

A MODEL STUDY ON BIOMORPHOLOGICAL DESCRIPTION

P. HOGEWEG

Theoretical Biology Group, University of Utrecht, Heidelberglaan 2, Utrecht, Netherlands
and

B. HESPER

Central Computing Institute, University of Leiden, Leiden, Netherlands

(Received 10 July 1973)

Abstract—Description of the morphology of organisms is mostly done in terms of features assuming certain states. Biological patterns are subjected to certain constraints imposed on them by the fact that they all are developed from a single cell. Using a model in which some of the constraints of biological development are incorporated we investigate the relation between different modes of description of such patterns.

Pattern generation Pattern recognition Numerical taxonomy Parallel rewriting systems
L-systems

I. INTRODUCTION

Organisms show for each species characteristic, patterns in which cells and regions of different types occur. Much biological work is done on the description of these patterns: in taxonomy and morphology the patterns are described and classified, in developmental biology sequences of patterns occurring during the development are described, while in evolutionary biology observed patterns are ordered into sequences in such a way that the transitions seem small. In all these areas of research, patterns are described in terms of "features" (subpatterns) assuming certain "states". Given this description, classification or ordering may be done either intuitively, selecting a few important features (classical taxonomy), or by cluster analysis or multidimensional scaling techniques where selection of features is less severe (numerical taxonomy). Nevertheless, in all cases the selection of features remains crucial. Moreover, the recognition of these features is sometimes difficult, and it is not clear which subpatterns should be compared (homology problem).

Patterns of multicellular organisms are subjected to some obvious constraints. All are produced by repeated cell division, via a sequence of patterns, from a single cell. All cells contain the same genetic material. Obviously the behaviour of the cells (division or differentiation) can be controlled only by local conditions working on the genetic material, not by the current global form of the organism.

In this paper we try to gain insight in the relations between global patterns and their morphological de-

scription in terms of subpatterns and the description of the local transformations [controlled by local (non morphological) conditions], by which the patterns are formed. These relations seem important for the methodology of pattern recognition in biology: often morphological similarities are interpreted as representing similarities on the genetic level. Moreover, we try to obtain some feeling about "small" transitions between morphological patterns under the stated constraints; small transitions between successive stages in development and those caused by small changes in the rules specifying the development (studied by developmental biology and evolutionary theory respectively).

We do this by studying a set of generative systems, defined on the basis of the above named characteristics of biological systems and the sequences of patterns generated by these systems.

Thus we obtain the following modes of description of the set of patterns:

- (1) The generative system (a set of rules specifying local transformations).
- (2) String representations of the sequences of patterns generated by these rules.
- (3) Pictorial representations of the strings (as three-dimensional branching patterns).
- (4) Description of the pictorial representations in terms of morphological features and the classification on the basis of these features.
- (5) Description of the sequences of the strings and the pictorial representations in terms of recurrence relations.

It should be stressed that we do not consider these generative systems as realistic models for biological development: they only serve to investigate pattern description and pattern recognition of patterns subjected to some of the most obvious constraints of biological development. Biological development may be subjected to more constraints and the simplified two-state system we used is obviously not realized in biological systems.

II. THE GENERATIVE SYSTEMS

We used for our experiments the generative systems proposed by Lindenmayer^(1,2) (and later called *L*-systems by van Dalen⁽³⁾) for the description of the sequence of developmental stages of growing filamentous organisms.

The systems may be described in terms of arrays of simultaneously operating, interacting cellular automata or in terms of growing strings of symbols. The description in terms of cellular automata suggests strongly the analogy with cellular organisms. In this paper we will use the linguistic terminology. The main characteristics of these generative systems, as compared to Chomsky's generative grammars, are:

(1) The production rules are applied to all symbols simultaneously.

(2) No terminal alphabet is defined, since the emphasis lies on the sequence of strings formed and not on alternative strings developed by the same grammar. In these systems we denote by the word "language" all strings formed from a given axiom, by applying the production rules any number of times and not only those strings which contain terminal symbols only.

(3) The starting string (axiom) is contained in the alphabet (Rozenberg and Doucet).⁽⁴⁾

Lindenmayer^(1,2) introduced three different types of *L*-systems, characterized by the influence of the context on the productions. In 0-*L*-systems the production rules are such that each symbol is to be transformed independently of the neighbouring symbols (context); in 1-*L* and 2-*L* systems the production rules are such that each symbol is to be transformed according to its one or two-sided context respectively. The context is locally defined by the neighbouring symbol(s) or, when there is no neighbouring alphabetic symbol, it is defined by end symbols representing the environment. The simultaneity of the productions introduces a non localness in the systems. Formal definitions of *L*-systems are given by van Dalen,⁽³⁾ Herman^(5,6) and Rozenberg and Doucet.⁽⁴⁾

The systems are called propagating if the empty string is not generated by any of the production rules.

For the description of branching structures Lindenmayer⁽²⁾ introduced brackets in the *L*-systems. In the bracketed deterministic *L*-systems which we have used, all the production rules have the forms: (1) $\delta(x.a.y) = b[c]$; (2) $\delta(x.a.y) = bc$; or (3) $\delta(x.a.y) = b$, where *a*, *b*, *c* are symbols of the alphabet, *x* is the left and *y* is the right context (possibly unspecified as in 0-*L* and 1-*L* systems). The brackets are terminal symbols and once introduced are included in all subsequent strings. The brackets are defined as changing the context of the symbols. In 0-*L* systems the presence of brackets has no influence on the subsequent development of a string of alphabetic symbols.

The rules that define the context in bracketed, propagating deterministic 1-*L* and 2-*L* systems (bracketed *PDi-L* systems; *i* = 1,2) are as follows: (see Fig. 1).

(1) When the left or right neighbouring symbol is an alphabetic symbol, this symbol defines the context (as is the case in non-bracketed *iL* systems).

(2) When the left neighbouring symbol is an opening bracket, the leftsided context is defined by the first alphabetic symbol in the string towards the left, which is separated from this opening bracket by an equal number (possibly 0) of opening and closing brackets.

(3) When the left neighbouring symbol is a closing bracket, the leftsided context is defined by the first alphabetic symbol towards the left which is separated from the symbol by an equal number of opening and closing brackets (including the neighbouring closing bracket).

Moreover, in case of a 2-*L* system:

(4) When the right neighbouring symbol is a closing bracket, the rightsided context is defined by the symbol at the end of the string, which represents the environment.

(5) When the right-sided symbol is an opening bracket, the rightsided context is defined by the first alphabetic symbol towards the right, which is separated from the symbol by an equal number of opening and closing brackets (including the neighbouring closing bracket).* (N.b. "left" and "right" are introduced by convention and could be interchanged. The form of some production rules then becomes: $\delta(x.a.y) = [b]c$.)

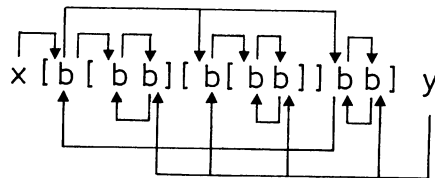


Fig. 1.

* For a formal definition of bracketed *i-L* systems, see Lindenmayer.⁽²⁾

The effect of the above definition is that each substring within brackets is "lifted out of the string". In the branching interpretation of the strings, the symbols which define the context are actually neighbouring symbols: the substrings between brackets are depicted as side-branches attached to the symbol as defined in (2), sticking out in the environment (4) and the main branches are attached as defined in (3) and (5).

It should be noted that this notation for branching structures is one of many possible notations and differs for example from the notation which Rounds⁽⁷⁾ and Tatcher⁽⁸⁾ use in their "tree grammars". They write a pair of brackets around all symbols connected to the same symbol: neighbouring symbols are not connected to each other but to the same symbol. Our notation makes it possible to code: (1) a variable number of symbols attached to one symbol; (2) a variable number of symbols in one branch; (3) a variable number of orders of branches (i.e. branches on branches); (4) 2 different types of connections of symbols.

Type 1. Both symbols are between the same pair of brackets, and are each other's left and right context.

Type 2. The rightmost symbol between one more pair of brackets than the leftmost symbols. The left symbol serves as left-sided context for the right symbol. The right symbol does not serve as context for the left symbol.

The difference in context only exists in 2-L systems, since only the production rules of 2-L systems are defined on rightsided context. The 2 types of connections are depicted as branches (straight lines) and side branches (line under some angle) respectively.*†

Lindenmayer^(2,9) used some of them for the description of some specific branching structures: the branching structure of the alga *Calithamnion roseum* and of the moss *Phascum cuspidatum* (the latter not deterministic). In this paper we describe a subset of bracketed PD2L systems. The subset under consideration is defined as follows:

- (1) The alphabet consists of 2 symbols only (0 and 1).
- (2) The environmental context is provided by one of the alphabetic symbols (in this case "1"; (the environmental symbols, i.e. the symbols at the end positions, are not subjected to the production rules).
- (3) The starting string (axiom) is "1".
- (4) From (1) and (2) it follows that such systems are completely specified by $2^3 = 8$ production rules, since

* A restriction of the notation is that type 1 connection cannot be established after (during development) a type 2 connection is established on a symbol.

† Little work has been done on the above defined bracketed PD2L systems.

there are two different symbols with two different left- and two different right-sided contexts.

(5) The production rules transform a symbol into a string of alphabetic symbols of length one or two. Transformation into a string of two alphabetic symbols will be called "division". The two symbols are always the same in one system. There are two types of division: type 1—continuation of a branch; and type 2—initiation of a sidebranch. Thus the production rules for division are of the form $\delta(a,b,c) = "11"$ or $\delta(a,b,c) = "1[1]"$, where $a,b,c \in \{0,1\}$.

(6) In the complete specification of each system both types of division occur once and only once.

(7) The production rules for non-dividing transformations are permuted in all possible ways.

The set of languages under consideration contains all languages generated by all possible sets of 8 production rules under the above named constraints. There are $7 \times 8 \times 64 = 3584$ such languages. A computer-program (in PL/I) was developed for generating the sets of production rules (the generative systems) and their languages. The main program generates the complete sets of production rules when given the two division rules, by permutation of all possibilities of the other 6 rules. Accordingly one run produces 64 generative systems. The program calls a subroutine for generating the languages from every set of production rules. This subroutine can accept a broad range of generative systems and generates the corresponding language up to a given number of cycles (25 cycles were used for the experiments described here). One production cycle transforms all symbols of the string formed in the previous cycle.

III. PICTORIAL REPRESENTATION

Only part of the set of systems generate strings longer than two symbols. All strings generated after 25 production cycles which are longer than two symbols are represented as branching patterns. The patterns are conceived as branching patterns in three dimensions and are plotted as projections of these in two dimensions. The following conventions are used to form the branching patterns in three dimensions from the bracketed strings:

- (1) Substrings between the same pair of brackets form a branch.
- (2) A sidebranch is attached to that symbol of another branch which defines the left-sided context of its left-most symbol.
- (3) Branches grow straight, sidebranches leave their main branch under a given angle.
- (4) All sidebranches attached to the same symbol are spread evenly over 360° while the direction of the

first sidebranch at the symbol is defined according to a "spiraling coefficient".

(5) Alphabetic symbols are each represented by a line of given length, ended by a dot.

A computer program (in PL/I) was written for calculating the positions of the endpoints of the lines in three-dimensional space, and plotting them in projection. Method: first the string is scanned and a count is made of the number of branches at every alphabetic symbol. Then the string is scanned again and the pattern is plotted at the same time.

The three-dimensional positions are calculated by conceiving the lines as being in the z -direction of a wandering base of which the position relative to the starting base is always known. The "wandering" is initiated by the symbols in the following way:

(1) An alphabetic symbol shifts the base over a unit length in the z -direction and rotates it around the z -axis over the spiralling angle given by the user (all illustrations given here are made with a spiralling angle of 180°).

(2) An opening bracket initiates the rotating of the base around the y -axis over a given spreading angle (in the illustrations 45°) and stores the base position at the point of attachment in a push down store.

(3) A closing bracket reclaims the base position at the point of attachment of the branch by popping the push down store, and positions the base in that point.

The program is equipped to accept a delta factor for the unit length and the spreading angle, which is added to them after each symbol representation line is plotted (a paper by Honda⁽¹⁰⁾ reports on an investigation on the variability of global form of branching patterns when varying such factors). Alternatively an array of node lengths may be used (this option is useful when the program is used for depicting data structures by their spanning subtrees: a numerical value of the relationships may be depicted as node length).

The so defined branching patterns represent the strings only as far as the sequence of the brackets is concerned: the representations are independent of the type of alphabetic symbols in the string, and represent only the number of them between brackets. The representation is not entirely unique for all the strings under consideration; they are unique though for the languages when the sequences of strings are represented as sequences of branching patterns.

The pictures give an easily perceived representation of the structure and variability of the generated patterns. Regularities are easily discovered by plotting the sequences of patterns which constitute the language, or plotting from or up to a specified order branch separately (examples of sequences of patterns are shown in Figs. 3 and 8, separate first order branches in Fig. 5).

We have considered the branching patterns as pictorial representations of the strings. We may, however, also consider the strings as string representations of the branching patterns. The strings represent the branching patterns as far as the relative positions of the sidebranches and branches is concerned: neither the angles between sidebranches and their main branch, nor the angles between sidebranches are represented in the strings. Additional information may be included in the strings: when they are used to represent biological branching patterns, the type of cells may be represented by the type of alphabetic symbols (possibly influencing production rules which describe transformation of the patterns but not represented in the plotted pictures).

Comparing the relationship between the patterns and the transformations of the patterns (the strings and the generative systems in our work) with this relationship in related studies on heuristic pattern generation, we note the following: the production rules here employed are themselves not dependent on spatially interpretable terms: the context may be defined entirely in terms of descent of the symbols. This is in contrast to the work of, for example, Ulam⁽¹¹⁾ in 2 and 3 dimensions, and the work of Cohen⁽¹²⁾ on branching patterns.

IV. THE GENERATED PATTERNS

The set of generative systems gives rise to about 1000 patterns growing above the two-symbol stage; many of the systems produced identical strings. The end result consists of 150 morphologically different patterns (see Table 1).

Examples of the pictorial representations of the patterns after 25 generation cycles are given in Figs. 2-11.

The set of pictures show a remarkable amount of gross morphological variability and different types are easily recognized in the set. Conspicuous morphological features of the patterns are e.g.:

(1) elongated or rosette forming (Figs. 11 and 7 respectively). [This morphological feature coincides with one specific generation rule: $\delta(1.0,1) = 11$ gives rise to elongated patterns; $\delta(1.0,1) \neq 11$ gives rise to rosettes.]

(2) Branches with straight or bended aspect (Fig. 4)—this aspect is caused by sidebranches on the end symbols of the branches.

(3) Number of sidebranches which may occur on one symbol.

(4) Maximal length of branches (e.g. limited to one symbol, Fig. 2).

(5) Number of sidebranches which occur on a branch.

Moreover, the sequences of patterns show some interesting characteristics.

Table 1

Number	δ Functions of argument triples								Recurrence description of languages (if known) description in terms of pictorial representation	
	000	001	010	011	100	101	110	111		
1									1	$S(n) = S(0)$ no development
2									11	$Sa(n) = Sa(n - 1) \parallel Sa(n - 1)$ exponentially growing straight line
3									1[1]	$Sb(n) = Sb(n - 1) \parallel Sb(n - 1) \parallel$ rosette; $\max(l(n)) = 1$ for all order branches: $\max(s(n)) = n$
4.1						0			0	$S(n) = S(1)$ for all $n \geq 1$; $S(1) = [0]$ no development
4.2						1			0	$S(n) = S(0)$ iff $n = \text{even}$ $S(n) = S(1)$ iff $n = \text{odd}$; $S(1) = [0]$ no development
4.3							1[1]		0	$Sb(n) = Sb(n - 2) \parallel x(n)$, rosettes; $\max[s(n)] = \text{trunc}(1/2 n)$ (= on top of mainbranch); all first order sidebranches have same development
4.3.1		1					1[1]		0	$Sb(n) = Sb(n - 2) \parallel S(0)$ iff $n = \text{even}$, see Fig. 2(a) $Sb(n) = Sb(n - 2) \parallel S(1)$ iff $n = \text{odd}$. $S(1) = [0]$, first order sidebranches only; $\max l(n) = 1$
4.3.2.1		0		1			1[1]		0	$Sb(n) = Sb(n - 2) \parallel x(n)$, $\max l(n) = 1$, see Fig. 2(b) rate of branching on higher order branches not equal for all branches
4.3.2.2		0		0			1[1]		0	$Sb(n) = Sb(n - 2) \parallel Sb(n - 3)$, see Fig. 2(c), 2 $\max l(n) = 1$; rate of branching equal for all branches of all orders
4.3.3		11					1[1]		0	$Sb(n) = Sb(n - 2) \parallel x(n)$, $\max l(n)$ depends on n : maximum length occurs in first order branches; see Table 2
4.3.2.3		0		11			1[1]		0	$Sb(n) = Sb(n - 2) \parallel x(n)$ $\max l(n) = f(n)$ (max length in first order branches) see Table 3
4.4							11		0	No recurrence formula known elongated patterns (length of mainbranch depends on n) (for exception see Table 4)
4.4.1		1[1]					11		0	see Table 4
4.4.2				1[1]			11		0	see Table 5
4.4.3		otherwise					11		0	see Table 6

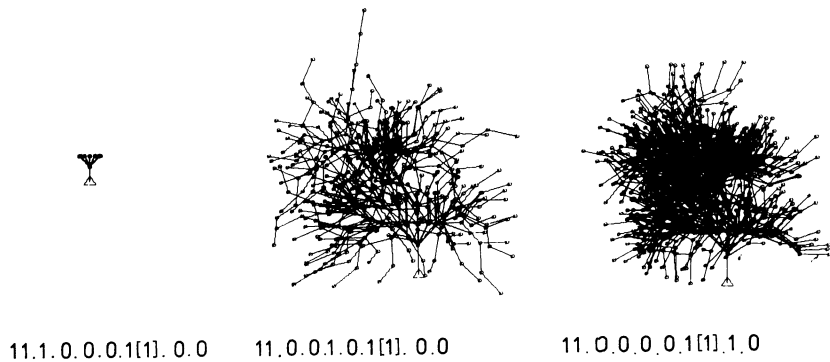


Fig. 2.

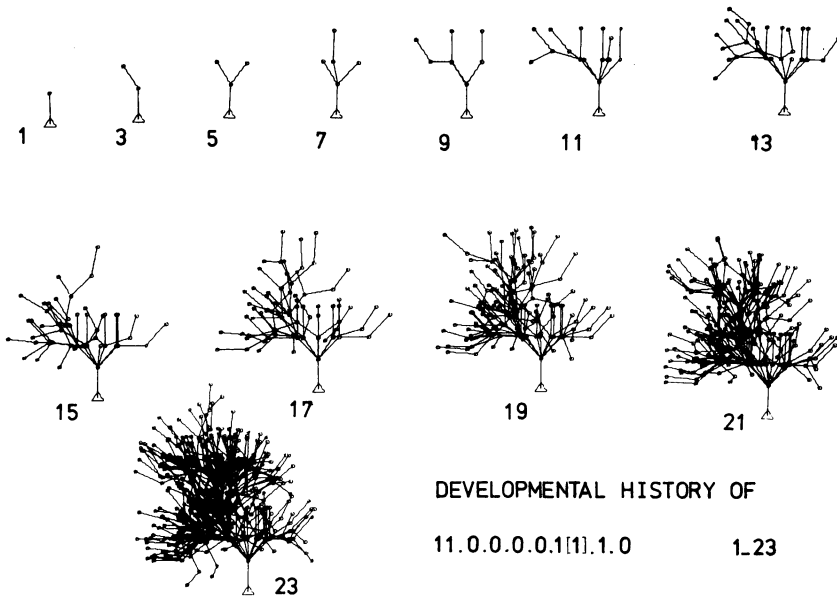


Fig. 3.

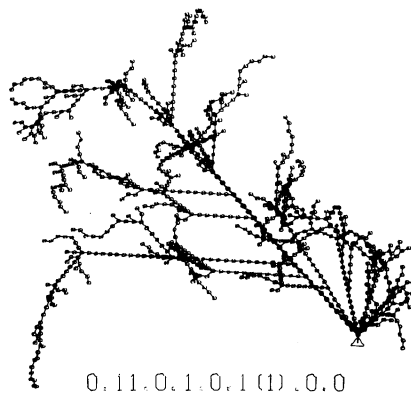
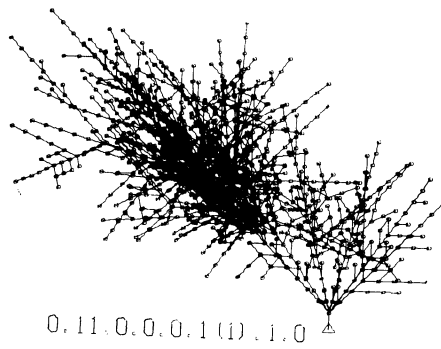


Fig. 4.

0.11.1.0.0.1(1).0.0

1e ORDER BRANCHES IN SEVERAL STAGES OF DEVELOPMENT

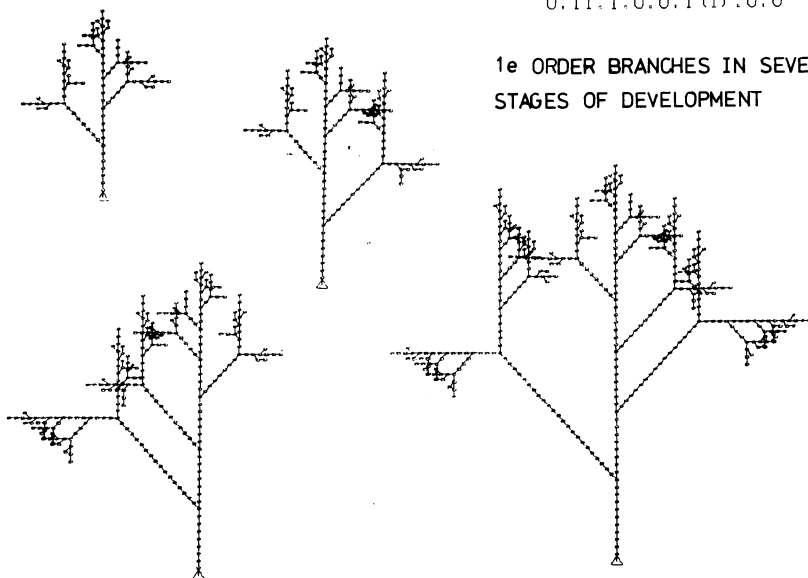
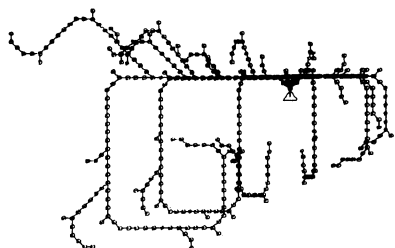
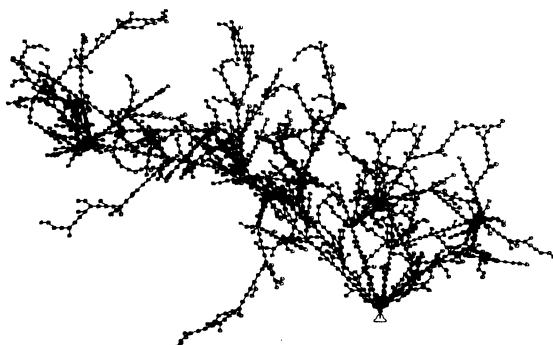


Fig. 5.

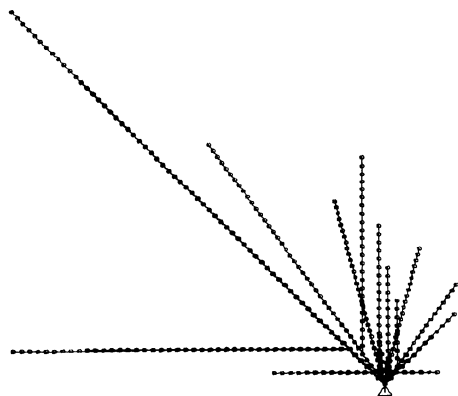


0.0.0.11.0.1(1).1.0

Fig. 6(a).

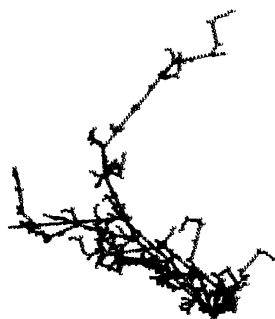


1.0.0.11.1.1(1).1.0



0.11.0.0.0.1(1).0.0

Fig. 6(b).



0.0.0.11.1.1(1).1.0

Fig. 7.

DEVELOPMENTAL HISTORY
OF 0.1[1].0.0.0.11.0.0.

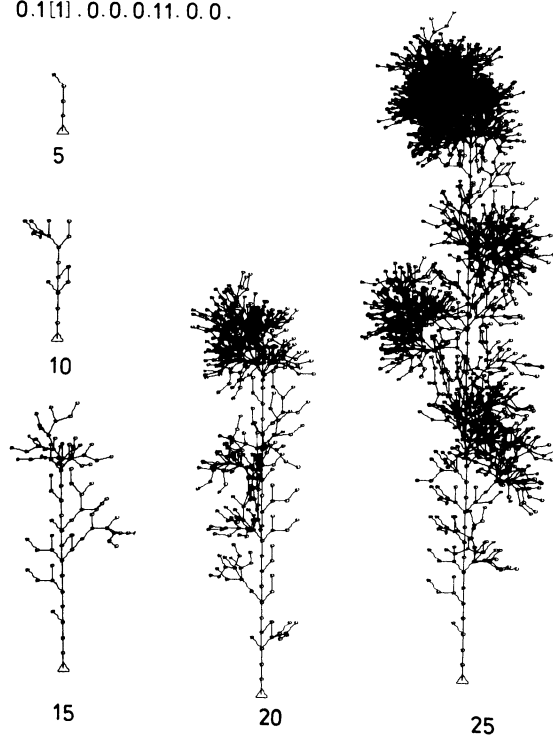


Fig. 8.

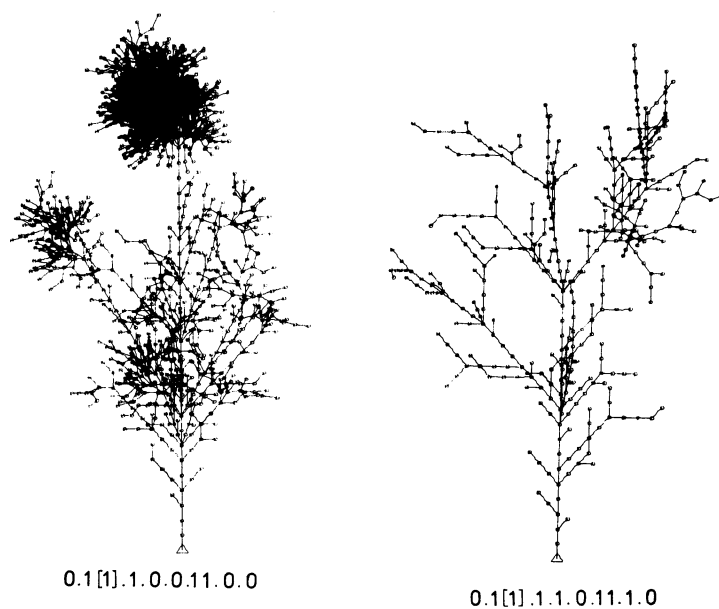


Fig. 9.

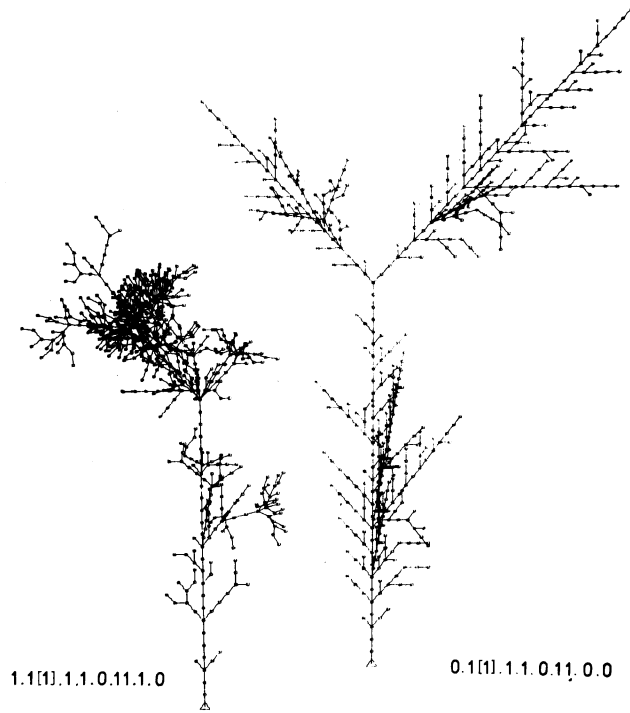


Fig. 10.

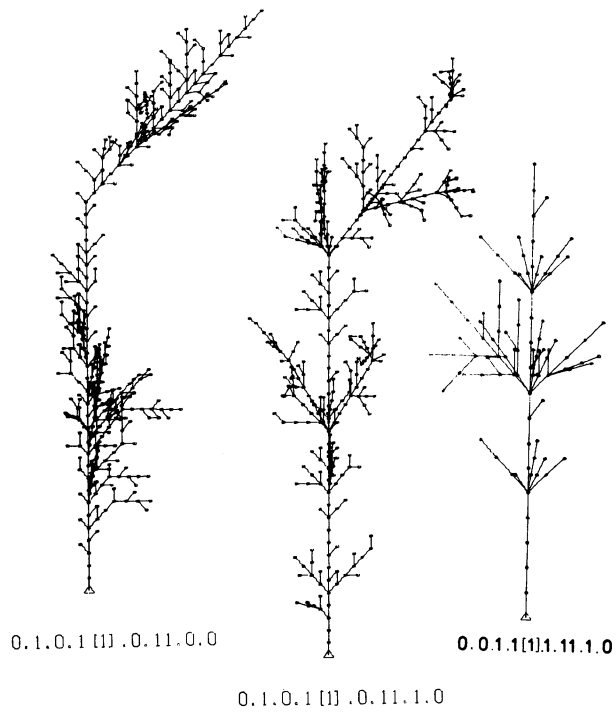


Fig. 11.

Table 2. Elaboration of case 4.3.3 in Table 1

Number	δ Functions of argument triples								Recurrence description of languages (if known) description in terms of pictorial representation
	000	001	010	011	100	101	110	111	
1		11				1[1]		0	$Sb(n) = Sb(n) x(n)$ $l(n,1) = f(n)$; elongated first order branches
1.1	1	11				1[1]		0	$Sb(n) = Sb(n-2) x(n) $; $x(n) = x(n-2) 1$ first order branches only; $\max l(n) = \text{trunc}(1/2n)$
1.2.1	0	11		1		1[1]		0	Clusters of sidebranches on top of the first order branches
1.2.1.1	0	11		1		1[1]	1	0	$s(25,1) = 8$, on top of first order branches
1.2.1.2	0	11		1		1[1]	0	0	$Sb(n) = S(n-2) 00 x(n)$ $x(n) = y(n) 11 x(n-2)$ $s(25,1) = 4$, on top of first order branches (see Fig. 4a)
1.2.1.3	0	11	1	1		1[1]		0	Branches of order > 1 may have sidebranches on last symbols
1.2.2	0	11		0		1[1]		0	First order branches straight (i.e. no sidebranches on last symbols)
1.2.2.1	0	11	0	0		1[1]		0	All order branches straight (see Fig. 4a)
1.2.2.1.1	0	11	0	0	0	1[1]	0	0	Maximum branching order: 2; only 2 sidebranches per first order branch, attached to second and sixth symbol from the base (see Fig. 6b)
1.2.2.1.2	0	11	0	0	1	1[1]		0	$Sb(n) = Sb(n-2) Sb(n-5) Sb(n-4)$ both patterns are exactly the same
1.2.2.2	0	11	1	0	0	1[1]	0	0	$Sb(n) = Sb(n-2) x(n)$ $x(n) = y(n) 11 x(n-2)$ $y(n) = [0000[1100 x(n-9) x(n-7)]$ all odd and all even order branches have the same development (see Fig. 5)

Table 3. Elaboration of case 4.3.2.3 of Table 1

Number	δ Functions of argument triples								Recurrence description of languages (if known) description in terms of pictorial representation
	000	001	010	011	100	101	110	111	
1				11		1[1]		0	$Sb(n) = Sb(n-2) x(n)$ first order branches elongated
1.1		1		11		1[1]		0	$Sb(n) = Sb(n-2) [1]$ first order sidebranches only. Number of sidebranches = $\text{trunc}(1/2n)$, their length = 1 (see Fig. 2a)
1.2		0		11		1[1]		0	First and higher order sidebranches present
1.2.1	0	0		11	0	1[1]		0	Length of first order branches = 2; on last symbol they bear one sidebranch only
1.2.1.1	0	0		11	0	1[1]	0	0	$Sb(n) = Sb(n-2) x(n)$ $x(n) = x(n-1) x(n-5) x(n-4)$ both patterns are equal; sidebranches along branches of order higher than one occur every second symbol, only one per symbol
1.2.1.2	0	0		11	0	1[1]	1	0	$Sb(n) = Sb(n-2) x(n)$ $x(n) = y(n) x(n-4)$ sidebranches along branches of order higher than one occur every fourth symbol, only one per symbol (see Fig. 6a)
1.2.2	otherwise								Elaborate rosette forming branching patterns (see Fig. 7)

(1) The patterns of the n th cycle may be composed of patterns of previous cycles. $S_{(n)} = S_{(m)} || S_{(l)}$ (see Fig. 2). The symbol “||” indicates concatenation of strings. In such cases we may speak of catenative systems. Rozenberg and Lindenmayer⁽¹³⁾ proved that all such patterns may be generated by PDOL grammars.

(2) The patterns of the n th cycle may be composed of subpatterns of previous cycles (see, e.g. Fig. 5, Table 2, 1.2.2.2.1.). This is a generalization of 1; together we will refer to them as “recurrence relation”. The formal structure of generalized recurrence relations are studied by Herman *et al.*⁽¹⁴⁾

(3) Subpatterns of previous patterns occur in the n th generation, but the n th generation cannot be described in terms of previous patterns; e.g. the top or basal parts of the patterns remain morphologically constant (see Fig. 11c).

(4) Subpatterns showing the same developments occur several times in one pattern (see, e.g. Fig. 8).

Tables 1–6 summarize the set of generated systems by three modes of description:

(1) The generative systems. These systems are partly specified in the tables. The stated descriptions apply to all systems (within one set under consideration) in which the specified rules occur.

(2) Gross morphological characteristics of the pictorial representation of the patterns.

(3) Recurrence relations in the sequences of the patterns (in terms of the strings).

Notation

Generative systems. In the first three tables the δ -function of the triples specified above each column are given in the column; when left blank, all permutation of the δ -function, within the constraints of the set of generative systems used, may be specified. In the other tables the δ -functions of all triples are ordered according to the number, in binary notation, formed by the

Table 4. Elaboration of case 4.4.1 of Table 1

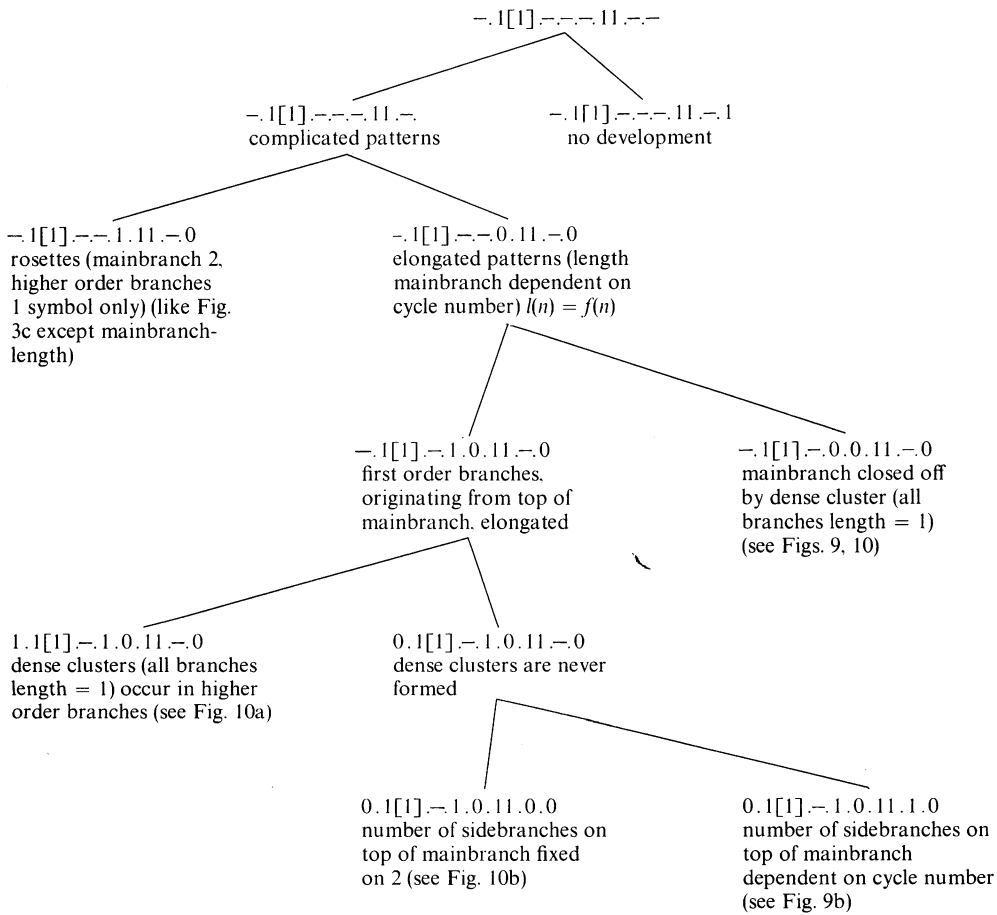


Table 5. Elaboration of case 4.4.2 of Table 1

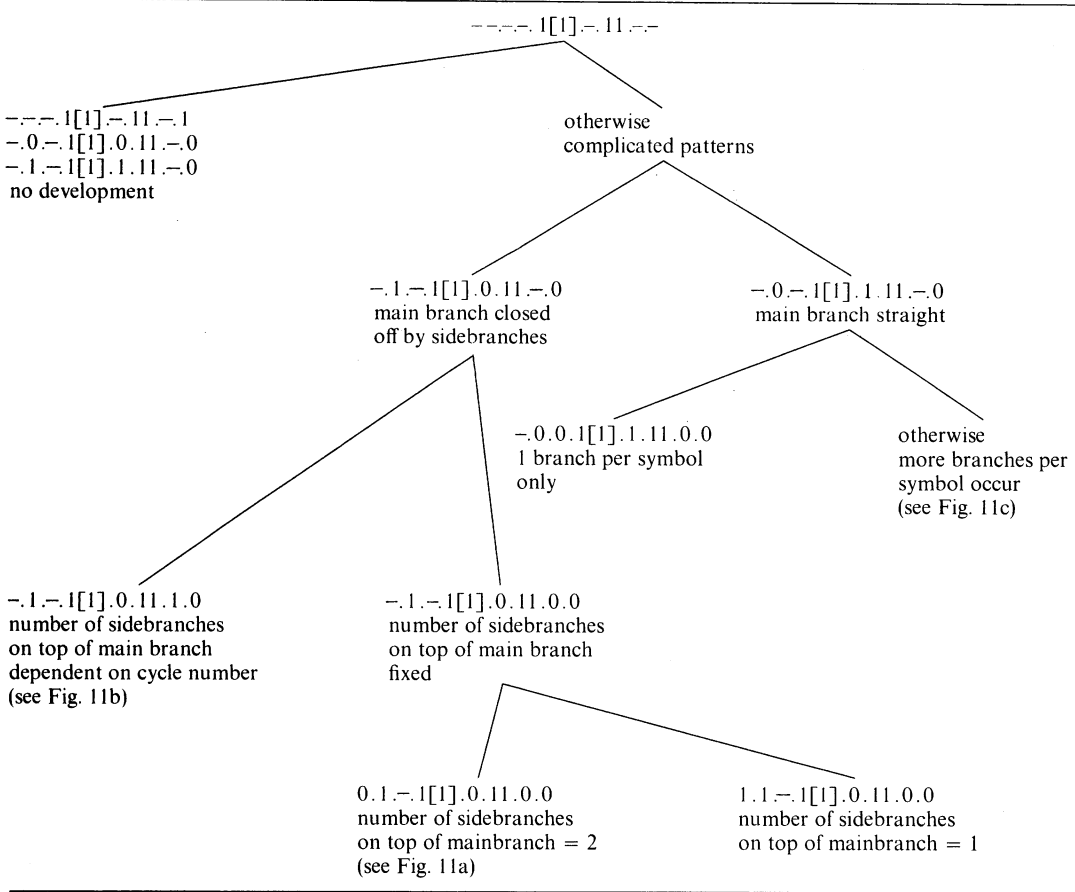
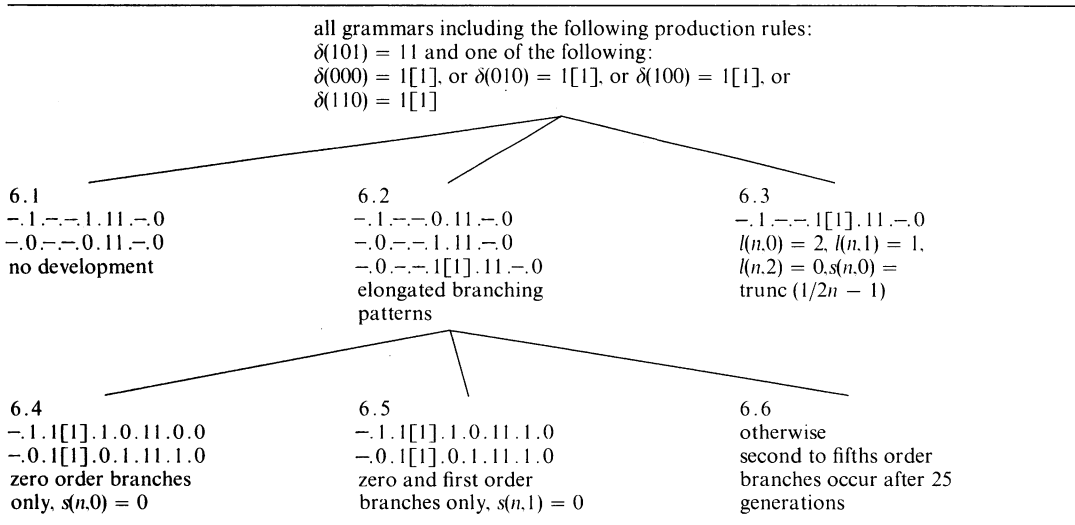


Table 6. Elaboration of case 4.4.3 of Table 1



triples, and are separated by a dot; non-specified δ -functions are indicated by “—”.

Strings. $S(n)$ the string formed by one generative system, from the axiom ($S(0)$) after n production cycles. For notational convenience we define: $Sa(n)$ the string $S(n)$ without both outermost brackets (one opening and one closing); $Sb(n)$ the string $S(n)$ without the left-most opening bracket; $x(n)$ denotes an arbitrary string at generation step n .

Pictorial representation. The terms as introduced in section III (branches, sidebranches, etc.) are used. Moreover, we define: $l(n,m)$ = length (= number of symbols) of branch (without sidebranches) of generation n and of order m ; n = cycle number: the given recurrence relations are given for all $n \geq 1$; $\max l(n,m)$ = maximum length of branches in generation n and of order m ; $s(n,m)$ = number of sidebranches per node in branch in order m and in generation n . $f(n)$ = a function of the cycle number; $\text{trunc}(x)$ = nearest lower integer to x for any number x ; the predicates $l(n,m)$ and $s(n,m)$ are also used as $l(n)$ and $s(n)$ whenever they apply to all order branches.

Furthermore: cluster = a node on which the number of sidebranches increases with n ; dense cluster = a cluster with $l = 1$ for branches of all orders; rosette = a cluster around the main branch which remains length 1; elongated pattern = a pattern of which the main branch increases with n .

V. DISCUSSION

In this section we point out some aspects of the results of our experiments with a set of generated patterns which seem to us to be of interest for the description of biological patterns as far as they may be considered to be entirely shaped by the constraints of a development which is local and is locally controlled. It should be noted that the morphological patterns are entirely defined by the generative systems and the conventions used in plotting; factors like influences through the substrate, gradients in the substrate etc. are not present in contrast to biological developing systems and models considered by others (Cohen⁽¹²⁾). If we compare, metaphorically, the role of the production rules in our systems with the role of the genetic instructions in organisms, we may see our branching patterns as an analogy of entirely genetically controlled biological systems.

1. Morphology

The generated patterns keep growing indefinitely in contrast to biological patterns. Also the fact that no spatial limitations are included in the systems gives rise to situations which are never observed in biological

systems (e.g. an unlimited number of side branches on one cell).

Nevertheless, close scrutiny of the pattern suggests several interesting analogies to biological systems. Especially interesting seems to us the occurrence of morphological stationary subpatterns. These subpatterns are composed of different symbols at every generation cycle but their position in the pattern is constant. This phenomenon is very common in organisms: organs are composed of a changing cell population and retain their form. Moreover, cells transplanted from other parts will assume the states of cells of the location to which they are transplanted. The occurrence of this phenomenon in our patterns, which are formed by very simple locally defined rule without explicit blocking mechanisms of the generative systems, suggests that its explanation may be much simpler than often thought.

2. Variability and classification of the set of patterns

The set of pictures show a remarkable amount of variability. Although all possible generative systems within the chosen set are considered, and no other constraints are introduced, the impression which the form variability makes on us is discontinuous. Comparing this with biological patterns we note that no environmental constraints are necessary to produce a set of patterns which show a “discontinuous” morphological variability.

The amount of morphological difference (intuitively defined) is not related to the number of differing production rules: the patterns of Fig. 2 are formed by a great variety of sets of production rules, while in other cases (see Figs. 7–10) single changes in production rules give rise to quite distinct patterns. The amount of morphological difference is often taken as a measure of the amount of genetic difference (because it is often the only obtainable measure, see Sokal and Sneath⁽¹⁵⁾) in this way obtaining a measure for phylogenetic relationship. A correlation between the amount of morphological and “genetic” difference is not found in our systems, in which no convergence is introduced by environmental constraints, in which case the above chain of reasoning is generally considered to be false.⁽¹⁵⁾

It is possible to obtain a monothetic, hierarchical classification on the morphological features of sets of our patterns. Within some subsets of the generated patterns, one can produce such a classification in which every fork is coincident with the specification of one production rule (see Table 4). For other subsets it is not possible because combinations of production rules give rise to one morphological recognizable feature only (see Table 6). In terms of biological theory

both monogenetic and polygenetic morphological features appear in our simple systems.

The monothetic, hierarchical classification may be used as a key. Some morphological features appear as separating features between higher taxons, others as separating lower taxons (compare the tables). Features, specified as separating taxons at specific levels, are often used in biological taxonomy, but their use is a much debated issue.⁽¹⁵⁾ The applicability of the concepts may be due to the simplicity of our systems: only two pattern developing rules are defined, namely the two kinds of division rules.

The statements made above about the relation between morphological differences between patterns and the differences between their generative systems are based on an intuitive classification using a few global morphological features only. The question rises how far our conclusions are dependent on the choice of the features.

To investigate this question we have clustered the patterns on the basis of a set of locally defined morphological features. There were for all orderbranches separately:

- (1) The occurrence of side branches along the branches.
- (2) Maximal number of side branches on one symbol.
- (3) The occurrence of sidebranches on end symbols.
- (4) The maximal number of these.
- (5) Length of the branches.

These features were extracted automatically from the strings. The dissimilarity of the patterns was calculated as the Euclidian distance after normalization of the features, and the patterns were clustered agglomeratively using several clustering criteria (Lance and Williams).⁽¹⁶⁾

The following was found: The resulting classification was stable under different clustering criteria and gave rise to 15 classes which may be distinguished in all dendrograms but which vary among each other in variance. This classification on these local features agrees very well with the intuitive classification given above in the tables: all classes may be described in terms of the earlier used global morphological features. The conclusions stated above thus seem relatively independent of the choice of the features. The fact that different clustering criteria give the same classification confirms the discontinuity of the morphology of the patterns.

3. Developmental sequences

As noted above some of the developmental sequences show recurrence relations. Looking at the pictorial representation of the sequences of patterns these may be observed easily; in this way the morphology

of the patterns may aid the study of the generated languages. The formal structure of interactionless recurrence systems (their inclusion in the Chomsky hierarchy and in the different classes of *L*-systems) have been studied by Rozenberg and Lindenmayer⁽¹³⁾ (local catenative systems) and Herman *et al.*⁽¹⁴⁾ (general recurrence systems). Recurrence relations are observed in several biological structures. For example the branching patterns of inflorescences are classified on the basis of the occurrence of recurrence relations. It seems interesting to note that in plant morphology the local catenative inflorescences are considered most "primitive" while we find that the structure of their generative systems is also the most simple. Besides the recurrence systems, we see in our set of patterns a constancy of a part of the pattern during development, while the rest of the pattern may not be described in terms of previous patterns, as well as patterns in which no such regularity is observed (there is no specific area of growth in our patterns). The non-recurrent nature of our patterns is due to interaction among the symbols.

The differences between successive patterns in a sequence are quite large: the transformations may hardly be articulated in terms of global morphological features. Nevertheless the sequences may be easily recognized as such and intermingled sets of patterns can be separated in its composing sequences by human observers.

SUMMARY

Description of the morphology of organisms is mostly done in terms of features assuming certain states. Biological patterns are subjected to certain constraints imposed on them by the fact that they all are developed from a single cell. Using a type of parallel rewriting systems which incorporates some of the constraints of biological development, we generated a set of patterns. Of these patterns the following modes of description were compared:

- (1) The generative system.
- (2) The (sequences of) strings generated by these systems.
- (3) The (sequences of) pictorial representations of these strings (as 3-dimensional branching patterns).
- (4) The description of the pictorial representations in terms of morphological features and the classification of the patterns on the bases of these features.
- (5) The recurrence relations occurring in the sequences of patterns. The relevance of the results for the methodology of pattern description and classification in biology are discussed.

REFERENCES

1. A. Lindenmayer. Mathematical models for cellular interactions in development—I: Filaments with one-sided inputs. *J. Theor. Biol.* **18**, 280 (1968).
2. A. Lindenmayer. Mathematical models for cellular interactions in development—II: Simple and branching filaments with two-sided inputs. *J. Theor. Biol.* **18**, 300 (1968).
3. D. van Dalen. A note on some systems of Lindenmayer. *Math. Systems Theory* **5**, 128 (1971).
4. G. Rozenberg and P. G. Doucet. On OL-languages. *Information Control* **19**, 302 (1971).
5. G. T. Herman. The computing ability of a developmental model for filamentous organisms. *J. Theor. Biol.* **25**, 421 (1969).
6. G. T. Herman. Role of environment in developmental models. *J. Theor. Biol.* **29**, 329 (1970).
7. W. C. Rounds. Context free grammars on trees. *2nd ACM symp. theory of computing*, pp. 143 (1969).
8. J. W. Thatcher. Characterizing derivation trees of context free grammars through a generalization of finite automata theory. *J. Comp. System Sci.* **1**, 317 (1966).
9. A. Lindenmayer. Developmental systems without cellular interactions, their languages and grammars. *J. Theor. Biol.* **30**, 455 (1971).
10. H. Honda. Description of the form of trees by parameters of the tree-like body: effects of the branching angle and the branch length on the shape of the tree like body. *J. Theor. Biol.* **31**, 331 (1971).
11. S. M. Ulam. On some mathematical problems connected with patterns and growth of figures. *Proc. Symp. in Applied Math.*, Vol. 14, pp. 215. Reprinted in *Essays on Cellular Automata*, Burks, ed., Univ. of Illinois Press, Chicago (1970).
12. D. Cohen. Computer simulation of biological pattern generation processes. *Nature* **216**, 246 (1967).
13. G. Rozenberg and A. Lindenmayer. Developmental systems with locally catenative formulas. *Acta Informatica* **2**, 214 (1973).
14. G. T. Herman, A. Lindenmayer and G. Rozenberg. Description of developmental languages using recurrence systems. *Math. Systems Theory*, to be published.
15. R. R. Sokal and P. H. A. Sneath. *Principles of Numerical Taxonomy*. Freeman, San Francisco (1963).
16. G. N. Lance and W. T. Williams. A general theory of classificatory sorting strategies. I. Hierarchical systems. *Computer J.* **9**, 373 (1968).