Emptiness under isolation and the Value problem for hierarchical probabilistic automata

Rohit Chadha[1], A. Prasad Sistla[2], and Mahesh Viswanathan[3]

[1] University of Missouri, USA
chadhar@missouri.edu
[2] University of Illinois at Chicago, USA
sistla@uic.edu
[3] University of Illinois at Urbana-Champaign, USA
vmahesh@illinois.edu

Abstract. k-Hierarchical probabilistic automata (k-HPA) are probabilistic automata whose states are stratified into $k+1$ levels such that from any state, on any input symbol, at most one successor belongs to the same level, while the remaining belong to higher levels. Our main result shows that the emptiness and universality problems are decidable for k-HPAs with isolated cut-points; recall that a cut-point x is isolated if the acceptance probability of every word is bounded away from x. Our algorithm for establishing this result relies on computing an approximation of the value of an HPA; the value of a probabilistic automaton is the supremum of the acceptance probabilities of all words. Computing the exact value of a probabilistic automaton is an equally important problem and we show that the problem is **co-R.E.**-complete for k-HPAs, for $k \geq 2$ (as opposed to $\mathbf{\Pi^0_2}$-complete for general probabilistic automata). On the other hand, we also show that for 1-HPAs the value can be computed in exponential time.

1 Introduction

k-Hierarchical probabilistic automata (HPAs) [11] are a syntactic sub-class of probabilistic automata, whose states are stratified into $k+1$ levels. Like probabilistic automata, the next state on an input symbol is determined stochastically. However, transitions are required to "respect levels" — from any state q, on any input symbol a, at most one possible next state belongs to the same level as q, with the others being constrained to belong to levels higher than q's. Such automata can recognize languages over finite (hierarchical probabilistic finite automata) or infinite words (hierarchical probabilistic Muller automata) depending on the notion of accepting runs. Given a threshold x, the language recognized by an HPA $\mathcal{A}$ is the collection of all input strings such that the measure of all accepting runs on the input is $> x$.

HPAs arise naturally as models of client-server systems with stochastic server failures, concurrent systems under probabilistic context-bounded schedulers, and business enterprise systems [2] and these have been analyzed using automated tools for HPAs [4]. HPAs were introduced in [11] as a computationally tractable subclass of probabilistic automata. When the acceptance threshold is extremal, i.e., 0 or 1, many verification problems for HPAs become decidable. While (general) probabilistic Büchi automata can recognize non-regular languages with acceptance threshold 0 and 1, HPAs were shown to recognize only regular languages [11]. Classical (qualitative) verification questions like emptiness and universality are decidable in low complexity classes (**NL** and **PSPACE**). In contrast, the emptiness problem for probabilistic Büchi automata with threshold 0 is undecidable [1] and $\mathbf{\Pi^0_2}$-complete [11, 9].

Surprisingly, however, the landscape changes completely when the threshold is taken to be $x \in (0, 1)$. Even 1-HPAs [4] can recognize non-regular languages when the acceptance threshold is $\frac{1}{2}$ [10]. Though emptiness and universality problems are decidable for 1-HPAs [10], these problems are undecidable for 2-HPAs (and higher) [4]. In this paper, we present results that support the thesis that, despite the many negative results about HPAs in [8, 4], HPAs are indeed a computationally tractable model of open probabilistic systems. Specifically, we present results that show that the value of HPAs can be approximated to a given degree of precision ϵ unlike general probabilistic automata. Hence HPAs can be "approximately verified"; we can declare the language of a PFA to be non-empty/empty if the value of the HPA is at least ϵ more/less than the threshold.

The main results in this paper pertain to HPAs with isolated cut-points and the value problem for HPAs. A threshold x is said to be isolated for a probabilistic automaton (not necessarily hierarchical) $\mathcal{A}$, if there is an $\epsilon > 0$ such that the acceptance probability of any word is either at most $x - \epsilon$ or at least $x + \epsilon$, i.e., the probability of acceptance of any word is bounded away from x. Automata with isolated cut-points describe algorithms to which algorithmic techniques like amplification can be applied, and are constant space analogs of complexity classes **BPP** and **RP**. An important classical result due to Rabin [18] is that though probabilistic automata over finite words (PFA) can recognize non-regular languages when the threshold $x \in (0, 1)$, they only recognize regular languages when x is isolated. The extension of this result to automata on infinite words is not known. In this paper, we show that HPAs on infinite words with isolated cut-points recognize ω-regular languages.

Even though probabilistic finite automata (PFAs) with isolated cut-points recognize regular languages, it is not known if the following problem is decidable: Given a PFA with an isolated cut-point x, determine if some input string is accepted with probability $> x$. Our main result is that for HPAs with isolated cut-points, this emptiness problem is decidable. Our result applies to both HPAs over finite words and HPAs over infinite words. In fact, we show that checking if an HPA's (with isolated cut-point) language is equal to any given regular language is decidable; thus, even checking universality is decidable.

Our proof for the decidability of emptiness under isolation is based on solving another classical problem for probabilistic automata, namely, computing the

[4] 0-HPAs are just deterministic machines. Thus, 1-HPAs are automata with fewest number of levels that have some stochastic behavior.

value of an automaton. The value of an automaton is the *least upper bound* of the acceptance probabilities of all words. The decision version of the value problem is known to undecidable [5, 15, 12]. We show that for HPAs (over finite or infinite words) the value can be *approximated* to precision γ ($\gamma < 1$) in time that is doubly exponential in the size of the automaton and exponential in $\mathsf{poly}(\log(\frac{1}{\gamma}))$. The approximation algorithm is obtained by observing that in an HPA, for any finite word v, there is a "short" word u such that the distribution on states after u is very "close" to the distribution after input v; the length of u only depends on the size of the automaton and the approximation factor. Thus to approximate the value of an HPA up-to γ, we compute the maximum of the acceptance probabilities of all "short" words. Having an algorithm to approximate the value of an automaton immediately gives us an algorithm for checking language emptiness of an HPA $\mathcal{A}$ with isolated cut-point x as follows. We progressively compute the value of $\mathcal{A}$ with increasing precision. Suppose at some point the value is approximated by v with precision γ. If $v-\gamma > x$ then we know that $\mathcal{A}$ has a non-empty language. On the other hand, if $v + \gamma < x$ then $\mathcal{A}$'s language is empty. Since x is isolated, we are guaranteed that eventually the precision γ is low enough to ensure that one of these two conditions hold.

In addition to the algorithm to approximate the value of an HPA, we characterize the precise complexity of the value problem for HPAs (on both finite and infinite words). We show that the value problem is in **EXPTIME** for 1-HPAs, as follows. First, we prove that the value of a 1-HPA is a fraction whose size (i.e., the length of the binary representation of its denominator) is at most exponential in the number of states of the automaton. Then, we present an algorithm that computes the value exactly, by employing binary search on rationals [19], together with an algorithm for emptiness checking for 1-HPA [10]. We show that the value problem is **co-R.E.**-complete for 2-HPAs (and higher). In contrast, for general PFAs, the value problem is known to be $\mathbf{\Pi_2^0}$-complete [12]. Finally, we also show that the problem of checking if a cut-point x is not isolated for a probabilistic automaton $\mathcal{A}$ can be reduced in polynomial time to the value problem. This, along with the results for the value problem for HPAs, shows that the problem of checking isolation is **R.E.**-complete for 2-HPAs (and higher), and is in **co-R.E.** for 1-level HPA.

The paper is organized as follows. We describe closely related work next. Section 2 contains notations and definitions. Section 3 has definitions of HPAs and results on the regularity of the language under isolated cut points. Section 4 has results on the approximation of HPAs and decidability of emptiness under isolation. Section 5 has the results for the value problem and isolated cut point problem. Section 6 has conclusions.

Related Work. We summarize results on the emptiness problem, the value problem, and the isolation problem. The undecidability of the emptiness problem for probabilistic automata with non-extremal thresholds was shown for finite words in [13] and for infinite words in [8]. The emptiness problem for 2-HPAs (and higher) with non-extremal thesholds is also undecidable [8, 4]. When the cut-point is isolated, the decidability of the emptiness problem for general probabilistic automata is not known. However, for unary PFAs [6] and eventually weakly ergodic PFAs [12] the emptiness problem is decidable when the cut-point is isolated. Eventually weakly ergodic PFAs are incomparable to HPAs considered here; Figure 1(c) in [12] is an example of a HPA that is not eventually weakly ergodic. The value problem is undecidable for PFAs [5], even for extremal thresholds [15]; it is known to be $\mathbf{\Pi^0_2}$-complete [12]. The problem of checking if the value is 1 was shown to be **PSPACE**-complete for *leak-tight automata* [14] which is sub-class of probabilistic automata that includes HPAs considered here. For value other than 1, no decidability results are known (other than those presented here). The isolation problem was shown to be $\mathbf{\Pi^0_2}$-complete [12] for general probabilistic automata.

2 Preliminaries

We assume that the reader is familiar with probability distributions, stochastic matrices finite-state automata, regular languages, Muller automata and ω-regular languages. The set of natural numbers will be denoted by $\mathbb{N}$, the closed unit interval by $[0, 1]$ and the open unit interval by $(0, 1)$. The power-set of a set X will be denoted by 2^X. The absolute value of a real number r shall be denoted by $|r|$. A non-negative rational number x is uniquely represented as a fraction $\frac{y}{z}$ where $y, z \in \mathbb{N}$ are relatively prime to each other, and $y \leq z$. In this case, we define the *size* of x to be the number of bits in the binary representation of z.

Sequences. Given a finite set S, $|S|$ denotes the cardinality of S. Given a sequence (finite or infinite) $\kappa = s_0 s_1 \ldots$ over S, $|\kappa|$ will denote the length of the sequence (for infinite sequence $|\kappa|$ will be ω), and $\kappa[i]$ will denote the ith element s_i of the sequence with $\kappa[0]$ being the first. We will use ϵ to denote the (unique) empty string/sequence. For natural numbers i, j, $i \leq j < |\kappa|$, $\kappa[i : j]$ is the sequence $s_i \ldots s_j$. For $i < |\kappa|$, $\kappa[i : \infty]$ is the sequence $s_i s_{i+1} \ldots$ if $|\kappa| = \omega$, and is the sequence $s_i \ldots s_{|\kappa|-1}$ if $|\kappa|$ is finite. As usual S^* will denote the set of all finite sequences/strings/words over S, S^+ will denote the set of all finite non-empty sequences/strings/words over S and S^ω will denote the set of all infinite sequences/strings/words over S. We will use u, v, w to range over elements of S^*, α, β, γ to range over infinite words over S^ω.

Given $u \in S^*$ and $\kappa \in S^* \cup S^\omega$, $u\kappa$ is the sequence obtained by concatenating the two sequences in order. Given $\mathsf{L}_1 \subseteq S^*$ and $\mathsf{L}_2 \subseteq S^* \cup S^\omega$, the set $\mathsf{L}_1\mathsf{L}_2$ is defined to be $\{u\kappa \mid u \in \mathsf{L}_1 \text{ and } \kappa \in \mathsf{L}_2\}$. Given $u \in S^+$, the word u^ω is the unique infinite sequence formed by repeating u infinitely often. For an infinite word $\alpha \in S^\omega$, we write $\mathsf{inf}(\alpha) = \{s \in S \mid s = \alpha[i] \text{ for infinitely many } i\}$.

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

Definition 1. *A* finite-state probabilistic automaton (PA) *over a finite alphabet* Σ *is a tuple* $\mathcal{A} = (Q, q_s, \delta, \mathsf{Acc})$ *where* Q *is a finite set of states,* $q_s \in Q$ *is the initial state,* $\delta : Q \times \Sigma \times Q \to [0,1]$ *is the transition relation such that for all* $q \in Q$ *and* $a \in \Sigma$*,* $\delta(q,a,q')$ *is a rational number and* $\sum_{q' \in Q} \delta(q,a,q') = 1$*, and* Acc *is an* acceptance condition *(to be defined later).*

Notation: The transition function δ of PA $\mathcal{A}$ on input a can be seen as a square matrix δ_a of order $|Q|$ with the rows labeled by "current" state, columns labeled by "next state" and the entry $\delta_a(q,q')$ equal to $\delta(q,a,q')$. Given a word $u = a_0a_1\dots a_n \in \Sigma^+$, δ_u is the matrix product $\delta_{a_0}\delta_{a_1}\dots\delta_{a_n}$. For the empty word $\epsilon \in \Sigma^*$ we take δ_ϵ to be the identity matrix. Finally for any $Q_0 \subseteq Q$, we say that $\delta_u(q, Q_0) = \sum_{q' \in Q_0} \delta_u(q,q')$. Given a state $q \in Q$ and a word $u \in \Sigma^+$, $\mathsf{post}(q,u) = \{q' \mid \delta_u(q,q') > 0\}$. For a set $C \subseteq Q$, $\mathsf{post}(C,u) = \cup_{q \in C} \mathsf{post}(q,u)$.

Intuitively, the PA starts in the initial state q_s and if after reading $a_0, a_1 \dots, a_i$ it is in state q, then the PA moves to state q' with probability $\delta_{a_{i+1}}(q,q')$ on symbol a_{i+1}. A *run* of the PA $\mathcal{A}$ starting in a state $q \in Q$ on an input $\kappa \in \Sigma^* \cup \Sigma^\omega$ is a sequence $\rho \in Q^* \cup Q^\omega$ such that $|\rho| = 1 + |\kappa|, \rho[0] = q$ and for each $i \geq 0$, $\delta_{\kappa[i]}(\rho[i], \rho[i+1]) > 0$. Unless otherwise stated, a *run* for us will mean a run starting in the initial state q_s.

Given a word $\kappa \in \Sigma^* \cup \Sigma^\omega$, the PA $\mathcal{A}$ can be thought of as a (possibly infinite-state) (sub)-Markov chain. The set of states of this (sub)-Markov Chain is the set $\{(q,v) \mid q \in Q, v \text{ is a prefix of } \kappa\}$ and the probability of transitioning from (q,v) to (q',u) is $\delta_a(q,q')$ if $u = va$ for some $a \in \Sigma$ and 0 otherwise. This gives rise to the standard σ-algebra on Q^ω defined using cylinders and the standard probability measure on (sub)-Markov chains [20, 16]. We shall henceforth denote the σ-algebra as $\mathcal{F}_{\mathcal{A},\kappa}$ and the probability measure as $\mu_{\mathcal{A},\kappa}$.

Acceptance conditions and PA **languages**. The language of a PA $\mathcal{A} = (Q, q_s, \delta, \mathsf{Acc})$ over an alphabet Σ is defined with respect to the acceptance condition Acc and a threshold $x \in [0,1]$. We consider two kinds of acceptance conditions.

Finite acceptance: When defining languages over finite words, the acceptance condition Acc is given in terms of a finite set $Q_f \subseteq Q$. In this case we call the PA $\mathcal{A}$, a probabilistic finite automaton (PFA). Given a finite acceptance condition $Q_f \subseteq Q$ and a finite word $u \in \Sigma^*$, a run ρ of $\mathcal{A}$ on u is said to be accepting if the last state of ρ is in Q_f. The set of accepting runs on $u \in \Sigma^*$ is measurable [20] and we denote its measure by $\mathsf{P}_{\mathcal{A}}(u)$. Note that $\mathsf{P}_{\mathcal{A}}(u) = \delta_u(q_s, Q_f)$. Given a PFA, a rational threshold $x \in [0,1]$ and the language of finite words $\mathsf{L}_{>x}(\mathcal{A}) = \{u \in \Sigma^* \mid \mathsf{P}_{\mathcal{A}}(u) > x\}$ is the set of finite words accepted by $\mathcal{A}$ with probability $> x$.

Muller acceptance: For Muller acceptance, the acceptance condition Acc is given in terms of a finite set $F \subseteq 2^Q$. In this case, we call the PA $\mathcal{A}$, a probabilistic Muller automaton (PMA). Given a Muller acceptance condition $F \subseteq 2^Q$, a run ρ of $\mathcal{A}$ on an infinite word $\alpha \in \mathcal{A}$ is said to be *accepting* if $\mathsf{inf}(\rho) \in F$. Once again, the set of accepting runs is measurable [20]. Given a word α, the measure of the set of accepting runs is denoted by $\mathsf{P}_{\mathcal{A}}(\alpha)$. Given a PMA $\mathcal{A}$, a rational threshold $x \in [0,1]$, the language of infinite words $\mathsf{L}_{>x}(\mathcal{A}) = \{\alpha \in \Sigma^\omega \mid \mathsf{P}_{\mathcal{A}}(\alpha) > x\}$ is the set of infinite words accepted by PMA $\mathcal{A}$ with probability $> x$.

Changing the cutpoint. The following proposition allows us to change non-extremal cutpoints. It is proved for PFAs in [17]. The same construction also works for PMAs.

Proposition 1. *For any* PA $\mathcal{A}$*, rationals* $x, y \in (0,1)$*, there is a* PA $\mathcal{B}$ *constructible in linear time such that* $\mathsf{L}_{>x}(\mathcal{A}) = \mathsf{L}_{>y}(\mathcal{B})$*.*

The value problem. For a PA $\mathcal{A}$, let $\mathsf{value}(\mathcal{A})$ denote the least upper bound of the set $\{\mathsf{P}_{\mathcal{A}}(u) \mid u \in \Sigma^*\}$ when $\mathcal{A}$ is a PFA and of the set $\{\mathsf{P}_{\mathcal{A}}(\alpha) \mid \alpha \in \Sigma^\omega\}$ when $\mathcal{A}$ is a PMA. The *value computation problem* for a PA is the problem of computing $\mathsf{value}(\mathcal{A})$ for a given $\mathcal{A}$. The *value decision problem* is the problem of deciding for a given PA $\mathcal{A}$ and a rational $x \in [0,1]$ whether $\mathsf{value}(\mathcal{A}) = x$.

The isolation decision problem. For a PA $\mathcal{A}$, a rational threshold $x \in [0,1]$ is said to be an *isolated cut-point* of $\mathcal{A}$ if there is an $\epsilon > 0$ such that for each word κ (where $\kappa \in \Sigma^*$ when $\mathcal{A}$ is a PFA and $\kappa \in \Sigma^\omega$ otherwise), we have that $|\mathsf{P}_{\mathcal{A}}(\kappa) - x| > \epsilon$. If such an ϵ exists then ϵ is said to be *a degree of isolation.* The *isolation decision problem* is the problem of deciding for a given PA $\mathcal{A}$ and a rational $x \in [0,1]$ whether x is an isolated cutpoint of $\mathcal{A}$.

We have the following relation between the isolated cutpoint decision problem and the value decision problem. (See Appendix A for a proof).

Proposition 2. *For each* PA $\mathcal{A} = (Q, q_s, \delta, \mathsf{Acc})$ *and* $x \in (0,1)$*, there is a constructible* PA $\mathcal{B}$ *such that* $\mathsf{value}(\mathcal{B}) = \frac{1}{4}$ *iff* x *is not a isolated cutpoint of* $\mathcal{A}$*.*

3 Hierarchical Probabilistic Automata

Intuitively, a hierarchical probabilistic automaton is a PA such that the set of its states can be stratified into totally-ordered levels. From a state q on each letter a, the machine can transit with non-zero probability to at most one state in the same level as q, and all other probabilistic successors belong to higher levels.

Definition 2. *For* $k \in \mathbb{N}$*, a* k-hierarchical probabilistic automaton *(*HPA*) is a probabilistic automaton* $\mathcal{A} = (Q, q_s, \delta, \mathsf{Acc})$ *over alphabet* Σ *such that* Q *can be partitioned into* $k+1$ *sets* $Q_0, Q_1, \ldots, Q_k$ *satisfying the following properties:*

- $q_s \in Q_0$*;*
- *for every* i*,* $0 \le i \le k$ *and every* $q \in Q_i$ *and* $a \in \Sigma$*,* $|\mathsf{post}(q,a) \cap Q_i| \le 1$*; and,*
- *for every* i*,* $0 < i \le k$*,* $q \in Q_i$ *and* $a \in \Sigma$*,* $\mathsf{post}(q,a) \cap Q_j = \emptyset$ *for every* $j < i$*.*

For any k-HPA $\mathcal{A}$*, as given above, for* $0 \le i \le k$*, we call the elements of* Q_i*,* level i *states of* $\mathcal{A}$*. We call a* HPA *a* HPFA/HPMA *if* Acc *is a finite acceptance/Muller acceptance condition respectively.*

Let us fix a k-HPA $\mathcal{A} = (Q, q_s, \delta, \mathsf{Acc})$ over alphabet Σ. Observe that given any state $q \in Q_0$ and any word $\kappa \in \Sigma^* \cup \Sigma^\omega$, $\mathcal{A}$ has at most one run ρ on α where all states in ρ belong to Q_0. We now present a couple of useful definitions. A set $W \subseteq Q$ is said to be a *witness* set if W has at most one level 0 state, i.e., $|W \cap Q_0| \leq 1$. Observe that for any word $u \in \Sigma^*$, $\mathsf{post}(q_s, u)$ is a witness set, i.e., $|\mathsf{post}(q_s, u) \cap Q_0| \leq 1$. We will say a word $\kappa \in \Sigma^* \cup \Sigma^\omega$ (depending on whether $\mathcal{A}$ is an automaton on finite or infinite words) is *definitely accepted* from witness set W iff for every $q \in W$ with $q \in Q_i$ (for $0 \leq i \leq k$) there is an accepting run ρ on κ starting from q such that for every j, $\rho[j] \in Q_i$ and $\delta_{\kappa[j]}(\rho[j], \rho[j+1]) = 1$. In other words, κ is definitely accepted from witness set W if and only if κ is accepted from every state q in W by a run where you stay in the same level as q, and all transitions in the run are taken with probability 1. Observe that the set of all words definitely accepted from a witness set W is regular. Furthermore, its emptiness can be checked in **PSPACE**.

Proposition 3. *For any* HPA $\mathcal{A}$ *and witness set* W*, the language* $\mathsf{L}_W = \{\kappa | \kappa$ *is definitely accepted by* $\mathcal{A}$ *from* $W\}$ *is regular. The emptiness of* L_W *can be checked in* **PSPACE**.

That the emptiness of L_W can be checked in **PSPACE** follows from the observation that $\mathsf{L}_W = \cap_{q \in W} \mathsf{L}_{\{q\}}$ and $\mathsf{L}_\emptyset$ (as defined above) is the set of all strings.

Definition 3. *A witness set* W *is said to be* good *if the language* L_W *(defined in Proposition 3) is non-empty.*

Witness sets play an important role in the acceptance of strings. is This characterized by the following Proposition.

Proposition 4. *For a* HPA $\mathcal{A}$*, threshold* $x \in [0,1]$*, and word* κ*,* $\kappa \in \mathsf{L}_{>x}(\mathcal{A})$ *if and only if there is a non-empty witness set* W*,* $u \in \Sigma^*$ *and* $\kappa' \in \Sigma^* \cup \Sigma^\omega$ *such that* $\kappa = u\kappa'$*,* κ' *is definitely accepted by* $\mathcal{A}$ *from* W*, and* $\delta_u(q_0, W) > x$*.*

Proposition 4 immediately implies the following (proof in Appendix C).

Proposition 5. *For a* HPMA $\mathcal{A} = (Q, q_s, \delta, \mathsf{Acc})$ *let* $\mathcal{GW}$ *be the set of good non-empty witness sets of* $\mathcal{A}$*. For* $W \in \mathcal{GW}$*, let* $\mathcal{A}_W = (Q, q_s, \delta, W)$ *be the* PFA *that has* W *as the set of its final states. We have that* $\mathsf{value}(\mathcal{A}) = \max_{W \in \mathcal{GW}} \mathsf{value}(\mathcal{A}_W)$*.*

3.1 Regularity of HPAs with Isolated Cut-points

Probabilistic automata, though finite state, are known to recognize non-regular languages, whether we consider automata on finite or infinite words [18, 1, 7]. One important result due to Rabin [18] is that $\mathsf{L}_{>x}(\mathcal{A})$ is regular for any PFA $\mathcal{A}$ if x is isolated for $\mathcal{A}$. We extend this observation to any HPMA.

Theorem 1. *Let* $\mathcal{A}$ *be a* HPMA *and let* $x \in [0,1]$ *be such that* x *is isolated for* $\mathcal{A}$*. Then* $\mathsf{L}_{>x}(\mathcal{A})$ *is* ω*-regular.*

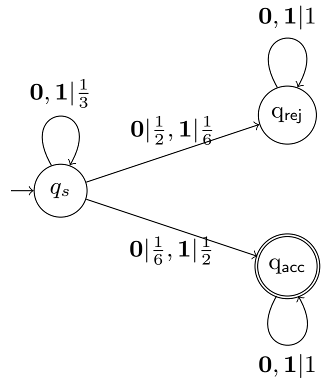

Fig. 1. $\mathcal{A}_{\mathsf{isolated}}$

Example 1. Now, we given an example HPMA with isolated cut points. The HPMA $\mathcal{A}_{\mathsf{isolated}}$ is given in Figure 1. It's set of states is $\{q_s, \mathrm{q}_{\mathsf{rej}}, \mathrm{q}_{\mathsf{acc}}\}$ where q_s is the initial state. It's acceptance condition is given by $\{\{\mathrm{q}_{\mathsf{acc}}\}\}$. Below, we show that, for each $n \geq 1$, the value $x_n = \frac{3}{4}(1-(\frac{1}{3})^n)$ is an isolated cut point with degree of isolation $\frac{1}{6}(\frac{1}{3})^n$. We also show that $L_n = \mathbf{1}^n\{\mathbf{0},\mathbf{1}\}^\omega$ is the set of infinite strings accepted with probability $> x_n$. For any finite sequence u, let the acceptance and rejection probabilities of u be the probabilities of reaching the states $\mathrm{q}_{\mathsf{acc}}, \mathrm{q}_{\mathsf{rej}}$, respectively, on input u staring from q_s. Observe that every sequence in L_n, is accepted by $\mathcal{A}$, with probability greater than the probability of acceptance of the finite string $u = 1^n0$, which is $\sum_{0\leq i<n}(\frac{1}{3})^i\frac{1}{2} + (\frac{1}{3})^n\frac{1}{6}$ and this value equals $\frac{3}{4}(1-(\frac{1}{3})^n) + (\frac{1}{3})^n\frac{1}{6}$. Now, consider any input sequence not in L_n, i.e., sequence in $\{\mathbf{0},\mathbf{1}\}^\omega \setminus L_n$. The probability of rejection of any such string is $>$ the probability of rejection of the finite string $1^{n-1}01$ which is $\sum_{0\leq i<n-1}(\frac{1}{3})^i\frac{1}{6} + (\frac{1}{3})^{n-1}\frac{1}{2} + (\frac{1}{3})^n\frac{1}{6}$. The later value, after some simplification, is seen to be $y = \frac{1}{4}(1-(\frac{1}{3})^{n-1}) + \frac{5}{3}(\frac{1}{3})^n$. From this, we see that the probability of acceptance of every infinite sequence not in L_n is less than $1-y$. After some simplification, it is seen that $1-y = \frac{3}{4}(1-(\frac{1}{3})^n) - \frac{1}{6}(\frac{1}{3})^n$. From these observations, we see that every infinite sequence in $\{\mathbf{0},\mathbf{1}\}^\omega \setminus L_n$ is accepted with probability $< x_n - \frac{1}{6}(\frac{1}{3})^n$, and every sequence in L_n is accepted with probability $> x_n + \frac{1}{6}(\frac{1}{3})^n$. Hence x_n is an isolated cut point with degree of isolation $\frac{1}{6}(\frac{1}{3})^n$.

4 Emptiness under isolation

We now show that the emptiness and universality problems are decidable for k-HPAs with isolated cut-points, even when the degree of isolation is not known. In order to establish the above result, we recall the definition of max-norms in matrices.

Definition 4. *For a $n \times n$ matrix δ, let δ_{ij} be the entry in i-th row and j-th column. We say that $\|\delta\| = \max_{i,j} |\delta_{ij}|$.*

For the rest of this section, we fix the input alphabet Σ. The decision procedure for checking emptiness and universality of k-HPAs depends on Lemma 1, which states that the "effect" of input word u on a k-HPFA $\mathcal{A}$ can be approximated by a short word v upto a given degree of approximation. The Lemma shows that for each ϵ, the matrix $\delta_u - \delta_v$ has max-norm less than equal to ϵ.

Lemma 1. *Given a k-HPA $\mathcal{A} = (Q, q_s, \delta, \mathsf{Acc})$ and a rational $0 < \epsilon < 1$, there is a computable $\ell_{\mathcal{A},\epsilon} \in \mathbb{N}$ such that for each word $u \in \Sigma^*$, there is a word $v \in \Sigma^*$ such that $|v| \leq \ell_{\mathcal{A},\epsilon}$ and $\|\delta_u - \delta_v\| < \epsilon$. Furthermore $\ell_{\mathcal{A},\epsilon} \leq (\log\lceil \frac{2^{(b+1)k} n^{2k}}{\epsilon} \rceil)^k 2^{(b+2)k} n^{n+k}$ where $n = |Q|$ and b is the maximum size of the transition probabilities.*

The proof of Lemma 1 is presented in Appendix D. We sketch its key ideas. The proof proceeds by induction on k.

- We first observe that if $\mathcal{A}$ is a 0-HPA, then all transition probabilities are either 0 or 1. Hence the stochastic matrix δ_u is such that each entry is either 0 or 1 and each row consists of exactly one non-zero entry. Since there are only n^n matrices, if $|u| > n^n$ then there will be $i < j$ such that matrices $\delta_{u[0:i]}$ and $\delta_{u[0:j]}$ are the same. So, we can remove the word $u[i+1:j]$ from u without affecting the probability of transitioning from one state to another.
- Suppose that we have established the Lemma for $k = k_0$. In the induction step, we have to prove it for $k = k_0 + 1$. Fix a level 0 state q of the PA $\mathcal{A}$. For each prefix w of u, it is the case that there is at most one level 0 state in $\mathsf{post}(q, w)$. Assume that there is exactly one level 0 state in $\mathsf{post}(q, w)$. For each $i < |u|$, we will say that there is a leak at position i if on the input $u[i]$, some probability moves to higher levels. Now, between two consecutive leaks, the automaton $\mathcal{A}$ is essentially a k_0-HPA obtained by moving all states down one level. Thus, we can use the Induction Hypothesis to shorten the words between leaks. After we reach a point when there are too "many" leaks, the probability of being in level 0 is small and can be ignored. This informal argument only shows that the qth row of δ_u can be approximated by a short word. Some bookkeeping is needed to ensure that the same short word works for every row.

Using Lemma 1, we can show that for a k-HPA, $\mathsf{value}(\mathcal{A})$ can be computed within a given degree of accuracy.

Theorem 2. *There is an algorithm, which given a k-HPA $\mathcal{A} = (Q, q_s, \delta, \mathsf{Acc})$, and a rational $\epsilon \in (0, 1)$ computes x such that $|\mathsf{value}(\mathcal{A}) - x| \leq \epsilon$. The algorithm is exponential in $\mathsf{poly}(\log(\frac{1}{\gamma}))$ and doubly exponential in the size of $\mathcal{A}$.*

Proof. The algorithm for the case when $\mathcal{A}$ is a HPFA works as follows. Given $\mathcal{A}$ and ϵ as given in the lemma, the algorithm computes $\ell_{\mathcal{A},\frac{\epsilon}{n}}$ where $n = |Q|$, enumerates all input sequence of length at most $\ell_{\mathcal{A},\frac{\epsilon}{n}}$, computes and outputs

Input: Integer k, a k-HPA $\mathcal{A} = (Q, q_s, \delta, \mathsf{Acc})$ and rational $x \in [0, 1]$
Output: YES if $\mathsf{L}_{>x}(\mathcal{A}) = \emptyset$ and NO if $\mathsf{L}_{>x}(\mathcal{A}) \neq \emptyset$

$n \leftarrow |Q|$
approx_value $\leftarrow 0$
$\epsilon \leftarrow \frac{1}{2}$
if $\mathcal{A}$ *is a* HPFA **then**
    $\mathcal{GW} \leftarrow \{\mathsf{Acc}\}$
else
    $\mathcal{GW} \leftarrow \{W \mid W \subseteq Q, W \neq \emptyset, W$ is a good witness$\}$
end
while true **do**
    Compute $\ell_{\mathcal{A}, \frac{\epsilon}{n}}$ as given in Lemma 1
    approx_value $\leftarrow \max_{W \in \mathcal{GW}, v \in \Sigma^*, |v| \leq \ell_{\mathcal{A}, \frac{\epsilon}{n}}} \delta_v(q_s, W)$
    if approx_value $> x$ **then**
        return NO
    else
        if approx_value $< x - \epsilon$ **then**
            return YES
        else
            $\epsilon \leftarrow \frac{\epsilon}{2}$
        end
    end
end

Fig. 2. Procedure for checking emptiness of HPAs

the maximum of the acceptance probabilities of these strings. If x is the value output by the algorithm, using Lemma 1, it is easy to see that $|\mathsf{value}(\mathcal{A}) - x| \leq \epsilon$. The time bounds follow from the bound on $\ell_{\mathcal{A}, \frac{\epsilon}{n}}$ in Lemma 1. For the case of HPMA, we appeal to Proposition 5 which allows us to approximate the value using HPFAs. □

The above algorithm to approximate the value of a HPA immediately gives us an algorithm that given a HPA $\mathcal{A}$ and a rational x such that x is an isolated cutpoint of $\mathcal{A}$ checks if the regular language $\mathsf{L}_{>x}(\mathcal{A})$ is empty or not, even if a degree of isolation is not known. The algorithm is obtained as follows. We progressively compute the value of $\mathcal{A}$ with increasing precision. Suppose at some point the value is approximated by v with precision ϵ. If $v - \epsilon > x$ then we know that $\mathcal{A}$ has a non-empty language. On the other hand, if $v + \epsilon < x$ then $\mathcal{A}$'s language is empty. Since x is isolated, we are guaranteed that eventually the precision ϵ is low enough to ensure that one of these two conditions hold. This is carried out in the following Theorem.

Theorem 3. *The algorithm in Figure 2 solves the following problem: Given a k-HPA $\mathcal{A} = (Q, q_s, \delta, \mathsf{Acc})$ and a rational $x \in (0, 1)$ such that x is an isolated cut-point for $\mathcal{A}$, decide if $\mathsf{L}_{>x}(\mathcal{A})$ is empty.*

Proof. Let the number of states of $\mathcal{A}$ be n. Let ϵ_m be the value of variable ϵ at the beginning of the mth iteration of the while loop. Clearly $\epsilon_m = \frac{1}{2^m}$. Consider first the case when $\mathcal{A}$ is a HPFA. The case when $\mathcal{A}$ is a HPMA follows a similar argument and is shown in Appendix E.

Clearly if the procedure outputs NO then $\mathsf{L}_{>x}(\mathcal{A}) \neq \emptyset$. Now suppose that the the algorithm outputs YES. Let ϵ_{m_0} be the value of ϵ when the algorithm outputs YES. As the program outputs YES, for each word w such that $|w| \leq \ell_{\mathcal{A}, \frac{\epsilon_{m_0}}{n}}$, we have that $\delta_w(q_s, Q_f) + \epsilon_{m_0} < x$. Fix a finite word u. Thanks to Lemma 1, there is finite word v such that $|v| \leq \ell_{\mathcal{A}, \frac{\epsilon_{m_0}}{n}}$ and $\delta_u(q_s, Q_f) < \delta_v(q_s, Q_f) + \epsilon_{m_0} < x$. Thus, if the algorithm outputs YES then $\mathsf{L}_{>x}(\mathcal{A}) = \emptyset$. Notice that if the algorithm terminates then it gives the correct answer even if x is not isolated.

We claim that the algorithm in Figure 2 terminates if $\mathsf{L}_{>x}(\mathcal{A}) \neq \emptyset$ or if $\mathsf{value}(\mathcal{A}) < x$. If $\mathsf{L}_{>x}(\mathcal{A}) \neq \emptyset$ then fix a word u such that $\delta_u(q_s, Q_f) > x$. Let $\epsilon' = \delta_u(q_s, Q_f) - x$. Let m_0 be the smallest integer such that $n\epsilon_{m_0} < \epsilon'$. Thanks to Lemma 1, there is a finite word v such that $|v| \leq \ell_{\mathcal{A}, \frac{\epsilon_{m_0}}{n}}$ and $\delta_v(q_s, Q_f) > \delta_u(q_s, Q_f) - n\epsilon_{m_0} = x + \epsilon' - n\epsilon_{m_0} > x$. Thus $\mathsf{approx\_value} > x$ in the m_0th unrolling of the while loop and the algorithm terminates.

If $\mathsf{value}(\mathcal{A}) < x$ then let $\epsilon' = x - \mathsf{value}(\mathcal{A})$. Let m_0 be the smallest integer such that $\epsilon_{m_0} < \epsilon'$. It is easy to see that the algorithm will terminate in the m_0th unrolling of the loop as for every word w, it is the case that $\delta_w(q_s, Q_f) + \epsilon_{m_0} \leq (x - \epsilon') + \epsilon_{m_0} < x$.

The Theorem follows from the fact that if x is an isolated cutpoint of $\mathcal{A}$ and $\mathsf{L}_{>x}(\mathcal{A}) = \emptyset$ then $\mathsf{value}(\mathcal{A}) < x$. □

Next, we show that if x is isolated for a PA $\mathcal{A}$ then we can decide if $\mathsf{L}_{>x}(\mathcal{A})$ is contained in/contains a given regular language R (where R is a regular language over finite or infinite words depending on whether $\mathcal{A}$ is a HPFA or a HPMA). Observe that this also implies that the problem of checking whether $\mathsf{L}_{>x}(\mathcal{A})$ is universal or not is also decidable if x is an isolated cutpoint of $\mathcal{A}$. (See Appendix F for a proof.)

Theorem 4. *Let* $\bowtie \in \{\subseteq, \supseteq, =\}$. *There is an algorithm that given a regular language* R*, a* k-HPA $\mathcal{A} = (Q, q_s, \delta, \mathsf{Acc})$*, a rational* $x \in (0,1)$ *such that* x *is an isolated cut-point for* $\mathcal{A}$*, decides if* $\mathsf{L}_{>x}(\mathcal{A}) \bowtie R$.

5 On the value decision problem

For a PFA $\mathcal{A}$, the problem of checking if $x \in [0,1]$ is an isolated cut-point is $\mathbf{\Sigma_2^0}$-complete [12]. Observe that 1 is an isolated cutpoint of a PFA $\mathcal{A}$ iff $\mathsf{value}(\mathcal{A}) < 1$. An immediate consequence is that the value decision problem for PFAs is $\mathbf{\Pi_2^0}$-complete. For HPFAs, the problem of checking if 1 is isolated is known to be **PSPACE**-complete [14]. The same result holds for checking if 0 is an isolated cutpoint for a HPA. We now show that the problem checking whether $x \in (0,1)$ is an isolated cutpoint for a HPA is **R.E.**-complete and the value problem is **co-R.E.**-complete. Hence, the isolated cut point decision problem and the value decision problem are simpler for HPAs. We start by proving that the value problem is **co-R.E.**-complete. The proof of containment in **co-R.E.** relies on Lemma 1, which allows us to approximate the effect of each finite word on an automaton by a short word. The hardness result is obtained by a modification of the proof of undecidability of emptiness problem for the 2-HPAs [2, 4, 3].

Theorem 5. *For each $k \geq 2$, the value decision problem for k-HPAs is* **co-R.E.**-*complete.*

Proof. We first establish that the value problem is in **co-R.E.** Consider first the case for HPFAs. Let $\mathcal{A} = (Q, q_s, \delta, \mathsf{Acc})$ be a HPFA. Let $value(\mathcal{A}, x)$ be the predicate

$$value(\mathcal{A}, x) = (\forall v \in \Sigma^*.\ \mathsf{P}_{\mathcal{A}}(v) \leq x) \wedge \forall m \in \mathbb{N}.\ (\exists u \in \Sigma^*.\ \mathsf{P}_{\mathcal{A}}(u) > x - \frac{1}{m}).$$

It is easy to see that $\mathsf{value}(\mathcal{A}) = x$ iff $value(\mathcal{A}, x)$ is true.

Let $|Q| = n$ and let $fvalue(\mathcal{A}, x)$ be the predicate

$$fvalue(\mathcal{A}, x) = (\forall v \in \Sigma^*.\ \mathsf{P}_{\mathcal{A}}(v) \leq x) \wedge \\ \forall m \in \mathbb{N}.\ (\exists v \in \Sigma^*.\ |v| \leq \ell_{\mathcal{A}, \frac{1}{mn}} \wedge \mathsf{P}_{\mathcal{A}}(v) > x - \tfrac{1}{2m}).$$

It is easy to see that if $fvalue(\mathcal{A}, x)$ is true then so is $value(\mathcal{A}, x)$.

Assume now that $value(\mathcal{A}, x)$ is true. Then for each $m \in \mathbb{N}$, there is a $u \in \Sigma^*$ such that $\mathsf{P}_{\mathcal{A}}(u) > x - \frac{1}{m}$. Fix m, u. Thanks to Lemma 1 there is a v such that $|v| \leq \ell_{\mathcal{A}, \frac{1}{mn}}$ and

$$\delta_u(q_s, q) - \frac{1}{mn} < \delta_v(q_s, q) < \delta_u(q_s, q) + \frac{1}{mn} \text{ for each } q \in Q. \tag{1}$$

Fix v. Therefore we get from Equation 1 above that

$$\delta_v(q_s, Q_f) > \sum_{q \in Q_f} (\delta_u(q_s, q) - \frac{1}{mn}) = \delta_u(q_s, Q_f) - \frac{|Q_f|}{mn} > x - \frac{1}{m} - \frac{1}{m}.$$

It follows that $fvalue(\mathcal{A}, x)$ is also true if $value(\mathcal{A}, x)$ is true. Hence, $\mathsf{value}(\mathcal{A}) = x$ iff $fvalue(\mathcal{A}, x)$ is true. Note that the problem of checking that for given v, if $\mathsf{P}_{\mathcal{A}}(v) \leq x$ is decidable. Also the problem of checking that given $m \in \mathbb{N}$, $(\exists v \in \Sigma^*.\ |v| \leq \ell_{\mathcal{A}, \frac{1}{mn}} \wedge \mathsf{P}_{\mathcal{A}}(v) > x - \frac{1}{2m})$ is decidable since $\ell_{\mathcal{A}, \frac{1}{mn}}$ is computable. Thus, the value problem is in **co-R.E.**

Now consider the theorem for HPMAs. Let $\mathcal{A} = (Q, q_s, \delta, \mathsf{Acc})$ be a HPMA. Let $\mathcal{GW}$ be the set of good non-empty witness sets of $\mathcal{A}$. For $W \in \mathcal{GW}$, let $\mathcal{A}_W = (Q, q_s, \delta, W)$ be the PFA that has W as the set of its final states. Thanks to Proposition 5, we have that $\mathsf{value}(\mathcal{A}) = \mathsf{max}_{W \in \mathcal{GW}}\ \mathsf{value}(\mathcal{A}_W)$. This implies that $\mathsf{value}(\mathcal{A}) = x$ iff one of the predicates $\{value(\mathcal{A}_W, x) \mid W \in \mathcal{GW}\}$ is true. The upper bound follows in this case.

The lower bound is proved in Appendix G. □

Using Proposition 2, we can convert the non-isolation decision problem to the value decision problem. Thus the problem of checking whether a cut-point x is isolated for a HPA $\mathcal{A}$ is in **R.E.**. We can show that the non-isolation decision problem is **co-R.E.**-hard also using the same reduction that is used to prove **co-R.E.**-hardness of the value problem. This yields the following Theorem (The proof is in Appendix H).

Theorem 6. *For each $k \geq 2$, the isolation decision problem for k-*HPA*s is* **R.E.**-*complete.*

5.1 Computing the Least Upperbound for 1-HPA

In this section, we give an **EXPTIME** algorithm for computing the value of a 1-HPA. The key technical observation to make this possible is a necessary and sufficient condition for when x is the value of a 1-HPFA. The observation is that there is always an exponentially bound "ultimately periodic" witness for the value being x; this is the content of the next Lemma.

Lemma 2. *Let $\mathcal{A} = (Q, q_s, \delta, Q_f)$ be an* 1-HPFA *over an alphabet Σ, and $n = |Q|$. Then, for any x, $x = \mathsf{value}(\mathcal{A})$ iff there is no string that is accepted by $\mathcal{A}$ with probability $> x$ and at least one of the following conditions is satisfied.*

1. *$\exists u \in \Sigma^*$ such that $|u| \leq 2^n$ and $\mathsf{P}_{\mathcal{A}}(u) = x$.*
2. *$\exists u, v \in \Sigma^*$ such that $|u|, |v| \leq 2^n$, there exists a good witness set $W \subseteq Q_1$ such that $W \subseteq \mathsf{post}(q_s, u)$, $\mathsf{post}(W, v) \subseteq W$, $\mathsf{post}(q_s, u) \cap Q_0 = \mathsf{post}(q_s, uv) \cap Q_0$, $\forall i \geq 0$, $\delta_{uv^{i+1}}(q_s, W) > \delta_{uv^i}(q_s, W)$ and $\lim_{i\to\infty} \delta_{uv^i}(q_s, W) = x$.*

Condition 1 of the lemma corresponds to the case when there is an input string that is accepted with the maximum possible probability $\mathsf{value}(\mathcal{A})$. If there is no input string that is accepted with probability $\mathsf{value}(\mathcal{A})$, then Condition 2 of the lemma asserts that there are finite sequences u, v and a good witness set W, such that $\delta_{uv^i}(q_s, W)$ increases monotonically with increasing values of i, and the limit of this monotonic sequence equals $\mathsf{value}(\mathcal{A})$. The proof is in Appendix I.

The next observation bounds the size of the probability of reaching a set of states C on an input u, as a function of $|u|$.

Lemma 3. *Let $\mathcal{A} = (Q, q_s, \delta, \mathsf{Acc})$ be a* 1-HPA *over an alphabet Σ and $n = |Q|$. Then, for any $u \in \Sigma^+$, $q \in Q_0$ and $C \subseteq Q$, the size of $\delta_u(q, C)$ is $\leq |u|nr$ where r is the maximum of the sizes of the transition probabilities of $\mathcal{A}$.*

Proof. The lemma is proved by a simple induction on $|u|$. In the base case, $|u| = 1$, the observation follows from the fact that $\delta_u(q, C)$ is the sum of atmost n transition probabilities of $\mathcal{A}$. For the inductive step, assume that the observation is true for all strings of length $\leq k$. Let $u = av$ be a string of length $k+1$ where $a \in \Sigma$ and $v \in \Sigma^k$. Clearly, $\delta_u(q, C) = \sum_{q' \in Q_1} \delta_a(q, q')\delta_v(q', C) + \delta_a(q, q_1)\delta_v(q_1, C)$ where $q_1 \in Q_0$ be such that $\delta_a(q, q_1) > 0$. Observe that, for $q' \in Q_1$, $\delta_v(q', C)$ is either 0 or 1. Now, it is easy to see that size of $\delta_u(q, C)$ is $\leq$ the sum of the sizes of $\delta_a(q,q')$ for less than n distinct $q' \in Q_1$, the sizes of $\delta_a(q,q_1)$ and $\delta_v(q_1,C)$. Using the induction hypothesis for v and observing that the sizes of $\delta_a(q,q')$, $\delta_a(q,q_1)$ are both $\leq r$, we get the desired result. □

The last technical lemma we need bounds the size of the value of a 1-HPFA using Lemmas 2 and 3.

Lemma 4. *Let* $\mathcal{A} = (Q, q_s, \delta, Q_f)$ *be a* 1-HPFA *over an alphabet* Σ *and* $n = |Q|$. *Then, the size of* $\mathsf{value}(\mathcal{A})$ *is* $\leq 4rn2^n$ *where* r *is the maximum of the sizes of the transition probabilities of* $\mathcal{A}$.

Proof. Let $x = \mathsf{value}(\mathcal{A})$. Clearly no string is accepted by $\mathcal{A}$ with probability $> x$. Further, from Lemma 2, we see that either of the conditions (1), (2) , stated there, is satisfied. Suppose condition (1) is satisfied. Then, there is a string $u \in \Sigma^*$, such that $|u| \leq 2^n$ and $x = \mathsf{P}_{\mathcal{A}}(u)$. Now, our result follows from Lemma 3.

Now, suppose condition (2) of Lemma 2 is satisfied; let u, v, W be as specified in that condition. Let $\mathsf{post}(q_s, u) \cap Q_0 = \mathsf{post}(q_s, uv) \cap Q_0 = \{q_1\}$. It is easy to see that

$$\begin{aligned}\lim_{i\to\infty} \delta_{uv^i}(q_s, W) &= \delta_u(q_s, W) + \delta_u(q_s, q_1)\delta_v(q_1, W) \textstyle\sum_{i=0}^{\infty}(\delta_v(q_1, q_1))^i \\ &= \delta_u(q_s, W) + \delta_u(q_s, q_1)\frac{\delta_v(q_1,W)}{1-\delta_v(q_1,q_1)}\end{aligned}$$

Since, $|u|, |v| \leq 2^n$, we see from Lemma 3 that the sizes of $\delta_u(q_s, W)$, $\delta_v(q_1, W)$, $\delta_u(q_s, q_1)$ and $\delta_v(q_1, q_1)$ are all $\leq rn2^n$. From this and the above equation, it is easy to see that the size of $\lim_{i\to\infty} \delta_{uv^i}(q_s, W)$ is atmost the sum of the sizes of $\delta_u(q_s, W)$, $\delta_v(q_1, W)$, $\delta_u(q_s, q_1)$, and $\delta_v(q_1, q_1)$. From this we observe that the size of $\lim_{i\to\infty} \delta_{uv^i}(q_s, W)$, and hence the size of x is $\leq 4rn2^n$. □

We are now ready to present the main result of this section — an exponential time algorithm to compute the value of a 1-HPA.

Theorem 7. *The value of a* 1-HPA $\mathcal{A}$ *can be computed in exponential time. The value decision problem is in* **EXPTIME** *and is* **PSPACE**-*hard.*

Proof. First we consider the case for HPFAs. Let $\mathcal{A} = (Q, q_s, \delta, \mathsf{Acc})$ be the given HPFA over an alphabet Σ, and $n = |Q|$. There is a näive double exponential time algorithm that computes $\mathsf{value}(\mathcal{A})$ using Lemma 2. Such an algorithm enumerates all triples (u, v, W) such that $|u|, |v| \leq 2^n$, $W \subseteq Q_1$ and all the properties stated in condition (2) of Lemma 2 are satisfied. It computes $\mathsf{value}(\mathcal{A})$ to be the maximum of $\lim_{i\to\infty} \delta_{uv^i}(q_s, W)$ over all such triples (u, v, W). It is easy to see that such an algorithm is of time complexity double exponential in n.

Now, we give an algorithm, that computes $\mathsf{value}(\mathcal{A})$, of time complexity only single exponential in n. Let $N = 4rn2^n$ and $M = 2^N$ where r is the maximum of the sizes of the transitional probabilities of $\mathcal{A}$. From Lemma 4, we see that the size of $\mathsf{value}(\mathcal{A})$ is $\leq N$. Let $\mathsf{value}(\mathcal{A}) = \frac{y}{z}$. The above observation implies that $y, z \leq M$. Now, we employ an approach based on binary search on rationals [19] to compute the exact value of $\mathsf{value}(\mathcal{A})$. Essentially, this approach divides the unit interval $[0, 1]$ into $2M^2$ sub-intervals of equal length, i.e., each of length $\frac{1}{2M^2}$. Then, using binary search that employs queries of the form "$\mathsf{L}_{>x}(\mathcal{A}) = \emptyset$?", where $x = \frac{k}{2M^2}$ for some $k \leq 2M^2$, this approach determines the unique integer $\ell \leq 2M^2$ such that $\mathsf{value}(\mathcal{A})$ is in the interval $[\frac{\ell}{2M^2}, \frac{\ell+1}{2M^2})$. (Every such interval has atmost one rational number of the form $\frac{y_1}{z_1}$ where $y_1, z_1 \leq M$).

Once such an interval is identified, the exact value of $\mathsf{value}(\mathcal{A})$ is computed using a simple algorithm, given in [19], of complexity $O(\log M)$, i.e., of complexity $O(N)$. Each query of the form "$\mathsf{L}_{>x}(\mathcal{A}) = \emptyset$?" can be answered using the algorithm for the emptiness problem for 1-HPA as given in [10, 3]; the algorithm given in [10] is of complexity linear in the size of x and exponential in n. Since the size of x used in the above algorithm is $\leq N$, the time complexity of a single invocation of this algorithm during the binary search is seen to be $O(r8^n)$. Further more, there are at most N such invocations and hence the over all complexity of performing the binary search is $O(r^2 16^n)$. Furthermore, the complexity of the second step of the algorithm, i.e., the step in which the actual values of $\mathsf{value}(\mathcal{A})$ is computed, is also of time complexity $O(N)$. Hence the overall time complexity of the above algorithm for computing $\mathsf{value}(\mathcal{A})$ is $O(r^2 16^n)$.

For HPMAs, we use Proposition 5. Let $\mathcal{A}$ be a HPMA. $\mathcal{GW}$ be the set of good non-empty witness sets of $\mathcal{A}$. Using this proposition, we compute $\mathsf{value}(\mathcal{A})$ to be $\mathsf{max}_{W \in \mathcal{GW}}\ \mathsf{value}(\mathcal{A}_W)$, where $\mathcal{A}_W = (Q, q_s, \delta, W)$. Since $\mathsf{value}(\mathcal{A}_W)$ can be computed in time $O(r^2 16^n)$ and $|\mathcal{GW}| \leq 2^n$, we see that the time complexity of computing $\mathsf{value}(\mathcal{A})$ is $O(r^2 32^n)$.

Thus, it is easy to see that the value decision problem is in **EXPTIME**. It can be shown to be **PSPACE**-hard using the same techniques used to prove that the emptiness problem for 1-HPAs is **PSPACE**-hard in [10, 3] □

6 Conclusions

In this paper, we presented a number of results on HPAs. First, we showed that for a k-HPA, the effect of any string (i.e., the transition probability matrix of the string) can be approximated by that of a short string of bounded length, for a given precision. This can be used to approximate the value of a k-HPA with arbitrary precision, and decide the emptiness of the language of a k-HPA with an isolated cut-point. These observations allowed us to prove that the problem of computing the value of a k-HPA (for $k \geq 2$) is **co-R.E.**-complete. For a 1-HPA, we showed that it's value can be computed exactly in exponential time. A couple of problems for 1-HPAs remain open — the decidability of the isolation problem and the exact complexity of the value problem which has been shown to be in **EXPTIME**.

Acknowledgements. We thanks the anonymous reviewers for their useful comments and suggestions. Rohit Chadha was partially supported by NSF CNS 1314338 and CNS 1553548. Mahesh Viswanathan was partially supported by NSF CNS 1314485. A. Prasad Sistla was partially supported by CCF 1319754 and CNS 1314485.

References

1. C. Baier and M. Größer. Recognizing ω-regular languages with probabilistic automata. In *20th IEEE Symp. on Logic in Computer Science*, pages 137–146, 2005.
2. Y. Ben. *Model Checking Open Probabilistic Systems using Hierarchical probabilistic automata*. PhD thesis, University of Illinois, Chicago, 2016.
3. Y. Ben, R. Chadha, A. P. Sistla, and M. Viswanathan. Decidable and expressive classes of probabilistic automata. https://drive.google.com/open?id=0Bygmq8nf_becNUFESjZzNEJGVlE, 2016. Manuscript under review.
4. Y. Ben and A. P. Sistla. Model checking failure-prone open systems using probabilistic automata. In *13th International Symposium on Automated Technology for Verification and Analysis*, volume 9364 of *Lecture Notes in Computer Science*, pages 148–165. Springer, 2015.
5. A. Bertoni. The solution of problems relative to probabilistic automata in the frame of the formal languages theory. In *GI Jahrestagung*, pages 107–112, 1974.
6. R. Chadha, D. Kini, and M. Viswanathan. Decidable problems for unary PFAs. In *Proceedings of the International Conference on Quantitative Evaluation of Systems*, pages 329–344, 2014.
7. R. Chadha, A. P. Sistla, and M. Viswanathan. On the expressiveness and complexity of randomization in finite state monitors. *Journal of the ACM*, 56(5), 2009.
8. R. Chadha, A. P. Sistla, and M. Viswanathan. Probabilistic Büchi automata with non-extremal acceptance thresholds. In *Intl. Conference on Verification, Model checking and Abstract Interpretation*, pages 103–117, 2010.
9. R. Chadha, A. P. Sistla, and M. Viswanathan. Power of randomization in automata on infinite strings. *Logical Methods in Computer Science*, 7(3):1–22, 2011.
10. R. Chadha, A. P. Sistla, M. Viswanathan, and Y. Ben. Decidable and expressive classes of probabilistic automata. In *16th International Conference on Foundations of Software Science and Computation Structures*, volume 9034 of *Lecture Notes in Computer Science*, pages 200–214. Springer, 2015.
11. R. Chadha, A.P. Sistla, and M. Viswanathan. Power of randomization in automata on infinite strings. In *Proceedings of the International Conference on Concurrency Theory*, pages 229–243, 2009.
12. R. Chadha, A.P. Sistla, and M. Viswanathan. Probabilistic automata with isolated cut-points. In *Proceedings of the International Symposium on Mathematical Foundation of Computer Science*, pages 254–265, 2013.
13. A. Condon and R. J. Lipton. On the complexity of space bounded interactive proofs (extended abstract). In *Symp. on Foundations of Computer Science*, pages 462–467, 1989.
14. N. Fijalkow, H. Gimbert, and Y. Oualhadj. Deciding the value 1 problem for probabilistic leaktight automata. In *IEEE Symp. on Logic in Computer Science*, pages 295–304, 2012.
15. H. Gimbert and Y. Oualhadj. Probabilistic automata on finite words: Decidable and undecidable problems. In *Intl. Colloquium on Automata, Languages and Programming*, pages 527–538, 2010.
16. J. Kemeny and J. Snell. *Denumerable Markov Chains.* Springer-Verlag, 1976.
17. A. Paz. *Introduction to Probabilistic Automata.* Academic Press, 1971.
18. M. O. Rabin. Probabilistic automata. *Inf. and Control*, 6(3):230–245, 1963.
19. S.Kwek and K. Mehlhorn. Optimal search for rationals. *Information Processing Letters*, 86(1):23–26, 2003.
20. M. Vardi. Automatic verification of probabilistic concurrent finite-state programs. In *Symp. on Foundations of Computer Science*, pages 327–338, 1985.

A Proof of Proposition 2

Thanks to Proposition 1 we can choose x to be $\frac{1}{2}$ as the construction therein preserves isolation and non-isolation.

First consider the case when $\mathcal{A}$ is a PFA. Let $\mathcal{A} = (Q, q_s, \delta, Q_f)$. Consider the PFA, $\mathcal{A}^c = (Q, q_s, \delta, Q \setminus Q_f)$ obtained by interchanging the final and non-final states of Q_f. It is easy to see that for any word u, $\mathsf{P}_{\mathcal{A}^c}(u) = 1 - \mathsf{P}_{\mathcal{A}}(u)$. Let $\mathcal{B} = (Q', q'_s, \delta', Q'_f)$ be the cross product of $\mathcal{A}$ and $\mathcal{A}^c$. That is, $Q' = Q \times Q$, $q'_s = (q_s, q_s)$, $Q'_f = Q_f \times Q \setminus Q_f$ and $\delta'((q_a, q_b), (q_c, q_d)) = \delta(q_a, q_c)\delta(q_b, q_d)$ for each $q_a, q_b, q_d, q_c \in Q$. It can be shown that for each finite word u,

$$\begin{aligned}\mathsf{P}_{\mathcal{B}}(u) &= \mathsf{P}_{\mathcal{A}}(u)\mathsf{P}_{\mathcal{A}^c}(u)\\ &= \mathsf{P}_{\mathcal{A}}(u)(1 - \mathsf{P}_{\mathcal{A}}(u))\\ &= \mathsf{P}_{\mathcal{A}}(u) - (\mathsf{P}_{\mathcal{A}}(u))^2\\ &= \tfrac{1}{4} - (\tfrac{1}{4} - \mathsf{P}_{\mathcal{A}}(u) + (\mathsf{P}_{\mathcal{A}}(u))^2)\\ &= \tfrac{1}{4} - (\tfrac{1}{2} - \mathsf{P}_{\mathcal{A}}(u))^2.\end{aligned}$$

Now, it is easy to see that $\mathsf{value}(\mathcal{B}) < \frac{1}{4}$ if $\frac{1}{2}$ is an isolated cutpoint of $\mathcal{A}$ and $\mathsf{value}(\mathcal{B}) = \frac{1}{4}$ if $\frac{1}{2}$ is not isolated for $\mathcal{A}$. $\mathcal{B}$ is the required automaton.

Now consider the case when $\mathcal{A}$ is a PMA. Let $\mathcal{A} = (Q, q_s, \delta, \mathsf{Acc})$ where $\mathsf{Acc} \subseteq 2^Q$. Consider the PMA, $\mathcal{A}^c = (Q, q_s, \delta, 2^Q \setminus \mathsf{Acc})$. It is easy to see that for any infinite word α, $\mathsf{P}_{\mathcal{A}^c}(\alpha) = 1 - \mathsf{P}_{\mathcal{A}}(\alpha)$. Let $\mathcal{B} = (Q', q'_s, \delta', \mathsf{Acc}')$ be the cross product of $\mathcal{A}$ and $\mathcal{A}^c$. That is, $Q' = Q \times Q$, $q'_s = (q_s, q_s)$, and $\delta'((q_a, q_b), (q_c, q_d)) = \delta(q_a, q_c)\delta(q_b, q_d)$ for each $q_a, q_b, q_d, q_c \in Q$. A set $A \in \mathsf{Acc}'$ iff the set $\mathsf{proj}_1(A) = \{q_1 \in Q \mid \exists q_2 \in Q.\ (q_1, q_2) \in A\}$ is in the set Acc and the set $\mathsf{proj}_2(A) = \{q_2 \in Q \mid \exists q_1 \in Q.\ (q_1, q_2) \in A\}$ is in the set $2^Q \setminus \mathsf{Acc}$. We can show that $\mathcal{B}$ is the required PMA along the same lines as in the case for PFAs.

B Proof of Proposition 4

We first observe that the existence of a witness set satisfying the properties of the proposition implies that $\kappa \in \mathsf{L}_{>x}(\mathcal{A})$.

($\Rightarrow$) For the converse, first observe that in the case of HPFAs, we can take $u = \kappa$, $\kappa' = \epsilon$, and $W = Q_f$. Thus, we are left with only establishing this result for the case of HPMA. We prove this by induction on the number of levels k. Observe that, without loss of generality, we can assume that $q_s \in Q_0$.

Base Case $k = 0$. $\mathcal{A}$ in this case is a deterministic automaton (no probabilitistic transitions) and hence the result follows trivially — take $u = \epsilon$, $\kappa' = \kappa$ and $W = \{q_s\}$.

Induction step. Let the claim be true for all $k \leq m$. Let $\mathcal{A} = (Q, q_s, \delta, \mathsf{Acc})$ be a $(m+1)$-HPMA and let $\mathsf{P}_{\mathcal{A}}(\kappa) = x + \epsilon$ where $\epsilon > 0$. Observe that $\delta_{\kappa[0:r]}(q_s, Q_0)$ is a non-increasing function. Let A_0 be the set of all accepting runs where all states are in Q_0; observe that A_0 is measurable. We will consider two cases based on $\mu_{\mathcal{A},\kappa}(\mathsf{A}_0)$.

Case a. Consider the case when $\mu_{\mathcal{A},\kappa}(\mathsf{A}_0) < \frac{\epsilon}{4}$. Let ρ be the unique sequence of Q_0 states $\mathcal{A}$ visits on κ, i.e., $\rho[r] = \mathsf{post}(q_s, \kappa[0 : r] \cap Q_0$ if $\mathsf{post}(q_s, \kappa[0 : r] \cap Q_0 \neq \emptyset$. Now $\mu_{\mathcal{A},\kappa}(\mathsf{A}_0) < \frac{\epsilon}{4}$ implies that there is r_0 such that either (i) $\delta_{\kappa[0:r_0]}(q_s, Q_0) < \frac{\epsilon}{4}$, or (ii) $\{\rho[r] \mid r \geq r_0\} = \mathsf{inf}(\rho) \notin \mathsf{Acc}$. Let $S' = \mathsf{post}(q_s, \kappa[0 : r_0] \setminus Q_0$, $v = \kappa[0 : r_0]$, and $\kappa_1 = \kappa[r_0 + 1 : \infty]$. For each $q \in W'$, if q is a level k_q state then let $\mathcal{A}_q$ be the HPMA obtained from $\mathcal{A}$ by taking the initial state to be q and restricting the set of states to be the states whose level is $\geq k_q$. Let $x_q = \mathsf{P}_{\mathcal{A}_q}(\kappa_1)$ and let $S \subseteq S'$ be the set of states $\{q \mid x_q > \frac{\epsilon}{4}\}$.

We make the following observations.

- $\mathsf{P}_{\mathcal{A}}(\kappa) = x + \epsilon \leq \left(\sum_{q \in S'} \delta_v(q_s, q) x_q\right) + \frac{\epsilon}{4}$. Since
$$\sum_{q \in S' \setminus S} \delta_v(q_s, q) x_q < \frac{\epsilon}{4},$$
we get that
$$\sum_{q \in S} \delta_v(q_s, q) x_q > x + \frac{\epsilon}{2}.$$
Therefore,
$$\sum_{q \in S} \delta_v(q_s, q)(x_q - \frac{\epsilon}{4}) > x + \frac{\epsilon}{4}.$$
- Note for each $q \in S$, $\mathcal{A}_q$ can be viewed as a HPMA with rank $\leq m$ and $\mathcal{A}_q$ accepts κ_1 with probability $x_q > x_q - \frac{\epsilon}{4}$. By induction hypothesis, for each $q \in S$, fix a witness set W_q of $\mathcal{A}_q$, index r_q with $u_q = \kappa_1[0 : r_q]$ and $kappa_q = \kappa_1[r_q + 1 : \infty]$, such that $\kappa_1 = u_q \kappa_q$, κ_q is definitely accepted from W_q and $\delta_u(q, W_q) > x_q - \frac{\epsilon}{4}$. Let $r = \max\{r_q \mid q \in S\} + 1$. Take $u = \kappa[0 : r_0 + r]$, $\kappa' = \kappa[r_0 + r + 1 : \infty]$, and $W = \cup_{q \in S} W_q$. Observe that κ' is definitely accepted from W, and from the first observation, $\delta_u(q_s, W) > x$.

Case b. Consider the case when $\mu_{\mathcal{A},\kappa}(\mathsf{A}_0) \geq \frac{\epsilon}{4}$. Since $\mathcal{A}$ is hierarchical, this means that there is r_0 such that for all $r \geq r_0$, $\delta_{\kappa[0:r]}(q_s, Q_0) = \delta_{\kappa[0:r_0]}(q_s, Q_1)$. In other words, $\kappa[r_0 + 1 : \infty]$ is definitely accepted from $\mathsf{post}(q_s, \kappa[0 : r_0]) \cap Q_0$. Take $S' = \mathsf{post}(q_s, \kappa[0 : r_0]) \setminus Q_0$, and repeat the argument from Case a to find r and W. One can argue that taking $u = \kappa[0 : r_0 + r]$, $\kappa' = \kappa[r_0 + r + 1 : \infty]$, and the witness set to be $W \cup (\mathsf{post}(q_s, u) \cap Q_0)$ satisfies the conditions of the Proposition. ☐

C Proof of Proposition 5

Let $x = \mathsf{value}(\mathcal{A})$ and let $y = \mathsf{max}_{W \in \mathcal{GW}}\ \mathsf{value}(\mathcal{A}_W)$. For each $\epsilon > 0$, we have that there is an infinite word α such that $\mathsf{P}_{\mathcal{A}}(\alpha) > x - \epsilon$. Fix α. Thanks to Proposition 4, there is a non-empty witness W_α and a finite prefix u_α of α such that $\mathsf{P}_{\mathcal{A}_{W_\alpha}}(u_\alpha) > x - \epsilon$. Thus $x - \epsilon < y$. Since ϵ is arbitrary, we get that $x \leq y$.

Let $W_{\mathsf{max}} \in \mathcal{GW}$ be the set such that $\mathsf{value}(\mathcal{A}_{W_{\mathsf{max}}}) = y$. For each $\epsilon > 0$ there is a word u_ϵ such that $\mathsf{P}_{\mathcal{A}_{W_{\mathsf{max}}}}(u_\epsilon) > y - \epsilon$. Thanks to the definition of a good witness set, there is an infinite word α_ϵ that is definitely accepted from W_{max}. It is easy to see that the word $u_\epsilon, \alpha_\epsilon$ is accepted by $\mathcal{A}$ with at least probability $\mathsf{P}_{\mathcal{A}_{W_{\mathsf{max}}}}(u_\epsilon) > y - \epsilon$. Thus $x > y - \epsilon$. Since ϵ is arbitrary, we get that $x \geq y$ also.

D Proof of Lemma 1

We shall need the following proposition.

Proposition 6. *Let $\delta_1, \delta_2, \dots, \delta_p$ and $\delta'_1, \delta'_2, \dots, \delta'_p$ be $n \times n$ stochastic matrices and let $\epsilon_i = \|\delta_i - \delta'_i\|$ for each $1 \leq i \leq p$. Then $\left\|\delta_1 \dots \delta_p - \delta'_1 \dots \delta'_p\right\| \leq n \sum_{i=1}^{p-1} \epsilon_i + \epsilon_p$.*

Proof. We make the following observations:

(a) If δ is a stochastic $n \times n$ matrix and η is a $n \times n$ matrix then $\|\delta\eta\| \leq \|\eta\|$. This is because for row $1 \leq i \leq n$ and column $1 \leq j \leq n$,

$$|\delta\eta(i,j)| \leq \textstyle\sum_{k=1}^{n} \delta(i,k)|\eta(k,j)| \leq \|\eta\| \sum_{k=1}^{n} \delta(i,k) \leq \|\eta\|.$$

(b) If δ is a $n \times n$ matrix and η is a stochastic $n \times n$ matrix then $\|\delta\eta\| \leq n\|\delta\|$. This is because for row $1 \leq i \leq n$ and column $1 \leq j \leq n$,

$$|\delta\eta(i,j)| = |\textstyle\sum_{k=1}^{n} \delta(i,k)\eta(k,j)| \leq \|\delta\| \sum_{k=1}^{n} |\eta(k,j)| \leq \|\delta\| n.$$

Now going back to our Proposition, we claim by induction that for each $0 \leq j \leq p-1$,

$$\left\|\delta_{p-j} \dots \delta_p - \delta'_{p-j} \dots \delta'_p\right\| \leq n \sum_{i=p-j}^{p-1} \epsilon_i + \epsilon_p. \quad (2)$$

Clearly Equation 2 holds for $j = 0$. Assume that Equation 2 holds for $j = j_0$. Thanks to Observation (a) above, we get

$$\left\|\delta'_{p-j_0-1}(\delta_{p-j_0} \dots \delta_p - \delta'_{p-j_0} \dots \delta'_p)\right\| \leq n \sum_{i=p-j_0}^{p-1} \epsilon_i + \epsilon_p. \quad (3)$$

Also, by Observation (b) above, we get

$$\left\|(\delta_{p-j_0-1} - \delta'_{p-j_0-1})\delta_{p-j_0} \dots \delta_p\right\| \leq n\epsilon_{p-j_0-1}. \quad (4)$$

Therefore, using triangle inequality and Equations 3 and 4 above, we have that

$$\begin{aligned}\left\|\delta_{p-j_0-1} \dots \delta_p - \delta'_{p-j_0-1} \dots \delta'_p\right\| &\leq \left\|\delta_{p-j_0-1}\delta_{p-j_0} \dots \delta_p - \delta'_{p-j_0-1}\delta_{p-j_0} \dots \delta_p\right\| + \\ &\quad \left\|\delta'_{p-j_0-1}\delta_{p-j_0} \dots \delta_p - \delta'_{p-j_0-1}\delta'_{p-j_0} \dots \delta'_p\right\| \\ &\leq n\epsilon_{p-j_0-1} + n \textstyle\sum_{i=p-j}^{p-1} \epsilon_i + \epsilon_p.\end{aligned}$$

The proposition follows. ∎

We prove the existence of $\ell_{\mathcal{A},\epsilon}$. The proof of the upper bound on $\ell_{\mathcal{A},\epsilon}$ follows. The proof is by induction on k.

Base Case: $k = 0$. Let $|Q| = n$ and $\ell_{\mathcal{A},\epsilon} = n^n$. It is easy to see that for each word $w \in \Sigma^*$, each row of the matrix δ_w contains exactly one non-zero entry and this entry is 1. The base case follows immediately from the following claim.

Claim. For each word $u \in \Sigma^*$, there is a word v such that $|v| \leq n^n$ and $\delta_u(q, q') = \delta_v(q, q')$ for each $q, q' \in Q$.

Proof. (**Proof of Claim**.) Fix u. The claim is trivially true if $|u| \leq n^n$. Now consider the case $|u| > n^n$. Observe that for each $0 \leq j \leq |u| - 1$, the $n \times n$ matrix $\delta_{u[0:j]}$ is such that each row contains exactly one non-zero entry and this entry is 1. As there are atmost n^n such matrices, it follows that there must be $0 \leq j_1 < j_2 \leq |u| - 1$ such that $\delta_{u[0:j_1]} = \delta_{u[0:j_2]}$. This implies that $\delta_{u[0:j_1]}\delta_{u[j_2+1:|u|-1]} = \delta_{u[0:j_2]}\delta_{u[j_2+1:|u|-1]} = \delta_u$. Let $v_1 = u[0 : j_1]\ u[j_2 + 1 : |u| - 1]$. Observe that $|v_1| < |u|$ and $\delta_u = \delta_{v_1}$. If $|v_1| \leq n^n$, the claim will follow. Otherwise, we can similarly construct v_2 such that $|v_2| < |v_1|$ and $\delta_{v_1} = \delta_{v_2}$. Therefore, $|v_2| < |v_1| < |u|$ and $\delta_u = \delta_{v_2}$. If $|v_2| \leq n^n$ then the claim will follow. Otherwise, we can proceed in a similar fashion and construct $v_3, v_4 \ldots, v_r$ such that $|v_r| < |v_{r-1}| < \cdots < |v_1| < |u|$ and $\delta_u = \delta_{v_r}$ until $|v_r| \leq n^n$. (**End: Proof of claim**) □

Induction Hypothesis: Assume that the Lemma is true for each $k \leq k_0$.

Fix a (k_0+1)-HPA $\mathcal{A} = (Q, q_s, \delta, \mathsf{Acc})$ and $0 < \epsilon < 1$. For each $0 \leq i \leq k_0+1$, let $Q_i \subseteq Q$ be the level i states. Let $|Q| = n$ and let $|Q_0| = n_0$. Without loss of generality, we can assume that there is a state $q \in Q_0$ and an $a \in \Sigma$ such that $0 < \delta_a(q, Q_0) < 1$. Otherwise, we can consider $\mathcal{A}$ to be a k_0-HPA. Let p_{max} be the maximum value in the set $\{\delta_a(q, Q_0) \mid q \in Q_0, a \in \Sigma, 0 < \delta_a(q, Q_0) < 1\}$. We have that $0 < p_{\mathsf{max}} < 1$. Fix the smallest natural number $r > 0$ such that $p_{\mathsf{max}}^r \leq \frac{\epsilon}{8}$. Let $\epsilon_0 = \frac{\epsilon}{2(rn_0+1)n}$ and $\epsilon_1 = n\epsilon_0$.

Let q_{rej} be a new state $\notin Q$. Consider now the k_0-HPA $\mathcal{B} = (Q \cup \{q_{\mathsf{rej}}\}, q_s, \delta^{\mathcal{B}}, \mathsf{Acc})$ where for each letter a,

- $\delta_a^{\mathcal{B}}(q_1, q_2) = \delta_a(q_1, q_2)$ if $q_1, q_2 \in Q \setminus Q_0$,
- $\delta_a^{\mathcal{B}}(q_1, q_2) = 1$ if $q_1 \in Q_0$ and $\delta_a(q_1, q_2) = 1$,
- $\delta_a^{\mathcal{B}}(q_1, q_{\mathsf{rej}}) = 1$ if $q_1 \in Q_0$ and $|\mathsf{post}(q_1, a)| > 1$, and
- $\delta_a^{\mathcal{B}}(q_{\mathsf{rej}}, q_{\mathsf{rej}}) = 1$.

It is easy to see that $\mathcal{B}$ is a k_0-HPA with $Q_0 \cup Q_1 \cup \{q_{\mathsf{rej}}\}$ as level 0 states, Q_{i+1} being level i states for $i > 0$. Let $\ell_{\mathcal{B},\epsilon_0}$ be as in Induction Hypothesis. Let $\ell_{\mathcal{A},\epsilon} = (rn_0 + 1)\ell_{\mathcal{B},\epsilon_0} + rn_0$.

Now, fix a word $u = a_0a_1 \ldots a_{m-1}$ of length m. For each $i < m$ and $q \in Q_0$, we shall say that there is a q-*leak* at position i if $\delta_{u[0:i-1]}(q, Q_0) > \delta_{u[0,i]}(q, Q_0)$. We will say that a q-leak at position $i < m$ is *interesting* if there are at most $(r - 1)$ q-leaks amongst $0, 1, \ldots i - 1$. We say a position $i < m$ is *significant* if there is an interesting q-leak at position i for some $q \in Q_0$. It is easy to see that there are at most rn_0 significant positions in u. Let $i_1, i_2, \ldots, i_p$ be the significant positions in u. Let $u_1, u_2, \ldots, u_{p+1}$ be such that $u = u_1 a_{i_1} u_2 a_{i_2} \ldots u_p a_{i_p} u_{p+1}$. We have that $p \leq rn_0$.

Now, for each $1 \leq j \leq p+1$, thanks to Induction Hypothesis, there is a word v_j such that $|v_j| \leq \ell_{\mathcal{B},\epsilon_0}$ and

$$\left\|\delta^{\mathcal{B}}_{u_j} - \delta^{\mathcal{B}}_{v_j}\right\| < \epsilon_0.$$

Let $v = v_1 a_{i_1} v_2 a_{i_2} \ldots v_p a_p v_{p+1}$. It is easy to see that $|v| \leq (p+1)\ell_{\mathcal{B},\epsilon_0} + p \leq (rn_0+1)\ell_{\mathcal{B},\epsilon_0} + rn_0 \leq \ell_{\mathcal{A},\epsilon}$. Furthermore, thanks to Proposition 6 and the fact that $p \leq rn_0$, it is the case that

$$\left\|\delta^{\mathcal{B}}_{u} - \delta^{\mathcal{B}}_{v}\right\| < \sum_{i=1}^{p} n\epsilon_0 + \epsilon_0 = p\epsilon_1 + \epsilon_0 \leq (p+1)\epsilon_1 \leq \frac{\epsilon}{2}. \tag{5}$$

Observe that by construction of $\mathcal{B}$, for each finite word w, state $q \in Q \setminus Q_0$ and state $q' \in Q$, we have that $\delta_w(q,q') = \delta^{\mathcal{B}}_w(q,q')$. Thus, using Equation 5, we get that for each state $q \in Q \setminus Q_0$ and state $q' \in Q$,

$$|\delta_u(q,q') - \delta_v(q,q')| \leq \frac{\epsilon}{2}. \tag{6}$$

So the induction hypothesis will follow if we can show the following claim.

Claim. For each $q \in Q_0$ and $q' \in Q$, we have that

$$|\delta_v(q,q') - \delta_u(q,q')| < \epsilon.$$

Proof. (**Proof of Claim.**) Fix q. For the sake of simplicity, we will assume that for each $i < m$, $\delta_{u[0:i]}(q, Q_0) \neq 0$. The case when $\delta_{u[0:i]}(q, Q_0) = 0$ for some i can be dealt with similarly. Let $q_0 = q$. For $1 \leq t \leq p+1$, let q_t be the unique state in Q_0 such that $\delta_{u_1 a_{i_1} u_2 a_{i_2} \ldots a_{i_{t-1}} u_t}(q, q_t) > 0$, and for $1 \leq t \leq p$, let q'_t be the state in Q_0 such that $\delta_{a_t}(q_t, q'_t) > 0$. Let $p_q \leq p$ be the largest number such that such that there is an interesting q-leak at position i_{p_q}.

Fix $1 \leq t \leq p_q$. Observe, that by definition of interesting q-leaks, it is the case that $\delta_{u_t}(q'_{t-1}, q_t) = 1$. By construction of $\mathcal{B}$,

$$\delta^{\mathcal{B}}_{u_t}(q'_{t-1}, q_t) = \delta_{u_t}(q'_{t-1}, q_t) = 1 \wedge \delta^{\mathcal{B}}_{u_t}(q'_{t-1}, q'') = \delta_{u_t}(q'_{t-1}, q'') = 0 \text{ for } q'' \in Q \setminus Q_0. \tag{7}$$

Also note that by construction of v_t, $\delta^{\mathcal{B}}_{v_t}(q'_{t-1}, q_t) > \delta^{\mathcal{B}}_{u_t}(q'_{t-1}, q_t) - \epsilon > 1 - \epsilon$. Since q'_{t-1}, q_t are states in Q_0 and the probability of transitioning from any state q_a to q_b in $\mathcal{B}$ is either 0 or 1, we get that $\delta^{\mathcal{B}}_{v_t}(q'_{t-1}, q_t) = 1$. Once again this implies that

$$\delta^{\mathcal{B}}_{v_t}(q'_{t-1}, q_t) = \delta_{v_t}(q'_{t-1}, q_t) = 1 \wedge \delta^{\mathcal{B}}_{v_t}(q'_{t-1}, q'') = \delta_{v_t}(q'_{t-1}, q'') = 0 \text{ for } q'' \in Q \setminus Q_0. \tag{8}$$

Now, there are two cases to consider.

– The first case is when the number of interesting q-leaks is $< r$. Note that in this case, for each $p_q < i \leq p+1$, it is the case that $\delta_{u_i}(q'_{i-1}, q_i) = 1$. Using this, we get once again as in Equations 7 and 8 that for each $p_q < i \leq p+1$ and $q'' \in Q \setminus Q_0$,

$$\delta^{\mathcal{B}}_{u_i}(q'_{i-1}, q_i) = \delta_{u_i}(q'_{i-1}, q_t) = 1 \wedge \delta^{\mathcal{B}}_{u_i}(q'_{t_i-1}, q'') = \delta_{u_i}(q'_{i-1}, q'') = 0.$$
$$\delta^{\mathcal{B}}_{v_i}(q'_{i-1}, q_i) = \delta_{v_i}(q'_{i-1}, q_t) = 1 \wedge \delta^{\mathcal{B}}_{v_i}(q'_{t_i-1}, q'') = \delta_{u_i}(q'_{i-1}, q'') = 0.$$

This observation in combination with Equations 7 and 8 yields that for each $q' \in Q$, $\delta_u(q, q') = \delta^{\mathcal{B}}_{u_1}\delta_{a_1}\delta^{\mathcal{B}}_{u_2}a_{i_2}\ldots a_{i_{p-1}}\delta^{\mathcal{B}}_{u_p}(q, q')$ and that $\delta_v(q, q') = \delta^{\mathcal{B}}_{v_1}\delta_{a_1}\delta^{\mathcal{B}}_{v_2}a_{i_2}\ldots a_{i_{p-1}}\delta^{\mathcal{B}}_{v_p}(q, q')$. Thus, using Proposition 6 and the fact that $p \leq rn_0$, we get that for each $q' \in Q$,

$$|\delta_u(q, q') - \delta_v(q, q')| < \sum_{i=1}^{p} n\epsilon_0 + \epsilon_0 = p\epsilon_1 + \epsilon_0 \leq (p+1)\epsilon_1 \leq \frac{\epsilon}{2}$$

The claim follows.

– The second case is when the number of interesting q-leaks is r. Let w be the word $u_1 a_{i_1} \ldots u_{p_q} a_{i_{p_q}}$ and w' be the word $v_1 a_{i_1} \ldots v_{p_q} a_{i_{p_q}}$. Let $\delta^1 = \delta^{\mathcal{B}}_{u_1}\delta_{a_1}\delta^{\mathcal{B}}_{u_2}a_{i_2}\ldots a_{i_{p-1}}\delta^{\mathcal{B}}_{u_{p_q}}$ and $\delta^2 = \delta^{\mathcal{B}}_{v_1}\delta_{a_1}\delta^{\mathcal{B}}_{v_2}a_{i_2}\ldots a_{i_{p-1}}\delta^{\mathcal{B}}_{v_p}$.
It is easy to see that Equations 7 and 8 imply that

$$\delta_w(q, q'_{p_q}) = \delta_{w'}(q, q'_{p_q}) \leq p^r_{\mathsf{max}} \leq \frac{\epsilon}{8} \tag{9}$$

and that for each $q'' \in Q$

$$\delta_w(q, q'') = \delta^1(q, q'') \text{ and } \delta_{w'}(q, q'') = \delta^2(q, q''). \tag{10}$$

It is easy to see that $\delta_u(q, Q_0) \leq \delta_w(q, q_{p_q}) \leq \frac{\epsilon}{8}$ and that $\delta_v(q, Q_0) \leq \delta_{w'}(q, q_{p_q}) \leq \frac{\epsilon}{8}$.
Hence, we only need to prove the claim for the case when $q' \in Q \setminus Q_0$. Fix $q' \in Q \setminus Q_0$.
Let u^{p_q} be the word $u_{p_q} a_{i_{p_q+1}} \ldots a_{i_p} u_{p+1}$. Equation 10 implies that

$$\begin{aligned}\delta_u(q, q') &= \delta_w(q, q'_{p_q})\delta_{u^{p_q}}(q'_{p_q}, q') + \textstyle\sum_{q'' \in Q \setminus Q_0} \delta_w(q, q'')\delta_{u^{p_q}}(q'', q')\\ &= \delta_w(q, q'_{p_q})\delta_{u^{p_q}}(q'_{p_q}, q') + \textstyle\sum_{q'' \in Q \setminus Q_0} \delta^1(q, q'')\delta^{\mathcal{B}}_{u^{p_q}}(q'', q')\\ &= \delta_w(q, q'_{p_q})(\delta_{u^{p_q}}(q'_{p_q} q') - \delta^{\mathcal{B}}_{u^{p_q}}(q'_{p_q}, q')) + \textstyle\sum_{q'' \in Q} \delta^1(q, q'')\delta^{\mathcal{B}}_{u^{p_q}}(q'', q')\\ &= \delta_w(q, q'_{p_q})(\delta_{u^{p_q}}(q'_{p_q}, q') - \delta^{\mathcal{B}}_{u^{p_q}}(q'_{p_q}, q')) + \delta^1\delta^{\mathcal{B}}_{u^{p_q}}(q, q').\end{aligned}$$

Similarly, we can show that

$$\delta_v(q, q') = \delta_{w'}(q, q'_{p_q})(\delta_{v^{p_q}}(q'_{p_q}, q') - \delta^{\mathcal{B}}_{v^{p_q}}(q'_{p_q}, q')) + \delta^2\delta^{\mathcal{B}}_{v^{p_q}}(q, q')$$

where v^{p_q} is the word $v_{p_q} a_{i_{p_q+1}} \ldots a_{i_p} v_{p+1}$.
As $\delta_w(q, q'_{p_q}) = \delta_{w'}(q, q'_{p_q})$ by Equation 9, we get that

$$\begin{aligned}\delta_u(q, q') - \delta_v(q, q') = \delta_w(q, q'_{p_q})(\delta_{u^{p_q}}(q'_{p_q}, q'') - \delta^{\mathcal{B}}_{u^{p_q}}(q'_{p_q}, q') - \delta_{v^{p_q}}(q'_{p_q}, q') +\\ \delta^{\mathcal{B}}_{v^{p_q}}(q'_{p_q}, q')) + (\delta^1\delta^{\mathcal{B}}_{u^{p_q}}(q, q') - \delta^2\delta^{\mathcal{B}}_{v^{p_q}}(q, q')).\end{aligned}$$

Hence using Equation 9 again,

$$\begin{aligned}|\delta_u(q,q') - \delta_v(q,q')| &\le 4\delta_w(q,q'_{p_q}) + |(\delta^1\delta^{\mathcal{B}}_{u^{p_q}}(q,q') - \delta^2\delta^{\mathcal{B}}_{v^{p_q}}(q,q')))| \\ &\le 4\tfrac{\epsilon}{8} + |(\delta^1\delta^{\mathcal{B}}_{u^{p_q}}(q,q') - \delta^2\delta^{\mathcal{B}}_{v^{p_q}}(q,q')))|.\end{aligned}$$

Now, using Proposition 6 and the fact that $p \le rn_0$, we get that

$$|(\delta^1\delta^{\mathcal{B}}_{u^{p_q}}(q,q') - \delta^2\delta^{\mathcal{B}}_{v^{p_q}}(q,q')))| < \sum_{i=1}^{p} n\epsilon_0 + \epsilon_0 = p\epsilon_1 + \epsilon_0 \le (p+1)\epsilon_1 \le \frac{\epsilon}{2}.$$

The claim follows. (**End: Proof of claim**)

□

We only have to show the upper bound for $\ell_{\mathcal{A},\epsilon}$. Without loss of generality, we will assume that for a k-HPA $\mathcal{A}$, there is at least one state for each level i of $\mathcal{A}$. Furthermore, $b > 1$.

From the existence proof, we have that if $\mathcal{A}$, is a 0-HPA then $\ell_{\mathcal{A},\epsilon} \le n^n$. If $\mathcal{A}$ is a $(k+1)$-HPA then $\ell_{\mathcal{A},\epsilon} = (rn_0+1)\ell_{\mathcal{B},\epsilon_0} + rn_0$ where $\mathcal{B}, r, n, n_0, \epsilon_0$ are as given in the Induction Hypothesis of the existence proof.

Recall that p_{max} is the maximum value in the set $\{\delta_a(q,Q_0) \mid q \in Q_0, a \in \Sigma, 0 < \delta_a(q,Q_0) < 1\}$ and that $r > 0$ is the smallest number such that $p^r_{\mathsf{max}} \le \frac{\epsilon}{8}$. This implies that $p^{r-1}_{\mathsf{max}} \ge \frac{\epsilon}{8}$ and hence $(r-1)\log\frac{1}{p_{\mathsf{max}}} \le \log\frac{8}{\epsilon}$. Now, we have the following:

Claim. $\log\frac{1}{p_{\mathsf{max}}} > 1 - p_{\mathsf{max}}$.

Proof. (**Proof of claim**) For $x \ge 1$ let $\ln(x)$ denote the logarithm of x to the base e. Now, we have that $\log x = (\log_2 e)\ln x$ and hence $\log x > \ln x$. Hence the claim will follow if we can show that $\ln x > 1 - \frac{1}{x}$ for $x > 1$. Indeed, let $f(x) = \ln x - (1 - \frac{1}{x})$. It suffices to show that $f(x) > 0$ for $x > 1$. Now, the derivative of f is $f'(x) = \frac{1}{x} - \frac{1}{x^2} = x(1 - \frac{1}{x})$. For $x > 1$, $f'(x) > 0$ and $f(x)$ is a strictly increasing function. This implies that $f(x) > f(1)$ for each $x > 1$. As $f(1) = 0$, the claim follows. (**End: Proof of claim**) □

Hence $(r-1)(1-p_{\mathsf{max}}) < \log\lceil\frac{8}{\epsilon}\rceil$. If p_{max} is the quantity $\frac{c}{d}$ where c and d are nonnegative integers with $c < d$, we get that $(r-1)(d-c) < d\log(\lceil\frac{8}{\epsilon}\rceil)$ and hence $r - 1 < d\log(\lceil\frac{8}{\epsilon}\rceil)$ As $d \le 2^b$ we get that $r < 1 + 2^b\log\lceil\frac{8}{\epsilon}\rceil < 2^b(\log\lceil\frac{8}{\epsilon}\rceil + 1) = 2^b\log\lceil\frac{16}{\epsilon}\rceil$.

From the induction hypothesis [5] we have that

$$\ell_{\mathcal{A},\epsilon} \le 2rn\ell_{\mathcal{B},\epsilon_0} \le 2rn(\log\lceil\frac{2^{(b+1)k_0}n^{2k_0}}{\epsilon_0}\rceil)^{k_0}2^{(b+2)k_0}n^{n+k_0}.$$

Since $\epsilon_0 = \frac{\epsilon}{2(rn_0+1)n}$, we get that

$$\frac{2^{(b+1)k_0}n^{2k_0}}{\epsilon_0} \le \frac{2rn^2 2^{(b+1)k_0}n^{2k_0}}{\epsilon} = \frac{r2^{(b+1)k_0+1}n^{2(k_0+1)}}{\epsilon}.$$

[5] A careful examination of the existence proof tells us that q_{rej} does not play a role in bounding $\ell_{\mathcal{A},\epsilon}$.

From this and the fact that $r < 2^b \log\lceil \frac{16}{\epsilon} \rceil$, we get that

$$\frac{2^{(b+1)k_0} n^{2k_0}}{\epsilon_0} \leq \frac{2^{(b+1)(k_0+1)} n^{2(k_0+1)}}{\epsilon} \log\lceil \frac{16}{\epsilon} \rceil.$$

This implies that

$$\log\lceil \frac{2^{(b+1)k_0} n^{2k_0}}{\epsilon_0} \rceil \leq 2 \log\lceil \frac{2^{(b+1)(k_0+1)} n^{2(k_0+1)}}{\epsilon} \rceil.$$

Hence, we get that

$$\ell_{\mathcal{A},\epsilon} \leq r(\log\lceil \frac{2^{(b+1)(k_0+1)} n^{2(k_0+1)}}{\epsilon} \rceil)^{k_0} 2^{(b+2)k_0+2} n^{n+k_0+1}.$$

Given the fact that $r \leq 2^b \log\lceil \frac{16}{\epsilon} \rceil$, the above inequation implies

$$\ell_{\mathcal{A},\epsilon} \leq (\log\lceil \frac{2^{(b+1)(k_0+1)} n^{2(k_0+1)}}{\epsilon} \rceil)^{k_0+1} 2^{(b+2)k_0+2+b} n^{n+k_0+1}.$$

This completes the proof of the Lemma.

E Proof of Theorem 3

We need to prove the theorem in the case that $\mathcal{A}$ is a HPMA.

Clearly if the procedure outputs NO then $\mathsf{L}_{>x}(\mathcal{A}_W) \neq \emptyset$ for some good non-empty witness W. This immediately implies that $\mathsf{L}_{>x}(\mathcal{A}) \neq \emptyset$. Now suppose that the the algorithm outputs YES. Let ϵ_{m_0} be the value of ϵ when the algorithm outputs YES. As the program outputs YES, for each word w such that $|w| \leq \ell_{\mathcal{A},\frac{\epsilon_{m_0}}{n}}$ and each witness $W \in \mathcal{GW}$ we have that $\delta_w(q_s, W) + \epsilon_{m_0} < x$. Fix a finite word u and a finite witness W. Thanks to Lemma 1, there is finite word v such that $|v| \leq \ell_{\mathcal{A},\frac{\epsilon_{m_0}}{n}}$ and $\delta_u(q_s, W) < \delta_v(q_s, W) + \epsilon_{m_0} < x$ for each $W \in \mathcal{GW}$. Thus, $\delta_u(q_s, W) < x$ for each finite word u and $W \in \mathcal{GW}$. Proposition 4 implies that $\mathsf{L}_{>x}(\mathcal{A}) = \emptyset$. Thus, if the algorithm terminates then it gives the correct answer even if x is not isolated.

We claim that the algorithm in Figure 2 terminates if $\mathsf{L}_{>x}(\mathcal{A}) \neq \emptyset$ or if $\mathsf{value}(\mathcal{A}) < x$. If $\mathsf{L}_{>x}(\mathcal{A}) \neq \emptyset$ then fix a word u and a $W \in \mathcal{GW}$ such that $\delta_u(q_s, W) > x$. Such a word and a witness set exist thanks to Proposition 4. Let $\epsilon^0 = \delta_u(q_s, W) - x$. Let m_0 be the smallest integer such that $n\epsilon_{m_0} < \epsilon^0$. Thanks to Lemma 1, there is a finite word v such that $|v| \leq \ell_{\mathcal{A},\frac{\epsilon_{m_0}}{n}}$ and $\delta_v(q_s, W) > \delta_u(q_s, W) - n\epsilon_{m_0} = x + \epsilon^0 - n\epsilon_{m_0} > x$. Thus $\mathsf{approx\_value} > x$ in the m_0th unrolling of the while loop and the algorithm terminates.

If $\mathsf{value}(\mathcal{A}) < x$ then thanks to Proposition 5, $\mathsf{value}(\mathcal{A}_W) \leq \mathsf{value}(\mathcal{A})$ for each witness $W \in \mathcal{GW}$. Let $\epsilon^0 = x - \mathsf{value}(\mathcal{A}, x)$. Let m_0 be the smallest integer such that $\epsilon_{m_0} < \epsilon^0$. It is easy to see that the algorithm will terminate in the m_0th unrolling of the loop as for every finite word w and witness $W \in \mathcal{GW}$, it is the case that $\delta_w(q_s, W) + \epsilon_{m_0} \leq (x - \epsilon^0) + \epsilon_{m_0} < x$.

The Theorem follows from the fact that if x is an isolated cutpoint of $\mathcal{A}$ and $\mathsf{L}_{>x}(\mathcal{A}) = \emptyset$ then $\mathsf{value}(\mathcal{A}) < x$.

F Proof of Theorem 4

It suffices to consider the cases when $\bowtie$ is $\subseteq$ or when $\bowtie$ is $\supseteq$. We prove the two parts of the theorem separately. For the purposes of this proof, a deterministic automaton DA is a PA such that all transition probabilities are either 1 or 0.

1. *Claim.* There is an algorithm such that given a k-HPFA $\mathcal{A}$, $x \in (0,1)$ such that x is isolated for $\mathcal{A}$ and a regular language R, decides if $\mathsf{L}_{>x}(\mathcal{A}) \cap R = \emptyset$.

   Proof. (**Proof of Claim**.) Let $\mathcal{A} = (Q, q_s, \delta, Q_f)$. Let R be given by the DFA $\mathcal{A}' = (Q', q'_s, \delta', Q'_f)$. Consider the PFA $\mathcal{A}^{\mathsf{new}} = (Q^{\mathsf{new}}, q_s^{\mathsf{new}}, \delta^{\mathsf{new}}, Q_f^{\mathsf{new}})$ where $Q^{\mathsf{new}} = Q \times Q'$, $q_s^{\mathsf{new}} = (q_s, q'_s)$, $Q_f^{\mathsf{new}} = Q_f \times Q'_f$ and for each $q_1, q_2 \in Q$, $q'_1, q'_2 \in Q'$, $\delta^{\mathsf{new}}((q_1, q'_1)(q_2, q'_2)) = \delta(q_1, q_2)\delta'(q'_1, q'_2)$. It is easy to see that $\mathcal{A}^{\mathsf{new}}$ is also a k-HPA with level i states of $\mathcal{A}^{\mathsf{new}}$ being the set of all states (q, q') such that q is a level i state of $\mathcal{A}$. Furthermore, it is easy to see that for each finite word u, $\mathsf{P}_{\mathcal{A}^{\mathsf{new}}}(u) = \mathsf{P}_{\mathcal{A}}(u)$ if $u \in R$ and is 0 otherwise. This immediately implies the following:
   (a) $\mathsf{L}_{>x}(\mathcal{A}^{\mathsf{new}}) = \mathsf{L}_{>x}(\mathcal{A}) \cap R$.
   (b) Since x is an isolated cutpoint for $\mathcal{A}$ and $x \neq 0$, x is also an isolated cut-point of $\mathcal{A}^{\mathsf{new}}$.
   The result now follows from Theorem 3. (**End: Proof of claim**)

   Consider first the case when $\bowtie$ is $\subseteq$. Let $\mathcal{A} =$ be a k-HPFA with $\mathsf{Acc} = Q_f$. Observe that $\mathsf{L}_{>x}(\mathcal{A}) \subseteq R$ iff $\mathsf{L}_{>x}(\mathcal{A}) \cap (\Sigma^* \setminus R) = \emptyset$. Now $\Sigma^* \setminus R$ is also a regular language and its DFA can be computed from the DFA of R. The result now follows from the Claim 1.
   Consider now the case when $\bowtie$ is $\subseteq$. Let $\mathcal{A} = (Q, q_s, \delta, \mathsf{Acc})$ be a k-HPFA with $\mathsf{Acc} = Q_f$. Observe that $\mathsf{L}_{>x}(\mathcal{A}) \supseteq R$ iff $(\Sigma^* \setminus \mathsf{L}_{>x}(\mathcal{A})) \cap R = \emptyset$. Now, let $\mathcal{A}^c = (Q, q_s, \delta, Q \setminus Q_f)$. As x is an isolated cut-point of $\mathcal{A}$, it is easy to see that $1-x$ is an isolated cutpoint of $\mathcal{A}^c$ and that $\Sigma^* \setminus \mathsf{L}_{>x}(\mathcal{A}) = \mathsf{L}_{>1-x}(\mathcal{A}^c)$. The result now follows from Claim 1.
2. *Claim.* There is an algorithm such that given a k-HPMA $\mathcal{A}$, $x \in (0,1)$ such that x is isolated for $\mathcal{A}$ and an ω-regular language R, decides if $\mathsf{L}_{>x}(\mathcal{A}) \cap R = \emptyset$.

   Proof. (**Proof of Claim**.) Let $\mathcal{A} = (Q, q_s, \delta, \mathsf{Acc})$. Let R be given by the DMA $\mathcal{A}' = (Q', q'_s, \delta', \mathsf{Acc}')$.
   Consider the PMA $\mathcal{A}^{\mathsf{new}} = (Q^{\mathsf{new}}, q_s^{\mathsf{new}}, \delta^{\mathsf{new}}, \mathsf{Acc}^{\mathsf{new}})$ where $Q^{\mathsf{new}} = Q \times Q'$, $q_s^{\mathsf{new}} = (q_s, q'_s)$ and for each $q_1, q_2 \in Q$, $q'_1, q'_2 \in Q'$, $\delta^{\mathsf{new}}((q_1, q'_1)(q_2, q'_2)) = \delta(q_1, q_2)\delta'(q'_1, q'_2)$. $\mathsf{Acc}^{\mathsf{new}}$ is defined as follows. A set $A \in \mathsf{Acc}^{\mathsf{new}}$ if and only if the set $\mathsf{proj}_1(A) = \{q_1 \in Q \mid \exists q_2 \in Q'.\ (q_1, q_2) \in A\}$ is in the set Acc and the set $\mathsf{proj}_2(A) = \{q_2 \in Q' \mid \exists q_1 \in Q.\ (q_1, q_2) \in A\}$ is in the set Acc'.
   It is easy to see that $\mathcal{A}^{\mathsf{new}}$ is also a k-PMA with level i states of $\mathcal{A}^{\mathsf{new}}$ being the set of all states (q, q') such that q is a level i state of $\mathcal{A}$. Furthermore, it is easy to see that for each infinite word α, $\mathsf{P}_{\mathcal{A}^{\mathsf{new}}}(\alpha) = \mathsf{P}_{\mathcal{A}}(\alpha)$ if $\alpha \in R$ and is 0 otherwise. This immediately implies the following:
   (a) $\mathsf{L}_{>x}(\mathcal{A}^{\mathsf{new}}) = \mathsf{L}_{>x}(\mathcal{A}) \cap R$.

(b) Since x is an isolated cutpoint for $\mathcal{A}$ and $x \neq 0$, x is also an isolated cut-point of $\mathcal{A}^{\mathsf{new}}$.

The result now follows from Theorem 3. (**End: Proof of claim**)

Consider first the case when $\bowtie$ is $\subseteq$. Let $\mathcal{A} =$ be a k-HPMA with $\mathsf{Acc} = Q_f$. Observe that $\mathsf{L}_{>x}(\mathcal{A}) \subseteq R$ iff $\mathsf{L}_{>x}(\mathcal{A}) \cap (\Sigma^\omega \setminus R) = \emptyset$. Now $\Sigma^* \setminus R$ is also a regular language and its DMA can be computed from the DMA of R. The result now follows from the Claim 1.

Consider now the case when $\bowtie$ is $\subseteq$. Let $\mathcal{A} = (Q, q_s, \delta, \mathsf{Acc})$ be a k-HPMA with $\mathsf{Acc} = Q_f$. Observe that $\mathsf{L}_{>x}(\mathcal{A}) \supseteq R$ iff $(\Sigma^\omega \setminus \mathsf{L}_{>x}(\mathcal{A})) \cap R = \emptyset$. Now, let $\mathcal{A}^c = (Q, q_s, \delta, 2^Q \setminus \mathsf{Acc})$. As x is an isolated cut-point of $\mathcal{A}$, it is easy to see that $1 - x$ is an isolated cutpoint of $\mathcal{A}^c$ and that $\Sigma^\omega \setminus \mathsf{L}_{>x}(\mathcal{A}) = \mathsf{L}_{>1-x}(\mathcal{A}^c)$. The result now follows from Claim 1.

G Proof of Theorem 5

Now, we first show that the problem of checking if $\mathsf{value}(\mathcal{A}) = \frac{1}{2}$ for a given 2-HPFA $\mathcal{A}$ is **co-R.E.**-hard. We do this by reducing the non-halting problem of a 2-counter machine to the above problem. More specifically, we present an algorithm that given a deterministic 2-counter machine T, outputs a 2-HPA $\mathcal{A}_T$ such that $\mathsf{value}(\mathcal{A}_T) = \frac{1}{2}$ iff T does not halt starting from the initial configuration,i.e., one in which both counters have zero values. In our earlier papers [4, 3], we gave a reduction from the halting problem of a deterministic 2-counter machines to the problem of checking if a given 2-HPA accepts at least one string with probability $> \frac{1}{2}$. The constrcution of [4, 3], given a 2-counter machine T on input 0 outputs a 2-HPA $\mathcal{B}_T$ such that $\mathcal{B}_T$ accepts a string with probability $> \frac{1}{2}$ iff the unique computation of T starting from the initial configuration is a halting computation. The input to $\mathcal{B}_T$ is a sequence of configurations of T which are triples of the form (q, a^i, b^j) where q is the control state of T, i, j are the values of the two counters. There is a special symbol τ that indicates the end of the computation. $\mathcal{B}_T$ has two special absorbing states state q_a and q_{rej}. q_a is the only final state of $\mathcal{B}_T$. In the following, we shall call the probability of rejecting an input word u to be the probability of being in q_{rej}. $\mathcal{B}_T$ has the following properties:

1. If its input sequence does not contain τ then it is accepted with probability 0. Furthermore, once τ is encountered then the sum of probabilities of being in q_a and in q_{rej} is one.
2. If its input sequence is $w\tau v$ and a prefix of w is a valid halting computation of T then it is accepted with probability $> \frac{1}{2}$. Furthermore probability of accepting $w\tau v$ is exactly the probability of accepting $u\tau$ where u represents the halting computation of T.
3. All words $u\tau w$ such that u is a valid prefix of a computation of T are accepted with probability $< \frac{1}{2}$. If T does not halt, then the least upper bound of acceptance probabilities of the words $u\tau w$ such that u is a valid prefix of it's non-halting computation is $\frac{1}{2}$.

4. On all input sequences u that are not valid computations, the probability of rejecting u is $> \frac{1}{2}$.

We slightly modify $\mathcal{B}_T$, as given below, to obtain $\mathcal{A}_T$. We modify $\mathcal{B}_T$, to obtain $\mathcal{A}_T$, so that $\mathcal{A}_T$ goes to the rejecting state q_{rej} from all states including q_a with probability 1 upon encountering the halting state in the state component of an input configuration. In all other cases, $\mathcal{A}_T$ has same transition probabilities as $\mathcal{B}_T$. Thus, $\mathcal{A}_T$ satisfies properties (1), (3) and (4) given above and further it accepts any sequence that contains the halting state with probability 0. Assume that T halts. Let ℓ be the length of the input sequence representing the halting computation. Thanks to property (4), if the input sequence u is not a valid computation, and does not contain the halting state, then there is a prefix v of u that is of length $< \ell$ which represents an invalid computation, and is rejected with probability $> \frac{1}{2}$. It is easy to see that this implies that $\mathsf{value}(\mathcal{A}_T) < \frac{1}{2}$. Property (3) implies that $\mathsf{value}(\mathcal{A}_T) = \frac{1}{2}$ if T does not halt starting from the initial configuration. As a consequence $\mathsf{value}(\mathcal{A}_T) = \frac{1}{2}$ iff T does not halt starting from the initial configuration.

It is easy to see that if we consider $\mathcal{A}_T$ to be a HPMA with the acceptance condition being $\{\{q_a\}\}$, it is easy to see that $\mathsf{value}(\mathcal{A}_T) = \frac{1}{2}$ iff T does not halt starting from the initial configuration.

It is fairly straightforward to reduce the problem of checking if $\mathsf{value}(\mathcal{A}) = \frac{1}{2}$, for a given 2-HPA $\mathcal{A}$, to the problem of checking if $\mathsf{value}(\mathcal{B}) = x$, for a given 2-HPA $\mathcal{B}$ and a rational $x \in (0, 1)$, showing that the later problem is also **co-R.E.**-hard. We briefly present this reduction. We introduce a new input symbol a and a new start state r, and modify $\mathcal{A}$ as follows, to obtain $\mathcal{B}$. If $x < \frac{1}{2}$, from the start state r, on input a, $\mathcal{B}$ goes to a reject state (which is an absorbing state) with probability $1 - 2x$ and goes to the start of $\mathcal{A}$ with probability $2x$. If $x > \frac{1}{2}$, from state r, on input a, $\mathcal{B}$ goes to a new accept state (which is also an absorbing state) with probability $2x - 1$, and goes to the start state of $\mathcal{A}$ with probability $2(1 - x)$. From all the states of $\mathcal{A}$, $\mathcal{B}$ on input a goes to the reject state with probability 1, and on all other input symbols $\mathcal{B}$ has the same transitions as $\mathcal{A}$. The accept states of $\mathcal{B}$ are those of $\mathcal{A}$ and the newly added accept state for the case when $x > \frac{1}{2}$. Now, consider an input string $w = au$ where $u\Sigma^*$. Let y be the probability of acceptance of u by $\mathcal{A}$. It is easily see that w is accepted by $\mathcal{B}$ with probability $2xy$ or $2(x + y - xy) - 1$ depending on whether $x < \frac{1}{2}$ or $x > \frac{1}{2}$, respectively. From this, it is easy to see that $\mathsf{value}(\mathcal{B}) = x$ iff $\mathsf{value}(\mathcal{A}) = \frac{1}{2}$.

H Proof of Theorem 6

For the upper bound, we appeal to Proposition 2 which states that for each PA $\mathcal{A} = (Q, q_s, \delta, \mathsf{Acc})$ and $x \in (0, 1)$, there is a constructible PA $\mathcal{B}$ such that $\mathsf{value}(\mathcal{B}) = \frac{1}{4}$ iff x is not a isolated cutpoint of $\mathcal{A}$. The result follows by observing that for $k \geq 1$, if $\mathcal{A}$ is a k-HPA then $\mathcal{B}$ is a $2k$-HPA.

To show that the isolated cut point problem is **R.E.**-hard for 2-HPFA, we use the same reduction as given in the proof of Theorem 5. There, for a given 2-counter machine T, we constructed a 2-HPFA $\mathcal{B}_T$. It is fairly easy to see that

$\frac{1}{2}$ is an isolated cut point of $\mathcal{B}_T$ iff T halts starting from the initial configuration. This shows that the problem of deciding if $\frac{1}{2}$ is an isolated cut point of a given 2-HPFA $\mathcal{A}$ is **R.E.**-hard. Using the same technique as given in the proof of Theorem 5, we also show that for any given rational $x \in (0,1)$ and 2-HPMA $\mathcal{A}$, the problem of determining if x is an isolated cut point of $\mathcal{A}$ with respect to x is **R.E.**-hard.

I Proof of Lemma 2

First we prove the if part of the lemma. Assume that there is no string that is accepted by $\mathcal{A}$ with probability greater than x and at least one of the two conditions holds. Clearly, if condition (1) holds, then $x = \mathsf{value}(\mathcal{A})$. Now assume that condition (2) holds. Since $\forall i \geq 0$, $\delta_{uv^{i+1}}(q_s, W) > \delta_{uv^i}(q_s, W)$, it has to be that $\mathsf{post}(q_s, u) \cap Q_0 \neq \emptyset$. Let $\mathsf{post}(q_s, u) \cap Q_0 = \{q_1\}$. Clearly, $\delta_v(q_1, q_1) > 0$. Let w be any string in L_W. Then, $\forall i \geq 0$, $\mathsf{P}_{\mathcal{A}}(uv^i w) = \delta_{uv^i}(q_s, W) + \delta_{uv^i}(q_s, q_1)\delta_w(q_1, Q_f)$. From Lemma 3.8 of [3], we observe that the sequence $s = \left(\mathsf{P}_{\mathcal{A}}(uv^i w)\right)_{i=0}^{\infty}$ is either a monotonically increasing or a monotonically decreasing sequence. Observe that $\delta_{uv^i}(q_s, q_1) = \delta_u(q_s, q_1)(\delta_v(q_1, q_1))^i$ and hence $\lim_{i\to\infty} \delta_{uv^i}(q_s, q_1) = 0$. From this, we see that $\lim_{i\to\infty} \mathsf{P}_{\mathcal{A}}(uv^i w) = \lim_{i\to\infty} \delta_{uv^i}(q_s, W) = x$. Observe that the sequence s cannot be a decreasing sequence, since that would imply that it's first element $\mathsf{P}_{\mathcal{A}}(uw) > x$, contradicting the assumption that no sequence is accepted with probability greater than x. Hence s is an increasing sequence. Since $\lim_{i\to\infty} \mathsf{P}_{\mathcal{A}}(uv^i w) = x$, we have $\mathsf{value}(\mathcal{A}) = x$.

Now, we prove the "only if" part. Assume that $x = \mathsf{value}(\mathcal{A})$. Clearly no string is accepted by $\mathcal{A}$ with probability $> x$. Now, we have two cases. The first case is when there exists an input string that is accepted with probability x. From Lemma 3.12 of [3], we get condition (1). Now we consider the case when there is no string that is accepted with probability x; we will show that condition (2) holds in this case. In this case, we see that there exists an infinite sequence $(u_i)_{i=1}^{\infty}$ of input strings such that $x - \frac{1}{2^i} < \mathsf{P}_{\mathcal{A}}(u_i) < x$ for $i > 0$. Now, for each $i > 0$, let u'_i be the shortest input sequence such that $\mathsf{P}_{\mathcal{A}}(u'_i) > x - \frac{1}{2^i}$. Observe that u'_i may be shorter than u_i. Clearly, the length of u'_i increases unboundedly with i. Furthermore, it is not difficult to see that $\exists i_0 > 0$ such that $\forall i \geq i_0$, $|u'_i| > 2^n$. Fix any $k \geq i_0$. Using the same argument as given in the proof of Lemma 3.9 of [3], it can be shown that there exist strings v_k, w_k, t_k and a set $X_k \subseteq Q_1$ such that the following hold:

(a) $u'_k = v_k w_k t_k$ and $|v_k w_k| \leq 2^n$;
(b) $\mathsf{post}(q_s, v_k) \cap Q_0 = \mathsf{post}(q_s, v_k w_k) \cap Q_0 \neq \emptyset$;
(c) $X_k = \{q \in \mathsf{post}(q_s, v_k) \cap Q_1 \mid \mathsf{post}(q, w_k t_k) \subseteq Q_f\} = \{q \in \mathsf{post}(q_s, v_k w_k) \cap Q_1 \mid \mathsf{post}(q, t_k) \subseteq Q_f\}$;
(d) for all $\ell > 0$, $\mathsf{P}_{\mathcal{A}}(s_{k,\ell}) > \mathsf{P}_{\mathcal{A}}(s_{k,\ell-1})$ where $s_{k,\ell} = v_k(w_k)^\ell t_k$.

Condition (b) states that the Q_0-state reached after v_k is same as the one reached after $v_k w_k$. Condition (c) states that the two sets of Q_1-states, reached after v_k (and after $v_k w_k$), from which a final state, i.e., a state in Q_f, is reached at the end of β_k, are both same and are equal to some X_k. This implies that $\mathsf{post}(X_k, w_k) \subseteq X_k$. Condition (d) states that the acceptance probability of $s_{k,\ell}$ increases with ℓ. Observe that $u'_k = s_{k,1}$ and $x - \frac{1}{2^k} < \mathsf{P}_{\mathcal{A}}(s_{k,1}) < x$. Hence, we have $x - \frac{1}{2^k} < \mathsf{P}_{\mathcal{A}}(s_{k,\ell}) < x$ for all $\ell > 0$. Hence $x - \frac{1}{2^k} < \lim_{\ell\to\infty} \mathsf{P}_{\mathcal{A}}(s_{k,\ell}) \leq x$. Now observe that the probability, of the automaton being in a Q_0-state after $v_k(w_k)^\ell$, goes to zero in the limit as $\ell \to \infty$. From this we see that $\lim_{\ell\to\infty} \mathsf{P}_{\mathcal{A}}(s_{k,\ell}) = \lim_{\ell\to\infty} \delta_{v_k(w_k)^\ell}(q_s, X_k)$. Hence, $x - \frac{1}{2^k} < \lim_{\ell\to\infty} \delta_{v_k(w_k)^\ell}(q_s, X_k) \leq x$.

Since $|v_k w_k| \leq 2^n$ and $|X_k| \leq n$, we see that there exists an infinite sequence of integers $i_0 < k_1 < ... < k_j < ...$ such that, $\forall j > 0,\ v_{k_j} = v', w_{k_j} = w',\ X_{k_j} = W'$ for some v', w', W'. Notice that W' is a good witness set. We have

$$\forall j > 0,\ x - \frac{1}{2^{k_j}} < \lim_{\ell\to\infty} \delta_{v'(w')^\ell}(q_s, W') \leq x$$

Since, $\lim_{j\to\infty}\ x - \frac{1}{2^{k_j}} = x$, it follows that $\lim_{\ell\to\infty} \delta_{v'(w')^\ell}(q_s, W') = x$. From this, we see that condition (2) of the lemma holds for $u = v', v = w', W = W'$.

□