# Decision Questions for Probabilistic Automata on Small Alphabets

**Paul C. Bell**
Department of Computer Science, Liverpool John Moores University, UK

**Pavel Semukhin**
Department of Computer Science, University of Oxford, UK

## Abstract

We study the emptiness and $\lambda$-reachability problems for unary and binary Probabilistic Finite Automata (PFA) and characterise the complexity of these problems in terms of the degree of ambiguity of the automaton and the size of its alphabet. Our main result is that emptiness and $\lambda$-reachability are solvable in EXPTIME for polynomially ambiguous unary PFA and if, in addition, the transition matrix is over $\{0, 1\}$, we show they are in NP. In contrast to the Skolem-hardness of the $\lambda$-reachability and emptiness problems for exponentially ambiguous unary PFA, we show that these problems are NP-hard even for finitely ambiguous unary PFA. For binary polynomially ambiguous PFA with commuting transition matrices, we prove NP-hardness of the $\lambda$-reachability (dimension 9), nonstrict emptiness (dimension 37) and strict emptiness (dimension 40) problems.

**2012 ACM Subject Classification** Theory of computation → Formal languages and automata theory; Theory of computation → Computability; Theory of computation → Probabilistic computation

**Keywords and phrases** Probabilistic finite automata, unary alphabet, emptiness problem, bounded ambiguity

**Digital Object Identifier** 10.4230/LIPIcs.MFCS.2021.15

**Related Version** *Full Version*: https://arxiv.org/abs/2105.10293

## 1 Introduction

There are many possible extensions of the fundamental notion of a nondeterministic finite automaton. One such notion is that of a Probabilistic Finite Automata (PFA) which was first introduced by Rabin [19]. In a PFA $\mathcal{P}$ over a (finite) input alphabet $\Sigma$ the outgoing transitions from a state, for each input letter of $\Sigma$, form a probability distribution, as does the initial state vector. Thus, a word $w \in \Sigma^*$ is accepted with a certain probability, which we denote $\mathcal{P}(w)$.

There are a variety of interesting questions that one may ask about a PFA $\mathcal{P}$ over an alphabet $\Sigma$. In this article we focus on two decision questions, that of *$\lambda$-reachability* and *emptiness*. The $\lambda$-reachability problem is stated thus: given a probability $\lambda \in [0, 1]$, does there exist some word $w \in \Sigma^*$ such that $\mathcal{P}(w) = \lambda$? In the (strict) emptiness problem, we ask if there exists *any* word $w \in \Sigma^*$ such that $P(w) \geq \lambda$ (resp. $P(w) > \lambda$). We also mention the related *cutpoint isolation* problem – to determine if for each $\epsilon > 0$, there exists a word $w \in \Sigma$ such that $|\mathcal{P}(w) - \lambda| < \epsilon$.

In general, the emptiness problem is undecidable for PFA [18], even over a binary alphabet when the automaton has 25 states [12]. The cutpoint isolation problem is undecidable [4] even for PFA with 420 states over a binary alphabet [5]. The problem is especially interesting given the seminal result of Rabin that if a cutpoint $\lambda$ is isolated, then the cutpoint language associated with $\lambda$ is necessarily regular [19].

We may ask which restrictions of PFA may lead to decidability of the previous problems. In this paper we are interested in PFA of *bounded ambiguity*, where the ambiguity of a word denotes the number of accepting runs of that word in the PFA. A PFA $\mathcal{P}$ is $f$-ambiguous,

for a function $f : \mathbb{N} \to \mathbb{N}$, if every word of length $n$ has at most $f(n)$ accepting runs. A run is *accepting* if the probability of that run ending in a final state is strictly positive. The degree of ambiguity is thus a property of the NFA underlying a PFA (i.e. the NFA produced by setting all nonzero transition probabilities to 1). We may consider the notions of finite, polynomial or exponential ambiguity of $\mathcal{P}$ based on whether $f$ is bounded by a constant, is a polynomial or else is exponential, respectively. Characterisations of the degree of ambiguity of NFA are given by Weber and Seidel [23].

The authors of [8] show that emptiness for PFA remains undecidable even for polynomially ambiguous automata (quadratic ambiguity), show PSPACE-hardness results for finitely ambiguous PFA and that emptiness is in NP for the class of $k$-ambiguous PFA for every $k > 0$. The emptiness problem for PFA was later shown to be undecidable for linearly ambiguous automata [7].

Another restriction is to constrain input words of the PFA to come from a given language $\mathcal{L}$. If $\mathcal{L}$ is a *letter-bounded* language, then the emptiness and $\lambda$-reachability problems remain undecidable for polynomially ambiguous PFA, even when all transition matrices commute [2]. In contrast, the *cutpoint-isolation* problem is decidable even for exponentially ambiguous PFA when inputs are constrained to come from a given letter-bounded context-free language, although it is NP-hard for 3-state PFA on letter-bounded inputs [3].

Our main results are as follows. We show that the $\lambda$-reachability and emptiness problems for probabilistic finite automata are:

- In EXPTIME for the class of polynomially ambiguous unary PFA and are NP-complete if, in addition, the transition matrix is over $\{0, 1\}$ [Theorem 4 and Corollary 11].
- NP-hard for polynomially ambiguous PFA over a binary alphabet with *fixed* and *commuting* transition matrices of dimension 40 (strict emptiness problem), 37 (nonstrict emptiness problem) and 9 ($\lambda$-reachability problem) [Theorem 12].

We also show NP-hardness for the class of finitely ambiguous unary PFA with $\{0, 1\}$ transition matrix [Theorem 10]. Our hardness results rely on the NP-hardness of solving binary quadratic equations and the universality problem for unary regular expressions. An interesting question, that is left open, is to find out the exact computational complexity of the above problems in the case of polynomially ambiguous unary PFA, i.e. to close the gap between the EXPTIME upper bound and NP lower bound.

## 2 Probabilistic Finite Automata and Notation

We denote by $\mathbb{Q}^{n\times n}$ the set of all $n \times n$ matrices over $\mathbb{Q}$. Given two column vectors $u \in \mathbb{Q}^n$ and $v \in \mathbb{Q}^m$, we denote by $[u|v]$ the column vector $(u_1, \ldots, u_n, v_1, \ldots, v_m)^T \in \mathbb{Q}^{n+m}$. For a sequence of vectors $u_1, u_2, \ldots, u_k$, we write $[u_1|u_2|\cdots|u_k]$ for the column vector which stacks the vectors on top of each other.

Given $A = (a_{ij}) \in \mathbb{Q}^{m\times m}$ and $B \in \mathbb{Q}^{n\times n}$, we define the direct sum $A \oplus B$ and Kronecker product $A \otimes B$ of $A$ and $B$ by:

$$A \oplus B = \left[\begin{array}{c|c} A & \mathbf{0}_{m,n} \\ \hline \mathbf{0}_{n,m} & B \end{array}\right], \quad A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \cdots & a_{1m}B \\ a_{21}B & a_{22}B & \cdots & a_{2m}B \\ \vdots & \vdots & & \vdots \\ a_{m1}B & a_{m2}B & \cdots & a_{mm}B \end{bmatrix},$$

where $\mathbf{0}_{i,j}$ denotes the zero matrix of dimension $i \times j$. Note that neither $\oplus$ nor $\otimes$ are commutative in general. The following useful properties of $\oplus$ and $\otimes$ are well known.

▶ **Lemma 1.** *Let $A, B, C, D \in \mathbb{Q}^{n\times n}$. Then we have:*

- *Associativity: $(A\otimes B)\otimes C = A\otimes(B\otimes C)$ and $(A\oplus B)\oplus C = A\oplus(B\oplus C)$, thus $A\otimes B\otimes C$ and $A\oplus B\oplus C$ are unambiguous.*
- *Mixed product properties: $(A\otimes B)(C\otimes D) = (AC\otimes BD)$ and $(A\oplus B)(C\oplus D) = (AC\oplus BD)$.*
- *If $A$ and $B$ are stochastic matrices, then so are $A\oplus B$ and $A\otimes B$.*
- *If $A, B \in \mathbb{Q}^{n\times n}$ are both upper-triangular, then so are $A\oplus B$ and $A\otimes B$.*

See [13] for proofs of the first three properties of Lemma 1. The fourth property follows directly from the definition of the direct sum and Kronecker product and is not difficult to prove.

A Probabilistic Finite Automaton (PFA) $\mathcal{P}$ with $n$ states over an alphabet $\Sigma$ is defined as $\mathcal{P} = (u, \{M_a | a \in \Sigma\}, v)$ where $u \in \mathbb{Q}^n$ is the *initial probability distribution*; $v \in \{0,1\}^n$ is the *final state vector* and each $M_a \in \mathbb{Q}^{n\times n}$ is a (row) stochastic matrix. We will primarily be interested in *unary* and *binary* PFA, for which $|\Sigma| = 1$ and $|\Sigma| = 2$ respectively. For a word $w = a_1 a_2 \cdots a_k \in \Sigma^*$, we define the acceptance probability $\mathcal{P}(w) : \Sigma^* \to \mathbb{Q}$ of $\mathcal{P}$ as:

$$\mathcal{P}(w) = u^T M_{a_1} M_{a_2} \cdots M_{a_k} v \in [0,1],$$

which denotes the acceptance probability of $w$.[1]

For a given cutpoint $\lambda \in [0,1]$, we define the following languages: $L_{\geq\lambda}(\mathcal{P}) = \{w \in \Sigma^* \mid \mathcal{P}(w) \geq \lambda\}$, a nonstrict cutpoint language, and $L_{>\lambda}(\mathcal{P}) = \{w \in \Sigma^* \mid \mathcal{P}(w) > \lambda\}$, a strict cutpoint language. The (strict) emptiness problem for a cutpoint language is to determine if $L_{\geq\lambda}(\mathcal{P}) = \emptyset$ (resp. $L_{>\lambda}(\mathcal{P}) = \emptyset$). We are also interested in the *$\lambda$-reachability* problem, for which we ask if there exists a word $w \in \Sigma^*$ such that $\mathcal{P}(w) = \lambda$.

## 2.1 PFA Ambiguity

The degree of ambiguity of a finite automaton is a structural parameter, roughly indicating the number of accepting runs for a given input word. See [23] for a thorough discussion of ambiguity for nondeterministic automata and [2, 3, 7, 8] for connections to PFA.

Let $w \in \Sigma^*$ be an input word of an NFA $\mathcal{N} = (Q, \Sigma, \delta, Q_I, Q_F)$, with $Q$ the set of states, $\Sigma$ the input alphabet, $\delta \subset Q \times \Sigma \times Q$ the transition function, $Q_I$ the set of initial states and $Q_F$ the set of final states. For each $(p, w, q) \in Q \times \Sigma^* \times Q$, define $\mathrm{da}_{\mathcal{N}}(p, w, q)$ as the number of paths for $w$ in $\mathcal{N}$ leading from state $p$ to $q$. The *degree of ambiguity* of $w$ in $\mathcal{N}$, denoted $\mathrm{da}_{\mathcal{N}}(w)$, is defined as the number of all *accepting paths* for $w$ (starting from an initial and ending in a final state). The *degree of ambiguity* of $\mathcal{N}$, denoted $\mathrm{da}(\mathcal{N})$, is the supremum of the set $\{\mathrm{da}_{\mathcal{N}}(w) \mid w \in \Sigma^*\}$. $\mathcal{N}$ is called infinitely ambiguous if $\mathrm{da}(\mathcal{N}) = \infty$, finitely ambiguous if $\mathrm{da}(\mathcal{N}) < \infty$, and unambiguous if $\mathrm{da}(\mathcal{N}) \leq 1$. The *degree of growth* of the ambiguity of $\mathcal{N}$, denoted $\deg(\mathcal{N})$, is defined as the minimum degree of a univariate polynomial $h$ with positive integral coefficients such that for all $w \in \Sigma^*$, $\mathrm{da}_{\mathcal{N}}(w) \leq h(|w|)$ (if such a polynomial exists, in which case $\mathcal{N}$ is called polynomially ambiguous, otherwise the degree of growth is infinite and $\mathcal{N}$ is called exponentially ambiguous).

The above notions relate to NFA. We may derive an analogous notion of ambiguity for PFA by considering an embedding of a PFA $\mathcal{P}$ to an NFA $\mathcal{N}$ in such a way that for each letter $a \in \Sigma$, if the probability of transitioning from a state $i$ to state $j$ is nonzero under $\mathcal{P}$, then there is an edge from state $i$ to $j$ under $\mathcal{N}$ for letter $a$. The initial states of $\mathcal{N}$ are those of $\mathcal{P}$ having nonzero initial probability and the final states of $\mathcal{N}$ and $\mathcal{P}$ coincide. We then say that $\mathcal{P}$ is finitely/polynomially/exponentially ambiguous if $\mathcal{N}$ is (respectively).

[1] Some authors interchange the order of $u$ and $v$ and use column stochastic matrices, although the two definitions are trivially equivalent.

A state $q \in Q$ in an NFA (resp. PFA) is called *useful* if there exists an accepting path which visits $q$ (resp. an accepting path of nonzero probability which visits $q$). We can characterise whether an NFA $\mathcal{A}$ (and thus a PFA by the above embedding) has finite, polynomial or exponential ambiguity using the following properties:
**EDA** – There is a useful state $q \in Q$ such that, for some word $v \in \Sigma^*$, $da_{\mathcal{A}}(q, v, q) \geq 2$.
**IDA**$_d$ – There are useful states $r_1, s_1, \ldots, r_d, s_d \in Q$ and words $v_1, u_2, v_2, \ldots, u_d, v_d \in \Sigma^*$ such that for all $1 \leq i \leq d$, $r_i$ and $s_i$ are distinct and $(r_i, v_i, r_i), (r_i, v_i, s_i), (s_i, v_i, s_i) \in \delta$ and for all $2 \leq i \leq d$, $(s_{i-1}, u_i, r_i) \in \delta$.

▶ **Theorem 2** ( [14, 20, 23]). *An NFA (or PFA) $\mathcal{A}$ having the EDA property is equivalent to it being exponentially ambiguous. For any $d \in \mathbb{N}$, an NFA (or PFA) $\mathcal{A}$ having property IDA$_d$ is equivalent to $deg(\mathcal{A}) \geq d$.*

Clearly, if $\mathcal{N}$ agrees with IDA$_d$ for some $d > 0$, then it also agrees with IDA$_1, \ldots,$ IDA$_{d-1}$. An NFA (or PFA) is thus finitely ambiguous if it does not possess property IDA$_1$.

## 3 Unary PFA

Our main focus is on unary automata. We begin by giving a simple folklore proof that the $\lambda$-reachability and emptiness problems are as computationally difficult as the famous Skolem problem, which is only know to be decidable for instances of depth 4 [22]. See also [1] for connections to reachability problems for Markov chains.

▶ **Theorem 3.** *The $\lambda$-reachability and emptiness problems for unary exponentially ambiguous Probabilistic Finite Automata are Skolem-hard.*

**Proof.** *(Folklore).* The $\lambda$-reachability problem for unary exponentially ambiguous PFA can be shown Skolem-hard based on the well known matrix formulation of Skolem's problem [11] and Turakainen's technique showing the equivalence of (strict) cutpoint language acceptance of generalised automata and exponentially ambiguous probabilistic automata [21].

The emptiness problem can be shown Skolem-hard by encoding the positivity problem which is known to be Skolem-hard, see [17] for example. ◀

We now move to prove our main result, specifically that the emptiness and $\lambda$-reachability problems for polynomially ambiguous unary probabilistic finite automata are in EXPTIME. Note again that without the restriction of polynomial ambiguity the problem is Skolem-hard by Theorem 3 and thus not even known to be decidable.

▶ **Theorem 4.** *The $\lambda$-reachability and (strict) emptiness problems for unary polynomially ambiguous Probabilistic Finite Automata are decidable in EXPTIME.*

In order to establish Theorem 4, we need to prove a series of lemmas.

The next lemma states that we may consider a unary polynomially ambiguous PFA whose transition matrix is upper-triangular. This will prove useful since in that case the eigenvalues of the transition matrix are rational nonnegative. In general, a polynomially ambiguous unary PFA may have a transition matrix with complex eigenvalues. The proof of the lemma relies on the analysis of strongly connected components (SCCs) of the underlying transition graph of a PFA.

▶ **Lemma 5.** *Let $\mathcal{P} = (u, A, v)$ be a polynomially ambiguous unary Probabilistic Finite Automaton with acceptance function $\mathcal{P}(a^k) = u^T A^k v$. Then we can compute in EXPTIME a set of $d$ polynomially ambiguous unary PFAs $\{\mathcal{P}_s = (u_s, U, v') \mid 0 \leq s \leq d-1\}$ such that $U$ is rational upper-triangular and $\mathcal{P}(a^k) = \mathcal{P}_s(a^r) = u_s^T U^r v'$, where $k = rd + s$ with $0 \leq s \leq d-1$.*

**Proof.** We will identify $\mathcal{P}$ and its underlying graph in which an edge $(p, q)$ exists iff $A_{p,q} \neq 0$. Two states $p, q$ of a PFA are said to be *connected* if there exists a path from $p$ to $q$ and from $q$ to $p$. We partition the set of states into Strongly Connected Components (SCC) denoted $S_1, S_2, \ldots, S_\ell$ so that for any SCC $S_j$, either $|S_j| = 1$, or else any two states in $S_j$ are connected. These SCCs can be computed in linear time.

A polynomially ambiguous PFA does not have the EDA property (see Sec. 2.1). This implies that every $S_j$, with $|S_j| > 1$, consists of a single directed cycle, possibly with transitions to other SCCs. To see this, suppose there are two different directed cycles inside $S_j$ of lengths $m$ and $n$ and a common vertex $p$. Then one can construct two different paths of length $mn$ from $p$ to $p$ by going $m$ times along the first cycle and $n$ time along the second cycle, respectively, contradicting the assumption that $\mathcal{P}$ does not have the EDA property.

Note that if there exists a path from a state $p \in S_{j_1}$ to some $q \in S_{j_2}$, then there does not exist any path from any state in $S_{j_2}$ to a state in $S_{j_1}$, otherwise $S_{j_1}$ and $S_{j_2}$ would merge to a single SCC (since all vertices are then connected). This implies that the connected components $S_1, S_2, \ldots, S_\ell$ can be reordered in such a way that there are no transitions from $S_j$ to $S_i$ for $i < j$. Hence there exists a permutation matrix $P$ such that the following matrix is stochastic block upper-triangular:

$$B = PAP^{-1} = \begin{pmatrix} B_1 & * & \cdots & * \\ 0 & B_2 & \ddots & * \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & B_\ell \end{pmatrix},$$

such that each $B_j \in \mathbb{Q}^{d_j \times d_j}$, where $d_j$ is the size of $S_j$, and $B_j \preceq P_j$, where $P_j \in \mathbb{N}^{d_j \times d_j}$ is a permutation matrix, and the entries $*$ are arbitrary. Here $M \preceq N$ means that $M$ is entrywise less than $N$, i.e. $M_{i,j} \leq N_{i,j}$.

Let $d = \text{lcm}\{d_j \mid 1 \leq j \leq \ell\}$ (in fact, we can simply take $d = \prod_{j=1}^{\ell} d_j$). We then see that:

$$U := B^d = PA^dP^{-1} = \begin{pmatrix} B_1^d & * & \cdots & * \\ 0 & B_2^d & \ddots & * \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & B_\ell^d \end{pmatrix}.$$

Note that each $B_j^d \preceq P_j^d = I_j$, where $I_j \in \mathbb{N}^{d_j \times d_j}$ is the identity matrix, and the entries $*$ are arbitrary. Therefore, each $B_j^d$ is diagonal, and so $U$ is clearly upper-triangular.

We then define $\mathcal{P}_s = (u_s, U, v')$, for $0 \leq s \leq d-1$, with $u_s^T = u^T A^s P^{-1}$ and $v' = Pv$ noting that $Pv$ is a binary vector as required of a final state vector. We now see that:

$$\mathcal{P}(a^k) = u^T A^k v = u^T A^s A^{rd} v = u^T A^s P^{-1}(PA^{rd}P^{-1})Pv = u_s^T U^r v' = \mathcal{P}_s(a^r)$$

for $k = rd + s$ with $0 \leq s \leq d-1$ as required. Here we used the identity $U^r = PA^{rd}P^{-1}$.

Finally, note that $d$ can be exponential in the number of states of $\mathcal{P}$, which in turn is bounded by the input size. Hence computing $U$ and all $u_s$, for $0 \leq s \leq d-1$, takes exponential time. ◀

The next lemma gives us an efficient method to compute an explicit formula for the acceptance probability function of a unary PFA with upper-triangular transition matrix.

▶ **Lemma 6.** *Let $\mathcal{P} = (u, A, v)$ be a unary probabilistic finite automaton such that $A$ is rational upper-triangular, and let $\lambda_0 = 1 > \lambda_1 > \cdots > \lambda_m \geq 0$ be distinct eigenvalues of $A$. Then there exist a constant $c \in \mathbb{Q}$ and univariate polynomials $p_1, \ldots, p_m$ over $\mathbb{Q}$, all of which can be computed in polynomial time, such that*

$$\mathcal{P}(a^k) = c + \sum_{i=1}^{m} p_i(k)\lambda_i^k.$$

**Proof.** First, we write $A$ in *Jordan normal form* $A = S^{-1}JS$, where $S$ is a nonsingular ($\det(S) \neq 0$) matrix consisting of the generalised eigenvectors of $A$. Recall that $A$ is a rational upper-triangular matrix. It follows that $J$ and $S$ must have rational entries. Moreover, to compute $J$ and $S$, we need to solve systems of linear equations over $\mathbb{Q}$, which can be done in polynomial time. Computing $S^{-1}$ also requires polynomial time. Matrix $J$ has the form $J = \bigoplus_{i=0}^{m} \bigoplus_{j=1}^{n_i} J_{\ell_{i,j}}(\lambda_i)$, where $J_{\ell_{i,j}}(\lambda_i)$ is a $\ell_{i,j} \times \ell_{i,j}$ *Jordan block* and $n_i$ is the geometric multiplicity of $\lambda_i$ (hence $\sum_{j=1}^{n_i} \ell_{i,j}$ is the algebraic multiplicity of $\lambda_i$). Recall that a Jordan block $J_\ell(\lambda)$ of size $\ell \times \ell$ that corresponds to an eigenvalue $\lambda$ has the form:

$$J_\ell(\lambda) = \begin{pmatrix} \lambda & 1 & 0 & \cdots & 0 \\ 0 & \lambda & 1 & \cdots & 0 \\ 0 & 0 & \lambda & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \lambda \end{pmatrix} \in \mathbb{Q}^{\ell \times \ell}.$$

Noting that $\binom{x}{y} = 0$ if $y > x$, we see that

$$J_\ell(\lambda)^k = \begin{pmatrix} \lambda^k & \binom{k}{1}\lambda^{k-1} & \binom{k}{2}\lambda^{k-2} & \cdots & \binom{k}{\ell-1}\lambda^{k-(\ell-1)} \\ 0 & \lambda^k & \binom{k}{1}\lambda^{k-1} & \cdots & \binom{k}{\ell-2}\lambda^{k-(\ell-2)} \\ 0 & 0 & \lambda^k & \cdots & \binom{k}{\ell-3}\lambda^{k-(\ell-3)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \lambda^k \end{pmatrix}. \tag{1}$$

Note that the entries of $J_\ell(\lambda)^k$ have the form $q_{i,j}(k)\lambda^k$, where $q_{i,j}(k)$ are polynomials over $\mathbb{Q}$ that can be computed in polynomial time. Namely, $q_{i,i+p}(k) = \binom{k}{p}\lambda^{-p}$ for $0 \leq p \leq \ell - i$, and $q_{i,j}(k) = 0$ for $i > j$. Note that even though $p$ appears in the exponent of $\lambda^{-p}$ and as $p!$ in $\binom{k}{p}$, these values are still computable in PTIME from the input data because $p$ is bounded by the dimension of the matrix, which in turn is bounded by the input size.

Next, we note that $J^k = \bigoplus_{i=0}^{m} \bigoplus_{j=1}^{n_i} J_{\ell_{i,j}}(\lambda_i)^k$. Hence the entries of $J^k$ have the form $p_{s,t}(k)\lambda_i^k$, where $p_{s,t}(k)$ are polynomials over $\mathbb{Q}$. So we can write the function $\mathcal{P}(a^k)$ as follows:

$$\mathcal{P}(a^k) = u^T A^k v = (u^T S^{-1}) J^k (Sv).$$

Note that in the above equation, $u^T S^{-1}$ and $Sv$ are rational vectors. It follows that

$$\mathcal{P}(a^k) = \sum_{i=0}^{m} p_i(k)\lambda_i^k$$

for some polynomials $p_i(k)$ over $\mathbb{Q}$. In fact, these polynomials are rational linear combinations of those $p_{s,t}(k)$ that multiply $\lambda_i^k$ in the expression for $J^k$, and so they can be computed in polynomial time.

Finally, recall that $\lambda_0 = 1$ and note that the Jordan blocks that correspond to the dominant eigenvalues of a stochastic matrix have size $1 \times 1$ (for the proof of this fact see, e.g. [9, Theorem 6.5.3]). It follows from (1) that the terms $\lambda_0^k$ in the formula for $J^k$ are multiplied by constant polynomials $p_{s,t}(k) = 1$. Hence $p_0(k) = c$ for some constant $c \in \mathbb{Q}$. ◀

The next technical lemma is crucial in our later analysis of the running time of the algorithms for the emptiness and $\lambda$-reachability problems presented in Lemmas 8 and 9.

▶ **Lemma 7.** *Let $D \in \mathbb{R}$ be such that $\ln D > 2$. Then for all $x > 3D \ln D$, we have $D \ln x < x$.*

**Proof.** Our goal is to find $x_0 > 0$ such that every $x > x_0$ satisfies $D \ln x < x$. First, let us make a substitution $x = Dt$, where $t > 1$. Then we can rewrite $D \ln x < x$ as follows

$$\begin{aligned} &D \ln(Dt) < Dt, \\ &\ln t + \ln D < t. \end{aligned}$$

We want to find $t_0 > 1$ such that every $t > t_0$ satisfies $\ln t + \ln D < t$. Let us make another substitution $t = \ln D + u \ln\ln D$, where $u > 0$. Then we can write the previous inequality as

$$\begin{aligned} &\ln(\ln D + u \ln\ln D) + \ln D < \ln D + u \ln\ln D, \\ &\ln\left(\ln D\left(1 + u\frac{\ln\ln D}{\ln D}\right)\right) < u \ln\ln D, \\ &\ln\ln D + \ln\left(1 + u\frac{\ln\ln D}{\ln D}\right) < u \ln\ln D. \end{aligned} \tag{2}$$

So we need to find $u_0 > 0$ such that for all $u > u_0$, the inequality (2) holds. In order to do this, we can replace the left-hand side of (2) with a larger value using $\ln\left(1 + u\frac{\ln\ln R}{\ln R}\right) < u\frac{\ln\ln R}{\ln R}$. Thus we obtain

$$\ln\ln D + u\frac{\ln\ln D}{\ln D} < u \ln\ln D,$$

$$1 + \frac{u}{\ln D} < u, \qquad \ln D + u < u \ln D, \qquad \frac{\ln D}{\ln D - 1} < u.$$

Recall that by our assumption $\ln D > 2$. In this case, $\frac{\ln D}{\ln D - 1} < 2$, and hence we can choose $u_0 = 2$. This gives us the values $t_0 = \ln D + u_0 \ln\ln D = \ln D + 2\ln\ln D$ and $x_0 = Dt_0 = D(\ln D + 2\ln\ln D)$. Since $\ln\ln D < \ln D$, we can choose $x_0$ to be $x_0 = 3D \ln D$. ◀

We now proceed to the proof of our main result. We split the analysis into two cases depending on whether or not the cutpoint $\lambda$ coincides with the limit $\lim_{k\to\infty} \mathcal{P}(a^k)$, which is unique by Lemma 6.

▶ **Lemma 8.** *Let $\mathcal{P} = (u, A, v)$ be a unary probabilistic finite automaton such that $A$ is rational upper-triangular, and let $\lambda \in [0,1] \cap \mathbb{Q}$ be a cutpoint. Assuming that $\lambda \neq \lim_{k\to\infty} \mathcal{P}(a^k)$, the (strict) emptiness and $\lambda$-reachability problems for $\mathcal{P}$ and $\lambda$ are decidable in EXPTIME.*

**Proof.** By Lemma 6, we can write $\mathcal{P}(a^k) = c + \sum_{i=1}^m p_i(k)\lambda_i^k$, where $1 > \lambda_1 > \cdots > \lambda_m$ are the eigenvalues of $A$ and $c$ and the coefficients of $p_i$ are rational numbers that can be computed in polynomial time. By assumption, $\lim_{k\to\infty} \mathcal{P}(a^k) = c \neq \lambda$. Let $\epsilon = \frac{|c-\lambda|}{2}$. We now determine a natural number $k_0 > 0$ such that $\mathcal{P}(a^k) \in (c - \epsilon, c + \epsilon)$ for all $k > k_0$.

Let each $p_i(k)$ have the form $p_i(k) = a_{i,s}k^s + a_{i,s-1}k^{s-1} + \cdots + a_{i,0}$, where $s \leq n$ is the size of the largest Jordan block in the Jordan normal form of $A$ (we do not assume here that $a_{i,s} \neq 0$). Then for all $k > 0$ we have

$$\left|\sum_{i=1}^{m} p_i(k)\lambda_i^k\right| \leq \lambda_1^k \sum_{i=1}^{m} |p_i(k)| \leq \lambda_1^k k^s \sum_{i=1}^{m}\sum_{j=0}^{s} |a_{i,j}| = d\, k^s \lambda_1^k,$$

where $d = \sum_{i=1}^{m}\sum_{j=0}^{s} |a_{i,j}| \in \mathbb{Q}$ can be computed in polynomial time by Lemma 6.

Let $k_1 > 0$ be a number to be defined later such that for all $k > k_1$,

$$k^s < \left(\frac{1}{\sqrt{\lambda_1}}\right)^k = \lambda_1^{-\frac{k}{2}}.$$

Then for all $k > k_1$, we have $d\, k^s \lambda_1^k < d\lambda_1^{\frac{k}{2}}$. Thus we need to find $k_0 \geq k_1$ such that for all $k > k_0$, we have $\lambda_1^{\frac{k}{2}} < \epsilon/d$. Note that if $\epsilon/d \geq 1$, then we can take $k_0 = k_1$. Hence we assume that $\epsilon/d < 1$.

The inequality $\lambda_1^{\frac{k}{2}} < \epsilon/d$ is equivalent to $k \ln \lambda_1 < 2\ln(\epsilon/d)$. Since $\ln \lambda_1 < 0$, the previous inequality is equivalent to

$$k > \frac{2\ln(\epsilon/d)}{\ln \lambda_1} = \frac{2\ln(d/\epsilon)}{-\ln \lambda_1}. \tag{3}$$

To determine $k_0$, we need an upper bound on the right-hand side of (3). We will use the fact that for any rational $r > 1$, $\ln r < \log_2 r \leq \log_2 \lceil r \rceil < \mathrm{bins}(\lceil r \rceil)$, where $\mathrm{bins}(n)$ is the size of the binary representation of $n$. Thus $\mathrm{bins}(\lceil r \rceil)$ gives a polynomially computable integer upper bound for $\ln r$.

Next, using the fact that $\ln(1 + x) < x$ for $x \neq 0$, we obtain

$$\ln \lambda_1 = \ln(1 + (\lambda_1 - 1)) < \lambda_1 - 1,$$

which gives $-\ln \lambda_1 > 1 - \lambda_1$. Hence a polynomially computable upper bound on the right-hand side of (3) is

$$\frac{2\ln(d/\epsilon)}{-\ln \lambda_1} < \frac{2\,\mathrm{bins}(\lceil d/\epsilon \rceil)}{1 - \lambda_1}. \tag{4}$$

Next we compute a value $k_1$ such that for all $k > k_1$:

$$k^s < \lambda_1^{-\frac{k}{2}} \qquad \text{or, equivalently,} \qquad C \ln k < k, \tag{5}$$

where $C = \dfrac{2s}{-\ln \lambda_1}$. Using the fact that $\ln(1 + x) < x$ for $x \neq 0$, we obtain $C < \dfrac{2s}{1 - \lambda_1}$. Hence in order to find $k_1$, we can replace $C$ in (5) with $D = \dfrac{2s}{1 - \lambda_1}$. In addition, we can assume that $\ln D > 2$, since otherwise we can replace $D$ with a larger value that satisfies this condition, e.g. with $D = 9$. Now, Lemma 7 implies that every $k > 3D \ln D$ satisfies $D \ln k < k$. To make this value polynomially computable, we can choose it to be

$$k_1 = 3\lceil D \rceil \mathrm{bins}(\lceil D \rceil), \text{ where } D = \max\left\{\frac{2s}{1 - \lambda_1}, 9\right\}.$$

Finally, combining the right-hand side of (4) with the above formula, we can define

$$k_0 = \max\left\{\frac{2\,\mathrm{bins}(\lceil d/\epsilon\rceil)}{1-\lambda_1},\ 3\lceil D\rceil \mathrm{bins}(\lceil D\rceil)\right\}.$$

Note that all the values that appear in the above formula, e.g. $\epsilon$, $d$ and $D$, can be computed in polynomial time from the input data.

At this point we have derived a polynomially computable $k_0$ such that $\mathcal{P}(a^k) = u^T A^k v \in (c-\epsilon, c+\epsilon)$ and, in particular, $\mathcal{P}(a^k) \neq \lambda$ for all $k > k_0$. Now, to decide the $\lambda$-reachability problem, we need to check for each integer $k \in [0, k_0]$ whether $u^T A^k v = \lambda$. Note that the number of integers in $[0, k_0]$ is equal to $2^{\mathrm{bins}(k_0)}$, which is exponential in the instance size. Also, computing $A^k$ for a given $k \in [0, k_0]$ takes exponential time because $\mathrm{bins}(A^k) = \mathcal{O}(2^{\mathrm{bins}(k_0)}\mathrm{bins}(A))$. So, the overall algorithm is in EXPTIME.

In a similar way, we can decide the (strict) emptiness problem in EXPTIME. For instance, suppose $\lambda > c$. Then for all $k > k_0$, we have $\mathcal{P}(a^k) < c + \epsilon < \lambda$. Thus deciding whether there exists $k$ such that $\mathcal{P}(a^k) < \lambda$ is trivial. Suppose we want to know if there exists $k$ such that $\mathcal{P}(a^k) \geq \lambda$. In this case, we need to check for each integer $k \in [0, k_0]$ whether $u^T A^k v \geq \lambda$. By the same argument as before, this can be done in EXPTIME. ◀

▶ **Lemma 9.** *Let $\mathcal{P} = (u, A, v)$ be a unary polynomially ambiguous probabilistic finite automaton such that $A$ is upper-triangular and let $\lambda \in [0,1] \cap \mathbb{Q}$ be a cutpoint. Assuming that $\lambda = \lim_{k\to\infty} \mathcal{P}(a^k)$, the (strict) emptiness and $\lambda$-reachability problems for $\mathcal{P}$ and $\lambda$ are decidable in EXPTIME.*

**Proof.** Recall that by Lemma 6, we can write $\mathcal{P}(a^k) = c + \sum_{i=1}^m p_i(k)\lambda_i^k$, where $1 > \lambda_1 > \cdots > \lambda_m$ are the eigenvalues of $A$ and $c$ and the coefficients of $p_i$ are rational numbers computable in polynomial time. By our assumption, $\lambda = \lim_{k\to\infty} \mathcal{P}(a^k) = c$. As before, let each $p_i(k)$ have the form $p_i(k) = a_{i,s}k^s + a_{i,s-1}k^{s-1} + \cdots + a_{i,0}$, where $s \leq n$ (we do not assume here that $a_{i,s} \neq 0$).

In addition, assume that the leading coefficient of $p_1(k)$ is $a_{1,t}$, for some $t \leq s$. Without loss of generality, suppose $a_{1,t} > 0$; the case when $a_{1,t} < 0$ is similar. First, we compute $k_0$ such that $p_1(k) > \frac{1}{2}a_{1,t}k^t$ for all $k > k_0$. To do this, we will use the following inequalities:

$$a_{1,t}k^t + a_{1,t-1}k^{t-1} + \cdots + a_{1,0} > \frac{1}{2}a_{1,t}k^t \iff \frac{1}{2}a_{1,t}k^t + a_{1,t-1}k^{t-1} + \cdots + a_{1,0} > 0$$

$$\text{and } |a_{1,t-1}k^{t-1} + \cdots + a_{1,0}| \leq k^{t-1}(|a_{1,t-1}| + \cdots + |a_{1,0}|) = k^{t-1}\sum_{j=0}^{t-1}|a_{1,j}| \quad \text{if } k \geq 1.$$

So, the inequality $p_1(k) > \frac{1}{2}a_{1,t}k^t$ follows from $\frac{1}{2}a_{1,t}k^t > k^{t-1}\sum_{j=0}^{t-1}|a_{1,j}|$, which is equivalent to $k > \frac{2}{a_{1,t}}\sum_{j=0}^{t-1}|a_{1,j}|$. Therefore, we conclude that

$$p_1(k) > \frac{1}{2}a_{1,t}k^t \quad \text{for all } k \text{ such that } k > k_0 := \max\left\{1,\ \frac{2}{a_{1,t}}\sum_{j=0}^{t-1}|a_{1,j}|\right\}. \tag{6}$$

Now we want to find $k_1 \geq k_0$ such that for all $k > k_1$, we have

$$\lambda_1^k p_1(k) + \lambda_2^k p_2(k) + \cdots + \lambda_m^k p_m(k) > 0. \tag{7}$$

Note that

$$|\lambda_2^k p_2(k) + \cdots + \lambda_m^k p_m(k)| \leq \lambda_2^k(|p_2(k)| + \cdots + |p_m(k)|) \leq dk^s\lambda_2^k, \tag{8}$$

where $d = \sum_{i=2}^m \sum_{j=0}^s |a_{i.j}|$. Using (6) and (8), we see that (7) holds whenever $k > k_0$ and $dk^s\lambda_2^k < \frac{1}{2}a_{1,t}k^t\lambda_1^k$, which is equivalent to

$$\frac{2dk^{s-t}}{a_{1,t}} < \left(\frac{\lambda_1}{\lambda_2}\right)^k \qquad \text{or} \qquad \ln\frac{2d}{a_{1,t}} + (s-t)\ln k < k\ln\frac{\lambda_1}{\lambda_2}$$
$$\frac{1}{\ln\lambda_1/\lambda_2}\left(\ln\frac{2d}{a_{1,t}} + (s-t)\ln k\right) < k. \tag{9}$$

We will use the following inequality

$$\ln\frac{\lambda_1}{\lambda_2} = -\ln\frac{\lambda_2}{\lambda_1} = -\ln\left(1+\frac{\lambda_2-\lambda_1}{\lambda_1}\right) > -\frac{\lambda_2-\lambda_1}{\lambda_1} > \lambda_1-\lambda_2.$$

Then we can replace (9) with a stronger inequality

$$\frac{1}{\lambda_1-\lambda_2}\left(\ln\frac{2d}{a_{1,t}} + (s-t)\ln k\right) < k. \tag{10}$$

In the following, we will assume $t < s$ since otherwise (10) simplifies to $\frac{1}{\lambda_1-\lambda_2}\ln\frac{2d}{a_{1,t}} < k$. Let us make the substitution $k = t\left(\frac{2d}{a_{1,t}}\right)^{-\frac{1}{s-t}}$, where $t > 0$. Then (10) can be written as

$$\frac{1}{\lambda_1-\lambda_2}\left(\ln\frac{2d}{a_{1,t}} + (s-t)\ln t + (s-t)\frac{-1}{s-t}\ln\frac{2d}{a_{1,t}}\right) < t\left(\frac{2d}{a_{1,t}}\right)^{-\frac{1}{s-t}}$$
$$\left(\frac{2d}{a_{1,t}}\right)^{\frac{1}{s-t}}\frac{s-t}{\lambda_1-\lambda_2}\ln t < t.$$

Let $D = \max\left\{9,\ \left(\frac{2d}{a_{1,t}}\right)^{\frac{1}{s-t}}\frac{s-t}{\lambda_1-\lambda_2}\right\}$. Here 9 is needed to satisfy the requirement $\ln D > 2$ in Lemma 7. Then by Lemma 7, the above inequality holds when $t > 3D\ln D$. Therefore, (10) and hence (9) holds when $k > 3\left(\frac{2d}{a_{1,t}}\right)^{-\frac{1}{s-t}} D\ln D$. To make this bound polynomially computable, we can simplify it as follows. Suppose that $2d \geq a_{1,t}$. Then (9) holds when

$$k > k_1 := 3\lceil E\rceil \mathrm{bins}(\lceil E\rceil), \quad \text{where } E = \max\left\{9,\ \frac{2d}{a_{1,t}}\cdot\frac{s-t}{\lambda_1-\lambda_2}\right\}$$

because in this case $\left(\frac{2d}{a_{1,t}}\right)^{\frac{1}{s-t}} \leq \frac{2d}{a_{1,t}}$ and $\left(\frac{2d}{a_{1,t}}\right)^{-\frac{1}{s-t}} \leq 1$. On the other hand, if $2d < a_{1,t}$, then (9) holds when

$$k > k_1 := 3\left\lceil\frac{a_{1,t}}{2d}E\right\rceil \mathrm{bins}(\lceil E\rceil), \quad \text{where } E = \max\left\{9,\ \frac{s-t}{\lambda_1-\lambda_2}\right\}$$

because in this case $\left(\frac{2d}{a_{1,t}}\right)^{-\frac{1}{s-t}} < \left(\frac{2d}{a_{1,t}}\right)^{-1}$ and $\left(\frac{2d}{a_{1,t}}\right)^{\frac{1}{s-t}} < 1$.

Finally, we conclude that (7) holds for all $k > k_2 := \max\{k_0, k_1\}$, where both $k_0$ and $k_1$ are computable in PTIME. In other words, we obtained a polynomially computable value $k_2$ such that $\mathcal{P}(a^k) > c = \lambda$ for all $k > k_2$. Using the same argument as at the end of the proof of Lemma 8, we can show that the (strict) emptiness and $\lambda$-reachability problems are decidable in EXPTIME. ◀

We are now ready to give a proof of Theorem 4.

**Proof of Theorem 4.** Let $\mathcal{P} = (u, A, v)$ be a polynomially ambiguous unary PFA. By Lemma 5, we can compute in EXPTIME a set of $d$ polynomially ambiguous unary PFAs $\{\mathcal{P}_s = (u_s, U, v') \mid 0 \leq s \leq d-1\}$ such that $U$ is rational upper-triangular and

$$\mathcal{P}(a^{rd+s}) = \mathcal{P}_s(a^r) = u_s^T U^r v',$$

where $0 \leq s \leq d-1$.

Suppose $\lambda$ is a given cutpoint. If we want to decide whether there exists $k$ such that $\mathcal{P}(a^k) = \lambda$ (or $\mathcal{P}(a^k) \geq \lambda$), we can check for every $s$ from 0 to $d-1$ whether there exists $r$ such that $\mathcal{P}_s(a^r) = \lambda$ (or $\mathcal{P}_s(a^r) \geq \lambda$, respectively), which can be done in EXPTIME using Lemmas 8 and 9. Namely, we will use Lemma 8 if $\lambda \neq c_s$ and Lemma 9 if $\lambda = c_s$ for the current values of $s \in [0, d-1]$. Finally, we note that even though the value of $d$ can be exponential in the input size, the whole procedure can still be done in EXPTIME. ◀

Skolem's problem is at least NP-hard [6] implying that the $\lambda$-reachability and emptiness problems are also NP-hard, at least for PFA of exponential ambiguity. Our next result shows that NP-hardness can be established even for unary PFAs of *finite ambiguity*.

▶ **Theorem 10.** *The $\lambda$-reachability and emptiness problems for unary finitely ambiguous Probabilistic Finite Automata $\mathcal{P} = (u, A, v)$ with $\{0,1\}$-matrix $A$ are NP-hard.*

**Proof.** The NP-hardness of Skolem's problem was established in [6]. Specifically, Corollary 1.3 of [6] states that the problem of determining, for a given matrix $A \in \{0,1\}^{n \times n}$ and row vectors $b, c \in \{0,1\}^n$, if $b^T A^k c = 0$ for some $k \geq 0$ is NP-hard. Examination of the proof of this corollary shows that in fact $\mathcal{P}$ is finitely ambiguous as we shall show.

The proof of Theorem 1.1 of [6] shows a reduction of 3SAT on $m$ clauses with $n$ letters to a unary rational expression $E$ of the form:

$$E = \bigcup_{j=0}^{k} a^{z_j}(a^{r_j})^*,$$

where $k = \mathcal{O}(n^3 m)$ and $z_j, r_j = \mathcal{O}(n^6)$ as is not difficult to see from the proof in [6]. Notice then that each $z_j, r_j$ represented in unary has a polynomial size in terms of the 3SAT instance and thus $E$ also has a polynomial representation size.

We may then invoke Kleene's theorem [15] to state that the language recognised by $E$ is also recognised by an NFA $\mathcal{P} = (b, \{A\}, c)$ which thus allows the derivation of Corollary 1.3 of [6]. Note that $E$ is simply the union of rational expressions of the form $E_j = a^{z_j}(a^{r_j})^*$. Each $E_j$ can be transformed to an NFA $\mathcal{N}_j$ with $z_j + r_j + 1$ states $S_j = n_{0,j}, \ldots, n_{z_j,j}, n_{z_j+1,j}, \ldots, n_{z_j+r_j,j}$ with initial state $n_{0,j}$, final state $n_{z_j+1,j}$ and transition function $\delta : S_j \times \{a\} \to S_j$ given by $\delta(n_{i,j}, a) = n_{i+1,j}$ for $0 \leq i \leq z_j + r_j - 1$ and $\delta(n_{z_j+r_j}, a) = n_{z_j+1,j}$.

We may then form an NFA $\mathcal{N}$ by $\mathcal{N} = \bigcup_{j=0}^{k} \mathcal{N}_j$ with the usual construction. In this case, $\mathcal{N}$ has set of initial states $\{n_{0,j} \mid 1 \leq j \leq k\}$, set of final states $\{n_{z_j+1,j} \mid 1 \leq j \leq k\}$ and states in disjoint subsets $S_j$ and $S_{j'}$ with $j \neq j'$ are not connected. This implies by the IDA property of [23] that $\mathcal{N}$ is finitely ambiguous since there does not exist any state with two outgoing transitions (by which reasoning we also know that each row of $\mathcal{N}$'s transition matrix has exactly one entry 1 with all others 0). In fact one may see that $\mathcal{N}$ is $k$-ambiguous with $k = \mathcal{O}(n^3 m)$. The number of states of $\mathcal{N}$ is $d = \sum_{j=0}^{k} z_j + r_j + 1 = \mathcal{O}(n^9 m)$ which is polynomial in the 3SAT instance representation size.

We note that actually $\mathcal{N}$ is already close to a PFA. Since each row is zero except for exactly one entry 1, matrix $A$ is stochastic. We thus consider Probabilistic Finite Automaton $\mathcal{P} = (u, \{A\}, c)$ where $u = \frac{b}{|b|}$ is the initial (stochastic) vector. $\mathcal{P}$ has polynomial ambiguity

since $\mathcal{N}$ does. Therefore, deciding if there exists $k \geq 0$ such that $\mathcal{P}(a^k) = 0$ or $\mathcal{P}(a^k) \leq 0$ is NP-hard to determine, proving NP-hardness of the $\lambda$-reachability and emptiness problems. Since we did not modify $\mathcal{N}$ to derive $\mathcal{P}$ other than to scale the initial vector, the degree of ambiguity is retained. ◀

▶ **Corollary 11.** *The $\lambda$-reachability and emptiness problems for unary polynomially ambiguous PFA $\mathcal{P} = (u, A, v)$ with $\{0,1\}$-matrix $A$ are NP-complete.*

**Proof.** NP-hardness follows from Theorem 10 since finite ambiguity is a stronger property than polynomially ambiguity. To prove the NP upper bound, we will show that the algorithm in the proof of Theorem 4 can be done in NP. We again use Lemmas 5, 6, 8 and 9. Note that the value $d$ from Lemma 5 can be exponential. However, its binary presentation has polynomial size. So, instead of cycling though all $s$ from 0 to $d-1$, we can nondeterministically guess in polynomial time a value $s \in [0, d-1]$.

Next, we note that the values of $k_0$ in Lemma 8 and $k_2$ in Lemma 9 also have binary representations of polynomial size. Again, instead of checking every $k$ in $[0, k_0]$ or $[0, k_2]$, we can nondeterministically guess $k$ in polynomial time.

Finally, in the verification step of our algorithm we need to compute the matrices $A^d$, $A^s$ and $(A^d)^k$. This can be done in polynomial time using exponentiation by squaring. Indeed, the exponentiation by squaring requires polynomially many steps. Also, any power of a stochastic $\{0,1\}$-matrix is also a stochastic $\{0,1\}$-matrix, so the entries of the power matrices do not grow in size. ◀

## 4 Binary PFA

The following theorem shows that the $\lambda$-reachability and emptiness problems are NP-hard for binary PFA of *polynomial ambiguity* with *commuting* transition matrices (and the matrices can be assumed fixed in the case of $\lambda$-reachability and nonstrict emptiness). The emptiness problem for non-commutative binary PFA over 25 states is known to be undecidable, at least over exponentially ambiguous PFA [12]. Emptiness is also undecidable for exponentially ambiguous commutative PFA, although with many more states and a larger alphabet [2].

▶ **Theorem 12.** *The $\lambda$-reachability and emptiness problems are NP-hard for binary probabilistic finite automata of polynomial ambiguity with commuting matrices of dimension* 9 *for $\lambda$-reachability,* 37 *for nonstrict emptiness, and* 40 *for strict emptiness. Moreover, the matrices can be assumed* fixed *for the $\lambda$-reachability and nonstrict emptiness problems.*

**Proof.** We use a reduction from the solvability of binary quadratic Diophantine equations. Namely, given an equation of the form $ax^2 + by - c = 0$, where $a, b, c \in \mathbb{N}$, it is NP-hard to determine if there exists $x, y \in \mathbb{N}$ satisfying the equation [16]. We begin with the $\lambda$-reachability problem before considering the emptiness problem.

**$\lambda$-Reachability reduction.** Let $A = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$ and note that $A^k = \begin{pmatrix} 1 & k \\ 0 & 1 \end{pmatrix}$ and that $(A \otimes A)^k_{1,4} = (A^k \otimes A^k)_{1,4} = k^2$. We form a weighted automaton[2] $W_1$ on binary alphabet $\Sigma = \{h, g\}$ in the following way to encode $ax^2 + by$ (we will deal with $c$ later). Let $W_1 = (u_1, \phi, v_1)$ where $u_1, v_1 \in \mathbb{N}^7$ and $\phi : \Sigma^* \to \mathbb{N}^{7\times 7}$. We define $u_1 = (a, 0, 0, 0, b, 0, 0)^T$, $v_1 = (0, 0, 0, 1, 0, 1, 0)^T$ and $\phi(\ell) = \frac{1}{4}\phi'(\ell)$ for $\ell \in \{h, g\}$ with

[2] For our purposes here, by a *weighted automaton* we simply mean an automaton whose initial vector, final vector, and transition matrices are over nonnegative integers.

$$\phi'(h) = \left(\begin{array}{c|c} (A \otimes A) \oplus I_2 & t_1 \\ \hline \mathbf{0}^6 & 4 \end{array}\right), \quad \phi'(g) = \left(\begin{array}{c|c} I_4 \oplus A & t_2 \\ \hline \mathbf{0}^6 & 4 \end{array}\right),$$

with $\mathbf{0}^k = (0, 0, \ldots, 0) \in \mathbb{N}^k$, $t_1 = (0, 2, 2, 3, 3, 3)^T$ and $t_2 = (3, 3, 3, 3, 2, 3)^T$. We see then that each row of $\phi'(\ell)$ is nonnegative and sums to 4, thus $\phi(\ell)$ is stochastic for $\ell \in \{g, h\}$. Furthermore, by the mixed product property of the Kronecker product, we see that $((A \otimes A) \oplus I_2)^x = (A^x \otimes A^x) \oplus I_2$ and $(I_4 \oplus A)^y = I_4 \oplus A^y$ for $x, y \in \mathbb{N}$ and thus by the block upper triangular structure of $\phi'(h), \phi'(g)$, we see that

$$\phi'(h^x g^y) = \left(\begin{array}{c|c} (A^x \otimes A^x) \oplus A^y & t_{xy} \\ \hline \mathbf{0}^6 & 4^{x+y} \end{array}\right),$$

where $t_{xy}$ is a nonnegative vector maintaining the row sum at $4^{x+y}$. We now see that

$$u_1^T \phi(h^x g^y) v_1 = \frac{ax^2 + by}{4^{x+y}} \tag{11}$$

We define a second weighted automaton $W_2 = (u_2, \psi, v_2)$ with $u_2 = (c, 0)^T$, $v_2 = (0, 1)^T$ and $\psi : \Sigma^* \to \mathbb{N}^{2\times 2}$ with $\psi(\ell) = \frac{1}{4}\psi'(\ell)$ for $\ell \in \{h, g\}$ defined thus: $\psi'(h) = \psi'(g) = \begin{pmatrix} 1 & 3 \\ 0 & 4 \end{pmatrix}$. We therefore see that

$$u_2^T \psi(h^x g^y) v_2 = \frac{c(4^{x+y} - 1)}{4^{x+y}} = c(1 - \frac{1}{4^{x+y}}) \tag{12}$$

We now join $W_1$ and $W_2$ into a 9-state PFA $\mathcal{P} = (u, \gamma, v)$ where $u = \frac{1}{a+b+c}[u_1|u_2]$, $v = [v_1|v_2]$ and $\gamma(\ell) = \phi(\ell) \oplus \psi(\ell)$. Combining Eqns (11) and (12) we see that

$$\begin{aligned} u^T \gamma(h^x g^y) v &= \frac{1}{a+b+c}\left(\frac{ax^2 + by}{4^{x+y}} + c(1 - \frac{1}{4^{x+y}})\right) \\ &= \frac{1}{a+b+c}\left(c + \frac{ax^2 + by - c}{4^{x+y}}\right) \end{aligned} \tag{13}$$

which equals $\frac{c}{a+b+c}$ if and only if $ax^2 + by - c = 0$. Note that $\gamma(h)$ and $\gamma(g)$ commute by their structure since clearly $(A \otimes A) \oplus I$ and $I_4 \oplus A$ commute, giving $(A \otimes A) \oplus A$ in both cases (as a consequence of the mixed product properties of Lemma 1) and the rightmost vector of the matrix simply retains the row sum at 1 for such a product since the matrices are stochastic. Both $\gamma(h)$ and $\gamma(g)$ are upper-triangular thus $\mathcal{P}$ is polynomially ambiguous.

**Nonstrict Emptiness reduction.** We now show the proof of the emptiness problem. We showed that the $\lambda$-reachability problem is NP-hard by deriving a PFA $\mathcal{P}$ over the binary alphabet $\{h, g\}$ such that $\mathcal{P}(h^x g^y)$ is given by Eqn. 13. We note however that a non solution to $ax^2 + by - c = 0$ can be positive or negative and thus we may be above or below the threshold $\frac{c}{a+b+c}$. This encoding thus cannot be used to show the NP-hardness of the *emptiness* problem.

Instead, we can use a similar encoding of the quartic polynomial given by $(ax^2 + by - c)^2 = a^2x^4 + 2abx^2y + b^2y^2 + c^2 - 2acx^2 - 2bcy$ with $a, b, c \in \mathbb{N}$. Note that we arranged the four positive terms first, followed by the two negative terms. Clearly $(ax^2 + by - c)^2$ is nonnegative and equals zero if and only if $ax^2 + by - c = 0$. We will derive a PFA $\mathcal{P}_2$ such that

$$\mathcal{P}_2(h^x g^y) = \frac{1}{z}\left((2ac + 2bc) + \frac{1}{16^{x+y}}(ax^2 + by + c)^2\right),$$

where $z = a^2 + 2ab + b^2 + c^2 + 2ac + 2bc$, with the property that $\mathcal{P}_2(h^x g^y) \geq \frac{2ac+2bc}{z}$ with equality if and only if $(ax^2 + by - c)^2 = 0$ which is NP-hard to determine. To this end, we compute the following four matrices $\{H_+, G_+, H_-, G_-\}$, the idea being that $H_+$ and $G_+$ will be used to compute the positive four terms and $H_-$ and $G_-$ will compute the negative terms:

$$\begin{aligned}
H_+ &= \underbrace{(A \otimes A \otimes A \otimes A)}_{x^4} \oplus \underbrace{(A \otimes A \otimes I_2)}_{x^2y} \oplus \underbrace{(I_2 \otimes I_2)}_{y^2} \oplus \underbrace{1}_{1} \\
G_+ &= \underbrace{(I_2 \otimes I_2 \otimes I_2 \otimes I_2)}_{x^4} \oplus \underbrace{(I_2 \otimes I_2 \otimes A)}_{x^2y} \oplus \underbrace{(A \otimes A)}_{y^2} \oplus \underbrace{1}_{1} \\
H_- &= \underbrace{(A \otimes A)}_{x^2} \oplus \underbrace{I_2}_{y} \\
G_- &= \underbrace{(I_2 \otimes I_2)}_{x^2} \oplus \underbrace{A}_{y}
\end{aligned}$$

and by the mixed product property of Kronecker products of Lemma 1),

$$\begin{aligned}
H_+^x G_+^y &= (A^x \otimes A^x \otimes A^x \otimes A^x) \oplus (A^x \otimes A^x \otimes A^y) \oplus (A^y \otimes A^y) \oplus 1 \\
H_-^x G_-^y &= (A^x \otimes A^x) \oplus A^y
\end{aligned}$$

Note that $H_+^x G_+^y$ and $H_-^x G_-^y$ each contain the positive and negative (respectively) term of $(ax^2 + by - c)^2$, excluding the coefficients, e.g. $(H_+^x G_+^y)_{1,16} = x^4$ and $(H_+^x G_+^y)_{17,24} = x^2y$ etc. Note also that $H_+G_+ = G_+H_+$ and $H_-G_- = G_-H_-$ which also follows from the mixed product properties and thus matrices $\{H_+, G_+\}$ and $\{H_-, G_-\}$ commute.

As before, we may now increase the dimension of each matrix $\{H_+, H_-, G_+, G_-\}$ by 1 to ensure a common row sum (of 16 in this case) by adding a new column on the right hand side of each matrix, and then divide each matrix by this common value to give $\{H'_+, H'_-, G'_+, G'_-\}$ so that each of these matrices is row stochastic. Matrices $\{H'_+, G'_+\}$ and $\{H'_-, G'_-\}$ still commute since this change only has an effect on the final column of the matrix.

We now show how to handle each term of $(ax^2+by-c)^2$. We first handle the positive terms. We define $u_1 = (a^2, 0, \ldots, 0)^T \in \mathbb{Q}^{16}$, $u_2 = (2ab, 0, \ldots, 0)^T \in \mathbb{Q}^8$, $u_3 = (b^2, 0, 0, 0)^T \in \mathbb{Q}^4$ and $u_4 = c^2$ and then let $u_+ = [u_1|u_2|u_3|u_4|0] \in \mathbb{Q}^{30}$. We let $v_1 = (0, \ldots, 0, 1)^T \in \mathbb{Q}^{16}$, $v_2 = (0, \ldots, 0, 1)^T \in \mathbb{Q}^8$, $v_3 = (0, 0, 0, 1)^T \in \mathbb{Q}^4$ and $v_4 = 1$, and let $v_+ = [v_1|v_2|v_3|v_4|0] \in \mathbb{Q}^{30}$. We then see that

$$\begin{aligned}
& u_+^T (H'_+)^x (G'_+)^y v_+ \\
= \; & \frac{1}{16^{x+y}} \left( u_1^T (A^x \otimes A^x \otimes A^x \otimes A^x) v_1 + u_2^T (A^x \otimes A^x \otimes A^y) v_2 + u_3^T (A^y \otimes A^y) v_3 + u_4^T v_4 \right) \\
= \; & \frac{1}{16^{x+y}} \left( a^2 x^4 + 2abx^2 y + b^2 y^2 + c^2 \right) \qquad (14)
\end{aligned}$$

We next handle the negative terms, which is essentially accomplished by switching final and non-final states in the final state vectors to follow. Define $u_5 = (2ac, 0, 0, 0)^T \in \mathbb{Q}^4$ and $u_6 = (2bc, 0)^T \in \mathbb{Q}^2$ and let $u_- = [u_5|u_6|0] \in \mathbb{Q}^7$. We let $v_5 = (0, 0, 0, 1)^T \in \mathbb{Q}^4$ and $v_6 = (0, 1)^T \in \mathbb{Q}^2$. Define $v_- = [v_5|v_6|0] \in \mathbb{Q}^7$. We then see that

$$\begin{aligned}
& u_-^T (H'_-)^x (G'_-)^y (\mathbf{1} - v_-) \\
= \; & (2ac + 2bc) - \frac{1}{16^{x+y}} \left( u_5^T (A^x \otimes A^x) v_5 + u_6^T A^y v_6 + 0 \right) \\
= \; & (2ac + 2bc) - \frac{1}{16^{x+y}} \left( 2acx^2 + 2bcy \right), \qquad (15)
\end{aligned}$$

where $\mathbf{1} = (1, 1, \ldots, 1)^T \in \mathbb{Q}^7$. We used here the fact that $X\mathbf{1} = \mathbf{1}$ for a row stochastic matrix $X$. We finally define that $H = H'_+ \oplus H'_- \in \mathbb{Q}^{37 \times 37}$ and $G = G'_+ \oplus G'_- \in \mathbb{Q}^{37 \times 37}$, both of which are row stochastic and commute, and let $u_\star = \frac{[u_+|u_-]}{z} \in \mathbb{Q}^{37}$ and $v_\star = [v_+|(\mathbf{1} - v_-)] \in \mathbb{Q}^{37}$,

with $z = a^2 + 2ab + b^2 + c^2 + 2ac + 2bc$ to normalise vector $u_\star$. We see then that $u_\star$ is a stochastic vector as required. We define the PFA $\mathcal{P}_2 = (u_\star, \{H, G\}, v_\star)$ and we can now compute that

$$\begin{aligned}
& \mathcal{P}_2(h^x g^y) = u_\star^T H^x G^y v_\star \\
= \; & u_\star^T (H'^x_+ G'^y_+ \oplus H'^x_- G'^y_-) v_\star \\
= \; & \frac{1}{16^{x+y}} \left( \frac{[u_+|u_-]}{z}^T \left( \begin{array}{c|c|cc} \multicolumn{2}{c|}{H_+^x G_+^y} & \multicolumn{2}{c}{\multirow{2}{*}{$\mathbf{0}$}} \\ \multicolumn{2}{c|}{\mathbf{0} \quad | \quad 16^{x+y}} & \multicolumn{2}{c}{} \\ \hline \multicolumn{2}{c|}{\multirow{2}{*}{$\mathbf{0}$}} & \multicolumn{2}{c}{\multirow{2}{*}{$H_-^x G_-^y \quad | \quad *$}} \\ \multicolumn{2}{c|}{} & \multicolumn{2}{c}{$\mathbf{0} \quad | \quad 16^{x+y}$} \end{array} \right) [v_+|(\mathbf{1} - v_-)] \right) \\
= \; & \frac{1}{z 16^{x+y}} \left( u_+^T H_+^x G_+^y v_+ + u_-^T H_-^x G_-^y (\mathbf{1} - v_-) \right) \\
= \; & \frac{1}{z} \left( u_+^T (H'_+)^x (G'_+)^y v_+ + u_-^T (H'_-)^x (G'_-)^y (\mathbf{1} - v_-) \right) \\
= \; & \frac{1}{z} \left( (2ac + 2bc) + \frac{1}{16^{x+y}} \left( a^2 x^4 + 2abx^2 y + b^2 y^2 + c^2 \right) - \frac{1}{16^{x+y}} \left( 2acx^2 + 2bcy \right) \right) \\
= \; & \frac{1}{z} \left( (2ac + 2bc) + \frac{1}{16^{x+y}} (ax^2 + by - c)^2 \right) \qquad (16)
\end{aligned}$$

where $*$ denote the column vectors used to ensure row sums of $16^{x+y}$ and $\mathbf{0}$ denotes zero matrices of appropriate sizes. We also used Eqns (14) and (15).

Since $(ax^2 + by - c)^2$ is nonnegative, we see that $u_\star^T H^x G^y v_\star \geq \frac{2ac+2bc}{z}$ with equality if and only if $(ax^2 + by - c)^2 = 0$, which is NP-hard to determine. Therefore using cutpoint $\lambda = \frac{2ac+2bc}{z} \in \mathbb{Q} \cap [0, 1]$ means the (nonstrict) emptiness problem is NP-hard (i.e. does there exist $x, y \in \mathbb{N}$ such that $u_\star^T H^x G^y v_\star \leq \lambda$ is NP-hard). As before, matrices $H$ and $G$ are upper-triangular and commute by their structure, and therefore the result holds.

**Strict Emptiness reduction.** Finally we show how to handle the strict emptiness problem. We proceed with a technique inspired by [10]. By (16), if $P_2(h^x g^y) = u_\star^T H^x G^y v_\star \neq \frac{1}{z}(2ac + 2bc)$, then $u_\star^T H^x G^y v_\star \geq \frac{1}{z}\left((2ac + 2bc) + \frac{1}{16^{x+y}}\right)$ therefore $P_2(h^x g^y) \leq \frac{1}{z}(2ac + 2bc)$ if and only if $P_2(h^x g^y) < \frac{1}{z}\left((2ac + 2bc) + \frac{1}{16^{x+y}}\right)$.

Let us adapt $\mathcal{P}_2$ in the following way to create a new PFA $\mathcal{P}_3$. Note that $\mathcal{P}_2$ has 6 initial states (by $u_\star$). We add three new states to $\mathcal{P}_3$, denoted $q_0, q_F$ and $q_*$. State $q_0$ is a new initial state of $\mathcal{P}_3$ which, for any input letter, has probability $\frac{1}{2 \cdot 6}$ of moving to each of the 6 initial states of $\mathcal{P}_2$ and probability $\frac{1}{2}$ to move to new state $q_F$. State $q_F$ is a new final state that remains in $q_F$ for any input letter with probability $1 - \frac{1}{16z}$ and moves to a new non-accepting absorbing sink state $q_*$ with probability $\frac{1}{16z}$. We now see that for any $a \in \{h, g\}$:

$$\mathcal{P}_3(aw) = \frac{1}{2}\mathcal{P}_2(w) + \frac{1}{2}\left(1 - \frac{1}{16^{|w|} z^{|w|}}\right)$$

If there exists $w_1 = h^x g^y$ with $x, y \geq 0$ such that $\mathcal{P}_2(w_1) \leq \frac{1}{z}(2ac + 2bc)$ then $\mathcal{P}_2(w_1) = \frac{1}{z}(2ac + 2bc)$ and thus:

$$\mathcal{P}_3(aw_1) = \frac{1}{2}\left(\frac{1}{z}(2ac + 2bc)\right) + \frac{1}{2}\left(1 - \frac{1}{16^{|w_1|} z^{|w_1|}}\right) < \frac{1}{2}\left(\frac{1}{z}(2ac + 2bc) + 1\right).$$

For any $w_2 = h^x g^y$ with $x, y \geq 0$ such that $\mathcal{P}_2(w_2) > \frac{1}{z}(2ac + 2bc)$ then $\mathcal{P}_2(w_2) \geq \frac{1}{z}(2ac + 2bc) + \frac{1}{16^{x+y}}$ by (16). Thus:

$$\mathcal{P}_3(aw_2) \geq \frac{1}{2}\left(\frac{1}{z}(2ac + 2bc) + \frac{1}{16^{|w_2|}}\right) + \frac{1}{2}\left(1 - \frac{1}{16^{|w_2|} z^{|w_2|}}\right) > \frac{1}{2}\left(\frac{1}{z}(2ac + 2bc) + 1\right).$$

Thus determining if there exists $w = h^x g^y$ such that $\mathcal{P}_3(w) < \frac{1}{2}\left(\frac{1}{z}(2ac + 2bc) + 1\right)$, i.e. the strict emptiness problem for $\mathcal{P}_3$ on cutpoint $\frac{1}{2}\left(\frac{1}{z}(2ac + 2bc) + 1\right)$, is NP-hard. The modifications to $\mathcal{P}_2$ retain polynomial ambiguity since $q_0$ and $q_F$ have no incoming (non self looping) edges and $q_*$ has no outgoing edges, therefore property EDA does not hold. Commutativity of the PFA is unaffected since $\mathcal{P}_3$ is identical to $\mathcal{P}_2$ except for adding three new states, behaving identically for both input letters. Note that $\mathcal{P}_3$ has $37 + 3 = 40$ states. ◀

## References


1 S. Akshay, Timos Antonopoulos, Joël Ouaknine, and James Worrell. Reachability problems for Markov chains. *Information Processing Letters*, 115(2):155–158, 2015.

2 Paul C. Bell. Polynomially ambiguous probabilistic automata on restricted languages. In *International Colloquium on Automata, Languages, and Programming*, number 105 in ICALP'19, pages 1–14, 2019.

3 Paul C. Bell and P. Semukhin. Decidability of cutpoint isolation for probabilistic finite automata on letter-bounded inputs. In *International Conference on Concurrency Theory*, number 22 in CONCUR'20, pages 1–16, 2020.

4 A. Bertoni, G. Mauri, and M. Torelli. Some recursively unsolvable problems relating to isolated cutpoints in probabilistic automata. In *Automata, Languages and Programming*, volume 52, pages 87–94, 1977.

5 V. Blondel and V. Canterini. Undecidable problems for probabilistic automata of fixed dimension. *Theory of Computing Systems*, 36:231–245, 2003.

6 V. D. Blondel and N. Portier. The presence of a zero in an integer linear recurrent sequence is NP-hard to decide. *Linear Algebra and its Applications*, pages 91–98, 2002.

7 L. Daviaud, M. Jurdzinski, R. Lazic, F. Mazowiecki, G. A. Pérez, and J. Worrell. When is containment decidable for probabilistic automata? In *International Colloquium on Automata, Languages, and Programming*, number 121 in ICALP'18, pages 1–14, 2018.

8 N. Fijalkow, C. Riveros, and J. Worrell. Probabilistic automata of bounded ambiguity. In *28th International Conference on Concurrency Theory (CONCUR)*, pages 19:1–19:14, 2017.

9 S. Friedland. *Matrices: Algebra, Analysis and Applications*. World Scientific Publishing Company Pte Limited, 2015. URL: `https://books.google.co.uk/books?id=y8fACwAAQBAJ`.

10 H. Gimbert and Y. Oualhadj. Probabilistic automata on finite words: decidable and undecidable problems. In *International Colloquium on Automata, Languages and Programming (ICALP'10)*, volume 2, pages 527–538, 2010.

11 V. Halava, T. Harju, M. Hirvensalo, and J. Karhumäki. Skolem's problem — on the border between decidability and undecidability. In *TUCS Technical Report Number 683*, 2005.

12 M. Hirvensalo. Improved undecidability results on the emptiness problem of probabilistic and quantum cut-point languages. *SOFSEM 2007: Theory and Practice of Computer Science, Lecture Notes in Computer Science*, 4362:309–319, 2007.

13 R. A. Horn and C. R. Johnson. *Topics in matrix analysis*. Cambridge University Press, 1991.

14 O. Ibarra and B. Ravikumar. On sparseness, ambiguity and other decision problems for acceptors and transducers. In *Proc. STACS 1986*, volume 210, pages 171–179, 1986.

15 S. C. Kleene. Representation of events in nerve nets and finite automata. *Automata Studies, Annals of Mathematical Studies*, 34, 1956.

16 K. L. Manders and L. Adleman. NP-complete decision problems for binary quadratics. *Journal of Computer and System Sciences*, 16:168–184, 1978.

17 J. Ouaknine and J. Worrell. Positivity problems for low-order linear recurrence sequences. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA'14, pages 366–379, SODA, 2014.

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

19 M. O. Rabin. Probabilistic automata. *Information and Control*, 6:230–245, 1963.

**20** C. Reutenauer. *Propertiétés arithmétiques et topologiques de séries rationnelles en variables non commutatives*. Thèse troisième cycle, Université Paris VI, 1977.

**21** P. Turakainen. Generalized automata and stochastic languages. *Proceedings of the American Mathematical Society*, 21:303–309, 1969.

**22** N. K. Vereshchagin. The problem of appearance of a zero in a linear recurrence sequence (in russian). *Mat. Zametki*, 38(2), 1985.

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