.2.5. Recall that for L systems it is the following. Let S = $\langle W_S, \delta_S \rangle$ be a semi PD2L system such that $\ell g(\delta_S(a,b,c)) = 1$ for all $b \in W_S$ and all $a,c \in W_S \cup \{\lambda\}$, and there is a letter m in W_S such that $\delta_S(m,m,\lambda) = \delta_S(m,m,m) = m$. The problem is to design an S satisfying the restrictions above such that $\delta^{k(n)}(m^n) = f^n$, $f \in W_S$, for all natural numbers n and a minimal function k of n, while $\delta^t(m^n) \in (W_S - \{f\})^n$ for all t, $0 \le t < k(n)$. BALZER [1967] proved that there is a minimal time solution $k(n) = 2n-2$. In the PD2L case we can achieve a solution in e.g. $k(n) = n-1$ by dropping the restriction $\delta_S(m,m,\lambda) = m$ and having both letters m on the ends of an initial string act like "soldiers receiving the firing command from a general" in the firing squad terminology. Assume that S = $\langle W_S, \delta_S \rangle$ is such a semi PD2L system simulating a firing squad with $k(n) = n-1$. Let G = $\langle W, \delta, w \rangle$ be any (P)D2L system. We define the (P)D2L system G' = $\langle W', \delta', w' \rangle$ as follows:

$$W' = W \times W_S;\; w' = (a_1,m)(a_2,m)\ldots(a_k,m) \quad \text{for } w = a_1a_2\ldots a_k,$$

$$\delta'((a,a'),(b,b'),(c,c')) = (b,b'') \quad \text{for } \delta_S(a',b',c') = b'' \text{ and } a'b'c' \ne fff,$$

$$\delta'((a,f),(b,f),(c,f)) = \begin{cases} (b_1,m)(b_2,m)\ldots(b_h,m) & \text{for } \delta(a,b,c) = b_1b_2\ldots b_h, \\ \lambda & \text{for } \delta(a,b,c) = \lambda \end{cases}$$

We easily see that if $\delta(v) = v'$ for $v,v' \in W^*$ then

$$\delta'^{\ell g(v)}((a_1,m)(a_2,m)\dots(a_k,m)) = (b_1,m)(b_2,m)\dots(b_\ell,m)$$

where $v = a_1a_2\dots a_k$ and $v' = b_1b_2\dots b_\ell$; and $\delta'^{\ell g(v)}((a_1,m)(a_2,m)\dots(a_k,m)) = \lambda$ for $v' = \lambda$. Therefore we have:

LEMMA 4.33. *Let* G *be any* (P)D2L *system. We can effectively find a* (P)D2L *system* G' *such that*

$$(1) \qquad f_{G'}(t) = \begin{cases} f_G(0) & \textit{for all } t \textit{ such that } 0 \le t < f_G(0), \\ f_G(\tau+1) & \textit{for all } t \textit{ such that } \sum_{i=0}^{\tau} f_G(i) \le t < \sum_{i=0}^{\tau+1} f_G(i). \end{cases}$$

Since we can simulate an arbitrary (but fixed) number of r firing squads in sequence plus a number j of production steps of G' for each production step of G, we can effectively find a (P)D2L system G' for each (P)D2L system G such that:

$$f_{G'}(t) = \begin{cases} f_G(0) & \text{for all } t \text{ such that } 0 \le t < r\, f_G(0) + j \\ f_G(\tau+1) & \text{for all } t \text{ such that } r \sum_{i=0}^{\tau} f_G(i) + (\tau+1)j \le t < \\ & \qquad r \sum_{i=0}^{\tau+1} f_G(i) + (\tau+2)j. \end{cases}$$

Let us call the operation to obtain a growth function $f_{G'}$ from f_G as defined in (1) *FSS*. Then $f_{G'} = FSS(f_G)$.

A cascade of r firing squads working inside each other, such that one production step of a (P)D2L system G is simulated if the outermost squad fires, gives us a (P)D2L system G' such that $f_{G'} = FSS^r(f_G)$, i.e.

$$(2) \qquad f_{G'}(t) = \begin{cases} f_G(0) & \text{for all } t \text{ such that } 0 \le t < f_G(0)^r, \\ f_G(\tau+1) & \text{for all } t \text{ such that } \sum_{i=0}^{\tau} f_G(i)^r \le t < \sum_{i=0}^{\tau+1} f_G(i)^r. \end{cases}$$

EXAMPLE 4.34. Suppose that f_G is exponential, say $f_G(t) = 2^t$. Then $FSS(f_G) = f$ where $f(t) = 2^{\tau+1}$ for $\Sigma_{i=0}^{\tau} 2^i \le t < \Sigma_{i=0}^{\tau+1} 2^i$. Hence $f(2^{\tau+1}-1) = 2^{\tau+1}$ and $f(t) = 2^{\lfloor \log_2 t \rfloor}$, i.e., $f(t) \in \Theta(t)$, and we can obtain analogous results for arbitrary exponential functions.*) (footnote following page).

EXAMPLE 4.35. Suppose that f_G is polynomial, e.g. $f_G(t) = p(t)$ where $p(t)$ is a polynomial of degree r. Then $FSS(f_G) = f$ where $f(\Sigma_{i=0}^{t} p(i)) = p(t+1)$. Since $\Sigma_{i=0}^{t} p(i) = q(t)$ where $q(t)$ is a polynomial of degree r+1 we have $f(t) \in \Theta(t^{r/r+1})$. By (2) we see that $FSS^j(f_G) = f$ where $f(t) \in \Theta(t^{r/r+j})$.

Hence we have:

THEOREM 4.36. *For each rational number* r, $0 < r \leq 1$, *we can effectively find a* PD2L *system* G *such that* $f_G(t) \in \Theta(t^r)$.

PROOF. Since $r = r'/r"$, such that $r"$, r' are natural numbers and $r" \geq r'$, and according to SZILARD [1971] we can, for every monotonic ultimately polynomial function g, find a PD0L system G' such that $f_{G'} = g$: by Example 4.35 we can find a PD2L system G such that $f_G(t) \in \Theta(t^{r'/r"})$. □

EXAMPLE 4.37. Let $f_G(t) = \lfloor \log_2 t \rfloor$. Then $FSS(f_G) = f$, where $f((t-1)2^{t+1}+4) = t+1$, i.e. $f(t) \in \Theta(\log t)$.

Hence we see that the relative slowing down gets less when the growth function is slower. By Theorem 4.32 everything we have obtained for D2L systems holds for D1L systems if we substitute $\lfloor t/2 \rfloor$ for t in the expression for the growth function and add 1. However, even for D1L systems we can achieve a greater slowing down. Let G be some D2L system. We can construct a D1L system G' which simulates G such that for each production step of G, G' does the following.

(a) G' counts all strings of length $f_G(t)$ over an r letter alphabet by the method of Example 4.26. When an increase of length is due on e.g. the left side,

(b) G' initializes a firing squad, making use of the simulation technique of Lemma 4.31. When the firing squad fires, G' simulates one production step of G and subsequently starts again at (a).

Hence, if $h(t) \leq f_G(t) \leq g(t)$ for a D2L system G and monotonic increasing functions h and g then we can effectively find a D1L system G' such that $f_{G'}(\Sigma_{i=0}^{t} r^{h(i)}) < g(t+1)$. For instance, if $f_G(t) = t$ then $f_{G'}(t) < \log_r t$, $t > 1$.

---

*) Recall that $f \in \Theta(g)$ asserts that f is of the same *order of magnitude* as g, i.e. $c_1 g(t) < f(t) < c_2 g(t)$ for all t and some constants c_1, c_2.

We can combine processes like the above to obtain stranger and stranger, slower and slower growth functions. Similar to the above application of the Firing Squad Synchronization problem we could apply the solution to the *French Flag problem* (see e.g. HERMAN and LIU [1973]).

The next theorem tells us under what operations the family of growth functions is closed. In particular, the subfamilies of (P)D2L, (P)D1L and (P)D0L growth functions are closed under (i) - (iii).

THEOREM 4.38. *Growth functions are closed under* (i) *addition,* (ii) *multiplication with a natural number* $r > 0$*,* (iii) *entier division of the argument by a natural number* $r > 0$*,* (iv) FSS. *Growth functions are not closed under* (v) *subtraction,* (vi) *division,* (vii) *composition.*

PROOF.

(i) Let $G_1 = \langle W_1, \delta_1, w_1 \rangle$ and $G_2 = \langle W_2, \delta_2, w_2 \rangle$ be two D1L systems with disjoint alphabets. Define $G_3 = \langle W_1 \cup W_2, \delta_3, w_1 w_2 \rangle$. Then it is easy to construct δ_3, given δ_1 and δ_2, such that $f_{G_3} = f_{G_1} + f_{G_2}$.

(ii) Follows from (i).

(iii) Let $G_1 = \langle W_1, \delta_1, w_1 \rangle$ be a D1L system. Define $G_2 = \langle W_2, \delta_2, w_2 \rangle$ such that $f_{G_2}(t) = f_{G_1}(\lfloor t/r \rfloor)$. This is easily achieved by introducing a cycle of length r for each direct production of G_1.

(iv) By Lemma 4.33.

(v)-(vi) Trivial.

(vii) 2^t is a growth function while $2^{(2^t)}$ is not. □

We conclude this section with some conjectures. The evidence in favor of in particular Conjecture 1 is overwhelming, but we have not been able to derive a formal proof.

Conjecture 1. Growth functions are not closed under multiplication. (E.g. $2^{t+\lfloor \log_2 t \rfloor}$ can hardly be a growth function.)

Conjecture 2. Unbounded growth functions are closed under function inverse. (E.g. if $f(t) = r^t$ is a growth function for a constant r then $g(t) \sim f^{-1}(t) = \log_r t$ is a growth function too.)

Conjecture 3. There are no PD1L growth functions $f(t) \in \Theta(t^r)$ where r is not a natural number. (It is hard to see how a string can determine its own length in the PD1L case.)

4.3.3. THE HIERARCHY

The first PD1L growth function of growth type 1½ was *Gabor's Sloth* in HERMAN and ROZENBERG [1975, p.338]. The examples in Sections 4.3.1 and 4.3.2 provide us with an ample supply of this growth type. A more difficult problem is to construct a DIL system of growth type 2½. The first (and presumably up till now only) DIL system of this growth type is the PD2L system of KARHUMÄKI [1974a,b] with a growth function f such that $2^{\sqrt{t}} \leq f(t) \leq (2^{\sqrt{3}})^{\sqrt{t}}$. We give a brief outline how this PD2L system works.

At certain intervals in the string sequence the string has the form $(ga^k)^m g$. Thus consider the word $ga^k g$. The letter g is called a node. These nodes always send messengers b and $\bar{b}$ to the right and left, respectively. At the same time g changes to an inactive form $\bar{g}$ (which does not send any messengers). While moving on, messengers b and $\bar{b}$ duplicate every letter. When b and $\bar{b}$ meet, they create a new node which is in inactive form. Furthermore, b and $\bar{b}$ disappear and new messengers f and $\bar{f}$ are born. They travel to the right and left, respectively. At the beginning, g sends to the right and left also another messenger. These messengers travel at half the speed of the other messengers. When such a messenger and $\bar{f}$ meet, this slow messenger changes to the messenger f. Now we have three messengers travelling on. Moreover, these are synchronized in the sense that they reach each an inactive node simultaneously. When this happens, they disappear and transform the nodes to the active form g. During this process we increase the number of a's between letters g by one. So the word $ga^k g$ has changed to the form $ga^{k+1}ga^{k+1}g$, i.e., it has essentially duplicated. Note that the time in which the string duplicates its length increases linearly. This process yields the above growth function of type 2½. With regards to growth type 1½ the reader might have noticed that we have realized PD1L growth functions which grow logarithmically or as a fractional power. KARHUMÄKI [1974a] showed how to realize PD1L growth functions asymptotically equal to the function $(\log_p t)^r$, for natural numbers p and r, which lie between logarithmic functions and fractional powers. From the above examples (and Theorems 4.32, 4.38) it follows:

THEOREM 4.39. *There are* PD1L *growth functions of types* 1½, 2, 2½, 3 *which are not* DOL *growth functions.*

Hence the family of (P)DOL growth functions is properly contained in in the family of (P)D1L growth functions. However, if we restrict ourselves to the class of bounded growth functions the situation is different.

<u>THEOREM 4.40</u>. *Let* G *be any* DIL *system such that* f_G *is of* (i) *growth type* 0, *or,* (ii) *growth type* 1. *Then we can construct a* DOL *system* G' *such that* $f_{G'} = f_G$.

<u>PROOF</u>.

(i) Let $f_G(t) > 0$ for all $t \le t_0$ for some t_0 and $f_G(t) = 0$ otherwise. Then $f_{G'} = f_G$ where $G' = \langle W', \delta', w' \rangle$ is a DOL system constructed as follows:

$$W' = \{a_0, a_1, \ldots, a_{t_0}, b\};\ w' = a_0 b^{f_G(0)-1};$$

$$\delta'(a_i) = a_{i+1} b^{f_G(i+1)-1} \qquad \text{for all } i,\ 0 \le i < t_0,$$

$$\delta'(b) = \delta'(a_{t_0}) = \lambda.$$

(ii) If f_G is of growth type 1 for some DIL system G then f_G is ultimately periodic, i.e. $f_G(t) = f_G(t-p)$ for all $t > t_0+p$ for some t_0 and p. The construction of the appropriate DOL system G' is similar to the construction in (i). □

<u>COROLLARY 4.41</u>. *The family of bounded* (P)DIL *growth functions coincides with the family of bounded* (P)DOL *growth functions.*

<u>THEOREM 4.42</u>. *Let* $G = \langle W, \delta, w \rangle$ *be a unary (i.e.* $\#W = 1$*)* DIL *system. Then there is a* DOL *system* G' *such that* $f_{G'} = f_G$.

<u>PROOF</u>. Suppose f_G is bounded. By Theorem 4.40 the theorem holds. Suppose f_G is unbounded, and let G be a D(m,n)L system. Furthermore, let $p = \ell g(\delta(a^m, a, a^n))$, $x = \Sigma_{i=0}^{m-1} \ell g(\delta(a^i, a, a^n)) + \Sigma_{j=0}^{n-1} \ell g(\delta(a^m, a, a^j))$. Since f_G is unbounded there is a t_0 such that $f_G(t_0) \ge 2(m+n) + x + 1$. For all $t \ge t_0$ the following equation holds:

(3) $$f_G(t+1) = p(f_G(t) - m - n) + x.$$

Case 1. $p = 0$. Then $f_G(t) \le (m+n)y$ where $y = \max\{\ell g(\delta(v_1, a, v_2)) \mid v_1, v_2 \in W^*\}$. Therefore f_G is bounded: contradiction.

Case 2. $p = 1$. Then $x - m - n > 0$ since f_G is bounded otherwise. It is easy to construct a D0L system G' such that $f_{G'} = f_G$ in this case.

Case 3. $p > 1$. Construct a D0L system $G'' = \langle W'', \delta'', w'' \rangle$ as follows:

$$W'' = \{a_0, a_1, a_2, a_3\};\ \{\delta''(a_0) = \lambda,\ \delta''(a_1) = a_0 a_1 a_3^{p-2},$$

$$\delta''(a_2) = a_2 a_3^{x+p-1},\ \delta''(a_3) = a_3^{p}\};$$

$$w'' = (a_0 a_1)^{m+n} a_2 a_3^{f_G(t_0)-2(m+n)-1}.$$

It is easy to prove by induction on t that $f_{G''}(t) = f_G(t+t_0)$ for all t. By using Theorem 4.40 we construct a D0L system $G' = \langle W', \delta', w' \rangle$ such that $W'' \subseteq W'$, $\delta'' \subseteq \delta'$, $\delta'^{t_0}(w') = w''$ and $f_{G'}(t) = f_G(t)$ for $0 \le t < t_0$. Then $f_{G'} = f_G$. □

It may be worthwhile to note that the solution to the difference equation (3) is given by:

$$f_G(t) = \begin{cases} f_G(t_0) + (x-m-n)(t-t_0) & \text{for } p = 1, \\ p^{t-t_0} f_G(t_0) + (x - p(m+n)) \dfrac{1-p^{t-t_0}}{1-p} & \text{for } p > 1, \end{cases}$$

for all $t > t_0$.

Therefore, the growth function of a unary DIL system is either linear or purely exponential, which by Theorem 4.5 gives us:

<u>COROLLARY 4.43</u>. *The family of growth functions of unary* DIL *systems is properly contained in the family of growth functions of* D0L *systems.*

<u>THEOREM 4.44</u>. *There is a binary* PD1L *system* $G = \langle W, \delta, w \rangle$, *(i.e.* $\#W = 2$*), with a one letter initial string such that there is no* D0L *system* G' *such that* $f_{G'} = f_G$.

<u>PROOF</u>. Let $G = \langle W, \delta, w \rangle$ be a PD(1,0)L system where

$$W = \{a,b\};\ w = a;\ \{\delta(\lambda,a,\lambda) = b,\ \delta(\lambda,b,\lambda) = aa,\ \delta(a,a,\lambda) = a$$

$$\delta(b,a,\lambda) = b,\ \delta(b,b,\lambda) = b,\ \delta(a,b,\lambda) = aa\}.$$

The initial sequence of produced strings is:

$$a,\ b,\ aa,\ ba,\ aab,\ baaa,\ aabaa,\ ba^3ba,\ a^2ba^4b,\ ba^3ba^5,$$
$$a^2ba^4ba^4,\ ba^3ba^5ba^3,\ a^2ba^4ba^6ba^2,\ ba^3ba^5ba^7ba,\ a^2ba^4ba^6ba^8b,$$
$$ba^3ba^5ba^7ba^9, \ldots .$$

Every second time step one b is introduced on the left and starts moving along the string to the right. Every time step b moves one place to the right and leaves a string a^2 on the place it formerly occupied. When a letter b reaches the right end of the string it disappears in the next step leaving aa. Therefore, on the one hand, every second production step there enters a length increasing element in the string; on the other hand, with exponentially increasing time intervals one of these elements disappears. The strings where a b has just disappeared in the above sequence are:

$$\delta^5(a) = baaa,\ \delta^9(a) = ba^3ba^5,\ \delta^{15}(a) = ba^3ba^5ba^7ba^9.$$

Now introduce the notational convenience $\Pi_{i=1}^{x}\ v(i)$ where $v(i)$ is a function from $\mathbb{N}$ into W^*. E.g. if $v(i) = a^ib^{2i}$ then $\Pi_{i=1}^{3}\ v(i) = ab^2a^2b^4a^3b^6$.

<u>CLAIM</u>. $\delta^{t(x)}(a) = \prod_{i=1}^{2^x} ba^{2i+1}$ where $t(x) = 2^{x+1} + 2x + 3$.

<u>Proof of claim</u>. By induction on x.

$x = 0$. $\delta^5(a) = ba^3$.

$x > 0$. Suppose the claim is true for all $x \le n$. Then

$$\delta^{t(n)}(a) = \prod_{i=1}^{2^n} ba^{2i+1} = \ldots ba^{2.2^n+1}.$$

This last occurrence of b will just have disappeared at time $t' = t(n) + 2.2^n + 2 = t(n+1)$. The distance with the preceding occurrence of b was 2.2^n-1 and therefore

$$\text{(4)} \qquad \delta^{t(n+1)}(a) = \ldots ba^{2.2^n-1+2(2.2^n+2)-2(2^n+1)} = \ldots ba^{2.2^{n+1}+1}.$$

At time t(n) the total number of occurrences of b in the string was 2^n; at time t(n+1) this is $2^n + 2^n + 1 - 1 = 2^{n+1}$ and

(5) $$\delta^{t(n+1)}(a) = ba^3b\ldots\ .$$

Since it is easy to see that for all $t \geq 0$ holds: if $\delta^t(a) = v_1ba^{i_1}ba^{i_2}bv_2$ for some v_1, v_2 then $i_2 = i_1 + 2$, it follows from (4) and (5) that $\delta^{t(n+1)}(a) = \Pi_{i=1}^{2n+1}\ ba^{2i+1}$, which proves the claim. *End of proof of Claim.*

Hence,

$$f_G(t(x)) = \sum_{i=1}^{2^x} 2(i+1) = 2^x(2^x + 3) = 1/4(t(x) - 2x - 3)(t(x)-2x+3)$$

$$= 1/4\ t(x)^2 - x\ t(x) + x^2 - 9/4.$$

Since $t(x) = 2^{x+1} + 2x + 3$ we have $x \approx \lfloor \log_2 t(x)/2 \rfloor$ and therefore:

(6) $$f_G(t(x)) \approx 1/4\ t(x)^2 - \lfloor \log_2 t(x)/2 \rfloor t(x) + \lfloor \log_2 t(x)/2 \rfloor^2 - 9/4.$$

From (6) and the general formula for a DOL growth function in Theorem 4.5 it follows that f_G cannot be a DOL growth frunction since

$$f_G(t) - 1/4\ t^2 \sim t \log t. \qquad \square$$

That context dependent L systems using a two letter alphabet cannot yield all DOL growth functions is ascertained by the counterexample $f(0) = f(1) = f(2) = 1$ and $f(t) = t$ for $t > 2$, which is surely a (P)DOL growth function.

COROLLARY 4.45. *The family of binary* (P)D1L *growth functions has a nonempty intersection with the family of* (P)DOL *growth functions and neither contains the other.*

An open problem in this area is: does the family of (P)D1L growth functions coincide with the family of (P)D2L growth functions. A proof of Conjecture 3 would show that the family of PD1L growth functions is properly contained in the family of PD2L growth functions. Using a similar technique as in Lemma 4.31 we can, however, say the following.

THEOREM 4.46.

(i) *If* f(t) *is a* PD2L *growth function then* f(t) *is a* D(2,0)L *growth function.*

(ii) *If* $f(t)$ *is a* D2L *growth function then* $f(t)+1$ *is a* D(2,0)L *growth function.*

PROOF.

(i) Let $G = \langle W,\delta,w\rangle$ be a PD2L system. Define a D(2,0)L system $G' = \langle W',\delta', w'\rangle$ as follows.

$$W' = W \cup W \times \{\not{c}\} \text{ where } \not{c} \notin W;\ w' = a_1 a_2 \ldots a_{n-1}(a_n,\not{c}) \text{ for}$$
$$w = a_1 a_2 \ldots a_n\ ;$$

$$\delta'(ab,c,\lambda) = \delta(a,b,c), \qquad \delta'(\lambda,c,\lambda) = \lambda,$$

$$\delta'(ab,(c,\not{c}),\lambda) = \delta(a,b,c)a_1 a_2 \ldots a_{m-1}(a_m,\not{c})$$
$$\text{if } \delta(b,c,\lambda) = a_1 a_2 \ldots a_m,$$

$$\delta'(\lambda,(c,\not{c}),\lambda) = a_1 a_2 \ldots a_{m-1}(a_m,\not{c})$$
$$\text{if } \delta(\lambda,c,\lambda) = a_1 a_2 \ldots a_m,$$

for all $b,c \in W$ and all $a \in W \cup \{\lambda\}$. Then $\delta'^t(w') = b_1 b_2 \ldots b_{m-1}(b_m,\not{c})$ if $\delta^t(w) = b_1 b_2 \ldots b_m$, and therefore $f_{G'} = f_G$.

(ii) Let $G = \langle W,\delta,w\rangle$ be a D2L system. Define a D(2,0)L system $G' = \langle W',\delta', w'\rangle$ as follows.

$$W' = W \cup \{\not{c}\} \text{ where } \not{c} \notin W;\ w' = w\not{c};$$

$$\delta'(ab,c,\lambda) = \delta(a,b,c),\ \delta'(\lambda,c,\lambda) = \lambda,$$

$$\delta'(ab,\not{c},\lambda) = \delta(a,b,\lambda)\not{c},\ \delta'(\lambda,\not{c},\lambda) = \not{c},$$

for all $b,c \in W$ and all $a \in W \cup \{\lambda\}$. Then $\delta'^t(w') = \delta^t(w)\not{c}$ and therefore $f_{G'}(t) = f_G(t)+1$. □

ROZENBERG [1972a] proved that a D(m,n)L system can be simulated in real time by a D(k,ℓ) system if $k+\ell = m+n$ and $k,\ell,m,n > 0$. Therefore, by using the same trick as above we have the following:

COROLLARY 4.47.

(i) *If* $f(t)$ *is a* PD(m,n)L *growth function then* $f(t)$ *is a* D(k,ℓ) *growth function where* $k+\ell = m+n$.

(ii) *If* f *is a* D(m,n)L *growth function then* $f(t) + 1$ *is a* D(k,ℓ) *growth function where* $k + \ell = m + n$.

In particular, (i) *and* (ii) *hold for* $k = m + n$ *and* $\ell = 0$ *and conversely.*

4.3.4. DECISION PROBLEMS

According to Sections 4.1, and 4.2 and the beginning of Section 4.3 (and the references contained therein) the analysis, synthesis, growth equivalence, classification and structural problems all have affirmative solutions for context free growth, i.e., there is an algorithm which gives the required answer or decides the issue for these cases. Here it is shown that for DIL systems, these problems all have a negative solution, essentially because already PD1L systems can simulate any effective process. (Note that by Theorems 4.40 and 4.42 the above problems have a positive solution if we restrict ourselves to unary DIL systems or DIL systems with a bounded growth function). Furthermore, it will appear that similar questions concerning growth ranges of DIL systems have similar answers.

First we need the notion of a *Tag system.* A Tag system is a 4 tuple $T = \langle W,\delta,w,\beta\rangle$ where W is a finite nonempty *alphabet,* δ is a total mapping from W into $W^{\star}$, $w \in WW^{\star}$ is the *initial string,* and β is a positive integer called the *deletion number.* The operation of a Tag system is inductively defined as follows: the initial string w is generated by T in 0 steps. If $w_t = a_1a_2\ldots a_n$ is the t-th string generated by T then $w_{t+1} = a_{\beta+1}a_{\beta+2}\ldots$ $\ldots a_n\delta(a_1)$ is the (t+1)-th string generated by T. Cf. MINSKY [1967].

LEMMA 4.48. (MINSKY). *It is undecidable for an arbitrary Tag system with* $\beta = 2$ *and a given positive integer* k *whether* T *derives a string of length less than or equal to* k. *In particular it is undecidable whether* T *derives the empty word.*

We shall now show that if it is decidable whether or not an arbitrary PD1L system has a growth function of growth type 1 then it is decidable whether or not an arbitrary Tag system with deletion number 2 derives the empty word λ. Therefore, by Lemma 4.48 it is undecidable whether a PD1L system has a growth function of type 1.

Let $T = \langle W_T,\delta_T,w_T,2\rangle$ be any Tag system with deletion number 2. Define a PD(1,0)L system $G = \langle W,\delta,w\rangle$ as follows:

$$W = W_T \cup W_T' \cup W_T \times W_T \cup \{\not{c},\$\},$$

where

$$W_T' = \{\underset{\approx}{a} \mid a \in W_T\},\ W_T' \cap W_T = \emptyset \text{ and } \not{c},\$ \notin W_T \cup W_T';$$

$$w = w_T\not{c};$$

$$\delta(\lambda,a,\lambda) = \delta(\$,a,\lambda) = \delta(\$,(a,b),\lambda) = \underset{\approx}{a},$$

$$\delta(\lambda,\underset{\approx}{a},\lambda) = \delta(\$,\underset{\approx}{a},\lambda) = \delta(\lambda,\$,\lambda) = \delta(\$,\$,\lambda) = \delta(\$,\not{c},\lambda)$$
$$= \delta(\lambda,\not{c},\lambda) = \$,$$

$$\delta(a,b,\lambda) = \delta(\underset{\approx}{a},(b,c),\lambda) = \delta(a,(b,c),\lambda) = b,$$

$$\delta(a,\not{c},\lambda) = \not{c},$$

$$\delta(\underset{\approx}{b},c,\lambda) = \delta((a,b),c,\lambda) = (c,b),$$

$$\delta(\underset{\approx}{b},\not{c},\lambda) = \delta((a,b),\not{c},\lambda) = \delta_T(b)\not{c},$$

for all $a,b,c \in W_T$ and all $\underset{\approx}{a},\underset{\approx}{b} \in W_T'$.

A sample derivation is:

T	G
$a_1a_2a_3a_4a_5$	$a_1a_2a_3a_4a_5\not{c}$
$a_3a_4a_5\delta_T(a_1)$	$\underset{\approx}{a}_1a_2a_3a_4a_5\not{c}$
$a_5\delta_T(a_1)\delta_T(a_3)$, etc.	$\$(a_2,a_1)a_3a_4a_5\not{c}$
	$\$\underset{\approx}{a}_2(a_3,a_1)a_4a_5\not{c}$
	$\$\$a_3(a_4,a_1)a_5\not{c}$
	$\$\$\underset{\approx}{a}_3a_4(a_5,a_1)\not{c}$
	$\$\$\$(a_4,a_3)a_5\delta_T(a_1)\not{c}$
	$\$\$\$\underset{\approx}{a}_4(a_5,a_3)\delta_T(a_1)\not{c}$, etc.

In the simulating PD1L system G signals depart from the left, with distances of one letter in between, and travel to the right at an equal speed of one letter per time step. Therefore, the signals cannot clutter up. It is clear that if the Tag system T derives the empty word, then there is a time t_0 such that $\delta^{t_0}(w) = \$^k\not{c}$ and $\delta^t(w) = \$^{k+1}$ for some k and for all $t > t_0$. Conversely, the only way for G to be of growth type 1 is to generate a string of the form $\$^k ¢$. (If the string always contains letters other than \$ and ¢ then at each second production step there appears a new occurrence of \$ and the string grows indefinitely long.) Therefore, T derives the empty word iff G is of growth type 1. Since it is undecidable whether or not an arbitrary Tag system with deletion nember 2 derives the empty word it is undecidable whether or not a PD1L is of growth type 1.

THEOREM 4.49.

(i) *It is undecidable whether or not an arbitray* PD1L *system is of growth type* i, $i \in \{1,1\frac{1}{2},2,2\frac{1}{2},3\}$.

(ii) *It is undecidable whether or not an arbitrary* D1L *system is of growth type* i, $i \in \{0,1,1\frac{1}{2},2,2\frac{1}{2},3\}$.

(iii) *It is undecidable whether an arbitrary* PD1L *system has an unbounded growth function.*

PROOF. (i). Let $G_1 = \langle W_1,\delta_1,w_1\rangle$ be a PD(1,0)L system simulating a Tag system T as discussed above. Let $G_2 = \langle W_2,\delta_2,w_2\rangle$ be a PD(1,0)L system of growth type i, $i \in \{1,1\frac{1}{2},2,2\frac{1}{2},3\}$ such that $W_2 \cap W_1 = \emptyset$. Define $G_3 = \langle W_3,\delta_3,w_3\rangle$ as follows:

$$W_3 = W_2 \cup \{\$\};\ w_3 = w_2;$$

$$\delta_3 = \delta_2 \cup \{\delta_3(\$,\$,\lambda) = \delta_3(\lambda,\$,\lambda) = \$\}$$
$$\cup \{\delta_3(\$,a,\lambda) = \delta_2(\lambda,a,\lambda) \mid a \in W_2\}.$$

Clearly, $f_{G3} = f_{G2}$. Now construct a PD(1,0)L system $G_4 = \langle W_4,\delta_4,w_4\rangle$ as follows:

$$W_4 = W_3 \cup W_1;\ w_4 = w_1;$$

$$\delta_4 = \delta_3 \cup (\delta_1 - \{\delta_1(\$,¢,\lambda) = \$\}) \cup \{\delta_4(\$,¢,\lambda) = w_3\}.$$

If there is a time t_0 such that $\delta_1^{t_0}(w_1) = \$^k ¢$ for some k then $\delta_4^{t_0}(w_4) = \$^k ¢$ and $\delta_4^{t+t_0+1}(w_4) = \$^k \delta_3^t(w_3)$ for all t, i.e., $f_{G4}(t+t_0+1) = f_{G2}(t)+k$. If there is no such time t_0 then $f_{G4}(t) = f_{G1}(t)$ for all t. In this latter case it is easy to see that $f_{G1}(t) \in \Theta(t)$, that is, G_4 is of growth type 2.

By the previous discussion it is undecidable whether such a time t_0 exists and therefore whether f_{G_4} is of growth type 2 or i, $i \in \{1,1\frac{1}{2},2\frac{1}{2},3\}$.
(ii). Follows by a similar argument if we talk about D(1,0)L systems instead of PD(1,0)L systems. (Hint: change everywhere $\delta.(\lambda,\$,\lambda) = \$$ into $\delta.(\lambda,\$,\lambda) = \lambda$ and let i range over $\{1,1\frac{1}{2},2,2\frac{1}{2},3\}$. Then the analogon of G_4 here is of growth type 0 or i, $i \in \{1,1\frac{1}{2},2,2\frac{1}{2},3\}$ depending on whether or not time t_0 as described above exists.)
(iii) Follows from (i). □

In the DOL case we have seen that we can effectively express the growth function by a difference equation, as a generalized exponential polynomial or by a generating function. These expressions of the growth function have in common that we can, e.g., ascertain the speed of growth from them, check equality etc. Although for deterministic context sensitive L systems the system itself is already a formalism for expressing the growth function, it follows from the above that we can not derive a useful expression for a DIL growth function in general; useful in the sense that it can help us decide properties like in the DOL case, such as an analytic expression. Hence we cannot hope to express DIL growth functions by analytic or other "*useful*" means and we have:

COROLLARY 4.50. *There is no algorithm which, for an arbitrary* PDIL *system* G, *gives an explicit expression for* f_G *in analytic form, by a generating function (or in any other form which is useful).*

The undecidability of whether a (P)D1L system is of a certain growth type holds (because of the proof method) also for further refinements in the growth type classification. We could have proved Theorem 4.49 by simulating Turing machines by PD1L systems (cf. Lemmas 4.30 and 4.31) and thereby reduce everything to the printing problem for Turing machines which is known to be undecidable too, MINSKY [1967]. This, however, would have caused some difficulties with the slow growth types.

Theorem 4.49 has some interesting corollaries. Recall that two DIL systems G_1, G_2 are said to be *language equivalent* if $L(G_1) = L(G_2)$. Now it is known that the language equivalence problem for e.g. 0L systems is undecidable (cf. HERMAN and ROZENBERG [1975]) but that the language equivalence problem for DOL systems is decidable, ČULIK and FRIS [1977a,b]. By the special tractable nature of PD1L systems it might well be that the language equivalence problem is decidable in this case. However, in the proof of Theorem 4.49 (i) it is clearly undecidable whether $L(G_4) = L(G_1)$. Therefore we have:

COROLLARY 4.51. *The language equivalence problem for* PD1L *systems is undecidable. (according to Theorem* 4.53 *this even is the case when we already know that both* PD1L *systems concerned are of the same growth type* i*,* $i \in \{2,2\frac{1}{2},3\}$.)

In the PROCEEDINGS of an Open House in Unusual Automata Theory [1972, p. 20] V.I. Varshavsky proposed the following problem: "Consider the class of D2L grammars producing strings which stabilize at a certain length. Make some reasonable assumptions about the maximal production length (e.g. 2) and axiom length (e.g. 1) and find the maximal stable string length as a function of the number of letters in the alphabet." The restrictions as stated in the above problem are no restrictions on the generating power of any usual subfamily of DIL systems since it is clear that by enlarging the alphabet we can simulate any DIL system G_1 by a DIL system G_2 where G_2 takes k_1 production steps to generate the axiom of G_1 and takes a constant number k_2 of productions steps of G_2 to simulate one production step of G_1, i.e. $\delta_2^{k_1+k_2t}(w_2) = \delta_1^t(w_1)$ for all t. (This is similar to deriving e.g. the Chomsky Normal Form for context free grammars.) Suppose we restrict ourselves to the family of PD1L systems and there is a function as proposed by Varshavsky where, moreover, this function is computable. Then it would also be decidable whether or not a PD1L system G simulating a Tag system T ever generates a string of the form $\$^k\not{c}$ for some k: contradicting Lemma 4.48. Therefore, we have

COROLLARY 4.52. *Let* V_i *by the family of* PD1L *systems* $G = \langle W,\delta,w\rangle$ *such that* $\#W = i$, $w \in W$, $\ell g(\delta(a,b,\lambda)) \leq 2$ *for all* $b \in W$ *and* $a \in W \cup \{\lambda\}$, *and* $\ell g(\delta^{t_0+t}(w)) = \ell g(\delta^{t_0}(w))$ *for some* t_0 *and all* t. *Let* $v(i) = \max\{\ell g(v) \mid v \in L(G)$ and $G \in V_i\}$. *There is no computable function* f *such that* $v(i) \leq f(i)$ *for all* i, *i.e.* v *increases faster then any computable function and hence Varshavsky's problem has a negative solution.*

THEOREM 4.53.

(i) *It is undecidable whether or not two* PD1L *systems are growth equivalent even if we have the advance information that they are of the same growth type* i, $i \in \{2,2\frac{1}{2},3\}$.

(ii) *It is undecidable whether or not two* D1L *systems are growth* *equivalent even if we have the advance information that they are of the same growth type* i, i $\in \{1\frac{1}{2},2,2\frac{1}{2},3\}$.

(iii) *The growth equivalence of two* DIL *systems is decidable if we have the advance information that they both have bounded growth functions.*

PROOF. Take an arbitrary Tag system T and simulate it with a PD1L system G_1 as in the proof of Theorem 4.49.

(i) Now construct two variants of G_1, called G_2 and G_3, which act like G_1 until $\$\not{c}$ occurs in a string; then G_2 and G_3 start different growths albeit of the same growth type i, i $\in \{2,2\frac{1}{2},3\}$. Now let f be another growth function of type i. Since PD1L growth functions are closed under addition (Theorem 4.38) both $g = f_{G_2} + f$ and $h = f_{G_3} + f$ are PD1L growth functions of type i, say of G_4 and G_5. If $\$\not{c}$ never occurs in a string then $f_{G_4} = f_{G_5} = f_{G_1} + f$ and $f_{G_1}(t) \in \Theta(t)$. If $\$\not{c}$ occurs in a string then $f_{G_4} \neq f_{G_5}$. Since it is undecidable whether $\$\not{c}$ occurs in a string it is undecidable whether or not $f_{G_4} = f_{G_5}$, where it is known that both f_{G_4} and f_{G_5} are of growth type i, i $\in \{2,2\frac{1}{2},3\}$.

(ii) Similar to (i). Since we talk here about D1L systems we can slow the growth function f_{G_1} down to $f_{G_1'}$ where $f_{G_1'} < \log_r t$, $r > 1$, (cf. discussion after Example 4.37).

(iii) Trivial. □

Note that the theorem above leaves open the decidability of the question of two PD1L systems being growth equivalent if we are informed in advance that they are both of growth type $1\frac{1}{2}$. This is because in our simulation method of Tag systems all simulating PD1L systems are either of growth type 1 or growth type 2.

THEOREM 4.54. *It is undecidable whether two* PD2L *systems are growth equivalent even if we are informed in advance that they are both of growth type* $1\frac{1}{2}$.

PROOF. Take a PD2L system G_1 simulating a Tag system T. Construct a PD2L system G_2 which simulates G_1 such that $f_{G_2}(t) < \log_r f_{G_1}(t)$ (cf. discussion after Example 4.37). Since $f_{G_1}(t) \in \Theta(t)$ or $f_{G_1}(t) \le m$ for some constant m, f_{G_2} is of growth type $1\frac{1}{2}$ or 1. Then use the method of proof of Theorem 4.53 (i). □

Theorems 4.49, 4.53 and 4.54 have analogues for the growth ranges of DIL systems. The *growth range* of a DIL system G is defined by R(G) = $\{\ell g(v) \mid v \in L(G)\}$. Although the results on growth ranges are not corollaries of Theorem 4.49, 4.53 and 4.54 they follow by the same proof method. Two DIL systems G_1 and G_2 are said to be *growth range equivalent* iff $R(G_1) = R(G_2)$.

THEOREM 4.55. *The growth range equivalence is undecidable for two* PD1L *systems* G_1 *and* G_2 *even if we have advance information that they both are of growth type* i, $i \in \{2, 2\frac{1}{2}, 3\}$.

PROOF. The proof of Theorem 4.53 (i) will do since we can choose f_{G_1} and f_{G_2} such that they are strictly increasing at different rates iff a substring $\$\cancel{c}$ occurs. □

Under appropriate interpretation we can prove the undecidability of growth range type classification etc. analogous to Theorems 4.49, 4.53 and 4.54. Note, however, that the growth range type can be different from the growth function type of a DIL system. E.g. $f_G(t) = 2^{\lfloor \log_2 t \rfloor}$ is of growth type 1 whereas $R(G) = \{2^i \mid i \geq 0\}$ and therefore is exponential.

4.4. BIBLIOGRAPHICAL COMMENTS

The first paper in the field of growth functions of L systems was by SZILARD [1971] who treated the analysis and synthesis problem for DOL systems with the generating function approach. In PAZ and SALOMAA [1973] growth functions of DOL systems are investigated from the point of view of integral sequential word functions and algorithms are obtained for the solution of the analysis, synthesis and growth equivalence problems. The difference equation method appears in DOUCET [1973], PAZ and SALOMAA [1973] and SALOMAA [1973b]. Section 4.1 is based on these papers. Section 4.2 is based on VITÁNYI [1973] and VITÁNYI [1976b]; and Section 4.3 on VITÁNYI [1974b]. The first example of a D1L system with subpolynomial growth is due to G.T. Herman ("Gabor's Sloth") as is the idea of simulating Tag systems with D1L systems (cf. HERMAN [1969]). An example of a D2L system with subexponential growth was first given by KARHUMÄKI [1974a,b]. An overview paper of some of the material contained in this chapter is HERMAN and VITÁNYI [1976]. The study of growth functions and related topics has become a very active field within the study of formal power series. See e.g.

SALOMAA [1976a,b] or SALOMAA and SOITTALA [1978].

CHAPTER 5

PHYSICAL TIME GROWTH FUNCTIONS ASSOCIATED WITH LINDENMAYER SYSTEMS OPERATING IN PHYSIOLOGICAL TIME

'Physiological time varies - in rate does it? and if so in what sense? - from one organism to another, and from one stage to another in the development of a single one.'

in P.B. MEDAWAR [1945].

The closed form solution of DOL growth functions we met in the previous chapter, a combination of polynomial and exponential terms, cannot account for the empirically derived *sigmoidal* growth curves we meet in developmental biology such as the logistic growth function $A/(1+Be^{-kt})$ or the monomolecular growth function $A(1-Be^{-kt})$, to name a few well-known ones, see MEDAWAR [1945]. Apart from this, there are also troubles with reconciling the theoretical framework and its mathematical consequences following from the L system model with experimental results obtained by biologists. If, however, we drop the assumption that changes (= rewriting of strings) in the system occur at unit time intervals, we can describe in the model phenomena like progressive dissipation of growth energy, biological rhythms, changes in environmental conditions which influence the growth rate etc. Thus we derive a hybrid model by assuming descrete cells and instantaneous cell division but continuous time. The number of past rewritings then corresponds to physiological time and the total time consumed to physical time. It is shown how, e.g., exponential growth in physiological time may lead to a logistic growth curve in physical time and, similarly, linear growth in physiological time to monomolecular growth in physical time. Some extensions of the model are discussed and an interpretation in terms of table Lindenmayer systems with a computable control word is given. The strength of the results seems to lie in the fact that the new model relates stereotype elemental (cellular) behavior to empirically observed overall growth curves.

If we want to obtain sigmoidal growth curves with the original L systems then not even the introduction of cell interaction does help us out.

In the first place we get quite unlikely flows of messages through the organism (see e.g. HERMAN and ROZENBERG [1975]) which are more suitable to electronic computers, and in fact give the organism the computing power of one. In the second place, we are still not able to obtain growth which, always increasing the size of the organism, tends towards stability in the limit. The slowest increasing growth we can obtain by allowing cell interaction is logarithmic and thus cannot account for the asymptotic behavior of sigmoidal growth functions like the logistic and monomolecular ones. Thirdly, context sensitive L systems are highly vulnerable to disturbances: a small disturbance usually causes a completely different behavior, contrary to biological organisms which are robust enough not to be swayed from their chosen path by minor disturbances.

Apart from this it can be argued that, for instance, purely exponential growth such as met in the theory of L systems, does not reflect biological reality: in a short time the organism would fill the universe! However, it has been shown that under continuous culture conditions bacteria and monocellular algae can easily be kept under exponential growth as can filamentous algae, Lück [private communication]. Of course, if the culture medium remains unaltered in time, as is eventually the case, there will be a sigmoidal growth curve. Mostly, growth curves of higher plants show this form. Sometimes, there is also a very long, nearly linear, median phase. Lianes grow that way. In any case, that real growth normally stops somehow is not necessarily related to food constraints but can also be the results of higher hierarchical processes such as flowering. Actually, however, in the last decades serious experimental workers seem only to consider the first so-called exponential phase.

Growth functions as occurring in developmental biology have a purely empirical origin. The size of an organism is plotted graphically against its age. The resulting curve is expressed, as accurately as need be, by means of an algebraic equation. No biological significance is attributed to the exact form this equation takes. The growth function's chief function is to facilitate the analysis of the curve of growth (MEDAWAR [1945]).

In this paper we attempt to clarify what in our view are some of the shortcommings of the otherwise quite appealing model of Lindenmayer and how to overcome them. As examples we show how to derive logistic and monomolecular growth curves.

In biology, as opposed to the usual automata theoretical approaches, we meet the problem of environment. In an organism each cell has an environment (apart from the adjacent cells) which is going to influence its behavior, c.q. division rate. In algae this is the surrounding water from which it draws its food. In larger plants the environment consists of the outside world, and inside the organism, e.g., the vessels which transport nutricients. Furthermore, growth inhibitors, temperature and, for all we know, the phases of the moon will influence the growth rate of the organism. Of course, every one of these exogenous influences may occasion changes in endogenous parameters. Apart from this, e.g. the following empirical generalizations are mentioned by MEDAWAR [1945].

(i) Size is a monotonic increasing function of age.

(ii) Usually, what results from growth is itself capable of growing.

(iii) Under the actual conditions of development living tissue progressively loses power to reproduce itself at the rate it was formed.

In automata theory we are dealing with abstractions which are not subject to physical constraints, and identical cells do identical things at all times. In actual organisms, differences in environment in space and time are going to create differences in cell behaviour such as division rates etc. So even if we assume that a cell is essentially an autonomous unit, changes and divisions so not occur at unit time intervals, but division times are governed by environmental parameters, like concentration and accessibility of nutricients, growth inhibitors, enzymes, temperature, light. It will come as no surprise that this is corroborated by experimental evidence.

The biologist observes very little real differences in cell types/ states (e.g., cells with distinct stereotype behavior). Erickson, in his experiments with growth in corn cobs, essentially distinguishes between cells in the core and those in the surrounding tissue only, and insists that all cells in one of these areas behave more or less alike. The Lücks, experimenting on algae, distinguish between four cell types (according to ancestry). Under changing environmental conditions they observe changes in size and division times only [private communication].

To account for differences in cell behavior induced by time or extracellular agents, the automata theorist is inclined to postulate a very large number of cell states. In doing so, he makes no distinction between the autonomous properties of cells, and changes in division times due to extracellular agents. We can overcome this difficulty by assuming but a few different cell types and taking intervals between changes in the model as a variable quantity. We shall call the elapsed time *physical* or *real* time and the number of times the model has undergone changes *physiological* time. This is in agreement with biological terminology. To quote MEDAWAR [1945] again:

..."Growth is more rapid earlier in life than later, and if the time intervals are equal in length - are days for example - the approximation will correspondingly be less efficient at the beginning than at the end. The length of the chosen interval should evidently bear some relation to the work done by the organism in its life span; to the organism's "physiological age" in fact... (Physiological time is biology's claim to be considered at least as obscure to the lay mind as theoretical physics. The organism it is argued, dispenses a Time of its own making by a just measure of the work done...)"

We want to show that the underlying model of L systems, even without cellular interactions, gains in adequacy and explaining power if we treat the time intervals between changes of cell states and divisions as a function of elapsed time, environmental parameters, and possibly the number of previous changes. Hence we consider L systems operating in physiological time and their associated physical or real time growth functions. Later on we solve some examples yielding well known growth curves. In the last section we formulate some extensions of the model on which the automata theorist might want to turn loose his bag of tricks, and show some relations with so-called table L systems.

To be more precise about the ideas we have in mind, recall the theory of DOL growth functions as explained in Section 4.1. Now imagine that the clock, which governs the discrete time rewriting of the string of cells does not tick at unit time intervals, but rather at variable time intervals corresponding with the relative slowing down or speeding up of the growth of the organism, under the influence of changes in the environmental and internal parameters, and maybe related with the number of previous rewritings. Each such variable length time interval then corresponds to the time elapsed between two consecutive rewritings of the string. That is, the time interval between the occurrences of the ι-th and $\iota+1$-th elements of $S(G)$ is given by $\tau_{\iota+1} - \tau_\iota$ where $\tau_{\iota+1}$ is the time elapsed up to the occurrence of the $\iota+1$-th element of $S(G)$ and τ_ι is the time elapsed up to the occurrence of the ι-th element of $S(G)$. To be able to use analytical methods we give the relation between ι and τ_ι by a continuous function t: $\mathbb{R}_+ \to \mathbb{R}_+$ (where $\mathbb{R}_+$ denotes as usual the positive real numbers) such that

$t(\iota) = \tau_\iota$ for all $\iota \in \mathbb{N}$. By its genesis t is strictly increasing on $\mathbb{N}$ and we consider only such functions t as are strictly increasing on $\mathbb{R}_+$ too. The function t can be interpreted as mapping the *physiological* time ι to the *physical* or *real* time τ_ι. Then the size (c.q. weight or number of cells) of the modeled organism at real time $t(\iota)$ is given by $L_G(t(\iota)) = f_G(\iota)$. (If complex constants enter in f_G this can have as its effect that values of L_G are complex for $\iota \in \mathbb{R}_+ - \mathbb{N}$. We circumvent this difficulty by either taking $L_G(t(\iota))$ equal to the absolute value of $f_G(\iota)$ in such cases or by only ascribing a physical interpretation to L_G and f_G for $\iota \in \mathbb{N}$.) Since $t: \mathbb{R}_+ \to \mathbb{R}_+$ is strictly increasing, there does also exist the inverse mapping $t^{-1} = i: \mathbb{R}_+ \to \mathbb{R}_+$ defined by $i(\tau) = \iota$ if $t(\iota) = \tau$. Then $\lfloor i(\tau) \rfloor$ gives the number of rewritings, starting from the initial string at time zero, which have occured up to time τ as a function of the real time τ elapsed. It seems reasonable to assume that the time delay between two consecutive stages (rewritings) of an organism is related to, e.g., the concentration of nutricients it has access to and the waste products and growth inhibitors it secretes. Such concentrations will be related to the organism's size and history in that environment. So the fundamental relation is

$$L_G(\tau) = f_G(i(\tau))$$

where $i: \mathbb{R}_+ \to \mathbb{R}_+$ is the physiological time as a function of the real time. Similarly, $t: \mathbb{R}_+ \to \mathbb{R}_+$ is the real time as a function of the physiological time. The function i is found by describing (if advantagous by differential equations) the relations between $t(\iota)$, $L_G(\tau)$, the influences of the environmental parameters which are not influenced by the organism such as temperature, day and night cycles, and the influences of the environmental parameters which are influenced by the organism such as food concentration. To take a simple example where we do not ascribe a physical meaning to $t(\iota)$. Suppose that $f_G(\iota) = 2^\iota$ and $t(\iota) = \iota^2$. Then $i(\tau) = \sqrt{\tau}$ and $L_G(\tau) = 2^{\sqrt{\tau}}$, a real time growth function of the so-called subexponential growth type. One assumption we have made is that the relative changes of time intervals in between the rewriting of a letter does not depend on the letter itself or its position in the string. The theory could be extended to take care of this too, cf. Section 5.2.

Below we will show by some examples that we can derive well-known biological growth functions by the above method. The problem of constructing real time growth functions for an organism modeled in physiological time by a DOL system consists in finding a plausible set of physical constraints (for instance, a set of differential equations), solving $i(\tau)$, and solving $L_G(\tau)$ from $f_G(i(\tau))$. In the sequel of this chapter we denote, for convenience sake, the values ι of the function $i(\tau)$ and the function itself both by i and, similarly, τ and t both by t. Which one is meant, the function or its value, will be clear from the context. So we will write $\frac{dt(i)}{di} = 1/\frac{di(t)}{dt}$ for $\frac{dt(\iota)}{d\iota} = 1/\frac{di(\tau)}{d\tau}$.

5.1. SIGMOÏDAL GROWTH FUNCTIONS OF LINDENMAYER SYSTEMS OPERATING IN PHYSIOLOGICAL TIME.

In this section we investigate some examples of growth behavior we are liable to meet according to the theory developed above. We shall be concerned with algae-like organisms which (I) reside in a closed environment containing an initial amount of food stuff, (II) are subject to periodic speeding up and slowing down of division rates (i.e., some sort of biological rhythm), and (III), (I) and (II) together.

(I). ORGANISMS IN A CLOSED ENVIRONMENT CONTAINING AN INITIAL AMOUNT OF NUTRICIENTS.

Suppose we have (fig. 5.1) a (filamentous) organism residing in a trough filled with water from which it draws its food. We shall assume that (i) the organism uses no food to maintain itself but only to grow; (ii) it excretes no waste products etc. which inhibit its growth; (iii) at all times the concentration of food throughout the trough is uniform; (iv) no parameters influence the growth except the concentration of food.

Let a(t) be the concentration of nutricients at time t. Assume that for $a(t) \geq a_0$ the environment is optimal and the organism grows according to the modeling DOL system, i.e., physiological time and real time are the same. After some time, say t_0 time units, the food level has been depleted to a_0 and the growth rate starts slowing down. Since the surface of the filamentous organism is proportional to its length (or the amount of cells it is made up of), i.e., the value of L_G, we choose our differential equations as follows.

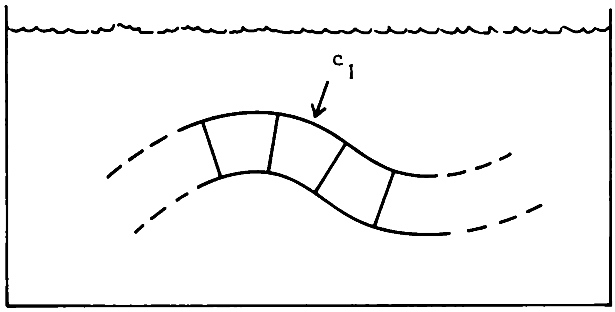


Figure 5.1.

$$(2) \qquad \frac{da(t)}{dt} = - c_1 f_G(t)a(t),$$

where c_1 is the nutricient absorption constant pro unit of organism. This yields

$$(3) \qquad a(t) = a(0)e^{-\int_0^t c_1 f_G(x)dx},$$

and substituting $a(t_0) = a_0$ yields t_0. From t_0 onwards the division times of cells grow larger because there is a food shortage and for $t \geq t_0$ we have

$$(4) \qquad \frac{da(t)}{dt} = - c_1 L_G(t)a(t),$$

$$(5) \qquad \frac{dt(i)}{di} = g(a(t)),$$

$$(6) \qquad L_G(t) = f_G(i(t)),$$

for some function g yet to be chosen. Since t is the inverse function of i (5) leads to

$$(7) \qquad \frac{di(t)}{dt} = 1/g(a(t)).$$

Considering everything in phase-space, (4), (6) and (7) give

(8) $$\frac{da}{di} = - c_1 f_G(i) a g(a),$$

and hence (with some abuse of notation)

(9) $$\int_{a=a_0}^{a(i)} \frac{1}{ag(a)}\, da = - c_1 \int_{i=t_0}^{i} f_G(i) di.$$

At this point we might wonder whether it is necessary to give a(t) a strong and explicit interpretation as food concentration. The fact that real growth normally stops somehow is not necessarily connected with exhaustive constraints but can also be the result of higher integrated processes such as flowering. See LÜCK [1966] for a discussion about largely independent levels of organization in a plant's hierarchical make up. Therefore, perhaps, it would be better to give a(t) a more mathematical purpose than a too restricted biological significance. For instance, integration constants may always enter into a(t).

<u>EXAMPLE 5.1.</u>: *the logistic growth curve.*

Assume that $f_G(i) = 2^i$ and $g(a) = c_2/a$. $t \le t_0$. According to (3):

$$a_0 = a(0)e^{-\int_0^{t_0} c_1 2^t dt}$$

which yields

$$t_0 = \log_2\left(1 + \frac{\ln 2}{c_1} \ln \frac{a(0)}{a_0}\right).$$

Substituting f_G and g in (9) yields, for $t \ge t_0$,

$$\frac{1}{c_2}\,(a(i) - a_0) = -c_1 \frac{1}{\ln 2}\,(2^i - 2^{t_0}).$$

Substitute $a(i) = c_2 \frac{di}{dt}$ and we have to solve i in

(10) $$\frac{di}{dt} = \frac{a_0}{c_2} - \frac{c_1}{\ln 2}\,(2^i - 2^{t_0}),$$

via separation of i and t,

$$(11)\qquad \int_{i=t_0}^{i} \frac{1}{A+B2^i}\,di = \int_{t=t_0}^{t} dt,$$

with

$$A = \frac{a_0 \ln 2 + c_1 c_2 2^{t_0}}{c_2 \ln 2}$$

and

$$B = \frac{-c_1}{\ln 2},$$

which yields, after substitution of $y = 2^i$,

$$(12)\qquad \int_{y=2^{t_0}}^{y} \frac{1}{A\,y \ln 2}\,dy - \int_{y=2^{t_0}}^{y} \frac{B}{A(A+By)\ln 2}\,dy = \int_{t=t_0}^{t} dt.$$

Solving i in (12) we obtain

$$(13)\qquad i(t) = \frac{1}{\ln 2} \quad \ln \frac{GA}{1-GB} \quad \text{with } G = \frac{2^{t_0} e^{A(t-t_0)\ln 2}}{A + B\,2^{t_0}}\ .$$

Substituting $i(t)$ in $f_G(i) = 2^i$:

$$L_G(t) = 2^{i(t)}$$

$$= \frac{-A/B}{1-1/BG}$$

$$= \frac{\dfrac{a_0 \ln 2}{c_1 c_2} + 2^{t_0}}{1 + \dfrac{a_0 \ln 2}{c_1 c_2 2^{t_0}}\, e^{-\left(\frac{a_0 \ln 2}{c_2} + 2^{t_0} c_1\right)(t-t_0)}}$$

which is of the form $\frac{X}{1+Ye^{-kt}}$.: the *logistic* or autocatalytic curve.

For $t = t_0$ we obtain: $L_G(t_0) = 2^{t_0} = 1 + \frac{\ln 2}{c_1} \ln \left(\frac{a(0)}{a_0}\right)$

For $t \to \infty$ we obtain: $L_G \max = 2^{t_0} + \frac{a_0 \ln 2}{c_1 c_2}$

$$= 1 + \frac{\ln 2}{c_1} \ln \left(\frac{a(0)}{a_0}\right) + \frac{a_0 \ln 2}{c_1 c_2}.$$

This yields the growth curve depicted in Figure 5.2 in which for $t \le t_0$: $L_G(t) = f_1(t) = 2^t$ and for $t \ge t_0$: $L_G(t) = f_2(t) =$ the above logistic growth function. The only parameters involved are c_1, c_2, $a(0)$ and a_0.

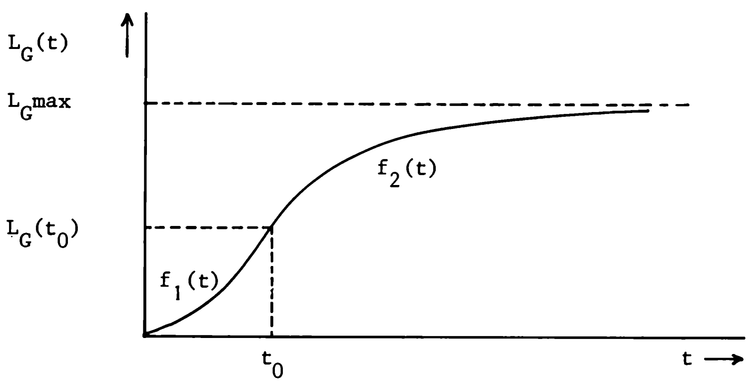


Figure 5.2.

EXAMPLE 5.2: *the monomolecular growth curve*. Assume that $f_G(t) = t+1$ and $g(a) = c_2/a$. Then, according to (3) we can solve t_0 from

$$a_0 = a(0)e^{-\int_0^{t_0} c_1(t+1)dt}$$

which yields $t_0 = -1 \pm \sqrt{1 + \frac{2}{c_1} \ln \frac{a(0)}{a_0}}$ and since $\frac{a(0)}{a_0}$ is greater than 1 for $t_0 > 0$, clearly,

$$t_0 = -1 + \sqrt{1 + \frac{2}{c_1} \ln \frac{a(0)}{a_0}}$$

and

$$L_G(t_0) = f_G(t_0) = \sqrt{1 + \frac{2}{c_1} \ell n \frac{a(0)}{a_0}} \ .$$

From (9) we see that, for $t \geq t_0$,

$$a(i) - a_0 = \frac{c_1 c_2}{2} ((t_0 + 1)^2 - (i + 1)^2)$$

$$= \frac{c_1 c_2}{2} (L_G(t_0)^2 - (i + 1)^2) \ .$$

Substituting $a(i) = c_2 \frac{di}{dt}$ we get

$$\frac{di}{dt} = \frac{a_0}{c_2} + \frac{c_1}{2} (t_0 + 1)^2 - \frac{c_1}{2} (i + 1)^2$$

and

$$\int_{i=t_0}^{i} \frac{di}{A-B(i+1)^2} = \int_{t=t_0}^{t} dt$$

with

$$A = \frac{a_0}{c_2} + \frac{c_1}{2} (t_0 + 1)^2$$

$$B = \frac{c_1}{2}$$

which yields

$$t = t_0 - \frac{1}{2\sqrt{AB}} \ln \frac{\sqrt{A/B}+(t_0+1)}{\sqrt{A/B}-(t_0+1)} + \frac{1}{2\sqrt{AB}} \ln \frac{\sqrt{A/B}+(i+1)}{\sqrt{A/B}-(i+1)} \ .$$

Setting

$$t_0 - \frac{1}{2\sqrt{AB}} \ln \frac{\sqrt{A/B}+(t_0+1)}{\sqrt{A/B}-(t_0+1)} \text{ to } Z,$$

$\frac{1}{2\sqrt{AB}}$ to Y and $\sqrt{A/B}$ to X we have, after some computation,

$$L_G(t) = f_G(i(t)) = i(t) + 1 = X\left(1 - \frac{2}{1+e^{-Z/Y} \cdot e^{t/Y}}\right)$$

and

$$L_G \max = \lim_{t\to\infty} L_G(t) = X = \sqrt{\frac{2a_0}{c_1c_2} + L_G(t_0)^2}.$$

The growth curve looks like Figure 5.3:

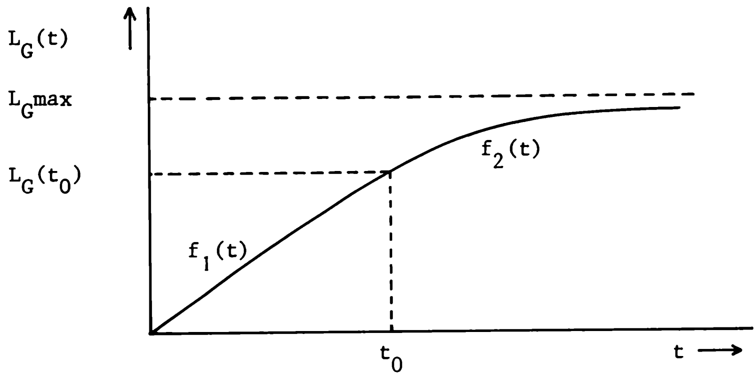


Figure 5.3.

$t < t_0$: $L_G(t) = f_1(t) = t+1$: linear,

$t \geq t_0$: $L_G(t) = f_2(t) = X(1-2(1+e^{-Z/Y} \cdot e^{t/Y})^{-1})$,

$t \gg t_0$: $L_G(t) \approx X(1-2e^{+Z/Y}e^{-t/Y})$: the *monomolecular* growth curve;

where

$$t_0 = -1 + \sqrt{1 + \frac{2}{c_1} \ln \frac{a(0)}{a_0}},$$

$$L_G(t_0) = \sqrt{1 + \frac{2}{c_1} \ln \frac{a(0)}{a_0}},$$

$$L_G\max = \sqrt{\frac{2a_0}{c_1c_2} + L_G(t_0)^2} = \sqrt{\frac{2a_0}{c_1c_1} + \frac{2}{c_1} \ln \frac{a(0)}{a_0} + 1}.$$

Hence we see that between the two extremes of unbounded DOL growth, viz. exponential and linear, the chosen set of differential equations, which depict the depletion of food, always yields a sigmoidal growth curve. Therefore, all unbounded DOL growth functions yield a sigmoidal growth curve under these conditions.

(II). ORGANISMS WITH A PERIODICAL CHANGE OF DIVISION RATE.

In biology we meet a phenomenon called biological rhythms. Examples are circadian rhythms, florescence etc. Such phenomena might be connected with the hierarchical organization of multicellular organisms, changes from daylight to night etc. According to the observations of the Lücks [private communication] the algae they observe show the following growth behavior. Under optimal conditions the algae behave in essence like a rather simple DOL system, LÜCK [1975], where each transition takes place after a unit time interval of 48 hours.

However, each fifth time interval the organism alternatively skips the required transition or executes two consecutive transitions in one time interval. Thus, after each period of ten time intervals the organism reaches the stage we would expect from the DOL model, but in between it periodically speeds up and slows down its growth rate. According to the discussion in the beginning of the chapter this means that

$$L_G(t) = f_G(i(t))$$

where i(t) is the function inverse of

$$t(i) = \begin{cases} i & \text{for } 0 \le i \bmod 10 < 5 \\ i+1 & \text{for } 5 \le i \bmod 10 \le 9. \end{cases}$$

Therefore,

$$i(t) = \begin{cases} t & \text{for } 0 \le t \bmod 10 < 5 \\ t-1 & \text{for } 5 \le t \bmod 10 \le 9. \end{cases}$$

Suppose $f_G(i) = 2^{i/5}$ then $L_G(t) = 2^{i(t)/5}$ and the growth curve is as depicted in Figure 5.4.

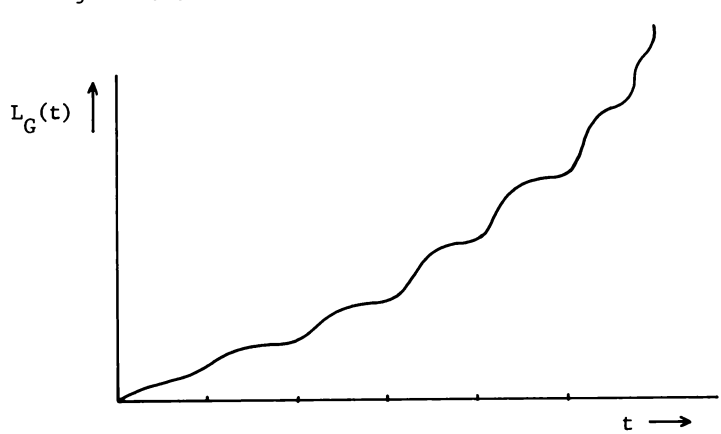


Fig. 5.4.

(III). COMBINATION OF (I) AND (II).

A combination of (I) and (II), i.e., an organism residing in a closed environment and showing periodic speed ups and slowing downs of growth rate, yields

$$L_G(t) = f_G(i(i'(t))),$$

where i is a function as found in (I) and i' a function as found in (II). The resulting growth curve looks like Figure 5.5, where we assume that the periodicity is independent of the organism's interaction with the environment.

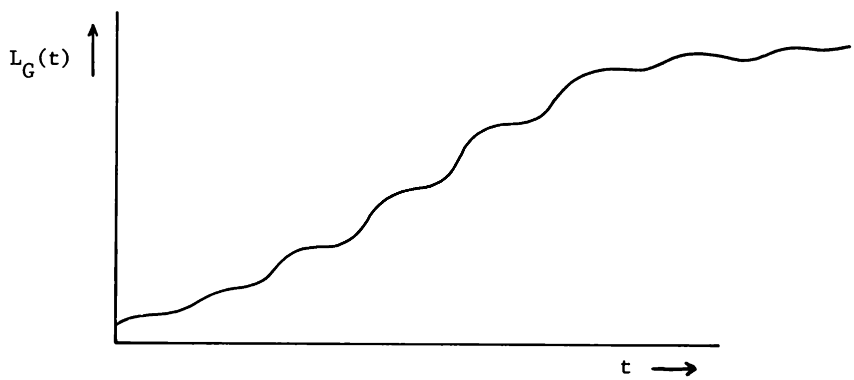


Figure 5.5.

5.2. SOME POSSIBLE EXTENSIONS AND AN INTERPRETATION IN TERMS OF TABLE L SYSTEMS.

The assumption that the relation between physiological time and real time is the same for all cell types in the organism can be relaxed, and we obtain in general that a is rewritten as $f(t,a) \in \{a, \delta(a)\}$, $a \in W$ and $t \in \mathbb{N}$. Then the growth matrix at time t is

$$M_G(t) = \begin{pmatrix} \overline{f(t,a_1)} \\ \overline{f(t,a_2)} \\ \vdots \\ \overline{f(t,a_n)} \end{pmatrix} \qquad \text{with} \quad W = \{a_1, a_2, \ldots, a_n\},$$

where $\overline{f(t,a_i)}$ will be $\overline{a_i}$ or $\overline{\delta(a_i)}$ depending on t. (In our previous approach this would mean that $M_G(t)$ is either the unit matrix I or M_G depending on t.) The above is useful to express different division times of different cell types without having to introduce different cell states to account for distinct delays in division rates. We could even go farther, and use the DTOL model. Recall that a DTOL *system* (deterministic context free table L system) is a triple $G = \langle W,\{\delta_1,\delta_2,\dots,\delta_k\},w\rangle$ such that for all i, $1 \le i \le k$, $G_i = \langle W,\delta_i,w\rangle$ is a DOL system. Recall from Section 3.3 that a *control word* u is an element of $\{1,2,\dots,k\}^*$. A word v is said to derive a word v' in G under the control word $u = i_1 i_2 \dots i_\ell$ if

$$v' = \delta_{i_\ell}\delta_{i_{\ell-1}}\dots\delta_{i_2}\delta_{i_1}(v).$$

Now we define, for $A = \{M_G(t) \mid t \in \mathbb{N}\}$ (A is finite) a DTOL system

$$G = \langle W,\{\delta_1,\delta_2,\dots,\delta_k\},w\rangle$$

where k is the number of elements in A and each table δ_i corresponds to the distinct element of A for which it is the associated set of rewriting rules, i.e., $A = \{M_{G_1},M_{G_2},\dots,M_{G_k}\}$, where

$$M_{G_i} = \begin{pmatrix} \overline{\delta_i(a_1)} \\ \overline{\delta_i(a_2)} \\ \vdots \\ \overline{\delta_i(a_n)} \end{pmatrix}$$

for all i, $1 \le i \le k$. Now a computable function $h\colon \mathbb{N} \to \{1,2,\dots,k\}$ is defined which has as its argument the real time t and is composed from functions which compute from the relevant parameters which table $\delta_{h(t)}$ is applicable at time t. Then the word sequence

$$S_h(G) = w,\ \delta_{h(1)}(w),\ \delta_{h(2)}\delta_{h(1)}(w),\dots,\delta_{h(t)}\delta_{h(t-1)}\dots$$
$$\dots\delta_{h(1)}(w),\dots$$

gives us the required developmental history of the modeled organism and the lengths of the successive elements of $S_h(G)$ give us the associated real time growth function.

<u>EXAMPLE</u>. Suppose we have $G = <\{a\},\{\delta(a) = a^2\},a>$ and $f(i) = 2^i$. If $t = i^2$ Then $L_G(t) = 2^{\sqrt{t}}$. The present approach would model the organism as follows.

$$G' = <\{a\},\{\delta_1(a) = a, \delta_2(a) = a^2\},a>$$

Hence $M_{G_1} = (1)$, $M_{G_2} = (2)$ and

$$h(t) = \begin{cases} 2 \text{ if } t \text{ is a square} \\ 1 \text{ if } t \text{ is not a square} \end{cases}$$

which yields $L_G(t) = 2^{\lfloor \sqrt{t} \rfloor}$.

We might note here that the approach taken previously in this chapter always leads to DTOL systems with *two* tables: if the physiological L system was $G = <W,\delta,w>$ then the associated DTOL system will be $G' = <W,\{\delta_1,\delta_2\}, w>$ where δ_1 is the identity function and $\delta_2 = \delta$. The associated function h satisfies

$$h(t) = \begin{cases} 2 \text{ if } t = t(i) \text{ for some } i \in \mathbb{N} \\ 1 \text{ if } t \neq t(i) \text{ for all } i \in \mathbb{N}. \end{cases}$$

As a further extension of the ideas presented above we could, e.g., make the choice of table, for rewriting a letter at time t, depend on the geometric position in the string of that occurrence of the letter. For instance, the tip of a root grows while the basal part does not. In this case, as in this section in general, not only the derived string sequence could be different from that of the underlying DOL system, but also the set of derived strings could differ from that of the underlying DOL system which does not happen with the approach in Section 5.1.

5.3. FINAL REMARKS

Although this chapter is concerned with L systems, i.e., models for filamentous organisms such as algae, the method used above should be applicable to more-dimensional growth as well. First find a, preferably context free, model of how the organism grows in physiological time (the essential cell ancestry and division pattern) and then try to find the functional relation between physiological time and real time. The advantages of such a procedure are that we have both one (qualitative) fundamental physiological time model and that the transition from one type of growth to another, e.g. from exponential to logistic, does not require changing the model but is a consequence of the functional relation between physiological and real time which governs the quantitative aspects of the matter.

Among experimentalists it is considered that the over-all approximations like exponential, logistic etc. growth curves have nothing to do with elemental (cellular) behavior. Furthermore, usually only the initial exponential stage is studied; the latter stages of growth are more or less neglected. We have tried to establish a relation between elemental behavior and the over-all growth curve and we have introduced as a most significant state of a growing organism, or of the history of a growing organism, the stage at which the growth ceases to be exponential and becomes sigmoidal: at time t_0.

The presented ideas should not be of interest solely for people working with algae but for every experimentalist who tries to fit theoretical growth functions to observed data.

5.4. BIBLIOGRAPHICAL COMMENTS

Chapter 5 is based on VITÁNYI [1977b]. Interesting discussions with P.G. DOUCET, R.O. ERICKSON, J. GRASMAN, H.B. LÜCK, J. LÜCK, W.J. SAVITCH and A.R. SMITH III concerning the subject matter of this chapter were valuable.

CHAPTER 6

EPILOGUE: EVALUATION OF RESULTS

In this final chapter we evaluate the work presented in this monograph. It does not lead up to one or a few main results but rather, textbook like, covers part of the field. Some results or topics are, for various reasons, more interesting than others. The two main themes are language classification (Ch. 3) and growth functions (Chs. 4 and 5). The techniques used are mainly combinatorial. To the author Chapter 5 seems an interesting one, in particular seen from the viewpoint of applications, to which purpose L systems were originally introduced.

It is shown there that, for very good reasons, the assumption of a unit time interval for transitions ought to be replaced by variable lengths time intervals related to physical constraints. The resulting model, with differential equations modeling these physical constraints, lead us to sigmoidal growth curves hitherto unattainable in the theory of growth functions of L systems. It also leads to a clear connection between elemental (cellular) behavior and the overall curve of growth and a theoretical interpretation about the transition from exponential to sigmoidal growth not requiring any change of model at the transition point. Theoretical exploration of the ideas presented in Chapter 5 and, hopefully, applications of these ideas in empirical investigations could prove most rewarding.

The treatment in Sections 3.2 and 3.3 of the generating power of context sensitive parallel rewriting with and without the use of various additions such as nonterminals, several types of homomorphic closures of the derived languages, and combinations of these generating power enhancing devices, fills a previously existing gap in L theory. Whereas the context free case was thoroughly studied, the context sensitive case was largely left open. In Section 3.2 the most important results are summarized in Table 3.1, where for the four main types of deterministic context sensitive L systems (PD1L, D1L, PD2L and D2L) the effect on the language generating power of the additional features is shown. The various (32) families of

XYZ L languages are provided with a least upper bound and a greatest lower bound (both with respect to set inclusion) of language families chosen from amongst the main families of the Chomsky hierarchy. We compared the power of XYZ L systems with these language families, since they are well understood and mathematically natural because of their closures under many operations. It appears that the pure L language families we consider are rather unevenly spread out, in the sense that none of them contains all finite languages but with erasing production rules they contain non-context sensitive and without erasing production rules still non-context free languages. In the latter case all languages produced are in $DLBA_{exp}$ by the fact that we cannot have more than a constant multiplicative factor length difference between two consecutive words in the string sequence. For the same reason we can replace RE as a l.u.b. on pure L language families, and their non-erasing homomorphic closures, by RE_{exp} which is a subclass of the recursively enumerable languages defined by

$$RE_{exp} = FIN \cup \{L \in RE \mid \exists c \in \mathbb{N}, \forall v \in L, \exists v' \in L$$
$$[\ell g(v) < \ell g(v') \leq c\ell g(v)]\}.$$

RE_{exp} is the somewhat strange class of recursively enumerable languages, of which the words in a language ordered by length differ in length at most a constant multiple. (Note that this class contains indeed strictly recursively enumerable languages.) It is easy to see that in Table 3.1 we can replace RE by RE_{exp} everywhere in rows (i) - (iii) but nowhere else. As a consequence of Theorem 3.59 the addition of but the simplest form of erasing homomorphism gives all types of context sensitive L systems the full power to generate all r.e. languages. Notice that this is a specific property of the parallel rewriting feature which enables the system to simulate a firing squad; for the set of sentential forms of a sequential rewriting (Chomsky type) grammar the erasing homomorphism would not be that powerful. In general we see that by adding more context and/or more additional features, like the use of nonterminals or closure under homomorphisms, the families of DIL languages induced by the various classes of L systems contract, as it were, more and more to a neat Chomsky type language family; where some types of context are more aided by some types of additional features than others. One-sided context, especially in combination with λ-free production rules, is more resistant to the beneficial effects of additional features than the others. This is due to prefix properties accompanying one-sided context, of which Theorem 3.57 gives the most beautiful example. Particularly nice is also the role played by the family of finite languages of which (the proof of) Theorem 3.54 illustrates how surprisingly difficult it can be to generate them even by quite powerful means. (The difficulty lies of course again with the problem of recurring prefixes.) Another interesting point is that if we can generate all regular sets with the means we consider here, we can also generate all $DLBA_{exp}$ languages. It appears that, for parallel rewriting plus the considered additional devices, FIN and REG play the part of a turning point: by adding more and more we sometimes hesitatingly pass FIN, but when we pass REG we immediately jump to $DLBA_{exp}$ or higher in generating power. That extensions of PD2L languages are exactly the DLBA languages is the subject of Theorem 3.55 which, together with an older result of van Dalen that EP2L equals CS, yields an analogue of the classic LBA problem in L theory.

Section 3.3 considers the same questions as Section 3.2 but now for the general case of (nondeterministic) L systems with or without tables. Here the results are summarized in Figure 3.2. Most important in this section is Lemma 3.63 which, together with the earlier Theorem 3.55, yields a pleasant restatement of the LBA problem as a trade-off in L parlance. Whereas in the classic statement of the problem we have to prove that nondeterminism gives (no) additional power over determinism for LBAs, here we have to prove something different: a sharp trade-off between two-sided context with one set of production rules and one-sided context with two sets of production rules is (not) possible for EPDTIL systems. Hence it is a real quid pro quo which was not the case in the original statement of the LBA problem. Nice as it is, unfortunately the result does not seem to bring the solution for the LBA problem any nearer.

In Section 3.4 we investigate what have earlier been called adult languages. For this somewhat scabrous name we have substituted the less biologically motivated, but more descriptive, name of stable string languages. In this section we prove that by combinations of four fundamental lemmas most of the extant results, and quite a few new ones, concerning the subject can be derived. It appears that the stable string device is exactly as powerful a language squeezing mechanism as the use of nonterminals, except when context sensitivity is absent in any form, even the mild one of the use of tables. In that case the stable string device is strictly less powerful than the use of nonterminals, and we generate exactly the context free languages. Note that the stable string feature is only relevant to the type of parallel rewriting we consider; it would do nothing for sequential rewriting.

In Section 3.1 we consider structural (combinatorial) investigations of DOL systems: quite a few properties of the derived sequences and languages are understood by the methods used (recursive letters, associated digraphs etc.). One of the nice results here is the one that relates the maximal cardinality of a finite DOL language over an n letter alphabet with the maximal order of a permutation of the n-th degree (Theorem 3.10 and Corollary 3.17), which in turn leads us into the realm of analytic number theory in Section 3.1.1.1. There two new, more or less natural, number theoretic functions are introduced which are related to, but behave sometimes quite different from, the one mentioned above. The brute force computer aided investigations of these functions by O. Østerby led, amongst other things, to a counter example to an implied conjecture of E. Landau to the effect that the largest number we can obtain by taking the least common multiple of a partition of the sum of the first k primes in positive integral summands is the product of the first k primes. With regard to L system theory it is shown that a DOL system with an n letter alphabet can generate a finite language of cardinality about $e^{\sqrt{n \log n}}$. Furthermore, the results of Section 3.1.1 lead us to the interesting fact that, if a DOL system generates a finite language, we can ascertain this fact and solve the membership problem both, by solely examining the first n productions of each letter in the alphabet (of cardinality n). The result rests on an application of the Chinese remainder theorem and led N. Jones and S. Skyum to quite interesting results concerning the complexity of recognizing deterministic context free (tabled extension) L languages. See Theorem 3.15, the lemmas leading up to it and the discussion afterwards.

One problem in L theory, which has been open for quite a time, is to give a method for deciding whether the string sequence generated by a DOL system is locally catenative, that is, whether there is an integer n_0 such that from the n_0-th string onwards each string in the sequence consists of the concatenation of strings which occur earlier in the sequence. The problem has been called L problem #2, L problem #1 being the DOL equivalence problem which recently has been solved by K. Čulik and I. Fris. Using the methods developed earlier in Section 3.1 we study aspects of L problem #2 in

Section 3.1.2.2. First we derive a necessary (but not sufficient) condition on the recursive structure of a D0L system for it to be locally catenative. Secondly, we show that if a D0L system is locally catenative then n_0 may be very large: about $e^{\sqrt{n \log n}}$ for a D0L system using n letters. From the necessary condition above it appears namely, that there is an intrinsic connection between locally catenative D0L systems and D0L systems generating a finite language: locally catenative D0L systems are, as it were, exponentially growing versions of D0L systems generating finite languages. From this it follows that we might have to look very far in the string sequence before the locally catenative property appears. Thirdly it is proven that the string sequence of a D0L system is locally catenative iff the monoid generated by the language produced by the system is finitely generated. Thus we relate a property of the derived string sequence with a property of the derived language. Theorems 3.28, 3.31 and 3.33 are the interesting ones here.

Of interest is also Section 3.5; not because of the results obtained, but because of the ideas hinted at to generalize L systems so as to obtain a model in which we may cope with the physical constraints relevant to the growth of bulky organisms. The ideas of Section 3.5 are presented at greater lengths, and in more worked-out fashion, in the author's master thesis (VITÁNYI [1971]).

The second half (Chs. 4 and 5) of this monograph deals with the subject of L growth functions; which subject is regarded, among L people, as being abundantly biologically motivated. Above we already expounded on our vested interest in Chapter 5. In Chapter 4 we explain first, in Section 4.1, the fundamentals of the analytical theory of D0L growth functions. Section 4.2, which ties in with Section 3.1.2.1 treats the structural (or more combinatorial) approach to D0L growth functions. It is shown there how to classify D0L growth on the basis of structural properties alone, which is computationally more easy (but less accurate) than the analytical methods of Section 4.1. Theorem 4.21 tells us how to do this on the basis of the structural characterization of types of D0L growths, and Theorem 3.27 even tells us how to determine the degree of polynomial growth in this way. Important in Section 4.2 is the structural proof of Theorem 4.14 to the effect that there exists no D0L growth functions of the subpolynomial (type 1½) and the subexponential (type 2½) variety. Although strongly suggested by the analytical expression for a D0L growth function, to the author's knowledge this is the easiest (even first?) proof for this fact.

In Section 4.3 appears about all which is known about context sensitive L growth. It is shown there, that virtually everything is undecidable in this realm which, surprisingly enough considering the proof of this fact, was an open problem for some time: the decidability status of the equivalence problem for context sensitive L growth was stated as an open problem in at least one textbook and a research paper. An important problem in this area, which is still open, consists of whether D1L growth is properly included in D2L growth and, similarly, whether PD1L growth is properly included in PD2L growth. Theorem 4.32 tells us that in both cases the classes are approximately equal, but the precise problem stays unsolved. Several examples solving at one time open problems are given or described. In Section 4.3.2 we show how to synthesize some context sensitive growth functions by, e.g., using a firing squad synchronization. It appears in Theorem 4.36 that for each rational number r we can find a context sensitive L system with growth of order of magnitude t^r. We then show in Section 4.3.3 several results about the hierarchy induced in growth functions by the amount of context in the system and, e.g., the number of letters used. The most interesting one says that context sensitive growth over a one letter alphabet is in the realm of context free growth (Theorem 4.42), but context sensitive growth over two letters gets outside of that realm (Theorem 4.44) Yet the latter does not contain all context free growth. A general matter which appears from Chapter 4 is, that, although we can realize much more growth functions with context sensitive L systems than with context free L systems (especially of the unbounded subpolynomial type), we are still not able to obtain sigmoidal growth (Theorem 4.25). To obtain sigmoidal growth we require a (justified) change of model as described in Chapter 5 where we obtain this type of, biologically important, growth even for the context free model.

Future research and open problems. As said before, work in L systems ought to provide techniques which are of use to the practicing biologist. As such, work in the area disclosed by Chapter 5 (or even the topics touched on in Section 3.5) seems most important. On the more theoretical side, L problem #2 of deciding whether a D0L system has the locally catenative property is a venerable old problem, and its solution would possibly shed light on more generally interesting properties of (monoid) morphisms.

Interesting within L system theory is whether D2L growth properly includes D1L growth, and the corresponding problem for the λ-free case. (Also e.g. the problem of whether PD2L growth is strictly included in D1L growth). A further investigation in structural properties of DOL systems, e.g., how degree of exponential growth is related to structural properties of the homomorphisms, as for instance appearing in the associated digraphs of Section 3.1.2, is also needed. Sections 3.2 - 3.4 seem pretty much complete short of solving the LBA problem. A further list of (at the time) open problems in L theory appears in LINDENMAYER and ROZENBERG [1976]. (Recall that ČULIK and FRIS [1977a,b] have already solved L problem #1.)

BIBLIOGRAPHY

Although the references concerning L system theory contained in this bibliography are fairly extensive, they are by no means intended to be exhaustive or to be the most important ones. In this monograph we restricted ourselves to those which are relevant to our treatment of the subject. For more or less complete L system bibliographies see ROZENBERG and SALOMAA [1974], HERMAN and ROZENBERG [1975] and LINDENMAYER and ROZENBERG [1976]. Furthermore, there regularly appear L system bibliographies edited by G. Rozenberg of the Math. Dept. of the University of Antwerp (U.I.A.).

BAKER, R. & G.T. HERMAN, [1970], *CELIA - a cellular linear iterative array simulator*. In: Proceedings of the Fourth Conference on Applications of Simulation, 64-73.

_________ & ___________, [1972a], *Simulation of organisms using a developmental model, Part I: Basic description,* International Journal of Bio-Medical Computing, 3, 201-215.

_________ & ___________, [1972b], *Simulation of organisms using a developmental model, Part II: The heterocyst formation problem in blue-green algae,* International Journal of Bio-Medical Computing, 3, 251-267.

BALZER, R., [1967], *An 8 state minimal solution to the firing squad synchronization problem,* Information and Control, 10, 22-42.

BERSTEL, J., [1971], *Sur les poles et le quotient de Hadamard de séries N-rationelles,* Comptes Rendus Acad. Sc. Paris, t. 272, 1079-1081.

BERSTEL, J. & M. MIGNOTTE, [1975], *Deux propriétés décidables des suites récurrentes linéaires*. In: Seminaire d'Informatique, Univ. Louis Pasteur, Strasbourg.

CHOMSKY, N., [1957], *Syntactic Structures.* Mouton, 's-Gravenhage.

ČULIK II, K., [1975], *Weighted growth functions of DOL systems and growth functions of parallel graph rewriting systems*. In: Abstracts of the Conference on Formal Languages, Automata and Development held at Noordwijkerhout, The Netherlands.

ČULIK II, K. & I. FRIS, [1977a], *The decidability of the equivalence problem for DOL systems,* Information and Control, 35, 20-39.

ČULIK II, K. & I. FRIS, [1977b], *The sequence equivalence for DOL systems is decidable*. In: Proc. IV Coll. on Automata, Languages and Programming, Lecture Notes in Computer Science, 52, Springer Verlag, Berlin, 148-163.

DALEN, D. van, [1971], *A note on some systems of Lindenmayer,* Mathematical Systems Theory, 5, 128-140.

DOUCET, P., [1972], *On the membership question in some Lindenmayer systems,* Indagationes Mathematicae, 34, 45-52.

__________, [1973], *The growth of word length in DOL systems*. In: Proc. of an Open House in Unusual Automata Theory, Comp. Sci. Dept., Univ. of Aarhus, Tech. Rept. DAIMI PB-15, 83-94.

__________, [1975], *On the applicability of L systems in developmental biology*. In: Abstracts of the Conference on Formal Languages, Automata and Development held at Noordwijkerhout, The Netherlands.

__________, [1976], *Studies on L Systems*. Ph.D. Thesis, Univ. of Utrecht.

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

______________ & ____________, [1974b], *Trade-off between the use of nonterminals, codings and homomorphisms in defining languages for some classes of rewriting systems*. In: Proc. II Coll. on Automata, Languages and Programming, Lecture Notes in Computer Science, 14, Springer-Verlag, Berlin, 473-480 (also, Acta Informatica 3, (1974), 265-283.)

______________ & ____________, [1974c], *DOL systems with rank*. In: L systems, Lecture Notes in Computer Science, 15, Springer-Verlag, Berlin, 136-141.

______________ & ____________,[1978], *Simplifications of homomorphisms,* Information and Control, (Forthcoming).

EHRENFEUCHT, A., G. ROZENBERG & S. SKYUM, [1976], *A relationship between ETOL and EDTOL Languages,* Theoretical Computer Science, 1, 325-330.

ENGELFRIET, J., E.M. SCHMIDT & J. van LEEUWEN, [1977], *Stack automata and classes of nonnested macro languages,* Comp. Sci. Dept., Univ. of Utrecht, Tech. Rept. RUU - CS - 77 - 2.

FRIJTERS, D., [1976], *Control Mechanisms and Computer Descriptions of Inflorescence Development*. Ph. D. Thesis, Univ. of Utrecht.

FRIJTERS, D. & A. LINDENMAYER, [1974], *A model for the growth and flowering of Aster novae-angliae on the basis of table <1,0> L systems*. In: L systems, Lecture Notes in Computer Science, 15, Springer-Verlag, Berlin, 24-52.

____________ & ______________, [1976], *Developmental descriptions of branching patterns with paracladial relationships*. In: A. Lindenmayer and G. Rozenberg (eds.), Automata, Languages, Development. North-Holland, Amsterdam, 57-73.

HARARY, F., [1971], *Graph Theory*. Addison-Wesley.

HARDY, G.H. & E.M. WRIGHT, [1945], *An Introduction to the Theory of Numbers*. Oxford Univ. Press.

HARTMANIS, J. & A.B. HUNT III, [1974], *The LBA problem and its importance in the theory of computing.* In: R.M. Karp (ed.), Complexity of Computation, SIAM-AMS Proc. Vol. VII. Am. Math. Soc., Providence, R.I.

HELLENDOORN, P.H. & A. LINDENMAYER, [1974], *Phyllotaxis in Bryophyllum tubiflorum: morphogenetic studies and computer simulations,* Acta Botanica Neerlandica, 23, 473-492.

HERMAN, G.T., [1969], *The computing ability of a developmental model for filamentous organisms,* Journal of Theoretical Biology, 25, 421-435.

____________, [1971], *Models for cellular interactions in development without polarity of individual cells. Part I: General description and the problem of universal computing ability,* International Journal of Systems Sciences, 2, 271-289.

____________, [1972], *Models for cellular interactions in development without polarity of individual cells. Part II: Problems of synchronization and regulation,* International Journal of Systems Sciences, 3, 149-175.

HERMAN, G.T., [1973], *Polar organisms with apolar individual cells*. In: P. Suppes et. al. (eds.), Logic, Methodology and Philosophy of Science IV. North-Holland, Amsterdam, 665-675.

____________, [1974], *Closure properties of some families of languages associated with biological systems,* Information and Control, 24, 101-121.

HERMAN, G.T. & W.H. LIU, [1973], *The daughter of CELIA, the French flag and the firing squad,* Simulation, 21, 33-41.

HERMAN, G.T., W.H. LIU, S. ROWLAND & A. WALKER, [1974], *Synchronization of growing cellular arrays,* Information and Control, 25, 103-121.

HERMAN, G.T. & G. ROZENBERG, [1975], *Developmental Systems and Languages.* North-Holland, Amsterdam.

HERMAN, G.T. & G.L. SCHIFF, [1975], *Simulation of multi-gradient models of organisms in the context of L systems,* Journal of Theoretical Biology, 54, 35-46.

HERMAN, G.T. & P.M.B. VITÁNYI, [1976], *Growth functions associated with biological development,* American Mathematical Monthly, 83, 1-15.

HERMAN, G.T. & A. WALKER, [1975], *Context free languages in biological systems,* International Journal of Computer Mathematics, 4, 369-391.

____________ & _________, [1976], *On the stability of some biological schemes with cellular interactions,* Theoretical Computer Science, 2, 115-130.

HOGEWEG, P., [1976], *Topics in Biological Pattern Analysis.* Ph.D. Thesis, Univ. of Utrecht.

HOGEWEG, P. & B. HESPER, [1974], *A model study of biomorphological description,* Pattern Recognition, 6, 165-179.

HOPCROFT, J.E. & J.D. ULLMAN, [1969], *Formal Languages and their Relation to Automata.* Addison-Wesley, Reading, Mass.

JONES, N.D. & S. SKYUM, [1976], *Recognition of deterministic ETOL languages in polynomial time*, Comp. Sci. Dept., Univ. of Aarhus, Tech. Rept. DAIMI PB-63.

__________ & ________, [1977a], *Upper bounds on the complexity of some problems concerning L systems*, Comp. Sci. Dept., Univ. of Aarhus, Tech. Rept. DAIMI PB-69.

__________ & ________, [1977b], *Lower bounds on the complexity of some problems concerning L systems*, Comp. Sci. Dept., Univ. of Aarhus, Tech. Rept. DAIMI PB-70.

__________ & ________, [1977c], *Complexity of some problems concerning L systems*. In: Proc. IV Coll. on Automata, Languages and Programming, Lecture Notes in Computer Science, 52, Springer-Verlag, Berlin, 301-308.

__________ & ________, [1977d], *Recognition of deterministic ETOL languages in logarithmic space*, Information and Control, 35, 177-181.

KARHUMÄKI, J., [1974a], *Some growth functions of context dependent L systems*. In: L Systems, Lecture Notes in Computer Science, 15, Springer-Verlag, Berlin, 127-135.

____________, [1974b], *An example of a PD2L system with growth type $2\frac{1}{2}$*, Information Processing Letters, 2, 131-134.

KNUTH, D., [1969a], *Fundamental Algorithms*, Addison-Wesley, Reading, Mass.

________, [1969b], *Seminumerical Algorithms*, Addison-Wesley, Reading, Mass.

LANDAU, E., [1903], *Über die Maximalordnung der Permutationen gegebenen Grades*, Archiv der Math. und Phys., Dritte Reihe, 5, 92-103.

LEE, K.P. & G. ROZENBERG, [1974], *TIL systems and languages*, Dept. of Math., Univ. of Antwerp (U.I.A.), Tech. Rept. 74-13. (Also, Information Sciences, 12 (1977), 203-227.)

LEEUWEN, J. van, [1974], *Notes on pre-set pushdown automata*. In: L Systems, Lecture Notes in Computer Science, 15, Springer-Verlag, Berlin, 177-188.

LEEUWEN, J. van, [1975a], *The tape complexity of context independent developmental languages,* Journal of Computer and System Sciences, 11, 203-211.

_______________, [1975b], *Deterministically recognizing EOL languages in time* $O(n^{3.81})$, Mathematical Centre, Amsterdam, Tech. Rept. IN 9/75.

_______________, [1975c], *The membership question for ETOL languages is polynomially complete,* Information Processing Letters, 3, 138-143.

_______________, [1976], *A study of complexity in hyper-algebraic families.* In: A. Lindenmayer and G. Rozenberg (eds.), Automata, Languages, Development. North-Holland, Amsterdam, 323-333.

LINDENMAYER, A., [1968a], *Mathematical models for cellular interactions in development, Part I,* Journal of Theoretical Biology, 18, 280-299.

_______________, [1968b], *Mathematical models for cellular interactions in development, Part II,* Journal of Theoretical Biology, 18, 300-315.

_______________, [1973], *Cellular automata, formal languages and developmental systems.* In: P. Suppes et. al. (eds.), Logic, Methodology and Philosophy of Science IV. North-Holland, Amsterdam, 677-691.

_______________, [1975], *Developmental systems and languages in their biological context.* Contribution to G.T. Herman and G. Rozenberg, Developmental Systems and Languages. North-Holland, Amsterdam.

LINDENMAYER, A. & G. ROZENBERG (eds.), [1976], *Automata, Languages, Development.* North-Holland, Amsterdam.

LINNA, M., [1977], *The decidability of the DOL prefix problem,* International Journal of Computer Mathematics, 6, 127-141.

LÜCK, H.B., [1966], *Sur la traduction morphogénétique de la cinétique cellulaire pendant la croissance des entre-noeuds de Tradescantia Flumensis Vel.,* Bull. de la Société Botanique de France, 220-225.

LŨCK, H.B., [1975], *Elementary behavior rules as a foundation for morphogenesis,* Journal of Theoretical Biology, 54, 23-34.

LŨCK, H.B. & J. LŨCK, [1976], *Cell number and cell size in filamentous organisms in relation to ancestrally and positionally dependent generation times.* In: A. Lindenmayer and G. Rozenberg (eds.), Automata, Languages, Development. North-Holland, Amsterdam, 109-124.

LŨCK, J., [1977], *Complémentarité Apico-Basale dans l'Organisation des Tissus Végétaux.* Ph.D. Thesis, Univ. d'Aix-Marseille III. Cahiers de Biologique Théoretique Vol. 1, Fasc. 1.

MEDAWAR, P.B., [1945], *Size, shape and age.* In: W.E. le Gros Clark and P.B. Medawar (eds.), Essays on Growth and Form. Oxford Univ. Press.

MILLO, R.A. de, R.J. LIPTON & A.J. PERLIS, [1977], *Social processes and proofs of theorems and programs.* In: Fourth ACM Symp. on Principles of Programming Languages, ACM, New York, N.Y., 206-214.

NIELSEN, M., [1974], *On the decidability of some equivalence problems for DOL systems,* Information and Control, 25, 166-193.

MINSKY, M., [1967], *Computation: Finite an Infinite Machines. Prentice-Hall.*

NIELSEN, M., G. ROZENBERG, A. SALOMAA & S. SKYUM, [1974a], *Nonterminals, homomorphisms and codings in different variations of OL systems. Part I: Deterministic systems,* Acta Informatica, 4, 87-106.

__________, __________, __________ & ________, [1974b], *Nonterminals, homomorphisms and codings in different variations of OL systems. Part II: Nondeterministic systems,* Acta Informatica, 3, 357-364.

PAZ, A. & A. SALOMAA, [1973], *Integral sequential word functions and growth equivalence of Lindenmayer systems,* Information and Control, 23, 313-343.

PENTTONEN, M., [1975], *One sided and two sided context in formal grammars,* Information and Control, 25, 371-392.

POLLUL, W. & D. SCHÜTT, [1975], *Characterization of growth in deterministic Lindenmayer systems,* Indagationes Mathematicae, 37, 183-188.

PROCEEDINGS of an Open House in Unusual Automata Theory, [1972], Comp. Sci. Dept., Univ. of Aarhus, Tech. Rept. DAIMI PB-15.

PROCEEDINGS of the 1974 Conference on Biologically Motivated Automata Theory, [1974], IEEE, New York.

RABIN, M.O. & H. WANG, [1963], *Words in the history of a Turing machine with fixed input,* Journal of the ACM, 10, 526-527.

ROBINSON, R.W., [1970], *Enumeration of acyclic digraphs.* In: Proc. Second Chapel Hill Conf. on Comb. Math. and its Applications, Univ. of North Carolina, Chapel Hill, N.C., 391-399.

ROGERS, H., [1967], *Theory of Recursive Functions and Effective Computability.* McGraw-Hill, New York.

ROZENBERG, G., [1972a], *L systems with interactions: the hierarchy,* Dept. Comp. Sci., State Univ. of New York at Buffalo, Tech. Rept. 28-72.

____________, [1972b], *Propagating L systems with interactions,* Dept. Comp. Sci., State Univ. of New York at Buffalo, Tech. Rept. 29-72.

____________, [1973a], *TOL systems and languages,* Information and Control, 23, 357-381.

____________, [1973b], *Extensions of tabled OL systems and languages,* International Journal of Computer and Information Sciences, 2, 311-334.

____________, [1974], *On a family of acceptors for some classes of developmental languages,* International Journal of Computer Mathematics, 4, 199-228.

ROZENBERG, G. & P.G. DOUCET, [1971], *On OL languages,* Information and Control, 19, 302-318.

ROZENBERG, G. & K.P. LEE, [1975], *Some properties of the class of L languages with interactions,* Journal of Computer and System Sciences, 11, 129-147.

ROZENBERG, G. & A. LINDENMAYER, [1973], *Developmental systems with locally catenative formulas,* Acta Informatica, 2, 214-248.

ROZENBERG, G. & A. SALOMAA (eds.), [1974], *L Systems,* Lecture Notes in Computer Science, 15, Springer-Verlag, Berlin.

RUOHONEN, K., [1974], *Some regenerative Lindenmayer models.* Licentiate Thesis, Dept. of Math., Univ. of Turku.

__________, [1975], *On the synthesis of DOL growth,* Ann. Acad. Sci. Fenn. Ser. A I Mathematica, 1, 143-154.

__________, [1976], *A symmetric D2L model for string regeneration.* In: A. Lindenmayer and G. Rozenberg (eds.), Automata, Languages, Development. North-Holland, Amsterdam, 263-270.

SALOMAA, A., [1973a], *Formal Languages.* Academic Press, New York.

__________, [1973b], *On exponential growth in Lindenmayer systems,* Indagationes Mathematicae, 35, 23-30.

__________, [1973c], *On sentential forms of context free grammars,* Acta Informatica, 2, 40-49.

__________, [1974], *Iteration grammars and Lindenmayer AFLs.* In: L Systems, Lecture Notes in Computer Science, 15, Springer-Verlag, Berlin, 250-253.

__________, [1975a], *Formal power series and growth functions of Lindenmayer systems,* Lecture Notes in Computer Science, 26, Springer-Verlag, Berlin, 101-113.

__________, [1975b], *Comparative decision problems between sequential and parallel rewriting.* In: Proceedings of the International Symposium on Uniformly Structured Automata and Logic, Tokyo, 62-66.

__________, [1976a], *Growth functions of Lindenmayer systems: some new approaches.* In: A. Lindenmayer and G. Rozenberg (eds.), Automata, Languages, Development. North-Holland, Amsterdam, 271-282.

__________, [1976b], *L systems: a parallel way of looking at formal languages. New ideas and recent developments.* In: K.R. Apt and J.W. de Bakker (eds.), Foundations of Computer Science II, Part 2, Mathematical Centre Tracts, 82, Mathematical Centre, Amsterdam, 67-110.

SALOMAA, A. & M. SOITTALA, [1978], *Automata-Theoretic Aspects of Formal Power Series,* Lecture Notes in Computer Science, Springer-Verlag, Berlin. (Forthcoming.)

SAVITCH, W.J., [1975], *Some characterizations of Lindenmayer systems in Chomsky type grammars and stack machines,* Information and Control, 27, 37-60.

SOITTALA, M., [1976], *On DOL synthesis problem.* In: A. Lindenmayer and G. Rozenberg (eds.), Automata, Languages, Development. North-Holland, Amsterdam, 313-321.

STAFLEU, G., [1973], *A generating formula for a branching system,* Theoretical Biology Group, Univ. of Utrecht, Working Report.

SUDBOROUGH, I.H., [1977], *The time and tape complexity of developmental languages.* In: Proc. IV Coll. on Automata, Languages and Programming, Lecture Notes in Computer Science, 52, Springer-Verlag, Berlin, 509-523.

SZILARD, A., [1971], *Growth functions of Lindenmayer systems,* Comp. Sci. Dept., Univ. of Western Ontario, Tech. Rept. 4.

TURING, A.M., [1936], *On computable numbers, with an application to the Entscheidungsproblem,* Proc. London Math. Soc., 42, 230-265.

____________, [1952], *The chemical basis of morphogenesis,* Philos. Trans. Royal Society London Ser. A, 237, 37-72.

VEEN, A.H. & A. LINDENMAYER, [1973], *Generation of phyllotactic patterns by diffusion,* manuscript.

VITÁNYI, P.M.B., [1971], *On generalized cellular automata and sexually reproducing cellular automata.* Master's Thesis, Math. Dept., Delft Technological University.

_______, [1972a], *Context variable Lindenmayer systems and some simple regenerative structures,* Mathematical Centre, Amsterdam, Tech. Rept. NR 24/72.

_______________, [1972b], *DOL languages and a feasible solution for a word problem,* Mathematical Centre, Amsterdam, Tech. Rept. MR 138/72.

_______________, [1973], *Structure of growth in Lindenmayer systems,* Indagationes Mathematicae, 35, 247-253.

VITÁNYI, P.M.B., [1974a], *On the size of DOL languages*. In: L Systems, Lecture Notes in Computer Science, 15, Springer-Verlag, Berlin, 78-92.

______________, [1974b], *Growth of strings in context dependent Lindenmayer systems*. In: L Systems, Lecture Notes in Computer Science, 15, Springer-Verlag, Berlin, 104-123.

______________, [1976a], *Deterministic Lindenmayer languages, nonterminals and homomorphisms,* Theoretical Computer Science, 2, 49-71.

______________, [1976b], *Digraphs associated with DOL systems*. In: A. Lindenmayer and G. Rozenberg (eds.), Automata, Languages, Development. North-Holland, Amsterdam, 335-346.

______________, [1977a], *Context sensitive table Lindenmayer languages and a relation to the LBA problem,* Information and Control, 33, 217-226.

______________, [1977b], *Physical time growth functions associated with developmental models operating in physiological time,* Mathematical Centre, Amsterdam, Tech. Rept. IW 87/77.

VITÁNYI, P.M.B. & A. WALKER, [1978], *Stable string languages of Lindenmayer systems,* Information and Control, 37, (Forthcoming.)

WADDINGTON, C.H., [1957], *The Strategy of the Genes*. Allen and Unwin, London.

WALKER, A., [1974a], *Dynamically stable strings in developmental systems, self repair, and the Chomsky hierarchy*. In: Proc. 1974 Conf. on Biologically Motivated Automata Theory, IEEE, New York, 46-49.

__________, [1974b], *Adult languages of L systems and the Chomsky hierarchy*. In: L Systems, Lecture Notes in Computer Science, 15, Springer-Verlag, Berlin, 201-215.

__________, [1974c], *Formal Grammars and the Stability of Biological Organisms*. Ph.D. Thesis, Dept. Comp. Science, State Univ. of New York at Buffalo..

WALKER, A., [1975], *Formal grammars and regeneration capability of biological systems,* Journal of Computer and System Sciences, 11, 252-261.

WOLPERT, L., [1968], *The French flag problem: A contribution to the discussion on pattern development and regulation.* In: C.H. Waddington (ed.), Towards a Theoretical Biology, Vol. 1. Edinburgh Univ. Press, Edinburgh, 125-133.

WOOD, D., [1976], *Iterated a-NGSM maps and Γ systems,* Information and Control, 32, 1-26.

ØSTERBY, O., [1973], *Prime decompositions with minimum sum,* Comp. Sci. Dept., Univ. of Aarhus, Tech. Rept. DAIMI PB-19.

___________, [1976], *Prime decompositions with minimum sum,* BIT, 16, 451-458.

SAMENVATTING

De gemeenschappelijke interessesfeer van de informatica en de biologie is niet leeg. In het bijzonder bevat zij problemen verbonden met de ontwikkelingsbiologie. De ontwikkeling van een organisme kan worden opgevat als de uitvoering van een ontwikkelingsprogramma dat aanwezig is in de bevruchte eicel. Door de cellulariteit van hogere organismen worden wij in staat gesteld de organismen in ontwikkeling te beschouwen als dynamische collectieven van passend geprogrammeerde automaten. Tot de taak van de ontwikkelingsbiologie behoort dan onder meer de fundamentele algoritme af te leiden uit het verloop van de groei en ontwikkeling van het organisme. Onderzoek aan algoritmische processen vormt weer een van de centrale onderwerpen uit de informatica. Het is dan ook niet vreemd dat de theoretische informatica, en de automaten- en formele talentheorie in het bijzonder, waardevolle inzichten verschaft in problemen betreffende biologische ontwikkeling. Omgekeerd hebben biologische motivaties en voorbeelden nieuwe ideeën bijgedragen tot de theoretische informatica. De voorlopers van de automatentheorie, zoals de zenuwnetten van McCulloch en Pitts en de reproducerende automaten van von Neumann, waren biologisch gemotiveerd.

Omstreeks 1968 stelde A. Lindenmayer de L systemen voor als modellen voor eendimensionale groei en ontwikkeling. Deze modellen sluiten nauw aan bij de theorie der formele talen. Dientengevolge werd het onderzoek naar de mathematische aspecten hiervan ter hand genomen door een allengs groeiende groep wiskundigen en beoefenaren van de theoretische informatica. De in aanvang biologische motivatie raakte in veel gevallen op de achtergrond. L systemen vormen een parallelle variant op de, eerder door N. Chomsky in de mathematische linguistiek ingevoerde, generatieve grammatica's. De (mathematische) theorie van L systemen heeft dan ook veel problemen (met of zonder biologische motivatie) en bewijstechnieken gemeen met de formele talentheorie en vormt daarbinnen tegenwoordig een hoofdstroming. Door het parallel toepassen van de herschrijfregels lenen L systemen zich beter tot een wiskundige aanpak dan de oudere generatieve grammatica's. Verder leidt de studie van L systemen tot tal van nieuwe problemen en toepassingen.

In deze monografie komen drie aspecten van L systemen uitgebreid aan de orde: de *structuur* van deterministische contextvrije L systemen, de *taalklassificatie* van families van talen voortgebracht door contextgevoelige L systemen, en de *groeifuncties* van deterministische L systemen. Bij al deze aspecten speelt de door het L systeem gegenereerde woordrij een centrale rol.

- De structuur van een deterministisch contextvrij L systeem (DOL) bestaat uit de, door het gebruikte homomorfisme bepaalde, voortbrengingsrelaties tussen de letters van het alfabet van de DOL. Deze relaties worden uitgedrukt door een klassificatie van de letters in typen en door met het homomorfisme geassocieerde eindige gerichte grafen. De samenhang tussen eigenschappen van de structuur van een DOL systeem en de globale eigenschappen van de door het systeem voortgebrachte woordrij, taal, en groeifunctie, worden onderzocht. Noodzakelijke en voldoende voorwaarden onder welke een DOL systeem een eindige taal voortbrengt worden gegeven en leiden tot het vaststellen van de maximale cardinaliteit van een eindige DOL taal over een alfabet van n letters. De orde van grootte hiervan is $e^{\sqrt{n \log n}}$. Het probleem van het vaststellen of een woord v door een (P)DOL systeem G wordt voortgebracht heet het *lidmaatschapsprobleem* voor (P)DOL talen. De benodigde tijd/geheugen ruimte voor een algoritme dat dit probleem oplost, wordt uitgedrukt in de lengte van v voor het *gewone lidmaatschapsprobleem*, en in de lengte van v en de beschrijving van G voor het *algemene lidmaatschapsprobleem*. Uit de resultaten volgt dat het algemene lidmaatschapsprobleem voor (E)PDOL systemen en de eindigheid van DOL talen beslist kunnen worden door een deterministische algoritme die in polynomiale tijd werkt. Indien een DOL taal oneindig is, kan de kwestie of een woord v door een DOL systeem G over een alfabet van n letters voortgebracht wordt opgelost worden door n * lengte (v) woorden van de met G geassocieerde woordrij te genereren en met v te vergelijken. Met inachtneming van speciale voorzieningen voor de letters die het lege woord voortbrengen, leidt dit tot een polynomiale deterministische algoritme. In het geval dat de taal die door G voortgebracht wordt eindig is, kan deze taal echter uit $e^{\sqrt{n \log n}}$ woorden van gelijke lengte bestaan. Door gebruik te maken van de gevonden eigenschappen van DOL systemen die een eindige taal voortbrengen en door, onder meer, het toepassen van de Chinese reststelling blijkt ook voor dit geval een uitvoerbare polynomiale deterministische algoritme mogelijk. Enige getallentheoretische functies, voortspruitende uit het verband tussen de grootte van het alfabet en de grootte van een eindige DOL taal, worden gedefinieerd en onderzocht. Deze fucties zijn varianten op de reeds door E. Landau onderzochte maximale orde van een permutatie van de n - de graad.

DOL systemen zijn *lokaal-katenatief* indien de voortgebrachte woordrij een zekere fibonacci-achtige eigenschap bezit. De structuur van zulke lokaal-katenatieve D0L systemen wordt bepaald. Een structureel verband, tussen lokaal-katenatieve D0L systemen en D0L systemen met eindige taal, leidt tot de conclusie dat de lokaal-katenatieve eigenschap pas zeer laat in de gegenereerde woordrij hoeft op te treden. Een en ander heeft zijn consequenties voor het zoeken naar een algoritme die uitmaakt of een gegeven D0L systeem de lokaal-katenatieve eigenschap heeft. Voorts wordt aangetoond dat het bezit van de lokaal-katenatieve eigenschap equivalent is met het eindig voortgebracht zijn van de door de D0L taal voortgebrachte monoïde.

De *groeifunctie* van een deterministisch L systeem beeldt het argument i af op de lengte van het i-de woord in de geassocieerde woordrij. Onder meer wordt de structuur bepaald van D0L systemen waarvan de groeifunctie van de orde van grootte van een polynoom van een gegeven graad is.

Voorts worden structurele condities aangegeven voor het al dan niet regulier of contextvrij zijn van D0L talen.

Een en ander vormt het onderwerp van Sectie 3.1.

- Het onderzoek naar *talen* van L systemen houdt ons bezig in Secties 3.2 - 3.4. In Sectie 3.2 worden voor deterministische contextgevoelige L systemen de pure taalfamilies, de taalfamilies verkregen met gebruikmaking van hulpsymbolen, en de afsluitingen van de voorgaande taalfamilies onder diverse typen homomorfismen onderzocht en volledig geklassificeerd in de Chomsky hierarchie. Hierdoor wordt de kracht van parallelle (deterministische) taalvoortbrenging (al dan niet geholpen door hulpsymbolen en/of homomorfe afbeeldingen) vergeleken met de kracht van de traditionele sequentieële taalvoortbrenging door generatieve grammatica's. De belangrijkste resultaten zijn bevat in Tafel 3.1 en Figuur 3.1. Sectie 3.3 behandelt soortgelijke kwesties voor al dan niet deterministische tafel L systemen. Dit zijn L systemen voorzien van meerdere stellen herschrijfregels, waarbij op een en hetzelfde woord slechts regels uit één stel toegepast mogen worden. Onder meer blijkt, dat het klassieke LBA probleem uit de formele talentheorie equivalent is met de kwestie of het gebruik van één tafel en tweézijdige context evenveel kracht biedt met betrekking tot taalgeneratie als het gebruik van tweé tafels en éénzijdige context, voor een bepaalde klasse L systemen. De belangrijkste resultaten van deze sectie zijn bevat in Figuur 3.2. In Sectie 3.4 bekijken we een alternatieve vorm van taalvoortbrenging die eigen is aan de L systemen. De woorden in een pure L taal die invariant zijn onder de herschrijfregels vormen de met het L systeem geassocieerde taal van *stabiele woorden*. Aangetoond wordt dat de op deze wijze uit een klasse van L systemen verkregen taalfamilie in het algemeen gelijk is aan de taalfamilie welke uit die klasse verkregen wordt door gebruik te maken van hulpsymbolen. Een uitzondering wordt gevormd door de klassen van deterministische L systemen met één tafel en door de klasse van contextvrije L systemen met één tafel. In het laatste geval is de familie van talen die bestaan uit stabiele woorden gelijk aan de familie van contextvrije talen, en dus strikt bevat in de taalfamilie door deze klasse van L systemen voortgebracht met gebruik van hulpsymbolen. De belangrijkste resultaten in deze sectie zijn bevat in de Figuren 3.3 en 3.4. In Sectie 3.5 wordt een L systeem variant geïntroduceerd in verband met een regeneratieprobleem.

- In het voorafgaande werd reeds gesproken over *groeifuncties* geassocieerd met L systemen. Sectie 4.1 behandelt de reeds eerder bekende analytische theorie van D0L groeifuncties. In Sectie 4.2 wordt het verband bestudeerd tussen de structuur van een D0L systeem en zijn groeifunctie. Voor de klassificatie van een D0L groeifunctie bijvoorbeeld, zal het beschouwen van de structuur van het D0L systeem vaak meer, snellere, maar soms vagere, informatie geven dan de analytische methode. Groeifuncties van contextgevoelige L systemen vormen het onderwerp van Sectie 4.3. Zulke groeifuncties zijn voorheen weinig onderzocht. In de onderhavige sectie wordt een aanzet gegeven tot een theorie hierover. Onder andere worden de analyse, synthese, hierarchieën en klassificatie van deze groeifuncties bestudeerd. Het blijkt dat vrijwel alle kwesties die voor de contextvrije variant beslisbaar zijn, voor de contextgevoelige variant niet beslisbaar zijn.

Een der grote tekorten van de hierboven geschetste theorie wordt gevormd door het onvermogen om hierbinnen de, in de biologie veel voorkomende, *sigmoïdale* groeifuncties te genereren. In Hoofdstuk 5 worden de oorzaken hiervan aangegeven en wordt de conventie van het eenheidstijdsinterval tussen opeenvolgende woorden in de woordrij vervangen door het toelaten van tijdsintervallen van variabele lengte. De rangorde van een woord in de woordrij wordt geassocieerd met de physiologische tijd van het gemodelleerde organisme, en de som van de voorafgaande tijdsintervallen wordt geassocieerd met de physische tijd die verstreken is tot het verschijnen van bovengenoemd woord. We verkrijgen dan een hybride L systeem met discrete herschrijving, dat opereert in continue tijd. Hierbij is de lengte van de verschillende tijdsintervallen afhankelijk van, bijvoorbeeld, physische gegevenheden. Dit beantwoordt meer aan de eisen die qua adequaatheid van modellering van de biologische ontwikkeling van een organisme aan L systemen gesteld mogen worden. Het blijkt dat, onder redelijke aannamen, de bekende logistische en monomoleculaire (sigmoïdale) groeifuncties uit de biologie verkregen worden.

- Hoofdstuk 1 geeft een inleiding tot de theorie der L systemen, en behandelt het verband met de biologie enerzijds en met de wiskunde en de theoretische informatica anderzijds. Hoofdstuk 2 geeft, in Sectie 2.1, enige begrippen en stellingen uit de theorie der formele talen en, in Sectie 2.2, definities uit de theorie der L systemen. In Hoofdstuk 6 worden de in het voorafgaande verkregen resultaten geëvalueerd.

STELLINGEN

1. Varshavsky definieert de functie $L(n)$ als de maximale eindige lengte van een configuratie, die kan groeien uit één geactiveerde automaat, in een lineaire cellulaire ruimte bestaande uit identieke eindige toestandsautomaten met n toestanden. De functie $L(n)$ stijgt sneller dan enige berekenbare functie, zelfs indien het transport van informatie in de cellulaire ruimte slechts in één richting kan plaatsvinden.

   V.I. Varshavsky, Some effects in the collective behaviour of automata. In: B. Meltzer and D. Michie (eds.), *Machine Intelligence* 7, Edinburgh University Press, Edinburgh, 1972, 389-403.

   P.M.B. Vitányi, On a problem in the collective behavior of automata, *Discrete Mathematics* 14 (1976) 99-101.

2. Zij $d(G,n)$ het minimale quotient van het aantal kanten, die gelijkgekleurde punten verbinden, en het totaal aantal kanten in een toekenning van n kleuren aan de punten van een graaf G.
   (i) $d(G,n) \leq 1/n$; $d(K_p,n) < 1/n$, $p \geq 2$, en $\lim_{p\to\infty} d(K_p,n) = 1/n$ voor alle n.
   (ii) Een n-partitie van de punten in G zodanig dat het bovengenoemde quotient niet het globale optimum $d(G,n)$ geeft, maar een lokaal optimum, in de zin dat de verandering van de kleur van één punt geen verbetering oplevert, kan gevonden worden met een $\mathcal{O}(epn)$ deterministische algoritme. (Hierbij is e het aantal kanten en p het aantal punten in G.) Dit lokale optimum geeft een quotient $\leq 1/n$.
   (iii) Het bepalen van $d(G,n)$ is NP-compleet.
   (iv) Er bestaat een algoritme voor het bepalen van $d(G,n)$ die werkt in deterministische tijd $\mathcal{O}(p\{{p \atop n}\})$, waarbij $\{{p \atop n}\}$ een Stirling getal van de tweede soort is.

   P.M.B. Vitányi, How "good" can a graph be n-colored, MC rapport IW 81, Mathematisch Centrum, Amsterdam, 1977.

3. Geen klasse van oneindige recursief opsombare talen die alle oneindige recursieve talen bevat is recursief opsombaar. Dit in tegenstelling tot de klassen der eindige-, recursieve-, en recursief opsombare talen, welke alle recursief opsombaar zijn.

P. van Emde Boas & P.M.B. Vitányi, A note on the recursive enumerability of some classes of recursively enumerable languages, *Information Sciences* (te verschijnen).

Th. M.V. Janssen, G. Kok & L.G.L.T. Meertens, On restrictions on transformational grammars reducing the generating power, *Linguistics and Philosophy* 1 (1977) 111-118.

4. De volgende pompstelling geldt voor Dyck talen D. Indien $w_{n_0} = \alpha\beta^{n_0}\gamma\delta^{n_0}\mu \in D$ voor $n_0 >$ lengte $(\alpha\beta\gamma\delta\mu)$ dan geldt voor alle $n \geq 1$ dat $w_n = \alpha\beta^n\gamma\delta^n\mu \in D$.
N.B. Deze stelling vertoont qua uiterlijk een oppervlakkige, maar misleidende, gelijkenis met de uvwxy stelling. Het ongewone van de stelling ten opzichte van andere pompstellingen bestaat onder meer hierin, dat we met behulp van deze stelling woorden zowel kunnen oppompen als leegpompen.

P.M.B. Vitányi & W.J. Savitch, On inverse deterministic pushdown transductions, *Journal of Computer and System Sciences* (te verschijnen).

5. Een deterministische stapelvertaler (deterministic pushdown transducer) is een model voor een simpele compiler die brontalen in objekttalen afbeeldt. Zij $S(\mathrm{L})$ (c.q. $S(\mathcal{L})$) de klasse van brontalen waarvan iedere taal door een deterministische stapelvertaler in de objekttaal L (c.q. in een objekttaal L uit de klasse van objekttalen $\mathcal{L}$) afgebeeld kan worden, terwijl het complement van de brontaal in het complement van L afgebeeld wordt. Dat wil zeggen, $S(\mathrm{L}) = \mathrm{dpdt}^{-1}(\mathrm{L})$ en $S(\mathcal{L}) = \mathrm{dpdt}^{-1}(\mathcal{L})$, waarbij dpdt^{-1} de klasse van inverse deterministische stapelvertalingen voorstelt. Zij verder $\mathrm{dgsm}^{-1}(\mathcal{D})$ de klasse van inverse deterministische gegeneraliseerde sequentieële machine-afbeeldingen van de klasse $\mathcal{D}$ van Dyck talen; $\mathrm{dgsm}^{-1}(\mathcal{D})$ bevat bijvoorbeeld de klasse van blokstructuurtalen met reguliere restricties.
   (i) Brontalen in $S(\mathrm{dgsm}^{-1}(\mathcal{D}))$ worden herkend door deterministische Turing machines met één band in tijd $\mathcal{O}(n^2)$ en geheugenruimte $\mathcal{O}(n)$. Verder worden zij herkend door deterministische off-line Turing machines in geheugenruimte $\mathcal{O}(\log^2 n)$. Dientengevolge geldt dat $S(\mathrm{dgsm}^{-1}(\mathcal{D}))$ echt bevat is in de DLBA talen. Ofschoon $S(\mathrm{dgsm}^{-1}(\mathcal{D}))$

triviaal de deterministische contextvrije talen bevat, bevat zij niet alle contextvrije talen.

$$L = \{a^i b^j \mid i \le j\} \cup \{a^i b^j c^k \mid i + j = k\} \cup \{a,b,c\}^*\{\cent\},$$

$\cent \notin \{a,b,c\}$, is een contextvrije taal die niet tot $S(\mathrm{dgsm}^{-1}(\mathcal{D}))$ behoort.

(ii) Het is onbeslisbaar of een taal in $S(\mathrm{dgsm}^{-1}(\mathcal{D}))$ leeg is. De afsluiting van $S(\mathrm{dgsm}^{-1}(\mathcal{D}))$ onder homomorfismen is de klasse van recursief opsombare talen.

(iii) Voor iedere objekttaal L bestaat er een taal $\bar{L}$ die kompleet is voor $S(L)$ met betrekking tot geheugenruimte.

P.M.B. Vitányi & W.J. Savitch, On inverse deterministic pushdown transductions, *Journal of Computer and System Sciences* (te verschijnen).

6. Het door von Neumann in 1953 opgestelde model voor de zelfreproductie van automaten kan worden opgevat als asexuele voortplanting. Het inbedden in cellulaire ruimten van een model voor de sexuele voortplanting van automaten blijkt een natuurlijke extensie te zijn van von Neumann's werk; het ontwerpen en onderzoeken van de daarmee samenhangende automaten-genetica levert bovendien mogelijk waardevolle inzichten betreffende de biologische erfelijkheidstheorie.

J. von Neumann, *Theory of Self-Reproducing Automata* (A.W. Burks ed.). University of Illinois Press, Ill., 1966.

P.M.B. Vitányi, Sexually reproducing cellular automata, *Mathematical Biosciences* 18 (1973) 23-54.

P.M.B. Vitányi, Genetics of reproducing automata. In: *Proc. 1974 Conference on Biologically Motivated Automata Theory*, IEEE, New York, 1974, 166-171.

7. De klasse van recursief opsombare verzamelingen kan als volgt gekarakteriseerd worden.

(i) Voor iedere recursief opsombare verzameling $L \subseteq \Sigma^*$ bestaat er een reguliere verzameling R zodanig dat

$$L = h(\bar{D}_2 \cap h(\bar{D}_2 \cap R)),$$

waarbij de reguliere verzameling $R \subseteq (\Sigma \cup \{a,b,\bar{a},\bar{b},0,1,\bar{0},\bar{1}\})^*$, $\Sigma \cap \{a,b,\bar{a},\bar{b},0,1,\bar{0},\bar{1}\} = \emptyset$; $\bar{D}_2$ de Dyck-achtige verzameling: shuffle $(D_2,(\Sigma \cup \{a,b,\bar{a},\bar{b}\})^*)$ is, met D_2 de Dyck taal over $\{0,1,\bar{0},\bar{1}\}$; en h het homomorfisme van $(\Sigma \cup \{a,b,\bar{a},\bar{b},0,1,\bar{0},\bar{1}\})^*$ op $(\Sigma \cup \{a,b,\bar{a},\bar{b}\})^*$ is, dat gedefinieerd wordt door $h(c) = c$ voor $c \in \Sigma$, $h(c) = \varepsilon$ voor $c \in \{0,1,\bar{0},\bar{1}\}$, $h(a) = 0$, $h(\bar{a}) = \bar{0}$, $h(b) = 1$, en $h(\bar{b}) = \bar{1}$.

(ii) Iedere recursief opsombare verzameling over Σ kan uitgedrukt worden in de vorm $h(\bar{D} \cap L)$ waarbij L een deterministische contextvrije taal is; $\bar{D}$ een Dyck-achtige taal: shuffle (D,Σ^*) met D een Dyck taal over een alfabet Δ, $\Delta \cap \Sigma = \emptyset$; en h het homomorfisme is, gedefinieerd door $h(a) = \varepsilon$ voor $a \in \Delta$ en $h(a) = a$ voor $a \in \Sigma$.

(iii) Iedere recursief opsombare verzameling kan worden uitgedrukt in de vorm $\tau^{-1}(L)$. Hierbij is L een deterministische contextvrije taal; τ een gemerkte Dyck taal-substitutie, dat wil zeggen, er is een Dyck taal D zodanig dat voor iedere a in het domein van τ geldt dat $\tau(a) = aD$, waarbij de respectieve alfabetten van D en de recursief opsombare verzameling disjunct zijn.

P.M.B. Vitányi & W.J. Savitch, On inverse deterministic pushdown transductions, *Journal of Computer and System Sciences* (te verschijnen).

8. De navolgende stelling moet gezien worden in het kader van het uitbreiden van de verzameling instructies van het basis Turing machine-model (meer in het bijzonder ten behoeve van snellere geheugen toegang), zonder dat de berekeningscomplexiteitsklassen hierdoor (noemenswaardig) veranderen. Dat wil zeggen, we ontwerpen machinemodellen die zich bevinden tussen de standaard meerband Turing machine en de registerautomaat, en die dezelfde complexiteitsklassen hebben als het standaard meerband Turing machine-model.
Een Turing machine M met meerdere lees-schrijfkoppen per band en met de toegevoegde instructie:

"herzet kop i naar de positie van kop j op de band in één machinestap, ongeacht de afstand tussen beiden"

kan gesimuleerd worden door een standaard meerband Turing machine M' in lineaire tijd. Dit houdt in dat, als de originele machine M een berekening uitvoert in tijd $T(n)$, waarbij n de lengte van de invoer is, de simulerende machine de berekening zal uitvoeren in tijd $cT(n)$ voor een constante c. Is M deterministisch, dan kan de simulatietijd verkort worden tot $n + \varepsilon T(n)$ voor willekeurig kleine constante ε. Is M niet-deterministisch dan is $T(n)$ voldoende. Voor $k > 1$ geldt:

- Elke bandeenheid met k koppen en herzetinstructies kan zo gesimuleerd worden door $8k - 8$ standaard bandeenheden met één kop per band.
- Is de gesimuleerde bandeenheid niet-deterministisch dan volstaan $4k$ standaard bandeenheden met één kop per band.

W.J. Savitch & P.M.B. Vitányi, Linear time simulation of multihead Turing machines with head-to-head jumps. In: *Proc. IVth Int. Coll. on Automata, Languages and Programming,* Lecture Notes in Computer Science 52, Springer, Heidelberg, 1977, 453-464.

9. Zij M een deterministische stapelautomaat (dpda) met n_1 toestanden, n_2 stapelsymbolen en ℓ de lengte van het langste woord waarmee het bovenste symbool op de stapel overschreven kan worden. Zij $S(n_1,n_2,\ell)$ het maximale aantal nietlezende stappen dat een dpda kan maken zonder in een berekeningslus te geraken en zonder de stapel te verlagen tot onder de aanvangshoogte. Een en ander onder de aanname dat de dpda input kan lezen. Dan is

$$((\ell - 1)^{n_1}\ell^{(n_1+1)(n_2-2)} - \ell^{n_2-2})/((\ell - 1)\ell^{n_2-2} - 1)$$

$$\leq S(n_1,n_2,\ell) \leq (\ell^{n_1 n_2} - 1)/(\ell - 1),$$

voor $n_1 \geq 1$, $n_2 \geq 3$ en $\ell \geq 2$.

Zij $T(n)$ het aantal stappen in de langste berekening van een dpda met parameters n_1, n_2, ℓ tot het moment van het lezen van de n-de letter van de invoer. Dan geldt

$$(2n - 1)((\ell - 1)^{n_1}\ell^{(n_1+1)(n_2-2)} - \ell^{n_2-2})/((\ell - 1)\ell^{n_2-2} - 1)$$

$$\leq T(n) \leq (n - 1)(\ell^{n_1 n_2} - 1)/(\ell - 1),$$

voor $n_1 \geq 1$, $n_2 \geq 3$ en $\ell \geq 2$.

Voor de overgebleven waarden van de parameters n_1, n_2, ℓ gelden andere, eenvoudiger, grenzen

P.M.B. Vitányi, Achievable high scores in ε-moves and running times in DPDA computations, MC Rapport IW 70, Mathematisch Centrum, Amsterdam, 1976.

10. Het door E. Landau gesuggereerde vermoeden, dat de maximale orde van een permutatie van de n-de graad gelijk is aan het produkt van de eerste k priemgetallen indien n gelijk is aan de som van de eerste k priemgetallen, is onjuist.

E. Landau, Über die Maximalordnung der Permutationen gegebenen Grades, *Archiv der Math. und Phys., Dritte Reihe,* 5 (1903), 92-103.

O. Østerby, Prime decompositions with minimum sum, *BIT* 16 (1976) 451-458.

Dit proefschrift, Sectie 3.1.1.1.

11. De Erdösgraaf heeft de wiskundingen als punten. Twee punten zijn verbonden door een kant indien de corresponderende wiskundigen tenminste één gemeenschappelijk artikel geschreven hebben. De afstand tussen twee punten is gelijk aan het kleinste aantal kanten in een pad dat deze twee punten verbindt. Een subgraaf van de Erdösgraaf, waarin ieder punt afstand 1 heeft tot ieder ander punt, vormt een *kliek*. We beschouwen nu de verzameling van potentieële klieken in de Erdösgraaf waarvan elke kliek de geslachtsverhouding van de algemene menselijke populatie weerspiegelt, zeg 108 1/8 : 100, en waarbij de verhouding tussen het aantal kanten dat personen van gelijk geslacht verbindt en het aantal kanten dat personen van verschillend geslacht verbindt gelijk is aan $1 + \varepsilon$, $-0{,}0001 \le \varepsilon \le +0{,}0001$. Deze verzameling bevat één kliek: K_{666}.

C. Goffman, And what is your Erdös Number?, *American Mathematical Monthly,* 76 (1969) 791.

P. Erdös, On the fundamental problem of mathematics, *American Mathematical Monthly,* 79 (1972) 149-150.

12. Als redenen dat de voorgevels van de 17e eeuwse Amsterdamse huizen naar voren hellen (op de vlucht gebouwd zijn) wordt veelal een der hierna genoemden aangevoerd.
    (a) Bij de verstening van het oorspronkelijk houten huis werd de overkragende constructie daarvan min of meer aangehouden *(traditie)*.
    (b) De huizen waren zo beter bestand tegen hemelwaterdoorslag *(komfort)*.
    (c) Het aanzicht van een gevelwand voldoet zo beter in verband met de perspectivische vertekening *(estetisch)*.
    (d) De bouwwijze biedt meer gemak met hijsen *(praktisch)*.

De werkelijke reden zal echter meer van konstruktieve aard zijn: door de voorgevel aan de ankers te laten *hangen* ontstaat een stabielere *konstruktie*.