



Repositorio Institucional de la Universidad Autónoma de Madrid

<https://repositorio.uam.es>

Esta es la **versión de autor** de la comunicación de congreso publicada en:
This is an **author produced version** of a paper published in:

IBM Journal of Research and Development 41.6 (1997): 727 – 736

DOI: <http://dx.doi.org/10.1147/rd.416.0727>

Copyright: © 1997 IBM

El acceso a la versión del editor puede requerir la suscripción del recurso
Access to the published version may require subscription

A study on the different representations of fractal curves by L systems and their equivalences

Authors:

- Manuel Alfonseca (Professor)

Address:

Universidad Autónoma
Dept. Ingeniería Informática
Ciudad Universitaria de Cantoblanco
28049 Madrid SPAIN
Telephone: (34 1) 397 4467

FAX: (34 1) 397 5277

Email: Manuel.Alfonseca@ii.uam.es

Formerly Senior Technical Staff Member at the IBM Madrid Scientific Center

- Alfonso Ortega (Graduate student, Universidad Autonoma of Madrid).

Email: AOrtega@neptuno.ii.uam.es

Formerly working in LAB2000.

Index terms

fractal, L system, turtle graphic, vector graphic, parallel derivation grammar, Lindenmayer.

Abstract

To represent fractals by means of L systems, a graphic interpretation of the L system is required. Two families of graphic interpretations have been used: turtle graphics and vector graphics. Both are proved to be equivalent for two interesting families of L systems that include many of the fractals in the literature. The equivalence theorems make it possible to start from one L system in one of the families and obtain other systems that represent the same fractal. Sometimes, a fractal that was assumed not to be representable by an L system in one of the families can be shown to be in this way. Another point shown is the fact that supposed deficiencies in L systems, that have prompted proposals of extensions, are really deficiencies in the graphic translation scheme.

Introduction

This paper looks at two kinds of objects: fractal objects (mathematical constructs), and L systems (a type of formal grammars). Both are known to be intimately related, in the sense that some types of fractals are easily represented by L systems by means of a given graphic interpretation. However (and this is not always done in the literature) one must separate clearly between the L system itself and the graphic interpretation that generates the fractal, otherwise deficiencies in the latter may be ascribed to the former.

The paper introduces fractals, L systems and their relationship, looks at the two most used graphic interpretations, uses them to classify the set of fractals and proves the equivalence between two interesting classes. Some examples of derivation of equivalent L systems are given.

Fractals

Fractal objects, as defined in 1975 by Benoit B. Mandelbrot [ref. 3, 9], have certain special properties, such as self-similarity (containing copies of themselves), underivability at every point, and a Hausdorff dimension greater than their geometric dimension. They are appropriate for the description of natural shapes, and have been used successfully to code and compress images [ref. 1, 2, 4]. Fractals have been generated or represented by many different means, including Recursive mathematical families of equations. In the Mandelbrot set, the fractal curve is the limit between the domains of convergence and divergence.

- Recursive transformations (generators) applied to an initial shape (the initiator).
- Fractional Brownian movements.

In this paper, we are mainly interested in the second family of fractal objects, which have been variously represented by means of L systems and their different extensions, geometric recursion systems, iterated function systems, mutually recursive function systems, and the like.

L systems

In 1968, Aristid Lindenmayer [ref. 8] defined a new type of grammar (the parallel derivation grammar) differing from normal Chomsky grammars because derivation is not sequential (a single rule is applied at every step), but parallel (as many rules as possible are applied at every step).

Parallel derivation grammars, also called L systems, can be classified in various ways:

- Context sensitive IL systems.
- Context free OL systems.
- Deterministic DL systems.
- Propagative PL systems.
- EL systems, with extensions.
- TL systems, with tables.

The previous types may be combined. In this way, a D0L system is a deterministic context free L system; a PD0L system is propagative, deterministic and context free; an EIL system is context sensitive with extensions; and so forth.

A D0L system is defined as the three-fold (Σ, P, w) , where Σ is an alphabet (a finite non-empty set of symbols); P is a set of production rules of the form $A : \Rightarrow x$, where $A \in \Sigma$ is a symbol in the alphabet and $x \in \Sigma^*$ is a (possibly empty) word or string of symbols in the alphabet; and $w \in \Sigma^*$ is the starting word or axiom.

In a D0L system, every symbol appears exactly once at the left of a production rule. This restriction makes the system deterministic.

A D0L scheme is the two-fold (Σ, P) , of an alphabet and a set of production rules, and represents the family of all the D0L systems sharing those two components and differing in the axiom.

An example of a D0L system is [L-System 1]:

$$(\{F, +, -\}, P, F++F++F)$$

where P is the following set of production rules:

$$F : \Rightarrow F-F++F-F$$

$$+ : \Rightarrow +$$

$$- : \Rightarrow -$$

A derivation of a word in a D0L system is the new word obtained when each symbol in the first word is replaced by the right part of the production rule whose left part is that symbol. In the previous example, we can get the following derivation from the axiom:

$$F++F++F \rightarrow F-F++F-F++F-F++F-F++F-F$$

The word obtained can become the starting point of a new derivation, and so on.

L systems have been successfully applied in the simulation of biologic processes, such as plant growth, development of leaves, pigmentation of snail shells, etc. [ref. 7].

Fractals and L systems

L systems are very appropriate to represent fractal objects obtained by means of recursive transformations [ref. 5]. The initiator maps to the axiom of the L system, the generator becomes the production rules, while the recursive applications of the generator to the initiator correspond to the successive derivations of the axiom. Something else is needed, however: a graphic interpretation that makes it possible to convert each of the words generated by the L system into a visible fractal object.

It is very important to separate the L system from its associated graphic interpretation. Otherwise, a problem in the latter may be mistaken for a deficiency in the former. This has happened before in the literature in this area, and extensions to L systems have been proposed [ref. 6, 11] in cases where classic L systems are appropriate, and the problem lays in an incomplete graphic interpretation.

Two different families of graphic interpretations of L systems have been used:

- Turtle graphics
- Vector graphics

Turtle graphics

Created in 1980 by Seymour Papert [ref. 10], the graphic is the trail left by an invisible "turtle", whose state at every instant is defined by its position and the direction it is looking to. The state of the turtle changes as it moves a step forward, or as it rotates a given angle in the same position.

Turtle graphics interpretations come in different levels of complexity. In the simplest one, the alphabet of a DOL system consists of just three symbols:

$$\Sigma = \{ F, +, - \}$$

The graphics interpretation of a word is as follows

- | | |
|----------|---|
| F | The turtle moves one step forward, in the direction it is looking at, leaving a visible trail. We will call F a <i>draw</i> letter. |
| + | The turtle rotates a positive angle α . |
| - | The turtle rotates a negative angle α . |

With this interpretation, if we take $\alpha=60$ degrees, L-system 1 generates the well-known Koch snowflake curve.

Additional rules complicate the turtle graphics and make it possible to generate fractals of different families. For instance:

- Upper case letters different from F have no graphic representation and leave the state of the turtle unchanged. In the following, we will call such letters *non-graphic* letters.
- Lower case f makes the turtle move a step forward, with no visible trail. In the following, we will call letters such as f *move* letters.
- An open parenthesis pushes the state of the turtle into a stack; a closed parenthesis pops the top of the stack and restores the turtle state. This rule makes it possible to represent branching fractals. Of course, we are only interested in strings where open and closed parentheses are paired according to the usual syntax rules (a closed parenthesis without a previous open one would generate an empty stack exception).
- Additional *draw* and *move* letters may be defined to act the same as F and f .
- Symbol $!$ makes the turtle rotate 180 degrees.
- Braces $\{ \}$ indicate that the area enclosed in the braces must be filled with some color.
- Other symbols may be defined to change colors.

In this paper, we are interested in all the indicated extensions except the last three.

In summary: a given fractal may be represented by means of three components: an L system, a concrete turtle interpretation, a distance step, and an angle step. The distance step is unnecessary, if appropriate scale factors are applied.

Turtle graphics are very flexible. Appropriate extensions make it possible to represent complex information (such as branching, color filling, and so forth). On the other hand, this representation is inherently slow: the turtle state at any point is a non-trivial function of the complete history of the previous turtle movements. Therefore, a string must be converted sequentially into pixel positions by means of a complex loop.

Vector graphics

In this family of interpretations, every symbol in the alphabet of the L system is associated to a vector in a rectangular cartesian system. A word (a string of symbols) is represented by the catenation of the vectors of the symbols that make the word.

In the simplest case, a fractal may be defined by two components: an L system, and a mathematical application $V:\Sigma\rightarrow\mathfrak{R}^2$. (the vector interpretation). We assume that all vectors associated to the symbols in the alphabet produce visible movements. The graphic representation of each derivation of the L system is a set of straight connected segments.

This vector interpretation allows us to represent branching fractals, since it is always possible to return to the start of the branch if for every symbol in the alphabet there is another associated to the opposite vector. However, a different vector interpretation is needed to build fractals that are not connected, such as the Cantor sets. This can be covered by means of a strict extension to the vector interpretation, which replaces V by the mathematical application: $V:\Sigma\rightarrow\{0,1\} \times \mathbb{R}^2$, which includes, for each symbol, a visibility coefficient (a 0 or a 1), indicating that the vector displacement should be visible (a 1) or invisible (a 0). This extension makes it also possible to represent branching fractals.

Vector graphics are less flexible than turtle graphics. Complex extensions, such as area filling and coloring are not easy. On the other hand, vector composition is a straightforward operation and can be performed with a very simple loop, which means that vector graphics are usually faster than turtle graphics.

A graphical classification of D0L Systems and Schemes

We propose a classification of D0L Systems and Schemes, using as criteria the graphical representations they must be subject to, to represent fractal curves.

In the following, we will consider the following turtle graphics interpretation (T, α) :

The alphabet of the L System can be expressed as the union of the four disjoint subsets: $N, D, M, \{+, -, (,)\}$.

- $+$ increases the turtle angle by α .
- $-$ decreases the turtle angle by α .
- $($ pushes the turtle state into a stack.
- $)$ pops and restores the turtle state from the stack.
- $A \in N$ leaves the turtle state unchanged.
- $F \in D$ moves the turtle one step forward, in the direction of its current angle, leaving a visible trail.
- $f \in M$ moves the turtle one step forward, in the direction of its current angle, with no visible trail.
- $\alpha = 2k\pi/n$, where k and n are two integers.

We call TGD0L to the set of all the D0L systems (schemes) that represent some fractal (family of fractals) by means of the turtle graphics interpretation (T, α) .

We call VGD0L to the set of all the D0L systems (schemes) that represent some fractal (family of fractals) by means of a vector graphics interpretation.

Definition 1: Two L systems are *fractal-equivalent* if they represent the same fractal curve by means of some graphic interpretation.

Definition 2: Two L schemes are *fractal-equivalent* if, for each L system in the first one (i.e. a given axiom), there exists a fractal-equivalent L system in the second one (with a possibly different axiom).

As we will see later, two L systems may be fractal-equivalent while their respective L schemes are not. Two D0L systems may be fractal-equivalent in any combination (i.e. both may be TGD0L, or VGD0L, or one each).

Given an L system in TGD0L or VGD0L, that represents a fractal with a given graphic interpretation, we are interested in finding a fractal-equivalent L system in the opposite set, for the following reasons:

- Since vector graphics are usually faster than turtle graphics, given a TGD0L system, we may be interested in finding a fractal-equivalent VGD0L system for performance reasons.
- Since turtle graphics are more flexible than vector graphics, given a VGD0L system, we may be interested in finding a fractal-equivalent TGD0L system if we want to fill areas or apply different colors to different sections of the fractal.

A general transformation algorithm from any TGD0L scheme to a fractal-equivalent VGD0L scheme may not be possible. However, we have proved an equivalence theorem between two interesting subsets of both sets, which we define as follows:

Definition 3: A string under a turtle graphics interpretation is said to be *angle-invariant* if the direction of the turtle at the beginning of the string is the same as its direction at the end of the string.

Since the only symbols in the turtle graphics interpretations we are considering that change the turtle direction are $+$ $-$ $)$, a string in an L system in TGD0L, whose incremental angle α is $2k\pi/n$, is angle-invariant if parentheses are paired in the usual way (we are only interested in this) and the number of plus signs minus the number of minus signs outside the parenthesis blocks is either zero (the simplest case) or a multiple of n (which would allow for an integer number of full circles). Inside paired parentheses, there are no restrictions to the number or combinations of $+$ $-$ signs.

Definition 4: An L scheme in TGD0L is said to be *angle-invariant* if the right hand side of all its rules is an angle-invariant string.

We call AITGD0L the set of all angle-invariant schemes in TGD0L. Most of the interesting TGD0L systems in the literature (but not all) are AITGD0L. As an example of a TGD0L system that is not AITGD0L, let us mention the following [L-System 2]:

$$(\{F, G, +, -\}, P, F)$$

where P is the following set of production rules:

$$\begin{aligned} F &::= F+G \\ G &::= F-G \\ + &::= + \\ - &::= - \end{aligned}$$

Using the typical turtle graphics interpretation, with $\{F, G\}$ the set of draw symbols and $\alpha=90$ degrees, this L system describes the well-known dragon fractal. It is clear that the strings $F+G$ and $F-G$ are not angle invariant. However, we will show later (applying one of our equivalence theorems) that there exists an AITGD0L system that is fractal-equivalent to L-System 2.

Definition 5: A set of real numbers is said to be *rationally-related* if the quotient of any two of them is a rational number.

For any rationally-related set of real numbers, there must exist a real number r , such that all the numbers in the set are integer multiples of r . (The proof is trivial).

Definition 6: An L scheme in VGD0L with a vector graphics interpretation VI is said to be *rationally-related* if both the set of the modules and the set of the angles of all the vectors in VI are rationally-related.

For any rationally-related scheme with a vector graphics interpretation VI, there exist two real numbers, r and α , such that all the modules of the vectors in VI are positive integer multiples of r , and all the angles of the vectors in VI are positive integer multiples of α .

We call RRVGD0L the set of all rationally-related schemes in VGD0L. Most of the interesting VGD0L systems in the literature are RRVGD0L (we haven't found one that isn't).

The two sets AITGD0L and RRVGD0L are interesting, in the sense that many fractals usually represented by L systems belong to them. We are now going to prove that those sets are fractal-equivalent.

Theorem 1

For every AITGD0L system which represents a fractal with the usual turtle graphics interpretation and $\alpha=2k\pi/n$, there exists a fractal-equivalent RRVGD0L system.

For every AITGD0L scheme which represents a set of fractals with the usual turtle graphics interpretation and $\alpha=2k\pi/n$, there exists a fractal-equivalent RRVGD0L scheme.

Constructive proof

Informal description:

The algorithm we propose starts from an AITGD0L system and builds a fractal-equivalent RRVGD0L system. Every symbol in the AITGD0L system splits into n varieties in the RRVGD0L system (one per possible angle in the turtle state). All vectors in the target system are unitary (their module is 1), and all their angles are a multiple of α , therefore the resulting system is RRVGD0L. The strings generated by each variety are rotated appropriately. A plus sign in a string (a rotation of α) becomes a change from variety i to variety $i+1$ for every symbol after the sign. A minus sign changes variety i to variety $i-1$. In this way, the direction of movement of the turtle is replicated by the vectors. The effect of parentheses pairs (branching) is simulated by moving back to the branching point with a second set of opposite vectors (n per symbol). This set is unnecessary if there are no parentheses in the rules of the source system.

Formal description:

Assume the following AITGD0L system:

$$L = (\Sigma, P, w)$$

$$\Sigma = N \cup D \cup M \cup \{+, -, (,)\}$$

N = a set of *non-graphic* letters

D = a set of *draw* letters

M = a set of *move* letters

$$+ ::= + \in P$$

$$- ::= - \in P$$

$$(::= (\in P$$

$$) ::=) \in P$$

$$s ::= x(s) \in P \text{ for all } s \in N \cup D \cup M$$

$$\text{where } x(s) \in \Sigma^*.$$

We build another L system and a vector interpretation represented by the two-fold (L', V) where

$L' = (\Sigma', P', w')$ is a D0L system, such that for each $s \in N \cup D \cup M$

$$s'(i) \in \Sigma' \ (i \in Z_n)$$

$$s''(i) \in \Sigma' \ (i \in Z_n)$$

$$s'(i) ::= C'[x(s), i, 0] \in P'$$

$$s''(i) ::= C''[x(s), i, 0] \in P'$$

where $C'[x, i, k] : \Sigma^* \times Z_n^2 \rightarrow \Sigma^*$, where Z_n is a finite group of n elements (the set of integers module n). The transformation C' is recursively defined as follows:

$$C'[\lambda, i, k] = \lambda \text{ for all } i, k$$

$$C'[s, i, k] = s'(i+k)$$

$$\text{for all } s \in N \cup D \cup M$$

$$C'[s.y, i, k] = C'[s, i, k].C'[y, i, k]$$

$$\text{for all } s \in N \cup D \cup M, y \in \Sigma^*$$

$$C'[+, i, k] = \lambda \text{ for all } i, k$$

$$C'[-, i, k] = \lambda \text{ for all } i, k$$

$$C'[+.y, i, k] = C'[y, i, k+1]$$

$$\text{for all } y \in \Sigma^*$$

$$C'[-.y, i, k] = C'[y, i, k-1]$$

$$\text{for all } y \in \Sigma^*$$

$$C'[(.x.), i, k] = C'[x, i, k].C''[f(x), i, k]^{-1}$$

where $f(x)$ is what remains of x after we have eliminated all parentheses pairs and whatever is inside them.

The transformation C'' is defined exactly as C' , replacing every occurrence of $s'(i)$ by $s''(i)$ and vice versa.

The axiom, $w' = C'[w, 0, 0]$.

Finally, the vector interpretation V is defined as follows:

- For all $s \in N$, and all $i \in Z_n$,

$$V(s'(i)) = V(s''(i)) = (0, 0, 0)$$
- For all $s \in D$, and all $i \in Z_n$,

$$V(s'(i)) = (1, \cos(i.\alpha), \sin(i.\alpha))$$

$$V(s''(i)) = (1, -\cos(i.\alpha), -\sin(i.\alpha))$$
- For all $s \in M$, and all $i \in Z_n$,

$$V(s'(i)) = (0, \cos(i.\alpha), \sin(i.\alpha))$$

$$V(s''(i)) = (0, -\cos(i.\alpha), -\sin(i.\alpha))$$

where the vector (v, x, y) consists of a visibility $v \in \{0, 1\}$ and two cartesian coordinates x, y .

If Σ does not include the two parentheses, the set of symbols $s''(i)$ can be eliminated from L' .

It is easy to prove, by inspection of the algorithm, that the fractal curve represented by (L', V) is the same represented by L with the standard turtle graphic interpretation. The second part of the theorem (its application to schemes) is trivial, since for every system (axiom) used with the source scheme, the alphabet and the production rules in the target system are the same and only the target axiom changes (i.e. all the target systems belong to the same scheme).

Application example for theorem 1

We have seen that L-system 1 generates the Koch snowflake curve with the simplest turtle graphics interpretation, and $\alpha=60$ degrees. Its axiom is $F++F++F$, and its rules are:

$$\begin{aligned} F &::= F-F++F-F \\ + &::= + \\ - &::= - \end{aligned}$$

In this case, $n=6$, $k=1$. Applying the above algorithm, this system can be converted to the equivalent (L', V) two-fold, where L' is the following D0L system:

$$(\{A, B, C, D, E, F\}, P, ACE)$$

P is the set of production rules:

$$\begin{aligned} A &::= AFBA \\ B &::= BACB \\ C &::= CBDC \\ D &::= DCED \\ E &::= EDFE \\ F &::= FEAF \end{aligned}$$

where we have renamed the symbols in the new alphabet in the following way:

$$\begin{aligned} A &= F'(0) \\ B &= F'(1) \\ C &= F'(2) \\ D &= F'(3) \\ E &= F'(4) \\ F &= F'(5) \end{aligned}$$

V is the following vector interpretation:

Symb	Vis	X	Y
A	1	1	0
B	1	0.5	r
C	1	-0.5	r
D	1	-1	0
E	1	-0.5	-r
F	1	0.5	-r

where r is one half of the square root of 3.

It is easy to see that this system represents the same fractal, whose fifth iteration appears in figure 1.

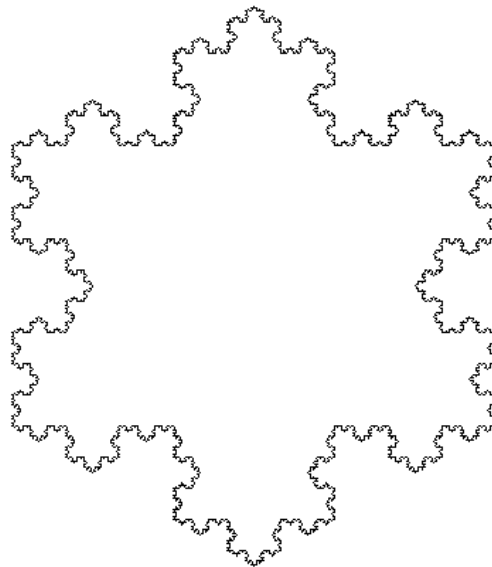


Figure 1

Theorem 2

For every RRVGD0L system which represents a fractal with a vector graphics interpretation, there exists a fractal-equivalent AITGD0L system.

For every RRVGD0L scheme which represents a set of fractals with a vector graphics interpretation, there exists a fractal-equivalent AITGD0L scheme.

Constructive proof

Informal description:

The algorithm we propose starts from an RRVGD0L system and builds a fractal-equivalent AITGD0L system. Every symbol in the RRVGD0L system splits into two varieties in the AITGD0L system, one that propagates the derivations of the original symbol, the other that disappears in the next derivation, whose purpose is to draw the vector to its full length. (If all the vectors are the same length, the second variety is not needed). All symbols in the target system start by moving the turtle angle to the corresponding vector angle (with the appropriate number of + or - signs), then generate the vector drawing or movement, finally return to the null angle (with the opposite set of + and - signs). The result is obviously an AITGD0L system without branching (parentheses).

Formal description:

Assume the following RRVGD0L system:

$$L=(\Sigma, P, w)$$

with the vector graphic interpretation $V:\Sigma \rightarrow \{0,1\} \times \mathbb{R}^2$. Remember that, for any RRVGD0L scheme, there exist two real numbers, r and α , such that all the modules of the vectors are positive integer multiples of r , and all their angles are positive integer multiples of α .

We perform the following operations:

- Every vector in V is converted to polar coordinates. This gives us a set of modules R and a set of angles, A .
- Compute the two numbers, r and α .
- Build the D0L system $L' = (\Sigma', P', w')$, such that for each $s \in \Sigma$, $s' \in \Sigma'$, $s'' \in \Sigma'$, and for each
$$s ::= x(s) \in P,$$
$$s' ::= C[x(s), \alpha(s)] \in P'$$
$$s'' ::= \lambda \in P'$$

where $\alpha(s)$ is the angle of the vector associated to symbol s in $V(s)$, and the transformation

$C[x,p] : \Sigma^* \mathbb{R} \rightarrow \Sigma'^*$, is recursively defined as follows:

$$C[\lambda, p] = \lambda \text{ for all } p$$

$$C[s, p] = A.B.C$$

for all $s \in \Sigma$,

where s is

not

associated to

and

$$A = \begin{cases} \alpha(s) \geq p : \text{a string of } (\alpha(s)-p)/\alpha + \text{signs} & \text{the null vector} \\ \alpha(s) < p : \text{a string of } (p-\alpha(s))/\alpha - \text{signs} \end{cases}$$

$B = s'$ catenated to a string of $(r(s)/r)-1$ copies of symbol s''

$$C = \begin{cases} \alpha(s) \geq p : \text{a string of } (\alpha(s)-p)/\alpha - \text{signs} \\ \alpha(s) < p : \text{a string of } (p-\alpha(s))/\alpha + \text{signs} \end{cases}$$

$$C[s,p] = s'$$

for all $s \in \Sigma$, where s is associated to the null vector

$$C[s,y,p] = C[s,p].C[y,p]$$

for all $s \in \Sigma$, $y \in \Sigma^*$

- We separate the set Σ' in the following way:

$$\Sigma' = N \cup D \cup M \cup \{+, -\}$$

$$N = \{s', s'' \mid s \text{ is associated to the null vector}\}$$

(*non-graphic* letters)

$$D = \{s', s'' \mid s \text{ has 1 visibility and is not associated to the null vector}\}$$

(*draw* letters)

$$M = \{s', s'' \mid s \text{ has 0 visibility and is not associated to the null vector}\}$$

(*move* letters)

- We add the following rules to P' :

$$+ ::= + \in P'$$

$$- ::= - \in P'$$

- The axiom, $w' = C[w,0]$.

It is easy to see that the fractal curve represented by the original two-fold (L, V) , is also represented by the three-fold (L', T, α) , where T is the standard turtle graphics interpretation and α (the value used above) is the elemental angle step. The second part of the theorem (its application to schemes) is trivial, since for every system (axiom) used with the source scheme,

the alphabet and the production rules in the target system are the same and only the target axiom changes (i.e. all the target systems belong to the same scheme).

Application example for theorem 2

The L system obtained by the application of theorem 1 can now be subject to the algorithm of theorem 2 to build a different equivalent D0L system with turtle graphics interpretation. By inspection of the six vector, we find that $r=1$ and $\alpha=60$ degrees or $\pi/3$ radians. $\Sigma'=\{A', B', C', D', E', F', +, -\}$. Applying the algorithm we find that P' contains the following production rules:

$$\begin{aligned} A' &::= A' - F' ++ B' - A' \\ B' &::= B' - A' ++ C' - B' \\ C' &::= C' - B' ++ D' - C' \\ D' &::= D' - C' ++ E' - D' \\ E' &::= E' - D' ++ F' - E' \\ F' &::= F' - E' ++ A' - F' \\ + &::= + \\ - &::= - \end{aligned}$$

All of the symbols, A' , B' , C' , D' , E' , F' , are draw symbols. The sets of move and non-graphic symbols are empty in this case.

The axiom becomes $A' ++ C' ---- E' ++$.

It is clear that this system also represents the same fractal, whose fifth iteration is shown in figure 1. In fact, it is easy to see by direct observation that all the draw symbols are really the same, and this system is fractal-equivalent to the original one.

Other examples

Three of the next four examples have been designed by different authors and are common in the literature. The third one is our own.

- The Cantor set of the third order is represented by the following D0L system:

$$(\{F, f\}, P, F)$$

where P is the set of production rules:

$$\begin{aligned} F &::= FfF \\ f &::= fff \end{aligned}$$

with the standard turtle graphics interpretation, where F is a draw symbol, f is a move symbol, and the value of α is immaterial. Figure 2 shows the axiom and the first five derivations of this system.



Figure 2

The same fractal may be generated by means of the equivalent D0L system:

$$(\{A, B\}, P, A)$$

where P is the following set of rules:

$$A ::= ABA$$

$$B ::= BBB$$

with the following vector interpretation:

Symb	Vis	X	Y
A	1	1	0
B	0	1	0

The two preceding fractals can be obtained from one another by application of the two equivalence theorems.

- L-system 2 (representing the dragon fractal) is not an AITGD0L system, therefore theorem 1 is not applicable. However, by observation of the derivations of the fractal, we can find the following fractal-equivalent VGD0L system:

$$A ::= AB$$

$$B ::= CB$$

$$C ::= CD$$

$$D ::= AD$$

with the following vector interpretation:

Symb	Vis	X	Y
A	1	1	0
B	1	0	1
C	1	-1	0
D	1	0	-1

This system is RRVGD0L. Therefore, theorem 2 is applicable, which makes it possible to obtain the following fractal-equivalent AITGD0L system:

$$(\{F, G, +, -\}, P, F)$$

where P is the following set of production rules:

$$\begin{aligned} F &::= F+G- \\ G &::= +H-G \\ H &::= H+I- \\ I &::= +F-I \\ + &::= + \\ - &::= - \end{aligned}$$

Figure 3 shows the tenth derivation of this system. It should be observed that, although L-system 2 is fractal-equivalent to the two systems given above, the corresponding schemes are not equivalent (the last two are, of course, since they are the result of the application of theorem 2). Therefore, schemes may not be equivalent even when some systems in them are. We have not proved that an AITGD0L scheme fractal-equivalent to L-system 2 does not exist, but have a strong suspicion that this is the case.

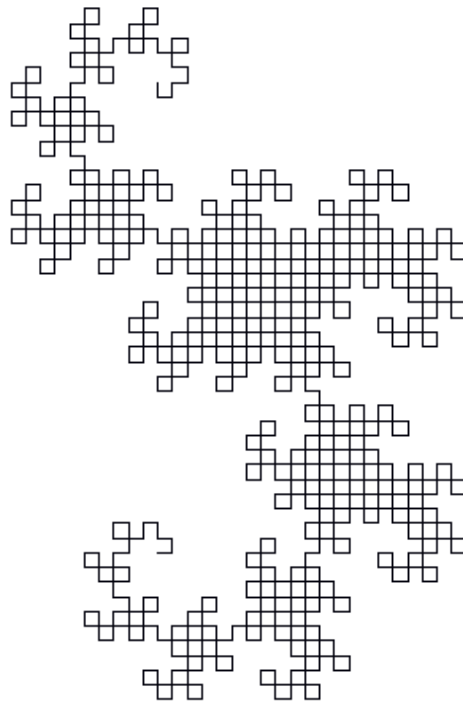


Figure 3

- The PDOL scheme

A ::= ABB

B ::= BC

C ::= CD

D ::= DAC

with axiom CCCC, a vector graphic interpretation and the following two column vector definition:

Symb	Vis	X	Y
A	1	8	0
B	1	0	0.75
C	1	-4	0
D	1	0	-0.75

generates a fractal curve whose tenth derivation is uncannily similar to a human hand (see figure 4).

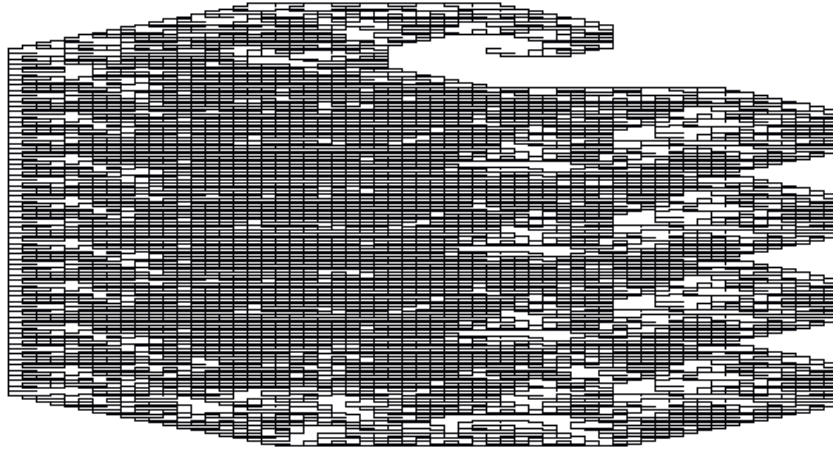


Figure 4

By applying theorem 2, we obtain the following L system that generates the same fractal with a turtle graphics interpretation:

- $r = 0.25$
- $\alpha = 90$ degrees or $\pi/2$ radians
- $\Sigma' = \{A', A'', B', B'', C', C'', D', D'', +, -\}$.
- P' contains the following production rules:

$$\begin{aligned}
 A' &::= A' A''^{31} + B' B'' B'' B' B'' B'' - \\
 B' &::= B' B'' B'' + C' C''^{15} - \\
 C' &::= C' C''^{15} + D' D'' D'' - \\
 D' &::= D' D'' D'' + A' A''^{31} - - C' C''^{15} + \\
 A'' &::= \lambda \\
 B'' &::= \lambda \\
 C'' &::= \lambda \\
 D'' &::= \lambda \\
 + &::= + \\
 - &::= -
 \end{aligned}$$

where the exponents represent symbol repetition.

- All of the symbols, A' , A'' , B' , B'' , C' , C'' , D' , D'' , are draw symbols. The sets of move and non-graphic symbols are empty in this case.
- The axiom is $++C' C''^{15} C' C''^{15} C' C''^{15} C' C''^{15} - -$, where each set of C'' includes 15 items.

If we apply theorem 1 to the previous result, we would get another fractal-equivalent RRVGD0L system, where all the vectors have the same module.

- The following D0L system:

$$(\{A, F, +, -, (,)\}, P, A)$$

where P is the set of rules:

$$\begin{aligned} A &::= F - ((A) + (A)) + F (+ F (A)) - (A) \\ F &::= FF \\ + &::= + \\ - &::= - \\ (&::= (\\) &::=) \end{aligned}$$

represents a fractal whose fifth derivation is shown in figure 5, taking $\alpha=22.5$ degrees. Both A and F are *draw* letters. This system is not AITGD0L, but we mention it because, in a paper published in 1986, P. Prusinkiewicz [ref. 11] asserted that this fractal, and others like it, could not be represented by means of 0L systems, and proposed an extension (pL systems) to make it possible. The fact that we have been able to represent it with a D0L system indicates that the extension may be unnecessary: 0L systems are sufficiently powerful to represent these curves. The restriction was not in them, but in the turtle graphics interpretation.

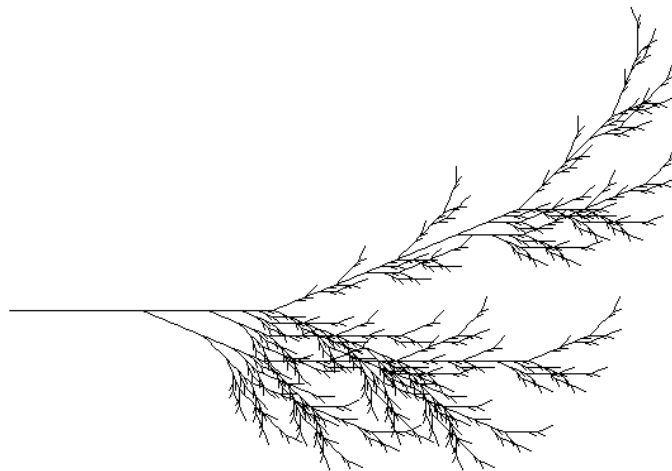


Figure 5

Conclusion

D0L systems, given an appropriate graphic interpretation, are quite powerful to represent large families of fractal curves, even some that previous authors had assumed required nonstandard

extensions. It is thus important to isolate the D0L system from its graphic interpretation, which can be very different and belong either to the turtle or the vector family.

We have proved a fractal-equivalence theorem between two families of L systems, one associated with a turtle graphics interpretation, the other with vector graphics. The two families are interesting because most of the fractals in the literature can be represented by means of them. The two theorems make it possible to obtain representations in those families for fractals that were assumed not to belong to them.

References

1. Barnsley, M.F.: "Fractals everywhere", Academic Press, Boston, 1988.
2. Bedford, T.; Dekking, F.M.; Breeuwer, M.; Keane, M.S.; van Schooneveld, D.: "Fractal Coding of Monochrome Images", *Signal Processing: Image Communication*, Vol 6, pp. 405-419, 1994.
3. Casey, S.D.; Reingold, N.F.: "Self-similar Fractal Sets: Theory and Procedure", *IEEE Computer Graphics and Applications*, pp. 73-82, May 1994.
4. Culik II, K.; Dube, S.: "New Methods for Image Generation and Compression", *Proc. Conf. on Facts and New Trends in Computer Science*, ed. H. Maurer, Springer-Verlag, Berlin, 1991, pp. 69-90.
5. Culik II, K.; Dube, S.: "L-Systems and Mutually Recursive Function Systems", *Acta Informatica*, Vol 30, pp. 279-302, 1993.
6. Giessmann, E.G.: "Generation of fractal curves by generalizations of Lindenmayer's L-systems". *Proc. 1st IFIP Conf. on Fractals in the Fundamental and Applied Sciences*, ed. H.O. Peitgen, J.M. Henriques, L.F. Penedo, North Holland, Amsterdam, 1991, pp.147-157.
7. Kaandorp, J.: "Fractal Modelling. Growth and Form in Biology", Springer, Berlin, 1994.
8. Lindenmayer, A.: "Mathematical models for cellular interactions in development" (two parts). *J. Theor. Biol.* 18,. pp. 280-315, 1968.
9. Mandelbrot, B.B.: *The Fractal Geometry of Nature*. W.H.Freeman, San Francisco, 1982.
10. Papert, S.: "Mindstorms: Children, Computers, and Powerful Ideas", Basic Books, New York, 1980.
11. Prusinkiewicz, P.: "Graphical Applications of L-Systems", *Proc. Graphics Interface 86 and Vision Interface 86*, ed: M. Wein, E.M. Kidd, Vancouver, Canada, 1986, pp. 247-253.