ELSEVIER

Physica D 155 (2001) 1–33

PHYSICA D

www.elsevier.com/locate/physd

Inaccessibility and undecidability in computation, geometry, and dynamical systems

Asaki Saito [a,*], Kunihiko Kaneko [b]

[a] *Laboratory for Mathematical Neuroscience, Brain Science Institute, RIKEN, 2-1 Hirosawa, Wako, Saitama 351-0198, Japan*
[b] *Department of Pure and Applied Sciences, University of Tokyo, Tokyo 153-8902, Japan*

Received 13 October 1999; received in revised form 5 February 2001; accepted 5 February 2001

Communicated by Y. Kuramoto

Abstract

Non-self-similar sets defined by a decision procedure are numerically investigated by introducing the notion of inaccessibility to (ideal) decision procedure, that is connected with undecidability. A halting set of a universal Turing machine (UTM), the Mandelbrot set and a riddled basin are mainly investigated as non-self-similar sets with a decision procedure. By encoding a symbol sequence to a point in a Euclidean space, a halting set of a UTM is shown to be geometrically represented as a non-self-similar set, having different patterns and different fine structures on arbitrarily small scales. The boundary dimension of this set is shown to be equal to the space dimension, implying that the ideal decision procedure is inaccessible in the presence of error. This property is shown to be invariant under application of "fractal" code transformations. Thus, a characterization of undecidability is given by the inaccessibility to the ideal decision procedure and its invariance against the code transformations. It is also shown that the distribution of halting time of the UTM, decays with a power law (or slower), and that this characteristic is also unchanged under code transformation. The Mandelbrot set is shown to have these features including the invariance against the code transformation, in common, and is connected with undecidable sets. In contrast, although a riddled basin, as a geometric representation of a certain context-free language, has the boundary dimension equal to the space dimension and a power law halting time distribution, these properties are not invariant against the code transformation. Thus, the riddled basin is ranked as middle between an ordinary fractal and a halting set of a UTM or the Mandelbrot set. Last, we propose undecidability for analog computation, and discuss the appropriate condition for coding.

PACS: 89.80.+h; 05.45.+b

Keywords: Inaccessible decision procedure; Undecidability; Boundary dimension; Non-self-similar set; Code transformation

1. Introduction

For defining sets, recursive procedures are generally utilized. Such recursive procedures can roughly be divided into two kinds, one is a generating procedure (generator) which generates elements of a set, and the other is a decision procedure (recognizer) which decides whether a given object is included in a set. In this paper, we focus our attention on sets defined by the latter. As a set defined by a decision procedure (i.e., a halting set of a decision procedure), we consider the following three sets from different category:

* Corresponding author. Tel.: +81-48-467-9780; fax: +81-48-467-9693.
E-mail addresses: saito@brain.riken.go.jp (A. Saito), kaneko@complex.c.u-tokyo.ac.jp (K. Kaneko).

0167-2789/01/$ – see front matter
PII: S0167-2789(01)00232-9

- a halting set of an automaton treated in classical computation theory; [1]
- a geometric set defined by a decision procedure;
- a basin of attraction treated in dynamical systems study. [2]

Although these three sets are different in their origin, they are similarly defined according to procedures of decision. So far, the halting sets have not been studied from this unified standpoint on decision procedures. In this paper, by using the same methods, we study various halting sets of a decision procedure in computation theory, geometry and dynamical systems study.

1.1. Computation

In classical computation theory [1–4], halting sets of an automaton (decision procedure) with various computational power are treated, which include regular set, context-free language and recursively enumerable set, and so forth [2]. Basically, these objects are studied only by means of proving theorems. Thanks to this approach, rigorous mathematical theory was made, but this theory is often hard to be understood intuitively. [3] Relation with other fields is also not made clear.

On the other hand, in dynamical systems study, basins (halting sets) and transient processes (decision processes) to an attractor are studied extensively and intensively. Indeed some analytic approaches are developed for these studies, but their application is limited to some ideal cases. For dynamical systems studies, experimental approaches with numerical simulation using digital computers are generally adopted and are effective, in contrast with rare use of these approaches in classical computation theory.

Noticing the unified standpoint of decision procedures, we study geometric properties of a halting set of automaton by mapping a symbol sequence to a point in a Euclidean space, and investigate dynamical systems characteristics of a decision procedure determined by an automaton.

In this work, we aim to re-interpret computation from a viewpoint of dynamical systems, by positively adopting experimental approaches with numerical simulations. This study is based on the recognition that both the halting sets of automaton in computation theory and the basins in dynamical systems study have a common feature as halting sets of decision procedure.

1.2. Fractal geometry and nonlinear dynamical systems

In fractal geometry [5–7], sets having infinitely fine structures are treated. However, these sets are studied basically from a viewpoint of self-similarity. Among sets having infinitely fine structures, however, there exist not only self-similar sets but non-self-similar sets that cannot simply be characterized by self-similarity. For example, there are sets having a "different" fine structure on an arbitrarily small scale, treated in detail in the following sections. Many interesting sets are included in the non-self-similar sets. However, non-self-similar sets have been scarcely studied so far, as compared with self-similar sets (i.e., ordinary fractals).

Likewise, in dynamical systems study [8–12], systems with chaotic dynamics are often studied which have exponential orbital instability. However, there exist also interesting nonlinear dynamical systems that cannot simply be characterized by chaotic behavior [13], while these dynamical systems have been scarcely studied, as compared with ordinary chaos.

On the other hand, as will be mentioned in Section 3, geometric representation of the halting set of the universal Turing machine, [4] obtained by encoding a symbol sequence to a point in a Euclidean space, is shown to be a non-self-similar set, having a different fine structure on an arbitrarily small scale (see also [14]). In addition, a dynamical system in which a Turing machine is embedded generally has a

[1] A given symbol string is decided to be included in the set when the automaton with that symbol string goes to a halting state.

[2] A given initial point is decided to be included in the set when the dynamical system with that initial point converges to an attractor.

[3] For example, overuse of reduction to absurdity.

[4] See Section 3 for the brief explanation, and also [2–4], for example.

different instability other than chaotic instability, and is shown to be qualitatively different from chaos (see also [15–17]).

In this paper, we study such non-self-similar sets, by treating general decision procedures not restricted within dynamical systems simply characterized by chaotic behavior. We try to explore a new type of geometric and dynamical systems properties simultaneously by re-interpreting computation.

1.3. Code

When we discuss properties of geometric sets or dynamical systems in which (a halting set of) an automaton is embedded, coding causes an inevitable problem. These geometric and dynamical systems properties must be discussed together with a code which maps a symbol sequence to a real number. So far, the problem of coding is not seriously discussed. Here we study a nature of codes explicitly.

1.4. Analog computation

Our study is related to analog computation since we do not restrict a decision procedure of a set to a discrete one.[5] Although several analog computation models have been studied so far (e.g. [18–22]), classes of computable real functions do not coincide with each other, in contrast with the case of classical discrete computation theory. Also, ordinary studies of analog computation (including the cited studies) start by constructing a model in the beginning, without explicit consideration on physical realizability. Only after the model construction, physical realizability, especially noise effects, is either considered only insufficiently, or not considered at all.

Here, our approach to analog computation does not start by constructing a model in the beginning, in contrast with ordinary approach. Instead, we aim to clarify a necessary condition for analog computation on the basis of above studies of dynamical systems and computation.[6] To propose the condition, physical realizability (noise effects) is seriously considered.

1.5. Overview

In the present paper, a halting set of a universal Turing machine from classical computation theory, the Mandelbrot set[7] from fractal geometry, and riddled basin structure[8] from nonlinear dynamical systems study are mainly investigated as concrete objects, all of which are defined by decision procedure. We will see both geometric properties of the sets and dynamical systems characteristics of decision procedures. We mainly study the boundary dimension for the former and halting time distribution for the latter. For studying these features, numerical approaches borrowed from nonlinear dynamical systems studies are adopted.

Especially we focus on the cases where dimension of boundary of a set asymptotically approaches (and is equal to) the dimension of space, and where halting time distribution of decision procedure of a set decays according to a power law or slower. When the boundary dimension is equal to the space dimension, the decision procedure of the set has so strong uncertainty that one cannot approach the ideal decision procedure, in the presence of error. Such system is qualitatively different from a system with a boundary dimension less than space dimension, or a system with an ordinary chaotic unpredictability. We survey this strong uncertainty and its behavior against the transformation of code.

This paper is organized as follows. In Section 2, we study the code that maps a symbol sequence to a real number, to prepare for the subsequent sections. There it is shown that the preservation of (boundary) dimension is generally not guaranteed under the transformation from one code to another.

In Section 3, we treat a halting set of a universal Turing machine, aiming to see how undecidability can

[5] Note that the concept of procedure is essentially "computational", where the meaning of "computation" is not necessarily restricted within classical one.

[6] Another different study of analog computation based on dynamical systems is presented in [23].

[7] See Section 4 for the brief explanation, and also [5,10,11], for example.

[8] See Section 5 for the brief explanation, and also [24,25], for example.

be characterized. The geometric representation of the halting set of a universal Turing machine, according to the code introduced in Section 2, is shown to have different patterns and have different fine structures on arbitrarily small scales. Furthermore, the boundary dimension of this set is shown to approach the space dimension, as the computation time is increased. In other words, the decision procedure of the set has the strong uncertainty mentioned above. In addition, this result is shown to be invariant against certain "fractal" code transformations. Thus, a characterization of the undecidability of the halting problem of a universal Turing machine is given by the strong uncertainty implying the inaccessibility to the ideal decision procedure, and by the invariance of the strong uncertainty against the code transformations. It is also shown that the distribution of halting time of the universal Turing machine, decays with a power law (or slower), and that this characteristic is also unchanged under code transformation.

In Section 4, we treat the Mandelbrot set to show similarity with the above halting set of a universal Turing machine. In particular, the Mandelbrot set has common properties with the halting sets of universal Turing machine, as for the boundary dimension and the halting time distribution. The invariance against the application of "fractal" function, which corresponds to a code transformation, is also common. Thus, the Mandelbrot set can be connected with undecidable sets from the standpoint of the strong uncertainty with the inaccessibility to the ideal decision procedure.

In Section 5, we treat riddled basin structure. Riddled basin of a certain simple dynamical system geometrically represents a certain context-free language. Concerning the boundary dimension, even though the set has the boundary dimension equal to the space dimension, this property of the boundary dimension is not invariant against a code transformation, in strong contrast with the halting sets of a universal Turing machine and the Mandelbrot set. Absence of the invariance is also shown concerning the halting time distributions of the riddled basin structure. Thus, the riddled basin is ranked as "middle" between ordinary fractals like Cantor set and a halting set of a universal Turing machine or the Mandelbrot set.

In Section 6, we study geometric and dynamical systems characteristics of converted (universal) Turing machines which are obtained by modifying a universal Turing machine. In particular, relation between modification and computational universality is pursued.

Following these results, we propose a novel concept for analog computation in Section 7, and discuss the appropriate condition for coding.

2. Code

As will be shown in the following, dynamical feature of an automaton (e.g., universal Turing machine and pushdown automaton [2]), as well as the geometric property of its halting set, generally depends on the way of embedding a symbol sequence to a number (i.e., code). In this section, we discuss briefly the problem of coding in connection with dynamical systems.

2.1. Code

In this paper, we use the word "code" as a mapping which transforms an infinite symbol string on a finite alphabet to a real number. Furthermore, "symbol string" indicates a finite symbol string, whereas "symbol sequence" indicates an infinite one.

Here, we restrict codes which map a symbol sequence to a real number represented by the same label of Markov partition [9] of a certain piecewise-linear map. To be concrete, a symbol sequence of n symbols is encoded into a real number on the interval [0, 1], by choosing a piecewise-linear map $f : [0, 1] \to [0, 1]$ defined by

$$f(x) = \frac{1}{\alpha_{i+1} - \alpha_i}(x - \alpha_i) \quad \text{for } x \in [\alpha_i, \alpha_{i+1}],$$

where $\alpha_0 = 0$, $\alpha_n = 1$ and $\alpha_i < \alpha_{i+1}$. Then, we encode a symbol sequence to a real number which has the same label of Markov partition of this piecewise-linear map, by labeling the interval between α_i, and α_{i+1} as 'i'. As a result, a finite symbol string, corresponding to the set of all symbol sequences with the prefix,

is encoded by this code, to an interval which has the same label of Markov partition. [9]

In particular, for a 2-symbol sequence ($n = 2$), the code has only one parameter α (i.e., α_1). When $\alpha = \frac{1}{2}$, this code corresponds to the binary encoding, using a symbol sequence as base-2 expansion of a real number.

2.2. Code transformation

Now we consider transformation of a code with 2-symbol sequences. Suppose a 2-symbol sequence **s** is chosen. Then, by using a code with parameter α (denoted as code α), one real number, denoted as x_α, is decided for **s**. Similarly, by using another code with parameter β (code β), another real number, x_β, is decided for **s**. By varying **s**, one can obtain a one-to-one mapping, denoted as $t_{\alpha\to\beta}$, from x_α to x_β. This $t_{\alpha\to\beta} : [0, 1] \to [0, 1]$ transforms encoding by code α into that by code β. Thus, we call such a mapping $t_{\alpha\to\beta}$ code transformation.

When $\alpha = \frac{1}{2}$, $t_{1/2\to\beta}$ becomes Lebesgue's singular function L_β [26,27], where Lebesgue's singular function $L_\beta : [0, 1] \to [0, 1]$ is defined by the functional equation,

$$L_\beta(\tfrac{1}{2}x) = \beta L_\beta(x),$$
$$L_\beta(\tfrac{1}{2}(1 + x)) = \beta + (1 - \beta)L_\beta(x),$$

where $0 < \beta < 1$. [10]

Fig. 1 shows $L_{1/3}$ (i.e., $t_{1/2\to1/3}$). When $\beta \neq \frac{1}{2}$, this "fractal" singular function L_β is shown to have following properties [26,27]:

- continuous;
- strictly monotone increasing from 0 to 1;
- the differential coefficient is 0 almost everywhere.

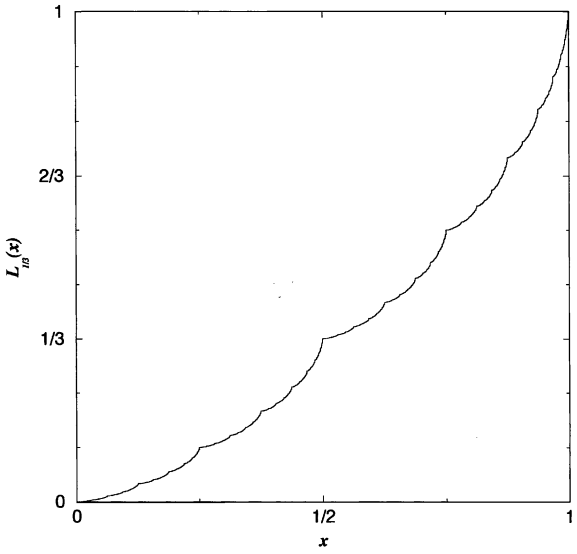

Fig. 1. Lebesgue's singular function $L_{1/3}$ (i.e., code transformation $t_{1/2\to1/3}$).

If $\beta = \frac{1}{2}$, solution of the functional equation is, of course, simply $L_{1/2}(x) = x$ which corresponds to the identity transformation.

Concerning the fractal (boundary) dimension [11] which is important in subsequent sections, the invariance of the (boundary) dimension under the fractal transformations is not guaranteed, whereas the (boundary) dimension is preserved under the application of diffeomorphism [28]. Not only in the case of the 2-symbol code, but generally the preservation of (boundary) dimension is not guaranteed under the transformation from one code to another.

For example, consider the middle-thirds Cantor set consisting of real numbers with no '1's in their base-3 expansion. In other words, it is obtained by encoding all symbol sequences on the alphabet $\{0, 2\}$, using the Markov partition of the map $f : [0, 1] \to [0, 1]$ [12]

[9] A code satisfying this condition is considered to be appropriate, but this condition may be too strong. The question what are the appropriate condition and class of codes remains unsolved, which will be discussed in Section 7. Here we do not treat other types of codes.

[10] Since encoding using code $\frac{1}{2}$ is binary encoding $\frac{1}{2}x$ corresponds to the shift of a 2-symbol sequence to right with the insertion of the symbol '0' at the left most cell. The real number for this symbol sequence using code β is obtained by multiplying the real number for the original symbol sequence using code β by β. Thus $t_{1/2\to\beta}$ satisfies the first equation. It is easy to check $t_{1/2\to\beta}$ also satisfies the second equation.

[11] We treat box-counting dimension and Hausdorff dimension as fractal dimension. The word "boundary dimension" is used as box-counting dimension or Hausdorff dimension of boundary of a set, unless otherwise mentioned.

[12] The set of symbol strings on the alphabet $\{0, 1, 2\}$, where the symbol strings end with the symbol '1' with no '1' before that '1', is, of course, encoded to the complement of the Cantor set with the base-3 code. This set of symbol strings is included in the class of regular sets, i.e., accepted by a finite automaton [2].

$f(x) = 3x \mod 1$.

The fractal dimension of this middle-thirds Cantor set is $\log 2/\log 3$. Since the Lebesgue measure of this set is zero, the boundary dimension of this set (also the boundary dimension of the complement of this set) is also $\log 2/\log 3$.

Then, if we transform the code using f to another 3-symbol code using the Markov partition of g

$$g(x) = \frac{1}{\alpha_{i+1} - \alpha_i}(x - \alpha_i) \quad \text{for } x \in [\alpha_i, \alpha_{i+1}],$$

where $\alpha_0 = 0$, $\alpha_3 = 1$ and $\alpha_i < \alpha_{i+1}$, the (boundary) dimension D of the resulting Cantor set is given by the solution of equation [7]

$$\alpha_1^D + (1 - \alpha_2)^D = 1.$$

Thus, we can change the (boundary) dimension D by changing α_1 and α_2 (e.g., if $\alpha_1 = \frac{1}{4}$ and $\alpha_2 = \frac{3}{4}$, then $D = \frac{1}{2}$). Hence for a Cantor set, the (boundary) dimension is not preserved under the above code transformation.

3. Halting set of universal Turing machine

In this section, by using codes introduced in the previous section, we investigate both geometric properties of halting set of a universal Turing machine and dynamical systems characteristics of its decision procedure, to characterize undecidability.

3.1. Turing machine

First, we briefly describe the Turing machine [1–4]. The Turing machine (TM) is the most important model of computation, and is also an abstract model of a digital computer.

A TM basically consists of a finite control, a tape and a tape head. A finite control has finite states $\{q_1, q_2, \ldots, q_m\}$. A tape is one-dimensional, and consists of infinite cells on which one of the symbols in finite alphabet $\{s_1, s_2, \ldots, s_r\}$ is written. A tape head reads one cell on the tape at each time step.

Depending on the symbol read by the tape head and on the state of the finite control, the TM

1. changes the state of the finite control;
2. rewrites the symbol on the cell which is read by the tape head;
3. shifts the tape head either to the left or right neighbor cell.

The finite control has two special states, one is the initial state and the other is the halting state(s). The finite control is prepared in the initial state at the beginning of the TM's movement (i.e., "computation"), and the TM halts when the finite control arrives at the halting state.

It is well known that the TM can solve various problems, or perform various "computations", by properly encoding the problem on the tape initially by using the alphabet $\{s_1, s_2, \ldots, s_n\}$, and by also properly setting transition function δ: $\{q_1, q_2, \ldots, q_m\} \times \{s_1, s_2, \ldots, s_n\} \to \{q_1, q_2, \ldots, q_m\} \times \{s_1, s_2, \ldots, s_n\} \times \{L, R\}$, where '$L$' denotes the motion of the tape head to left and 'R' to right, respectively. Because of this "computational" ability and equivalence of "computational" power with other various models of computation, the Church–Turing thesis [2] has been broadly accepted, which asserts that the intuitive concept of what is "computable" can be identified with what can be performed by TM. [13]

As the most important consequence of this formulation of computation, it has been understood that there exists a class of undecidable (i.e., not computable) decision problems within well-defined problems. A decision problem is decidable if it can be solved by a TM which is guaranteed to halt in finite time, and a problem is undecidable if it cannot be solved by a TM guaranteed to halt in finite time. Especially, the most famous undecidable problem is the halting problem of the following universal Turing machine.

One can construct a particular TM that is able to simulate any TM by accepting, on its initial tape, the description of the simulated TM, together with the description of input which would be given to the simulated TM. Such a TM is called universal Turing machine (UTM). Therefore a single UTM can perform all computation that can be performed by any TM.

[13] In spite of "vagueness" in the former, the latter is mathematically strictly defined.

The halting problem of a UTM is a question to decide whether the UTM will eventually halt (i.e., enter a halting state) or not (i.e., run forever and never halt), for a given initial tape with a given symbol string. This question is undecidable, i.e., no TM can answer this question with a guaranty to halt in finite time. Of course, there are TMs whose halting problems are decidable. In other words, the problems can be solved by another TM which always halts with finite time.

A halting set of a TM is a set from whose symbol strings, written on the initial tape, the TM will eventually halt. The set is called recursively enumerable. The class of recursively enumerable sets is the most complex in the Chomsky hierarchy [29,30] of the formal languages. A set of symbol strings with a decidable decision procedure is called recursive. [14] In the class of recursively enumerable sets, some sets are recursive (i.e., decidable) while some sets are undecidable. The halting set of UTM is known as a recursively enumerable set but not a recursive set [2].

3.2. Input for TM

Usually, we consider a symbol string to be an input for TM. In other words, on an initial tape, we consider a special symbol called "blank symbol", written on every cell except the symbol string. This use of symbol string as an input is equivalent to the use of a symbol sequence such that blank symbols are inserted to (both) the end(s) of that symbol string. [15]

However, we have to be careful when we consider a halting problem. If a TM halts on an input, it also halts on inputs obtained by changing symbols on such cells that are not read by the tape head until the TM halts, for any symbols (not just blank symbols). Of course, the number of changeable cells is not necessary to be finite in contrast with the usual input. Thus, to study the halting problem where halting is the only concern, we can consider the halting problem on all symbol sequences, instead of restricting inputs so that blank symbol is written on except a finite number of cells. [16] Therefore, as a halting set, we consider a set of symbol sequences instead of symbol strings.

3.3. Different patterns and different structure on any small scales

Let us consider the structure of a halting set of a UTM. A UTM can simulate any TM. In considering a halting set, this fact indicates that a mere single halting set of UTM contains all halting sets of TM.

To put it more concretely, suppose that a UTM has a one-way infinite tape with the tape alphabet $\{0, 1\}$, and that the UTM simulates a TM which also has a one-way infinite tape with the same tape alphabet. Then, assume that the symbol sequence on the initial tape of the UTM is a concatenated symbol sequence composed of the description of the simulated TM, followed by the symbol sequence that would be supplied for the simulated TM. In this setting, if a symbol sequence **s** is included in the halting set of the simulated TM (i.e., the simulated TM halts on **s**) and symbol string **p** is the description of the simulated TM, the concatenated symbol sequence **ps** is included in the halting set of the UTM. Thus, the halting set of the UTM contains all halting sets of TM. [17]

Now we consider the geometry of the halting set of UTM, obtained by using code $\frac{1}{2}$, i.e., the binary encoding, with the decreasing weight from left to right of the symbol sequence. From the above fact, the geometric representation of the halting set of the UTM contains different patterns, i.e., a region for each halting set of a TM, constructed by contracting the geometric representation of the halting set of the TM obtained by using the same code $\frac{1}{2}$, by the rate $2^{-|\mathbf{p}|}$, where $|\mathbf{p}|$ is the length of the description **p** of the TM. As a consequence, the geometry of the halt-

[14] In other words, a set is recursive if there is a TM, guaranteed to halt in finite time, that can answer whether a given symbol string is in the set or not.

[15] To be more precise, symbol sequence consisting only of blank symbols is appended to the end(s) of that symbol string.

[16] Another treatment of languages on infinite symbol strings is presented in, e.g. [31,32].

[17] To symbol sequences of each of halting sets of TM, the description **p** of the corresponding TM is prefixed.

ing set of the UTM by using code $\frac{1}{2}$ has a different structure on an arbitrarily small scale. [18]

Also in the case of a UTM having a two-way infinite tape with the alphabet $\{0, 1\}$, a similar result is obtained about the geometry of the halting set of the UTM by using code $\frac{1}{2}$ and by mapping it into a two-dimensional space. [19] In general, the present discussion is extended to geometry of an undecidable halting set of a UTM with n tape symbols, obtained by encoding symbol sequences as base-n expansion of real numbers. Such set has different fine structures on arbitrarily small scales and have different patterns, in strong contrast with ordinary self-similar fractals.

3.4. Universal Turing machine

Now, we will numerically investigate both boundary dimension of the halting set of UTM as a geometric property and halting time distribution of UTM as a dynamical feature of decision procedures. As a UTM, we mainly treat three UTMs, Rogozhin's UTM(24, 2), Minsky's UTM(7, 4) and Rogozhin's UTM(10, 3) [3,33], where UTM(m, n) denotes a UTM with m internal states and n tape symbols. (However, it should be noted that treating a specific 'U'TM is equal to treating all TMs including all UTMs.) We briefly describe each UTM, and then explain the coding method.

3.4.1. Rogozhin's UTM(24, 2)

Rogozhin's UTM(24, 2) has 24 internal states besides a halting state $\{q_1, q_2, \ldots, q_{24}\}$ and a two-way infinite tape with tape alphabet $\{0, 1\}$. q_1 is the initial state for this UTM. Table 1 shows the transition function of Rogozhin's UTM(24, 2). [20]

Since Rogozhin's UTM(24, 2)'s tape is two-way infinite one, we map a symbol sequence of the halting set of Rogozhin's UTM(24, 2) to a point of a two-dimensional space. To be more precise, we map the halting set into two unit squares, denoted by square 0 and square 1, respectively. First, we divide a symbol sequence in the halting set into three parts, one cell (called "center cell") which is read by the head at the beginning of the computation, and the right and left sides of the symbol sequence, from the center cell. Then we represent the right and left sides of the symbol sequence by a pair of real numbers given by code α, respectively. Finally, this pair is put into either square 0 or square 1 depending on whether the symbol on the center cell is equal to '0' or '1', where the right side of the symbol sequence corresponds to the horizontal axis, and the left side to the vertical axis.

We name this geometric representation of the halting set of Rogozhin's UTM(24, 2) by using code α to be UL_α, Fig. 2 shows $UL_{1/2}$ and $UL_{1/3}$.

3.4.2. Minsky's UTM(7, 4)

Minsky's UTM(7, 4) has seven internal states besides a halting state $\{q_1, q_2, \ldots, q_7\}$ and a bi-infinite tape with tape alphabet $\{y, 0, 1, A\}$. q_2 is the initial state for this UTM. The symbol 'y' is read by the head of Minsky's UTM(7, 4) at the beginning of its computation. Table 2 shows the transition function of Minsky's UTM(7, 4).

To obtain a geometric representation of the halting set of Minsky's UTM(7, 4) in a two-dimensional space, first we transform the tape alphabet $\{y, 0, 1, A\}$ of a symbol sequence into $\{3, 0, 1, 2\}$, respectively. (Other choices of transformation from the tape

[18] We can easily construct a TM which accepts (halts on) a symbol string of arbitrary length and does not accept other symbol strings. Thus the geometry of the halting set of the UTM cannot have a geometric structure only up to a finite scale (e.g., classical geometric sets like line segment). Similarly, the geometry of the halting set of the UTM does not have a self-similar structure (i.e., a same structure) on an arbitrarily small scale, like a Cantor set, since it contains an arbitrary halting set of TM, including a TM which only halts on a symbol sequence of binary expansion of a point corresponding to a fractal with an arbitrary structure. (In fact, it contains various fractals which can be drawn by a digital computer.) Hence the geometry of the halting set of the UTM must have different fine structures on arbitrarily small scales.

[19] The result is obtained by mapping symbol sequence of the halting set to a pair of real numbers by dividing the symbol sequence at the cell which is read by the tape head at the beginning of computation, and using code $\frac{1}{2}$ for each, see also next section.

[20] There are further restrictions, especially on inputs, to use Rogozhin's UTM(24, 2) "properly". Still, we do not need to exclude inputs that do not use the UTM properly since our main concern lies in dynamical and geometrical aspects of the undecidability of the halting problem. This remark is also applied to the cases of other UTMs.

Table 1
The transition function of Rogozhin's UTM(24, 2) [33]

	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8
0	$0, q_5, R$	$1, q_1, R$	$0, q_4, L$	$1, q_{12}, L$	$1, q_1, R$	$0, q_7, L$	$0, q_8, L$	$0, q_7, L$
1	$1, q_2, R$	$1, q_3, L$	$0, q_2, L$	$0, q_9, L$	$0, q_6, L$	$1, q_7, L$	$0, q_6, L$	$1, q_2, R$
	q_9	q_{10}	q_{11}	q_{12}	q_{13}	q_{14}	q_{15}	q_{16}
0	$0, q_{19}, R$	$1, q_4, L$	$0, q_4, L$	$0, q_{19}, R$	$0, q_{10}, R$	$0, q_{15}, L$	$0, q_{16}, R$	$0, q_{15}, R$
1	$1, q_4, L$	$0, q_{13}, R$	Halt	$1, q_{14}, L$	$1, q_{24}, R$	$1, q_{11}, L$	$1, q_{17}, R$	$1, q_{10}, R$
	q_{17}	q_{18}	q_{19}	q_{20}	q_{21}	q_{22}	q_{23}	q_{24}
0	$0, q_{16}, R$	$0, q_{19}, R$	$1, q_3, L$	$1, q_{18}, R$	$0, q_{22}, R$	$1, q_{10}, L$	$1, q_{21}, R$	$0, q_{13}, R$
1	$1, q_{21}, R$	$1, q_{20}, R$	$1, q_{18}, R$	$0, q_{18}, R$	$1, q_{23}, R$	$1, q_{21}, R$	$0, q_{21}, R$	$0, q_3, L$

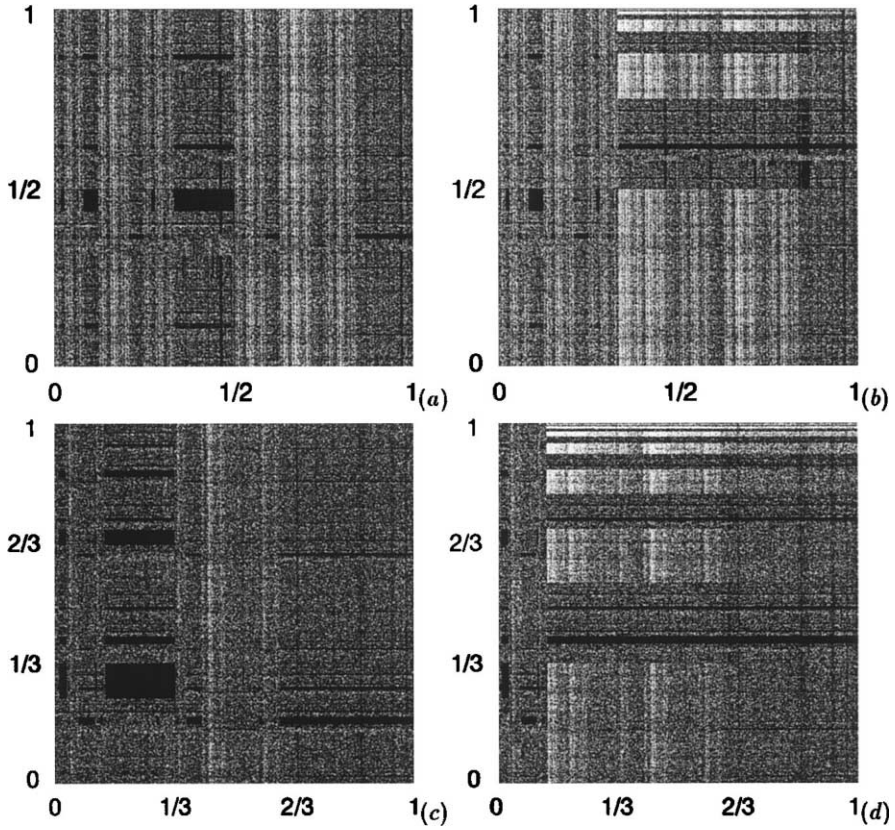

Fig. 2. The halting set of Rogozhin's UTM(24, 2), represented by code $\frac{1}{2}$ and code $\frac{1}{3}$. Dots correspond to inputs halted within 500 steps ($UL_{1/2}(500)$ and $UL_{1/3}(500)$ in the text). (a) Square 0 of $UL_{1/2}$ (500). (b) Square 1 of $UL_{1/2}(500)$. (c) Square 0 of $UL_{1/3}(500)$. (d) Square 1 of $UL_{1/3}(500)$. Inputs are chosen at random in each grid cell (1024×1024 grid and 729×729 grid, respectively).

Table 2
The transition function of Minsky's UTM(7, 4) [3]

	q_1	q_2	q_3	q_4	q_5	q_6	q_7
y	$0, q_1, L$	$0, q_1, L$	y, q_3, L	y, q_4, L	y, q_5, R	y, q_6, R	$0, q_7, R$
0	$0, q_1, L$	y, q_2, R	Halt	y, q_5, R	y, q_3, L	A, q_3, L	y, q_6, R
1	$1, q_2, L$	A, q_2, R	A, q_3, L	$1, q_7, L$	A, q_5, R	A, q_6, R	$1, q_7, R$
A	$1, q_1, L$	y, q_6, R	$1, q_4, L$	$1, q_4, L$	$1, q_5, R$	$1, q_6, R$	$0, q_2, R$

alphabet to {0, 1, 2, 3} meet with the same results below.) Then we divide the symbol sequence into three parts as before, and we represent the right and left sides of the symbol sequence by a pair of real numbers given by a 4-symbol code, respectively. Finally, this pair is put on a unit square.

Fig. 3 shows the geometric representation of the halting set of Minsky's UTM(7, 4) obtained by base-4 code.

3.4.3. Rogozhin's UTM(10, 3)

Rogozhin's UTM(10, 3) has 10 internal states besides a halting state $\{q_1, q_2, \ldots, q_{10}\}$ and a two-way infinite tape with tape alphabet $\{0, 1, b\}$. q_1 is the initial state for this UTM. Table 3 shows the transition function of Rogozhin's UTM(10, 3). Here, geometric representation of the halting set of this UTM is obtained in the same way, using a 3-symbol code and three unit squares.

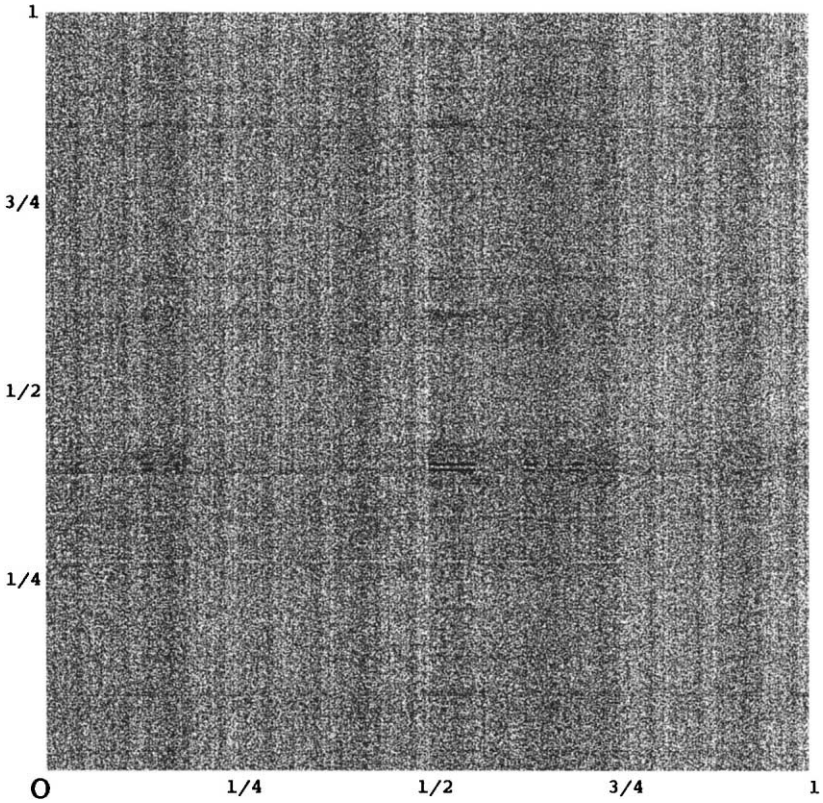

Fig. 3. The halting set of Minsky's UTM(7, 4), represented by base-4 code. Dots correspond to inputs halted within 500 steps. Inputs are chosen at random in each grid cell (1024 × 1024 grid).

Table 3
The transition function of Rogozhin's UTM(10, 3) [33]

	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8	q_9	q_{10}
0	$1, q_1, R$	$0, q_3, L$	$0, q_2, L$	$1, q_1, R$	b, q_3, L	$1, q_7, L$	$0, q_8, R$	$1, q_6, L$	$1, q_{10}, L$	$0, q_5, R$
1	$0, q_2, L$	$0, q_2, L$	b, q_6, L	$1, q_5, R$	$1, q_5, R$	$1, q_6, L$	Halt	$1, q_8, R$	$0, q_{10}, R$	$0, q_{10}, R$
b	b, q_4, R	b, q_2, L	b, q_1, R	$1, q_4, L$	b, q_5, R	b, q_6, L	b, q_9, L	b, q_8, R	$0, q_4, L$	b, q_9, R

3.5. Boundary dimension

Now we study the dimension of a halting set of UTM. Geometric representation of halting set of UTM has positive Lebesgue measure (i.e., a fat fractal [8,34]). In fact, our UTMs are numerically shown to halt for some symbol sequences on the initial tape. When the head of a UTM, moves to the right and left at most i and j sites, respectively, during its computation, before it halts, then the UTM with symbol sequences of the same symbols up to i and j sites and any symbols beyond them, also halts, because the UTM can only read up to i and j sites before it halts, as mentioned already. These symbol sequences correspond to rectangular region with any of our code (in the case of $UL_{1/2}$ of Rogozhin's UTM(24, 2), they correspond to rectangular region with sides 2^{-i} and 2^{-j}), and this region is contained in the geometric representation of the halting set of the UTM. Therefore, the halting set of the UTM has positive two-dimensional Lebesgue measure (at least $2^{-(i+j)}$ in the case of $UL_{1/2}$). Dots in Figs. 2 and 3 correspond to such symbol sequences.

Thus, its fractal dimension is the same as the dimension of the space (i.e., $D = 2$). Instead, we investigate the box-counting dimension of the boundary (i.e., the exterior dimension [8,35]) of geometric representation of halting set of UTM to study its fine structure.

The definition of the box-counting dimension of a set S in N-dimensional space is equivalent to $D_0 = N - \lim_{\epsilon\to 0} \log V[S(\epsilon)]/\log\epsilon$, where $V[S(\epsilon)]$ is the N-dimensional volume of the set $S(\epsilon)$ created by fattening the original set S by ϵ [8]. In the following, based on this definition, we numerically estimate the box-counting dimension of the boundary (the boundary dimension) of geometric representation of halting set of each of the above mentioned UTMs. Especially we investigate the boundary dimension of geometric representation of halting set of Rogozhin's UTM(24, 2) with code $\frac{1}{2}$ and code $\frac{1}{3}$ (i.e., $UL_{1/2}$ and $UL_{1/3}$, respectively), and those of Minsky's UTM(7, 4) and Rogozhin's UTM(10, 3) with base-4 and base-3 code, respectively.

Of course, it is impossible to take infinite time for the numerical simulations. Thus we treat the set of inputs on which each UTM will halt within a given finite step n (denoted as $UL_\alpha(n)$ in the case of Rogozhin's UTM(24, 2)), and investigate the boundary dimension of the set $(UL_\alpha(n))$. Then we survey the asymptotic behavior of the boundary dimension of the set $(UL_\alpha(n))$ as n is increased (to infinity).

3.5.1. $UL_{1/2}$

The detailed procedure to study the boundary dimension D_0 of $UL_{1/2}$ is as follows. First, we choose a point (X, Y) in either unit square 0 or 1 of a two-dimensional space at random.[21] Then we perturb this point (X, Y) to $(X + \epsilon, Y)$, $(X, Y + \epsilon)$ and $(X + \epsilon, Y + \epsilon)$. Then we decide whether Rogozhin's UTM(24, 2), starting from each of these four tapes at the center cell, will halt within n steps or not (i.e., in $UL_{1/2}(n)$ or not). If all four points are in $UL_{1/2}(n)$ or none of them are in $UL_{1/2}(n)$, we regard there to be no boundary in the square of length ϵ made from these four points. Otherwise we regard that there is a boundary in the square. We repeat this procedure for a large number of points and evaluate the fraction of squares with a boundary, denoted as $f(\epsilon)$, which gives the estimation of $V[S(\epsilon)]$. Varying ϵ, we obtain the scaling of $f(\epsilon)$ with ϵ, and can evaluate $N - D_0$ (i.e., $2 - D_0$).

[21] Choosing square 0 or 1 is equal to determining the symbol on the center cell.

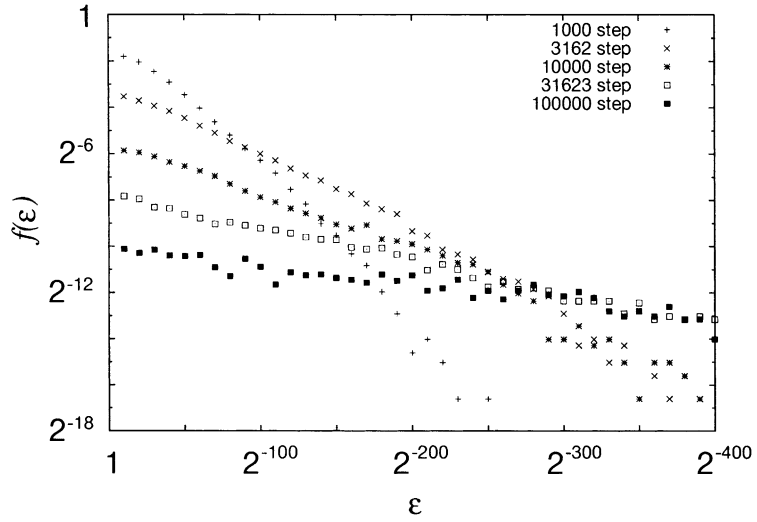

Fig. 4. Log–log plot of $f(\epsilon)$ (i.e., the fraction of squares with a boundary) of $UL_{1/3}(n)$ with ϵ for computation time $n = 1000$, 3162, 10 000, 31 623 and 100 000. The slope of $f(\epsilon)$ becomes smaller with the increase of n.

In Fig. 4, the log–log plot of $f(\epsilon)$ of $UL_{1/2}(n)$ with ϵ is shown for several values of n. It can be fit as $f(\epsilon) \sim \epsilon^{2-D_0}$ for small ϵ, from which one can obtain $2 - D_0$ for each n. Fig. 5 displays the log–log plot of $2 - D_0$ versus the computation time n. It shows that $2 - D_0$ of the boundary of $UL_{1/2}(n)$ approaches zero roughly as $n^{-0.45}$ with the increase of the computation time n. In other words, when we increase n, the box-counting dimension of the boundary of $UL_{1/2}(n)$ approaches two, which is the same as the dimension of the space. Thus the box-counting dimension of the boundary of $UL_{1/2}$ is estimated to be two.

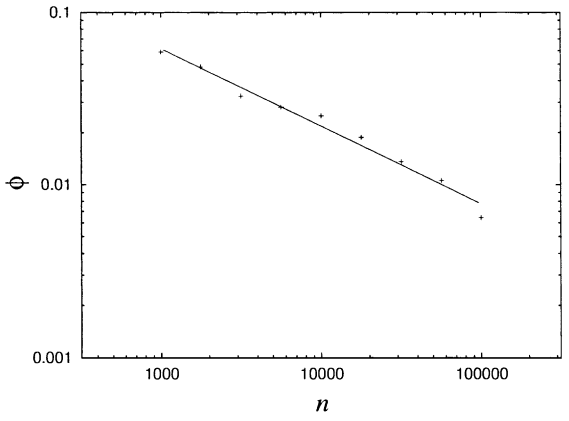

Fig. 5. Log–log plot of $2 - D_0$ (i.e., uncertainty exponent ϕ) of $UL_{1/2}(n)$ versus the computation time n, obtained from Fig. 4. It approaches zero roughly as $n^{-0.45}$ with the increase of the computation time n. The data are least-square fit to a straight line, $1.4n^{-0.45}$.

Here we briefly refer to the uncertainty exponent $\phi = N - D_0$ [8,35]. Suppose there exists a set A in a certain N-dimensional space and our ability to determine the position of points has an uncertainty ϵ. $\bar{A}$ denotes the complement of A and S denotes the boundary separating A and $\bar{A}$. If we have to determine which set a given point lies in, the probability $f(\epsilon)$ of making a mistake in such determination is proportional to $V[S(\epsilon)]$. Thus, if the box-counting dimension of the boundary S is D_0, $f(\epsilon)$ is proportional to $\epsilon^{N-D_0} (= \epsilon^{\phi})$. If ϕ is small, then a large decrease in ϵ leads to only a relatively small decrease in $f(\epsilon)$. Thus ϕ is called uncertainty exponent.

The above result that the uncertainty exponent of the boundary of $UL_{1/2}(n)$ becomes smaller with the increase of computation time n indicates that it becomes more difficult to decrease mistake in determination of $UL_{1/2}(n)$. The null uncertainty exponent of the boundary of $UL_{1/2}$ (i.e., $D_0 = 2$) indicates that the probability of making a mistake (equally, the volume of ϵ-neighborhood of the boundary $V[S(\epsilon)]$) does not depend on ϵ.

A decision procedure is defined in ideal condition without errors. On the other hand, robustness of a decision procedure against a noise is essentially different, depending on if the boundary dimension of a set is equal to the space dimension or less. Indeed, if the boundary dimension is less than the space dimension with the uncertainty exponent $\phi > 0$, one can decrease the volume of ϵ-neighborhood of a boundary ($V[S(\epsilon)]$) to any amount in principle, by decreasing ϵ. In other words, one can arbitrarily approach the ideal decision procedure (i.e., ideal condition without errors) by improving accuracy in principle, even under presence of errors. On the other hand, if the boundary dimension is equal to the space dimension with the uncertainty exponent $\phi = 0$, it is impossible to approach the ideal decision procedure (i.e., the ideal condition) as long as there exist errors no matter how small they are.

Consequently, decision procedure of the halting set of UTM by using code $\frac{1}{2}$ has so strong uncertainty that one cannot approach the ideal decision procedure, in the presence of error. This result is also reasonable, since unlike chaotic unpredictability (i.e., sensitive dependence on initial conditions), the halting problem of UTM is undecidable even if descriptions of inputs are known exactly. In this way, the undecidability is explained from the viewpoint of the uncertainty exponent, i.e., the boundary dimension.

3.5.2. $UL_{1/3}$

In the case using code $\frac{1}{3}$, i.e., for obtaining the boundary dimension of $UL_{1/3}$, the detailed procedure is as follows. It is the same as the case of $UL_{1/2}$ that first we choose a point (X, Y) in either unit square 0 or 1 of a two-dimensional space at random, and perturb the point (X, Y) to $(X + \epsilon, Y)$, $(X, Y + \epsilon)$ and $(X + \epsilon, Y + \epsilon)$. Then, we decide the labels of Markov partition for $X, Y, X + \epsilon$ and $Y + \epsilon$ (i.e., decoded symbol sequences $(\text{code}\,\frac{1}{3})^{-1}(X)$, $(\text{code}\,\frac{1}{3})^{-1}(Y)$, $(\text{code}\,\frac{1}{3})^{-1}(X + \epsilon)$ and $(\text{code}\,\frac{1}{3})^{-1}(Y + \epsilon)$), and compose four input symbol sequences for corresponding four points. For example, in the case of (X, Y) in square 0, the corresponding input symbol sequence is concatenation of $(\text{code}\,\frac{1}{3})^{-1}(Y)$, '0', $(\text{code}\,\frac{1}{3})^{-1}(X)$, where the symbol sequence $(\text{code}\,\frac{1}{3})^{-1}(Y)$ is flip-flopped (turned over). Subsequent procedure is the same as for the case of $UL_{1/2}$.

In Fig. 6, the log–log plot of $f(\epsilon)$ of $UL_{1/3}(n)$ with ϵ is shown for several values of n. It can be fit as $f(\epsilon) \sim \epsilon^{\phi}$ for small ϵ, from which one can obtain ϕ

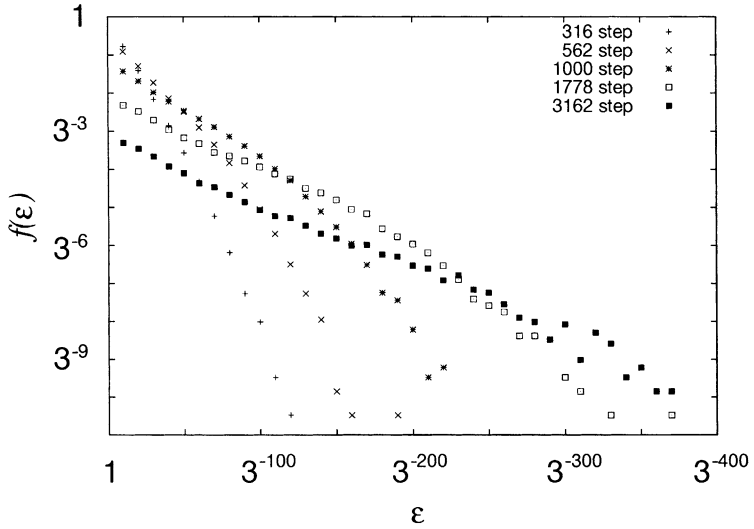

Fig. 6. Log–log plot of $f(\epsilon)$ (i.e., the fraction of squares with a boundary) of $UL_{1/3}(n)$ with ϵ for computation time $n = 316$, 562, 1000, 1778 and 3162. The slope of $f(\epsilon)$ becomes smaller with the increase of n, similarly in Fig. 4.

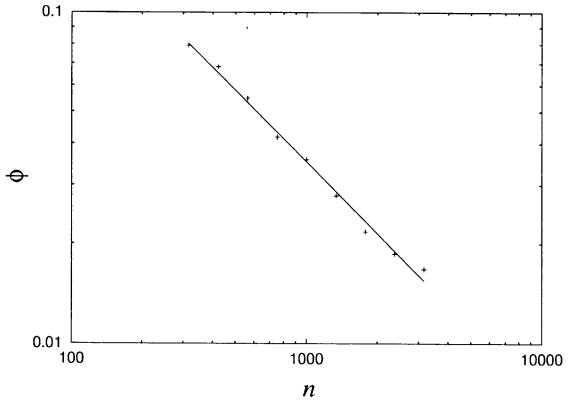

Fig. 7. Log–log plot of uncertainty exponent ϕ (i.e., $2 - D_0$) of $UL_{1/3}(n)$ versus the computation time n, obtained from Fig. 6. It also approaches zero with a power law (roughly as $n^{-0.70}$) as in the case of $UL_{1/2}$.

for each n. Fig. 7 shows the log–log plot of ϕ versus the computation time n, and ϕ also approaches zero with a power law (roughly as $n^{-0.70}$) as in the case of $UL_{1/2}$. In other words, with the increase of n, the box-counting dimension of the boundary of $UL_{1/3}(n)$ approaches two, i.e., the dimension of the space, in the same way as $UL_{1/2}$. Thus the box-counting dimension of the boundary of $UL_{1/3}$ is also estimated to be two.

Consequently, similarly to $UL_{1/2}$, the boundary dimension of $UL_{1/3}$ is also equal to the space dimension two, and decision procedure of $UL_{1/3}$ also has strong uncertainty so that one cannot approach the ideal decision procedure, in the presence of error. Thus, our result of the strong uncertainty of UL_α is not based on the property of the code. [22]

Note that $UL_{1/3}$ is the set obtained by transforming $UL_{1/2}$ by the code transformation $t_{1/2\to 1/3}$. As seen in the case of Cantor set in Section 2, code transformation in general, does not preserve the (boundary) dimension. However, the above results show that the property that boundary dimension is equal to space dimension, i.e., the strong uncertainty, is preserved in the case of the halting set of UTM. Thus, one could say the property that a decision procedure is not robust against a noise is robust under application of even "fractal" code transformations.

In conclusion, the boundary dimension of geometric representation of the halting set of UTM is equal to the space dimension. There exists strong uncertainty such that one cannot approach the ideal decision procedure in the presence of error. Furthermore, the above property is conserved even by "fractal" code transformations. Thus, a characterization of the undecidability is given by the strong uncertainty implying inaccessibility to the ideal decision procedure, and by invariance of the strong uncertainty against code transformations. It should be noted that this characterization of undecidability is intuitively more comprehensible than ordinary one in classical computation theory.

3.5.3. Geometric representation of halting set of other UTMs

The boundary dimension of geometric representation of halting set of other UTMs is also equal to the space dimension. Since UTMs can imitate each other, the geometric representations of halting set for any UTMs contain each other, and each has the same boundary dimension, that is equal to the space dimension.

Indeed, we have numerically studied the boundary dimension of geometric representation of halting sets of Minsky's UTM(7, 4) and Rogozhin's UTM(10, 3), obtained by using base-4 code and base-3 code, respectively. Fig. 8 shows the log–log plot of the uncertainty exponent ϕ of these sets, versus the computation time n. In these systems again, ϕs approach zero with some power of n, as in the previous cases (roughly as $n^{-0.60}$ and $n^{-0.65}$, respectively). Thus the boundary dimension of these sets is also estimated to be two, i.e., the dimension of the space in which they lie.

3.6. Halting time distribution

We now focus on dynamical systems characteristics of decision procedure of such geometric representation of halting set of UTM. In particular, we study the distribution of halting time of the decision procedure, for Rogozhin's UTM(24, 2) with code $\frac{1}{2}$ and code $\frac{1}{3}$ (i.e, $UL_{1/3}$ and $UL_{1/3}$, respectively), and Minsky's

[22] The result of the boundary dimension equal to the space dimension of the halting set of UTM is independent of code α, see also Section 5.3.

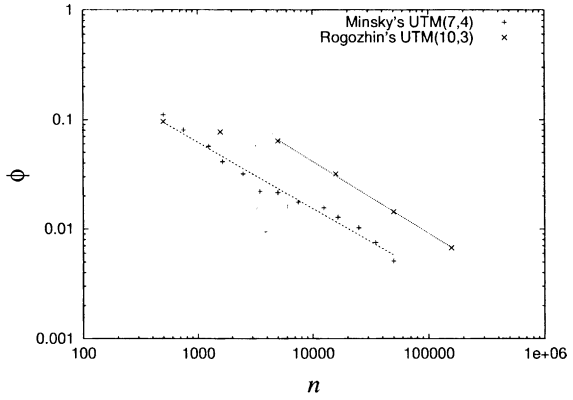

Fig. 8. Log–log plot of uncertainty exponent $\phi(2-D_0)$ of Minsky's UTM(7, 4) using base-4 code and Rogozhin's UTM(10, 3) using base-3 code, versus the computation time n. Both ϕs approach zero, roughly as $n^{-0.60}$ and $n^{-0.65}$, respectively, with the increase of the computation time n.

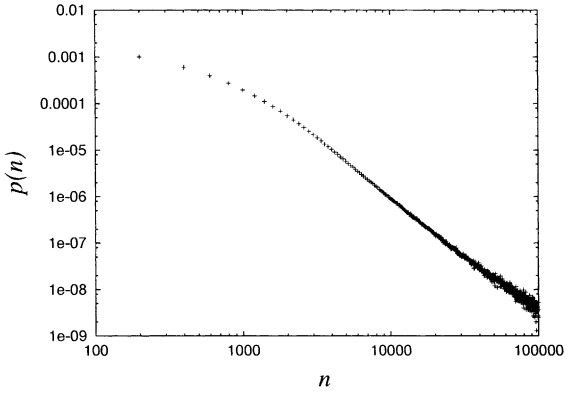

Fig. 9. Log–log plot of $p(n)$ (i.e., the fraction of the initial points halting with computation time n) with n in the case of $UL_{1/2}$. $p(n)$ is fitted by $p(n) \sim n^{-2.4}$.

UTM(7, 4) and Rogozhin's UTM(10, 3) with base-4 and base-3 code, respectively.

3.6.1. Cantor set

First, let us consider the halting time distribution of the complement of the Cantor set, which corresponds to a regular set, before treating those of UTMs. The complement of the Cantor set is the set of initial points which eventually leave the interval [0, 1] by the map $f(x) = 3x\,(x < \frac{1}{2})$, $f(x) = 3x - 2\,(x \geq \frac{1}{2})$. Thus, if $x \in [0, 1]$ is given, the decision procedure for the set is deciding whether $f^k(x) \notin [0, 1]$ for some $k > 0$. In this case, the length of a chaotic transient is computation time for the decision process.

In the construction of the complement of the Cantor set, the fraction of the points which leave [0, 1] decays exponentially with time n when initial points are scattered uniformly in [0, 1]. In general, the distribution of transient time decays exponentially for transient chaos (i.e., construction process of ordinary fractals), where remaining points with null measure forms fractal set [8].

3.6.2. $UL_{1/2}$

On the other hand, in the case of our $UL_{1/2}$, the fraction of the initial points halting with computation time n, denoted as $p(n)$, is found to decay according to a power law. Fig. 9 shows the log–log plot of $p(n)$ with n, where $p(n)$ is fitted by $p(n) \sim n^{-2.4}$.

In contrast with this numberical result, $p(n)$ is expected to decay slower than power law if n is further increased. The reason is as follows. The halting probability of a UTM, known as Chaitin number Ω [36], is obtained by summing $p(n)$ over n. Ω is known as random number [37]. If $p(n)$ decayed with a power law up to $n \to \infty$, there were an algorithm to tell any bit of halting probability Ω. This would contradict the randomness of Ω. Thus $p(n)$ must decay slower than a power law. [23]

3.6.3. $UL_{1/3}$

Similarly we have studied $p(n)$ for $UL_{1/3}$. As shown in Fig. 10, $p(n)$ is found to decay with a power law ($p(n) \sim n^{-2.7}$) as in the case of $UL_{1/2}$. Thus this character of the halting time distribution is also unchanged under code transformation.

[23] Suppose $p(n)$ were bounded upwards by cn^a for sufficiently large n, where a is a constant satisfying $a < -1$ and c is a positive constant. Then, Ω satisfied $\sum_{n=1}^{N} p(n) \leq \Omega \leq \sum_{n=1}^{N} p(n) + \int_{n+1}^{\infty} cn^a\,\mathrm{d}n$, where N is a sufficiently large natural number. The computation time necessary to calculate $\sum_{n=1}^{N} p(n)$ is finite. Thus, one could improve the precision of Ω in any amount by taking large N. (The error were at most $\int_{N+1}^{\infty} cn^a\,\mathrm{d}n$, which is also computable.) Hence, if the power law holds, the power should approach -1 for $n \to \infty$, because $\sum_{n}^{\infty} n^a$ diverges with $a = -1$.

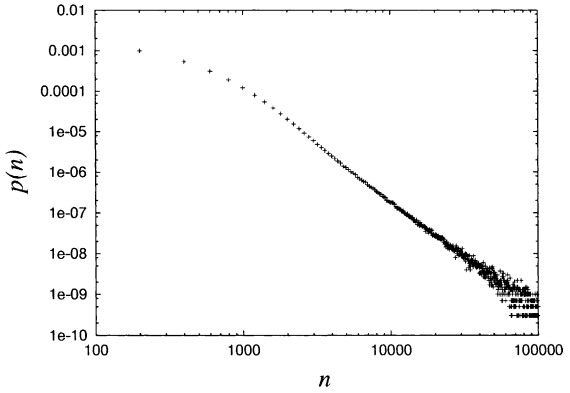

Fig. 10. Log–log plot of $p(n)$ (i.e., the fraction of the initial points halting with computation time n) with n in the case of $UL_{1/3}$. $p(n)$ also decays with a power law ($p(n) \sim n^{-2.7}$) as in the case of $UL_{1/2}$.

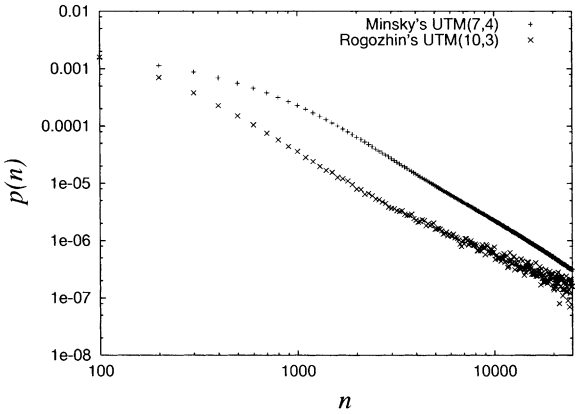

Fig. 11. Log–log plot of the halting time distribution of Minsky's UTM(7, 4) and Rogozhin's UTM(10, 3), obtained by using base-4 code and base-3 code, respectively. $p(n)$ (i.e., the fraction of the initial points halting with computation time n) also decays with a power law ($p(n) \sim n^{-2.1}$ and $p(n) \sim n^{-1.5}$, respectively), in the same way as in Rogozhin's UTM(24, 2).

In any case, the construction of both $UL_{1/2}$ and $UL_{1/3}$ takes much longer time than the construction of ordinary fractal sets by transient chaos. Furthermore, it should be stressed that the halting time distribution of undecidable system obeys power law distribution up to large n. Indeed this power law property is not restricted to the above Rogozhin's UTM(24, 2), as shown next.

3.6.4. Geometric representation of halting set of other UTMs

We have numerically studied the halting time distribution of geometric representation of halting set of Minsky's UTM(7, 4) and Rogozhin's UTM(10, 3). In Fig. 11, we have plotted the halting time distribution of Minsky's UTM(7, 4) with base-4 code and Rogozhin's UTM(10, 3) with base-3 code. Again this shows the power law property ($p(n) \sim n^{-2.1}$ and $p(n) \sim n^{-1.5}$, respectively).

Remark. In this section, it is numerically found that the uncertainty exponent and the halting time distribution decay as power laws of computation time n. The powers, however, depend on the UTM and on the code. At this stage, we conjecture that these results imply absence of universal power exponent.

4. Mandelbrot set

In this section, we deal with the Mandelbrot set, to show its common features with the above halting set of a UTM.

4.1. Mandelbrot set

The Mandelbrot set [10,11], denoted as M, is the subset of the complex plane $\mathbf{C}$, and is defined as $M = \{c \in \mathbf{C} || Q_c^n(0)| \not\to \infty\}$ for $Q_c(z) = z^2 + c$ (equivalently, $M = \{c \in \mathbf{C} | K_c \text{ is connected}\}$, where K_c is the filled-in Julia set of Q_c). Fig. 12 shows M.

Although self-similarity of M has often been studied as a structure containing small copies of M itself, M is also known as a set with extraordinarily complex structure. Indeed, M has been called one of the most intricate and beautiful objects in mathematics [10,11]. Likewise, M has Misiurewicz points where M and corresponding filled-in Julia set K_c are similar. Thus, similarly with geometric representation of halting set of UTM, M has different patterns and has a different fine structure on an arbitrarily small scale, in contrast with ordinary self-similar fractals.

There is a decision procedure for M. It is shown that if $|c| > 2$, then $|Q_c^n(0)| \to \infty$ and thus $c \in M$. Also

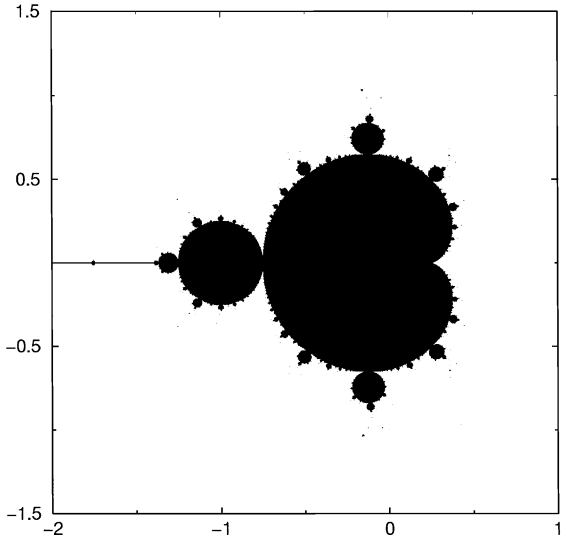

Fig. 12. The Mandelbrot set M (precisely, $M(500)$).

if $|Q_c^k(0)| > 2$ for some $k > 0$, then $|Q_c^n(0)| \to \infty$ [10,11]. Thus, if $c \in \mathbf{C}$ is given, the decision procedure of M (precisely, that of the complement of M) is given by deciding whether $|Q_c^k(0)| > 2$ for some $k > 0$.

Although their origins are different, M and halting set of UTM are not distinguishable as a set defined by a decision procedure. Also, there is no guarantee if the decision procedure of M (precisely, that of the complement of M) will halt, similarly with the halting problem of UTM. Hence it is interesting to study if there is a relation, in some sense, between M and recursively enumerable sets [18,38].

Here we investigate the M's boundary dimension and the halting time distribution of M's decision procedure in the same way as the halting set of UTM, to point out that (the decision procedures of) M and the halting sets of UTM have common properties.

4.2. Boundary dimension

4.2.1. M

It is proven that Hausdorff dimension, which gives a lower bound for box-counting dimension, of the boundary of M is equal to the space dimension two [39], i.e., the maximal value in two-dimensional space. To see also the asymptotic behavior of such boundary dimension, we study the box-counting dimension of the boundary of M numerically in the same way as UL_α. Here again, we cannot treat M directly by means of numerical simulation as in the case for UL_α. Thus we study the boundary dimension of $M(n)$, i.e., the set of points which are not decided not included in M

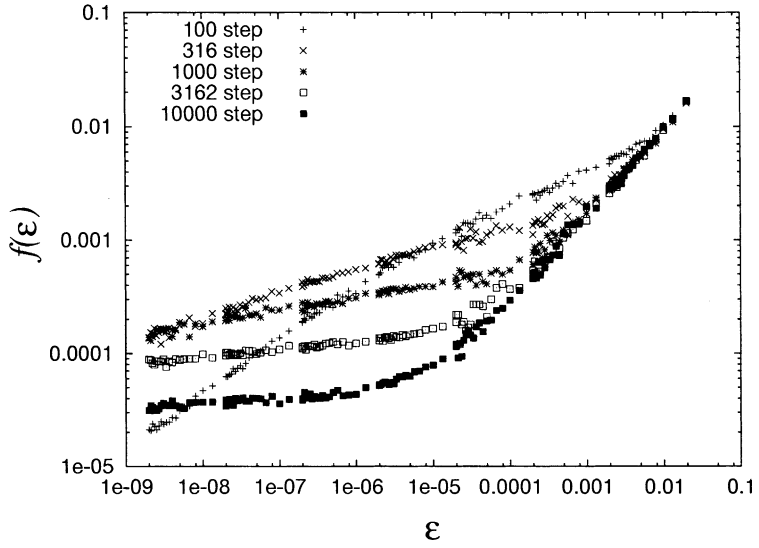

Fig. 13. Log–log plot of $f(\epsilon)$ (i.e., the fraction of squares with a boundary) of $M(n)$ with ϵ for computation time $n = 100$, 3162, 1000, 3162 and 10 000. The slope of $f(\epsilon)$ becomes smaller with the increase of n.

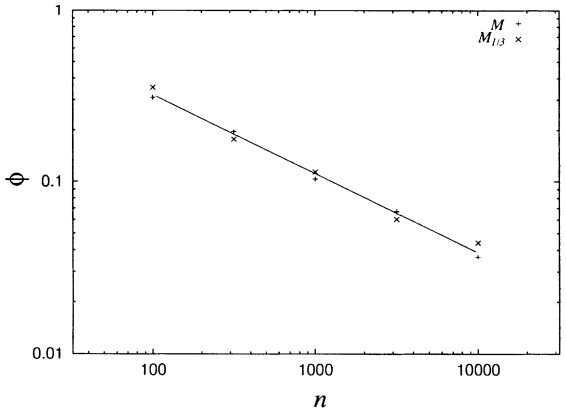

Fig. 14. Log–log plot of uncertainty exponent ϕ (i.e., $2 - D_0$) of $M(n)$ and $M_{1/3}(n)$ versus the computation time n. Both ϕs approach zero roughly as $n^{-0.45}$ with the increase of the computation time n. This power law decay of ϕs is in common with the case of halting set of UTM.

within a given finite step n. The detailed procedure for this case is almost same as for UL_α's.

The asymptotic behavior of the boundary dimension of $M(n)$ with the increase of n is given in Fig. 13, where $f(\epsilon)$, the fraction of squares with a boundary of $M(n)$, is plotted with ϵ with a logarithmic scale for several values of n. Fig. 14 also shows the log–log plot of ϕ (i.e., $2 - D_0$) versus the computation time n. As is shown, the box-counting dimension of the boundary of $M(n)$ approaches two with the increase of the computation time n. The approach is given by $2 - cn^{-0.45}$, where c is a positive constant. This asymptotic convergence to the dimension of the space is common with the case of halting set of UTM.

4.2.2. $M_{1/3}$

Since M is already a geometric set, it intrinsically does not matter how to encode. However, it is possible to apply a fractal function which corresponds to a code transformation, to M.

Now, by using Lebesgue's singular function L_β which is a transformation from encoding with code $\frac{1}{2}$ into encoding with code β, we can define $L'_\beta : [-2, 2] \to [-1, 1]$[24]

[24] In other words, the map corresponds to a transformation from a real number to the real number on Markov partition, which has the same label as the base-2 expansion of the original real number.

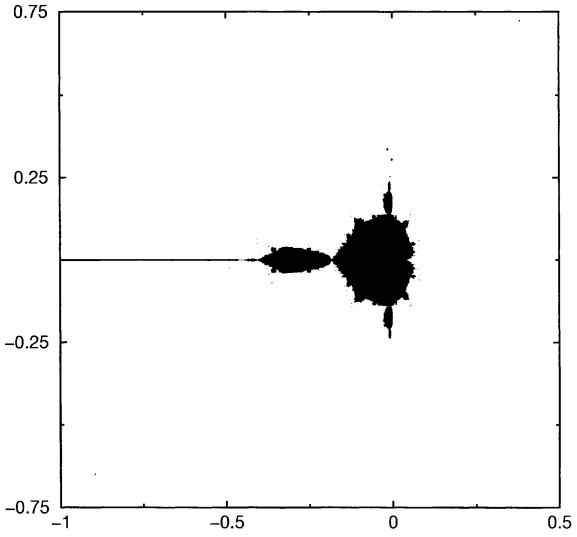

Fig. 15. $M_{1/3}$ obtained by $t_{\to 1/3}(M)$ (precisely, $M_{1/3}(500)$).

$$L'_\beta(x) = \begin{cases} L_\beta(\frac{1}{2}x) & \text{for } x \in [0, 2], \\ -L_\beta(-\frac{1}{2}x) & \text{for } x \in [-2, 0], \end{cases}$$

and also define a transformation $t_{\to\beta} : x + \mathrm{i}y \mapsto L'_\beta(x) + \mathrm{i}L'_\beta(y)$, where $x, y \in [-2, 2]$.

Let M_β the set obtained by applying $t_{\to\beta}$ to M (i.e., $t_{\to\beta}(M)$). In the following, we will numerically study boundary dimension of M_β in the case of $\beta = \frac{1}{3}$ (i.e., $M_{1/3}$; Fig. 15).

In the same way as for M, the asymptotic behavior of the boundary dimension of $M_{1/3}(n)$ with the increase of n is studied by considering $M_{1/3}(n)$ obtained by $t_{\to 1/3}(M(n))$. In Fig. 16, $f(\epsilon)$, the fraction of squares with a boundary of $M_{1/3}(n)$, is plotted with ϵ with a logarithmic scale for several values of n. Fig. 14 also shows the log–log plot of ϕ versus the computation time n as before. As is shown, the box-counting dimension of the boundary of $M_{1/3}(n)$ approaches two by $2 - cn^{-0.45}$ with the increase of the computation time n. Thus the box-counting dimension of the boundary of $M_{1/3}$ is also estimated to be two. This property of boundary dimension including asymptotic behavior is common with the previous cases.

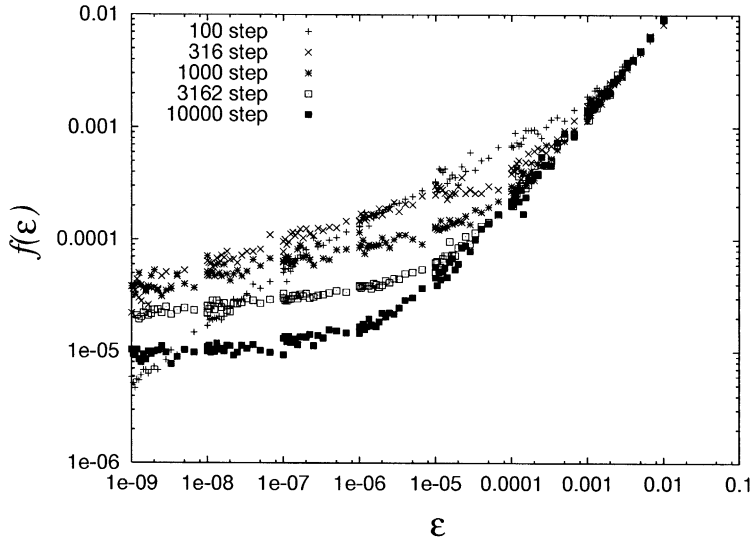

Fig. 16. Log–log plot of $f(\epsilon)$, the fraction of squares with a boundary of $M_{1/3}(n)$, with ϵ for computation time $n = 100$, 316, 1000, 3162 and 10 000. The slope of $f(\epsilon)$ becomes smaller with the increase of n.

Consequently, similarly with the case of undecidable halting set of UTM, decision procedures of M and $M_{1/3}$ also indicate the strong uncertainty, in the sense that one cannot approach the ideal decision procedure, in the presence of error. This strong uncertainty is also unchanged under application of a fractal function which corresponds to a code transformation. Now it is clear that M and halting set of UTM have common features (which were expected by naive arguments previously) from the standpoint of the strong uncertainty.

4.3. Halting time distribution

We have also studied the halting time distribution $p(n)$ for M and $M_{1/3}$, defined as the fraction of the

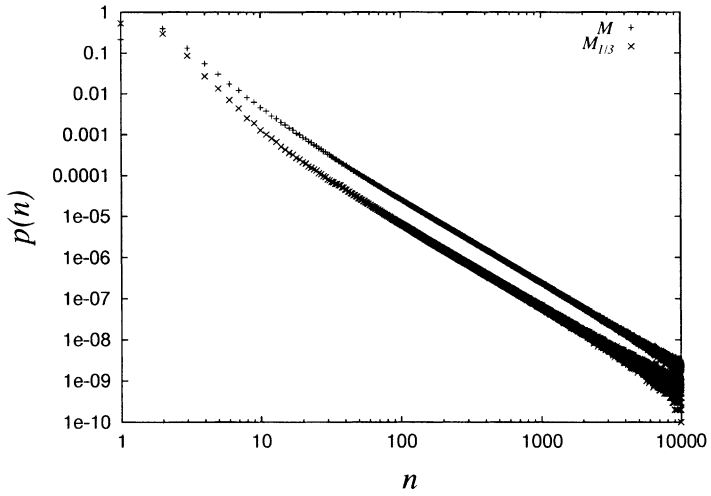

Fig. 17. Log–log plot of the halting time distribution $p(n)$ for M and $M_{1/3}$ (i.e., the fraction of the points that are decided not included in M or $M_{1/3}$ with computation time n). It decays with a power law (or slower) roughly as $n^{-2.0}$, in common with halting sets of UTM.

points that are decided not included in M or $M_{1/3}$ with computation time n. As shown in Fig. 17, $p(n)$ is found to decay with a power law (or slower) as $n^{-2.0}$.

Thus, decision procedure of both M and the halting set of UTM have long-time tail in the halting time distribution, and this property is also preserved under the application of a fractal function corresponding to the code transformation. These results are in a strong contrast with the construction process of ordinary fractals (i.e., transient chaos).

5. Riddled basin

Here we treat riddled basin structure, in the same manner. The riddled basin of a certain simple dynamical system will be shown to have different character, e.g., against "fractal" code transformations, from a halting set of a UTM and the Mandelbrot set.

5.1. Riddled basin

Basin of a class of chaotic attractors has a complex structure, called riddled basin, which was not observed for a simple low-dimensional dynamical system. The basin of an attractor is said to be "riddled" [24,25,40], when, for any point in the basin of the attractor, any of its ϵ-vicinity includes points with a non-zero volume in the phase space, that belong to another attractor's basin. Especially, when basins are riddled with each other, the basins are called "intermingled" [25].

Although an ordinary chaotic system with a single attractor exhibits unpredictability, its "qualitative" behavior is predictable that the state of the system tends to be confined to the attractor. However, in the case of a riddled basin, even such qualitative behavior (i.e., which attractor is the eventual one) is unpredictable by any small error because there is always a positive probability that an arbitrary small error will put the initial condition into another attractor's basin [40].

A simple model which has a riddled basin is introduced by Ott et al. [25]; a two-dimensional map of the region $0 \le x \le 1$, $0 \le y \le 1$ by

$$x_{n+1} = \begin{cases} \dfrac{1}{\alpha} x_n & \text{for } x_n < \alpha, \\ \dfrac{1}{1-\alpha}(x_n - \alpha) & \text{for } x_n > \alpha, \end{cases}$$

$$y_{n+1} = \begin{cases} \gamma y_n & \text{for } x_n < \alpha, \\ \delta y_n & \text{for } x_n > \alpha, \end{cases}$$

where $\gamma > 1$ and $0 < \delta < 1$. The map in the region $y > 1$ is chosen so that orbits in $y > 1$ move to an attractor in $y > 1$ and thus never return to $y < 1$.

The line segment $I = \{x, y | y = 0, 0 \le x \le 1\}$ is invariant for the map. I is a riddled basin chaotic attractor according to the definition of Milnor [41] if the perpendicular Lyapunov exponent

$$h_\perp = \alpha \ln \gamma + (1-\alpha) \ln \delta$$

is negative [25]. We concern basin structure of $y = 0$ attractor and $y > 1$ attractor on the horizontal line segment $\{x, y | y = y_0, 0 \le x \le 1\}$ with $0 < y_0 < 1$.

By assuming a randomly chosen initial x_0 in $(0, 1)$ with a specific initial y_0, we have a random walk in $\ln y$, starting at $\ln y_0$, where a step of size $-\ln \delta$ to the left has probability $1-\alpha$ while a step of size $\ln \gamma$ to the right has probability α [25]. The initial point (x_0, y_0) belongs to the $y > 1$ attractor's basin if the random walk ever reaches $\ln y \ge \ln 1 (= 0)$. On the other hand, (x_0, y_0) belongs to the $y = 0$ attractor's basin if, as $n \to +\infty$, $\ln y_n \to -\infty$ without reaching $\ln y_n \ge \ln 1$ for all n.

Let 'L' denote a step to left and 'R' to right, respectively. Then consider the set of all (sample) paths of random walk starting from $\ln y_0$ until it passes $\ln 1$ for the first time. A path of such a random walk is expressed by a symbol string on the alphabet $\{L, R\}$ (e.g., RR, $LRRR$, $LRLRR$, ...), and the set of all paths are decided by γ, δ and y_0. In case the ratio of step sizes between to the left and to the right is a rational number (i.e., $-\ln \delta / \ln \gamma$ is rational), a set of random walk paths until it passes $\ln 1$ for the first time can be accepted by a pushdown automaton.

Roughly speaking, a pushdown automaton (PDA) [2] is a non-deterministic finite automaton with a stack.

A stack is a one-way infinite memory, i.e., oneway sequence of cells on each of which one stack symbol can be written. We regard the end cell of a stack as "top". PDA consists of a finite control with a finite set of states Q, an input tape with an input alphabet Σ, and a stack with a stack alphabet Γ. At each transition, depending on the current finite control state, the input tape symbol read by input tape head, and the top stack symbol, PDA changes the finite control state, pops the top stack symbol off out of the stack and pushes a symbol string (or empty string) on the stack, and moves the input tape head one symbol ahead. Also, without an input tape symbol, PDA can change the finite control state and manipulate the stack in the same way as mentioned above, and in this case, the input tape head does not move. Thus, act of PDA is determined by transition function δ which maps each element of $Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma$ to a finite subset of $Q \times \Gamma^*$. At the beginning of PDA's movement, the finite control is prepared in the initial state, and the initial stack symbol is written on the top of its stack. A PDA accepts a symbol string written on its input tape if there exists a computation path where its stack is empty. The set of symbol strings accepted by a PDA is called context-free, and it is decidable (recursive) in contrast with a halting set of UTM.

Coming back to the problem of the riddled basin, it is easy to construct a pushdown automaton which accepts the paths of random walk until it passes $\ln 1$ for the first time when the ratio of step sizes is rational. [25] Thus, a set of these paths is a context-free language. In the following, we will call the set satisfying this condition, which is decided by (γ, δ, y_0) as random walk context-free language (RWCFL).

As a concrete example of RWCFL, we explain the case of $\gamma\delta = 1$ and $y_0 = \delta$ (RWCFL($\gamma\delta = 1, y_0 = \delta$)). Because $\gamma\delta = 1$, the step size to left $(-\ln\delta)$ and that to right $(\ln\gamma)$ are the same. Also, because $y_0 = \delta$, the random walk starts at $\ln\delta$, i.e., one step to the left of $\ln 1$. In this setting, the paths of random walk starting from $\ln y_0$ until it passes $\ln 1$ for the first time are R, LRR, $LRLRR$, $LLRRR$, ... The set of all these paths is RWCFL($\gamma\delta = 1, y_0 = \delta$). The Dyck language [2] which is a set of properly nested strings of brackets is well-known context-free language, and this RWCFL($\gamma\delta = 1, y_0 = \delta$) corresponds to the Dyck language by removing the right most R of these paths.

Although these paths are symbol strings, we can also regard these as symbol sequences because, once a random walk passes $\ln 1$, it will move to the $y > 1$ attractor no matter what symbols follow the original symbol string (e.g., $RR{*}{*}\cdots$, $LRRR{*}{*}\cdots$, $LRLRR{*}{*}\cdots$, ...). Thus, this simple dynamical system virtually executes encoding RWCFL(γ, δ, y_0) on the horizontal line segment $\{x, y | y = y_0, 0 \le x \le 1\}$ by using code α, as the basin of the $y > 1$ attractor.

5.2. Scaling

In [42], scaling property of the Dyck language using binary encoding is studied. The argument there is also valid in the case of our RWCFL. Let S be a subset of interval $[0, 1]$. A map $f : [0, 1] \to [0, 1]$ is a scaling of S when f preserves membership of S (i.e., $x \in S$ if and only if $f(x) \in S$). Thus, a scaling f corresponds to a copy of the original S.

In the case of Cantor sets treated previously, all scalings can be generated by composition of two functions f_0 and f_2, where $f_0(x) = \alpha_1 x$ and $f_2(x) = (1 - \alpha_2)x + \alpha_2$. Thus, f_0 and f_2 form a basis for the scalings of Cantor set.

On the other hand, in the case of our RWCFL (i.e., the riddled basin of the simple dynamical system), it is impossible to generate all scalings by composition of a finite number of functions, because there are symbol strings in RWCFL that cannot be represented by concatenation of smaller symbol strings of RWCFL. Thus, the set of scalings of RWCFL (the riddled basin of the simple dynamical system) does not have a finite basis, in contrast with Cantor sets.

5.3. Boundary dimension

Now we deal with dimensional property of such geometric sets. The riddled basin of $y = 0$ attractor of

[25] Basically such a PDA is constructed as follows: first, suppose $l, r \in \mathbf{Z}$, and let the ratio of l to r equal the ratio of the step size of 'L' to that of 'R'. Then, let the PDA push l stack symbols on the stack when its input tape head reads 'L', whereas let r stack symbols be popped off when 'R' is read.

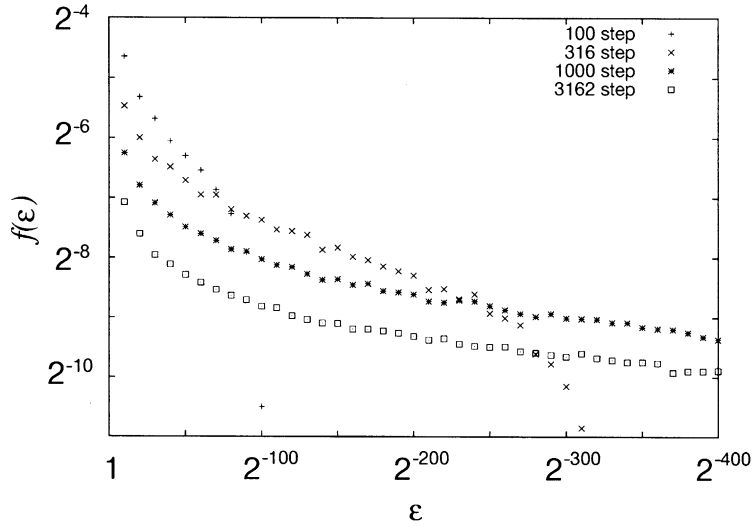

Fig. 18. Log–log plot of $f(\epsilon)$ (i.e., the fraction of subintervals with a boundary) with ϵ for computation time $n = 100$, 316, 1000 and 3162, for the RWCFL with code $\frac{1}{2}$. The slope of $f(\epsilon)$ becomes smaller with the increase of n.

the simple model is the complement of the geometric representation of RWCFL(γ, δ, y_0) with code α. Since the riddled basin is a fat fractal, we also consider the dimension of the boundary in this case, to study its fine structure. Of course, the boundary dimension of the riddled basin is equal to the boundary dimension of RWCFL(γ, δ, y_0) with code α.

The uncertainty exponent ϕ is calculated by the analysis of the simple model using diffusion approximation to simulate the random walk in [25]. The basic parameters of the diffusion approximation are the average drift per iterate, ν, and the diffusion per iterate, D. For the simple model, ν is equal to the perpendicular Lyapunov exponent, $\nu = h_\perp$, and $D = \frac{1}{2}\{\alpha(\ln\gamma - \nu)^2 + (1-\alpha)(\ln\delta - \nu)^2\} = \frac{1}{2}\alpha(1-\alpha)(\ln\gamma - \ln\delta)^2$. Under the condition $|\ln y_0| \gg 1$ and $|h_\perp| \ll \ln\gamma, -\ln\delta$, for the validity of the diffusion approximation, it is given by

$$\phi = \frac{h_\perp^2}{4Dh_\parallel},$$

where $h_\parallel = \alpha\ln(1/\alpha) + (1-\alpha)\ln(1/(1-\alpha))$ is the Lyapunov exponent for the dynamics in I [25].

Thus, according to the above formula, for given RWCFL(γ, δ, y_0) with rational $-\ln\delta/\ln\gamma$, one can change the boundary dimension of the geometric representation of the RWCFL, by changing α, i.e., the code transformation. In particular, if α_c satisfies $h_\perp = 0$ (i.e., $\alpha_c\ln\gamma + (1-\alpha_c)\ln\delta = 0$), the above formula implies that the boundary dimension of the RWCFL with code α_c (and only with that α_c) is equal to the space dimension. However, this equality is easily broken, in the present case, by code transformation (i.e., changing α). This fact is in strong contrast with

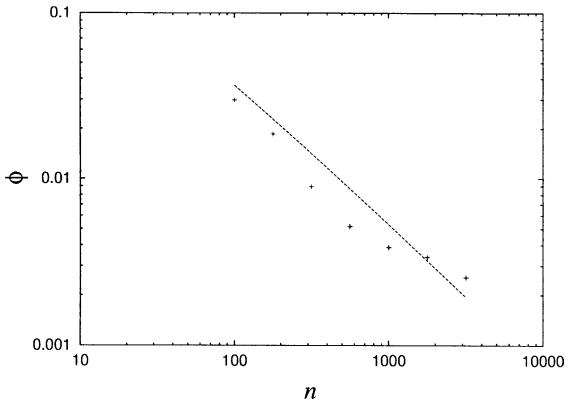

Fig. 19. Log–log plot of uncertainty exponent ϕ (i.e., $1-D_0$) of the RWCFL with code $\frac{1}{2}$ versus the computation time n, obtained from Fig. 18. It approaches zero roughly as n^{-1} with the increase of the computation time n. The dashed line shows an analytic estimate, given by $\phi = -[\log_2\pi^{-1/2}(-\arctan((n-1)/2)^{1/2} + \pi/2)]/n$.

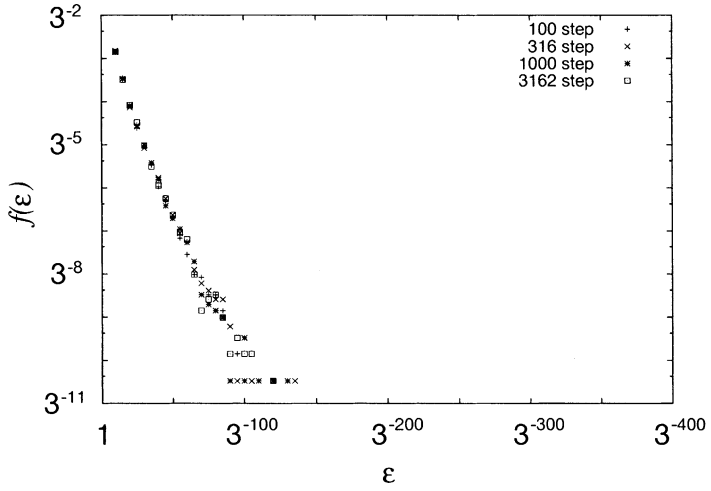

Fig. 20. Log–log plot of $f(\epsilon)$ of the RWCFL with code $\frac{1}{3}$ with ϵ for computation time $n = 100$, 316, 1000 and 3162. The boundary dimension remains less than one even with the increase of n. Note the difference from the case with code $\frac{1}{2}$ in Fig. 18.

the case of a halting set of a UTM and the Mandelbrot set.

If $h_\perp = 0$, however, $I = \{x, y | y = 0, 0 \leq x \leq 1\}$ is no longer an attractor even according to the definition of Milnor. As $h_\perp \to 0$ with $h_\perp < 0$, the measure of the riddled basin for the $y = 0$ attractor approaches zero (and the measure of the RWCFL with the corresponding code converges to one). Thus, even though the boundary dimension is equal to space dimension with the code α_c, the decision problem of the RWCFL virtually disappears since a point in the measure zero set cannot actually be chosen. This fact is also in strong contrast with the case of the halting set of a UTM and the Mandelbrot set.

As an example, we numerically examine the boundary dimension of the RWCFL such that $\gamma\delta = 1$, $y_0 = \delta$ (that corresponds to the Dyck language) with code $\frac{1}{2}$ and code $\frac{1}{3}$ (i.e., $\alpha = \frac{1}{2}$ and $\alpha = \frac{1}{3}$, respectively). Here, we also treat the set of paths which are decided to be in the RWCFL within a given finite step n. In other words, the set of initial points which are decided in $y > 1$ attractor's basin within n steps is computed, with the same procedure as before.

Fig. 18 shows the asymptotic behavior of the boundary dimension for $\alpha = \frac{1}{2}$ with the increase of n, where $f(\epsilon)$, the fraction of subintervals including a boundary, is plotted with a logarithmic scale for several values of n.

Fig. 19 shows the log–log plot of ϕ (i.e., $1 - D_0$) versus n. As is shown, the box-counting dimension of the boundary approaches the space dimension one roughly as $1 - cn^{-1}$ with the increase of the (computation) time n,[26] similarly with the case of a halting set of a UTM and the Mandelbrot set.

On the other hand, in Fig. 20, the boundary dimension is studied for $\alpha = \frac{1}{3}$ for several values of (computation) time n. This set is transformed from the set with code $\frac{1}{2}$ by the code transformation $t_{1/2\to 1/3}$. Unlike the case of $\alpha = \frac{1}{2}$ (binary encoding), Fig. 20 shows no asymptotic approach of ϕ to zero with the increase of n. Thus, the boundary dimension is estimated to be less than one even in the limit of $n \to \infty$. In terms of the uncertainty exponent, the probability of making a mistake (equally $V[S(\epsilon)]$) can be decreased to any amount, in principle, by a decrease in ϵ. Thus,

[26] The boundary dimension of the RWCFL($\gamma\delta = 1$, $y_0 = \delta$) with $\alpha = \frac{1}{2}$ is estimated to be

$$D_0 \approx \lim_{m\to\infty} \frac{\log_2(1/\sqrt{\pi}(-\arctan m^{1/2} + (\pi/2))}{2m+1} + 1,$$

by adopting a random walk representation, see Appendix A.

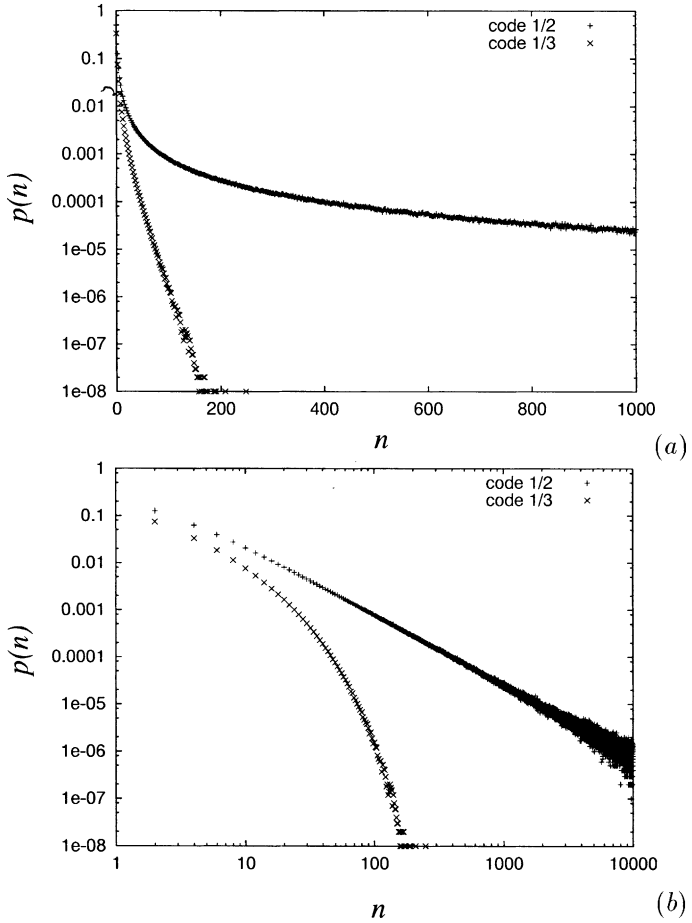

Fig. 21. (a) Semi-log plot and (b) log–log plot of the halting time distribution $p(n)$ with n for the RWCFL with both code $\frac{1}{2}$ and code $\frac{1}{3}$. $p(n)$ with code $\frac{1}{2}$ is found to decay with a power law as $n^{-3/2}$, in the same way as the halting set of UTM and the Mandelbrot set. On the other hand, $p(n)$ with code $\frac{1}{3}$ is found to decay exponentially with time n, as in an ordinary fractal set.

the decision procedure of this set does not have the strong uncertainty mentioned previously.

Now, it is confirmed that one can change the boundary dimension of the geometric representation of the RWCFL by changing code. In particular, even if the boundary dimension of the RWCFL with code α_c (and only with that α_c) is equal to the space dimension, this property is easily broken by code transformation (i.e., changing α), unlike the halting set of a UTM and the Mandelbrot set.

After all, there exist various classes in non-self-similar sets that are not simply characterized by self-similarity, especially in sets that have the boundary dimension equal to the space dimension. RWCFL, an example of context-free language, is ranked as "middle" between a halting set of a UTM and a self-similar set like Cantor set which corresponds to a regular language.

So far, we have fixed an RWCFL and changed α (code). Now, let us change RWCFL by fixing α (code), i.e., a Markov partition of a piecewise-linear map characterized by α. For given α, there is a pair (γ, δ) such that $\alpha \ln \gamma + (1-\alpha) \ln \delta$ is equal to, or arbitrarily close to, zero, satisfying $-\ln \delta / \ln \gamma$ is rational. Hence the geometric representation of the RWCFL specified by these γ and δ with appropriate y_0 using code α has

boundary dimension equal to, or arbitrarily close to, the space dimension.[27] Thus, for any code α or any Markov partition of a piecewise-linear map, there is RWCFL such that boundary dimension is equal to, or arbitrarily close to, the space dimension.

Now, let us reconsider a halting set of a UTM. Basically, a halting set of a UTM contains all halting sets of TM. Hence it naturally contains all RWCFL. In this case, although the boundary dimension of each individual RWCFL is varied by the code transformation, within the halting set of UTM there is an RWCFL, whose boundary dimension is equal to, or arbitrarily close to the space dimension for an arbitrary code into a Markov partition of a piecewise-linear map. Thus, it is realized from this point that the boundary dimension of the halting set of a UTM is equal to, or arbitrarily close to the space dimension for an arbitrary code into a Markov partition of a piecewise-linear map.

5.4. Halting time distribution

Here, we study the decision procedure of RWCFL, i.e., $y > 1$ attractor's basin. Again, we have numerically studied the halting time distribution $p(n)$ for the RWCFL such that $\gamma\delta = 1$, $y_0 = \delta$ (corresponds to the Dyck language) with code $\frac{1}{2}$ and code $\frac{1}{3}$ (i.e., $\alpha = \frac{1}{2}$ and $\alpha = \frac{1}{3}$, respectively), defined as the fraction of the points that are decided to be included in the RWCFL with (computation) time n.

As shown in Fig. 21, $p(n)$ of $\alpha = \frac{1}{2}$ is found to decay with a power law as $n^{-3/2}$ (see Appendix A). Thus, in this case with boundary dimension equal to space dimension, the decision procedure of $\alpha = \frac{1}{2}$ shows the same behavior as a halting set of a UTM and the Mandelbrot set, from the aspect of halting time distribution.

On the other hand, in the case of $\alpha = \frac{1}{3}$, $p(n)$ is found to decay exponentially with time n. Thus, the decision procedure of $\alpha = \frac{1}{3}$ without strong uncertainty, shows the same behavior as ordinary fractal sets, as for the halting time distribution.

[27] Note that RWCFL signifies the way to choose labels of Markov partition for letting boundary dimension equal to, or arbitrarily close to, the space dimension.

Hence, it is shown that the power law decay of $\alpha = \frac{1}{2}$ is easily broken by the code transformation (i.e., changing α), unlike a halting set of a UTM and the Mandelbrot set.

6. Converted (universal) Turing machine

6.1. Adding 'ϵ' symbol

Now we consider modification of a UTM by adding 'ϵ' symbol to the original tape alphabet of the UTM, where 'ϵ' symbol does nothing: suppose that 'ϵ' symbol is written on a cell of the tape of a TM and the tape head of the TM moves to this cell from the left neighbor cell, with the finite control state q_i. Then, at the next time step, the tape head of the TM moves to the right neighbor cell without changing anything, i.e., this TM leaves the 'ϵ' symbol on the cell as it is, and shifts its tape head to the right with the finite control state q_i. It is similarly defined when the tape head moves from the right side to the left side.

Of course, even if a UTM is converted in this way by adding this 'ϵ' symbol, the resulting TM is also universal. Practically, we consider ϵUTM obtained from Rogozhin's UTM(24, 2) by adding 'ϵ' symbol to the tape alphabet of Rogozhin's UTM(24, 2), and investigate both the boundary dimension and the halting time distribution of the halting set of ϵUTM, similarly as the previous cases.

In Fig. 22, the log–log plot of $f(\epsilon)$ of ϵUTM using base-3 code (by transforming the tape alphabet $\{0, 1, \epsilon\}$ into $\{0, 1, 2\}$) is shown for several values of the computation time n. In this case also, ϕ approaches zero roughly as $n^{-0.40}$, as in the previous cases. Thus the boundary dimension of the geometric representation of the halting set of ϵUTM using base-3 code, is also estimated to be two, i.e., the dimension of the space. The decision procedure of ϵUTM possesses inaccessibility similarly as the previous cases of UTMs.

In Fig. 23, the halting time distribution of ϵUTM with base-3 code is plotted. This also shows the power law property ($p(n) \sim n^{-2.6}$).

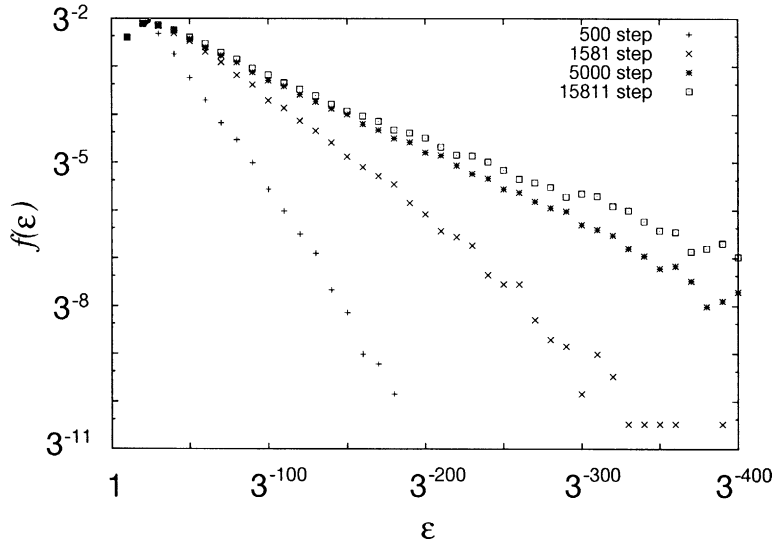

Fig. 22. Log–log plot of $f(\epsilon)$ (the fraction of squares with a boundary) of ϵUTM using base-3 code, with ϵ for computation time $n = 500$, 1581, 5000 and 15 811. The slope of $f(\epsilon)$ becomes smaller with the increase of n.

6.2. Changing halting state

Now we modify a UTM by changing its halting state. For a given UTM, a "broken" (U)TM is defined by changing one of the finite control states of the given UTM as a new halting state for the "broken" (U)TM. Also, this "broken" (U)TM has the same transition function and tape alphabet as the original UTM. We say "broken" (U)TM at q_i if the new halting state for the "broken" (U)TM is q_i. (Because this TM has the same (number of) internal states and tape symbols as the original UTM, it is appropriate as a contrast study.)

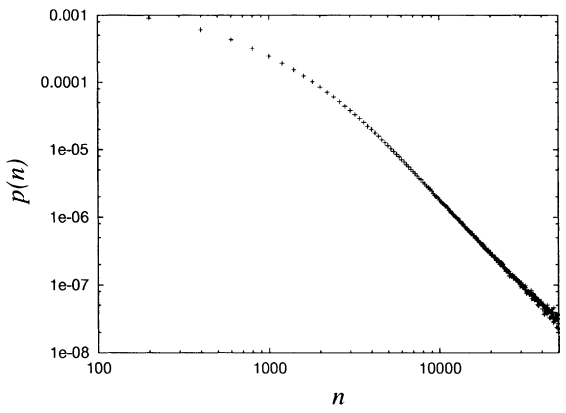

Fig. 23. Log–log plot of the halting time distribution of ϵUTM obtained by using base-3 code. $p(n)$ (i.e., the fraction of the initial points halting with computation time n) also decays with a power law ($p(n) \sim n^{-2.6}$), similarly as the cases of UTMs.

By this modification, it is prospective that the "broken" (U)TM is no longer a UTM, and that the halting problem of this "broken" (U)TM becomes decidable. However, it is, of course, possible that the halting problem of this "broken" (U)TM is still undecidable. For example, it may be possible that this "broken" (U)TM, with a new description on the initial tape, becomes a new UTM.

Now, we study both the boundary dimension and the halting time distribution of geometric representation of the halting set of "broken" (U)TMs obtained from Rogozhin's UTM(24, 2) and Minsky's UTM(7, 4), similarly as above.

6.2.1. "Broken" (U)TMs obtained from Rogozhin's UTM(24, 2)

Since Rogozhin's UTM(24, 2) has 24 internal states besides a halting state, the number of possible "broken" (U)TMs obtained from Rogozhin's UTM(24, 2) is 23, besides trivial destruction at the

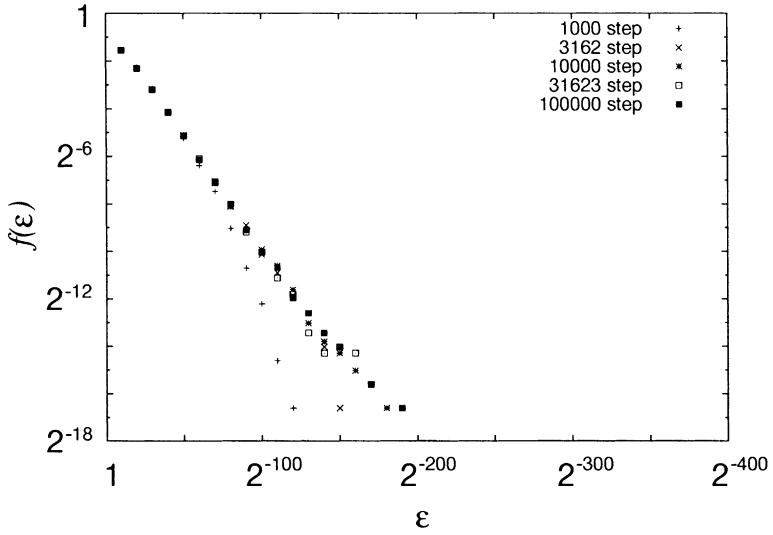

Fig. 24. Log–log plot of $f(\epsilon)$ of "broken" (U)TM at q_{16} using code $\frac{1}{2}$, with ϵ for computation time $n = 1000$, 3162, 10 000, 31 623 and 100 000. The boundary dimension remains less than the space dimension even with the increase of n. Note the difference from the case of Rogozhin's UTM(24, 2) with code $\frac{1}{2}$ in Fig. 4.

initial state q_1. We have numerically studied the boundary dimension and the halting time distribution of these "broken" (U)TMs, by using code $\frac{1}{2}$ (i.e., base-2 code). From the numerical results, these "broken" (U)TMs with code $\frac{1}{2}$ can roughly be classified into three groups:

Group 1. The boundary dimension approaches the space dimension and the halting time distribution obeys a power law distribution.

Group 2. The boundary dimension remains less than the space dimension and the halting time distribution decays much faster than that of original Rogozhin's UTM(24, 2) and group 1.

Group 3. It shows the same characteristics as those in group 2, but the Lebesgue measure of the geometric representation of the halting set is estimated to be the total measure, two, and the complement is a skinny fractal, in contrast with the case of group 1 and group 2.

"Broken" (U)TMs at $q_{18}, q_{20}, q_{21}, q_{22}$, and q_{23} are found to be included in group 1. The boundary dimension of the geometric representation of the halting set of these "broken" (U)TMs approaches, and is estimated to be, two, i.e., the space dimension. Also, the halting time distribution of these "broken" (U)TMs decays according to a power law. Thus the characteristics of the original UTM are preserved.

"Broken" (U)TMs at $q_5, q_6, q_7, q_8, q_9, q_{10}, q_{13}, q_{15}, q_{16}, q_{17}, q_{19}$, and q_{24} are found to be included in group 2. As an example, in Fig. 24, the log–log plot of $f(\epsilon)$ of "broken" (U)TM at q_{16} is shown for several values of the computation time n. The boundary dimension of the geometric representation of the halting set of "broken" (U)TM at q_{16} using code $\frac{1}{2}$, remains less than the space dimension with the increase of n. Thus, the boundary dimension is estimated to be less than the space dimension even in the limit of $n \to \infty$. In Fig. 25, the halting time distribution of "broken" (U)TM at q_{16} with code $\frac{1}{2}$ is plotted. It is found to decay much faster than that of Rogozhin's UTM(24, 2) with code $\frac{1}{2}$.

The rest of "broken" (U)TMs (those at $q_2, q_3, q_4, q_{11}, q_{12}$, and q_{14}) are found to be included in group 3. These show the same behavior as in group 2, but, here, the Lebesgue measure of the geometric representation of the halting set is estimated to be two that is the total Lebesgue measure of both square 0 and square 1.

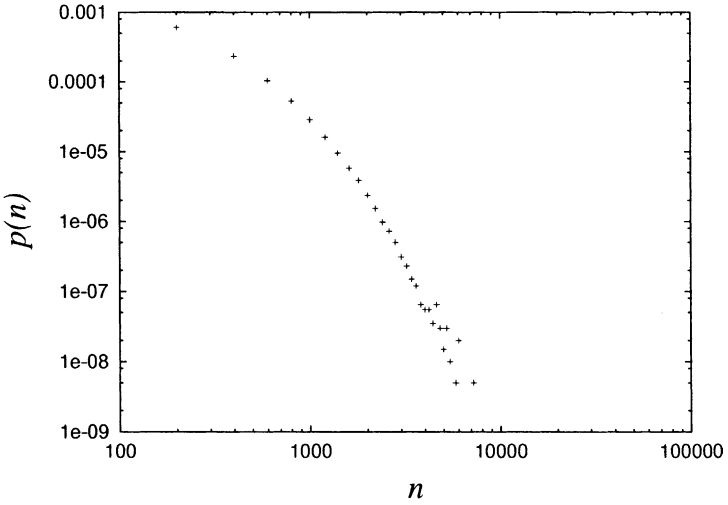

Fig. 25. Log–log plot of $p(n)$ with n in the case of "broken" (U)TM at q_{16} using code $\frac{1}{2}$. This $p(n)$ is found to decay much faster than that of Rogozhin's UTM(24, 2) with code $\frac{1}{2}$ in Fig. 9.

In other words, the Lebesgue measure of non-halting inputs is estimated to be zero, in contrast with the case of groups 1 and 2.

To sum up, the decision procedure of the halting sets of the "broken" (U)TMs in group 1 has inaccessibility to the ideal decision procedure, in the presence of error. Thus, the halting problem of the "broken" (U)TMs in group 1 is expected to be undecidable. On the other hand, for the groups 2 and 3, the UTM is broken, and the TM is expected to lose undecidability.

6.2.2. "Broken" (U)TMs obtained from Minsky's UTM(7, 4)

Since Minsky's UTM(7, 4) has seven internal states besides a halting state, the number of possible "broken" (U)TMs obtained from Minsky's UTM(7, 4) is six. We have numerically studied the boundary dimension and the halting time distribution of these "broken" (U)TMs, by using base-4 code. In contrast with "broken" (U)TMs from Rogozhin's UTM(24, 2), all the "broken" (U)TMs from Minsky's UTM(7, 4) are found to belong to groups 2 or 3. No "broken" (U)TM exhibits an approach of boundary dimension to the space dimension. Thus all the TMs are expected to lose undecidability (and to be no longer universal) with the change of table.

The difference between the present "broken" (U)TMs and those from Rogozhin's UTM(24, 2) is understood as follows. As a measure of complexity of TM, the number of commands of TM, given by the product of the number of the internal states and the number of the symbols in the tape alphabet, is proposed by Shannon [43]. This complexity measure is equal to 48 for Rogozhin's UTM(24, 2) and 28 for Minsky's UTM(7, 4). By changing q_i as a new halting state, the commands at q_i cannot be used for the "broken" (U)TM at q_i, and the above complexity measure for the "broken" (U)TM is equal to 46 and 24, respectively. Therefore, "damage" from destruction of a internal state is more serious to the "broken" (U)TMs obtained from Minsky's UTM(7, 4).

Indeed, another measure of complexity of TM is proposed, as the number of commands really used by TM. It is not more than 46 and 24, respectively. However, the least known number of commands for being UTM is 22, that is obtained from UTM with four internal states and six tape symbols [33]. The number for the "broken" (U)TM obtained from Minsky's UTM(7, 4) is much closer to 22 than that from Rogozhin's UTM(24, 2). Hence, it is expected

that "broken" (U)TMs from Minsky's UTM(7, 4) lose undecidability.

7. Discussion: analog computation and code

7.1. Undecidability in analog computation

In Section 1.4, we mentioned physical realizability of models of computation. Indeed several analog computation models have been studied so far (e.g. [18–22]), but each model defines each computability, and relation of each other is not made so clear.

On the other hand, from a viewpoint of physical realizability, the (in)accessibility is considered to give one condition for any models of computation. Note that we have shown inaccessibility of undecidable decision procedure in the sense of the Turing model. Here, we propose that this inaccessibility holds for any other models with undecidability and propose the following statement: *in a model of analog computation, a procedure should be uncomputable (undecidable), if it has inaccessibility in the sense that the ideal procedure is not approached in the presence of error*.

This proposition is based on the following consideration: from a viewpoint of physical realizability, it is actually impossible to avoid errors (noise) in computational operation and observation. Thus, it is irrelevant to count a procedure with the inaccessibility as computable, since one cannot even approach the ideal procedure on which the model is founded, in the presence of error. Such procedure having the inaccessibility should be included in uncomputable procedures.

Concerning the Mandelbrot set M, Penrose suggests that M (and also the complement of M) is not "recursive" (i.e., "undecidable") [38]. By our condition of physical realizability, M should also be undecidable. A remark should be made here: Blum, Shub, and Smale have already pointed out and proved the undecidability of M over $\mathbf{R}$ according to their well-known formulation of analog computation [18]. However, all fractals (like a typical Julia set) are also undecidable according to their formulation. In our criterion that takes into account of the precision and the inaccessibility, we can properly distinguish ordinary fractals from more complex sets that are hard to handle.

7.2. Class of appropriate codes

About the coding, we have introduced the mapping from a symbol sequence to a real number represented by the same label of Markov partition of a certain piecewise-linear map, so that the effect of farther cells gets smaller in the real number. Otherwise, it is impossible to discuss the distance in the symbol sequence, necessary to consider geometric and dynamical systems properties. Conversely, such coding may be required to construct an analog computation machine from a dynamical system. However, the question about the appropriate condition for coding remains unanswered yet, and a wider class of codes seems to exist.

As the first step to address this question, we show that use of too general codes are meaningless (1). Then we point out that even within a computable transformation, there is a case that inaccessibility is not preserved (2). Then we discuss possible limitation on the ability to detect decidability, by our codes into a Markov partition of a piecewise-linear map (3). Finally, following the above arguments, we discuss a certain class of codes that we want to clarify in future studies (4).

1. First of all, use of all codes (i.e., all mappings from a symbol sequence to a real number) is too general.

   For example, there is a code which solves a halting problem of a UTM. Consider a code which first maps a symbol sequence to another symbol sequence, and then maps the resulting symbol sequence to a real number. Its detail is given as follows. If a given symbol sequence **s** is included in the halting set of the UTM, then **s** is transformed to the concatenated symbol sequence **1s**, and then **1s** is transformed to a real number given by, e.g., the base-n encoding. Otherwise **s** is transformed to **0s**, and then **0s** is transformed to a real number given by the same code.

   This code executes computation using the same computational power as a TM with a halting problem oracle [2], at the stage of mapping a symbol sequence to another symbol sequence. As a result,

an undecidable set can be constructed from a decidable set at this stage, and this code does not preserve decidability.

Of course, if this code is used, it is impossible to distinguish a geometric representation of a decidable set from that of an undecidable set. In particular, there is a decidable set whose dimensional property is the same as undecidable sets.

Not only the halting problem of UTM but also all decision problems can be solved only by the coding, in the similar way as above (this is mere transfer of computation to code). It is not at all realistic that all computation can be executed only by codes, and therefore it is necessary to restrict a class of codes, instead of considering all codes.

2. Following (1), let us restrict the mapping from a symbol sequence to another symbol sequence within computable transformations t_c. A computable transformation $t_c : x \in \Sigma^* \mapsto t_c(x) \in \Sigma'^*$ is defined as injection satisfying $t_c(x)$ to x as well as x to $t_c(x)$ is computable and $t_c(\Sigma^*)$ is decidable. This computable transformation t_c conserves both undecidability and decidability of sets (i.e., for a given $S \subset \Sigma^*$, $t_c(S)$ is undecidable if and only if S is undecidable), in contrast with the code in (1) that solves a halting problem of UTM.

   Within the codes into a Markov partition of a piecewise-linear map, we have shown inaccessibility and its invariance against code transformations, for the halting problem of UTM. The question, then, is if these properties are preserved for all the computable transformations. Unfortunately, this is not the case.

   Indeed, let us consider a geometric representation of the halting set of a UTM, obtained by first mapping a symbol string in the halting set using a computable transformation t_c and then by mapping the resulting symbol string using our code into a Markov partition of a piecewise-linear map. For example, suppose a halting set of a UTM on tape alphabet $\{s_0, s_1\}$ is given, and consider adding a new tape symbol 's_{new}' to the tape alphabet. In this setting, if we transform $\{s_0, s_{new}, s_1\}$ to $\{0, 1, 2\}$ (this transformation is a computable transformation) and apply 3-symbol code of a piecewise-linear map to the set, then the boundary of the geometric representation of the halting set is embedded in a Cantor set. This is equivalent with modifying the UTM by adding unhalt symbol 'u'.[28] Similarly we can embed the boundary to a Cantor set by increasing redundancy, e.g., by transforming the halting set of the UTM by changing $s_0 \to s_0 s_0$ and $s_1 \to s_1 s_1$ (this transformation is also a computable transformation).

   In the above cases, the boundary dimension is less than the space dimension. Thus, it is impossible to require each decision procedure of all undecidable sets to be inaccessible since above two examples of a code are considered to be appropriate.

3. On the other hand, contrary to above facts, possibly there could be a decidable language (not shown yet) that "outwit" our codes into a Markov partition of a piecewise-linear map, and its decision procedure of geometric representation possesses inaccessibility, with those codes.

   Even if this turns out true, the class of codes adopted in the present paper is restricted too much, and one should find a class of appropriate codes, that is broader than those by a Markov partition of a piecewise-linear map.

   Here, it seems that what code is appropriate will not be derived from computation theory but will be determined by our "sense", in particular, on dynamical system. Indeed, even the simple 2-symbol code α into a Markov partition of a piecewise-linear map cannot be appropriate from computation theory, if α is an uncomputable real number. However, as for the boundary dimension, e.g., there is almost no difference between uncomputable real α and its neighboring rational, as seen in this paper.

4. It will be important to determine the class of appropriate codes. Following the argument so far, we

[28] Similarly if halt symbol 'h' is chosen as 's_{new}', then the halting set is modified by adding the regular set of the symbol strings on the alphabet $\{s_0, h, s_1\}$, whose symbol strings end with the first symbol 'h'. This resulting set is, of course, undecidable (but note, however, that this modification of the set does not satisfy the above condition for computable transformation of symbol string). Since the added regular set corresponds to the complement of the Cantor set, the boundary is also embedded in the Cantor set.

want to investigate realizability of the following class of codes in future studies:

For $S \subset \Sigma^*$, S is undecidable if and only if there exists at least one computable transformation t_c: $\Sigma^* \to \Sigma'^*$ such that decision procedure of $t_c(S)$ with any of the codes in the class, is inaccessible.

Here we have considered a computable transformation as a "filter" for transformations that lower boundary dimension, as mentioned in above examples of (2). For at least one set (which can be said "concentrated") included in all undecidable sets derived from the original undecidable set, we require decision procedure of the set to be inaccessible, with any codes in the class. At the same time, for any of decidable sets, we require decision procedure of the set to be accessible, with at least one of codes in the class. Our codes into a Markov partition of a piecewise-linear map are considered to be included in this class.

To clarify this class of coding might be directly connected with the explicit statement of the "physical" Church–Turing thesis, that any physically realizable computers cannot be more powerful than the classical models of computation [44]. For example, if this class exactly coincides with our possible class of codes, then it seems to be difficult to utilize analog states to do more than the discrete TM, and this fact will seem to support the "physical" Church Turing thesis.

8. Conclusion

Our main results in this paper are as follows:

- Boundary dimension equal to space dimension indicates that decision procedure of a set has so strong uncertainty that one cannot approach the ideal decision procedure, in the presence of error. Decision procedure of geometric representation of halting set of a UTM has this strong uncertainty, and this property is preserved under code transformations. A characterization of undecidability of the halting problem of a UTM is given by the strong uncertainty implying inaccessibility to the ideal decision procedure, and by invariance of the strong uncertainty against code transformations.
- The Mandelbrot set, known as a set with extraordinarily complex structure, can be connected with undecidable sets by both the inaccessibility and its invariance against "fractal" function which corresponds to a code transformation.
- Riddled basin of a certain simple dynamical system, having different unpredictability from chaotic unpredictability, represents geometrically a certain context-free language. The riddled basin is ranked as middle between an ordinary fractal and a halting set of a UTM or the Mandelbrot set. There exist various classes in non-self-similar sets that are not simply characterized by self-similarity, especially in sets that have boundary dimension equal to space dimension.

Acknowledgements

The authors would like to thank M. Taiji, T. Ikegami, S. Sasa, O. Watanabe, M. Shishikura and I. Shimada for useful discussions. This work is partially supported by a Grant-in-Aid for Scientific Research from the Ministry of Education, Science, and Culture of Japan.

Appendix A

Both the boundary dimension and the halting time distribution for the RWCFL($\gamma\delta = 1, y_0 = \delta$) with $\alpha = \frac{1}{2}$ can be estimated by considering random walks. The number of paths starting from -1 at time 0 and reaching 0 first at time $2m+1$ is $(1/(m+1))\binom{2m}{m}$ [45]. Thus, the halting time distribution $p(2m+1)$, i.e., the fraction of the points that are decided to be in the $y > 1$ attractor's basin at time $2m+1$, is

$$p(2m+1) = \frac{1}{m+1}\binom{2m}{m}2^{-(2m+1)} \approx \frac{1}{2\sqrt{\pi}}\frac{1}{m^{3/2}+m^{1/2}}.$$

The measure of the $y > 1$ attractor's basin is $\lim_{N\to\infty}\sum_{m=0}^{N} p(2m+1)$. This is equal to 1 that is

known as ruin probability [45] (i.e., the complement of the $y > 1$ attractor's basin is a skinny fractal). From this, the total measure of the points that are decided to be in the $y > 1$ attractor's basin up to time $2m + 1$ is approximated to be $-1/\sqrt{\pi}(-\arctan m^{1/2} + \pi/2) + 1$.

Now, cover the interval $[0, 1]$ by a grid of subintervals, where the length of subintervals is $\epsilon = 2^{-(2m+1)}$. The number of subintervals needed to cover the boundary (i.e., the skinny fractal) is estimated $N(\epsilon) \approx (1/\sqrt{\pi})(-\arctan m^{1/2} + \pi/2)2^{2m+1}$. Thus, the box-counting dimension of the boundary is

$$D_0 = \lim_{\epsilon \to 0} -\frac{\log N(\epsilon)}{\log \epsilon} \approx \lim_{m \to \infty} \frac{\log_2 (1/\sqrt{\pi})(-\arctan m^{1/2} + \pi/2)}{2m + 1} + 1.$$

These results of both the boundary dimension and the halting time distribution agree well with the numerical experiments.

References

[1] A.M. Turing, On computable numbers with an application to the Entscheidungsproblem, Proc. London Math. Soc. 42 (1936) 230.

[2] J.E. Hopcroft, J.D. Ullman, Introduction to Automata Theory, Languages and Computation, Addison-Wesley, Reading, MA, 1979.

[3] M.L. Minsky, Computation: Finite and Infinite Machines, Prentice-Hall, Englewood Cliffs, NJ, 1967.

[4] M. Davis, Computability & Unsolvability, Dover, New York, 1982.

[5] B.B. Mandelbrot, The Fractal Geometry of Nature, Freeman, New York, 1983.

[6] M.F. Barnsley, Fractals Everywhere, Academic Press, Boston, 1988.

[7] K.J. Falconer, The Geometry of Fractal Sets, Cambridge University Press, Cambridge, 1985.

[8] E. Ott, Chaos in Dynamical Systems, Cambridge University Press, Cambridge, 1993.

[9] J. Guckenheimer, P. Holmes, Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields, Springer, New York, 1983.

[10] R.L. Devaney, A First Course in Chaotic Dynamical Systems: Theory and Experiment, Addison-Wesley, Reading, MA, 1992.

[11] R.L. Devaney, An Introduction to Chaotic Dynamical Systems, Addison-Wesley, Redwood City, CA, 1989.

[12] E. Atlee Jackson, Perspectives of Nonlinear Dynamics, Cambridge University Press, Cambridge, 1991.

[13] S. Wolfram, Undecidability and Intractability in Theoretical Physics, Phys. Rev. Lett. 54 (1985) 735.

[14] A. Saito, K. Kaneko, Geometry of undecidable systems, Prog. Theor. Phys. 99 (1998) 885.

[15] C. Moore, Generalized shifts: unpredictability and undecidability in dynamical systems, Nonlinearity 4 (1991) 199

[16] C. Moore, Unpredictability and undecidability in dynamical systems, Phys. Rev. Lett. 64 (1990) 2354.

[17] I. Shimada, Talk at International Symposium on Information Physics 1992, Kyushu Institute of Technology.

[18] L. Blum, et al., Complexity and Real Computation, Springer, New York, 1998.

[19] M.B. Pour-El, J.I. Richards, Computability in Analysis and Physics, Springer, Berlin, 1989.

[20] H.T. Siegelmann, E.D. Sontag, Analog computation via neural networks, Theor. Comput. Sci. 131 (1994) 331

[21] H.T. Siegelmann, Neural Networks and Analog Computation: Beyond the Turing Limit, Birkhauser, Boston, 1999.

[22] C. Moore, Recursion theory on the reals and continuous-time computation, Theor. Comput. Sci. 162 (1996) 23.

[23] H.T. Siegelmann, S. Fishman, Analog computation with dynamical systems, Physica D 120 (1998) 214.

[24] E. Ott, et al., Scaling behavior of chaotic systems with riddled basins, Phys. Rev. Lett. 71 (1993) 4134.

[25] E. Ott, et al., The transition to chaotic attractors with riddled basins, Physica D 76 (1994) 384.

[26] T. Kamae, S. Takahashi, Ergodic Theory and Fractals, Springer, Tokyo, 1993.

[27] G. de Rham, Sur quelques courbes definies par des equations fonctionnelles, Rend. Sem. Mat. Torino 16 (1957) 101.

[28] P. Grassberger, Generalized dimensions of strange attractors, Phys. Lett. A 97 (1983) 227 .

[29] N. Chomsky, Three models for the description of language, IRE Trans. Inform. Theory 2 (1956) 113.

[30] N. Chomsky, On certain formal properties of grammars, Inform. Cont. 2 (1959) 137.

[31] L. Staiger, ω-Languages, in: G. Rozenberg, A. Salomaa (Eds.), Handbook of Formal Languages, Vol. 3, Springer, Berlin, 1997.

[32] W. Thomas, Automata on infinite objects, in: J. van Leeuwen (Ed.), Handbook of Theoretical Computer Science B, Elsevier, Amsterdam, 1990.

[33] Y. Rogozhin, Small universal Turing machines, Theor. Comput. Sci. 168 (1996) 215.

[34] J.D. Farmer, Sensitive dependence on parameters in nonlinear dynamics, Phys. Rev. Lett. 55 (1985) 351.

[35] C. Grebogi, et al., Exterior dimension of fat fractals, Phys. Lett. A 110 (1985) 1.

[36] G.J. Chaitin, Information-theoretic Incompleteness, World Scientific, Singapore, 1992.

[37] M. Li, P.M.B. Vitanyi, An Introduction to Kolmogorov Complexity and its Applications, Springer, New York, 1997.

[38] R. Penrose, The Emperor's New Mind, Oxford University Press, Oxford, 1989.

[39] M. Shishikura, The boundary of the Mandelbrot set has Hausdorff dimension two, Complex analytic methods in dynamical systems (Rio de Janeiro, 1992), Asterisque 222 (1994) 389.

[40] J.C. Sommerer, E. Ott, A physical system with qualitatively uncertain dynamics, Nature 365 (1993) 138.
[41] J. Milnor, On the concept of attractor, Commun. Math. Phys. 99 (1985) 177.
[42] C. Moore, Generalized one-sided shifts and maps of the interval, Nonlinearity 4 (1991) 727.
[43] C.E. Shannon, Automata Studies, Princeton University Press, Princeton, NJ, 1956.
[44] R. Penrose, Shadows of the Mind, Oxford University Press, Oxford, 1994.
[45] W. Feller, An Introduction to Probability Theory and its Applications, Wiley, New York, 1950.