RA

stichting
mathematisch
centrum

REKENAFDELING RA NR 24/72 APRIL

P.M.B. VITÁNYI

CONTEXT-VARIABLE LINDENMAYER SYSTEMS AND SOME SIMPLE REGENERATIVE STRUCTURES

2e boerhaavestraat 49 amsterdam

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

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

Contents

0. Introduction

An automata theoretic model for developmental growth in filamenteous organisms has been proposed by Lindenmayer (1968). In a (k,l) L-System we rewrite every letter of a string simultaneously according to its context, consisting of the k left and l right letters. Here we shall introduce Context-Variable Lindenmayer-Systems, where a letter of a string is rewritten according to a selection of letters from that string. The criterion for the selection is an attribute of the letter concerned. These Systems will appear to be especially suited to model certain properties of "full-growthness" and "regeneration". The accompanying languages are called Context-Variable languages.

1. Context-Variable Lindenmayer Systems

Differences between Chomsky generative grammars and Lindenmayer Systems as language generators are:

(i) In the former one letter of a string is rewritten in each time step while in the latter all letters are rewritten simultaneously.

(ii) In the Chomsky approach only terminal strings are elements of the language while in L-Systems all strings derived are elements of the language, i.e. no distinction is made between terminal and non terminal letters.

The main feature that distinguishes Context-Variable L-Systems from (k,l) ones is that in Context-Variable L-Systems the relative place of the context of a letter can vary from time to time and from place to 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 full-grown size and structure, which, however, is not terminal. 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.

Def. 1.1. A Context-Variable Lindenmayer System or C-V L-System is a 3-tuple $G = <\sum, \delta, \sigma>$ such that

(i) The alphabet $\sum$ is a nonempty finite set and elements of $\sum$ are called letters.

(ii) The transition function δ maps strings $x \in \sum^+$ onto strings $y \in \sum^*$ such that each element b_j of y has a superscript $\tau_j \in I^*$, i.e.

$$\delta(a_1 a_2 \ldots a_n) = b_1^{\tau_1} b_2^{\tau_2} \ldots b_m^{\tau_m}$$

where

$$x = a_1 a_2 \ldots a_n$$

$$y = b_1^{\tau_1} b_2^{\tau_2} \ldots b_m^{\tau_m}$$

$$\tau_j = \rho_0^{(j)} \rho_1^{(j)} \ldots \rho_{n_j}^{(j)} \qquad 1 \leq j \leq m$$

with

$$a_i, b_j \in \sum$$

$$\rho_h^{(j)} \in I \qquad \text{with } 1 \leq i \leq n,\ 1 \leq j \leq m,\ 0 \leq h \leq n_j.$$

In the above definition δ is deterministic; the generalization to the non deterministic case is done in the obvious way. In this report we shall only be concerned with the deterministic case.

(iii) The axiom σ is a word over $\sum$, each letter possessing a superscript which is a string over I, i.e.

$$\sigma = a_1^{\tau_1} a_2^{\tau_2} \ldots a_m^{\tau_m}$$

where

$$\tau_j = \rho_0^{(j)} \rho_1^{(j)} \ldots \rho_{n_j}^{(j)} \in I^*$$

$$a_j \in \sum \qquad 1 \leq j \leq m.$$

We also call the axiom the initial description of the C-V L-System.

Remark 1. The superscript $\tau_j = \rho_0^{(j)}\rho_1^{(j)} \ldots \rho_{n_j}^{(j)}$ selects in string $b_1^{\cdot}b_2^{\cdot} \ldots b_j^{\cdot} \ldots b_m^{\cdot}$ the context $h(b_j)$ according to which b_j is going to be rewritten:

$$h(b_j) = b_{j+\rho_0^{(j)}}\, b_{j+\rho_1^{(j)}} \ldots b_{j+\rho_{n_j}^{(j)}} \ .$$

If $j + \rho_i^{(j)} \leq 0$ or if $j + \rho_i^{(j)} > m$ we substitute the empty word λ for $b_{j+\rho_i^{(j)}}$ in $h(b_j)$. We will henceforth assume that $\rho_0^{(j)} = 0$ and omit $\rho_0^{(j)}$ from the superscript of b_j .

The C-V L-System generates words as follows:
Let $x = a_1^{\tau_1} a_2^{\tau_2} \ldots a_m^{\tau_m}$ be a string. Then x generates y directly, written as $x \Rightarrow y$, if

$$x = a_1^{\tau_1} a_2^{\tau_2} \ldots a_m^{\tau_m}$$

$$y = \alpha_1\alpha_2 \ldots \alpha_m$$

and for every j, $1 \leq j \leq m$,

$$\alpha_j = \delta(a_j\, a_{j+\rho_1^{(j)}}\, a_{j+\rho_2^{(j)}} \ldots a_{j+\rho_{n_j}^{(j)}})$$

with

$$\rho_1^{(j)}\rho_2^{(j)} \ldots \rho_{n_j}^{(j)} = \tau_j \ .$$

$\overset{*}{\Rightarrow}$ denotes the reflexive and transitive closure of $\Longrightarrow$ and $x \overset{*(k)}{\Longrightarrow} y$ 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 produces, generates or derives y, and if $x \overset{*(k)}{\Longrightarrow} y$ then y is derived in k steps from x and $x \overset{*(k)}{\Longrightarrow} y$ is a k-derivation of y from x. A string $x = a_1^{\tau_1} a_2^{\tau_2} \ldots a_m^{\tau_m}$ is called a description, and an element of x is called a cell.

Def. 1.2. A C-V L-Language is a set $L(G) \subseteq \sum^*$ where

$$L(G) = \{a_1 \dots a_n \mid \sigma \overset{*}{\Longrightarrow} a_1^{\cdot} \dots a_n^{\cdot}\} .$$

Example 1

Let

$$G = \langle\{a\}, \{a \to a^{-1}a^{+1}, aa \to \lambda\}, a\rangle .$$

Then

$$a \Longrightarrow a^{-1}a^{+1} \Longrightarrow a^{-1}a^{+1}a^{-1}a^{+1} \Longrightarrow a^{-1}a^{+1}a^{-1}a^{+1}$$

$$\Longrightarrow \dots \Longrightarrow a^{-1}a^{+1}a^{-1}a^{+1} \Longrightarrow \dots .$$

We notice that when the description has reached a certain full-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 $G(k)$ shall 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.

$L^a(0) = \{\lambda, a\}$

$L^a(-1) = \{\lambda, a, aa\}$

$L^a(-2) = \{a, aa, aaaa\}$

$L^a(-3) = L^a(-2)$

etc.

We describe the general form of an $L^a(k)$-language by:

Theorem 1. Let $G(k)$ and $L^a(k)$ be as above.

a) For $k > 0$ and k is even

$$L^a(k) = \{a^{2^t} \mid 0 \leq t \leq {}^2\log(k) + 1\} \cup \{a^{2k}\}.$$

For $k > 0$ and k is odd

$$L^a(k) = L^a(k+1).$$

b) For $k < -1$

$$L^a(k) = L^a(-k-1).$$

c) $L^a(0) = \{\lambda, a\}$

$L^a(-1) = \{\lambda, a, aa\}$.

Proof. By $\delta^t(a)$ we mean $a_1^{\cdot} a_2^{\cdot} \ldots a_n^{\cdot}$ if

$$a \xRightarrow{*(t)} a_1^{\cdot} a_2^{\cdot} \ldots a_n^{\cdot} .$$

a) For $k > 0$

(i) $t \leq {}^2\log(k)$. $|\delta^t(a)| = 2^t \leq k$. *)

*) $|x|$ denotes the length of x.

There are no cells $a_i^{\pm k}$ in $\delta^t(a)$ such that production rule $aa \to \lambda$ is to be applied. Therefore all cells divide and $|\delta^{t+1}(a)| = 2^{t+1}$.

(ii) $\quad {}^2\log(k) < t \leq {}^2\log(k) + 1$.

For all cells a_{2i}^{+k} and $a_{2^t-2i+1}^{-k}$ $(i>0)$, 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 $|\delta^{t+1}(a)| = 2k+2$.

(iii) $\quad t > {}^2\log(k) + 1$.

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 $|\delta^{t+1}(a)| = 2k+2$.

b) is proven in a similar way as a).

c) follows from the productions.

<u>Corollary</u> $\quad \bigcup_{k\in I} L^a(k) = \{a^{4n} \mid n \geq 0\} \cup \{a, aa\}$.

The C-V 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 full-grown size is reached but for one move. Next the C-V L-System grows on the remainder and stays at the same size and structure, although at each generation 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 (full-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 full-grown stage. This is reminiscent of the remarkable regenerative properties of flatworms. The discussed C-V 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 C-V L-Systems, e.g. with more cellular states. Does there exist a complexity bound, e.g. expressed in the size of the alphabet (and presumably δ), above which only partial regeneration is possible?
We may qualify questions like this by distinguishing several kinds of regeneration, e.g.

(i) Starting with one cell in a special state, i.e. reproduction.

(ii) Starting from arbitrary parts of a full 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 full grown description, etcetera.

Note that there is a difference between cases where we remove an end part of a full-grown description, and cases where we remove a middle part. We illustrate this with the following example (k=2).
The full grown description is:

$$a^{-2}a^{+2}a^{-2}a^{+2}.$$

Regeneration with the left-end (skin) cell removed:

$$a^{+2}a^{-2}a^{+2} \Longrightarrow a^{-2}a^{+2}a^{-2}a^{+2}.$$

The two cells right have divided, while the new leftmost cell has disappeared in the production. Regeneration with the third (middle) cell removed:

$$a^{-2}a^{+2}a^{+2} \Longrightarrow a^{-2}a^{+2}a^{-2}a^{+2}a^{-2}a^{+2} \Longrightarrow a^{-2}a^{+2}a^{-2}a^{+2} .$$

All three cells divide in the first production. In the second production only the two outermost cells divide and the others disappear: the full-grown size is reached.

We observe that the removal of different parts of the full-grown description may yield different courses for the regenerative process. The above is suggestive of biological interpretations like the surrounding of a wound by wound-tissue which is discarded after the healing process has been completed.

In the appendix we shall consider some closure (or rather non-closure) properties of $L^a(k)$ languages, so as to get an insight into what place the considered structures take with respect to the other language generating devices.

2. The Extended French Flag Problem

Usually the French Flag problem is states 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 full-grown stage, no cell staying static, and furthermore, when we chop off a piece of this structure regrow the missing piece until the full-grown stage has been reached again.

When we discuss the French Flag in this context what we want is:

(i) One cell divides and gives rise to a full-grown French Flag of a certain size which retains the same pattern and structure while individual cells are disappearing and dividing all the time.

(ii) When we chop off a piece of the full-grown French Flag it regrows the missing piece.

We will present a C-V L-System which does (i) and (ii).

As the system has to reach a certain full-grown size, clearly the production rules depend on this size. When we want a different full-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 such a kind of partial regeneration has not to be justified further. We may point out that "higher" organisms which are more differentiated mostly lose regenerative properties to a certain extend which seems to be the price to be paid for a more complex structure.
(Is there a maximal number of letters above which unlimited regeneration is not possible anymore? What about other types of regeneration?)
We shall exhibit an example of a Context Variable Lindenmayer System with maximal a two neighbor context, which, starting from a single cell attains a full-grown description, i.e. the French Flag

$$a^{\circ}a^{\circ}a^{\circ}a^{\circ}b^{\circ}b^{\circ}b^{\circ}b^{\circ}c^{\circ}c^{\circ}c^{\circ}c^{\circ} \;.$$

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 $a^{\circ}a^{\circ}a^{\circ}a^{\circ}$ the <u>head</u>, $b^{\circ}b^{\circ}b^{\circ}b^{\circ}$ the <u>trunk</u> and $c^{\circ}c^{\circ}c^{\circ}c^{\circ}$ the <u>tail</u> of the French Flag.

$\Sigma = \{a, b, c\}$. The transition function is specified by the following rules (we only write those we need and leave the others open):

$$a \rightarrow a^{-1+1}b^{-1+1}$$

$$b \rightarrow b^{-1+1}c^{+1-1}$$

$$c \rightarrow c^{+1-1}c^{+1-1}$$

$aa \to \lambda$

$bb \to \lambda$

$cc \to c^{+1-1}c^{+1-1}$

$ab \to a^{+2+1}a^{-1+1}$

$bc \to b^{+2+1}b^{-1+1}$

$ba \to b^{-1+1}c^{+1-1}$

$cb \to c^{+1-1}c^{+1-1}$

$aaa \to a^{-1+1}a^{-2+2}$

$bbb \to b^{-1+1}b^{-2+2}$

$ccc \to \lambda$

$aab \to a^{-1+1}a^{-1+1}$

$bbc \to b^{-1+1}b^{-1+1}$

$aba \to a^{-1+1}a^{-2+2}$

$bcb \to b^{-1+1}b^{-2+2}$

$cbc \to c^{+1-1}c^{+1-1}$

$bac \to b^{+2+1}b^{-1+1}$

$ccb \to c^{+1-1}c^{+1-1}$

$bab \to \lambda$

$bcc \to \lambda$

$cbb \to \lambda$

$cba \to \lambda$

$acb \to \lambda$

Starting from axiom a we obtain the following production:

$$(1)\quad a \Longrightarrow a^{-1+1}b^{-1+1} \Longrightarrow a^{+2+1}a^{-1+1}b^{-1+1}c^{+1-1}$$

$$\Longrightarrow a^{-1+1}a^{-2+2}a^{-1+1}a^{-1+1}b^{+2+1}b^{-1+1}c^{+1-1}c^{+1-1}$$

$$\Longrightarrow a^{-1+1}a^{-2+2}a^{-1+1}a^{-1+1}b^{-1+1}b^{-2+2}b^{-1+1}b^{-1+1}c^{+1-1}c^{+1-1}c^{+1-1}c^{+1-1}$$

$$\Longrightarrow a^{-1+1}a^{-2+2}a^{-1+1}a^{-1+1}b^{-1+1}b^{-2+2}b^{-1+1}b^{-1+1}c^{+1-1}c^{+1-1}c^{+1-1}c^{+1-1}$$

$$\Longrightarrow \text{idem.}$$

We call this full-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 $a^{-1+1}a^{-2+2}a^{-1+1}a^{-1+1}b^{-1+1}$.)

$$(2.1)\quad a^{-1+1} \Longrightarrow a^{-1+1}b^{-1+1} \overset{*}{\Longrightarrow} \text{FF by (1)}$$

$$(2.2)\quad a^{-1+1}a^{-2+2} \Longrightarrow a^{-1+1}b^{-1+1} \overset{*}{\Longrightarrow} \text{FF by (1)}$$

$$(2.3)\quad a^{-1+1}a^{-2+2}a^{-1+1} \Longrightarrow a^{-1+1}b^{-1+1} \overset{*}{\Longrightarrow} \text{FF by (1)}$$

$$(2.4)\quad a^{-1+1}a^{-2+2}a^{-1+1}a^{-1+1} \Longrightarrow a^{-1+1}a^{-2+2} \overset{*}{\Longrightarrow} \text{FF by (2.2)}$$

$$(2.5)\quad a^4b^{-1+1} \Longrightarrow a^4b^{-1+1}c^{+1-1} \Longrightarrow a^4b^{+2+1}b^{-1+1}c^{+1-1}c^{+1-1} \Longrightarrow \text{FF}$$

$$(2.6)\quad a^4b^{-1+1}b^{-2+2} \Longrightarrow a^4b^{-1+1}c^{+1-1} \overset{*}{\Longrightarrow} \text{FF by (2.5)}$$

$$(2.7)\quad a^4b^{-1+1}b^{-2+2}b^{-1+1} \Longrightarrow a^4b^{-1+1}c^{+1-1} \overset{*}{\Longrightarrow} \text{FF by (2.5)}$$

(2.8) $a^4 b^{-1+1} b^{-2+2} b^{-1+1} b^{-1+1} \Longrightarrow a^4 b^{-1+1} b^{-2+2} \overset{*}{\Longrightarrow}$ FF by (2.6)

(2.9) $a^4 b^4 c^{+1-1} \Longrightarrow a^4 b^4 c^{+1-1} c^{+1-1} \Longrightarrow$ FF

(2.10) $a^4 b^4 c^{+1-1} c^{+1-1} \Longrightarrow$ FF

(2.11) $a^4 b^4 c^{+1-1} c^{+1-1} c^{+1-1} \Longrightarrow$ FF.

Hence all left parts regenerate completely.
The reader may verify that the full-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) $a^3 b^4 c^4 \Longrightarrow$ FF

(3.2) $a^2 b^4 c^4 \Longrightarrow a^2 b^4 c^4$

(3.3) $a\ b^4 c^4 \overset{*}{\Longrightarrow}$ FF

(3.4) $b^4 c^4 \Longrightarrow b^4 c^4$

(3.5) $b^3 c^4 \Longrightarrow b^4 c^4$

(3.6) $b^2 c^4 \Longrightarrow b^2 c^4$

(3.7) $b\ c^4 \overset{*}{\Longrightarrow} b^4 c^4$

(3.8) $c^4 \Longrightarrow c^4$

(3.9) $c^3 \Longrightarrow c^4$

(3.10) $c^2 \Longrightarrow c^4$

(3.11) $c \overset{*}{\Longrightarrow} c^4$.

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) $a^{-1+1}a^{-1+1} \Longrightarrow \lambda$

(ii) $a^{-1+1}a^{-1+1}\eta \overset{*}{\Longrightarrow} a^2b^4c^4$ for $\eta \neq \lambda$.

(4.2) Every part of FF containing cells of the trunk but no head cells grows to a full-grown description b^4c^4, except parts of the form

$$b^{-1+1}b^{-1+1}\eta$$

(i) $b^{-1+1}b^{-1+1} \Longrightarrow \lambda$

(ii) $b^{-1+1}b^{-1+1}\eta \overset{*}{\Longrightarrow} b^2c^4$ for $\eta \neq \lambda$.

(4.3) Every part of FF consisting of tail cells grows to a full tail c^4, i.e. a full-grown description.

3. Open Problems

Def. 3.1. A C-V L-System $G = \langle\sum, \delta, \sigma\rangle$ stabilizes at ω if ω is the full-grown description of G.

A C-V production scheme is a pair $S = \langle\sum, \delta\rangle$.

Def. 3.2. A C-V production scheme $S = \langle\sum, \delta\rangle$ stabilizes at $\omega \in \sum^*$ if for all $\sigma \in \sum^*$ the C-V L-System $G = \langle\sum, \delta, \sigma\rangle$ stabilizes at ω.

1. Given an $\omega \in \sum^*$, does there always exist a C-V production scheme that stabilizes at ω. Find an algorithm which produces such a C-V production scheme.

2. If the answer to 1 is begative in general, then characterize the class (or a sub-class) for which the answer is positive.

A sub-class as meant in 2 is e.g.

$$\{a^{4n} \mid n \geq 0\}.$$

By example 1 G(k), k is an even natural number, stabilizes at a^{2k} for every axiom $\sigma \in \{a\}^*$.

3. Given $\omega \in \Sigma^*$, does there always exist a C-V L-System $G = \langle\Sigma', \delta, a\rangle$, $\Sigma' \supseteq \Sigma$ and $a \in \Sigma$, such that G stabilizes at ω. Give an algorithm to obtain such a G. (Can every word be generated by a C-V L-System with a one letter axiom such that the word is a full-grown description of that C-V L-System.)

4. If the answer to 3 is negative, then characterize the class (or a sub-class) for which the answer is positive.

Again, $\{a^{4n} \mid n \geq 0\}$ is such a sub-class.

5. Given $\omega \in \Sigma^*$, does there always exist a production scheme $S = \langle\Sigma', \delta\rangle$, $\Sigma' \supseteq \Sigma$, such that for all σ, $\omega = \eta\sigma\xi$, $G = \langle\Sigma', \delta, \sigma\rangle$ stabilizes at ω. (Is universal regeneration possible for every word?) If not, characterize the class (or a sub-class) for which the answer is positive.

$\{a^{4n} \mid n \geq 0\}$ is such a sub-class.

One criterion for the finiteness of C-V L-Languages is whether the produced description ever stabilizes.

6. Can we indicate conditions under which a C-V L-System stabilizes.

Appendix

Theorem 2. The family of $L^a(k)$ languages is not closed under (i) complementation, (ii) union, (iii) Kleenean star (*), (iv) Kleenean cross (+), (v) concatenation, (vi) intersection with regular sets; but it is closed under (vii) intersection.

Proof.

(i) $\{a\}^* \setminus L^a(1)$ contains aaa, and aaa $\notin \bigcup_{k\in I} L^a(k)$.

(ii) $L^a(6) \cup L^a(10) = \{a^n \mid n = 1,2,4,8,12,16,20\}$. From theorem 1 follows $L^a(k) \neq L^a(6) \cup L^a(10)$ for all k.

(iii) $L^a(k)^*$ contains aaa and aaa $\notin \bigcup_{k\in I} L^a(k)$.

(iv) as (iii).

(v) $L(1) \,.\, L(1) = \{a^n \mid n = 2,3,4,5,6,8\} \neq L^a(k)$ for all k.

(vi) $L^a(1) \cap \{aaaa\} = \{aaaa\} \neq L^a(k)$ for all k.

(vii) $L^a(k_1) \cap L^a(k_2) = \{a^{2^t} \mid 0 \leq t \leq \min({}^2\log(k_1), {}^2\log(k_2)) + 1\} = L^a(k)$ for $k = \max\{2^{t-1} \mid 2^t \leq \min(2k_1, 2k_2)\}$.

Lemma 3. The family of $L^a(k)$ languages is strictly contained in the family of regular languages over a one letter alphabet.

Proof. All $L^a(k)$ languages are finite.

Lemma 4. The intersection of the family of $L^a(k)$ languages with the family of OL-languages [Rozenberg & Doucet, 1971] consists of those $L^a(k)$ languages for which $L^a(k) \neq L^a(-k-1)$, viz. $\{L^a(0), L^a(-1)\}$.

Proof. Consider the following OL-Systems:

$$S_1 = <\{a\}, \{a \to a, a \to \lambda\}, a^h>$$

then

$$L(S_1) = \{a^t \mid 0 \leq t \leq h\}$$

and

$$L(S_1) = L^a(-1) \qquad \text{for } h = 2$$

$$L(S_1) = L^a(0) \qquad \text{for } h = 1.$$

If h > 2 then

$$a^3 \in L(S_1) \quad \text{and} \quad a^3 \notin L^a(h) \qquad \text{for all } h.$$

$$S_2 = <\{a\}, \{a \to a\}, a^h>$$

$$L(S_2) = \{a^h\} \text{ and } |L(S)| = 1 \neq |L^a(k)| \quad \text{for all } k.$$

$$S_3 = <\{a\}, \{a \to \lambda\}, a^h>$$

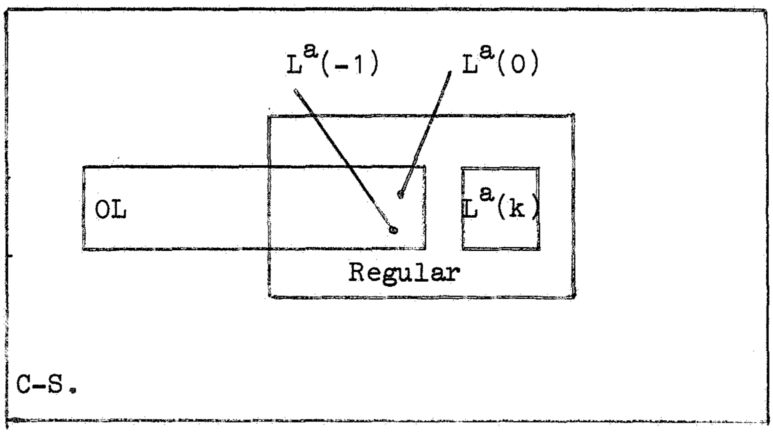
$$L(S_3) = \{a^h, \lambda\} = L^a(0) \qquad \text{for } h = 1.$$

$$L(S_3) \neq L^a(0) \qquad \text{for all } h > 1$$

and

$$|L(S_3)| = 2 \neq L^a(k) \qquad \text{for } k \neq 0.$$

All other OL-Systems over one letter alphabet have a production $a \to a^x$ where $x > 1$, and therefore generate an infinite language.



<u>Remark</u>. $L^a(k)$ languages are finite (containing usually more than 2 elements) and are generated in a deterministic fashion. It is not possible to generate finite languages containing more than two elements deterministically by either formal grammars or 0L-Systems.

References

G.T. Herman, Polar organisms with apolar individual cells, in: IV Int. Congress for Logic, Methodology and Philosophy of Science, 1971 (in press).

A. Lindenmayer, Mathematical models for cellular interaction in development I & II, J. Theoret. Biol. 18(1968), 280-315.

G. Rozenberg & P.G. Doucet, On OL-languages, Information & Control 19(1971), 302-318.