Approximating Probabilistic Automata by Regular Languages

Rohit Chadha[1]
University of Missouri
Columbia, USA
chadhar@missouri.edu

A. Prasad Sistla[2]
University of Illinois, Chicago
Chicago, USA
sistla@uic.edu

Mahesh Viswanathan[3]
University of Illinois, Urbana-Champaign
Urbana, USA
vmahesh@illinois.edu

Abstract

A probabilistic finite automaton (PFA) $\mathcal{A}$ is said to be regular-approximable with respect to (x, y), if there is a regular language that contains all words accepted by $\mathcal{A}$ with probability at least $x+y$, but does not contain any word accepted with probability at most x. We show that the problem of determining if a PFA $\mathcal{A}$ is regular-approximable with respect to (x, y) is not recursively enumerable. We then show that many tractable sub-classes of PFAs identified in the literature – hierarchical PFAs, polynomially ambiguous PFAs, and eventually weakly ergodic PFAs – are regular-approximable with respect to all (x, y). Establishing the regular-approximability of a PFA has the nice consequence that its value can be effectively approximated, and the emptiness problem can be decided under the assumption of isolation.

2012 ACM Subject Classification Theory of computation → Probabilistic computation

Keywords and phrases Probabilistic Finite Automata, Regular Languages, Ambiguity

Digital Object Identifier 10.4230/LIPIcs.CSL.2018.14

1 Introduction

Probabilistic finite automata (PFA), introduced by Rabin [26], are finite state machines that read symbols from an input string and whose state transitions are determined by the input symbol being read and the result of a coin toss. For an input string w, the probability of accepting w is the measure of all runs of the automaton on w that end in an accepting state. Given a threshold x, the language recognized by a PFA is the collection of all words w whose probability of acceptance is at least x. Probabilistic finite automata serve as convenient models of open stochastic systems. Despite their simplicity, PFAs are a surprisingly powerful model of computation and typical decision problems of PFAs are undecidable. For example, the classical decision problem that arises when verifying a design described by a PFA against regular specifications, namely emptiness, is undecidable [11].

[1] NSF CNS 1314338 and NSF CNS 1553548
[2] NSF CNS 1314485 and NSF CCF 1564296
[3] NSF CSR 1422798 and NSF CPS 1329991

27th EACSL Annual Conference on Computer Science Logic (CSL 2018).
Editors: Dan Ghica and Achim Jung; Article No. 14; pp. 14:1–14:23
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

The reason for the computational hardness of problems involving PFAs is because they can "simulate" powerful computational models like Turing machines. The question we ask is if, despite this evidence of expressive power, languages recognized by PFAs can be "approximated" by regular languages, in a sense that we will make precise later in this introduction. If PFAs can be approximated by regular languages, it opens up the possibility of solving some of these decision problems partially. For example, if we want to verify that a stochastic open system modeled by a PFA meets a regular specification, we could approximate the PFA language by a regular language, and then check containment/emptiness. This approach would be similar to the effective role finite state abstractions have played in verifying real world designs.

So what type of regular approximations are we talking about? For a PFA $\mathcal{A}$, let $\mathsf{L}_{\geq x}(\mathcal{A})$ and $\mathsf{L}_{\leq x}(\mathcal{A})$ be the sets of strings accepted with probability $\geq x$ and $\leq x$, respectively. We say that $\mathcal{A}$ is *regular-approximable with respect to* (x, y) if there is a regular language L that separates $\mathsf{L}_{\geq x+y}(\mathcal{A})$ and $\mathsf{L}_{\leq x}(\mathcal{A})$, i.e., $\mathsf{L}_{\geq x+y}(\mathcal{A}) \subseteq L$ and $L \cap \mathsf{L}_{\leq x}(\mathcal{A}) = \emptyset$ (i.e., $L \subseteq \mathsf{L}_{>x}(\mathcal{A})$). Thus, L is a "over-approximation" of $\mathsf{L}_{\geq x+y}(\mathcal{A})$ and an "under-approximation" of $\mathsf{L}_{>x}(\mathcal{A})$. Such a notion of separability has been previously studied in the context of PFAs [24]. Separability using regular languages have played a significant role in understanding the expressive power of formal languages and coming up with decision procedures [12, 25].

First, even if $\mathsf{L}_{\geq x+y}(\mathcal{A})$ and $\mathsf{L}_{\leq x}(\mathcal{A})$ are not regular, $\mathcal{A}$ maybe regular-approximable with respect to (x, y) (see Example 7). On the other hand, there are PFAs $\mathcal{A}$ and (x, y) such that $\mathcal{A}$ is not regular-approximable with respect to (x, y) (see [24] and Theorem 8). So how difficult is it to check regular-approximability? We show that the problem of determining if a PFA $\mathcal{A}$ is regular-approximable with respect to (x, y) is not recursively enumerable (Theorem 9). Our proof relies on showing that a closely related problem of determining if a PFA $\mathcal{A}$ is regular-approximable with respect to *some* (x, y) is $\mathbf{\Sigma^0_2}$-hard; $\mathbf{\Sigma^0_2}$ is the second level of the arithmetic hierarchy.

Given that determining if a PFA $\mathcal{A}$ is regular-approximable with respect to (x, y) is undecidable, we try to identify sufficient conditions that guarantee the regular-approximability of PFAs in a very strong sense. In particular, we identify conditions under which a PFA is guaranteed to be regular-approximable with respect to *every* pair (x, y). Further, we'd like to identify when the regular language approximating the PFA can be effectively constructed from $\mathcal{A}$ and (x, y). PFAs that satisfy such strong properties are amenable to automated analysis. We show that problems that are undecidable (or open) for general PFAs, become decidable in such situations. We give examples of two such problems. The first is the value problem for PFAs, where the goal is to compute the supremum of the acceptance probabilities of all input words. When a PFA $\mathcal{A}$ represents the product of an open probabilistic system and an incorrectness property given as deterministic automaton on the system executions, then value of $\mathcal{A}$ gives a tight upper bound on the probability of incorrectness of the system on all input sequences. Decision versions of the value problem are known to be $\mathbf{\Sigma^0_2}$-complete. The second problem is checking emptiness under isolation. A threshold x is said to be isolated for PFA $\mathcal{A}$ with a degree of isolation ϵ if the acceptance probability of every word is ϵ-bounded away from x. A classical result is that when x is isolated, the language $\mathsf{L}_{\geq x}(\mathcal{A})$ is regular [26]. The emptiness under isolation problem, is to determine if the language $\mathsf{L}_{\geq x}(\mathcal{A})$ is empty, under the promise that x is an isolated cut-point for $\mathcal{A}$ (but no degree of isolation is given). The decidability of this is a long standing open problem. We prove that for PFAs that are effectively regular-approximable (that is regular separator L can be constructed for every (x, y)), the value problem can be approximated with arbitrary precision (Theorem 11) and the emptiness under isolation is decidable (Corollary 12).

Our semantic condition that identifies when a PFA is regular-approximable is as follows. A *leaky* transition is a transition whose probability is less than 1. A PFA $\mathcal{A}$ is said to be *leak monotonic* if for every ϵ, there is a number N_ϵ such that, for any input u, the measure of all accepting runs ρ on u that have at least N_ϵ leaks is $< \epsilon$. In other words, runs with many leaky transitions contribute very little to the acceptance probability of a word. We prove that leak monotonic PFAs are regular-approximable with respect to every (x, y) (Corollary 20). If a leak monotonic PFA in addition has the property that N_ϵ can be computed from ϵ, then one can show that the regular separator of $\mathsf{L}_{\geq x+y}(\mathcal{A})$ and $\mathsf{L}_{\leq x}(\mathcal{A})$ can also be effectively constructed (Corollary 20). The deterministic automaton B that recognizes the regular separator has the property that its computation on any input u can be used to approximately compute $\mathcal{A}$'s acceptance probability as follows – one can associate a function from states of B to $[0, 1]$ such that the label of the state reached on reading u is an approximation of the acceptance probability of u.

Our last set of results in the paper show that many of the tractable sub-classes of PFAs discovered, enjoy the nice decidability properties because of regular-approximability. Hierarchical PFAs [9] are those that obey the restriction that states can be partitioned into a hierarchy of ranks, and transitions from a state only go to states of the same or higher rank (for a precise definition, see paragraph before Theorem 26). Another class of PFAs are those with polynomial ambiguity [16]. These are PFAs with the property that on any input u, the number of accepting runs on u (not its probability) is bounded by a polynomial function of the input length $|u|$. Both these sub-classes of PFAs are effectively leak monotonic, and hence effectively regular-approximable. Thus their value can be effectively approximated, and the emptiness problem is decidable under the promise of isolation for these classes. These results for hierarchical PFAs subsume [8], and are new for polynomial ambiguous PFAs. Our results also show the existence of a large class of non-trivial PFAs that exhibit exponential ambiguity but are nonetheless still leak monotonic and hence regular approximable; Theorem 21 gives a method of obtaining such PFAs (Figure 2a shows such a PFA $\mathcal{A}_z$). In this paper, we also show that the emptiness problem is undecidable for linearly ambiguous PFAs, thus resolving an open problem posed in [16], and tightening the decidability results presented in [16]. Another tractable class of PFAs is that of eventually weakly ergodic PFAs [10]. We show that though eventually weakly ergodic PFAs are not leak monotonic, they are effectively regular-approximable. Once again, as a consequence, the decidability results proved in [10] follow from observations made here.

The rest of the paper is organized as follows. We conclude this section with a discussion of closely related work. Basic definitions and notations are introduced in Section 2. Regular-approximability is defined and the undecidability of deciding of a PFA regular-approximable with respect to (x, y) is proved in Section 3. Next, in Section 4, we give the semantic definition of leak monotonicity, its relation to regular-approximability, and its application to computing the value and deciding the emptiness problem. Section 5 presents results establishing the regular-approximability of hierarchical PFAs and polynomially ambiguous PFAs, and Section 6 shows that eventually weakly ergodic PFAs are also regular-approximable. Conclusions are presented in Section 7. All missing proofs can be found in the Appendix.

Related Work

The problem of checking whether the language recognized by a PFA is regular known to be undecidable [17, 4]. As mentioned above, regular-approximability of PFAs was first studied in [24], where Paz gave an alternate, semantic characterization of regular-approximable PFAs. We are not aware of any further work on this topic in the context PFAs, though separation using regular languages has been used to obtain expressiveness and decidability results [12, 25]. ♮-acyclic automata and their generalization *leak-tight automata* [15, 14], are special classes of PFAs for which the value 1 problem is decidable. The classes of leaktight and leak monotonic automata (introduced in this paper) are incomparable – PFA $\mathcal{A}_3$ in Figure 1c on page 7 is leaktight but not leak monotonic. On the other hand, consider any PFA $\mathcal{A}$ that is not leaktight, and let $\mathcal{B}$ be PFA that is identical to $\mathcal{A}$, but with an empty set of final states. $\mathcal{B}$ is still not leaktight, but $\mathcal{B}$ is trivially leak monotonic (and hence regular approximable). The relationship between ♮-acyclic automata/leaktight automata and regular-approximable automata still needs further investigation. In particular, it is open whether ♮-acyclic and leaktight automata are a subclass of regular approximable automata. Bounding the ambiguity of PFAs as been a way to identify subclasses of PFAs for which certain computational problems become decidable [6, 8, 16]. However, all these results only pertain to automata with *constant* ambiguity and their subclasses. In this paper, we obtain positive results for more general classes of PFAs that go beyond polynomially ambiguous automata. The undecidability of the emptiness problem for linearly ambiguous automata was also independently observed in [13].

2 Preliminaries

We assume that the reader is familiar with probability distributions, stochastic matrices, finite-state automata, and regular languages. The set of natural numbers will be denoted by $\mathbb{N}$, the closed unit interval by $[0,1]$ and the open unit interval by $(0,1)$. The power-set of a set X will be denoted by 2^X. For any function $f\colon X \to Y$ and $Y_1 \subseteq Y$, $f^{-1}(Y_1)$ is the set $\{x \in X \mid f(x) \in Y_1\}$. If X is a finite set $|X|$ will denote its cardinality. We assume that the reader is familiar with the arithmetic hierarchy.

Sequences. Given a finite sequence $s = s_0 s_1 \dots$ over S, $|s|$ will denote the length of s and $s[i]$ will denote the ith element s_i of the sequence with $s[0]$ being the first. We will use λ to denote the (unique) empty string/sequence. For natural numbers i, j, $i \le j < |s|$, $s[i:j]$ is the sequence $s_i \dots s_j$. As usual S^* will denote the set of all finite sequences/strings/words over S, S^+ will denote the set of all finite non-empty sequences/strings/words over S.

Given $u \in S^*$ and $v \in S^*$, uv is the sequence obtained by concatenating the two sequences in order. Given $\mathsf{L}_1 \subseteq S^*$ and $\mathsf{L}_2 \subseteq S^*$, the set $\mathsf{L}_1\mathsf{L}_2$ is defined to be $\{uv \mid u \in \mathsf{L}_1 \text{ and } v \in \mathsf{L}_2\}$.

Ambiguity and Pumping Lemma

Let $\mathcal{A}$ be a nondeterministic automaton recognizing a regular language over alphabet Σ. The degree of ambiguity [22, 21, 27] of $\mathcal{A}$ on input word $u \in \Sigma^*$, denoted $d_{\mathcal{A}}(u)$, is the number of accepting runs of $\mathcal{A}$ on u. It is shown in [28, 20] that the degree of ambiguity of a NFA $\mathcal{A}$ is one the following.

1. $\mathcal{A}$ is *finitely ambiguous* if there is a constant k such that $d_{\mathcal{A}}(u) \le k$ for all input words $u \in \Sigma^*$.
2. $\mathcal{A}$ is *polynomially ambiguous* if there is a non-constant polynomial $p\colon \mathbb{N} \to \mathbb{N}$ such that $d_{\mathcal{A}}(u) \le p(|u|)$ for all all input words $u \in \Sigma^*$; if p has degree 1 or 2 then $\mathcal{A}$ is said to be linearly or quadratically ambiguous, respectively.
3. $\mathcal{A}$ is *exponentially ambiguous* if for every polynomial $p : \mathbb{N} \to \mathbb{N}$, there is a word $u \in \Sigma^*$ such that $d_{\mathcal{A}}(u) > p(|u|)$.

A *trim* NFA is an automaton that does not have any silent edges. The following can be concluded from the results of [28]:

▶ **Lemma 1.** *The problems of deciding whether a trim $\mathcal{A}$ is finitely ambiguous, whether $\mathcal{A}$ is polynomially ambiguous and whether $\mathcal{A}$ is exponentially ambiguous are decidable in polynomial time. If $\mathcal{A}$ is polynomially ambiguous then a constant C and a constant ℓ can be computed in polynomial time such that $d_{\mathcal{A}}(u) \leq C|u|^{\ell}$ for all input words $u \in \Sigma^*$.*

The following lemma, used in parts of the paper, states a simple property of regular languages and is easily proved along the same lines as the standard pumping lemma.

▶ **Lemma 2.** *For a regular language $L \subseteq \Sigma^*$, where $|\Sigma| \geq 2$, there exists an integer constant $N > 0$ such that the following property holds for each $a \in \Sigma$ and each $k \geq 1$: if there exists a string of the form $u_1 a u_2 a ... u_k a \in L$, where each $u_i \in (\Sigma \setminus \{a\})^*$, for $1 \leq i \leq k$, then there exists such a string such that $|u_i| \leq N$, for each i, $1 \leq i \leq k$.*

Probabilistic automaton (PFA)

Informally, a PFA is like a finite-state deterministic automaton except that the transition function from a state on a given input is described as a probability distribution which determines the probability of the next state.

▶ **Definition 3.** A *finite-state probabilistic automaton (*PFA*)* [26, 24] on finite strings over a finite alphabet Σ is a tuple $\mathcal{A} = (Q, q_s, \delta, Q_f)$ where Q is a finite set of *states*, $q_s \in Q$ is the *initial state*, $\delta : Q \times \Sigma \times Q \to [0, 1]$ is the *transition relation* such that for all $q \in Q$ and $a \in \Sigma$, $\delta(q, a, q')$ is a rational number and $\sum_{q' \in Q} \delta(q, a, q') = 1$, and $Q_f \subseteq Q$ is the set of accepting/final states. We say that the PFA $\mathcal{A}$ is a *deterministic* automaton if, for every $q \in Q, a \in \Sigma$ there exists exactly one $q' \in Q$ such that $\delta(q, a, q') = 1$.

▶ Notation. The transition function δ of PFA $\mathcal{A}$ on input a can be seen as a square matrix δ_a of order $|Q|$ with the rows labeled by "current" state, columns labeled by "next state" and the entry $\delta_a(q, q')$ equal to $\delta(q, a, q')$. Given a word $u = a_0 a_1 \ldots a_n \in \Sigma^+$, δ_u is the matrix product $\delta_{a_0} \delta_{a_1} \ldots \delta_{a_n}$. For the empty word $\lambda \in \Sigma^*$ we take δ_λ to be the identity matrix. Finally for any $Q_0 \subseteq Q$, we say that $\delta_u(q, Q_0) = \sum_{q' \in Q_0} \delta_u(q, q')$. Given a state $q \in Q$ and a word $u \in \Sigma^+$, $\mathsf{post}(q, u) = \{q' \mid \delta_u(q, q') > 0\}$. For a set $C \subseteq Q$, let $\mathsf{post}(C, u) = \cup_{q \in C} \mathsf{post}(q, u)$.

Intuitively, the PFA starts in the initial state q_s and if after reading $a_0, a_1 \ldots, a_i$ it is in state q, then the PFA moves to state q' with probability $\delta_{a_{i+1}}(q, q')$ on symbol a_{i+1}. A *run* of the PFA $\mathcal{A}$ starting in a state $q \in Q$ on an input $u \in \Sigma^*$ is a sequence $\rho \in Q^*$ such that $|\rho| = 1 + |u|, \rho[0] = q$ and for each $i \geq 0$, $\delta_{u[i]}(\rho[i], \rho[i+1]) > 0$. The probability measure of such a run ρ on u is defined to be the value $\prod_{0 \leq i < |\rho|} \delta_{u[i]}(\rho[i], \rho[i+1])$. We say that the run ρ is an *accepting* run if $\rho[|\rho|] \in Q_f$, i.e., it ends in an accepting state. Unless otherwise stated, a *run* for us will mean a run starting in the initial state q_s. The probability of acceptance of $u \in \Sigma^*$ by the PFA $\mathcal{A}$, denoted by $\mathsf{P}_{\mathcal{A}}(u)$, is defined to be the sum of probability measures of all accepting runs of $\mathcal{A}$ on u. Note that $\mathsf{P}_{\mathcal{A}}(u) = \delta_u(q_s, Q_f)$.

PFA languages

Given a PFA $\mathcal{A}$, a rational threshold $x \in [0, 1]$ and $\lozenge \in \{<, \leq, =, \geq, >\}$, the language $\mathsf{L}_{\lozenge x}(\mathcal{A}) = \{u \in \Sigma^* \mid \mathsf{P}_{\mathcal{A}}(u) \lozenge x\}$ is the set of finite words accepted by $\mathcal{A}$ with probability $\lozenge x$. If $\mathcal{A}$ is a deterministic automaton then we let $\mathsf{L}(\mathcal{A})$ denote the language $\mathsf{L}_{\geq 1}(\mathcal{A})$. In general, the language $\mathsf{L}_{\lozenge x}(\mathcal{A})$ for a PFA $\mathcal{A}$, threshold x, and $\lozenge \in \{<, \leq, =, \geq, >\}$, may be non-regular. However, when x is an extremal threshold ($x \in \{0, 1\}$), it is regular.

▶ **Proposition 4.** *For any* PFA $\mathcal{A}$*, the languages* $\mathsf{L}_{\lozenge x}(\mathcal{A})$ *is effectively regular for* $x \in \{0, 1\}$ *and* $\lozenge \in \{<, \leq, =, \geq, >\}$.

Given a PFA $\mathcal{A}$ and rational threshold x, the problem of checking whether $\mathsf{L}_{>x}(\mathcal{A}) = \emptyset$ is known to be **co-R.E.**-complete [24, 11].

Isolated cut-points

For a PFA $\mathcal{A}$, a rational threshold $x \in [0,1]$ is said to be an *isolated cut-point* of $\mathcal{A}$ if there is an $\epsilon > 0$ such that for each word $u \in \Sigma^*$, $|\mathsf{P}_{\mathcal{A}}(u) - x| > \epsilon$. If such an ϵ exists, then ϵ is said to be *a degree of isolation.* An important observation about PFAs with isolated cut-points, is that their language is regular; however, the deterministic finite automaton recognizing this language is known to be constructible only given a degree of isolation.

▶ **Theorem 5** (Rabin [26]). *For any* PFA $\mathcal{A}$ *with an isolated cut-point* x*, the languages* $\mathsf{L}_{\lozenge x}(\mathcal{A})$ *are regular, where* $\lozenge \in \{<, \leq, =, \geq, >\}$.

The *isolation decision problem* is the problem of deciding for a given PFA $\mathcal{A}$ and a rational $x \in [0,1]$ whether x is an isolated cut-point of $\mathcal{A}$. The isolation decision problem is known to be undecidable [3], even when x is 0 or 1 [18]. The problem is known to be $\mathbf{\Sigma_2^0}$-complete [10].

The value problem. For a PFA $\mathcal{A}$, let $\mathsf{value}(\mathcal{A})$ denote the least upper bound of the set $\{\mathsf{P}_{\mathcal{A}}(u) \mid u \in \Sigma^*\}$. The *value computation problem* for a PFA is the problem of computing $\mathsf{value}(\mathcal{A})$ for a given $\mathcal{A}$. The *value decision problem* is the problem of deciding for a given PFA $\mathcal{A}$ and a rational threshold $x \in [0,1]$ whether $\mathsf{value}(\mathcal{A}) = x$. The value decision problem is known to be undecidable [3, 18] and known to be $\mathbf{\Pi_2^0}$-complete [10] even when x is taken to be 1 [10].

3 Approximability and Value problem

3.1 Regular Approximability.

The problem of approximating a PFA by a regular language was first discussed by Paz [24]. We will say that PFA $\mathcal{A}$ can be approximated by a regular language L at a threshold x with precision y if L separates the languages $\mathsf{L}_{\geq x+y}(\mathcal{A})$ and $\mathsf{L}_{\leq x}(\mathcal{A})$. Formally,

▶ **Definition 6.** Given $x, y \in [0,1]$ such that $y > 0$, a PFA $\mathcal{A} = (Q, q_s, \delta, Q_f)$ over Σ is said to be *regular-approximable* with respect to the pair (x, y) if there is a regular language L such that $\mathsf{L}_{\geq x+y}(\mathcal{A}) \subseteq L \subseteq \mathsf{L}_{>x}(\mathcal{A})$.

It is easy to see that $\mathcal{A}$ is regular-approximable with respect to (x, y) if either $\mathsf{L}_{>x}(\mathcal{A})$ or $\mathsf{L}_{\geq x+y}(\mathcal{A})$ is a regular set. We say that the pair (x, y), $x, y \in [0,1]$, is a *trivial* pair if either $x = 0$ or $x + y \geq 1$. It is seen that every PFA is regular-approximable with respect to every trivial pair thanks to Proposition 4.

▶ **Example 7.** Consider the PFA $\mathcal{A}_1$, shown in Figure 1a. It has been shown in [7] that both $\mathsf{L}_{>\frac{1}{2}}(\mathcal{A}_1)$ and $\mathsf{L}_{\geq\frac{1}{2}}(\mathcal{A}_1)$ are non-regular. Further, given this observation, we can also conclude that $\mathsf{L}_{\geq\frac{3}{4}}(\mathcal{A}_1)$ is non-regular. This is because $\mathsf{L}_{\geq\frac{1}{2}}(\mathcal{A}_1) = \mathbf{1}\{\mathbf{0},\mathbf{1}\}^* \cup \mathbf{0}\mathsf{L}_{\geq\frac{3}{4}}(\mathcal{A}_1)$. Inspite of this, we can show that a regular language can separate $\mathsf{L}_{\geq\frac{3}{4}}(\mathcal{A}_1)$ and $\mathsf{L}_{\leq\frac{1}{2}}(\mathcal{A}_1)$, i.e., $\mathcal{A}_1$ is regular-approximable with respect to the pair $(\frac{1}{2}, \frac{1}{4})$. Observe that $\mathsf{L}_{\geq\frac{2}{3}}(\mathcal{A}_1) = \mathbf{1}\{\mathbf{0},\mathbf{1}\}^*$ is a regular set. Since $\mathsf{L}_{\geq\frac{3}{4}}(\mathcal{A}_1) \subseteq \mathsf{L}_{\geq\frac{2}{3}}(\mathcal{A}_1) \subseteq \mathsf{L}_{>\frac{1}{2}}(\mathcal{A}_1)$, we can conclude that $\mathcal{A}_1$ is regular-approximable with respect to the pair $(\frac{1}{2}, \frac{1}{4})$. In fact, as we will show later, $\mathcal{A}_1$ is regular-approximable with respect to every pair (x, y) where $y > 0$.

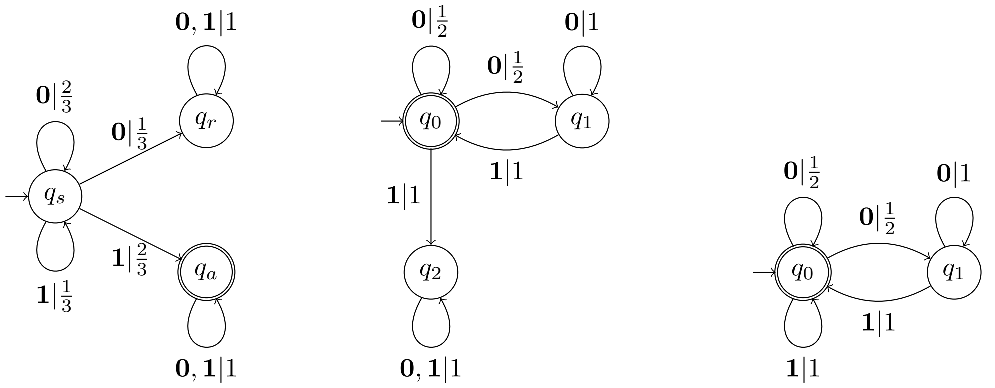

Figure 1 On the left (a) is PFA $\mathcal{A}_1$, in the middle (b) is PFA $\mathcal{A}_2$, and on the right (c) is PFA $\mathcal{A}_3$. In these pictures, for states q and q' and input letter a, if $\delta(q, a, q') > 0$ then we label the edge from q to q' by $a|\delta(q, a, q')$. The initial state is indicated by a dangling $\rightarrow$ and the final state by two concentric circles.

While $\mathcal{A}_1$ is an example of a PFA that is regular-approximable with respect to every pair (x, y) such that $y > 0$, the following theorem shows the existence of a PFA that is not regular-approximable with respect to any non-trivial pair.

▶ **Theorem 8.** *There exists a* PFA $\mathcal{A}$ *that is not regular-approximable with respect to any pair* (x, y) *such that* $x, y > 0$ *and* $x + y < 1$.

Proof. We prove the theorem by construction. Consider the PFA $\mathcal{A}_2$ over the input alphabet $\Sigma = \{\mathbf{0}, \mathbf{1}\}$, shown in Figure 1b. This automaton was used in [1] to show that the language recognized by a Probabilistic Büchi automaton (PBA) with threshold 0 can be nonregular.

We make the following observations, which are easily seen. Every word starting with $\mathbf{1}$ or that contains two consecutive $\mathbf{1}$s is accepted by $\mathcal{A}_2$ with probability zero. For every $k > 0$ and every $z, 0 < z < 1$, there is a word in $(\mathbf{0}^*\mathbf{1})^k$ that is accepted with probability $\geq z$.

Consider any pair (x, y) such that $x, y > 0$ and $x + y < 1$. We show that $\mathcal{A}_2$ is not regular-approximable with respect to (x, y), by contradiction. Assume for contradiction, that there is a regular language L such that $\mathsf{L}_{\geq x+y}(\mathcal{A}_2) \subseteq L \subseteq \mathsf{L}_{>x}(\mathcal{A}_2)$. Since L is a regular language, let N be the constant satisfying Lemma 2. Now, let $k \in \mathbb{N}$ be any integer such that $(1 - \frac{1}{2^N})^k \leq x$. Such a k exists since $x > 0$. From our earlier observation, we see that there exists a string $u \in (\mathbf{0}^*\mathbf{1})^k$ that is in $\mathsf{L}_{\geq x+y}(\mathcal{A}_2)$. Clearly, $u \in L$. Now, from Lemma 2, we see that there exists a string $v = \mathbf{0}^{n_1}\mathbf{1}\mathbf{0}^{n_2}\mathbf{1}\cdots\mathbf{0}^{n_k}\mathbf{1}$ where $n_i \leq N$, for $1 \leq i \leq k$ such that $v \in L$. Word v is accepted by $\mathcal{A}_2$, with probability $\prod_{1\leq i\leq k}(1 - \frac{1}{2^{n_i}})$. Since each $n_i \leq N$, we have $(1 - \frac{1}{2^{n_i}}) \leq (1 - \frac{1}{2^N})$. From this we see that the probability of acceptance of v by $\mathcal{A}_2$ is $\leq (1 - \frac{1}{2^N})^k \leq x$. Hence $v \notin \mathsf{L}_{>x}(\mathcal{A}_2)$ which contradicts our assumption that $L \subseteq \mathsf{L}_{>x}(\mathcal{A}_2)$. ◀

The following theorem shows that the problem of checking if a given PFA $\mathcal{A}$ is regular-approximable with respect to a given pair (x, y) is undecidable.

▶ **Theorem 9.** *Given a* PFA $\mathcal{A}$ *and rational values* $x, y \in [0, 1]$*, the problem of checking if* $\mathcal{A}$ *is approximable with respect to* (x, y)*, is undecidable. Formally the language* Approx $= \{(\mathcal{A}, x, y) \,|\, x, y \in [0, 1], \mathcal{A}$ *is a* PFA *that is regular-approximable w.r.t.* $(x, y)\}$ *is undecidable.*

3.2 Value Problem and Emptiness under isolation

PFAs that are effectively regular-approximable for every pair (x, y) enjoy nice properties.

▶ **Definition 10.** We say that $\mathcal{A}$ is *regular-approximable* if it is regular-approximable with respect to every pair (x, y) such that $x, y \in [0, 1]$ and $y > 0$. We further say that $\mathcal{A}$ is *effectively* regular-approximable if there is a procedure that, given x and y terminates and outputs a deterministic automaton that accepts a language L where $\mathsf{L}_{\geq x+y}(\mathcal{A}) \subseteq L \subseteq \mathsf{L}_{>x}(\mathcal{A})$. A class $\mathcal{C}$ of regular-approximable PFAs is said to be *effectively* regular-approximable if there is a procedure that, given $\mathcal{A} \in \mathcal{C}$, x and y terminates and outputs a deterministic automaton that accepts a language L where $\mathsf{L}_{\geq x+y}(\mathcal{A}) \subseteq L \subseteq \mathsf{L}_{>x}(\mathcal{A})$.

We shall establish later that the class of hierarchical probabilistic automata (HPAs) is effectively regular-approximable (See Theorem 26). It has been shown in [5, 8, 2] that the emptiness problem and the value decision problem continues to be undecidable if we restrict our attention to HPAs. Thus, there is no algorithm that given an effectively regular-approximable PFA $\mathcal{A}$ computes its value. Nevertheless, we now show that if $\mathcal{A}$ is effectively regular-approximable then its value can be computed to a given precision.

▶ **Theorem 11.** *There is a procedure* ComputeVal *that given an effectively regular-approximable* PFA $\mathcal{A}$ *and* $\epsilon > 0$ *terminates and returns an interval* $[z_1, z_2]$ *such that* $\mathsf{value}(\mathcal{A}) \in [z_1, z_2]$ *and* $z_2 - z_1 \leq \epsilon$.

Proof. ComputeVal works as follows. Initially, it checks if there is u such that $\mathsf{P}_{\mathcal{A}}(u) = 1$ or if for every u, $\mathsf{P}_{\mathcal{A}}(u) = 0$. If either of these conditions hold then it returns the corresponding value as $\mathsf{value}(\mathcal{A})$. Observe that these conditions can be checked thanks to Proposition 4. If neither of these conditions holds, it acts as follows. It maintains two variables z_1, z_2 such that $0 \leq z_1 < z_2 \leq 1$ and $\mathsf{value}(\mathcal{A}) \in [z_1, z_2]$. Initially z_1, z_2 are set to $0, 1$ respectively.

The following procedure is iterated until $z_2 - z_1 \leq \epsilon$. In each iteration, it first computes a deterministic automaton $\mathcal{B}$ such that $\mathsf{L}_{\geq x+y}(\mathcal{A}) \subseteq L(\mathcal{B}) \subseteq \mathsf{L}_{>x}(\mathcal{A})$ where $x = z_1 + \frac{z_2 - z_1}{3}$ and $y = \frac{z_2 - z_1}{3}$. Such an automaton $\mathcal{B}$ can be computed since $\mathcal{A}$ is effectively regular-approximable. (Observe that both $x + y - z_1$ and $z_2 - x$ are equal to $\frac{2}{3}(z_2 - z_1)$.) Now, the algorithm checks if $L(\mathcal{B}) = \emptyset$. If $L(\mathcal{B}) = \emptyset$ then this implies $\mathsf{L}_{\geq x+y}(\mathcal{A}) = \emptyset$ and hence $\mathsf{value}(\mathcal{A})$ lies in the interval $[z_1, x + y]$; in this case, it repeats the above procedure by setting $z_2 = x + y$ and keeping z_1 unchanged. On the other hand, if $L(\mathcal{B}) \neq \emptyset$, then this implies that $\mathsf{value}(\mathcal{A})$ lies in the interval $[x, z_2]$; so, in this case the algorithm sets $z_1 = x$, keeps z_2 unchanged and repeats the above procedure.

Notice that the length of the interval (z_1, z_2) at the beginning of each succeeding iteration is $\frac{2}{3}$rd of its value at the beginning of the preceding iteration; further, at the beginning of the first iteration, its value is 1. From this we see that this algorithm terminates after k iterations where k is the least value such that $(\frac{2}{3})^k \leq \epsilon$, that is, $k = \lceil \log_{\frac{3}{2}}(\frac{1}{\epsilon}) \rceil$. From our arguments, we see that at the beginning of each iteration, we have $\mathsf{value}(\mathcal{A}) \in (z_1, z_2)$ and when it terminates $z_2 - z_1 \leq \epsilon$. Thus, it returns an interval in which $\mathsf{value}(\mathcal{A})$ lies and its length is at most ϵ. Observe that, in the above procedure, we only need to check the emptiness of $L(\mathcal{B})$ in each iteration; no explicit computation of $\mathcal{B}$ is needed. ◀

An immediate consequence of the above observation is that if $\mathcal{A}$ is effectively regular-approximable and x is an isolated cut-point of $\mathcal{A}$, then we can check the emptiness of $\mathsf{L}_{>x}(\mathcal{A})$.

▶ **Corollary 12.** *There is a procedure* IsoEmpty *that given an effectively regular-approximable* PFA $\mathcal{A}$ *and a threshold* x *such that* x *is an isolated cut-point of* $\mathcal{A}$, *terminates and decides if* $\mathsf{L}_{>x}(\mathcal{A}) = \emptyset$.

Proof. Observe that $\mathcal{A}$ is isolated at x with a degree of isolation ϵ_0 then either $\mathsf{value}(\mathcal{A}) \geq x + \epsilon_0$ or $\mathsf{value}(\mathcal{A}) \leq x - \epsilon_0$. IsoEmpty works iteratively as follows. Initially it sets $\epsilon = \frac{1}{2}$ and uses the algorithm ComputeVal in Theorem 11 to compute $[z_1, z_2]$ such that $\mathsf{value}(\mathcal{A}) \in [z_1, z_2]$ and $z_2 - z_1 \leq \epsilon$. If $x \in [z_1, z_2]$ then it sets $\epsilon = \frac{\epsilon}{2}$ and repeats. Otherwise if $z_1 > x$ then it returns 1 and if $z_2 < x$ then it returns 0. It is easy to see that IsoEmpty always returns the correct answer and terminates when ϵ takes a value $< \epsilon_0$. ◀

4 Leak monotonicity and complexity

We shall now identify a *semantic* class of PFAs that are regular-approximable. Our proof of the fact that polynomial ambiguous automata are regular-approximable shall be established by showing that they belong to this class. In order to define these classes, we shall need the concept of a *leak*. Intuitively, a leak happens at a position i in a run $q_0 q_1 \ldots q_n$ of $\mathcal{A}$ on input u if the probability of transitioning from q_i to q_{i+1} is non-zero and yet is less than 1.

▶ **Definition 13.** Consider a PFA $\mathcal{A} = (Q, q_s, \delta, Q_f)$ over an alphabet Σ. We say that a triple (q, a, q'), where $q, q' \in Q$ and $a \in \Sigma$, is a *leaky* transition of $\mathcal{A}$ if $0 < \delta(q, a, q') < 1$. Let $u \in \Sigma^*$ be a finite word and ρ be a run of $\mathcal{A}$ on u. We let $\mathsf{NbrLeaks}(\mathcal{A}, u, \rho)$ to be the number of leaky transitions in ρ with respect to the word u; formally, it is $|\{i \mid 0 \leq i < |\rho|, \delta(\rho[i], u[i], \rho[i+1]) < 1\}|$.

4.1 Leak Monotonicity

The class of PFAs that we will be interested in are PFAs in which the measure of accepting a word is concentrated mostly in runs with a few leaks. We formalize this intuition below:

▶ **Definition 14.** Let $\epsilon \in (0, 1)$ be a rational number. We say that $\mathcal{A}$ is *ϵ-leak monotonic* if there exists some $N_\epsilon \in \mathbb{N}$ such that for all $u \in \Sigma^*$, the measure of accepting runs of $\mathcal{A}$ on u having at least N_ϵ leaks is strictly less than ϵ. Such an N_ϵ will be called a horizon of ϵ-leak monotonicity of $\mathcal{A}$.

▶ **Example 15.** The PFA $\mathcal{A}_1$ in Figure 1a on page 7, can be shown to be ϵ-leak monotonic by taking N_ϵ to be any integer n such that $(\frac{2}{3})^n \leq \epsilon$. In contrast, the PFA $\mathcal{A}_2$ in Figure 1b is not ϵ-leak monotonic for any $\epsilon \in (0, 1)$. This is an immediate consequence of Theorem 8 and Theorem 16 established below.

The following theorem connects ϵ-leak monotonicity with regular-approximability.

▶ **Theorem 16.** *If $\mathcal{A}$ is a* PFA *over an alphabet Σ which is ϵ-leak monotonic then $\mathcal{A}$ is regular-approximable with respect to every pair (x, ϵ), for $x \in [0, 1]$ and $\epsilon > 0$.*

Proof. Let PFA $\mathcal{A} = (Q, q_s, \delta, Q_f)$ over alphabet Σ be ϵ-leak monotonic. Let $N \in \mathbb{N}$ be an integer such that $\forall u \in \Sigma^*$, the probability measure, of all accepting runs of $\mathcal{A}$ on u having at least N leaks, is at most ϵ. Let $x \in [0, 1]$. Now, we give the construction of a deterministic automaton $\mathcal{B}$ on alphabet Σ such that $\mathsf{L}_{\geq x+\epsilon}(\mathcal{A}) \subseteq \mathsf{L}(\mathcal{B}) \subseteq \mathsf{L}_{>x}(\mathcal{A})$.

Without loss of generality, let $Q = \{q_0, q_1, ..., q_{n-1}\}$ with the start state $q_s = q_0$. For any $u \in \Sigma^*$, let $LeakPr_u$ be a $n \times N$ matrix such that, for $0 \leq i < n$ and $0 \leq j < N$, $LeakPr_u(i, j)$ is the probability measure of all runs ρ of $\mathcal{A}$ on input u starting from q_0, ending in state q_i and having exactly j leaky transitions, i.e., $\mathsf{NbrLeaks}(\mathcal{A}, u, \rho) = j$.

Consider the automaton (not necessarily finite) $\mathcal{B} = (R, r_0, \delta', R_f)$ where $R = \{LeakPr_u \mid u \in \Sigma^*\}$; r_0 is the matrix such that $r_0(0, 0) = 1$ and $r_0(i, j) = 0$ when $i \neq 0$ or $j \neq 0$; $R_f = \{r \mid (\sum_{i:q_i \in Q_f} \sum_{0 \leq j < N} r(i, j)) > x\}$. We define δ' as follows. Let $r \in R$ and $a \in \Sigma$.

By definition, there exists $u \in \Sigma^*$ such that $r = LeaksPr_u$. Let $r' = LeaksPr_{ua}$. Fix any i, j such that $0 \leq i < n$ and $0 \leq j < N$. Let p_1 be the sum of all $r(i', j)$ such that $\delta(q_{i'}, a, q_i) = 1$, i.e., the transition $(q_{i'}, a, q_i)$ is not a leaky transition of $\mathcal{A}$. Let p_2 be a value defined as follows: if $j = 0$ then $p_2 = 0$, otherwise p_2 is the sum of $r(i', j-1) \cdot \delta(q_{i'}, a, q_i)$ where the sum is taken over all i' such that $\delta(q_{i'}, a, q_i) < 1$, i.e., $(q_{i'}, a, q_i)$ is a leaky transition of $\mathcal{A}$. It is easily shown that $r'(i, j) = p_1 + p_2$. We call r' as the a-successor of r. Observe that the values p_1, p_2 for a given pair i, j are independent of u and hence, the relationship between r, r', as given above, is independent of u. This leads us to the following definition of δ'. We define δ' so that $\delta'(r, a, r') = 1$ iff r' is the a-successor of r. Now, by induction on $|u|$, we can easily show that, for any $r \in R$, $\delta'_u(r_0, r) = 1$ iff $r = LeaksPr_u$.

Now, we show that R is a finite set and bound its size. Let D be the maximum of the denominators of the non-zero transition probabilities of $\mathcal{A}$. The probability of any run of $\mathcal{A}$, on some input, having less than N leaks is a rational number $\frac{x'}{y'}$ where y' is a positive integer such that $y' \leq D^N$. For any state $r \in R$ and for any i, j, $i < n, j < N$, the value of the entry $r(i, j)$ is the sum of the probabilities of some runs of $\mathcal{A}$ each having fewer than N leaks; the least common multiple of the denominators of these probabilities is bounded by $D^{N \cdot D^N}$. Hence $r(i, j)$ is either zero, or is a rational number whose denominator is bounded by $D^{N \cdot D^N}$. This implies that the number of distinct values $r(i, j)$ can take is bounded by $1 + (D^{N \cdot D^N})^2 = 1 + D^{2N \cdot D^N}$. Since r has $n \cdot N$ such entries, we see that $|R|$, which is the number of distinct values r can take, is bounded by $(1 + D^{2N \cdot D^N})^{n \cdot N}$ and hence is finite.

Now we show that $\mathsf{L}_{\geq x+\epsilon}(\mathcal{A}) \subseteq \mathsf{L}(\mathcal{B}) \subseteq \mathsf{L}_{>x}(\mathcal{A})$. Consider any $u \in \Sigma^*$. The set of accepting runs of $\mathcal{A}$ on u can be partitioned into two sets X_1, X_2 which are, respectively, the sets of runs having less than N leaks, or having at least N leaks. Let z_1, z_2, respectively, be the probability measures of these two sets of runs. Clearly, $\mathsf{P}_{\mathcal{A}}(u) = z_1 + z_2$. Based on the value of N, we have $z_2 \leq \epsilon$. Suppose that r is the unique state in R such that $\delta'_u(r_0, r) = 1$. Then, from our earlier observations, we see that $\sum_{i:q_i \in Q_f} \sum_{j<N} r(i, j) = z_1$. If $u \in \mathsf{L}_{\geq x+\epsilon}(\mathcal{A})$ then $z_1 > x$ since $z_2 < \epsilon$, and from the definition of R_f, it follows that $r \in R_f$ and $u \in \mathsf{L}(\mathcal{B})$. Thus, we see that $\mathsf{L}_{\geq x+\epsilon}(\mathcal{A}) \subseteq \mathsf{L}(\mathcal{B})$. If $u \in \mathsf{L}(\mathcal{B})$ then, from the definition of R_f, we have $z_1 > x$ and hence $u \in \mathsf{L}_{>x}(\mathcal{A})$. Thus, we see that $\mathsf{L}(\mathcal{B}) \subseteq \mathsf{L}_{>x}(\mathcal{A})$. ◀

▶ Remark. The deterministic automaton $\mathcal{B}$ that we construct for an ϵ-leak monotonic PFA $\mathcal{A}$ in the proof of Theorem 16 has the following property: for each input string u, the state r that is reached in $\mathcal{B}$ on input u, starting from its initial state, gives the probability of acceptance of u by A with precision ϵ. Equivalently, there is a function f from the states of $\mathcal{B}$ to $[0, 1]$ such that $f(q) \leq \mathsf{P}_{\mathcal{A}}(u) < f(q) + \epsilon$. $f(q)$ can be computed in time polynomial in the size of the representation of q. The above observations imply that the value of $\mathcal{A}$ lies in the interval $[v, v + \epsilon]$ where $v = \max f(q)$. Thus, if $\mathcal{B}$ can be constructed then value of $\mathcal{A}$ can be approximated within ϵ.

However, there are regular-approximable PFAs that are not ϵ-leak monotonic for any ϵ.

▶ **Proposition 17.** *There is a* PFA $\mathcal{A}$ *that is regular-approximable but not ϵ-leak monotonic for any $\epsilon \in (0, 1)$.*

Proof. Consider the PFA $\mathcal{A}_3$ shown in Figure 1c on page 7. Given $x \in (0, 1)$, let n_x be the largest integer such that $\frac{1}{2}^{n_x} > x$. It is easy to see that $\mathsf{L}_{>x}(\mathcal{A}_3) = \lambda + \{\mathbf{0}, \mathbf{1}\}^*\mathbf{1}(\lambda + \mathbf{0} + \mathbf{0}^2 + \ldots \mathbf{0}^{n_x})$ where λ is the empty word. Thus, $\mathsf{L}_{>x}(\mathcal{A}_3)$ is regular for each x and hence regular-approximable. Furthermore, observe that for each n, the word $u_n = (\mathbf{01})^n$ is accepted by $\mathcal{A}_3$ with probability 1. In addition, for each n, u_n has exactly 2^n runs, each of which is accepting and has exactly n leaks. From these observations, it is easy to see that $\mathcal{A}_3$ is not ϵ-leak monotonic for any ϵ – for every possible horizon N_ϵ there are infinitely many words such that the measure of accepting runs having at least N_ϵ leaks is 1. ◀

The following theorem shows that the problem of checking if a given PFA is ϵ-leak monotonic with respect to given $\epsilon \in (0,1)$ is undecidable.

▶ **Theorem 18.** *Given a* PFA $\mathcal{A}$ *and a rational value* $\epsilon \in (0,1)$*, the problem of checking if* $\mathcal{A}$ *is* ϵ*-leak monotonic is undecidable. Formally the set* $\mathsf{LeakMon} = \{(\mathcal{A}, \epsilon) \,|\, \epsilon \in (0,1), \mathcal{A}$ *is a* PFA *that is* ϵ*-leak monotonic*$\}$ *is undecidable.*

It is easy to see that we can give a simple algorithm that takes as input $\mathcal{A}, x, N$ and constructs the deterministic automaton $\mathcal{B}$ defined in the proof of Theorem 16. Such an algorithm starts with an initial set of states of $\mathcal{B}$ which is taken to be r_0 and enlarges this set by choosing an unexplored state from it, and explores it by constructing and adding all its a-successors, that are not already present, to the set of states, for each $a \in \Sigma$. This algorithm terminates when no new states can be added. Hence if we can compute a horizon of ϵ-leak monotonicity of an ϵ-leak monotonic $\mathcal{A}$ then we can compute the regular language that approximates $\mathsf{L}_{>x}(\mathcal{A})$ for every threshold x.

▶ **Definition 19.** We say that a PFA $\mathcal{A}$ is *leak monotonic* if $\mathcal{A}$ is ϵ-leak monotonic with respect to every $\epsilon \in (0,1)$. $\mathcal{A}$ is said to be *effectively leak monotonic* if there is an algorithm that given ϵ outputs a horizon of ϵ-leak monotonicity of $\mathcal{A}$. A class $\mathcal{C}$ of leak monotonic PFAs is said to be *effectively* leak monotonic if there is a procedure that, given $\mathcal{A} \in \mathcal{C}$ and $\epsilon > 0$ terminates and outputs a horizon of ϵ-leak monotonicity of $\mathcal{A}$.

The PFA $\mathcal{A}_1$ given in Figure 1a on page 7 is leak monotonic. We have the following as a consequence of Theorem 16.

▶ **Corollary 20.** *If a* PFA *is (effectively) leak monotonic then it is (effectively) regular-approximable.*

The following theorem allows us to construct leak monotonic PFAs from smaller leak monotonic PFAs.

▶ **Theorem 21.** *If a* PFA $\mathcal{A} = (Q, \delta, q_s, Q_f)$ *over* Σ *is such that* Q *can be partitioned into sets* $Q_0, \ldots, Q_m$ *such that* $q_s \in Q_0$ *and the following conditions hold:*

1. *For each* $i \geq 1, q \in Q_i$ *and* $a \in \Sigma$, $\mathsf{post}(q,a) \subseteq Q_i$.
2. *There is a constant* $m > 0$ *such that from every state in* Q_0 *and on every input* u *of length at least* m*, some state outside* Q_0 *is reachable, and*
3. *For* $i > 0$*, the restriction of* $\mathcal{A}$ *to each* Q_i*, when started in any state* $q \in Q_i$*, is leak monotonic,*

then $\mathcal{A}$ *is leak monotonic.*

4.2 Leak Complexity

In this subsection, we introduce a *syntactic* class of PFAs that are leak monotonic. The syntactic class of PFAs shall be defined through the concept of *leak complexity* defined below.

▶ **Definition 22.** Let $f : \mathbb{N} \to \mathbb{N}$ be a function. We say that the *leak complexity* of $\mathcal{A}$ is given by f (or is simply f) if for all $u \in \Sigma^*$, for all $\ell \in \mathbb{N}$, the number of accepting runs of $\mathcal{A}$ on u having exactly ℓ leaks is at most $f(\ell)$, i.e., $|\{\rho \,|\, \rho$ is an accepting run of $\mathcal{A}$ on u and $\mathsf{NbrLeaks}(\mathcal{A}, u, \rho) = \ell\}| \leq f(\ell)$.

Notice that we are only using the accepting runs to define the leak complexity. Further, observe that if f, g are functions from $\mathbb{N}$ to $\mathbb{N}$ such that $f(\ell) \leq g(\ell)$ for all $\ell \in \mathbb{N}$, and the leak complexity of $\mathcal{A}$ is given by f, then its leak complexity is also given by g. We try to use the tightest function to specify the leak complexity of a PFA.

We shall be interested in PFAs whose leak complexity is given by special functions.

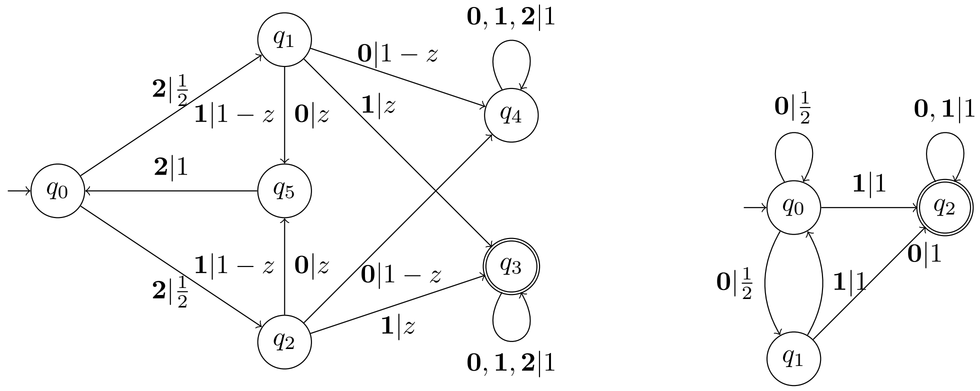

Figure 2 Automaton $\mathcal{A}_z$ on the left (a) and Automaton $\mathcal{A}_5$ on the right (b).

▶ **Definition 23.** Let $\mathcal{A} = (Q, \delta, q_s, Q_f)$ be a PFA.

- $\mathcal{A}$ is said to have *polynomial leak complexity* if its leak complexity is given by a polynomial function h.
- For $\mathcal{A}$, let $MaxTrPr(\mathcal{A})$ be maximum probability of a leaky transition, i.e., the value $\max\{\delta(q, a, q') \mid 0 < \delta(q, a, q') < 1, q, q' \in Q, a \in \Sigma\}$. We say that $\mathcal{A}$ has *sub-exponential leak complexity* if there exist constants $c, d > 0$ such that $d < \frac{1}{MaxTrPr(\mathcal{A})}$ and the leak complexity of $\mathcal{A}$ is $c \cdot d^\ell$.

Clearly, if $\mathcal{A}$ has polynomial leak complexity then it has sub-exponential leak complexity.

▶ **Example 24.** For the PFA $\mathcal{A}_1$ in Figure 1a on page 7, on any input, the number of accepting runs having ℓ leaks is at most 1 and hence its leak complexity is constant. Figure 2a shows a PFA $\mathcal{A}_z$ over the input alphabet $\Sigma = \{\mathbf{0}, \mathbf{1}, \mathbf{2}\}$ that has sub-exponential leak complexity, but not polynomial leak complexity. Here $z \in (0, 1)$ is a number that is left unspecified. In the figure, all unspecified transitions, from states q_0, q_1, q_2, q_5, on the appropriate input symbols, go to the reject state q_4 with probability 1. Both q_3, q_4 are absorbing states in which q_3 is the accepting state. It is not difficult to see that all accepting runs of $\mathcal{A}_z$ on an input word have an even number of leaks. Furthermore, for an even ℓ, the number of accepting runs having ℓ leaks is exactly $2^{\frac{\ell}{2}}$, i.e., $(\sqrt{2})^\ell$. Observe that $MaxTrPr(\mathcal{A}_z) = z$ if $z > \frac{1}{2}$ else it is $1 - z$. Hence, $\mathcal{A}_z$ has subexponential leak complexity iff $1 - \frac{1}{\sqrt{2}} < z < \frac{1}{\sqrt{2}}$. Thus, for example, $\mathcal{A}_z$ has sub-exponential leak complexity if $z = \frac{2}{3}$. On the other hand $\mathcal{A}_z$ does not have subexponential leak complexity if $z = \frac{3}{4}$. However, note that $\mathcal{A}_z$ is leak monotonic for each $z \in (0, 1)$ as $\mathcal{A}_z$ satisfies conditions of Theorem 21 with $m = 2, Q_0 = \{q_0, q_1, q_5, q_2\}, Q_1 = \{q_4\}$ and $Q_2 = \{q_3\}$.

We show that every PFA that has sub-exponential leak complexity is leak monotonic.

▶ **Theorem 25.** *If a* PFA $\mathcal{A}$ *over an alphabet* Σ *has sub-exponential leak complexity then* $\mathcal{A}$ *is leak monotonic and hence regular-approximable.*

Proof. Let $\mathcal{A} = (Q, q_s, \delta, Q_f)$ be a PFA over alphabet Σ with sub-exponential leak complexity. This means, there exist constants $c, d > 0$ such that $d < \frac{1}{MaxTrPr(\mathcal{A})}$ and the leak complexity of $\mathcal{A}$ is $c \cdot d^\ell$, i.e. on every word $u \in \Sigma^*$ the number of accepting runs of $\mathcal{A}$ on u having ℓ leaks is bounded by $c \cdot d^\ell$. We prove the theorem by showing that $\mathcal{A}$ is leak monotonic and appealing to Corollary 20. Let $\epsilon \in [0, 1]$ be such that $\epsilon > 0$. Let $p = d \cdot MaxTrPr(\mathcal{A})$.

Observe that $0 < p < 1$ since $d < \frac{1}{MaxTrPr(\mathcal{A})}$. Now, let $N \in \mathbb{N}$ be the smallest integer such that

$$N > \frac{\log(\frac{c}{\epsilon\cdot(1-p)})}{\log\frac{1}{p}} \tag{1}$$

Consider any $u \in \Sigma^*$. Let z_2 be the probability measure of accepting runs of $\mathcal{A}$ having at least N leaks. The probability of any single run having ℓ leaks is bounded by $(MaxTrPr(\mathcal{A}))^\ell$. Since there are at most $c \cdot d^\ell$ accepting runs of $\mathcal{A}$ on u having ℓ leaks, we see that $z_2 \leq \sum_{\ell\geq N} c \cdot d^\ell \cdot (MaxTrPr(\mathcal{A}))^\ell$. Using $p = d \cdot MaxTrPr(\mathcal{A})$, we have

$$z_2 \leq \sum_{\ell\geq N} c \cdot p^\ell \;= c \cdot p^N \cdot \sum_{\ell\geq 0} p^\ell.$$

From this we see that $z_2 \leq c \cdot p^N \cdot \frac{1}{1-p}$. Now using inequality (1) and raising both its two sides to the power of 2, after simplification, we get $(\frac{1}{p})^N > \frac{c}{\epsilon\cdot(1-p)}$, which leads to $\epsilon > p^N \cdot \frac{c}{1-p} \geq z_2$. Hence, we see that $\mathcal{A}$ is ϵ-leak monotonic. Clearly this holds for every $\epsilon \in [0, 1]$ such that $\epsilon > 0$. Hence $\mathcal{A}$ is leak monotonic. ◀

Observe that the proof of Theorem 25 also shows that if the (sub-exponential) leak complexity function of $\mathcal{A}$ is known (or can be computed) then $\mathcal{A}$ is effectively regular-approximable. Theorem 25 can be used to identify classes of PFAs that are leak monotonic. In conjunction with Theorem 21 and Theorem 16, it can be used to identify regular-approximable PFAs . We conclude by showing that the class of Hierarchical PFAs (HPA)s (introduced in [9, 6]) is effectively leak monotonic.

Hierarchical PFAs (HPA)s

(HPAs), introduced in [9, 6], are defined as follows. A k-HPA $\mathcal{A}$ on Σ is a probabilistic automaton whose states can be partitioned into $k + 1$ levels $Q_0, Q_1, \ldots, Q_k$ such that for any state q and input symbol $a \in \Sigma$, at most one successor state is at the same level, and others are higher level states. In other words for each $q \in Q_i$ and $a \in \Sigma$, $\mathsf{post}(q, a) \subseteq Q_i \cup Q_{i+1} \cdots \cup Q_k$ and $|\mathsf{post}(q, a) \cap Q_i| \leq 1$. Without loss of generality, we can assume that the initial state is at level 0. The following theorem shows that the class of HPAs are effectively leak monotonic and hence regular-approximable.

▶ **Theorem 26.** *Every k-HPA $\mathcal{A}$ with n-states and $k > 0$, has leak complexity at most $n^k\ell^{k-1}$. Hence, the class of hierarchical probabilistic automata is effectively leak monotonic and hence regular-approximable.*

▶ **Example 27.** The automaton $\mathcal{A}_1$ in Figure 1a on page 7 is a 1-HPA whose leak complexity is 1. Automaton $\mathcal{A}_z$ in Figure 2a on page 12 is not a HPA.

Thanks to Theorem 11 and Corollary 12, the values of HPAs can be approximated and emptiness checked under isolation. These facts are also established in [2] through an alternative proof.

5 Ambiguity and Approximability

We now identify a large class of PFAs which are effectively leak monotonic. Any PFA $\mathcal{A}$ over Σ can be viewed as a non-deterministic finite automaton NFA $\mathsf{nfa}(\mathcal{A})$ over Σ by ignoring the probability of transitioning from one state to another: $\mathsf{nfa}(\mathcal{A})$ has the same set of states as

$\mathcal{A}$ and there is a transition from state q to q' on a in $\mathsf{nfa}(\mathcal{A})$ iff $\delta(q,a,q') > 0$. The degree of ambiguity of $\mathcal{A}$ on word u is the degree of ambiguity of $\mathsf{nfa}(\mathcal{A})$ on word u. We will be interested in PFAs that are polynomially ambiguous. We have the following observation.

▶ **Proposition 28.** *If a* PFA $\mathcal{A}$ *has polynomial leak complexity with polynomial* $h(\ell)$ *then* $\mathcal{A}$ *is polynomially ambiguous with polynomial* $nh(n)$.

Proof. Let $\mathcal{A}$ have polynomial leak complexity with polynomial $h(\ell)$. Any accepting run of $\mathcal{A}$ on a word of length n can have at most n leaks. Thus the number of accepting runs of $\mathcal{A}$ on a word of length n is bounded above by $\sum_{\ell=1}^{n} h(\ell) \leq nh(n)$. ◀

From the proof of Theorem 26 and Proposition 28, we can conclude that every HPA is polynomially ambiguous. However, the converse is not true. We give an example of a linearly ambiguous PFA that is not a HPA.

▶ **Example 29.** Consider the PFA $\mathcal{A}_5$ on $\Sigma = \{\mathbf{0}, \mathbf{1}\}$ shown in Figure 2b on page 12. $\mathcal{A}_5$ is not hierarchical. This can be seen as follows. Since $S = \{q_0, q_1\}$ form a strongly connected component, they must be in the same level. However, then $\mathsf{post}(q_0, \mathbf{0}) = \{q_0, q_1\}$ has two successors in the same level. Next, observe that on input $\mathbf{0}^k$ there are only two runs that remain in S. Thus, on input $\mathbf{0}^k$ there are $k-1$ accepting runs. On the other hand, on input $\mathbf{0}^k\mathbf{1}$ there is exactly one run that remains in S, and this run ends in q_0. Further, the number of accepting runs on $\mathbf{0}^k\mathbf{1}$ is k. Now a general input over Σ is either $u = \mathbf{0}^{k_1}\mathbf{1}\mathbf{0}^{k_2}\mathbf{1}\cdots\mathbf{1}^{k_n}$ or $u\mathbf{1}$. Putting the above observations together, we have the number of accepting runs on u is $k_1 + k_2 + \cdots k_{n-1} + (k_n - 1)$ and on $u\mathbf{1}$ is $k_1 + k_2 + \cdots k_n$. Thus, $\mathcal{A}_5$ has linear ambiguity.

Thanks to Theorem 26 and Proposition 28, we can conclude that a k-HPA is polynomially ambiguous with polynomial $O(n^k)$. Since the value decision problem and emptiness problem of 2-HPAs are undecidable [8, 2], we get that the value decision problem and emptiness problem for quadratically ambiguous PFAs is also undecidable. The emptiness problem for quadratically ambiguous PFAs is shown to be undecidable in [16] as well. The problem of emptiness of linearly ambiguous PFAs was left open. A close examination of the 2-HPAs constructed in the undecidability proof of the emptiness problem for 2-HPAs established in [2], shows that the resulting automata have only linear ambiguity (instead of quadratic ambiguity). This observation proves that the emptiness problem of linearly ambiguous automata is undecidable. This result (with a different proof) was also independently observed in [13].

▶ **Theorem 30.** *The emptiness problem for linearly ambiguous* PFA*s is undecidable.*

In contrast, we will show that polynomially ambiguous automata are effectively regular-approximable, which will imply that their value can be approximated and emptiness under isolation be checked thanks to Theorem 11 and Corollary 12. We establish this by showing that every polynomially ambiguous PFA has polynomial leak complexity. This is a consequence of Lemma 32 below, which will allow us to bound leak complexity from bounds on degree of ambiguity. We need one further definition.

▶ **Definition 31.** For a PFA $\mathcal{A}$ on Σ, word $u \in \Sigma^*$ and $\ell \in \mathbb{N}$, let $\mathsf{accruns}(\mathcal{A}, u, \ell)$ be the set of accepting runs of $\mathcal{A}$ on u with leaks $\leq \ell$. Formally, $\mathsf{accruns}(\mathcal{A}, u, \ell)$ is the set $\{\rho \mid \rho \text{ is accepting and } \mathsf{NbrLeaks}(\mathcal{A}, u, \rho) \leq \ell\}$.

We now show that for any word u and any integer ℓ, there is a *short* word v such that v has at least as many accepting runs with at most ℓ leaks as u does.

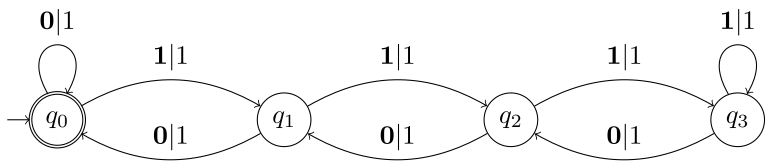

Figure 3 Deterministic automaton $\mathcal{A}_6$ that is not eventually weakly ergodic.

▶ **Lemma 32.** *Let $\mathcal{A}$ be a* PFA *with m states. For any word u and integer $\ell > 0$, there is a word v of length $\leq m + ((\ell+1)m)^m$ such that* $|\mathsf{accruns}(\mathcal{A}, v, \ell)| \geq |\mathsf{accruns}(\mathcal{A}, u, \ell)|$.

Polynomial ambiguity implies polynomial leak complexity follows from Lemma 32.

▶ **Theorem 33.** *If* PFA *$\mathcal{A}$ with m states is polynomially ambiguous with polynomial $p(n)$ then $\mathcal{A}$ has polynomial leak complexity with polynomial $h(\ell) = p(m + ((\ell+1)m)^m)$.*

Proof. Let $\mathcal{A}$ be a PFA with m states. Fix an input word u and an integer ℓ. From Lemma 32, there is a word v such that $|v| \leq m + ((\ell+1)m)^m$ and $|\mathsf{accruns}(\mathcal{A}, u, \ell)| \leq |\mathsf{accruns}(\mathcal{A}, v, \ell)|$. Now $\mathsf{accruns}(\mathcal{A}, v, \ell)$ is a subset of the accepting runs of $\mathcal{A}$ on input v. Since $\mathcal{A}$ is polynomially ambiguous, we get $\mathsf{accruns}(\mathcal{A}, v, \ell) \leq p(|v|) = p(m + ((\ell+1)m)^m)$. ◀

Thanks to Theorem 33, we get that

▶ **Corollary 34.** *The class of polynomially ambiguous* PFA*s is effectively regular-approximable. The value of a polynomially ambiguous* PFA *can be approximated to any degree of precision and emptiness checked under isolation.*

6 Eventually Weakly Ergodic PFAs

Not all effectively regular-approximable PFAs are leak monotonic. We exhibit a class of PFAs from the literature that is effectively regular-approximable but not leak monotonic. Recall that a Markov Chain is ergodic if it is aperiodic and its underlying transition graph is strongly connected. Ergodicity in the context of PFAs have been studied in [29, 23, 19]. Intuitively, a PFA is weakly ergodic if any sequence of input symbols has only one terminal strongly connected component and this component is aperiodic. Weak ergodicity was generalized in [10] to define a new class of PFAs, called *eventually weakly ergodic* PFAs. Informally, a PFA $\mathcal{A}$ is eventually weakly ergodic if its states can be partitioned into sets $Q_T, Q_1, \ldots, Q_r$ and there is an ℓ such that in the transition graph on any word of length ℓ, $Q_1, \ldots, Q_r$ are the only terminal strongly connected components, and in addition, they are aperiodic. (See Appendix F for the formal definition.) Every unary PFA turns out to be eventually weakly ergodic [10]. The problem of checking whether a PFA is eventually weakly ergodic is also decidable [10].

▶ **Example 35.** The PFA $\mathcal{A}_3$ in Figure 1c on page 7 is eventually weakly ergodic but not leak monotonic. This can be seen by taking $\ell = 1, Q_T = \emptyset, Q_1 = \{q_0, q_1\}$. On the other hand, the deterministic automaton $\mathcal{A}_6$ in Figure 3 is shown to be **not** eventually weakly ergodic in [10]. Thus, the class of leak monotonic automata and eventually weakly ergodic automata are not comparable.

Using the techniques of [10], we can show that the class of weakly ergodic PFAs is effectively regular-approximable. (See Appendix F for the proof.)

▶ **Theorem 36.** *The class of eventually weakly ergodic* PFA*s is effectively regular-approximable.*

Thus, we can approximate the value of eventually weakly ergodic PFAs and check emptiness under isolation for eventually weakly ergodic PFAs. Please note that the latter result is also given in [10].

7 Conclusions

In this paper, we addressed the problem of regular-approximability of PFAs. We showed that regular-approximability problem is undecidable. We also showed that if a PFA is regular-approximable then its value can be computed with arbitrary precision. We also showed that emptiness problem is decidable for regular-approximable PFAs when the given cut-point is isolated. We defined a class of PFAs, called leak monotonic PFAs and showed them to be regular-approximable. For PFAs belonging to this class, we gave an effective procedure for computing a deterministic automaton that approximates the language accepted by the given PFA with a given minimum probability threshold. We showed that PFAs with polynomial ambiguity as well as all HPAs are leak monotonic. We also introduced leak complexity and showed that PFAs with sub-exponential leak complexity are leak monotonic. We also solved an open problem showing that the emptiness problem is undecidable for PFAs with linear ambiguity. Finally, we showed that eventually weakly ergodic PFAs are also regular-approximable. As part of future work, it will be interesting to investigate algorithms to decide if a given PFA has sub-exponential leak complexity. The decidability of determining whether a given PFA is leak monotonic and checking emptiness under isolation for general PFAs are some other open problems.

References

1 C. Baier and M. Größer. Recognizing ω-regular languages with probabilistic automata. In *20th IEEE Symposium on Logic in Computer Science*, pages 137–146, 2005.

2 Y. Ben and A. P. Sistla. Model checking failure-prone open systems using probabilistic automata. In *13th International Symposium on Automated Technology for Verification and Analysis*, volume 9364 of *Lecture Notes in Computer Science*, pages 148–165. Springer, 2015.

3 A. Bertoni. The solution of problems relative to probabilistic automata in the frame of the formal languages theory. In *GI Jahrestagung*, pages 107–112, 1974.

4 A. Bertoni. Mathematical methods of the theory of stochastic automata. In *3rd Symposium of Mathematical Foundations of Computer Science*, volume 28 of *Lecture Notes in Computer Science*, pages 9–22. Springer, 1975.

5 R. Chadha, A. P. Sistla, and M. Viswanathan. Probabilistic Büchi automata with non-extremal acceptance thresholds. In *11th International Conference on Verification, Model checking and Abstract Interpretation*, pages 103–117, 2010.

6 R. Chadha, A. P. Sistla, and M. Viswanathan. Power of randomization in automata on infinite strings. *Logical Methods in Computer Science*, 7(3):1–22, 2011.

7 R. Chadha, A. P. Sistla, M. Viswanathan, and Y. Ben. Decidable and expressive classes of probabilistic automata. In *18th International Conference on Foundations of Software Science and Computation Structures*, volume 9034 of *Lecture Notes in Computer Science*, pages 200–214. Springer, 2015.

8 R. Chadha, A. Prasad Sistla, and M. Viswanathan. Emptiness under isolation and the value problem for hierarchical probabilistic automata. In *Foundations of Software Science and Computation Structures - 20th International Conference, FOSSACS 2017*, volume 10203 of *Lecture Notes in Computer Science*, pages 231–247, 2017.

9 R. Chadha, A.P. Sistla, and M. Viswanathan. Power of randomization in automata on infinite strings. In *20th International Conference on Concurrency Theory*, pages 229–243, 2009.

10 R. Chadha, A.P. Sistla, and M. Viswanathan. Probabilistic automata with isolated cut-points. In *38th International Symposium on Mathematical Foundation of Computer Science*, pages 254–265, 2013.

11 A. Condon and R. J. Lipton. On the complexity of space bounded interactive proofs (extended abstract). In *30th Annual Symposium on Foundations of Computer Science*, pages 462–467, 1989.

12 W. Czerwinski and S. Lasota. Regular separability of one counter automata. In *32nd IEEE Symposium on Logic in Computer Science*, pages 1–12, 2017.

13 L. Daviaud, M. Jurdzinski, R. Lazic, F. Mazowiecki, G. A. Pérez, and James Worrell. When is containment decidable for probabilistic automata? In *45th International Colloquium on Automata, Languages, and Programming*, 2018. To appear.

14 N. Fijalkow, H. Gimbert, E. Kelmendi, and Youssouf Oualhadj. Deciding the value 1 problem for probabilistic leaktight automata. *Logical Methods in Computer Science*, 11(2), 2015.

15 N. Fijalkow, H. Gimbert, and Y. Oualhadj. Deciding the value 1 problem for probabilistic leaktight automata. In *27th IEEE Symposium on Logic in Computer Science*, pages 295–304, 2012.

16 N. Fijalkow, C. Riveros, and J. Worrell. Probabilistic automata of bounded ambiguity. In *the International Conference on Concurrency Theory*, pages 19:1–19:14, 2017.

17 N. Fijalkow and M. Skrzypczak. Irregular behaviours for probabilistic automata. In *Reachability Problems*, pages 33–36, 2015.

18 H. Gimbert and Y. Oualhadj. Probabilistic automata on finite words: Decidable and undecidable problems. In *37th International Colloquium on Automata, Languages and Programming*, pages 527–538, 2010.

19 J. Hajnal and M. S. Bartlett. Weak ergodicity in non-homogeneous markov chains. *Mathematical proceedings of the Cambridge Philosophical Society*, 54(02):233–246, 1958.

20 O. H. Ibarra and B. Ravikumar. On sparseness, ambiguity and other decision problems for acceptors and transducers. In *3rd Annual Symposium on Theoretical Aspects of Computer Science*, pages 171–179, 1986.

21 G. Jacob. Un algorithme calculant le cardinal, fini ou infini, des demi-groupes de matrices. *Theoretical Computer Science*, 5(2):183–204, 1977.

22 A. Mandel and I. Simon. On finite semigroups of matrices. *Theoretical Computer Science*, 5(2):101–111, 1977.

23 A. Paz. Definite and quasidefinite sets of stochastic matrices. *Proceedings of the American Mathematical Society*, 16(4):634–641, 1965. URL: `http://www.jstor.org/stable/2033893`.

24 A. Paz. *Introduction to Probabilistic Automata.* Academic Press, 1971.

25 T. Place and M. Separation for dot-depth two. In *32nd IEEE Symposium on Logic in Computer Science*, pages 1–12, 2017.

26 M. O. Rabin. Probabilistic automata. *Information and Control*, 6(3):230–245, 1963.

27 C. Reutenauer. Propriétés arithmétiques et topologiques de séries rationnelles en variables non commutatives, 1997. Thése troisiéme cycle, Université Paris VI.

28 A. Weber and H. Seidl. On the degree of ambiguity of finite automata. *Theoretical Computer Science*, 88(2):325–349, 1991.

29 J. Wolfowitz. Products of indecomposable, aperiodic, stochastic matrices. *Proceedings of the American Mathematical Society*, 14(5):733–737, 1963.

A Proof of Theorem 9

Proof. Let $\mathsf{SomeApprox}$ be the set of all PFAs $\mathcal{A}$ such that there is a non-trivial rational pair (x, y) such that $(\mathcal{A}, x, y) \in \mathsf{Approx}$. We show that $\mathsf{SomeApprox}$ is $\mathbf{\Sigma_2^0}$-hard where $\mathbf{\Sigma_2^0}$ is the second level in the arithmetical hierarchy. This automatically implies that Approx is not even recursively enumerable; for if it were recursively enumerable this would imply that $\mathsf{SomeApprox}$ is also recursively enumerable which will be a contradiction.

Let $\mathsf{ValueNot1} = \{\mathcal{A} \mid \mathcal{A}$ is a PFA and $\mathsf{value}(\mathcal{A}) < 1\}$. It has been shown in [10] that $\mathsf{ValueNot1}$ is $\mathbf{\Sigma_2^0}$-complete. We prove that $\mathsf{SomeApprox}$ is $\mathbf{\Sigma_2^0}$-hard by reducing $\mathsf{ValueNot1}$ to $\mathsf{SomeApprox}$. Our reduction, given a PFA $\mathcal{A}$ over Σ, constructs a PFA $\mathcal{B}$ such that $\mathsf{value}(\mathcal{A}) < 1$ iff $\mathcal{B} \in \mathsf{SomeApprox}$. Let $\mathcal{A} = (Q, q_s, \delta, Q_f)$ be any PFA over some alphabet Σ. Now, we define $\mathcal{B}$ as follows. If $\exists u \in \Sigma^*$ such that $\mathsf{P}_{\mathcal{A}}(u) = 1$ then $\mathcal{B}$ is simply the PFA $\mathcal{A}_2$ given in Figure 1b on page 7; observe that in this case, $\mathcal{A} \notin \mathsf{ValueNot1}$, and $\mathcal{B} \notin \mathsf{SomeApprox}$ as shown by Theorem 8. Note that the above condition can be checked effectively thanks to Proposition 4. If there is no such a string u, then we define $\mathcal{B}$ to be a PFA over the alphabet $\Sigma' = \Sigma \cup \{\sharp\}$ defined as follows. $\mathcal{B} = (Q', q_s, \delta', Q_f)$ where $Q' = Q \cup \{q_r\}$ where $q_r \notin Q$ and δ' defined as follows: $\delta'(q, a, q') = \delta(q, a, q')$ for $q, q' \in Q$ and $a \in \Sigma$; $\delta'(q, \sharp, q_s) = 1$ for $q \in Q_f$; $\delta'(q, \sharp, q_r) = 1$ for $q \notin Q_f$; $\delta'(q_r, a, q_r) = 1$ for all $a \in \Sigma'$. Now, we make the following observations. For any $u \in \Sigma^*$, the acceptance probabilities of u by $\mathcal{A}$ and $\mathcal{B}$ are the same. Now consider any string v of the form $u_1\sharp u_2\sharp ... u_k\sharp$ where each $u_i \in \Sigma^*$, for $1 \le i \le k$. It is easy to see that $\mathsf{P}_{\mathcal{B}}(v) = \prod_{1 \le i \le k} \mathsf{P}_{\mathcal{A}}(u_i)$. Also, $\mathsf{value}(\mathcal{B}) = \mathsf{value}(\mathcal{A})$.

Now, we show that $\mathcal{A} \in \mathsf{ValueNot1}$ iff $\mathcal{B} \in \mathsf{SomeApprox}$. Suppose $\mathcal{A} \in \mathsf{ValueNot1}$. In this case, take any $x, y \in (0, 1)$ such that $\mathsf{value}(\mathcal{A}) < x < x + y < 1$. Clearly such x, y exist, since $\mathsf{value}(\mathcal{A}) < 1$. Since $\mathsf{value}(\mathcal{B}) = \mathsf{value}(\mathcal{A})$, we have $\mathsf{value}(\mathcal{B}) < x < x + y < 1$. Clearly $\mathsf{L}_{>x}(\mathcal{B}) = \mathsf{L}_{\ge x+y}(\mathcal{B}) = \emptyset$. Since the empty set is a regular set, we see that $\mathcal{B}$ is approximable with respect to (x, y) and hence $\mathcal{B} \in \mathsf{SomeApprox}$. Now, assume $\mathcal{A} \notin \mathsf{ValueNot1}$. This means $\mathsf{value}(\mathcal{A}) = 1$. Now, we have two cases. In the first case, $\exists u \in \Sigma^*$ such that $\mathsf{P}_{\mathcal{A}}(u) = 1$. In this case, by construction, $\mathcal{B}$ is the automaton $\mathcal{A}_2$ which is not in $\mathsf{SomeApprox}$. The second case is when there is no such string u. This means, for each $i > 0, \exists u_i \in \Sigma^*$ such that $\mathsf{P}_{\mathcal{A}}(u_i) > (1 - \frac{1}{2^i})$. Since $\mathsf{P}_{\mathcal{B}}(u_i) = \mathsf{P}_{\mathcal{A}}(u_i)$, we have $\mathsf{P}_{\mathcal{B}}(u_i) > (1 - \frac{1}{2^i})$. We show that $\mathcal{B} \notin \mathsf{SomeApprox}$ by contradiction. Suppose $\mathcal{B} \in \mathsf{SomeApprox}$. This means $\exists x, y$ and a regular language over $L \subseteq (\Sigma')^*$ such that $0 < x < x + y < 1$ and $\mathsf{L}_{\ge x+y}(\mathcal{B}) \subseteq L \subseteq \mathsf{L}_{>x}(\mathcal{B})$. Since L is a regular language, there exists an integer $N > 0$ satisfying Lemma 2. Now, let $z_1 = \max\{\mathsf{P}_{\mathcal{A}}(u') \mid u' \in \Sigma^*, |u'| \le N\}$. Fix an integer $k > 0$ such that $(z_1)^k \le x$. Now, let $v \in \Sigma^*$ be any string such that $v = u_i$ for some $i > 0$ such that $(\mathsf{P}_{\mathcal{A}}(v))^k \ge x + y$. Clearly such a string v exists. Now consider the string $w = (v\sharp)^k$ in $(\Sigma')^*$. Now, we have $\mathsf{P}_{\mathcal{B}}(w) = (\mathsf{P}_{\mathcal{A}}(v))^k \ge x + y$. Hence $w \in L$. Now applying Lemma 2, we see that there exists a string $w' = w_1\sharp w_2\sharp \cdots w_k\sharp$ such that $w_i \in \Sigma^*, |w_j| \le N$, for $1 \le j \le k$ and $w' \in L$. Clearly, $\mathsf{P}_{\mathcal{A}}(w_i) \le z_1$, for each i, $1 \le i \le k$. Now, $\mathsf{P}_{\mathcal{B}}(w) = \prod_{1 \le i \le k} \mathsf{P}_{\mathcal{A}}(w_i) \le (z_1)^k$. Since $(z_1)^k \le x$, we see that $\mathsf{P}_{\mathcal{B}}(w) \le x$ which contradicts our assumption that $L \subseteq \mathsf{L}_{>x}(\mathcal{B})$. ◀

B Proof of Theorem 18

Proof. Let $\mathsf{SomeLeakMon}$ be the set of all PFAs $\mathcal{A}$ such that there is an ϵ such that $(\mathcal{A}, \epsilon) \in \mathsf{LeakMon}$. We show that $\mathsf{SomeLeakMon}$ is $\mathbf{\Sigma_2^0}$-hard where $\mathbf{\Sigma_2^0}$ is the second level in the arithmetical hierarchy, which implies that $\mathsf{LeakMon}$ is not even recursively enumerable. As in the proof of Theorem 9, $\mathsf{ValueNot1} = \{\mathcal{A} \mid \mathcal{A}$ is a PFA and $\mathsf{value}(\mathcal{A}) < 1\}$ which is a $\mathbf{\Sigma_2^0}$-hard problem. We can conclude the theorem by reducing $\mathsf{ValueNot1}$ to $\mathsf{SomeLeakMon}$.

Our reduction, given a PFA $\mathcal{A} = (Q, q_s, \delta, Q_f)$ over Σ, constructs a PFA $\mathcal{B}$ such that $\mathsf{value}(\mathcal{A}) < 1$ iff $\mathcal{B} \in \mathsf{SomeLeakMon}$. Let $\mathcal{A} = (Q, q_s, \delta, Q_f)$ be any PFA over some alphabet Σ. Now, we define $\mathcal{B}$ as follows. If $\exists u \in \Sigma^*$ such that $\mathsf{P}_{\mathcal{A}}(u) = 1$ then $\mathcal{B}$ is simply the PFA $\mathcal{A}_3$ given in Figure 1c on page 7; observe that in this case, $\mathcal{A} \notin \mathsf{ValueNot1}$, and $\mathcal{B} \notin \mathsf{SomeLeakMon}$.

If there is no such a string u, then we define $\mathcal{B}$ to be a PFA over the alphabet Σ as follows. $\mathcal{B} = (Q \times \{1,2\}, (q_s, 1), \delta', Q_f \times \{1,2\})$ where $\delta'((q,i), a, (q',j)) = \frac{1}{2}\delta(q,a,q')$ for $q, q' \in Q, a \in \Sigma$ and $i, j \in \{1,2\}$.

Now, we make the following observations. For any $u \in \Sigma^*$, the acceptance probabilities of u by $\mathcal{A}$ and $\mathcal{B}$ are the same. Thus, $\mathsf{value}(\mathcal{B}) = \mathsf{value}(\mathcal{A})$. Furthermore, every accepting run of $\mathcal{B}$ on u has $|u|$ leaks. Using these observations, we shall show that $\mathcal{A} \in \mathsf{ValueNot1}$ iff $\mathcal{B} \in \mathsf{SomeLeakMon}$.

Suppose $\mathcal{A} \in \mathsf{ValueNot1}$. Then there must exist ϵ_0 such that $\mathsf{value}(\mathcal{B}) = \mathsf{value}(\mathcal{A}) < \epsilon_0 < 1$. As no word is accepted by $\mathcal{B}$ with probability $\geq \epsilon_0$, $\mathcal{B}$ is ϵ_0-leak monotonic with horizon $N_{\epsilon_0} = 0$.

Suppose $\mathcal{A} \notin \mathsf{ValueNot1}$. Then $\mathsf{value}(\mathcal{A}) = 1$. As there is no word accepted by $\mathcal{A}$ with probability 1 and Σ is finite, we get that there must be an infinite sequence of non-empty words $u_1, u_2, \ldots$ such that for each i, $|u_i| < |u_{i+1}|$ and $\mathsf{P}_{\mathcal{A}}(u_i) > 1 - \frac{1}{i}$. Suppose, for contradiction, $\mathcal{B} \in \mathsf{SomeLeakMon}$. This means that there must exist $\epsilon_0 \in (0,1)$ and N_{ϵ_0} such that $\mathcal{B}$ is ϵ_0-leak monotonic with horizon N_{ϵ_0}. Please note that as $\epsilon_0 < 1$, there must exist a j_0 such that $1 - \frac{1}{i} > \epsilon_0$ for all $i \geq j_0$. Fix $k = \max(N_{\epsilon_0}, j_0)$. Consider the word u_k. We have that $|u_k| \geq k \geq N_{\epsilon_0}$ and every run of $\mathcal{B}$ on u_k has exactly $|k|$ leaks. As N_{ϵ_0} is a horizon of ϵ_0-leak monotonicity we must have $\mathsf{P}_{\mathcal{A}}(u_k) < \epsilon_0$. This contradicts the fact that $\mathsf{P}_{\mathcal{A}}(u_k) = 1 - \frac{1}{k} > \epsilon_0$. ◀

C Proof of Theorem 21

Proof. For $i > 0$, $q \in Q_i$, let $\mathcal{A}_{i,q}$ be the restriction of $\mathcal{A}$ to the set Q_i of states with starting state q. For any $\epsilon \in (0,1)$, let $N_\epsilon > 0$ be a constant such that, for each $i > 0$, $q \in Q_i$ and each $u \in \Sigma^*$, the measure of the set of accepting runs of $\mathcal{A}_{i,q}$ on u, having at least N_ϵ leaks, is less than ϵ. Such a constant N_ϵ exists since each $\mathcal{A}_{i,q}$ is leak monotonic. Now let p be the minimum of the probabilities of reaching a state in $Q \setminus Q_0$, from any state in Q_0, on any input string of length exactly m, where m is the constant specified in the theorem. Clearly $p > 0$. Now, fix an $\epsilon \in (0,1)$. We specify a constant M_ϵ such that on every $u \in \Sigma^*$, the measure of the set of accepting runs of $\mathcal{A}$ on u, having at least M_ϵ leaks, is less than ϵ. Let n' be the smallest integer such that $(1-p)^{n'} < \frac{\epsilon}{2}$ and let $L_{\frac{\epsilon}{2}} = m \cdot n'$. Observe that for any $u \in \Sigma^*$ of length at least $L_{\frac{\epsilon}{2}}$, $\delta_u(q_s, Q_0) < \frac{\epsilon}{2}$, i.e., the probability that $\mathcal{A}$ is in some state in Q_0 after u is $< \frac{\epsilon}{2}$.

Now, let $M_\epsilon = L_{\frac{\epsilon}{2}} + N_{\frac{\epsilon}{2}}$. We show that M_ϵ satisfies the desired property. Now, consider any input string $u \in \Sigma^*$. If $|u| < M_\epsilon$ then the measure of the set of all runs of $\mathcal{A}$ on u having at least M_ϵ leaks is zero. So, assume that $|u| \geq M_\epsilon$. Let u_1 be the prefix of u of length $L_{\frac{\epsilon}{2}}$ and $u_2 \in \Sigma^*$ be the suffix of u following u_1, i.e., $u = u_1 u_2$. For any $i > 0$, $q \in Q_i$, let p_q be the probability measure of the set of all runs of $\mathcal{A}_{i,q}$, on input u_2, having at least $N_{\frac{\epsilon}{2}}$ leaks. Observe that $p_q < \frac{\epsilon}{2}$. Now, we see that the probability measure of the set of all accepting runs of $\mathcal{A}$ on u, having at least M_ϵ leaks, is bounded by $\frac{\epsilon}{2} + \sum_{q \in Q \setminus Q_0} \delta_{u_1}(q_s, q) \cdot p_q$. In the above expression, the first term in the sum bounds the probability of all such runs that remain entirely with in Q_0 and the second term bounds the probability of all such runs that end in a state outside Q_0. Since $p_q < \frac{\epsilon}{2}$ for $q \in Q \setminus Q_0$ and since $\sum_{q \in Q \setminus Q_0} \delta_{u_1}(q_s, q) \leq 1$, we see that the probability measure of the set of all accepting runs of $\mathcal{A}$ on u, having at least M_ϵ leaks, is less than ϵ. ◀

D Proof of Theorem 26

Proof. The theorem is an easy consequence of Theorem 25, Theorem 16 and the following claim:

▶ **Claim.** *Every k-HPA $\mathcal{A}$ with n-states and $k > 0$, has leak complexity at most $n^k \ell^{k-1}$.*

Proof. We prove this claim by induction on k. The base case is when $k = 1$. In this case, any accepting run that has ℓ leaks, either completely stays at level 0 or goes from a level 0 state to a higher level state making a non-leaky transition, or it goes to a level 1 state exactly after the ℓ^{th} leak (this is so because there can not be any leaks from level 1 states). Clearly, there can be at most m such runs that end in a level 1 accepting state, where m is the number of level 1 states. Thus, the total number of such runs can be at most $1 + m \leq n$, which is a constant independent of ℓ.

Now, assume that the claim is true for any $k > 0$. We show that that claim holds for $(k+1)$-HPA as well. Consider a $(k+1)$-HPA $\mathcal{A}$ on an input alphabet Σ. Let m be the total number of states at levels 1 and higher. Consider an input $u \in \Sigma^*$. Let X be the set of accepting runs of $\mathcal{A}$ on an input u, having $\ell > 0$ leaks. Let ℓ' be the maximum of the number of leaks from a level 0 state in any of the runs in X. Observe that $\ell' \leq \ell$. The set X can be partitioned into $\ell' + 1$ disjoint sets $X_b, X_1, ..., X_{\ell'}$, where X_b is a singleton consisting of the run that stays at level 0 or transitions from a level 0 state to a higher level state using a non-leaky transition, and X_i are the set of runs that made a transition from a level 0 state to a higher level state on the i^{th} leak, for $1 \leq i \leq \ell'$. For each i, $1 \leq i \leq \ell'$, let u_i be the prefix of the input after which the i^{th} leak occurred, and v_i be the suffix of u following u_i. All runs in X_i have the same prefix, say ρ_i, until the level 0 state from which the i^{th} leak occurred and they transition to one or more of the m higher level states after this leak. Thus, we can partition X_i into $m_i \leq m$ disjoint sets $X_{i,1}, ..., X_{i,m_i}$ such that all runs in $X_{i,j}$ transition to the same higher level state, say $q_{i,j}$, after the i^{th} leak, which is immediately after ρ_i. Now $X_{i,j}$ is simply the set of runs having prefix ρ_i followed by the set $X'_{i,j}$ of all accepting runs of $\mathcal{A}$ starting from the state $q_{i,j}$ on the input v_i and having $\ell - i$ leaks. Since $q_{i,j}$ is a higher level state, the restriction of $\mathcal{A}$ having $q_{i,j}$ as a start state is a k'-HPA for some $k' \leq k$. Now by the induction hypothesis, we see that the number of runs in $X'_{i,j}$ and hence those in $X_{i,j}$ is bounded by $n^k \cdot (\ell - i)^{k-1}$. From this we see that the number of runs in X_i is bounded by $m \cdot n^k \cdot (\ell - i)^{k-1}$. From this we see that $|X| \leq 1 + \sum_{1 \leq i \leq \ell'} m \cdot n^k \cdot (\ell - i)^{k-1}$. Since $\ell' \leq \ell$, we get $|X| \leq 1 + m \cdot n^k \cdot \ell^k \leq n^{k+1}\ell^k$. ◀

The Theorem follows. ◀

E Proof of Lemma 32

Proof. Fix u and ℓ. Let v be the word of the shortest length such that $|\mathsf{accruns}(\mathcal{A}, v, \ell)| \geq |\mathsf{accruns}(\mathcal{A}, u, \ell)|$. We will show that length of v is $\leq m + ((\ell + 1)m)^m$. Observe that the set of finite non-empty prefixes of $\mathsf{accruns}(\mathcal{A}, v, \ell)$ can be arranged as a tree $\mathbb{T}$ as follows. The initial state q_s is the root of the tree. If ρq is a prefix of some run in $\mathsf{accruns}(\mathcal{A}, v, \ell)$ then ρq is a child of ρ. Attach to each node ρ of $\mathbb{T}$, two labels: a state label $st(\rho)$ which is the last state of ρ and a leak label $lk(\rho)$ which is the number of leaks in ρ. For each depth i, let c_i be the set of nodes at depth i. We say that a leak occurs at node ρ if there is a state q' such that $\rho q'$ is in the tree $\mathbb{T}$ and $lk(\rho q') = lk(\rho) + 1$. Observe that if there is a leak at a node ρ at depth i with state label ρ then there is a leak at every node ρ' at depth i with state label q. We say that a leak occurs at depth i if a leak occurs at some node $\rho \in c_i$. We show that leaks in $\mathbb{T}$ cannot be too far apart.

▶ **Claim.** *Let $i, j \leq |v|$ be such that $j - i > m^m$ then there is a $i \leq k \leq j$ such that a leak occurs at depth k.*

Proof. We proceed by contradiction. Assume that there are i and j with $j-i > m^m$ with no leak occurring at any depth k between i and j. Consider any node $\rho \in c_i$. By our assumption, for each $i \leq k \leq j$, there is a *unique* descendant of ρ_k of ρ. The leak label of ρ_k is exactly the leak label of ρ. Furthermore, for any two nodes ρ and ρ' of c_i with the same state labels, the state labels of ρ_k and ρ'_k are exactly the same. From this, it is easy to see that there are k_1 and k_2 with $i \leq k_1 < k_2 \leq j$ such that for each node ρ of c_i, the state and leak labels of ρ_{k_1} and ρ_{k_2} are exactly the same. Let w be the string obtained from u by deleting the subword $u[k_1 + 1 : k_2]$ from v. It is easy to see that $\mathsf{accruns}(\mathcal{A}, w, \ell) \geq \mathsf{accruns}(\mathcal{A}, v, \ell)$ contradicting the minimality of v. ◀

A similar argument shows that there must be an $i \leq m$ such that there is a leak at depth i. Thus, we can conclude the Lemma if we can show that there are at most $(\ell + 1)^m$ depths at which a leak can occur; this is so due to the fact that the first depth at which a leak occurs is in the first m input symbols, and there are at most $(\ell + 1)^m$ depths at which leaks can occur and there are at most m^m input symbols between two successive such depths.

▶ **Claim.** *There are at most $(\ell + 1)^m$ depths at which a leak can occur.*

Proof. For each depth i, we define a function $\mathsf{sml}_i : Q \to \{\bot, 1, 2, \ldots, \ell\}$ as follows

$$\mathsf{sml}_i(q) = \begin{cases} \bot & \text{if } \{\rho \mid \rho \in c_i, st(\rho) = q, lk(\rho) > 0\} = \emptyset \\ n & \text{if } n = \min\{j > 0 \mid \exists \rho \in c_i, st(\rho) = q \text{ and } lk(\rho) = j\} \end{cases}.$$

Since there are only $(\ell + 1)^m$ possible functions sml_i, it suffices to show that for any two depths $i < j$ such that there is a leak at some depth $i \leq k < j$, we have that $\mathsf{sml}_i \neq \mathsf{sml}_j$. Observe that if there is no leak up-to depth i, then the latter is trivially true. So, we consider the case when there has been at least one leak before depth i.

To each depth j such that there is a leak before depth j, we associate an integer $1 \leq \mathsf{level}_j \leq \ell + 1$. If there is no leak at depth j, $\mathsf{level}_j = \ell + 1$. Otherwise level_j is the smallest integer $1 \leq r \leq \ell + 1$ such there is a leak at node ρ of c_j with leak label r.

Fix j such that there is a leak before depth j. We make the following two observations:

(a) For each $r < \mathsf{level}_j$, we have that $|\mathsf{sml}_j^{-1}(\{1, 2, \ldots, r\})| \geq |\mathsf{sml}_{j+1}^{-1}(\{1, 2, \ldots, r\})|$. This follows from the fact that there is a surjection g from the set $\mathsf{sml}_j^{-1}(\{1, 2, \ldots, r\})$ to the set $\mathsf{sml}_{j+1}^{-1}(\{1, 2, \ldots, r\})$ defined as follows. Let $q \in \mathsf{sml}_j^{-1}(\{1, 2, \ldots, r\})$. The definition of sml implies that there is a unique state q' such that $\delta(q, v[j], q') = 1$. Let $g(q) = q'$. The function g is easily seen to be a surjection.

(b) If there is a leak at depth j then $|\mathsf{sml}_j^{-1}(\{1, 2, \ldots, \mathsf{level}_j\})| > |\mathsf{sml}_{j+1}^{-1}(\{1, 2, \ldots, \mathsf{level}_j\})|$. This can be concluded as follows. Let $A \subseteq \mathsf{sml}_j^{-1}(\{1, 2, \ldots, \mathsf{level}_j)\}$ be the set of states q such that there is no leak at any node $\rho \in c_j$ with state label q. Clearly A is a proper subset of $\mathsf{sml}_j^{-1}(\{1, 2, \ldots, \mathsf{level}_j\})$. We can again define a surjection g from A onto $|\mathsf{sml}_{j+1}^{-1}(\{1, 2, \ldots, \mathsf{level}_j\})|$ as in (a) above.

Now, let $i < j$ be such that such that there is a leak at some depth $i \leq k < j$. Let $r = \min(\mathsf{level}_t \mid i \leq t < j)$. Observations (a) and (b) above imply that $|\mathsf{sml}_i^{-1}(\{1, 2, \ldots, r\})| > |\mathsf{sml}_j^{-1}(\{1, 2, \ldots, r\})|$. Thus, $\mathsf{sml}_i \neq \mathsf{sml}_j$. ◀

This concludes the proof of the Lemma. ◀

F Eventually weakly ergodic PFAs are regular-approximable

We recall the formal definition of eventually weakly ergodic PFAs.

▶ **Definition 37.** A PFA $\mathcal{A} = (Q, \delta, q_s, Q_f)$ is said to be *eventually weakly ergodic* if there is a partition $Q_T, Q_1, \ldots, Q_r$ of Q and a natural number $\ell > 0$ such that the following conditions hold:

- For each word u of length ℓ, each $1 \le i \le r$ and state $q_i \in Q_i$, $\mathsf{post}(q_i, u) \subseteq Q_i$.
- For each word u of length ℓ and each $1 \le i \le r$, there exists a state $q_i^u \in Q_i$ such that $q_i^u \in \mathsf{post}(q_i, u)$ for each $q_i \in Q_i$.
- For each word u of length ℓ and each state $q \in Q_T$, $\mathsf{post}(q, u) \cap (\cup_{1 \le j \le r} Q_j) \neq \emptyset$.

It is shown in [10] that the acceptance probability of each word u can be approximated by a short word v. In order to describe this result, we recall the following definition from [10]:

▶ **Definition 38.** Given an alphabet Σ and natural numbers $\ell, \ell' > 0$ such that ℓ' divides ℓ, let $\mathsf{c}_{(\ell,\ell')} : \Sigma^* \to \Sigma^*$ be defined as follows.

$$\mathsf{c}_{(\ell,\ell')}(u) = \begin{cases} u & \text{if } |u| < \ell' + 2\ell; \\ u_0 u_1 v_1 & \text{if } u = u_0 u_1 w v_1, |u_0| < \ell', |u_1| = \ell, w \in (\Sigma^{\ell'})^+ \text{ and } |v_1| = \ell \end{cases}.$$

▶ Remark. Observe that $\mathsf{c}_{(\ell,\ell')}(\cdot)$ is well defined. If $|u| \ge \ell' + 2\ell$ then there are unique u_0, u_1, w, v_1 such that $u = u_0 u_1 w v_1, |u_0| < \ell', |u_1| = \ell, w \in (\Sigma^{\ell'})^+, |v_1| = \ell$.

The following is shown in [10].

▶ **Lemma 39.** *Given an eventually weakly ergodic* PFA $\mathcal{A} = (Q, \delta, q_s, Q_f)$ *and* $y > 0$, *there are* $\ell > 0$ *and* $\ell' > 0$ *s.t.* ℓ' *divides* ℓ *and*

$$\forall u \in \Sigma^*. \ |\mathsf{P}_{\mathcal{A}}(u) - \mathsf{P}_{\mathcal{A}}(\mathsf{c}_{(\ell,\ell')}(u))| < \frac{y}{2}.$$

Furthermore, if y is rational then ℓ, ℓ' can be computed from $\mathcal{A}$ and y.

Given x, y, Lemma 39 can be used to show that an eventually weakly ergodic PFA $\mathcal{A}$ is regular-approximable with respect to (x, y). The proof proceeds as follows. First, we compute ℓ', ℓ as given in Lemma 39. Next, we construct a regular language L that approximates $\mathsf{L}_{>x}(\mathcal{A})$ as follows. L is the union of two regular languages L_{short} and L_{long}. $\mathsf{L}_{short} = \{u \in \Sigma^* \mid |u| < \ell' + 2\ell, \mathsf{P}_{\mathcal{A}}(u) > x\}$. It is easy to see that L_{short} is finite and hence regular.

We construct L_{long} by constructing a NFA $\mathcal{B}$ that recognizes L_{long}. The set of states of $\mathcal{B}$ is a union of four sets Q_0, Q_1, Q_2, Q_3 defined as follows:

- $Q_0 = \{u_0 \in \Sigma^* \mid |u_0| < \ell'\}$.
- $Q_1 = \{(u_0, u_1) \in \Sigma^* \mid |u_0| < \ell', |u_1| \le \ell\}$.
- $Q_2 = \emptyset$ if $\ell' = 1$ else $Q_2 = \{(u_0, u_1, i) \in \Sigma^* \mid |u_0| < \ell', |u_1| = \ell, 1 \le i \le \ell' - 1\}$.
- $Q_3 = \{(u_0, u_1, v_1) \in \Sigma^* \mid |u_0| < \ell', |u_1| = \ell, |v_1| \le \ell\}$.

The transition relation of $\mathcal{B}$ is defined as follows. For each input symbol a:

- For each $u_0 \in Q_0$, there is a transition from u_0 to $(u_0, a) \in Q_1$ on a. Furthermore, there is also a transition from u_0 to $u_0 a$ if $|u_0 a| < \ell'$.
- For each $(u_0, u_1) \in Q_1$ such that $|u_1| < \ell$, there is a transition from (u_0, u_1) to $(u_0, u_1 a)$ on a.
- For each $(u_0, u_1) \in Q_1$ such that $|u_1| = \ell$, there are two transitions on input a:
  1. There is a transition to $(u_0, u_1, a) \in Q_3$ on a.
  2. If $\ell' = 1$ then there is a transition from (u_0, u_1) to itself on a. If $\ell' > 1$ then there is a transition from (u_0, u_1) to $(u_0, u_1, 1) \in Q_2$ on a.

- For each $(u_0, u_1, i) \in Q_2$ such that $i < \ell' - 1$, there is a transition to $(u_0, u_1, i+1) \in Q_2$ on input a.
- For each $(u_0, u_1, \ell' - 1) \in Q_2$, there is a transition to $(u_0, u_1) \in Q_1$ on input a.
- For each $(u_0, u_1, v_1) \in Q_3$ such that $|v_1| < \ell$, there is a transition to $(u_0, u_1, v_1 a) \in Q_3$ on input a.
- There are no other transitions of $\mathcal{B}$.

The initial state of $\mathcal{B}$ is the empty string λ. The set of final states of $\mathcal{B}$ is the set:

$$\{(u_0, u_1, v_1) \in Q_3 \mid |v_1| = \ell, \mathsf{P}_{\mathcal{A}}(u_0 u_1 v_1) \geq x + \frac{y}{2}\}.$$

Thanks to Lemma 39, it is easy to see that $\mathsf{L}_{\geq x+y}(\mathcal{A}) \subseteq \mathsf{L} \subseteq \mathsf{L}_{>x}(\mathcal{A})$.