

Self-Similarity In Plants: Integrating Mathematical And Biological Perspectives

Przemyslaw Prusinkiewicz

Department of Computer Science, University of Calgary

2500 University Drive N.W., Calgary, Alberta, Canada T2N 1N4

Email: pwp@cpsc.ucalgary.ca

Abstract

Self-similarity is a conspicuous feature of many plants. Geometric self-similarity is commonly expressed in terms of affine transformations that map a structure into its components. Here we introduce topological self-similarity, which deals with the configurations and neighborhood relations between these components instead. The topological self-similarity of linear and branching structures is characterized in terms of recurrence systems defined within the theory of L-systems. We first review previous results, relating recurrence systems to the patterns of development that can be described using deterministic context-free L-systems. We then show that topologically self-similar structures may become geometrically self-similar if additional geometric constraints are met. This establishes a correspondence between recurrence systems and iterated function systems, which is of interest as a mathematical link between L-systems and fractals. The distinction between geometric and topological self-similarity is useful in biological applications, where topological self-similarity is more prevalent than geometric self-similarity.

Reference

P. Prusinkiewicz: Self-similarity in plants: Integrating mathematical and biological perspectives. In M. Novak (Ed.): *Thinking in Patterns. Fractals and Related Phenomena in Nature*. World Scientific, Singapore, pp. 103–118.

SELF-SIMILARITY IN PLANTS: INTEGRATING MATHEMATICAL AND BIOLOGICAL PERSPECTIVES

PRZEMYSŁAW PRUSINKIEWICZ

*Department of Computer Science, University of Calgary
2500 University Drive N.W., Calgary, Alberta, Canada T2N 1N4
E-mail: pwp@cpsc.ucalgary.ca*

Self-similarity is a conspicuous feature of many plants. Geometric self-similarity is commonly expressed in terms of affine transformations that map a structure into its components. Here we introduce topological self-similarity, which deals with the configurations and neighborhood relations between these components instead. The topological self-similarity of linear and branching structures is characterized in terms of recurrence systems defined within the theory of L-systems. We first review previous results, relating recurrence systems to the patterns of development that can be described using deterministic context-free L-systems. We then show that topologically self-similar structures may become geometrically self-similar if additional geometric constraints are met. This establishes a correspondence between recurrence systems and iterated function systems, which is of interest as a mathematical link between L-systems and fractals. The distinction between geometric and topological self-similarity is useful in biological applications, where topological self-similarity is more prevalent than geometric self-similarity.

1 Introduction

In her 1950 book, *Natural Philosophy of Plant Form*¹, the eminent British botanist Agnes Arber² wrote (p. 7):

It is well to return, even at long last, to such early work as is notably rich in content, to see whether it still offers suggestions, which formerly passed unheeded because the time was not ripe for them, but which the intellectual climate would now foster. Originality is so rare in the human mind that we need to harvest it to the last gleanings.

In this paper, we follow Arber's suggestion by revisiting the notion of self-similarity in plants. Mandelbrot³ defined self-similarity by referring to an underlying generative process (such as the Koch construction) as follows:

When each piece of a shape is geometrically similar to the whole, both the shape and the cascade that generates it are called self-similar.

Selected plant structures, such as the inflorescences of cauliflower and broccoli, compound fern leaves, and branching structures of trees, are often presented as canonical examples of self-similarity in the literature on fractals⁴. Yet aspects of self-similarity were characterized by botanists⁵ even before the term itself was coined. One of the best such characterizations belongs to Arber herself, who wrote the following in Chapter IX ("Repetitive branching and the *Gestalt* type, with special reference to parallelism", p. 142) of her book¹:

The relation to one another of a compound leaf, a simple leaf, and a mere lobe or hair, may perhaps be described as *identity-in-parallel*. A

leaflet of a compound leaf comes in, as it were, on both sides of the equation: to the compound leaf, the leaflet stands in the relation of part to whole, but it is also the equivalent to the compound leaf as the whole, though in another generation.

This quotation is interesting for several reasons. First, Arber’s *identity-in-parallel* clearly anticipated the notion of self-similarity in a botanical context. Second, Arber referred figuratively to an equation, in which a form, “though in another generation”, would appear on both sides. Third, Arber did not imply that this equation must necessarily have a geometric character. This leads us to the key question considered in the present paper:

How can the equation anticipated by Arber be formulated in mathematical terms?

We first examine iterated function systems as one possible interpretation of Arber’s identity-in-parallel equation, and point to the botanical inadequacy of this interpretation (Section 2). We then show that Arber’s identity-in-parallel can also be formalized in a different way, at the level of plant topology rather than geometry. This *topological self-similarity* can be expressed using *recurrence systems*^{6,7} (Section 3) and their variant, *catenative formulas*⁸ (Section 4), both of which have been defined within the theory of *L-systems*^{9,10}. A method for conceptualizing and visualizing recurrence systems and catenative formulas makes use of data flow networks (Section 5). Recurrence systems may describe both linear and branching structures, which makes them well suited to characterize self-similarities in plant architecture (Section 6). Furthermore, with an appropriate geometric interpretation, recurrence systems may yield forms that are self-similar in both the topological and geometric sense (Section 7).

2 Identity-in-parallel and iterated function systems

One obvious candidate for the identity-in-parallel equation is the global characterization of fractals^{11,12,13}, defined by the equation:

$$A = \bigcup_{i=1}^m T_i(A). \tag{1}$$

Here, the self-similar form A is the *attractor* of the set $\{T_1, T_2, \dots, T_m\}$ of contracting transformations (usually similarities or affine transformations). This set is referred to as an *iterated function system* (IFS). Consistent with Arber’s description, the attractor A appears on both sides of Equation 1. Furthermore, assuming that the initial structure $A^{(0)}$ is given, this equation can easily be extended to a sequence of “generations”:

$$A^{(n)} = \bigcup_{i=1}^m T_i(A^{(n-1)}); \quad n = 1, 2, 3, \dots \tag{2}$$

The above definitions of an IFS and its attractor can be extended to cases where different parts of a form are mapped into each other, instead of the whole form being

mapped into its own parts. These cases are captured by the closely related notions of recurrent IFS¹⁴, controlled IFS¹⁵, and language-restricted IFS^{16,17}. In all cases, the different parts A_1, A_2, \dots, A_z of the attractor satisfy the set of equations:

$$A_j = \bigcup_{i=1}^{m_j} T_{j_i}(A_{j_i}); \quad j = 1, 2, \dots, z \quad (3)$$

where z is the total number of parts, and m_j is the number of transformations T_{j_i} that map properly re-indexed parts A_1, A_2, \dots, A_z into part A_j . Unfortunately, even with these extensions, iterated function systems do not adequately characterize the self-similarity of plants. On one hand, IFS are insufficiently constrained: a small change in parameter values can change an attractor representing a branching structure into a set of unconnected points or segments, thus violating fundamental properties of the structures being modeled. On the other hand, IFS are too constrained: they impose a strict geometric correspondence between the form and its parts. Such correspondence is not frequently found in real plants, which is why plant-like structures generated using IFS are confined to the small set of examples that appear repetitively in the literature.

3 L-systems, recurrence systems, and self-similarity

In this section, we consider identity-in-parallel and self-similarity from a different perspective, focused on topology (the arrangement of components in a structure) rather than geometry. This approach is rooted in the theory of L-systems^{9,10,15}. After background definitions, we first review a theorem linking DOL-systems (deterministic context-free L-systems) to recurrence relations between the generated strings of symbols^{6,7}. We call these relations *recurrence systems* in a slight modification of the original definition⁶ of this term. We then show that the recurrence systems closely correspond to Arber's notion of identity-in-parallel, and thus constitute a description of self-similarity.

Definition 1 (from¹⁵). Let V denote a set of symbols called an alphabet, V^* the set of all words (strings of symbols) over V , and V^+ the set of all nonempty words over V . A *DOL-system* is an ordered triplet $\mathcal{G} = \langle V, \omega, P \rangle$ where V is the *alphabet* of the system, $\omega \in V^+$ is a nonempty word called the *axiom* and $P : V \rightarrow V^*$ is a finite *set of productions*. A production $(a, \chi) \in P$ is written as $a \rightarrow \chi$. The letter a and the word χ are called the *predecessor* and the *successor* of this production, respectively.

Definition 2 (from¹⁵). Let $\mu = a_1 \dots a_m$ be an arbitrary word over V . The word $\nu = \chi_1 \dots \chi_m \in V^*$ is *directly derived* from (or *generated* by) μ , noted $\mu \Rightarrow \nu$, if and only if $a_i \rightarrow \chi_i$ for all $i = 1, \dots, m$. A word ν is generated by \mathcal{G} in a derivation of *length* n if there exists a *developmental sequence* of words $\mu_0, \mu_1, \dots, \mu_n$ such that $\mu_0 = \omega$, $\mu_n = \nu$ and $\mu_0 \Rightarrow \mu_1 \Rightarrow \dots \Rightarrow \mu_n$.

Theorem 1 (from⁶). Consider a DOL-system $\mathcal{G} = \langle V, \omega, P \rangle$, and for each $a \in V$

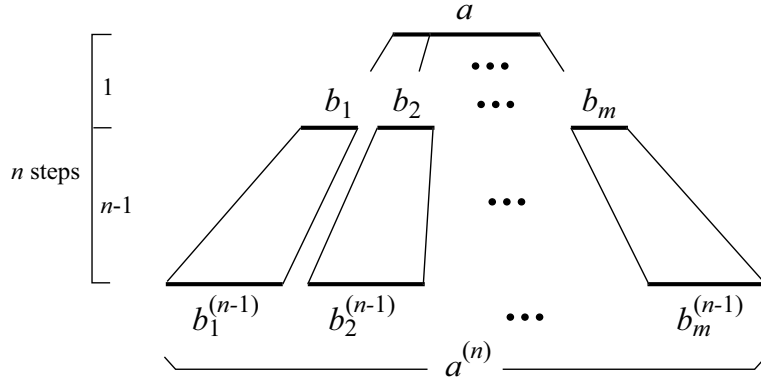


Figure 1: Illustration of the proof of Theorem 1

and $n \geq 0$ denote by $a^{(n)}$ the word derived from a in a derivation of length n :

$$a \xrightarrow{n} a^{(n)}. \quad (4)$$

If $a \rightarrow b_1 b_2 \dots b_m$ is a production in \mathcal{G} , then for any $n \geq 1$ the word $a^{(n)}$ satisfies the recurrence formula:

$$a^{(n)} = b_1^{(n-1)} b_2^{(n-1)} \dots b_m^{(n-1)}. \quad (5)$$

Proof. We decompose the derivation $a \xrightarrow{n} a^{(n)}$ into the first step and the remaining $n - 1$ steps (Figure 1):

$$a \xrightarrow{1} b_1 b_2 \dots b_m \xrightarrow{n-1} b_1^{(n-1)} b_2^{(n-1)} \dots b_m^{(n-1)}. \quad (6)$$

Thus, $a^{(n)} = b_1^{(n-1)} b_2^{(n-1)} \dots b_m^{(n-1)}$ \square .

Definition 3. Given a DOL-system $\mathcal{G} = \langle V, \omega, P \rangle$, we call the set of the recurrence formulas given by Equation 5, along with the initial conditions $a^{(0)} = a$ for all $a \in V$, the *recurrence system* associated with \mathcal{G} .

Note. Recurrence systems can also be specified independently of L-systems. For a formal definition and equivalence results see^{6,7}.

Example 1. Consider L-system $\mathcal{G}_1 = \langle \{a, b\}, a, P \rangle$ with productions

$$a \rightarrow ab, \quad b \rightarrow a. \quad (7)$$

This L-system can be viewed as a model of the development of a filamentous organism, with symbols a and b representing individual cells. The first production states that, over a certain time interval, a cell of type a divides into adjacent cells a and b . The second production states that, over the same time interval, a cell b changes its state into a . The above model is related to the development of the

step	L-system	recurrence system
	$\omega = a$ $a \rightarrow ab \quad b \rightarrow a$	$a^{(0)} = a$ $a^{(n)} = a^{(n-1)}b^{(n-1)}$
0		
1		
2		
3		

↑ same sequence ↑

Figure 2: A comparison of two methods for computing sequences of words generated by L-system \mathcal{G}_1 : by direct application of L-system productions (7) (left) and using recurrence relations (8) (right).

filamentous bacterium *Anabaena*, which is characterized by the unequal divisions of cells^{15,18}. To keep the example simple, we ignore here the polarity of cells, which would determine whether a cell a divides into ab or ba .

According to Theorem 1, the recurrence system associated with \mathcal{G}_1 is:

$$\begin{aligned} a^{(0)} &= a & b^{(0)} &= b \\ a^{(n)} &= a^{(n-1)}b^{(n-1)} & b^{(n)} &= a^{(n-1)} \end{aligned} \quad (8)$$

where $n = 1, 2, 3, \dots$. These equations provide an alternative to the usual method for generating words in an L-system, as illustrated in Figure 2. The alternative method closely corresponds to Arber's characterization of identity-in-parallel. We can make this evident by paraphrasing Arber's words quoted in the introduction to express the sample relation $a^{(n)} = a^{(n-1)}b^{(n-1)}$:

A substring $a^{(n-1)}$ of the whole string $a^{(n)}$ comes in on both sides of the equation: to the whole string, the substring stands in the relation of part to whole, but it is also the equivalent to the whole string, though in another generation.

Equating identity-in-parallel with self-similarity leads to the first, most straightforward interpretation of recurrence relations as a formal description of topological self-similarity. According to this interpretation, the words on which a recurrence system operates represent consecutive developmental stages (generations n) of a growing structure or a set of related structures (such as the structures derived from

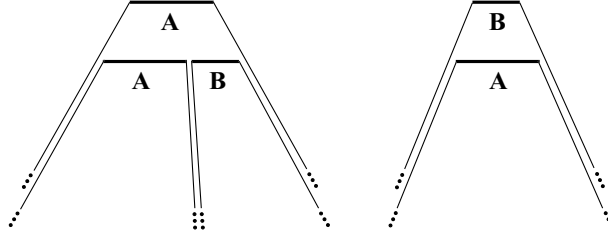


Figure 3: Self-similarity of the developmental sequences generated by L-system \mathcal{G}_1 from Example 1. Sequence **A** has itself and sequence **B** as its parts. Sequence **B** has sequence **A** as its part. This is a summary representation of the relations shown in more detail in Figure 2 (right).

a and b in the *Anabaena* example). The recurrence system specifies how the younger stages (with a lower index n) can be combined to produce the older stages. The pattern of the recurrence relations is independent of the age (generation step) n : for all n greater than some minimum value, different developmental stages of the same sequence are related to each other in the same manner.

The second interpretation of recurrence systems as a formalization of topological self-similarity deals with entire infinite developmental sequences, rather than individual words. To see this, let $\{\mu_0, \mu_1, \mu_2, \dots\}$ and $\{\nu_0, \nu_1, \nu_2, \dots\}$ denote two developmental sequences over some alphabet V , and \mathcal{N} and \circ denote operations on these sequences defined as follows:

$$\mathcal{N}(\xi, \{\mu_0, \mu_1, \mu_2, \dots\}) = \{\xi, \mu_0, \mu_1, \mu_2, \dots\}, \quad (9)$$

$$\{\mu_0, \mu_1, \mu_2, \dots\} \circ \{\nu_0, \nu_1, \nu_2, \dots\} = \{\mu_0\nu_0, \mu_1\nu_1, \mu_2\nu_2, \dots\}. \quad (10)$$

Thus, the \mathcal{N} operator represents a unit delay of a developmental sequence, with the initial element replaced by a given word ξ . The \circ operator represents concatenation of developmental sequences, defined as the sequence resulting from pairwise concatenation of corresponding strings in the argument sequences. By applying these definitions to the sequences

$$\mathbf{A} = \{a^{(0)}, a^{(1)}, a^{(2)}, \dots\}, \quad \mathbf{B} = \{b^{(0)}, b^{(1)}, b^{(2)}, \dots\}, \quad (11)$$

defined by the recurrence system (8), we obtain:

$$\mathbf{A} = \mathcal{N}(a, \mathbf{A} \circ \mathbf{B}), \quad \mathbf{B} = \mathcal{N}(b, \mathbf{A}). \quad (12)$$

Thus, the developmental sequences **A** and **B** include themselves as subsequences, as illustrated in Figure 3. This self-similarity of developmental sequences corresponds to the self-similarity of “cascades” in Mandelbrot’s definition quoted in the introduction.

The third interpretation of recurrence systems exposes self-similarity within individual words. To see this, let us formally interpret productions of an L-system \mathcal{G} as local zooming operations. A derivation step $\mu_i \Rightarrow \mu_{i+1}$ then represents the effect of globally zooming into the sequence μ_i or, conversely, zooming out of the sequence μ_{i+1} . Returning to Example 1 and Figure 2, a pair of words $\{a^{(n)}, b^{(n)}\}$

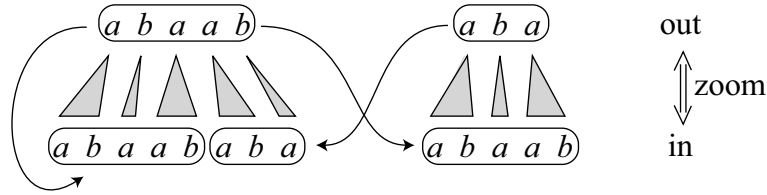


Figure 4: Self-similarity of the words generated by L-system \mathcal{G}_1 from Example 1. Productions and their inverses are formally treated as scaling operations.

with $n \geq 1$ is then equal to a combination of “scaled” versions of these words themselves, and in this sense these words are self-similar (Figure 4).

Independent of the interpretation used, Theorem 1 shows that recurrence relations are a mathematical consequence of any developmental process that can be modeled with a DOL-system. This is important from the biological perspective, because it explains why repetitive developmental processes in plants lead to topological self-similarity in developmental sequences and the resulting structures.

4 Catenative formulas

An important variant of recurrence relations are *catenative formulas*^{7,8}, which have the form

$$a^{(n)} = a^{(n-d_1)} a^{(n-d_2)} \dots a^{(n-d_m)} . \quad (13)$$

Here n is greater than some minimum value, and delays d_1, d_2, \dots, d_m are fixed positive integer numbers (i.e., they do not depend on n). Catenative formulas express older words as combinations of younger words from the same developmental sequence^a. A detailed account of the application of this concept to the description of filamentous organism (algae) is presented in¹⁹.

Referring once again to L-system \mathcal{G}_1 from Example 1 and the related recurrence system (8), we observe that $b^{(n-1)} = a^{(n-2)}$ for all $n \geq 2$. Using this substitution, we can express words $a^{(n)}$ as combinations of younger (lower n) words from the same sequence as follows:

$$a^{(-1)} = b , \quad a^{(0)} = a , \quad a^{(n)} = a^{(n-1)} a^{(n-2)} . \quad (14)$$

Thus, the developmental sequence generated by L-system \mathcal{G}_1 satisfies the catenative formula (14).

Catenative formulas are important as a topological counterpart of iterated function systems; in contrast, recurrence systems are a counterpart of recurrent iterated function systems. In general, we may also consider a combination of both notions: recurrence systems that operate simultaneously on several sequences of words, and involve different delays⁶.

^aThe term commonly used in literature is *locally* catenative formula. It reflects the property that “to get a new word by catenation of some previous words it is enough to remember previous words at most p steps back in the sequence”⁷. However, while in this sense catenative formulas are local in time, there are not local in space. For this reason, we have dropped the reference to locality from their name.

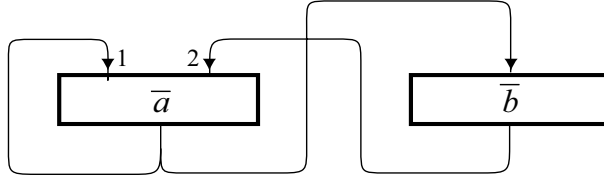


Figure 5: A data-flow representation of the recurrence relations given by Equation 8 and illustrated in Figure 2

5 Data-flow network representation of recurrence systems

Iterated function systems are sometimes conceptualized as *Multiple Reduction Copy Machines* (MRCM)⁴, which repetitively combine reduced copies of an original figure to produce a sequence of approximations of the attractor according to Equation 2. Similarly, recurrent IFS are conceptualized using *networked MRCM*⁴, which operate on a set of figures according to Equation 3. In the domain of words, a related device, called a *catenation machine*, was introduced by Mavaddat^{20,21} as a formal tool for hardware design. Below we apply catenation machines to represent recurrence systems in an intuitive, diagrammatic manner.

Referring to Figure 2, let us observe that the recurrence relations, which specify how the previously obtained words are combined into a new word, are the same at each level of a particular developmental sequence: independent of the derivation length $n \geq 1$. This is reflected in the repetitive pattern of lines showing “which word goes where” on the right side of Figure 2.

Instead of drawing the repetitive pattern, we can visualize recurrence relations between the old and new words using a *data-flow network*. Such a network is a directed multigraph, with the nodes labeled by symbols of the recurrence system alphabet (distinguished by a bar from the actual symbols) and arrows determined by the recurrence relations. Specifically, if $a^{(n)} = b_1^{(n-1)}b_2^{(n-1)} \dots b_m^{(n-1)}$ is a recurrence relation then the nodes $\bar{b}_1, \bar{b}_2, \dots, \bar{b}_m$ are connected to node \bar{a} by arrows that point to \bar{a} . These arrows are ordered in the same way as the symbols b_i in the recurrence relation. For instance, Figure 5 shows the network representing recurrence system (8) from Example 1. In general, the same letter b_i may occur in a production successor several times, and thus there may be several arrows from node \bar{b}_i to node \bar{a} , which is why the network is potentially a multigraph.

The nodes represent processing and storage elements of the network, and are capable of holding arbitrarily long words. Each node is initialized with its corresponding symbol (e.g., node \bar{b}_i initially holds symbol b_i). The network operates in synchronous steps, in which each node concatenates the words received from its input nodes. This operation give the network its name coined by Mavaddat: the catenation machine. The network generates sequences of words according to the recurrence relations associated with a given L-system. After n steps, each node \bar{b}_i will hold the word $b_i^{(n)}$.

A node with a single input can be interpreted as a delay operation, which blurs the distinction between data-flow networks representing recurrence systems

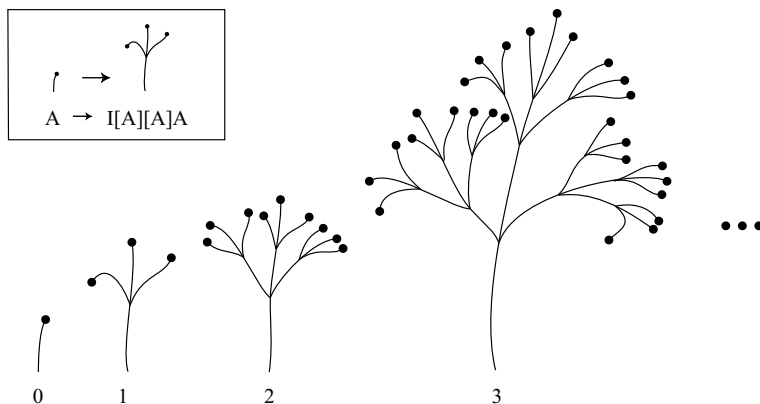


Figure 6: A sample developmental sequence with the topology described by L-system \mathcal{G}_2

and catenative formulas. For example, the network shown in Figure 5 represents not only recurrence system (8), but also catenative formula (14).

Unlike recurrence relations, which can be viewed as a shorthand notation for an infinite sequence of equations corresponding to $n = 1, 2, 3, \dots$, the data-flow network representations are finite, with no explicit reference to index n . This reflects an essential feature of self-similarity: the repetitive character of relations between components of a structure, which in our case are represented by words. The data-flow networks capture these relations in a succinct and intuitive way, and therefore provide a convenient graphical characterization of topological self-similarities in developing structures.

6 Extension to branching structures

The relationship between development and recurrence systems examined in Section 3 holds not only for linear structures, but also for branching structures, which are paramount in the kingdom of plants. The extension of L-systems to branching structures makes use of the *bracketed string notation* introduced by Lindenmayer⁹.

Example 2. Let us consider a DOL-system $\mathcal{G}_2 = \langle V, \omega, P \rangle$, in which the alphabet V consists of four symbols: letter A denoting the apex of a branching structure, letter I denoting a branch segment, and a pair of brackets $[,]$ delimiting branches. The axiom is a single apex A , and the production set P has a single non-identity production:

$$A \rightarrow I[A][A]A. \quad (15)$$

(We do not explicitly specify the identity productions for the remaining symbols, such as $I \rightarrow I$.) According to production (15), in a given time interval the apex A creates a branching structure consisting of a segment, two lateral apices, and a terminal apex.

The above L-system generates the developmental sequence shown in Figure 6. It can be interpreted as a schematic depiction of the development of a carrot-like leaf,

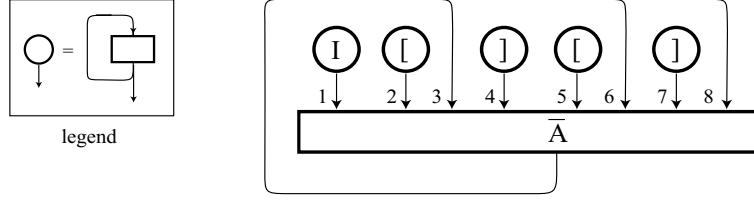


Figure 7: A data-flow representation of the recurrence relations given by Equation 16

for example. The apices are represented as circles and the segments are represented as lines. Their lengths and shapes have been chosen arbitrarily, since L-system \mathcal{G}_2 only describes the branching topology of the generated structures.

In order to formally characterize the self-similar aspects of this developmental sequence, we apply Theorem 1 to construct the equivalent recurrence system:

$$A^{(0)} = A, \quad A^{(n)} = I[A^{(n-1)}][A^{(n-1)}]A^{(n-1)}. \quad (16)$$

This system is represented by the data-flow diagram in Figure 7. The structure $A^{(n)}$, which represents the n -th stage of the development beginning with a single apex A , can be viewed as a branching configuration of a segment I and three copies of the younger structure $A^{(n-1)}$. Thus, the structure $A^{(n)}$ and the developmental sequence that generates it are topologically self-similar in all three senses of this word described in Section 3. At the same time, the structures and the developmental sequence shown in Figure 6 are not geometrically self-similar. This illustrates our thesis that topological self-similarity captures a wider class of biologically relevant phenomena, compared to geometric self-similarity.

7 Relation between topological and geometric self-similarity

In some cases, topologically-self-similar structures and developmental sequences can be assigned a geometric interpretation that makes them geometrically self-similar as well. This possibility is interesting from both the mathematical and the biological perspective, as it highlights the conceptual relation between topological and geometric self-similarities. The following discussion is based on a geometric extension of the L-system from Example 2.

Example 3. Let us consider a DOL-system $\mathcal{G}_3 = \langle V, \omega, P \rangle$, in which the alphabet V consists of the four symbols $A, I, [,]$ introduced in Example 2, and additional symbols $+$ and $-$ that indicate the direction of branching (to the left and to the right, respectively). The axiom is a single apex A , and the production set P has two non-identity productions:

$$A \rightarrow I[+A][-A]IA, \quad I \rightarrow II. \quad (17)$$

Compared to production (15), the first production above inserts an additional segment I between the branching point and the terminal apex A . The second production replaces each segment I with the pair II in every derivation step. These changes

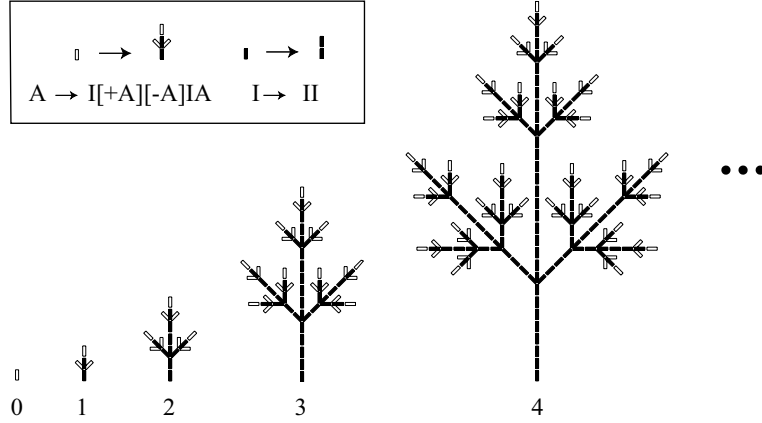


Figure 8: Developmental sequence generated by L-system \mathcal{G}_3 with the turtle interpretation

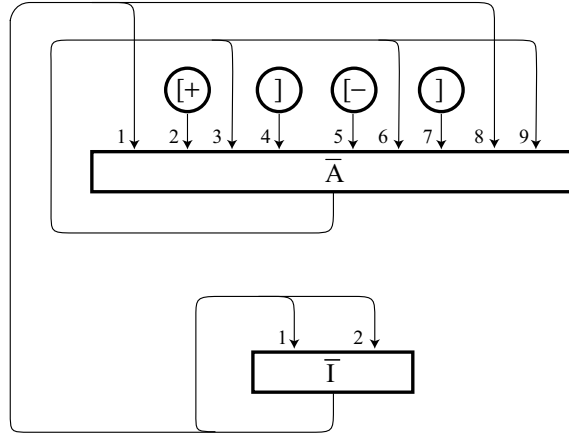


Figure 9: A data-flow representation of the recurrence relations given by Equation 18

become relevant when symbols are assigned a geometric interpretation. In our example, we use the turtle interpretation of L-system strings^{15,22,23}. Specifically, we assume that all apices A and segments I are represented as lines of equal length, and all branching angles have the same magnitude of 45° . Under these assumptions, the L-system \mathcal{G}_3 generates the developmental sequence shown in Figure 8.

In order to formally characterize the self-similar aspects of this developmental sequence, we apply Theorem 1 to construct the equivalent recurrence system:

$$\begin{aligned} A^{(0)} &= A & A^{(n)} &= I^{(n-1)}[+A^{(n-1)}][-A^{(n-1)}]I^{(n-1)}A^{(n-1)} \\ I^{(0)} &= I & I^{(n)} &= I^{(n-1)}I^{(n-1)} \end{aligned} \quad (18)$$

The data-flow representation of this systems is shown in Figure 9. We extract geometric information from the recurrence system (18) using the following theorem.

Theorem 2 (from ¹⁵). Consider turtle interpretation $\mathcal{J} : V^* \rightarrow S$ as a mapping from the set V^* of words over an alphabet V into the set S of geometric figures (sets of points in the plane or in 3D space). Furthermore, let $\mathcal{T}(\mu)$ represent the change in the turtle state (position and orientation) resulting from the interpretation of word $\mu \in V^*$. Then for any decomposition $\mu_1\mu_2$ of the word μ such that μ_1, μ_2 do not contain unbalanced right brackets, the following holds:

$$\mathcal{J}(\mu_1\mu_2) = \mathcal{J}(\mu_1) \cup \mathcal{J}(\mu_2)\mathcal{T}(\mu_1) . \quad (19)$$

According to this theorem, the turtle interpretation of the word $\mu = \mu_1\mu_2$ is a form (set of points in two or three dimensions) equal to the union of:

1. the turtle interpretation of the word μ_1 , and
2. the turtle interpretation of the word μ_2 , repositioned by the transformation $\mathcal{T}(\mu_1)$ that results from the interpretation of the word μ_1 .

By applying Theorem 2 to the recurrence relation given by Equation 18, we thus obtain:

$$\begin{aligned} \mathcal{J}(A^{(n)}) &= \mathcal{J}(I^{(n-1)}) \cup \\ &\quad \mathcal{J}(A^{(n-1)}) \mathcal{T}(I^{(n-1)+}) \cup \\ &\quad \mathcal{J}(A^{(n-1)}) \mathcal{T}(I^{(n-1)-}) \cup \\ &\quad \mathcal{J}(I^{(n-1)}) \mathcal{T}(I^{(n-1)}) \cup \\ &\quad \mathcal{J}(A^{(n-1)}) \mathcal{T}(I^{(n-1)}I^{(n-1)}) , \\ \mathcal{J}(I^{(n)}) &= \mathcal{J}(I^{(n-1)}) \cup \mathcal{J}(I^{(n-1)}) \mathcal{T}(I^{(n-1)}) . \end{aligned} \quad (20)$$

A geometric interpretation of these equations is shown in Figure 10. The generated developmental sequence exhibits an aspect of geometric self-similarity: a pair of structures $\{A^{(n)}, I^{(n)}\}$ can be obtained by combining younger developmental stages $\{A^{(n-1)}, I^{(n-1)}\}$ of the same structures. This combination, however, involves geometric transformations $\mathcal{T}(I^{(n-1)})$ that change from one step n to another. In contrast, the definition of geometric self-similarity embedded in the notion of iterated function systems (Equations 1–3) postulates that transformations T_i be fixed and not depend on the iteration number n .

In the example under consideration, we can achieve this independence by changing the interpretation of symbols A and I from one derivation step to another. To this end, let us observe that all transformations \mathcal{T} in recurrence system (20) include the same term $I^{(n-1)}$ (equal to $I^{(3)}$ in Figure 10). According to the recurrence formulas $I^{(0)} = I$ and $I^{(n)} = I^{(n-1)}I^{(n-1)}$ (Equation 18), the segment $I^{(n)}$ contains twice as many symbols I as the segment $I^{(n-1)}$. Thus, if the length of line segments represented by symbol I in step n is reduced by one half with respect to the lines represented by the same symbol I in step $n - 1$, the corresponding transformations $\mathcal{T}(I^{(n)})$ and $\mathcal{T}(I^{(n-1)})$ will be the same, independent of n . The forms $\mathcal{J}(A^{(n)})$ and $\mathcal{J}(I^{(n)})$, derived from A and I in n steps, can then be obtained by combining the forms $\mathcal{J}(A^{(n-1)})$ and $\mathcal{J}(I^{(n-1)})$ using transformations T_1, T_2, \dots, T_5 that are independent of n . Furthermore, since the length of segments represented by symbol I (and, consistently, symbol A) is decreased by one half between consecutive

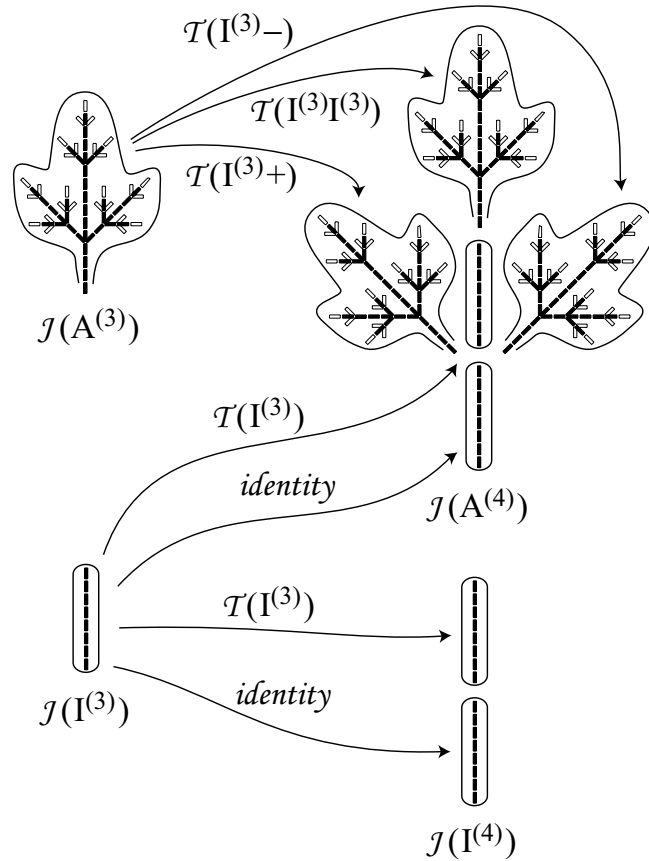


Figure 10: Geometric relations between components of the developmental sequence generated by L-system \mathcal{G}_3 , as revealed by Equation 20

derivation steps, the forms $\mathcal{J}(A^{(n-1)})$ and $\mathcal{J}(I^{(n-1)})$ must also be scaled by one half before they are combined into $\mathcal{J}(A^{(n)})$ and $\mathcal{J}(I^{(n)})$ (Figure 11a).

The resulting geometric self-similarity can be characterized by a data flow diagram, in which the nodes assemble new figures by computing the set-theoretic union of figures received at the inputs, and edges represent transformations. In order to distinguish these diagrams from catenation machines, we now represent the nodes as circles (Figure 11b). The order of inputs is irrelevant, because the union of sets is a commutative operation.

The graph shown in Figure 11b represents the form of self-similarity found in recurrent iterated function systems and their variants (Section 2). A comparison of this graph with the catenation machine in Figure 9 points to the close relationship between both notions. Nevertheless, it is evident from the number of assumptions introduced in the above example that topological self-similarity yields geometric self-similarity only in special situations.

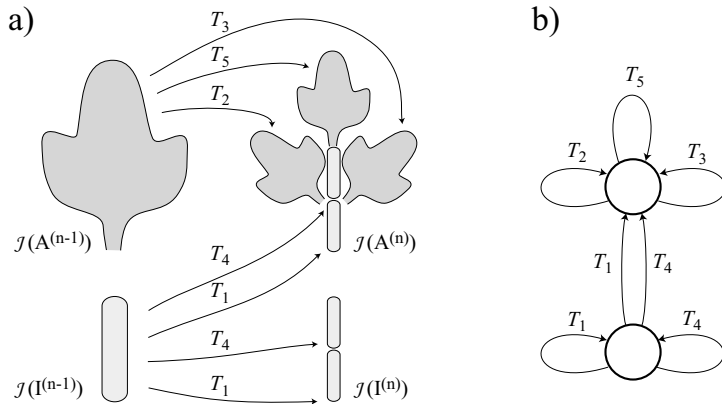


Figure 11: a) Geometric relations between components of the developmental sequence similar to that shown in Figure 10, but including an additional scaling of the components. Transformations T_1 to T_5 do not depend on the developmental step n . b) Control graph of the recurrent IFS associated with Figure a.

8 Conclusions

We revisited the notion of self-similarity in the context of plant modeling. We observed that the usual geometric self-similarity does not adequately capture self-similarity in plants, and we investigated the notion of topological self-similarity as an alternative. To this end, we revisited the notion of recurrence systems introduced in the theory of L-systems, and we concluded that they can be viewed as a formal characterization of the topological self-similarity in linear and branching structures.

Several questions are open for further research. We only considered “ordinary” (non-parametric) L-systems, and it would be interesting to extend our discussion to parametric L-systems^{15,24} as well. It would also be interesting to investigate the general conditions under which topological self-similarity yields geometric self-similarity. More specific questions concern the relationship between delays in cate-native formulas (Equation 13) and scaling transformations in the corresponding recurrent IFS. In order to firmly establish recurrence systems in the domain of fractals, it would be worthwhile to formally extend previous characterizations of the relations between L-systems, Koch constructions, and iterated function systems^{16,17,25,26,27} to recurrence systems. The postulated structure of relationships between these formalisms is shown in Figure 12.

The last class of problems deals with applications of recurrence relations and topological self-similarity to the analysis and synthesis of real biological structures. As is well known, geometric self-similarity makes it possible to represent intricate forms using a minimum amount of data. The question is, to what extent topological self-similarity could be used in an analogous fashion, reducing the amount of data needed to describe and generate complex branching structures. In addition, it would be interesting to investigate whether the relationship between development and self-similarity could be used to infer models of plant development on the basis of self-similarities observed in mature plants.

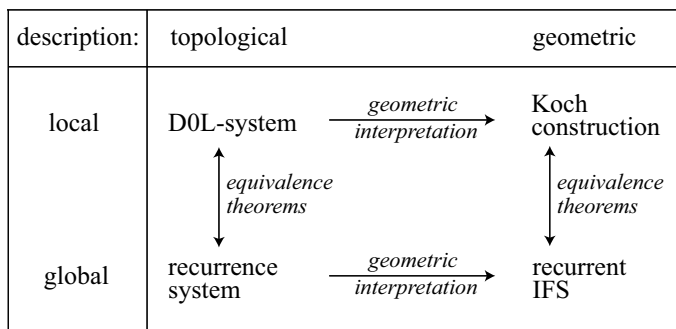


Figure 12: Postulated relationships between L-systems, Koch constructions, recurrence systems, and recurrent iterated function systems

Acknowledgments

I would like to thank Lynn Mercer and Brendan Lane for insightful comments and editorial help. This work was supported in part by a Discovery Grant from the Natural Sciences and Engineering Research Council of Canada.

References

1. A. Arber. *Natural philosophy of plant form*. University Press, Cambridge, 1950.
2. R. Schmid. Agnes Arber, neé Robertson (1879–1960): Fragments of her life, including her place in biology and women’s studies. *Annals of Botany*, 88:1105–1128, 2001.
3. B. B. Mandelbrot. *The fractal geometry of nature*. W. H. Freeman, San Francisco, 1982.
4. H.-O. Peitgen, H. Jürgens, and D. Saupe, editors. *Chaos and fractals. New frontiers of science*. Springer-Verlag, New York, 1992.
5. W. Troll. *Die Infloreszenzen*, volume I. Gustav Fischer Verlag, Stuttgart, 1964.
6. G. T. Herman, A. Lindenmayer, and G. Rozenberg. Description of developmental languages using recurrence systems. *Mathematical Systems Theory*, 8(4):316–341, 1975.
7. G. T. Herman and G. Rozenberg. *Developmental systems and languages*. North-Holland, Amsterdam, 1975.
8. G. Rozenberg and A. Lindenmayer. Developmental systems with locally catenative formulas. *Acta Informatica*, 2:214–248, 1973.
9. A. Lindenmayer. Mathematical models for cellular interaction in development, Parts I and II. *Journal of Theoretical Biology*, 18:280–315, 1968.
10. A. Lindenmayer. Developmental systems without cellular interaction, their languages and grammars. *Journal of Theoretical Biology*, 30:455–484, 1971.
11. J. E. Hutchinson. Fractals and self-similarity. *Indiana University Journal of Mathematics*, 30(5):713–747, 1981.

12. M. F. Barnsley and S. Demko. Iterated function systems and the global construction of fractals. *Proceedings of the Royal Society of London Ser. A*, 399:243–275, 1985.
13. M. F. Barnsley. *Fractals everywhere*. Academic Press, San Diego, 1988.
14. M. F. Barnsley, J. H. Elton, and D. P. Hardin. Recurrent iterated function systems. *Constructive Approximation*, 5:3–31, 1989.
15. P. Prusinkiewicz and A. Lindenmayer. *The algorithmic beauty of plants*. Springer-Verlag, New York, 1990. With J. S. Hanan, F. D. Fracchia, D. R. Fowler, M. J. M. de Boer, and L. Mercer.
16. P. Prusinkiewicz and M. Hammel. Automata, languages, and iterated function systems. In J. C. Hart and F. K. Musgrave, editors, *Fractal Modeling in 3D Computer Graphics and Imagery*, pages 115–143. ACM SIGGRAPH, 1991. Course Notes C14.
17. P. Prusinkiewicz and M. Hammel. Language-restricted iterated function systems, Koch constructions, and L-systems. In J. C. Hart, editor, *New directions for fractal modeling in computer graphics*, pages 4.1–4.14. ACM SIGGRAPH, 1994. Course Notes 13.
18. G. J. Mitchison and M. Wilcox. Rules governing cell division in *Anabaena*. *Nature*, 239:110–111, 1972.
19. H. B. Lück and J. Lück. Cell number and cell size in filamentous organisms in relation to ancestrally and positionally dependent generation times. In A. Lindenmayer and G. Rozenberg, editors, *Automata, languages, development*, pages 109–124. North-Holland, Amsterdam, 1976.
20. F. Mavaddat. Catenation machines and their relation to DTOL systems. Technical Report CS-91-26, Department of Computer Science, University of Waterloo, Waterloo, Canada, August 1991.
21. F. Mavaddat. Data-path synthesis as grammar inference. In *Proceedings of IFIP-INPG Workshop on Control Dominated Synthesis*, page 10 pp., Grenoble, France, September 1992.
22. A. L. Szilard and R. E. Quinton. An interpretation for DOL systems by computer graphics. *The Science Terrapin*, 4:8–13, 1979.
23. P. Prusinkiewicz. Graphical applications of L-systems. In *Proceedings of Graphics Interface '86 — Vision Interface '86*, pages 247–253, 1986.
24. J. S. Hanan. *Parametric L-systems and their application to the modelling and visualization of plants*. PhD thesis, University of Regina, June 1992.
25. P. Prusinkiewicz and G. Sandness. Koch curves as attractors and repellers. *IEEE Computer Graphics and Applications*, 8(6):26–40, November 1988.
26. J.C. Hart. The object instancing paradigm for linear fractal modeling. In *Graphics Interface '92*, pages 224–231. CIPS, 1992.
27. T. Ju, S. Shaefer, and R. Goldman. Recursive turtle programs and iterated affine transformations. Manuscript, Department of Computer Science, Rice University, Houston, TX, October 2003.