On the Expressiveness and Complexity of Randomization in Finite State Monitors

Rohit Chadha
Univ. of Illinois at Urbana-Champaign
and
A. Prasad Sistla
Univ. of Illinois at Chicago
and
Mahesh Viswanathan
Univ. of Illinois at Urbana-Champaign

In this paper we introduce the model of *finite state probabilistic monitors* (FPM), which are finite state automata on infinite strings that have probabilistic transitions and an absorbing reject state. FPMs are a natural automata model that can be seen as either randomized run-time monitoring algorithms or as models of open, probabilistic reactive systems that can fail. We give a number of results that characterize, topologically as well as with respect to their computational power, the sets of languages recognized by FPMs. We also study the emptiness and universality problems for such automata and give exact complexity bounds for these problems.

Categories and Subject Descriptors: D.2.4 [**Software Engineering**]: Program Verification; F.1.1 [**Theory of Computation**]: Models of Computation; F.1.2 [**Theory of Computation**]: Modes of Computation

1. INTRODUCTION

In this paper we introduce the model of *Finite state Probabilistic Monitors* (FPMs) and study their expressive power and complexity of their decision problems. FPMs are finite state automata on infinite strings that choose the next state on a transition based on a probability distribution, in addition to the input symbol read. They have a special *reject state*, with the property that once the automaton enters the reject state, it remains there no matter what the input symbol read is. On an infinite input word α, a FPM defines an infinite-state Markov chain and the word α is said to be accepted (rejected) with probability x if the probability of reaching the reject state is $1 - x$ (x respectively).

FPMs are natural models of computation that arise in a variety of settings. They can be seen as a special case of Probabilistic Büchi Automata (PBA) [Baier and Gröβer 2005] [1]. PBAs are like FPMs, except that instead of a reject state they have a set of *accept states*. PBAs are a generalization of probabilistic finite automata of [Rabin 1963; Salomaa 1973; Paz 1971] to infinite strings and the probability of acceptance of an infinite word α is the probability measure of runs on α that satisfy the Büchi acceptance condition, namely, those that visit an accepting state infinitely often. A FPM is nothing but a PBA $\mathcal{B}$, where instead

[1] In some ways, FPMs considered here are more general than the PBA model. Baier et. al. consider PBAs to be language acceptors that accept all strings whose measure of accepting runs in non-zero. We, on the other hand, consider various probability thresholds for acceptance, and as will be seen in this paper, this generalization makes a big difference to both the expressive power and the complexity of decision problems.

of the Büchi acceptance condition, the accepting runs are defined by a ω-regular safety language (or the complement of a safety language) over the states of $\mathcal{B}$.

Another context where FPMs arise is as *randomized monitoring algorithms*, and this was our original motivation for studying them. Run-time monitoring of system components is an important technique that has widespread applications. It is used to detect erroneous behavior in a component that either has undergone insufficient testing or has been developed by third party vendors. It is also used to discover emergent behavior in a distributed network like *intrusion*, or more generally, perform *event correlation*. In all these scenarios, the monitor can be abstractly viewed as an algorithm that observes an unbounded stream of events to and from the component being examined. Based on what the monitor has seen up until some point, the monitor may decide that the component is behaving erroneously (or is under attack) and raise an "alarm", or may decide that nothing worrisome has been observed. Hence, the absence of an error (or intrusion) is indicated implicitly by the monitor by the absence of an alarm. On the flip side, a behavior is deemed incorrect by the monitor based on what has been observed in the past, and this decision is independent of what maybe observed in the future. FPMs are *finite state* monitors that make probabilistic decisions while they are observing the input stream. The unique, absorbing reject state corresponds to the fact that once a behavior is deemed incorrect by a monitor, it does not matter what events are seen in the future.

Finally, FPMs are also models of open, finite state probabilistic systems that can fail. An open reactive system is a system that takes an infinite sequence of inputs from the environment and performs a computation by going through a sequence of states. Finite state automata on infinite strings have been used to model the behavior of such systems that have a finite number of possible states. Probabilistic open reactive systems are systems whose behavior on an input sequence is probabilistic in nature. Such probabilistic behavior can be either due to the randomization employed in the algorithm executed by the system or due to other uncertainties, such as component failure, modeled probabilistically. Probabilistic Büchi Automata (PBA) can be used to model the behavior of such systems having a finite state space. FPMs which are PBAs having a single absorbing reject state can be used to model open probabilistic systems that can fail; the failure state is modelled by the reject state of the FPM. Decision problems on the language accepted by the FPM can help answer a number of verification problems on such systems, including, determining the probability of failure on some or all input sequences, and checking ω-regular safety properties (or their complement).

Motivated by these applications we study a number of questions about FPMs. The first question we ask is whether randomization is useful in the context of run-time monitoring. We consider both algorithms with one-sided and two-sided errors. We say that a property is *monitorable with strong acceptance* (MSA) if there is a monitor for the property that never deems a correct behavior to be erroneous, but may occasionally pass an incorrect behavior. On the other hand, a property is *monitorable with weak acceptance* (MWA) if there is a monitor that may raise false alarms on a correct behavior, but would never fail to raise an alarm on an incorrect one. Similarly we define classes of properties that have monitors with two-sided errors. We say a property is *monitorable with strict cut-points* (MSC) if, for some x, there is monitor such that on behaviors α satisfying the property, the probability that the monitor rejects α is strictly less than x. Finally, a property is *monitorable with non-strict cut-points* (MNC) if, for some x, there is monitor such that on behaviors α satisfying the property, the probability that the monitor rejects α is at most x.

We compare the classes of properties monitorable with FPMs with those that can be monitored deterministically. Deterministic monitors are well understood [Schneider 2000] to be only able to detect violations of *safety properties* [Alpern and Schneider 1985; Lamport 1985; Sistla 1985]. In addition, if the deterministic monitor is finite state, then the monitorable properties are regular safety languages. We contrast this with our observations about properties monitorable by FPMs, which are summarized in Figure 5 on page 18. We show that while the class MSA is exactly the class of ω-regular safety properties, the class MNC not only contains all ω-regular safety properties, but also contains some non-regular safety properties. However, even though the class MNC is uncountable, it is properly contained in the class of all safety properties. The classes MWA and MSC allow us to go beyond deterministic monitoring along a different axis. We show that MWA strictly contains all the ω-regular almost safety properties [2]. Even though, MWA contains some non-regular properties, they are very close to being ω-regular. More precisely, we show that the safety closure of any property in MWA (i.e., the smallest safety property containing it) and the safety closure of its complement, are both ω-regular. We note here that if we were to define the class MWA for probabilistic automata over finite strings then we will only obtain regular languages. We show that MWA is strictly contained in MSC which in turn is strictly contained in the class of all almost safety properties. Finally, we show that MSC and MNC are incomparable.

We also consider a sub-class of FPMs. Let us call a FPM $\mathcal{M}$ *x-robust* if for some $\epsilon > 0$, the probability that $\mathcal{M}$ rejects any string is bounded away from x by ϵ, i.e., it either rejects a string with probability greater than $x + \epsilon$ or with probability less than $x - \epsilon$. A *robustly monitorable property* then is just a property that has a robust monitor. Robustly monitorable properties are a natural class of properties. They are the constant space analogs of the complexity classes RP and co-RP, when x is 0 or 1. They also are a generalization of the notion of isolated cut-points [Rabin 1963; Salomaa 1973; Paz 1971] in finite string probabilistic automata to infinite strings. We show that robustly monitorable properties are exactly the same as the class of ω-regular safety properties.

We conclude the summary of our expressiveness results with one more observation about the advantage of randomization in the context of monitoring. Even though MSA and robust monitors recognize the same class of languages as deterministic monitors, they can be exponentially more succinct. More specifically, there exists a family $\{L_i\}_{i=0}^{\infty}$ such that each L_i is an ω-regular safety language. Each L_i can be recognized by a robust FPMs M_i with $i + 2$ states that accepts every string in L_i with probability 1, and rejects every string not in L_i with probability at least 2^i. In contrast the smallest deterministic monitor for L_i has 2^i states. Thus, FPMs are space efficient algorithms for monitoring regular safety properties. Although, non-deterministic automata also exhibit the succinctness property, they are not directly executable.

The next series of questions that we investigate in this paper, is the complexity of the emptiness and universality problems for the various classes of FPM languages introduced. Apart from being natural decision problems for automata, emptiness and universality are important in this context for a couple of reasons. For FPMs that model monitoring algorithms, emptiness and universality checking serve as sanity checks: if the language of a monitor is empty then it means that it is too conservative, and if the language is universal

[2] An almost safety property is a countable union of safety properties.

then it means that it is too liberal. On the other hand, for FPMs modeling open, probabilistic reactive systems, emptiness and universality are the canonical verification problems for such systems.

We give exact complexity bounds for these decision problems for all the classes of FPMs considered; our results are summarized in Figure 6 on page 26. We show that the emptiness problems for monitors with one sided error, i.e., for the classes MSA and MWA, are **PSPACE**-complete. These results are interesting in the light of the fact that checking non-emptiness of a non-deterministic finite state automaton on infinite strings, with respect to any of the commonly used acceptance conditions, is in polynomial time. We also show that the emptiness problem for monitors with two sided errors is undecidable. More specifically, we show that the emptiness problem for the class MNC is **R.E.**-complete, while it is **co-R.E.**-complete for the class MSC. Next, we show that the universality problem for the class MSA is **NL**-complete while for MWA it is **PSPACE**-complete. This problem is **co-R.E.**-complete for the class MNC, while it is Π^1_1-complete for MSC. Many of these results, for both the lower and upper bounds, are quite surprising, and their proofs non-trivial. We conclude the introduction by highlighting the reason why the complexity for the emptiness and universality problems is so different for MSC and MNC. Words which have rejection probability x (which is the threshold of the FPM) can only be detected in the limit, whereas words rejected with probability $> x$ are witnessed by a finite prefix.

Paper Outline. The rest of paper is organized as follows. We first discuss related work. Then in Section 2, we motivate FPMs with some examples. In Section 3 we give basic definitions and properties of safety and almost safety languages. We formally define FPM's in Section 4 and the language classes MSA, MWA, MNC and MSC. Our expressiveness results are presented in Section 5 and the complexity, decidability results are presented in Section 6. We conclude in Section 7.

1.1 Related Work

The model of FPMs introduced here is a special case of Probabilistic Büchi Automata (PBA) introduced in [Baier and Gröβer 2005], which accept infinite strings whose accepting runs have non-zero measure — the language $\mathcal{R}_{<1}(\mathcal{M})$ of a monitor $\mathcal{M}$ is the same as the language $\mathcal{L}(\mathcal{B})$ of a PBA $\mathcal{B}$ which has the same transition structure as $\mathcal{M}$ and has as accept states the non-reject states of $\mathcal{M}$. [Baier and Gröβer 2005] show that there are PBAs $\mathcal{B}$ such that $\mathcal{L}(\mathcal{B})$ is non-regular. The emptiness and universality problems for such PBAs has been shown to be undecidable in [Baier et al. 2008]. Thus, our results on MWA demonstrate that we have identified a restricted form of PBAs that are effective. Under the almost-sure semantics of PBAs, introduced in [Baier et al. 2008], a PBA $\mathcal{B}$ is taken to accept all strings whose accepting runs have measure 1. Thus, the universality problem of $\mathcal{R}_{<1}(\mathcal{M})$ can also be seen as the emptiness problem for almost-sure semantics of special kinds of PBAs. The emptiness problem of PBAs with almost-sure semantics was shown to be in **EXPTIME** in [Baier et al. 2008]; the fact that the universality problem for MWA is **PSPACE**-complete once again demonstrates that FPMs are more effective than general PBAs.

We draw upon, and generalize, many proof techniques introduced in the context of finite strings and probabilistic automata [Rabin 1963; Salomaa 1973; Paz 1971] to infinite strings. In particular, the iteration lemma and its proof presented here, is closely related to a similar result in the context of finite strings (though it is generalized here to infinite strings). Also, the proof that robust monitors define ω-regular languages is inspired by a similar result for finite strings by [Rabin 1963], that says that probabilistic finite automata with isolated cut-points define regular (finite word) languages. However, our proof for infinite word languages, crucially relies on the fact that robust monitors define safety languages, and no such topological property is relied upon in the finite case. Another proof technique that we use extensively is the **co-R.E.**-hardness proof for the emptiness of PFAs on finite words due to [Condon and Lipton 1989]. The proof is by a reduction from deterministic two-counter machines and uses a *weak equality test* by [Freivalds 1981] designed to check (using coin tosses and finite memory) whether two counters have the same value. The **co-R.E.**-hardness of the emptiness for the class MSC and universality for the class MNC, proved herein, are proved as a consequence of **co-R.E.**-hardness of emptiness of probabilistic automata with cut-points. The **R.E.**-hardness of the emptiness for the class MNC and Π^1_1-hardness for the class MSC shown here are proved by a non-trivial modification of the weak-equality test of [Freivalds 1981].

Our work was initially inspired by a desire to understand the power of randomization in the context of run time monitoring. Observing the dynamic behavior of a system has long been understood to be an important technique to detect faults; see [Plattner and Nievergelt 1981; Mansouri-Samani and Sloman 1992; Schroeder 1995] for early surveys about the area. A number of tools have been built in the recent past to analyze the behavior of software systems; examples of such systems include [Diaz et al. 1994; rov 1997; Ranum et al. 1997; Paxson 1997; Jeffery et al. 1998; Kim et al. 1999; Kortenkamp et al. 2001; Gates and Roach 2001; Havelund and Rosu 2001; Chen and Rosu 2007]. The importance of this area can be gauged by the success of a workshop series devoted exclusively to this topic [rv- 2008]. The theoretical foundations of what can and cannot be monitored was initiated by [Schneider 2000] where it was observed that only safety properties [Alpern and Schneider 1985; Lamport 1985; Sistla 1985] can be deterministically monitored; this characterization was refined to take into account computability considerations in [Viswanathan 2000]. Though safety properties constitute an upper bound on the class of deterministically monitorable properties, in practice, properties other than safety are also monitored by either under or over approximating them to safety properties [Amorium and Rosu 2005; Margaria et al. 2005], or by using a multi-valued interpretation of whether formula holds on a finite prefix [Pnueli and Zaks 2006; Bauer et al. 2006].

There has been quite a bit of work on enforcing different security properties through run time monitoring by employing different types of deterministic automata. For example, truncation automata and suppression automata have been proposed in [Schneider 2000] and [Ligatti et al. 2005], respectively. Both these automata effectively enforce properties that are only safety properties. In [Ligatti et al. 2005; 2009; Ligatti 2006], a new kind of automata called edit automata are proposed. These automata can buffer outputs generated by the system and release them only when the required property is satisfied by the buffered sequence of outputs. The authors show that edit automata can be used to enforce a class of properties called reasonable infinite renewable properties, which include many non-safety properties. In general, edit automata can buffer arbitrary number of outputs and hence can use unbounded amount of memory. They also can delay as well as transform the outputs generated by the system that is being monitored. Compared to these, FPMs only have finite number of states, and do not delay or modify the system outputs, i.e., they are passive in nature. On the other hand, they employ randomization and can monitor non-safety properties with some error probability. It will be interesting to investigate whether infinite renewable properties can be monitored using FPMs. This will be part of future work.

Further, in [Hamlen et al. 2006], the authors propose a taxonomy of enforceable security policies. Using this they broadly classify existing enforcement techniques into three different classes based on— static analysis, execution monitoring and program rewriting. Although FPMs are based on execution monitoring, they employ randomization which is quite novel and is not covered by the above classification. Randomization can make the monitor unpredictable making it difficult for an adversary to exploit it. Lastly, the use of statistical techniques is ubiquitous in intrusion detection systems [Group 2001]. However, the study of designing randomized monitors for formally specified properties was only recently initiated [Sistla and Srinivas 2008]. In this paper we continue this line of work. We would, however, like to contrast this line of work with that of [Sammapun et al. 2007], where they consider the design of randomized monitors for probabilistic systems. We also like to point out that [Gondi et al. 2009] presents and compares different monitoring techniques, i.e., deterministic, probabilistic and hybrid, for monitoring properties, specified by automata on infinite strings, of systems modeled as Hidden Markov Chains. It proposes and uses two different accuracy measures, called *acceptance* and *rejection* accuracies. It performs both theoretical as well as experimental analysis of these techniques on realistic systems and shows that randomization can be quite effective in achieving high levels of accuracies, especially when combined with deterministic techniques.

Finally, we would like to point out that a preliminary version of these results was presented in [Chadha et al. 2008]. The main difference between these two papers is as follows. First, since [Chadha et al. 2008] was an extended abstract it did not contain many proofs, which have been presented fully for the first time here. Second, the proof of Theorem 5.15 had an incorrect expression in [Chadha et al. 2008] which has been corrected here. Finally, while we stated the results for the emptiness problem of MWA and MSC in [Chadha et al. 2008], we had neither considered the emptiness problem for MSC and MNC, nor the universality problem for any of the classes of FPMs.

2. MOTIVATION

In this section, we motivate FPMs with some examples in monitoring and verification. The first two examples are on monitoring and the last two are on verification.

Monitoring non-safety properties. Consider a system that outputs an infinite sequence of symbols from $\{a, b\}$. Suppose we want to monitor the property which states that the symbol a should eventually appear in the input sequence. Since this property is not a safety property, it can not be monitored using deterministic monitors. On the other hand the probabilistic monitor M, shown in Figure 1, monitors this property. From the initial state q_0, after each input symbol, $\mathcal{M}$ acts as follows. If the input symbol is b then $\mathcal{M}$ goes to the reject state q_r with probability $\frac{1}{2}$ and stays in the initial state with probability $\frac{1}{2}$. If the input symbol is a then it goes to an accept state q_a with probability one. Both q_r and q_a are absorbing states, i.e., on further inputs $\mathcal{M}$ remains in those states with probability one. It is easy to see that, an input sequence that violates the property, i.e., that does not have a appearing in it, will be rejected with probability one. On the other hand, a good sequence, i.e., one that satisfies the property, may also be rejected, but with probability less than one. The probability of rejection of such a sequence gracefully degrades, i.e., increases, with the number of input symbols before the first a in the input. It is to be

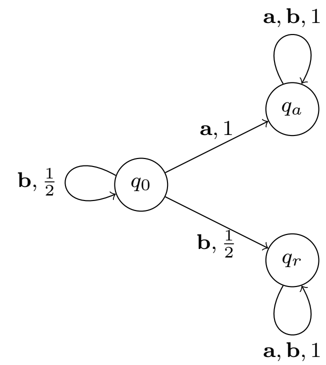

Fig. 1. FPM for $\Diamond a$

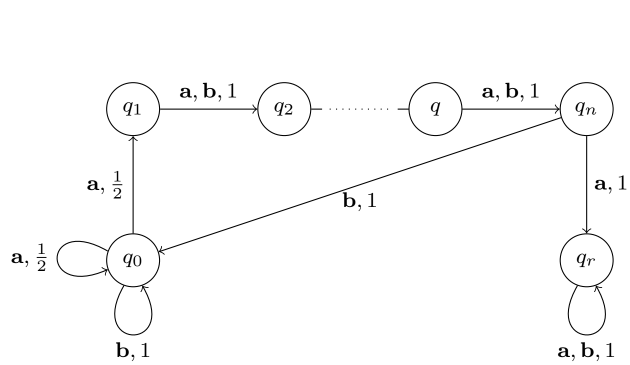

Fig. 2. Robust FPM with exponentially fewer states

noted that one can give deterministic algorithms, e.g., employing counters and/or using over-approximations, for monitoring this property; such an algorithm rejects the sequence that does not contain a and that also rejects some other good sequences. Combining such methods with probabilities, one can ensure that more good sequences are accepted (see [Sistla and Srinivas 2008; Gondi et al. 2009]).

Succinct monitors. The previous example illustrates a property that can be monitored in a probabilistic sense, but that can not be monitored using a deterministic monitor. Probabilistic monitors can also be used to generate succinct monitors even for safety properties. Let $n > 0$ be a positive integer. Consider the following ω-regular safety property L_n, over the input alphabet $\{a, b\}$, that requires that the n^{th} symbol after every a should be the symbol b. It is not difficult to show that any deterministic monitor for this property has to have at least 2^n states. On the other hand, the probabilistic monitor N with $n+2$ states, shown in Figure 2, monitors this property as follows. In the initial state q_0, when ever the monitor N sees a, with probability $\frac{1}{2}$ it remains in q_0, and with probability $\frac{1}{2}$ it goes to another state, say q_1. From q_1 it counts n input symbols and checks if the n^{th} symbol is b; if that symbol is b then it goes back to the initial state q_0, else it goes to the reject state q_r. It is not difficult to see that if the input sequence satisfies the property L_n then N rejects it with zero probability. On the other, if it does not satisfy L_n then it is rejected with probability at least $\frac{1}{2^n}$. This is because it catches the error if for all the n symbols before the faulty a the machine did not take the transition q_0 to q_1, while at the faulty a it does take this transition. Notice that N is different from M in the sense that it rejects all bad sequences with probability at least $\frac{1}{2^n}$ and rejects all good sequences with probability 0. It is also to be noted that if the input sequence has infinite number of errors, i.e., infinite number of occurrences of a such that the n^{th} symbol after it is is not a b, then the sequence is rejected by N with probability 1 (in general, the probability of rejecting an input sequence increases with the number of errors). This monitor has only $n+2$ states and is succinct compared to any deterministic monitors. Thus, we see that probabilistic monitors can give space efficient monitors. Note that non-deterministic automata can also be succinct language recognizers, but they are not executable.

The above two examples show that randomization can be used to monitor for liveness properties and can also be used for obtaining succinct monitors for safety properties. One can also give deterministic monitors for both of the above properties that err sometimes, with the monitor for the later property being succinct. These examples show that randomization is an alternate/complementary technique for monitoring. It has the advantage of being unpredictable making it difficult to be exploited by an adversary when monitoring for security properties. The reader is referred to [Sistla and Srinivas 2008; Gondi et al. 2009] for various practical examples and for comparison between deterministic, randomized and hybrid techniques for monitoring various properties, including liveness properties, specified by automata on infinite strings.

Checking failure freedom property. Consider a system consisting of k number of machines (for some $k > 0$) and a repair kit (this system may be a robot exploring Martian surface). At any time, only a subset of the machines are operational. The state of the system is denoted by the set of machines that are operational. There is also a special state denoting the failure of the repair kit as well as all the machines. The repair kit repairs failed machines over time. At any time the system takes an input from outside and performs work denoted by the input. During the processing of the input, some machines may fail and some previously failed machines may get repaired, or the repair kit and all the machines may fail. Both failures of machines and the repair kit, as well as repairing of machines is probabilistic in nature. Thus after each input, the state of the system may change according to a discrete probability distribution which may depend on the input symbol as well as the current state. One may want to determine the sequences of inputs on which the systems fails with more than a critical probability, or determine if the system never fail with more than a critical probability.

Verifying ω-regular safety properties of probabilistic open systems. FPMs can also be used for verifying safety properties of open probabilistic systems. As indicated in the introduction, such a system can be modeled as PBA, call it as *system automaton*. Now, suppose we want to verify that on every infinite input sequence, the sequence of states the system goes through satisfies a safety property specified by a deterministic finite state automaton A, with probability greater than some x. The input symbols to A are the states of the system automaton. The automaton A has an error state which is an absorbing state. Any sequence of inputs leading to the error state is a bad sequence. An infinite sequence of inputs to A is good if none of its prefixes is bad. To perform the verification, we take a synchronous product of the system automaton and the property automaton A. Each state of the product is a pair of the form (s, q) where s is the system state and q is the state of A. There is a transition from (s, q) to (s', q') with probability p on input a iff i) there is a transition from s on input a to s' in the system automaton with probability p and ii) there is a transition on input s from q to q' in A. We convert this product automaton to a FPM $\mathcal{M}$, by merging all states of the form (s, q), where q is the error state of A, into a single absorbing state which is the rejecting state. Now, on every infinite input sequence the system satisfies the safety property specified by A with probability greater than x iff every input sequence is rejected by $\mathcal{M}$ with probability less than $1 - x$.

3. PRELIMINARIES

Sequences. Let S be a finite set. We let $|S|$ denote the cardinality of S. Let $\kappa = s_1, s_2, \ldots$ be a possibly infinite sequence over S. The length of κ, denoted as $|\kappa|$, is defined to be the number of elements in κ if κ is finite, and ω otherwise. S^* denotes the set of finite sequences, S^+ the set of finite sequences of length ≥ 1 and S^ω denotes the set of infinite sequences. If η is a finite sequence, and κ is either a finite or an infinite sequence then $\eta\kappa$ denotes the concatenation of the two sequences in that order. If $R \subseteq S^*$ and $R' \subseteq S^* \cup S^\omega$, the set $RR' = \{\eta\kappa \mid \eta \in R \text{ and } \eta \in R'\}$.

For integers i and j such that $1 \leq i \leq j < |\kappa|$, $\kappa[i : j]$ denotes the (finite) sequence $s_i, \ldots s_j$ and the element $\kappa(i)$ denotes the element s_i. A *finite prefix* of κ is any $\kappa[1 : j]$ for $j < |\kappa|$. We denote the set of κ's finite prefixes by $\mathit{Pref}(\kappa)$.

Languages. Given a finite alphabet Σ, a *language L of finite words* over Σ is a set of finite sequences over Σ and a *language $\mathcal{L}$ of infinite words* over Σ is a set of infinite sequences over Σ.

Metric topology on Σ^ω. Given a finite alphabet Σ, one can define a metric $d : \Sigma^\omega \times \Sigma^\omega \to \mathbb{R}^+$ on Σ^ω as follows. For $\alpha_1, \alpha_2 \in \Sigma^\omega, \alpha_1 \neq \alpha_2$, $d(\alpha_1, \alpha_2) = \frac{1}{2^j}$ where j is the unique integer such that $\alpha_1(j) \neq \alpha_2(j)$ and $\forall i < j, \alpha_1(i) = \alpha_2(i)$. Also $d(\alpha, \alpha) = 0$ for all $\alpha \in \Sigma^\omega$. Given $\alpha \in \Sigma^\omega$ and $\mathsf{r} \in \mathbb{R}$, the set $\mathsf{B}(\alpha, \mathsf{r}) = \{\beta \mid d(\alpha, \beta) < \mathsf{r}\}$ is said to be an *open ball* with center α and radius r. A language $\mathcal{L} \subseteq \Sigma^\omega$ is said to be *open* if for every $\alpha \in \mathcal{L}$ there is a r_α such that $\mathsf{B}(\alpha, \mathsf{r}_\alpha) \subseteq \mathcal{L}$. It can be shown that a language $\mathcal{L}$ is open iff $\mathcal{L} = L\Sigma^\omega$ for some $L \subseteq \Sigma^*$. A language $\mathcal{L}$ is *closed* if its complement $\Sigma^\omega \setminus \mathcal{L}$ is open. Given a language $\mathcal{L} \subseteq \Sigma^\omega$, the set $\mathsf{cl}(\mathcal{L}) = \{\alpha \mid \forall \mathsf{r}, \mathsf{B}(\alpha, \mathsf{r}) \cap \mathcal{L} \neq \emptyset\}$ is the smallest closed set containing $\mathcal{L}$.

Safety Languages. Given an alphabet Σ and a language $\mathcal{L} \subseteq \Sigma^\omega$, we denote the set of prefixes of $\mathcal{L}$ by $\mathit{Pref}(\mathcal{L})$, i.e., $\mathit{Pref}(\mathcal{L}) = \bigcup_{\alpha \in \mathcal{L}} \mathit{Pref}(\alpha)$. Following [Lamport 1985; Alpern and Schneider 1985], a language $\mathcal{L}$ is a *safety property* (also, called *safety* language) if for every $\alpha \in \Sigma^\omega$: $\mathit{Pref}(\alpha) \subseteq \mathit{Pref}(\mathcal{L}) \quad \Rightarrow \quad \alpha \in \mathcal{L}$. In other words, $\mathcal{L}$ is a safety property if it is *limit closed* – for every infinite string α, if every prefix of α is a prefix of some string in $\mathcal{L}$, then α itself is in $\mathcal{L}$. Safety languages coincide exactly with the closed languages in the metric topology d defined above [Perrin and Pin 2004]. We will denote the set of safety languages by Safety.

Almost Safety Languages. It is well known that the class of safety languages are closed under finite union and countable intersection [Sistla 1985], but not under countable unions. We say that a language $\mathcal{L}$ is an *almost safety* property/language if it is a countable union of safety languages, i.e., $\mathcal{L} := \cup_{0 \leq i < \infty} \mathcal{L}_i$ where $\mathcal{L}_i$, for $0 \leq i < \infty$, is a safety language. For $\mathcal{L}$, as given above, and $i \geq 0$, let $\mathcal{M}_i = \cup_{0 \leq j \leq i} \mathcal{L}_j$. It is easy to see that, for each $i \geq 0$, $\mathcal{M}_i$ is a safety language and $\mathcal{M}_i \subseteq \mathcal{M}_{i+1}$. Thus the languages $\mathcal{M}_0, ..., \mathcal{M}_i, ...$ form an increasing chain of safety languages with $\mathcal{L}$ as its limit. Thus, we see that, although an almost safety language is not a safety language, it nevertheless can be approximated more and more accurately by safety languages. Please note that the complement of an almost safety property can be written as a countable intersection of open sets of the metric topology d defined above. We will denote the set of almost safety languages by $\mathsf{AlmostSafety}$.

Automata and ω-regular Languages. A *Büchi automaton* $\mathcal{A}$ on infinite strings over a finite alphabet Σ is a 4-tuple (Q, Δ, q_0, F) where Q is a finite set of states, $\Delta \subseteq Q \times \Sigma \times Q$ is a transition relation, $q_0 \in Q$ is an initial state, and $F \subseteq Q$ is a set of accepting/final automaton states. If for every $q \in Q$ and $a \in \Sigma$, there is exactly one q' such that $(q, a, q') \in \Delta$ then $\mathcal{A}$ is called a deterministic automaton. Let $\alpha = a_1, \ldots$ be an infinite sequence over

Σ. A *run* r *of* $\mathcal{A}$ on α is an infinite sequence $r_0, r_1, \ldots$ over Q such that $r_0 = q_0$ and for every $i > 0$, $(r_{i-1}, a_i, r_i) \in \Delta$. The run r is *accepting* if some state in F appears infinitely often in r. The automaton $\mathcal{A}$ *accepts* the string α if it has an accepting run over α. The *language accepted (recognized) by* $\mathcal{A}$, denoted by $L(\mathcal{A})$, is the set of strings that $\mathcal{A}$ accepts. A language $\mathcal{L}$ is called ω*-regular* if it is accepted by some finite state Büchi automaton. We will denote the set of ω-regular languages by $\mathsf{Regular}$. $\mathsf{Regular}$ is closed under finite boolean operations.

A language $\mathcal{L} \in \mathsf{Regular} \cap \mathsf{AlmostSafety}$ iff there is a deterministic Büchi automaton which accepts its complement $\Sigma^\omega \setminus \mathcal{L}$ [Perrin and Pin 2004; Thomas 1990]. It is well-known that a language $\mathcal{L} \in \mathsf{Regular} \cap \mathsf{Safety}$ iff there is a deterministic Büchi automaton $\mathcal{A} = (Q, \Delta, q_0, \{q_r\})$ such that $\forall a \in \Sigma, (q_r, a, q_r) \in \Delta$ and $\mathcal{A}$ recognizes the complement $\Sigma^\omega \setminus \mathcal{L}$. A language $\mathcal{L} \in \mathsf{Regular} \cap \mathsf{Safety}$ iff there is a Büchi automaton $\mathcal{A}$ (not necessarily deterministic) such that each state of $\mathcal{A}$ is a final state [Perrin and Pin 2004]. This implies that a language $\mathcal{L} \in \mathsf{Safety}$ is ω-regular iff the set of finite prefixes of $\mathcal{L}$, $\mathit{Pref}(\mathcal{L}) \subseteq \Sigma^*$, is a regular language (here $\mathit{Pref}(\mathcal{L})$ is a language of finite words).

Probability Spaces. Let V be a set and E be a set of subsets of V. We say that E is a σ-algebra on V if E contains the empty set, is closed under complementation and also under finite as well as countable unions. Let F be a set of subsets of V. A σ-algebra generated by F is the smallest σ-algebra that contains F. A probability space is a triple (V, E, μ) where E is a σ-algebra on V and μ is a probability function [Papoulis and Pillai 2002] on E.

4. FINITE STATE PROBABILISTIC MONITORS

We will now define finite state probabilistic monitors which can be viewed as probabilistic automata over infinite strings that have a special *reject state*. The transition relation from a state on a given input is described as a probability distribution that determines the probability of transitioning to that state. The transition relation ensures that the probability of transitioning from the reject state to a non-reject state is 0. A FPM can thus be seen as a generalization of deterministic finite state monitors where the deterministic transition is replaced by a probabilistic transition. Alternately, it can be viewed as the restriction of probabilistic monitors [Sistla and Srinivas 2008] to finite memory. From an automata-theoretic viewpoint, they are special cases of probabilistic Büchi automata described in [Baier and Gröβer 2005] which generalized the probabilistic finite automata [Rabin 1963; Salomaa 1973; Paz 1971] on finite words to infinite words. The main difference here is that instead of having a set of accepting states, we have one (absorbing) reject state. Formally,

Definition: A *finite state probabilistic monitor (*FPM*)* over a finite alphabet Σ is a tuple $\mathcal{M} = (Q, q_s, q_r, \delta)$ where Q is a finite set of states; $q_s \in Q$ is the initial state; $q_r \in Q$ is the reject state, and; $\delta : Q \times \Sigma \times Q \to [0, 1]$ is the probabilistic transition relation such that for all $q \in Q$ and $a \in \Sigma$, $\sum_{q' \in Q} \delta(q, a, q') = 1$ and $\delta(q_r, a, q_r) = 1$. In addition, if $\delta(q, a, q')$ is a rational number for all $q, q' \in Q, a \in \Sigma$, then we say that $\mathcal{M}$ is a rational finite state probabilistic monitor (RatFPM).

It will be useful to view the transition function δ of a FPM on an input a as a matrix δ_a with the rows labeled by "current" state and columns labeled by "next" state and the entry $\delta_a(q, q')$ denoting the probability of transitioning from q to q'. Formally,

Notation: Given a FPM, $\mathcal{M} = (Q, q_s, q_r, \delta)$ over Σ, and $a \in \Sigma$, δ_a is a square matrix of order $|Q|$ such that $\delta_a(q, q') = \delta(q, a, q')$. Given a word $u = a_1 a_2 \ldots a_n \in \Sigma^+$, δ_u is the matrix product $\delta_{a_1}\delta_{a_2}\ldots\delta_{a_n}$. Please note that δ_ϵ is not defined. However, we shall sometimes abuse notation and say that $\delta_\epsilon(q_1, q_2)$ is 1 if $q_1 = q_2$ and 0 otherwise. For $Q_1 \subseteq Q$, we say $\delta_u(q, Q_1) = \sum_{q_1 \in Q_1} \delta_u(q, q_1)$.

Intuitively, the matrix entry $\delta_u(q, q')$ denotes the probability of being in q' after having read the input word u and having started in q. Please note that $\sum_{q' \in Q} \delta_u(q, q') = 1$ for all $u \in \Sigma^+$ and $q, q' \in Q$.

The behavior of a FPM $\mathcal{M}$ on an input word $\alpha = a_1 a_2, \ldots, .$ can be described as follows. The FPM starts in the initial state q_s and if reading input symbols $a_1 a_2 a_3 \ldots a_i$ results in state q, then it moves to state q' with probability $\delta_{a_{i+1}}(q, q')$ on symbol a_{i+1}. An infinite sequence of states, $\rho \in Q^\omega$, is a *run* of the FPM $\mathcal{M}$. We say that a run ρ is *rejecting* if the reject state occurs infinitely often in ρ. A run ρ is said to be *accepting* if the run is not a rejecting run. In order to determine the probability of rejecting the word α, the FPM $\mathcal{M}$ can be thought of as a infinite state Markov chain which gives rise to the standard probability measure on Markov Chains [Vardi 1985; Kemeny and Snell 1976]:

Definition: Given a FPM, $\mathcal{M} = (Q, q_s, q_r, \delta)$ on the alphabet Σ and a word $\alpha \in \Sigma^\omega$, the *probability space generated* by $\mathcal{M}$ and α is the probability space $(Q^\omega, \mathcal{F}_{\mathcal{M},\alpha}, \mu_{\mathcal{M},\alpha})$ where

—$\mathcal{F}_{\mathcal{M},\alpha}$ is the smallest σ-algebra on Q^ω generated by the collection $\{\mathsf{C}_\eta \mid \eta \in Q^+\}$ where $\mathsf{C}_\eta = \{\rho \in Q^\omega \mid \eta \text{ is a prefix of } \rho\}$.

—$\mu_{\mathcal{M},\alpha}$ is the unique probability measure on $(Q^\omega, \mathcal{F}_{\mathcal{M},\alpha})$ such that $\mu_{\mathcal{M},\alpha}(\mathsf{C}_{q_0 \ldots q_n})$ is
  —0 if $q_0 \neq q_s$,
  —1 if $n = 0$ and $q_0 = q_s$, and
  —$\delta(q_0, \alpha(1), q_1) \ldots \delta(q_{n-1}, \alpha(n), q_n)$ otherwise.

The set of rejecting runs and the set of accepting runs for a given word α can be shown to be measurable which gives rise to the following definition.

Definition: Let $\mu_{\mathcal{M},\alpha}$ be the probability measure defined by the FPM, $\mathcal{M} = (Q, q_s, q_r, \delta)$ on Σ and the word $\alpha \in \Sigma^\omega$. Let $\mathsf{rej} = \{\rho \in Q^\omega \mid q_r \text{ occurs infinitely often in } \rho\}$ and $\mathsf{acc} = Q^\omega \setminus \mathsf{rej}$. The quantity $\mu_{\mathcal{M},\alpha}(\mathsf{rej})$ is said to be the *probability of rejecting* α and will be denoted by $\mu^{rej}_{\mathcal{M},\,\alpha}$. The quantity $\mu_{\mathcal{M},\alpha}(\mathsf{acc})$ is said to be the *probability of accepting* α and will be denoted by $\mu^{acc}_{\mathcal{M},\,\alpha}$. We have that $\mu^{rej}_{\mathcal{M},\,\alpha} + \mu^{acc}_{\mathcal{M},\,\alpha} = 1$.

Please note that as the probability of transitioning from a reject state to a non-reject state is 0, the probability of rejecting an infinite word can be seen as a limit of the probability of rejecting its finite prefixes. This is the content of the following Lemma.

LEMMA 4.1. *For any* FPM, $\mathcal{M} = (Q, q_s, q_r, \delta)$ *over* Σ, *and any word* $\alpha = a_1 a_2 \ldots \ldots \in \Sigma^\omega$, *the sequence of reals numbers* $\{\delta_{a_1 a_2 \ldots a_n}(q_s, q_r) \mid n \in \mathbb{N}\}$ *is an non-decreasing sequence. Furthermore,* $\mu^{rej}_{\mathcal{M},\,\alpha} = \lim_{n \to \infty} \delta_{a_1 a_2 \ldots a_n}(q_s, q_r)$.

The rest of this section is organized as follows. We first present some special monitors and technical constructions involving FPMs. We then conclude this section by introducing the class of monitorable languages that we consider in this paper.

4.1 Some Tools and Techniques

The special monitors and monitor constructions that we present in this section, will be used both in the definition of the classes of monitorable languages, as well as in establishing the expressiveness results in Section 5. We will also generalize the iteration lemma for probabilistic automata over finite words to FPMs.

Scaling acceptance/rejection probabilities. Given a FPM $\mathcal{M}$ and a real number $x \in [0,1]$, we can construct monitors, where the acceptance (or rejection) probability of a word is scaled by a factor x. We begin by proving such a proposition for acceptance probabilities, before turning our attention to rejection probabilities.

PROPOSITION 4.2. *Given a* FPM, *$\mathcal{M}$ on Σ, and a real number $x \in [0,1]$, there is a* FPM, *$\mathcal{M}_x$, such that for any $\alpha \in \Sigma^\omega$, $\mu^{acc}_{\mathcal{M}_x,\,\alpha} = x \times \mu^{acc}_{\mathcal{M},\,\alpha}$.*

PROOF. Let $\mathcal{M} = (Q, q_s, q_r, \delta)$. Pick a new state q_{s_0} not in Q and let $\mathcal{M}_x = (Q \cup \{q_{s_0}\}, q_{s_0}, q_r, \delta_x)$ where the transition function δ_x is defined as follows. For each $a \in \Sigma$

—$\delta_x(q,a,q') = \delta(q,a,q')$ if $q, q' \in Q$.
—$\delta_x(q_{s_0},a,q') = x \times \delta(q_s,a,q')$ if $q' \in Q \setminus \{q_r\}$.
—$\delta_x(q_{s_0},a,q_r) = 1 - x + x \times \delta(q_s,a,q_r)$.
—$\delta_x(q,a,q') = 0$ if $q' = q_{s_0}$.

Now, for any $u \in \Sigma^*$, we can show by induction that $\delta_x(q_{s_0},u,q_r) = (1-x) + x \times \delta(q_s,u,q_r)$ and $\delta_x(q_{s_0},u,q) = x \times \delta(q_s,u,q)$ for $q \neq q_r$. It is easy to see that $\mathcal{M}_x$ is the desired FPM. □

Similarly we can construct a FPM $\mathcal{M}^x$ which scales the probability of rejecting any word by a factor of x.

PROPOSITION 4.3. *Given a* FPM, *$\mathcal{M}$ on Σ, and a real number $x \in [0,1]$, there is a* FPM, *$\mathcal{M}^x$, such that for any $\alpha \in \Sigma^\omega$, $\mu^{rej}_{\mathcal{M}^x,\,\alpha} = x \times \mu^{rej}_{\mathcal{M},\,\alpha}$.*

PROOF. Let $\mathcal{M} = (Q, q_s, q_r, \delta)$. Pick two new states q_{r_0} and q_{r_1} not occurring in Q and let $\mathcal{M}^x = (Q \cup \{q_{r_0}, q_{r_1}\}, q_s, q_{r_0}, \delta^x)$ where δ^x is defined as follows. For each $a \in \Sigma$

—$\delta^x(q,a,q') = \delta(q,a,q')$ if $q, q' \in Q \setminus \{q_r\}$.
—$\delta^x(q_r,a,q_{r_0}) = x$.
—$\delta^x(q_r,a,q_{r_1}) = 1 - x$.
—$\delta^x(q,a,q) = 1$ if $q \in \{q_{r_0}, q_{r_1}\}$.
—$\delta^x(q,a,q') = 0$ if none of the above hold.

Now, for any $u \in \Sigma^*$ and $a \in \Sigma$, we can show that $\delta^x(q_s,ua,q_{r_0}) = x \times \delta(q_s,u,q_r)$. Using Lemma 4.1, we get the desired result. □

We get as an immediate consequence of Proposition 4.2 and Proposition 4.3.

PROPOSITION 4.4. *Given a* FPM *$\mathcal{M}$ on Σ and a real number $x \in [0,1]$, there are* FPM*'s $\mathcal{M}_x$ and $\mathcal{M}^x$ such that for any $\alpha \in \Sigma^\omega$, $\mu^{acc}_{\mathcal{M}_x,\,\alpha} = x \times \mu^{acc}_{\mathcal{M},\,\alpha}$ and $\mu^{rej}_{\mathcal{M}^x,\,\alpha} = x \times \mu^{rej}_{\mathcal{M},\,\alpha}$.*

Product automata. Given two FPM's, $\mathcal{M}_1$ and $\mathcal{M}_2$, we can construct a new FPM $\mathcal{M}$ such that the probability that $\mathcal{M}$ accepts a word α is the product of the probabilities that $\mathcal{M}_1$ and $\mathcal{M}_2$ accept the same word α.

PROPOSITION 4.5. *Given two* FPM, *$\mathcal{M}_1$ and $\mathcal{M}_2$ on Σ, there is a* FPM, *$\mathcal{M}_1 \otimes \mathcal{M}_2$ such that for any $\alpha \in \Sigma^\omega$, $\mu^{acc}_{\mathcal{M}_1 \otimes \mathcal{M}_2, \alpha} = \mu^{acc}_{\mathcal{M}_1, \alpha} \times \mu^{acc}_{\mathcal{M}_2, \alpha}$.*

PROOF. Let $\mathcal{M}_1 = (Q_1, q_{s_1}, q_{r_1}, \delta_1)$ and $\mathcal{M}_2 = (Q_2, q_{s_2}, q_{r_2}, \delta_2)$. Pick a new state q_r not occurring in $Q_1 \cup Q_2$. Let $\mathcal{M}_1 \otimes \mathcal{M}_2 = (Q, q_s, q_r, \delta)$ where

—$Q = \{q_r\} \cup ((Q_1 \setminus \{q_{r_1}\}) \times (Q_2 \setminus \{q_{r_2}\}))$

—$q_s = (q_{s_1}, q_{s_2})$

—For each $a \in \Sigma$,
  —$\delta((q_1, q_2), a, (q_1', q_2')) = \delta_1(q_1, a, q_1')\delta_2(q_2, a, q_2')$.
  —$\delta((q_1, q_2), a, q_r) = 1 - \sum_{q \neq q_r} \delta((q_1, q_2), a, q)$.
  —$\delta(q_r, a, q_r) = 1$ and $\delta(q_r, a, q) = 0$ for $q \neq q_r$.

Given any finite word $u \in \Sigma^+$, we can shown by induction on the length of u that for any $q_1, q_1' \in Q_1 \setminus \{q_{r_1}\}$ and $q_2, q_2' \in Q_2 \setminus \{q_{r_2}\}$, we have $\delta_u((q_1, q_2), (q_1', q_2')) = \delta_1(q_1, u, q_1') \times \delta_2(q_2, u, q_2')$.

Now, given a $\alpha = a_0 a_1 \ldots$, let $u_j = a_1 \ldots a_j$. Using Lemma 4.1, we get

$$\begin{aligned}
\mu^{acc}_{\mathcal{M}_1 \otimes \mathcal{M}_2, \alpha} &= 1 - \mu^{rej}_{\mathcal{M}_1 \otimes \mathcal{M}_2, \alpha} \\
&= 1 - \lim_{j \to \infty} \delta_{u_j}(q_s, q_r) \\
&= \lim_{j \to \infty} (1 - \delta_{u_j}(q_s, q_r)) \\
&= \lim_{j \to \infty} (\textstyle\sum_{q \neq q_r} \delta_{u_j}(q_s, q)) \\
&= \lim_{j \to \infty} \textstyle\sum_{q_1' \neq q_{r_1}} \sum_{q_2' \neq q_{r_2}} \\
&\qquad (\delta_1(q_{s_1}, u_j, q_1')\delta_2(q_{s_2}, u_j, q_2')) \\
&= \lim_{j \to \infty} ((\textstyle\sum_{q_1' \neq q_{r_1}} \delta_1(q_{s_1}, u_j, q_1')) \\
&\qquad \times (\textstyle\sum_{q_2' \neq q_{r_2}} \delta_2(q_{s_2}, u_j, q_2')) \\
&= \lim_{j \to \infty} ((1 - \delta_1(q_{s_1}, u_j, q_{r_1})) \\
&\qquad \times (1 - \delta_2(q_{s_2}, u_j, q_{r_2}))) \\
&= \mu^{acc}_{\mathcal{M}_1, \alpha} \times \mu^{acc}_{\mathcal{M}_2, \alpha}. \quad \square
\end{aligned}$$

Given two FPM's, $\mathcal{M}_1$ and $\mathcal{M}_2$, we can construct a new FPM $\mathcal{M}$ such that the probability that $\mathcal{M}$ rejects a word α is the product of the probabilities that $\mathcal{M}_1$ and $\mathcal{M}_2$ reject the same word α.

PROPOSITION 4.6. *Given two* FPM, *$\mathcal{M}_1$ and $\mathcal{M}_2$ on Σ, there is a* FPM, *$\mathcal{M}_1 \times \mathcal{M}_2$ such that for any $\alpha \in \Sigma^\omega$, $\mu^{rej}_{\mathcal{M}_1 \times \mathcal{M}_2, \alpha} = \mu^{rej}_{\mathcal{M}_1, \alpha} \times \mu^{rej}_{\mathcal{M}_2, \alpha}$.*

PROOF. Let $\mathcal{M}_1 = (Q_1, q_{s_1}, q_{r_1}, \delta_1)$ and $\mathcal{M}_2 = (Q_2, q_{s_2}, q_{r_2}, \delta_2)$. Let $\mathcal{M}_1 \times \mathcal{M}_2 = (Q, q_s, q_r, \delta)$ where

—$Q = Q_1 \times Q_2$

—$q_s = (q_{s_1}, q_{s_2})$

—$q_r = (q_{r_1}, q_{r_2})$

—For each $a \in \Sigma$, $\delta((q_1, q_2), a, (q_1', q_2')) = \delta_1(q_1, a, q_1')\delta_2(q_2, a, q_2')$.

Given any finite word $u \in \Sigma^+$, we can show by induction on the length of u that for any $q_1, q_1' \in Q_1$ and $q_2, q_2' \in Q_2$, we have $\delta_u((q_1, q_2), (q_1', q_2')) = \delta_1(q_1, u, q_1') \times \delta_2(q_2, u, q_2')$. The result now follows from Lemma 4.1. $\square$

Iteration Lemma. We present herein a technical lemma that will be used to demonstrate that certain languages are not recognized by probabilistic monitors. Please note that the iteration lemma is a generalization of a similar lemma for probabilistic automata over finite words [Nasu and Honda 1968; Paz 1971] and the generalization relies on the fact that the probability of rejection of an infinite word can be taken as a limit of probability of rejection of its finite prefixes (see Lemma 4.1).

LEMMA 4.7. *Given a* FPM, $\mathcal{M}$ *on* Σ, *and a finite word* $u \in \Sigma^+$, *there is a natural number* k *and real numbers* $c_0, \ldots c_{k-1} \in \mathbb{R}$ *(depending only upon* $\mathcal{M}$ *and* u*) such that for all* $v \in \Sigma^+, \alpha \in \Sigma^\omega$,

$$\mu^{rej}_{\mathcal{M},\, vu^k\alpha} = c_{k-1}\mu^{rej}_{\mathcal{M},\, vu^{k-1}\alpha} + \ldots + c_0\mu^{rej}_{\mathcal{M},\, v\alpha}$$

and $c_{k-1} + \ldots + c_0 = 1$.

PROOF. Let $\mathcal{M} = (Q, q_s, q_r, \delta)$. Consider the matrix δ_u. From elementary linear algebra there is a polynomial $p(x) = x^k - c_{k-1}x^{k-1} - c_{k-2}x^{k-2} - \ldots - c_0$ (*the minimal polynomial* of δ_u) such that

(1) $p(\delta_u) = 0$ and

(2) $p(\lambda) = 0$ where λ is an eigenvalue of δ_u.

Since $p(\delta_u) = 0$, we get $\delta_u^k = c_{k-1}\delta_u^{k-1} + \ldots c_0\mathbf{I}$ where $\mathbf{I}$ is the identity matrix. Now if $\alpha = a_1a_2\ldots$ then we get for each $n > 0$,

$$\delta_v\delta_u^k\delta_{a_1\ldots a_n} = c_{k-1}\delta_v\delta_u^{k-1}\delta_{a_1\ldots a_n} + \ldots + c_0\delta_v\delta_{a_1\ldots a_n}.$$

This implies that

$$\delta_{vu^k a_1\ldots a_n} = c_{k-1}\delta_{vu^{k-1}a_1\ldots a_n} + \ldots + c_0\delta_{va_1\ldots a_n}.$$

Thus,

$$\delta_{vu^k a_1\ldots a_n}(q_s, q_r) = c_{k-1}\delta_{vu^{k-1}a_1\ldots a_n}(q_s, q_r) + \ldots + c_0\delta_{va_1\ldots a_n}(q_s, q_r).$$

Taking the limit ($n \to \infty$) on both sides, we get by Lemma 4.1

$$\mu^{rej}_{\mathcal{M},\, vu^k\alpha} = c_{k-1}\mu^{rej}_{\mathcal{M},\, vu^{k-1}\alpha} + \ldots + c_0\mu^{rej}_{\mathcal{M},\, v\alpha}.$$

Now, please note that 1 is an eigenvalue of δ_u (with an eigenvector all of whose entries are 1). Thus, we get $p(1) = 0$ which implies that $c_{k-1} + \ldots + c_0 = 1$. $\square$

The unit interval as the acceptance (or rejection) probability of a monitor. We will now define two monitors that will be used for proving expressiveness results. These monitors will be defined on the alphabet $\Sigma = \{\mathbf{0}, \mathbf{1}\}$. By associating the numeral 0 to $\mathbf{0}$ and associating the numeral 1 to $\mathbf{1}$, we can associate $\alpha = a_1a_2\ldots \in \Sigma$ to a real number $\mathsf{bin}(\alpha)$ by thinking of α as the "binary" real number $0.a_1a_2\ldots$. Formally,

Definition: Let $\Sigma = \{\mathbf{0}, \mathbf{1}\}$, $\mathsf{bin}(\mathbf{0}) = 0$ and $\mathsf{bin}(\mathbf{1}) = 1$. We define the functions $\mathsf{bin}(\cdot) : \Sigma^+ \to [0,1]$ as $\mathsf{bin}(a_1a_2\ldots a_k) = \sum_{j=1}^{j=k} \frac{a_j}{2^j}$, and $\mathsf{bin}(\cdot) : \Sigma^\omega \to [0,1]$ as $\mathsf{bin}(a_1a_2\ldots) =$

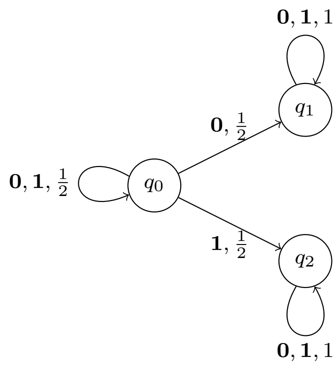

Fig. 3. Transition diagram for FPMs $\mathcal{M}_{\mathsf{Id}}$ and $\mathcal{M}_{1-\mathsf{Id}}$

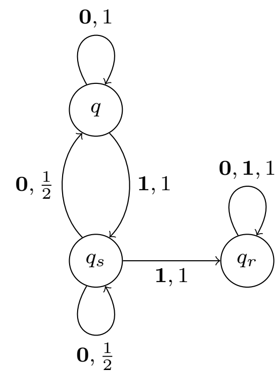

Fig. 4. FPM $\mathcal{M}$ used in proof of Theorem 5.8

$\sum_{j=1}^{j=\infty} \frac{a_j}{2^j}$. If $x \in [0,1]$ is an irrational number, let $\mathsf{wrd}(x) \in \Sigma^\omega$ be the unique word such that $\mathsf{bin}(\mathsf{wrd}(x)) = x$.

The following proposition will be useful.

PROPOSITION 4.8. *Let* $\Sigma = \{\mathbf{0}, \mathbf{1}\}$ *and let* $x \in [0,1]$ *be an irrational number. Then the languages* $\mathcal{L}_x = \{\alpha \in \Sigma^\omega \mid \mathsf{bin}(\alpha) \leq x\}$ *and* $\mathcal{L}^x = \{\alpha \in \Sigma^\omega \mid \mathsf{bin}(\alpha) < x\}$ *are not* ω*-regular.*

PROOF. We will just show that $\mathcal{L}_x$ is not ω-regular. The proof of non-ω-regularity of $\mathcal{L}^x$ is similar.

Please note that the language $\mathcal{L}_x$ defines an equivalence relation (the Myhill-Nerode equivalence) on Σ^*– $u \equiv_{\mathcal{L}_x} v$ iff for all $\alpha \in \Sigma^\omega$, $u\alpha \in \mathcal{L}_x \Leftrightarrow v\alpha \in \mathcal{L}_x$. If $\mathcal{L}_x$ is ω-regular, then there should be only a finite number of $\equiv_{\mathcal{L}_x}$ classes (see [Thomas 1990]). We will show that this is not the case.

By confusing $\mathbf{0}$ with the numeral 0 and $\mathbf{1}$ with the numeral 1, consider the binary expansion of $x = .a_1a_2\ldots$. Let $\mathsf{wrd}(x) = a_1a_2\ldots$ and for each $i > 0$, x_i be the suffix $a_ia_{i+1}\ldots$. Since x is irrational, no two suffixes x_i and x_j for $i < j$ can be the same infinite word.

So given $i < j$, let $u_i = a_1a_2\ldots a_i$ and $u_j = a_1a_2\ldots a_j$. Let k be the smallest natural number such that $a_{i+k} \neq a_{j+k}$. Let $\beta = a_{i+1}a_{i+2}\ldots a_{i+k-1}\mathbf{1}\mathbf{0}^\omega$. Please note that $\beta = a_{j+1}a_{j+2}\ldots a_{j+k-1}\mathbf{1}\mathbf{0}^\omega$. There are two cases: either $a_{i+k} = \mathbf{1}$ and $a_{j+k} = \mathbf{0}$, or $a_{i+k} = \mathbf{0}$ and $a_{j+k} = \mathbf{1}$. If $a_{i+k} = \mathbf{1}$ and $a_{j+k} = \mathbf{0}$, it can be easily shown that $\mathsf{bin}(u_i\beta) < x$ while $\mathsf{bin}(u_j\beta) > x$. If $a_{i+k} = \mathbf{0}$ and $a_{j+k} = \mathbf{1}$, $\mathsf{bin}(u_i\beta) > x$ while $\mathsf{bin}(u_j\beta) < x$. Hence $u_i \not\equiv_{\mathcal{L}_x} u_j$ for $i < j$. Thus, $\mathcal{L}_x$ is not ω-regular. □

We point out here that the proof of non-ω-regularity of $\mathcal{L}^x$ and $\mathcal{L}_x$ is a modification of the proof that the language of finite words $\{u \in \Sigma^* \mid \mathsf{bin}(u) < x\}$ is not regular (see [Paz 1971; Salomaa 1973]).

We can construct two monitors whose probability of accepting a word α is $\mathsf{bin}(\alpha)$ and $1 - \mathsf{bin}(\alpha)$ respectively as follows.

LEMMA 4.9. *Let $\Sigma = \{\mathbf{0}, \mathbf{1}\}$. There are $RatFPM$'s, $\mathcal{M}_{\mathsf{Id}}$ and $\mathcal{M}_{1-\mathsf{Id}}$ on Σ, such that for any word $\alpha \in \Sigma^\omega$,*

$$\mu^{acc}_{\mathcal{M}_{\mathsf{Id}},\,\alpha} = \mathsf{bin}(\alpha) \text{ and } \mu^{acc}_{\mathcal{M}_{1-\mathsf{Id}},\,\alpha} = 1 - \mathsf{bin}(\alpha).$$

PROOF. The transition diagrams of the FPMs $\mathcal{M}_{\mathsf{Id}}$ and $\mathcal{M}_{1-\mathsf{Id}}$ are shown in Figure 3. Let $Q = \{q_0, q_1, q_2\}$ and $\delta : Q \times \Sigma \times Q \to [0, 1]$ be defined as follows. The states q_1 and q_2 are absorbing, *i.e.*, $\delta(q_1, \mathbf{0}, q_1) = \delta(q_1, \mathbf{1}, q_1) = \delta(q_2, \mathbf{0}, q_2) = \delta(q_2, \mathbf{1}, q_2) = 1$. For transitions out of q_0, $\delta(q_0, \mathbf{0}, q_0) = \delta(q_0, \mathbf{0}, q_1) = \delta(q_0, \mathbf{1}, q_0) = \delta(q_0, \mathbf{1}, q_2) = \frac{1}{2}$ Let $\mathcal{M}_{\mathsf{Id}} = (Q, q_0, q_1, \delta)$ and $\mathcal{M}_{1-\mathsf{Id}} = (Q, q_0, q_2, \delta)$.

Given $u \in \Sigma^+$, it can be easily shown by induction (on the length of u) that $\delta_u(q_0, q_0) = \frac{1}{2^{|u|}}$, $\delta_u(q_0, q_2) = \mathsf{bin}(u)$ and $\delta_u(q_0, q_1) = 1 - \mathsf{bin}(u) - \frac{1}{2^{|u|}}$. Using Lemma 4.1, it can easily shown that for any word $\alpha \in \Sigma^\omega$, $\mu^{acc}_{\mathcal{M}_{\mathsf{Id}},\,\alpha} = 1 - \mu^{rej}_{\mathcal{M}_{\mathsf{Id}},\,\alpha} = \mathsf{bin}(\alpha)$ and $\mu^{acc}_{\mathcal{M}_{1-\mathsf{Id}},\,\alpha} = 1 - \mu^{rej}_{\mathcal{M}_{1-\mathsf{Id}},\,\alpha} = 1 - \mathsf{bin}(\alpha)$. $\square$

We point out here that if we consider the reject state as an accept state and view the monitor $\mathcal{M}_{1-\mathsf{Id}}$ as a probabilistic automaton over finite words, the resulting automaton is the probabilistic automaton (see [Salomaa 1973]) often used to show that non-regular languages are accepted by probabilistic automaton.

4.2 Monitored languages

We conclude this section, by defining the classes of probabilistic monitors that we will consider in this paper. Recall that in the case of deterministic monitors, the language rejected is defined as the set of words upon which the monitor reaches the "unique" reject state. The language permitted by the monitor is defined as the complement of the words rejected. For probabilistic monitoring, on the other hand, the set of languages permitted may depend upon the probability of rejecting a word. In other words, it is reasonable to think of a language permitted by a FPM to be the set of words which are rejected with a probability strictly less than (or just less than) a threshold x. This gives rise to the following definition.

Definition: Given a FPM $\mathcal{M}$ on Σ and $x \in [0, 1]$,

—$\mathcal{R}_{<x}(\mathcal{M}) = \{\alpha \in \Sigma^\omega \mid \mu^{rej}_{\mathcal{M},\,\alpha} < x\}$.

—$\mathcal{R}_{\leq x}(\mathcal{M}) = \{\alpha \in \Sigma^\omega \mid \mu^{rej}_{\mathcal{M},\,\alpha} \leq x\}$.

Thus potentially, given $x \in [0, 1]$, we can define two subclasses of languages over an alphabet Σ— $\{\mathcal{L} \mid \exists \mathcal{M} \text{ s.t. } \mathcal{L} = \mathcal{R}_{<x}(\mathcal{M})\}$ and $\{\mathcal{L} \mid \exists \mathcal{M} \text{ s.t. } \mathcal{L} = \mathcal{R}_{\leq x}(\mathcal{M})\}$. It turns out that as long as x is strictly between 0 and 1, the exact value of x does not affect the classes defined.

LEMMA 4.10. *Let $\mathcal{M}$ be a* FPM *on an alphabet Σ. Then, given $x_1, x_2 \in (0, 1)$, there is a $\mathcal{M}'$ such that $\mathcal{R}_{\leq x_1}(\mathcal{M}) = \mathcal{R}_{\leq x_2}(\mathcal{M}')$ and $\mathcal{R}_{<x_1}(\mathcal{M}) = \mathcal{R}_{<x_2}(\mathcal{M}')$.*

PROOF. First consider the case $x_2 \leq x_1$. Let $x_3 = \frac{x_2}{x_1}$. By Proposition 4.4, there is a FPM $\mathcal{M}^{x_3}$ such that for all $\alpha \in \Sigma$ such that $\mu^{rej}_{\mathcal{M}^{x_3},\,\alpha} = x_3 \times \mu^{rej}_{\mathcal{M},\,\alpha}$. An easy calculation shows that $\mathcal{R}_{\leq x_2}(\mathcal{M}^{x_3}) = \mathcal{R}_{\leq x_1}(\mathcal{M})$ and $\mathcal{R}_{<x_2}(\mathcal{M}^{x_3}) = \mathcal{R}_{<x_1}(\mathcal{M})$.

Now, consider the case $x_1 \leq x_2$. Let $x_3 = \frac{1-x_2}{1-x_1}$. By Proposition 4.4, there is a FPM

$\mathcal{M}_{x_3}$ such that for all $\alpha \in \Sigma$, $\mu^{acc}_{\mathcal{M}_{x_3},\alpha} = x_3 \times \mu^{acc}_{\mathcal{M},\alpha}$. Now for any word α,

$$\begin{aligned}
\mu^{rej}_{\mathcal{M}_{x_3},\alpha} \leq x_2 &\Leftrightarrow 1 - x_2 \leq 1 - \mu^{rej}_{\mathcal{M}_{x_3},\alpha} \\
&\Leftrightarrow 1 - x_2 \leq \mu^{acc}_{\mathcal{M}_{x_3},\alpha} \\
&\Leftrightarrow 1 - x_2 \leq x_3 \times \mu^{acc}_{\mathcal{M},\alpha} \\
&\Leftrightarrow 1 - x_1 \leq \mu^{acc}_{\mathcal{M},\alpha} \\
&\Leftrightarrow 1 - \mu^{acc}_{\mathcal{M},\alpha} \leq x_1 \\
&\Leftrightarrow \mu^{rej}_{\mathcal{M},\alpha} \leq x_1.
\end{aligned}$$

Hence, $\mathcal{R}_{\leq x_2}(\mathcal{M}_{x_3}) = \mathcal{R}_{\leq x_1}(\mathcal{M})$. Similarly, $\mathcal{R}_{<x_2}(\mathcal{M}_{x_3}) = \mathcal{R}_{<x_1}(\mathcal{M})$. $\square$

Please note that $\mathcal{R}_{<0}(\mathcal{M}) = \emptyset$ and $\mathcal{R}_{\leq 1}(\mathcal{M}) = \Sigma^\omega$ for any FPM. Hence, we restrict our attention to four classes of languages as follows.

Definition: Given an alphabet Σ and a language $\mathcal{L} \subseteq \Sigma^\omega$.

—$\mathcal{L}$ is said to be *monitorable with strong acceptance* if there is a monitor $\mathcal{M}$ such that $\mathcal{L} = \mathcal{R}_{\leq 0}(\mathcal{M})$. We will denote the class of such properties by MSA.

—$\mathcal{L}$ is said to be *monitorable with weak acceptance* if there is a monitor $\mathcal{M}$ such that $\mathcal{L} = \mathcal{R}_{<1}(\mathcal{M})$. We will denote the class of such properties by MWA.

—$\mathcal{L}$ is said to be *monitorable with strict cut-point* if there is a monitor $\mathcal{M}$ and $0 < x < 1$ such that $\mathcal{L} = \mathcal{R}_{<x}(\mathcal{M})$. Such properties will be denoted by MSC.

—$\mathcal{L}$ is said to be *monitorable with non-strict cut-point* if there is a monitor $\mathcal{M}$ and $0 < x < 1$ such that $\mathcal{L} = \mathcal{R}_{\leq x}(\mathcal{M})$. Such properties will be denoted by MNC.

We point out here that in the literature on probabilistic automata over finite words [Rabin 1963; Paz 1971; Salomaa 1973], there is usually no distinction between strict and non-strict inequality. Also, if we were to define the classes MWA and MSA for probabilistic automata over finite words, they will turn out to be subclasses of regular languages. As we will see, over infinite words, while the class of languages MSA coincides with ω-regular and safety languages, the class MWA may contain non-ω-regular languages.

5. EXPRESSIVENESS

In this Section, we will compare the relative expressiveness of the class of Languages MSA, MWA, MSC and MNC defined in Section 4.2. For this Section, we will assume that the alphabet Σ is fixed unless otherwise stated. We will also assume that Σ contains at least 2 elements (if Σ contains only one element, then Σ^ω consists of exactly only one element). We summarize our results in Figure 5 below. The rest of this section is organized as follows. We begin by establishing the results for the classes MSA and MNC. We then consider the classes MWA and MSC. We conclude the section by proving the results for robust monitors.

5.1 Monitored Languages MSA and MNC.

Please recall that given an alphabet Σ, MSA$= \{\mathcal{L} \subseteq \Sigma^\omega \mid \exists$ FPM $\mathcal{M}$ s.t. $\mathcal{L} = \mathcal{R}_{\leq 0}(\mathcal{M})\}$ and MNC$= \{\mathcal{L} \subseteq \Sigma^\omega \mid \exists$ FPM $\mathcal{M}$ and $x \in (0,1)$ s.t. $\mathcal{L} = \mathcal{R}_{\leq x}(\mathcal{M})\}$. We start by showing that both MSA and MNC are subclasses of safety languages.

THEOREM 5.1. MSA,MNC $\subseteq$ Safety.

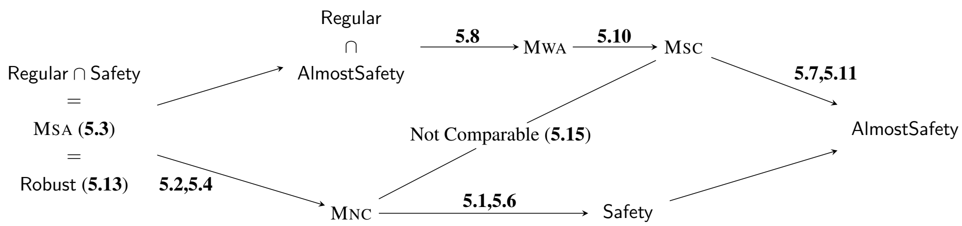

Fig. 5. An arrow from class $\mathbf{A}$ to $\mathbf{B}$ indicates the strict containment of class $\mathbf{A}$ in $\mathbf{B}$. The numbers on the arrows refer to the theorem in the paper that establishes this relationship.

PROOF. Let $\mathcal{M} = (Q, q_r, q_s, \delta)$ be a FPM on an alphabet Σ and let $0 \le x < 1$. It suffices to show that the set $\mathcal{L} = \Sigma^\omega \setminus \mathcal{R}_{\le x}(\mathcal{M})$ is an open set. Pick any word $\alpha = a_1 a_2 \ldots \in \mathcal{L}$. Then, we must have by definition and Lemma 4.1, $\mu^{rej}_{\mathcal{M},\,\alpha} = \lim_{n\to\infty} \delta_{a_1 \ldots a_n}(q_s, q_r) > x$. Hence, there is a $l > 0$ such that $\delta_{a_1 \ldots a_l}(q_s, q_r) > x$. Now, consider the open ball $\mathsf{B}(\alpha, \frac{1}{2^l}) = a_1 \ldots a_l \Sigma^\omega$. Clearly $\alpha \in \mathsf{B}(\alpha, \frac{1}{2^l})$ and again by Lemma 4.1, $\mu^{rej}_{\mathcal{M},\,\beta} \ge \delta_{a_1 \ldots a_n}(q_s, q_r) > x$ for any $\beta \in a_1 \ldots a_l \Sigma^\omega$. Thus $\mathsf{B}(\alpha, \frac{1}{2^l}) \subseteq \mathcal{L}$. Hence, $\mathcal{L}$ is an open set. $\square$

We will now show that any ω-regular safety language is contained in the classes MSA and MNC.

THEOREM 5.2. Regular ∩ Safety ⊆ MSA *and* Regular ∩ Safety ⊆ MNC.

PROOF. If $\mathcal{L}$ is ω-regular and safety, then there is a deterministic Büchi automaton $\mathcal{B} = (Q, \Delta, q_s, \{q_r\})$ with $(q_r, a, q_r) \in \Delta$ for each $a \in \Sigma$ such that $\Sigma^\omega \setminus \mathcal{L}$ is the language recognized by $\mathcal{B}$ (see Section 3). Now consider the FPM, $\mathcal{M} = (Q, q_s, q_r, \delta)$ where $\delta(q, a, q')$ is 1 if $(q, a, q') \in \Delta$ and is 0 otherwise. It can be shown easily that $\mu^{rej}_{\mathcal{M},\,\alpha} = 0 \Leftrightarrow \alpha \in \mathcal{L}$ and $\mu^{rej}_{\mathcal{M},\,\alpha} = 1 \Leftrightarrow \alpha \in \Sigma^\omega \setminus \mathcal{L}$. Hence $\mathcal{L} = \mathcal{R}_{\le x}(\mathcal{M})$ for all $0 \le x < 1$. $\square$

We will now show that the the class of ω-regular and safety languages coincides exactly with the class MSA. Thus, as a consequence of Theorem 5.2, MSA ⊆ MNC.

THEOREM 5.3. MSA= Regular ∩ Safety.

PROOF. In light of Theorem 5.2, we only need to show that MSA⊆ Regular ∩ Safety. Pick $\mathcal{L}$ ∈MSA. Let $\mathcal{M} = (Q, q_s, q_r, \delta)$ be a monitor such that $\mathcal{R}_{\le 0}(\mathcal{M}) = \mathcal{L}$. Please note that $\mathcal{L}$ is a safety language by Theorem 5.1. Thus, it suffices to show that $\Sigma^\omega \setminus \mathcal{L}$ is ω-regular.

Now, in order to show that $\Sigma^\omega \setminus \mathcal{L}$ is a ω-regular set, consider the Büchi automaton $\mathcal{B} = (Q, \Delta, q_s, \{q_r\})$ where $(q, a, q') \in \Delta$ iff $\delta(q, a, q') > 0$. Let $\mathcal{L}_1$ be the language recognized by $\mathcal{B}$. We claim that $\mathcal{L}_1 = \Sigma^\omega \setminus \mathcal{L}$.

It is easy to show that a word $\alpha = a_1 a_2 \ldots \in \mathcal{L}_1$ iff there is a finite prefix $a_1 \ldots a_l$ of α and a sequence of states $q_0 = q_s,\, q_1, \ldots q_l = q_r$ such that $(q_i, a_{i+1}, q_{i+1}) \in \Delta$ for all $0 \le i < l$. Since $(q_i, a_{i+1}, q_{i+1}) \in \Delta$ iff $\delta(q, a_{i+1}, q') > 0$, it can be easily shown that there is a sequence of states $q_0 = q_s,\, q_1, \ldots q_l = q_r$ such that $(q_i, a_{i+1}, q_{i+1}) \in \Delta$ for all $0 \le i < l$ iff $\delta_{a_1 \ldots a_l}(q_s, q_r) > 0$. Thus $\alpha \in \mathcal{L}_1$ iff there is a finite prefix $a_1 \ldots a_l$ such that

$\delta_{a_1 \ldots a_l}(q_s, q_r) > 0$. In light of Lemma 4.1, $\alpha \in \mathcal{L}_1$ iff $\mu^{rej}_{\mathcal{M},\,\alpha} > 0$. Thus, $\mathcal{L}_1 = \Sigma^\omega \setminus \mathcal{L}$ as required. $\Box$

Hence, we have MSA $\subseteq$ MNC $\subseteq$ Safety. We will now show that each of these containments is strict. Please note that since our alphabet is finite, the set MSA= Regular $\cap$ Safety is countable while the set Safety is uncountable. We can show that the set MNC contains an uncountable number of safety languages that are not ω-regular. The proof uses the automaton $\mathcal{M}_{(1-\mathsf{Id})}$ constructed in Lemma 4.9. As we pointed out before, the same automaton viewed as probabilistic automaton over finite words is often used to prove that non-regular languages (of finite words) can be recognized by probabilistic finite automata.

THEOREM 5.4. *The class* MNC *contains an uncountable number of safety languages which are not ω-regular. Thus* MSA $\subsetneq$ MNC.

PROOF. It suffices to prove the result for the case where the alphabet contains two elements. Let $\Sigma = \{\mathbf{0}, \mathbf{1}\}$. Consider the FPM $\mathcal{M}_{(1-\mathsf{Id})}$ constructed in Lemma 4.9. We have for every word $\alpha \in \Sigma^\omega$, $\mu^{rej}_{\mathcal{M}_{(1-\mathsf{Id})},\,\alpha} = 1 - \mu^{acc}_{\mathcal{M}_{(1-\mathsf{Id})},\,\alpha} = \mathsf{bin}(\alpha)$. Thus, for each irrational $x \in (0,1)$, the language $\mathcal{R}_{\leq x}(\mathcal{M}_{(1-\mathsf{Id})}) = \{\alpha \mid \mathsf{bin}(\alpha) \leq x\}$ which is not a ω-regular language by Proposition 4.8. $\Box$

One may argue that the non-regularity in Theorem 5.4 is a consequence of the irrationality of cut-point x in the proof. However,

PROPOSITION 5.5. *There is a $RatFPM$, $\mathcal{M}$ on $\Sigma = \{\mathbf{0}, \mathbf{1}\}$, such that the language $\mathcal{R}_{\leq \frac{1}{2}}(\mathcal{M})$ is not ω-regular.*

PROOF. Consider the FPM $\mathcal{M}_{\mathsf{Id}}$ constructed in Lemma 4.9. Let $\mathcal{M} = \mathcal{M}_{\mathsf{Id}} \otimes \mathcal{M}_{\mathsf{Id}}$ as defined in Proposition 4.5. Now, $\mu^{acc}_{\mathcal{M},\,\alpha} = \mu^{acc}_{\mathcal{M}_{\mathsf{Id}},\,\alpha} \times \mu^{acc}_{\mathcal{M}_{\mathsf{Id}},\,\alpha} = (\mathsf{bin}(\alpha))^2$. Thus, $\mu^{rej}_{\mathcal{M}_{\mathsf{Id}},\,\alpha} = 1 - (\mathsf{bin}(\alpha))^2$. Hence, $\mathcal{R}_{\leq \frac{1}{2}}(\mathcal{M}) = \{\alpha \mid \mathsf{bin}(\alpha) \geq \sqrt{\frac{1}{2}}\}$ which is not ω-regular by Proposition 4.8, and the fact that the class of ω-regular languages is closed under complementation. $\Box$

We finally show that class of safety languages contains strictly the class MNC.

THEOREM 5.6. MNC $\subsetneq$ Safety.

PROOF. In light of Theorem 5.1, we just need to show that the containment is strict. Let $\Sigma = \{\mathbf{0}, \mathbf{1}\}$. We need to show that there is a safety language $\mathcal{L} \subseteq \Sigma^\omega$ such that for any monitor $\mathcal{M}$ and $x \in (0,1)$, $\mathcal{L} \neq \mathcal{R}_{\leq x}(\mathcal{M})$. Let $L = \{\mathbf{0}^j\mathbf{1}(\mathbf{0}^*\mathbf{1})^*\mathbf{0}^j\mathbf{1} \mid j \in \mathbb{N}, j > 0\}$. Let $\mathcal{L}_1 = L\Sigma^\omega$ and $\mathcal{L} = \Sigma^\omega \setminus \mathcal{L}_1$. Now $\mathcal{L}_1$ is an open set and hence $\mathcal{L}$ is a safety language.

Assume that there is some $\mathcal{M}$ and some $x \in (0,1)$ such that $\mathcal{L} = \mathcal{R}_{\leq x}(\mathcal{M})$. Then by Lemma 4.7, there are constants $c_0, c_1, \ldots, c_k \in \mathbb{R}$ such that for all $\alpha \in \Sigma^\omega$ $\mu^{rej}_{\mathcal{M},\,\mathbf{0}^k\alpha} = c_{k-1}\mu^{rej}_{\mathcal{M},\,\mathbf{0}^{k-1}\alpha} + \ldots + c_0\mu^{rej}_{\mathcal{M},\,\alpha}$ and $c_{k-1} + \ldots + c_0 = 1$.

Consider the set $\mathsf{Pos} = \{i \mid c_i > 0\}$ and let $i_1, i_2, \ldots, i_r$ be an enumeration of the elements of Pos. Please note that Pos is a non-empty set (otherwise c_i's do not add up-to 1). Also observe that for $\alpha = \mathbf{010}^{i_1+1}\mathbf{10}^{i_2+1}\ldots\mathbf{10}^{i_r+1}\mathbf{1}^\omega$, it is the case that $\mathbf{0}^i\alpha \in \mathcal{L}_1$ iff $i \in \mathsf{Pos}$ and hence $\mathbf{0}^i\alpha \notin \mathcal{L}$ iff $i \in \mathsf{Pos}$. Therefore, each $i \in \mathsf{POS}$, we have $\mu^{rej}_{\mathcal{M},\,\mathbf{0}^i\alpha} > x$ and for $i \notin \mathsf{Pos}$, $\mu^{rej}_{\mathcal{M},\,\mathbf{0}^i\alpha} \leq x$. Please note that $\mathcal{L} = \mathcal{R}_{\leq x}(\mathcal{M})$ by assumption. Thus, we have

$$\begin{aligned}
&\mu^{rej}_{\mathcal{M},\,\mathbf{0}^k\alpha} - x \\
&= c_{k-1}\mu^{rej}_{\mathcal{M},\,\mathbf{0}^{k-1}\alpha} + \ldots + c_0\mu^{rej}_{\mathcal{M},\,\alpha} - x \times 1 = \\
&= c_{k-1}\mu^{rej}_{\mathcal{M},\,\mathbf{0}^{k-1}\alpha} + \ldots + c_0\mu^{rej}_{\mathcal{M},\,\alpha} - x(c_{k-1} + \ldots + c_0) \\
&= c_{k-1}(\mu^{rej}_{\mathcal{M},\,\mathbf{0}^{k-1}\alpha} - x) + \ldots + c_0(\mu^{rej}_{\mathcal{M},\,\alpha} - x).
\end{aligned}$$

Now, the left-hand side of the above equation is ≤ 0 as $\mu^{rej}_{\mathcal{M},\,\mathbf{0}^k\alpha} \leq x$. The right hand side is > 0 as $c_i > 0 \Rightarrow \mu^{rej}_{\mathcal{M},\,\mathbf{0}^i\alpha} - x > 0$ and $c_i \leq 0 \Rightarrow \mu^{rej}_{\mathcal{M},\,\mathbf{0}^i\alpha} - x \leq 0$. Thus, we obtain a contradiction. $\Box$

Consider the language $L \subseteq \Sigma^+$ defined in the proof above. We point out here that the language $L\Sigma^*$ is often used as an example to show that there are context-free languages over finite words which are not accepted by any probabilistic automaton [Paz 1971]. The proof uses the iteration lemma for probabilistic finite words and is similar to the proof outlined above. Summarizing the results of this Section, we have Regular $\cap$ Safety = MSA $\subsetneq$ MNC $\subsetneq$ Safety.

5.2 Monitored Languages MWA and MSC.

Recall that given an alphabet Σ, MWA= $\{\mathcal{L} \subseteq \Sigma^\omega \mid \exists$ FPM $\mathcal{M}$ s.t. $\mathcal{L} = \mathcal{R}_{<1}(\mathcal{M})\}$ and MSC= $\{\mathcal{L} \subseteq \Sigma^\omega \mid \exists$ FPM $\mathcal{M}$ and $x \in (0,1)$ s.t. $\mathcal{L} = \mathcal{R}_{<x}(\mathcal{M})\}$. Note that if we were to allow for "infinite" state monitoring, the class MWA would coincide with AlmostSafety [Sistla and Srinivas 2008]. However, FPM's as defined in this paper have only finite memory. We start by showing that both MWA and MSC are subclasses of almost safety languages. Recall that a language $\mathcal{L} \subseteq \Sigma^\omega$ is an almost safety language if and only if it can written as a countable union of safety languages. Of course, the containment MWA $\subseteq$ AlmostSafety can also be viewed as a special case of correspondence between AlmostSafety and the monitoring with infinite states [Sistla and Srinivas 2008].

THEOREM 5.7. *MWA, MSC $\subseteq$ AlmostSafety.*

PROOF. Please note that for any FPM $\mathcal{M}$, any $0 < x \leq 1$ and any word $\alpha \in \Sigma^\omega$, we have $\mathcal{R}_{<x}(\mathcal{M}) = \cup_{j=1}^{\infty}\{\alpha \mid \mu^{rej}_{\mathcal{M},\,\alpha} \leq x - \dfrac{1}{j}\}$. The result follows by observing that $\{\alpha \mid \mu^{rej}_{\mathcal{M},\,\alpha} \leq x - \frac{1}{j}\}$ is a safety language for each j by Theorem 5.1. $\Box$

We will now show that the class of ω-regular and almost safety languages is strictly contained in the class MWA. The proof of containment relies on the fact that if a language $\mathcal{L} \subseteq \Sigma^\omega$ is ω-regular and almost safety then its complement is recognized by a deterministic Büchi automaton [Perrin and Pin 2004; Thomas 1990] (see Section 3). Once this is observed, the proof of containment follows the lines of the proof of the fact that every AlmostSafety language is monitorable with infinite "memory" [Sistla and Srinivas 2008]. The strictness of the containment is witnessed by a probabilistic monitor which is a modified version of the probabilistic Büchi automaton defined in [Baier and Gröβer 2005].

THEOREM 5.8. *Regular $\cap$ AlmostSafety $\subsetneq$ MWA.*

PROOF. We first show that Regular $\cap$ AlmostSafety $\subseteq$ MWA. Let $\mathcal{L} \in$ Regular $\cap$ AlmostSafety. Since $\mathcal{L}$ is ω-regular and almost safety, there is a deterministic Büchi automaton $\mathcal{B} = (Q, \Delta, q_s, Q_f)$ such that $\Sigma^\omega \setminus \mathcal{L}$ is the language recognized by $\mathcal{B}$. Now pick a new state q_r and consider the FPM, $\mathcal{M} = (Q \cup \{q_r\}, q_s, q_r, \delta)$ where for each $a \in \Sigma$

$$\delta(q, a, q') = \begin{cases} \frac{1}{2} & q \in Q_f, q' = q_r \\ \frac{1}{2} & q \in Q_f, q' \in Q, (q, a, q') \in \Delta \\ 1 & q = q' = q_r \\ 1 & q \in Q \setminus Q_f, q' \in Q, (q, a, q') \in \Delta \\ 0 & otherwise \end{cases}$$

It can be shown easily that $\mathcal{L} = \mathcal{R}_{<1}(\mathcal{M})$. Hence, $\mathsf{Regular} \cap \mathsf{AlmostSafety} \subseteq$ MWA.

In order to show that the containment is strict, we just need to show that there is a FPM $\mathcal{M}$ such that $\mathcal{R}_{<1}(\mathcal{M})$ is not a ω-regular language. Let $\Sigma = \{\mathbf{0}, \mathbf{1}\}$ and consider the FPM, $\mathcal{M} = \{Q, q_s, q_r, \delta\}$, shown in Figure 4 on page 15, where $Q = \{q_s, q, q_r\}$ and δ is defined as follows. $\delta(q_s, \mathbf{1}, q_r) = 1, \delta(q_s, \mathbf{0}, q) = \frac{1}{2}, \delta(q_s, \mathbf{0}, q_s) = \frac{1}{2}, \delta(q, \mathbf{0}, q) = 1, \delta(q, \mathbf{1}, q_s) = 1$ and $\delta(q_r, \mathbf{0}, q_r) = \delta(q_r, \mathbf{1}, q_r) = 1$.

Now, note that any word $\alpha \in \Sigma^\omega$ that contains two consecutive occurrences of $\mathbf{1}$ is rejected by $\mathcal{M}$ with probability 1. Also note that the word $\mathbf{0}^\omega$ is accepted by $\mathcal{M}$ with probability 1. Thus, $\mathcal{R}_{<1}(\mathcal{M}) \subseteq \{\mathbf{0}^\omega\} \cup \{\mathbf{0}^{n_1}\mathbf{1}\mathbf{0}^{n_2}\mathbf{1}\ldots\mathbf{0}^{n_k}\mathbf{1}\mathbf{0}^\omega \mid n_1, n_2, n_k > 0\} \cup \{\mathbf{0}^{n_1}\mathbf{1}\mathbf{0}^{n_2}\mathbf{1}\mathbf{0}^{n_3}\mathbf{1}\ldots \mid n_i > 0 \text{ for each } i\}$. Now, given $n_1, n_2, \ldots, n_k > 0$ and $1 \leq i \leq k$, let $u_i = \mathbf{0}^{n_1}\mathbf{1}\mathbf{0}^{n_2}\mathbf{1}\ldots\mathbf{1}\mathbf{0}^{n_i}$ and $v_i = \mathbf{0}^{n_1}\mathbf{1}\mathbf{0}^{n_2}\mathbf{1}\ldots\mathbf{1}\mathbf{0}^{n_i}\mathbf{1}$. We make the following observations:

(1) $\delta_{v_i}(q_s, q_s) = \delta_{u_i}(q_s, q_s) \times \delta_{\mathbf{1}}(q_s, q_s) + \delta_{u_i}(q_s, q) \times \delta_{\mathbf{1}}(q, q_s) = \delta_{u_i}(q_s, q)$.

(2) $\delta_{v_i}(q_s, q) = \delta_{u_i}(q_s, q_s) \times \delta_{\mathbf{1}}(q_s, q) + \delta_{u_i}(q_s, q) \times \delta_{\mathbf{1}}(q, q) = 0$.

(3) $\delta_{u_{i+1}}(q_s, q_s) = \delta_{v_i}(q_s, q_s) \times (\delta_{\mathbf{0}^{n_{i+1}}}(q_s, q_s) + \delta_{\mathbf{0}^{n_{i+1}}}(q, q_s)) = \delta_{v_i}(q_s, q_s) \times (1/2)^{n_{i+1}}$.

(4) $\delta_{u_{i+1}}(q_s, q) = \delta_{v_i}(q_s, q_s) - \delta_{u_{i+1}}(q_s, q_s) = \delta_{v_i}(q_s, q_s) \times (1 - (1/2)^{n_{i+1}})$.

(5) Thus, we get that $\delta_{v_{i+1}}(q_s, q_s) = \delta_{v_i}(q_s, q_s) \times (1 - (1/2)^{n_{i+1}}) = \prod_{j=1}^{i+1}(1 - (1/2)^{n_j})$ and $\delta_{v_{i+1}}(q_s, q) = 0$.

Therefore, any word of the form $\mathbf{0}^{n_1}\mathbf{1}\mathbf{0}^{n_2}\mathbf{1}\ldots\mathbf{0}^{n_k}\mathbf{1}\mathbf{0}^\omega$ is accepted by $\mathcal{M}$ with probability > 0. Furthermore, the probability of accepting a word of the form $\mathbf{0}^{n_1}\mathbf{1}\mathbf{0}^{n_2}\mathbf{1}\ldots$ is $\prod_{k=1}^{\infty}(1 - (1/2)^{n_k})$. Thus, it can be easily checked that $\mathcal{R}_{<1}(\mathcal{M})$ is the union of two disjoint languages $\mathcal{L}_1 = \mathbf{0}\mathbf{0}^*(\mathbf{1}\mathbf{0}\mathbf{0}^*)^*\mathbf{0}^\omega$ and $\mathcal{L}_2 = \{\mathbf{0}^{n_1}\mathbf{1}\mathbf{0}^{n_2}\mathbf{1}\mathbf{0}^{n_3}\mathbf{1}\ldots\ldots \mid n_i > 0 \text{ for each } i \text{ and } \prod_{k=1}^{\infty}(1 - (1/2)^{n_k}) > 0\}$. Now $\mathcal{L}_1$ is a ω-regular language, but $\mathcal{L}_2$ is not. Thus, $\mathcal{R}_{<1}(\mathcal{M})$ is not a ω-regular language. $\square$

We will shortly show that the class MWA is strictly contained in the class MSC. However, before we proceed, we will need the following lemma which shows that even if a language $\mathcal{L} \in$ MWA is not ω-regular, the languages $\mathsf{cl}(\mathcal{L})$ and $\mathsf{cl}(\Sigma^\omega \setminus \mathcal{L})$ must be ω-regular. Recall that for a language $\mathcal{L}_1$, $\mathsf{cl}(\mathcal{L}_1)$ is the smallest safety language containing $\mathcal{L}_1$. Hence, ω-regularity of $\mathsf{cl}(\mathcal{L}_1)$ and $\mathsf{cl}(\Sigma^\omega \setminus \mathcal{L}_1)$ is a necessary (but not sufficient) condition for an almost safety language $\mathcal{L}_1$ to belong to MWA.

LEMMA 5.9. *Let $\mathcal{L} \in$ MWA. Then the safety languages $\mathsf{cl}(\mathcal{L})$ and $\mathsf{cl}(\Sigma^\omega \setminus \mathcal{L})$ are ω-regular. There is an almost safety language $\mathcal{L}_1$ such that $\mathsf{cl}(\mathcal{L}_1)$ and $\mathsf{cl}(\Sigma^\omega \setminus \mathcal{L}_1)$ are ω-regular but $\mathcal{L}_1 \notin$ MWA.*

PROOF. Let $\mathcal{M} = (Q, q_s, q_r, \delta)$ be such that $\mathcal{L} = \mathcal{R}_{<1}(\mathcal{M})$. For $q \in Q \setminus \{q_r\}$, let $\mathcal{M}_q = (Q, q, q_r, \delta)$, *i.e.*, the FPM obtained by making q the initial state.

We first show that $\mathsf{cl}(\mathcal{L})$ is ω-regular. Please note that if $\mathcal{L}$ is empty, then this is trivially true. Let $\mathcal{L}$ be non-empty. Let $Q_0 = \{q \in Q | q \neq q_r \text{ and there is some } \alpha \text{ s.t. } \mu^{acc}_{\mathcal{M}_q,\alpha} > 0\}$. Consider the Büchi automaton $\mathcal{B} = (Q_0, \Delta, q_s, Q_0)$ where $\Delta(q, a, q')$ iff $\delta(q, a, q') > 0$. We claim that the language $\mathcal{L}(\mathcal{B})$ recognized by $\mathcal{B}$ is the set $\mathsf{cl}(\mathcal{R}_{<1}(\mathcal{M}))$. Please note that since the set of accepting states of $\mathcal{B}$ is the set of states of $\mathcal{B}$, the language $\mathcal{L}(\mathcal{B})$ is a safety language. Hence, in order to show that $\mathcal{L}(\mathcal{B}) = \mathsf{cl}(\mathcal{R}_{<1}(\mathcal{M}))$, it suffices to show that $\mathcal{L}(\mathcal{B}) \subseteq \mathsf{cl}(\mathcal{R}_{<1}(\mathcal{M}))$ and $\mathcal{R}_{<1}(\mathcal{M}) \subseteq \mathcal{L}(\mathcal{B})$.

We will show that $\mathcal{L}(\mathcal{B}) \subseteq \mathsf{cl}(\mathcal{R}_{<1}(\mathcal{M}))$. Let $\alpha = a_1a_2 \ldots \in \mathcal{L}(\mathcal{B})$ and for $x > 0$ let $\mathsf{B}(\alpha, x)$ be the open ball of radius x centered at α. Pick $k > 0$ such that $\frac{1}{2^k} < x$. Clearly, $a_1a_2 \ldots a_k\Sigma^\omega \subseteq \mathsf{B}(\alpha, x)$. Therefore, it suffices to show that $a_1a_2 \ldots a_k\Sigma^\omega \cap \mathcal{R}_{<1}(\mathcal{M}) \neq \emptyset$. Now since $\alpha \in \mathcal{L}(\mathcal{B})$, there are some $k + 1$ states $q_0 = q_s, q_1, \ldots q_k \in Q_0$ such that $(q_0, a_1, q_1), (q_1, a_2, q_2), \ldots, (q_{k-1}, a_k, q_k) \in \Delta$. By definition $\delta(q_i, a_i, q_{i+1}) > 0$ for each $0 \leq i < k$ and there is word β such that $\mu^{acc}_{\mathcal{M}_{q_k},\beta} > 0$. Consider the word $\alpha_1 = a_0a_1 \ldots a_k\beta$. It can be easily shown that $\mu^{acc}_{\mathcal{M},\alpha_1} > 0$. Thus $a_1a_2 \ldots a_k\Sigma^\omega \cap \mathcal{R}_{<1}(\mathcal{M}) \neq \emptyset$.

Now we show that $\mathcal{R}_{<1}(\mathcal{M}) \subseteq \mathcal{L}(\mathcal{B})$. Pick $\alpha = a_1a_2a_n \ldots \in \mathcal{R}_{<1}$ and fix it. In order to show that $\alpha \in \mathcal{L}(\mathcal{B})$, by Koning's lemma, it suffices to show that for each $k > 1$ there is a path in $\mathcal{B}$ from q_s on input symbols $a_1, a_2 \ldots a_{k-1}$. Let $\alpha_k = a_ka_{k+1} \ldots$ and $u_k = a_1a_2 \ldots a_{k-1}$.

We have by definition $\mu^{acc}_{\mathcal{M},\alpha} > 0$. Please note that it can be shown that for each k, $\mu^{acc}_{\mathcal{M},\alpha} = \sum_{q \in Q\setminus\{q_r\}} \delta_{u_k}(q_s, q)\mu^{acc}_{\mathcal{M}_q,\alpha_k}$. Now since $\mu^{acc}_{\mathcal{M}_q,\beta} = 0$ for every $q \in Q\setminus(Q_0 \cup \{q_r\})$ and $\beta \in \Sigma^\omega$, we get that $\mu^{acc}_{\mathcal{M},\alpha} = \sum_{q \in Q_0} \delta_{u_k}(q_s, q)\mu^{acc}_{\mathcal{M}_q,\alpha_k}$. If there is no path on input $a_1a_2 \ldots a_{k-1}$ in the automaton $\mathcal{B}$, then $\delta_{u_k}(q_s, q) = 0$ for every $q \in Q_0$ which contradicts $\mu^{acc}_{\mathcal{M},\alpha} > 0$. Hence, there is a path in $\mathcal{B}$ on input symbols $a_1, a_2 \ldots a_{k-1}$.

Now in order to show that $\mathsf{cl}(\Sigma^\omega \setminus \mathcal{R}_{<1}(\mathcal{M}))$ is ω-regular, we can assume that $\mathcal{R}_{<1}(\mathcal{M}) \neq \Sigma^\omega$ (otherwise, $\mathsf{cl}(\Sigma^\omega \setminus \mathcal{R}_{<1}(\mathcal{M})) = \emptyset$ which is ω-regular). We now construct the Büchi automata $\mathcal{B} = (\rho, \Delta, \{q_s\}, \rho)$ where $\rho \subseteq \wp(Q)$ (the power-set of Q) and Δ are defined as follows. A set $Q_1 \subseteq Q$ belongs to ρ iff there is a $\beta \in \Sigma^\omega$ such that for any $q \in Q_1, \mu^{rej}_{\mathcal{M}_q,\beta} = 1$. Please note that $\{q_s\} \in \rho$. Now $(Q_1, a, Q_2) \in \Delta$ iff $\mathsf{post}(Q_1, a) = \{q' | \exists q \in Q_1 \text{ s.t. } \delta(q, a, q') > 0\} \subseteq Q_2$. We can once again show that $\mathsf{cl}(\Sigma^\omega \setminus \mathcal{R}_{<1}(\mathcal{M})) = \mathcal{L}(\mathcal{B})$.

However, the ω-regularity of $\mathsf{cl}(\mathcal{L})$ and $\mathsf{cl}(\Sigma^\omega \setminus \mathcal{L})$ is not sufficient to guarantee inclusion of $\mathcal{L}$ in MWA. Let $\Sigma = \{\mathbf{0}, \mathbf{1}\}$. Now, consider the Language $\mathcal{L}_2 = \{\mathbf{0}, \mathbf{1}\}^*\mathbf{11}\{\mathbf{0}, \mathbf{1}\}^\omega$. The set $\mathcal{L}_2$ is an open set and hence an almost safety language. Let $\mathcal{L}_1 = \mathcal{L}_2 \cup \{\mathbf{010^210^31} \ldots\}$. Now, the set $\{\mathbf{010^210^31} \ldots\}$ is a closed set and hence an almost safety language. It can be easily shown that $\mathsf{cl}(\mathcal{L}_1) = \Sigma^\omega$ and $\mathsf{cl}(\{\mathbf{0}, \mathbf{1}\}^\omega \setminus \mathcal{L}_1) = \{\mathbf{0}, \mathbf{1}\}^\omega \setminus \mathcal{L}_2$ both of which are ω-regular.

Now, we show that $\mathcal{L}_1 \neq \mathcal{R}_{<1}(\mathcal{M}_1)$ for any FPM $\mathcal{M}_1$ by contradiction. Suppose there is a $\mathcal{M}'$ such that $\mathcal{L} = \mathcal{R}_{<1}(\mathcal{M}')$. Then, by the Lemma 4.7, there are real numbers $c_0, \ldots c_{k-1} \in \mathbb{R}$ such that for all $v \in \Sigma^+, \alpha \in \Sigma^\omega$,

$$\mu^{rej}_{\mathcal{M}',v\mathbf{0}^k\alpha} = c_{k-1}\mu^{rej}_{\mathcal{M}',v\mathbf{0}^{k-1}\alpha} + \ldots + c_0\mu^{rej}_{\mathcal{M}',v\alpha}$$

and $c_{k-1} + \ldots + c_0 = 1$.

Now, pick $v_1 = \mathbf{010}^2 \dots \mathbf{10}^{k+1}\mathbf{1}$. and $\alpha_1 = \mathbf{0010}^{k+3}\mathbf{10}^{k+4}\mathbf{1}\dots$. We get

$$\mu^{rej}_{\mathcal{M}',\, v_1\mathbf{0}^k\alpha_1} = c_{k-1}\mu^{rej}_{\mathcal{M}',\, v_1\mathbf{0}^{k-1}\alpha_1} + \dots + c_0\mu^{rej}_{\mathcal{M}',\, v_1\alpha_1}.$$

Now $v_1\mathbf{0}^j\alpha_1 \in \Sigma^\omega \setminus \mathcal{L}_1$ for all $0 \leq j < k$. Hence, $\mu^{rej}_{\mathcal{M}',\, v_1\mathbf{0}^j\alpha_1} = 1$ for all $0 \leq j < k$. Hence, we get

$$\mu^{rej}_{\mathcal{M}',\, v\mathbf{0}^k\alpha} = c_{k-1} + \dots + c_0 = 1.$$

However, $v_1\mathbf{0}^k\alpha_1 = \mathbf{010}^2\mathbf{10}^3 \dots \in \mathcal{L}_1$ and therefore

$$\mu^{rej}_{\mathcal{M}',\, v\mathbf{0}^k\alpha} < 1.$$

Thus, we have arrived at a contradiction. □

We are ready to show that the class MSC strictly contains the class MWA.

THEOREM 5.10. MWA $\subsetneq$ MSC.

PROOF. We start by showing that MWA $\subseteq$ MSC. Given a FPM $\mathcal{M}$ and $x \in (0, 1)$, let $\mathcal{M}' = \mathcal{M}^x$ be the FPM defined in Proposition 4.4 such that $\mu^{rej}_{\mathcal{M}',\, \alpha} = x \times \mu^{rej}_{\mathcal{M},\, \alpha}$ for every word α. Now, it follows easily that $\mathcal{R}_{<1}(\mathcal{M}) = \mathcal{R}_{<x}(\mathcal{M}')$. Hence, MWA $\subseteq$ MSC.

In order to show that the containment is strict, construct $\mathcal{M}$ as in the proof of Theorem 5.5 such that for every α, $\mu^{rej}_{\mathcal{M},\, \alpha} = 1 - (\mathsf{bin}(\alpha))^2$. Thus, $\mathcal{R}_{<\frac{1}{2}}(\mathcal{M}) = \{\alpha \mid \mathsf{bin}(\alpha) > \sqrt{\frac{1}{2}}\}$. Now, it can be shown that $\mathsf{cl}(\mathcal{R}_{<\frac{1}{2}}(\mathcal{M})) = \{\alpha \mid \mathsf{bin}(\alpha) \geq \sqrt{\frac{1}{2}}\}$ which is not ω-regular by Proposition 4.8. Thus, if there is some FPM $\mathcal{M}'$ such that $\mathcal{R}_{<1}(\mathcal{M}') = \mathcal{R}_{<\frac{1}{2}}(\mathcal{M})$ then $\mathsf{cl}(\mathcal{R}_{<1}(\mathcal{M}'))$ is not ω-regular which contradicts Lemma 5.9. □

Finally, we show that the class MSC is strictly contained in the class of almost safety languages. The proof is similar to the proof of Theorem 5.6 which showed that there is a safety language $\mathcal{L}$ that is not contained in MNC. Indeed the same safety language used in the proof of Theorem 5.6 (any safety language is also an almost safety language) witnesses the strictness of the containment MSC $\subseteq$ $\mathsf{AlmostSafety}$.

THEOREM 5.11. MSC $\subsetneq$ $\mathsf{AlmostSafety}$.

PROOF. Please note that MSC $\subseteq$ $\mathsf{AlmostSafety}$ as a consequence of Theorem 5.7. Consider the safety language $\mathcal{L}$ defined in the proof of Theorem 5.6 as follows. Let $L = \{\mathbf{0}^j\mathbf{1}(\mathbf{0}^*\mathbf{1})^*\mathbf{0}^j\mathbf{1} | j \in \mathbb{N}, k > 0\}$. Let $\mathcal{L}_1 = L\Sigma^\omega$ and $\mathcal{L} = \Sigma^\omega \setminus \mathcal{L}_1$. We can show that $\mathcal{L} \notin$ MSC by an argument similar to proof of $\mathcal{L} \notin$MNC sketched in Theorem 5.7. □

Summarizing the results of this Section, we have $\mathsf{Regular} \cap \mathsf{AlmostSafety}$ $\subsetneq$ MWA $\subsetneq$ MSC $\subsetneq$ $\mathsf{AlmostSafety}$. Please note that since MSA coincides with ω-regular safety languages, MSA is strictly contained in MWA and MSC. Also, since there are ω-regular almost safety languages which are not safety, it follows immediately that neither MWA nor MSC is contained in MNC. Therefore, a natural question to ask is if MNC $\subseteq$ MSC? We will answer the question in negative in Section 5.3 as the proof requires a result which we will prove in Section 5.3.

5.3 Robust Monitors

In this Section, we will show that the class MNC $\not\subseteq$ MSC. The proof will utilize a result on robust monitors. Robust monitors are probabilistic FPM's such that there is a separation between probability of rejecting a word in the language permitted by the monitor and the probability of rejecting a word in the language rejected by the monitor. Formally,

Definition: A FPM, $\mathcal{M}$ on Σ, is said to be x-*robust* for $x \in (0,1)$ if there is an $\epsilon > 0$ such that for any $\alpha \in \Sigma^\omega$, $|\mu^{rej}_{\mathcal{M},\,\alpha} - x| > \epsilon$.

Observe that if $\mathcal{M}$ is x-robust then the languages $\mathcal{R}_{<x}(\mathcal{M})$ and $\mathcal{R}_{\leq x}(\mathcal{M})$ coincide. Robustness is a generalization of the concept of isolated cut-points defined for probabilistic automata over finite words [Rabin 1963] - x is said to be an isolated cut-point for a probabilistic automaton over finite words if there is an $\epsilon > 0$ such that for every finite word u the probability of accepting u is bounded away from x by ϵ. It was shown in [Rabin 1963] that if x is an isolated cut-point then the language of finite words accepted by the probabilistic automaton is a regular language. We can generalize this result to probabilistic monitors and demonstrate that the language $\mathcal{R}_{<x}(\mathcal{M})$ is a ω-regular safety language. The proof relies on the fact that $\mathcal{R}_{\leq x}(\mathcal{M})$ is a safety language which implies that ω-regularity of $\mathcal{R}_{\leq x}(\mathcal{M})$ is equivalent to the regularity of the set of its finite prefixes (see Section 3). This observation is crucial in the proof we present, whereas the result in [Rabin 1963] does not depend on any such topological consideration. The proof that the set of finite prefixes of $\mathcal{R}_{\leq x}(\mathcal{M})$ is regular, however, follows an argument similar to the result in [Rabin 1963]. The proof depends on the following result, proved in [Rabin 1963].

PROPOSITION 5.12. *Given $n \in \mathbb{N}, n > 0$, let $\mathcal{P}_n \subseteq \mathbb{R}^n$ be the set of vectors defined as $\{(\xi_1, \ldots \xi_n) \mid 0 \leq \xi_j \leq 1, \sum_{j=1}^n \xi_j = 1\}$. Given $\epsilon \in \mathbb{R}, \epsilon > 0$, let $\mathcal{U} \subseteq \mathcal{P}_n$ be a set such that for any $(\xi_1, \ldots \xi_n), (\xi'_1, \ldots \xi'_n) \in \mathcal{U}$, $\sum_j^n |\xi_j - \xi'_j| > \epsilon$. Then $\mathcal{U}$ must be finite.*

Using the above observation, we can show,

THEOREM 5.13. *Let $\mathcal{M}$ be x-robust for some $x \in (0,1)$. Then $\mathcal{R}_{<x}(\mathcal{M}) = \mathcal{R}_{\leq x}(\mathcal{M})$ is a ω-regular safety language.*

PROOF. Let $\mathcal{M} = (Q, q_s, q_r, \delta)$ be x-robust and let $\epsilon > 0$ be such that $|\mu^{rej}_{\mathcal{M},\,\alpha} - x| > \epsilon$ for all $\alpha \in \Sigma^\omega$. Let $\mathcal{L} = \mathcal{R}_{\leq x}(\mathcal{M})$ and let $L \subseteq \Sigma^*$ be the set of finite prefixes of $\mathcal{L}$. In other words $L = \{u \in \Sigma^* \mid \exists \alpha \in \Sigma^\omega \text{ s. t. } u\alpha \in \mathcal{L}\}$.

Since $\mathcal{L}$ is a safety language, in order to prove that $\mathcal{L}$ is ω-regular, it suffices to show that L is a regular language (viewed as a language over finite words). We will demonstrate that L is a regular language by demonstrating that it has finite Myhill-Nerode index, *i.e.*, the number of equivalence classes $\equiv_L$ is finite. Recall that for two finite words $u_1, u_2 \subseteq \Sigma^*$, $u_1 \equiv_L u_2$ if for all $v \in \Sigma^*$, $u_1 v \in L$ iff $u_2 v \in L$.

Now assume that u_1 and u_2 are two finite words such that $u_1 \not\equiv_L u_2$. Then, there is some $v \in \Sigma^*$ such that either $u_1 v \in L$ and $u_2 v \notin L$ or $u_1 v \notin L$ and $u_2 v \in L$.

First, consider the case $u_1 v \in L$ and $u_2 v \notin L$. Now, since $u_1 v \in L$, there is some $\alpha \in \Sigma^\omega$ such that $u_1 v \alpha \in \mathcal{L}$. Pick one such word, say α_0 and fix it. Thus $u_1 v \alpha_0 \in \mathcal{L}$. Also since $u_2 v \notin L$, $u_2 v \alpha_0 \notin \mathcal{L}$. By definition of $\mathcal{L}$, we get $\mu^{rej}_{\mathcal{M},\,u_1 v \alpha_0} \leq x$ and $\mu^{rej}_{\mathcal{M},\,u_2 v \alpha_0} > x$. Since x is an isolated cut-point, we get $\mu^{rej}_{\mathcal{M},\,u_1 v \alpha_0} < x - \epsilon$ and $\mu^{rej}_{\mathcal{M},\,u_2 v \alpha_0} > x + \epsilon$.

By Lemma 4.1, there exists a finite prefix of α_0, say v', such that $\delta_{u_1 v v'}(q_s, q_r) < x - \epsilon$ and $\delta_{u_2 v v'}(q_s, q_r) > x + \epsilon$. Thus, $\delta_{u_2 v v'}(q_s, q_r) - \delta_{u_1 v v'}(q_s, q_r) > 2\epsilon$. Now, $\delta_{u_2 v v'}(q_s, q_r) - \delta_{u_1 v v'}(q_s, q_r) = \sum_{q \in Q} (\delta_{u_2}(q_s, q) - \delta_{u_1}(q_s, q)) \delta_{v v'}(q, q_r)$. Thus, we get that $\sum_{q \in Q} (\delta_{u_2}(q_s, q) - \delta_{u_1}(q_s, q)) \delta_{v v'}(q, q_r) > 2\epsilon$.

Now, we have that $\sum_{q \in Q} (\delta_{u_2}(q_s, q) - \delta_{u_1}(q_s, q)) \delta_{v v'}(q, q_r) \leq \sum_{q \in Q} |\delta_{u_2}(q_s, q) - \delta_{u_1}(q_s, q)| |\delta_{v v'}(q, q_r)|$. Since $0 \leq \delta_{v v'}(q, q_r) \leq 1$, we get $\sum_{q \in Q} |\delta_{u_2}(q_s, q) - \delta_{u_1}(q_s, q)| \geq$

$\sum_{q\in Q}(\delta_{u_2}(q_s,q) - \delta_{u_1}(q_s,q))\delta_{vv'}(q,q_r) > 2\epsilon$. Similarly, if $u_1 v \notin L$ and $u_2 v \in L$ then $\sum_{q\in Q}|\delta_{u_2}(q_s,q) - \delta_{u_1}(q_s,q)| > 2\epsilon$.

Therefore, if $u_1 \not\equiv_L u_2$, it must be the case that $\sum_{q\in Q}|\delta_{u_2}(q_s,q) - \delta_{u_1}(q_s,q)| > 2\epsilon$. Also, please note that $\sum_{q\in Q}\delta_{u_1}(q_s,q) = \sum_{q\in Q}\delta_{u_2}(q_s,q) = 1$. By Proposition 5.12, it follows that there can only finite number of equivalence classes. $\square$

The above proof is sound even if only one side of the rejection probabilities is bounded away. Therefore, we get the following corollary.

COROLLARY 5.14. *Let $\mathcal{M}$ be a monitor such that there is an $\epsilon > 0$ such that for each $\alpha \in \Sigma^\omega$ either $\mu^{rej}_{\mathcal{M},\alpha} = 1$ or $\mu^{rej}_{\mathcal{M},\alpha} \leq 1-\epsilon$, then $\mathcal{R}_{<1}(\mathcal{M}) \in \mathsf{Regular} \cap \mathsf{Safety}$.*

PROOF. Follows immediately from the fact that $\mathcal{M}$ is $1-\frac{\epsilon}{2}$-robust and $\mathcal{R}_{<1}(\mathcal{M}) = \mathcal{R}_{<1-\frac{\epsilon}{2}}(\mathcal{M}) \in \mathsf{Regular} \cap \mathsf{Safety}$. $\square$

Now, we are ready to show that MSC and MNC are incomparable.

THEOREM 5.15. MSC $\not\subseteq$ MNC *and* MNC $\not\subseteq$ MSC.

PROOF. Observe that since MSC contains almost safety languages that are not safety languages, we get that MSC $\not\subseteq$ MNC. In order to show that MNC $\not\subseteq$ MSC, let $\Sigma = \{\mathbf{0},\mathbf{1}\}$. We will show that there is a RatFPM $\mathcal{M}$ on Σ, such that for any FPM $\mathcal{M}'$ on Σ, and any $x \in [0,1]$, $\mathcal{R}_{\leq\frac{15}{16}}(\mathcal{M}) \neq \mathcal{R}_{<x}(\mathcal{M}')$.

Now, by repeated use of Lemma 4.9, Proposition 4.5 and Proposition 4.6, we can construct a RatFPM $\mathcal{M}$ such that for all $\alpha \in \Sigma^\omega$, $\mu^{acc}_{\mathcal{M},\alpha} = (\mathsf{bin}(\alpha))^4(1-\mathsf{bin}(\alpha)^2)^2$. Now a word $\alpha \in \mathcal{R}_{\leq\frac{15}{16}}(\mathcal{M}) \Leftrightarrow \mu^{rej}_{\mathcal{M},\alpha} \leq \frac{15}{16} \Leftrightarrow 1-\mu^{acc}_{\mathcal{M},\alpha} \leq \frac{15}{16} \Leftrightarrow \frac{1}{16} \leq (\mathsf{bin}(\alpha))^4(1-\mathsf{bin}(\alpha)^2)^2 \Leftrightarrow \frac{1}{4} \leq \mathsf{bin}(\alpha)^2(1-\mathsf{bin}(\alpha)^2) \Leftrightarrow 0 \leq -\frac{1}{4}+\mathsf{bin}(\alpha)^2(1-\mathsf{bin}(\alpha)^2) \Leftrightarrow 0 \leq -(\frac{1}{2}-\mathsf{bin}(\alpha)^2)^2 \Leftrightarrow \frac{1}{\sqrt{2}} = \mathsf{bin}(\alpha)$.

Now there is only one word β, namely $\mathsf{wrd}(\frac{1}{\sqrt{2}})$, (the "binary expansion" of $\frac{1}{\sqrt{2}}$) such that $\mathsf{bin}(\beta) = \frac{1}{\sqrt{2}}$. Thus $\mathcal{R}_{\leq\frac{15}{16}}(\mathcal{M}) = \{\mathsf{wrd}(\frac{1}{\sqrt{2}})\}$. Now, $\mathcal{R}_{\leq\frac{15}{16}}(\mathcal{M})$ is not ω-regular since every ω-regular language must contain an ultimately periodic word [Perrin and Pin 2004; Thomas 1990], and $\mathsf{wrd}(\frac{1}{\sqrt{2}})$ is not ultimately periodic (as $\frac{1}{\sqrt{2}}$ is irrational).

We proceed by contradiction. Suppose there is some $\mathcal{M}'$ and x such that $\mathcal{R}_{\leq\frac{15}{16}}(\mathcal{M}) = \mathcal{R}_{<x}(\mathcal{M}')$. Let $y = \mu^{rej}_{\mathcal{M}',\mathsf{wrd}(\frac{1}{\sqrt{2}})}$. By definition $y < x$ and for all words $\alpha \neq \mathsf{wrd}(\frac{1}{\sqrt{2}})$, $\mu^{rej}_{\mathcal{M}',\alpha} \geq x$. Let $x_1 = \frac{x+y}{2}$. Clearly $\mathcal{M}'$ is x_1-robust. Thus, by Theorem 5.13, $\mathcal{R}_{<x_1}(\mathcal{M}')$ is ω-regular. This contradicts the fact that $\mathcal{R}_{<x_1}(\mathcal{M}') = \mathcal{R}_{\leq\frac{15}{16}}(\mathcal{M}) = \mathsf{wrd}(\frac{1}{\sqrt{2}})$ is not ω-regular. $\square$

6. DECISION PROBLEMS

In this section, we consider the problems of checking emptiness and universality of RatFPMs (with rational cut-points). Our results for these problems are summarized in Figure 6.

6.1 Emptiness Problem: Upper Bounds

In this section we will prove the upper bounds for the emptiness problem for monitors in the classes MSA, MWA, MSC, and MWA. We will assume that the automata given as input to the emptiness problem is a RatFPM i.e., a probabilistic monitor with rational transition

	EMPTINESS	UNIVERSALITY
MSA	**PSPACE**-complete (Theorems 6.1 and 6.6)	**NL**-complete (Theorems 6.10 and 6.16)
MWA	**PSPACE**-complete (Theorems 6.4 and 6.6)	**PSPACE**-complete (Theorems 6.13 and 6.17)
MSC	**co-R.E.**-complete (Theorems 6.4 and 6.8)	Π_1^1-complete (Theorems 6.15 and 6.19)
MNC	**R.E.**-complete (Theorems 6.5 and 6.9)	**co-R.E.**-complete (Theorems 6.14 and 6.18)

Fig. 6. Table summarizing the complexity of the emptiness and universality problems for the various classes of monitors.

probabilities.

MSA**:** We begin by considering the class of monitors in MSA. We show that the emptiness problem is in **PSPACE** by reducing it to the universality problem of Büchi automata.

THEOREM 6.1. *Let $\mathcal{M} = (Q, q_s, q_r, \delta)$ be a RatFPM on an alphabet Σ. The problem of checking the emptiness of $\mathcal{R}_{\leq 0}(\mathcal{M})$ is in* **PSPACE***.*

PROOF. We construct a non-deterministic Büchi automaton $\mathcal{B}$, which is essentially the same as $\mathcal{M}$, except that the transition probabilities are discarded. Formally, $\mathcal{B} = (Q, \Delta, q_s, q_r)$ where $\Delta = \{(q, a, q') \mid \delta(q, a, q') > 0\}$. Note that q_r is the only final state of the Büchi automaton $\mathcal{B}$. It is easy to see that $\mathcal{R}_{\leq 0}(\mathcal{M}) = \emptyset$ iff $L(\mathcal{B}) = \Sigma^\omega$ where $L(\mathcal{B})$ is the the language accepted by the Büchi automaton $\mathcal{B}$. The universality problem for Büchi automata is known to be in PSPACE, and hence the result follows. □

MWA **and** MSC**:** Next we establish that for monitors in MWA and MSC, the emptiness problems are in **PSPACE** and **R.E.**, respectively. The proofs for both of these rely on a technical lemma that we state and prove before presenting the upper bound for the emptiness problem. We begin with some definitions that we will need.

Definition: For a FPM $\mathcal{M} = (Q, q_s, q_r, \delta)$, the *deterministic graph* associated with it is the edge-labeled multi-graph $(2^Q, \mathsf{E})$, where $\mathsf{E} \subseteq 2^Q \times \Sigma \times 2^Q$ is defined as follows: $(S, a, S') \in \mathsf{E}$ iff $S' = \{q' \in Q \mid \exists q \in S \text{ s.t. } \delta(q, a, q') > 0\}$. We denote the deterministic graph of $\mathcal{M}$ by $G(\mathcal{M})$.

A vertex $S \in 2^Q$ of $G(\mathcal{M})$ will be said to be *good* if $q_r \notin S$. A *cycle* in $G(\mathcal{M})$ is a sequence of edges $(S_0, a_0, S_1), (S_1, a_1, S_2), ..., (S_{n-1}, a_{n-1}, S_n)$ such that $S_0 = S_n$; S_0 is the starting vertex of the cycle, and $a_0 \ldots a_{n-1}$ is the input sequence associated with the cycle. Finally, we say that the cycle is *good* if all the nodes on it are good.

LEMMA 6.2. *Let $\mathcal{M} = (Q, q_s, q_r, \delta)$ be a* FPM *on an alphabet Σ, and $x \in (0, 1]$. The language $\mathcal{R}_{<x}(\mathcal{M})$ is non-empty iff there exists a finite word u, a set $S \subseteq Q$ such that S lies on a good cycle in $G(\mathcal{M})$, and $\delta_u(q_s, S) > 1 - x$, where $\delta_u(q_s, S) = \sum_{q' \in S} \delta_u(q_s, q')$.*

PROOF. We prove the "if" part of the lemma as follows. Let u, S be such that $\delta_u(q_s, S) > 1 - x$ and S lies on a good cycle in $G(\mathcal{M})$. Let C be the good cycle on which S lies. Without loss of generality, we assume that S is the starting state of C and v is the finite input sequence associated with C. Since C is a good cycle and $\delta_u(q_s, S) > 1 - x$, it is not difficult to see that, for $\alpha = uv^\omega$, $\mu^{acc}_{\mathcal{M},\alpha} > 1 - x$, and hence $\mu^{rej}_{\mathcal{M},\alpha} < x$. Thus $\alpha \in \mathcal{R}_{<x}(\mathcal{M})$.

The "only if" part of the lemma is proved as follows. Assume that for some $\alpha \in \Sigma^\omega$, $\mu^{rej}_{\mathcal{M},\alpha} < x$. This means $\mu^{acc}_{\mathcal{M},\alpha} > 1 - x$. Let $\alpha = a_1 a_2 \ldots$ and $k = 2^{|Q|}$. Note that

the number of states of $G(\mathcal{M})$ is bounded by k. For every, $j \geq 1$, let T_j be the set of all $q' \in Q$ such that $\delta_{\alpha[j+1:j+k]}(q', q_r) > 0$; i.e., T_j is the set of all states from which q_r can be reached on the input sequence $\alpha[j+1 : j+k]$.
Claim: There exists an $i \geq 1$ such that $\delta_{\alpha[1:i]}(q_s, T_i) < x$.

Before proving the above claim, let us observe that the "only if" part of the lemma follows from it. Let $i \geq 1$ be such that $\delta_{\alpha[1:i]}(q_s, T_i) < x$. Let S_i be the set of $q' \in Q \setminus T_i$ such that $\delta_{\alpha[1:i]}(q_s, q') > 0$. It is easy to see that $\delta_{\alpha[1:i]}(q_s, S_i) > 1 - x$. Note that from each of the states in S_i, the state q_r can not be reached in the automaton $\mathcal{M}$ on the input sequence $\alpha[i+1 : i+k]$. From this it follows that, the sequence $S_i, S_{i+1}, \ldots, S_{i+k}$ of vertices of $G(\mathcal{M})$ reached on the input sequence $\alpha[i+1 : i+k]$ starting from S_i are all good. Since k equals the number of vertices of $G(\mathcal{M})$, using pigeon hole principle, we see that the above sequence contains a cycle and it is a good cycle. Furthermore, we see that $\delta_{\alpha[1:j]}(q_s, S_j) > 1 - x$ for every j such that $i \leq j \leq i+k$. Thus, the "only if" part of the lemma follows.
Proof of the claim: We conclude this proof by showing the claim by contradiction. Assume the claim is not true. This means for every $j \geq 1$, $\delta_{\alpha[1:j]}(q_s, T_j) \geq x$. Let $j_0 < j_1 < \ldots < j_\ell \ldots$ be the infinite sequence of natural numbers such that for each $\ell \geq 0$, $j_\ell = \ell \cdot k$. Clearly, for each $\ell > 0$, $\delta_{\alpha[1:j_\ell]}(q_s, T_{j_\ell}) \geq x$ and $j_{\ell+1} = j_\ell + k$. Let p be the value $\min\{\delta_{u'}(q', q_r) \mid u' \in \Sigma^k, \delta_{u'}(q', q_r) > 0\}$; note that this value is well defined and > 0. Let $F_0 = 0$ and for every $\ell > 0$, let $F_\ell = \delta_{\alpha[1:j_\ell]}(q_s, q_r)$, i.e., it is the probability that $\mathcal{M}$ is in state q_r after the finite prefix $\alpha[1 : j_\ell]$. We claim that for each $\ell > 0$

$$F_{\ell+1} \geq F_\ell + (x - F_\ell)p = (1-p)F_\ell + xp.$$

In order to see this, first observe that $q_r \in T_\ell$ for each $l > 0$. Also, note that it must be the case that $q_s \in T_1$ (otherwise the claim is simply true). Therefore, $F_1 = \delta_{\alpha[1:j_1]}(q_s, q_r) \geq p \geq 0 + (x-0)p = F_0 + (x - F_0)p$.

For each $q \in Q$ and $\ell > 0$, let the probability of being in state q after reading the input $\alpha[1 : j_\ell]$ be denoted as $pr_\ell(q)$. Note that $pr_\ell(q) = \delta_{\alpha[1:j_\ell]}(q_s, q)$, $pr_\ell(q_r) = F_\ell$ and $\sum_{q \in T_{j_\ell}} pr_\ell(q) = \delta_{\alpha[1:j_\ell]}(q_s, T_{j_\ell}) \geq x$. Also, for each $q \in Q$ and $\ell > 0$, let $pr_{\ell+1}(q_r|q)$ be the probability of being in state q_r after reading the input $\alpha[j_\ell + 1 : j_{\ell+1}]$ having started in the state q. In other words, let $pr_{\ell+1}(q_r|q) = \delta_{\alpha[j_\ell+1:j_{\ell+1}]}(q, q_r)$. Note that $pr_{\ell+1}(q_r|q_r) = 1$, $pr_{\ell+1}(q_r|q) \geq p$ for each $q \in T_{j_\ell}$ and $pr_{\ell+1}(q_r|q) = 0$ for all $q \in Q \setminus T_{j_\ell}$. It follows from definition that for each $\ell > 1$,

$$\begin{aligned}
F_{\ell+1} &= \delta_{\alpha[1:j_{\ell+1}]}(q_s, q_r) \\
&= \textstyle\sum_{q \in Q} pr_\ell(q) pr_{\ell+1}(q_r|q) \\
&= pr_\ell(q_r) pr_{\ell+1}(q_r|q_r) + \textstyle\sum_{q \in T_{j_\ell} \setminus \{q_r\}} pr_\ell(q) pr_{\ell+1}(q_r|q) + \textstyle\sum_{q \in Q \setminus T_\ell} pr_\ell(q) pr_{\ell+1}(q_r|q) \\
&= F_\ell + \textstyle\sum_{q \in T_{j_\ell} \setminus \{q_r\}} pr_\ell(q) pr_{\ell+1}(q_r|q) + 0 \\
&\geq F_\ell + \textstyle\sum_{q \in T_{j_\ell} \setminus \{q_r\}} pr_\ell(q) p \\
&\geq F_\ell + p((\textstyle\sum_{q \in T_{j_\ell}} pr_\ell(q)) - pr_\ell(q_r)) \\
&\geq F_\ell + p(x - F_\ell).
\end{aligned}$$

Thus, we have that for each ℓ, $F_{\ell+1} \geq F_\ell + (x - F_\ell)p = (1-p)F_\ell + xp$. By induction on ℓ, it follows that

$$F_{\ell+1} \geq (1-p)^l F_0 + xp \sum_{0 \leq r \leq \ell} (1-p)^r$$

From this, we see that,

$$(\lim_{\ell\to\infty} F_\ell) \geq xp \sum_{0\leq r<\infty} (1-p)^r \;=\; x$$

Hence $\mu^{rej}_{\mathcal{M},\alpha} \geq x$, which contradicts the fact that $\alpha \in \mathcal{R}_{<x}(\mathcal{M})$. $\square$

An immediate consequence of the proof of Lemma 6.2 is that non-emptiness of $\mathcal{R}_{<x}(\mathcal{M})$ for a FPM $\mathcal{M}$ means that there is an *ultimately periodic* word in $\mathcal{R}_{<x}(\mathcal{M})$.

COROLLARY 6.3. *Let $\mathcal{M} = (Q, q_s, q_r, \delta)$ be a* FPM *on an alphabet Σ, and $x \in (0,1]$. $\mathcal{R}_{<x}(\mathcal{M}) \neq \emptyset$ if and only if there exist $u, v \in \Sigma^*$ such that $uv^\omega \in \mathcal{R}_{<x}(\mathcal{M})$.*

We thus obtain the upper bounds for checking the emptiness of MWA and MSC.

THEOREM 6.4. *Let $\mathcal{M} = (Q, q_s, q_r, \delta)$ be a RatFPM on an alphabet Σ, and rational $x \in (0,1)$ be a rational number. Emptiness of $\mathcal{R}_{<1}(\mathcal{M})$ can be determined in* **PSPACE**, *while emptiness of $\mathcal{R}_{<x}(\mathcal{M})$ can be determined in* **co-R.E.**.

PROOF. From Lemma 6.2, we know that if $\mathcal{R}_{<x}(\mathcal{M})$ is non-empty then there is u, S such that S is on a good cycle and $\delta_u(q_s, S) > 1 - x$. So the algorithm for checking non-emptiness, non-deterministically guesses $S \subseteq Q$ and the string u. Then the semi-decision first checks that S is on a good cycle of $G(\mathcal{M})$, which can be done using space that is polynomial in the size of $\mathcal{M}$ without explicitly constructing $G(\mathcal{M})$. Next, it is easy to see that there is a semi-decision procedure to check if $\delta_u(q_s, S) > 1 - x$. This proves that the non-emptiness of $\mathcal{R}_{<x}(\mathcal{M})$ is recursively enumerable.

Based on the observations in the previous paragraph, in order to prove that non-emptiness of $\mathcal{R}_{<1}(\mathcal{M})$ is in **PSPACE**, all we need to show is that the check $\delta_u(q_s, S) > 0$ can be accomplished in **PSPACE**. Notice that $\delta_u(q_s, S) > 0$ iff the vertex S' reached on the sequence u in $G(\mathcal{M})$ is such that $S \cap S' \neq \emptyset$. Thus the **PSPACE** upper bound can be shown by observing that this check can be done by guessing the symbols of u incrementally following a path in $G(\mathcal{M})$. $\square$

MNC**:** We will conclude this section on upper bounds by showing that the emptiness problem for MNC is recursively enumerable.

THEOREM 6.5. *Let $\mathcal{M} = (Q, q_s, q_r, \delta)$ be a RatFPM on an alphabet Σ, and rational $x \in (0,1)$ be a rational number. The problem of checking the emptiness of $\mathcal{R}_{\leq x}(\mathcal{M})$ is in* **R.E.**.

PROOF. A collection $W \subseteq \Sigma^+$ of finite strings will be called a *witness* if for every $u \in W$, $\delta_u(q_s, q_r) > x$ and for every $\alpha \in \Sigma^\omega$, there is $u \in W$ such that u is a prefix of α. Observe that if there is a witness set W then for every $\alpha \in \Sigma^\omega$, $\mu^{rej}_{\mathcal{M},\alpha} > x$, and so $\mathcal{R}_{\leq x}(\mathcal{M}) = \emptyset$.

Our main observation is that $\mathcal{R}_{\leq x}(\mathcal{M}) = \emptyset$ if and only if there is a *finite* set $W \subseteq \Sigma^+$ such that W is a witness. Before proving this, let us note that this gives us a semi-decision procedure to check the emptiness of $\mathcal{R}_{\leq x}(\mathcal{M})$. The algorithm to check emptiness will guess a finite set of finite strings W, and check that W is indeed a witnessing set. It is easy to see that checking if W is a witness is decidable, which proves the theorem.

We now prove the key technical claim that $\mathcal{R}_{\leq x}(\mathcal{M}) = \emptyset$ if and only if there is a *finite* set $W \subseteq \Sigma^+$ such that W is a witness. Clearly, the existence of a finite witnessing set implies that $\mathcal{R}_{\leq x}(\mathcal{M}) = \emptyset$, and so the main challenge is in proving the converse.

Let $\mathcal{R}_{\leq x}(\mathcal{M}) = \emptyset$. Consider the set $G = \{u \in \Sigma^* \mid \delta_u(q_s, q_r) \leq x\}$. Note that the empty string $\epsilon \in G$. The set G can be viewed as vertices of a Σ-branching tree, because G is prefix closed. We will abuse notation and view G both as a tree and a collection of strings. Suppose G is not finite. Then by König's Lemma, G has an infinite path, which means that there is $\alpha \in \Sigma^\omega$ such that every prefix of α is in G. This means that for every prefix u of α, $\delta_u(q_s, q_r) \leq x$, and hence $\mu^{rej}_{\mathcal{M},\alpha} \leq x$. This contradicts our assumption that $\mathcal{R}_{\leq x}(\mathcal{M}) = \emptyset$. Hence G must be finite.

Consider the set $W = (G\Sigma) \setminus G$. Observe that W is finite. Second, since $W \subseteq \Sigma^+ \setminus G$, we have for every $u \in W$, $\delta_u(q_s, q_r) > x$. Finally, since G is finite, for every $\alpha \in \Sigma^\omega$, there are infinite number of prefixes of α that are not in G; from this, it follows that there is a $u \in W$ such that u is a prefix of α. This shows that W is a finite witness set, and this completes the proof of the theorem. $\square$

6.2 Emptiness Problem: Lower Bounds

In this section, we will show that the upper bounds proved in Section 6.1 for the various classes of monitors are tight. We will first prove lower bounds for MWA and MSA, before we consider the classes MSC and MNC.

MWA **and** MSA**:** We will show the **PSPACE**-hardness of the emptiness problem for MWA and MSA.

THEOREM 6.6. *For a RatFPM $\mathcal{M}$, the problems of checking the emptiness of $\mathcal{R}_{\leq 0}(\mathcal{M})$ and $\mathcal{R}_{<1}(\mathcal{M})$ are* **PSPACE**-*hard.*

PROOF. The **PSPACE**-hardness problem for $\mathcal{R}_{<1}(\mathcal{M})$ might appear deceptively similar to the **PSPACE**-hardness of the universality problem of non-deterministic automata; however, we could not find any easy reduction from the later to the former problem. We prove the hardness result by reducing the membership problem for a language in **PSPACE**. Let $L \in$**PSPACE** and let M be a single tape deterministic Turing machine that accepts L in space $p(n)$ for some polynomial p where n is the length of its input. Without loss of generality, let $M = (Q, \Sigma, \Gamma, B, \delta, q_0, q_a, q_r)$, where Q is a finite set of control states; Σ, Γ are the input and tape alphabets respectively, such that $\Sigma \subsetneq \Gamma$; $B \in \Gamma \setminus \Sigma$ is the blank symbol; $\delta : Q \times \Gamma \to Q \times \Gamma \times \{-1, 1\}$ is the transition function, with -1 denoting moving the tape head left, and 1 denoting moving the tape head right; $q_0 \in Q$ is the initial state; $q_a \in Q$ is the unique accepting state, i.e., if $x \in L$ then M on input x eventually reaches control state q_a; $q_r \in Q$ is the unique rejecting state, i.e., if $x \notin L$ then M on input x eventually reaches q_r. Let $c = |\Gamma| + |(\Gamma \times Q)|$, $m = p(n)$ and $k = c^m$. Without loss of generality, we can make the following simplifying assumptions about M.

1. M cannot take any further steps once its control state is either q_a or q_r. Thus, q_a and q_r are halting states of the machine M.

2. When started in any configuration, M reaches one of the halting states q_a or q_r within k steps.

Let $\mathrm{Config} = \Gamma^*(\Gamma \times Q)\Gamma^*$. A configuration of M on an input of length n is a string of length $m = p(n)$ in Config, where the unique symbol in $\Gamma \times Q$ indicates the head position as well as the control state. For a configuration $s = s_1 s_2 \cdots s_m \in \mathrm{Config}$, $st(s)$ denotes the control state in s, $pos(s)$ denotes the position of the head, and $sym(s)$ denotes the input symbol being scanned. More formally, if $i = pos(s)$ then $s_i = (st(s), sym(s))$.

Let $\Sigma_n = \{1, \ldots p(n)\} \times (\Gamma \times Q)$. For a configuration s, $strip(s)$ will denote the triple $(pos(s), (sym(s), st(s))) \in \Sigma_n$ and for a sequence of configurations $\rho = s_1, s_2, \ldots s_\ell$, $strip(\rho) \in \Sigma_n^*$ is the word $strip(s_1)strip(s_2) \cdots strip(s_\ell)$.

Recall that a *valid computation* of M starting from a configuration s is a sequence of configurations $s_1, s_2, \ldots s_\ell$ such that $s = s_1$ and for each $i < \ell$, s_{i+1} follows from s_i by one step of M. The initial configuration on an input of size n is of the form $(\Sigma \times \{q_0\})\Sigma^{n-1}B^{m-n}$. We will say that a word $u \in \Sigma_n^*$ is *valid from configuration* s iff there is a valid computation ρ starting from s such that $u = strip(\rho)$; observe that because M is deterministic there is at most one valid computation ρ such that $u = strip(\rho)$. Finally we will say $u \in \Sigma_n^*$ is valid, if it is valid from some configuration s.

Let $\sigma = \sigma_1\sigma_2 \cdots \sigma_n$ be an input of length n to M. We will construct an RatFPM $\mathcal{C}_\sigma$ such that $\mathcal{R}_{<1}(\mathcal{C}_\sigma) \neq \emptyset$ (and $\mathcal{R}_{\leq 0}(\mathcal{C}_\sigma) \neq \emptyset$) if and only if $\sigma \in L$. Formally, $\mathcal{C}_\sigma = (Q^{\mathcal{C}}, q_s^{\mathcal{C}}, q_r^{\mathcal{C}}, \delta^{\mathcal{C}})$ will be a RatFPM over the alphabet $\Sigma_{\mathcal{C}} = \Sigma_n \cup \{\tau\}$. The set of states $Q^{\mathcal{C}} = \{q_s^{\mathcal{C}}, q_r^{\mathcal{C}}\} \cup (\{1, \ldots, m\} \times \Gamma \times Q) \cup (\{1, \ldots, m\} \times \Gamma)$. Thus, a state of $\mathcal{C}_\sigma$ is either $q_s^{\mathcal{C}}$, or $q_r^{\mathcal{C}}$, or of the form (i, a), where $1 \leq i \leq m$ and $a \in \Gamma \cup (\Gamma \times Q)$. Intuitively, $\mathcal{C}_\sigma$ in state (i, a) denotes that i^{th} element of the current configuration of the computation of M has value a.

Before defining the transitions of $\mathcal{C}_\sigma$, we define some concepts that we find useful. Consider any symbol $(i, (b, q)) \in \Sigma_n$. Note that such a triple can either be an input symbol or a state of $\mathcal{C}_\sigma$. Let $\delta(q, b) = (q', c, d)$. For such a pair (i, a), we define $next\_state(i, (b, q))$ and $next\_val(i, (b, q))$ to be q' and c, respectively. We also define $next\_pos(i, (b, q))$ to be $i + d$. Given two such triples $(i, (b, q))$ and $(i', (b', q'))$, we say that $(i', (b', q'))$ is a successor of $(i, (b, q))$ iff $i' = next\_pos(i, (b, q))$ and $q' = next\_state(i, (b, q))$.

The transitions of $\mathcal{C}_\sigma$ are defined as follows. From the initial state $q_s^{\mathcal{C}}$, on input τ, there are transitions to the states (i, u_i) , for each $i \in \{1, ..., m\}$ where $u_1 = (1, (\sigma_1, q_0))$, and for $1 < i \leq n$, $u_i = (i, \sigma_i)$, and for $n < i \leq m$, $u_i = (i, B)$; the probability of each of these transitions is $\frac{1}{m}$. Thus the input τ "sets up" the initial configuration when $\mathcal{C}_\sigma$ is in the initial state $q_s^{\mathcal{C}}$. From every other state on input τ there is a transition to the reject state $q_r^{\mathcal{C}}$ with probability 1. Also, from the state $q_s^{\mathcal{C}}$, on every input other than τ there is a transition to $q_r^{\mathcal{C}}$ with probability 1. From $q_r^{\mathcal{C}}$, on every input there is a transition back to itself with probability 1.

We now define the transitions on the input symbols in Σ_n. Consider an input symbol x of the form $(i, (b, q))$. Intuitively, such an input denotes $strip(s)$ of the next configuration s in the computation; thus, it denotes the new head position, new state, and the new symbol being scanned. If $q = q_r$, then from every state there is a transition to the reject state $q_r^{\mathcal{C}}$ with probability 1 on input x. We have the following transitions on input x for the case when $q = q_a$. From every state, of $\mathcal{C}_\sigma$, of the form (j, c), where $c \in \Gamma$, there is a transition to $q_s^{\mathcal{C}}$ with probability 1. From every state of the form $(j, (c, q'))$, there is a transition to $q_s^{\mathcal{C}}$ with probability 1 if x is a successor of $(j, (c, q'))$; otherwise there is a transition to $q_r^{\mathcal{C}}$ from $(j, (c, q'))$ with probability 1. Intuitively, these transitions allow the automaton to start from the initial state again, if the computation described by the input symbols is accepting. Next, we describe the transitions when $q \notin \{q_a, q_r\}$. From a state of the form (j, c), where $c \in \Gamma$, transitions on input $x = (i, (b, q))$ are defined as follows. If $j = i$ and $b \neq c$ then there is a single transition with probability 1 to the reject state $q_r^{\mathcal{C}}$; this means the contents of the cell specified by x does not match with the one specified by the state. If $j = i$ and $c = b$ then there is a single transition with probability 1 to the state

$(i, (b, q))$. If $j \neq i$ then there are two transitions each with probability $\frac{1}{2}$ to the states (j, c) and $(i, (b, q))$. Note that the former transition is a self loop denoting that the contents of the cell did not change; the later, called *cross transition*, is to the state denoting new head position and control state. Transitions from a state of the form $(j, (c, q'))$ on input $x = (i, (b, q))$, where $q \neq q_a, q_r$, are defined as follows. If x is a successor of the pair $(j, (c, q'))$ then there are two transitions, each with probability $\frac{1}{2}$, to the states (j, d) and $(i, (b, q))$, where $d = next\_val(j, (c, q'))$. If the above condition does not hold then there is a single transition to the reject state $q_r^{\mathcal{C}}$ with probability 1.

Before proving the correctness of the reduction, we formally spell out some properties $\mathcal{C}_\sigma$ satisfies. These properties capture the intuition behind the correctness. Consider an input sequence $u \in \Sigma_n^*$ such that u does not contain τ and does not contain any symbol of the form $(i, (b, q_a))$ or of the form $(i, (b, q_r))$ for any i, b.

A. Let ρ be a valid computation starting from configuration s and ending in configuration s' such that $u = strip(\rho)$. Suppose further that the j^{th} symbol of s is $a \in \Gamma \cup (\Gamma \times Q)$. Then on input u from state (j, a), $\mathcal{C}_\sigma$ has a non-zero probability of reaching the state $e \in Q^{\mathcal{C}}$ iff $e = (i, b)$ for some $i \in \{1, ..., m\}$, $b \in \Gamma \cup (\Gamma \times Q)$ and the i^{th} symbol of s' is b, and either $i = j$ or u contains a symbol of the form $(i, (c, q'))$ for some c, q'. This is because of the cross transitions. One consequence of this is that, on input u from state (j, a), $\mathcal{C}_\sigma$ has probability zero of reaching the rejecting state $q_r^{\mathcal{C}}$.

B. If u is not valid starting from any configuration then from any state (j, a), on input u, $\mathcal{C}_\sigma$ reaches the reject state $q_r^{\mathcal{C}}$ with probability at least $\frac{1}{2^{|u|}}$ where $|u|$ is the length of u. The above property is proved to hold as follows. Since u is not valid from any configuration, it can be shown that one of the following two conditions is satisfied: (i) there exist two input symbols x, y appearing consecutively in that order in u such that y is not a successor of u; (ii) there exist two input symbols $x = (i, (b, q)), y = (i, (c, r))$ such that x appears sometimes before y in u, no other symbol of the form $(i, (c', q'))$, for any c', q', appears in between them and $c \neq next\_val(i, (b, q))$. Let u' be the smallest prefix of u that violates condition (i) or (ii). We show that $q_r^{\mathcal{C}}$ is reachable from (j, a) on input u' by a sequence of transitions of $\mathcal{C}_\sigma$. The lower bound on the probability follows since probability of each of the transitions is at least $\frac{1}{2}$. Assume that u' violates condition (i). Then, $u' = u''xy$. The state x is reachable from (j, a) on input $u''x$ and there is a transition from state x to $q_r^{\mathcal{C}}$ on input y and hence $q_r^{\mathcal{C}}$ is reachable from (j, a). Now assume that u' violates condition (ii). In this case $u' = vxwy$ where v, w are input sequences and x, y are input symbols as given in (ii). It should be easy to see that the states $x, (i, next\_val(x))$, respectively, are reachable from (j, a) on the input sequences vx, vxw. From this, we see that $q_r^{\mathcal{C}}$ is reachable from (j, a) on input u' since there is a transition from $(i, next\_val(x))$ on the input symbol y to $q_r^{\mathcal{C}}$, as $c \neq next\_val(x)$.

C. Let s_0 be the initial configuration, i.e., $s_0 = (\sigma_1, q_0), \sigma_2, ..., \sigma_n$. Let $u \in \Sigma_n^*$ be such that it is not valid from s_0. Then, the finite string τu is rejected by $\mathcal{C}_\sigma$ with probability greater than or equal to $(\frac{1}{2})^{|u|}$.

We will now show that our construction of $\mathcal{C}_\sigma$ satisfies the following property. If ρ is the accepting computation of M on σ then $\mu^{acc}_{\mathcal{C}_\sigma, \alpha} = 1$, where $\alpha = (\tau strip(\rho))^\omega$. On the other hand, if M does not accept σ then $\mu^{rej}_{\mathcal{C}_\sigma, \alpha} = 1$ for every $\alpha \in \Sigma_{\mathcal{C}}^\omega$. Based on this property, $\mathcal{R}_{\leq 0}(\mathcal{C}_\sigma) \neq \emptyset$ and $\mathcal{R}_{<1}(\mathcal{C}_\sigma) \neq \emptyset$ iff M accepts σ, which proves the **PSPACE**-hardness of the non-emptiness problem. We now prove that our claim holds.

Let ρ be the accepting computation of M on σ. Observe that on input $(i,(b,q_a))$, $\mathcal{C}_\sigma$ goes to the initial state $q_s^{\mathcal{C}}$ with probability 1 from every state except the reject state $q_r^{\mathcal{C}}$. This coupled with property **A** above, ensures that $\delta_u^{\mathcal{C}}(q_s^{\mathcal{C}}, q_s^{\mathcal{C}}) = 1$, where $u = \tau strip(\rho)$. Thus, u^ω is accepted with probability 1 by $\mathcal{C}_\sigma$.

Now we prove the more difficult part of the claim, namely, that if M rejects σ, $\mathcal{C}_\sigma$ rejects every input sequence with probability 1. The proof is by cases on the form of the input word α to $\mathcal{C}_\sigma$. Observe that if α is not of the right form, i.e., does not begin with τ or does not have a τ immediately following a symbol of the form $(i,(b,q_a))$ or every τ (except the first one) is not preceded by a symbol of the form $(i,(b,q_a))$ then α is rejected with probability 1. Next if α contains any symbol of the form $(i,(b,q_r))$ then also α is rejected with probability 1.

Let us now consider the case when α has infinitely many symbols of the form $(i,(b,q_a))$. Since $\mathcal{C}_\sigma$ will reject any input in which such symbols are not immediately followed by τ, we can assume without loss of generality that $\alpha = \tau u_1 \tau u_2 \tau \cdots$, where $u_i \in \Sigma_n^*$ and ends with a symbol of the form $(i,(b,q_a))$. Now, since σ is rejected by M, for every $i \geq 1$, we have the following properties. The sequence u_i is not valid from the initial configuration. Hence from property **C**, $\mathcal{C}_\sigma$, on input τu_i, reaches the reject state with probability at least $(\frac{1}{2})^{|u_i|}$. From the simplifying assumption 2, we have $|u_i| \leq k$, and hence the probability of rejection of τu_i is at least $(\frac{1}{2})^k$. Hence, $\mu^{rej}_{\mathcal{C}_\sigma, \alpha} = 1$.

Finally, suppose α has only finitely many symbols of the form $(i,(b,q_a))$ (each of which is followed by τ) and no symbols of the form $(i,(b,q_r))$. Observe that we may also assume that every τ symbol (except the first) is immediately preceded by a symbol of the form $(i,(b,q_a))$ (as otherwise α will be rejected with probability 1). Thus $\alpha = u'\tau\beta$, where $\beta \in \Sigma_n^\omega$ and does not contain any symbol of the form $(i,(b,q_a))$ or $(i,(b,q_r))$. Let us divide β into sequences of length k, i.e., $\beta = u_1 u_2 \cdots$, where $|u_i| = k$. Based on the simplifying assumption 2, made about M, we can conclude that each u_i is not valid (from any configuration). Let S_0 be the set of states of $\mathcal{C}_\sigma$ reached after the input sequence $u'\tau$, and for $j \geq 1$, let S_j be the set of states of $\mathcal{C}_\sigma$ reached after the input sequence $u'\tau u_1 ... u_j$. From the property **B**, we see that for $j \geq 1$, from every state p' in S_{j-1}, the automaton state $q_r^{\mathcal{C}}$ is reachable on the input sequence u_j with probability at least $\frac{1}{2^k}$. Hence we see that α is rejected with probability 1. $\square$

MSC**:** We will show that the emptiness problem for MSC is **co-R.E.**-hard. The proof will rely on **co-R.E.**-hardness of the emptiness problem for Probabilistic Finite Automata (PFA). Therefore, before presenting our proof, we recall the basic definitions associated with such automata. Formally, a PFA over alphabet Σ is $\mathcal{M} = (Q, q_s, F, \delta)$, where Q is a finite set of states, $q_s \in Q$ is the initial state, $F \subseteq Q$ are the final states, and $\delta : Q \times \Sigma \times Q \to [0,1]$ is the probabilistic transition function that satisfies the same properties as the one for FPMs. Given $x \in [0,1]$, $\mathcal{L}_{>x}(\mathcal{M}) = \{u \in \Sigma^* \mid \sum_{q \in F} \delta_u(q_s, q) > x\}$. The main result that we will use is the following.

THEOREM 6.7 CONDON-LIPTON [CONDON AND LIPTON 1989]. *Given a* PFA $\mathcal{M}$ *over alphabet* Σ *the problem of determining if* $\mathcal{L}_{>\frac{1}{2}}(\mathcal{M}) = \emptyset$ *is* **co-R.E.**-*complete.*

Using this result we can show,

THEOREM 6.8. *For a* FPM $\mathcal{M}$ *over an alphabet* Σ, *and rational* $x \in (0,1)$ *the problem of determining if* $\mathcal{R}_{<x}(\mathcal{M}) = \emptyset$ *is* **co-R.E.**-*hard.*

PROOF. We will actually prove this result for the case when $x = \frac{1}{2}$; Lemma 4.10 will then allow us to conclude for any $x \in (0,1)$. The proof of **co-R.E.**-hardness relies on a reduction from the emptiness problem of PFAs. Let $\mathcal{M} = (Q, q_s, F, \delta)$ be a PFA over the alphabet Σ. We will construct a FPM $\mathcal{M}'$ such that $\mathcal{L}_{>\frac{1}{2}}(\mathcal{M}) = \emptyset$ iff $\mathcal{R}_{<\frac{1}{2}}(\mathcal{M}') = \emptyset$.

Pick a new symbol $\tau \notin \Sigma$ and let $\Sigma' = \Sigma \cup \{\tau\}$. Formally $\mathcal{M}' = (Q', q_s, q_r, \delta')$ will be a FPM over alphabet Σ'. The set of states $Q' = Q \cup \{q_a, q_r\}$, where q_a, q_r will be assumed to be new states not in Q. δ' will be defined as follows. First we describe transitions out of the new states q_a and q_r: for every $a \in \Sigma'$, $\delta'(q_a, a, q_a) = \delta'(q_r, a, q_r) = 1$. Next we describe the transitions on the new symbol τ: if $q \in F$ then $\delta'(q, \tau, q_a) = 1$ and if $q \in Q \setminus F$ then $\delta(q, \tau, q_r) = 1$. Finally, for all the old states $q \in Q$, on all the input symbols $a \in \Sigma$, we will define transitions as follows:

$$\delta'(q, a, q') = \begin{cases} \frac{1}{3} & \text{if } q' = q_a \text{ or } q' = q_r \\ \frac{1}{3}\delta(q, a, q') & \text{if } q' \in Q \end{cases}$$

We will now prove the correctness of the above reduction. On any string $\alpha \in \Sigma^\omega$ (i.e., when α does not have any τ symbols), we have $\mu^{rej}_{\mathcal{M}',\alpha} = \sum_{i=1}^{\infty} (\frac{1}{3})^i = \frac{1}{2}$. Thus, $\mathcal{R}_{<\frac{1}{2}}(\mathcal{M}') \cap \Sigma^\omega = \emptyset$. Let us now consider $\alpha \in (\Sigma')^\omega \setminus \Sigma^\omega$. Such a word can be written as: $\alpha = u\tau\alpha'$, where $u \in \Sigma^*$. Let $\rho = \sum_{q \in F} \delta_u(q_s, q)$, i.e., the acceptance probability of u in $\mathcal{M}$. Now, suppose $\rho > \frac{1}{2}$. Then

$$\begin{aligned} \mu^{acc}_{\mathcal{M}',\alpha} &\geq (\textstyle\sum_{i=1}^{|u|} (\frac{1}{3})^i) + (\frac{1}{3})^{|u|}\rho \\ &= \tfrac{1}{2}(1 - (\tfrac{1}{3})^{|u|}) + (\tfrac{1}{3})^{|u|}\rho > \tfrac{1}{2} \end{aligned}$$

By a symmetric argument, if $\rho \leq \frac{1}{2}$, then $\mu^{rej}_{\mathcal{M}',\alpha} \geq \frac{1}{2}$. Thus, a word of the form $u\tau\alpha'$ is in $\mathcal{R}_{<\frac{1}{2}}(\mathcal{M}')$ iff $u \in \mathcal{L}_{>\frac{1}{2}}(\mathcal{M})$. Hence, $\mathcal{R}_{<\frac{1}{2}}(\mathcal{M}') = \emptyset$ iff $\mathcal{L}_{>\frac{1}{2}}(\mathcal{M}) = \emptyset$. $\square$

MNC**:** We will show that the emptiness problem for MNC is **R.E.**-hard. The proof will be a highly non-trivial modification of the proof of **co-R.E.**-hardness of emptiness of PFA [Condon and Lipton 1989].

THEOREM 6.9. *For a* FPM $\mathcal{M}$ *and rational* $x \in (0,1)$ *the problem of determining if* $\mathcal{R}_{\leq x}(\mathcal{M}) = \emptyset$ *is* **R.E.**-*hard.*

PROOF. It suffices to show that the problem of determining whether $\mathcal{R}_{\leq\frac{1}{2}}(\mathcal{M})$ is non-empty is **co-R.E.**-hard. This is exhibited by a reduction from the problem of non-termination of a deterministic two-counter machine on an empty input. The reduction is carried in two steps.

The first step of the construction is a modification of the PFA used in proof of **co-R.E.**-hardness of emptiness of PFA. Given a deterministic 2-counter machine $\mathcal{C}$ we construct a monitor $\mathcal{M}$ as follows. The monitor $\mathcal{M}$ has an *absorbing accept state* q_a and an *absorbing reject state* q_r (by an absorbing state q, we mean that probability of transitioning from q to q is $\mathbf{1}$ for every input symbol). The input alphabet of $\mathcal{M}$ is the set of control states of $\mathcal{C}$, left bracket (, right bracket), $\mathbf{0}$, $\mathbf{1}$, a and b. The constructed monitor checks whether a given sequence of inputs determine a valid computation of $\mathcal{C}$ on empty input. The computation of $\mathcal{C}$ is represented as a sequence of consecutive configurations of $\mathcal{C}$. A configuration of $\mathcal{C}$ represented by the tuple (q, i, i') is input as left bracket (; followed by a control state q; followed by two bits which indicate whether the two counters are zero or not (the bit $\mathbf{0}$ a counter value of 0 and the bit $\mathbf{1}$ represents a non-zero counter value); followed by the input sequence $a^i b^{i'}$ which represent the counter values in unary; and ending in right bracket). If q_s is the initial state of $\mathcal{M}$, the construction will ensure that for any word α and $j \in \mathbb{N}$, $\delta_{\alpha[1:j]}(q_s, q_r) \geq \delta_{\alpha[1:j]}(q_s, q_a)$. If a word α represents a valid infinite computation then $\delta_{\alpha[1:j]}(q_s, q_r) = \delta_{\alpha[1:j]}(q_s, q_a)$ for all $j \geq 1$ such that $\alpha(j) =)$. If a word α does not represent a valid infinite computation then there would be a j_0 such that $\delta_{\alpha[1:j]}(q_s, q_r) > \delta_{\alpha[1:j]}(q_s, q_a)$ for all $j \geq j_0$.

The machine $\mathcal{M}$ checks if in the first input configuration, the control state is the start state of $\mathcal{C}$ and both the counter values are 0. Otherwise, it rejects the input with probability 1 (*i.e.*, makes a transition to q_r with probability 1). Similarly, if ever the monitor $\mathcal{M}$ receives an input symbol of the wrong kind (for example, if the monitor receives a state q when it is expecting a or b) then $\mathcal{M}$ rejects the rest of the input with probability 1.

The monitor $\mathcal{M}$ also has to check that consecutive input configurations, say (q, i, i') and (q', j, j'), are in accordance with the transition function of the 2-counter automaton. If there is no transition from q to q' in $\mathcal{C}$ or if one of the counter values j, j' is zero (or, non-zero) when it is not supposed to be, then the rest of the input is rejected with probability 1. The main challenge is in checking the counter values across the transition using only finite number of states. This problem reduces to checking whether two strings have equal length using only finite memory of the monitors. For the case of PFA's, this is accomplished by a *weak equality test* described in [Condon and Lipton 1989; Freivalds 1981] as follows. We recall the equality test described there. Suppose we want to check that for two strings a^i and a^j, $i = j$. Then while processing a^i

(1a). Toss $2i$ fair coins and note if all of them turned heads.

(2a). Toss a separate set of i fair coins and note if all of them turned heads.

(3a). Toss yet another set of i fair coins and note if all of them turned heads.

While processing a^j

(1b). Toss $2j$ fair coins and note if all of them turned heads.

(2b). Toss a separate set of j fair coins and note if all of them turned heads.

(3b). Toss yet another set of j fair coins and note if all of them turned heads.

As in [Condon and Lipton 1989], we define event A as *either* all coins turn up heads in (1a) *or* all coins turn up heads in (1b). We define event B as *either* all coins turn up heads in (2a) as well as (2b) *or* all coins turn up heads in (3a) as well as (3b). We reject (that is make a transition to q_r with probability 1) if event A is true and B is not true and accept (that is make a transition to q_a with probability 1) if B is true and A is not true. If $i = j$ then as explained in [Condon and Lipton 1989], the probability of acceptance and rejection is equal (note each of them may be less than $\frac{1}{2}$), otherwise probability of transitioning to reject state is strictly larger than probability of acceptance. We shall slightly modify this test. The main problem in using this test directly is that the input configuration may have an unbounded number of a's and in that case we never make a transition to either q_a or q_r. Hence, in this case, we will not be able to guarantee that there is a j_0 such that $\delta_{\alpha[1:j]}(q_s, q_r) > \delta_{\alpha[1:j]}(q_s, q_a)$ for each $j > j_0$.

We modify the weak equality test as follows. We will still conduct the experiments (1a), (2a), (3a), as above. Let A_0 be the event such that all coins turn up heads in (1a). Now, when we process a^j we will still conduct experiments (1b), (2b) and (3b) but take certain extra actions while scanning the input as described below.

—Assume first the case that event A_0 is true. Now, while scanning a^j, we check if both of the following events happen- at least one coin turns up tails in (2a) or (2b), and at least one coin turns up tails in (3a) or (3b). If both of the above events are true then we reject the input. Note that if j is unbounded, this rejection will happen with probability 1 given that A_0 is true. Also note that if j is bounded and if both the above events happened, then the input would have been rejected anyways in the original weak equality test. If j is bounded and we have not rejected the input, then at the end of scanning of a^j we check for events A and B as above. If A is true and B is false, we reject the input. If A is false and B is true, then we accept the input. Otherwise we continue processing the rest of the input.

—Assume now that A_0 is false. Now, while scanning a^j, check if all three of the following events happen- at least one coin turns up tails in (1b), at least one coin turns up tails in (2a) or (2b), and at least one coin turns up tails in (3a) or (3b). If all three of the above events are true then we start accepting (that is transiting to q_a) and rejecting (that is transiting to q_r) the succeeding input with probability $\frac{1}{3}$ each until we detect the end of a^j. Note if j is unbounded, then with probability 1 all three events will happen and asymptotically we will accept and reject with probability $\frac{1}{2}$. If j is bounded, then at the end of scanning of a^j we check for events A and B as above. (note if the above three events happen at some point while scanning a^j then both A and B are going to be false). If A is true and B is false, we reject the input. If A is false and B is true, then we accept the input. Otherwise we continue processing the rest of the input.

These tests ensure that if j is unbounded, then the probability of transitioning to the reject state is $> \frac{1}{2}$. In the case j is bounded and $i = j$ then probability of transitioning to accept state q_a is exactly equal to the probability of transitioning to reject state q_r. If j is bounded and $i \neq j$ then probability of transitioning to accept state q_a is strictly less than the probability of transitioning to reject state q_r.

Finally, while checking the input, the monitor $\mathcal{M}$ rejects if the state of the current configuration is a halting state. Let q_s be the start state of the monitor $\mathcal{M}$ and δ the transition function of $\mathcal{M}$. For α, let $\delta_\alpha(q_s, q_a) = \lim_{j\to\infty} \delta_{\alpha[1:j]}(q_s, q_a)$ and $\delta_\alpha(q_s, q_r) = \lim_{j\to\infty} \delta_{\alpha[1:j]}(q_s, q_r)$. It is easy to see that for the monitor $\mathcal{M}$, a finite word u and an infinite word α, the following facts are true–

(1) $\delta_u(q_s, q_r) \geq \delta_u(q_s, q_a)$.

(2) If u represents a valid finite computation of the counter machine $\mathcal{C}$, u does not contain the halting state and u ends in the symbol) then $\delta_u(q_s, q_r) = \delta_u(q_s, q_a) < \frac{1}{2}$.

(3) If u contains a halting state then $\delta_u(q_s, q_r) > \frac{1}{2}$.

(4) If the input α does not represent a valid infinite computation then there is some j_0 such that $\delta_{\alpha[1:j]}(q_s, q_r) > \delta_{\alpha[1:j]}(q_s, q_a)$ for all $j > j_0$.

(5) If there is some unbounded counter in some input configuration, then $\delta_\alpha(q_s, q_r) > \frac{1}{2}$. In particular, if α does not contain an infinite number of states of the counter machine $\mathcal{C}$ then $\delta_\alpha(q_s, q_r) > \frac{1}{2}$. Furthermore, if word α does not contain an infinite number of states of the counter machine then $\delta_\alpha(q_s, q_a) + \delta_\alpha(q_s, q_r) = 1$.

(6) If the input α represents a valid infinite computation then $\delta_\alpha(q_s, q_r) = \delta_\alpha(q_s, q_a)$. Furthermore, if $\alpha(j) =)$ then $\delta_{\alpha[1:j]}(q_s, q_r) = \delta_{\alpha[1:j]}(q_s, q_a)$.

Now, we obtain $\mathcal{M}_1$ by modifying $\mathcal{M}$ as follows. Whenever we process the control state in a new input configuration, we accept (that is make a transition to q_a) and reject (that is make a transition to q_r) with probability $\frac{1}{3}$. It is easy to see that for $\mathcal{M}_1$ and word α, we have that

(1) If α contains a halting state or represents an invalid computation then $\mu^{rej}_{\mathcal{M}_1,\alpha} > \frac{1}{2}$.

(2) If the input α represents a valid infinite computation then $\mu^{rej}_{\mathcal{M}_1,\alpha} = \frac{1}{2}$.

Thus, $\mathcal{R}_{\leq\frac{1}{2}}(\mathcal{M}_1)$ is non-empty iff $\mathcal{C}$ has an infinite computation. $\square$

6.3 Universality Problem: Upper Bounds

In this section, we will establish upper bounds for the universality problem of RatFPM's.

MSA**:** The universality problem for MSA is easily seen to be in **NL**.

THEOREM 6.10. *For a RatFPM $\mathcal{M}$ on alphabet Σ, the problem of checking the universality of $\mathcal{R}_{\leq 0}(\mathcal{M})$ is in* **NL**.

PROOF. Let $\mathcal{M} = (Q, q_s, q_r, \delta)$ and let Σ be the alphabet. Consider the direct graph $\mathsf{G} = (V, E)$ constructed (in log-space) as follows. The set of vertices V is Q and $(q_1, q_2) \in E$ iff there exists an $a \in \Sigma$ such that $\delta(q_1, a, q_2) > 0$. It is easy to see that $\mathcal{R}_{\leq 0}(\mathcal{M})$ is not universal iff there is a directed path from q_s to q_r in G. The result follows from the fact that the latter problem is known to be in **NL**, and **NL**=co-**NL** [Szelepcsenyi 1987; Immerman 1988]. $\square$

MWA**:** We will show that the universality problem for MWA is in **PSPACE**. The proof depends on showing that the set $\mathcal{R}_{=1}(\mathcal{M}) = \{\alpha \in \Sigma^\omega | \mu^{rej}_{\mathcal{M},\alpha} = 1\}$ contains an ultimately periodic word.

LEMMA 6.11. *Given a* FPM $\mathcal{M} = (Q, q_s, q_r, \delta)$*, a state $q \in Q$ and a finite word $u \in \Sigma^+$, let $\mathsf{post}(q, u) = \{q' \in Q \mid \delta_u(q, q') > 0\}$. The set $\mathcal{R}_{=1}(\mathcal{M}) = \{\alpha \in \Sigma^\omega | \mu^{rej}_{\mathcal{M},\alpha} = 1\}$ is non-empty iff there are $u, v \in \Sigma^+$ such that the following conditions hold–*

—$\mathsf{post}(q_s, u) = \mathsf{post}(q_s, uv)$

—*For each $q \in \mathsf{post}(q_s, u)$, $q_r \in \mathsf{post}(q, v)$.*

PROOF. ($\Rightarrow$). Fix α such that $\mu^{rej}_{\mathcal{M},\alpha} = 1$. For each integer $j > 0$, let $Q_j = \mathsf{post}(q_s, \alpha[1 : j])$. Now since $Q_j \subseteq Q$ and Q is a finite set, there exists an infinite sequence of natural numbers $1 < j_1 < j_2 < j_3, \ldots$ such $Q_{j_r} = Q_{j_s}$ for all $r, s \in \mathbb{N}$. Let $u = \alpha[1 : j_1]$. As $\mu^{rej}_{\mathcal{M},\alpha} = 1$, it must be the case that for each $q \in \mathsf{post}(q_s, u)$ there is some $k_q \geq 1$ such that $\delta_{\alpha[j_1+1, j_1+k_q]}(q, q_r) > 0$. Pick j_r such that $j_r \geq j_1 + k_q$ for each $q \in \mathsf{post}(q_s, u)$. Let $v = \alpha[j_1 + 1, j_r]$. Clearly u, v are the required words.

($\Leftarrow$). Suppose that u, v are such that $\mathsf{post}(q_s, u) = \mathsf{post}(q_s, uv)$ and for each $q \in \mathsf{post}(q_s, u)$, $q_r \in \mathsf{post}(q, v)$. From $\mathsf{post}(q_s, u) = \mathsf{post}(q_s, uv)$, we have that $\mathsf{post}(q_s, u) = \mathsf{post}(q_s, uv^j)$ for all $j \in \mathbb{N}$. Let $Q_0 = \mathsf{post}(q_s, u) = \mathsf{post}(q_s, uv^j)$.

Consider the word $\alpha = uv^\omega$. We show that $uv^\omega \in \mathcal{R}_{=1}(\mathcal{M})$. Note that we have $\mu^{rej}_{\mathcal{M},\alpha} = \lim_{j\to\infty} \delta_{uv^j}(q_s, q_r) = 1 - \lim_{j\to\infty} \delta_{uv^j}(q_s, Q_0 \setminus \{q_r\})$. Thus, it suffices to show that $\lim_{j\to\infty} \delta_{uv^j}(q_s, Q_0 \setminus \{q_r\}) = 0$. Observe that–

(1) $\delta_u(q_s, Q_0 \setminus \{q_r\}) \leq 1$.

(2) Let $x = \min_{q\in Q_0}(\delta_v(q, q_r))$. We have that $x > 0$. It is easy to see that $\delta_{uv^{j+1}}(q_s, Q_0 \setminus \{q_r\}) \leq (1 - x)\delta_{uv^j}(q_s, Q_0 \setminus \{q_r\})$.

Thus, by induction $\delta_{uv^j}(q_s, Q_0 \setminus \{q_r\}) \leq (1-x)^j$. As $x > 0$, we get that

$$\lim_{j \to \infty} \delta_{uv^j}(q_s, Q_0 \setminus \{q_r\}) = 0. \quad \square$$

We get as an immediate corollary that there is an ultimately periodic word in $\mathcal{R}_{=1}(\mathcal{M}) = \{\alpha \in \Sigma^\omega | \mu^{rej}_{\mathcal{M},\alpha} = 1\}$.

COROLLARY 6.12. *Let $\mathcal{M} = (Q, q_s, q_r, \delta)$ be a* FPM *on an alphabet Σ. Then $\mathcal{R}_{=1}(\mathcal{M}) = \{\alpha \in \Sigma^\omega | \mu^{rej}_{\mathcal{M},\alpha} = 1\}$ is non-empty iff there are $u, v \in \Sigma^+$ such that $uv^\omega \in \mathcal{R}_{=1}(\mathcal{M})$.*

THEOREM 6.13. *For a RatFPM $\mathcal{M}$ on alphabet Σ, the problem of checking the universality of $\mathcal{R}_{<1}(\mathcal{M})$ is in* **PSPACE**.

PROOF. Let $\mathcal{M} = (Q, q_s, q_r, \delta)$. The **PSPACE** algorithm actually checks the non-universality for $\mathcal{R}_{<1}(\mathcal{M})$ by appealing to Lemma 6.11. The algorithm proceeds by first guessing u incrementally and storing the "current" value of $\mathsf{post}(q_s, u)$. After the guess is complete, it stores $\mathsf{post}(q_s, u)$ in its memory and starts guessing v incrementally. While guessing v incrementally, it stores $\mathsf{post}(q, v)$ for each $q \in \mathsf{post}(q_s, u)$. After its stops guessing v it checks that a) $q_r \in \mathsf{post}(q, v)$ for each $q \in \mathsf{post}(q_s, u)$ and b) $\mathsf{post}(q_s, u) = \cup_{q \in \mathsf{post}(q_s,u)}(\mathsf{post}(q, v))$ (note that $\mathsf{post}(q_s, uv) = \cup_{q \in \mathsf{post}(q_s,u)}(\mathsf{post}(q, v))$). $\square$

MNC**:** The universality problem for MNC is easily seen to be in **co-R.E.**.

THEOREM 6.14. *For a RatFPM $\mathcal{M}$ on alphabet Σ and rational $x \in (0, 1)$, the problem of checking the universality of $\mathcal{R}_{\leq x}(\mathcal{M})$ is in* **co-R.E.**.

PROOF. Let $\mathcal{M} = (Q, q_s, q_r, \delta)$. Now $\mathcal{R}_{\leq x}(\mathcal{M})$ is not universal iff there is a word α such that $\mu^{rej}_{\mathcal{M},\alpha} > x$. The latter is true iff there is a finite word $u \in \Sigma^*$ such that $\delta_u(q_s, q_r) > x$. The result now follows. $\square$

MSC**:** The universality problem for MSC is in Π^1_1.

THEOREM 6.15. *For a RatFPM $\mathcal{M}$ on alphabet Σ and rational $x \in (0, 1)$, the problem of checking the universality of $\mathcal{R}_{<x}(\mathcal{M})$ is in Π^1_1.*

PROOF. Let $\mathcal{M} = (Q, q_s, q_r, \delta)$. Now $\mathcal{R}_{<x}(\mathcal{M})$ is not universal iff there is a word $\alpha \in \Sigma^\omega$ such that $\mu^{rej}_{\mathcal{M},\alpha} \geq x$. Now,

$$(\mu^{rej}_{\mathcal{M},\alpha} \geq x) \Leftrightarrow \forall n > 0. \exists k > 0. (\delta_{\alpha[1:k]}(q_s, q_r) > x - 1/n).$$

Thus, we get

$$\mathcal{R}_{<x}(\mathcal{M}) = \Sigma^\omega \Leftrightarrow \forall \alpha. \exists n > 0. \forall k > 0. (\delta_{\alpha[1:k]}(q_s, q_r) \leq x - 1/n).$$

From this, it is easy to see that the problem of checking the universality of $\mathcal{R}_{<x}(\mathcal{M})$ is in Π^1_1. $\square$

6.4 Universality Problem: Lower Bounds

In this section, we will show that the upper bounds proved in Section 6.3 for the various classes of monitors are tight.

MSA**:** The **NL**-hardness of the universality problem for MSA is proved by a reduction from graph reachability problem.

THEOREM 6.16. *For a RatFPM $\mathcal{M}$, the problem of checking the universality of $\mathcal{R}_{\leq 0}(\mathcal{M})$ is* **NL**-*hard.*

PROOF. We will reduce the reachability problem in directed graphs to the non-universality problem for MSA. Since reachability is **NL**-complete, and **NL**=co-**NL** [Szelepcsenyi 1987; Immerman 1988], the hardness of universality follows.

Let $\mathsf{G} = (V, E)$ be a directed graph with V as the set of vertices and E as the set of edges. Given $v_1, v_2 \in V$ the reachability problem is the problem of determining whether there is a directed path from v_1 to v_2. For the reachability problem, we can assume that for any $v \in V$, the set $E_v = \{v' \in V \mid (v, v') \in E\}$ is non-empty.

We reduce (in log-space) the reachability problem to the universality problem of MSA as follows. We pick a symbol a and let $\Sigma = \{a\}$. We construct a monitor $\mathcal{M} = (V, v_1, v_2, \delta)$ where δ is defined as follows.

$$\delta(v, a, v') = \begin{cases} 1 & \text{if } v = v' = v_2 \\ \frac{1}{|E_v|} & \text{if } (v, v') \in E \\ 0 & \text{otherwise} \end{cases}$$

It is easy to see that $\mathcal{R}_{\leq 0}(\mathcal{M}) = \Sigma^\omega$ iff there is no directed path from v_1 to v_2 in G. □

MWA**:** The **PSPACE**-hardness of the universality problem for MWA is proved by a reduction from the universality problem of Finite State Machines.

THEOREM 6.17. *For a RatFPM $\mathcal{M}$, the problem of checking the universality of $\mathcal{R}_{<1}(\mathcal{M})$ is* **PSPACE**-*hard.*

PROOF. We reduce the problem of universality of Finite State Machines to the universality of $\mathcal{R}_{<1}(\mathcal{M})$. Given a finite state machine $\mathcal{A} = (Q, \Delta, q_s, F)$ over the alphabet Σ, $q \in Q$ and $a \in \Sigma$, let $\Delta_{q,a} = \{q' \mid \delta(q, a, q') \in \Delta\}$. Please note that for universality problem, we can assume that $\Delta_{q,a}$ is non-empty, *i.e.*, $|\Delta_{q,a}| > 0$.

We reduce (in polynomial-time) the universality problem of $\mathcal{A}$ to the universality problem of MWA as follows. Pick a new symbol $\tau \notin \Sigma$ and two new states $q_a, q_r \notin Q$, and construct a RatFPM $\mathcal{M} = (Q \cup \{q_a, q_r\}, q_s, q_r, \delta)$ over $\Sigma \cup \{\tau\}$ where δ is defined as follows–

$$\delta(q, a, q') = \begin{cases} 1 & \text{if } q = q', q \in \{q_a, q_r\} \text{ and } a \in \Sigma \cup \{\tau\} \\ 1 & \text{if } q \in F, q' = q_a \text{ and } a = \tau \\ 1 & \text{if } q \in Q \setminus F, q' = q_r \text{ and } a = \tau \\ \frac{1}{|\Delta_{q,a}|} & \text{if } q, q' \in Q \text{ and } (q, a, q') \in \Delta \\ 0 & \text{otherwise} \end{cases}$$

For any word α over the alphabet $\Sigma \cup \{\tau\}$, the following are easy to check–

(1) If α does not contain τ then $\mu^{rej}_{\mathcal{M},\alpha} = 0$.

(2) If $\alpha = u\tau\beta$ and u does not contain τ, then $\mu^{rej}_{\mathcal{M},\alpha} = 1$ iff u is not accepted by $\mathcal{A}$.

Thus $\mathcal{A}$ is universal iff $\mathcal{R}_{<1}(\mathcal{M})$ is universal. The result now follows. □

MNC**:** The **co-R.E.**-hardness of determining the emptiness of $\mathcal{L}_{>\frac{1}{2}}(\mathcal{M})$ yields the **co-R.E.**-hardness of the universality problem for MNC.

THEOREM 6.18. *For a RatFPM $\mathcal{M}$ on an alphabet Σ and $x \in (0,1)$, the problem of checking the universality of $\mathcal{R}_{\leq x}(\mathcal{M})$ is* **co-R.E.**-*hard.*

PROOF. We can assume without loss of generality that $x = \frac{1}{2}$. Given a PFA $\mathcal{M}' = (Q, q_s, F, \delta)$ over the alphabet Σ, pick two new states $q_a, q_r \notin Q$ and a new symbol $\tau \notin \Sigma$. Now, construct a monitor $\mathcal{M} = \{Q \cup \{q_a, q_r\}, q_s, q_r, \delta'\}$ over $\Sigma \cup \{\tau\}$ where δ' is defined as follows–

$$\delta'(q,a,q') = \begin{cases} 1 & \text{if } q = q', q \in \{q_a, q_r\} \text{ and } a \in \Sigma \cup \{\tau\} \\ 1 & \text{if } q \in F, q' = q_r \text{ and } a = \tau \\ 1 & \text{if } q \in Q \setminus F, q' = q_a \text{ and } a = \tau \\ \delta(q,a,q') & \text{otherwise} \end{cases}$$

For any word α over the alphabet $\Sigma \cup \{\tau\}$, the following are easy to check–

(1) If α does not contain τ then $\mu^{rej}_{\mathcal{M},\alpha} = 0$.

(2) If $\alpha = u\tau\beta$ and u does not contain τ, then $\mu^{rej}_{\mathcal{M},\alpha} = \sum_{q' \in F} \delta_u(q_s, q')$.

Thus, the language $\mathcal{L}_{>\frac{1}{2}}(\mathcal{M}')$ is empty iff $\mathcal{R}_{\leq\frac{1}{2}}(\mathcal{M})$ is universal. The result follows. □

MSC**:** We now show that the universality problem for MSC is Π^1_1-hard.

THEOREM 6.19. *For a RatFPM $\mathcal{M}$ on alphabet Σ and $x \in (0,1)$, the problem of checking the universality of $\mathcal{R}_{<x}(\mathcal{M})$ is Π^1_1-hard.*

PROOF. The following problem is known to be Π^1_1-complete.
Problem: Given a *non-deterministic* two-counter machine $\mathcal{C}$ and a control state q of $\mathcal{C}$, check whether all computations of $\mathcal{C}$ visit q only a finite number of times.

Given a non-deterministic two-counter machine $\mathcal{C}$ and a state q of $\mathcal{C}$, we will construct a RatFPM such that $\mathcal{R}_{<\frac{1}{2}}(\mathcal{M})$ is universal iff computations of $\mathcal{C}$ visit q only a finite number of times. This monitor is constructed in two steps. For the first step, we construct a monitor which is a small modification of the monitor $\mathcal{M}_1$ constructed in the proof of **co-R.E.**-hardness of emptiness of $\mathcal{R}_{\leq\frac{1}{2}}$ (see Theorem 6.9). Observe that in that construction, given a *deterministic* two-counter machine $\mathcal{C}^1$ we constructed a monitor $\mathcal{M}_1 = (Q^1, q_s, q_r, \delta^1)$ with the following properties–

(1) The alphabet Σ of $\mathcal{M}_1$ consists of (,), the states of $\mathcal{C}^1$, **0**, **1**, a and b.

(2) $\mathcal{M}_1$ has two absorbing states – an absorbing accept state q_a and an absorbing reject state q_r.

(3) For all finite strings u on Σ, $\delta^1_u(q_s, q_a) < \frac{1}{2}$.

(4) For any infinite word α, $\lim_{j\to\infty} \delta^1_{\alpha[1:j]}(q_s, q_a) = \frac{1}{2}$ iff α represents an infinite valid computation of $\mathcal{C}^1$. The infinite computation is represented as a sequence of configurations $(q_0, 0, 0)(q_1, i_1, j_1)(q_2, i_2, j_2)\ldots$.

Now, given a non-deterministic two-counter machine $\mathcal{C}$, let $t_1, t_2, \ldots t_k$ be the set of all the possible transitions of $\mathcal{C}$. We choose new symbols τ^i for each t_i. The modified monitor $\mathcal{M}_2$ has as an alphabet $\Sigma_{\mathsf{new}} = \Sigma \cup \{\tau^1, \tau^2, \ldots, \tau^k\}$. $\mathcal{M}_2$ works almost exactly as $\mathcal{M}_1$ except that it has to learn which transition is being used to go from (q_l, i_l, j_l) to $(q_{l+1}, i_{l+1}, j_{l+1})$. This is done by assuming that the input computation is given in the form $(q_0, 0, 0)\tau_0(q_1, i_1, j_1)\tau_1(q_2, i_2, j_2)\ldots$. Here $\tau_l \in \{\tau^1, \tau^2, \ldots \tau^k\}$ and "informs" $\mathcal{M}_2$ which transition to check while processing the new configuration. Of course, if in between configurations, $\mathcal{M}_2$ does not receive such a symbol, $\mathcal{M}_2$ rejects the rest of the input with probability 1. It is easy to see that $\mathcal{M}_2 = (Q^2, q_s, q_r, \delta^2)$ satisfies the following properties–

(1) $\mathcal{M}_2$ has two absorbing states – an absorbing accept state q_a and and an absorbing reject state q_r.
(2) For all finite strings u on Σ, $\delta^2_u(q_s, q_a) < \frac{1}{2}$.
(3) For any infinite word α, $\lim_{j\to\infty} \delta^2_{\alpha[1:j]}(q_s, q_a) = \frac{1}{2}$ iff α represents an infinite valid computation of $\mathcal{C}$.

Now, we construct $\mathcal{M}$ as follows. The alphabet of $\mathcal{M}$ will be Σ_{new}. For the states, we will pick a new state $q_{\mathsf{new}} \notin Q^2$ and set $Q = Q^2 \cup q_{\mathsf{new}}$. The state q_{new} will be the reject state of $\mathcal{M}$. The state q_s is the start state of $\mathcal{M}$. The transition function δ of $\mathcal{M}$ is obtained by modifying δ^2 as follows. The accept state q_a of $\mathcal{M}_2$ is no longer absorbing. From q_a we will make a transition to q_{new} with probability $\frac{1}{2}$ whenever we see the input symbol q (which is the given state in the Π^1_1-hard problem). With probability $\frac{1}{2}$ we will remain in q_a on the input symbol q. In other words $\delta(q_a, q, q_{\mathsf{new}}) = \delta(q_a, q, q_a) = \frac{1}{2}$. $\delta(q_2, c, q_2') = \delta^2(q_2, c, q_2')$ for all $q_2, q_2' \in Q^2$, $c \in \Sigma_{\mathsf{new}}$ except when q_2 is q_a and c is the input symbol q. Also, $\delta(q_{\mathsf{new}}, c, q_{\mathsf{new}}) = 1$ for all $c \in \Sigma_{\mathsf{new}}$. For all other possible states and symbols, the transition probability is 0.

It is easy to see that for any word α, $\lim_{j\to\inf} \delta_{\alpha[1:j]}(q_s, q_{\mathsf{new}}) \leq \frac{1}{2}$. Furthermore, $\lim_{j\to\inf} \delta_{\alpha[1:j]}(q_s, q_{\mathsf{new}}) = \frac{1}{2}$ iff α represents a valid infinite computation of $\mathcal{C}$ and visits q infinitely often. Thus $\mathcal{R}_{<\frac{1}{2}}(\mathcal{M})$ is universal iff all computations of $\mathcal{C}$ visit q only finitely many times. $\square$

7. CONCLUSIONS

In this paper, we investigated the power of randomization in finite state monitors. We have classified the languages defined by FPMs based on the rejection probability and proved a number of results characterizing these classes. Interestingly, some of these classes allow us to go beyond safety and ω-regularity, but be within almost safety. We have also presented complexity results on the problem of checking emptiness and universality of languages defined by FPMs. In the future, we would like to explore applying the techniques developed here for practical monitoring and verification needs.

7.1 Acknowledgments

Rohit Chadha was supported in part by NSF grants CCF04-29639 and NSF CCF04-48178. A. Prasad Sistla was supported in part by NSF CCF-0742686. Mahesh Viswanathan was supported in part by NSF CCF04-48178 and NSF CCF05-09321.

REFERENCES

1997. Time rover. `http://www.time-rover.com`.

1999–2008. *Proceedings of the Workshop in Runtime Verification*. Electronic Notes in Theoretical Computer Science.

ALPERN, B. AND SCHNEIDER, F. 1985. Defining liveness. *Information Processing Letters 21*, 181–185.

AMORIUM, M. AND ROSU, G. 2005. Efficient monitoring of omega-languages. In *Proceedings of the International Conference on Computer Aided Verification'*. 364–378.

BAIER, C., BERTRAND, N., AND GRÖSSER, M. 2008. On decision problems for probabilistic büchi automata. In *11th International Conference on Foundations of Software Science and Computational Structures, FoSSaCS*. 287–301.

BAIER, C. AND GRÖβER, M. 2005. Recognizing ω-regular languages with probabilistic automata. In *Proceedings of the IEEE Symposium on Login in Computer Science*. 137–146.

BAUER, A., LEUCKER, M., AND SCHALLHART, C. 2006. Monitoring of real-time properties. In *Proceedings of the 26th Conference on Foundations of Software Technology and Theoretical Computer Science*. 260–272.

CHADHA, R., SISTLA, A. P., AND VISWANATHAN, M. 2008. On the expressiveness and complexity of randomization in finite state monitors. In *LICS 2008- 23rd Annual IEEE Symposium on Logic in Computer Science*. IEEE Computer Society, 18–29.

CHEN, F. AND ROSU, G. 2007. MOP: An efficient and generic runtime verification framework. In *Proceedings of the ACM International Conference on Object-Oriented Programming, Systems, Languages, and Applications*. 569–588.

CONDON, A. AND LIPTON, R. J. 1989. On the complexity of space bounded interactive proofs (extended abstract). In *30th Annual Symposium on Foundations of Computer Science*. 462–467.

DIAZ, M., JUANOLE, G., AND COURTIAT, J. P. 1994. Observer — a concept for formal on-line validation of distributed systems. *IEEE Transactions on Software Engineering 20,* 12, 900–913.

FREIVALDS, R. 1981. Probabilistic two-way machines. In *Mathematical Foundations of Computer Science*. 33–45.

GATES, A. AND ROACH, S. 2001. Dynamics: Comprehensive support for run-time monitoring. Vol. 55. Elsevier Publishing.

GONDI, K., PATEL, Y., AND SISTLA, A. P. 2009. Monitoring the full range of omega-regular properties of stochastic systems. In *Proceedings of International Conference on Verification, Model Checking and Abstract Interpretation*. Lecture Notes in Computer Science, vol. 5403. Springer, 105–119.

GROUP, N. 2001. Intrusion detection systems — Group Test.

HAMLEN, K. W., MORRISETT, G., AND SCHNEIDER, F. 2006. Computability classes for enforcement mechanisms. *ACM Trans. Program. Lang. Syst. 28,* 1, 175–205.

HAVELUND, K. AND ROSU, G. 2001. Monitoring Java Programs with JavaPathExplorer. Vol. 55. Elsevier Publishing.

IMMERMAN, N. 1988. Nondeterministic Space is Closed under Complementation. *SIAM Journal of Computing 17*, 935–938.

JEFFERY, C., ZHOU, W., TEMPLER, K., AND BRAZELL, M. 1998. A lightweight architechture for program execution monitoring. In *ACM Workshop on Program Analysis for Software Tools and Engineering*.

KEMENY, J. AND SNELL, J. 1976. *Denumerable Markov Chains*. Springer-Verlag.

KIM, M., VISWANATHAN, M., B.-ABDULLAH, H., KANNAN, S., LEE, I., AND SOKOLSKY, O. 1999. Formally specified monitoring of temporal properties. In *Proceedings of the IEEE Euromicro Conference on Real Time Systems*. 114–122.

KORTENKAMP, D., MILAM, T., SIMMONS, R., AND FERNANDEZ, J. L. 2001. Collecting and Analyzing Data from Distributed Control Programs. Vol. 55. Elsevier Publishing.

LAMPORT, L. 1985. Logical foundation, distributed systems- methods and tools for specification. *Springer-Verlag Lecture Notes in Computer Science 190*.

LIGATTI, J. 2006. Policy enforcement via program monitoring. Ph.D. thesis, Princeton University.

LIGATTI, J., BAUER, L., AND WALKER, D. 2005. Edit automata: Enforcement mechanisms for run-time security policies. *International Journal of Information Security 4,* 1–2 (Feb.), 2–16.

LIGATTI, J., BAUER, L., AND WALKER, D. 2009. Run-Time Enforcement of Nonsafety Policy. *ACM Transactions on Information and System Security 12,* 3, 1–41.

MANSOURI-SAMANI, M. AND SLOMAN, M. 1992. Monitoring Distributed Systems (A Survey). Research Report DOC92/93, Imperial College, London.

MARGARIA, T., SISTLA, A. P., STEFFEN, B., AND ZUCK, L. D. 2005. Taming interface specifications. In *Proceedings of 16th International Conference on Concurrency Theory (CONCUR)*. 548–561.

NASU, M. AND HONDA, N. 1968. Fuzzy events realized by finite probabilistic automata. *Information and Control 12,* 4, 284–303.

PAPOULIS, A. AND PILLAI, S. U. 2002. *Probability, Random Variables and Stochastic Processes*. McGraw Hill, New York.

PAXSON, V. 1997. Automated packet trace analysis of TCPimplementations. *Computer Communication Review 27,* 4.

PAZ, A. 1971. *Introduction to Probabilistic Automata*. Academic Press.

PERRIN, D. AND PIN, J.-E. 2004. *Infinite Words:Automata, Semigroups, Logic and Games*. Elseiver.

PLATTNER, B. AND NIEVERGELT, J. 1981. Monitoring program execution: A survey. *IEEE Computer*, 76–93.

PNUELI, A. AND ZAKS, A. 2006. Psl model checking and run-time verification via testers. In *Proceedings of the 14th International Symposium on Formal Methods*. 573–586.

RABIN, M. O. 1963. Probabilitic automata. *Information and Control 6,* 3, 230–245.

RANUM, M. J., LANDFIELD, K., STOLARCHUK, M., SIENKIEWICZ, M., LAMBETH, A., AND WALL, E. 1997. Implementing a generalized tool for network monitoring. In *Proceedings of the Systems Administration Conference*. 1–8.

SALOMAA, A. 1973. *Formal Languages*. Academic Press.

SAMMAPUN, U., LEE, I., SOKOLSKY, O., AND REGEHR, J. 2007. Statistical runtime checking of probabilistic properties. In *Proceedings of the 7th International Workshop on Runtime Verification*. 164–175.

SCHNEIDER, F. B. 2000. Enforceable security policies. *ACM Transactions on Information Systems Security 3,* 1, 30–50.

SCHROEDER, B. A. 1995. On-line monitoring: A Tutorial. *IEEE Computer*, 72–78.

SISTLA, A. P. 1985. On characterization of safety and liveness properties in temporal logic. In *Proceedings of the ACM Symposium on Principle of Distributed Computing*.

SISTLA, A. P. AND SRINIVAS, A. R. 2008. Monitoring temporal properties of stochastic systems. In *Proceedings of International Conference on Verification, Model Checking and Abstract Interpretation*. Lecture Notes in Computer Science, vol. 4905. Springer, 294–308.

SZELEPCSENYI, R. 1987. The method of forcing for nondeterministic automata. *Bulletin of the EATCS 33*, 96–100.

THOMAS, W. 1990. Automata on infinite objects. In *Handbook of Theoretical Computer Science*. Vol. B. 133–192.

VARDI, M. 1985. Automatic verification of probabilistic concurrent systems. In *26th annual Symposium on Foundations of Computer Science*. IEEE Computer Society Press, 327–338.

VISWANATHAN, M. 2000. Foundations for the run-time analysis of software systems. Ph.D. thesis, University of Pennsylvania.