GRAPHICAL APPLICATIONS OF L-SYSTEMS

Przemyslaw Prusinkiewicz

Technical Report CS-85-20

COMPUTER SCIENCE
DEPARTMENT

GRAPHICAL APPLICATIONS OF L-SYSTEMS

Przemyslaw Prusinkiewicz

Technical Report CS-85-20

Department of Computer Science
University of Regina
Regina, Saskatchewan CANADA S4S 0A2

November 1985

This work was partially supported by a grant from the Natural Sciences and Engineering Research Council of Canada, number A0324.

ISSN 0828-3494
ISBN 0-7731-0035-0

GRAPHICAL APPLICATIONS OF L-SYSTEMS

Przemyslaw Prusinkiewicz

Department of Computer Science
University of Regina
Regina, Saskatchewan S4S 0A2
CANADA

1. INTRODUCTION.

Rewriting systems can be used to generate pictures in two different ways. In the first case, a rewriting system operates directly on two-dimensional objects, such as arrays [Kirsch 1964, Dacey 1970], graphs [Rosenfeld and Milgram 1972, Pfaltz 1972], or "shapes" [Gips 1975, Stiny 1975]. In the second case, a string grammar (in the broad sense of the word, including parallel rewriting systems) is used to define strings of symbols. A graphic interpretation function subsequently maps these strings into pictures. This paper is devoted to this second approach. After the idea of applying string grammars to pictures is put into a historic perspective in Section 2, attention is focused on L-systems. The necessary definitions related to L-systems are collected in Section 3. Sections 4 and 5 concentrate on pictures generated by 0L-systems under two particular interpretations, the chain-code and the turtle interpretation, respectively. Examples of pictures are given and the classes of pictures generated under both interpretations are compared. Two approaches for extending the gamut of generated pictures are discussed in Sections 6 and 7. The first approach relies on extending the generative power of L-systems beyond that of 0L-systems. The second approach employs more sophisticated interpretation functions. Section 8 presents some open problems.

2. THE HISTORICAL BACKGROUND.

The idea of applying formal (string) languages to describe pictures emerged a few years after Chomsky established the fundamental concept of a phrase-structure grammar. Narasimhan [1962, 1966] and Ledley [1964, 1965] are credited with the first results in this area. Their interest was in recognizing handwritten characters and chromosomes, respectively. An approach designed for describing a wider class of pictures using string grammars was proposed by Shaw [1969]. For a survey of these early results, see Fu [1980].

The early research concentrated on picture recognition. Pictures were described as strings of symbols which represented selected primitives, such as straight segments, sharp V-turns, wide U-turns or branches. In some cases, relations between picture elements, such as ABOVE, BELOW, of INSIDE, were also considered as primitives. The actual recognition was performed by parsing the resulting strings.

For the purpose of picture recognition, the exact geometry of primitives is usually of secondary importance, as long as they can be properly identified. In the case of picture generation, however, the correspondence between string symbols and picture primitives must be specified in more detail. One example of a suitable specification, known as chain coding, is due to Freeman [1961]. Picture description using string grammars and chain coding was first investigated by Feder [1968]. He showed that the languages of chain codes describing such classes of figures as straight lines of arbitrary slope, circles of arbitrary radius, or convex figures in a plane, are all context sensitive. It was subsequently pointed out (for example, by Fu [1980]) that even intuitively simpler classes of pictures, such as the set of all rectilinear squares in an integer grid, correspond to context-sensitive chain-code languages. This discouraged to a certain degree a further study of chain-code languages, for it is believed that context-sensitive grammars are difficult to construct and they do not provide an intuitively clear description of languages. Nevertheless, picture generation using Chomsky grammars and the chain interpretation has recently received a considerable attention [Maurer, Rozenberg and Welzl 1982, Sudborough and Welzl 1985].

In order to describe growth of living organisms, Lindenmayer [1968] introduced the notion of parallel rewriting systems. The Lindenmayer systems, or L-systems, attracted the interest of many researchers, and the theory of L-systems was soon extensively developed [Herman and Rozenberg 1975, Lindenmayer and Rozenberg 1976]. However, although a geometrical interpretation of strings was at the origin of L-systems, they were not applied to picture generation until 1984, when Aono and Kunii [1984], and Smith [1984] used them to create realistic-looking images of trees and plants.

This paper further investigates graphical applications of L-systems. It is shown that these applications are not limited to the generation of trees. In particular, L-systems can be used to generate many fractal curves. Some of these curves were discovered or popularized by Mandelbrot [1982]. Others are believed to be presented here for the first time.

3. L-SYSTEMS.

This section summarizes fundamental definitions and notations related to L-systems. For their tutorial introduction, see Salomaa [1973], and Herman and Rozenberg [1975].

Let V denote an alphabet, V^* - the set of all words over V, and V^+ - the set of all nonempty words over V.

Definition 3.1. A **0L-system** is an ordered triple $G = \langle V, \omega, P \rangle$ where V is the **alphabet** of the system, $\omega \in V^+$ is a nonempty word called the **axiom**, and $P \subset V \times V^*$ is a finite **set of productions**. If a pair (a, x) is a production, we write $a \rightarrow x$. The letter a and the word x are called the **predecessor** and the **successor** of this production, respectively. It is assumed that for any letter $a \in V$, there is at least one word $x \in V^*$ such that $a \rightarrow x$. A 0L-system is **deterministic** iff for each $a \in V$ there is exactly one $x \in V^*$ such that $a \rightarrow x$.

Definition 3.2. Let $G = \langle V, \omega, P \rangle$ be a 0L-system, and suppose that $p = a_1 \ldots a_n$ is an arbitrary word over V. We will say that the word $q = x_1 \ldots x_n \in V^*$ is **directly derived** from (or **generated** by) p and write $p \Rightarrow q$ iff for all $i = 1, \ldots, n$, $a_i \rightarrow x_i$. A sequence $S(G)$ of words $q_0, q_1, q_2, \ldots$ such that $q_0 = \omega$ and $q_0 \Rightarrow q_1 \Rightarrow q_2 \Rightarrow \ldots$ is called a **sequence generated by** G. A word z is generatedby G iff it belongs to a sequence $S(G)$. The set of all words z generated by G is called the **language generated by** G and denoted $L(G)$.

Remark: A 0L-system is deterministic iff the sequence $S(G)$ is unique.

4. GENERATING PICTURES USING 0L-SYSTEMS WITH CHAIN INTERPRETATION.

This sections introduces the notion of a graphic interpretation of a string, defines the chain interpretation, and provides examples of pictures generated by 0L-systems under this interpretation.

Definition 4.1. A **picture** π is a set of points in the plane: $\pi \in 2^{\Re \times \Re}$. A function $I: V^* \rightarrow 2^{\Re \times \Re}$ mapping strings over alphabet V into pictures is called a (graphic) **interpretation function**.

Definition 4.2. Let π denote a picture drawn in a square grid by connecting 4-adjacent [1] vertices of this grid with straight line segments of unit length. In general, the resulting line may be self-intersecting, and the same line segment can be drawn more than once. Next, let each line segment be assigned a letter A, B, C, or D, according to the slope of the lines (Fig. 1). The string of letters representing the entire picture is called the **chain representation** of π.

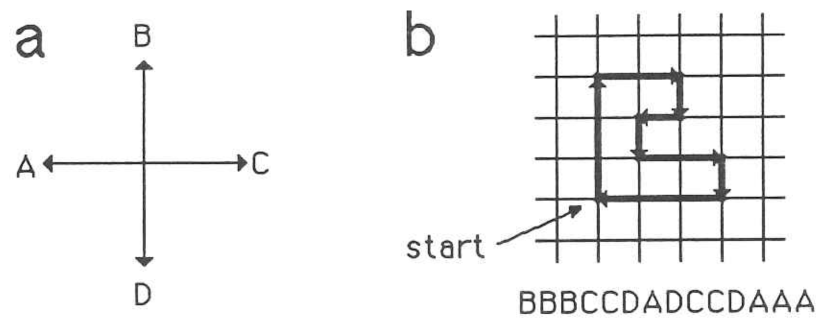


Fig. 1. (a) Assignment of letters to line segments in chain coding.
(b) The chain representation of an example picture.

A picture π may have many chain representations depending, for example, on the starting point assumed. However, a given string describes a unique figure (up to translations). Consequently, chain coding is an interpretation function defined on the alphabet $V_c = \{A, B, C, D\}$.

Figure 2 presents chain interpretations of words z generated by some deterministic 0L-systems G over alphabet V_c. Data under each picture indicate the length of derivation, the axiom ω of the L-system, and its set of productions P. Note the variety of the shapes obtained, and the simplicity of the underlying L-systems.

5. GENERATING LINE DRAWINGS USING L-SYSTEMS WITH TURTLE INTERPRETATION.

In this section an alternative interpretation function, called turtle coding [Papert 1980, Abelson and diSessa 1982], is considered.

---

[1] Grid vertices (x_1, y_1) and (x_2, y_2) are 4-adjacent if $|x_1-x_2| + |y_1-y_2| = 1$.

Definition 5.1. Let π denote a picture drawn in a rectangular grid by connecting adjacent vertices of the grid with straight line segments, as described in Def. 4.2. Furthermore, let each line segment be represented by letter F, and a $+90°$ or $-90°$ change of the segment slope with respect to the previous segment be denoted by character $+$ or $-$, respectively. The string of characters: F, $+$ and $-$ representing the entire picture is called a **turtle representation** of π.

A given picture may have many turtle representations depending, among other factors, on the way each angle is represented. For example, a turn of $180°$ can be represented as $++$, $--$, $++-+$, etc. However, a given string of characters F, $+$ and $-$ describes a unique figure (up to translations and rotations of $n \cdot 90°$). Consequently, the turtle coding is an interpretation function defined on the alphabet $V_t = \{F, +, -\}$.

The Koch curve shown in Fig. 2a is generated by the following L-system:

3
F–F–F–F–
F → F+F–F–FF+F+F–F
+ → +
– → –

Analogously, L-systems generating curves shown in Figs. 2b-2d under the turtle interpetation can be found. The existence of different L-systems generating the same sequence of figures motivates the following definitions.

Definition 5.2. Let $G = \langle V, \omega, P \rangle$ be an L-system, and suppose that I is an interpretation function defined on alphabet V. A pair $H = (G, I)$ is then called a **picture-generation system**, or **P-system** in short.

Definition 5.3. Let $\pi_{10}, \pi_{11}, \pi_{12}, \ldots$ and $\pi_{20}, \pi_{21}, \pi_{22}, \ldots$ denote sequences of pictures generated by deterministic P-systems H_1 and H_2, respectively. Systems H_1 and H_2 are called **equivalent** iff for any $i=0,1,2,\ldots$ pictures π_{1i} and π_{2i} are congruent.

A natural question is: What is the relationship between the families of pictures generated by 0L-systems under the chain and turtle interpretation? A partial answer to this question is provided by theorems 5.1 and 5.2, stated here without proofs.

Theorem 5.1. Families of figures generated by 0L-systems under chain and turtle interpretation are incomparable.

Definition 5.4. Let h denote a cyclic permutation of the sequence of letters A,B,C,D such as $h(A) = A$, $h(B) = C$, $h(C) = D$, and $h(D) = A$. A set of productions P is **closed under permutation** h if for any production $a \rightarrow b_1 b_2 ... b_n$ in P, production $h(a) \rightarrow h(b_1)h(b_2)...h(b_n)$ also belongs to P.

Theorem 5.2. Let a P-system H_c consists of a 0L-system $G_c = \langle V_c, \omega_c, P_c \rangle$ and the chain interpretation. If the set of productions P_c is closed under cyclic permutation h there exists a P-system H_t consisting of a 0L-system $G_t = \langle V_t, \omega_t, P_t \rangle$ and the turtle interpretation, equivalent to H_c.

Example: Systems H_t exist for all pictures shown in Fig. 2, and for picture shown in Fig. 3b.

6. GRAPHICAL APPLICATIONS OF GENERALIZED L-SYSTEMS.

The gamut of pictures generated by P-systems can be extended using two approaches: by enhancing the generative power of L-systems beyond 0L-systems, or by using interpretation functions more complex than chain or turtle interpretation. Various extensions to 0L-systems have been thoroughly studied in the past [Salomaa 1973, Herman and Rozenberg 1975, Lindenmayer and Rozenberg 1976], and many of them have straightforward graphical applications. **2L-systems** use productions of the form $a_l < a > a_r \rightarrow x$; this notation means that the letter a can produce word x iff a is preceded by letter a_l and followed by a_r. Letters a_l and a_r form the left and the right context of a in this production. Productions in **1L-systems** have one-side context only; consequently, they are either of the form $a_l < a \rightarrow x$ or $a > a_r \rightarrow x$ (Fig. 4a). 0L-systems, 1L-systems and 2L-systems belong to a wider class of **(k,l)-systems**. In a (k,l)-system, the left context is a word of length k, and the right context is a word of length l. L-systems can be generalized even further by allowing productions of the form $u_l < u > u_r \rightarrow x$, where all three components of the predecessor are words of arbitrary length (Figs. 4b and 4c). Additionally, auxiliary symbols may be allowed to occur in the words being derived. In extended L-systems, or **EL-systems**, auxiliary symbols have the function of nonterminal letters in Chomsky grammars: they can be used in intermediate stages of a derivation, but must not appear in the final words of the language. In graphical applications, another approach to auxiliary symbols is useful:

they may occur in any stage of the derivation, but must be erased from the string just before its interpretation (Fig. 4d).

The generalizations of L-systems outlined so far are based on modifications of the form of productions. Another approach, based on a controlled use of productions, also seems promising. In **TL-systems** productions are grouped into sets called tables. All productions used in a given step of the derivation must belong to the same table. For example, a plant may result from applying separate tables to generate its stem, leaves, and flowers. In this case, the sequence of tables to be used at each step would be predetermined. Alternatively, tables can be selected according to probabilistic rules.

7. EXTENDING INTERPRETATION FUNCTIONS.

The second approach to extending the gamut of pictures generated by P-systems involves interpretation functions other than chain and turtle coding. A simple modification of these functions introduces a distinction between visible and invisible line segments. The visible segments are represented using letters A, B, C, D and F, as previously. The corresponding invisible segments are represented by lower-case letters: a, b, c, d and f. The invisible segments make it possible to define unconnected pictures (Fig. 5a).

Another modification of the interpretation functions uses a triangle or hexagonal grid instead of the square grid; in the case of turtle coding this corresponds to associating turns of $\pm 120°$ or $\pm 60°$ to the symbols $+$ and $-$ (Figs. 5b-d). Pursuing the same idea, symbols $+$ and $-$ may represent turns by arbitrary angles α and β. In consequence of the Eulerian formula, no underlying grid can be assumed in this general case. Finally, line segments need not be confined to the plane. For example, the chain interpretation can be expanded by symbols denoting line segments in the positive and negative direction of axis z. A corresponding L-system will then describe a three-dimensional object rather than a two-dimensional picture.

The original L-systems, as described by Lindenmayer [1968], used braces to group sequences of segments into branches. This idea was preserved in L-systems generating trees for computer imagery purposes [e.g. Smith 1984]. However, grouping lines with braces can also be used for other purposes. For example, lines within a pair of braces may define the boundary of a filled polygon. This is particularly attractive in the case of three-dimensional objects, since polygons may represent faces of polyhedra.

8. CONLUDING REMARKS.

This paper shows that L-systems can be used to generate a wide variety of pictures. However, many problems related to the graphical applications of L-systems remain open. One possible direction of future research consists of finding L-systems and interpretation functions suitable for generating visually attractive images. As is often the case with fractals, these images can be appealing either because of their abstract beauty, or because of their similarity to real-life objects. Another research direction consists of exploring formal properties of L-systems related to picture generation. This is parallel to the study of graphical applications of Chomsky languages initiated by Maurer, Rozenberg and Welzl [1982]. Example problems falling in this category are: Given an L-system and an interpretation function, is the resulting line closed? Is it self-intersecting or tree-like? Are there any segments drawn more than once? Are they drawn infinitely many times (if the derivation length tends to infinity)? What is the function relating the diameter of the picture to the derivation length? Answers to these and similar questions are interesting not only from the theoretical point of view. They may also help to construct L-systems which will generate pictures with given properties.

ACKNOWLEDGMENT.

This research was supported by grant number A0324 from the Natural Sciences and Engineering Research Council of Canada.

REFERENCES.

Abelson, H., and diSessa, A. A. [1982]: *Turtle geometry*. M.I.T. Press, Cambridge and London.

Dacey, M. F. [1970]: The syntax of a triangle and some other figures. *Pattern Recognition*, vol. 2, pp. 11-31.

Davis, C., and Knuth, D. E. [1970]: Number representations and dragon curves. *J. of Recreational Mathematics* vol. 3, pp. 66-81 and 133-149.

Feder, J. [1968]: Languages of encoded line patterns. *Information and Control* 13, pp. 230-244.

Freeman H. [1961]: On encoding arbitrary geometric configurations. *IRE Trans. Electron. Comput.*, vol. 10, pp. 260-268.

Fu, K. S. [1980]: Syntactic (linguistic) pattern recognition. In Fu K. S. (Ed.): *Digital pattern recognition*, Springer-Verlag, Berlin - Heiedlberg - New York, pp.95-134.

Gips, J. [1975]: *Shape grammars and their uses; artificial perception, shape generation and computer aesthetics.* Birkhäuser-Verlag, Basel and Stuttgart.

Herman, G. T., and Rozenberg, G. [1975]: *Developmental systems and languages*. North-Holland, Amsterdam and Oxford.

Kirsch, R. A. [1964]: Computer interpretation of English text and picture patterns. *IEEE Trans. Electron. Comput.*, vol. EC-13, pp. 363-376.

Ledley, R. S. [1964]: High-speed automatic analysis of biomedical pictures. *Science*, vol. 146, No. 3641, pp. 216-223.

Ledley, R. S., *et al.* [1965]: FIDAC: Film input to digital automatic computer and associated syntax-directed pattern-recognition programming system. In J. T. Tippet et al. (Eds.): *Optical and electro-optical informtation processing*, M.I.T. Press, Cambridge, pp. 591-613.

Lindenmayer, A. [1968]: Mathematical models for cellular interaction in development, Parts I and II. *Journal of Theoretical Biology*, vol. 18, pp. 280-315.

Lindenmayer, A., and Rozenberg, G. (Eds.) [1976]: *Automata, languages, development*. North-Holland, Amsterdam - New York - Oxford.

Mandelbrot, B. B. [1982]: *The fractal geometry of nature*. W. H. Freeman, San Francisco.

Maurer, H. A., Rozenberg, G., and Welzl, E. [1982]: Using string languages to describe picture languages. *Information and Control*, vol. 54, pp. 155-185.

Narasimhan, R. [1962]: A linguistic approach to pattern recognition. Rep. 21, Digital Computer Laboratory, University of Ilinois, Urbana.

Narasimhan, R [1966]: Syntax-directed interpretation of classes of pictures. *Comm. ACM*, vol. 9, pp. 166-173.

Papert, S. [1980]: *Mindstorms: children, computers, and powerful ideas*. Basic Books.

Pfaltz, J. L. [1972]: Web grammars and picture description. *Computer Graphics and Image Processing*, vol. 1, pp. 193-220.

Rosenfeld, A., and Milgram, D. L. [1972]: Web automata and web grammars. *Machine Intelligence*, vol. 7, pp. 307-324.

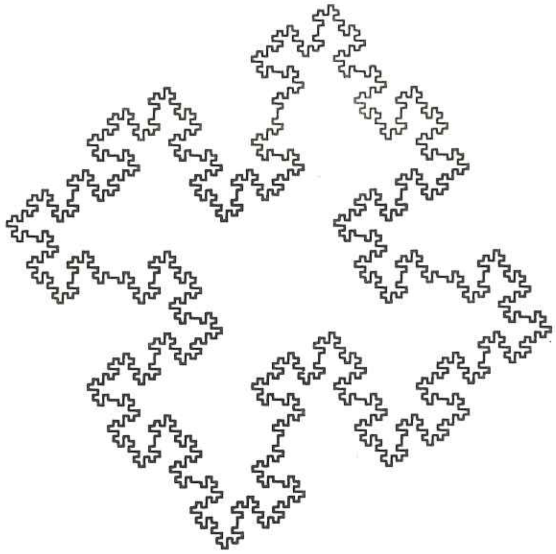
Salomaa, A. [1973]: *Formal languages*. Academic Press, New York and London.

Shaw, A. C. [1969]: A formal picture description scheme as a basis for a picture processing system. Information and Control, vol. 14, pp. 9-52.

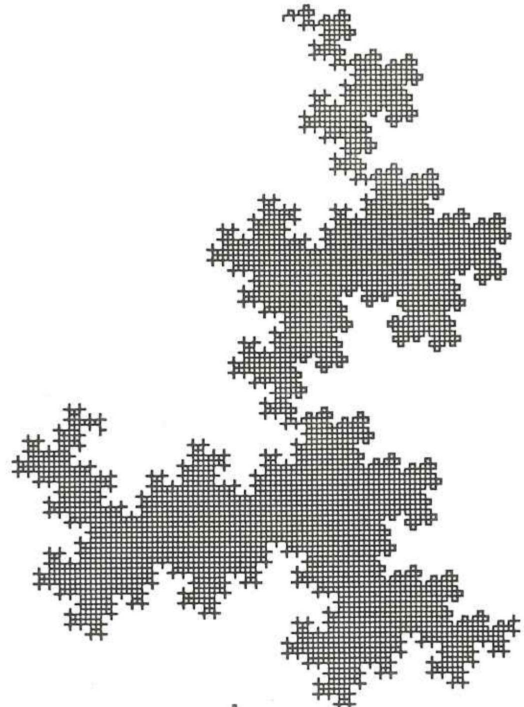
Sierpinski, W. [1915]: Sur une courbe dont tout point est un point de ramification. *Comptes Rendus* (Paris) vol. 160 p. 302.

Sudborough, I. H., and Welzl, E. [1985]: Complexity and decidability for chain code picture languages. *Theoretical Computer Science*, vol. 36, pp. 173-202.

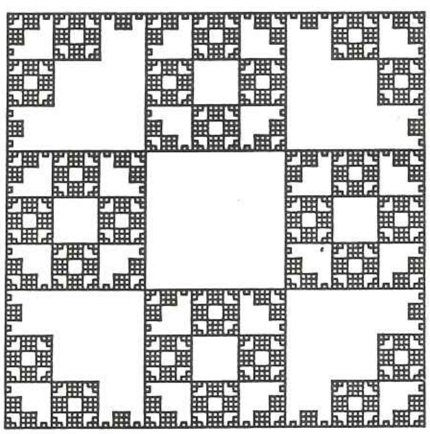
Stiny, G. [1975]: *Pictorial and formal aspects of shape and shape grammars*. Birkhäuser-Verlag, Basel and Stuttgart.



3 **a**
ABCD
A --> ABADDABA
B --> BCBAABCB
C --> CDCBBCDC
D --> DADCCDAD



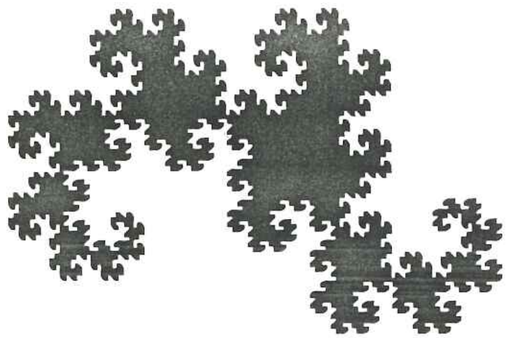
6 **b**
B
A --> ABBDA
B --> BCCAB
C --> CDDBC
D --> DAACD



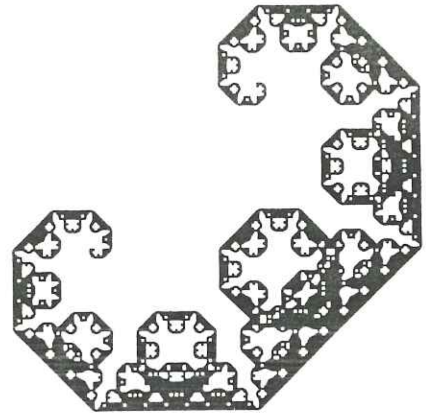
4 **c**
ABCD
A --> AABCDAA
B --> BBCDABB
C --> CCDABCC
D --> DDABCDD

4 **d**
ABCD
A --> AABCDABA
B --> BBCDABCB
C --> CCDABCDC
D --> DDABCDAD

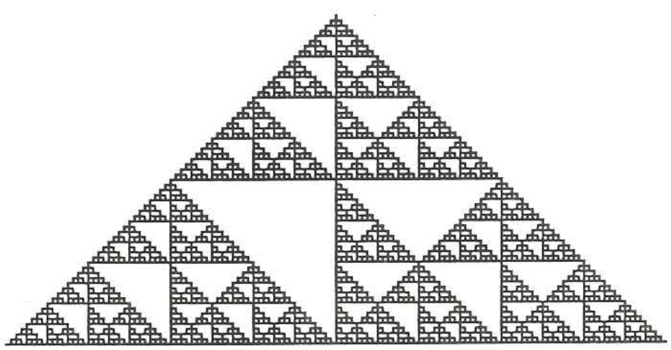
Fig. 2. Examples of pictures generated using OL-systems under chain interpretation. Picture (a) is the quadratic Koch island from Mandelbrot [1982].



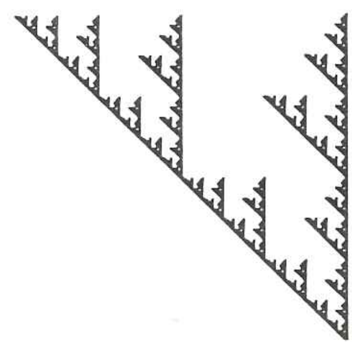
14 a
B
A --> AB
B --> CB
C --> CD
D --> AD



13 b
A
A --> BC
B --> CD
C --> DA
D --> AB

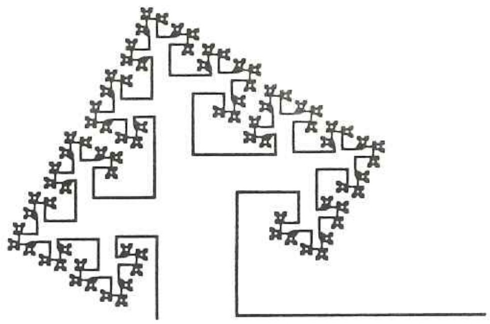


14 c
C
A --> AB
B --> AD
C --> DC
D --> CB



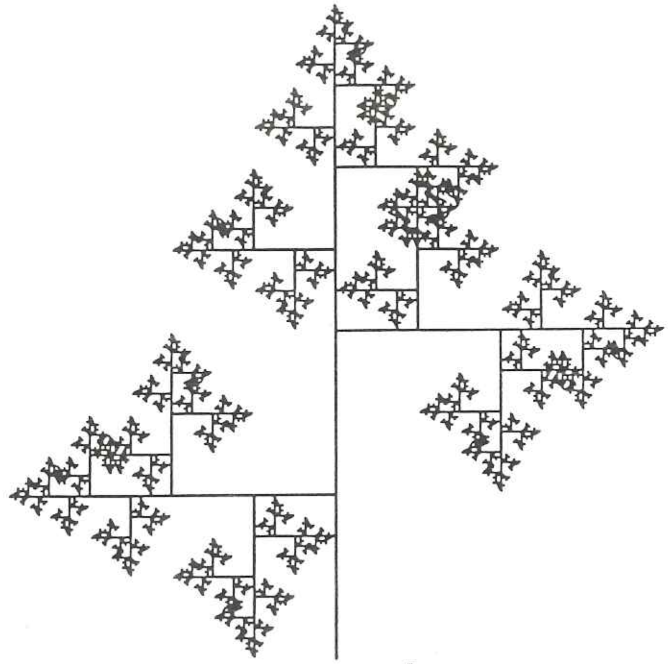
14 d
A
A --> CB
B --> BA
C --> DA
D --> CD

Fig. 3. Examples of pictures generated using OL-systems under chain interpretation. Picture (a) is the dragon curve [Davis and Knuth 1970]. Picture (b) is the C curve from Abelson and diSessa [1982].



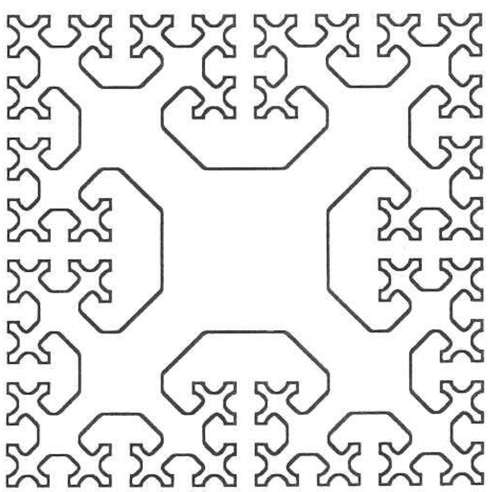
a

```
6
BC
A > B --> ADABCB
B > C --> BABCDC
C > D --> CBCDAD
D > A --> DCDABA
  A   --> AA
  B   --> BB
  C   --> CC
  D   --> DD
```



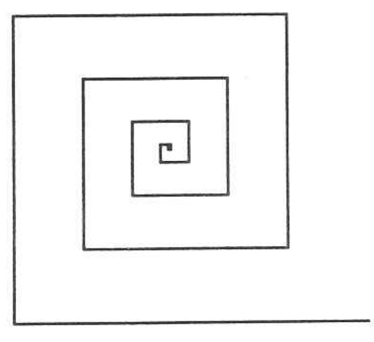
b

```
7
-F++F
F++F --> F+F++F+F-F++F-F++FFF
  F  --> FF
  +  --> +
  -  --> -
```



c

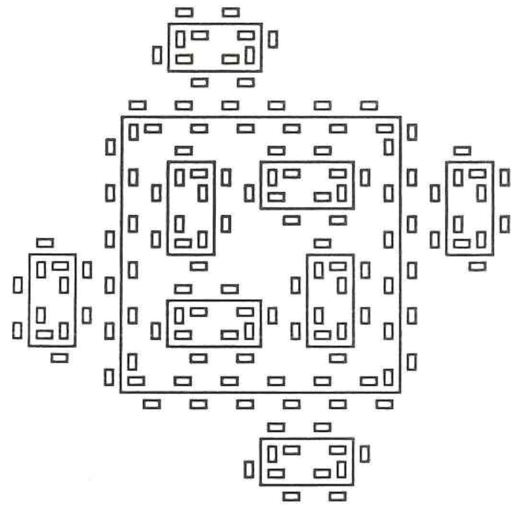
```
4
FF-FFFF-FFFF-FFFF-FF
F-F --> +F+++F+F+++F+FF-FFFF-
          FFFF-FF+F+++F+F+++F+
+F+++F --> +F+++F+F+++F
     F   --> FF
     +   --> +
```

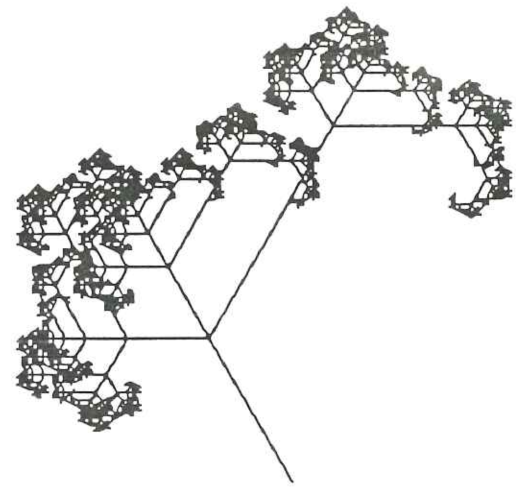


d

```
18
X
  X   --> X+F
+ < F --> FY
  Y   --> YF
  F   --> F
  +   --> +
```

Fig. 4. Examples of pictures obtained using generalized L-systems. In these examples the set of productions is ordered. If more than one production can be applied, the production higher in the list is chosen.



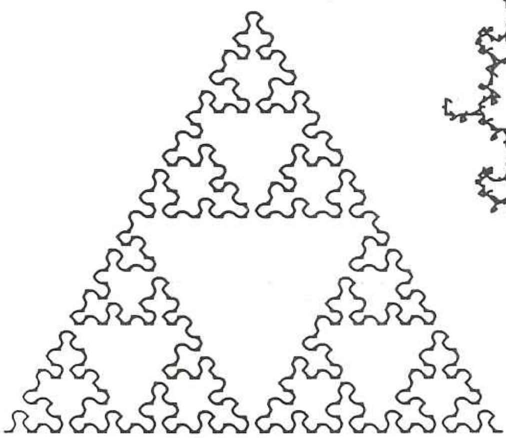


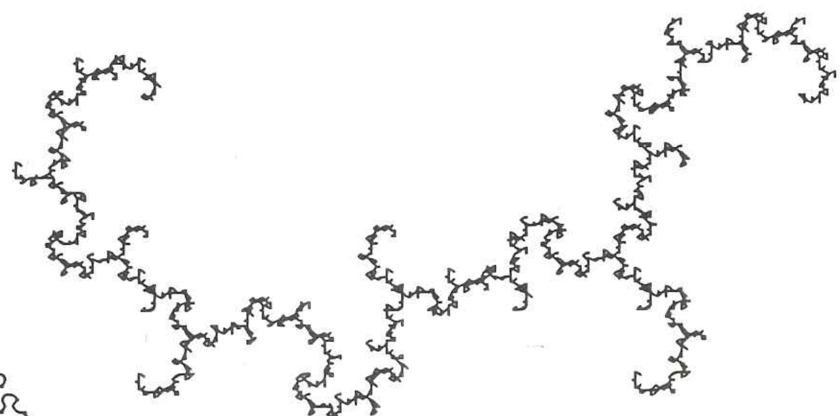
a

2
F+F+F+F
F --> F+f-FF+F+FF+Ff+FF-f+
FF-F-FF-Ff-FFF
f --> ffffff
+ --> +
- --> -

b

7
+F+++F
F+++F --> F+FF+++FF+F+++F--F+++FF
F --> FF
\+ --> +
\- --> -





c

6
YF
XF --> -YF+XF+YF-
YF --> +XF-YF-XF+
\+ --> +
\- --> -

d

8
YF
XF --> +XF-YF-YF+
YF --> -XF+XF+YF-
\+ --> +
\- --> -

Fig. 5. Examples of pictures generated using invisible lines or triangular grid. Picture (a) is a "combination of lakes and islands" from Mandelbrot [1982]. Picture (c) is an arrowhead [Sierpiński 1915].