Decidable Problems for Unary PFAs

Rohit Chadha[1], Dileep Kini[2], and Mahesh Viswanathan[2]

[1] University of Missouri
[2] University of Illinois at Urbana-Champaign

Abstract. Given a PFA A and a cut-point λ, the isolation problem asks if there is a bound $\epsilon > 0$ such that the acceptance probability of every word is bounded away from λ by ϵ. In this paper we show that the isolation problem for PFAs with a unary input alphabet is (a) coNP-complete, if the cut-point is 0 or 1, and (b) is in $\mathsf{coNP}^{\mathsf{RP}}$ and coNP-hard, if the cut-point is in $(0, 1)$. We also show that the language containment problem, language equivalence problem, the emptiness problem and the universality problem for unary PFAs with limit isolated cut-points is in the fourth level of counting hierarchy $\mathsf{C}_4\mathsf{P}$ (and hence in PSPACE).

1 Introduction

Probabilistic finite automata (PFA), introduced by Rabin [18], are a generalization of deterministic finite automata that model finite state, randomized algorithms that process an input string one-way. Given a cut-point $\lambda \in [0, 1]$, an input string w is accepted by a PFA A iff the probability of reaching a final/accept state of A on input w from the initial state is $> \lambda$, and $L_{>\lambda}(A)$ denotes the collection of all strings accepted by A with cut-point λ. The emptiness problem for PFAs is not only an important mathematical problem, but it has applications in verifying useful properties of open, probabilistic, reactive systems like sensor networks, biochemical reactions, and software [14, 15, 13, 1]. For example, checking emptiness is equivalent to checking if drug concentrations in certain organs is always below toxicity levels [13]. In such contexts, one would, in fact, like to ensure a stronger property, namely, that the drug concentrations are well below acceptable levels, and not just barely acceptable.

Checking such "robust" properties of a system is closely related to another important problem of PFAs, namely, the *isolation* problem. A cut-point λ is said to be isolated for an automaton A if there is an $\epsilon > 0$ (called a *degree of isolation*) such that A's probability of reaching an accepting state on any word is either $> \lambda + \epsilon$ or is $< \lambda - \epsilon$. PFAs with isolated cut-points enjoy many nice properties. First, they represent algorithms with bounded error, for which, the error probability can be driven down below a fixed level using repeated experiments [18]. Second, λ being isolated ensures that $L_{>\lambda}(A)$ is regular; note, if λ is not isolated then $L_{>\lambda}(A)$ may not be regular [**?**,18] (even for unary alphabet [17, 20]). Finally, certain PFAs with isolated cut-points are *stable*, in that small changes to the transition probabilities don't change the language with respect to cut-point λ [18].

Even though the emptiness and isolation problems have important practical applications, these problems have been shown to be computationally very hard. The emptiness problem is co-r.e.-complete [17, 7] and the isolation problem is Σ_2^0-complete [4, 10, 6]. However, these lower bounds only apply when the input alphabet of the PFA has at least two symbols. The decidability of the isolation problem for PFAs over the unary input alphabet was claimed to be true in Bertoni's original paper [4] but was not proved. Furthermore, there was no complexity analysis for the isolation problem. The decidability of the emptiness problem for PFAs over the unary alphabet is a long-standing open problem.

In this paper we consider these decision problems for PFAs over a unary input alphabet. Unary PFAs are nothing but (finite state) Markov chains, which are a standard model to define the semantics of probabilistic systems, and have been used in a number of contexts. Our main result is about the complexity of the isolation problem for unary PFAs. When $\lambda \in (0, 1)$, we show that the isolation problem is decidable and is in $\mathsf{coNP}^{\mathsf{RP}}$.[3] Note that since RP is contained in NP (see [3]), this implies that the problem of checking isolation when $\lambda \in (0, 1)$ is in the second level of polynomial hierarchy. Furthermore, given that RP is believed to be P, this would imply the problem to be in coNP which matches the lower bound of coNP-hardness mentioned ahead. Our procedure also gives a way to compute a degree of isolation if the PFA is isolated. Our result is proved as follows. Let us call a PFA A isolated in the limit if there is a $n_0 > 0$ such that the probability of accepting any string a^n, with $n > n_0$ is bounded away from λ. Thus λ is an isolated cut-point for a PFA A iff A is isolated in the limit and the probability of accepting any "short" string (i.e., one whose length is less than n_0) is bounded away from λ. We first prove that the problem of checking if A is isolated in the limit is in coNP. Next, we show that if A is isolated in the limit, then the bound n_0 is "small". More precisely, we show that this number n_0 can be represented in binary using polynomially (in the size of A) many bits. Using this observation, we can conclude that if a PFA A is isolated in the limit, then λ is not isolated if there is some string (of exponential length) that is accepted with probability λ. The check of whether a string of length ℓ is accepted with probability λ can be reduced to checking if a straight-line program of length ℓ using addition, multiplication, and subtraction computes a real number that is equal to 0. Based on this observation, and results on the complexity of the EquSLP problem [21], we conclude that checking the isolation of a PFA over the unary alphabet is in $\mathsf{coNP}^{\mathsf{RP}}$. Next, we show that if the cut-point λ is either 0 or 1 the isolation problem is easier. We show that for these extremal cut-points, the isolation problem is in coNP. The proof uses observations about the complexity of the universality problem for NFAs [23]. We also show that the isolation problem is coNP-hard.

Our techniques for checking isolation for unary PFAs have a few consequences. One can show that if A and B are limit isolated PFAs then the problem of checking $L_{>\lambda}(A) \subseteq L_{>\lambda}(B)$ is in $\mathsf{coNP}^{\mathsf{C_3P}}$. The complexity class $\mathsf{coNP}^{\mathsf{C_3P}}$ lies

[3] RP is the set of decision problems that can be decided by randomized polynomial-time Turing Machines with one-sided error (see Section 2.3).

in the fourth level of counting hierarchy and hence in PSPACE [25]. That means that the language equivalence problem, the emptiness problem and the universality problem for limit isolated PFAs are decidable in $\mathsf{coNP}^{\mathsf{C_3P}}$ and hence in PSPACE. These results need to be contrasted with the fact that the decidability of the emptiness problem for unary PFAs (when λ is not necessarily isolated) is still open. Similarly, the decidability of the emptiness problem for PFAs with isolated cut-points, with input alphabet having at least 2 symbols, is also open.

The rest of the paper is organized as follows. We discuss preliminary definitions and results in Section 2. Our results on complexity of checking limit isolation and isolation for unary PFAs are discussed in Section 3 and Section 4. We discuss the problems of language emptiness, containment and universality for limit isolated unary PFAs in Section 5. We present our conclusions in Section 6.

Related Work. As pointed out in the introduction, the undecidability of the emptiness and isolation problems for PFAs was established in [17, 7, 4, 10, 6]. The efficient decidability of bisimulation for probabilistic systems can be exploited to efficiently check a strong version of equivalence of PFAs [24, 8, 11, 12] — here PFAs A and B are said to be equivalent if the acceptance probability of each input string is the same in both A and B. The decidability of language equivalence is a consequence of a more general result on minimizing weighted automata [**?**]. Model checking of Markov chains with respect to PCTL properties is a mature technology [19]. However, such tools cannot answer emptiness, containment, and language equivalence of unary PFAs because such properties are not expressible in PCTL. Convergence properties of Markov chains have been widely studied (see [9, 16]) but questions of complexity of isolation pertain to both transient and asymptotic behavior of Markov chains which, to the best of our knowledge, has not been studied. The decidability of checking emptiness for isolated unary PFAs can also be derived as a consequence of the results in [5] which establishes that the problem of checking emptiness of isolated eventually weakly ergodic PFAs is decidable. However, [5] does not contain any complexity analysis. Furthermore, our results on complexity of emptiness checking applies to limit isolated unary PFAs which are a strict super-set of isolated unary PFAs.

2 Preliminaries

We introduce some notation and recall some standard notation. We will fix some notational conventions used in this paper. We will assume that for any finite S of k elements, we have a *fixed* enumeration $\{1, 2, \ldots, k\}$ of the elements of S. We will identify elements of S with the corresponding numeral.

2.1 Distributions, Stochastic Matrices and Markov chains

Distributions. Given a finite set S, a distribution over S is any function $\mu : S \mapsto [0, 1]$ such that $\sum_{s \in S} \mu(s) = 1$. Since S is finite, μ can be thought of as a vector with $|S|$ coordinates. The set of all distributions over S is represented by

$dist(S)$. For a set $S' \subseteq S$, we write $\mu(S') = \sum_{i \in S'} \mu(i)$. For any two distributions $\mu, \nu \in dist(S)$ the distance between them is defined as

$$\boldsymbol{d}(\mu, \nu) = \sum_{i \in S} \frac{|\mu(i) - \nu(i)|}{2} = \max_{S' \subseteq S} |\mu(S') - \nu(S')|.$$

The distance $\boldsymbol{d}$ is a metric.

Stochastic matrices. A stochastic matrix δ is a square matrix with non-negative entries such that each row of the matrix sums up to one. This distance between $n \times n$ matrices δ_1,δ_2 is defined as:

$$\boldsymbol{d}(\delta_1, \delta_2) = \max_i \sum_j |\delta_1(i,j) - \delta_2(i,j)|.$$

We use $\delta(;,j)$ to represent the jth column of the matrix δ. We use $\sigma_t(\delta)$ to denote the sequence of matrices $\delta^t, \delta^{2t}, \delta^{3t}, \ldots$ and use $\widehat{\delta^t}$ to denote the limit $\lim_{r \to \infty} \delta^{rt}$ if it exists. When clear from the context, we shall drop δ from $\sigma_t(\delta)$ and just write σ_t. A stochastic matrix δ is called *positive* if all of its entries are strictly positive.

A stochastic matrix of dimension $n \times n$ can be represented as a directed graph with n vertices, and an edge from i to j if $\delta(i,j) > 0$. A maximally strongly connected component is called a *Bottom Strongly Connected Component* (BSCC) if it has no outgoing edges. A *transient state* is a state which is not in a BSCC, and a *terminal state* is one which is within a BSCC. δ is said to be *irreducible* if it has only one BSCC and no transient states. The collection of all BSCCs of a δ will be represented by $\mathcal{C}_\delta$. The set of all transient states of δ will be denoted by T_δ. Lower case c_δ will be used to denote individual BSCCs. When clear from the context, we shall drop the subscript δ.

The *period* of a vertex is defined as the g.c.d (greatest common divisor) of all the cycle lengths going through the vertex. For a SCC, the periods of all the vertices in that component will be the same and will be defined as the period of that component. A component is called *aperiodic* if its period is 1. δ is said to be *aperiodic* if all vertices have period 1. The *ultimate period* of δ is the l.c.m (least common multiple) of the periods of its BSCCs. Since BSCCs and their related periods can be computed in polynomial time we have the following:

Proposition 1. *The ultimate period of a $n \times n$ matrix δ can be computed in polynomial time and is a number with $O(n \log n)$ bits.*

The following Lemma is proved in [22]

Lemma 1. *For any $n \times n$ stochastic matrix δ, if δ is an aperiodic and irreducible stochastic matrix then δ^{n^2} is positive.*

A stochastic matrix γ with dimensions $n \times n$ is called a *contraction map with contracting factor* $\alpha < 1$ if for all distributions μ and ν of dimension n it is the case that $\boldsymbol{d}(\mu\gamma, \nu\gamma) < \alpha \boldsymbol{d}(\mu, \nu)$.

Proposition 2. *For any $n \times n$ stochastic matrix δ if δ is positive then δ is contracting with contracting factor $1 - n \min_{i,j} \delta(i,j)$.*

Proof. Let s be $\min_{i,j} \delta(i,j)$. Observe that $s \leq \frac{1}{n}$.

$$\begin{aligned}
2\boldsymbol{d}(\mu\delta, \nu\delta) &= \sum_j \big|\mu\delta(j) - \nu\delta(j)\big| = \sum_j \big|\sum_i \big(\mu(i)\delta(i,j) - \nu(i)\delta(i,j)\big)\big| \\
&= \sum_j \big|\sum_i \big(\mu(i)(\delta(i,j) - s) - \nu(i)(\delta(i,j) - s) + s(\mu(i) - \nu(i))\big)\big| \\
&= \sum_j \big|\sum_i \big(\mu(i)(\delta(i,j) - s) - \nu(i)(\delta(i,j) - s)\big) + s\big(\sum_i \mu(i) - \sum_i \nu(i)\big)\big| \\
&= \sum_j \big|\sum_i \big((\mu(i) - \nu(i))(\delta(i,j) - s)\big) + s(1-1)\big| \\
&\leq \sum_j \sum_i \big|(\mu(i) - \nu(i))\big|(\delta(i,j) - s) = \sum_i \sum_j \big|(\mu(i) - \nu(i))\big|(\delta(i,j) - s) \\
&\leq \sum_i \big|(\mu(i) - \nu(i))\big| \sum_j (\delta(i,j) - s) = \sum_i \big|(\mu(i) - \nu(i))\big|(1 - ns) \\
&\leq 2(1 - ns)\boldsymbol{d}(\mu, \nu).
\end{aligned}$$

□

Markov Chains A Markov chain M is a tuple (Q, δ, μ_0) where Q is a finite set of states, δ is stochastic matrix of dimension $|Q| \times |Q|$ and $\mu_0 \in dist(Q)$. δ is referred to as the transition matrix and μ_0 denotes the initial distribution. The Markov chain represents an infinite sequence of distributions $\mu_0, \mu_1, ..$ where $\mu_i = \mu_0\delta^i$.

2.2 Probabilistic Finite Automata

A PFA [18] is like a deterministic automaton except that the transition on an input symbol is probabilistic.

Definition 1. *A Probabilistic Finite Automaton (PFA) A is a tuple $(Q, \Sigma, (\delta_\sigma)_{\sigma\in\Sigma}, \mu_0, Q_F)$, where Q is a finite set of states, Σ is a finite alphabet, $\mu_0 \in dist(Q)$ is the initial distribution, $Q_F \subseteq Q$ is the set of final states, and $(\delta_\sigma)_{\sigma\in\Sigma}$ is an indexed set of stochastic matrices with dimension $|Q| \times |Q|$.*

For a symbol a, $\delta_a(s,t)$ represents the probability of going from state s to t on input symbol a. For any input word $w \in \Sigma^*$ of length n the probability of going from s to t along $w = a_1a_2\cdots a_n$ is then given by $\delta_w(s,t)$ where δ_w is the matrix $(\delta_{a_1} \cdot \delta_{a_2} \cdots \delta_{a_n})$. The distribution reached on input $w \in \Sigma^*$ in A is then given by $\mu_0\delta_w$.

Definition 2. *The acceptance probability of a word $w \in \Sigma^*$ on PFA A is given by $\sum_{q\in Q_F} \mu_0\delta_w(q)$ or $\mu_0\delta_w\eta_F$ where η_F is the column vector such that $\eta_F(j) = 1$ if $j \in Q_F$ and $\eta_F(j) = 0$ otherwise.*

We will say that η_F is the vector corresponding to Q_F.

Languages defined by PFAs need not be regular (even over unary alphabet [17][20]). Emptiness checking turns out to be undecidable in general [7] but is still open for the unary case.

Since this paper only considers PFAs over a unary alphabet, the remaining definitions only apply to unary PFAs.

We will assume that the unique letter in the unary alphabet is a, and we will drop the index a in the transition δ_a. Thus, the probability of accepting the string of length a^ℓ is given by $\mu_0\delta^\ell\eta_F$. We will often use μ_k to denote $\mu_0\delta^k$. A unary PFA is essentially a Markov chain with a subset of the states designated as final states. The language of PFA is defined with respect to a cut-point λ:

Definition 3. *Given a cut-point $\lambda \in [0,1]$ the language accepted by a unary PFA A with respect to λ denoted by $L_{>\lambda}(A)$ is*

$$\{a^n \mid \mu_0\delta^n\eta_F > \lambda\}.$$

We are interested in special kinds of cut-points, called *isolated* cut-points [18]. A cut-point λ is isolated for a PFA A if the acceptance probabilities of all the words are bounded away from λ.

A cut-point is said to be extremal if it is either 0 or 1, and non-extremal if it is the open interval $(0,1)$.

Definition 4. *The cut-point λ is said to be isolated for A if there exists an $\epsilon > 0$ such that for all $n > 0$,*

$$|\mu_0\delta^n\eta_F - \lambda| > \epsilon.$$

ϵ is known as a degree of isolation.

When λ is isolated, the language recognized $L_{>\lambda}(A)$ is known to be regular [18]. We introduce the notion of *limit isolation*, which generalizes the notion of isolated cut-points. We say that a cut-point is limit isolated if it is isolated for asymptotically large inputs. Formally,

Definition 5. *The cut-point λ is said to be limit isolated for A if there exists $\epsilon > 0$ and $n_0 > 0$ and such that for all $n > n_0$,*

$$|\mu_0\delta^n\eta_F - \lambda| > \epsilon.$$

ϵ is known as a degree of limit isolation.

Note 1. A PFA A is isolated at λ iff it is limit isolated at λ and there is no word that is accepted with probability exactly λ. It is also easy to see that the language recognized $L_{>\lambda}(A)$ is regular if λ is limit isolated.

The two definitions lead us to the problems of checking if a given rational cut-point is limit isolated or isolated, which we will tackle in sections 3 and 4 respectively.

The following proposition implies that checking whether 0 is isolated (limit isolated, respectively) is as hard as checking if 1 is isolated (limit isolated, respectively) and vice-versa.

Proposition 3. *For any PFA A, there is another PFA B such that 0 is an isolated (limit isolated respectively) cut-point of A iff 1 is an isolated (limit isolated respectively) cut-point of B.*

Proof. We just need to interchange final and non-final states. That is if $A = (Q, \Sigma, (\delta_\sigma)_{\sigma\in\Sigma}, \mu_0, Q_F)$ then we can take $B = (Q, \Sigma, (\delta_\sigma)_{\sigma\in\Sigma}, \mu_0, Q \setminus Q_F)$.

2.3 Complexity

The complexity class RP consists of problems which can be solved using a randomized polynomial time algorithm that always returns "yes" on yes-instances, and returns "no" with probablity at least $\frac{1}{2}$ on no-instances. We know that RP is contained in NP.

The counting hierarchy CH is a class of decision problems contained within PSPACE, which was introduced by Wagner [25]. The 0-th level, $\mathsf{C_0P}$, is defined as P. The k-th level of the hierarchy is denoted by $\mathsf{C}_k\mathsf{P}$ and is defined recursively as $\mathsf{C}_{k+1}\mathsf{P} = \mathsf{PP}^{\mathsf{C}_k\mathsf{P}}$. Here PP denotes the class of decision problems for which there are polynomial time randomized algorithms which answer "yes" with probability $> \frac{1}{2}$ on yes-instances, and answer "no" with probability $\geq \frac{1}{2}$ on no-instances. The whole counting hierarchy is contained in PSPACE.

In this paper we will assume every rational number is represented as $\frac{p}{q}$ where p and q are integers in binary. So, when we say a rational r can be computed in polynomial time given rationals $r_1, \ldots, r_k$, it implies that r can also be represented using polynomially many bits in the inputs $r_1, \ldots, r_k$.

2.4 Straight Line Programs

We will use *straight line programs* (SLP) to represent the computation of quantities such as acceptance probability of a word. A SLP over a set of variables V is a sequence of statements of the form $x := E$ where $x \in V$; E is either a constant in $\{0, 1\}$, a variable in V, or an expression of the form $e_1 \circ e_2$ where the operator $\circ \in \{+, -, *\}$ and $e_i \in \{0, 1\} \cup V$. Furthermore, each variable occurring on the right hand side of an assignment must occur in the left hand side of (some) earlier assignment. The value of a SLP is defined as the value assigned in its last statement. EquSLP is the problem of deciding if the value returned by the SLP is 0. PosSLP is defined as the problem of determining whether the value of the given SLP is positive. EquSLP was shown to be in coRP in [21]. A recent result [2] shows that PosSLP is in $\mathsf{P}^{\mathsf{C}_3\mathsf{P}}$ and hence in the 4-th level of counting hierarchy.

3 Limit Isolation

We prove that the problem of checking if a cut-point (extremal or non-extremal) is limit-isolated for a unary PFA is coNP-complete. In order to prove these results, we recall some standard facts about Markov chains. The proofs of these facts can be found in [9] and hence are omitted.

Theorem 1. *Let $c \in \mathcal{C}$ be a BSCC of a Markov Chain $M = (Q, \delta, \mu_0)$, p be the period of c, then for any state j in c:*

1. *If i is a transient state of M then $\lim_{r\to\infty} \delta^{pr}(i,j)$ exists and can be calculated in time polynomial in the size of δ.*
2. *If i is in c, then $\lim_{r\to\infty} \delta^{pr}(i,j)$ exists and can be calculated in time polynomial in the size of δ.*
3. *If i is neither a transient state of M nor in c then $\lim_{r\to\infty} \delta^{pr}(i,j) = 0$ (in fact $\delta^{\ell}(i,j) = 0$ for all ℓ).*

This leads to the following corollary (recall that the ultimate period of δ is the l.c.m of the period of its BSCCs).

Corollary 1. *For any stochastic matrix δ, with ultimate period p, $\widehat{\delta^p} = \lim_{r\to\infty} \delta^{pr}$ exists.*

We are ready to show that limit isolation is coNP-complete.

Theorem 2. *The problem of checking given a unary PFA A and a rational cut-point λ whether λ is limit-isolated for A is* coNP*-complete.*

Proof. (**Upper Bound**). Let $A = (Q, \Sigma, \delta, \mu_0, Q_F)$ and let p be the ultimate period of δ. According to Corollary 1, there are possibly p different limits towards which the Markov chain approaches in a cyclic manner. That is for each $0 \leq k < p$, we have that $\lim_{r\to\infty} \delta^{k+pr}$ exists.

If λ is not a limit isolated cut-point then it is easy to see that there is a $0 \leq k < p$ such that $\lim_{r\to\infty} \mu_0\delta^{k+pr}\eta_F$ is λ. The witness for a *no* answer to our problem is therefore going to be this number k which requires only $n \log n$ bits to be represented.

The result will follow if we can compute the distribution $\mu_k\widehat{\delta^p} = \lim_{r\to\infty} \mu_0\delta^{k+pr}$ in polynomial time. This can be achieved as follows. Note that $\mu_k\widehat{\delta^p}(i) = 0$ for any transient state i. We only have to compute $\mu_k\widehat{\delta^p}(i)$ for terminal states i.

Consider a BSCC c_i of δ. Let its period be p_i, let k_i be k mod p_i. Note that p can be exponentially large but each of the p_is at most n. Although $\sigma_{p_i} = \delta^{p_i}, \delta^{2p_i}, \ldots$ need not converge, it follows from Theorem 1 that the columns corresponding to c_i do converge to a limit. Now $\widehat{\delta^{p_i}}(;,j) = \widehat{\delta^p}(;,j)$ for any state $j \in c_i$ because σ_p is a subsequence of σ_{p_i}. So the entire matrix $\widehat{\delta^p}$ can be calculated in polynomial time. Essentially the jth column of $\widehat{\delta^p}$ is identical to the jth column of $\widehat{\delta^{p_i}}$. In order to calculate $\delta^k\widehat{\delta^p}$ observe that its jth column $\delta^k\widehat{\delta^p}(;,j) = \delta^k\widehat{\delta^{p_i}}(;,j) = \delta^{k_i}\widehat{\delta^{p_i}}(;,j)$ where again p_i is the period of the BSCC c_i that contains j. Note that Now $k_i < p_i \leq n$ and so δ^{k_i} can be calculated in polynomial time. The upper bound follows.

(**Lower Bound**). In order to prove hardness we use the reduction in [**?**,23] which is used to show coNP-hardness of the universality problem for unary non-deterministic finite automata (NFA). We briefly describe the salient features of the reduction; for further details the reader should refer to [**?**]. The original

reduction is from 3SAT to non-universality of unary NFA. Given a 3SAT formula ϕ with n variables and m clauses, [**?**] constructs a NFA N_ϕ as a union of m cyclic automata. Intuitively, each cycle corresponds to a clause, has an initial state and a cycle accepts if and only if the input encodes an assignment that does not satisfy that clause. So N_ϕ accepts every input iff ϕ is unsatisfiable. The only non-determinism in N_ϕ is from having to choose a cycle at the beginning, so we can transform it into a PFA P_ϕ by choosing amongst the cycles uniformly at random. Since there are only m cycles any word that is accepted by N_ϕ is accepted by P_ϕ with probability at least $\frac{1}{m}$; otherwise it is accepted with probability 0. So 0 is an isolated cut-point for P_ϕ iff ϕ is unsatisfiable. Finally we observe that P_ϕ is isolated at 0 iff it is limit isolated at 0, which proves that 0 is a limit-isolated cut-point for P_ϕ iff ϕ is unsatsifiable. We have already observed that if a cut-point is isolated then it is also limit-isolated. For the converse observe that the constructed unary NFA N_ϕ is a disjoint union of cycles. Let d be the lcm of all the cycles of N_ϕ. Now it is easy to see that for each j, the probability distribution on the states of the unary PFA P_ϕ on input a^j is the same as the probability distribution on input $a^{j \bmod d}$. So if there is some word a^j that is accepted with 0 probability then 0 cannot be a limit-isolated cut-point. □

Remark 1. Please note that the lower bound proof of Theorem 2 can be modified if the cut-point λ is not extremal; simply add an additional state with a self loop, which you choose initially with probability λ. Also, we could have taken the cut-point to be 1 thanks to Proposition 3. Thus, complexity of limit-isolation does not depend on whether the cut-point is extremal or not. Also, note that the lower bound proof also establishes the coNP-hardness of the isolation problem.

4 Complexity of Isolation Checking

We will prove that the problem of checking whether λ is isolated is in $\mathsf{coNP}^{\mathsf{RP}}$ (see Theorem 3). For extremal cut-points, i.e., when λ is 0 or 1, we will show the problem to be coNP-complete (see Theorem 4). We start by discussing non-extremal cut-points.

Non-extremal cut-points. Broadly speaking, the proof for showing that isolation is in $\mathsf{coNP}^{\mathsf{RP}}$ is as follows:

- We can use Theorem 2 to check if the cut-point λ is limit isolated for A. If it is not limit-isolated then we know that the cut-point is not isolated.
- If it is limit isolated, then λ will be isolated iff there is no word a^ℓ accepted with probability λ. We will show that that this word cannot be too long (see Lemma 3).
- We can then guess this word, construct a straight-line program such that its value is 0 iff the probability of accepting this word is λ, and check if it evaluates to 0 or not (see Lemma 2).

We start by showing that the problem of deciding given a PFA A and a number n in binary, whether a word a^n is accepted with probability $=\lambda$ is in coRP and $> \lambda$ is in the counting hierarchy.

Lemma 2. *Given unary PFA A, a non-negative integer n in binary and a rational number λ, the problem of checking:*

1. *if a^n is accepted with probability equal to λ is in coRP.*
2. *if a^n is accepted with probability greater than λ lies in $\mathsf{P}^{\mathsf{C}_3\mathsf{P}}$.*

Proof. The word a^n is accepted with probability $\mu_0\delta^n\eta_F$ where μ_0 is the initial distribution, δ the transition matrix and η_F the vector corresponding to the final states. In order to find out if this quantity is equal to λ, one can write a straight line program p that calculates $\mu_0\delta^n\eta_F - \lambda$. The program is the usual square-and-multiply algorithm for exponentiation and it is going to be $O(\log_2 n)$ long because the number of iterations in the algorithm is equal to the number of bits required to represent n. The value of the program p is equal to (greater than) 0 iff a^n is accepted with probability exactly (greater than) λ. Now, we can check if $val(p) = 0$ in coRP [21] and $val(p) > 0$ in $\mathsf{P}^{\mathsf{C}_3\mathsf{P}}$ [2]. The result follows. □

We will now show that if a limit isolated PFA accepts a word with probability exactly λ then this word cannot be too long. This fact is proved in Lemma 3 with the help of auxiliary Propositions 4, 5 and 6. We start by proving a result about irreducible stochastic matrices. Recall that $\widehat{\delta^t}$ is used to denote the limit of the sequence $\lim_{r\to\infty} \delta^{rt}$.

Proposition 4. *Given an irreducible stochastic matrix δ with period p and rational $\epsilon \in (0,1)$ there exists a number k, computable in polynomial time, such that for all $\ell \geq k : \boldsymbol{d}(\delta^{p\ell}, \widehat{\delta^p}) \leq \epsilon$.*

Proof. A stochastic matrix γ with all positive entries acts as a contraction map on the set of distributions. The associated contraction factor α is $(1-ns)$ where s is the smallest entry in γ (see Proposition 2). So we have

$$\begin{aligned}\boldsymbol{d}(\mu\gamma^i, \mu\widehat{\gamma}) &= \lim_{j\to\infty} \boldsymbol{d}(\mu\gamma^i, \mu\gamma^j) \leq \lim_{j\to\infty} \sum_{i'=i}^{j-1} \boldsymbol{d}(\mu\gamma^{i'}, \mu\gamma^{i'+1}) \\ &\leq \lim_{j\to\infty} \sum_{i'=i}^{j-1} \alpha^{i'}\, \boldsymbol{d}(\mu, \mu\gamma) \leq \frac{\alpha^i}{1-\alpha} = \frac{(1-ns)^i}{ns} \leq \frac{e^{-nsi}}{ns}.\end{aligned}$$

Choosing $i > \frac{1}{ns}\log\frac{2}{ns\epsilon}$ will give us $\boldsymbol{d}(\mu\gamma^i, \mu\widehat{\gamma}) \leq \frac{\epsilon}{2}$ and because the μ is arbitrary we also have $\boldsymbol{d}(\gamma^i, \widehat{\gamma}) \leq \epsilon$.

Coming back to δ, the graph of δ^p consists of p disjoint irreducible and aperiodic components. It is enough to show the above bound on each of the individual components (because the distance between the matrices takes maximum across rows), so consider δ^p to be irreducible and aperiodic. From Lemma 1, we know that δ^{pn^2} has all positive entries. The smallest entry of δ^{pn^2}, say s, requires only

polynomially many bits to be represented. According to the above observation, for $i \geq \frac{1}{ns} \log \frac{2}{ns\epsilon}$ we have $\boldsymbol{d}(\delta^{pn^2 i}, \widehat{\delta^p}) \leq \epsilon$. If $\frac{1}{ns\epsilon} = \frac{x}{y}$, and j represents the number of bits of y then we can choose $k = \lceil \frac{n}{s}(j+1) \rceil$, which is computable in polynomial time. □

We now bound the number of steps required so that the probability of being in a transient state is small.

Proposition 5. *Given a stochastic matrix δ and rational $\epsilon \in (0,1)$ there exists a number k, computable in polynomial time such that for all $\ell \geq k$ it is the case that for all distributions μ_0, $\sum_{j \in T_\delta} \mu_0 \delta^\ell(j) \leq \epsilon$ where T_δ is the set of transient states of δ.*

Proof. Here we are required to show that after k steps the probability of being in a transient state is small. Every transient state has a path of length at most n to at least one terminal state, so choose one for each transient state. Let u be the minimum probability associated with any of those paths. So after every n steps each transient state loses at least u fraction of its probability to a terminal state, or in other words the probability of being in any transient state reduces by a factor of u. Hence after $k'n$ steps the probability of being in a transient state is at most $(1-u)^{k'}$, and choosing $k' \geq \frac{1}{u} \log \frac{1}{\epsilon}$ makes $(1-u)^{k'} \leq \epsilon$. So choosing k to be a number bigger then $\frac{n}{u} \log \frac{1}{\epsilon}$ we have our required number. □

We now bound the length of input needed to be close to the limit distribution $\mu\widehat{\delta^p}$ where p is the ultimate period of δ.

Proposition 6. *Given a stochastic matrix δ, a distribution μ and rational $\epsilon \in (0,1)$ there exists a k, computable in polynomial time such that for all $\ell \geq k$: $\boldsymbol{d}(\mu\delta^{p\ell}, \mu\widehat{\delta^p}) \leq \epsilon$ where p is the ultimate period of δ.*

Proof. (Sketch.) First we use Proposition 5 to get a k_1 such that it suffices to take k_1 steps to get to a distribution where the probability of being in any transient state is less than $\frac{\epsilon}{4}$. This ensures that for $l \geq k_1$, the probability of being in any BSCC c after pl steps is at least $1 - \frac{\epsilon}{4}$. This means that taking any more steps beyond k_1 can only perturb the probability in terminal states by a small amount which adds up to $\frac{\epsilon}{4}$ across all BSCCs. Let us focus on one BSCC c. Taking, k_2 steps beyond the k_1 will do two things to c:

i) bring in more probability from the transient states
ii) distribute the probability already present in c (i.e., the probability of being in c) at step k_1 according to μ_c, the stationary distribution of c.

The first effect can only result in pumping at most a small probability into c, which adds at most $\frac{\epsilon}{4}$ to the distance. The probability already present in c after k_1 steps is close to the limiting probability, and hence the contribution of the second effect into the distance can be made small by choosing k_2 according to Proposition 4 for the BSCC c with the bound $\frac{\epsilon}{4}$. Instead of choosing k_2 for a particular c, we can choose it to be the maximum across all c which will give us the desired result. We formalize these ideas in the calculations included in the Appendix. □

We can prove a similar result about the matrix products as well.

Lemma 3. *Given a stochastic matrix δ and a rational $\epsilon \in (0,1)$, there exists a number k, computable in polynomial time, such that for all $\ell \geq k : \boldsymbol{d}(\delta^{p\ell}, \widehat{\delta^p}) \leq \epsilon$ where p is the ultimate period of δ.*

Proof. The distance between the matrices can be broken into

$$\begin{aligned}\boldsymbol{d}(\delta^{p\ell}, \widehat{\delta^p}) &= \max_i \sum_j |\delta^{p\ell}(i,j) - \widehat{\delta^p}(i,j)| = \max_i \sum_j |\nu_i\delta^{p\ell}(j) - \nu_i\widehat{\delta^p}(j)| \\ &= \max_i (2\,\boldsymbol{d}(\nu_i\delta^{p\ell}, \nu_i\widehat{\delta^p})). \qquad \text{Here } \nu_i \text{ represents the distribution with probability 1 at state } i\end{aligned}$$

Proposition 6 tells us we can choose a k of appropriate size such that for any μ, the distance $\boldsymbol{d}(\mu\delta^{p\ell}, \mu\widehat{\delta^p})$ for $\ell \geq k$ is below $\frac{\epsilon}{2}$. □

We are ready to establish the complexity of the problem of checking if λ is an isolated cut-point for a unary PFA A.

Theorem 3. *Given a unary PFA A and a rational λ, the problem of checking if A is isolated at λ is in* $\mathsf{coNP}^{\mathsf{RP}}$ *and is* coNP*-hard.*

Proof. The lower bound follows from the proof of Theorem 2. For the $\mathsf{coNP}^{\mathsf{RP}}$ upper bound, let us consider the complement of the problem where A is not limit isolated at λ. In this case either λ is not a limited isolated cut-point or there is some string which is accepted with probability λ. If the given PFA is not limit isolated then we guess this fact and check if it is true in NP (Thanks to Theorem 2). So assume that the PFA is limit isolated. We now need to check if there is any "short" string accepted with probability λ.

Let $A = (Q, \Sigma, \delta, \mu_0, Q_F)$ and let p be the ultimate period of A. Let $\widehat{\delta^p} = \lim_{t\to\infty} \delta^{pt}$. For each $r > 0$, let $\mu_r = \mu_0\delta^r$.

Consider $\epsilon_r = |\mu_r\widehat{\delta^p}\eta - \lambda|$. Since any $\mu_r\widehat{\delta^p}$ can be computed in polynomial time (see proof of Theorem 2), it is the case that ϵ_r can be computed in polynomial time (given μ_0, δ, r). Suppose the length of the string accepted with probability λ is ℓ. Let $\ell = pq + r$ where $r = \ell \bmod p$. According to Lemma 3, there exists a k_r (computable in polynomial time) such that if $q > k_r$ then $\boldsymbol{d}(\mu_r\delta^{pq}, \mu_r\widehat{\delta^p}) \leq \frac{\epsilon_r}{2}$. Since $\boldsymbol{d}(\mu_r\delta^{pq}, \mu_r\widehat{\delta^p}) \geq |\mu_r\delta^{pq}\eta - \mu_r\widehat{\delta^p}\eta|$, we get that a^ℓ will not be accepted with probability λ if $q > k_r$.

Now, the decision procedure for checking non-isolation proceeds as follows. It first guesses $0 \leq r < p$, then it computes ϵ_r and subsequently computes k_r. Now, it guesses $q \leq k_r$ and then it computes $\ell = pq + r$. ℓ requires only polynomially many bits (because k_r is computable in polynomial time from r). Hence we can use the procedure of Lemma 2 as an oracle to check if a^ℓ is accepted with probability exactly λ. Note that this final check is done by a coRP algorithm and hence the non-isolation is in $\mathsf{NP}^{\mathsf{coRP}}$. Note that $\mathsf{NP}^{\mathsf{coRP}}$ is exactly the class $\mathsf{NP}^{\mathsf{RP}}$ since we can always switch the yes/no answer from the oracle-calls. This results in a $\mathsf{coNP}^{\mathsf{RP}}$ upper bound for the limit isolation problem in the non-extremal case. □

Extremal cut-points. For extremal cut-points, the upper bound matches the lower bound.

Theorem 4. *Given a unary PFA A, the problem of checking if 0 is isolated is* coNP*-complete. Similarly checking if 1 is isolated is also* coNP*-complete.*

Proof. The lower bound follows from the proof of Theorem 2. For upper bound, first thing to note is that the coNP upper bound for limit isolation proved in Theorem 2 also holds for the cut-point 0. So in case it is limit isolated, we need to check if there is a string accepted with probability 0. If μ_0 is the initial distribution, let $Q_I = \{q|\mu_0(q) > 0\}$. A word is accepted by A with probability 0 iff it has no path from a state in Q_I to a final state. So checking if 0 is isolated reduces to the universality checking of unary NFA which is known to be in coNP [23]. □

5 Other Decidable Problems

In this section, we observe that the problems of language containment, equality, emptiness, and universality are all in counting hierarchy for unary PFAs with limit isolated cut-points. We need one proposition.

Proposition 7. *Given a PFA A with ultimate period p, a number $0 \leq r < p$ and a rational cut-point λ such that λ is limit isolated for A, there is a number k computable in polynomial time s.t $\forall q \geq k$, $a^{pq+r} \in L_{>\lambda}(A)$ iff $a^{pk+r} \in L_{>\lambda}(A)$.*

Proof. Let $A = (Q, \Sigma, (\delta_\sigma)_{\sigma\in\Sigma}, \mu_0, Q_F)$. Let $\mu_r = \mu_0\delta^r$, let $\widehat{\delta^p} = \lim_{t\to\infty} \delta^{pt}$ and let η_F be the vector corresponding to F. Consider $\epsilon_r = |\mu_r\widehat{\delta^p}\eta_F - \lambda|$. ϵ_r can be computed in polynomial time (see proof of Theorem 2). According to Lemma 3, there is a k computable in polynomial time such that if $q > k$ then $\boldsymbol{d}(\mu_r\delta^{pq}, \mu_r\widehat{\delta^p}) \leq \frac{\epsilon_r}{2}$. Since $\boldsymbol{d}(\mu_r\delta^{pq}, \mu_r\widehat{\delta^p}) \geq |\mu_r\delta^{pq}(Q_f) - \mu_r\widehat{\delta^p}(Q_f)|$, we get that a^{pq+r} has acceptance probability $> \lambda$ iff a^{pk+r} has acceptance probability $> \lambda$.

Theorem 5. *Given two unary PFAs A and B and rational cut-points λ_1 and λ_2, such that λ_1 and λ_2 are limit isolated for A and B respectively, the following problems are in* $\mathsf{coNP}^{\mathsf{C_3P}}$.

1. $L_{>\lambda_1}(A) \subseteq L_{>\lambda_2}(B)$.
2. $L_{>\lambda_1}(A) = L_{>\lambda_2}(B)$.
3. $L_{>\lambda_1}(A) = \emptyset$.
4. $L_{>\lambda_1}(A) = \Sigma^*$.

Proof. Without loss of generality, we can assume that the ultimate periods of A and B are the same (since we can always add unreachable cycles). Let the ultimate period be p. The algorithm for checking containment proceeds as follows. The algorithm is going to guess a number ℓ such that a^ℓ is accepted by A and rejected by B. Note, ℓ can be written as $\ell = pq + r$ where $q = \ell \mathsf{\ div\ } p$ and $r = \ell \mathsf{\ mod\ } p$. Hence we have to guess q and r. First, the algorithm guesses the offset $0 \leq r < p$ which is a polynomial-sized number.

Thanks to Proposition 7, there is a k_A such for all $q_A \geq k_A$, a^{pq_A+r} is accepted by A iff the string a^{pk_A+r} is accepted. Furthermore, k_A can be computed in

polynomial time from A and r. Similarly, there is a k_B such that for all $q_B \geq k_B$, a^{pq_B+r} is accepted by B iff the string a^{pk_B+r} is accepted. Let $k = \max(k_A, k_B)$.

By construction, we can conclude that if a^ℓ with $\ell = pq + r$ is in the language of A but not in the language of B then we can take $q \leq k$. So, now the algorithm guesses $q \leq k$ and then checks that i) $a^\ell \in L_{>\lambda_1}(A)$ and ii) $a^\ell \notin L_{>\lambda_2}(B)$. These checks can be carried out by $\mathsf{P}^{\mathsf{C}_3\mathsf{P}}$ algorithms as in Lemma 2 and the result follows. The other problems of language equality, emptiness, and universality follow immediately from the result for language containment. □

6 Conclusions

In this paper we established the complexity of a variety of decision problems for unary PFAs. In particular, we showed that the isolation problem is in coNP-complete, when the cut-point is extremal, and is in $\mathsf{coNP}^{\mathsf{RP}}$ when the cut-point is not extremal. We also show that limit isolation of unary PFAs allows us to conclude that language, containment, equality, emptiness, and universality are decidable within PSPACE.

Acknowledgements: We thank anonymous referees for pointing out that EquSLP is in coRP, and observing that Bertoni had claimed the decidability of the isolation problem in the conclusions of [4]. Rohit Chadha was supported by NSF grant CNS 1314338. Dileep Kini was supported by NSF grant SHF 1016989. Mahesh Viswanathan was supported by NSF grant CNS 1314485.

References

1. Manindra Agrawal, S. Akshay, Blaise Genest, and P. S. Thiagarajan. Approximate verification of the symbolic dynamics of markov chains. In *LICS*, pages 55–64, 2012.
2. Eric Allender, Peter Bürgisser, Johan Kjeldgaard-Pedersen, and Peter Bro Miltersen. On the complexity of numerical analysis. *SIAM J. Comput.*, 38(5):1987–2006, 2009.
3. S. Arora and B. Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009.
4. A. Bertoni. The solution of problems relative to probabilistic automata in the frame of formal language theory. In *Proceedings of the 4th GI Jahrestagung*, volume 26 of *Lecture Notes in Computer Science*, pages 107–112, 1974.
5. R. Chadha, A. P Sistla, and M. Viswanathan. Probabilistic automata with isolated cut-points. In *MFCS*, volume 8087 of *Lecture Notes in Computer Science*, pages 254–265. Springer, 2013.
6. R. Chadha, A.P. Sistla, and M. Viswanathan. Probabilistic automata with isolated cut-points. In *38th International Symposium on Mathematical Foundations on Computer Science*, pages 254–265, 2013.
7. Anne Condon and Richard 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.
8. L. Doyen, T. A. Henzinger, and J.-F. Raskin. Equivalence of labeled markov chains. *Inernational Journal of Foundations of Computer Science*, 19(3):549–563, 2008.

9. F.R. Gantmacher. *Applications Of The Theory Of Matrices.* Dover Books on Mathematics Series. Dover, 2005.
10. H. Gimbert and Y. Oualhadj. Probabilistic automata on finite words: Decidable and undecidable problems. In *Proceedings of the International Colloquium on Automata, Languages and Programming*, pages 527–538, 2010.
11. Stefan Kiefer, Andrzej S. Murawski, Joël Ouaknine, Björn Wachter, and James Worrell. Language equivalence for probabilistic automata. In *Proceedings of the 23rd International Conference on Computer Aided Verification (CAV)*, volume 6806 of *LNCS*, pages 526–540, Snowbird, Utah, USA, 2011. Springer.
12. Stefan Kiefer, Andrzej S. Murawski, Joël Ouaknine, Björn Wachter, and James Worrell. On the complexity of the equivalence problem for probabilistic automata. In *Proceedings of the 15th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS)*, volume 7213 of *LNCS*, pages 467–481, Tallinn, Estonia, 2012. Springer.
13. Vijay Anand Korthikanti, Mahesh Viswanathan, Gul Agha, and YoungMin Kwon. Reasoning about mdps as transformers of probability distributions. In *QEST*, pages 199–208, 2010.
14. Y. Kwon and G. Agha. Linear Inequality LTL (iLTL): A Model Checker for Discrete Time Markov Chains. *Lecture Notes In Computer Science*, pages 194–208, 2004.
15. Y.M. Kwon and G. Agha. A Markov Reward Model for Software Reliability. In *Parallel and Distributed Processing Symposium, 2007. IPDPS 2007. IEEE International*, pages 1–6, 2007.
16. J.R. Norris. *Markov Chains.* Cambridge University press, 1997.
17. A. Paz. *Introduction to probabilistic automata.* Academic Press, 1971.
18. M.O. Rabin. Probabilistic automata. *Information and Computation*, 6(3):230–245, 1963.
19. J. M. Rutten, M. Kwiatkowska, G. Norman, and D. Parker. *Mathematical Techniques for Analyzing Concurrent and Probabilistic Systems.* AMS, 2004.
20. A. Salomaa, M. Soittola, F.L. Bauer, and D. Gries. *Automata-theoretic aspects of formal power series.* Texts and monographs in computer science. Springer-Verlag, 1978.
21. Arnold Schönhage. On the power of random access machines. In Hermann A. Maurer, editor, *ICALP*, volume 71 of *Lecture Notes in Computer Science*, pages 520–529. Springer, 1979.
22. E. Senata. *Non-negative Matrices and Markov Chains.* George Allen & Unwin Ltd, 1973.
23. Larry J. Stockmeyer and Albert R. Meyer. Word problems requiring exponential time: Preliminary report. In *Proc. of the 5th Ann. ACM Symposium on Theory of Computing,*, pages 1–9, 1973.
24. Wen-Guey Tzeng. A polynomial-time algorithm for the equivalence of probabilistic automata. *SIAM J. Comput.*, 21(2):216–227, April 1992.
25. Klaus W. Wagner. Some observations on the connection between counting and recursion. *Theoretical Computer Science*, 47(3):131–147, 1986.

Appendix

Calculations from Proposition 6: A *sub-distribution* over a finite set S is a function $\mu : S \mapsto [0, 1]$ such that $\sum_{s \in S} \mu(s) \leq 1$. The distance between sub-distributions can be defined in the same way we do for distributions.

We first describe the notation we will use in the following calcualtions: $\mu_{\mathcal{T}}$ and $\mu_{\mathcal{C}}$ denote the sub-distributions on the transient states and the BSCCs after pk_1 steps. For any (sub-)distribution μ , $\mu \restriction c$ denotes the vector with the entries in the states in c alone. The matrix δ restricted to the states of c is written as δ_c. Starting from μ, and having taken pk_1 steps, the probability of being in component c is denoted by $\pi_c^{pk_1}$, and the relative distribution on a component c is given by $\mu_c^{pk_1}$, i.e for any $i \in c$, $\mu_c^{pk_1}(i) = \mu\delta^{pk_1}(i)/\pi_c^{pk_1}$. Starting from μ the probability of being in c in the limit is given by $\hat{\pi_c}$ and the relative distribution on c in the limit is give by $\hat{\mu_c}$. Now we are ready to proceed.

$$\begin{aligned}
&\boldsymbol{d}\big(\mu\delta^{p\ell}, \mu\widehat{\delta^p}\big) = \boldsymbol{d}\big(\mu\delta^{pk_1}\delta^{pk_2}, \mu\widehat{\delta^p}\big) = \boldsymbol{d}\big((\mu_{\mathcal{T}} + \mu_{\mathcal{C}})\delta^{pk_2}, \mu\widehat{\delta^p}\big) \\
=&\boldsymbol{d}\big(\mu_{\mathcal{T}}\delta^{pk_2} + \mu_{\mathcal{C}}\delta^{pk_2}, \mu\widehat{\delta^p}\big) \leq \underbrace{\frac{\sum_j \mu_{\mathcal{T}}\delta^{pk_2}(j)}{2}}_{\text{Apply Prop 5}} + \boldsymbol{d}\big(\mu_{\mathcal{C}}\delta^{pk_2}, \mu\widehat{\delta^p}\big)
\end{aligned}$$

Let us focus on $\boldsymbol{d}\big(\mu_{\mathcal{C}}\delta^{pk_2}, \mu\widehat{\delta^p}\big)$

$$= \sum_{c\in\mathcal{C}} \boldsymbol{d}\big((\mu_{\mathcal{C}}\delta^{pk_2}) \restriction c\,, (\mu\widehat{\delta^p}) \restriction c\big) = \sum_{c\in\mathcal{C}} \boldsymbol{d}\big(\pi_c^{pk_1}\mu_c^{pk_1}\delta_c^{pk_2},\ \hat{\pi_c}\ \hat{\mu_c}\widehat{\delta_c{}^p}\big)$$

We have $(\mu_{\mathcal{C}}\delta^{pk_2}) \restriction c\ = \pi_c^{pk_1}\mu_c^{pk_1}\delta_c^{pk_2}$ because when we start from $\mu_{\mathcal{C}}$ there is no probability of being in any transient state, so we can ignore the transient states, and then the BSCCs cannot communicate so they evolve independently.

$$\begin{aligned}
&= \sum_{c\in\mathcal{C}}\sum_{j\in c} \big|\pi_c^{pk_1}\mu_c^{pk_1}\delta_c^{pk_2}(j) - \hat{\pi_c}\ \hat{\mu_c}\widehat{\delta_c{}^p}(j)\big| \\
&= \sum_{c\in\mathcal{C}}\sum_{j\in c} \big|\pi_c^{pk_1}\big(\mu_c^{pk_1}\delta_c^{pk_2}(j) - \hat{\mu_c}\widehat{\delta_c{}^p}(j)\big) + (\pi_c^{pk_1} - \hat{\pi_c}\ )\hat{\mu_c}\widehat{\delta_c{}^p}(j)\big| \\
&\leq \Big(\sum_{c\in\mathcal{C}} \pi_c^{pk_1} \sum_{j\in c} \big|\mu_c^{pk_1}\delta_c^{pk_2}(j) - \hat{\mu_c}\widehat{\delta_c{}^p}(j)\big|\Big) + \Big(\sum_{c\in\mathcal{C}} (\hat{\pi_c} - \pi_c^{pk_1}) \sum_{j\in c} \hat{\mu_c}\widehat{\delta_c{}^p}(j)\Big) \\
&\leq \Big(\sum_{c\in\mathcal{C}} \pi_c^{pk_1} \overbrace{\boldsymbol{d}\big(\mu_c^{pk_1}\delta_c^{pk_2}(j),\ \hat{\mu_c}\widehat{\delta_c{}^p}(j)\big)}^{\text{Apply Prop 4}}\Big) + \overbrace{\sum_{c\in\mathcal{C}}(\hat{\pi_c} - \pi_c^{pk_1})}^{\text{Apply prop 5}} \\
&\leq \Big(\sum_{c\in\mathcal{C}} \pi_c^{pk_1}\frac{\epsilon}{2}\Big) + \frac{\epsilon}{4} \leq \frac{3\epsilon}{4}
\end{aligned}$$

Therefore $\boldsymbol{d}\big(\mu\delta^{\ell}, \mu\widehat{\delta^p}\big) \leq \epsilon$.