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, Columbia, USA
chadhar@missouri.edu
[2] University of Illinois, Chicago, USA
sistla@uic.edu
[3] University of Illinois at Urbana-Champaign, Urbana, 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) [12] 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

© Springer-Verlag GmbH Germany 2017
J. Esparza and A.S. Murawski (Eds.): FOSSACS 2017, LNCS 10203, pp. 231–247, 2017.
DOI: 10.1007/978-3-662-54458-7_14

tools for HPAs [4]. HPAs were introduced in [12] 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 [12]. 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 $\mathbf{\Pi_2^0}$-complete [9,12].

Surprisingly, however, the landscape changes completely when the threshold is taken to be $x \in (0,1)$. Even 1-HPAs[1] can recognize non-regular languages when the acceptance threshold is $\frac{1}{2}$ [11]. Though emptiness and universality problems are decidable for 1-HPAs [11], 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 [4,8], 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 [19] 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

[1] 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 [5,13,16]. 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 [20], together with an algorithm for emptiness checking for 1-HPA [11]. 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 $\mathbf{\Pi^0_2}$-complete [13]. 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. For brevity, we have omitted some proofs which can be found in [10].

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 [14] and for infinite words in [8]. The emptiness problem for 2-HPAs (and higher) with non-extremal thresholds is also undecidable [4,8]. 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 [13] the emptiness problem is decidable when the cut-point is isolated. Eventually weakly ergodic PFAs are incomparable to HPAs considered here; Fig. 1(c) in [13] is an example of a HPA that is not eventually weakly ergodic. The value problem is undecidable for PFAs [5], even for extremal thresholds [16]; it is known to be $\mathbf{\Pi_2^0}$-complete [13]. The problem of checking if the value is 1 was shown to be **PSPACE**-complete for *leak-tight automata* [15] 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 $\mathbf{\Pi_2^0}$-complete [13] 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) $\kappa = s_0 s_1 \dots$ 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 $s_i \dots s_j$. For $i < |\kappa|$, $\kappa[i : \infty]$ is the sequence $s_i s_{i+1} \dots$ if $|\kappa| = \omega$, and is the sequence $s_i \dots s_{|\kappa|-1}$ 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, $\delta : Q \times \Sigma \times Q \rightarrow [0, 1]$ 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 $u = a_0 a_1 \dots a_n \in \Sigma^+$, 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 $\{(q, v) \mid q \in Q, v$ 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 [17,21]. 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 [21] 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 [21]. 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 [18]. 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(s). 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.

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*,* $0 \leq i \leq k$ *and every* $q \in Q_i$ *and* $a \in \Sigma$*,* $|\mathsf{post}(q,a) \cap Q_i| \leq 1$*; and,*
- *for every* i*,* $0 < i \leq k$*,* $q \in Q_i$ *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 $0 \leq i \leq k$, 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. This is 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.

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. Then $\mathsf{value}(\mathcal{A}) = \mathsf{max}_{W \in \mathcal{GW}}\, \mathsf{value}(\mathcal{A}_W)$.*

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 [1,7,19]. One important result due to Rabin [19] 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.

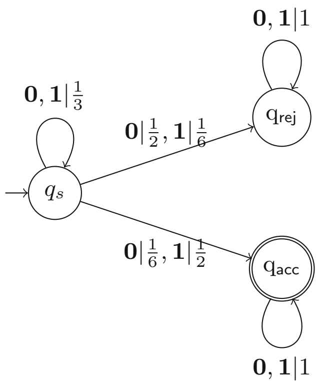
$\mathbf{0},\mathbf{1}|1$
$\mathbf{0},\mathbf{1}|\frac{1}{3}$
$\mathbf{0}|\frac{1}{2},\mathbf{1}|\frac{1}{6}$
$\mathsf{q}_{\mathsf{rej}}$
q_s
$\mathbf{0}|\frac{1}{6},\mathbf{1}|\frac{1}{2}$
$\mathsf{q}_{\mathsf{acc}}$
$\mathbf{0},\mathbf{1}|1$

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

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.*

Example 1. Consider the HPMA $\mathcal{A}_{\mathsf{isolated}}$ shown in Fig. 1. It has 3 states — $q_s, \mathsf{q}_{\mathsf{rej}}, \mathsf{q}_{\mathsf{acc}}$ — with q_s as initial state. The acceptance condition is given by $\{\{\mathsf{q}_{\mathsf{acc}}\}\}$. For any n, let $x_n = \frac{3}{4}(1-(\frac{1}{3})^n)$. We will show a couple of properties about $\mathcal{A}_{\mathsf{isolated}}$ and x_n. First we show that x_n is an isolated cut-point with degree of isolation $\frac{1}{6}(\frac{1}{3})^n$, and second, that the set of infinite strings accepted with probability $> x_n$ is exactly $L_n = \mathbf{1}^n\{\mathbf{0},\mathbf{1}\}^\omega$.

For any $u \in \{\mathbf{0},\mathbf{1}\}^*$, let the acceptance and rejection probabilities of u be the probabilities of reaching the states $\mathsf{q}_{\mathsf{acc}}, \mathsf{q}_{\mathsf{rej}}$, respectively, on input u staring from q_s. Observe that every string in L_n, is accepted by $\mathcal{A}$, with probability greater than the acceptance probability of $u = 1^n0$, which is $\sum_{0\le i<n}(\frac{1}{3})^i\frac{1}{2}+(\frac{1}{3})^n\frac{1}{6}$ and is equal to $\frac{3}{4}(1-(\frac{1}{3})^n)+(\frac{1}{3})^n\frac{1}{6}$. Now, consider any input sequence not in L_n, i.e., sequence in $\{\mathbf{0},\mathbf{1}\}^\omega \setminus L_n$. The probability of rejecting any such string is $>$ the rejection probability the string $1^{n-1}01$. The rejection probability of $1^{n-1}01$ is $\sum_{0\le i<n-1}(\frac{1}{3})^i\frac{1}{6}+(\frac{1}{3})^{n-1}\frac{1}{2}+(\frac{1}{3})^n\frac{1}{6}$, which after some simplification, is $y = \frac{1}{4}(1-(\frac{1}{3})^{n-1})+\frac{5}{3}(\frac{1}{3})^n$. From these observations, it follows that, for any $\kappa \notin L_n$, the probability of accepting κ is $< 1-y = \frac{3}{4}(1-(\frac{1}{3})^n)-\frac{1}{6}(\frac{1}{3})^n = x_n - \frac{1}{6}(\frac{1}{3})^n$. In addition, for any $\kappa \in L_n$, the probability of accepting κ is $> x_n + \frac{1}{6}(\frac{1}{3})^n$. 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 $\leq \epsilon$.

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.*

We sketch the key ideas of the proof of Lemma 1. 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 $\mathsf{poly}(\log(\frac{1}{\epsilon}))$ 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 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. $\square$

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.

```
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

Theorem 3. *The algorithm in Fig. 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.

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 Fig. 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 $\mathcal{A}$.

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 $\mathbf{\Sigma^0_2}$-complete [13]. 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 $\mathbf{\Pi^0_2}$-complete. For HPFAs, the problem of checking if 1 is isolated is known to be **PSPACE**-complete [15]. 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].

Theorem 5. *For each $k \geq 2$, the value decision problems for k-*HPFA*s and for k-*HPMA*s are* **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 $\mathsf{isValue}(\mathcal{A}, x)$ be the predicate

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

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

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

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

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

Assume now that $\mathsf{isValue}(\mathcal{A}, x)$ 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 Eq. 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 $\mathsf{isValueSm}(\mathcal{A}, x)$ is also true if $\mathsf{isValue}(\mathcal{A}, x)$ is true. Hence, $\mathsf{value}(\mathcal{A}) = x$ iff $\mathsf{isValueSm}(\mathcal{A}, x)$ 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 $\{\mathsf{isValue}(\mathcal{A}_W, x) \mid W \in \mathcal{GW}\}$ is true. The upper bound follows in this case.

The lower bound is proved by a modification of the proof of undecidability of emptiness problem for the 2-HPAs [2–4]. □

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.

Theorem 6. *For each $k \geq 2$, the isolation decision problems for k-*HPFA*s and k-*HPMA*s are **R.E.**-complete.*

5.1 Computing the Value of 1-HPAs

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 bounded "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 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 one of the conditions (1) or (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 $\leq 4rn2^n$. $\square$

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 problems for* 1-HPA*s and* 1-HPMA*s are in* **EXPTIME** *and are* **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 doubly exponential in n.

Now, we give an algorithm, that computes $\mathsf{value}(\mathcal{A})$, of time complexity only singly exponential in n. Let $N = 4rn2^n$ and $M = 2^N$ where r is the maximum of the sizes of the transition 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 [20] 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 [20], 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 [3,11]; the algorithm given in [11] 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 [3,11]. □

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 NSF CNS 1553548. A. Prasad Sistla was partially supported by NSF CCF 1319754 and NSF CNS 1314485. Mahesh Viswanathan was partially supported by NSF CNS 1314485.

References

1. Baier, C., Größer, M.: Recognizing ω-regular languages with probabilistic automata. In: 20th IEEE Symposium on Logic in Computer Science, pp. 137–146 (2005)
2. Ben, Y.: Model Checking Open Probabilistic Systems using Hierarchical probabilistic automata. Ph.D. thesis, University of Illinois, Chicago (2016)
3. Ben, Y., Chadha, R., Sistla, A.P., Viswanathan, M.: Decidable and expressive classes of probabilistic automata. https://www.cs.uic.edu/pub/Sistla/Publications/HPAjournal2016.pdf (2016). Manuscript under review
4. Ben, Y., Sistla, A.P.: Model checking failure-prone open systems using probabilistic automata. In: Finkbeiner, B., Pu, G., Zhang, L. (eds.) ATVA 2015. LNCS, vol. 9364, pp. 148–165. Springer, Cham (2015). doi:10.1007/978-3-319-24953-7_11
5. Bertoni, A.: The solution of problems relative to probabilistic automata in the frame of the formal languages theory. In: Siefkes, D. (ed.) GI 1974. LNCS, vol. 26, pp. 107–112. Springer, Heidelberg (1975). doi:10.1007/3-540-07141-5_213
6. Chadha, R., Kini, D., Viswanathan, M.: Decidable problems for unary PFAs. In: Norman, G., Sanders, W. (eds.) QEST 2014. LNCS, vol. 8657, pp. 329–344. Springer, Cham (2014). doi:10.1007/978-3-319-10696-0_26
7. Chadha, R., Sistla, A.P., Viswanathan, M.: On the expressiveness and complexity of randomization in finite state monitors. J. ACM 56(5) (2009)
8. Chadha, R., Sistla, A.P., Viswanathan, M.: Probabilistic Büchi automata with non-extremal acceptance thresholds. In: International Conference on Verification, Model Checking and Abstract Interpretation, pp. 103–117 (2010)
9. Chadha, R., Sistla, A.P., Viswanathan, M.: Power of randomization in automata on infinite strings. Log. Methods Comput. Sci. **7**(3), 1–22 (2011)
10. Chadha, R., Sistla, A.P., Viswanathan, M.: Emptiness under isolation and the value problem for hierarchical probabilistic automata (2017). https://www.cs.uic.edu/pub/Sistla/Publications/ValueProblem.pdf
11. Chadha, R., Sistla, A.P., Viswanathan, M., Ben, Y.: Decidable and expressive classes of probabilistic automata. In: Pitts, A. (ed.) FoSSaCS 2015. LNCS, vol. 9034, pp. 200–214. Springer, Heidelberg (2015). doi:10.1007/978-3-662-46678-0_13
12. Chadha, R., Sistla, A.P., Viswanathan, M.: Power of randomization in automata on infinite strings. In: Bravetti, M., Zavattaro, G. (eds.) CONCUR 2009. LNCS, vol. 5710, pp. 229–243. Springer, Heidelberg (2009). doi:10.1007/978-3-642-04081-8_16
13. Chadha, R., Sistla, A.P., Viswanathan, M.: Probabilistic automata with isolated cut-points. In: Chatterjee, K., Sgall, J. (eds.) MFCS 2013. LNCS, vol. 8087, pp. 254–265. Springer, Heidelberg (2013). doi:10.1007/978-3-642-40313-2_24
14. Condon, A., Lipton, R.J.: On the complexity of space bounded interactive proofs (extended abstract). In: Symposium on Foundations of Computer Science, pp. 462–467 (1989)
15. Fijalkow, N., Gimbert, H., Oualhadj, Y.: Deciding the value 1 problem for probabilistic leaktight automata. In: IEEE Symposium on Logic in Computer Science, pp. 295–304 (2012)
16. Gimbert, H., Oualhadj, Y.: Probabilistic automata on finite words: Decidable and undecidable problems. In: Abramsky, S., Gavoille, C., Kirchner, C., Meyer auf der Heide, F., Spirakis, P.G. (eds.) ICALP 2010. LNCS, vol. 6199, pp. 527–538. Springer, Heidelberg (2010). doi:10.1007/978-3-642-14162-1_44

17. Kemeny, J.G., Snell, J.L.: Denumerable Markov Chains. Springer, New York (1976)
18. Paz, A.: Introduction to Probabilistic Automata. Academic Press, Orlando (1971)
19. Rabin, M.O.: Probabilistic automata. Inf. Control **6**(3), 230–245 (1963)
20. Kwek, S., Mehlhorn, K.: Optimal search for rationals. Inf. Process. Lett. **86**(1), 23–26 (2003)
21. Vardi, M.: Automatic verification of probabilistic concurrent finite-state programs. In: Symposium on Foundations of Computer Science, pp. 327–338 (1985)