Decidable and Expressive classes of Probabilistic Automata

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

[1] University of Missouri, USA
[2] Univ. of Illinois, Chicago, USA
[3] Univ. of Illinois, Urbana-Champaign, USA

Abstract. Hierarchical probabilistic automata (HPA) are probabilistic automata whose states are partitioned into levels such that for any state and input symbol, at most one transition with non-zero probability goes to a state at the same level, and all others go to states at a higher level. We present expressiveness and decidability results for 1-level HPAs that work on both finite and infinite length input strings; in a 1-level HPA states are divided into only two levels (0 and 1). Our first result shows that 1-level HPAs, with acceptance threshold $1/2$ (both in the finite and infinite word cases), can recognize non-regular languages. This result is surprising in the light of the following two facts. First, all earlier proofs demonstrating the recognition of non-regular languages by probabilistic automata employ either more complex automata or irrational acceptance thresholds or HPAs with more than two levels. Second, it has been previously shown that simple probabilistic automata (SPA), which are 1-level HPAs whose accepting states are all at level 0, recognize only regular languages. We show that even though 1-level HPAs with threshold $1/2$ are very expressive (in that they recognize non-regular languages), the non-emptiness and non-universality problems are both decidable in **EXPTIME**. To the best our knowledge, this is the first such decidability result for any subclass of probabilistic automata that accept non-regular languages. We prove that these decision problems are also **PSPACE**-hard. Next, we present a new sufficient condition when 1-level HPAs recognize regular languages (in both the finite and infinite cases). Finally, we show that the emptiness and universality problems for this special class of HPAs is **PSPACE**-complete.

1 Introduction

Probabilistic automata (PA) [13, 12, 1, 10] are finite state machines that have probabilistic transitions on input symbols. Such machines can either recognize a language of finite words (probabilistic finite automata PFA [13, 12]) or a language of infinite words (probabilistic Büchi/Rabin/Muller automata [1, 10, 6]) depending on the notion of accepting run; on finite input words, an accepting run is one that reaches a final state, while on an infinite input, an accepting run is one whose set of states visited infinitely often satisfy a Büchi, Rabin, or Muller acceptance condition. The set of accepting runs in all these cases can be shown to be measurable and the probability of this set is taken to be probability of accepting the input word. Given an acceptance threshold x, the language $\mathsf{L}_{>x}(\mathcal{A})$ ($\mathsf{L}_{\geq x}(\mathcal{A})$) of a PA $\mathcal{A}$ is the set of all inputs whose acceptance probability is $> x$ ($\geq x$). In this paper the threshold x is always a rational number in $(0, 1)$.

Hierarchical probabilistic automata (HPA) are a syntactic subclass of probabilistic automata that are computationally more tractable for extremal thresholds [5] — problems of emptiness and universality which are undecidable for PAs on infinite words with threshold 0 become decidable for HPAs. Over finite words, the problem of deciding whether the infimum of acceptance probabilities is 0 also becomes decidable for HPAs [8], even though it is undecidable for general PAs [9]. Intuitively, a HPA is a PA whose states are stratified into (totally) ordered levels with the property that from any state q, and input a, the machine can transition with non-zero probability to at most one state in the same level as q, and all other probabilistic successors belong to a higher level. Such automata arise naturally as models of *client-server systems*. Consider such a system where clients can request services of multiple servers that can fail (catastrophically) with some probability. The state of the automaton models the global state of all the servers and inputs to the machine correspond to requests from the client to the servers. The levels of the automaton correspond to the number failed servers, with the lowest level modeling no failures. Since failed servers can't come back, the transitions in such a system satisfy the hierarchical nature. While HPAs are tractable with extremal thresholds, the emptiness and universality problems are undecidable for HPA with threshold $\frac{1}{2}$ [4]. In fact, solving these decision problems for 6-level HPAs is undecidable [4]. In this paper, we investigate how the landscape changes when we restrict our attention to 1-level HPAs.

1-level HPAs (henceforth simply called HPAs) are machines whose states are partitioned into two levels (0 and 1), with initial state in level 0, and transitions satisfying the hierarchical structure. These automata model client-server systems where only one server failure is allowed. Despite their extremely simple structure, we show that (1-level) HPAs turn out to be surprisingly powerful — they can recognize non-regular languages over finite and infinite words (even with threshold $\frac{1}{2}$). This result is significant because all earlier constructions of PFAs [12, 13] and probabilistic Büchi automata [10, 2] recognizing non-regular languages use either more complex automata or irrational acceptance thresholds or HPAs with more than two levels. Moreover, this result is also unexpected because it was previously shown that *simple probabilistic automata* only recognize regular languages [4, 5]. The only difference between (1-level) HPAs and simple probabilistic automata is that all accepting states of a simple probabilistic automaton are required to be in level 0 (same level as the initial state).

Next, we consider the canonical decision problems of emptiness and universality for (1-level) HPAs with threshold x. Decision problems for PAs with non-extremal thresholds are often computationally harder than similar questions when the threshold is extremal (either 0 or 1), and the problems are always undecidable [7, 5, 2, 12]. Even though 1-level HPAs are expressive, we show that both emptiness and universality problems for 1-level HPAs are decidable in **EXPTIME** and are **PSPACE**-hard. As far as we know, this is the first decidability result for any subclass of PAs with non-extremal thresholds that can recognize non-regular languages. Our decision procedure relies on observing that when the language of a HPA $\mathcal{A}$ is non-empty (or non-universal), then there is an input whose length is exponentially bounded in the size of the HPA that witnesses this fact.

Finally, we introduce a special subclass of (1-level) HPAs called *integer* HPAs. Integer HPA are HPAs where from any level 0 state q, on any input a, the probability of transitioning to a level 1 state is an integer multiple of the probability of the (unique) transition to a level 0 state on a from q. With this restriction, we can show that integer HPA with threshold x only recognize regular languages (over finite and infinite words). For integer HPAs, we show that the canonical decision decision problems of emptiness and universality are **PSPACE**-complete.

The rest of the paper is organized as follows. Section 2 has basic definitions, and introduces HPAs along with some useful propositions. The results characterizing the expressiveness and decidability of HPAs are presented in Section 3. The results on integer HPAs are presented in Section 4. Section 5 contains concluding remarks.

2 Preliminaries

We assume that the reader is familiar with finite state automata, regular languages, Büchi automata, 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.

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. 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 $\kappa \in S^* \cup S^\omega$, natural numbers $i, j \leq |\kappa|$, $\kappa[i : j]$ is the finite sequence $s_i \ldots s_j$ and $\kappa[i : \infty]$ is the infinite sequence $s_i s_{i+1} \ldots$, where $s_k = \kappa[k]$. The set of *finite prefixes* of κ is the set $\mathit{Pref}(\kappa) = \{\kappa[0 : j] \mid j \in \mathbb{N}, j \leq |\kappa|\}$. 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 $\mathsf{L}_1 \subseteq \Sigma^*$ and $\mathsf{L}_2 \subseteq \Sigma^\omega$, the set $\mathsf{L}_1\mathsf{L}_2$ is defined to be $\{u\alpha \mid u \in \mathsf{L}_1 \text{ and } \alpha \in \mathsf{L}_2\}$. Given $u \in S^+$, the word u^ω is the unique infinite sequence formed by repeating u infinitely often. An infinite word $\alpha \in S^\omega$ is said to be *ultimately* periodic if there are finite words $u \in S^*$ and $v \in S^+$ such that $\alpha = uv^\omega$. For an infinite word $\alpha \in S^\omega$, we write $\mathsf{inf}(\alpha) = \{s \in S \mid s = \alpha[i] \text{ for infinitely many } i\}$.

Languages. Given a finite alphabet Σ, a language L of finite words is a subset of Σ^*. A language L of infinite words over a finite alphabet Σ is a subset of Σ^ω. We restrict only to finite alphabets.

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 automata (*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*.

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 an 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 $C \subseteq Q$, $post(C, u) = \cup_{q \in C}\, post(q, u)$.

Intuitively, the PA starts in the initial state q_s and if after reading $a_0, a_1 \dots, a_i$ results in state q, then it 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 $i \geq 0$, $\delta_{\kappa[i]}(\rho[i], \rho[i+1]) > 0$.

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 $\{(q, v) \mid q \in Q, v \in \mathit{Pref}(\kappa)\}$ 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 [14, 11]. 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 three 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 [14]. and we shall denote its measure by $\mu^{acc,f}_{\mathcal{A},\,u}$. Note that $\mu^{acc,f}_{\mathcal{A},\,u} = \delta_u(q_s, Q_f)$. Given a rational threshold $x \in [0, 1]$ and $\rhd \in \{\geq, >\}$, the language of finite words $\mathsf{L^f}_{\rhd x}(\mathcal{A}) = \{u \in \Sigma^* \mid \mu^{acc,f}_{\mathcal{A},\,u} \rhd x\}$ is the set of finite words accepted by $\mathcal{A}$ with probability $\rhd x$.

Büchi acceptance: Büchi acceptance condition defines languages over infinite words. For Büchi acceptance, 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 Büchi automaton (PBA). Given a Büchi acceptance condition Q_f, a run ρ of $\mathcal{A}$ on an infinite word $\alpha \in \Sigma^\omega$ is said to be *accepting* if $\inf(\rho) \cap Q_f \neq \emptyset$. The set of accepting runs on $\alpha \in \Sigma^\omega$ is once again measurable [14] and we shall denote its measure by $\mu^{acc,b}_{\mathcal{A},\,\alpha}$. Given a rational threshold $x \in [0, 1]$ and $\rhd \in \{\geq, >\}$, the language of infinite words $\mathsf{L^b}_{\rhd x}(\mathcal{A}) = \{\alpha \in \Sigma^\omega \mid \mu^{acc,b}_{\mathcal{A},\,\alpha} \rhd x\}$ is the set of infinite words accepted by PBA $\mathcal{A}$ with probability $\rhd x$.

Muller acceptance: Muller acceptance condition, like Büchi acceptance condition, defines languages over infinite words. (Please see Appendix A for details). In this case,

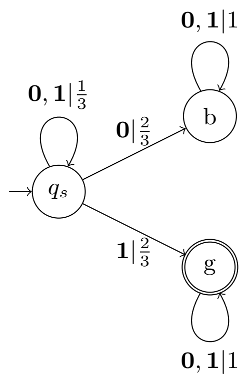

Fig. 1. PA $\mathcal{A}_{\mathsf{int}}$

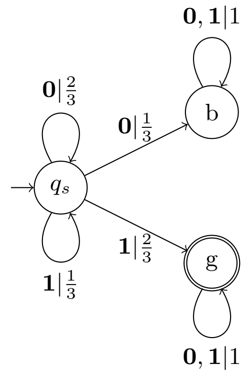

Fig. 2. PA $\mathcal{A}_{\frac{1}{3}}$

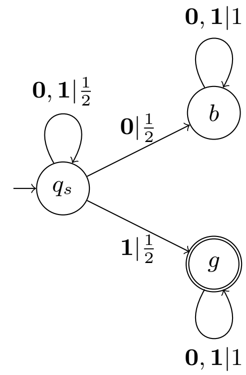

Fig. 3. PA $\mathcal{A}_{\text{Rabin}}$

we call the PA $\mathcal{A}$. probabilistic Muller automaton (PMA). Given a a rational threshold $x \in [0,1]$ and $\triangleright \in \{\geq, >\}$, the language of infinite words $\mathsf{L}^{\mathsf{m}}{}_{\triangleright x}(\mathcal{A}) = \{\alpha \in \Sigma^\omega \mid \mu^{acc,m}_{\mathcal{A},\,\alpha} \triangleright x\}$ is the set of infinite words accepted by PMA $\mathcal{A}$ with probability $\triangleright x$.

2.1 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, for each letter a, the machine can transition with non-zero probability to at most one state in the same level as q, and all other probabilistic successors belong to a higher level. We define such automata for the special case when the states are partitioned into two levels (level 0 and level 1).

Definition 2. A 1-*level 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 two sets Q_0 and Q_1 with the following properties.

- $q_s \in Q_0$,
- For every $q \in Q_0$ and $a \in \Sigma$, $|\mathsf{post}(q,a) \cap Q_0| \leq 1$
- For every $q \in Q_1$ and $a \in \Sigma$, $\mathsf{post}(q,a) \subseteq Q_1$ and $|\mathsf{post}(q,a)| = 1$.

Given a 1-level HPA $\mathcal{A}$, we will denote the level 0 and level 1 states by the sets Q_0 and Q_1 respectively.

Example 1. Consider the PAs $\mathcal{A}_{\mathsf{int}}$, $\mathcal{A}_{\frac{1}{3}}$, and $\mathcal{A}_{\mathsf{Rabin}}$ shown in Figs. 1, 2, and 3 respectively. All three automata have the same set of states ($\{q_s, \mathrm{b}, \mathrm{g}\}$), same initial state ($q_s$), same alphabet ($\{\mathbf{0}, \mathbf{1}\}$), the same acceptance condition ($Q_f = \{\mathrm{g}\}$ if finite/Büchi, and $F = \{\{\mathrm{g}\}\}$ if Muller) and the same transition structure. The only difference is in the probability of transitions out of q_s. All three of these automata are (1-level) HPAs; we can take $Q_0 = \{q_s\}$, and $Q_1 = \{\mathrm{g}, \mathrm{b}\}$. Though all three are very similar automata, we will show that $\mathcal{A}_{\mathsf{int}}$ and $\mathcal{A}_{\mathsf{Rabin}}$ are symptomatic of automata that accept only regular languages (with rational thresholds), while the other ($\mathcal{A}_{\frac{1}{3}}$) accepts non-regular languages (with rational thresholds). The automata $\mathcal{A}_{\mathsf{Rabin}}$ was originally presented in [13] and it is known to accept a non-regular language with an *irrational threshold* [13, 3]. Similarly it can be shown that $\mathcal{A}_{\mathsf{int}}$ also accepts a non-regular language with an irrational threshold.

Notation: For, the rest of the paper, by a HPA we shall mean 1-level HPA, unless otherwise stated.

Let us fix a HPA $\mathcal{A} = (Q, q_s, \delta, \mathsf{Acc})$ over alphabet Σ with Q_0 and Q_1 being the level 0 and level 1 states. 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 $i \in \{0, 1\}$) 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.

Proposition 1. *For any* HPA $\mathcal{A}$ *and witness set* W*, the language*

$$\mathsf{L}_W = \{\kappa \mid \kappa \textit{ is definitely accepted by } \mathcal{A} \textit{ from } W\}$$

is regular.

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

Proposition 2. *For any* HPA $\mathcal{A}$ *and witness set* W*, the problem of checking the emptiness of* L_W *(as defined in Proposition 1) is in* **PSPACE**.

Proof. Let $\mathcal{A}'$ be the (non-probabilistic) automata obtained by removing all transitions that either have probability < 1 or go between states belonging to different levels. Checking the emptiness of L_W can be reduced to checking the emptiness of the intersection of the languages of different copies of $\mathcal{A}'$ that have as start states different members of W. $\square$

For a set $C \subseteq Q_1$, a threshold $x \in (0, 1)$, and a word $u \in \Sigma^*$, we will find it useful to define the following quantity $\mathsf{val}(C, x, u)$ given as follows. If $\delta_u(q_s, Q_0) \neq 0$ then

$$\mathsf{val}(C, x, u) = \frac{x - \delta_u(q_s, C)}{\delta_u(q_s, Q_0)}$$

On the other hand, if $\delta_u(q_s, Q_0) = 0$ then

$$\mathsf{val}(C, x, u) = \begin{cases} +\infty \text{ if } \delta_u(q_s, C) < x \\ 0 \quad\ \text{ if } \delta_u(q_s, C) = x \\ -\infty \text{ if } \delta_u(q_s, C) > x \end{cases}$$

The quantity $\mathsf{val}(C, x, u)$ measures the fraction of $\delta_u(q_s, Q_0)$ that still needs to move to C such that the probability of reaching C exceeds the threshold x. This intuition is captured by the following proposition whose proof follows immediately from the definition of $\mathsf{val}(C, x, u)$.

Proposition 3. *Consider a* HPA $\mathcal{A}$ *with threshold* x*, and words* $u, v \in \Sigma^*$*. Let* $C, D \subseteq Q_1$ *such that* $\mathsf{post}(C, v) = D$*. The following properties hold.*

- *If* $\mathsf{val}(C, x, u) < 0$ *then* $\delta_{uv}(q_s, D) > x$*.*
- *If* $\mathsf{val}(C, x, u) = 0$ *then* $\delta_u(q_s, C) = x$*.*

Witness sets and the value function play an important role in deciding whether a word κ is accepted by a HPA. In particular, κ is accepted iff κ can be divided into strings u, κ' such that $\mathcal{A}$ reaches a witness set W with "sufficient probability" on u, and κ' is definitely accepted from W. We state this intuition precisely next.

Proposition 4. *For a* HPA $\mathcal{A}$*, threshold* $x \in [0, 1]$*, and word* κ*,* $\kappa \in \mathsf{L}^{\mathsf{a}}{}_{>x}(\mathcal{A})$ *(where* $\mathsf{a} \in \{\mathsf{f}, \mathsf{b}, \mathsf{m}\}$*) if and only if there is a 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 one of the following holds.*

- *Either* $W \subseteq Q_1$ *and* $\mathsf{val}(W, x, u) < 0$*, or*
- $W \cap Q_0 \neq \emptyset$ *and* $0 \leq \mathsf{val}(W \cap Q_1, x, u) < 1$*.*

3 Expressiveness and decidability

One-level HPAs have a very simple transition structure. Inspite of this, we will show that HPA can recognize non-regular languages (Section 3.1). Even though it has been shown before that PFAs [12, 13] and PBAs [10, 2] recognize non-regular languages, all the examples before, use either more complex automata or irrational acceptance thresholds or HPAs with more than two levels. We shall then show that even though HPAs can recognize non-regular languages, nevertheless the emptiness and universality problems of HPAs are decidable (Section 3.2).

3.1 Non-regular languages expressed by 1-level HPA

We will now show that HPA can recognize non-regular languages, under both finite acceptance and Büchi acceptance conditions. We consider a special type of HPA which we shall call *simple absorbing* HPA (SAHPA).

Definition 3. *Let* $\mathcal{A} = (Q, q_s, \delta, \mathsf{Acc})$ *be a* HPA *over an alphabet* Σ *with* Q_0 *and* Q_1 *as the sets of states at level* 0 *and* 1 *respectively.* $\mathcal{A}$ *is said to be a* simple absorbing HPA *(*SAHPA*) if*

- $Q_0 = \{q_s\}, Q_1 = \{\mathrm{g}, \mathrm{b}\}$.
- *The states* g, b *are absorbing, i.e., for each* $a \in \Sigma$, $\delta_a(\mathrm{g}, \mathrm{g}) = 1$ *and* $\delta_a(\mathrm{b}, \mathrm{b}) = 1$

For an $\kappa \in \Sigma^* \cup \Sigma^\omega$, $\mathrm{GoodRuns}(\kappa)$ *is the set of runs* ρ *of* $\mathcal{A}$ *on* κ *such there is an* $i \geq 0$ *with* $\rho(j) = \mathrm{g}$ *for all* $i \leq j \leq |\kappa|$. *A word* $\alpha \in \Sigma^\omega$ *is said to be* always alive *for* $\mathcal{A}$ *if for each* $i > 0$ $\delta_{\alpha[0:i]}(q_s, q_s) > 0$.

Example 2. All three automata $\mathcal{A}_{\mathsf{int}}$, $\mathcal{A}_{\frac{1}{3}}$ and $\mathcal{A}_{\mathsf{Rabin}}$ (Example 1) shown in Figs. 1, 2, and 3 are simple absorbing HPAs.

The following lemma states some important properties satisfied by SAHPA. (See Appendix B for a proof.)

Lemma 1. *Let* $\mathcal{A} = (Q, q_s, \delta, \mathsf{Acc})$ *be a* SAHPA *over an alphabet* Σ *with* Q_0 *and* Q_1 *as the sets of states at level* 0 *and* 1 *respectively. For any always alive* $\alpha \in \Sigma^\omega$,

1. *if* α *is ultimately periodic and* $\mu_{\mathcal{A},\alpha}(\mathrm{GoodRuns}(\alpha)) = x$ *then the set* $\{\mathsf{val}(\{\mathrm{g}\}, x, \alpha[0:i]) \mid i \in \mathbb{N}, i \geq 0\}$ *is a finite set,*
2. *if* $\lim_{i\to\infty} \delta_{\alpha[0:i]}(q_s, q_s) = 0$ *and* $x \in (0,1)$ *then* $\mu_{\mathcal{A},\alpha}(\mathrm{GoodRuns}(\alpha)) = x \Leftrightarrow \forall i \geq 0, \mathsf{val}(\{\mathrm{g}\}, x, \alpha[0:i]) \in [0,1]$.

Now, we shall show that SAHPA can recognize non-regular languages. We start by recalling a result originally proved in [13]. Let $\Sigma = \{\mathbf{0}, \mathbf{1}\}$. Any word $\kappa \in \Sigma^* \cup \Sigma^\omega$ can be thought of as the binary representation of a word in the unit interval $[0,1]$ by placing a decimal in front of it. Formally,

Definition 4. *Let* $\Sigma = \{\mathbf{0}, \mathbf{1}\}$. *The map* $\Sigma^* \cup \Sigma^\omega \to [0,1]$ *be the unique map such that* $\mathsf{bin}(\epsilon) = 0$ *and* $\mathsf{bin}(a\kappa_1) = \frac{\bar{a}}{2} + \frac{1}{4}\mathsf{bin}(\kappa_1)$, *where* $\bar{a} = 0$ *if* $a = \mathbf{0}$ *and* 1 *otherwise.*

Note that $\mathsf{bin}(\alpha)$ is irrational iff α is an infinite word which is not an ultimately periodic. The following is shown in [13].

Theorem 1. $\Sigma = \{\mathbf{0}, \mathbf{1}\}$ *and* $\alpha \in \Sigma^\omega$ *be a word which is not ultimately periodic. Given* $\rhd \in \{>, \geq\}$,

- $\{u \in \Sigma^* \mid \mathsf{bin}(u) \rhd \mathsf{bin}(\alpha)\}$ *is not regular.*
- $\{\gamma \in \Sigma^\omega \mid \mathsf{bin}(\gamma) \rhd \mathsf{bin}(\alpha)\}$ *is not* ω*-regular.*

We make some observations about the automaton $\mathcal{A}_{\frac{1}{3}}$ shown in Fig. 2 in Lemma 2. (See Appendix C for a proof).

Lemma 2. *Let* $\mathcal{A}_{\frac{1}{3}}$ *be the* SAHPA *over the alphabet* $\Sigma = \{\mathbf{0}, \mathbf{1}\}$ *defined in Example 1. Let* $\alpha \in \Sigma^\omega$ *be such that* α *is not an ultimately periodic word. We have that for each* $\kappa \in \Sigma^* \cup \Sigma^\omega$,

$$\mathsf{bin}(\kappa) < \mathsf{bin}(\alpha) \Leftrightarrow \mu_{\mathcal{A},\kappa}(\mathrm{GoodRuns}(\kappa)) < \mu_{\mathcal{A},\alpha}(\mathrm{GoodRuns}(\alpha))$$

and

$$\mathsf{bin}(\kappa) > \mathsf{bin}(\alpha) \Leftrightarrow \mu_{\mathcal{A},\kappa}(\mathrm{GoodRuns}(\kappa)) > \mu_{\mathcal{A},\alpha}(\mathrm{GoodRuns}(\alpha)).$$

We have:

Theorem 2. *Consider the* SAHPA $\mathcal{A}_{\frac{1}{3}}$ *over the alphabet* $\Sigma = \{\mathbf{0}, \mathbf{1}\}$ *defined in Example 1. Consider the finite acceptance condition and the Büchi acceptance condition defined by setting* $\mathsf{Acc} = \{\mathrm{g}\}$. *Given* $\rhd \in \{>, \geq\}$, *we have that the language of finite words* $\mathsf{L^f}_{\rhd\frac{1}{2}}(\mathcal{A})$ *is not regular and the language of infinite words* $\mathsf{L^b}_{\rhd\frac{1}{2}}(\mathcal{A})$ *is not* ω*-regular.*

Proof. Given $u \in \Sigma^*$, we shall denote $\mathsf{val}(\{\mathrm{g}\}, \frac{1}{2}, u)$ by val_u. We observe some properties of the value val_u.

Claim (A). For any $u \in \Sigma^*$,

- $\mathsf{val}_{u\mathbf{0}} = \frac{3}{2}\mathsf{val}_u$ and $\mathsf{val}_{u\mathbf{1}} = 3\mathsf{val}_u - 2$.
- If $\mathsf{val}_u \in [0, 1]$ then it is of the form $\frac{p}{2^i}$ where p is an odd number and $i - 1$ is the number of occurrences of $\mathbf{0}$ in u.
- $\mathsf{val}_u \notin \{0, 1, \frac{2}{3}\}$.

Proof. The first part of the claim follows from observing that $\delta_{u\mathbf{0}}(q_s, q_s) = \frac{2}{3}\delta_u(q_s, q_s)$, $\delta_{u\mathbf{0}}(q_s, \mathrm{g}) = \delta_u(q_s, good), \delta_{u\mathbf{1}}(q_s, q_s) = \frac{1}{3}\delta_u(q_s, q_s)$ and that $\delta_{u\mathbf{1}}(q_s, \mathrm{g}) = \delta_u(q_s, good)+ \delta(q_s, q_s)\frac{2}{3}$. The second part can be shown easily by an induction on the length of u using the first part of the claim. (Observe that the base case is $\mathsf{bin}(\epsilon) = \frac{1}{2}$). The third part of the claim is an easy consequence of the second part. (End: Proof of Claim (A)) $\square$

We now show that there is exactly one word $\beta \in \Sigma^\omega$ such that $\mu_{\mathcal{A},\beta}(\mathrm{GoodRuns}(\beta)) = \frac{1}{2}$. As each $\alpha \in \Sigma^\omega$ is always live and $\lim_{i\to\infty} \delta_{\alpha[0:i]}(q_s, q_s) = 0$, it follows from Lemma 1 and Claim (A) that it suffices to show that there is exactly one word $\beta \in \Sigma^\omega$ such that $\forall i \geq 0,\ \mathsf{val}_{\beta[0:i]} \in (0, 1)$.

We prove this by constructing β, starting from the empty word and showing that it can be extended one letter at a time in exactly one way. Clearly, thanks to Claim (A), since $\mathsf{val}_{\mathbf{0}} = \frac{3}{4}$ and $\mathsf{val}_{\mathbf{1}} = -\frac{1}{2}$, $\beta[0]$ should be $\mathbf{0}$. Suppose we have constructed $\beta[0 : i]$. Now, thanks to Claim (A) if $0 < \mathsf{val}_{\beta[0:i]} < \frac{2}{3}$ then $0 < \mathsf{val}_{\beta[0:i]\mathbf{0}} < \frac{3}{2}\frac{2}{3} = 1$ and $\mathsf{val}_{\beta[0:i]\mathbf{1}} < 3\frac{2}{3} - 2 < 0$. If $\frac{2}{3} < \mathsf{val}_{\beta[0:i]} < 1$ then $\mathsf{val}_{\beta[0:i]\mathbf{0}} > \frac{3}{2}\frac{2}{3} = 1$ and $0 = 3\frac{2}{3} - 2 < \mathsf{val}_{\beta[0:i]\mathbf{1}} < 3.1 - 2 = 1$. Thus if $\mathsf{val}_{\beta[0:i]} < \frac{2}{3}$ then $\beta[i + 1]$ has to be $\mathbf{0}$, otherwise $\beta[i + 1]$ has to be $\mathbf{1}$. Thus, we see that there is exactly one word $\beta \in \Sigma^\omega$ such that $\mu_{\mathcal{A},\beta}(\mathrm{GoodRuns}(\beta)) = \frac{1}{2}$. We shall now show that the values $\mathsf{val}_{\beta[0:i]}$ are all distinct.

Claim (B). For each i, j such that $i \neq j$, $\mathsf{val}_{\beta[0:i]} \neq \mathsf{val}_{\beta[0:j]}$.

Proof. Fix i, j. Without loss of generality, we can assume that $j > i$. Note that thanks to Claim (A) that if there is an occurrence of $\mathbf{0}$ in $\beta[i + 1 : j]$ then $\mathsf{val}_{\beta[0:i]} \neq \mathsf{val}_{\beta[0:j]}$. If there is no occurrence of $\mathbf{0}$ in $\beta[i + 1 : j]$ then every letter of $\beta[i + 1 : j]$ must be a $\mathbf{1}$. Thus, the result will follow if we can show that for each $i + 1 \leq k < j$, we have that $\mathsf{val}_{\beta[1:k]\mathbf{1}} < \mathsf{val}_{\beta[1:k]}$. Using Claim (A), we have that

$$\mathsf{val}_{\beta[1:k]\mathbf{1}} < \mathsf{val}_{\beta[1:k]} \Leftrightarrow 3\mathsf{val}_{\beta[1:k]} - 2 < \mathsf{val}_{\beta[1:k]} \Leftrightarrow \mathsf{val}_{\beta[1:k]} < 1.$$

Now $\mathsf{val}_{\beta[1:k]} < 1$ by construction of β. The claim follows. (End: Proof of Claim (B))

$\square$

Now, thanks to Lemma 1 and Claim (B), we have that β is not ultimately periodic. The result follows from Lemma 2 and Theorem 1. □

Remark 1. Note that since any Büchi acceptance condition can be converted into an equivalent Muller acceptance condition, HPAs also recognize non-regular languages under Muller acceptance conditions.

3.2 Decision problems for 1-level HPA

We now show that the problems of checking emptiness and universality for HPAs are decidable, more specifically, they are in **EXPTIME**. We start by considering emptiness for the language $\mathsf{L}^{a}_{>x}(\mathcal{A})$ for a HPA $\mathcal{A}$. In order to construct the decision procedure for this language, we need to consider special kinds of witness sets. We will say that a witness set W is *good* if the language L_W defined in Proposition 1 is non-empty. We have the following.

Proposition 5. *Give a* HPA $\mathcal{A} = (Q, q_s, \delta, \mathsf{Acc})$*, threshold* $x \in [0,1]$ *and* $\mathsf{a} \in \{\mathsf{f}, \mathsf{b}, \mathsf{m}\}$*, the language* $\mathsf{L^a}_{>x}(\mathcal{A}) \neq \emptyset$ *iff there is a word* $u \in \Sigma^*$ *and a good non-empty set* H *such that* $\delta_u(q_s, H) > x$*.*

Now, the decision procedure for checking emptiness will be to check for non-emptiness. For non-emptiness, the decision procedure shall search for a word u in Proposition 5. The following lemma shows that, it is enough to search for words of exponential length.

Lemma 3. *Let* $\mathcal{A} = (Q, q_s, \delta, \mathsf{Acc})$ *be an* HPA *with* n *states (i.e.,* $|Q| = n$*) such that all the transition probabilities of* $\mathcal{A}$ *have size at most* r [4]*. Let* $x \in [0,1]$ *be a rational threshold of size at most* r*. For any* $\mathsf{a} \in \{\mathsf{f}, \mathsf{b}, \mathsf{m}\}$*,* $\mathsf{L^a}_{>x}$ *iff there is a word* u *and a good non-empty set* H*, such that* $|u| \leq 4rn8^n$ *and* $\delta_u(q_s, H) > x$*.*

Proof. Observe that if there is a word u and a good non-empty set H such that $\delta_u(q_s, H) > x$ then by Proposition 5, $\mathsf{L^a}_{>x}(\mathcal{A}) \neq \emptyset$. Thus, we only need to prove that nonemptiness of $\mathsf{L^a}_{>x}(\mathcal{A})$ guarantees the existence of u and H as in the lemma.

Let $gwords = \{(s, G) \mid G \neq \emptyset, G$ is good and $\delta_s(q_s, G) > x\}$. By Proposition 5, $gwords$ is non-empty. Fix $(s, G) \in gwords$ such that for every $(s_1, G_1) \in gwords$, $|s| < |s_1|$, i.e., s is the shortest word appearing in a pair in $gwords$. Note if $|s| \leq 2^n$ then the lemma follows.

Let us consider the case when $|s| > 2^n$. Let $k_1 = |s| - 1$. Observe that by our notation, $s = s[0 : k_1]$. Now, for any $0 \leq i \leq k_1$, let $Y_i = \mathsf{post}(q_s, s[0 : i]) \cap Q_1$ and $X_i = \{q \in Y_i : \mathsf{post}(q, s[i+1 : k_1]) \subseteq G\}$. Note that $X_i \subseteq Y_i$ and is good. Since $|s| > 2^n$ and $\mathcal{A}$ has n states, there must be i, j with $i < j \leq k_1$ such $X_i = X_j$ and $\mathsf{post}(q_s, s[0 : i]) \cap Q_0 = \mathsf{post}(q_s, s[0 : j]) \cap Q_0$. If $\mathsf{post}(q_s, s[0 : i]) \cap Q_0 = \emptyset$ then it is easy to see that $(s[0 : i]s[j+1 : k_1], G) \in gwords$ contradicting the fact that s is the shortest such word. Hence, fix j to be the smallest integer such that for some

[4] We say a rational number s has size r iff there are integers m, n such that $s = \frac{m}{n}$ and the binary representation of m and n has at most r-bits.

$i < j$, $X_i = X_j$ and $\mathsf{post}(q_s, s[0:i]) \cap Q_0 = \mathsf{post}(q_s, s[0:j]) \cap Q_0 \neq \emptyset$. Let q be the unique state in $\mathsf{post}(q_s, s[0:i]) \cap Q_0$.

Let $s[0:i] = v, s[i+1:j] = w, s[j+1:k_1] = t$; thus, $s = vwt$. Now, let $z_1 = \delta_v(q_s, X_i)$ and $y_1 = \delta_v(q_s, Q_1)$. Similarly, let $z_2 = \delta_w(q, X_j), y_2 = \delta_w(q, Q_1)$ and $z_3 = \delta_t(q, G)$. Since $X_i, X_j \subseteq Q_1$, $z_1 \leq y_1$ and $z_2 \leq y_2$. Also note that $|w| > 0$ by construction of j and that $y_2 = \delta_w(q, Q_1) > 0$ (by the minimality of length of s).

For any integer $\ell \geq 0$, let $u_\ell = vw^\ell$ and $s_\ell = u_\ell t$. Note that $u_0 = v$ and $s_1 = s$. Let $\ell > 0$. We observe that

$$\delta_{s_\ell}(q_s, G) = \delta_{u_{(\ell-1)}}(q_s, X_i) + (1 - \delta_{u_{(\ell-1)}}(q_s, Q_1)) \cdot z_2 + (1 - \delta_{u_\ell}(q_s, Q_1)) \cdot z_3$$

and

$$\delta_{s_{(\ell-1)}}(q_s, G) = \delta_{u_{(\ell-1)}}(q_s, X_i) + (1 - \delta_{u_{(\ell-1)}}(q_s, Q_1)) \cdot z_3. \tag{1}$$

Therefore,

$$\begin{aligned}\delta_{s_\ell}(q_s, G) - \delta_{s_{(\ell-1)}}(q_s, G) = (1 - \delta_{u_{(\ell-1)}}(q_s, Q_1)) \cdot z_2 - \\ (\delta_{u_\ell}(q_s, Q_1) - \delta_{u_{(\ell-1)}}(q_s, Q_1)) \cdot z_3.\end{aligned}$$

In addition, $\delta_{u_\ell}(q_s, Q_1) = \delta_{u_{(\ell-1)}}(q_s, Q_1) + (1 - \delta_{u_{(\ell-1)}}(q_s, Q_1)) \cdot y_2$ and hence $\delta_{u_\ell}(q_s, Q_1) - \delta_{u_{(\ell-1)}}(q_s, Q_1) = (1 - \delta_{u_{(\ell-1)}}(q_s, Q_1)) \cdot y_2$ Putting all the above together, we get for all $\ell > 0$,

$$\delta_{s_\ell}(q_s, G) - \delta_{s_{(\ell-1)}}(q_s, G) = (1 - \delta_{u_{(\ell-1)}}(q_s, Q_1)) \cdot (z_2 - y_2 \cdot z_3).$$

Since $s = s_1$ is the shortest word in $gwords$ and $s_0 = vt$ is a strictly smaller word than s_1, we must have that $\delta_{s_0}(q_s, G) \leq x$ and hence $\delta_{s_1}(q_s, G) > \delta_{s_0}(q_s, G)$. From this and the above equality, we see that $(1 - \delta_{u_0}(q_s, Q_1)) > 0$ and that $(z_2 - y_2 \cdot z_3) > 0$. This also means that, for all $\ell > 0$, $\delta_{s_\ell}(q_s, G) \geq \delta_{s_{(\ell-1)}}(q_s, G)$. Hence, $\lim_{\ell\to\infty} \delta_{s_\ell}(q_s, G)$ exists and is $\geq \delta_{s_1}(q_s, G)$. Since $s_1 = s$, we get that $\lim_{\ell\to\infty} \delta_{s_\ell}(q_s, G) > x$.

Observe that $\delta_w(q, Q_1) > 0$. Hence, one can show that $\lim_{\ell\to\infty} (1 - \delta_{u_{(\ell-1)}}(q_s, Q_1)) = 0$. This along with Equation (1) means that $\lim_{\ell\to\infty} \delta_{s_\ell}(q_s, G) = \lim_{\ell\to\infty} \delta_{u_\ell}(q_s, X_i)$. The right hand side of this equation is seen to be $z_1 + (1 - y_1) \cdot \frac{z_2}{y_2}$ and since $\lim_{\ell\to\infty} \delta_{s_\ell}(q_s, G) > x$, we get that $z_1 + (1 - y_1) \cdot \frac{z_2}{y_2} > x$. Observe that X_i is a good set. Let m be the minimum ℓ such that $\delta_{u_\ell}(q_s, X_i) > x$. Now, we show that the length of u_m is bounded by $4rn8^n$ and hence the lemma is satisfied by taking u to be u_m and H to be X_i. Observe that

$$\delta_{u_\ell}(q_s, X_i) = z_1 + (1 - y_1) \cdot (1 - (1 - y_2)^\ell) \cdot \frac{z_2}{y_2}.$$

From this, we see that m is the minimum ℓ such that

$$(1 - y_2)^\ell < 1 - \frac{(x - z_1)y_2}{(1 - y_1)z_2}.$$

That is, m is the minimum ℓ such that $\ell > \frac{\log(n_1)}{\log(n_2)}$, where

$$n_1 = \tfrac{(1-y_1)z_2}{(1-y_1)z_2 - (x - z_1)y_2} \text{ and } n_2 = \tfrac{1}{(1-y_2)}.$$

Now, observe that the probability of a run ρ of $\mathcal{A}$ starting from any state, on an input string of length at most 2^n is a product of 2^n fractions of the form $\frac{m_1}{m_2}$ where m_i, for $i = 1, 2$, is an integer bounded by 2^r. Hence the probability of such a run is itself a fraction whose numerator and denominator are bounded by 2^{r2^n}. Second, in an HPA with n states, on any input of length k, there are at most kn different runs; this is because once the run reaches a state in Q_1 the future is deterministic, and for any prefix, there is at most one run in a state in Q_0. Hence, $\delta_v(q_s, Q_1)$ is the sum of at most $n2^n$ such fractions. Therefore, y_1 is a fraction whose numerator and denominator are integers bounded by 2^{rn4^n}. By a similar argument, we see that z_1, y_2, z_2 are also fractions whose numerators and denominators are similarly bounded. Now, it should be easy to see that n_1 is bounded by 2^{4rn4^n} and hence $m \leq 4rn4^n$. Now, the length of $u_m = |vw| + (m-1)|w|$ which is easily seen to be bounded $m2^n$ since $|vw|$ and $|w|$ are bounded by 2^n. Hence $u_m \leq 4rn8^n$. □

Now, we have the following theorem.

Theorem 3. *Given a* HPA $\mathcal{A} = (Q, q_s, \delta, \mathsf{Acc})$*, a rational threshold* $x \in [0, 1]$ *and* $\mathsf{a} \in \{\mathsf{f}, \mathsf{b}, \mathsf{m}\}$*, the problem of determining if* $\mathsf{L^a}_{>x}(\mathcal{A}) = \emptyset$ *is in* **EXPTIME**.

Proof. It suffices to show that the problem of determining if $\mathsf{L^a}_{>x}(\mathcal{A}) \neq \emptyset$. Let $\mathcal{X}$ be the collection of all witness sets U such that $U \cap Q_0 \neq \emptyset$ and $U \cap Q_1$ is a good set; for a witness set $U \in \mathcal{X}$, we will denote by q_U the unique state in $U \cap Q_0$. Let $\mathcal{Y}$ be the collection of good witness sets. For a natural number $i > 0$, let

$$\mathsf{Prob}(U, i) = \max\{\delta_u(q_U, W) \mid u \in \Sigma^*,\ W \in \mathcal{Y},\ \mathsf{post}(U \cap Q_1, u) \subseteq W,\ |u| \leq i\}.$$

In the above definition, we take the maximum of the empty set to be 0. Let k be the bound given by Lemma 3 for the length of the word u. Lemma 3 implies that $\mathsf{L^a}_{>x}(\mathcal{A}) \neq \emptyset$ iff $\mathsf{Prob}(\{q_s\}, k) > x$. This observation yields a simple algorithm to check non-emptiness: compute $\mathsf{Prob}(\{q_s\}, k)$ and check if it is greater than x.

$\mathsf{Prob}(\cdot, \cdot)$ can be computed by an iterative dynamic programming algorithm as follows.

$$\begin{aligned}
\mathsf{Prob}(U, 1) &= \max\{\delta_a(q_U, W) \mid a \in \Sigma,\ W \in \mathcal{Y},\ \mathsf{post}(U \cap Q_1, a) \subseteq W\} \\
\mathsf{Prob}(U, i+1) &= \max\big(\{\mathsf{Prob}(U, i)\} \bigcup \\
&\qquad \{\delta_a(q_U, q_V)\mathsf{Prob}(V, i) + \delta_a(q_U, V \cap Q_1) \mid a \in \Sigma,\ V \in \mathcal{X}, \\
&\qquad\qquad \mathsf{post}(U \cap Q_1, a) \subseteq V\}\big)
\end{aligned}$$

Let us analyze the algorithm computing $\mathsf{Prob}(\cdot, \cdot)$. Let us assume that $\mathcal{A}$ has n states, and that $\delta_a(p, q)$ is of size at most r for any $a \in \Sigma$ and $p, q \in Q$. Thus, $\mathcal{X}$ and $\mathcal{Y}$ have cardinality at most 2^n, and by Proposition 2, the sets $\mathcal{X}$ and $\mathcal{Y}$ can be computed in **EXPTIME** (in fact, even in **PSPACE**). In addition, because $|\mathcal{X}|,\ |\mathcal{Y}| \leq 2^n$, the maximum in the above equations for computing Prob is over at most $O(2^n)$ terms. Thus, we would get an exponential time bound provided the arithmetic operations needed to compute Prob can also be carried out in exponential time. This requires us to bound the size of the numbers involved in computing $\mathsf{Prob}(U, i)$. Observe that for any witness set W and $q \in Q$, $\delta_a(q, W)$ is the sum of at most n rational numbers and so has size at most $r + n$. Hence, we can inductively show that the size of $\mathsf{Prob}(U, i)$ (for any U) is a rational number of size at most $2i(r + n)$. Since $i \leq k$ and k is at most exponential in n (by Lemma 3), the dynamic programming algorithm is in **EXPTIME**. □

The emptiness problem for the languages $\mathsf{L^a}_{\geq x}(\mathcal{A})$ can be shown to be decidable using similar methods (See Appendix D for a proof).

Theorem 4. *Given a* HPA $\mathcal{A}$*, a rational threshold* $x \in [0,1]$ *and* $\mathsf{a} \in \{\mathsf{f},\mathsf{b},\mathsf{m}\}$*, the problem of determining if* $\mathsf{L^a}_{\geq x}(\mathcal{A}) = \emptyset$ *is in* **EXPTIME***.*

Now, we give the following lower bound results for checking non-emptiness of the languages $\mathsf{L^a}_{\rhd x}(\mathcal{A}) \neq \emptyset$ for $\rhd \in \{>, \geq\}$ (see Appendix E for a proof).

Theorem 5. *Given a* HPA $\mathcal{A}$*,* $\mathsf{a} \in \{\mathsf{f},\mathsf{b},\mathsf{m}\}$*,* $\rhd \in \{>, \geq\}$*, the problem of determining if* $\mathsf{L^a}_{\rhd x}(\mathcal{A}) \neq \emptyset$ *is* **PSPACE***-hard.*

Now, Theorem 3 and Theorem 4 yield that checking non-universality is also decidable (see Appendix F for a proof).

Theorem 6. *Given a* HPA $\mathcal{A}$*,* $\mathsf{a} \in \{\mathsf{f},\mathsf{b},\mathsf{m}\}$*,* $\rhd \in \{>, \geq\}$*, the problem of checking universality of the language* $\mathsf{L^a}_{\rhd x}(\mathcal{A})$ *is in* **EXPTIME** *and is* **PSPACE***-hard.*

4 Integer HPAs

In the previous section we saw that even though (1-level) HPAs have a very simple transition structure, their ability to toss coins allows them to recognize non-regular languages. In this section, we will show that if we restrict the numbers that appear as transition probabilities in the automaton, then the HPA can only recognize regular languages (see Theorem 7). We will also show that the problems of checking emptiness and universality of this class of HPAs are **PSPACE**-complete (see Theorem 8). We will call this restricted class of HPAs, integer HPAs.

Definition 5. *An* integer HPA *is a (1-level)* HPA $\mathcal{A} = (Q, q_s, \delta, \mathsf{Acc})$ *over alphabet* Σ *with* Q_0 *and* Q_1 *being the level 0 and level 1 states, respectively, such that for every* $q \in Q_0$ *and* $a \in \Sigma$*, if* $\mathsf{post}(q,a) \cap Q_0$ *is non-empty and equal to* $\{q'\}$*, then for every* $q'' \in Q_1$*,* $\delta_a(q,q'')$ *is an integer multiple of* $\delta_a(q,q')$*.*

Example 3. Consider automata $\mathcal{A}_{\mathsf{int}}$, $\mathcal{A}_{\frac{1}{3}}$, and $\mathcal{A}_{\mathsf{Rabin}}$ from Example 1 that are shown in Figs. 1, 2, and 3. Observe that $\mathcal{A}_{\mathsf{int}}$ and $\mathcal{A}_{\mathsf{Rabin}}$ are integer automata. On the other hand, $\mathcal{A}_{\frac{1}{3}}$, which was shown to accept non-regular languages in Section 3.1, is not an integer automaton. The reason is because of the transition from q_s on symbol **0**; $\delta_{\mathbf{0}}(q_s, \mathrm{b}) = \frac{1}{3}$ is not an integer multiple of $\delta_{\mathbf{0}}(q_s, q_s) = \frac{2}{3}$.

The main result of this section is that for any integer HPA $\mathcal{A}$, and rational x, the language $\mathsf{L^a}_{>x}(\mathcal{A})$ is regular (for $\mathsf{a} \in \{\mathsf{f},\mathsf{b},\mathsf{m}\}$). The proof of this result will rest on observations made in Proposition 4 that states that a word κ is accepted exactly when a prefix of κ reaches a witness set with sufficient probability, and the rest of the word κ is definitely accepted from the witness set. Proposition 1 states that the words definitely accepted from any witness set is regular. Thus, the crux of the proof will be to show that there is a way to maintain the $\mathsf{val}(\cdot, x, \cdot)$ function for each witness set using only finite memory. This observation will rest on a few special properties of integer HPAs. (See Appendix G for a proof).

Proposition 6. *Let $\mathcal{A}$ be an integer* HPA *over alphabet Σ with level 0 and level 1 sets Q_0 and Q_1, $C \subseteq Q_1$, and x be a rational number $\frac{c}{d}$. For any $u \in \Sigma^*$, if* $\mathsf{val}(C, x, u) \in [0, 1]$ *then there is $e \in \{0, 1, 2, \ldots d\}$ such that* $\mathsf{val}(C, x, u) = \frac{e}{d}$.

The above proposition makes a very important observation — the set of relevant values that the function val can take are finite. Proposition 3 in Section 2.1 essentially says that when the function val takes on values either below 0 or above 1, either all extensions of the current input will have sufficient probability among witness sets in Q_1 or no extension will have sufficient probability. Thus, when measuring the quantity val what matters is only whether it is strictly less than 0, strictly greater than 1 or its exact value when it is in $[0, 1]$. Proposition 6 above, guarantees that val is finite when it lies within $[0, 1]$. This allows us to keep track of val using finite memory. This is captured in the following Lemma (see Appendix H for the proof.)

Lemma 4. *Consider an integer* HPA $\mathcal{A}$ *over alphabet Σ with Q_0 and Q_1 as level 0 and level 1 states. Let $x = \frac{c}{d}$ be a rational threshold. For an arbitrary $C \subseteq Q_1$, $q \in Q_0$, and $e \in \{0, 1, \ldots d\}$, the following six languages*

$$
\begin{aligned}
\mathsf{L}_{(q,C,e)} &= \{u \in \Sigma^* \mid \mathsf{post}(q_s, u) \cap Q_0 = \{q\} \textit{ and } \mathsf{val}(C, x, u) \le \tfrac{e}{d}\}\\
\mathsf{L}_{(q,C,-)} &= \{u \in \Sigma^* \mid \mathsf{post}(q_s, u) \cap Q_0 = \{q\} \textit{ and } \mathsf{val}(C, x, u) < 0\}\\
\mathsf{L}_{(q,C,+)} &= \{u \in \Sigma^* \mid \mathsf{post}(q_s, u) \cap Q_0 = \{q\} \textit{ and } \mathsf{val}(C, x, u) > 1\}\\
\mathsf{L}_{(*,C,e)} &= \{u \in \Sigma^* \mid \mathsf{post}(q_s, u) \cap Q_0 = \emptyset \textit{ and } \mathsf{val}(C, x, u) \le \tfrac{e}{d}\}\\
\mathsf{L}_{(*,C,-)} &= \{u \in \Sigma^* \mid \mathsf{post}(q_s, u) \cap Q_0 = \emptyset \textit{ and } \mathsf{val}(C, x, u) < 0\}\\
\mathsf{L}_{(*,C,+)} &= \{u \in \Sigma^* \mid \mathsf{post}(q_s, u) \cap Q_0 = \emptyset \textit{ and } \mathsf{val}(C, x, u) > 1\}
\end{aligned}
$$

are all regular.

We are ready to present the main result of this section.

Theorem 7. *For any integer* HPA $\mathcal{A}$, *rational threshold $x \in [0, 1]$, the languages* $\mathsf{L}^{\mathsf{a}}{}_{>x}(\mathcal{A})$ *and* $\mathsf{L}^{\mathsf{a}}{}_{\ge x}(\mathcal{A})$ *are regular (where* $\mathsf{a} \in \{\mathsf{f}, \mathsf{b}, \mathsf{m}\}$*).*

Proof. From Proposition 4, we can conclude that

$$
\mathsf{L}^{\mathsf{a}}{}_{>x}(\mathcal{A}) = \left(\bigcup_{C \subseteq Q_1,\ q \in Q_0 \cup \{*\}} \mathsf{L}_{(q,C,-)} \mathsf{L}_C \right) \cup \left(\bigcup_{C \subseteq Q_1,\ q \in Q_0, e \in [0,1)} \mathsf{L}_{(q,C,e)} \mathsf{L}_{C \cup \{q\}} \right)
$$

where L_W is the set of words definitely accepted from witness set W, as defined in Proposition 1. From Proposition 1 and Lemma 4, we can conclude that each of the languages on the right hand side is regular, and therefore, $\mathsf{L}^{\mathsf{a}}{}_{>x}(\mathcal{A})$ is regular.

The proof of regularity of $\mathsf{L}^{\mathsf{a}}{}_{\ge x}(\mathcal{A})$ is carried out in Appendix I. $\square$

The following theorem shows that the problems of checking emptiness and universality are **PSPACE**-complete for integer HPAs, thus giving a tight upper bound; the proof is in Appendix J.

Theorem 8. *Given an integer* HPA $\mathcal{A}$, $\mathsf{a} \in \{\mathsf{f}, \mathsf{b}, \mathsf{m}\}$, $\rhd \in \{>, \ge\}$, *the problem of determining if* $\mathsf{L}^{\mathsf{a}}{}_{\rhd x}(\mathcal{A}) = \emptyset$ *is* **PSPACE***-complete. Similarly, the problem of checking universality is also* **PSPACE***-complete.*

5 Conclusions

We investigated the expressiveness of (1-level) HPAs with non-extremal thresholds and showed, in spite of their very simple transition structure, they can recognize non-regular languages. Nevertheless, the canonical decision problems of emptiness and universality for HPAs turn out to be decidable in **EXPTIME** and are **PSPACE**-hard. Imposing a very simple restriction on the transition probabilities result in automata that we call integer HPAs which recognize only regular languages. For integer HPAs, the canonical decision problems turn out to be **PSPACE**-complete.

There are a few problems left open by our investigations. The first one is of course the gap in the complexity of deciding emptiness and universality for these problems. Our investigations in this paper were motivated by understanding the relationship between the number of levels in HPAs and the tractability of the model. The results in [4] suggest that problems become hard for 6-level HPAs and non-extremal thresholds. Our results here suggest that 1-level HPAs (with non-extremal thresholds) are tractable. Exactly where the boundary between decidability and undecidability lies is still open. Finally, as argued in the Introduction, HPAs arise naturally as models of client-server systems, and it would useful to apply the theoretical results here to such models.

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. C. Baier, M. Größer, and N. Bertrand. Probabilistic ω-automata. *Journal of the ACM*, 59(1):1–52, 2012.
3. 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.
4. 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.
5. 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.
6. K. Chatterjee and T. A. Henzinger. Probabilistic automata on infinite words: Decidability and undecidability results. In *Intl. Symp. on Automated Technology for Verification and Analysis*, pages 1–16, 2010.
7. 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.
8. 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.
9. 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.
10. M. Größer. *Reduction Methods for Probabilistic Model Checking*. PhD thesis, TU Dresden, 2008.
11. J. Kemeny and J. Snell. *Denumerable Markov Chains*. Springer-Verlag, 1976.
12. A. Paz. *Introduction to Probabilistic Automata*. Academic Press, 1971.
13. M. O. Rabin. Probabilistic automata. *Inf. and Control*, 6(3):230–245, 1963.
14. M. Vardi. Automatic verification of probabilistic concurrent finite-state programs. In *Symp. on Foundations of Computer Science*, pages 327–338, 1985.

A Probabilistic Muller Automata

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 are measurable [14]. Given a word α, the measure of the set of accepting runs is denoted by $\mu^{acc,m}_{\mathcal{A},\,\alpha}$. Given a a threshold $x \in [0,1]$ and $\triangleright \in \{\geq, >\}$, the language of infinite words $\mathsf{L^m}_{\triangleright x}(\mathcal{A}) = \{\alpha \in \Sigma^\omega \,|\, \mu^{acc,m}_{\mathcal{A},\,\alpha} \triangleright x\}$ is the set of infinite words accepted by PMA $\mathcal{A}$ with probability $\triangleright x$.

B Proof of Lemma 1

We prove each of the following.

1. Consider any always alive $\alpha \in \Sigma^\omega$ such that $\mu_{\mathcal{A},\alpha}(\mathrm{GoodRuns}(\alpha)) = x$. Consider a prefix $u \in \Sigma^*$ and $\gamma \in \Sigma^\omega$ such that $\alpha = u\gamma$. Now, it is easy to see that
$$\mu_{\mathcal{A},\alpha}(\mathrm{GoodRuns}(\alpha)) = \delta_u(q_s, \mathrm{g}) + \delta_u(q_s, q_s)\mu_{\mathcal{A},\gamma}(\mathrm{GoodRuns}(\gamma)).$$
Hence
$$\mu_{\mathcal{A},\gamma}(\mathrm{GoodRuns}(\gamma)) = \frac{\mu_{\mathcal{A},\alpha}(\mathrm{GoodRuns}(\alpha)) - \delta_u(q_s,\mathrm{g})}{\delta_u(q_s,q_s)}.$$
As $\mu_{\mathcal{A},\alpha}(\mathrm{GoodRuns}(\alpha)) = x$, we get that
$$\mu_{\mathcal{A},\gamma}(\mathrm{GoodRuns}(\gamma)) = \mathsf{val}(\{\mathrm{g}\}, u, x).$$
Now, if α is ultimately periodic then the number of its suffixes is finite; (if it is periodic with period p starting from $n > 0$, then for all $i \geq n$, the suffix of α, starting from the i^{th} symbol is same as it's suffix starting from the $(i{+}p)^{th}$ symbol). The result follows.
2. ($\Rightarrow$) For each $i > 0$ let us write $\alpha[0 : i]$ as u_i and $\alpha[i+1 : \infty]$ as γ_i. If $\mu_{\mathcal{A},\alpha}(\mathrm{GoodRuns}(\alpha)) = x$ then as in above part we have that for each i
$$\mu_{\mathcal{A},\gamma_i}(\mathrm{GoodRuns}(\gamma_i)) = \mathsf{val}(\{\mathrm{g}\}, x, u_i).$$
Thus, we get $\mathsf{val}(\{\mathrm{g}\}, x, u_i) \in [0,1]$ for each i.
($\Leftarrow$) Now assume that $\mathsf{val}(\{\mathrm{g}\}, x, \alpha[0 : i])) \in [0,1]$ for all i. Note that since $\lim_{i\to\infty} \delta_{u_i}(q_s,q_s) = 0$, we get that $\mu_{\mathcal{A},\alpha}(\mathrm{GoodRuns}(\alpha)) = \lim_{i\to\infty} \delta_{u_i}(q_s,\mathrm{g})$. Let $y = \mu_{\mathcal{A},\alpha}(\mathrm{GoodRuns}(\alpha))$. Now, if $y \neq x$ then we will get that $\lim_{i\to\infty}(x - \delta_{u_i}(q_s,\mathrm{g})) = x - y \neq 0$. Now, observe that
$$\lim_{i\to\infty} \mathsf{val}(\{\mathrm{g}\}, x, u_i) = \lim_{i\to\infty} \frac{(x - \delta_{u_i}(q_s,\mathrm{g}))}{\delta_{u_i}(q_s,q_s)}.$$
Since $\lim_{i\to\infty}(x - \delta_{u_i}(q_s,\mathrm{g})) \neq 0$ and $\lim_{i\to\infty} \delta_{u_i}(q_s,q_s) = 0$ we must get that there is an i such that $\mathsf{val}(\{\mathrm{g}\}, x, u_i) > 1$ or $\mathsf{val}(\{\mathrm{g}\}, x, u_i) < 0$, contradicting our assumption that $\mathsf{val}(\{\mathrm{g}\}, x, \alpha[0 : i])) \in [0,1]$ for all i.

C Proof of Lemma 2

Fix κ such that $\kappa \neq \alpha$. Note that if κ is a prefix of α, the lemma follows immediately from construction of $\mathcal{A}_{\frac{1}{3}}$. Assume that κ is not a prefix of α. In this case there must be a word $u \in \Sigma^*$, letters $a, b \in \Sigma, \kappa_1 \in \Sigma^* \cup \Sigma^\omega, \alpha_1 \in \Sigma^\omega$ such that $a \neq b, \kappa = ua\kappa_1$ and $\alpha = ub\alpha_1$.

Note that α_1 contains infinite occurrences of $\mathbf{0}$ and $\mathbf{1}$ (as α is not ultimately periodic). Using this fact, we can show that $0 < \mu_{\mathcal{A},\alpha_1}(\mathrm{GoodRuns}(\alpha_1)) < 1$. Also, note that we have by construction of $\mathcal{A}_{\frac{1}{3}}$, $\delta_u(q_s, q_s) > 0$.

Assume first $a = \mathbf{0}$ and $b = \mathbf{1}$. Then we will have $\mathsf{bin}(\kappa) < \mathsf{bin}(\alpha)$. It is easy to see that

$$\begin{aligned}\mu_{\mathcal{A},\kappa}(\mathrm{GoodRuns}(\kappa)) &= \delta_u(q_s, \mathrm{g}) + \tfrac{2}{3}\delta_u(q_s, q_s)\mu_{\mathcal{A},\kappa_1}(\mathrm{GoodRuns}(\kappa_1)) \\ &\leq \delta_u(q_s, \mathrm{g}) + \tfrac{2}{3}\delta_u(q_s, q_s)\end{aligned}$$

and

$$\begin{aligned}\mu_{\mathcal{A},\alpha}(\mathrm{GoodRuns}(\alpha)) &= \delta_u(q_s, \mathrm{g}) + \tfrac{2}{3}\delta_u(q_s, q_s) + \\ &\qquad\qquad \tfrac{1}{3}\delta_u(q_s, q_s)\mu_{\mathcal{A},\alpha_1}(\mathrm{GoodRuns}(\alpha_1)) \\ &> \delta_u(q_s, \mathrm{g}) + \tfrac{2}{3}\delta_u(q_s, q_s) \\ &> \mu_{\mathcal{A},\kappa}(\mathrm{GoodRuns}(\kappa)).\end{aligned}$$

Similarly, we can show that if $a = \mathbf{1}$ and $b = \mathbf{0}$ we have that $\mathsf{bin}(\kappa) > \mathsf{bin}(\alpha)$ and $\mu_{\mathcal{A},\kappa}(\mathrm{GoodRuns}(\kappa)) > \mu_{\mathcal{A},\alpha}(\mathrm{GoodRuns}(\alpha))$. The result follows.

D Proof of Theorem 4

Given a HPA $\mathcal{A} = (Q, q_s, \delta, \mathsf{Acc})$, a threshold $x \in [0, 1]$ and $\mathsf{a} \in \{\mathsf{f}, \mathsf{b}, \mathsf{m}\}$, we consider the problem of checking if the language $\mathsf{L^a}_{\geq x}(\mathcal{A}) \neq \emptyset$. We have the following lemma 6 that gives a necessary and sufficient conditions for checking the above problem. We also need the technical lemma 5 given below.

Lemma 5. *Consider an* HPA $\mathcal{A} = (Q, q_s, \delta, \mathsf{Acc})$ *over an alphabet* Σ*, a threshold* $x \in [0, 1]$ *and* $\mathsf{a} \in \{\mathsf{f}, \mathsf{b}, \mathsf{m}\}$ *such that* $\mathsf{L^a}_{>x}(\mathcal{A}) = \emptyset$ *and* $\kappa \in \mathsf{L^a}_{\geq x}(\mathcal{A})$*. For* $k \leq |\kappa|$*, let* $X_k \subseteq \mathsf{post}(q_s, \kappa[0 : k])$ *be the set of all states from which* $\kappa[k+1 : |\kappa|]$ *is definitely accepted. If, in addition,* κ *is such that there are integers* $i < j$ *such that* $\mathsf{post}(q_s, \kappa[0 : i]) \cap Q_0 = \mathsf{post}(q_s, \kappa[0 : j]) \cap Q_0$*,* $X_i \neq \emptyset$ *and* $X_i = X_j$ *then for every* $\ell \geq 0$*,* $\mu^{acc,a}_{\mathcal{A},\,\kappa(\ell)} = x$*, where* $\kappa(\ell) = \kappa[0 : i](\kappa[i+1 : j])^\ell \kappa[j+1 : |\kappa|]$*.*

Proof. Assume κ, i, j be as given in the statement of the lemma. Take $\ell = 0$. Then $\mu^{acc,a}_{\mathcal{A},\,\kappa(0)} \leq \mu^{acc,a}_{\mathcal{A},\,\kappa}$ since $\mathsf{L^a}_{>x}(\mathcal{A}) = \emptyset$. If $\mu^{acc,a}_{\mathcal{A},\,\kappa(0)} < \mu^{acc,a}_{\mathcal{A},\,\kappa}$ then using the argument given in the proof of lemma 3, we see that $\mu^{acc,a}_{\mathcal{A},\,\kappa(2)} > \mu^{acc,a}_{\mathcal{A},\,\kappa}$, which contradicts the assumption that $\mathsf{L^a}_{>x}(\mathcal{A}) = \emptyset$. □

Lemma 6. *Let* $\mathcal{A} = (Q, q_s, \delta, \mathsf{Acc})$ *be a* HPA *over alphabet* Σ *such that* $|Q| = n$*. For a threshold* $x \in [0, 1]$ *and* $\mathsf{a} \in \{\mathsf{f}, \mathsf{b}, \mathsf{m}\}$*, the language* $\mathsf{L^a}_{\geq x}(\mathcal{A}) \neq \emptyset$ *iff either* $\mathsf{L^a}_{>x}(\mathcal{A}) \neq \emptyset$ *or at least one of the following two conditions is satisfied:*

1. $\exists u \in \Sigma^*$ *such that* $|u| \leq 2^n$ *and there is a good set* G *such that* $\delta_u(q_s, G) = x$.
2. $\mathsf{a} \in \{\mathsf{b}, \mathsf{m}\}$, $\exists u, v \in \Sigma^*$ *such that* $|u| \leq 2^n$, $|v| \leq 2^{(n+2)\log n}$ *and* $\alpha = uv^\omega$ *is accepted by* $\mathcal{A}$ *with probability exactly* x, *i.e.,* $\mu^{acc,a}_{\mathcal{A},\,\alpha} = x$.

Proof. Without loss of generality, w evan assume that $x > 0$ (otherwise, the lemma is trivially true). We prove the lemma in only one direction, as the other direction is trivial. Assume that $\mathsf{L^a}_{\geq x}(\mathcal{A}) \neq \emptyset$ and $\mathsf{L^a}_{>x}(\mathcal{A}) = \emptyset$. Now, we show that either (1) or (2) is satisfied. Consider the case when $\mathsf{a} = \mathsf{f}$. Clearly $\exists u \in \Sigma^*$ and a good non-empty set G such that $\delta_u(q_s, G) = x$. Let u be the shortest such sequence. Now, if $|u| > 2^n$, then this would imply that there exist integers i, j such that $i < j$, $\mathsf{post}(q_s, u[0:i]) \cap Q_0 = \mathsf{post}(q_s, u[0:j]) \cap Q_0$, $X_i \neq \emptyset$ and $X_i = X_j$ where X_i, X_j are as defined in the statement of Lemma 5, and using Lemma 5 with $\ell = 0$, we get a shorter sequence satisfying the above property, a contradiction. Hence we see that condition (1) of the lemma is satisfied. Now consider the case when $\mathsf{a} = \mathsf{b}$ or $\mathsf{a} = \mathsf{m}$, i.e., $\mathcal{A}$ is a Büchi automaton or a Muller automaton. Now, if there is a finite sequence u and good set G such that $\delta_u(q_s, G) = x$, then the above argument also applies here and condition (1) is satisfied. Now, consider the case when there is an $\alpha \in \Sigma^\omega$ such that $\mu^{acc,a}_{\mathcal{A},\,\alpha} = x$. Let L_1 be the set of all such α. Now, consider any $\alpha \in L_1$. Then there exist integers i, j such that $i < j$, $\mathsf{post}(q_s, u) \cap Q_0 = \mathsf{post}(q_s, uv) \cap Q_0$, $X_i \neq \emptyset$ and $X_i = X_j$ where $u = \alpha[0:i]$, $v = \alpha[i{+}1:j]$, X_i, X_j are as defined earlier and the following conditions are satisfied for every $q \in X_i \cup (post(q_s, u) \cap Q_0)$: (a) $\mathsf{post}(q, v) = \{q\}$, (b) if $q \in X_i$ then the set Y_q of all states that appear on the finite run from q on the string v satisfies the acceptance condition given by Acc, i.e., $Y_q \cap \mathsf{Acc} \neq \emptyset$ if $\mathsf{a} = \mathsf{b}$ and $Y_q \in \mathsf{Acc}$ if $\mathsf{a} = \mathsf{m}$. Using Lemma 5, we see that for each $\ell \geq 0$, $\alpha_\ell = uv^\ell\alpha[j+1:\infty]$ is accepted with probability x. From this, it can be shown that $\mu^{acc,a}_{\mathcal{A},\,\beta} = x$ where $\beta = uv^\omega$. Now, let M be the set of all triples $(u, v, X_i \cup (post(q_s, u) \cap Q_0))$ where u, v, X_i are as defined above for some $\alpha \in L_1$.

Now let (u, v, U) be a triple in M such that $|u| + |v|$ is the minimum for all such triples in M and $|u|$ be the minimum among all triples having this minimum value. It is not difficult to show that $|u| \leq 2^n$. Now, we prove that $|v| \leq n^{n+2} = 2^{(n+2)\log n}$ through contradiction. Contrary to this assume that $|v| > n^{n+2}$. Let $U = \{q_1, q_2, ..., q_m\}$ for some $m \leq n$. Let ρ_p, $1 \leq p \leq m$, be the run of $\mathcal{A}$ on the input v starting from the state q_p; observe that length of ρ_p is $|v| + 1$ and $\rho_p[0] = q_p$. For any k,$0 \leq k \leq |v|$, let w_k be the vector $(\rho_1[k], ..., \rho_m[k])$. Now, for any $r = (r_1, ..., r_m) \in Q^m$, let $I_r = \{k \mid w_k = r, 0 \leq k < |v|\}$. Since $|v| > n^{n+2}$ and the cardinality of Q^m is bounded by n^n (since $m \leq n$), it follows that there exists an $r \in Q^m$, such that I_r has more than n^2 elements in it. Fix such an $r \in Q^m$. Let $i_1 < i_2 < ... < i_k$ be all the elements in I_r. Since $k > n^2$, it is easy to see that there exists j', $1 \leq j' < k$, such that for every p, $1 \leq p \leq m$, every state that appears in the sequence of states $\rho_p[i_{j'}], \rho_p[i_{j'} + 1], ..., \rho_p[i_{j'+1} - 1]$ also appears some where else, outside this subsequence, in the same run ρ_p. Now let $k' = i_{j'}$, $k'' = i_{j'+1}$ and $v' = (v_0, v_1, ..., v_{k'-1}, v_{k''}, v_{k''+1}, ..., v_{|v|-1})$, i.e., v' is obtained by removing the subsequence $(v_{k'}, ..., v_{k''-1})$ from v. Observe that, for each $p = 1, ..., m$, the run of $\mathcal{A}$ starting from state q_p on the input v' has the same set of states appearing in it as the run ρ_p. Using this and Lemma 5 repeatedly, it can be easily seen that the triple

$(u, v', U) \in M$ which contradicts the condition that $|u| + |v|$ is the minimum among all triples in M. $\square$

We now give an **EXPTIME** algorithm that checks if $\mathsf{L^a}_{\geq x}(\mathcal{A}) \neq \emptyset$. First observe that if $\mathsf{a} = \mathsf{f}$ then the proof of Lemma 3 and Lemma 6 shows that $\mathsf{L^a}_{\geq x}(\mathcal{A}) \neq \emptyset$ iff there exists a good set G and finite word u such that $|u| \leq 2^n$ and $\delta_u(q_s, G) \geq x$. Theorem 4 is easily proved on lines similar to that of Theorem 3 for this case.

Now if $\mathsf{a} = \mathsf{b}$ or $\mathsf{a} = \mathsf{m}$ then we first check if either of the following two conditions hold:

1. $\mathsf{L^a}_{>x}(\mathcal{A}) \neq \emptyset$.
2. There exists a good non-empty set G and finite word u such that $|u| \leq 2^n$ and $\delta_u(q_s, G) \geq x$.

If either of the above two conditions hold then we can conclude that $\mathsf{L^a}_{\geq x}(\mathcal{A}) \neq \emptyset$. Otherwise, the proof of Lemma 6 shows that $\mathsf{L^a}_{\geq x}(\mathcal{A}) \neq \emptyset$ iff there exists a set $\emptyset \subset C \subseteq Q_1$, a state $q_0 \in Q_0$ and words $u, v \in \Sigma^*$ such that $|u| \leq 2^n$, $|v| \leq 2^{(n+2)\log n}$ such that following hold:

1. $\mathsf{post}_u(q_s) = \{q_0\}$.
2. For every $q \in C \cup \{q_0\}$, $\mathsf{post}_v(q) = \{q\}$.
3. For every $q \in C$, the set Y_q of all states that appear on the (unique) finite run from q on the string v satisfies the acceptance condition given by Acc, i.e., $Y_q \cap \mathsf{Acc} \neq \emptyset$ if $\mathsf{a} = \mathsf{b}$ and $Y_q \in \mathsf{Acc}$ if $\mathsf{a} = \mathsf{m}$.
4. Let $\delta_u(q_s, q_0) = x_1, \delta_u(q_s, C) = z_1, \delta_v(q_0, C) = z_2, \delta_v(q_0, Q_1) = y_2$. Then
$$x_1 \frac{z_2}{y_2} + z_1 \geq x.$$

Observe that $\mathsf{val}(C, x, u) = \frac{x - z_1}{x_1}$. Hence Conditon 4 can be restated as

4. Let $\delta_u(q_s, q_0) = x_1, \delta_u(q_s, C) = z_1, \delta_v(q_0, C) = z_2, \delta_v(q_0, Q_1) = y_2$. Then
$$\frac{z_2}{y_2} \geq \mathsf{val}(C, x, u).$$

Suppose that there are $\emptyset \subset C \subseteq Q_1, q_0 \in Q_0, u, v \in \Sigma^*$, which satisfy the above conditions. Now, we make two more observations which will be useful in the algorithm.

(a) Having fixed C and q_0 above, we can always replace u by u_1 such that $|u_1| \leq 2^n$, $\mathsf{post}_{u_1}(q_s) = \{q_0\}$ and $\mathsf{val}(C, x, u_1) \leq \mathsf{val}(C, x, u)$. Hence, once C and q_0 has been fixed, we can take u to be the word that realizes the minimum of the set $\{\mathsf{val}(C, x, u_1) \mid |u_1| \leq 2^n, \mathsf{post}_{u_1}(q_s) = \{q_0\}\}$.

(b) Also, it can be shown that
$$\frac{z_2}{y_2} \geq \mathsf{val}(C, x, u) \Leftrightarrow \mathsf{val}(C, x, uv) \leq \mathsf{val}(C, x, u).$$
Hence, once we have picked u as described above, we can choose v to be the word such that $|v| \leq 2^{(n+2)\log n}$, v satisfies Conditions 2 and 3 above and minimizes $\mathsf{val}(C, x, uv)$.

Thus, we can say that $\mathsf{L^a}_{\geq x}(\mathcal{A})$ will be non-empty if there is a set $C \subseteq Q_1$ and a state $q_0 \in Q_0$ such that if we choose words u, v according to (a) and (b) then $\mathsf{val}(C, x, uv)$ must be smaller than $\mathsf{val}(C, x, u)$. Hence, it suffices to show that for any given C, q_0 we can check that u, v chosen according to (a) and (b) have the property that $\mathsf{val}(C, x, uv) \leq \mathsf{val}(C, x, u)$.

In order to achieve this, we first compute using a dynamic algorithm the value $\mathsf{val}(C, x, u)$ for u chosen according to (a) above. This is carried out as follows. For each $C_1 \subseteq Q_1$ and integer $0 \leq i \leq 2^n$, we define $Val_i(C_1, q)$ to be $\min\{\mathsf{val}(C_1, x, u) : |u| = i, \mathsf{post}_u(q_s) = q\}$; here we take the minimum of an empty set to be ∞.

The function Val_i is computed from Val_{i-1} as follows.

1. $Val_0(C_1, q_s) = x$.
2. $Val_0(C_1, q) = \infty$ if $q \neq q_s$
3. $Val_i(C_1, q_1) = \min\{\frac{Val_{i-1}(C_1^{a,q}, q) - \delta_a(q, C_1)}{\delta_a(q, q_1)} \mid a \in \Sigma, q \in Q, \delta_a(q, q') > 0\}$ where $C_1^{a,q} = \{r \in Q_1 : \delta_a(r, C_1) = 1\}$.

The required value is $\min\{Val_i(C, q_0) \mid 0 \leq i \leq 2^n\}$.

Now having computed the value $\mathsf{val}(C, x, u)$ for u chosen according to (a) above, we can similarly compute the value $\mathsf{val}(C, x, uv)$ for v chosen according to (b) above. This is carried out by a dynamic algorithm which follows the lines of the computation of $\mathsf{val}(C, x, u)$. Having computed these values, we just need to check that $\mathsf{val}(C, x, uv) \leq \mathsf{val}(C, x, u)$.

E Proof of Theorem 5

It is well known that the problem of determining if the intersection of the languages accepted by a given finite set of deterministic automata, on finite strings, is empty is **PSPACE**-complete. We reduce this problem to our problem.

Let $\mathcal{A}_1, \mathcal{A}_2, ..., \mathcal{A}_m$, where $m \geq 2$, be the given set of deterministic automata on finite strings over an input alphabet Σ'. For $i = 1, ..., m$, let $\mathcal{A}_i = (Q_i, q_i, \delta_i, F_i)$ where q_i is the initial state of $\mathcal{A}_i$, $F_i \subseteq Q_i$, $\delta_i : Q_i \times \rightarrow Q_i$, $F_i \subseteq Q_i$. We assume that the sets Q_i, $i = 1, ..., m$ are mutually disjoint.

Now, we define a HPA $\mathcal{A} = (Q, q_s, \delta, F)$ over $\Sigma = \Sigma' \cup \{a\}$ where a is any symbol not in Σ', as follows. $Q = \cup_{1 \leq i \leq m} Q_i \cup \{q_s, r\}$, where $q_s, r \notin Q_i$ for $i = 1, ..., m$ and $F = \cup_{1 \leq i \leq m} F_i$. Let $y = \frac{1}{2} - \frac{1}{4(m-1)}$. Now, we have the following transitions: $\delta(q_s, a, r) = y$ and $\delta(q_s, a, q_i) = \frac{1-y}{m}$ for $i = 1, ..., m$; $\delta(q_s, b, r) = 1$ for $\forall b \in \Sigma'$; for each $q \in Q_i, b \in \Sigma'$, $\delta(q, b, q') = 1$ iff $\delta_i(q, b) = q'$ and is 0 otherwsie, and $\delta(q, a, r) = 1$; finally, r is an absorbing state on all inputs.

Let $\rhd \in \{>, \geq\}$. It is fairly easy to see that for any $u \in (\Sigma')^*$, $\delta_{au}(q_s, F) \rhd \frac{1}{2}$ iff u is accepted by $\mathcal{A}_i$ for $i = 1, ...m$; furthermore, for any $v \in \Sigma^*$ that does not start with a or that contains a after the first symbol, $\delta_v(q_s, F) = 0$. Hence, $\mathsf{L^f}_{\rhd \frac{1}{2}}(\mathcal{A}) = \emptyset$ iff there are no input strings that are accepted by each $\mathcal{A}_i$, for $i = 1, ..., m$.

F Proof of Theorem 6

Let $\mathcal{A} = (Q, q_s, \delta, \mathsf{Acc})$. If Acc is a finite accepting condition that Acc is a set $Q_f \subseteq Q$. Consider $\mathsf{Acc}' = Q \setminus Q_f$ and let $\mathcal{B} = (Q, q_s, \delta, \mathsf{Acc}')$. Observe that $\mathsf{L^f}_{>x}(\mathcal{A}) = \Sigma^*$ iff $\mathsf{L^f}_{\geq 1-x}(\mathcal{B}) = \emptyset$ and that $\mathsf{L^f}_{\geq x}(\mathcal{A}) = \Sigma^*$ iff $\mathsf{L^f}_{>1-x}(\mathcal{B}) = \emptyset$. The upper bound follows Theorem 3 and Theorem 4.

If Acc is a Büchi acceptance condition then Acc is a set $Q_f \subseteq Q$. Let $Q' = Q \setminus Q_f$ and consider the Muller acceptance condition $\mathsf{Acc}' = 2^{Q'}$. Observe that $\mathsf{L^b}_{>x}(\mathcal{A}) = \Sigma^\omega$ iff $\mathsf{L^m}_{\geq 1-x}(\mathcal{B}) = \emptyset$ and that $\mathsf{L^b}_{\geq x}(\mathcal{A}) = \Sigma^\omega$ iff $\mathsf{L^m}_{>1-x}(\mathcal{B}) = \emptyset$. The upper bound now follows Theorem 3 and Theorem 4. Note, a priori it seems that the construction of $\mathcal{B}$ causes an exponential blowup thanks to Acc'. However, it can be shown that the algorithms in the proofs can still be carried out in **EXPTIME**. For example, when checking $\mathsf{L^m}_{>1-x}(\mathcal{B}) \neq \emptyset$ we have to guess a good witness set and then we have to check that it is good (with accepting conditions Acc'). Now, checking whether this witness set is good can be carried out without explicitly constructing the set Acc'. The other checks do not depend on Acc'.

Similarly, if Acc is a Muller acceptance condition then we can show that checking non-universality is in **EXPTIME**. The lower bounds can be proved along the lines of the proof of Theorem 5.

G Proof of Propositon 6

Before proving Proposition 6, we observe that integer HPA have the special property that transitions between states of level 0 have transition probability of the form $\frac{1}{m}$ where m is a natural number.

Proposition 7. *For an integer* HPA $\mathcal{A}$*, any pair of states* $q, q' \in Q_0$ *and* $a \in \Sigma$*, if* $\delta_a(q, q') \neq 0$ *then* $\delta_a(q, q') = \frac{1}{m}$ *for some* $m \in \mathbb{N}$.

Proof. Let $m = 1 + \sum_{q'' \in Q_1} \frac{\delta_a(q,q'')}{\delta_a(q,q')}$. Observe that $m \in \mathbb{N}$ by the properties of an integer HPA. Moreover $m \times \delta_a(q, q') = \sum_{q'' \in Q} \delta_a(q, q'') = 1$. The proposition follows. □

Proposition 6 is proved by induction on the length of u. Observe that for any $C \subseteq Q_1$, $\mathsf{val}(C, x, \epsilon) = x = \frac{c}{d}$, and so the proposition holds in the base case. Let the proposition hold for string u. Consider $a \in \Sigma$ and $C \subseteq Q_1$. We will show that the proposition holds for $\mathsf{val}(C, x, ua)$. Let $D = \{q \in Q_1 \,|\, \mathsf{post}(q, a) \in C\}$. Observe that if $\mathsf{val}(C, x, ua) \in [0, 1]$ then $\mathsf{val}(D, x, u) \in [0, 1]$. Hence, by the induction hypothesis, we know that there is e such that $\mathsf{val}(D, x, u) = \frac{e}{d}$. If $\delta_{ua}(q_s, Q_0) = 0$ and $\mathsf{val}(C, x, ua) \in [0, 1]$ then $\mathsf{val}(C, x, ua) = 0$ and the proposition follows. Let us consider the case when $\delta_{ua}(q_s, Q_0) \neq 0$. Let $\mathsf{post}(q_s, u) \cap Q_0 = \{q\}$ and $\mathsf{post}(q_s, ua) \cap Q_0 = \{q'\}$. We can make the following sequence of observations.

$$\mathsf{val}(C, x, ua) = \frac{x - \delta_{ua}(q_s, C)}{\delta_{ua}(q_s, Q_0)} = \frac{x - (\delta_u(q_s, D) + \delta_u(q_s, q)\delta_a(q, C))}{\delta_u(q_s, q)\delta_a(q, q')}$$

$$= \frac{\mathsf{val}(D, x, u)}{\delta_a(q, q')} - \frac{\delta_a(q, C)}{\delta_a(q, q')}$$

Proposition 7 implies that $\frac{1}{\delta_a(q,q')}$ is a natural number. Further, since $\mathcal{A}$ is an integer HPA, $\frac{\delta_a(q,C)}{\delta_s(q,q')}$ is also a natural number. Putting it all together the proposition follows.

H Proof of Lemma 4

The result is proved by constructing a single NFA that essentially recognizes each of the languages in the proposition. The NFA that we will construct will maintain the level 0 state that $\mathcal{A}$ could be in, guess the set C whose val we care to compute, and maintain a upper bound on val. Proposition 3 and 6 together ensure that there are only finitely many relevant values that val can take which can be maintained by a finite automaton.

We now give the precise construction of the NFA. Consider NFA $\mathcal{M} = (U, r_0, \delta', F)$, where the set of states is $U = (Q_0 \cup \{*\}) \times 2^{Q_1} \times (\{0, 1, 2, \ldots d\} \cup \{+, -\})$. Thus, a state is of the form (q, C, e), where $q \in Q_0 \cup \{*\}$, $C \subseteq Q_1$ and e is a natural number between 0 and d or $+$ or $-$. The invariant that the automaton will maintain is the following.

$$(q, C, e) \in \delta'(r_0, u) \text{ iff } \text{ either } (\mathsf{post}(q_s, u) \cap Q_0 = \emptyset \text{ and } q = *) \text{ or } \mathsf{post}(q_s, u) \cap Q_0 = \{q\}$$
$$\text{and } \begin{cases} \mathsf{val}(C, x, u) \leq \frac{e}{d} \text{ if } e \notin \{+, -\} \\ \mathsf{val}(C, x, u) < 0 \text{ if } e = - \end{cases}$$

To be consistent with the above invariant, the initial state $r_0 = (q_s, \emptyset, c)$. The transition function δ' will maintain this invariant and is defined as follows. $(q', C', e') \in \delta'((q, C, e), a)$ if (q', C', e') satisfies the following constraints.

- If $q = *$ or $Q_0 \cap \mathsf{post}(q, a) = \emptyset$ then $q' = *$. Otherwise, $q' \in Q_0 \cap \mathsf{post}(q, a)$.
- $\mathsf{post}(C, a) \subseteq C'$, and
- If $e \in \{+, -\}$ or $q = *$ then $e' = e$. When $q' = *$, $e' = +$ if $\delta_a(q, C') < \frac{e}{d}$, $e' = -$ if $\delta_a(q, C') > \frac{e}{d}$, and $e' = 0$ if $\delta(q, C') = \frac{e}{d}$. Otherwise, consider

  $$f = \frac{e}{\delta_a(q, q')} - d \cdot \frac{\delta_a(q, D)}{\delta_a(q, q')}$$

  Observe that f is an integer. Then, $e' = f$ if $0 \leq f \leq d$, $e' = +$ if $f > d$, and $e' = -$ if $f < 0$.

The transition function defined above maintains the required invariant.

The languages defined in the proposition can then be defined by making an appropriate choice of final states. To recognize the language $\mathsf{L}_{(q,C,e)}$ where $q \in Q_0 \cup \{*\}$, and $e \in \{0, 1, \ldots d\} \cup \{-\}$, we take $F = \{(q, C, e)\}$. Next, $\overline{\mathsf{L}_{(q,C,+)}}$, when $q \in Q_0 \cup \{*\}$ can be recognized by taking $F = \{(q, C, e) \mid e \neq +\}$. Hence $\mathsf{L}_{(q,C,+)}$ is also regular. This completes the proof of the proposition.

I Proof of Theorem 7

We need to show that the language $\mathsf{L^a}_{\geq x}(\mathcal{A})$ is regular for $\mathsf{a} \in \{\mathsf{f}, \mathsf{b}, \mathsf{m}\}$. From Proposition 1 and Lemma 4, we can conclude the regularity of the language

$$L_1 = \left(\bigcup_{C \subseteq Q_1,\ q \in Q_0 \cup \{*\},\ e \in \{0, -\}} \mathsf{L}_{(q,C,e)} \mathsf{L}_C \right) \cup \left(\bigcup_{C \subseteq Q_1,\ q \in Q_0, e \in [0,1)} \mathsf{L}_{(q,C,e)} \mathsf{L}_{C \cup \{q\}} \right).$$

Now, $\mathsf{L^f}_{\geq x}(\mathcal{A}) = L_1$ and so that completes the proof of this proposition for the finite input case. When the case of Büchi acceptance (or Muller acceptance), $\mathsf{L^b}_{\geq x}(\mathcal{A})$ contains strings that are accepted with probability x "in the limit", in addition to the strings in L_1. The challenge is in handling this case, and the details are defered to Appendix I.

We will define a language L_2 such that $\mathsf{L^a}_{\geq x}(\mathcal{A}) = L_1 \cup L_2$ (when $\mathsf{a} \in \{\mathsf{b}, \mathsf{m}\}$) and prove that L_2 is regular to complete the proof.

To define the language L_2, we need some auxiliary definitions, which we present first. Recall that the level 0 and 1 states of $\mathcal{A}$ are Q_0 and Q_1. Let the threshold $x = \frac{c}{d}$, and the input alphabet of $\mathcal{A}$ be Σ. For an input $\alpha \in \Sigma^\omega$, a *witnessing run* of $\mathcal{A}$ on α is an infinite sequence $\lambda = (q_0, C_0, e_0), (q_1, C_1, e_1), \ldots$ such that

- $q_0 = q_s$, $C_0 = \emptyset$, $e_0 = c$,
- For each i, $q_i \in Q_0$, $C_i \subseteq Q_1$, and $e_i \in \{0, 1, \ldots d\}$,
- For each i, $q_{i+1} \in \mathsf{post}(q_i, \alpha[i])$, and $\mathsf{post}(C_i, \alpha[i]) \cap Q_1 \subseteq C_{i+1}$, and
- For each i, $\mathsf{val}(C_i, x, \alpha[0 : i-1]) \leq \frac{e_i}{d}$.

In other words, a witnessing run is a run of the nondeterministic machine constructed in the proof Lemma 4. Given a witnessing run $\lambda = (q_0, C_0, e_0), (q_1, C_1, e_1), \ldots$ on input α, the set of runs associated with λ, denoted $R(\lambda, \alpha)$, is given as

$$R(\lambda, \alpha) = \{\rho \in Q^\omega \mid \forall i.\ \rho[i] \in \{q_i\} \cup C_i \text{ and } \delta_{\alpha[i]}(\rho[i], \rho[i+1]) > 0\}$$

In other words, $R(\lambda, \alpha)$ is the set of all runs of $\mathcal{A}$ on α that "conform" to the witnessing run λ. Finally, we say that a witnessing run λ on input α is *accepting* iff every run $\rho \in R(\lambda, \alpha)$ is an accepting run of $\mathcal{A}$.

Consider the language L_2 defined as follows.

$$L_2 = \{\alpha \in \Sigma^\omega \mid \text{there is an accepting witnessing run } \lambda \text{ on } \alpha\}$$

Proposition 8. *For any* HPA $\mathcal{A}$ *with threshold* $x = \frac{c}{d}$*, and* $\mathsf{a} \in \{\mathsf{b}, \mathsf{m}\}$*,* $\mathsf{L^a}_{\geq x}(\mathcal{A}) = L_1 \cup L_2$*.*

The crux of the proof is arguing that the language L_2 is regular. We do this by constructing a nondeterministic Büchi automata accepting L_2. The nondeterministic automaton described in the proof of Lemma 4 almost has all the ingredients to recognize the language L_2 — it keeps track of the states visited in level 0, and the val for some subset of states at level 1. However, it does not keep track of whether a final accepting state has been visited. We will fix this problem carrying out something like a "marked subset" construction and using the ideas of the automaton in Lemma 4. Details are given below.

Proposition 9. *The language* L_2 *is regular.*

Proof. We prove this observation for only the case when $\mathcal{A}$ is a PBA. We will sketch how the construction can be adapted for the Muller case at the end of this proof sketch.

Let $\mathcal{A} = (Q, q_s, \delta, Q_f)$, where Q_f is the set of final states, and we take the threshold $x = \frac{c}{d}$. Consider the following nondeterministic Büchi automaton $\mathcal{M} = (U, r_0, \delta', F)$. A state in U is a tuple of the form (q, C, m, e), where $q \in Q_0$, $C \subseteq Q_1$, $\mathsf{m} : C \cup$

$\{q\} \rightarrow \{0, 1\}$, and $e \in \{0, 1, 2, \ldots d\}$. Here m is a marking on $C \cup \{q\}$ that keeps track of whether a final state has been visited. The initial state $r_0 = (q_s, \emptyset, \mathsf{m}_*, c)$, where $\mathsf{m}_*(q_s) = 1$. The set of final states F will be all states (q, C, m, e) such that for every q' in the domain of m, $\mathsf{m}(q') = 1$.

Before defining the transition function δ' it is useful to spell out the invariant maintained by the automaton, to explain how the construction works. Suppose $\rho = (q_0, C_0, \mathsf{m}_0, e_0)$, $(q_1, C_1, \mathsf{m}_1, e_1), \ldots$ is an infinite run of $\mathcal{M}$ on input word α. The constraints this run will satisfy are the following.

- For each i, $q_i \in \mathsf{post}(q_s, \alpha[0:i]) \cap Q_0$, and $\mathsf{val}(C_i, x, \alpha[0:i]) \leq \frac{e_i}{d}$
- For each i, let j_i be the maximum position $< i$ such that for all q in the domain of m_{j_i}, $\mathsf{m}_{j_i}(q) = 1$. Note, such a j_i always exists because $r_0 = (q_0, C_0, \mathsf{m}_0, e_0)$ is such a state. For every state q such that $\mathsf{m}_i(q) = 1$, every run of $\mathcal{A}$ from some state $q' \in \{q_{j_i}\} \cup C_{j_i}$ to q on the word $\alpha[j_i : i]$ visits an accepting state in Q_f.

Observe that if this invariant is maintained then $\mathcal{M}$ will indeed accept the language L_2.

The transition function δ' that maintains this invariant is defined as follows. $(q', C', \mathsf{m}', e') \in \delta'((q, C, \mathsf{m}, e), a)$ if $(q', C', \mathsf{m}', e')$ satisfies the following constraints.

- $q' \in \mathsf{post}(q, a) \cap Q_0$
- $\mathsf{post}(C, a) \subseteq C'$
- $e' = \frac{e}{\delta_a(q,q')} - d \cdot \frac{\delta_a(q,D)}{\delta_a(q,q')}$ and $e' \in \{0, 1, \ldots d\}$
- m' is given as follows. For every $p' \in C' \setminus \mathsf{post}(C, a)$, $\mathsf{m}'(p') = 0$. For every $p' \in (C' \cup \{q'\}) \cap Q_f$, $\mathsf{m}'(p') = 1$. For $p' \in (C' \cup \{q'\}) \setminus Q_f$, m' is defined as follows. If $\mathsf{m}(p) = 1$ for every $p \in C \cup \{q\}$ then $\mathsf{m}'(p') = 0$. On the other hand, if $\mathsf{m}(p) = 0$ for some $p \in C \cup \{q\}$ then $\mathsf{m}'(p') = 1$ if and only if for every $p_1 \in C \cup \{q\}$ such that $p' \in \mathsf{post}(p_1, a)$, $\mathsf{m}(p_1) = 1$.

The automaton $\mathcal{M}$ constructed above recognizes L_2. If $\mathcal{A}$ has a Muller acceptance condition then $\mathcal{M}$ will intially guess an accepting set of states F, and the marking function will turn 1 only if the sub-run visits each of the states in F. The details are skipped.

Regularity of $\mathsf{L^a}_{\geq x}(\mathcal{A})$ now follows from Proposition 8, and the regularity of L_1 and L_2 (Proposition 9).

J Proof of Theorem 8

The proof of Theorem 7 shows that the languages accepted by an integer HPA are recognized by an an exponential sized (non-probabilistic) automaton. Thus the emptiness problem for integer HPA can be reduced to the emptiness problem for exponential sized automaton, giving us the **PSPACE** upper bound (we don't need to explicitly construct the automaton, just *run* the standard emptiness checking algorithm for (non-probabilistic) automaton). Next, recall that in the proof of Theorem 6, we showed that checking the universality of an HPA $\mathcal{A}$ can be reduced to checking the emptiness of another closely related HPA $\mathcal{B}$ simply by changing the accepting condition of $\mathcal{A}$. Thus, if $\mathcal{A}$ is an integer HPA then so is $\mathcal{B}$. Finally, since the emptiness problem of integer HPA can be decided in **PSPACE**, so can the universality problem.