Fractals from Regular Behaviours

Todd Schmid
University College, London, UK

Victoria Noquez
St. Mary's College of California, Moraga, CA, USA

Lawrence S. Moss
Indiana University, Bloomington, IN, USA

Abstract

We are interested in connections between the theory of fractal sets obtained as attractors of iterated function systems and process calculi. To this end, we reinterpret Milner's expressions for processes as contraction operators on a complete metric space. When the space is, for example, the plane, the denotations of fixed point terms correspond to familiar fractal sets. We give a sound and complete axiomatization of fractal equivalence, the congruence on terms consisting of pairs that construct identical self-similar sets in all interpretations. We further make connections to labelled Markov chains and to invariant measures. In all of this work, we use important results from process calculi. For example, we use Rabinovich's completeness theorem for trace equivalence in our own completeness theorem. In addition to our results, we also raise many questions related to both fractals and process calculi.

2012 ACM Subject Classification Theory of computation → Process calculi

Keywords and phrases fixed-point terms, labelled transition system, fractal, final coalgebra, equational logic, completeness

Digital Object Identifier 10.4230/LIPIcs.CALCO.2023.14

Funding *Todd Schmid*: Partially funded by ERC Grant Autoprobe (grant agreement 10100269).
Lawrence S. Moss: Supported by grant #586136 from the Simons Foundation.

Acknowledgements We would like to thank Alexandra Silva and Dylan Thurston for helpful discussions. The images in Figures 1 and 2 were made using SageMath and GIMP.

1 Introduction

Hutchinson noticed in [13] that many familiar examples of fractals can be captured as the set-wise fixed-point of a finite family of contraction (i.e., distance shrinking) operators on a metric space. He called these spaces *(strictly) self-similar*, since the intuition behind the contraction operators is that they are witnesses for the appearance of the fractal in a proper (smaller) subset of itself. For example, the famous Sierpiński gasket is the unique nonempty compact subset of the plane left fixed by the union of the three operators $\sigma_a, \sigma_b, \sigma_c : \mathbb{R}^2 \to \mathbb{R}^2$ in Figure 1. The Sierpiński gasket is a scaled-up version of each of its thirds.

The self-similarity of Hutchinson's fractals hints at an algorithm for constructing them: Each point in a self-similar set is the limit of a sequence of points obtained by applying the contraction operators one after the other to an initial point. In the Sierpiński gasket, the point $(1/4, \sqrt{3}/4)$ is the limit of the sequence

$$p,\ \sigma_b(p),\ \sigma_b\sigma_a(p),\ \sigma_b\sigma_a\sigma_a(p),\ \sigma_b\sigma_a\sigma_a\sigma_a(p),\ \dots \tag{1}$$

where the initial point p is an arbitrary element of $\mathbf{R}^2$ (note that σ_b is applied last). Hutchinson showed in [13] that the self-similar set corresponding to a given family of contraction operators is precisely the collection of points obtained in the manner just described. The limit of the

10th Conference on Algebra and Coalgebra in Computer Science (CALCO 2023).
Editors: Paolo Baldan and Valeria de Paiva; Article No. 14; pp. 14:1–14:18
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

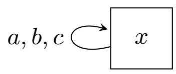
sequence in (1) does not depend on the initial point p because $\sigma_a, \sigma_b, \sigma_c$ are contractions. Much like digit expansions to real numbers, every stream of a's, b's, and c's corresponds to a unique point in the Sierpiński gasket. The point $(1/4, \sqrt{3}/4)$, for example, corresponds to the stream $(b, a, a, a, \dots)$ ending in an infinite sequence of a's. Conversely, every point in the Sierpiński gasket comes from (in general more than one) corresponding stream.

From a computer science perspective, the languages of streams considered by Hutchinson are the *traces* observed by one-state labelled transition systems, like the one in Figure 1. We investigate whether one could achieve a similar effect with languages of streams obtained from labelled transition systems having more than one state. Observe, for example, Figure 2. This twisted version of the Sierpiński gasket is constructed from a two-state labelled transition system. Each point in the twisted Sierpiński gasket corresponds to a stream of a's, b's, and c's, but not every stream corresponds to a point in the set: The limit corresponding to $(c, a, b, c, c, c, \dots)$ is $(3/4, \sqrt{3}/8)$, for example.

A labelled transition system paired with an interpretation of its labels as contractions on a complete metric space is the same data as a *directed-graph iterated function system* (GIFS), a generalization of iterated function systems introduced by Mauldin and Williams [18]. GIFSs generate their own kind of self-similar set, and much work has been done to understand the geometric properties of fractal sets generated by GIFSs [7–10, 18]. We take this work in a slightly different direction by presenting a coalgebraic perspective on GIFSs, seeing each labelled transition system as a "recipe" for constructing fractal sets.

In analogy with the theory of regular languages, we call the fractals generated by finite labelled transition systems *regular subfractals*, and give a logic for deciding if two labelled transition systems represent the same recipe under all interpretations of the labels. By identifying points in the fractal set generated by a labelled transition system with traces observed by the labelled transition system, it is reasonable to suspect that two labelled transition systems represent equivalent fractal recipes – i.e., they represent the same fractal under every interpretation – if and only if they are trace equivalent. This is the content of Theorem 4.4, which allows us to connect the theory of fractal sets to mainstream topics in computer science.

Labelled transition systems are a staple of theoretical computer science, especially in the area of process algebra [1], where a vast array of different notions of equivalence and axiomatization problems have been studied. We specifically use a syntax introduced by Milner in [22] to express labelled transition systems as terms in an expression language with recursion. This leads us to a fragment of Milner's calculus consisting of just the terms that constitute recipes for fractal constructions. Using a logic of Rabinovich [25] for deciding trace equivalence in Milner's calculus, we obtain a complete axiomatization of fractal recipe equivalence.



$$\sigma_a \begin{bmatrix} r \\ s \end{bmatrix} = \begin{bmatrix} \frac{1}{2}r + \frac{1}{4} \\ \frac{1}{2}s + \frac{\sqrt{3}}{4} \end{bmatrix} \qquad \sigma_b \begin{bmatrix} r \\ s \end{bmatrix} = \begin{bmatrix} \frac{1}{2}r \\ \frac{1}{2}s \end{bmatrix}$$

$$\sigma_c \begin{bmatrix} r \\ s \end{bmatrix} = \begin{bmatrix} \frac{1}{2}r + \frac{1}{2} \\ \frac{1}{2}s \end{bmatrix}$$

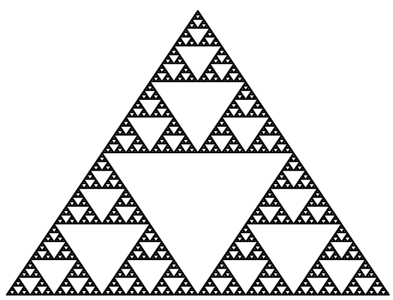
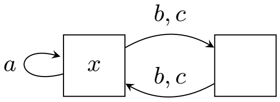


Figure 1 The Sierpiński gasket is the unique nonempty compact subset $\mathbf{S}$ of $\mathbf{R}^2$ such that $\mathbf{S} = \sigma_a(\mathbf{S}) \cup \sigma_b(\mathbf{S}) \cup \sigma_c(\mathbf{S})$. Each of its points corresponds to a stream emitted by the state x.



$$\sigma_a \begin{bmatrix} r \\ s \end{bmatrix} = \begin{bmatrix} \frac{1}{2}r + \frac{1}{4} \\ \frac{1}{2}s + \frac{\sqrt{3}}{4} \end{bmatrix} \qquad \sigma_b \begin{bmatrix} r \\ s \end{bmatrix} = \begin{bmatrix} \frac{r}{2} \\ \frac{s}{2} \end{bmatrix}$$

$$\sigma_c \begin{bmatrix} r \\ s \end{bmatrix} = \begin{bmatrix} \frac{r}{2} + \frac{1}{2} \\ \frac{s}{2} \end{bmatrix}$$

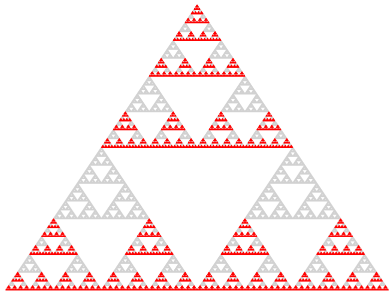


Figure 2 A twisted Sierpińksi gasket, depicted in red. In the construction of this set, σ_b and σ_c are applied twice to a single copy of σ_a applied to the set. This has the effect of systematically removing the "top" part of the Sierpiński gasket from its bottom thirds.

In his study of self-similar sets, Hutchinson also makes use of probability measures supported on self-similar sets, called *invariant measures*. Each invariant measure is specified by a probability distribution on the set of contractions generating its support. In the last technical section of the paper, we adapt the construction of invariant measures to a probabilistic version of labelled transition systems called *labelled Markov chains*, which allows us to give a measure-theoretic semantics to terms in a probabilistic version of Milner's specification language, the calculus introduced by Stark and Smolka [27]. Our measure-theoretic semantics of probabilistic process terms can be seen as a generalization of the trace measure semantics of Kerstan and König [14]. We offer a sound axiomatization of equivalence under this semantics and pose completeness as an open problem.

In sum, the contributions of this paper are as follows.

- In Section 3, we give a fractal recipe semantics to process terms using a generalization of iterated function systems.
- In Section 4, we show that two process terms agree on all fractal interpretations if and only if they are trace equivalent. This implies that fractal recipe equivalence is decidable for process terms, and it allows us to derive a complete axiomatization of fractal recipe equivalence from Rabinovich's axiomatization [25] of trace equivalence of process terms.
- Finally, we adapt the fractal semantics of process terms to the probabilistic setting in Section 5 and propose an axiomatization of probabilistic fractal recipe equivalence.

We start with a brief overview of trace semantics in process algebra and Rabinovich's Theorem (Theorem 2.7) in Section 2.

2 Labelled Transition Systems and Trace Semantics

Labelled transition systems are a widely used model of nondeterminism. Given a fixed finite set A of *action labels*, a *labelled transition system* (LTS) is a pair (X, α) consisting of a set X of *states* and a *transition function* $\alpha : X \to \mathcal{P}(A \times X)$. We generally write $x \xrightarrow{a}_\alpha y$ if $(a, y) \in \alpha(x)$, or simply $x \xrightarrow{a} y$ if α is clear from context, and say that x *emits* a *and transitions to* y.

Given a state x of an LTS (X, α), we write $\langle x \rangle_\alpha$ for the LTS obtained by restricting the relations $\xrightarrow{a}$ to the set of states *reachable* from x, meaning there exists a *path* of the form $x \xrightarrow{a_1} x_1 \to \cdots \to x_{n-1} \xrightarrow{a_n} x_n$. We refer to $\langle x \rangle_\alpha$ as either the LTS *generated by* x, or as the *process starting at* x. An LTS (X, α) is *locally finite* if $\langle x \rangle_\alpha$ is finite for all states x.

$$ae \xrightarrow{a} e \qquad \frac{e_1 \xrightarrow{a} f}{e_1 + e_2 \xrightarrow{a} f} \qquad \frac{e_2 \xrightarrow{a} f}{e_1 + e_2 \xrightarrow{a} f} \qquad \frac{e[\mu v\ e/v] \xrightarrow{a} f}{\mu v\ e \xrightarrow{a} f}$$

Figure 3 The relation $\xrightarrow{a} \subseteq \mathsf{Term} \times \mathsf{Term}$ defining (Term, γ).

Traces

In the context of the current work, nondeterminism occurs when a process branches into multiple threads that execute in parallel. Under this interpretation, to an outside observer (without direct access to the implementation details of an LTS), two processes that emit the same set of sequences of action labels are indistinguishable.

Formally, let A^* be the set of words formed from the alphabet A. Given a state x of an LTS (X, α), the set $\mathrm{tr}_\alpha(x)$ of *traces emitted by* x is the set of words $a_1 \dots a_n \in A^*$ such that there is a path of the form $x \xrightarrow{a_1} x_1 \to \cdots \to x_{n-1} \xrightarrow{a_n} x_n$ through (X, α). Two states x and y are called *trace equivalent* if $\mathrm{tr}(x) = \mathrm{tr}(y)$. Each trace language $\mathrm{tr}(x)$ is *prefix-closed*, which for a language L means that $w \in L$ whenever $wa \in L$.

Trace equivalence is a well-documented notion of equivalence for processes [3, 11], and we shall see it in our work on fractals as well.

▶ **Definition 2.1.** *A* stream *is an infinite sequence* $(a_1, a_2, \dots)$ *of letters from* A*. A state* x *in an LTS* (X, α) emits *a stream* $(a_1, \dots)$ *if for any* $n > 0$, $a_1 \cdots a_n \in \mathrm{tr}(x)$*. We write* $\mathrm{str}(x)$ *for the set of streams emitted by* x*.*

In our construction of fractals from LTSs, points are represented only by (infinite) streams. We therefore focus primarily on LTSs with the property that for all states x, $\mathrm{tr}(x)$ is precisely the set of finite prefixes of streams emitted by x. We refer to an LTS (X, α) satisfying this condition as *productive*. Productivity is equivalent to the absence of *deadlock* states, states with no outgoing transitions.

▶ **Lemma 2.2.** *Let* (X, α) *be an LTS. Then the following are equivalent: (i) for any* $x, y \in X$, $\mathrm{str}(x) = \mathrm{str}(y)$ *if and only if* $\mathrm{tr}(x) = \mathrm{tr}(y)$*; (ii) for any* $x \in X$, $\alpha(x) \neq \emptyset$*.*

Specification

We use the following language for specifying processes: Starting with a fixed countably infinite set $\{v_1, v_2, \dots\}$ of *variables*, the set of *terms* is given by the grammar

$$v \mid ae \mid e_1 + e_2 \mid \mu v\ e$$

where v is v_i for some $i \in \mathbb{N}$, $a \in A$, and e, e_1, e_2 are terms.

Intuitively, the process ae emits a and then turns into e, and $e_1 + e_2$ is the process that nondeterministically branches into e_1 and e_2. The process $\mu v\ e$ is like e, but with instances of v that appear free in e acting like `goto` expressions that return the process to $\mu v\ e$.

▶ **Definition 2.3.** *A* (process) term *is a term* e *in which every occurrence of a variable* v *appears both within the scope of a* $\mu v\ (-)$ *(*e *is* closed*) and within the scope of an* $a(-)$ *(*e *is* guarded*). The set of process terms is written* Term*. The set of process terms themselves form the LTS* (Term, γ) *defined in Figure 3.*

In Figure 3, we use the notation $e[g/v]$ to denote the expression obtained by replacing each *free* occurrence of v in e (one which does not appear within the scope of a $\mu v\ (-)$ operator) with the expression g. Given $e \in \mathsf{Term}$, the process *specified by* e is the LTS $\langle e \rangle_\gamma$.

$$
\begin{array}{llll}
\text{(ID)} & e + e \equiv e & \text{(CN)} & \dfrac{(\forall i)\ e_i \equiv f_i}{g[\vec{e}/\vec{v}] \equiv g[\vec{f}/\vec{v}]} \\
\text{(CM)} & e_2 + e_1 \equiv e_1 + e_2 & & \\
\text{(AS)} & e_1 + (e_2 + e_3) \equiv (e_1 + e_2) + e_3 & \text{(AE)} & \dfrac{}{\mu w\ e \equiv \mu v\ e[v/w]} \\
\text{(DS)} & a(e_1 + e_2) \equiv ae_1 + ae_2 & & \\
\text{(FP)} & \mu v\ e \equiv e[\mu v\ e/v] & \text{(UA)} & \dfrac{g \equiv e[g/v]}{g \equiv \mu v\ e}
\end{array}
$$

Figure 4 The axioms and rules of the provable equivalence relation in addition to those of equational logic (not shown). Here, $e, e_i, f, f_i, g \in \mathsf{Term}$ for all i. In (CN), g has precisely the free variables $v_1, \dots, v_n$, and no variable that appears free in f_i is bound in g for any i. In (AE), v does not appear free in e.

▶ Remark 2.4. The set of process terms, as we have named them, is the fragment of Milner's fixed-point calculus from [22] consisting of only the terms that specify productive LTSs.

Labelled transition systems specified by process terms are finite and productive, and conversely, every finite productive process is trace-equivalent to some process term.

▶ **Lemma 2.5** ([22, Proposition 5.1]). *For any $e \in \mathsf{Term}$, the set of terms reachable from e in (Term, γ) is finite. Conversely, if x is a state in a finite productive LTS (X, α), then there is a process term e such that $\mathrm{tr}(e) = \mathrm{tr}_\alpha(x)$.*

Axiomatization of trace equivalence

Given an interpretation of process terms as states in an LTS, and given the notion of trace equivalence, one might ask if there is an algebraic or proof-theoretic account of trace equivalence of process terms. Rabinovich showed in [25] that a complete inference system for trace equivalence can be obtained by adapting earlier work of Milner [22]. The axioms of the complete inference system include equations like $e_1 + e_2 = e_2 + e_1$ and $a(e_1 + e_2) = ae_1 + ae_2$, which are intuitively true for trace equivalence.

To be more precise, given any function with domain Term, say $\sigma : \mathsf{Term} \to Z$, call an equivalence relation $\sim$ *sound with respect to* σ if $e \sim f$ implies $\sigma(e) = \sigma(f)$, and *complete with respect to* σ if $\sigma(e) = \sigma(f)$ implies $e \sim f$. Then the smallest equivalence relation $\equiv$ on Term containing all the pairs derivable from the axioms and inference rules appearing in Figure 4 is sound and complete with respect to $\mathrm{tr} = \mathrm{tr}_\gamma \colon \mathsf{Term} \to \mathcal{P}(A^*)$.

▶ **Definition 2.6.** *Given $e_1, e_2 \in \mathsf{Term}$, we say that e_1 and e_2 are* provably equivalent *if $e_1 \equiv e_2$, and call $\equiv$* provable equivalence*.*

▶ **Theorem 2.7** (Rabinovich [25]). *Let $e_1, e_2 \in \mathsf{Term}$. Then $e_1 \equiv e_2$ iff $\mathrm{tr}(e_1) = \mathrm{tr}(e_2)$.*

▶ **Example 2.8.** Consider the processes specified by $e_1 = \mu w\ \mu v\ (a_1a_2v + a_1a_3w)$ and $e_2 = \mu v\ (a_1(a_2v + a_3v))$. The traces emitted by both e_1 and e_2 are those that alternate between a_1 and either a_2 or a_3. We can show these expressions are trace equivalent via the formal deduction in Figure 5.

Rabinovich's theorem tells us that, up to provable equivalence, our specification language consisting of process terms is really a specification language for languages of traces. In what follows, we are going to give an alternative semantics to process terms by using LTSs to

$$
\begin{aligned}
e_1 &= \mu w\ \mu v\ (a_1 a_2 v + a_1 a_3 w) \\
&\stackrel{\text{(FP)}}{\equiv} \mu v\ (a_1 a_2 v + a_1 a_3 e_1) \\
&\stackrel{\text{(FP)}}{\equiv} a_1 a_2 e_1 + a_1 a_3 e_1 \\
&\stackrel{\text{(DS)}}{\equiv} a_1 (a_2 e_1 + a_3 e_1) \\
&\stackrel{\text{(UA)}}{\equiv} \mu v\ (a_1 (a_2 v + a_3 v))
\end{aligned}
$$

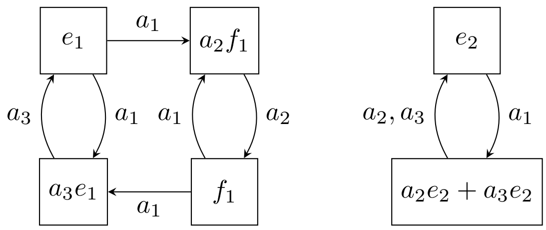


Figure 5 Deducing $e_1 \equiv e_2$. Above, $f_1 = \mu v\ (a_1 a_2 v + a_1 a_3 e_1)$.

generate fractal subsets of metric spaces. The main result of our paper is that these two semantics coincide: Two process terms are trace equivalent if and only if they generate the same fractals. This is the content of Sections 3 and 4 below.

3 Fractals from Labelled Transition Systems

In the Sierpiński gasket $\mathbf{S}$ from Figure 1, every point of $\mathbf{S}$ corresponds to a stream of letters from the alphabet $\{a, b, c\}$, and every stream corresponds to a unique point. To obtain the point corresponding to a particular stream $(a_1, a_2, a_3, \dots)$ with each $a_i \in \{a, b, c\}$, start with any $p \in \mathbb{R}^2$ and compute the limit $\lim_{n\in\mathbb{N}} \sigma_{a_1} \cdots \sigma_{a_n}(p)$. The point in the fractal corresponding to $(a_1, a_2, a_3, \dots)$ does not depend on p because $\sigma_a, \sigma_b, \sigma_c$ in Figure 1 are *contraction operators*.

▶ **Definition 3.1.** *Given a metric space (M, d), a* contraction operator *on (M, d) is a function $h : M \to M$ such that for some $r \in [0, 1)$, $d(h(x), h(y)) \leq r\ d(x, y)$ for any $x, y \in M$. The number r is called a* contraction coefficient *of h. The set of contraction operators on (M, d) is written* $\mathrm{Con}(M, d)$.

For example, with the Sierpiński gasket (Figure 1) associated to the contractions σ_a, σ_b, and σ_c, $r = 1/2$ is a contraction coefficient for all three maps. Now, given $p, q \in \mathbb{R}^2$,

$$d(\sigma_{a_1} \cdots \sigma_{a_n}(p), \sigma_{a_1} \cdots \sigma_{a_n}(q)) \leq \frac{1}{2^n}\ d(p, q)$$

for all n, so it follows that $\lim_{n\in\mathbb{N}} \sigma_{a_1} \cdots \sigma_{a_n}(p) = \lim_{n\in\mathbb{N}} \sigma_{a_1} \cdots \sigma_{a_n}(q)$. For any finite set of contraction operators $\{\sigma_{a_1}, \dots, \sigma_{a_n}\}$ indexed by A and acting on a complete metric space (M, d), every stream from A corresponds to a unique point in M.

▶ **Definition 3.2.** *A* contraction operator interpretation *is a function $\sigma\colon A \to \mathrm{Con}(M, d)$. We usually write $\sigma_a = \sigma(a)$. Given $\sigma\colon A \to \mathrm{Con}(M, d)$ and a stream $(a_1, \dots)$ from A, define*

$$\sigma_\omega\colon A^\omega \to M \qquad \sigma_\omega(a_1, \dots) = \lim_{n\in\mathbb{N}} \sigma_{a_1} \cdots \sigma_{a_n}(x) \tag{2}$$

where $x \in M$ is arbitrary. The self-similar set *corresponding to a contraction operator interpretation σ is the set $\mathbf{S}_\sigma = \{\sigma_\omega(a_1, \dots) \mid (a_1, \dots)$ is a stream from $A\}$.*

▶ Remark 3.3. Note that in (2), the contraction operators corresponding to the initial trace $(a_1, \dots, a_n)$ are applied in *reverse* order. That is, σ_{a_n} is applied before $\sigma_{a_{n-1}}$, $\sigma_{a_{n-2}}$ is applied before $\sigma_{a_{n-1}}$, and so on.

Regular Subfractals

Generalizing the fractals of Mandelbrot [17], Hutchinson introduced self-similar sets in [13] and gave a comprehensive account of their theory. In op. cit., Hutchinson defines a self-similar set to be the invariant set of an *iterated function system*. In our terminology, an iterated function system is equivalent to a contraction operator interpretation of a finite set A of actions, and the invariant set is the total set of points obtained from streams from A. The fractals constructed from a LTS paired with a contraction operator interpretation generalize Hutchinson's self-similar sets to nonempty compact sets of points obtained from certain subsets of the streams, namely the subsets emitted by the LTS.

Write $\mathbf{K}(M, d)$ for the set of nonempty compact subsets of (M, d). Given a state x of a productive LTS (X, α) and a contraction operator interpretation $\sigma : A \to \mathrm{Con}(M, d)$, we define $[\![-]\!]_{\alpha,\sigma} : X \to \mathbf{K}(M, d)$ by

$$[\![x]\!]_{\alpha,\sigma} = \{\sigma_\omega(a_1, \dots) \mid (a_1, \dots) \text{ emitted by } x\} \tag{3}$$

and call this the set *generated by the state* x. As we will see, $[\![x]\!]_{\alpha,\sigma}$ is always nonempty and compact.

▶ **Definition 3.4.** *Given a process term* $e \in \mathsf{Term}$ *and a contraction operator interpretation* $\sigma : A \to \mathrm{Con}(M, d)$*, the* regular subfractal semantics of e *corresponding to* σ *is* $[\![e]\!]_\sigma = [\![e]\!]_{\gamma,\sigma}$.

For example, the set of points depicted in Figure 2 is the regular subfractal semantics of $\mu v\ (av + b(bv + cv) + c(bv + cv))$ corresponding to the interpretation σ given in that figure. The regular subfractal semantics of e is a proper subset of the Sierpiński Gasket, and in particular does not contain the point corresponding to $(c, a, b, c, b, c, \dots)$.

Systems and Solutions

Self-similar sets are often characterized as the unique nonempty compact sets that solve systems of equations of the form

$$K = \sigma_1(K) \cup \cdots \cup \sigma_n(K)$$

with each σ_i a contraction operator on a complete metric space. For example, the Sierpiński gasket is the unique nonempty compact solution to $K = \sigma_a(K) \cup \sigma_b(K) \cup \sigma_c(K)$. In this section, we are going to provide a similar characterization for regular subfractals that will play an important role in the completeness proof in Section 4.

One way to think of an n-state LTS (X, α) is as a system of formal equations

$$x_i = a_{k_1} x_{j_1} + \cdots + a_{k_m} x_{j_m}$$

indexed by $X = \{x_1, \dots, x_n\}$, where $x_i \xrightarrow{a_{k_l}}_\alpha x_{j_l}$ for $k_1, \dots, k_m, j_1, \dots, j_m \leq n$.

▶ **Definition 3.5.** *Given a contraction operator interpretation* $\sigma : A \to \mathrm{Con}(M, d)$*, and an LTS* (X, α)*, we call a function* $\varphi : X \to \boldsymbol{K}(M, d)$ *a* (σ-)solution *to* (X, α) *if for any* $x \in X$*,*

$$\varphi(x) = \bigcup_{x \xrightarrow{a} y} \sigma_a(\varphi(y))$$

▶ **Example 3.6.** Let $\mathbf{S}$ be the Sierpiński gasket as a subset of $\mathbb{R}^2$. Let (X, α) be the LTS in Figure 1. Then we have a single state, x, with $x \xrightarrow{a,b,c} x$. The function $\varphi \colon X \to \mathbf{K}(\mathbb{R}^2, d)$ given by $\varphi(s) = \mathbf{S}$ is a solution to (X, α), because $\mathbf{S} = \sigma_a(\mathbf{S}) \cup \sigma_b(\mathbf{S}) \cup \sigma_c(\mathbf{S})$.

Finite productive LTSs have unique solutions.

▶ **Lemma 3.7.** *Let (M, d) be a complete metric space, $\sigma : A \to \mathrm{Con}(M, d)$, and (X, α) be a finite productive LTS. Then (X, α) has a unique solution φ_α.*

The proof of Lemma 3.7 makes use of the *Hausdorff metric* on $\mathbf{K}(M, d)$, defined

$$d(K_1, K_2) = \max \left\{ \sup_{u \in K_1} \inf_{v \in K_2} d(u, v), \sup_{v \in K_2} \inf_{u \in K_1} d(u, v) \right\} \tag{4}$$

This equips $\mathbf{K}(M, d)$ with the structure of a metric space. If M is complete, so is $\mathbf{K}(M, d)$. Incidentally, we need to restrict to *nonempty* sets in (4). This is the primary motivation for the guardedness condition which we imposed on our terms. We also recall the *Banach fixed-point theorem*, which allows for the computation of fixed-points by iteration.

▶ **Theorem 3.8** (Banach [2]). *Let (M, d) be a complete nonempty metric space and $f : M \to M$ a contraction map. Then $\lim_{n \in \mathbb{N}} f^n(q)$ is the unique fixed-point of f.*

Fix a complete nonempty metric space (M, d), a productive finite LTS (X, α), and a contraction operator interpretation $\sigma : A \to \mathrm{Con}(M, d)$. To compute the solution to (X, α), we iteratively apply a matrix-like operator to the set $\mathbf{K}(M, d)^X$ of vectors $[K_{x_1}, \ldots, K_{x_n}]$ with entries in $\mathbf{K}(M, d)$ indexed by X. Formally, we define

$$[\alpha]_\sigma : \mathbf{K}(M, d)^X \to \mathbf{K}(M, d)^X \qquad ([\alpha]_\sigma \vec{K})_x = \bigcup_{x \xrightarrow{a} y} \sigma_a(K_y)$$

for each $x \in X$. Intuitively, $[\alpha]_\sigma$ acts like an $X \times X$-matrix of unions of contractions.

Proof of Lemma 3.7. Every fixed-point of $[\alpha]_\sigma$ corresponds to a solution of (X, α). Given a fixed-point $\vec{F}$, i.e., $[\alpha]_\sigma \vec{F} = \vec{F}$, and defining $\varphi : X \to \mathbf{K}(M, d)^X$ by $\varphi(x) = F_x$, we see that

$$\varphi(x) = F_x = ([\alpha]_\sigma \vec{F})_x = \bigcup_{x \xrightarrow{a} y} \sigma_a(F_y) = \bigcup_{x \xrightarrow{a} y} \sigma_a(\varphi(y))$$

Conversely, if $\varphi : X \to \mathbf{K}(M, d)$ is a solution to (X, α), then defining $F_x = \varphi(x)$ we have

$$F_x = \varphi(x) = \bigcup_{x \xrightarrow{a} y} \sigma_a(\varphi(y)) = \bigcup_{x \xrightarrow{a} y} \sigma_a(F_y) = ([\alpha]_\sigma \vec{F})_x$$

for each $x \in X$. Thus, it suffices to show that $[\alpha]_\sigma$ has a unique fixed-point. By the Banach Fixed-Point Theorem 3.8, we just need to show that $[\alpha]_\sigma$ is a contraction operator. That is, $[\alpha]_\sigma \in \mathrm{Con}(\mathbf{K}(M, d))$, where d is the Hausdorff metric. This point is standard in the fractals literature; cf. [13]. ◀

Fractal Semantics and Solutions

Recall that the fractal semantics of a process term e with respect to a contraction operator interpretation $\sigma : A \to \mathrm{Con}(M, d)$ is the set $[\![e]\!]_\sigma$ of limits of streams applied to points in the complete metric space (M, d).

▶ **Theorem 3.9.** *Let (X, α) be a finite productive LTS and let $x \in X$. Given a complete metric space (M, d), and $\sigma : A \to \mathrm{Con}(M, d)$,*

1. *$[\![x]\!]_{\alpha,\sigma} \in \boldsymbol{K}(M, d)$, i.e., $[\![x]\!]_{\alpha,\sigma}$ is nonempty and compact.*
2. *$[\![-]\!]_{\alpha,\sigma} : X \to \boldsymbol{K}(M, d)$ is the unique solution to (X, α).*

In particular, (Term, γ) is locally finite, and so by Lemma 3.7 has a unique solution. Theorem 3.9 therefore implies that this solution is $[\![-]\!]_\sigma$.

Given a solution φ and a state x, call $\varphi(x)$ the *x-component* of the solution φ. We obtain the following, which can be seen as an analogue of Kleene's theorem for regular expressions [15], as a direct consequence of Theorem 3.9.

▶ **Theorem 3.10.** *A subset of a self-similar set is a regular subfractal if and only if it is a component of a solution to a finite productive LTS.*

4 Fractal Equivalence is Traced

We have seen that finite productive LTSs (LTSs that only emit infinite streams) can be specified by process terms. We also introduced a family of fractal sets called regular subfractals, those subsets of self-similar sets obtained from the streams emitted by a finite productive LTS. An LTS itself is representative of a certain system of equations, and set-wise the system of equations is solved by the regular subfractals corresponding to it. Going from process terms to LTSs to regular subfractals, we see that a process term is representative of a sort of *uninterpreted fractal recipe*, which tells us how to obtain a regular subfractal from an interpretation of action symbols as contractions on a complete metric space.

▶ **Definition 4.1.** *Given $e, f \in \mathsf{Term}$, we write $e \approx f$ if for every complete metric space (M, d) and every contraction operator interpretation $\sigma : A \to \mathrm{Con}(M, d)$, $[\![e]\!]_\sigma = [\![f]\!]_\sigma$. We say that e and f are* fractal equivalent *or that they are* equivalent fractal recipes *when $e \approx f$.*

▶ **Theorem 4.2.** *Let $e, f \in \mathsf{Term}$. Then $e \approx f$ if and only if $\mathrm{str}(e) = \mathrm{str}(f)$.*

In essence, this is a soundness/completeness theorem for our version of Rabinovich's logic with respect to its fractal semantics that we presented. Our proof relies on the logical characterization of trace equivalence that we saw in Theorem 2.7.

▶ **Lemma 4.3** (Soundness). *For any $e, f \in \mathsf{Term}$, if $e \equiv f$, then $e \approx f$.*

▶ **Theorem 4.4** (Completeness). *For any $e, f \in \mathsf{Term}$,if $e \approx f$, then $e \equiv f$.*

Proof. Consider the space (A^ω, d) of streams from A with the metric below:

$$d((a_1, \dots), (b_1, \dots)) = \inf \left\{2^{-n} \;\middle|\; (\forall i \leq n)\ a_i = b_i\right\}$$

This space is the *Cantor set on A symbols*, a compact metric space. For any productive LTS (X, α) and $x \in X$, $\mathrm{str}(x)$ is a nonempty closed subset of (A^ω, d), for the following reason: Given a Cauchy sequence $\{(a_1^{(i)}, \dots)\}_{i\in\mathbb{N}}$ in $\mathrm{str}(x)$, let $(a_1, \dots)$ be its limit in (A^ω, d). Then x emits every finite initial segment of $(a_1, \dots)$ because for any $N \in \mathbb{N}$ there is an $m \in \mathbb{N}$ such that $(a_1, \dots, a_m, a_{m+1}^{(m)}, \dots) \in \mathrm{str}(x)$ for $m > N$. By compactness of (A^ω, d), we therefore have $\mathrm{str}(x) \in \mathbf{K}(A^\omega, d)$, so $\mathrm{str}\colon X \to \mathbf{K}(A^\omega, d)$.

For each $a \in A$, let $\sigma_a : A^\omega \to A^\omega$ be the map $\sigma_a(a_1, \dots) = (a, a_1, \dots)$. Then $\sigma : A \to \mathrm{Con}(A^\omega, d)$. By construction, $\mathrm{str}(x) = \bigcup_{x \xrightarrow{a} y} \sigma_a(\mathrm{str}(y))$ for any $x \in X$. By the uniqueness of fixed points we saw in Lemma 3.7, we therefore have $\mathrm{str}(x) = [\![x]\!]_{\alpha,\sigma}$.

To finish the proof, consider (Term, γ). If $e, f \in \mathsf{Term}$ and $e \approx f$, then in particular, $\mathrm{str}(e) = \mathrm{str}(f)$, because $\mathrm{str} = [\![-]\!]_{\gamma,\sigma}$ with $\sigma\colon A \to \mathrm{Con}(A^\omega, d)$ as above. Since (Term, γ) is productive, $\mathrm{tr}(e) = \mathrm{str}(e)$ and $\mathrm{tr}(f) = \mathrm{str}(f)$, so in particular, e and f are trace equivalent. By Rabinovich's Theorem, Theorem 2.7, $e \equiv f$, as desired. ◀

5 A Calculus of Subfractal Measures

Aside from showing the existence of self-similar sets and their correspondence with contraction operator interpretations (in Hutchinson's terminology, iterated function systems), Hutchinson also shows that every probability distribution on the contractions corresponds to a unique measure, called the *invariant measure*, that satisfies a certain recursive equation and whose support is the self-similar set. In this section, we replay the story up to this point, but with Hutchinson's invariant measure construction instead of the invariant (self-similar) set construction. We make use of a probabilistic version of LTSs called *labelled Markov chains*, as well as a probabilistic version of Milner's specification language introduced by Stark and Smolka [27] to specify fractal measures. Similar to how fractal equivalence coincides with trace equivalence, *fractal measure equivalence* is equivalent to a probabilistic version of trace equivalence due to Kerstan and König [14].

Invariant measures

Recall that a *Borel probability measure* on a metric space (M, d) is a $[0, \infty]$-valued function ρ defined on the Borel subsets of M (the smallest σ-algebra containing the open balls of (M, d)) that is countably additive and assigns $\rho(\emptyset) = 0$ and $\rho(M) = 1$.

Hutchinson shows in [13] that, given $\sigma\colon A \to \mathrm{Con}(M, d)$, each probability distribution $\rho\colon A \to [0, 1]$ on A gives rise to a unique Borel probability measure $\hat{\rho}$, called the *invariant measure*, satisfying the equation below and supported by the self-similar set $\mathbf{S}_\sigma$:

$$\hat{\rho}(B) = \sum_{a\in A} \rho(a)\ \sigma_a^{\#}\hat{\rho}(B)$$

Here and elsewhere, the *pushforward measure* $f^{\#}\hat{\rho}$ with respect to a continuous map f is defined by $f^{\#}\hat{\rho}(B) = \hat{\rho}(f^{-1}(B))$ for any Borel subset B of (M, d).

We can view the specification ρ of the invariant measure $\hat{\rho}$ as a one-state Markov process with a loop labelled with each letter from A, similar to how self-similar sets are specified with a one-state productive LTS. We can adapt this construction to multiple states by moving from probability distributions on A to *labelled Markov chains*, where again, the labels are interpreted as contraction maps.

Labelled Markov Chains

Let $\mathcal{D}$ denote the finitely supported probability distribution functor on the category of sets.

▶ **Definition 5.1.** *A* labelled Markov chain *(LMC) is a pair (X, β) consisting of a set X of* states *and a function $\beta\colon X \to \mathcal{D}(A \times X)$. A* homomorphism *of LMCs $h\colon (X, \beta_X) \to (Y, \beta_Y)$ is a function $h : X \to Y$ such that $\mathcal{D}(h) \circ \beta_X = \beta_Y \circ h$. We write $x \xrightarrow{r|a}_\beta y$ if $\beta(x)(a, y) = r$, often dropping the symbol β if it is clear form context.*

As we have already seen, given a contraction operator interpretation $\sigma\colon A \to \mathrm{Con}(M, d)$, every state x of a productive LTS (X, α) with labels in A corresponds to a regular subfractal $[\![x]\!]_{\alpha,\sigma}$ of $\mathbf{S}_\sigma$. This regular subfractal is defined to be the continuous image of the set $\mathrm{str}(x)$ under the map $\sigma_\omega\colon (A^\omega, d_\sigma) \to (M, d)$, where d_σ is determined by the contraction coefficients of the σ_a's as follows: Given a nonzero contraction coefficient c_a of σ_a for each $a \in A$, define $d_\sigma((a_1, \dots), (b_1, \dots)) = \prod_{i=1}^{n} c_{a_i}$, where n is the least index such that $a_{n+1} \neq b_{n+1}$. The family $[\![x]\!]_{\alpha,\sigma}$ is characterized by its satisfaction of the equations representing the LTS (X, α).

Every LMC (X, β) has an *underlying* LTS $(X, \bar{\beta})$, where $\bar{\beta}(x) = \{(a, y) \mid \beta(x)(a, y) > 0\}$. For each $x \in X$, we are going to define a probability measure $\hat{\beta}_\sigma(x)$ on $\mathbf{S}_\sigma$ whose support is $[\![x]\!]_{\bar{\beta},\sigma}$, and that satisfies a recursive system of equations represented by the LMC (X, β). Roughly, $\hat{\beta}_\sigma(x)$ is the pushforward of a certain Borel probability measure $\hat{\beta}(x)$ on A^ω that does not depend on the contraction operator interpretation σ.

We begin by topologizing A^ω, using as a basis the sets of the form

$$B_{a_1\cdots a_n} = \{(a_1, \dots, a_n, b_1, \dots) \mid (b_1, \dots) \in A^\omega\}$$

Given a state x of a LMC (X, β) and a word $w = a_1 \cdots a_n$, we follow Kerstan and König [14] and define the *trace measure* of the basic open set B_w by

$$\hat{\beta}(x)(B_w) = \sum\{r_1 \cdots r_n \mid x \xrightarrow{r_1|a_1} x_1 \to \cdots \xrightarrow{r_n|a_n} x_n\} \tag{5}$$

where $\hat{\beta}(B_\epsilon) = \hat{\beta}(A^\omega) = 1$. This defines a unique Borel probability measure on (A^ω, d).

▶ **Proposition 5.2.** *Let $j \colon A^* \to [0, 1]$ satisfy $j(w) = \sum_{a\in A} j(wa)$ for any $w \in A^*$ and $j(\epsilon) = 1$, where ϵ is the empty word. Then there is a unique Borel probability measure ρ on (A^ω, d) such that for any $w \in A^*$, $\rho(B_w) = j(w)$.*

Proof. This is an easy consequence of the Identity and Extension Theorems for σ-finite premeasures. See Propositions 2.3 to 2.5 of [14]. ◀

In particular, given any LMC (X, β), $\hat{\beta}(x)(B_w) = \sum_{a\in A} \hat{\beta}(x)(B_{wa})$, so there is a unique Borel probability measure $\hat{\beta}(x)$ on A^ω such that (5) holds for any basic open set B_w.

▶ **Definition 5.3.** *Let (X, β) be a LMC, and $\sigma \colon A \to \mathrm{Con}(M, d)$ be a contraction operator interpretation in a complete metric space. For each $x \in X$, we define the* regular subfractal measure *corresponding to x to be $\hat{\beta}_\sigma(x) = \sigma_\omega^\# \hat{\beta}(x)$.*

Intuitively, the regular subfractal measure of a state in a LMC under a contraction operator interpretation computes the probability that, if run stochastically according to the probabilities labelling the edges, the sequence of points of M observed in the run eventually lands within a given Borel subset of (M, d).

Systems of Probabilistic Equations

Given a complete metric space (M, d), let $\mathbf{P}(M, d)$ be the set of Borel probability measures on (M, d). In previous sections, we made use of the fact that, when $\sigma \colon A \to \mathrm{Con}(M, d)$, we can see $\mathbf{K}(M, d)$ as a *semilattice with operators*, i.e., union acts as a binary operation $\cup\colon \mathbf{K}(M, d)^2 \to \mathbf{K}(M, d)$ and each $\sigma_a \colon \mathbf{K}(M, d) \to \mathbf{K}(M, d)$ distributes over $\cup$. Analogously, equipped with $\sigma \colon A \to \mathrm{Con}(M, d)$, $\mathbf{P}(M, d)$ is a *convex algebra with operators*. Formally, for any $r \in [0, 1]$, there is a binary operation $\oplus_r \colon \mathbf{P}(M, d)^2 \to \mathbf{P}(M, d)$ defined $(\rho_1 \oplus_r \rho_2)(B) = r\rho_1(B) + (1 - r)\rho_2(B)$, over which each $\sigma_a^\#$ distributes, i.e.,

$$\sigma_a^\#(\rho_1 \oplus_r \rho_2) = \sigma_a^\# \rho_1 \oplus_r \sigma_a^\# \rho_2$$

We also make use of a summation notation defined by

$$r_1 \cdot \rho_1 \oplus \cdots \oplus r_n \cdot \rho_n = \rho_n \oplus_{r_n} \left(\frac{r_1}{1 - r_n} \cdot \rho_i \oplus \cdots \oplus \frac{r_{n-1}}{1 - r_n} \cdot \rho_i \right)$$

for any $r_1, \dots, r_n \in [0, 1)$.

Given a contraction operator interpretation, an LMC (X, β) can be thought of as a system of equations with one side a polynomial term in a convex algebra with operators,

$$x_i = r_{i1} \cdot a_{i1} x_{k_1} \oplus r_{i2} \cdot a_{i2} x_{k_2} \oplus \cdots \oplus r_{im} \cdot a_{im} x_{k_m}$$

where $X = \{x_1, \dots, x_n\}$ and $x_i \xrightarrow{r_{ij}|a_{ij}} x_{k_m}$ for each $i, j \leq m$.

▶ **Definition 5.4.** *Let (X, β) be a LMC, and let $\sigma \colon A \to \mathrm{Con}(M, d)$. A* solution *to (X, β) is a function $\varphi \colon X \to \mathbf{P}(M, d)$ such that for any $x \in X$ and any Borel set B,*

$$\varphi(x)(B) = \sum_{x \xrightarrow{r|a} y} r \ \sigma_a^{\#}(\varphi(y))(B)$$

Every finite LMC admits a unique solution, and moreover, the unique solution is the regular subfractal measure from Definition 5.3.

▶ **Theorem 5.5.** *Let (X, β) be a LMC, $x \in X$, and $\sigma \colon A \to \mathrm{Con}(M, d)$. Then the map $\hat{\beta}_\sigma \colon X \to \mathbf{P}(M, d)$ is the unique solution to (X, β).*

Since the support of $\hat{\beta}(x)$ is precisely $\mathrm{str}(x)$, the support of $\hat{\beta}_\sigma(x)$ is precisely $\sigma_\omega(\mathrm{str}(x))$, which we have already seen is the regular subfractal determined by the state x of the underlying LTS of (X, β).

Probabilistic Process Algebra

Finally, we introduce a syntax for specifying LMCs. Our specification language is essentially the *productive* fragment of Stark and Smolka's process calculus [27], meaning that the expressions do not involve deadlock and all variables are guarded.

▶ **Definition 5.6.** *The set of* probabilistic terms *is given by the grammar*

$$v \mid ae \mid e_1 \oplus_r e_2 \mid \mu v\ e$$

Here $r \in [0, 1]$, and otherwise we make the same stipulations as in Definition 2.3. The set of probabilistic process terms PTerm *consists of the closed and guarded probabilistic terms.*

Instead of *languages* of streams, the analog of trace semantics appropriate for probabilistic process terms is a measure-theoretic semantics consisting of trace measures introduced earlier in this section (Equation (5)).

▶ **Definition 5.7.** *We define the LMC* (PTerm, δ) *in Figure 6 and call it the* syntactic LMC*. The* trace measure semantics $\mathrm{trm}(e)$ *of a probabilistic process term e is defined to be* $\mathrm{trm}(e) = \hat{\delta}(x)$*. Given $\sigma \colon A \to \mathrm{Con}(M, d)$, the* subfractal semantics of $e \in \mathsf{PTerm}$ corresponding to σ *is $\hat{\delta}_\sigma(e)$.*

Intuitively, the trace measure semantics of a process term e assigns a Borel set of streams B the probability that e eventually emits a word in B. Trace measure semantics can be computed inductively as follows.

▶ **Lemma 5.8.** *For any $w \in A^*$, $a \in A$, $e, e_i \in \mathsf{PTerm}$, and $r \in [0, 1]$, $\mathrm{trm}(e)(A^\omega) = 1$ and*

$$\mathrm{trm}(ae)(B_w) = \begin{cases} \mathrm{trm}(e)(B_u) & w = au \\ 0 & \textit{otherwise} \end{cases}$$

$$\mathrm{trm}(e_1 \oplus_r e_2)(B_w) = r\,\mathrm{trm}(e_1)(B_w) + (1 - r)\,\mathrm{trm}(e_2)(B_w)$$

$$\mathrm{trm}(\mu v\ e)(B_w) = \mathrm{trm}(e[\mu v\ e/v])(B_w)$$

$$\delta(ae)(b,f) = \begin{cases} 1 & f = e \text{ and } b = a \\ 0 & \text{otherwise} \end{cases}$$
$$\delta(e_1 \oplus_r e_2)(b,f) = r\delta(e_1)(b,f) + (1-r)\delta(e_2)(b,f)$$
$$\delta(\mu v\ e)(b,f) = \delta(e[\mu v\ e/v])(b,f)$$

Figure 6 The LMC structure (PTerm, δ). Above, $a, b \in A$, $\sum r_i = 1$, and $e, e_i, f \in \mathsf{PTerm}$.

$$\begin{array}{lc}
(\mathsf{ID}) & e \oplus_r e \equiv e \\
(\mathsf{CM}) & e_1 \oplus_r e_2 \equiv e_2 \oplus_{1-r} e_1 \\
(\mathsf{AS}) & (e_1 \oplus_r e_2) \oplus_s e_3 \equiv e_1 \oplus_{rs} (e_2 \oplus_{\frac{s(1-r)}{1-rs}} e_3) \\
(\mathsf{DS}) & a(e_1 \oplus_r e_2) \equiv ae_1 \oplus_r ae_2 \\
(\mathsf{FP}) & \mu v\ e \equiv e[\mu v\ e/v]
\end{array}
\qquad
\begin{array}{lc}
(\mathsf{CN}) & \dfrac{(\forall i)\ e_i \equiv f_i}{g[\vec{e}/\vec{v}] \equiv g[\vec{f}/\vec{v}]} \\
(\mathsf{AE}) & \dfrac{}{\mu w\ e \equiv \mu v\ e[v/w]} \\
(\mathsf{UA}) & \dfrac{g \equiv e[g/v]}{g \equiv \mu v\ e}
\end{array}$$

Figure 7 Axioms for probabilistic trace equivalence. Above, $e, e_1, e_2 \in \mathsf{PTerm}$, $a \in A$, $r, s \in [0,1]$, and $rs \neq 1$. Also, in (AE), v is not free in e.

Similar to the situation with trace semantics and regular subfractals, trace measure semantics and subfractal measure semantics identify the same probabilistic process terms.

▶ **Theorem 5.9.** *Let $e, f \in \mathsf{PTerm}$. Then $\mathrm{trm}(e) = \mathrm{trm}(f)$ if and only if for any contraction operator interpretation $\sigma\colon A \to \mathrm{Con}(M,d)$, $\hat{\hat{\delta}}_\sigma(e) = \hat{\hat{\delta}}_\sigma(f)$.*

Axiomatization

Figure 7 outlines an inference system for determining when the subfractal measures corresponding to two expressions coincide.

▶ **Definition 5.10.** *Given $e, f \in \mathsf{PTerm}$, write $e \equiv f$ and say that e and f are* provably equivalent *if the equation $e = f$ can be derived from inference rules in Figure 7.*

▶ **Theorem 5.11** (Soundness). *For any $e, f \in \mathsf{PTerm}$, if $e \equiv f$, then for any complete metric space (M,d) and any $\sigma\colon A \to \mathrm{Con}(M,d)$, $\hat{\hat{\delta}}_\sigma(e) = \hat{\hat{\delta}}_\sigma(f)$.*

Unlike the situation with trace equivalence, it is not known if these axioms are complete with respect to subfractal measure semantics. We leave this as a conjecture.

▶ **Conjecture 5.12** (Completeness). *Figure 7 is a complete axiomatization of trace measure semantics. That is, for any $e, f \in \mathsf{PTerm}$, if for any complete metric space (M,d) and any $\sigma\colon A \to \mathrm{Con}(M,d)$ we have $\hat{\hat{\delta}}_\sigma(e) = \hat{\hat{\delta}}_\sigma(f)$, then $e \equiv f$.*

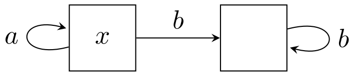
We expect that Conjecture 5.12 can be proven in a similar manner to Theorem 4.4.

6 A Question about Regular Subfractals

Certain regular subfractals that have been generated by LTSs with multiple states happen to coincide with self-similar sets using a different alphabet of action symbols and under a different contraction operator interpretation. For example, the twisted Sierpiński gasket in Figure 2 is the self-similar set generated by the iterated function system consisting of the compositions $\sigma_a, \sigma_b\sigma_b, \sigma_b\sigma_c, \sigma_c\sigma_b$, and $\sigma_c\sigma_c$.

▶ **Question 1.** Is every regular subfractal a self-similar set? In other words, are there regular subfractals that can only be generated by a multi-state LTS?

▶ **Example 6.1.** To illustrate the subtlety of this question, consider the following LTS.



The state x emits $(a, a, \ldots)$ (an infinite stream of a's) and $(a, \ldots, a, b, b, \ldots)$, a stream with some finite number (possibly 0) of a's followed by an infinite stream of b's. Now let $M = \mathbb{R}$ with Euclidean distance and consider the contraction operator interpretation $\sigma_a(r) = \frac{1}{2}r$ and $\sigma_b(r) = \frac{1}{2}r + \frac{1}{2}$. Let $K = \{0\} \cup \{\frac{1}{2^n} | n \geq 0\}$. Then K is the component of the solution at x. This example is interesting because unlike the Twisted Sierpiński gasket in Figure 2, there is no obvious finite set of compositions σ_a and σ_b such that K is the self-similar set generated by that iterated function system.

There is an LTS (X, α) with X a singleton set $\{x\}$, and a contraction operator interpretation σ_x whose solution is K. We take the set of action labels underlying X to be $B = \{f, g, h\}$ and use the contraction operator interpretation $\sigma_f(r) = 0$, $\sigma_g(r) = 1$ and $\sigma_h(r) = \frac{1}{2}r$. It is easy to verify that $K = \bigcup_{i \in \{f,g,h\}} \sigma_i(K)$.

But we claim that K is not obtainable using a single-state LTS and *the same contractions* $\sigma_a(r) = \frac{1}{2}r$ and $\sigma_b(r) = \frac{1}{2}r + \frac{1}{2}$, or using *any (finite) compositions* of σ_a and σ_b. Indeed, suppose there were such a finite collection $\sigma_1, \ldots \sigma_n$ consisting of (finite) compositions of σ_a and σ_b such that $K = \bigcup_{i=1}^{n} \sigma_i(K)$. Since $1 \in K$, we must be using the stream $(b, b, b, \ldots)$ (since if there is an a at position n, the number obtained would be $\leq 1 - \frac{1}{2^n} < 1$), so some σ_i must consist of a composition of σ_b some number $m \geq 1$ of times with itself. Similarly, the only way to obtain 0 is with $(a, a, a, \ldots)$, so there must be some σ_j which is a composition of σ_a some number of times $p \geq 1$ with itself. But then $\lim_{n\to\infty} \sigma_i \circ \sigma_j \circ \sigma_i^n(r) = 1 - (\frac{2^p - 1}{2^{m+p}}) > \frac{1}{2}$, since $m, p \geq 1$. That point must be in the subset of $\mathbb{R}$ generated by this LTS. However, it is not in K, since $\frac{1}{2} < 1 - (\frac{2^p-1}{2^{m+p}}) < 1$.

More generally, we cannot obtain K using a single-state LTS even if we allowed finite sums of compositions of σ_a and σ_b.

Once again, it is possible to find a single state LTS whose corresponding subset of $\mathbb{R}$ is K, but to do this we needed to *change the alphabet and also the contractions*. Perhaps un-coincidentally, the constant operators are exactly the limits of the two contractions from the original interpretation. Our question is whether this can always be done.

On the other hand, the thesis of Boore [7] may contain an answer to Question 1. Boore presents a (family) of 2-state GIFS whose *attractors*, total unions of their regular subfractals, are not self-similar. Attractors of GIFSs are not precisely the same as regular subfractals, so additional work is required to adapt Boore's work to answer Question 1.

7 Related Work

This paper is part of a larger effort of examining topics in continuous mathematics from the standpoint of coalgebra and theoretical computer science. The topic itself is quite old, and originates perhaps with Pavlovic and Escardó's paper "Calculus in Coinductive Form" [23]. Another early contribution is Pavlovic and Pratt [24]. These papers proposed viewing some structures in continuous mathematics – the real numbers, for example, and power series expansions – in terms of final coalgebras and streams. The next stage in this line of work was a set of papers specifically about fractal sets and final coalgebras. For example, Leinster [16] offered a very general theory of self-similarity that used categorical modules in connection with the kind of gluing that is prominent in constructions of self-similar sets. In a different direction, papers like [4] showed that for some very simple fractals (such as the Sierpiński gasket treated here), the final coalgebras were Cauchy completions of the initial algebras.

Generalizations of IFSs. Many generalizations of Hutchinson's self-similar sets have appeared in the literature. The generalization that most closely resembles our own is that of an attractor for a *directed-graph iterated function system* (GIFS) [18]. A LTS paired with a contraction operator interpretation is equivalent data to that of a GIFS, and equivalent statements to Lemma 3.7 can be found for example in [9, 18, 19]. As opposed to the regular subfractal corresponding to one state, as we have studied above, the geometric object studied in the GIFSs literature is typically the union of the regular subfractals corresponding to all the states (in our terminology), and properties such as Hausdorff dimension and connectivity are emphasized. We also distinguish our structures from GIFSs because we need to allow the interpretations of the labels to vary in our semantics.

Another generalization is Mihail and Miculescu's notion of attractor for a *generalized iterated function system* [19]. A generalized IFS is essentially that of Hutchinson's IFS with multi-arity contractions – equivalent to a single-state labelled transition system where labels have "higher arity". A common generalization of GIFSs and generalized IFSs could be achieved by considering coalgebras of the form $X \to \mathcal{P}(\coprod_{n\in\mathbb{N}} A_n \times X^n)$ and interpreting each $a \in A_n$ as an n-ary contraction. We suspect that a similar story to the one we have outlined in this paper is possible for this common generalization.

Process algebra. The process terms we use to specify labelled transition systems and labelled Markov chains are fragments of known specification languages. Milner used process terms to specify LTSs in [22], and we have repurposed his small-step semantics here. Stark and Smolka use probabilistic process terms to specify labelled Markov chains (in our terminology) in [27], and we have used them for the same purpose. Both of these papers also include complete axiomatizations of bisimilarity, and we have also repurposed their axioms.

However, fractal semantics is strictly coarser than bisimilarity, and in particular, bisimilarity of process terms is trace equivalence. Rabinovich added a single axiom to Milner's axiomatization to obtain a sound and complete axiomatization of trace equivalence of expressions [25], which allowed us to derive Theorem 4.4. In contrast, the axiomatization of trace equivalence for probabilistic processes is only well-understood for *finite* traces, see Silva and Sokolova's [26], which our probabilistic process terms do not exhibit. We use the trace semantics of Kerstan and König [14] because it takes into account infinite traces. Infinite trace semantics has yet to see a complete axiomatization in the literature.

Other types of syntax. In this paper, we used the specification language of μ-terms as our basic syntax. As it happens, there are two other flavors of syntax that we could have employed. These are *iteration theories* [5], and terms in the Formal Language of Recursion FLR, especially its FLR_0 fragment. The three flavors of syntax for fixed point terms are compared in a number of papers: In [12], it was shown that there is an equivalence of categories between FLR_0 structures and iteration theories, and Bloom and Ésik make a similar connection between iteration theories and the μ-calculus in [6]. Again, these results describe general matters of equivalence, but it is not completely clear that for a specific space or class of spaces that they are equally powerful or equally convenient specification languages. We feel this matter deserves some investigation.

Equivalence under hypotheses. A specification language fairly close to iteration theories was used by Milius and Moss to reason about fractal constructions in [21] under the guise of *interpreted solutions* to recursive program schemes [20]. Moreover, [21] contains important examples of reasoning about the equality of fractal sets under assumptions about the contractions. Based on the general negative results on reasoning from hypotheses in the logic of recursion [12], we would not expect a completeness theorem for fractal equivalence under hypotheses. However, we do expect to find sound logical systems which account for interesting phenomena in the area.

8 Conclusion

This paper connects fractals to trace semantics, a topic originating in process algebra. This connection is our main contribution, because it opens up a line of communication between two very different areas of study. The study of fractals is a well-developed area, and like most of mathematics it is pursued without a special-purpose specification language. When we viewed process terms as recipes for fractals, we provided a specification language that was not present in the fractals literature. Of course, one also needs a contraction operator interpretation to actually define a fractal, but the separation of syntax (the process terms) and semantics (the fractals obtained using contraction operator interpretations of the syntax) is something that comes from the tradition of logic and theoretical computer science. Similarly, the use of a logical system and the emphasis on soundness and completeness is a new contribution here.

All of the above opens questions about fractals and their specifications. Our most concrete question was posed in Section 6. We would also like to know if we can obtain completeness theorems allowing for extra equations in the axiomatization. Lastly, and most speculatively, since LTSs (and other automata) appear so frequently in decision procedures from process algebra and verification, we would like to know if our semantics perspective on fractals can provide new complexity results in fractal geometry.

We hope we have initiated a line of research where questions and answers come from both the analytic side and from theoretical computer science.

References

1 Jos C. M. Baeten. A brief history of process algebra. *Theor. Comput. Sci.*, 335(2-3):131–146, 2005. doi:10.1016/j.tcs.2004.07.036.

2 Stefan Banach. Sur les opérations dans les ensembles abstraits et leur application aux équations intégrales. *Fundamenta Mathematicae*, 3:133–181, 1922.

3 Jan A. Bergstra, Alban Ponse, and Scott A. Smolka, editors. *Handbook of Process Algebra*. North-Holland / Elsevier, 2001. doi:10.1016/b978-0-444-82830-9.x5017-6.

4 Prasit Bhattacharya, Lawrence S. Moss, Jayampathy Ratnayake, and Robert Rose. Fractal sets as final coalgebras obtained by completing an initial algebra. In Franck van Breugel, Elham Kashefi, Catuscia Palamidessi, and Jan Rutten, editors, *Horizons of the Mind. A Tribute to Prakash Panangaden - Essays Dedicated to Prakash Panangaden on the Occasion of His 60th Birthday*, volume 8464 of *Lecture Notes in Computer Science*, pages 146–167. Springer, 2014.

5 Stephen L. Bloom and Zoltán Ésik. *Iteration Theories - The Equational Logic of Iterative Processes*. EATCS Monographs on Theoretical Computer Science. Springer, 1993. doi:10.1007/978-3-642-78034-9.

6 Stephen L. Bloom and Zoltán Ésik. Solving polynomial fixed point equations. In *Mathematical foundations of computer science 1994 (Košice, 1994)*, volume 841 of *Lecture Notes in Comput. Sci.*, pages 52–67. Springer, Berlin, 1994.

7 Graeme C. Boore. *Directed graph iterated function systems.* PhD thesis, University of St. Andrews, 2011. URL: `http://hdl.handle.net/10023/2109`.

8 G. A. Edgar and R. Daniel Mauldin. Multifractal decompositions of digraph recursive fractals. *Proceedings of the London Mathematical Society*, s3-65(3):604–628, 1992. `doi:10.1112/plms/s3-65.3.604`.

9 Gerald A. Edgar. *Measure, Topology, and Fractal Geometry.* Springer New York, NY, 1 edition, 1990. `doi:10.1007/978-1-4757-4134-6`.

10 Kenneth J. Falconer. *The Geometry of Fractal Sets.* Cambridge Tracts in Mathematics. Cambridge University Press, 1986. URL: `https://www.cambridge.org/us/academic/subjects/mathematics/abstract-analysis/geometry-fractal-sets`.

11 C. A. R. Hoare. Communicating sequential processes. *Commun. ACM*, 21(8):666–677, 1978. `doi:10.1145/359576.359585`.

12 A. J. C. Hurkens, Monica McArthur, Yiannis N. Moschovakis, Lawrence S. Moss, and Glen T. Whitney. Erratum: "The logic of recursive equations". *J. Symbolic Logic*, 64, 1999.

13 John E. Hutchinson. Fractals and self similarity. *Indiana University Mathematics Journal*, 30(5):713–747, 1981. URL: `http://www.jstor.org/stable/24893080`.

14 Henning Kerstan and Barbara König. Coalgebraic Trace Semantics for Continuous Probabilistic Transition Systems. *Logical Methods in Computer Science*, Volume 9, Issue 4, December 2013. `doi:10.2168/LMCS-9(4:16)2013`.

15 S. C. Kleene. Representation of events in nerve nets and finite automata. In Claude Shannon and John McCarthy, editors, *Automata Studies*, pages 3–41. Princeton University Press, Princeton, NJ, 1956.

16 Tom Leinster. A general theory of self-similarity. *Adv. Math.*, 226(4):2935–3017, 2011.

17 Benoit B. Mandelbrot. *Fractals: Form, Chance, and Dimension.* Mathematics Series. W. H. Freeman, 1977. URL: `https://books.google.com/books?id=avw_AQAAIAAJ`.

18 R Daniel Mauldin and S. C. Williams. Hausdorff dimension in graph directed constructions. *Transactions of the American Mathematical Society*, 309(2):811–829, 1988. `doi:10.1090/S0002-9947-1988-0961615-4`.

19 Alexandru Mihail and Radu Miculescu. Generalized ifss on noncompact spaces. *Fixed Point Theory and Applications*, 1(584215), 2010. `doi:10.1155/2010/584215`.

20 Stefan Milius and Lawrence S. Moss. The category-theoretic solution of recursive program schemes. *Theor. Comput. Sci.*, 366(1-2):3–59, 2006. `doi:10.1016/j.tcs.2006.07.002`.

21 Stefan Milius and Lawrence S. Moss. Equational properties of recursive program scheme solutions. *Cah. Topol. Géom. Différ. Catég.*, 50(1):23–66, 2009.

22 Robin Milner. A complete inference system for a class of regular behaviours. *J. Comput. Syst. Sci.*, 28(3):439–466, 1984. `doi:10.1016/0022-0000(84)90023-0`.

23 Dusko Pavlovic and Martín Hötzel Escardó. Calculus in coinductive form. In *Thirteenth Annual IEEE Symposium on Logic in Computer Science, Indianapolis, Indiana, USA, June 21-24, 1998*, pages 408–417. IEEE Computer Society, 1998.

24 Dusko Pavlovic and Vaughan R. Pratt. The continuum as a final coalgebra. *Theor. Comput. Sci.*, 280(1-2):105–122, 2002.

25 Alexander Moshe Rabinovich. A complete axiomatisation for trace congruence of finite state behaviors. In Stephen D. Brookes, Michael G. Main, Austin Melton, Michael W. Mislove, and David A. Schmidt, editors, *Mathematical Foundations of Programming Semantics, 9th International Conference, New Orleans, LA, USA, April 7-10, 1993, Proceedings*, volume 802 of *Lecture Notes in Computer Science*, pages 530–543. Springer, 1993. `doi:10.1007/3-540-58027-1_25`.

26 Alexandra Silva and Ana Sokolova. Sound and complete axiomatization of trace semantics for probabilistic systems. In Michael W. Mislove and Joël Ouaknine, editors, *Twenty-seventh Conference on the Mathematical Foundations of Programming Semantics, MFPS 2011, Pittsburgh, PA, USA, May 25-28, 2011*, volume 276 of *Electronic Notes in Theoretical Computer Science*, pages 291–311. Elsevier, 2011. `doi:10.1016/j.entcs.2011.09.027`.

27 Eugene W. Stark and Scott A. Smolka. A complete axiom system for finite-state probabilistic processes. In Gordon D. Plotkin, Colin Stirling, and Mads Tofte, editors, *Proof, Language, and Interaction, Essays in Honour of Robin Milner*, pages 571–596. The MIT Press, 2000.