Fractals, Vol. 18, No. 2 (2010) 145–155

DOI: 10.1142/S0218348X10004804

World Scientific
www.worldscientific.com

**Articles**

# HIGHER BLOCK IFS 1: MEMORY REDUCTION AND DIMENSION COMPUTATIONS

RICHARD BEDIENT,* MICHAEL FRAME,‡,§ KEITH GROSS,*
JENNIFER LANSKI and BRENDAN SULLIVAN†
*Mathematics Department, Hamilton College
Clinton, NY 13323, USA
†Mathematics Department, Carnegie Mellon University
Pittsburgh, PA 15213, USA
‡Mathematics Department, Yale University
10 Hillhouse Ave., P. O. Box 208283
New Haven, CT 06520-8283, USA
§michael.frame@yale.edu

Received June 17, 2009
Accepted September 28, 2009

**Abstract**

By applying a result from the theory of subshifts of finite type,[1] we generalize the result of Frame and Lanski[2] to IFS with multistep memory. Specifically, we show that for an IFS $\mathcal{I}$ with $m$-step memory, there is an IFS with 1-step memory (though in general with many more transformations than $\mathcal{I}$) having the same attractor as $\mathcal{I}$.

*Keywords*: Iterated Function System; Subshift of Finite Type; Higher Block Shift.

## 1. INTRODUCTION: IFS AND MEMORY

Recall the standard formulation for an iterated function system (IFS).[3,4] Given contraction maps $T_1, \ldots, T_n : \mathbb{R}^2 \to \mathbb{R}^2$, define a function $\mathcal{T} : C(\mathbb{R}^2) \to C(\mathbb{R}^2)$, the compact subsets of $\mathbb{R}^2$, by $\mathcal{T}(B) = \cup_{i=1}^{n}\{T_i(x) : x \in B\}$. In the Hausdorff metric $h$ on $C(\mathbb{R}^2)$, $\mathcal{T}$ is a contraction map. Because

§Corresponding author. Partially supported by NSF DMS 0203203.

$C(\mathbb{R}^2)$ is complete in $h$, for any $B \in C(\mathbb{R}^2)$ the sequence $\mathcal{T}(B), \mathcal{T}^2(B), \mathcal{T}^3(B), \ldots$ converges to a unique $A \in C(\mathbb{R}^2)$, the *attractor* of the IFS $\mathcal{T}$. The set $A$ is characterized by $\mathcal{T}(A) = A$. In this formulation, note that the transformations $T_i$ are applied in all combinations: $\mathcal{T}(B) = \cup_{i=1}^n T_i(B)$, $\mathcal{T}^2(B) = \cup_{j=1}^n \cup_{i=1}^n T_j(T_i(B))$, and so on. Thus we could call this an *unrestricted* or *memoryless* IFS.

We say an IFS has memory if some compositions of the $T_i$ are forbidden. This notion has a fairly long history (for any topic in the field of fractals) and a substantial literature. Versions of this construction are called *graph-directed*, *recurrent*, *hierarchcal*, and *Markov* IFS. See Refs. 3, 5–17, for example. Distinctions between these types of IFS with memory can be found in Ref. 18.

An example is given in Sec. 2, but the meaning should be clear when we say an IFS has 1-step memory if certain pairs $T_i \circ T_j$ are forbidden. This information can be encoded in a *transition matrix* $M = [m_{ij}]$, where $m_{ij} = 0$ if $T_i \circ T_j$ is forbidden, and $m_{ij} = 1$ if $T_i \circ T_j$ is allowed. Note that if $T_{i_2} \circ T_{i_1}$ is forbidden, so also is every composition $T_{j_k} \circ \cdots \circ T_{j_1}$, where $i_2$ and $i_1$ are two consecutive indices in $j_k, \ldots, j_1$.

An IFS has *2-step memory* if (perhaps some pairs are forbidden and) some compositions $T_{i_3} \circ T_{i_2} \circ T_{i_1}$ are forbidden, where the triple $i_3 i_2 i_1$ does not contain a forbidden pair.

An IFS has *m-step memory* if it is determined by specifying forbidden combinations up through length $m+1$, and least one forbidden $(m+1)$-tuple does not contain a forbidden $j$-tuple for $1 \leq j \leq m$. Call an IFS with $m$-step memory an $m$-IFS. A standard IFS (without memory) is called a 0-IFS.

We represent the composition $T_{i_m} \circ \cdots \circ T_{i_1}$ by the $m$-string $i_m \cdots i_1$, called the *label* of the composition, and also by the sequence of transitions

$$i_1 \to \cdots \to i_m.$$

Note the presented order of the string agrees with that of the composition and of the transition sequence, taking note of the directions of the arrows in the sequence.

Early work on IFS with memory included computing the Hausdorff dimension of the attractor $A$.[17] Taking $r_j$ to be the contraction factor of the similarities $T_j$, the Hausdorff dimension of $A$ is the unique $d$ for which the spectral radius of $M(d) = [m_{ij} r_j^d]$ equals 1. Recall the spectral radius $\rho$ of a matrix $M$ is

$$\rho(M) = \max\{|\lambda_i| : \text{ where } \lambda_i \text{ is an eigenvalue of } M\}.$$

That is, the dimension is given by

$$\rho[m_{ij} r_j^d] = 1. \tag{1}$$

Because the matrix $M(d)$ is non-negative, the Perron-Frobenius theorem guarantees this maximum is achieved by a real eigenvalue. This formula generalizes the Moran equation $\sum_{i=1}^n r_i^d = 1$ for the dimension of IFS without memory. This was extended further to random constructions in Refs. 8 and 16. Replacing $m_{ij} r_j^d$ by $p_{ij}^q r_j^{\beta(q)}$, where $p_{ij}$ is the probability that the composition $T_i \circ T_j$ occurs, and $q$ is a parameter ranging over $\mathbb{R}$, the $f(\alpha)$ curve is obtained by the Legendre transform of $\beta(q)$ gotten by solving $\rho[p_{ij}^q r_i^{\beta(q)}] = 1$. See Refs. 10 and 12.

Applications of IFS with memory include compressing images,[19] developing variants of the chaos game approach to visualizing DNA sequences[20,21] to distinguish introns from exons[5] and to trace evolutionary relations of species,[22] analyzing nonlinear time series,[23] and defining Laplacians on fractals generated by IFS with memory.[11,24]

In Frame and Lanski[2] we investigated the circumstances under which the attractor of a 1-IFS could be realized as the attractor of a 0-IFS. The solution can be expressed neatly in the language of directed graphs. Associate each $T_i$ with a node of the graph, and place an edge from $i$ to $j$ if $T_j$ can follow $T_i$, that is, if the transition $i \to j$ is allowed. A node $i$ is called a *rome* if for every $j$ there is an edge $j \to i$. In Ref. 2, we called this a *full state*, being unaware of the sensible use of the word "rome" in this context. Doug Lind mentioned this language to us, but because this was in a conversation, we thought the word was "roam." Some confusing, though entertaining, Google searches resulted.

The main result of Ref. 2 is that a 1-IFS attractor can be realized as the attractor of a 0-IFS if and only if

(1) the 1-IFS graph has at least one rome, and
(2) there is a path to each non-rome from a rome.

Moreover, among those that can be realized as attractors of 0-IFS, this IFS requires infinitely may transformations if and only if the graph contains a cycle through non-rome nodes. Unknown to us at the time, our theorem answered a question posed in Layman and Womack.[13] An algorithm for finding a 0-IFS representation for a 1-IFS with a rome is presented in Máté.[15] All compositions of transformations containing a single rome in the terminal position suffice.

Our purpose here is to investigate what additional complications arise if we add more steps to the memory of the IFS. The existence of memory reduction is a straightforward adaptation of a result on subshifts of finite type. Other relations between IFS with different levels of memory can be more nuanced.

## 2. BASIC CONCEPTS AND SOME ILLUSTRATIVE EXAMPLES

We build most IFS from four transformations, $I = \{T_1, T_2, T_3, T_4\}$, where

$$\begin{aligned} T_1(x,y) &= \left(\frac{x}{2}, \frac{y}{2}\right) + (0,0), \\ T_2(x,y) &= \left(\frac{x}{2}, \frac{y}{2}\right) + \left(\frac{1}{2}, 0\right), \\ T_3(x,y) &= \left(\frac{x}{2}, \frac{y}{2}\right) + \left(0, \frac{1}{2}\right), \\ T_4(x,y) &= \left(\frac{x}{2}, \frac{y}{2}\right) + \left(\frac{1}{2}, \frac{1}{2}\right). \end{aligned} \tag{2}$$

As a 0-IFS, these transformations produce the filled-in unit square $S$.

Transformations (2) divide $S$ into subsquares with *addresses* determined by the appropriate compositions. For example, the $2^{-n} \times 2^{-n}$ subsquare

$$S_{i_n \cdots i_1} = T_{i_n} \circ \cdots \circ T_{i_1}(S)$$

has *address* $i_n \cdots i_1$. Note the order of indices of the address agrees with the order of the composition of transformations, and observe

$$S_{i_q} \supset S_{i_q i_{q-1}} \supset \cdots \supset S_{i_q \cdots i_1}. \tag{3}$$

Figure 1 illustrates Eq. (3). Note $S_2 \supset S_{23}$, for example.

Forbidding certain combinations of transformations gives rise to IFS with memory. For an IFS with tansformations $T_1, \ldots, T_m$, the *alphabet* $\mathcal{A}$ is $\{1, \ldots, m\}$. For most of our examples, $m = 4$. Suppose $\mathcal{F}$ is a finite collection of strings from this alphabet. Say the longest string in $\mathcal{F}$ has length $n+1$. The $n$-IFS determined by $\mathcal{F}$ forbids the compositions $T_{i_q} \circ \cdots \circ T_{i_1}$ where $i_q \cdots i_1 \in \mathcal{F}$. The set of all forbidden strings is the set of all strings on $\mathcal{A}$ that contain an element of $\mathcal{F}$ as a substring. We say $\mathcal{F}$ *generates* the collection of all forbidden strings.

The indexing of the $n$-IFS is meant to indicate that forbidden pairs are determined by 1-step memory, forbidden triples by 2-step memory, and so on.

If $A$ is the attractor of an IFS with memory based on transformations (2), then $A_{i_n \cdots i_1} = A \cap S_{i_n \cdots i_1}$ is the *address* $i_n \cdots i_1$ *region* of the attractor.

**Example 2.1.** 1-IFS with $\mathcal{F}_1 = \{14, 23, 32\}$

The regions $A_{14}, A_{23}$, and $A_{32}$ are empty, as is every region with address containing 14, 23, or 32. See the left side of Fig. 2. Note boxes indicating the length 3 address regions are included.

**Example 2.2.** 2-IFS with $\mathcal{F}_2 = \{14, 23, 32, 441\}$

The regions $A_{14}, A_{23}, A_{32}$, and $A_{441}$ are empty, as is every region with address containing 14, 23, 32, or 441. See the right side of Fig. 2. The most obvious difference between the left and right sides of Fig. 2 lies in address 441, but this implies other differences. Consider 344 and 244, for example.

Denote by $I(\mathcal{F})$ the attractor of the IFS $I$ with forbidden strings $\mathcal{F}$. We say $\mathcal{F}$ is a *generating set* for the attractor. Suppose the longest string in $\mathcal{F}$ has length $n$. Then there is a set $\mathcal{F}'$ with all strings having length $n$ and $I(\mathcal{F}) = I(\mathcal{F}')$. This is most easily seen through an example. Take $\mathcal{A} = \{1,2,3,4\}$ and

$$\mathcal{F} = \{11, 123\}. \quad \text{Then}$$

$$\mathcal{F}' = \{111, 112, 113, 114, 123\}.$$

In terms of addresses, every element $i_q \cdots i_1$ of $\mathcal{F}$ determines the region $A_{i_q \cdots i_1}$ of the attractor having address $i_q \cdots i_1$. If $q < n$, we replace $i_q \cdots i_1$ with

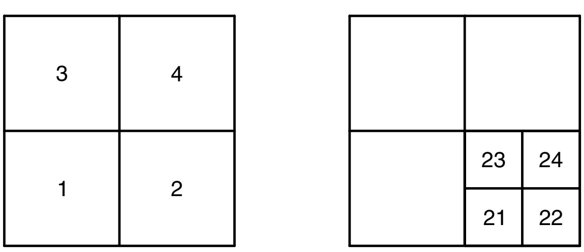


**Fig. 1** An illustration of addresses.

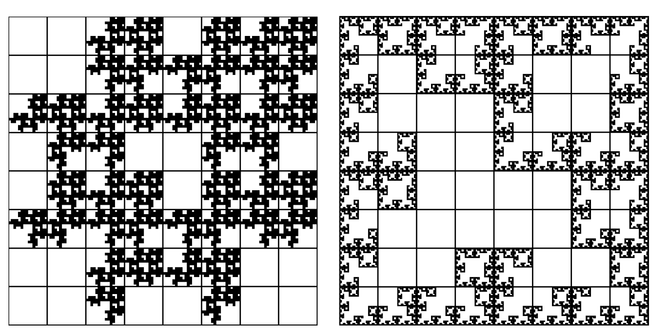

**Fig. 2** Attractors for Example 2.1 (**left**) and Example 2.2 (**right**).

the $4^{n-q}$ sequences $i_q \cdots i_1 j_{n-q} \cdots j_1$ where each $j_k$ takes on all four values 1, 2, 3, 4.

This observation suggests a generalization, relating forbidden addresses to longer addresses containing them. Because $\text{int}(A_{i_n \cdots i_1})$ can be empty even though $\text{int}(S_{i_n \cdots i_1}) \cap A \neq \emptyset$, we define the sense in which we call the region $A_{i_q \cdots i_1}$ empty.

To avoid some additional special cases of little interest, we assume

$$A \cap \text{int}(S) \neq \emptyset. \tag{4}$$

IFS for which (4) fails are left as exercises.

**Definition 2.1.** The region $A_{i_q \cdots i_1}$ is *empty* if $A \cap \text{int}(S_{i_q \cdots i_1}) = \emptyset$.

**Lemma 2.1.** *The region $A_{i_q \cdots i_1}$ is empty if and only if the address $i_q \cdots i_1$ is forbidden.*

**Proof.** If $i_q \cdots i_1$ is forbidden, then $T_{i_q} \circ \cdots \circ T_{i_1}$ cannot be applied. If there were some $x \in A \cap \text{int}(S_{i_q \cdots i_1})$, then

$$x = T_{i_q} \circ \cdots \circ T_{i_1}(y) \tag{5}$$

for some $y \in \text{int}(S)$. Because $x \in \text{int}(S_{i_q \cdots i_1})$, we have $x \neq T_{k_p} \circ \cdots \circ T_{k_1}(z)$ and so (5) is the only way $A$ could contain such an $x$. We see this is impossible because that composition is forbidden.

Now suppose $A \cap \text{int}(S_{i_q \cdots i_1}) = \emptyset$. If $T_{i_q} \circ \cdots \circ T_{i_1}$ were allowed, then the invariance of $A$ under the allowed compositions of the $T_i$ would show $A \supset T_{i_q} \circ \cdots \circ T_{i_1}(A)$. Then by condition (4),

$$T_{i_q} \circ \cdots \circ T_{i_1}(A) \cap T_{i_q} \circ \cdots \circ T_{i_1}(S) \neq \emptyset$$

contradicting $A \cap \text{int}(S_{i_q \cdots i_1}) = \emptyset$. □

**Lemma 2.2.** *If $i_q \cdots i_1$ is a forbidden address, so is $j_a \cdots j_1 i_q \cdots i_1 k_b \cdots k_1$ for all $a \geq 0$ and $b \geq 0$.*

**Proof.** If $i_q \cdots i_1$ is a forbidden address, then $A_{i_q \cdots i_1}$ is empty. By Eq. (3),

$$A_{i_q \cdots i_1} \supset A_{i_q \cdots i_1 k_b \cdots k_1}$$

so $A_{i_q \cdots i_1 k_b \cdots k_1} = \emptyset$ and $i_q \cdots i_1 k_b \cdots k_1$ is a forbidden address by Lemma 2.1.

Next,

$$A_{j_a \cdots j_1 i_q \cdots i_1 k_b \cdots k_1} = T_{j_a} \circ \cdots \circ T_{j_1}(A_{i_q \cdots i_1 k_b \cdots k_1})$$

If there were some $x \in A \cap \text{int}(S_{j_a \cdots j_1 i_q \cdots i_1 k_b \cdots k_1})$, then $x = T_{j_a} \circ \cdots \circ T_{j_1}(y)$ for some $y \in A \cap \text{int}(S_{i_q \cdots i_1 k_b \cdots k_1})$, but there is no such $y$. Applying Lemma 2.1 again, $j_a \cdots j_1 i_q \cdots i_1 k_b \cdots k_1$ is a forbidden address. □

By a *substring* of $i_q \cdots i_1$ we mean any $j_m \cdots j_1$ for which

$$i_q \cdots i_1 = p_a \cdots p_1 j_m \cdots j_1 q_b \cdots q_1$$

where we allow $a = 0$ or $b = 0$. A simple consequence of Lemmas 2.1 and 2.2 is the following

**Corollary 2.1.** *The region $A_{i_q \cdots i_1}$ is empty if and only if some substring of $i_q \cdots i_1$ belongs to some $\mathcal{F}$ determining this IFS.*

If $T_{i_n} \circ \cdots \circ T_{i_1}$ is forbidden, then we can conclude $\text{int}(S_{i_n \cdots i_1}) = \emptyset$. We cannot conclude $S_{i_n \cdots i_1} = \emptyset$, because $S_{i_n \cdots i_1}$ shares edges with four other address length $n$ subsquares. If one of these has nonempty interior, the common edge with $S_{i_n \cdots i_1}$ may be nonempty.

For later use we need another observation about edges.

**Lemma 2.3.** *If* $\text{int}(S_{i_n \cdots i_2 *}) = \emptyset$ *for* $* = 1, 2, 3,$ *and* 4, *then* $\text{int}(S_{i_n \cdots i_2}) = \emptyset$.

**Proof.** Because $\text{int}(S_{i_n \cdots i_2 *}) = \emptyset$, the four compositions $T_{i_n} \circ \cdots \circ T_{i_2} \circ T_*$ are forbidden. Then the common edge of $S_{i_n \cdots i_2 1}$ and $S_{i_n \cdots i_2 2}$ is empty, as are the other three common edges of the $S_{i_n \cdots i_2 *}$, and so $\text{int}(S_{i_n \cdots i_2}) = \emptyset$. □

Corollary 2.1 has a geometric characterization that emphasizes Mandelbrot's dictum "A fractal is as easily described by what has been removed as by what remains." For an $m$-IFS with attractor $A$ let

$$\mathcal{E}(A) = \{(i_n \cdots i_1) : \text{int}(S_{i_n \cdots i_1}) \cap A = \emptyset, 1 \leq n < \infty\}. \tag{6}$$

Then Corollary 2.1 can be restated as

$$\mathcal{E}(A) = \{(i_n \cdots i_1) : \text{a substring of } i_n \cdots i_1 \text{ belongs to } \mathcal{F}_{m+1}\},$$

where $\mathcal{F}_{m+1}$ is a generating set of forbidden sequences of the $m$-IFS.

If $A$ is the attractor of an $m$-IFS and $B$ is the attractor of an $n$-IFS, both built from the transformations of Eq. (2), then certainly

$$A = B \quad \text{if and only if} \quad \mathcal{E}(A) = \mathcal{E}(B). \tag{7}$$

## 3. REVIEW OF SUBSHIFTS OF FINITE TYPE

An excellent reference for subshifts is Lind and Marcus.[1] Given a finite alphabet $A$, the *full shift* on $A$ is

$$X = A^{\mathbb{Z}} = \{(\cdots x_{-1} x_0 x_1 \cdots) : x_i \in A\}.$$

For any collection $\mathcal{F}$ of finite strings of symbols from $A$, the shift space determined by $\mathcal{F}$ is $X_{\mathcal{F}}$, the elements of X containing no element of $\mathcal{F}$. Of course, $X_{\mathcal{F}}$ can equal $X_{\mathcal{G}}$ for different collections of finite strings $\mathcal{F}$ and $\mathcal{G}$. If $\mathcal{F}$ is a finite collection, then $X_{\mathcal{F}}$ is a *subshift of finite type* of the full shift $X$.

For example, take $A = \{1, 2, 3, 4\}$ and $\mathcal{F} = \{41, 32, 23\}$. Certainly $X_{\mathcal{F}}$ is a subshift of finite type; the left side of Fig. 2 is a geometric realization of this subshift using the transformations (2). Some care is needed in reading these strings. Typically, elements of a subshift are read left to right, while strings of transformations in an IFS are read right to left, consistent with the order of composition of functions.

On the other hand, the set

$$Y = \{x \in X : \text{ the block } 12^{2n}1 \text{ does not occur in } x, \text{ for any } n\}$$

is not a subshift of finite type: no finite set of forbidden strings specifies $Y$.

The notion of subshifts of finite type can be refined: call a subshift $Y$ *$N$-step* if $Y = X_{\mathcal{F}}$ where the longest strings in $\mathcal{F}$ have length $= N + 1$.

Given a (directed) graph $G$, the *vertex shift* of $G$ has alphabet $\mathcal{A}_G = \{v_i : v_i \text{ is a vertex of } G\}$ and is defined by

$$\widehat{X}_G = \{\cdots v_1 v_0 v_1 \cdots : \text{for all } i, v_i \in \mathcal{A}_G \text{ and } G \text{ has an edge from } v_i \text{ to } v_{i+1}\}.$$

Next we describe the subshift memory reduction addressed in Sec. 2.3 of Ref. 1. Suppose $X$ is a subshift over the alphabet $\mathcal{A}$. Denote by $B_N(X)$ the set of all allowed strings of length $N$ in $X$, and define

$$\beta_N : X \to (B_N(X))^{\mathbb{Z}}$$

by

$$(\beta_N(\mathbf{x}))_i = x_i x_{i+1} \cdots x_{i+N-1},$$

where $\mathbf{x} = \cdots x_{-1} x_0 x_1 \cdots$. The $N$th *higher block shift* $X^{[N]}$ is defined by

$$X^{[N]} = \beta_N(X).$$

Proposition 2.3.9 of Ref. 1 shows that if $X$ is an $N$-step shift, then $X^{[N]}$ is a 1-step shift and there is a graph $G$ with $X^{[N]} = \widehat{X}_G$. The vertices of $G$ are the allowed strings of length $N$ in $X$, and there is an edge from vertex $a_1 \cdots a_N$ to vertex $b_1 \cdots b_N$ if and only if

(1) $a_2 \cdots a_N = b_1 \cdots b_{N-1}$, and
(2) $a_1 a_2 \cdots a_N b_N$ is an allowed string in $X$.

Note that $\widehat{X}_G$ has alphabet the vertices of $G$, that is, the allowed length $N$ strings of $X$. The forbidden length 2 strings of $\widehat{X}_G$ are the pairs of length $N$ strings of $X$ for which either of these two conditions does not hold.

## 4. IFS MEMORY REDUCTION

Can the method given in Propositon 2.3.9 of Ref. 1 be applied to reduce every $n$-IFS to a 1-IFS? Consider this example. The right side of Fig. 2 shows the attractor of the 2-IFS $I(\mathcal{F}_2)$ of Example 2.2. Proposition 2.3.9 of Ref. 1 instructs us to convert the 2-IFS $I(\mathcal{F}_2)$ to the 2nd *higher block IFS* $I^{[2]}(\mathcal{F}_2)$. How is this done?

**Example 4.1.** Realizing a 2-IFS as a 1-IFS using the 2nd higher block IFS.

The 2-IFS of Example 2.2 forbids three pairs, 14, 23, and 32, so the 2nd higher block IFS has transformations $J$ consisting of the 13 pairs allowed by $\mathcal{F}_2$. That is,

$$S_k = T_i \circ T_j,$$

where $ij$ is the $k$th element of

$$\mathcal{V} = \{11, 12, 13, 21, 22, 24, 31, 33, 34, 41, 42, 43, 44\}.$$

In constructing the transition matrix for $I^{[2]}(\mathcal{F}_2)$, recall the column index is the source of the edge, the row index the target. Then to satisfy conditions of Proposition 2.3.9 of Ref. 1, the matrix entry in column $ij$ and row $km$ is 1 if and only if

(1) $i = m$, and
(2) $kij$ is an allowed string.

With these conditions, the matrix is

$$M = \begin{bmatrix} 1&1&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&1&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&1&1&0&0&0&0 \\ 1&1&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&1&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&1&1&1&1 \\ 1&1&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&1&1&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&1&1&1&1 \\ 1&1&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&1&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&1&1&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&1&1&1 \end{bmatrix}.$$

For example, the 0 in entry $M_{13,10}$ (row 13 and column 10) results from forbidding the triple 441.

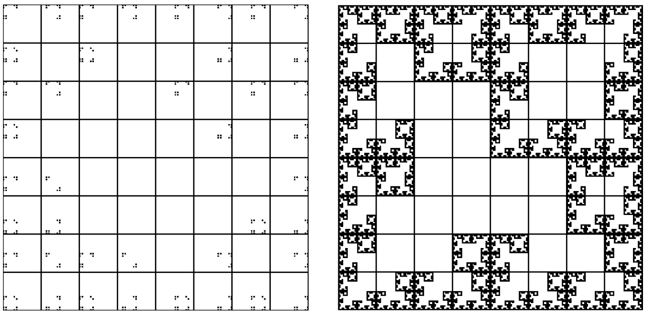

**Fig. 3** A graphical realization of the 2nd higher block IFS of Example 4.1. **Left**: Incorrect compositions. **Right**: Correct compositions.

Each vertex of the graph, and row and column of the matrix, is indexed by a composition of two transformations. For example $v_2$ corresponds to 12, i.e., $T_1 \circ T_2$, and $v_6$ to $T_2 \circ T_4$. A first guess at implementing the 1-IFS corresponding to the matrix $M$ would apply the composition $T_1 \circ T_2 \circ T_2 \circ T_4$ because $M_{2,6} = 1$. Generating the image in this way gives the picture on the left side of Fig. 3. This approach is incorrect, because it fails to account for the overlap imposed by condition 1 of Proposition 2.3.9. Taking this into account, $M_{2,6} = 1$ signals allowing the composition $T_1 \circ T_2 \circ T_4$. Interpreted this way, we obtain the picture on the right side of Fig. 3.

We see the attractor of the 2-IFS $I(\mathcal{F}_2)$ of Example 2.2 is identical with that of the 1-IFS $I^{[2]}(\mathcal{F}_2) = J(\mathcal{F}_3)$ of Example 4.1, where $\mathcal{F}_3$ is the set of pairs $pq$ for which $M_{pq} = 0$. How general is this observation?

Suppose $\mathcal{F}$ is a collection of strings of length $n+1$, so $I(\mathcal{F})$ is an $n$-IFS. The $n$th higher block IFS $I^{[n]}(\mathcal{F})$ has transformations

$$J = \{T_{i_n} \circ \cdots \circ T_{i_1} : i_n \cdots i_1 j \notin \mathcal{F} \text{ for at least one of } j = 1, 2, 3, \text{or } 4\}$$

The allowed transitions are $T_{i_n} \circ \cdots \circ T_{i_1}$ follows $T_{j_n} \circ \cdots \circ T_{j_1}$ if and only if

$$i_{n-1} \cdots i_1 = j_n \cdots j_2, \quad \text{and} \tag{8}$$

$$i_n \cdots i_1 j_1 \notin \mathcal{F}. \tag{9}$$

Denoting by $\mathcal{F}'$ the forbidden pairs of transformations from $J$, we define

$$I^{[n]}(\mathcal{F}) = J(\mathcal{F}').$$

Note this method cannot be applied to the problem of reducing a 1-IFS to a 0-IFS because the overlap condition would be vacuous.

Conditions under which a 1-IFS has the same attractor as a 0-IFS were derived in France and Lanski.[2] A consequence of Theorem 4.1 below is that the only obstruction to IFS memory reduction is contained in Ref. 2: for all $n > 1$, every $n$-IFS has the same attractor as some 1-IFS with a finite collection of transformations.

**Theorem 4.1.** *The $n$-IFS $I(\mathcal{F})$ and the 1-IFS $I^{[n]}(\mathcal{F})$ have the same attractor.*

**Proof.** Denote by $A^1$ the attractor of $I(\mathcal{F})$, and by $A^n$ the attractor of $I^{[n]}(\mathcal{F})$. Regions of $A^1$ can be given addresses that are $I$-strings, that is, strings over $I$. Regions of $A^n$ can be given addresses that are $J$-strings, but comparisons of $A^1$ and $A^n$ are easier if regions of $A^n$ are given addresses that are $I$-strings.

By Corollary 2.1, the region $A^1_{i_q \cdots i_1}$ is empty if and only if $i_q \cdots i_1$ contains a substring $j_{n+1} \cdots j_1 \in \mathcal{F}$. Then either

(1) for all $k = 1, 2, 3$, and 4, $j_{n+1} \cdots j_2 k \in \mathcal{F}$, or
(2) for some $k = 1, 2, 3$, or 4, $j_{n+1} \cdots j_2 k \notin \mathcal{F}$.

In Case 1, $T_{j_{n+1}} \circ \cdots \circ T_{j_2} \notin J$ and so $A^n_{j_{n+1} \cdots j_2}$ is empty. It follows that $A^n_{i_q \cdots i_1}$ is empty, regardless of whether or not all the length $n$ substrings of $i_q \cdots i_1$ belong to $J$, or if those with length $n-1$ overlaps are related by allowed $J$-transitions.

In Case 2, $T_{j_{n+1}} \circ \cdots \circ T_{j_2} \in J$, but because $j_{n+1} \cdots j_1 \in \mathcal{F}$, $A^n_{j_{n+1} \cdots j_1}$ is empty, regardless of whether or not $j_n \cdots j_1 \in J$. Arguing as in case 1, $A^n_{i_q \cdots i_1}$ is empty.

That is, every $I$-address empty in $A^1$ also is empty in $A^n$, so $A^n \subseteq A^1$.

For the other containment, suppose the region $A^n_{i_q \cdots i_1}$ is empty. Then either

(1) for some substring $j_n \cdots j_1$ of $i_q \cdots i_1$ we have $j_n \cdots j_1 k \in \mathcal{F}$ for $k = 1, 2, 3$, and 4, or
(2) condition 1 fails for all length $n$ substrings of $i_q \cdots i_1$, but a length $n+1$ substring $k_{n+1} \cdots k_1$ of $i_q \cdots i_1$ is an element of $\mathcal{F}$.

Condition 1 implies the composition $T_{j_n} \circ \cdots \circ T_{j_1}$ is not an element of $J$; condition 2 that $T_{k_{n+1}} \circ \cdots \circ T_{k_2}$ and $T_{k_n} \circ \cdots \circ T_{k_1}$ are elements of $J$, but the first cannot follow the second because of (9). (Note the overlap condition (8) need not be considered, because two length $n$ substrings $i_{a+n} \cdots i_{a+1}$ and $i_{a+n-1} \cdots i_a$ necessarily satisfy (8).) In both cases, $A^1_{i_q \ldots i_1}$ is empty. That is, every $I$ address empty in $A^n$ also is empty in $A^1$, so $A^1 \subseteq A^n$. □

**Example 4.2.** Realizing a 3-IFS as a 1-IFS using the 3rd higher block IFS.

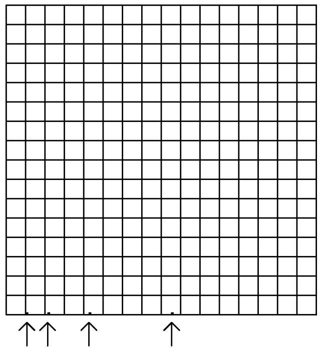

**Fig. 4** A graphical realization of the 3rd higher block IFS of Example 4.2. Arrows indicate the points of the attractor.

A 3rd higher bolck IFS could have as many as $4^3$ transformations. To keep this example manageable, we use a very simple 3-IFS, characterized by these allowed strings

$$\mathcal{A} = \{1112, 1121, 1211, 2111\}.$$

All other length 4 compositions are forbidden. The attractor consists of the four points comprising the 4-cycle obtained by iterating $T_2 \circ T_1 \circ T_1 \circ T_1$. That is, the points $(1/15, 0)$, $(2/15, 0)$, $(4/15, 0)$, and $(8/15, 0)$, having addresses $(1112)^\infty$, $(1121)^\infty$, $(1211)^\infty$, and $(2111)^\infty$. See Fig. 4. The transformations of the 3rd higher block IFS consist of the allowed triples

$$S_1 = T_1 \circ T_1 \circ T_1, \quad S_2 = T_1 \circ T_1 \circ T_2,$$
$$S_3 = T_1 \circ T_2 \circ T_1, \quad \text{and} \quad S_4 = T_2 \circ T_1 \circ T_1.$$

The transition matrix, imposed by conditions (8) and (9), is

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}.$$

Figure 4 also shows the attractor of this 1-IFS.

## 5. OTHER WAYS TO REDUCE MEMORY

The memory reduction from an $n$-IFS to a 1-IFS given in Theorem 4.1 establishes the existence of a solution of the memory reduction problem, but it does not address the issue of classifying such reductions, or even the simpler problem of finding the most efficient (in terms of fewest transformations) memory reduction. In Fig. 5 we see the attractor of the 2-IFS $I(\mathcal{F})$ with

$$\mathcal{F} = \{22, 23, 32, 33, 43, 421\}.$$

Because five compositions $T_i \circ T_j$ are forbidden, the 1-IFS given by the construction of Sec. 4 has $16 - 5 = 11$ transformations. Can we do better? That is, can we find a 1-IFS with the same attractor and fewer than 11 transformations?

**Example 5.1.** An efficient 1-IFS of Fig. 5.

The forbidden triple 421 is not a consequence of any of the forbidden pairs of $\mathcal{F}$. The first step is to subdivide $T_4$ and $T_2$, obtaining a new set of ten transformations

$$\begin{aligned} &R_1 = T_1, \quad R_2 = T_2 \circ T_1, \quad R_3 = T_2 \circ T_2, \\ &R_4 = T_2 \circ T_3, \quad R_5 = T_2 \circ T_4, \\ &R_6 = T_3, \quad R_7 = T_4 \circ T_1, \quad R_8 = T_4 \circ T_2, \\ &R_9 = T_4 \circ T_3, \quad R_{10} = T_4 \circ T_4. \end{aligned} \tag{10}$$

Some of these may be unnecessary, depending on the forbidden pairs of $T_i$. For example, if $T_4 \circ T_3$ is forbidden, then $R_9$ can be dropped from the $R_i$.

We find the pairs of $R_i$ that are forbidden as a consequence of the forbidden pairs and triples of $T_i$. Because some $R_i$ are compositions of two $T_i$, to identify all the forbidden pairs of $R_i$, we must consider superstrings of elements of $\mathcal{F}$.

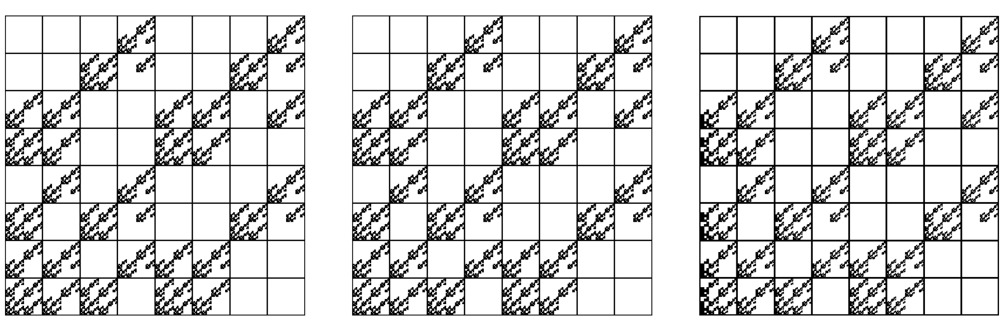

**Fig. 5** Attractors of the 2-IFS of Example 5.1 (**left**), the equivalent 1-IFS using the method of Sec. 4 (**middle**), and the efficient equivalent 1-IFS (**right**).

Specifically,

$$*22, 22*, *23, 23*, *32, 32*, *33,$$
$$33*, *43, 43*, *421, 421*, \qquad (11)$$

where $*$ stands for 1, 2, 3, and 4.

We explore $*22$ and $22*$ in detail, then state the results for the other superstrings in (11). We use the notation $\underset{\rightarrow}{R}$ to indicate translating $T$ strings (compositions of the $T_i$) to $R$ strings, and $a(j)_T$ for the $R$ string $a$ followed by the still untranslated $T$ string $j$. In these symbol strings we denote the 10 for $R_{10}$ by $X$.

$$\begin{aligned} *22 &= 122, 222, 322, 422 \\ &\overset{R}{\rightarrow} 13, 3(2)_T, 63, 8(2)_T \\ &= 13, 32, 33, 34, 35, 63, 82, 83, 84, 85; \\ 22* &= 221, 222, 223, 224 \\ &\overset{R}{\rightarrow} 31, 3(2)_T, 36, 3(4)_T \\ &= 31, 32, 33, 34, 35, 36, 37, 38, 39, 3X \\ &= 3. \end{aligned}$$

Not surprisingly, forbidding the $T$ string 22* forbids the $R$ transformation $R_3$. Then this transformation is not needed, and every forbidden string containing 3 can be deleted from the list of forbidden strings.

Some of these, *32 for example, include $R$ string of length 3 that are forbidden as conseqences of length 2 forbidden $R$ strings arising from other elements of (11). We list only the length 2 $R$ strings.

$$\begin{aligned} *32 &\overset{R}{\rightarrow} 42, 43, 44, 45, 92, 93, 94, 95 \\ 32* &\overset{R}{\rightarrow} 62, 63, 64, 65 \\ *23 &\overset{R}{\rightarrow} 14, 36, 64, 86 & 23* &\overset{R}{\rightarrow} 4 \\ *33 &\overset{R}{\rightarrow} 46, 96 & 33* &\overset{R}{\rightarrow} 66 \\ *43 &\overset{R}{\rightarrow} 19, 56, 69, X6 & 43* &\overset{R}{\rightarrow} 9 \\ *421 &\overset{R}{\rightarrow} 52, X2 & 421* &\overset{R}{\rightarrow} 81 \end{aligned}$$

Removing the length 2 $R$ strings containing forbidden length 1 $R$ strings, we obtain this 1-IFS

$$R_1, R_2, R_5, R_6, R_7, R_8, R_{10}, \qquad (12)$$

with forbidden strings

$$52, 56, 62, 65, 66, 81, 82, 85, 86, X2, X6. \qquad (13)$$

Writing the transition matrix, we see the $R$ addresses 1, 2, and 7 are romes, and there are transitions to each nonrome from some rome, so additional reduction in the level of memory is possible. This attractor can be realized as the attractor of a 0-IFS, but because there are loops between nonromes, $8 \leftrightarrow X$, for example, the equivalent 0-IFS requires infinitely many transformations.

**Example 5.2.** Another efficient 1-IFS equivalent to a 2-IFS.

For another example, recall the right side of Fig. 2, the attractor of the 2-IFS with forbidden strings

$$\mathcal{F}_2 = \{14, 23, 32, 441\}.$$

Because 441 does not contain any forbidden pair, to forbid it as a pair, we must subdivide $T_4$ into four transformations. Call the new transformations $S_i$:

$$S_1 = T_1, \quad S_2 = T_2, \quad S_3 = T_3, \quad S_4 = T_4 \circ T_1,$$
$$S_5 = T_4 \circ T_2, \quad S_6 = T_4 \circ T_3, \quad S_7 = T_4 \circ T_4. \quad (14)$$

Carrying out the analogous analysis, the forbidden $T$ strings *14, 41*, *23, 23*, *32, 32*, *441, 441* translate into the forbidden $R$ strings

$$14, 15, 16, 17, 23, 32, 44, 45, 46, 47, 53, 62, 71, 74.$$

The 1-IFS with these transformations and forbidden pairs generates the attractor pictured in the right side of Fig. 2.

While fairly straightforward, this construction is tedious. A more compact and systematic method of finding the efficient equivalent 1-IFS for a given 2-IFS is given in the conjecture. First, two definitions. The label $i$ of a transformation is *subdivided* if in the efficient equivalent IFS the transformation $T_i$ must be replaced by $T_i \circ T_1$, $T_i \circ T_2$, $T_i \circ T_3$, and $T_i \circ T_4$. A forbidden composition $i_n \circ \cdots \circ i_1$ is *primary* if no substring is forbidden.

**Conjecture.** Given a 2-IFS with forbidden strings $\mathcal{F}$, the equivalent efficient 1-IFS can be generated by these steps.

(1) Remove all non-primary strings from $\mathcal{F}$.
(2) For every $ijk \in \mathcal{F}$, subdivide $i$ and $j$. Say $S$ is the total number of subdivided labels. This gives the initial efficient generating set of transformations. Some of these may be removed in the process of reducing the forbidden strings.
(3) Efficient reduction of a forbidden triple $ijk$.
   (a) If $k$ is subdivided, $S + 4$ addresses are needed.
   (b) If $k$ is not subdivided, $S + 1$ addresses are needed.
(4) Efficient reduction of a forbidden pair $ij$.

(a) If $i$ is subdivided, then some generating transformation maps to address $ij$, so this transformation can be removed from the generating set. However, some elements still are necessary to forbid $*ij$.

(i) If $j$ is subdivided, then $4S + 4 - S$ addresses are needed.

(ii) If $j$ is not subdivided, then 4 addresses are needed.

(b) If $i$ is not subdivided and $j$ is subdivided, then $4S + 4$ addresses are needed.

(c) If neither $i$ nor $j$ are subdivided, then $S+1$ addresses are needed.

(5) Efficient reduction of a forbidden address $i$. Remove $T_i$ and all compositions including $T_i$ from the generating set of transformations; remove every forbidden address that includes $i$.

As an illustration, we apply this method to find the equivalent efficient IFS of Example 5.1. First note that every element of $\mathcal{F} = \{22, 32, 23, 33, 43, 421\}$ is a primary string. Next, the forbidden string 421 requires we subdivide 4 and 2, obtaining the generating set $R_1, \ldots, R_{10}$ of (10). Note $S = 2$.

By 3(a), 421 gives rise to 3 forbidden pairs. Because 2 and 4 are subdivided, *421 gives rise to two pairs $2421_T = 52_R$ and $4421_T = X2_R$, and 421* gives rise to $81_R$.

Applying 4(a) to 22, 23, and 43, the $R_3 = T_2 \circ T_2$, $R_4 = T_2 \circ T_3$, and $R_9 = T_4 \circ T_3$ can be eliminated from the generating set of transformations.

By 4(a)(i), 22 gives rise to 10 forbidden pairs. The $T$ string *22 gives the $R$ strings 13, 32, 33, 34, 35, 63, 82, 83, 84, and 85. The $T$ string 22* gives the $R$ strings 3*, the reason $R_3$ is eliminated.

By 4(a)(ii), both 23 and 43 give rise to 4 forbidden pairs: the $T$ string *23 gives the $R$ strings 14, 36, 64, and 86; 23* gives 4; *43 gives 19, 56, 69, and $X6$; 43* gives 9.

By 4(b), 32 gives rise to 12 forbidden pairs. From *32, $232_T$ produces $T$ strings 2321, 2322, 2323, and 2324, hence $R$ strings 42, 43, 44, and 45. Similarly, $432_T$ gives $R$ strings 92, 93, 94, and 95. The $T$ strings of 32* give the $R$ strings 62, 63, 64, and 65.

By 4(c), 33 gives rise to 3 forbidden pairs. From *33, the $T$ strings 233 and 433 give the $R$ strings 46 and 96. The $T$ strings 33* give the $R$ string 66.

Aggregating these results, the transformations that remain are those of (12). Removing the forbidden strings that contain 3, 4, or 9 we obtain the forbidden set (13).

The correctness of this conjecture, and its generalization to reductions of $n$-IFS to 1-IFS, will be explored in Fiross *et al.*[25]

## 6. SOME DIMENSION COMPUTATIONS

For 1-IFS the dimension of the attractor can be computed by applying Eq. (1) using the transition matrix and scaling factors of the 1-IFS . With Examples 4.1 and 5.1 we illustrate the computation of the dimension of a 2-IFS attractor by applying memory reduction and computing the dimension of the attractor of the 1-IFS obtained.

In Example 4.1 we might expect we would replace each 1 in the matrix $M$ with $(1/4)^d$ because each of the 13 transformations of $I^{[2]}(\mathcal{F}_2)$ is a composition $T_i \circ T_j$, so has contraction factor 1/4. This ignores the overlap of the row and column indices. For example, the 1 in $M_{26}$ refers to $T_1 \circ T_2$ following $T_2 \circ T_4$. Because of the overlap, this 1 allows the composition $T_1 \circ T_2 \circ T_4$. That is, in going from $T_2 \circ T_4$ to $T_1 \circ T_2 \circ T_4$, only the transformation $T_1$ is applied. Consequently, in computing the dimension of the attractor, each 1 of $M$ must be replaced by $(1/2)^d$. So in Example 4.1, Eq. (1) becomes

$$1 = \rho(M_{ij}(1/2)^d) = (1/2)^d \rho(M). \qquad (15)$$

The (numerical) eigenvalues of $M$ are

$$0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 3.15276, \\ -0.576379 \pm i\sqrt{0.549684},$$

so $\rho(M) \approx 3.15276$ and $d \approx \log_2(3.15276) \approx 1.65657$.

For Example 5.1, the transition matrix is

$$N = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}.$$

In the transformations (14), $S_1, S_2$, and $S_3$ have contraction factor 1/2, while $S_4, S_5, S_6$, and $S_7$ have contraction factor 1/4. Then Eq. (1) becomes

$$\rho \begin{bmatrix} x & x & x & x^2 & x^2 & x^2 & 0 \\ x & x & 0 & x^2 & x^2 & 0 & x^2 \\ x & 0 & x & x^2 & 0 & x^2 & x^2 \\ 0 & x & x & 0 & x^2 & x^2 & 0 \\ 0 & x & x & 0 & x^2 & x^2 & x^2 \\ 0 & x & x & 0 & x^2 & x^2 & x^2 \\ 0 & x & x & 0 & x^2 & x^2 & x^2 \end{bmatrix} = 1$$

where $x = (1/2)^d$. Numerical explorations show this equation is satisfied for $d \approx 1.65657$.

This agreement is no surprise: Examples 4.1 and 5.1 have the same attractor. Example 5.1 is included to illustrate the computational issues that arise if different scaling factors occur.

The main point is this: the method of Eq. (1) for computing dimensions of 1-IFS need not be extended to $n$-IFS. Rather, apply Theorem 4.1 to find a 1-IFS generating the same attractor and compute the dimension by applying Eq. (1) to this 1-IFS.

## 7. CONCLUSION

Adapting the concept of $n$th higher block codes from symbolic dynamics, the attractor of any $n$-IFS can be realized as the attractor of a 1-IFS, the $n$th higher block IFS. Then the theorem of Frame and Larski[2] determines which of these can be realized as the attractor of a 0-IFS. The $n$th higher block IFS often does not generate the attractor by a 1-IFS with the fewest transformations. Section 5 provides a method for finding the most efficient 1-IFS, in the sense of using the fewest transformations, having the same attractor as a given $n$-IFS. The conjecture of that section gives a quick way to find the minimum number of transformations and forbidden strings. Details will be provided in Gross *et al.*[25]

By showing that the attractor of an $n$-IFS can be realized as the attractor of a 1-IFS, the method of Mauldin and Williams[17] can be used to compute the Hausdorff dimension of the attractors of $n$-IFS.

Dimension is one measure of the complexity of compact subsets of Euclidean space. For attractors of IFS with memory, the length of memory might have served as another measure of complexity. Theorem 4.1 shows this is not a productive direction to pursue. On the other hand, this memory-reduction method does point out an interesting trade-off between the length of memory and number of transformations needed to generate a fractal.

In Bedient *et al.*[26] we explore some relations between IFS with different levels of memory, and build up a hierarchy of attractors resulting from different embeddings of $m$-step memory rules into $n$-step memory rules, for $n < m$.

## REFERENCES


1. D. Lind and B. Marcus, *An Introduction to Symbolic Dynamics and Coding* (Cambridge Univerisity Press, 1995).
2. M. Frame and J. Lanski, When is a recurrent IFS attractor a standard IFS attractor? *Fractals* **7** (1999) 257–266.
3. M. Barnsley, *Fractals Everywhere*, 2nd edn. (Academic Press, 1993).
4. K. Falconer, *Fractal Geometry. Mathematical Foundations and Applications* (John Wiley, New York, 1990).
5. D. Ashlock and J. Golden, Chaos automata: iterated function systems with memory, *Physica D* **181** (2003) 274–285.
6. M. Barnsley, J. Elton and D. Hardin, Recurrent iterated function systems, *Constr. Approx.* **5** (1989) 3–31.
7. A. Barbé, On a class of fractal matrices (III): limit structures and hierarchical iterated function systems, *Int. J. Bifurc. Chaos* **5** (1995) 1119–1156.
8. A. Barbé, F. von Haesler and G. Skordev, Limit sets of restricted random substitutions, *Fractals* **14** (2006) 37–47.
9. G. Edgar, *Integral, Probabilty, and Fractal Measures* (Springer-Verlag, New York, 1998).
10. G. Edgar and R. Mauldin, Multifractal decompositions of digraph recursive fractals, *Proc. Lond. Math. Soc.* **65** (1992) 604–628.
11. B. Hambly and S. Nyberg, Finitely ramified graph-directed fractals, spectral asymptotics and the multidimensional renewal theorem, *Proc. Edinburgh Math. Soc.* **46** (2003) 1–34.
12. J. King and J. Geronimo, Singularity spectrum for recurrent IFS attractors, *Nonlinearity* **6** (1992) 337–348.
13. J. Layman and T. Womack, Linear Markov iterated function systems, *Comput. Graph.* **14** (1990) 343–353.
14. P. Massopust, *Fractal Functions, Fractal Surfaces, and Wavelets* (Academic Press, San Diego, CA, 1994).
15. L. Máté, *On Full-Like Sofic Shifts*, preprint.
16. R. Mauldin, S. Graf and S. Williams, Exact Hausdorff dimension in random recursive constructions, *Proc. Natl. Acad. Sci. USA* **84** (1987) 3959–3961.
17. R. Mauldin and S. Williams, Hausdorff dimension in graph directed constructions, *Trans. Am. Math. Soc.* **309** (1988) 811–829.
18. J. Hart, *Iterated Function Systems and Recurrent Iterated Function Systems*, PhD thesis (Washington State University, Pullman, WA, USA, 1996).
19. H.-O. Peitgen, H. Jürgens and D. Saupe, *Chaos and Fractals. New Frontiers in Science*, 2nd edn. (Springer-Verlag, New York, 2004).
20. H. Jeffrey, Chaos game representation of gene structure, *Nucleic Acid Res.* **18** (1990) 2163–2170.
21. H. Jeffrey, Chaos game visualization of sequences, *Comput. Graph.* **16** (1992) 25–33.

22. B.-L. Hao, H. Lee and S.-Y. Zhang, Fractals related to long DNA sequences and complete genomes, *Chaos Solitons Fractals* **11** (2000) 825–836.
23. T. Suzuki, T. Ikeguchi and M. Suzuki, Applications of chaos game respresentation to nonlinear time series analysis, *Fractals* **14** (2006) 27–35.
24. V. Metz, "Laplacians" on finitely ramified, graph directed fractals, *Math. Ann.* **330** (2004) 809–828.
25. K. Gross, R. Bedient and M. Frame, Efficient memory reduction of IFS with memory, in preparation.
26. R. Bedient, M. Frame, K. Gross, J. Lanski and B. Sullivan, Higher block IFS 2: relations between IFS with different levels of memory, to appear in *Fractals*.