Stochastic Languages at Sub-stochastic Cost

Smayan Agarwal and Aalok Thakkar

Ashoka University

Abstract. When does a deterministic computational model define a probability distribution? What are its properties? This work formalises and settles this stochasticity problem for weighted automata, and its generalisation cost register automata (CRA).
We show that checking stochasticity is undecidable for CRAs in general. This motivates the study of the fully linear fragment, where a complete and tractable theory is established. For this class, stochasticity becomes decidable in polynomial time via spectral methods, and every stochastic linear CRA admits an equivalent model with locally sub-stochastic update functions. This provides a local syntactic characterisation of the semantics of the quantitative model.
This local characterisation allows us to provide an algebraic Kleene-Schützenberger characterisation for stochastic languages. The class of rational stochastic languages is the smallest class containing finite support distributions, which is closed under convex combination, Cauchy product, and discounted Kleene star. We also introduce Stochastic Regular Expressions as a complete and composable grammar for this class.
Our framework provides the foundations for a formal theory of probabilistic computation, with immediate consequences for approximation, sampling, and distribution testing.

Keywords: Quantitative Automata · Cost Register Automata · Stochastic Languages · Spectral Methods

1 Introduction

A fundamental question in formal language theory asks: *Which functions on strings give rise to probability distributions?* More precisely, given a quantitative automaton that assigns real-valued weights to words through algebraic computations, when does this function, after normalization, define a valid probability measure over Σ^*? This stochasticity problem sits at the confluence of automata theory, linear algebra, and probability theory, connecting classical questions about regularity with modern challenges in probabilistic modelling.

The question is deceptively simple. Unlike classical probabilistic automata [30, 29], which embed randomness directly into their operational semantics, weighted automata [23] and cost register automata (CRAs) [1] compute weights by accumulating numeric values through matrix multiplication or register updates, producing a function $f : \Sigma^* \to \mathbb{R}$ without internal probabilistic choices. The stochasticity question asks whether this deterministically computed function satisfies two global conditions: non-negativity ($f(w) \geq 0$ for all w) and normalization ($\sum_{w \in \Sigma^*} f(w) = 1$). Answering this requires reasoning about infinite summations over all possible strings–a task that, as we show, is undecidable in general.

Motivation. Probability distributions over strings pervade computer science. In natural language processing, stochastic grammars model syntactic structure [16, 28], while neural language models implicitly define distributions over token sequences [12]. Biological sequence analysis relies on probabilistic models of DNA and protein strings, where hidden Markov models capture evolutionary dynamics [25]. Formal verification employs probabilistic specifications to reason about quantitative properties [4, 7]. Information theory studies optimal compression via string distributions [19]. Quantitative automata theory offers formal models for their study. When they define stochastic languages, they provide interpretable, verifiable probability models with decidable properties and compositional structure. Understanding *when* they are stochastic, and *why*, is thus both theoretically natural and practically consequential.

Results and Contributions. Our focus on deterministically computed distributions provides a unifying framework for these domains. We study cost register automata as a model of computing probability distributions over strings. Our main contributions are:

1. We prove that determining whether a CRA defines a stochastic language is undecidable in general (Theorems 1 and 2). To recover decidability, we identify and analyse the fragment of CRAs with affine updates. We establish that stochasticity decision problem decidable for this fragment (Theorem 4).
2. Our second main contribution proves that every linear CRA defining a stochastic language admits a semantically equivalent representation in which stochasticity becomes local and can be checked by computing the sum of transition weights of each register (Theorem 5).
3. We provide a Kleene-Schützenberger characterization for rational stochastic languages (Theorem 6), and introduce stochastic regular expressions as a grammar for rational stochastic languages.

Our results chart a precise boundary between decidable and undecidable semantic properties of quantitative automata and provide sub-stochastic and local characterisation for CRA with linear updates. We also present three open questions on *stochasticity decision problem* for general quantitative models, *minimal approximation methods*, and *distribution testing* for stochastic languages. These results and problems bridge automata theory and statistical inference, opening new directions for both theoretical investigation, modelling of stochastic sequences, and validating probabilistic generative models against formal distributional specifications.

2 Related Work

Stochastic languages formalize probability distributions over words, with canonical representations via weighted automata [23] and probabilistic grammars [22]. These frameworks connect formal language theory with probabilistic modeling, supporting applications in grammar induction [17], and natural language processing [18]. Despite this extensive development, there is no established syntactic characterization that delineates exactly the class of stochastic languages definable by such models.

Early probabilistic automata, introduced by Rabin and Paz, localize randomness in the transition relation: outgoing transitions from each state form a stochastic distribution, and the probability of accepting a word is obtained by summing over all accepting runs labeled by that word [30, 29]. This yields a probabilistic acceptance semantics governed by cutpoint conditions. For ω-languages, probabilistic Büchi automata assign Markov chains to runs and classify behaviors by almost-sure, positive, or threshold probability acceptance [3]. Chatterjee, Doyen, and Henzinger later generalized these models to probabilistic weighted automata, where transitions are probabilistic but evaluations are quantitative, defining random variables over runs rather than word-level distributions [15, 9]. In concurrency theory, Segala and Lynch formalized probabilistic automata as systems with randomized transitions, introducing semantic notions of probabilistic bisimulation [32]. Across these variants, randomness arises operationally along runs rather than as a distribution over the word space.

On the other hand, Cost Register Automata (CRA), introduced by Alur et al. [1], provide a deterministic, register-based model for quantitative functions in which a finite set of registers is updated along a run using algebraic operations. The model generalizes classical weighted automata, while retaining explicit operational semantics. More recently, Benalioua, Lhote, and Reynier analyzed CRA with linear and affine updates over fields, demonstrating equivalence in expressive power with weighted automata and developing minimization algorithms with matching complexity bounds [8]. Together, these lines of work position CRA as a deterministic and compositional alternative to path-summing weighted models, offering a good fit for extending quantitative automata theory to probabilistic or normalized semantics.

A distinct line of work, initiated by Denis and Esposito [21, 22], studied rational stochastic languages through multiplicity automata. While such models provide an algebraic vehicle for representing stochastic languages, the literature still lacks a clear syntactic or compositional characterization analogous to the Kleene–Schützenberger theorem for rational stochastic languages. In particular, it remains unclear whether stochastic languages admit a closed algebra of regular-like expressions corresponding precisely to the distributions generated by these automata.

Our work addresses this gap by developing a cost register automata based framework for stochastic languages, proving a locally sub-stochastic characterisation, and introducing Stochastic Regular Expressions that give an equivalence between algebraic and automata-theoretic descriptions of stochastic languages.

3 Preliminaries and Notation

In this section we recall the basic definitions and notation used throughout the paper. Our focus is on *quantitative languages* over non-negative reals, their representation by *weighted* and *cost register automata*, and a few linear-algebraic tools used to reason about convergence.

3.1 Quantitative Languages

Let Σ be a finite alphabet and Σ^* the set of all finite words over Σ. For the purpose of our discussion, a *quantitative language* over Σ is a function $f : \Sigma^* \to \mathbb{R}$, that assigns a real-valued weight to each word. For a general treatment of quantitative languages, see [23, 24].

Definition 1 (Mass). *Let $L \subseteq \Sigma^*$ be a (possibly finite or infinite) set of words. The mass of a quantitative language $f : \Sigma^* \to \mathbb{R}$ on L is defined as*

$$\mathrm{mass}\,(f)_L = \sum_{w \in L} f(w),$$

whenever this sum is well-defined. In particular, the total mass of f is $\mathrm{mass}\,(f)_{\Sigma^*}$.

A quantitative language f is $f$inite-mass if mass_{Σ^*} is finite. [24]

Example 1. Let $\Sigma = \{a, b\}$. Let $f : \Sigma^* \to \mathbb{R}$ be defined as:

$$f(w) = \frac{1}{(|w| + 1)^2 \cdot 2^{|w|}},$$

where $|w|$ denotes the length of w (so that $|\varepsilon| = 0$). Then f is a finite-mass language as $\mathrm{mass}\,(f)_{\Sigma^*} = \frac{\pi^2}{6}$.

Definition 2 (Stochastic Language). *A quantitative language $f : \Sigma^* \to \mathbb{R}$ is stochastic if for all $w \in \Sigma^*$, $f(w) \geq 0$ and* $\mathrm{mass}\,(f)_{\Sigma^*} = 1$. *The set $\mathcal{S}(\Sigma^*)$ denotes the set of all stochastic languages over Σ^* [22, 24].*

Example 2 (Dirac Distribution). For a fixed string $w \in \Sigma^*$, the *Dirac distribution* δ_w is a stochastic language over Σ^* defined for all $u \in \Sigma^*$ as:

$$\delta_w(u) = \begin{cases} 1 & \text{if } u = w, \\ 0 & \text{otherwise.} \end{cases}$$

Example 3. Consider the function f in Example 1. The function $\overline{f} : w \mapsto \frac{6f(w)}{\pi^2}$ is a stochastic language. In general, if f is a finite-mass language such that for all $w \in \Sigma^*$, $f(w) \geq 0$, and $\mathrm{mass}\,(f)_{\Sigma^*} \neq 0$, the *normalized* function $\overline{f} : w \mapsto f(w)/\mathrm{mass}\,(f)_{\Sigma^*}$ is a stochastic language.

Definition 3. *We can now define some operations over stochastic languages. Let f_1 and f_2 be two stochastic languages, and let $\lambda \in (0, 1)$.*

1. *Their convex combination parametrised by λ is:*

$$\lambda f_1 + (1-\lambda) f_2 : w \mapsto \lambda f_1(w) + (1-\lambda) f_2(w)$$

2. *Their Cauchy product is:*

$$f_1 \cdot f_2 : w \mapsto \sum_{w=uv} f_1(u) \cdot f_2(v)$$

3. *The* d*iscounted Kleene Star of f_1 parameterised by λ is:*

$$(f_1)^*_\alpha : w \mapsto \sum_{k=1}^{\infty} \sum_{\substack{w_1,\ldots,w_k \in \Sigma^+ \\ w = w_1 \cdots w_k}} \alpha(1-\alpha)^{k-1} \prod_{i=1}^{k} f_1(w_i).$$

Intuitively, $f^*_\alpha(w)$ defines a distribution over strings obtained by concatenating k non-empty substrings, each independently drawn from r, with the total number of substrings following a shifted geometric distribution with parameter α. This definition implicitly sets $r(\epsilon) = 0$. This is in accordance with standard definitions [10]. It is immediate that stochastic languages are closed under the above three operators.

3.2 Quantitative Automata

Weighted finite automata (WFAs) are the canonical model for quantitative languages over semirings [10, 24, 5]. A *semiring* is a set K with two binary operations $+$ and $\cdot$ and two constant elements 0 and 1 such that $(K, +, 0)$ is a commutative monoid, $(K, \cdot, 1)$ is a monoid, and $+$ distributes over $\cdot$ operation. We will work with the semirings $(\mathbb{R}, +, \cdot, 0, 1)$ and $(\mathbb{R}_{\geq 0}, +, \cdot, 0, 1)$.

Definition 4 (Weighted Finite Automata (WFA)). *A weighted automaton $\mathcal{A}$ is a tuple $(\Sigma, Q, \lambda, M, \mu)$ over the semiring K where:*

1) Σ is a finite alphabet,
2) Q is a finite state set,
3) $\lambda : Q \to K$ is the initial weight vector,
4) $M : \Sigma \to K^{Q \times Q}$ assigns a transition matrix M_σ to each symbol,
5) $\mu : Q \to K$ is the final weight vector.

The semantics of $\mathcal{A}$ is the quantitative language

$$[\![\mathcal{A}]\!](w) = \lambda^\top M_{w_1} \cdots M_{w_n} \mu \quad \textit{for } w = w_1 \cdots w_n \in \Sigma^*.$$

The set $\mathcal{S}^{rat}_K(\Sigma^*)$ represents the set of stochastic languages over Σ^* that can be expressed by a weighted automaton over the semiring K. By restricting to the non-negative models in $\mathcal{S}^{rat}_{\mathbb{R}_{\geq 0}}(\Sigma^*)$, we give up a bit of expressivity. Indeed, $\mathcal{S}^{rat}_{\mathbb{R}_{\geq 0}}(\Sigma^*) \subsetneq \mathcal{S}^{rat}_{\mathbb{R}}(\Sigma) \subsetneq \mathcal{S}(\Sigma)$ [22].

Cost Register Automata (CRAs), introduced by Alur et al. [1], offer an alternative representation in which weights are stored explicitly in registers and updated deterministically during the run.

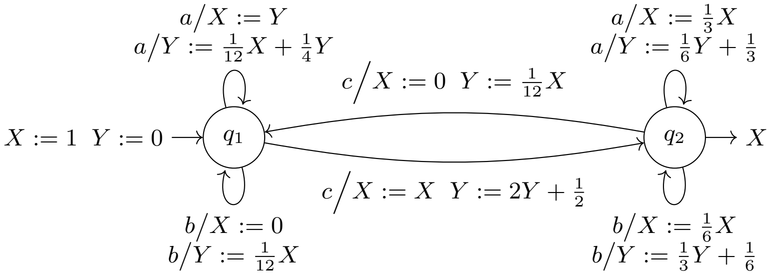
$a/X := Y$
$a/Y := \frac{1}{12}X + \frac{1}{4}Y$
$a/X := \frac{1}{3}X$
$a/Y := \frac{1}{6}Y + \frac{1}{3}$
$c/X := 0 \;\; Y := \frac{1}{12}X$
$X := 1 \;\; Y := 0$
q_1
q_2
X
$c/X := X \;\; Y := 2Y + \frac{1}{2}$
$b/X := 0$
$b/Y := \frac{1}{12}X$
$b/X := \frac{1}{6}X$
$b/Y := \frac{1}{3}Y + \frac{1}{6}$

Fig. 1. A two-state cost register automaton (CRA) over alphabet $\{a, b, c\}$ with registers X and Y. Transitions are labeled by input symbols and associated affine register updates. The initial valuation is $X = 1, Y = 0$, and the output in the accepting state q_2 is the value of register X.

Definition 5 (Cost Register Automaton). *A real-valued CRA is a tuple $\mathcal{A} = (\Sigma, Q, X, q_0, x_0, \delta, f, \mu)$ where*

1) Σ is a finite alphabet, and Q a finite state set;
2) X is a finite set of registers;
3) $q_0 \in Q$ is the initial state, and $x_0 : X \to \mathbb{R}$ the initial valuation;
4) $\delta : Q \times \Sigma \to Q$ is a deterministic transition function;
5) $f : Q \times \Sigma \times \mathbb{R}^X \to \mathbb{R}^X$ specifies register updates;
6) $\mu : Q \times \mathbb{R}^X \to \mathbb{R}$ is the output function.

The semantics $[\![\mathcal{A}]\!] : \Sigma^* \to \mathbb{R}$ is given by

$$[\![\mathcal{A}]\!](w) = \mu(q_n, x_n),$$

where $(q_0, x_0) \xrightarrow{w_1} (q_1, x_1) \xrightarrow{w_2} \cdots \xrightarrow{w_n} (q_n, x_n)$ is the unique run with $x_i = f(q_{i-1}, w_i, x_{i-1})$.

CRAs naturally capture deterministic weighted computations and are expressive enough to represent rational stochastic languages when instantiated with affine updates. We will use the cost register automata in Figure 1 as our running example.

3.3 Matrix Properties and Convergence

We recall standard results from non-negative matrix theory (see [27]) that will be used to analyse convergence of weighted computations. The results discussed here are for real valued square matrices in $\mathbb{R}^{n \times n}$.

1. A non-negative square matrix A is *irreducible* if for all i, j there exists $k \geq 1$ such that $(A^k)_{ij} > 0$.
2. The *spectral radius* $\rho(\cdot)$ of a square matrix is the maximum of the absolute values of its eigenvalues. The spectral radius of a bounded operator is the supremum of the absolute values of the elements of its spectrum.

3. A non-negative square matrix A is *row sub-stochastic* if for every row i, $\sum_j A_{ij} \leq 1$. It is strictly sub-stochastic if for at least one row, the sum is stictly less than 1.
4. (Perron-Frobenius Theorem) If a non negative matrix A is irreducible, then:

$$\min_i \sum_j A_{ij} \leq \rho(A) \leq \max_i \sum_j A_{ij}.$$

   In particular, if it is strictly sub-stochastic, then $\rho(A) < 1$.
5. (Neumann Series) For a square matrix A, the series $I + A + A^2 + A^3 \dots$ converges if and only if $\rho(A) < 1$. In this case, it converges to $(I - A)^{-1}$.

$\rho(A)$ or $(I - A)^{-1}$ can be computed in $O(n^3)$ time.

4 Problem Formulation and Decidability Results

This work is motivated by a central semantic question:

> *Given a cost register automaton (CRA), when does it define a stochastic language?*

This problem connects the operational view of quantitative automata with the analytic notion of probability distributions over strings.

Problem 1 (Stochasticity Decision Problem). Given a CRA $\mathcal{A}$, determine whether its semantics satisfies:

(i) Non-negativity: $[\![\mathcal{A}]\!](w) \geq 0$ for all $w \in \Sigma^*$, and
(ii) Normalization: the total mass equals one,

$$\text{mass}\left([\![\mathcal{A}]\!]\right)_{\Sigma^*} = \sum_{w \in \Sigma^*} [\![\mathcal{A}]\!](w) = 1.$$

Intuitively, a CRA represents a stochastic language when its register updates behave like probability-preserving transitions and the total accumulated weight over all strings equals 1. In what follows, we chart the boundary between decidable and undecidable instances of this problem.

4.1 Undecidability of the Stochasitcity Decision Problem

We first consider the more basic question of whether a given CRA has *finite total mass.* Even this relaxed problem turns out to be undecidable.

Theorem 1. *Given a CRA $\mathcal{A}$ with affine register updates, it is undecidable whether* $\text{mass}\left([\![\mathcal{A}]\!]\right)_{\Sigma^*}$ *converges.*

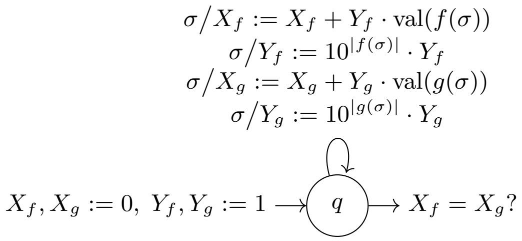
$\sigma/X_f := X_f + Y_f \cdot \mathrm{val}(f(\sigma))$
$\sigma/Y_f := 10^{|f(\sigma)|} \cdot Y_f$
$\sigma/X_g := X_g + Y_g \cdot \mathrm{val}(g(\sigma))$
$\sigma/Y_g := 10^{|g(\sigma)|} \cdot Y_g$
$X_f, X_g := 0,\ Y_f, Y_g := 1$
q
$X_f = X_g?$

Fig. 2. A single-state CRA used in the reduction from PCP in the proof of Theorem 1. The registers X_f and X_g accumulate the numerical value of interpretations of two transductions f and g over input w, and Y_f and Y_g maintain their lengths. The output function checks if $X_f = X_g$ and returns 1, or returns 0 otherwise.

Proof. We reduce from the *Post Correspondence Problem (PCP)*. Let the instance be given by pairs of strings $\{(f(i), g(i))\}_{i=1}^{k}$ over the digit alphabet $\Sigma = \{1, \ldots, 9\}$. We construct a single-state CRA $\mathcal{A}$ (Figure 2) with registers X_f, Y_f, X_g, Y_g initialized as $(0, 1, 0, 1)$. Upon reading symbol $\sigma \in \{1, \ldots, k\}$, the updates are:

$$X_f := X_f + Y_f \cdot \mathrm{val}(f(\sigma)), \quad Y_f := 10^{|f(\sigma)|} \cdot Y_f,$$
$$X_g := X_g + Y_g \cdot \mathrm{val}(g(\sigma)), \quad Y_g := 10^{|g(\sigma)|} \cdot Y_g.$$

The output $\mu(q, (X_f, Y_f, X_g, Y_g)) = 1$ iff $X_f = X_g$, and 0 otherwise.

For a word $w = \sigma_1 \cdots \sigma_n$, the registers (X_f, Y_f) encode the base-10 numeric value and positional multiplier for $f(w)$, and analogously for $g(w)$. Hence, $[\![\mathcal{A}]\!](w) = 1$ iff $f(w) = g(w)$.

If the PCP instance admits a solution w, then $[\![\mathcal{A}]\!](w^k) = 1$ for all $k \geq 1$, so

$$\mathrm{mass}\left([\![\mathcal{A}]\!]\right)_{\Sigma^*} \geq \sum_{k=1}^{\infty} [\![\mathcal{A}]\!](w^k) = \infty.$$

If no solution exists, $[\![\mathcal{A}]\!](w) = 0$ for all nonempty w, and $[\![\mathcal{A}]\!](\varepsilon) = 1$, giving $\mathrm{mass}\left([\![\mathcal{A}]\!]\right)_{\Sigma^*} = 1$. Thus, deciding convergence of $\mathrm{mass}\left([\![\mathcal{A}]\!]\right)_{\Sigma^*}$ is equivalent to solving PCP. □

The proof of Theorem 1 crucially relies on a non-polynomial output function (testing equality). This naturally leads us to consider the *fully polynomial fragment*, where both the register updates and output functions are restricted to polynomials. We show a general property of polynomial dynamical systems:

Theorem 2. *Let $p : \mathbb{R}^d \to \mathbb{R}^d$ be a polynomial map, $g : \mathbb{R}^d \to \mathbb{R}$ be a polynomial function, and $x_0 \in \mathbb{R}^d$ be an initial vector. It is undecidable to determine if the following series converges:*

$$S(p, g, x_0) = \sum_{n=0}^{\infty} g(p^{(n)}(x_0)). \tag{1}$$

Proof. We reduce from the halting problem for Turing machines. By classical constructions [11, 26], for every Turing machine M and input w, there exist d, a polynomial map $p : \mathbb{R}^d \to \mathbb{R}^d$, and rational vectors $x_0, q \in \mathbb{Q}^d$ such that the iteration $x_{n+1} = p(x_n)$ simulates the configuration of $M(w)$:

1. if $M(w)$ halts after N steps, then $x_n = q$ for all $n \geq N$, and
2. if $M(w)$ does not halt, then $\|x_n - q\| \geq 1$ for all n.

Define the function $g : \mathbb{R}^d \to \mathbb{R}$ by $g(x) = \|x - q\|^2 = \sum_{i=1}^{d}(x_i - q_i)^2$. Then $g(x_n) = 0$ eventually iff $M(w)$ halts. That is,

$$M(w) \text{ halts} \iff \sum_{n=0}^{\infty} g(p^{(n)}(x_0)) \text{ converges.}$$

Thus, deciding convergence of Equation 1 is undecidable. □

These results imply that checking whether a CRA has finite total mass is undecidable, both for CRAs restricted to *affine* register updates with *non-linear* outputs (Theorem 1), and for the *fully polynomial* fragment with polynomial updates and outputs (Theorem 2).

Moreover, given a CRA $\mathcal{A}$, is it undecidable whether $[\![\mathcal{A}]\!](w) \geq 0$ for all $w \in \Sigma^*$. These follow from classical results for weighted finite automata (WFAs) where the boundedness problem over the non-negative rationals is known to be undecidable [20]. Since CRAs strictly generalize WFAs [1], it follows that neither non-negativity nor normalization can be algorithmically checked for general cost register automata.

4.2 Linear Cost Register Automata

To recover decidability, we focus on a restricted and well-behaved fragment of cost register automata. First, to guarantee non-negativity of register valuations, we define the automaton over the positive semiring $(\mathbb{R}_{\geq 0}, +, \times, 0, 1)$ only, ensuring that all intermediate and output values remain non-negative. Second, we constrain both register updates and output expressions to be *affine*, that is, linear transformations combined with additive constants. The analysis of affine CRAs can be further simplified using a standard linearization technique:

Lemma 1. *The class of functions computable by CRAs with linear updates coincide with the class of functions computable by CRAs with affine updates.*

Proof. (Sketch)
Let $\mathcal{A} = (Q, \Sigma, X, \delta, \rho, q_0, \mu)$ be a CRA with affine updates of the form $x \mapsto A_{q,\sigma}x + b_{q,\sigma}$. We add a new register x_0 that always holds 1. Replacing each affine update by the linear map $(x, x_0) \mapsto \begin{bmatrix} A_{q,\sigma} & b_{q,\sigma} \\ 0 & 1 \end{bmatrix}(x, x_0)$ and output by $\mu'(x, x_0) = [c_q^\top \; d_q](x, x_0)$ preserves semantics. □

These restrictions yield the class of *linear CRAs*, which is placed within a well-studied computational model whose dynamics can be represented by systems of linear transformations over the non-negative reals. Linear CRAs admit an exact correspondence with weighted automata:

Theorem 3 (Alur et al. [1]). *The class of functions computed by CRAs with linear updates coincides with those computed by weighted automata.*

The key idea is that a linear CRA with k registers and m states can be converted to an equivalent weighted automaton with $m \cdot k$ states by treating each register-value pair as a state, with transitions derived from the linear update functions. This equivalence immediately yields numerous closure properties and normal forms for linear CRAs. We can now prove our first main result:

Theorem 4. *Given a CRA $\mathcal{A}$ over $\mathbb{R}_{\geq 0}$ with affine register updates and affine output function, the total mass* $\operatorname{mass}(\llbracket \mathcal{A} \rrbracket)_{\Sigma^*} = \sum_{w \in \Sigma^*} \llbracket \mathcal{A} \rrbracket(w)$ *is computable.*

Proof. By Lemma 1, we may assume without loss of generality that $\mathcal{A}$ has linear updates. By Theorem 3, $\mathcal{A}$ is equivalent to a weighted automaton. Let $\mathcal{W}$ be its *trimmed* version (containing only states that are both reachable from an initial state and co-reachable from a final state), with n states $Q = \{1, \ldots, n\}$.

Let $M_\sigma \in \mathbb{R}^{n \times n}_{\geq 0}$ be the transition matrix for symbol $\sigma \in \Sigma$, where $(M_\sigma)_{i,j}$ is the weight of transitioning from state i to state j on σ. Let $\iota \in \mathbb{R}^n_{\geq 0}$ be the initial weight vector and $c \in \mathbb{R}^n_{\geq 0}$ the final weight vector.

Define the transition matrix $M = \sum_{\sigma \in \Sigma} M_\sigma$. The trimmed automaton $\mathcal{W}$ is strongly connected by construction, and hence M a nonnegative irreducible matrix. The total weight of all strings of length k is given by $c^\top M^k \iota$. Therefore, the total mass is:

$$\operatorname{mass}(\llbracket \mathcal{A} \rrbracket)_{\Sigma^*} = \sum_{w \in \Sigma^*} \llbracket \mathcal{A} \rrbracket(w) = c^\top \left(\sum_{k=0}^{\infty} M^k \right) \iota .$$

This Neumann series converges if and only if the spectral radius $\rho(M) < 1$. The spectral radius of a nonnegative matrix irreducible matrix is computable as the largest real eigenvalue by the Perron-Frobenius theorem [27].

- If $\rho(M) \geq 1$, then $\operatorname{mass}(\llbracket \mathcal{A} \rrbracket)_{\Sigma^*}$ diverges.
- If $\rho(M) < 1$, then $\operatorname{mass}(\llbracket \mathcal{A} \rrbracket)_{\Sigma^*}$ converges to $c^\top (I - M)^{-1} \iota$.

The computation involves computing $\rho(M)$ via eigenvalue computation in $O((|Q| \cdot |X|)^3)$ time, and if the spectral radius $\rho(B) < 1$, then solving the linear system $(I - M)x = \iota$ via Gaussian elimination in $O((|Q| \cdot |X|)^3)$ time. Thus, the total mass is computable in polynomial time in the size of the input CRA $\mathcal{A}$. □

Example 4. The WFA in Figure 3 is equivalent to the CRA in Figure 1. For this WFA, we have:

$$M_a + M_b + M_c = \begin{pmatrix} 0 & \frac{1}{6} & 1 \\ \frac{3}{4} & \frac{1}{4} & 0 \\ 0 & \frac{1}{12} & \frac{1}{2} \end{pmatrix} .$$

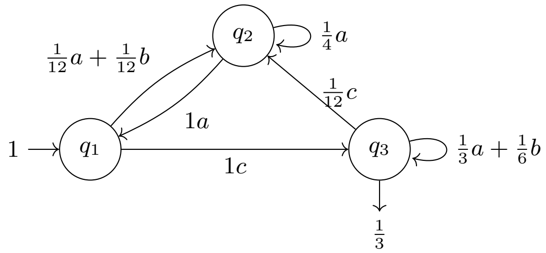

Fig. 3. A weighted automaton over $\Sigma = \{a, b, c\}$ with initial weight 1 at q_1 and final weight $\frac{1}{3}$ at q_3. Transition labels denote linear combinations of weighted symbol transitions (e.g., $\frac{1}{12}a + \frac{1}{12}b$ means weight $\frac{1}{12}$ on a and $\frac{1}{12}$ on b). This weighted automaton is equivalent to the cost register automata in Figure 1.

Its row sums are $(1\frac{1}{6}, 1, \frac{7}{12})$ and $\rho(M) = \frac{3}{4}$. The total mass $c^\top(I - M)^{-1}\iota = 1$, confirming that the automaton defines a normalized stochastic language.

In summary, while the stochasticity problem is undecidable in general (Theorem 1), it becomes efficiently decidable for the affine fragment. For affine CRAs over $\mathbb{R}_{\geq 0}$: $\rho(M) < 1$ ensures finite mass, and the positive semiring guarantees non-negativity. Normalizing the output by mass $(\llbracket \mathcal{A} \rrbracket)_{\Sigma^*}$ yields a stochastic language. This fragment is thus a useful model for representing probability distributions over strings.

5 Localizable Stochasticity and Sub-stochastic Cost

The decidability result in Theorem 4 establishes that the total mass of a linear CRA is computable whenever the combined transition matrix $M = \sum_{\sigma \in \Sigma} M_\sigma$ is irreducible and has spectral radius $\rho(M) < 1$.

In contrast, classical probabilistic models enforce local properties: each state's outgoing weights sum to 1, guaranteeing that the semantics forms a probability distribution without the need for a spectral analysis [10].

In this section, we show that every finite-mass weighted automaton (and hence every CRA with affine updates defining a stochastic language) admits an equivalent representation in which stochasticity is *localizable.*

5.1 Locally Sub-stochastic Weighted Automata

Formally, we show that any WFA with only reachable and co-reachable states and $\rho(M) < 1$ can be normalized so that the outgoing weights from each state sum to at most 1. We call this property *local sub-stochasticity.* Such normalization can be achieved by a similarity transform based on the Perron–Frobenius eigenvector, and it preserves the global semantics of the automaton.

Proposition 1 (Perron–Frobenius normalization). *Let $A \in \mathbb{R}^{n \times n}_{\geq 0}$ be an irreducible matrix with spectral radius $\rho(A) < 1$. Then there exists a positive*

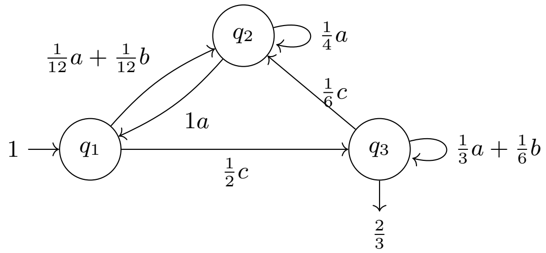

Fig. 4. A weighted automaton over $\Sigma = \{a, b, c\}$ with initial weight 1 at q_1 and final weight $\frac{2}{3}$ at q_3. This automaton is semantically equivalent to the one in Figure 3, and is sub-stochastic, that is the sum of outgoing weights of every state at most 1.

diagonal matrix $D = \operatorname{diag}(d_1, \ldots, d_n)$ *with* $d_i > 0$ *such that* $B = DAD^{-1}$ *is strictly row sub-stochastic.*

Proof (Sketch). By the Perron–Frobenius theorem, there exists a unique (up to scaling) positive eigenvector $v \in \mathbb{R}_{\geq 0}$ such that $Av = \rho(A)v$. Since $\rho(A) < 1$, we have $Av < v$ component-wise. Let $D = \operatorname{diag}(v_1, \ldots, v_n)$ and define $B = DAD^{-1}$. Then for each row i,

$$\sum_j B_{ij} = \sum_j \frac{v_i}{v_j} A_{ij} = \frac{(Av)_i}{v_i} \leq 1,$$

as required. As $\rho(A) < 1$, for some row i, $\sum_j B_{ij} < 1$. □

Given a WFA $\mathcal{W} = (\Sigma, Q, \lambda, \{M_\sigma\}, \mu)$, define $M = \sum_\sigma M_\sigma$. Applying Theorem 1 to M yields a diagonal matrix D such that the similarity transform $M' = DMD^{-1}$ is row-substochastic. We then define a *normalised automaton* $\mathcal{W}' = (\Sigma, Q, \lambda', \{M'_\sigma\}, \mu')$ with

$$M'_\sigma = DM_\sigma D^{-1}, \qquad \lambda' = D^{-1}\lambda, \qquad \mu' = D\mu.$$

The semantics of $\mathcal{W}'$ satisfies $[\![\mathcal{W}']\!](w) = [\![\mathcal{W}]\!](w)$ for all $w \in \Sigma^*$, since each path weight is invariant under this rescaling. Moreover, every row of M' now sums to at most 1, hence the automaton is *locally sub-stochastic*. Figure 4 illustrates this normalization for the running example.

This gives us characterisation of rational stochastic languages:

Theorem 5 (Local Sub-stochastic Characterisation). *A quantitative language* f *belongs to the class of stochastic rational languages* $\mathcal{S}^{rat}_{\mathbb{R}_{\geq 0}}(\Sigma^*)$ *if and only if there exists a* locally sub-stochastic *weighted automaton* $\mathcal{W}$ *such that* $[\![\mathcal{W}]\!] = f$.

The notion of rational stochastic languages has been investigated in several equivalent frameworks, including probabilistic automata and rational series over the non-negative reals ([10, 22]). The local sub-stochastic normalization bridges these perspectives, revealing that the analytic convergence condition $\rho(M) < 1$ is equivalent to the existence of a locally normalizable form.

5.2 Stochastic Regular Expressions

This local sub-stochasticity enables a Kleene-Schützenberger characterization of rational stochastic languages. We introduce *stochastic regular expressions* (SREs), which extend ordinary regular expressions with a probabilistic choice and discounted iteration operators. SREs capture exactly the class of quantitative languages definable by a sub-stochastic weighted automata.

Definition 6 (Syntax of Stochastic Regular Expressions). *Let Σ be a finite alphabet. The grammar of SREs is*

$$r ::= \delta_\sigma \mid \alpha r_1 + (1-\alpha) r_2 \mid r_1 \cdot r_2 \mid r_\alpha^*$$

where $\sigma \in \Sigma$, r_1, r_2 are SREs, and $\alpha \in (0,1)$.

Here δ_σ denotes the Dirac distribution concentrated on σ (see Example 2), the operator $\alpha r_1 + (1-\alpha) r_2$ represents convex combination, $r_1 \cdot r_2$ denotes the Cauchy product (probabilistic concatenation), and r_α^* denotes a *discounted Kleene star*, where the number of repetitions follows a shifted geometric distribution with parameter α (see Definition 3). We can now prove our characterisation theorem:

Theorem 6 (Kleene-Schützenberger Characterisation of Stochastic Regular Languages). *The class of stochastic regular languages, denoted by $\mathcal{S}^{rat}_{\mathbb{R}_{\geq 0}}(\Sigma^*)$, is the smallest class of quantitative languages over Σ that contains all Dirac distributions $\{\delta_\sigma \mid \sigma \in \Sigma\}$ and is closed under convex combinations, Cauchy products, and discounted Kleene star.*

The proof follows by constructing equivalences between locally sub-stochastic weighted automata and stochastic regular expressions (SREs). It parallels the construction between nondeterministic automata and regular expressions.

Lemma 2. *Every language defined by a locally sub-stochastic weighted automaton can be represented by a Stochastic Regular Expression (SRE).*

Proof (Sketch). We adapt the standard state-elimination construction, exploiting that local sub-stochasticity ensures all geometric series that appear during elimination converge.

Given a normalized weighted automaton $\mathcal{W} = (\Sigma, Q, \lambda, T, \mu)$, augment it with a new start state q_s and final state q_f, connected by weighted ε-transitions according to λ and μ. Then iteratively eliminate all intermediate states $q_k \in Q$, replacing paths $q_i \to q_j$ that pass through q_k by a single edge labelled with the expression

$$R_{ij} = T(q_i, \sigma, q_k)\, \delta_\sigma \cdot \Big(\sum_\sigma \tfrac{T(q_k,\sigma,q_k)}{W_k} \delta_\sigma \Big)^*_{1-T(q_k,\Sigma,q_k)} \cdot R_{kj},$$

where W_k is the total outgoing weight from q_k. Because $T(q_k, \Sigma, q_k) < 1$, every such geometric expansion is well defined. After all eliminations, the remaining edge $q_s \to q_f$ is labeled with an SRE r satisfying $[\![r]\!] = [\![\mathcal{W}]\!]$. □

Note that the size of the SRE can be exponential in the size of the weighted automaton [24]. Conversely:

Lemma 3. *Every SRE r defines a stochastic language that can be realized by a locally* stochastic weighted automaton*, that is, for every state, the sum of weights of outgoing transitions is exactly* 1. □

Proof. The proof proceeds by structural induction on r. For $r = \delta_\sigma$, define the two-state automaton $\mathcal{A}_\sigma = (\Sigma, \{q_i, q_f\}, \lambda, T, \mu)$ with $\lambda(q_i) = 1$, $T(q_i, \sigma, q_f) = 1$, and $\mu(q_f) = 1$. Clearly $\sum_w [\![\mathcal{A}_\sigma]\!](w) = 1$. For the operators:

1. (Convex Combination) For $r = \alpha r_1 + (1 - \alpha) r_2$, given automata $\mathcal{A}_1, \mathcal{A}_2$ for r_1, r_2 on disjoint state sets, scale the initial weights of $\mathcal{A}_1$ and $\mathcal{A}_2$ by α and $(1 - \alpha)$, respectively. Linearity of semantics yields $\sum_w [\![\mathcal{A}]\!](w) = 1$.
2. (Cauchy Product) For $r = r_1 \cdot r_2$, connect every final state of $\mathcal{A}_1$ to every initial state of $\mathcal{A}_2$ by ε-transitions weighted by $\mu_1(p)\lambda_2(q)$. This realizes the Cauchy product of distributions and preserves total mass.
3. (Discounted Kleene Star) For $r = (r_1)^*_\alpha$, introduce discount factor $\alpha \in (0, 1)$. Let $\mathcal{A}_1$ realize r_1. Each repetition contributes a multiplicative $(1-\alpha)$ factor, and termination contributes α. Thus

$$\sum_{w \in \Sigma^+} [\![\mathcal{A}]\!](w) = \sum_{k \geq 1} \alpha (1 - \alpha)^{k-1} \Big(\sum_w [\![\mathcal{A}_1]\!](w) \Big)^k = 1.$$

By construction, the total mass of the automaton is exactly 1 and for every state, the sum of weights of outgoing transitions is exactly 1. □

Together, these two lemmas complete the proof for Theorem 6. Intuitively, the nested discounted stars correspond to loops in the automaton whose expected number of traversals follows a geometric law. For our running example, the stochastic regular expression is:

$$\left(\left(\frac{1}{4} a + \frac{1}{4} b + \frac{1}{4} c \left(\frac{2}{3} a + \frac{1}{3} b \right)^*_{\frac{1}{2}} c \right)^*_{\frac{2}{3}} (a)^*_{\frac{3}{4}} a \right) c \left(\frac{2}{3} a + \frac{1}{3} b \right)^*_{\frac{1}{2}}$$

In summary, Theorems 5 and 6 provide a local characterisation of probabilistic behaviour while preserving global convergence semantics.

6 Approximation and Statistical Inference for Stochastic Languages

In this section, we move beyond exact characterization to the *quantitative* analysis of stochastic languages: their approximation, sampling, and learnability. We also introduce a minimal testing framework that connects these ideas to classical distribution testing in statistics.

Universal Approximation. We begin by showing that stochastic regular languages ($\mathcal{S}^{rat}_{\mathbb{R}_{\geq 0}}(\Sigma^*)$) are dense in the space of all stochastic languages ($\mathcal{S}(\Sigma)$) under the ℓ_1 (total variation) metric. This follows from the simple observation that any distribution over Σ^+ can be approximated arbitrarily well by truncating its support.

Theorem 7 (Universal Approximation). *For every $f \in \mathcal{S}(\Sigma^*)$ and $\varepsilon > 0$, there exists $r \in \mathcal{S}^{rat}_{\mathbb{R}_{\geq 0}}(\Sigma^*)$ such that $\|f - r\|_1 < \varepsilon$.*

Proof. Since f is a probability distribution over the countable set Σ^*, choose a finite $S \subseteq \Sigma^+$ such that $\sum_{w \in S} f(w) > 1 - \varepsilon/2$. Define r_S as the normalized restriction of f to S:

$$r_S(w) = \begin{cases} \dfrac{f(w)}{f(S)} & \text{if } w \in S, \\ 0 & \text{otherwise.} \end{cases}$$

Then r_S is a stochastic language with finite support and therefore $r_S \in \mathcal{S}^{rat}_{\mathbb{R}_{\geq 0}}(\Sigma^*)$. Moreover, $\|f - r_S\|_1 = 2(1 - f(S)) < \varepsilon$. □

Hence $\mathcal{S}^{rat}_{\mathbb{R}_{\geq 0}}(\Sigma^*)$ is dense in $\mathcal{S}(\Sigma)$ under ℓ_1 (and therefore also under ℓ_∞). Intuitively, the convex hull of Dirac distributions $\{\delta_w \mid w \in \Sigma^+\}$ suffices to approximate any stochastic language arbitrarily closely. This motivates a quantitative analogue of classical automata minimization.

Problem 2 (Minimal-State ε-Approximation). Given $r \in \mathcal{S}(\Sigma^*)$ and $\varepsilon > 0$, determine the minimal $k \in \mathbb{N}$ for which there exists a k-state weighted automaton $\mathcal{A}$ satisfying $\|r - [\![\mathcal{A}]\!]\|_1 \leq \varepsilon$.

When $\varepsilon = 0$ and r is rational, the minimal k coincides with the Hankel rank of r, computable via standard spectral methods (e.g. [14, 31, 21]). For $\varepsilon > 0$, Problem 2 becomes that of computing an *approximate Hankel rank*, a continuous relaxation of minimality that remains open in general.

Sampling from Stochastic Regular Expressions. Stochastic regular expressions (SREs) admit exact generative sampling procedures aligned with their semantics. Sampling proceeds recursively as follows:

(a) Atomic symbol: For $r = \delta_\sigma$, output σ.
(b) Convex combination: For $r = \alpha r_1 + (1 - \alpha) r_2$, sample $b \sim \text{Bernoulli}(\alpha)$ and recurse into r_b.
(c) Concatenation: For $r = r_1 \cdot r_2$, sample $w_1 \sim r_1$ and $w_2 \sim r_2$, and output $w_1 w_2$.
(d) Discounted star: For $r = r^*_\alpha$, draw $k \sim \text{Geom}(\alpha)$ and concatenate $w_1, \ldots, w_k$ sampled independently from r.

The expected length of a sample from r^*_α is $(1-\alpha)/\alpha$, hence finite for $\alpha > 0$. Each operator preserves independence and normalization, ensuring that the sampling process exactly realizes $[\![r]\!]$.

Learnability. The density result of Theorem 7 implies that stochastic languages can be approximated by finite-parameter models, suggesting learnability from data. Two settings naturally arise:

1. Given i.i.d. samples from an unknown $r \in \mathcal{S}(\Sigma^*)$, one can fit a weighted automaton by empirical estimation of Hankel moments, following the spectral learning paradigm for rational series [5, 21]. Sample complexity bounds follow from standard concentration inequalities for low-rank moment matrices.
2. If an oracle provides evaluation access to $[\![r]\!](w)$ for queries $w \in \Sigma^*$, the parameters of a linear CRA or weighted automaton can be recovered by solving a finite system of linear equations [2]. Under regularity assumptions, this system is solvable in polynomial time.

Distribution Testing. Finally, we consider the problem of distribution testing:

Problem 3 (Distribution Testing). Given $Q \in \mathcal{S}^{rat}_{\mathbb{R}_{\geq 0}}(\Sigma^*)$ and sample access to $P \in \mathcal{S}(\Sigma)$, design a randomized algorithm that, for tolerance $\varepsilon > 0$ and confidence $\delta \in (0, 1)$, satisfies:

$$\begin{aligned} \text{Accept,} \quad & \|P - Q\|_1 < \varepsilon, \\ \text{Reject,} \quad & \|P - Q\|_1 > \varepsilon, \end{aligned}$$

with probability at least $1 - \delta$.

This extends the classical problem on finite domains [6, 13]. A naive solution proceeds by a transforming it to the finite-support case by truncating the distribution tail. Let $\theta \in \mathbb{N}$ be the smallest length such that $\sum_{w \in \Sigma^{\leq \theta}} Q(w) > 1 - \varepsilon/3$. Construct the restricted distributions $P_{\leq \theta}$ and $Q_{\leq \theta}$ on $\Sigma^{\leq \theta}$ and apply any finite-domain tolerant tester (e.g. [13]) with thresholds $(\varepsilon/3, \varepsilon)$. This yields correctness guarantees identical to the finite setting, with sample complexity $\widetilde{\Theta}(|\Sigma|^{\frac{\theta+1}{2}}/\varepsilon^2)$.

7 Conclusion and Future Work

In this work, we advance the theory of stochastic languages through the lens of sub-stochastic weighted automata, establishing fundamental computational boundaries for quantitative automata models. Our main contributions are:

1. We show that determining whether a cost register automaton defines a stochastic language is undecidable in general, even for affine register updates (Theorem 1) and the fully polynomial fragment (Theorem 2). This holds for finite mass, non-negativity, and exact normalization.
2. By restricting to cost register automata over non-negative reals with affine updates, we recover decidability and efficient computation. For this fragment, expressively equivalent to weighted finite automata, the total mass is computable in polynomial time via spectral analysis (Theorem 4).

3. We establish that every stochastic weighted automaton admits a semantically equivalent *sub-stochastic* representation via *Perron-Frobenius normalization* (Theorem 5). This local characterization mirrors classical probabilistic automata, enabling convergence verification syntactically, without global spectral analysis.
4. We introduce *stochastic regular expressions* (SREs) as a compositional, probabilistic analogue of regular expressions and provide a Kleene-Schützenberger characterization for rational stochastic languages (Theorem 6).

Open Problems and Future Directions

Despite these advances, several fundamental questions remain open. We highlight three directions where sub-stochastic representations could play a decisive role:

1. *Decidability Beyond Linear Updates. (Problem 1)* Can syntactic conditions on non-linear cost register automata ensure stochasticity decidability? One can look at functions f that satisfy $\|f(x) - f(y)\| \leq L\|x - y\|$ with $L < 1$. Banach's fixed-point theorem guarantees convergence to a unique fixed point. For CRAs with contracting updates, we can explore if stochasticity is decidable and if spectral methods extend via linearization around fixed points? More generally, polynomial dynamical systems with well-understood asymptotics may admit decidable fragments. Investigating *semantic classes* of well-behaved CRAs could significantly expand the applicability of sub-stochastic methods. Approximate notions of stochasticity, e.g., ε-closeness, may suffice for practical purposes and connect to numerical optimization.
2. *Complexity of Minimal-State ε-Approximation (Problem 2).* Determining the minimal number of states to ε-approximate a stochastic language is a central open question. This connects to low-rank matrix approximation and non-negative factorization. Approximate minimal-state representations determine sample complexity in learning stochastic languages and illuminate the intrinsic complexity of probabilistic sequential models.
3. *Distribution Testing (Problem 3).* Efficient, instance-optimal testing algorithms are needed for stochastic languages, particularly for validating large language models. Some important questions are: Can we exploit local sub-stochastic properties of distributions for efficient testing? Can we use these methods to perform efficient black-box testing for language models? This problem bridges theory and practice, providing foundations for *trustworthy AI* and enabling rigorous, sample-efficient validation of probabilistic generative models.

In conclusion, our work establishes the connection between stochastic languages and their sub-stochastic representations. As probabilistic models become central to computing, formal methods to specify, analyse, and verify their distributional behaviour are increasingly urgent. Viewing stochastic languages through sub-stochastic models offers a principled foundation, laying the groundwork for a formal *theory of probabilistic computation.*

References

1. Alur, R., DAntoni, L., Deshmukh, J., Raghothaman, M., Yuan, Y.: Regular functions and cost register automata. In: 2013 28th Annual ACM/IEEE Symposium on Logic in Computer Science. pp. 13–22 (2013). https://doi.org/10.1109/LICS.2013.65
2. Angluin, D.: Learning regular sets from queries and counterexamples. Information and Computation **75**(2), 87–106 (1987). https://doi.org/https://doi.org/10.1016/0890-5401(87)90052-6
3. Baier, C., Größer, M., Bertrand, N.: Probabilistic ω-automata. Journal of the ACM **59**(1), 1–52 (2012)
4. Baier, C., Katoen, J.P.: Principles of Model Checking. MIT Press (2008)
5. Balle, B., Mohri, M.: Learning weighted automata. In: Maletti, A. (ed.) Algebraic Informatics. Lecture Notes in Computer Science, vol. 9270, pp. 1–21. Springer International Publishing, Cham (2015). https://doi.org/10.1007/978-3-319-23318-2_1
6. Batu, T., Fortnow, L., Rubinfeld, R., Smith, W.D., White, P.: Testing closeness of discrete distributions. Journal of the ACM **60**(1), 4:1–4:25 (Feb 2013). https://doi.org/10.1145/2432622.2432626, https://doi.org/10.1145/2432622.2432626
7. Behrmann, G., David, A., Larsen, K.G.: A tutorial on uppaal. In: Bernardo, M., Corradini, F. (eds.) Formal Methods for the Design of Real-Time Systems. SFM-RT 2004. Lecture Notes in Computer Science, vol. 3185, pp. 200–237. Springer (2004)
8. Benalioua, Y.I., Lhote, N., Reynier, P.A.: Minimizing cost register automata over a field. In: Proceedings of the 49th International Symposium on Mathematical Foundations of Computer Science (MFCS 2024). LIPIcs, vol. 306, pp. 23:1–23:19. Schloss Dagstuhl–Leibniz-Zentrum für Informatik (2024). https://doi.org/10.4230/LIPIcs.MFCS.2024.23
9. Bollig, B., Gastin, P., Monmege, B., Zeitoun, M.: A probabilistic kleene theorem? In: Proceedings of the 13th International Conference on Implementation and Application of Automata (CIAA 2012). Lecture Notes in Computer Science, Springer (2012), https://hal.science/hal-00713660/document, lSV, ENS Cachan, CNRS & Inria, and LaBRI, Univ. Bordeaux & CNRS
10. Bollig, B., Zeitoun, M.: Weighted automata. https://www.lsv.fr/ bollig/ (2015), lecture notes from the MPRI course on weighted automata, ENS Cachan
11. Bournez, O., Campagnolo, M.L., Graça, D.S., Hainry, E.: The general purpose analog computer and computable analysis are equivalent. Theoretical Computer Science **374**(1-3), 80–97 (2007). https://doi.org/10.1016/j.tcs.2006.12.035
12. Brown, P.F., Pietra, S.A.D., Pietra, V.J.D., Mercer, R.L.: The mathematics of statistical machine translation: Parameter estimation. Computational Linguistics **19**(2), 263–311 (1993)
13. Canonne, C.L., Jain, A., Kamath, G., Li, J.: The price of tolerance in distribution testing. In: COLT (2022), https://arxiv.org/abs/2106.13414
14. Carlyle, J., Paz, A.: Realizations by stochastic finite automata. Journal of Computer and System Sciences **5**(1), 26–40 (1971). https://doi.org/https://doi.org/10.1016/S0022-0000(71)80005-3, https://www.sciencedirect.com/science/article/pii/S0022000071800053
15. Chatterjee, K., Doyen, L., Henzinger, T.A.: Probabilistic weighted automata. In: Proceedings of CONCUR 2009. LNCS, vol. 5710, pp. 244–258. Springer (2009)

16. Chomsky, N.: Three models for the description of language. IRE Transactions on Information Theory **2**, 113–124 (1956)
17. Clark, A.: Learning context-free grammars with the syntactic concept lattice. In: International Colloquium on Grammatical Inference (ICGI 2010). Lecture Notes in Artificial Intelligence, vol. 6339, pp. 38–51. Springer (2010)
18. Cohn, T., Blunsom, P.: Spectral learning of sequential systems. In: Advances in Neural Information Processing Systems (NeurIPS 2009). pp. 1–8 (2009)
19. Cover, T.M., Thomas, J.A.: Elements of Information Theory. Wiley-Interscience, 2nd edn. (2006)
20. Czerwiński, W., Lefaucheux, E., Mazowiecki, F., Purser, D., Whiteland, M.: The boundedness and zero isolation problems for weighted automata over nonnegative rationals. In: Proceedings of the 37th Annual ACM/IEEE Symposium on Logic in Computer Science. LICS '22, Association for Computing Machinery, New York, NY, USA (2022). https://doi.org/10.1145/3531130.3533336, https://doi.org/10.1145/3531130.3533336
21. Denis, F., Esposito, Y., Habrard, A.: Learning rational stochastic languages. In: Lugosi, G., Simon, H.U. (eds.) Learning Theory. pp. 274–288. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)
22. Denis, F., Esposito, Y.: On rational stochastic languages. Fundamenta Informaticae **62**(3-4), 197–213 (2004)
23. Droste, M., Gastin, P.: Weighted automata and weighted logics. Theoretical Computer Science **380**(1-2), 69–86 (2007). https://doi.org/10.1016/j.tcs.2007.02.055
24. Droste, M., Kuich, W., Vogler, H. (eds.): Handbook of Weighted Automata. Springer (2009). https://doi.org/10.1007/978-3-642-01492-1
25. Durbin, R., Eddy, S.R., Krogh, A., Mitchison, G.: Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids. Cambridge University Press, Cambridge, UK (1998)
26. Hainry, E.: Decidability and Undecidability in Dynamical Systems and Hybrid Systems. Ph.d. thesis, École Normale Supérieure de Lyon (2006), https://tel.archives-ouvertes.fr/tel-00114112
27. Horn, R.A., Johnson, C.R.: Matrix Analysis. Cambridge University Press, Cambridge (2012)
28. Jelinek, F.: Statistical Methods for Speech Recognition. MIT Press, Cambridge, MA (1998)
29. Paz, A.: Introduction to Probabilistic Automata. Academic Press, New York (1971)
30. Rabin, M.O.: Probabilistic automata. Information and Control **6**(3), 230–245 (1963)
31. Schützenberger, M.: On the definition of a family of automata. Information and Control **4**(2), 245–270 (1961). https://doi.org/https://doi.org/10.1016/S0019-9958(61)80020-X
32. Segala, R., Lynch, N.A.: Probabilistic automata. Nordic Journal of Computing **2**(2), 250–273 (1995)