2019 57th Annual Allerton Conference on Communication, Control, and Computing
(Allerton)
Allerton Park and Retreat Center
Monticello, IL, USA, September 24-27, 2019

Algorithmic Randomness in Continuous-Time Markov Chains*

Xiang Huang[1], Jack H. Lutz[1], and Andrei N. Migunov[1]

***Abstract*— In this paper we develop the elements of the theory of algorithmic randomness in continuous-time Markov chains (CTMCs). Our main contribution is a rigorous, useful notion of what it means for an *individual trajectory* of a CTMC to be *random*. CTMCs have discrete state spaces and operate in continuous time. This, together with the fact that trajectories may or may not halt, presents challenges not encountered in more conventional developments of algorithmic randomness.**

Although we formulate algorithmic randomness in the general context of CTMCs, we are primarily interested in the *computational* power of stochastic chemical reaction networks, which are special cases of CTMCs. This leads us to embrace situations in which the long-term behavior of a network depends essentially on its initial state and hence to eschew assumptions that are frequently made in Markov chain theory to avoid such dependencies.

After defining the randomness of trajectories in terms of a new kind of martingale (algorithmic betting strategy), we prove equivalent characterizations in terms of constructive measure theory and Kolmogorov complexity.

***Index Terms*— algorithmic randomness, continuous time Markov chain, chemical reaction network**

I. Introduction

In this paper we develop the elements of the theory of algorithmic randomness in continuous-time Markov chains (CTMCs). Specifically, our main contribution is a rigorous, useful notion of what it means for an *individual trajectory* (also called a single orbit) of a CTMC C to be *random* with respect to C and an initial state–or probability distribution of initial states–of C. This is a first step toward carrying out *Kolmogorov's program* of replacing probabilistic laws stating that *almost every* trajectory has a given property with randomness laws stating that *every random* trajectory has the property. More generally, we are initiating an algorithmic "single orbit" approach (in the sense of Weiss [25]) to the dynamics of CTMCs. In a variety of contexts ranging from Bernoulli processes to ergodic theory, Brownian motion, and algorithmic learning, this algorithmic single-orbit approach has led to improved understanding of known results [10], [3], [14], [19], [22], [13], [4], [7], [1], [5], [20], [18], [6], [21]. In the context of fractal geometry, this approach has even led to recent solutions of classical open problems whose statements did not involve algorithms or single orbits [11], [12].

The fact that CTMCs have discrete state spaces and operate in continuous time, together with the fact that trajectories may or may not halt, presents challenges not encountered in more conventional developments of algorithmic randomness. Our formulation of randomness is nevertheless general. Because we are interested in the *computational* power of stochastic chemical reaction networks (which are special cases of CTMCs) we embrace situations in which the long-term behavior of a network depends essentially on its initial state. Our development thus does not make assumptions that are frequently used in Markov chain theory to avoid such dependencies.

Our approach is also general in another sense, one involving Kolmogorov's program, mentioned above. Once one has succeeded in replacing an "almost every" probabilistic law with an "every random" law, a natural next question is, "*How much* randomness is sufficient for the latter?" Saying that an individual object is random is saying that it "appears random" to a class of computations. Roughly speaking, an object is algorithmically random (or Martin-Löf random) if it appears random to all computably enumerable sets. But weaker notions of randomness such as computable randomness, polynomial-space randomness, polynomial-time randomness, and finite-state randomness, have also been extensively investigated. Three examples of answers to the "how much randomness suffices" question in the context of infinite binary sequences are that (i) every algorithmically random sequence satisfies Birkhoff's ergodic theorem [22]; (ii) every polynomial-time random sequence satisfies the Khinchin-Kolmogorov law of the iterated logarithm [23]; and (iii) every finite-state random sequence satisfies the strong law of large numbers [17].

Although we are primarily concerned with algorithmic randomness in the present paper, we want our randomness notion to be general enough to extend easily to other computational''levels" of randomness, so that "how much randomness" questions can be formulated and hopefully answered. For this reason, we define algorithmic randomness in CTMCs using the martingale (betting strategy) approach of Schnorr [15]. This approach extends to other levels of randomness in a straightforward manner, while our present state (i.e., lack) of knowledge in computational complexity theory does not allow us to extend other approaches (e.g., Martin-Löf tests or Kolmogorov complexity, which are known to be equivalent to the martingale approach at the algorithmic level [10], [3],

*This research was supported in part by National Science Foundation grants 1247051, 1545028, and 1900716.

[1]Department of Computer Science, Iowa State University, Ames, Iowa 50011, USA.

Email addresses:{huangx,lutz,amigunov}@iastate.edu

[14], [19]) to time-bounded complexity classes.

We develop our algorithmic randomness theory in stages. In section 2 we develop the underlying qualitative structure of *Boolean transition systems*, defined so that (i) state transitions are nontrivial, i.e., not from a state to itself, and (ii) trajectories may or may not terminate.

In section 3 we add probabilities, thereby defining *probabilistic transition systems*. For each probabilistic transition system $\mathscr{Q}$ and each initialization σ of $\mathscr{Q}$ we then define $(\mathscr{Q},\sigma)$-*martingales*, which are strategies for betting on the successive entries in a sequence of states of $(\mathscr{Q},\sigma)$. Following the approach of Schnorr [15], we then define a maximal state sequence q of $(\mathscr{Q},\sigma)$ to be *random* if there is no lower semicomputable $(\mathscr{Q},\sigma)$-martingale that *succeeds* on q, i.e., makes unbounded money betting along q. This notion of randomness closely resembles the well-understood theory of random sequences over a finite alphabet [10], [3], [14], [19], except that here the state set may be countably infinite; transitions from a state to itself are forbidden; and a state sequence may terminate, in which case it is random.

Section 4 is where we confront the main challenge of algorithmic randomness in CTMCs, the fact that they operate in continuous, rather than discrete, time. There we develop the algorithmic randomness of sequences $\mathbf{t} = (t_0, t_1, ...)$ of sojourn times t_i relative to corresponding sequences $\boldsymbol{\lambda} = (\lambda_0, \lambda_1, ...)$ of nonnegative real-valued rates λ_i. Each λ_i in such a sequence is regarded as defining an exponential probability distribution function F_{λ_i}, and the sojourn times t_i are to be independently random relative to these. We use a careful binary encoding of sojourn times to define $\boldsymbol{\lambda}$-*martingales* that bet along sequences of sojourn times, and we again follow the Schnorr approach, defining a sequence t of sojourn times to be $\boldsymbol{\lambda}$-*random* if there is no lower semicomputable $\boldsymbol{\lambda}$-martingale that succeeds in it.

In section 5 we put the developments of sections 3 and 4 together. A *trajectory* of a continuous-time Markov chain C is a sequence $\boldsymbol{\tau}$ of ordered pairs (q_n, t_n), where q_n is a state of C and t_n is the sojourn time that C spends in state q_n before jumping to state q_{n+1}. For each continuous-time Markov chain C, we define the notion of a C-*martingale*. Following Schnorr once again, we define a trajectory $\boldsymbol{\tau}$ of C to be *random* if no lower semicomputable martingale succeeds on it. We also give a Kolmogorov complexity characterization of the randomness of trajectories of continuous-time Markov chains.

II. Boolean transition systems

Before developing algorithmic randomness for sequences of states with respect to computable, probabilistic transition systems, we develop the underlying qualitative (not probabilistic) structure by considering transition systems that are Boolean. Some care must be taken to accommodate the fact that, in cases of interest, a sequence of states may either be infinite or end in a terminal state.

Formally, we define a *Boolean transition system* to be an ordered pair $\mathscr{Q} = (Q, \delta)$ where Q is a nonempty, countable set of *states*, and $\delta : Q \times Q \to \{0, 1\}$ is a *Boolean state transition matrix* satisfying $\delta(q, q) = 0$ for all $q \in Q$.

Intuitively, a Boolean transition system $\mathscr{Q} = (Q, \delta)$ is a nondeterministic structure that may be initialized to any nonempty set of states in Q. For $q, r \in Q$, the entry $\delta(q, r)$ in the Boolean transition matrix δ is the Boolean value (0 = false; 1 = true) of the condition that r is reachable from q in one "step" of $\mathscr{Q}$. The irreflexivity requirement that every $\delta(q, q) = 0$ (i.e., that δ have a zero diagonal) reflects the fact that, in all cases of interest in this paper, transitions are nontrivial changes of state. We formalize this intuition, because the formalism will be useful here.

We write $Q^{<\omega}$ for the set of all finite sequences of states in Q, Q^{ω} for the set of all infinite sequences of states in Q, and $Q^{\leq\omega} = Q^{<\omega} \cup Q^{\omega}$. The *length* of a sequence $\boldsymbol{q} \in Q^{\leq\omega}$ is

$$|\boldsymbol{q}| = \left\{ \begin{array}{ll} l & \text{if } \boldsymbol{q} = (q_0, q_1, ..., q_{l-1}) \in Q^{<\omega} \\ \omega & \text{if } \boldsymbol{q} \in Q^{\omega} \end{array} \right\}.$$

A sequence $\boldsymbol{q} \in Q^{\leq\omega}$ can thus be written as $\boldsymbol{q} = (q_i | i < |\boldsymbol{q}|)$ in any case. We write () for the *empty sequence* (sequence of length 0).

For $\boldsymbol{q}, \boldsymbol{r} = (r_i | i < |\boldsymbol{r}|) \in Q^{\leq\omega}$, we say that $\boldsymbol{q}$ is a *prefix* of $\boldsymbol{r}$, and we write $\boldsymbol{q} \sqsubseteq \boldsymbol{r}$, if $|\boldsymbol{q}| \leq |\boldsymbol{r}|$ and $\boldsymbol{q} = (r_i | i < |\boldsymbol{q}|)$. It is easy to see that $\sqsubseteq$ is a partial ordering of $Q^{\leq\omega}$.

An *initialization* of a Boolean transition system $\mathscr{Q} = (Q, \delta)$ is a Boolean-valued function $\sigma : Q \to \{0, 1\}$ whose *support* $supp(\sigma) = \{q \in Q | \sigma(q) \neq 0\}$ is nonempty.

A Boolean transition system $\mathscr{Q} = (Q, \delta)$ *admits* a sequence $\boldsymbol{q} = (q_i | i < |\boldsymbol{q}|) \in Q^{\leq\omega}$ with an initialization σ, and we say that $\boldsymbol{q}$ is $\mathscr{Q}$-*admissible* from σ, if the following conditions hold for all $0 \leq i < |\boldsymbol{q}|$.

(i) If $i = 0$, then $\sigma(q_i) = 1$.
(ii) If $i + 1 < |\boldsymbol{q}|$, then $\delta(q_i, q_{i+1}) = 1$.

A sequence $\boldsymbol{q} \in Q^{\leq\omega}$ that is $\mathscr{Q}$-admissible from σ is *maximal* if, for every sequence $\boldsymbol{r} \in Q^{\leq\omega}$ that is $\mathscr{Q}$-admissible from σ, $\boldsymbol{q} \sqsubseteq \boldsymbol{r} \implies \boldsymbol{q} = \boldsymbol{r}$.

We use the following notations.

$\mathbb{A}[\mathscr{Q}](\sigma) = \{\boldsymbol{q} \in Q^{\leq\omega} | \boldsymbol{q}$ is a maximal $\mathscr{Q}$-admissible sequence from $\sigma\}$.

When $\mathscr{Q}$ is obvious from the context, we omit it from the notation and write these sets as $Adm(\sigma)$ and $\mathbb{A}(\sigma)$. Note that elements of $Adm_{\mathscr{Q}}(\sigma)$ are required to be *finite* sequences.

Intuitively, $\mathbb{A}[\mathscr{Q}](\sigma)$ is the set of all possible "behaviors" of the Boolean transition system $\mathscr{Q} = (Q, \delta)$ with the state initialization $\sigma : Q \to \{0, 1\}$. The fact that δ is irreflexive implies that $q_i \neq q_{i+1}$ holds for all $i \in \mathbb{N}$ such that $i + 1 < |\boldsymbol{q}|$ in every admissible sequence $\boldsymbol{q} = (q_i | i < |\boldsymbol{q}|) \in \mathbb{A}[\mathscr{Q}](\sigma)$. In this paper we do *not* regard the indices $i = 0, 1, ...$ in a state sequence $\boldsymbol{q} = (q_0, q_1, ...)$ as successive instants in discrete time. In our main applications, the amount of time spent in state q_i varies randomly and

continuously, so it is more useful to think of the indices $i = 0, 1, ...$ as finite ordinal numbers, i.e., to think of q_i as merely the i^{th} state in the sequence $\boldsymbol{q}$.

Each $\boldsymbol{x} \in Adm_{\mathscr{Q}}(\sigma)$ is the *name* of the $\mathscr{Q}$-*cylinder*

$$\mathbb{A}_{\boldsymbol{x}}(\sigma) = \{\boldsymbol{q} \in \mathbb{A}[\mathscr{Q}](\sigma) | \boldsymbol{x} \sqsubseteq \boldsymbol{q}\}.$$

Each $\boldsymbol{x} \in Adm(\sigma)$ is a finite - and typically partial - specification of each sequence $\boldsymbol{q} \in \mathbb{A}_{\boldsymbol{x}}(\sigma)$. The collection

$$\mathscr{A}(\sigma) = \mathscr{A}[\mathscr{Q}](\sigma) = \{\mathbb{A}_{\boldsymbol{x}}(\sigma) | \boldsymbol{x} \in Adm_{\mathscr{Q}}(\sigma)\}$$

is a basis for a topology on $\mathbb{A}(\sigma)$. The open sets in this topology are simply the sets that are unions of (finitely or infinitely many) cylinders in $\mathscr{A}(\sigma)$. The metric (in fact, ultrametric) d on $Q^{\leq\omega}$ defined by

$$d(\boldsymbol{q}, \boldsymbol{r}) = 2^{-|\boldsymbol{p}|},$$

where $\boldsymbol{p}$ is the longest common prefix of $\boldsymbol{q}$ and $\boldsymbol{r}$ (and $2^{-\infty} = 0$), induces this same topology on $\mathbb{A}[\mathscr{Q}](\sigma)$ for each Boolean transition system $\mathscr{Q} = (Q, \delta)$ and each state initialization $\sigma : Q \to [0, 1]$. With this topology, $\mathbb{A}[\mathscr{Q}](\sigma)$ is a Polish space (a complete, separable metric space). The isolated points in $\mathbb{A}[\mathscr{Q}](\sigma)$ are (when they exist) the sequences in $\mathbb{A}[\mathscr{Q}](\sigma)$ that are finite, i.e., the sequences $x \in Q^{<\omega} \cap \mathbb{A}[\mathscr{Q}](\sigma)$. Such sequences $\boldsymbol{x}$ are said to *halt*, or *terminate*, in $\mathscr{Q}$ from σ.

A Boolean transition system $\mathscr{Q} = (Q, \delta)$ is *computable* if the elements of Q are naturally represented in such a way that (i) the Boolean-valued function δ is computable, and (ii) the set of *terminal states* (i.e., states $q \in Q$ such that $\delta(q, r) = 0$ for all $r \in Q$) is decidable. An initialization $\sigma : Q \to \{0, 1\}$ is *computable* if its support is decidable.

Boolean transition systems raise significant and deep problems in distributed computing [9], [2], but our focus here is on randomness, which we begin in the following section.

III. Random state sequences

This section develops the elements of algorithmic randomness for sequences of states with respect to computable, probabilistic transition rules.

Formally, we define a *probabilistic transition system* to be an ordered pair $\mathscr{Q} = (Q, \pi)$, where Q is a countable set of *states*, and $\pi : Q \times Q \to [0, 1]$ is a *probabilistic transition matrix*, by which we mean that π satisfies the following two conditions for each state $q \in Q$.

(1) $\pi(q, q) = 0$.
(2) The sum $\pi(q) = \sum_{r \in Q} \pi(q, r)$ is either 0 or 1.

If the sum $\pi(q)$ in condition 2 is 0, then q is a *terminal state*. If $\pi(q)$ is 1, then q is a *nonterminal state*.

If $\mathscr{Q} = (Q, \pi)$ is a probabilistic transition system, and we define $\delta : Q \times Q \to \{0, 1\}$ by

$$\delta(q, r) = sgn(\pi(q, r))$$

for all $q, r \in Q$, where $sgn : [0, \infty) \to \{0, 1\}$ is the signum function

$$sgn(x) = \begin{cases} 0 & \text{if } x = 0 \\ 1 & \text{if } x > 0 \end{cases},$$

then $\mathscr{Q}_B = (Q, \delta)$ is the Boolean transition system *corresponding* to Q. The essential difference between $\mathscr{Q}_B$ and $\mathscr{Q}$ is that, while $\delta(q, r)$ merely says whether it is *possible* for $\mathscr{Q}_B$ (or $\mathscr{Q}$) to transition from q to r in one step, $\pi(q, r)$ is the *quantitative probability* of doing so.

An *initialization* of a probabilistic transition system $\mathscr{Q} = (Q, \pi)$ is a discrete probability measure σ on Q, i.e., a function $\sigma : Q \to [0, 1]$ satisfying $\sum_{q \in Q} \sigma(q) = 1$. The *Boolean version* of such an initialization σ is the function $\sigma_B : Q \to \{0, 1\}$ defined by

$$\sigma_B(q) = sgn(\sigma(q))$$

for each $q \in Q$. It is clear that σ_B is an initialization of $\mathscr{Q}_B$.

Given a probabilistic transition system $\mathscr{Q} = (Q, \pi)$ and an initialization σ of $\mathscr{Q}$, we define the sets

$$Adm(\sigma) = Adm_{\mathscr{Q}}(\sigma) = Adm_{\mathscr{Q}_B}(\sigma_B),$$
$$\mathbb{A}(\sigma) = \mathbb{A}[\mathscr{Q}](\sigma) = \mathbb{A}_{\mathscr{Q}_B}(\sigma_B),$$

relying on the fact that the right-hand sets were defined in section 2. The notations and terminology in section 2 leading up to these definitions are similarly extended to probabilistic transition systems, as are the definitions of the $\mathscr{Q}$-cylinders $\mathbb{A}_{\boldsymbol{x}}(\sigma)$ and the basis $\mathscr{A}(\sigma)$ for the topology $\mathbb{A}(\sigma)$.

What we can do here that we could not do for Boolean transition systems is define a Borel probability measure on each set $\mathbb{A}[\mathscr{Q}](\sigma)$. Specifically, for each probabilistic transition system $\mathscr{Q} = (Q, \pi)$ and each initialization σ of $\mathscr{Q}$, define the function

$$\mu_{\mathscr{Q},\sigma} : Adm_{\mathscr{Q}}(\sigma) \to [0, 1]$$

as follows. Let $\boldsymbol{x} = (x_i | i < |\boldsymbol{x}|) \in Adm_{\mathscr{Q}}(\sigma)$. If $|\boldsymbol{x}| = 0$, then $\mu_{\mathscr{Q},\sigma}(x) = 1$. If $|\boldsymbol{x}| > 0$, then

$$\mu_{\mathscr{Q},\sigma}(\boldsymbol{x}) = \sigma(x_0) \prod_{i=0}^{|\boldsymbol{x}|-2} \pi(x_i, x_{i+1}). \tag{3.1}$$

Since $\boldsymbol{x}$ is a name of the cylinder $\mathbb{A}_{\boldsymbol{x}}[\mathscr{Q}](\sigma)$, each $\mu_{\mathscr{Q},\sigma}(\boldsymbol{x})$ here should be understood as an abbreviation of $\mu_{\mathscr{Q},\sigma}(\mathbb{A}_{\boldsymbol{x}}(\sigma))$, which is intuitively the probability that an element of $\mathbb{A}_{\boldsymbol{x}}[\mathscr{Q}](\sigma)$ begins with the finite sequence $\boldsymbol{x}$.

Observation 1: If a sequence $\boldsymbol{x} \in Adm_{\mathscr{Q}}(\sigma)$ does not terminate, then

$$\mu_{\mathscr{Q},\sigma}(\boldsymbol{x}) = \sum_{\boldsymbol{x} \sqsubseteq \boldsymbol{y} \in Adm_{\mathscr{Q}}(\sigma), |\boldsymbol{y}| = |\boldsymbol{x}|+1} \mu_{\mathscr{Q},\sigma}(\boldsymbol{y}) \tag{3.2}$$

The above observation implies that $\mu_{\mathscr{Q},\sigma}$ can, by standard techniques, be extended to a Borel probability measure on $\mathbb{A}[\mathscr{Q}](\sigma)$, i.e., to a function $\mu_{\mathscr{Q},\sigma}$ that assigns probability $\mu_{\mathscr{Q},\sigma}(E)$ to every Borel set $E \subseteq \mathbb{A}[\mathscr{Q}](\sigma)$.

Definition 1: If $\mathscr{Q}$ is a probabilistic transition system and σ is an initialization of $\mathscr{Q}$, then a $(\mathscr{Q}, \sigma)$-*martingale* is a function

$$d : Adm_{\mathscr{Q}}(\sigma) \to [0, \infty)$$

such that, for every non-terminating sequence $\boldsymbol{x} \in Adm_{\mathscr{Q}}(\sigma)$,

$$d(\boldsymbol{x})\mu(\boldsymbol{x}) = \sum_{\boldsymbol{x} \sqsubseteq \boldsymbol{y} \in Adm_{\mathscr{Q}}(\sigma), |\boldsymbol{y}| = |\boldsymbol{x}|+1} d(\boldsymbol{y})\mu_{\mathscr{Q},\sigma}(\boldsymbol{y}) \quad (3.3)$$

where $\mu = \mu_{\mathscr{Q},\sigma}$.

Intuitively, a $(\mathscr{Q}, \sigma)$-martingale d is a gambler that bets on the successive states in a sequence $\boldsymbol{q} = (q_i | i < |\boldsymbol{q}|) \in \mathbb{A}[\mathscr{Q}](\sigma)$. The gambler's initial capital is $d(())$, and its capital after betting on a prefix $x \in Adm_{\mathscr{Q}}(\sigma)$ of $\boldsymbol{q}$ is $d(\boldsymbol{x})$. The condition (3.3) says that the payoffs are fair with respect to the probability measure $\mu = \mu_{\mathscr{Q},\sigma}$ in the sense that the conditional expectation of the gambler's capital after betting on the state following $\boldsymbol{x}$ in $\boldsymbol{q}$ given that $\boldsymbol{x} \sqsubseteq \boldsymbol{q}$, is exactly the gambler's capital before placing this bet.

Definition 2: A $(\mathscr{Q}, \sigma)$-martingale d *succeeds* on a sequence $\boldsymbol{q} \in \mathbb{A}[\mathscr{Q}](\sigma)$ if the set

$$\{d(\boldsymbol{x}) | \boldsymbol{x} \in Adm_{\mathscr{Q}}(\sigma) \text{ and } \boldsymbol{x} \sqsubseteq \boldsymbol{q}\}$$

is unbounded.

The *success set* of a $(\mathscr{Q}, \sigma)$-martingale d is $S^{\infty}[d] = \{\boldsymbol{q} \in \mathbb{A}[\mathscr{Q}](\sigma) | d \text{ succeeds on } \boldsymbol{q}\}$.

Following standard practice, we develop randomness by imposing computability conditions on martingales. Recall that, if D is a discrete domain, then a function $f : D \to \mathbb{R}$ is *computable* if there is a computable function $\widehat{f} : D \times \mathbb{N} \to \mathbb{Q}$ such that, for all $x \in D$ and $r \in \mathbb{N}$,

$$|\widehat{f}(x, r) - f(x)| \leq 2^{-r}.$$

The parameter r here is called a *precision parameter*.

A function $f : D \to \mathbb{R}$ is *lower semi-computable* if there is a computable function $\widehat{f} : D \times \mathbb{N} \to \mathbb{Q}$ such that the following two conditions hold for all $x \in D$.

(i) For all $s \in \mathbb{N}, \widehat{f}(x, s) \leq \widehat{f}(x, s+1) < f(x)$.
(ii) $\lim_{s \to \infty} \widehat{f}(x, s) = f(x)$.

The parameter s is sometimes called a *patience parameter*, because the convergence in (ii) can be very slow. A probabilistic transition system $\mathscr{Q} = (Q, \pi)$ is *computable* if the elements of Q are naturally represented in such a way that (i) the probability transition matrix $\pi : Q \times Q \to [0, 1]$ is computable in the above sense, and (ii) the support of π and the set of terminal states are decidable. (It is well known ([8], [24]) that (ii) does not follow from (i). Fortunately, (ii) does hold in many cases of interest, including chemical reaction networks).

Similarly, an initialization σ of a probabilistic transition system $\mathscr{Q} = (Q, \pi)$ is *computable* if (i) the function $\sigma : Q \to [0, 1]$ is computable, and (ii) the *support* of σ is decidable.

Let $\mathscr{Q}$ be a probabilistic transition system that is computable, and let σ be an initialization of $\mathscr{Q}$ that is also computable. A state sequence $\boldsymbol{q} \in \mathbb{A}[\mathscr{Q}](\sigma)$ is (*algorithmically*) *random* if there is no lower semi-computable $(\mathscr{Q}, \sigma)$-martingale that succeeds on $\boldsymbol{q}$.

This notion of random sequences in $\mathbb{A}[\mathscr{Q}](\sigma)$ closely resembles the well-understood theory of random sequences on a finite alphabet [26], [16]. The main differences are that here the state set may be countably infinite; transitions from a state to itself are forbidden; and a state sequence may terminate, in which case it is clearly random.

IV. Random sequences of sojourn times

The "sojourn time" that a continuous-time Markov chain spends in a state before jumping to a new state may be any element of $(0, \infty]$, i.e., any duration t that is either a (strictly) positive real number or ∞. This section thus develops the elements of algorithmic randomness for sequences of durations $t \in (0, \infty]$ with respect to sequences of probability measures that occur in continuous-time Markov chains.

A *rate* in this paper is a non-negative real number $\lambda \in [0, \infty)$. We rely on context to distinguish this standard use of λ from the equally standard use of λ to denote the empty string.

We interpret each rate $\lambda > 0$ as a name of the exponential probability measure with rate λ, i.e., the probability measure on $(0, \infty]$ whose cumulative distribution function $F_{\lambda} : (0, \infty] \to [0, 1]$ is given by

$$F_{\lambda}(t) = 1 - e^{-\lambda t}$$

for all $t \in (0, \infty]$, where $e^{-\infty} = 0$. We interpret the rate $\lambda = 0$ as a name of the point-mass probability on $(0, \infty]$ that concentrates all the probability at ∞. This has the cumulative distribution function $F_0 : (0, \infty] \to [0, 1]$ given by

$$F_0(t) = \begin{cases} 0 & \text{if } t \in (0, \infty) \\ 1 & \text{if } t = \infty \end{cases}$$

We associate each string $w \in \{0, 1\}^*$ with the interval $I_w \subseteq [0, 1]$ defined as follows. Let w be the lexicographically i^{th} $(0 \leq i < 2^{|w|})$ element of $\{0, 1\}^{|w|}$ where $0^{|w|}$ is the 0^{th} element and $1^{|w|}$ is the $(2^{|w|} - 1)^{\text{st}}$ element. Then

$$I_w = (2^{-|w|}i, 2^{-|w|}(i+1)].$$

Note that, for each $w \in \{0, 1\}^*$ and $l \in \mathbb{N}$, the intervals I_{wu}, for $u \in \{0, 1\}^l$, form a *left-to-right partition* of I_w, i.e., a partition of I_w in which I_{wu} lies to the left of I_{wv} if and only if u lexicographically precedes v.

For each rate $\lambda \in [0, \infty)$ and each string $w \in \{0, 1\}^*$, define the interval

$$D_{\lambda}(w) = F_{\lambda}^{-1}(I_w) \subseteq (0, \infty].$$

Example 2: If $\lambda > 0$, then

$$D_{\lambda}(00) = (0, a_1], \quad D_{\lambda}(01) = (a_1, a_2],$$
$$D_{\lambda}(10) = (a_2, a_3], \quad D_{\lambda}(11) = (a_3, \infty],$$

where $a_1 = \frac{2ln2 - ln3}{\lambda}, a_2 = \frac{ln2}{\lambda}$, and $a_3 = \frac{2ln2}{\lambda}$. On the other hand, $D_0(00) = (0, \infty)$, $D_0(01) = D_0(10) = \emptyset$, and $D_0(11) = \{\infty\}$.

Observation 3: If $\lambda > 0$, then, for each $l \in \mathbb{N}$, the intervals $D_\lambda(w)$, for $w \in \{0,1\}^l$, form a left-to-right partition of $(0,\infty]$ into intervals that are equiprobable with respect to F_λ.

Example 2 shows that the assumption $\lambda > 0$ is essential here.

For each rate $\lambda \in [0,\infty)$, each duration $t \in (0,\infty]$, and each $w \in \{0,1\}^*$, we call w a *λ-approximation* (or a *partial λ-specification*) of t, and we write $w \sqsubseteq_\lambda t$, if $t \in D_\lambda(w)$.

A *rate sequence* is a nonempty sequence $\boldsymbol{\lambda} = (\lambda_i \mid 0 \le i < |\boldsymbol{\lambda}|) \in [0,\infty)^{\le\omega}$ with the property that, for each $0 \le i < |\boldsymbol{\lambda}|$,

$$i+1 < |\boldsymbol{\lambda}| \iff \lambda_i > 0.$$

(That is, either $\boldsymbol{\lambda}$ is finite with a single 0 entry, occurring at the end, or $\boldsymbol{\lambda}$ is infinite with no 0 entries.)

If $\boldsymbol{\lambda} = (\lambda_i \mid 0 \le i < |\boldsymbol{\lambda}|)$ is a rate sequence, then a *$\boldsymbol{\lambda}$-duration sequence* is a sequence

$$\boldsymbol{t} = (t_i | i < |\boldsymbol{\lambda}|) \in (0,\infty]^{\le\omega}$$

such that, for each $0 \le i < |\boldsymbol{\lambda}|$,

$$t_i < \infty \iff \lambda_i > 0.$$

We write $\boldsymbol{D_\lambda}$ for the set of all $\boldsymbol{\lambda}$-duration sequences. Note that

$$\boldsymbol{D_\lambda} = \begin{cases} (0,\infty)^{|\boldsymbol{\lambda}|-1} \times \{\infty\} & \text{if } |\boldsymbol{\lambda}| < \omega \\ (0,\infty)^\omega & \text{if } |\boldsymbol{\lambda}| = \omega \end{cases}$$

depends only on the length of $\boldsymbol{\lambda}$, not on the components of $\boldsymbol{\lambda}$.

If $\boldsymbol{\lambda} = (\lambda_i \mid 0 \le i < |\boldsymbol{\lambda}|)$ is a rate sequence, $\boldsymbol{t} = (t_i | i < |\boldsymbol{\lambda}|) \in \boldsymbol{D_\lambda}$ is a $\boldsymbol{\lambda}$-duration sequence, and $\boldsymbol{w} = (w_i | i < |\boldsymbol{w}|) \in (\{0,1\}^*)^{<\omega}$ is a finite sequence of binary strings with $|\boldsymbol{w}| \le |\boldsymbol{\lambda}|$, then we call $\boldsymbol{w}$ a *$\boldsymbol{\lambda}$-approximation* (or a *partial $\boldsymbol{\lambda}$-specification*) of $\boldsymbol{t}$, and we write $\boldsymbol{w} \sqsubseteq_{\boldsymbol{\lambda}} \boldsymbol{t}$, if $w_i \sqsubseteq_{\lambda_i} t_i$ holds for all $0 \le i < |\boldsymbol{w}|$.

If $\boldsymbol{\lambda}$ is a rate sequence and $\boldsymbol{w} \in (\{0,1\}^*)^{<\omega}$ is a finite sequence of binary strings with $|\boldsymbol{w}| \le |\boldsymbol{\lambda}|$, then the *$\boldsymbol{\lambda}$-cylinder generated by $\boldsymbol{w}$* is the set

$$\boldsymbol{D_\lambda}(\boldsymbol{w}) = \{\boldsymbol{t} \in \boldsymbol{D_\lambda} \mid \boldsymbol{w} \sqsubseteq_{\boldsymbol{\lambda}} \boldsymbol{t}\}$$

of $\boldsymbol{\lambda}$-duration sequences.

It is routine to verify that, for each rate sequence $\boldsymbol{\lambda}$, the collection

$$\mathscr{D}_{\boldsymbol{\lambda}} = \{D_{\boldsymbol{\lambda}}(\boldsymbol{w}) \mid \boldsymbol{w} \in (\{0,1\}^*)^{<\omega} \text{ and } |\boldsymbol{w}| \le |\boldsymbol{\lambda}|\}$$

is a semi-algebra of subsets of dl that generates the σ-algebra $\mathscr{B}_{\boldsymbol{\lambda}}$ of all Borel subsets of $\mathscr{D}_{\boldsymbol{\lambda}}$. If we define

$$\mu_{\boldsymbol{\lambda}} : \mathscr{D}_{\boldsymbol{\lambda}} \to [0,1]$$

by

$$\mu_{\boldsymbol{\lambda}}(\mathscr{D}_{\boldsymbol{\lambda}}(\boldsymbol{w})) = 2^{-\sum_{i=0}^{|\boldsymbol{w}|-1} |w_i|}$$

for all $\boldsymbol{w} = (w_i \mid i < |\boldsymbol{w}|) \in (\{0,1\}^*)^{<\omega}$ with $|\boldsymbol{w}| \le |\boldsymbol{\lambda}|$, then it follows by standard techniques that $\mu_{\boldsymbol{\lambda}}$ extends uniquely to a probability measure

$$\mu_{\boldsymbol{\lambda}} : \mathscr{B}_{\boldsymbol{\lambda}} \to [0,1].$$

Note that $\mathscr{B}_{\boldsymbol{\lambda}}$ only depends on the length of $\boldsymbol{\lambda}$, but $\mu_{\boldsymbol{\lambda}}$ also depends on the components of $\boldsymbol{\lambda}$. When convenient, we use the abbreviation

$$\mu_{\boldsymbol{\lambda}}(\boldsymbol{w}) = \mu_{\boldsymbol{\lambda}}(\boldsymbol{D_\lambda}(\boldsymbol{w})).$$

If $\boldsymbol{\lambda} = (\lambda_i \mid 0 \le i < |\boldsymbol{\lambda}|)$ is a rate sequence, then a *$\boldsymbol{\lambda}$-martingale* is a function

$$d : (\{0,1\}^*)^{<|\boldsymbol{\lambda}|} \to [0,\infty)$$

that satisfies the following two conditions for all $\boldsymbol{w} = (w_0, ..., w_{n-1}) \in (\{0,1\}^*)^{<|\boldsymbol{\lambda}|}$.

1) $d(\boldsymbol{w}) = \frac{d(w_0,...,w_{n-1}0) + d(w_0,...,w_{n-1}1)}{2}$.

2) If $n+1 < |\boldsymbol{\lambda}|$, then
$d(w_0, ..., w_{n-1}, \lambda) = d(w_0, ..., w_{n-1})$.

(Note that the λ entry on the left-hand side is the empty string.)

Intuitively, a $\boldsymbol{\lambda}$-martingale d is a strategy that a gambler may use for betting on approximations w_i of the durations t_i in a $\boldsymbol{\lambda}$-duration sequence $\boldsymbol{t} = (t_i \mid i < |\boldsymbol{t}|)$. The gambler's initial amount of money is the value $d(())$ of d at the empty sequence () of binary strings. If $\boldsymbol{w} = (w_0, ..., w_{n-1}) \sqsubseteq_{\boldsymbol{\lambda}} \boldsymbol{t}$, then $d(\boldsymbol{w})$ is the amount of money that the gambler has after betting on $\boldsymbol{w}$. This condition $\boldsymbol{w} \sqsubseteq_{\boldsymbol{\lambda}} \boldsymbol{t}$ means that each t_i is in the interval $\boldsymbol{D_\lambda}(w_i) \subseteq (0,\infty]$. If the gambler then chooses to bet on which of the subintervals $D_{\lambda_{n-1}}(w_{n-1}0)$ and $D_{\lambda_{n-1}}(w_{n-1}1)$ of $D_{\lambda_{n-1}}(w_{n-1})$ t_{n-1} lies in, condition 1 above says that the payoffs of these bets are fair with respect to the exponential probability measure with rate λ_{n-1}. (Note that $D_{\lambda_{n-1}}(w_{n-1}0)$ and $D_{\lambda_{n-1}}(w_{n-1}1)$ partition $D_{\lambda_{n-1}}(w_{n-1})$ into equiprobable subintervals, but these subintervals may have very different lengths.) Condition 2 above says that the extension from $(w_0, ..., w_{n-1})$ to $(w_0, ..., w_{n-1}, \lambda)$, does not involve a bet. The martingale has values $d(\boldsymbol{w})$ for all $\boldsymbol{w} \in (\{0,1\}^*)^{<|\boldsymbol{\lambda}|}$, but our intuitive gambler may place bets in many different orders. For example, the gambler may place a finite number of bets on approximations of t_1, then a finite number of bets on approximations of t_2, etc., but this ordering of bets is an intuitive fancy, not part of the definition of the $\boldsymbol{\lambda}$-martingale d.

A $\boldsymbol{\lambda}$-martingale d *succeeds* on a $\boldsymbol{\lambda}$-duration sequence $\boldsymbol{t}$ if the set

$$\{d(w) \mid \boldsymbol{w} \sqsubseteq_{\boldsymbol{\lambda}} \boldsymbol{t}\}$$

is unbounded. The *success set* of a $\boldsymbol{\lambda}$-martingale d is

$$S^\infty[d] = \{\boldsymbol{t} \in \boldsymbol{D_\lambda} \mid d \text{ succeeds on } \boldsymbol{t}\}.$$

V. Random CTMC trajectories

We now develop the theory of randomness for sequences of state-time pairs, representing trajectories of continuous-time Markov chains.

A. Continuous-time Markov chains

A CTMC is an ordered triple,

$$C = (Q, \boldsymbol{\lambda}, \sigma)$$

where Q is a countable set of states, $\boldsymbol{\lambda} : Q \times Q \to [0, \infty)$ is the *rate matrix* satisfying $\lambda(q, q) = 0$ for every $q \in Q$, and σ is the *state initialization* as described in section 3. Let $C = (Q, \boldsymbol{\lambda}, \sigma)$ be a CTMC. At each time $t \in [0, \infty)$ C is probabilistically in some state. At time $t = 0$, this state is chosen according to σ. For each state $q \in Q$, the real number

$$\lambda_q = \sum_{r \in Q} \lambda(q, r)$$

is the *rate out of* state q. If $\lambda_q = 0$, then q is a *terminal* state, meaning that, if C ever enters state q, then C remains in state q forever. If a state q is *nonterminal*, i.e., $\lambda_q > 0$ and C enters q at some time t, then the *sojourn time* for which C remains in state q before moving to a new state is a random variable that has the exponential distribution with rate λ_q. Hence the expected sojourn time of C in state q is $\frac{1}{\lambda_q}$. When C does move to a new state, it moves to state $r \in Q$ with probability

$$p(q, r) = \frac{\lambda(q, r)}{\lambda_q}.$$

Note that the CTMC model uses "continuous time" (times ranging over $(0, \infty]$) but "discrete state space". Accordingly, its state transitions, called *jump transitions*, are instantaneous. Mathematically, if C jumps from state q to state r at time t, we say that q is in the "new" state r at time t, having been in the "old" state q throughout some time interval $[s, t)$ where $s < t$.

A *trajectory* of a CTMC $C = (Q, \boldsymbol{\lambda}, \sigma)$ is a sequence $\boldsymbol{\tau}$ of the form

$$\boldsymbol{\tau} = ((q_n, t_n) \mid n \in \mathbb{N}) \in (Q \times (0, \infty))^\infty.$$

Intuitively, such a trajectory $\boldsymbol{\tau}$ denotes the turn of events in which $q_0, q_1, ...$ are the successive states of C and $t_0, t_1, ...$ are the successive sojourn times of C in these states. Accordingly, we write

$$state_{\boldsymbol{\tau}}(n) = q_n,\ soj_{\boldsymbol{\tau}}(n) = t_n$$

for each $n \in \mathbb{N}$. When convenient we write $\boldsymbol{\tau}$ as an ordered pair

$$\boldsymbol{\tau} = (\boldsymbol{q}, \boldsymbol{t}),$$

where

$$\boldsymbol{q} = (q_n \mid n \in \mathbb{N}), \boldsymbol{t} = (t_n \mid n \in \mathbb{N}).$$

There are two ways in which a trajectory $(\boldsymbol{q}, \boldsymbol{t})$ may fail to represent a "true trajectory" of the CTMC C in the above intuitive sense. First, it may be the case that $p(q_n, q_{n+1}) = 0$ (i.e. $\lambda(q_n, q_{n+1}) = 0$) for some $n \in \mathbb{N}$. This presents no real difficulty, since it merely says that the event "$state_{\boldsymbol{\tau}}(n) = q_n$ and $state_{\boldsymbol{\tau}}(n+1) = q_{n+1}$" has probability 0. The second way in which $(\boldsymbol{q}, \boldsymbol{t})$ may fail to represent a "true trajectory" is for some q_n to be a terminal state of C. We deal with this by defining the *length* of a trajectory $\boldsymbol{\tau} = (\boldsymbol{q}, \boldsymbol{t})$ to be

$$||\boldsymbol{\tau}|| = \min\{n \in \mathbb{N} \mid q_n \text{ is terminal }\},$$

where $min\ \emptyset = \infty$. We then intuitively interpret a trajectory $\boldsymbol{\tau} = (\boldsymbol{q}, \boldsymbol{t})$ with $||\boldsymbol{\tau}|| < \infty$ as the finite sequence

$$\boldsymbol{\tau}' = ((q_n, t'_n) \mid n \leq ||\boldsymbol{\tau}||),$$

where each

$$t'_n = \begin{cases} t_n & \text{if } n < ||\boldsymbol{\tau}|| \\ \infty & \text{if } n = ||\boldsymbol{\tau}|| \end{cases} \tag{V.1}$$

We write

$$\Omega = \Omega[C] = (Q \times (0, \infty))^\infty$$

for the set of all trajectories of a CTMC, C.

Elements of $(Q \times \{0, 1\}^*)^*$ are called *approximations* or *partial specifications* of trajectories. The *cylinder generated by* $w = (q_0, u_0), (q_1, u_1), ..., (q_{n-1}, u_{n-1}) \in (Q \times \{0, 1\}^*)^*$ is the set Ω_w of trajectories defined as follows: If q_i is terminal for some $0 \leq i < n - 1$ then $\Omega_w = \emptyset$. If q_i is nonterminal for all $0 \leq i < n - 1$ and q_{n-1} is terminal, then

$$\Omega_w = \{\boldsymbol{\tau} \in \Omega \mid (\forall 0 \leq i < n) state_{\boldsymbol{\tau}}(i) = q_i \text{ and } (\forall 0 \leq i < n - 1) soj_{\boldsymbol{\tau}}(i) \in D_{\lambda_i}(u_i)\}.$$

If q_i is nonterminal for all $0 \leq i < n$ then

$$\Omega_w = \{\boldsymbol{\tau} \in \Omega \mid (\forall 0 \leq i < n)[state_{\boldsymbol{\tau}}(i) = q_i \text{ and } soj_{\boldsymbol{\tau}}(i) \in D_{\lambda_i}(u_i)]\}.$$

The *probability* $\mu_C(\Omega_w)$, usually written $\mu_C(w)$, of a cylinder Ω_w, is defined as follows: If $n = 0$ (i.e. $w = \lambda$), then $\mu_C(w) = 1$. If q_i is terminal for some $0 \leq i < n - 1$, then $\mu_C(\Omega_w) = 0$. If q_i is nonterminal for all $0 \leq i < n-1$ and q_{n-1} is terminal, then

$$\mu_C(\Omega_w) = \sigma(q_0) \prod_{i=0}^{n-2} [p(q_i, q_{i+1}) 2^{-|u_i|}].$$

If $n > 0$ and q_i is nonterminal for all $0 \leq i < n$, then

$$\mu_C(\Omega_w) = \sigma(q_0) \prod_{i=0}^{n-2} p(q_i, q_{i+1}) \prod_{i=0}^{n-1} 2^{-|u_i|}.$$

A set $X \subseteq \Omega$ has *probability 0*, and we write $\mu_C(X) = 0$, if, for $\epsilon > 0$, there is a set $A \subseteq (Q \times \{0, 1\}^*)^*$ such that

$$X \subseteq \bigcup_{w \in A} \Omega_w$$

and

$$\sum_{w \in A} \mu_C(\Omega_w) \leq \epsilon$$

From now on we assume that the states $q \in Q$ have canonical representations, so that it is clear what it means for function $f : Q \to Q$, etc., to be computable.

A set $X \in \Omega$ has *constructive probability 0* (or is a *constructive null set*), and we write $\mu_{C,\text{constr}}(X) = 0$, if there is a computable function

$$g : \mathbb{N} \times \mathbb{N} \to (Q \times \{0,1\}^*)^*$$

such that, for every $k \in \mathbb{N}$,

$$X \subseteq \bigcup_{l=0}^{\infty} \Omega_{g(k,l)} \text{ and } \sum_{l=0}^{\infty} \mu_C(\Omega_{g(k,l)}) \leq 2^{-k}.$$

A set $X \subseteq \Omega$ has *constructive probability 1*, and we write $\mu_{C,\text{constr}}(X) = 1$, if $\mu_{C,\text{constr}}(\Omega \setminus X) = 0$.

Before we discuss C-martingales and their relation to the above probability space, let us overload the relation $\sqsubseteq$ to also compare partial specifications to partial specifications and to trajectories. If $w \in (Q \times \{0,1\}^{<\omega})^{<\omega}$ and $S \in (Q \times \{0,1\}^{\leq\omega})^{\leq\omega}$, we say $w \sqsubseteq S$ if:

1. $|v| \leq |w|$
2. For all i, $0 \leq i \leq |v| - 1, w[i] \sqsubseteq v[i]$ or $v[i] \sqsubseteq w[i]$
3. For all i, $0 \leq i \leq |v| - 1, state(w[i]) = state(v[i])$

We now introduce the notion of a C-martingale.

B. CTMC martingales

In place of $\mu_{\boldsymbol{\lambda}}, \mu_C$, and $\mu_{\mathscr{Q},\sigma}$ we will simply write μ, μ_{constr}. It should be clear from context which measure is being used.

If $C = (Q, \boldsymbol{\lambda}, \pi)$ is a CTMC, then a *C-martingale* is a function

$$d : (Q \times \{0,1\}^*)^* \to [0, \infty)$$

with the following two properties.

1) For all $w \in (Q \times \{0,1\}^*)^*$,

$$d(w)\mu(w) = \sum_{q \in Q} d(w(q,\lambda))\mu(w(q,\lambda)). \tag{V.2}$$

2) For all $w \in (Q \times \{0,1\}^*)^*$, $q \in Q$, and $u \in \{0,1\}^*$,

$$d(w(q,u))\mu(w(q,u)) = \sum_{b \in \{0,1\}} d(w(q,ub))\mu(w(q,ub)) \tag{V.3}$$

Intuitively, a C-martingale d is a strategy for betting on successive approximations w of a trajectory $\boldsymbol{\tau}$ of C. A gambler using d starts with initial capital $d(\lambda) \in [0, \infty)$. More generally, each value $d(w)$ is the amount of money that the gambler will have after betting on w. At this stage, the C-martingale d tells the gambler how it may proceed in either of the following two ways.

(i) The gambler may "move on" to bet on the value of $state_{\boldsymbol{\tau}}(|w|)$, which is the next state of $\boldsymbol{\tau}$. In this case condition (V.2) ensures that the payoffs for this bet are fair.

(ii) The gambler may "stay" with the current state, which is $state_{\boldsymbol{\tau}}(|w| - 1)$, and bet further on the approximate value of $soj_{\boldsymbol{\tau}}(|w| - 1)$. In this case condition (V.3) ensures that the payoffs for this bet are fair.

A C-martingale d *succeeds* on a trajectory $\boldsymbol{\tau}$ if, for every real number $\alpha > 0$, there exists $w \in (Q \times \{0,1\}^*)^*$ such that $w \sqsubseteq \boldsymbol{\tau}$ and $d(w) > \alpha$.

The *success set* of a C-martingale d is

$$S^\infty[d] = \{\boldsymbol{\tau} \in \Omega[C] \mid \text{d succeeds on } \boldsymbol{\tau}\}.$$

Theorem 4: For every CTMC C and every set $X \subseteq \Omega[C]$, the following two conditions are equivalent.

(1) $\mu(X) = 0$
(2) There is a C-martingale d such that $X \subseteq S^\infty[d]$.

Theorem 5: For every CTMC C and every set $X \subseteq \Omega[C]$, the following two conditions are equivalent.

(1) $\mu_{\text{constr}}(X) = 0$.
(2) There is a lower semi-computable C-martingale d such that $X \subseteq S^\infty[d]$.

Much like the classical setting, we call a trajectory $\boldsymbol{\tau}$ *Martin-Löf random* if $\{\boldsymbol{\tau}\}$ is not of constructive measure 0.

C. Kolmogorov complexity characterization

Random trajectories can also be characterized using *Kolmogorov complexity*. First, we briefly review this notion in the classical setting. We fix a universal *self-delimiting Turing machine* (see [10]), U. The Kolmogorov complexity, K, of a (finite) string x in $\{0,1\}^*$ is the length of a shortest program for a self-delimiting Turing machine which prints x. That is, $K : \{0,1\}^* \to \mathbb{N}$ is defined by

$$K(x) = \min\{|\pi| \mid U(\pi) = x \text{ and } \pi \in \{0,1\}^*\}.$$

When x is not a binary string, but some other finite object, $K(x)$ is defined from the above by routine coding.

Definition 3: The *profile* of a cylinder Ω_w of a CTMC is

$$\text{prof}(w) = (|u_1|, ..., |u_n|),$$

where $w = ((q_1, u_1), ..., (q_n, u_n))$.

Observation 6: For each CTMC C and each profile p,

$$\sum_{\{w : \text{prof}(w) = p\}} \mu_C(w) = 1.$$

The following two lemmas are analogous to standard results used in the Kolmogorov complexity characterization of algorithmically random sequences.

Lemma 7: For every cylinder, Ω_w of a CTMC C,

$$K(w) \leq l(w) + K(\text{prof}(w)) + O(1),$$

where $l(w) = \log \frac{1}{\mu_C(w)}$ is the "self-information" of w.

Lemma 8: There is a constant $c \in \mathbb{N}$ such that, for every profile p of a CTMC C and every $k \in \mathbb{N}$,

$$\mu_C\Bigg(\bigcup_{\substack{w \\ prof(w) = p \\ K(w) < l(w) + K(p) - k}} \Omega_w \Bigg) < 2^{c-k}.$$

Substituting $k + K(\text{prof}(w))$ for k here gives

$$\mu_C\Bigg(\bigcup_{\substack{w \\ prof(w)=p \\ K(w)<l(w)-k}} \Omega_w\Bigg) < 2^{c-k-K(p)}.$$

With these lemmas, we can establish the Kolmogorov complexity characterization of randomness for trajectory objects, which is exactly analogous to a well-known characterization of the algorithmic randomness of sequences over finite alphabets [26], [16].

Theorem 9: A trajectory $\boldsymbol{\tau}$ is Martin Löf random if and only if there exists $k \in \mathbb{N}$, such that for every $w \sqsubseteq \boldsymbol{\tau}$, $K(w) \geq l(w) - k$.

References

[1] Kelty Allen, Laurent Bienvenu, and Theodore Slaman. On zeros of Martin-Löf random Brownian motion. *Journal of Logic and Analysis*, 6, 2015.

[2] Wojciech Czerwinski, Slawomir Lasota, Ranko Lazic, Jérôme Leroux, and Filip Mazowiecki. The reachability problem for Petri nets is not elementary. *arXiv preprint arXiv:1809.07115*, 2018.

[3] Rodney G. Downey and Denis R. Hirschfeldt. *Algorithmic Randomness and Complexity*. Springer Science & Business Media, 2010.

[4] Willem L. Fouché. Fractals generated by algorithmically random Brownian motion. In *Conference on Computability in Europe*, pages 208–217. Springer, 2009.

[5] Willem L. Fouché. Kolmogorov complexity and the geometry of Brownian motion. *Mathematical Structures in Computer Science*, 25(7):1590–1606, 2015.

[6] Mrinalkanti Ghosh and Satyadev Nandakumar. Predictive complexity and generalized entropy rate of stationary ergodic processes. In *International Conference on Algorithmic Learning Theory*, pages 365–379. Springer, 2012.

[7] Bjørn Kjos-Hanssen and Anil Nerode. Effective dimension of points visited by Brownian motion. *Theoretical Computer Science*, 410(4-5):347–354, 2009.

[8] Ker-I Ko. *Complexity Theory of Real Functions*. Birkhäuser, 1991.

[9] Jérôme Leroux and Sylvain Schmitz. Demystifying reachability in vector addition systems. In *LICS 2015*, pages 56–67. IEEE, 2015.

[10] Ming Li and Paul Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer Science & Business Media, 2013.

[11] Neil Lutz. Fractal intersections and products via algorithmic dimension. In *42nd International Symposium on Mathematical Foundations of Computer Science (MFCS 2017)*, pages 58:1–58:12. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.

[12] Neil Lutz and Donald M. Stull. Bounding the dimension of points on a line. In *International Conference on Theory and Applications of Models of Computation*, pages 425–439. Springer, 2017.

[13] Satyadev Nandakumar. An effective ergodic theorem and some applications. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 39–44. ACM, 2008.

[14] André Nies. *Computability and Randomness*, volume 51. OUP Oxford, 2009.

[15] Claus-Peter Schnorr. A unified approach to the definition of random sequences. *Mathematical Systems Theory*, 5(3):246–258, 1971.

[16] Claus-Peter Schnorr. A survey of the theory of random sequences. In *Basic Problems in Methodology and Linguistics*, pages 193–211. Springer, 1977.

[17] Claus-Peter Schnorr and Hermann Stimm. Endliche Automaten und Zufallsfolgen. *Acta Informatica*, 1(4):345–359, 1972.

[18] Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9(Mar):371–421, 2008.

[19] Alexander Shen, Vladimir A. Uspensky, and Nikolay Vereshchagin. *Kolmogorov Complexity and Algorithmic Randomness*, volume 220. American Mathematical Society, 2017.

[20] Vladimir Vovk, Alex Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer Science & Business Media, 2005.

[21] Volodya Vovk, Alexander Gammerman, and Craig Saunders. Machine-learning applications of algorithmic randomness. In *Proceedings of the Sixteenth International Conference on Machine Learning*, pages 444–453. Morgan Kaufmann Publishers Inc., 1999.

[22] Vladimir V. V'yugin. Ergodic theorems for individual random sequences. *Theoretical Computer Science*, 207(2):343–361, 1998.

[23] Yongge Wang. *Randomness and Complexity*. PhD thesis, University of Heidelberg, 1996.

[24] Klaus Weihrauch. *Computable Analysis: An Introduction*. Springer, 2000.

[25] Benjamin Weiss. *Single Orbit Dynamics*. Number 95 in Regional Conference Series in Mathematics. American Mathematical Society, 2000.

[26] Alexander K. Zvonkin and Leonid A. Levin. The complexity of finite objects and the development of the concepts of information and randomness by means of the theory of algorithms. *Russian Mathematical Surveys*, 25(6):83, 1970.