

GRAPHICAL APPLICATIONS OF L-SYSTEMS

Przemyslaw Prusinkiewicz

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

ABSTRACT.

A new method for generating pictures is presented and illustrated with examples. The idea is to generate a string of symbols using an L-system, and to interpret this string as a sequence of commands which control a "turtle". Suitable generalizations of the notions of the L-system and of a turtle are introduced. The resulting mathematical model can be used to create a variety of (finite approximations of) fractal curves, ranging from Koch curves, to classic space-filling curves, to relatively realistic-looking pictures of plants and trees. All these pictures are defined in a uniform and compact way.

RESUME.

Une nouvelle méthode pour engendrer des images est présentée et illustrée avec des exemples. Cette méthode comprend deux étapes. On commence par engendrer une séquence de symboles avec un L-système. Ensuite on utilise cette séquence pour contrôler les mouvements d'une tortue qui trace l'image en question. Les notions d'un L-système et d'une tortue sont généralisées pour mieux correspondre au but de la création des images. Le modèle mathématique résultant s'applique à la création d'une large variété d'objets fractals, y compris des courbes de Koch, des courbes qui remplissent tout une aire plane, ainsi que des images relativement réalistes des plantes et des arbres. Toutes ces images sont définies d'une manière homogène et compacte.

KEYWORDS: L-systems, turtle geometry, fractals, space-filling curves, plants, trees.

1. INTRODUCTION.

Rewriting systems can be used to generate pictures in two different ways. In the first case, rewriting systems operate 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, string grammars (in the broad sense of the word, including parallel rewriting systems) are used to define strings of symbols. A graphic interpretation function subsequently maps these strings into pictures. This paper describes a method for generating pictures based on 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 [Lindenmayer 1968]. The necessary definitions related to OL-systems are collected in Section 3. Section 4 adapts the notion of a "turtle" [Papert 1980, Abelson and diSessa 1982] to the purpose of graphical interpretation of strings, and presents examples of pictures generated by OL-systems under this interpretation. Section 5 extends this basic approach in two directions: by generalizing the notion of the L-system, and by increasing the range of string symbols interpreted by the turtle. Section 6 presents conclusions and lists several open problems.

2. THE HISTORICAL BACKGROUND.

The idea of describing pictures using formal (string) languages 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 the recognition of 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, or INSIDE, were also considered as primitives. The actual picture recognition was performed by parsing the resulting strings.

In the case of picture generation, the correspondence between string symbols and picture primitives must be specified in more detail. The first such specification, known as chain coding, was developed by Freeman [1961]. Feder [1968] 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, for example the set of all rectilinear squares in an integer grid, correspond to context-sensitive chain-code languages. To a certain degree, this discouraged a further study of chain-code languages, for the context-sensitive grammars are usually difficult to construct and do not provide an intuitively clear description of languages. Neverthe-

less, picture generation using Chomsky grammars and the chain interpretation has recently received 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 a parallel rewriting system. 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. Approximately at the same time, Siromoney and Subramanian [1983] noticed that L-systems with chain-code interpretation could be used to generate some space-filling curves.

This paper further investigates graphical applications of L-systems. The emphasis is on the turtle interpretation rather than the chain coding, because the turtle interpretation allows for generating pictures which are not confined to a grid. The notions of the L-system and of a turtle are generalized to provide increased flexibility in picture specification. The resulting mathematical model is used to generate a wide spectrum of fractal curves.

3. 0L-SYSTEMS.

This section summarizes fundamental definitions and notations related to 0L-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 triplet $G = \langle V, \omega, P \rangle$ 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, χ) is a production, we write $a \rightarrow \chi$. The letter a and the word χ 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 $\chi \in V^*$ such that $a \rightarrow \chi$. A 0L-system is **deterministic** iff for each $a \in V$ there is exactly one $\chi \in V^*$ such that $a \rightarrow \chi$.

Definition 3.2. Let $G = \langle V, \omega, P \rangle$ be a 0L-system, and suppose that $\mu = a_1 \dots a_m$ is an arbitrary word over V . We will say that the word $v = \chi_1 \dots \chi_m \in V^*$ is **directly derived** from (or **generated** by) μ and write $\mu \Rightarrow v$ iff $a_i \rightarrow \chi_i$ for all $i = 1, \dots, m$. A word v is generated by G in a derivation of **length** n if there exists a sequence of words $\mu_0, \mu_1, \dots, \mu_n$ such that $\mu_0 = \omega$, $\mu_n = v$ and $\mu_0 \Rightarrow \mu_1 \Rightarrow \dots \Rightarrow \mu_n$.

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

This sections formalizes the notion of the turtle interpretation of a string 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: $\Pi \in 2^{R \times R}$. A function $I: V^* \rightarrow 2^{R \times R}$ mapping the set of strings

over the alphabet V into the set of pictures is called a (graphic) **interpretation function**.

Definition 4.2. A **state** of the turtle is a triplet (x, y, α) , where the coordinates (x, y) represent the turtle's **position**, and angle α , called the turtle's **heading**, is interpreted as the direction in which the turtle is facing. Given the **step size** d and the **angle increment** δ , the turtle can respond to commands represented by the following symbols:

- F** Move forward a step of length d . The state of the turtle changes to (x', y', α) , where $x' = x + d \cos \alpha$ and $y' = y + d \sin \alpha$. A line segment between points (x, y) and (x', y') is drawn.
- f** Move forward a step of length d without drawing a line.
- +** Turn right by angle δ . The next state of the turtle is $(x, y, \alpha + \delta)$. (Here we assume that the positive orientation of angles is clockwise.)
- Turn left by angle δ . The next state of the turtle is $(x, y, \alpha - \delta)$.
- |** Turn away. The state of the turtle changes to $(x, y, \alpha + 180^\circ)$.

All other symbols are ignored by the turtle (the turtle preserves its state).

Definition 4.3. Let v be a string, (x_0, y_0, α_0) - the initial state of the turtle, and d, δ - fixed parameters. The picture (set of lines) drawn by the turtle responding to the string v is called the **turtle interpretation** of v .

Figure 1 presents chain interpretations of the words generated by some deterministic 0L-systems. In all cases, the angle increment δ is equal to 90° . Data under each picture indicate the length of derivation n , the step size d (in pixels), the axiom ω of the 0L-system, and the set of productions P . Note the variety of the shapes obtained, and the simplicity of the underlying L-systems.

The Hilbert curve (Fig. 1f) is representative of classic space-filling curves. Other well-known space-filling curves were discovered by Peano [1890] and Sierpiński [1912]. The Peano curve is generated by the L-system with axiom **X** and productions:

$$\begin{aligned} X &\rightarrow XFYFX+F+YFXFY-F-XFYFX \\ Y &\rightarrow YFXFY-F-XFYFX+F+YFXFY \\ F &\rightarrow F \quad + \rightarrow + \quad - \rightarrow - \end{aligned}$$

A square-grid approximation of the Sierpiński curve is generated by the L-system with axiom **F+XF+F+XF** and productions:

$$\begin{aligned} X &\rightarrow XF-F+F-XF+F+XF-F+F-X \\ F &\rightarrow F \quad + \rightarrow + \quad - \rightarrow - \end{aligned}$$

Once again, note the simplicity of these descriptions. For the long history of creating possibly elegant and compact programs for generating space-filling curves, see [Null 1971, Wirth 1976, Goldschlager 1981, Witten and Wyvill 1983, Cole 1983, Cole 1985].

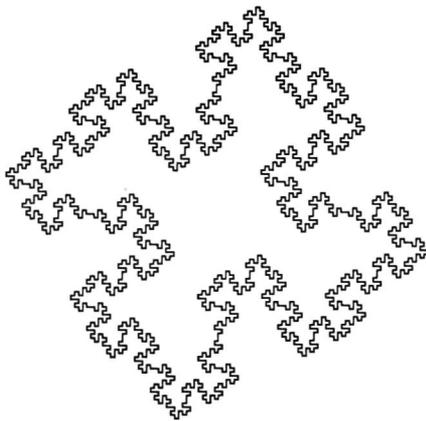
5. EXTENSIONS OF THE BASIC MODEL.

This section generalizes the notion of the L-system and extends the notion of the turtle interpretation of a string. The purpose is to increase the flexibility of picture specification.

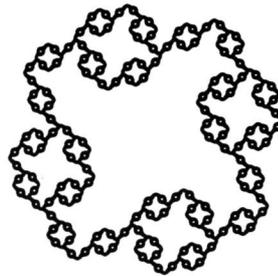
Various extensions to 0L-systems have been thoroughly studied in the past [Salomaa 1973, Herman and Rozenberg 1975, Lindenmayer and Rozenberg 1976]. Specifically, **2L-systems** use productions of the form $a_l < a > a_r \rightarrow \chi$; this notation means that the letter a (called the **strict predecessor**) can produce word χ iff a is preceded by letter a_l and followed by a_r . Thus, 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 $a_l < a \rightarrow \chi$ or $a > a_r \rightarrow \chi$. 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 . How-

ever, the strict predecessor is still limited to a single letter.

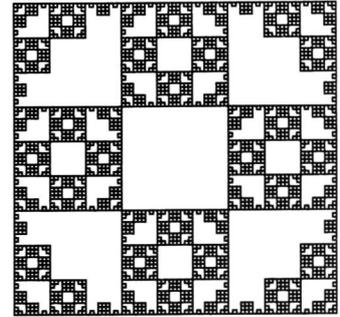
For the purpose of picture generation it is convenient to generalize L-systems even further by allowing for productions of the form $\eta_l < \eta > \eta_r \rightarrow \chi$, where all three components of the predecessor are words of arbitrary length. This modification of L-systems seriously affects the notion of the direct derivation of words. In 0L-systems and (k,l)-systems the strict predecessors of all productions consist of one letter. Consequently, in a derivation $\mu \Rightarrow v$ each letter of μ is a strict predecessor of some production. On the other hand, if the lengths of the strict predecessors η may vary, the partition of μ into strict predecessors depends on the particular productions used to derive v . The method for parti-



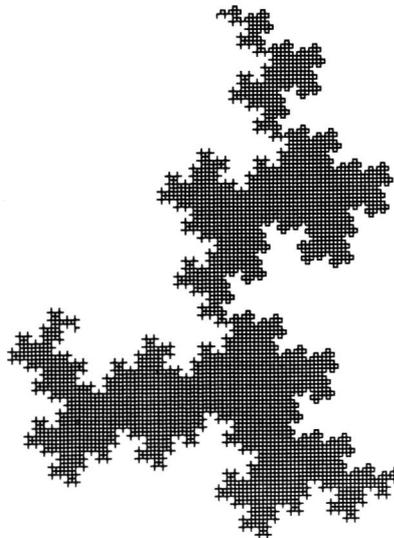
a $n=3, d=2$
 $F+F+F+F$
 $F \rightarrow F+F-F-FF+F+F-F$
 $+ \rightarrow + \quad - \rightarrow -$



b $n=4, d=1$
 $F+F+F+F$
 $F \rightarrow FF+F+F+F+F-F$
 $+ \rightarrow + \quad - \rightarrow -$



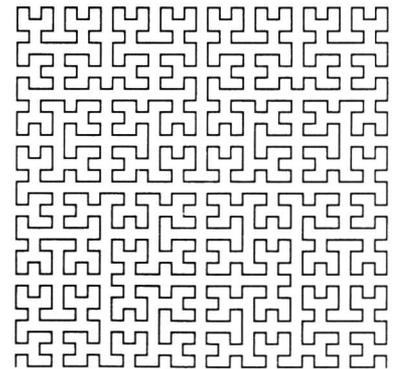
c $n=4, d=2$
 $F+F+F+F$
 $F \rightarrow FF+F+F+F+FF$
 $+ \rightarrow + \quad - \rightarrow -$



d $n=6, d=2$
 F
 $F \rightarrow F+FF|F+F$
 $+ \rightarrow + \quad | \rightarrow |$



e $n=14, d=1$
 X
 $X \rightarrow X+YF+$
 $Y \rightarrow -FX-Y$
 $F \rightarrow F \quad + \rightarrow + \quad - \rightarrow -$



f $n=5, d=6$
 X
 $X \rightarrow -YF+XFX+FY-$
 $Y \rightarrow +XF-YFY-FX+$
 $F \rightarrow F \quad + \rightarrow + \quad - \rightarrow -$

Fig. 1. Examples of pictures generated by 0L-systems under turtle interpretation. Figure (a) is a quadratic Koch island [Mandelbrot 1982]. Figure (e) is the dragon curve [Davis and Knuth 1970]. Figure (f) is the Hilbert [1891] curve.

tioning the word μ introduced below is based on scanning the string μ from left to right.

Definition 5.1. A **pL-system** (pseudo-L-system) is an ordered triplet $G = \langle V, \omega, p \rangle$, where V is the alphabet of the system, $\omega \in V^+$ is the axiom, and $p: \{1, \dots, N\} \rightarrow (V^* \times V^* \times V^*) \times V^*$ is an **ordered set of productions**. Instead of $p(i) = (\eta_l, \eta, \eta_r, \chi)$ we write that production $p(i)$ is equal to $\eta_l < \eta > \eta_r \rightarrow \chi$. Let l' , c' and r' denote the lengths of the strings η_l , η and η_r , respectively. Production p_i **matches** string $\mu = a_0 \dots a_m$ at position s , $0 \leq s \leq m$, iff the string μ can be represented as $a_0 \dots a_{s-l'-1} \eta_l \eta \eta_r a_{s+c'+r'} \dots a_m$. The production p_i is **applicable** to the string μ at position s iff it matches μ at position s and no production p_j preceding p_i ($j < i$) matches μ at this same position.

Note. In order to keep specifications of L-systems short, it is convenient to assume that all productions of the form $a \rightarrow a$, $a \in V$ are automatically appended at the end of any set p . Consequently, it is not necessary to list these productions when specifying p . On the other hand, any production of the form $a \rightarrow \chi$ entered explicitly into the set p will precede $a \rightarrow a$. Thus, production $a \rightarrow a$ can be "overwritten", if necessary.

Definition 5.2. Let $G = \langle V, \omega, p \rangle$ be a pL-system, and suppose that μ is an arbitrary word over V . We will say that the word v is **directly derived** from μ and write $\mu \Rightarrow v$ iff there exists a sequence of productions $p^{(t)} = \eta_l^{(t)} < \eta^{(t)} > \eta_r^{(t)} \rightarrow \chi^{(t)}$ and a sequence of positions $s^{(t)}$ ($t = 0, \dots, k$; $k \leq m$) such that:

- $\mu = \eta^{(0)} \dots \eta^{(k)}$,
- $v = \chi^{(0)} \dots \chi^{(k)}$,

- $p^{(0)}$ is applicable to the string μ at position $s^{(0)} = 0$,
- $p^{(t+1)}$ is applicable to the string μ at position $s^{(t+1)} = s^{(t)} + \text{length}(\eta^{(t)})$ for all $t = 0, 1, \dots, k-1$.

Example. Consider a pL-system G with the following productions:

$$p_1 = X < XY \rightarrow YX$$

$$p_2 = X > X \rightarrow YXX$$

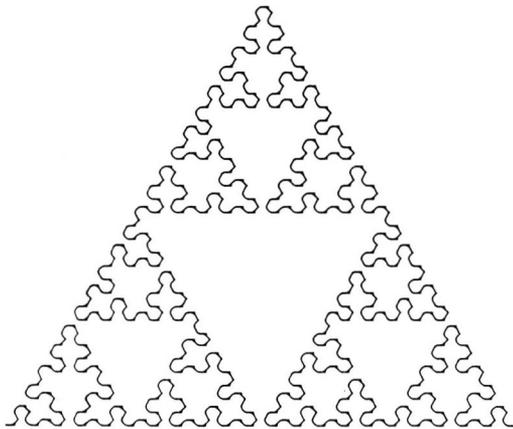
A word $\mu = \mathbf{XXYXY}$ will be partitioned into strict predecessors as follows: $\mathbf{X XY X Y}$. The corresponding successors are: $\mathbf{YXX YX X Y}$. Thus, the word v directly derived from μ is equal to $\mathbf{YXXYXXY}$.

The notion of the direct derivation is extended to the derivation of length n as in the case of OL-systems. Two curves generated by pL-systems are shown in Fig. 2.

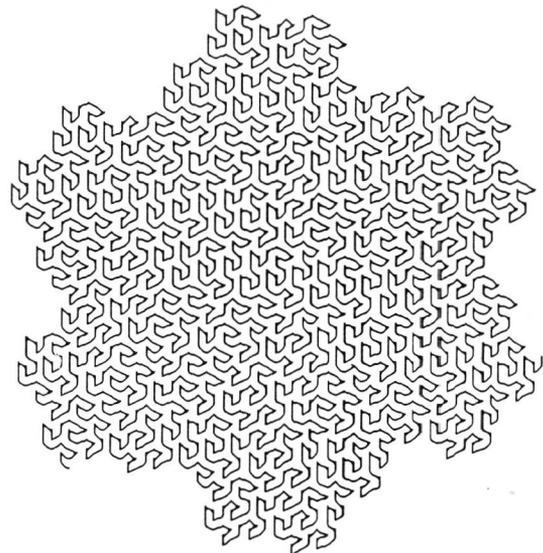
A useful extension of the set of commands interpreted by the turtle introduces two symbols "[" and "]" defined as follows:

- [Push current state of the turtle into a (pushdown) stack.
-] Pop a state from the stack, and make it the current state of the turtle. No line is drawn, although in general the position of the turtle is changed. If the stack is empty and no state can be popped, an error is reported and the string has no interpretation.

The above use of brackets is consistent with the original definition of L-systems by Lindenmayer [1968] where brackets were used to specify branches (of algae). This idea was also preserved in L-systems generating plants and trees for computer imagery purposes [e.g. Smith 1984]. Some plants and trees generated by OL-systems and pL-systems under the extended turtle interpretation are shown in Fig. 3.



a $n=6, d=4, \delta=60^\circ$
 YF
 $XF \rightarrow YF+XF+YF$
 $YF \rightarrow XF-YF-XF$



b $n=4, d=5, \delta=60^\circ$
 XF
 $XF \rightarrow XF+YF++YF-XF--XF\F-YF+$
 $YF \rightarrow -XF+YFYF++YF+XF--XF-YF$

Fig. 2. Examples of pictures generated by pL-systems under turtle interpretation. (a) The Sierpiński "arrowhead". (b) The Gosper space-filling curve. Both examples are taken from [Mandelbrot 1982].

Other graphical interpretation functions can also be considered. In the case of **chain interpretation** [Freeman 1961] letters **A**, **B**, **C** and **D** may be interpreted as commands moving the turtle to the left, up, to the right and down, respectively. These movements change position of the turtle by distance d , and are independent of the turtle's heading. Line segments connecting the old and the new position of the turtle are drawn. Under the chain interpretation, some interesting curves are generated by very simple L-

systems. For example, the dragon curve (Fig. 1e) is generated by a 0L-system with axiom **B** and productions **A** \rightarrow **AB**, **B** \rightarrow **CB**, **C** \rightarrow **CD** and **D** \rightarrow **AD**. Further extensions of the interpretation functions are also possible. For example, lines within a pair of parentheses may define the boundary of a filled polygon (Fig. 4). The turtle can also be allowed to move in three dimensions. An L-system will then describe a three-dimensional object rather than a two-dimensional picture.

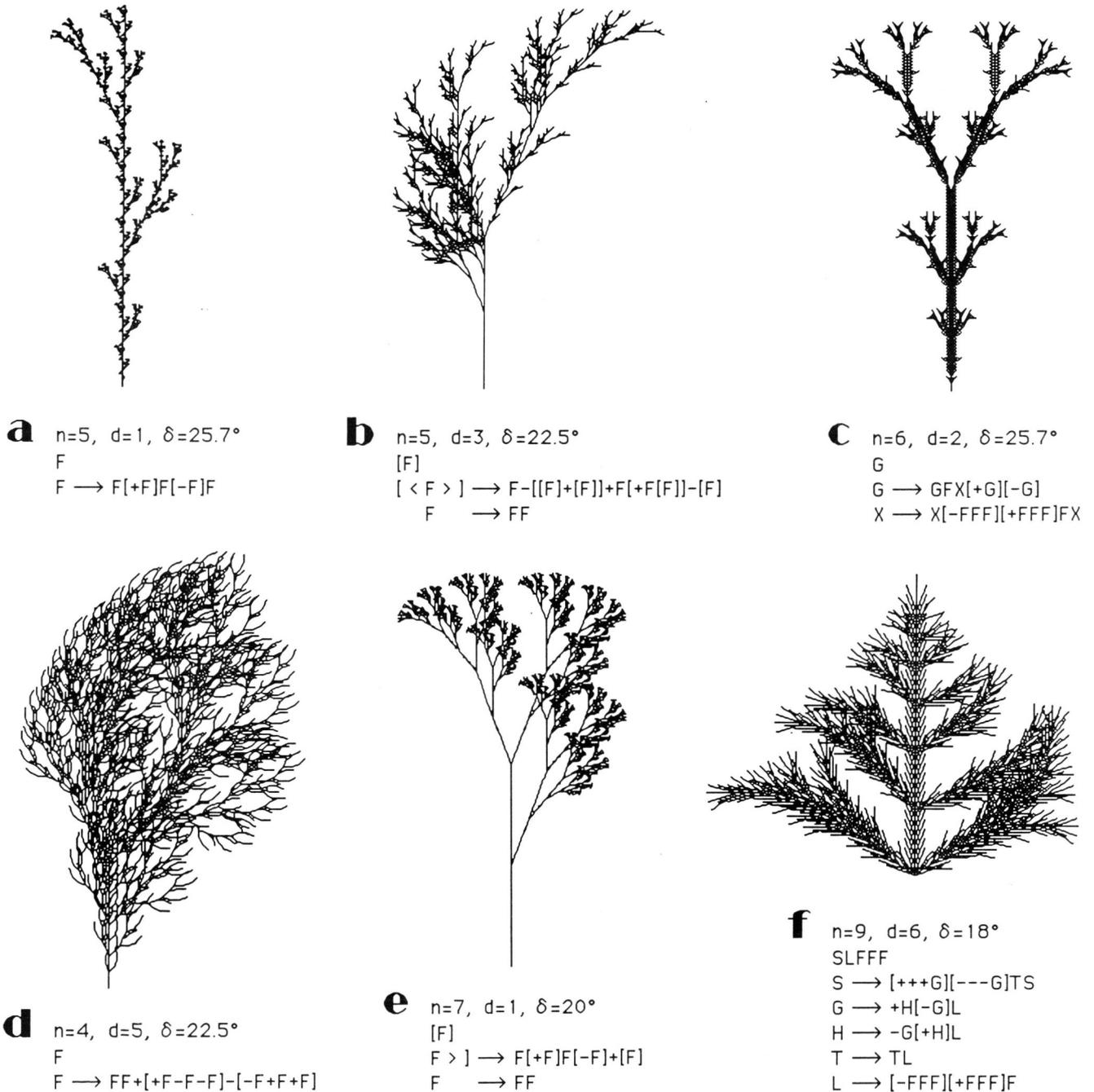


Fig. 3. Examples of plants and trees generated by 0L-systems and pL-systems under the extended turtle interpretation.



$n=4, d=8, \delta=30^\circ$
 T
 $T \rightarrow R+[T]--[L]R[+L]-[T]++T$
 $R \rightarrow F[--L][++L]F$
 $L \rightarrow \{[+FX-FX-FX+[+FX-FX-FX]\}$
 $FX \rightarrow FX$
 $F \rightarrow FF$

Fig 4. Example of a plant generated by a pL-system. Parentheses are grouping edges which define boundaries of filled polygons.

6. CONCLUDING REMARKS.

This paper presents a technique for generating pictures consisting of two steps:

- A string of symbols is generated with an L-system,
- This string is interpreted graphically as a sequence of commands controlling a turtle.

The notion of the L-system is generalized to include productions in which predecessors may have an arbitrary length. The L-systems generalized this way are called pL-systems. Furthermore, the turtle is equipped with a pushdown stack which allows it to return to a previously marked position. The resulting mathematical model is used to define a large variety of fractals ranging from simple Koch curves popularized by Mandelbrot [1982], to classic space-filling curves, to relatively realistic-looking plants and trees. All these pictures are defined in a uniform and compact way. Consequently, L-systems can be used to define fractals in a similar way that equations are used to define analytic curves in the Cartesian coordinates. In other words, the description of the Sierpiński arrowhead by the productions $XF \rightarrow -YF+XF+YF-$, $YF \rightarrow +XF-YF-XF+$ may be as "natural" as the description of a circle by the equations $x = R \cos\theta$, $y = R \sin\theta$.

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 are: Given two pL-systems and an interpretation function, are the resulting pictures congruent? Given a pL-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? Solutions to these problems are interesting not only from the theoretical point of view. They would also be useful when constructing pL-systems in order to generate pictures with given properties.

ACKNOWLEDGMENT.

The research described in this paper has been supported by grant number A0324 from the Natural Sciences and Engineering Research Council of Canada. This support is gratefully acknowledged.

REFERENCES.

- Abelson, H., and diSessa, A. A. [1982]: *Turtle geometry*. M.I.T. Press, Cambridge and London.
- Aono, M, and Kunii, T. L. [1984]: Botanical tree image generation. *IEEE Computer Graphics and Applications* 4, Nr. 5, pp. 10-34.
- Cole, A. J. [1983]: A note on space-filling curves. *Software - Practice and Experience* 13, pp. 1181-1189.
- Cole, A. J. [1985]: A note on Peano polygons and Gray codes. *Intl. Journal of Computer Mathematics* 18, pp. 3-13.
- Dacey, M. F. [1970]: The syntax of a triangle and some other figures. *Pattern Recognition* 2, pp. 11-31.
- Davis, C., and Knuth, D. E. [1970]: Number representations and dragon curves. *J. of Recreational Mathematics* 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.* 10, pp. 260-268.
- Fu, K. S. [1980]: Syntactic (linguistic) pattern recognition. In Fu K. S. (Ed.): *Digital pattern recognition*, Springer-Verlag, Berlin - Heidelberg - 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.
- Goldschlager, L. M. [1981]: Short algorithms for space-filling curves. *Software - Practice and Experience* 11, p. 99.
- Herman, G. T., and Rozenberg, G. [1975]: *Developmental systems and languages*. North-Holland, Amsterdam and Oxford.
- Hilbert, D. [1891]: Ueber stetige Abbildung einer Linie auf ein Flächenstück. *Math. Annln.* 38, pp. 459-460.

- Kirsch, R. A. [1964]: Computer interpretation of English text and picture patterns. *IEEE Trans. Electron. Comput.* **EC-13**, pp. 363-376.
- Ledley, R. S. [1964]: High-speed automatic analysis of biomedical pictures. *Science* **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. Tippett et al. (Eds.): *Optical and electro-optical information 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* **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* **54**, pp. 155-185.
- Narasimhan, R. [1962]: *A linguistic approach to pattern recognition*. Rep. 21, Digital Computer Laboratory, University of Illinois, Urbana.
- Narasimhan, R. [1966]: Syntax-directed interpretation of classes of pictures. *Comm. ACM* **9**, pp. 166-173.
- Null, A. [1971]: Space-filling curves, or how to waste time with a plotter. *Software - Practice and Experience* **1**, pp. 403-410.
- Papert, S. [1980]: *Mindstorms: children, computers, and powerful ideas*. Basic Books, New York.
- Peano, G. [1890]: Sur une courbe, qui remplit tout une aire plane. *Math. Annln.* **36**, pp. 157-160.
- Pfaltz, J. L. [1972]: Web grammars and picture description. *Computer Graphics and Image Processing* **1**, pp. 193-220.
- Rosenfeld, A., and Milgram, D. L. [1972]: Web automata and web grammars. *Machine Intelligence* **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* **14**, pp. 9-52.
- Sierpiński, W. [1912]: Sur une nouvelle courbe qui remplit tout une aire plane. *Bull. Acad. Sci. Cracovie, Série A*, pp. 462-478.
- Siromoney, R., and Subramanian, K. G. [1983]: Space-filling curves and infinite graphs. In H. Ehrig, M. Nagl and G. Rozenberg (Eds.): *Graph grammars and their application to computer science*, Springer-Verlag, Berlin - Heidelberg - New York - Tokyo.
- Smith, A. R. [1984]: Plants, fractals, and formal languages. *Computer Graphics* **18**, Nr. 3, pp. 1-10.
- Sudborough, I. H., and Welzl, E. [1985]: Complexity and decidability for chain code picture languages. *Theoretical Computer Science* **36**, pp. 173-202.
- Stiny, G. [1975]: *Pictorial and formal aspects of shape and shape grammars*. Birkhäuser-Verlag, Basel and Stuttgart.
- Wirth, N. [1976]: *Algorithms + data structures = programs*. Prentice-Hall, Englewood Cliffs.
- Witten, I. H., and Wyvill, B. [1983]: On the generation and use of space-filling curves. *Software - Practice and Experience* **13**, pp. 519-525.