Memory Reduction in Iterated Function Systems

Closing off (kind of) an avenue of measuring fractal complexity

Brendan W. Sullivan

Carnegie Mellon University
Undergraduate Math Club

February 15, 2012

1 Background
- Fractals
- Applications
- Iterated Function Systems

2 IFS with Memory
- 1-IFS
- Transition Graphs/Matrices
- Classification
- 2-IFS and beyond

3 Memory Reduction
- Theory
- Demonstration
- Results
- Conclusions and Future Work

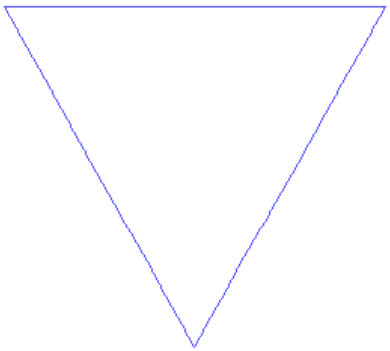
Definitions

A fractal is . . .

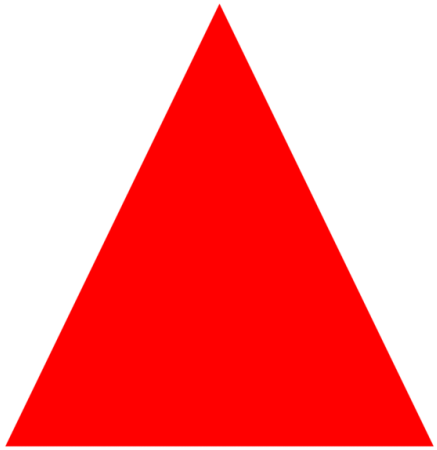
- . . . a set exhibiting self-similarity.
- . . . a set that "looks irregular; but more importantly, after it is magnified it still looks irregular." [1]
- . . . "by definition a set for which the Hausdorff-Besicovitch dimension strictly exceeps the topological dimension." [2]

Canonical examples

Figure: Simple, iteratively generated fractals



(a) Van Koch curve



(b) Sierpinski triangle (deterministic)

(c) Sierpinski triangle (probabilistic)

Dimensional analysis

Let X be a metric space. If $S \subseteq X$ and $d \in [0, \infty)$, the **d-dimensional Hausdorff measure of S** is

$$C_H^d(S) = \inf \left\{ \sum_i r_i^d \mid \exists \text{ cover of } S \text{ by balls with radii } r_i > 0 \right\}$$

and the **Hausdorff dimension** of S is

$$\dim_H(S) = \inf\{d \geq 0 \mid C_H^d(S) = 0$$

The **Lebesgue covering dimension** of a topological space X is the minimum n such that every finite open cover $\mathcal{A}$ of X admits a finite open cover which refines $\mathcal{A}$ in which no point is included in more than $n + 1$ elements.

Dimensional analysis

Example

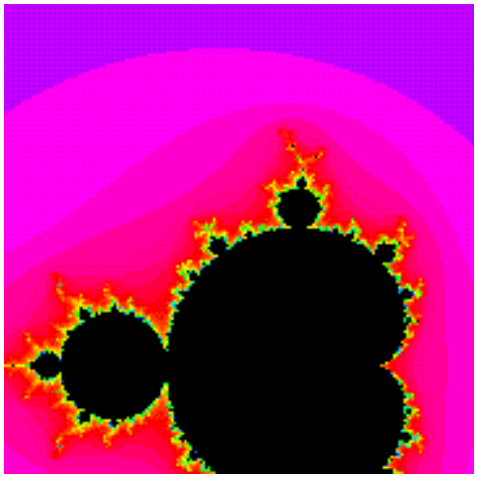
Cantor set: Hausdorff dim. $\frac{\ln 2}{\ln 3}$; topological dim. 0

Sierpinski $\triangle$: Hausdorff dim. $\frac{\ln 3}{\ln 2}$; topological dim. 1

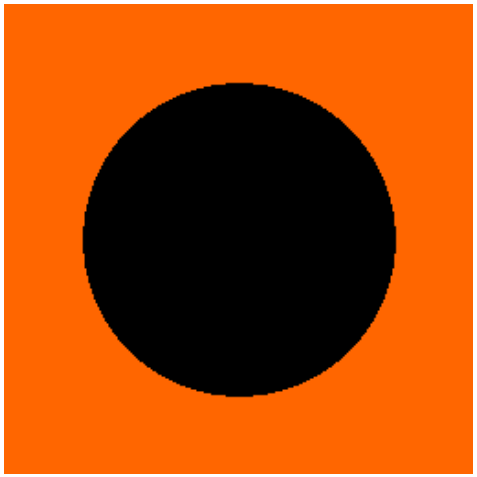
Van Koch curve: Hausdorff dim. $\frac{\ln 4}{\ln 3}$; topological dim. 1

Canonical examples

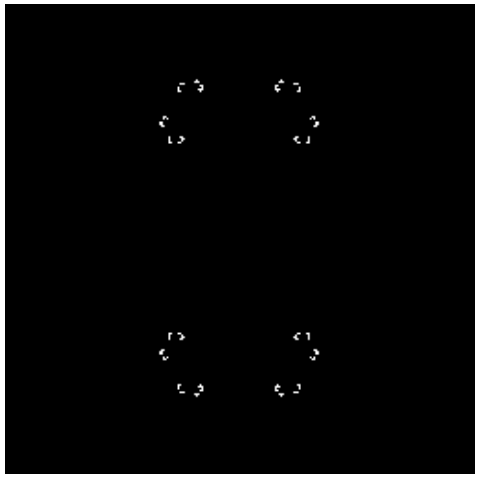
Figure: More complex, iteratively generated fractals



(a) Mandelbrot set



(b) Mandelbrot set growth

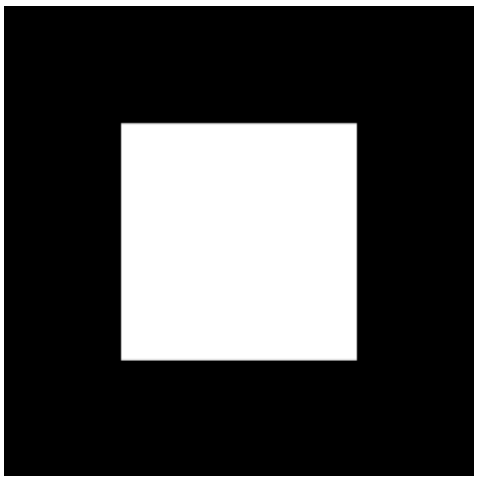


(c) Julia set

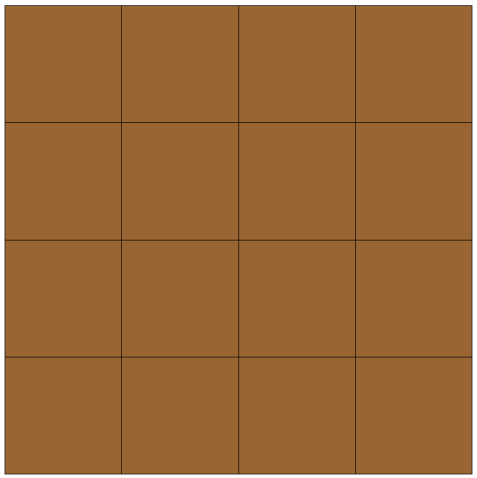
Canonical examples

Figure: Fractals generated by iterated function systems

(a) Shrinking box border

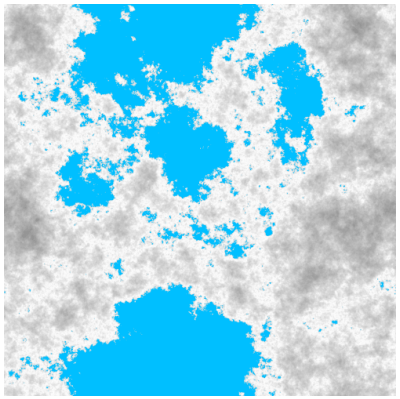


(b) Pointy leaf boxes



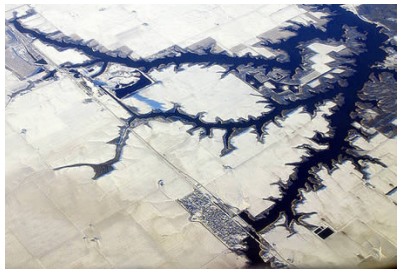
(c) Sierpinskitty trifelinegle

Nature



(a) Clouds [3]

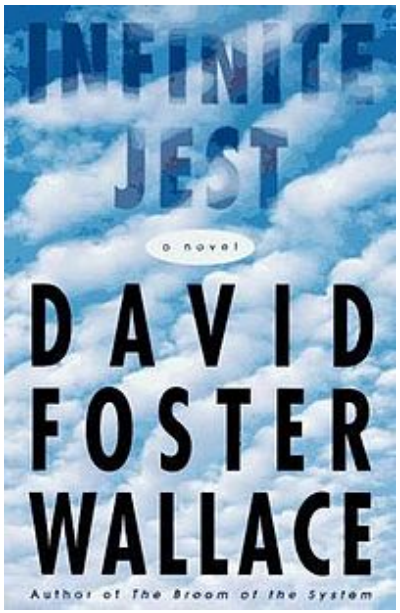
(b) Plants



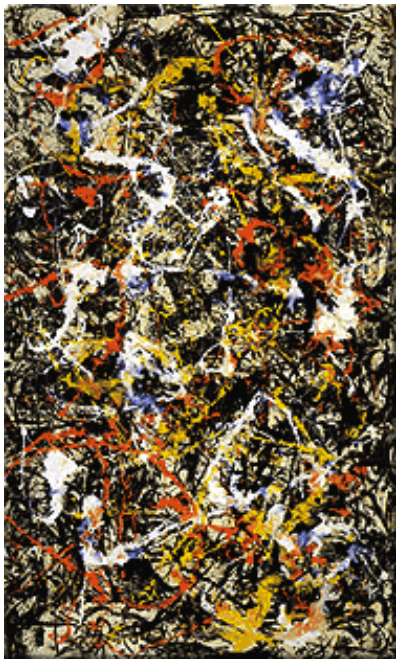
(c) Coastlines

(d) Snowflakes [4]

Art



(a) Literature



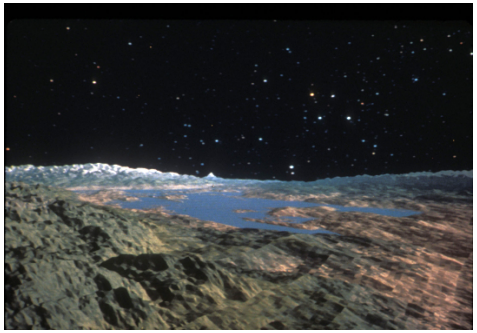
(b) Painting [5]

(c) Music [6]

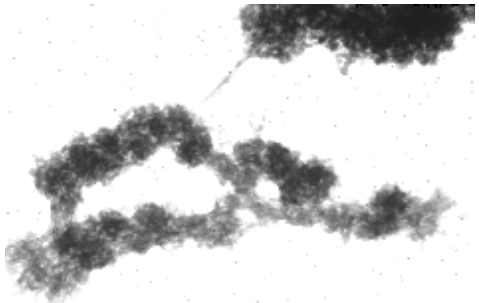
"Wind and Metal" [7]

Science and Computing

(a) Anatomy [9]



(b) Graphics [17]



(c) DNA [18]

Formal definition

Let (X, ρ) be a metric space and $T : X \to X$ a function.

Formal definition

Let (X, ρ) be a metric space and $T : X \to X$ a function.

Definition

We say T is a **contraction map** *iff*

$$\exists k \in [0, 1).\ \forall a, b \in X.\ \rho(T(x), T(y)) \leq k\rho(x, y)$$

Formal definition

Let (X, ρ) be a metric space and $T : X \to X$ a function.

Definition

We say T is a **contraction map** *iff*

$$\exists k \in [0, 1).\ \forall a, b \in X.\ \rho(T(x), T(y)) \leq k\rho(x, y)$$

Theorem (Banach Fixed Point)

If (X, ρ) is a complete metric space and $T : X \to X$ a contraction map, then T has exactly one fixed point; i.e.

$$\exists!\, \hat{x} \in X.\ T(\hat{x}) = \hat{x}$$

Formal definition

Definition

An **IFS** *is a finite set of contraction maps* $\mathcal{T} = \{T_i \mid i = 1, \ldots, N\}$ *on a complete metric space* (X, ρ).

Formal definition

Definition

An **IFS** *is a finite set of contraction maps* $\mathcal{T} = \{T_i \mid i = 1, \ldots, N\}$ *on a complete metric space* (X, ρ).

The **Hutchinson Operator** H *applies an IFS to any subset* $S \in \mathcal{P}(X)$ *via*

$$H(S) = \bigcup_{i=1}^{N} T_i(S)$$

Formal definition

Definition

An **IFS** *is a finite set of contraction maps* $\mathcal{T} = \{T_i \mid i = 1, \ldots, N\}$ *on a complete metric space* (X, ρ).

The **Hutchinson Operator** H *applies an IFS to any subset* $S \in \mathcal{P}(X)$ *via*

$$H(S) = \bigcup_{i=1}^{N} T_i(S)$$

Question: Does H have any "fixed points"?

Formal definition

Partial Answer:

Theorem (Hutchinson, 1981)

For $X = \mathbb{R}^d$ with the standard metric, every IFS admits a unique compact set $A \subset \mathbb{R}^d$ satisfying $H(A) = A$.

A is the **attractor** of the IFS and is a fractal.

Formal definition

Partial Answer:

Theorem (Hutchinson, 1981)

For $X = \mathbb{R}^d$ with the standard metric, every IFS admits a unique compact set $A \subset \mathbb{R}^d$ satisfying $H(A) = A$.

Proof.

Show that H is a contraction map on $K(X)$, the set of compact subsets of X. Apply Banach Fixed Point. □

A is the **attractor** of the IFS and is a fractal.

Formal definition

Constructive approach:

- Choose an initial compact set $S_0 \in K(X)$

Formal definition

Constructive approach:

- Choose an initial compact set $S_0 \in K(X)$
- Iteratively apply H:

$$S_{i+1} = H(S_0) = T_1(S_i) \cup \cdots \cup T_N(S_i)$$

Formal definition

Constructive approach:

- Choose an initial compact set $S_0 \in K(X)$
- Iteratively apply H:

$$S_{i+1} = H(S_0) = T_1(S_i) \cup \cdots \cup T_N(S_i)$$

- Take limit:

$$A = \lim_{i \to \infty} H^i(S_0)$$

Formal definition

Constructive approach:

- Choose an initial compact set $S_0 \in K(X)$
- Iteratively apply H:

$$S_{i+1} = H(S_0) = T_1(S_i) \cup \cdots \cup T_N(S_i)$$

- Take limit:

$$A = \lim_{i \to \infty} H^i(S_0)$$

Proof.

Corollary to Banach Fixed Point: this limit converges to A for any choice of S_0. □

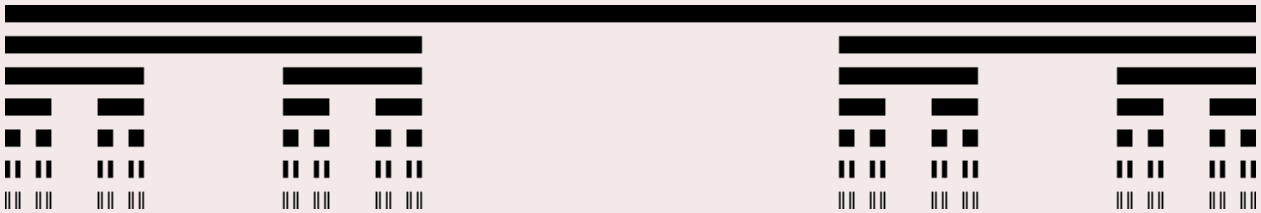
Examples

Standard contraction maps in $\mathbb{R}^d$ are scalings (with factor $r < 1$), rotations, reflections, translations.

Examples

Example

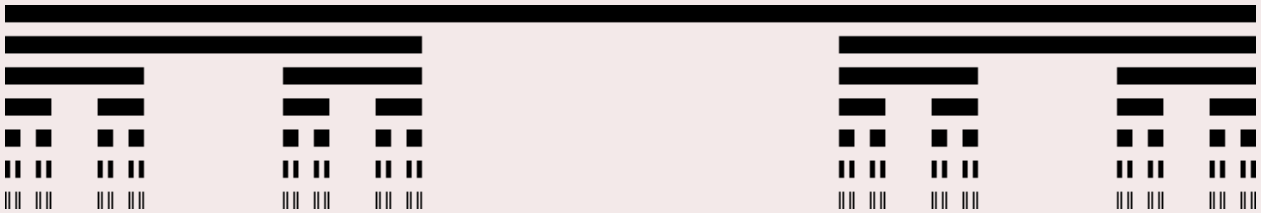
Cantor Set, $\mathcal{C}$:



Examples

Example

Cantor Set, $\mathcal{C}$:



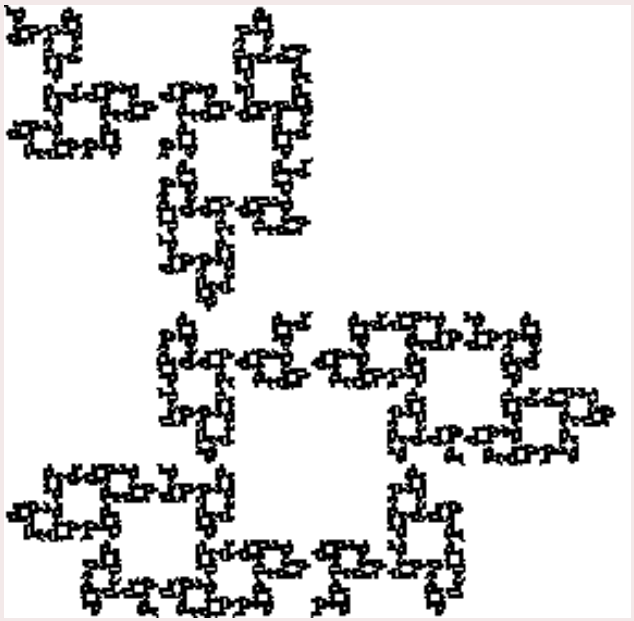
$$T_1(x) = \frac{x}{3} \qquad T_2(x) = \frac{x}{3} + \frac{2}{3}$$

$$\mathcal{C} = T_1(\mathcal{C}) \cup T_2(\mathcal{C})$$

Examples

Example

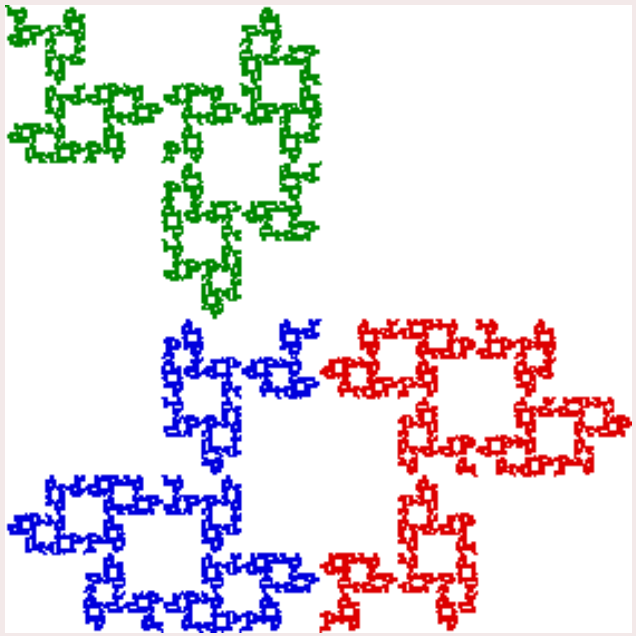
Cantor-style dust:



Examples

Example

Cantor-style dust:



Standard context

$X = [0,1]^2$ with standard metric and $\mathcal{T} = \{T_1, T_2, T_3, T_4\}$, where

$$T_1(x,y) = \left(\frac{x}{2}, \frac{y}{2}\right) + \left(0, 0\right)$$
$$T_2(x,y) = \left(\frac{x}{2}, \frac{y}{2}\right) + \left(\frac{1}{2}, 0\right)$$
$$T_3(x,y) = \left(\frac{x}{2}, \frac{y}{2}\right) + \left(0, \frac{1}{2}\right)$$
$$T_4(x,y) = \left(\frac{x}{2}, \frac{y}{2}\right) + \left(\frac{1}{2}, \frac{1}{2}\right)$$

Standard context

$X = [0,1]^2$ with standard metric and $\mathcal{T} = \{T_1, T_2, T_3, T_4\}$, where

$$T_1(x,y) = \left(\frac{x}{2}, \frac{y}{2}\right) + \left(0, 0\right)$$

$$T_2(x,y) = \left(\frac{x}{2}, \frac{y}{2}\right) + \left(\frac{1}{2}, 0\right)$$

$$T_3(x,y) = \left(\frac{x}{2}, \frac{y}{2}\right) + \left(0, \frac{1}{2}\right)$$

$$T_4(x,y) = \left(\frac{x}{2}, \frac{y}{2}\right) + \left(\frac{1}{2}, \frac{1}{2}\right)$$

3	4
1	2

Standard context

$X = [0,1]^2$ with standard metric and $\mathcal{T} = \{T_1, T_2, T_3, T_4\}$, where

$$T_1(x,y) = \left(\frac{x}{2}, \frac{y}{2}\right) + \left(0, 0\right)$$

$$T_2(x,y) = \left(\frac{x}{2}, \frac{y}{2}\right) + \left(\frac{1}{2}, 0\right)$$

$$T_3(x,y) = \left(\frac{x}{2}, \frac{y}{2}\right) + \left(0, \frac{1}{2}\right)$$

$$T_4(x,y) = \left(\frac{x}{2}, \frac{y}{2}\right) + \left(\frac{1}{2}, \frac{1}{2}\right)$$

33	34	43	44
31	32	41	42
13	14	23	24
11	12	21	22

Standard context

$X = [0,1]^2$ with standard metric and $\mathcal{T} = \{T_1, T_2, T_3, T_4\}$, where

$$T_1(x,y) = \left(\frac{x}{2}, \frac{y}{2}\right) + \left(0, 0\right)$$

$$T_2(x,y) = \left(\frac{x}{2}, \frac{y}{2}\right) + \left(\frac{1}{2}, 0\right)$$

$$T_3(x,y) = \left(\frac{x}{2}, \frac{y}{2}\right) + \left(0, \frac{1}{2}\right)$$

$$T_4(x,y) = \left(\frac{x}{2}, \frac{y}{2}\right) + \left(\frac{1}{2}, \frac{1}{2}\right)$$

333	334	343	344	433	434	443	444
331	332	341	342	431	432	441	442
313	314	323	324	413	414	423	424
311	312	321	322	411	412	421	422
133	134	143	144	233	234	243	244
131	132	141	142	231	232	241	242
113	114	123	124	213	214	223	224
111	112	121	122	211	212	221	222

Standard context

Being **in state** i means applying T_i
Addresses indicate the *reverse* order of the transformations required to land in that box.

Example

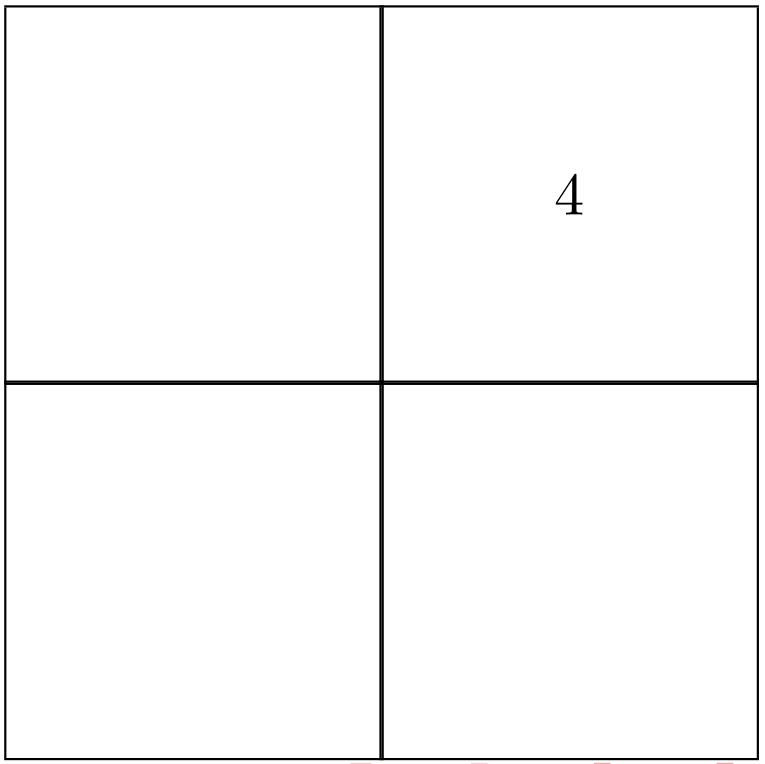
- Map composition: $(T_2 \circ T_1 \circ T_4)(X)$
- Address: 214
- State transformation: $4 \to 1 \to 2$

Standard context

Being **in state** i means applying T_i
Addresses indicate the *reverse* order of the transformations required to land in that box.

Example

- Map composition: $(T_2 \circ T_1 \circ T_4)(X)$
- Address: 214
- State transformation: $4 \to 1 \to 2$

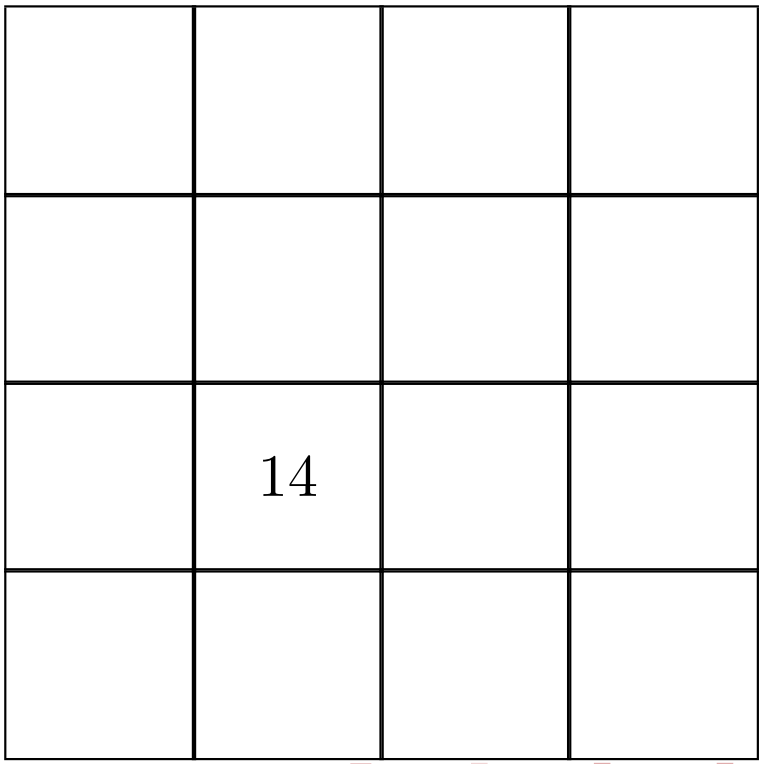


Standard context

Being **in state** i means applying T_i
Addresses indicate the *reverse* order of the transformations required to land in that box.

Example

- Map composition: $(T_2 \circ T_1 \circ T_4)(X)$
- Address: 214
- State transformation: $4 \to 1 \to 2$

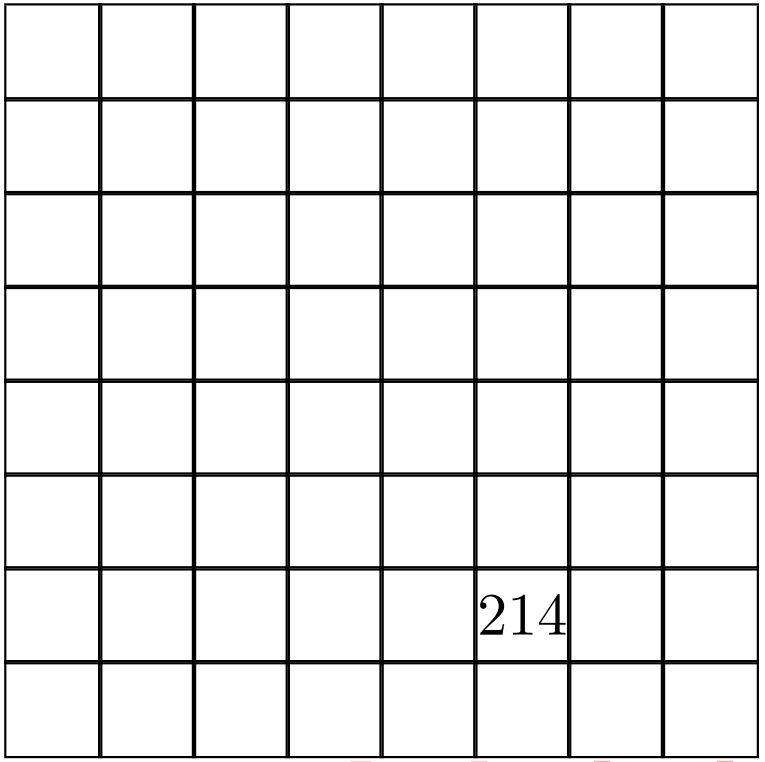


Standard context

Being **in state** i means applying T_i
Addresses indicate the *reverse* order of the transformations required to land in that box.

Example

- Map composition: $(T_2 \circ T_1 \circ T_4)(X)$
- Address: 214
- State transformation: $4 \to 1 \to 2$



Forbidden pairs: 1-IFS

Main idea: Restrict the constructive approach by disallowing certain pairs of transformations from occurring consecutively.

The system has "1 level of memory" because it looks at the currently-applied transformation in the construction to determine which transformations can be applied next.

Where you are (but not how you got there) affects where you are allowed to go.

Forbidden pairs: 1-IFS

Questions:

- What types of attractors does this yield?
- How does forbidding multiple pairs affect the attractors?
- Can we look at an attractor and determine which pairs were forbidden?
- Which attractors can be realized as a *standard* IFS (with "0 levels of memory") by redefining the set of contraction maps?
- Which attractors can be realized as a standard 0-IFS but require *infinitely* many contraction maps?
- Which attractors cannot be realized as a standard 0-IFS?

Forbidden pairs: 1-IFS

Questions:

- What types of attractors does this yield?
- How does forbidding multiple pairs affect the attractors?
- Can we look at an attractor and determine which pairs were forbidden?
- Which attractors can be realized as a *standard* IFS (with "0 levels of memory") by redefining the set of contraction maps?
- Which attractors can be realized as a standard 0-IFS but require *infinitely* many contraction maps?
- Which attractors cannot be realized as a standard 0-IFS?

Forbidden pairs: 1-IFS

Questions:

- What types of attractors does this yield?
- How does forbidding multiple pairs affect the attractors?
- Can we look at an attractor and determine which pairs were forbidden?
- Which attractors can be realized as a *standard* IFS (with "0 levels of memory") by redefining the set of contraction maps?
- Which attractors can be realized as a standard 0-IFS but require *infinitely* many contraction maps?
- Which attractors cannot be realized as a standard 0-IFS?

Forbidden pairs: 1-IFS

Questions:

- What types of attractors does this yield?
- How does forbidding multiple pairs affect the attractors?
- Can we look at an attractor and determine which pairs were forbidden?
- Which attractors can be realized as a *standard* IFS (with "0 levels of memory") by redefining the set of contraction maps?
- Which attractors can be realized as a standard 0-IFS but require *infinitely* many contraction maps?
- Which attractors cannot be realized as a standard 0-IFS?

Forbidden pairs: 1-IFS

Questions:

- What types of attractors does this yield?
- How does forbidding multiple pairs affect the attractors?
- Can we look at an attractor and determine which pairs were forbidden?
- Which attractors can be realized as a *standard* IFS (with "0 levels of memory") by redefining the set of contraction maps?
- Which attractors can be realized as a standard 0-IFS but require *infinitely* many contraction maps?
- Which attractors cannot be realized as a standard 0-IFS?

Forbidden pairs: 1-IFS

Questions:

- What types of attractors does this yield?
- How does forbidding multiple pairs affect the attractors?
- Can we look at an attractor and determine which pairs were forbidden?
- Which attractors can be realized as a *standard* IFS (with "0 levels of memory") by redefining the set of contraction maps?
- Which attractors can be realized as a standard 0-IFS but require *infinitely* many contraction maps?
- Which attractors cannot be realized as a standard 0-IFS?

Forbidden pairs: 1-IFS

Example

Cannot go from state 1 to state 4

Forbidden pairs: 1-IFS

Example

Cannot go from state 1 to state 4

$\iff$ T_4 cannot follow T_1

Forbidden pairs: 1-IFS

Example

Cannot go from state 1 to state 4

$$\iff T_4 \text{ cannot follow } T_1$$
$$\iff (T_4 \circ T_1)(A) = \varnothing$$

Forbidden pairs: 1-IFS

Example

Cannot go from state 1 to state 4

$\iff$ T_4 cannot follow T_1

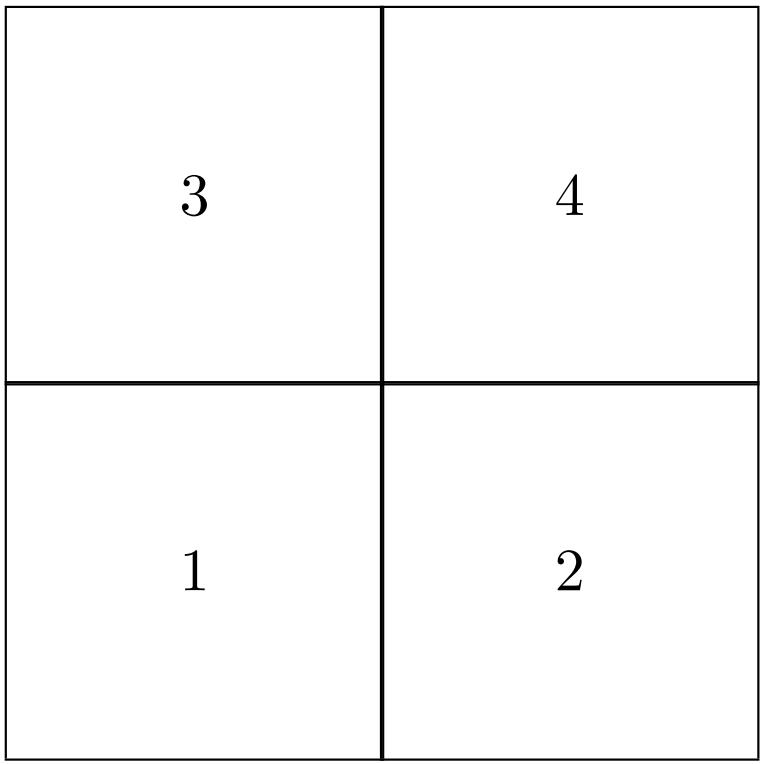
$\iff$ $(T_4 \circ T_1)(A) = \varnothing$

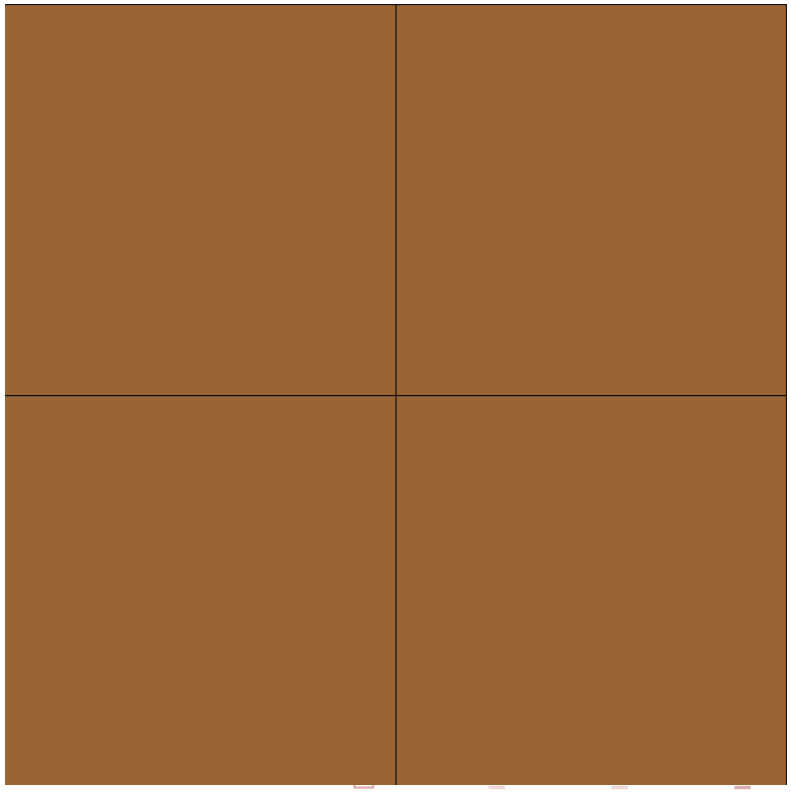
$\iff$ Any address with 41 as a substring is empty

Forbidden pairs: 1-IFS

Example

Any address with 41 as a substring is empty



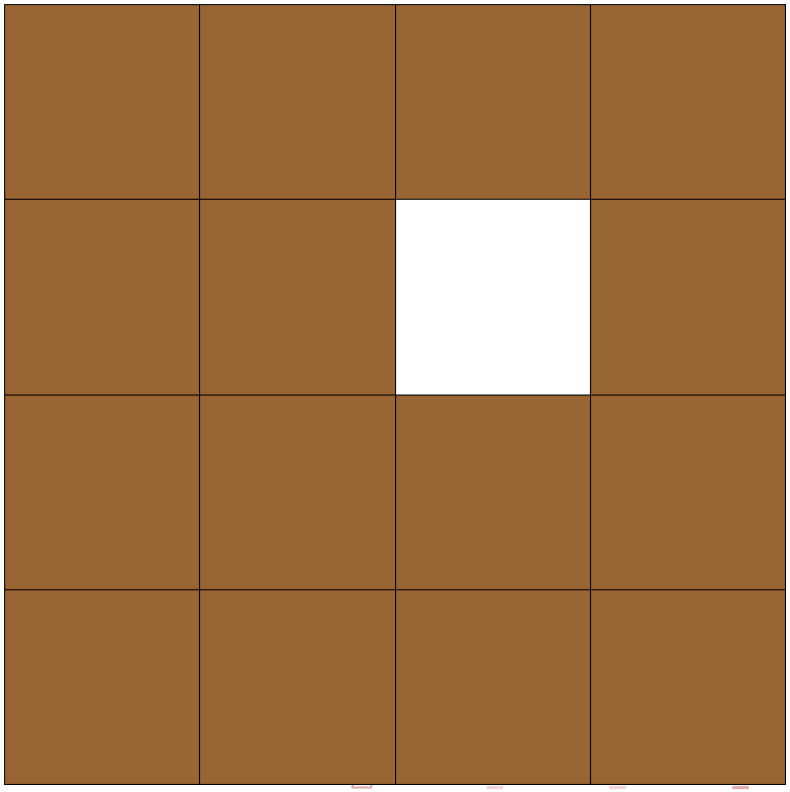


Forbidden pairs: 1-IFS

Example

Any address with 41 as a substring is empty

33	34	43	44
31	32		42
13	14	23	24
11	12	21	22

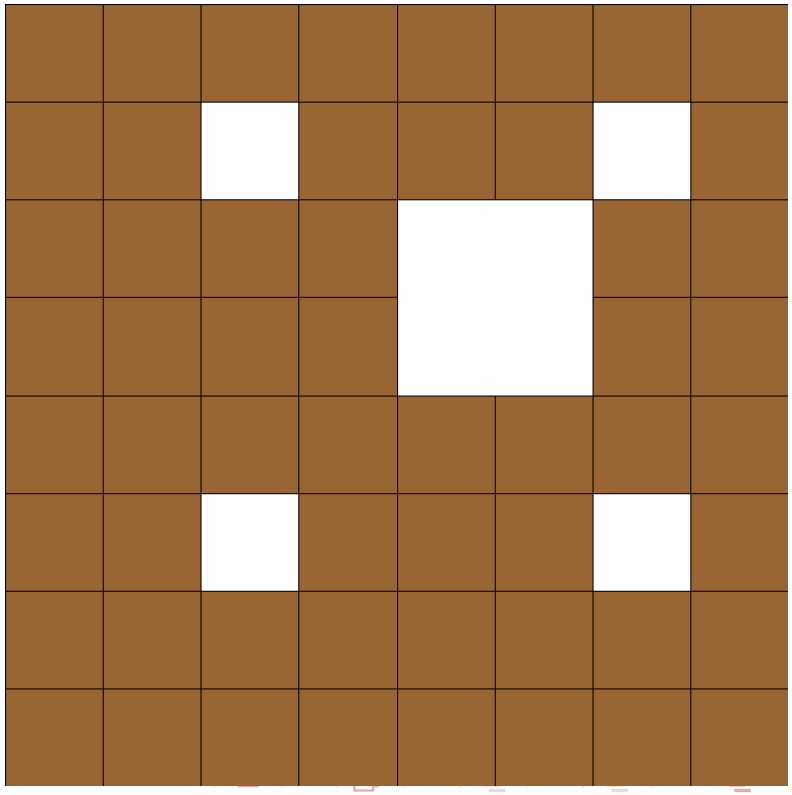


Forbidden pairs: 1-IFS

Example

Any address with 41 as a substring is empty

333	334	343	344	433	434	443	444
331	332		342	431	432		442
313	314	323	324			423	424
311	312	321	322			421	422
133	134	143	144	233	234	243	244
131	132		142	231	232		242
113	114	123	124	213	214	223	224
111	112	121	122	211	212	221	222

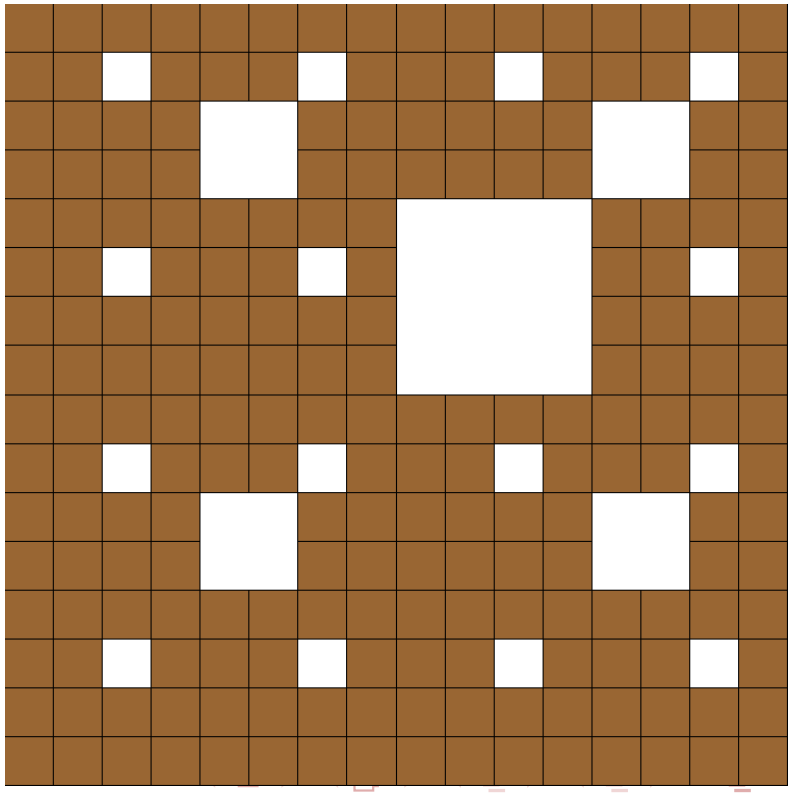


Forbidden pairs: 1-IFS

Example

Any address with 41 as a substring is empty

333	334	343	344	433	434	443	444
331	332		342	431	432		442
313	314	323	324			423	424
311	312	321	322			421	422
133	134	143	144	233	234	243	244
131	132		142	231	232		242
113	114	123	124	213	214	223	224
111	112	121	122	211	212	221	222

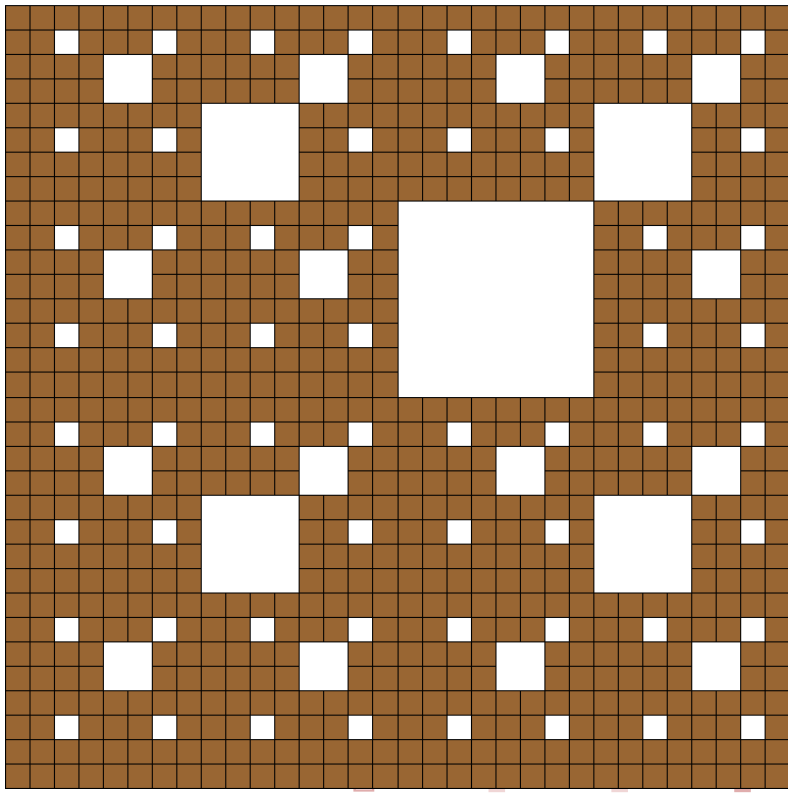


Forbidden pairs: 1-IFS

Example

Any address with 41 as a substring is empty

333	334	343	344	433	434	443	444
331	332		342	431	432		442
313	314	323	324			423	424
311	312	321	322			421	422
133	134	143	144	233	234	243	244
131	132		142	231	232		242
113	114	123	124	213	214	223	224
111	112	121	122	211	212	221	222

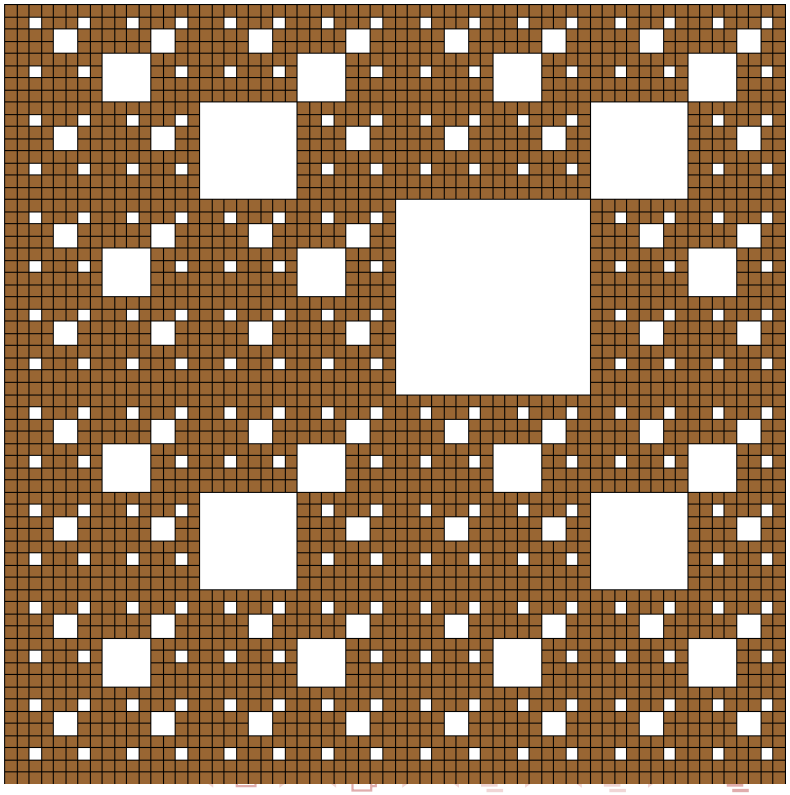


Forbidden pairs: 1-IFS

Example

Any address with 41 as a substring is empty

333	334	343	344	433	434	443	444
331	332		342	431	432		442
313	314	323	324			423	424
311	312	321	322			421	422
133	134	143	144	233	234	243	244
131	132		142	231	232		242
113	114	123	124	213	214	223	224
111	112	121	122	211	212	221	222

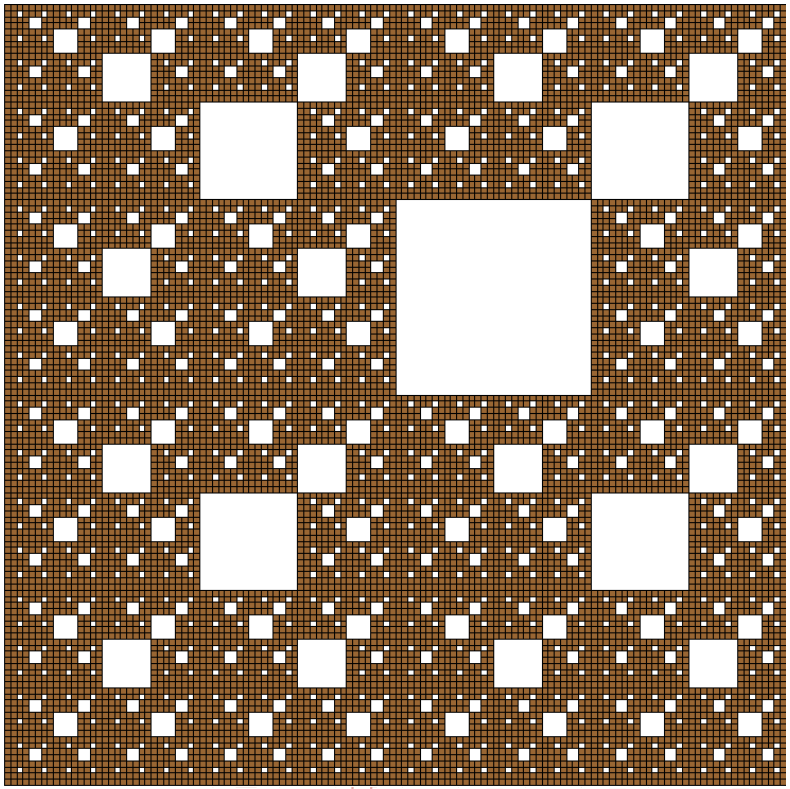


Forbidden pairs: 1-IFS

Example

Any address with 41 as a substring is empty

333	334	343	344	433	434	443	444
331	332		342	431	432		442
313	314	323	324			423	424
311	312	321	322			421	422
133	134	143	144	233	234	243	244
131	132		142	231	232		242
113	114	123	124	213	214	223	224
111	112	121	122	211	212	221	222

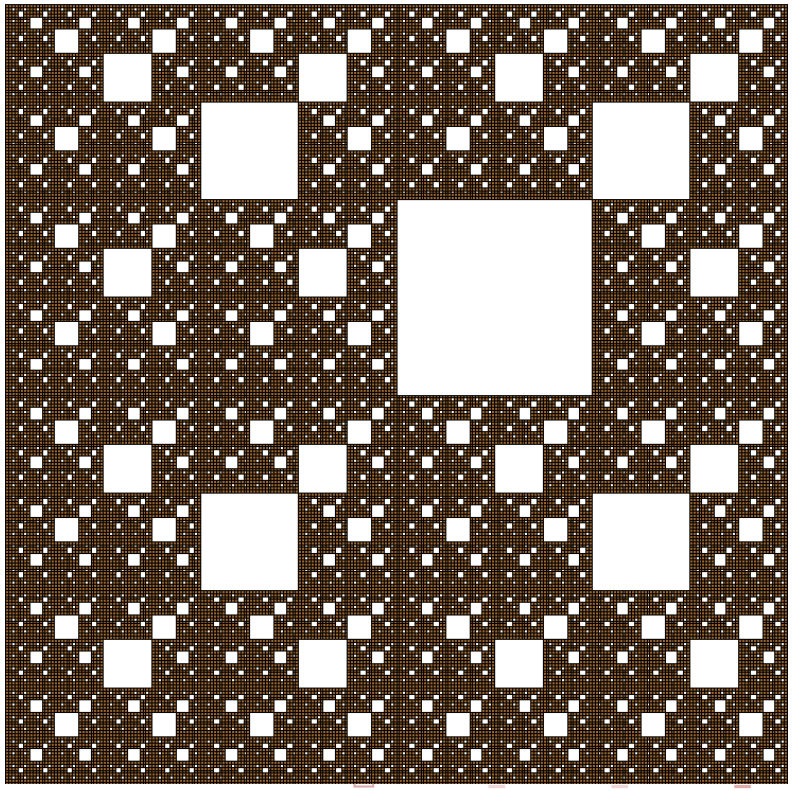


Forbidden pairs: 1-IFS

Example

Any address with 41 as a substring is empty

333	334	343	344	433	434	443	444
331	332		342	431	432		442
313	314	323	324			423	424
311	312	321	322			421	422
133	134	143	144	233	234	243	244
131	132		142	231	232		242
113	114	123	124	213	214	223	224
111	112	121	122	211	212	221	222

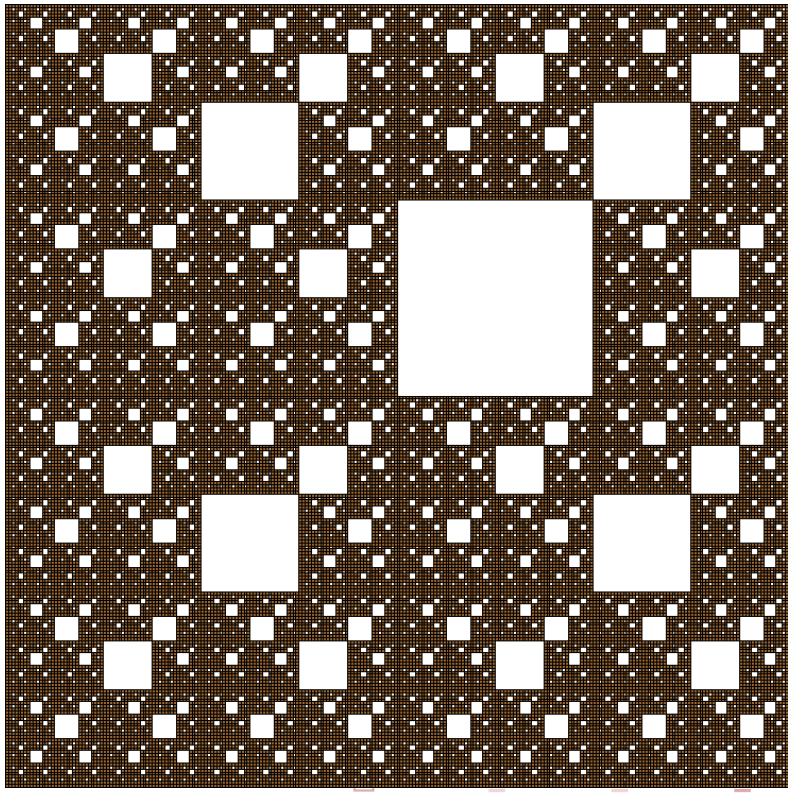


Forbidden pairs: 1-IFS

Example

Any address with 41 as a substring is empty

333	334	343	344	433	434	443	444
331	332		342	431	432		442
313	314	323	324			423	424
311	312	321	322			421	422
133	134	143	144	233	234	243	244
131	132		142	231	232		242
113	114	123	124	213	214	223	224
111	112	121	122	211	212	221	222



Forbidden pairs: 1-IFS

Example

Forbid $1 \to 4$, $4 \to 1$, $2 \to 3$, $3 \to 2$

Forbidden pairs: 1-IFS

Example

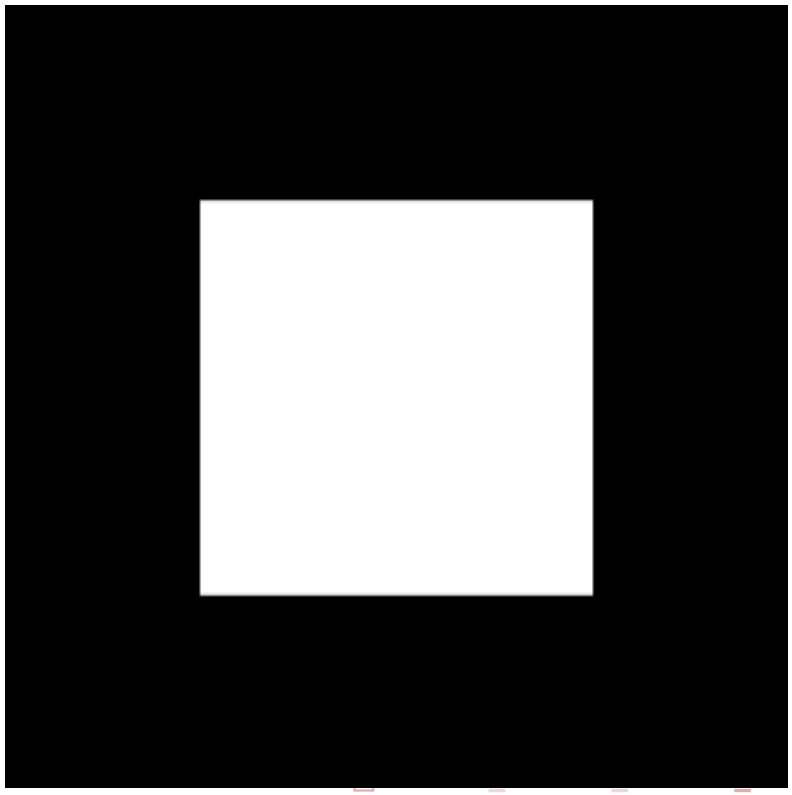
Addresses with 14, 41, 23, 32 are empty

Forbidden pairs: 1-IFS

Example

Addresses with 14, 41, 23, 32 are empty

33	34	43	44
31			42
13			24
11	12	21	22

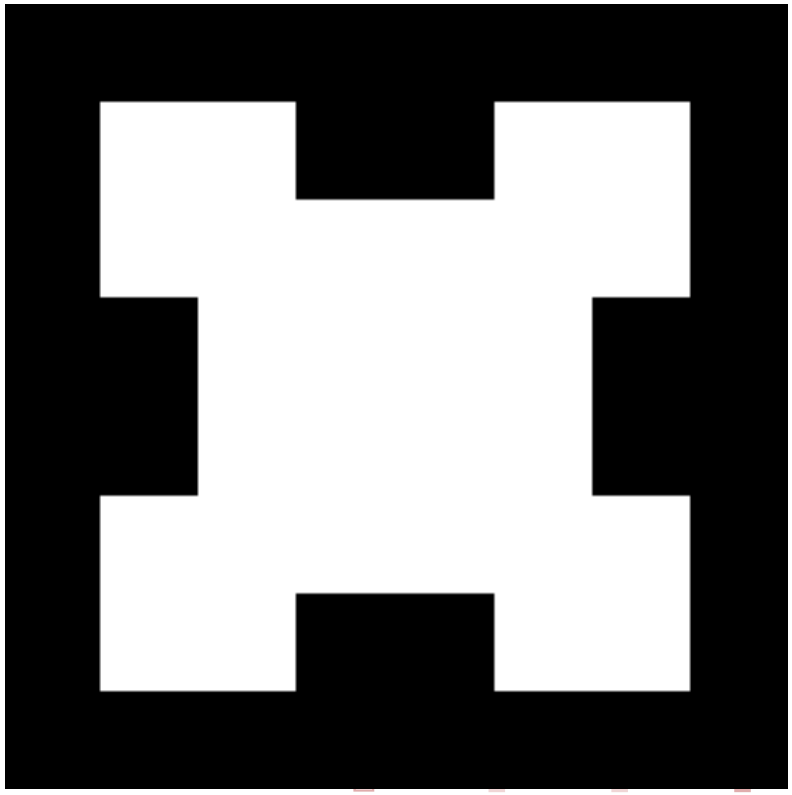


Forbidden pairs: 1-IFS

Example

Addresses with 14, 41, 23, 32 are empty

333	334	343	344	433	434	443	444
331			342	431			442
313							424
311	312					421	422
133	134					243	244
131							242
113			124	213			224
111	112	121	122	211	212	221	222

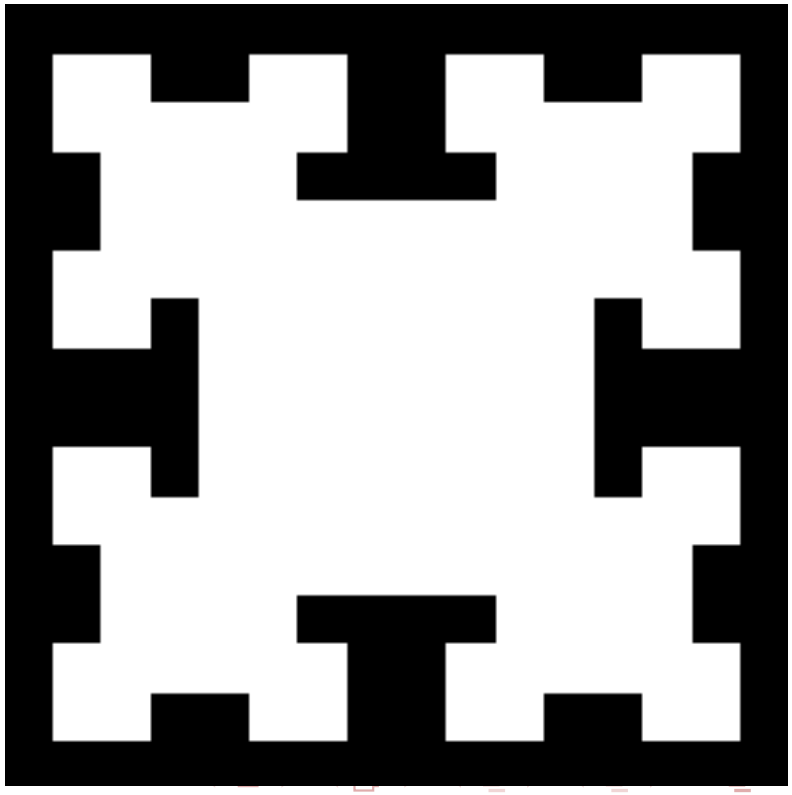


Forbidden pairs: 1-IFS

Example

Addresses with 14, 41, 23, 32 are empty

333	334	343	344	433	434	443	444
331			342	431			442
313							424
311	312					421	422
133	134					243	244
131							242
113			124	213			224
111	112	121	122	211	212	221	222

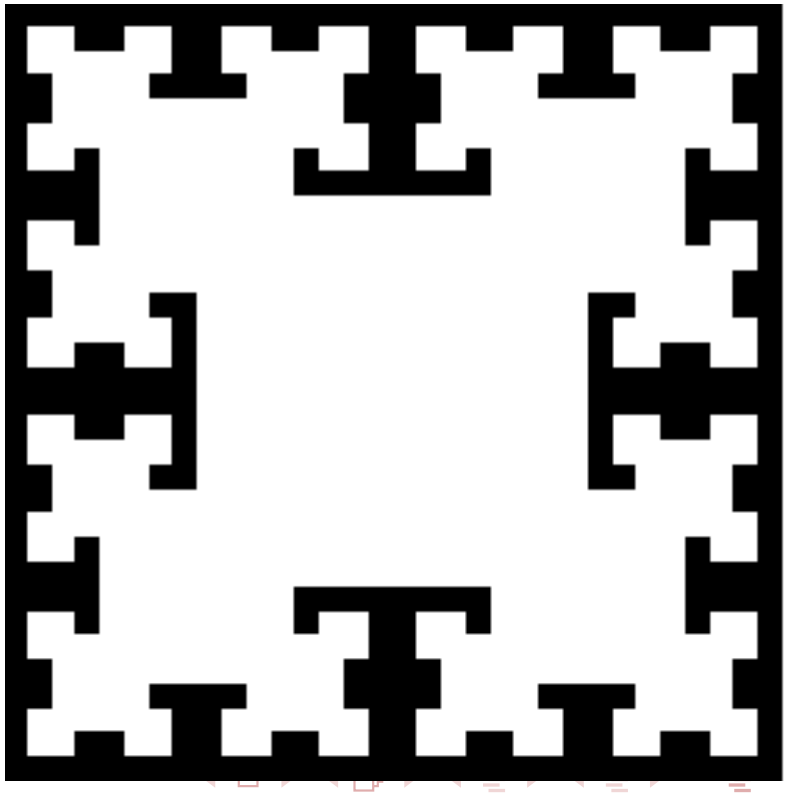


Forbidden pairs: 1-IFS

Example

Addresses with 14, 41, 23, 32 are empty

333	334	343	344	433	434	443	444
331			342	431			442
313							424
311	312					421	422
133	134					243	244
131							242
113			124	213			224
111	112	121	122	211	212	221	222

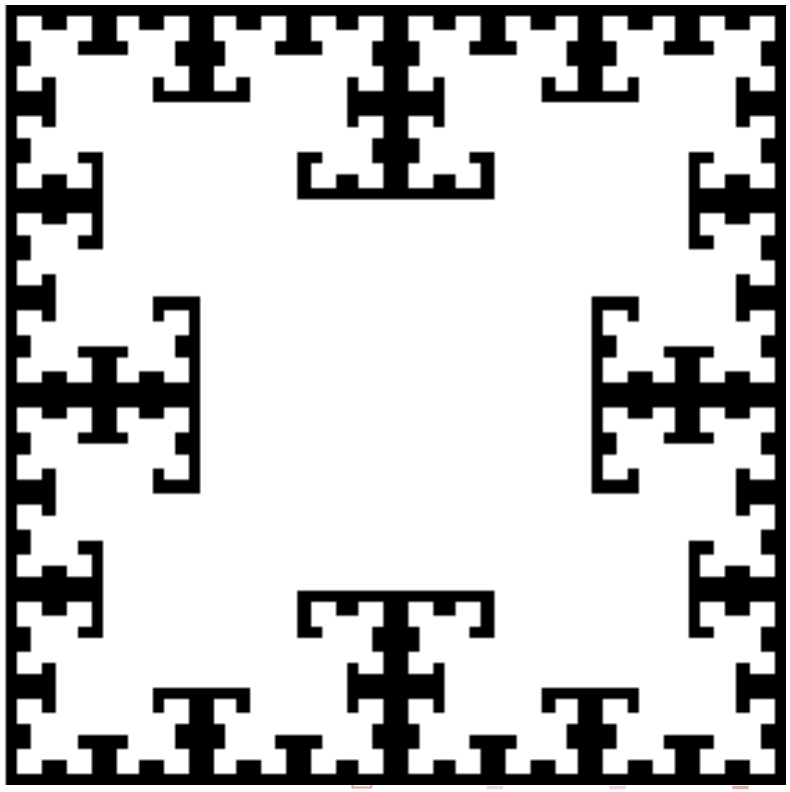


Forbidden pairs: 1-IFS

Example

Addresses with 14, 41, 23, 32 are empty

333	334	343	344	433	434	443	444
331			342	431			442
313							424
311	312					421	422
133	134					243	244
131							242
113			124	213			224
111	112	121	122	211	212	221	222

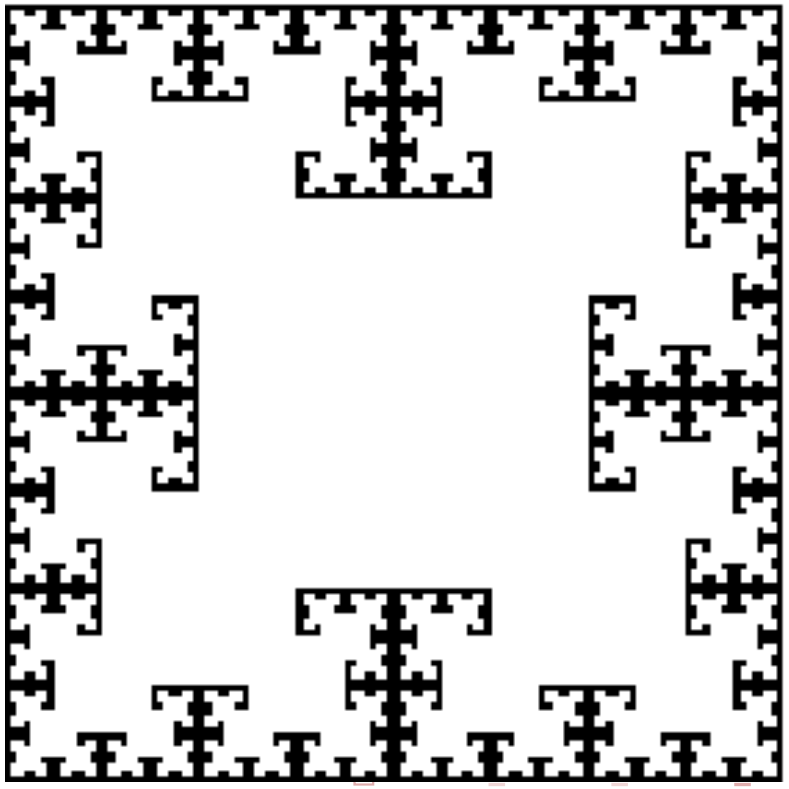


Forbidden pairs: 1-IFS

Example

Addresses with 14, 41, 23, 32 are empty

333	334	343	344	433	434	443	444
331			342	431			442
313							424
311	312					421	422
133	134					243	244
131							242
113			124	213			224
111	112	121	122	211	212	221	222

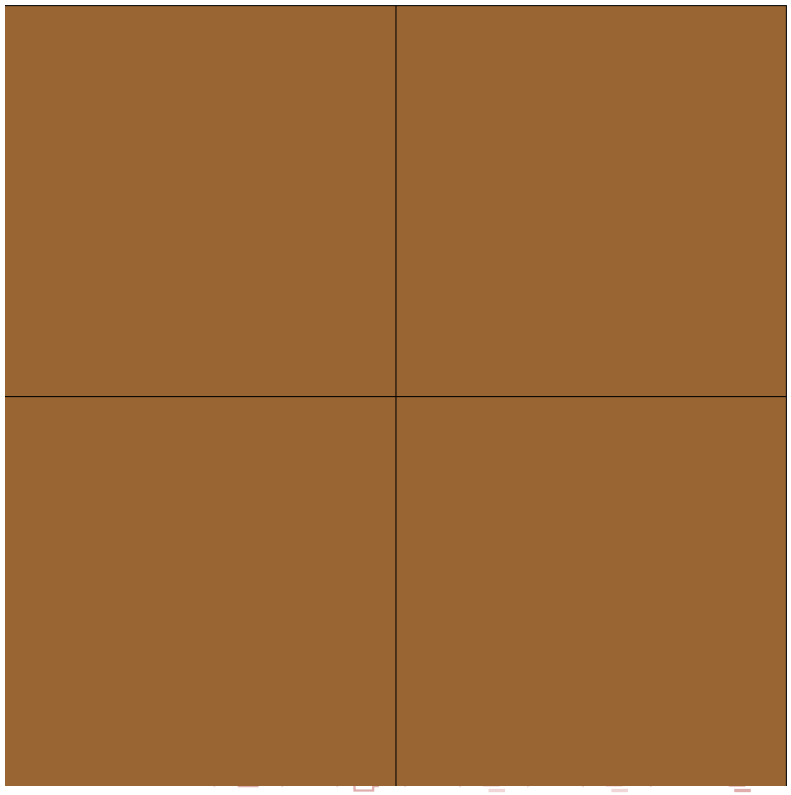


Forbidden pairs: 1-IFS

Example

Addresses with 22, 23, 33 are empty

		343	344		434	443	444
		341	342	431	432	441	442
313	314		324	413	414		424
311	312	321		411	412	421	
	134	143	144			243	244
131	132	141	142			241	242
113	114		124	213	214		
111	112	121		211	212		

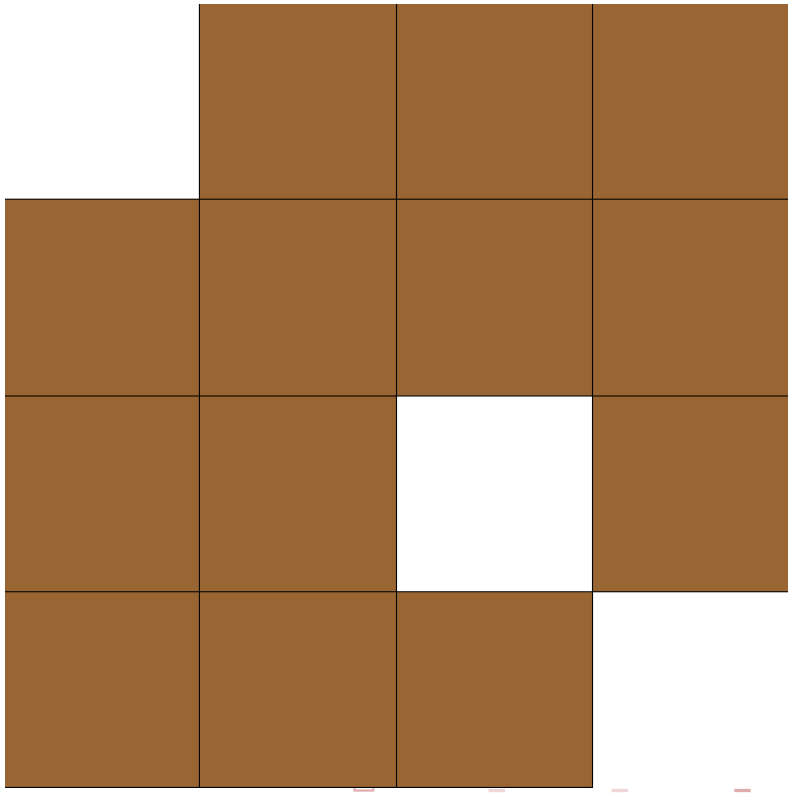


Forbidden pairs: 1-IFS

Example

Addresses with 22, 23, 33 are empty

		343	344		434	443	444
		341	342	431	432	441	442
313	314		324	413	414		424
311	312	321		411	412	421	
	134	143	144			243	244
131	132	141	142			241	242
113	114		124	213	214		
111	112	121		211	212		

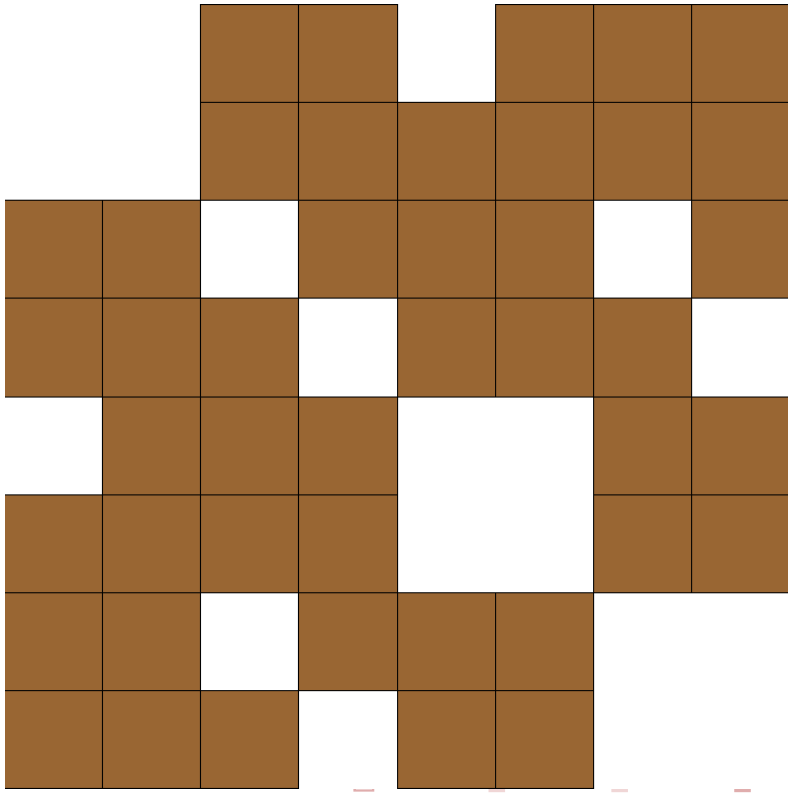


Forbidden pairs: 1-IFS

Example

Addresses with 22, 23, 33 are empty

		343	344		434	443	444
		341	342	431	432	441	442
313	314		324	413	414		424
311	312	321		411	412	421	
	134	143	144			243	244
131	132	141	142			241	242
113	114		124	213	214		
111	112	121		211	212		

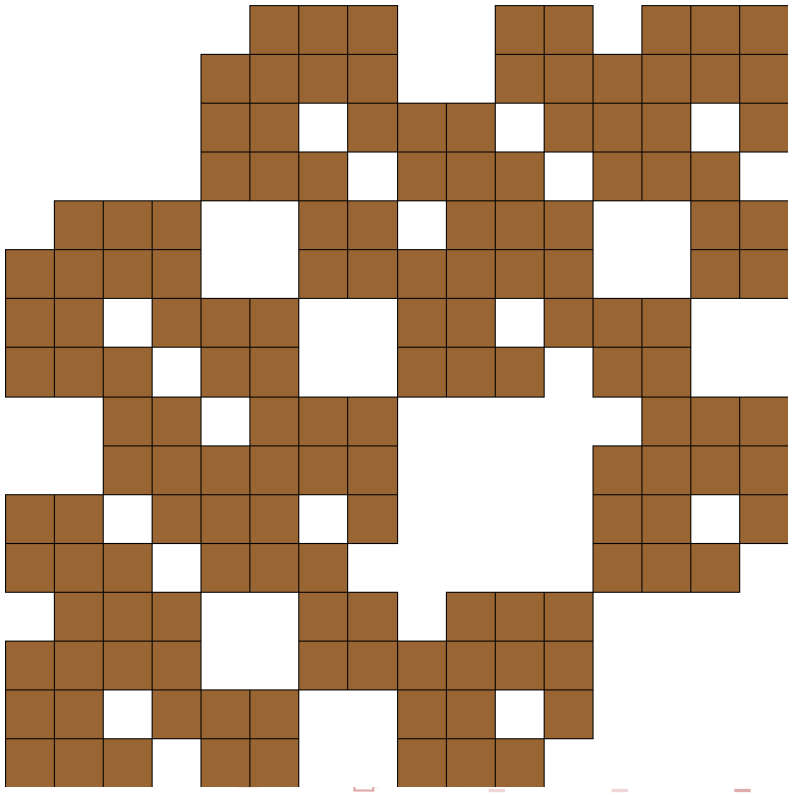


Forbidden pairs: 1-IFS

Example

Addresses with 22, 23, 33 are empty

		343	344		434	443	444
		341	342	431	432	441	442
313	314		324	413	414		424
311	312	321		411	412	421	
	134	143	144			243	244
131	132	141	142			241	242
113	114		124	213	214		
111	112	121		211	212		

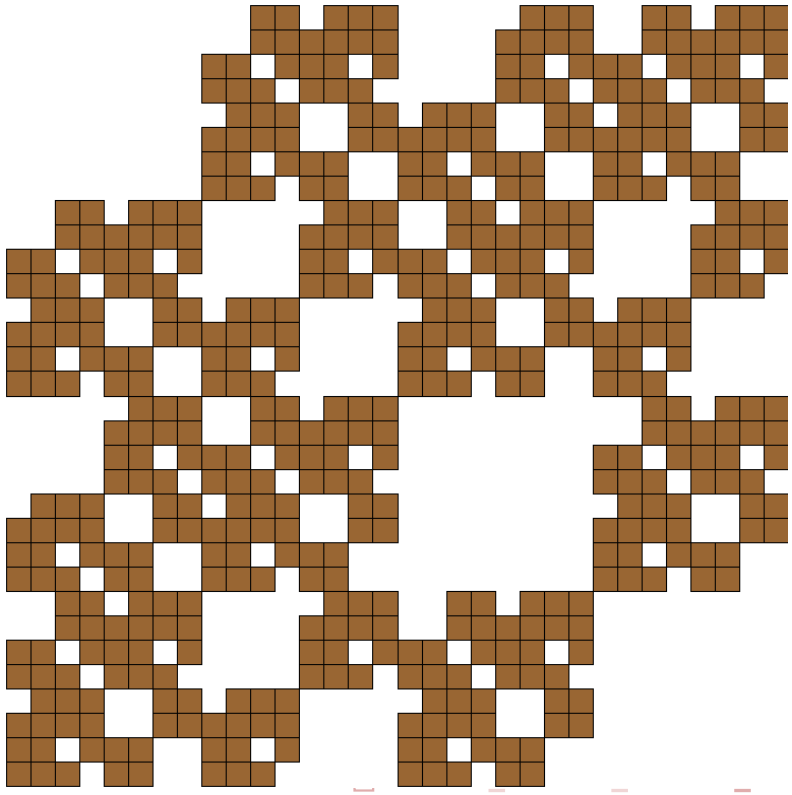


Forbidden pairs: 1-IFS

Example

Addresses with 22, 23, 33 are empty

		343	344		434	443	444
		341	342	431	432	441	442
313	314		324	413	414		424
311	312	321		411	412	421	
	134	143	144			243	244
131	132	141	142			241	242
113	114		124	213	214		
111	112	121		211	212		

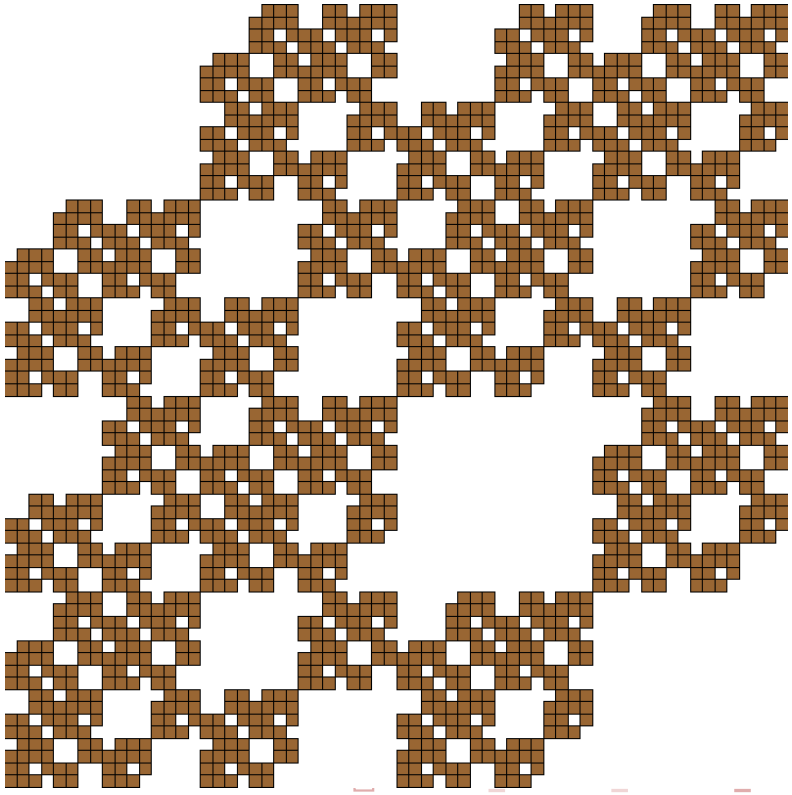


Forbidden pairs: 1-IFS

Example

Addresses with 22, 23, 33 are empty

		343	344		434	443	444
		341	342	431	432	441	442
313	314		324	413	414		424
311	312	321		411	412	421	
	134	143	144			243	244
131	132	141	142			241	242
113	114		124	213	214		
111	112	121		211	212		

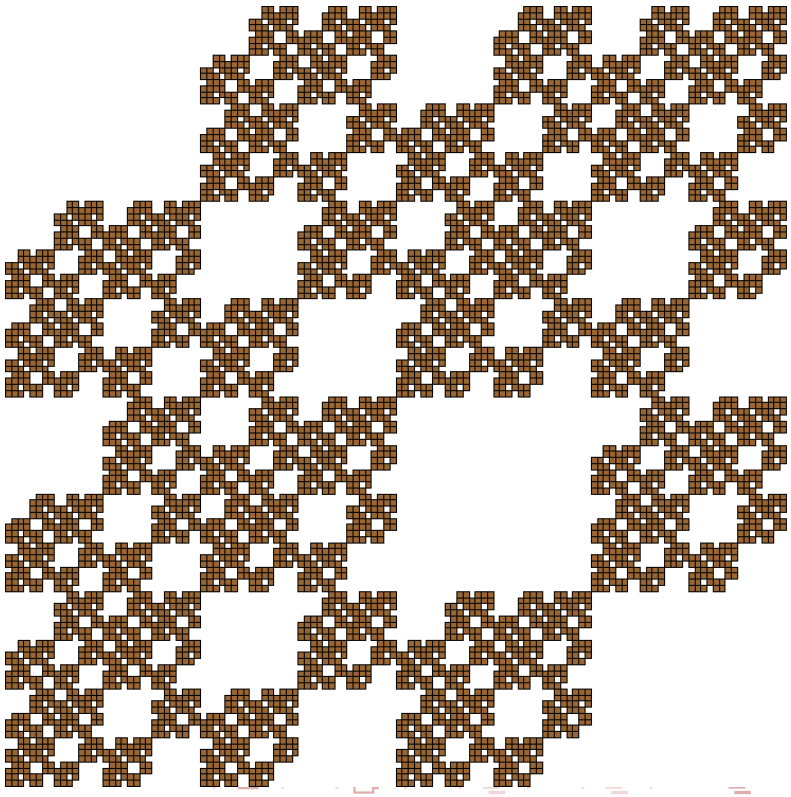


Forbidden pairs: 1-IFS

Example

Addresses with 22, 23, 33 are empty

		343	344		434	443	444
		341	342	431	432	441	442
313	314		324	413	414		424
311	312	321		411	412	421	
	134	143	144			243	244
131	132	141	142			241	242
113	114		124	213	214		
111	112	121		211	212		

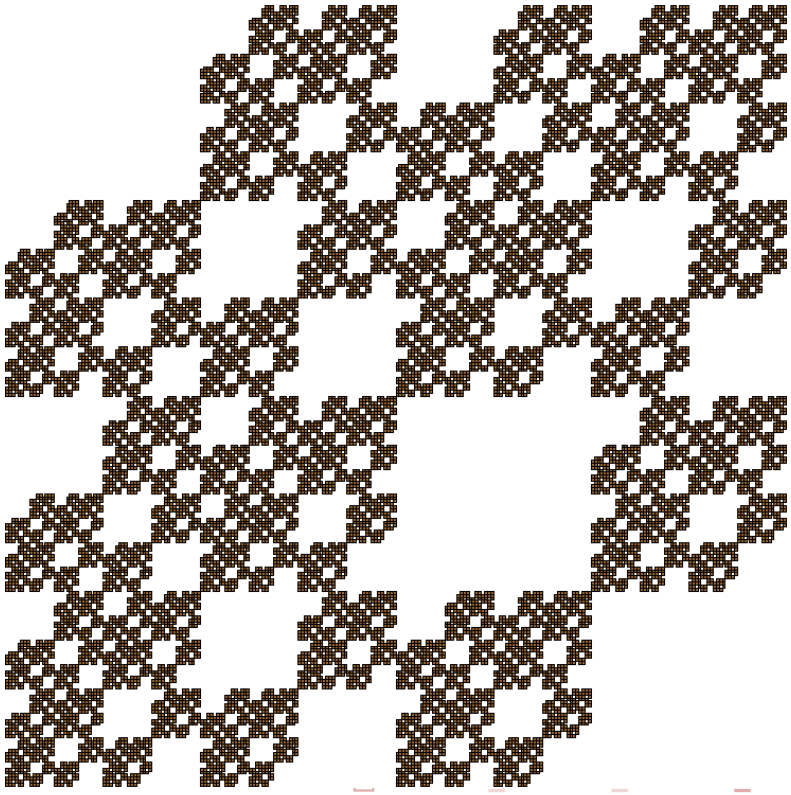


Forbidden pairs: 1-IFS

Example

Addresses with 22, 23, 33 are empty

		343	344		434	443	444
		341	342	431	432	441	442
313	314		324	413	414		424
311	312	321		411	412	421	
	134	143	144			243	244
131	132	141	142			241	242
113	114		124	213	214		
111	112	121		211	212		

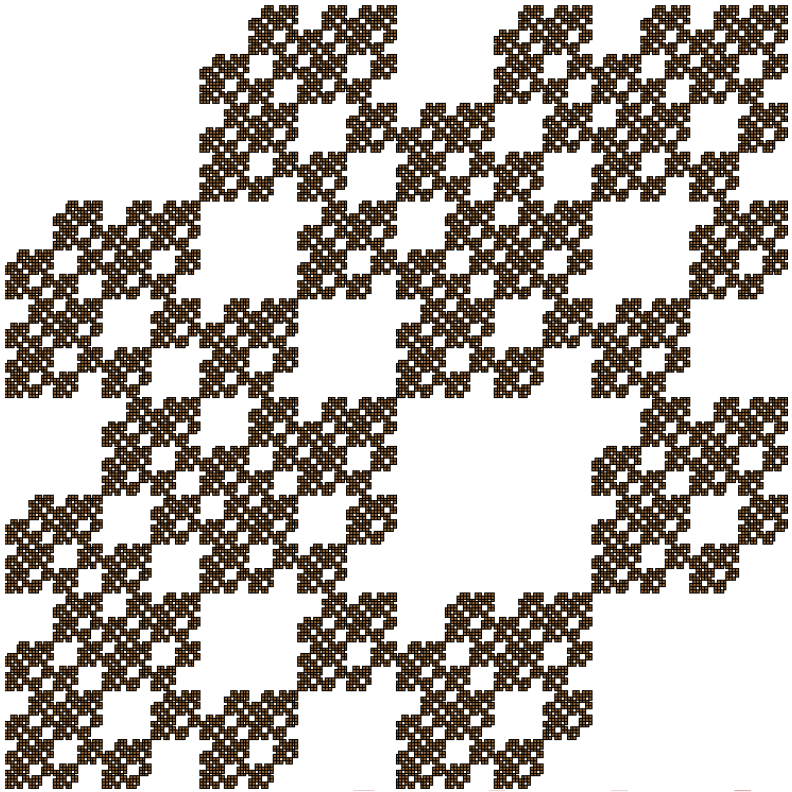


Forbidden pairs: 1-IFS

Example

Addresses with 22, 23, 33 are empty

		343	344		434	443	444
		341	342	431	432	441	442
313	314		324	413	414		424
311	312	321		411	412	421	
	134	143	144			243	244
131	132	141	142			241	242
113	114		124	213	214		
111	112	121		211	212		

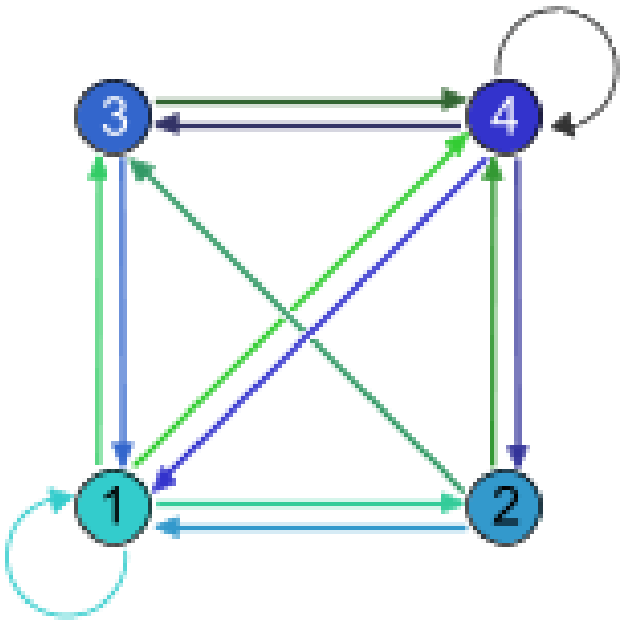


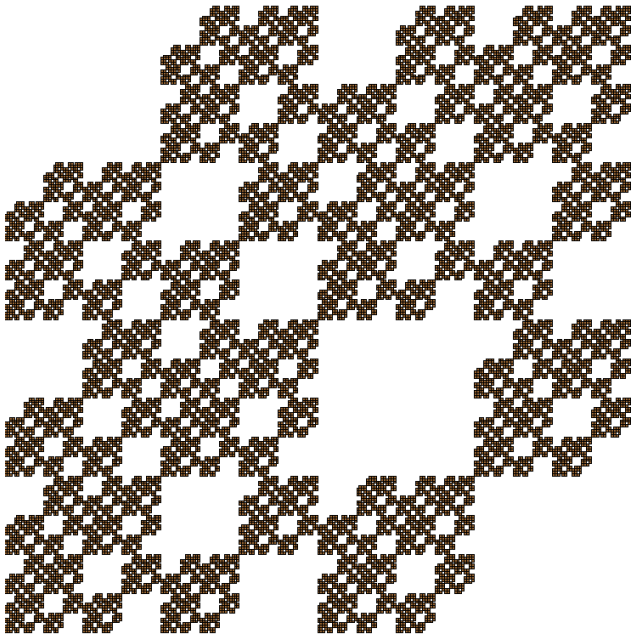
Representing allowed transitions

Vertex set is $\mathcal{T}$.
Directed edge from T_i to T_j if $i \to j$ is allowed.

Representing allowed transitions

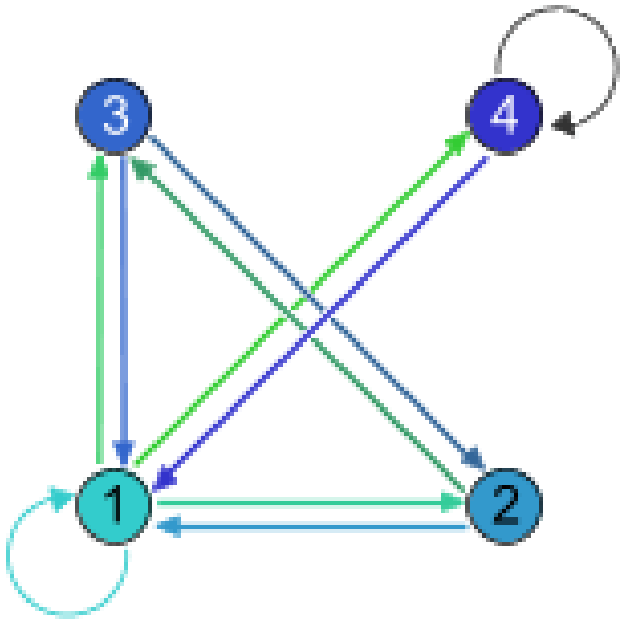
Vertex set is $\mathcal{T}$.
Directed edge from T_i to T_j if $i \to j$ is allowed.

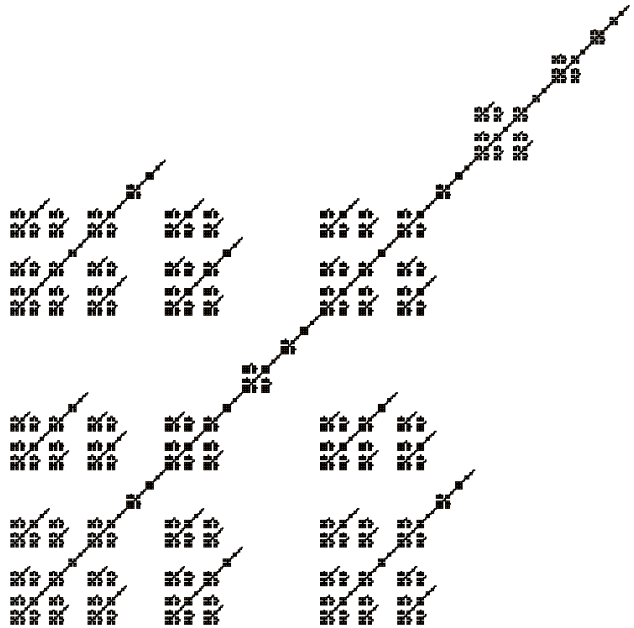




Representing allowed transitions

Vertex set is $\mathcal{T}$.
Directed edge from T_i to T_j if $i \to j$ is allowed.



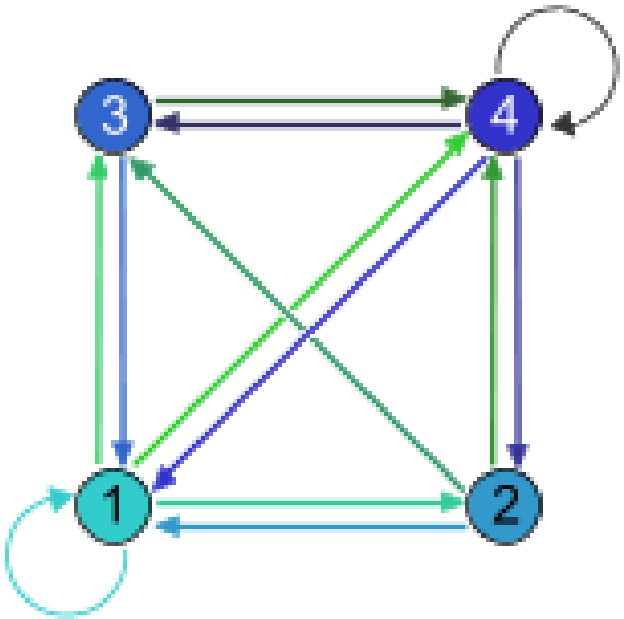


Transition matrices

Represent directed edges by a 0-1 matrix.
Rows/columns indexed by *states*.
$M_{ij} = 1 \iff j \to i$ is allowed.

Transition matrices

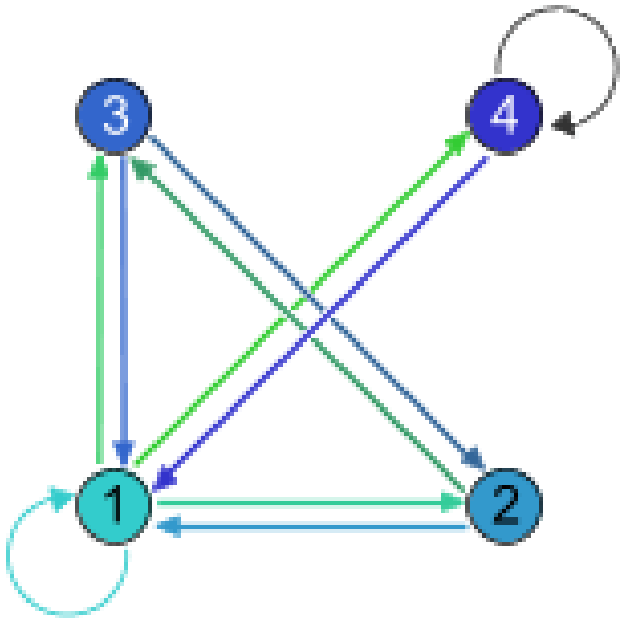
Represent directed edges by a 0-1 matrix.
Rows/columns indexed by *states*.
$M_{ij} = 1 \iff j \to i$ is allowed.



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

Transition matrices

Represent directed edges by a 0-1 matrix.
Rows/columns indexed by *states*.
$M_{ij} = 1 \iff j \to i$ is allowed.



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

Dimensional analysis

Can use transition matrix M to compute Hausdorff dimension of the attractor A.
Let r_j be the contraction factor of T_j.

Dimensional analysis

Can use transition matrix M to compute Hausdorff dimension of the attractor A.
Let r_j be the contraction factor of T_j.

Theorem

The Hausdorff dimension of A is the unique d for which the spectral radius of

$$M(d) = \left[m_{ij} r_j^d \right]_{ij}$$

is exactly 1.

Recall: the **spectral radius** of a matrix M is

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

Terminology

Definition

The attractor of a 1-IFS is . . .

- **IFS-able** *if it can be realized by a 0-IFS.*
- **∞-IFS-able** *if it can be realized by a 0-IFS with countably-many transformations.*
- **non-IFS-able** *if it cannot be realized by any 0-IFS, regardless of how many transformations are used.*

Terminology

Definition

A transformation T_i is called a **full state** *if it can immediately follow any other transformation;*

i.e. $1 \to i$, $2 \to i$, $3 \to i$, $4 \to i$ *are all allowed.*

Terminology

Definition

A transformation T_i is called a **full state** *if it can immediately follow any other transformation;*

i.e. $1 \to i$, $2 \to i$, $3 \to i$, $4 \to i$ *are all allowed.*

Also known as a **Rome**, *because all roads in the transition graph lead to it. (Not to be confused with a* **roam**.*)*

Corresponds to a row of 1s in transition matrix.

Main result: 1-IFS to 0-IFS

1. There exists a Rome.
2. Every transformation has a path to it starting at a Rome.
3. There are infinite sequences of non-Romes.

Main result: 1-IFS to 0-IFS

1. There exists a Rome.
2. Every transformation has a path to it starting at a Rome.
3. There are infinite sequences of non-Romes.

Theorem

The attractor of a 1-IFS is ...

- *IFS-able* $\iff$ *(1) and (2) hold.*
- ∞*-IFS-able* $\iff$ *(1) and (2) and (3) hold.*
- *non-IFS-able* $\iff$ *(1) or (2) fails.*

Main result: 1-IFS to 0-IFS

1. There exists a Rome.
2. Every transformation has a path to it starting at a Rome.
3. There are infinite sequences of non-Romes.

Theorem

The attractor of a 1-IFS is ...

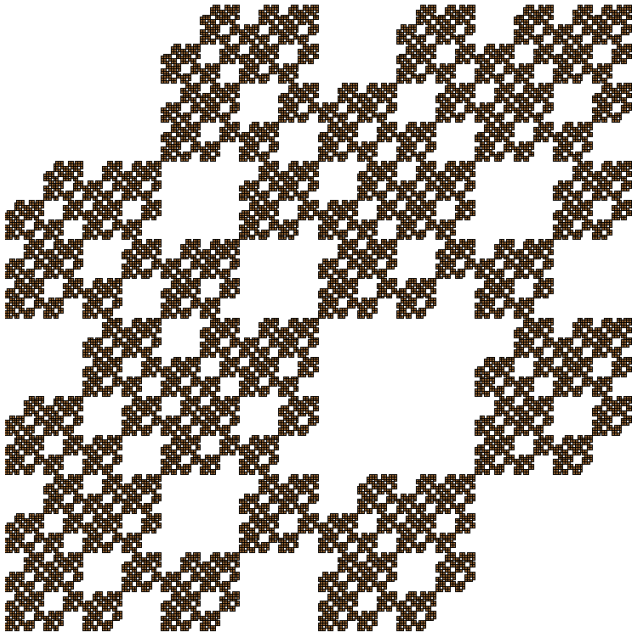
- *IFS-able* $\iff$ *(1) and (2) hold.*
- ∞*-IFS-able* $\iff$ *(1) and (2) and (3) hold.*
- *non-IFS-able* $\iff$ *(1) or (2) fails.*

Proof.

Manipulating strings and addresses ... [13] □

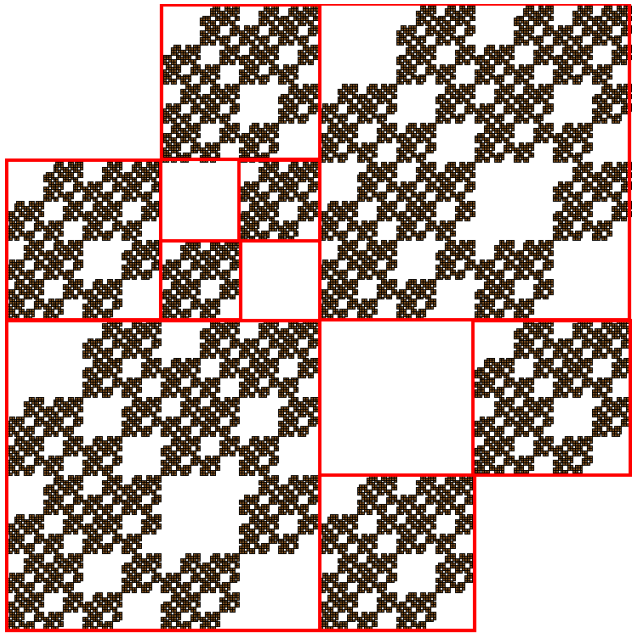
Classifying examples

Main idea: look for scaled copies of the attractor within itself.



Classifying examples

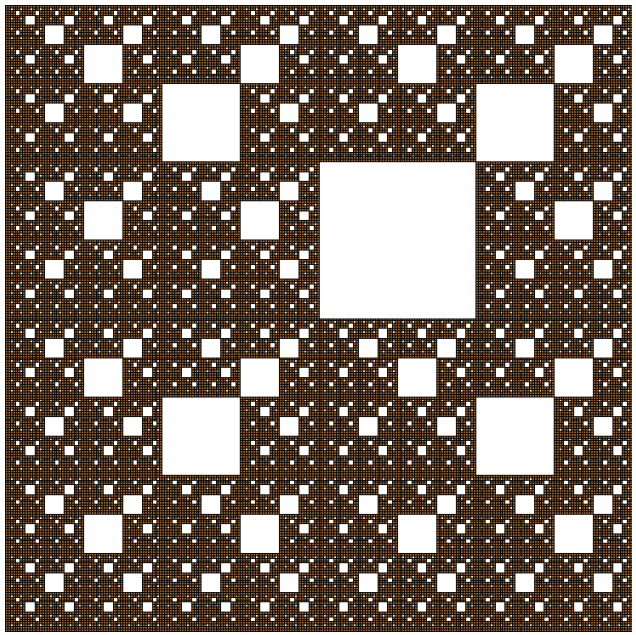
Main idea: look for scaled copies of the attractor within itself.



IFS-able with 8 transformations

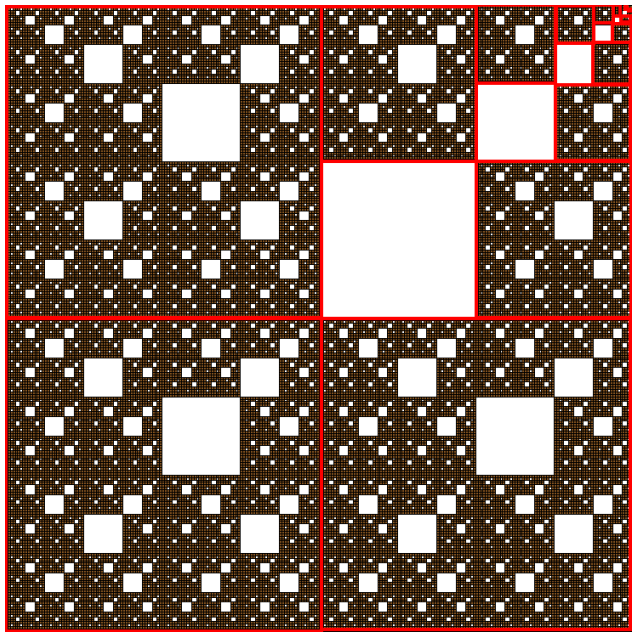
Classifying examples

Main idea: look for scaled copies of the attractor within itself.



Classifying examples

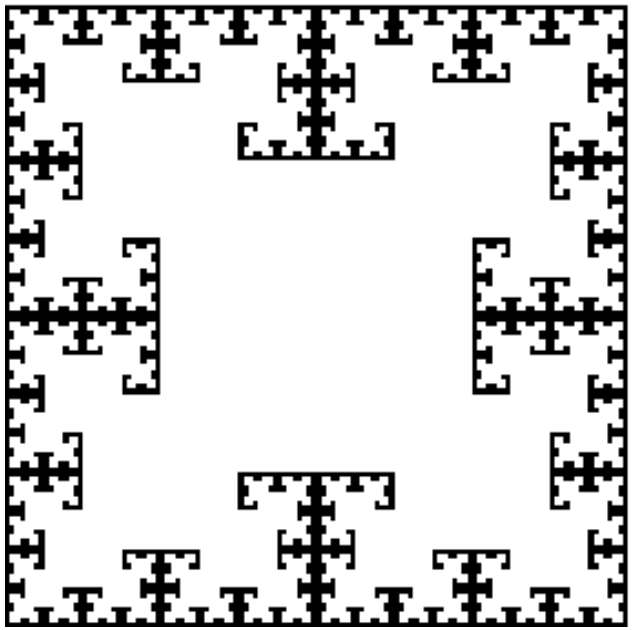
Main idea: look for scaled copies of the attractor within itself.



∞-IFS-able

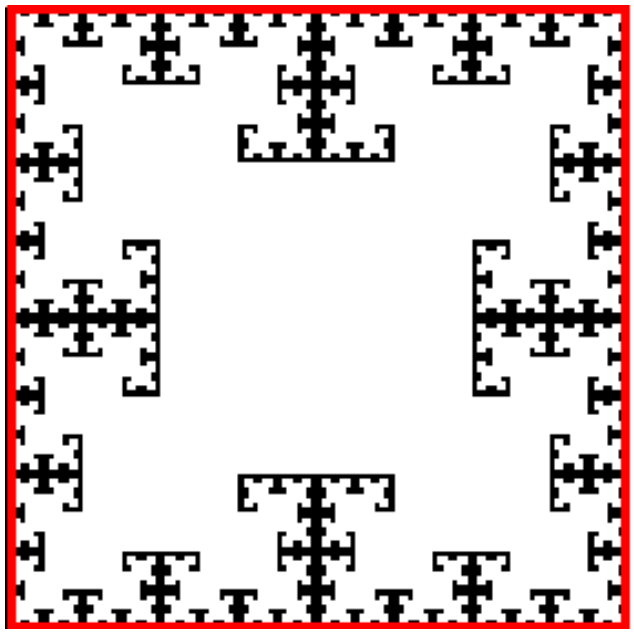
Classifying examples

Main idea: look for scaled copies of the attractor within itself.



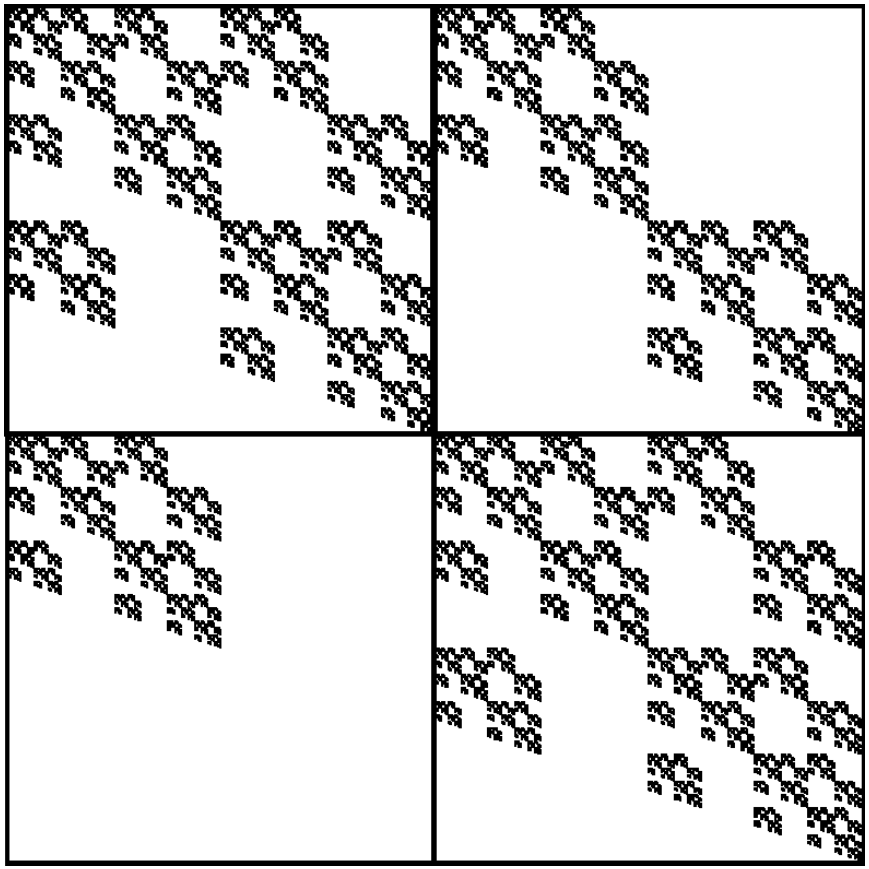
Classifying examples

Main idea: look for scaled copies of the attractor within itself.

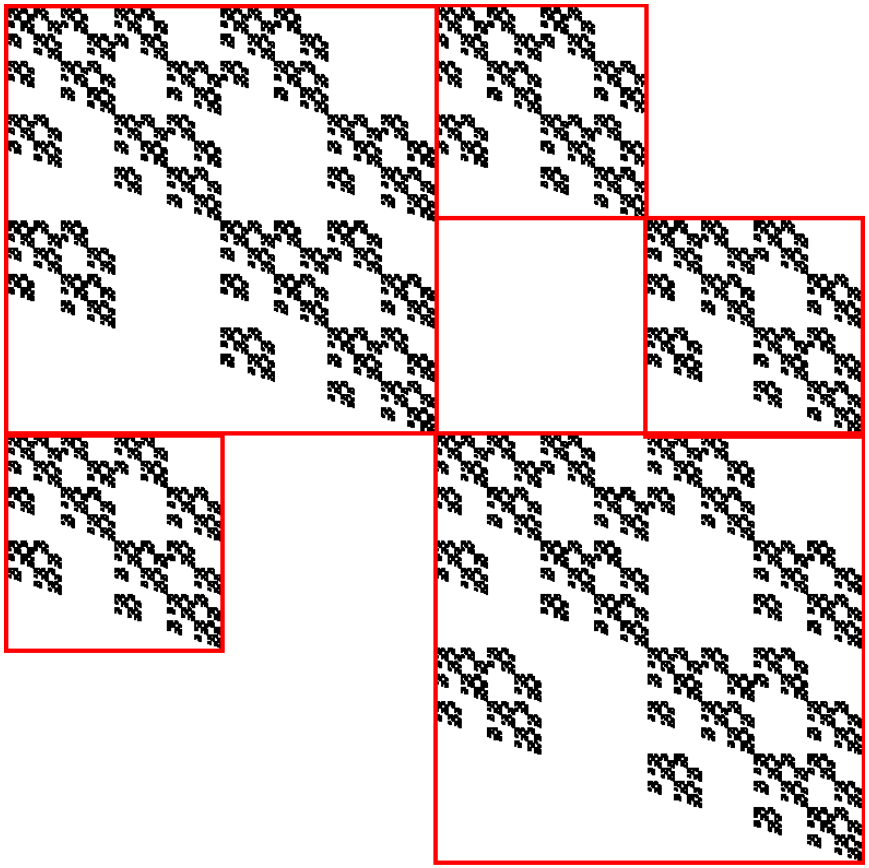


non-IFS-able

Your turn!

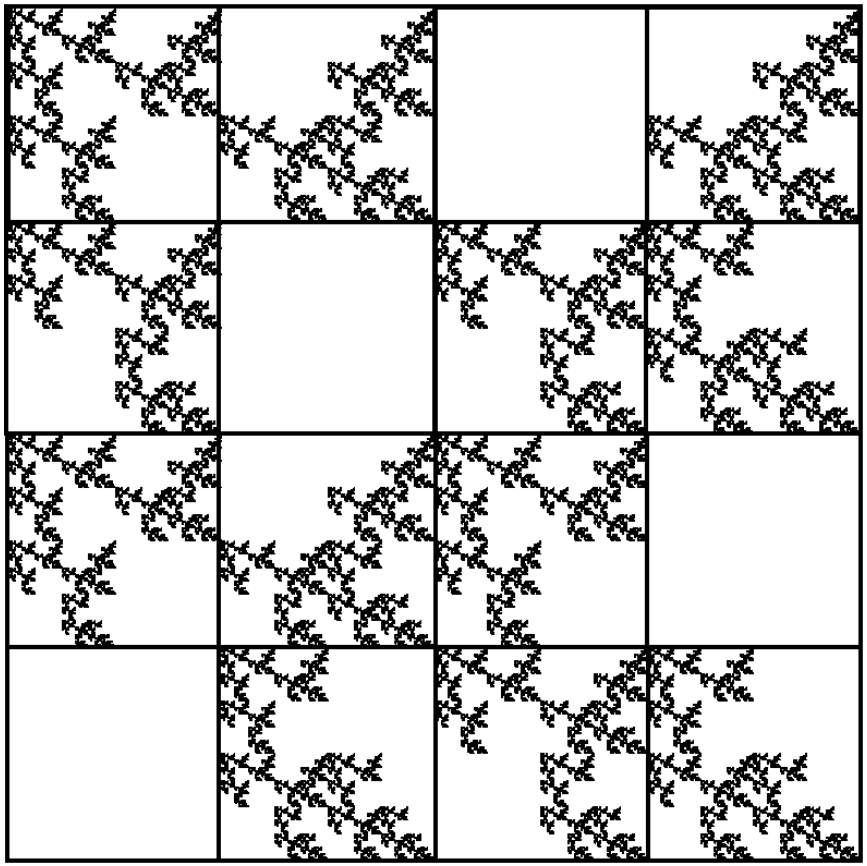


Your turn!

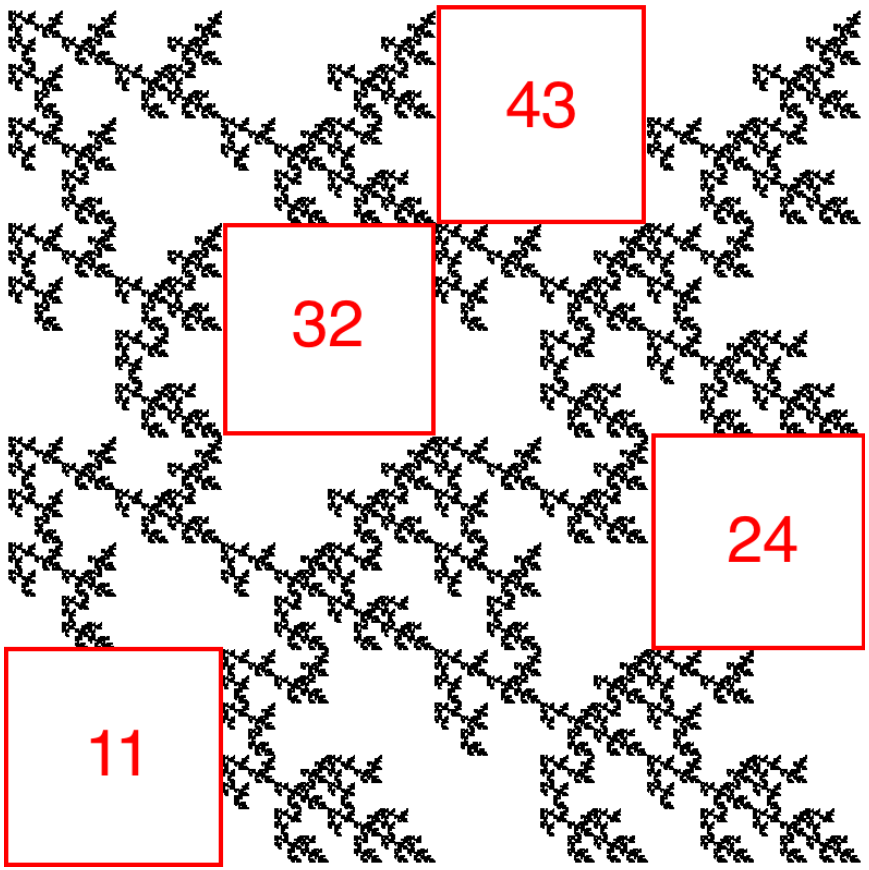


IFS-able with 5 transformations

Your turn!

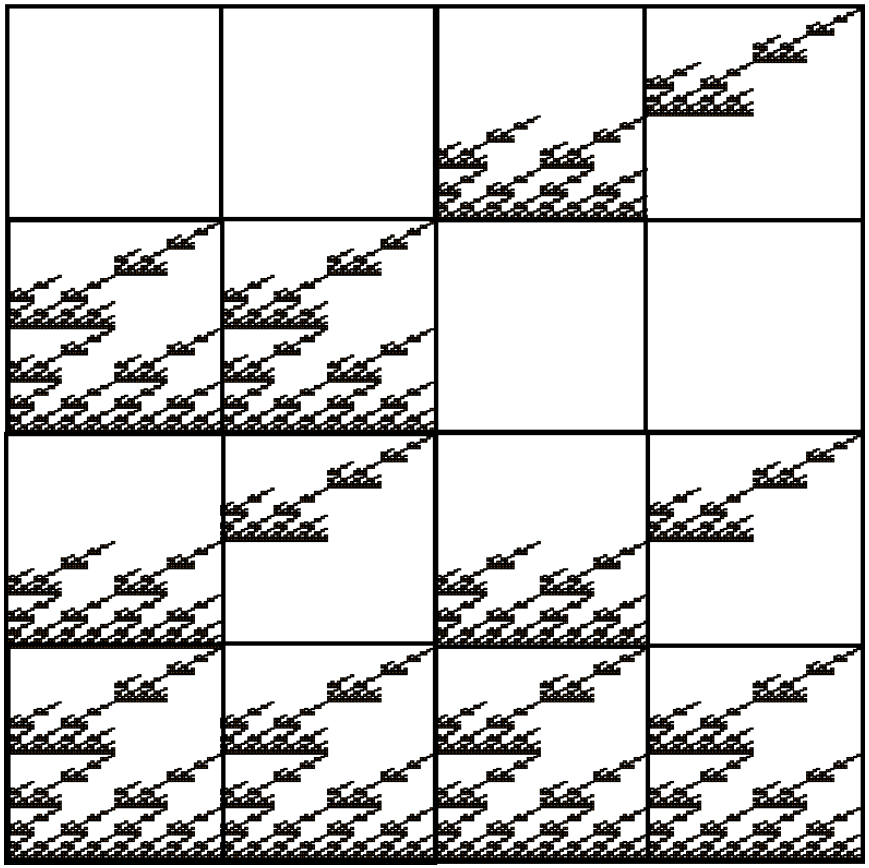


Your turn!

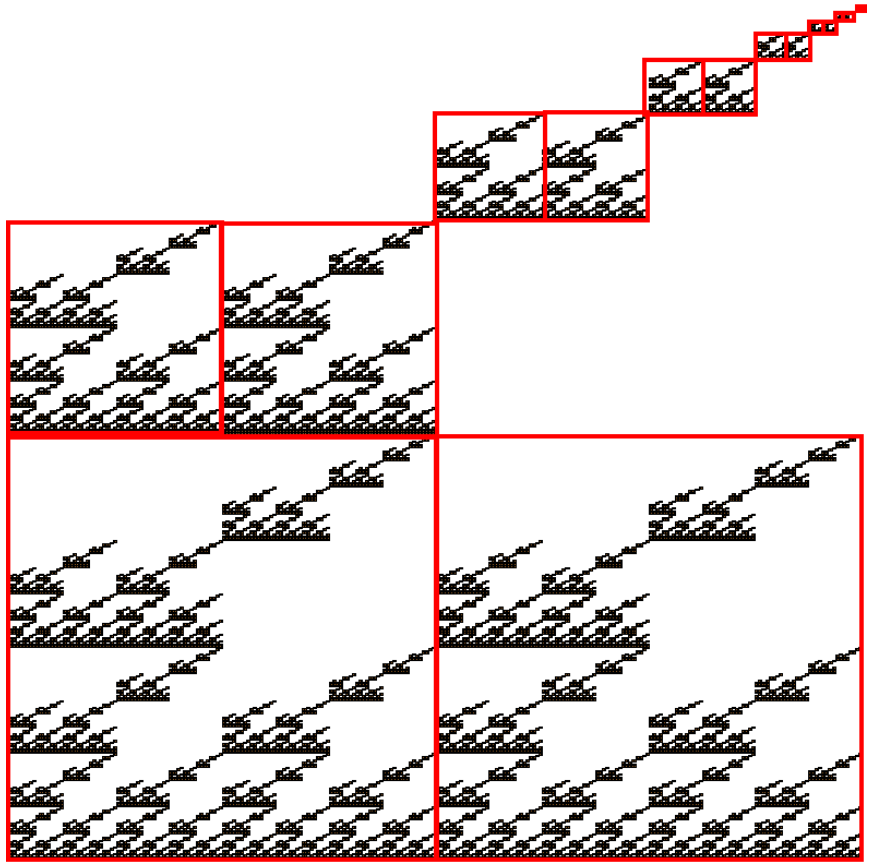


non-IFS-able (no Romes)

Your turn!



Your turn!



∞-IFS-able

Forbidden pairs and triples: 2-IFS

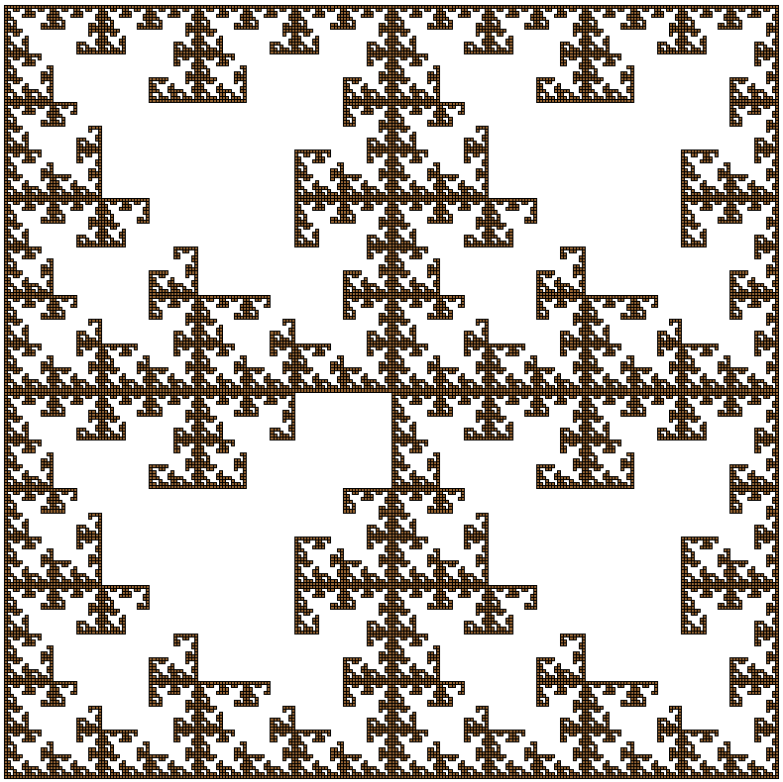
Define IFS by set $\mathcal{F}$ of forbidden strings.

2-IFS means $\mathcal{F}$ contains triples and pairs.

In general, n-IFS means $\mathcal{F}$ contains strings with length at most $n+1$, and contains at least one with exactly that length.

Forbidden pairs and triples: 2-IFS

Working example: $\mathcal{F} = \{14, 23, 32, 441\}$



Subshifts of finite type

Let A be a finite alphabet. Let X be the set of bi-infinite strings from A, called the **full shift**:

$$X = A^{\mathbb{Z}} = \{(\dots x_{-1}.x_0x_1\dots) \mid x_i \in A\}$$

Subshifts of finite type

Let A be a finite alphabet. Let X be the set of bi-infinite strings from A, called the **full shift**:

$$X = A^{\mathbb{Z}} = \{(\dots x_{-1}.x_0x_1\dots) \mid x_i \in A\}$$

Given set of words $\mathcal{F}$ from A, the **shift space determined by** $\mathcal{F}$ is the set of strings from X that contain no element of $\mathcal{F}$ as a substring, written as $X_{\mathcal{F}}$.

If $\mathcal{F}$ is finite, $X_{\mathcal{F}}$ is a **subshift of finite type**.

If the longest string in $\mathcal{F}$ has length $N+1$, say $X_{\mathcal{F}}$ is an **N-step shift**.

Graph vertex shifts

Let G be a directed graph. The **vertex shift** of G has alphabet $A = V$ (vertices of G) and is

$$\hat{X}_G = \{\ldots v_1.v_0v_1\cdots \mid \forall i.\ (v_i, v_{i+1}) \in E\}$$

the set of all infinite paths in G.

Graph vertex shifts

Let G be a directed graph. The **vertex shift** of G has alphabet $A = V$ (vertices of G) and is

$$\hat{X}_G = \{\ldots v_1.v_0v_1\cdots \mid \forall i.\ (v_i, v_{i+1}) \in E\}$$

the set of all infinite paths in G.

Lemma

Graph vertex shifts are 1-step shifts of finite type.

Goal: Exploit graph shifts to reduce n-IFS to 1-IFS.

Higher block shifts

Given $A, \mathcal{F}$, and $X_{\mathcal{F}}$ (an N-step shift).

Let $B_N(X_{\mathcal{F}})$ be the set of allowed strings of length N in $X_{\mathcal{F}}$.

Higher block shifts

Given $A, \mathcal{F}$, and $X_{\mathcal{F}}$ (an N-step shift).

Let $B_N(X_{\mathcal{F}})$ be the set of allowed strings of length N in $X_{\mathcal{F}}$.

Let $\beta_N : X_{\mathcal{F}} \to (B_N(X_{\mathcal{F}}))^{\mathbb{Z}}$ be defined by

$$(\beta_N(\underline{x}))_i = x_i x_{i+1} \dots x_{i+N-1}$$

where $\underline{x} = \dots x_{-1}.x_0 x_1 \dots$.

Higher block shifts

Example

Let $A = \{1, 2, 3, 4\}$, $\mathcal{F} = \{14, 23, 32, 441\}$ (i.e. $N = 2$).

Consider $\underline{x} = \ldots 12443 \ldots$

Higher block shifts

Example

Let $A = \{1, 2, 3, 4\}$, $\mathcal{F} = \{14, 23, 32, 441\}$ (i.e. $N = 2$).

Consider $\underline{x} = \ldots 12443 \ldots$

$$\beta_2(\underline{x}) = \ldots 12$$

Higher block shifts

Example

Let $A = \{1, 2, 3, 4\}$, $\mathcal{F} = \{14, 23, 32, 441\}$ (i.e. $N = 2$).

Consider $\underline{x} = \ldots 12443 \ldots$

$$\beta_2(\underline{x}) = \ldots 1224$$

Higher block shifts

Example

Let $A = \{1, 2, 3, 4\}$, $\mathcal{F} = \{14, 23, 32, 441\}$ (i.e. $N = 2$).

Consider $\underline{x} = \ldots 12443 \ldots$

$$\beta_2(\underline{x}) = \ldots 122444$$

Higher block shifts

Example

Let $A = \{1, 2, 3, 4\}$, $\mathcal{F} = \{14, 23, 32, 441\}$ (i.e. $N = 2$).

Consider $\underline{x} = \ldots 12443 \ldots$

$$\beta_2(\underline{x}) = \ldots 12244443 \ldots$$

Higher block shifts

Definition

The N**-th higher block shift** *is the image*

$$X_{\mathcal{F}}^{[N]} = \beta_N(X_{\mathcal{F}})$$

Higher block shifts

Definition

The N**-th higher block shift** *is the image*

$$X_{\mathcal{F}}^{[N]} = \beta_N(X_{\mathcal{F}})$$

Theorem

If $X_{\mathcal{F}}$ *is an* N*-step shift, then there exists a directed graph* G *such that*

$$X_{\mathcal{F}}^{[N]} = \hat{X}_G$$

i.e. N*-th higher block shift can be realized as a 1-step shift!*

Higher block shifts as graph shifts

Proof is in [14].
Vertices of G are $B_N(X_{\mathcal{F}})$, the allowed N-length strings.
Edge from $a_1 \dots a_N$ to $b_1 \dots b_N$ iff

1. $a_2 \dots a_N = b_1 \dots b_{N-1}$
2. $a_1 a_2 \dots a_N b_N$ is an allowed string in $X_{\mathcal{F}}$

Higher block shifts as graph shifts

Proof is in [14].
Vertices of G are $B_N(X_{\mathcal{F}})$, the allowed N-length strings.
Edge from $a_1 \dots a_N$ to $b_1 \dots b_N$ iff

1. $a_2 \dots a_N = b_1 \dots b_{N-1}$
2. $a_1 a_2 \dots a_N b_N$ is an allowed string in $X_{\mathcal{F}}$

Condition (1) ensures correct overlap.
Condition (2) ensures overlap is allowed.

Directed graph encodes which N-length strings of $X_{\mathcal{F}}$ can follow one another; i.e. it encodes which sequences of transformations are allowed by considering longer strings as the most basic elements.

Reducing 2-IFS to 1-IFS: $\mathcal{F} = \{14, 23, 32, 441\}$

$X_{\mathcal{F}}$ is all strings from $I = \{1, 2, 3, 4\}$ without a substring in $\mathcal{F}$.

Reducing 2-IFS to 1-IFS: $\mathcal{F} = \{14, 23, 32, 441\}$

$X_{\mathcal{F}}$ is all strings from $I = \{1, 2, 3, 4\}$ without a substring in $\mathcal{F}$.

$X_{\mathcal{F}}^{[2]}$ has alphabet J consisting of 13 allowed pairs:

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

Construct directed graph with J as vertex set.

Encode edges via transition matrix.

Ensure conditions (1) and (2) from Theorem in [14] are satisfied, i.e. *correct* and *allowed* overlaps.

Reducing 2-IFS to 1-IFS: $\mathcal{F} = \{14, 23, 32, 441\}$

$J = \{11, 12, 13, 21, 22, 24, 31, 33, 34, 41, 42, 43, 44\}$
Column is source of edge, row is target.

$M_{ij,km} = 1 \iff i = m$ and kij is allowed.

Reducing 2-IFS to 1-IFS: $\mathcal{F} = \{14, 23, 32, 441\}$

$J = \{11, 12, 13, 21, 22, 24, 31, 33, 34, 41, 42, 43, 44\}$
Column is source of edge, row is target.

$M_{ij,km} = 1 \iff i = m$ and kij is allowed.

Overlap conditions yield many 0 entries.
(Only 41/169 nonzero entries in this example.)
Helpful in computational applications, such as computing Hausdorff dimension, powers of transition matrix, etc.

$M_{ij,km} = 1 \iff m = i$ and kij is allowed. Recall $\mathcal{F} = \{14, 23, 32, 441\}$

$$M = \begin{array}{c} \\ 11 \\ 12 \\ 13 \\ 21 \\ 22 \\ 24 \\ 31 \\ 33 \\ 34 \\ 41 \\ 42 \\ 43 \\ 44 \end{array} \begin{array}{c} \begin{array}{cccccccccccc} 11 & 12 & 13 & 21 & 22 & 24 & 31 & 33 & 34 & 41 & 42 & 43 & 44 \end{array} \\ \left[\begin{array}{cccccccccccc} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{array}\right] \end{array}$$

$M_{ij,km} = 1 \iff m = i$ and kij is allowed. Recall $\mathcal{F} = \{14, 23, 32, 441\}$

$M =$

	11	12	13	21	22	24	31	33	34	41	42	43	44
11	1	1	1	0	0	0	0	0	0	0	0	0	0
12	0	0	0	1	1	1	0	0	0	0	0	0	0
13	0	0	0	0	0	0	1	1	1	0	0	0	0
21	1	1	1	0	0	0	0	0	0	0	0	0	0
22	0	0	0	1	1	1	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	1	1	1	1
31	1	1	1	0	0	0	0	0	0	0	0	0	0
33	0	0	0	0	0	0	1	1	1	0	0	0	0
34	0	0	0	0	0	0	0	0	0	1	1	1	1
41	1	1	1	0	0	0	0	0	0	0	0	0	0
42	0	0	0	1	1	1	0	0	0	0	0	0	0
43	0	0	0	0	0	0	1	1	1	0	0	0	0
44	0	0	0	0	0	0	0	0	0	0	1	1	1

$M_{ij,km} = 1 \iff m = i$ and kij is allowed. Recall $\mathcal{F} = \{14, 23, 32, 441\}$

$$M = \begin{array}{c} \\ 11 \\ 12 \\ 13 \\ 21 \\ 22 \\ 24 \\ 31 \\ 33 \\ 34 \\ 41 \\ 42 \\ 43 \\ 44 \end{array} \begin{array}{c} \begin{array}{cccccccccccc} 11 & 12 & 13 & 21 & 22 & 24 & 31 & 33 & 34 & 41 & 42 & 43 & 44 \end{array} \\ \left[\begin{array}{cccccccccccc} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{array}\right] \end{array}$$

Reducing 2-IFS to 1-IFS: $\mathcal{F} = \{14, 23, 32, 441\}$

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

Define $\mathcal{F}'$ to be the forbidden pairs from alphabet J:

$$\mathcal{F}' = \{ij, km \in J \mid M_{ij,km} = 0\}$$

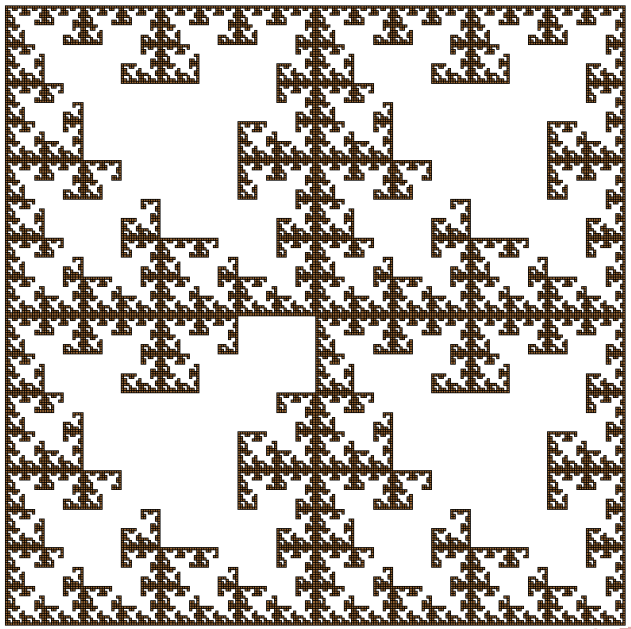
Reducing 2-IFS to 1-IFS: $\mathcal{F} = \{14, 23, 32, 441\}$

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

Define $\mathcal{F}'$ to be the forbidden pairs from alphabet J:

$$\mathcal{F}' = \{ij, km \in J \mid M_{ij,km} = 0\}$$

The same attractor is realized from this 1-IFS, $J(\mathcal{F}')$!



Reducing n-IFS to 1-IFS

Generalizing this procedure, any n-IFS (forbidden strings of length $\leq n+1$) can be reduced to a 1-IFS (forbidden pairs only):

1. List all allowed strings of length n
2. Populate transition matrix by following overlap conditions
3. Apply constructive procedure to this new IFS

Reducing n-IFS to 1-IFS

Generalizing this procedure, any n-IFS (forbidden strings of length $\leq n+1$) can be reduced to a 1-IFS (forbidden pairs only):

1. List all allowed strings of length n
2. Populate transition matrix by following overlap conditions
3. Apply constructive procedure to this new IFS

Theorem

The n-IFS $I(\mathcal{F})$ and the 1-IFS $J(\mathcal{F}')$ have the same attractor.

Proof.

Compare addresses of attractors, rewrite as I-strings. [11] □

Reducing n-IFS to 1-IFS

Observations:

- Procedure doesn't reduce 1-IFS to 0-IFS.
  Overlap conditions are vacuous.
- Previous results in [13] still helpful.
- Can now characterize all n-IFS as IFS-able / ∞-IFS-able / non-IFS-able

Reducing n-IFS to 1-IFS

Observations:

- Procedure doesn't reduce 1-IFS to 0-IFS.
  Overlap conditions are vacuous.
- Previous results in [13] still helpful.
- Can now characterize all n-IFS as IFS-able / ∞-IFS-able / non-IFS-able
- If $\mathcal{F}$ contains strings of length $n+1$, alphabet J may have up to 4^n elements!

Question: What is the most *efficient* memory reduction procedure, yielding the least number of transformations?

Efficient memory reduction: $\mathcal{F} = \{14, 23, 32, 441\}$

Notice 441 is a *primary string*, i.e. it does not contain a forbidden substring.

Efficient memory reduction: $\mathcal{F} = \{14, 23, 32, 441\}$

Subdivide T_4 into four transformations. Define new set $\mathcal{S}$ by

$$S_1 = T_1$$
$$S_2 = T_2$$
$$S_3 = T_3$$
$$S_4 = T_4 \circ T_1$$
$$S_5 = T_4 \circ T_2$$
$$S_6 = T_4 \circ T_3$$
$$S_7 = T_4 \circ T_4$$

Efficient memory reduction: $\mathcal{F} = \{14, 23, 32, 441\}$

Subdivide T_4 into four transformations. Define new set $\mathcal{S}$ by

$$S_1 = T_1$$
$$S_2 = T_2$$
$$S_3 = T_3$$
$$S_4 = T_4 \circ T_1$$
$$S_5 = T_4 \circ T_2$$
$$S_6 = T_4 \circ T_3$$
$$S_7 = T_4 \circ T_4$$

$$\mathcal{F}_{\mathcal{S}} = \{14, 15, 16, 17, 23, 32, 44, 45, 46, 47, 53, 62, 71, 74\}$$

Efficient memory reduction: $\mathcal{F} = \{14, 23, 32, 441\}$

Subdivide T_4 into four transformations. Define new set $\mathcal{S}$ by

$$
\begin{aligned}
S_1 &= T_1 \\
S_2 &= T_2 \\
S_3 &= T_3 \\
S_4 &= T_4 \circ T_1 \\
S_5 &= T_4 \circ T_2 \\
S_6 &= T_4 \circ T_3 \\
S_7 &= T_4 \circ T_4
\end{aligned}
\qquad
M = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array}
\begin{array}{c}
\begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\
\left[\begin{array}{ccccccc}
1 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 1 & 1 & 1 & 1 \\
1 & 0 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 1 & 1 & 1 & 1 \\
1 & 0 & 1 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 0 & 1 & 1 & 1
\end{array}\right]
\end{array}
$$

$$\mathcal{F}_{\mathcal{S}} = \{14, 15, 16, 17, 23, 32, 44, 45, 46, 47, 53, 62, 71, 74\}$$

Efficient memory reduction: 2-IFS to 1-IFS

Conjecture

Given 2-IFS, $I(\mathcal{F})$, efficiently equivalent 1-IFS is generated by

1. *Remove non-primary strings from $\mathcal{F}$*
2. *$\forall ijk \in \mathcal{F}$, subdivide i and j*
3. *Reduce forbidden ijk; 2 cases on whether k subdivided*
4. *Reduce forbidden ij; 4 cases on whether i, j subdivided*
5. *Reduce forbidden i; remove compositions*

Cases determine how many transformations needed in total.

To be proved and investigated for n-IFS in [15].

Results

- Can reduce any n-IFS to 1-IFS, perhaps efficiently.
- Can classify any n-IFS as IFS-able or not.
- Can apply method of [16] to calculate Hausdorff dimension.

Results

- Can reduce any n-IFS to 1-IFS, perhaps efficiently.
- Can classify any n-IFS as IFS-able or not.
- Can apply method of [16] to calculate Hausdorff dimension.
- Know that memory length is *not* a measure of fractal complexity.

Lingering questions

- Is the efficient procedure correct?
- Is the efficient procedure actually helpful in applications?

Lingering questions

- Is the efficient procedure correct?
- Is the efficient procedure actually helpful in applications?
- What exactly is the trade-off between memory length and # of transformations? Are certain formulations best for different applications?
- How many memory reductions are there with a fixed # of transformations?

Lingering questions

- Is the efficient procedure correct?
- Is the efficient procedure actually helpful in applications?
- What exactly is the trade-off between memory length and # of transformations? Are certain formulations best for different applications?
- How many memory reductions are there with a fixed # of transformations?
- What is the relationship between m-IFS and n-IFS? Are there embeddings? Is calculting Hausdorff dimension easier in certain settings? (Partially investigated in [12])

References I

G. Edgar.
Measure, Topology, and Fractal Geometry.
Springer, 2008.

B. B. Mandelbrot
The Fractal Geometry of Nature.
W.H. Freeman and Co., 1982.

Univ. Illinois Urbana-Champaign
Geometry Center Graphics Archive

http://www.geom.uiuc.edu/graphics/

References II

Wired

Earth's Most Stunning Natural Fractal Patterns

http://www.wired.com/wiredscience/2010/09/fractal-patterns-in-nature/?pid=164

Univ. New South Wales

Can Science Be Used To Further Our Understanding Of Art?

http://phys.unsw.edu.au/phys_about/PHYSICS!/FRACTAL_EXPRESSIONISM/fractal_taylor.html

Third Apex to Fractovia

http://www.fractovia.org/art/fmusic/index.html

Dmitry Kormann

http://bowerbird-studios.com/aicaramba/page2.html

References III

Yale Univ.

Fractal geometry course

http://classes.yale.edu/fractals/

Yale Univ., Fractals Panorama

http://classes.yale.edu/fractals/panorama/Biology/Physiology/Physiology.html

Eric Green, Univ. Wisc.

Iterated Function Systems

http://pages.cs.wisc.edu/ ergreen/honors_thesis/IFS.html

R. Bedient, M. Frame, K. Gross, J. Lanski, B. Sullivan

Higher Block IFS 1: Memory Reduction and Dimension Computations

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

References IV

R. Bedient, M. Frame, K. Gross, J. Lanski, B. Sullivan
Higher Block IFS 2: Relations Between IFS with Different Levels of Memory
Fractals, **Vol. 18, No. 4** (2010) 399-408

M. Frame, J. Lanski
When is a recurrent IFS attractor a standard IFS attractor?

Fractals, **7** (1999), 257-266.

D. Lind, B. Marcus
An Introduction to Symbolic Dynamics and Coding
Cambridge Univ. Press, 1995

References V

K. Gross, R. Bedient, M. Frame

Efficient memory reduction of IFS with memory

R. Mauldin, S. Williams

Hausdorff dimension in graph directed constructions

Trans. Am. Math. Soc. **309** (1988) 811-829

AI Game Programmers Guild, Rescue on Fractalus!

http://gameai.com/wiki/index.php?title=Rescue_on_Fractalus!

Applications of Fractals - Molecules

http://library.thinkquest.org/26242/full/ap/ap13.html

Fractals for the Classroom

http://www.squidoo.com/fractalsclassroom

THANK YOU