

Probabilistic Acceptors for Languages over Infinite Words

Christel Baier¹, Nathalie Bertrand², and Marcus Größer¹

¹ Technical University Dresden, Faculty Computer Science, Germany

² INRIA Rennes Bretagne Atlantique, France

Abstract. Probabilistic ω -automata are variants of nondeterministic automata for infinite words where all choices are resolved by probabilistic distributions. Acceptance of an infinite input word requires that the probability for the accepting runs is positive. In this paper, we provide a summary of the fundamental properties of probabilistic ω -automata concerning expressiveness, efficiency, compositionality and decision problems.

While classical finite automata can serve to recognize languages over finite words or tree-like structures, ω -automata are acceptors for languages consisting of infinite objects. ω -automata play a central role for verification purposes, reasoning about infinite games and logics that specify nondeterministic behaviors. Many variants of ω -automata have been studied in the literature that can be classified according to their inputs (e.g., words or trees), their acceptance conditions (e.g., Büchi, Rabin, Streett, Muller or parity acceptance) and their semantics of the branching structure (e.g., deterministic, nondeterministic, or alternating). See, e.g., [Tho97, GTW02] for an overview of automata over infinite objects.

In this paper, we study probabilistic variants of ω -automata for languages over infinite words. Although probabilistic finite automata (PFA) have attracted many researchers, see e.g. [Rab63, Paz66, Fre81, MHC03, DS90, BC03], probabilistic language acceptors for infinite words just have recently been studied. The formal definition of probabilistic ω -automata is roughly the same as for nondeterministic ω -automata, except that all choices are resolved by probabilistic distributions. Acceptance of an infinite word $\sigma = a_1 a_2 a_3 \dots$ requires that the generated sample run for σ (i.e., sequence of states that are passed in the automaton while reading σ letter by letter) meets the acceptance condition with positive probability. For instance, in the case of a probabilistic Büchi automaton (PBA), certain states are declared to be accepting and the acceptance condition requires to visit some accepting state infinitely often with positive probability.

As this definition of the accepted language via the criterion “*the probability for the accepting runs is > 0* ” appears to be the natural adaption of the definition of the accepted language of a nondeterministic automaton which relies on the criterion “*there is at least one accepting run*”, one might expect that probabilistic and nondeterministic ω -automata are rather close and enjoy similar properties. This, however, is not the case. The first surprising result is that PBA are more expressive than nondeterministic Büchi automata (NBA). Second, concerning the sizes of smallest automata for a given language, probabilistic and nondeterministic ω -automata are not comparable. That is,

there are languages that can be accepted by PBA of polynomial size, while all NBA for these languages have at least exponentially many states, and vice versa. Another interesting observation is that in the probabilistic setting the Büchi condition is somehow more powerful than in the nondeterministic case, as there exists a polynomial transformation from PBA to probabilistic automata with Streett acceptance. This is known to be impossible in the nondeterministic case [SV89].

On the other hand, the price we have to pay for this extra power of PBA is that we lose decidability of algorithmic problems, such as the emptiness, universality or equivalence problem. The undecidability results for PBA have several important consequences. First, the concept of PBA is not adequate for solving algorithmic problems that are related to the emptiness or universality problems. This, e.g., applies to the verification of nondeterministic systems against PBA-specifications. Second, PBA can be viewed as a special instance of partially-observable Markov decision processes (POMDPs) which are widely used in various areas, including robotics and stochastic planning (see, e.g., [Son71, PT87, Lov91]) and the negative results established for PBA yield the undecidability of various verification problems for POMDPs.

Organization. Section 1 recalls the definition of nondeterministic ω -automata with Büchi, Rabin or Streett acceptance conditions and introduces their probabilistic variants. Results on the expressiveness and efficiency of probabilistic Büchi, Rabin and Streett automata are summarized in Section 2. Composition operators for PBA are addressed in Section 3. Decision problems for PBA and the relation to POMDPs will be discussed in Section 4. Finally, Section 5 contains some concluding remarks.

The material of this paper is a summary of the results presented in the papers [BG05, BGG08]. Further details can be found there and in the thesis by Marcus Größer [Grö08].

1 From Nondeterministic to Probabilistic ω -Automata

We assume some familiarity with classical nondeterministic automata over finite or infinite words; see e.g. [Tho97, GTW02]. We just recall some basic concepts of nondeterministic ω -automata. Later we will adapt these concepts to the probabilistic setting.

Definition 1 (Nondeterministic ω -automata). *A nondeterministic ω -automaton is a tuple $\mathcal{N} = (Q, \Sigma, \delta, Q_0, \text{Acc})$, where*

- Q is a finite nonempty set of states,
- Σ is a finite nonempty input alphabet,
- $\delta: Q \times \Sigma \rightarrow 2^Q$ is a transition function that assigns to each state q and letter $a \in \Sigma$ a (possibly empty) set $\delta(q, a)$ of states,
- $Q_0 \subseteq Q$ is the set of initial states,
- Acc is an acceptance condition (which will be explained later).

\mathcal{N} is called *deterministic* if $|Q_0| = 1$ and $|\delta(q, a)| = 1$ for all $q \in Q$ and $a \in \Sigma$.

The intuitive operational behavior of a nondeterministic ω -automaton \mathcal{N} for some infinite input word $\sigma = a_1 a_2 a_3 \dots \in \Sigma^\omega$ is as follows. The automaton selects nondeterministically an initial state $q_0 \in Q_0$. Then, it attempts to read the first letter a_1 in

state q_0 . If q_0 does not have an outgoing a_1 -transition (i.e., $\delta(q_0, a_1) = \emptyset$) then the automaton rejects. Otherwise, the automaton reads the first letter a_1 and moves to some a_1 -successor q_1 of q_0 (i.e., some state $q_1 \in \delta(q_0, a_1)$) and attempts to read the remaining word $a_2 a_3 \dots$ from state q_1 . That is, the automaton rejects if $\delta(q_1, a_2) = \emptyset$. Otherwise the automaton reads letter a_2 and moves to some state $q_2 \in \delta(q_1, a_2)$ which is chosen nondeterministically, and so on.

Any maximal state-sequence $\pi = q_0 q_1 q_2 \dots$ that can be obtained in this way is called a *run* for σ . We write $\text{inf}(\pi)$ for the set of states $p \in Q$ such that $p = q_i$ for infinitely many indices $i \geq 0$.

Each finite run $q_0 q_1 \dots q_i$ (where \mathcal{N} fails to read letter a_{i+1} in the last state q_i because $\delta(q_i, a_{i+1})$ is empty) is said to be *rejecting*. The acceptance condition Acc imposes a condition on infinite runs and declares which of the infinite runs are *accepting*. Several acceptance conditions are known for nondeterministic ω -automata. We will consider three types of acceptance conditions:

Büchi: A Büchi acceptance condition Acc is a subset F of Q . The elements in F are called *final* or *accepting states*. An infinite run $\pi = q_0 q_1 q_2 \dots$ is called (Büchi) accepting if π visits F infinitely often, i.e., $\text{inf}(\pi) \cap F \neq \emptyset$.

Streett: A Streett acceptance condition Acc is a finite set of pairs (H_l, K_l) consisting of subsets H_l, K_l of Q , i.e., $\text{Acc} = \{(H_1, K_1), \dots, (H_\ell, K_\ell)\}$. An infinite run $\pi = q_0 q_1 q_2 \dots$ is called (Streett) accepting if for each $l \in \{1, \dots, \ell\}$ we have:

$$\text{inf}(\pi) \cap H_l \neq \emptyset \text{ or } \text{inf}(\pi) \cap K_l = \emptyset.$$

Rabin: A Rabin acceptance condition Acc is syntactically the same as a Streett acceptance condition, i.e., a finite set $\text{Acc} = \{(H_1, K_1), \dots, (H_\ell, K_\ell)\}$ where $H_l, K_l \subseteq Q$ for $1 \leq l \leq \ell$. An infinite run $\pi = q_0 q_1 q_2 \dots$ is called (Rabin) accepting if there is some $l \in \{1, \dots, \ell\}$ such that

$$\text{inf}(\pi) \cap H_l = \emptyset \text{ and } \text{inf}(\pi) \cap K_l \neq \emptyset.$$

Note that the semantics of Streett and Rabin acceptance conditions are duals of each other, i.e., for each infinite run π we have: π is accepting according to the Rabin condition Acc iff π is rejecting (i.e., not accepting) according to the Streett condition Acc . Furthermore, a Büchi acceptance condition F can be viewed as a special case of a Streett and Rabin condition with a single acceptance pair, namely $\{(F, Q)\}$ for the Streett condition and $\{(\emptyset, F)\}$ for the Rabin condition.

The *accepted language* of a nondeterministic ω -automaton \mathcal{N} with the alphabet Σ , denoted $\mathcal{L}(\mathcal{N})$, is defined as the set of infinite words $\sigma \in \Sigma^\omega$ that have at least one accepting run in \mathcal{N} .

$$\mathcal{L}(\mathcal{N}) = \{ \sigma \in \Sigma^\omega : \text{there exists an accepting run for } \sigma \text{ in } \mathcal{N} \}$$

In what follows, we write NBA to denote a nondeterministic Büchi automaton, NRA for nondeterministic Rabin automata and NSA for nondeterministic Streett automata. Similarly, the notations DBA, DRA and DSA are used to denote deterministic ω -automata with a Büchi, Rabin or Streett acceptance condition.

It is well-known that the classes of languages that can be accepted by NBA, DRA, NRA, DSA or NSA are the same. These languages are often called ω -regular and represented by ω -regular expressions, i.e., finite sums of expressions of the form $\alpha.\beta^\omega$ where α and β are ordinary regular expressions (representing regular languages over finite words) and the language associated with β is nonempty and does not contain the empty word. In the sequel, we will identify ω -regular expressions with the induced ω -regular language.

While deterministic ω -automata with Rabin and Streett acceptance (DRA and DSA) cover the full class of ω -regular languages, DBA are less powerful as, e.g., the language $(a+b)^*a^\omega$ cannot be recognized by a DBA. Hence, the class of DBA-recognizable languages is a proper subclass of the class of ω -regular languages.

Probabilistic ω -automata can be viewed as nondeterministic ω -automata where all nondeterministic choices are resolved probabilistically. That is, for any state p and letter $a \in \Sigma$ either p does not have any a -successor or there is a probability distribution for the a -successors of p .

Definition 2 (Probabilistic ω -automata). *A probabilistic ω -automaton is a tuple $\mathcal{P} = (Q, \Sigma, \delta, \mu_0, \text{Acc})$, where*

- Q is a finite nonempty set of states,
- Σ is a finite nonempty input alphabet,
- $\delta: Q \times \Sigma \times Q \rightarrow [0, 1]$ is a transition probability function such that for all $p \in Q$ and $a \in \Sigma$ either $\sum_{q \in Q} \delta(p, a, q) = 1$ or $\delta(p, a, \cdot)$ is the null-function (i.e. $\delta(p, a, q) = 0$ for all $q \in Q$),
- μ_0 is the initial distribution, i.e., a function $\mu_0: Q \rightarrow [0, 1]$ such that $\sum_{q \in Q} \mu_0(q) = 1$,
- Acc is an acceptance condition (as for nondeterministic ω -automata).

We refer to the states $q_0 \in Q$ where $\mu_0(q_0) > 0$ as initial states. If p is a state such that $\delta(q, a, p) > 0$ then we say that q has an outgoing a -transition to state p .

Acceptance conditions can be defined as in the nondeterministic case. In this paper, we just regard Büchi, Rabin and Streett acceptance and use the abbreviations PBA, PRA and PSA for probabilistic Büchi automata, probabilistic Rabin automata, and probabilistic Streett automata, respectively.

The intuitive operational behavior of a probabilistic ω -automaton \mathcal{P} for a given input word $\sigma = a_1 a_2 \dots \in \Sigma^\omega$ is similar to the nondeterministic setting, except that the nondeterminism is resolved internally by the probabilistic distributions μ_0 in the initial configuration and $\delta(q, a, \cdot)$ if the current state is q and the next letter to consume is a . That is, initially \mathcal{P} chooses at random an initial state p_0 according to the initial distribution μ_0 . If \mathcal{P} has consumed the first i input symbols a_1, \dots, a_i and the current state is p_i then \mathcal{P} moves with probability $\delta(p_i, a_{i+1}, p)$ to state p and tries to read the next input symbol a_{i+2} in state $p = p_{i+1}$. If there is no outgoing a_{i+1} -transition from the current state p_i , then \mathcal{P} rejects.

As in the nondeterministic case, the resulting state-sequence $\pi = p_0 p_1 p_2 \dots \in Q^* \cup Q^\omega$ is called a *run* for σ in \mathcal{P} . If \mathcal{P} rejects in state p_i , i.e., $\delta(p_i, a_{i+1}, \cdot)$ is the null function, then the obtained run is finite (and ends in state p_i). If the automaton never rejects while reading the letters a_i of the input word $\sigma = a_1 a_2 a_3 \dots \in \Sigma^\omega$, the generated

run is an infinite state-sequence $\pi = p_0 p_1 p_2 \dots \in Q^\omega$. Acceptance of a run according to a Büchi, Rabin or Streett acceptance condition is defined as in the nondeterministic setting.

Semantics of probabilistic ω -automata. While acceptance of an infinite word in a non-deterministic ω -automata requires the existence of an accepting run, a probabilistic ω -automaton accepts an infinite input word σ if the *acceptance probability* $\Pr^{\mathcal{P}}(\sigma)$ is > 0 . The formal definition of the acceptance probability relies on the view of an input word $\sigma \in \Sigma^\omega$ as a *scheduler* when \mathcal{P} is treated as a Markov decision process, i.e., an operational model for a probabilistic system where in each state q the letters that can be consumed in q are treated as actions that are enabled in q . Given a word/scheduler $\sigma = a_1 a_2 a_3 \dots \in \Sigma^\omega$, the behavior of \mathcal{P} under σ is given by a Markov chain \mathcal{M}_σ where the states are pairs (q, i) where $q \in Q$ stands for the current state and i is a natural number ≥ 1 that denotes the current word position. Stated differently, state (q, i) in the Markov chain \mathcal{M}_σ stands for the configuration that \mathcal{P} might have reached after having consumed the first $i-1$ letters a_1, \dots, a_{i-1} of the input word σ . Assuming that $\delta(q, a_{i+1}, \cdot)$ is not the null function, the transition probabilities from state (q, i) are given by the distribution $\delta(q, a_{i+1}, \cdot)$, i.e., from state (q, i) the Markov chain \mathcal{M}_σ moves with probability $\delta(q, a_{i+1}, p)$ to state $(p, i+1)$. In case that $\delta(q, a_{i+1}, \cdot) = 0$ then (q, i) is an absorbing state, i.e., a state without any outgoing transition. The runs for σ in \mathcal{P} correspond to the paths in \mathcal{M}_σ . We can now apply the standard concepts for Markov chains to reason about the probabilities of infinite paths and define the acceptance probability for the infinite word σ in \mathcal{P} , denoted $\Pr^{\mathcal{P}}(\sigma)$ or briefly $\Pr(\sigma)$, as the probability measure of the accepting runs for σ in the Markov chain \mathcal{M}_σ .

The formal definition of the accepted language of \mathcal{P} is

$$\mathcal{L}(\mathcal{P}) = \{ \sigma \in \Sigma^\omega : \Pr^{\mathcal{P}}(\sigma) > 0 \}$$

Sometimes we will add the subscript ‘‘Büchi’’, ‘‘Streett’’ or ‘‘Rabin’’ to make clear which type of acceptance condition is assumed and write $\mathcal{L}_{\text{Büchi}}(\mathcal{P})$, $\mathcal{L}_{\text{Rabin}}(\mathcal{P})$ or $\mathcal{L}_{\text{Streett}}(\mathcal{P})$, respectively.

Example 1 (Probabilistic Büchi automata). Let us have a look at a few examples of probabilistic ω -automata. In the pictures, if $\delta(q, a, \cdot) \neq 0$ then the probability $\delta(q, a, p)$ is attached to the a -transition from q to p . If $\delta(q, a, p) = 1$ then the edge is simply labeled with a . Similarly, if there is a single initial state q_0 (i.e., $\mu_0(q_0) = 1$, while $\mu_0(q) = 0$ for all other states q) we simply depict an incoming arrow to q_0 . For PBA, we depict the accepting states (i.e., the states $q \in F$) by squares, non-accepting states by circles.

Consider the PBA \mathcal{P} over the alphabet $\Sigma = \{a, b\}$ in the left part of Figure 1. State q_0 is initial, while state q_1 is accepting. More precisely, the initial distribution is given by $\mu_0(q_0) = 1$ and $\mu_0(q_1) = 0$, while the Büchi acceptance condition is given by the singleton $F = \{q_1\}$.

The accepted language $\mathcal{L}(\mathcal{P}) = \mathcal{L}_{\text{Büchi}}(\mathcal{P})$ is $(a+b)^*a^\omega$. If we feed \mathcal{P} with an infinite input word $\sigma \in (a+b)^*a^\omega$, then \mathcal{P} stays with positive probability in the initial state q_0 until the last b in σ has been read. From then on, \mathcal{P} moves almost surely to the accepting state q_1 and stays there forever when reading the infinite suffix a^ω . Thus, the acceptance probability for all words in $(a+b)^*a^\omega$ is positive. This yields that $(a+b)^*a^\omega \subseteq \mathcal{L}(\mathcal{P})$.

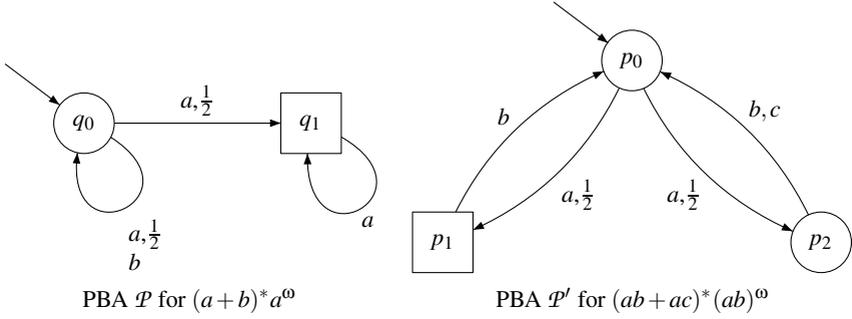


Fig. 1. Examples for PBA \mathcal{P} (left) and \mathcal{P}' (right)

It remains to show that $\mathcal{L}(\mathcal{P})$ is contained in $(a+b)^*a^\omega$. In fact, all words in $\mathcal{L}(\mathcal{P})$ contain only finitely many b 's because only state q_1 is accepting and only letter a can be read in state q_1 . Hence, each accepted word $\sigma \in \mathcal{L}(\mathcal{P})$ must have the suffix a^ω .

For another example, regard the PBA \mathcal{P}' over the alphabet $\Sigma = \{a, b, c\}$ shown in the right part of Figure 1. Let us first observe that the underlying nondeterministic Büchi automaton (NBA) that we obtain by ignoring the probabilities has an accepting run for each infinite word in $(ab+ac)^\omega$ with infinitely many b 's, no matter whether there are only finitely c 's or infinitely many c 's. Thus, the accepted language of the NBA is $((ac)^*ab)^\omega$. This language is different from the accepted language of the PBA \mathcal{P}' which is

$$\mathcal{L}(\mathcal{P}') = (ab+ac)^*(ab)^\omega.$$

Given an input word $\sigma \in (ab+ac)^*(ab)^\omega$, say $\sigma = x(ab)^\omega$ where $x \in (ab+ac)^*$, then with positive probability \mathcal{P} generates the run fragment $p_0 p_2 p_0 p_2 \dots p_0 p_2 p_0$ when reading x . For the remaining suffix $(ab)^\omega$, \mathcal{P} can always consume the next letter and almost surely \mathcal{P} will visit p_1 and p_2 infinitely often. This yields $\Pr^{\mathcal{P}'}(\sigma) > 0$ and $\sigma \in \mathcal{L}(\mathcal{P}')$. Vice versa, we have to show that $\mathcal{L}(\mathcal{P}')$ is a subset of $(ab+ac)^*(ab)^\omega$. It is obvious that all accepted words $\sigma \in \mathcal{L}(\mathcal{P}')$ belong to $((ac)^*ab)^\omega$.³ The intuitive argument why any word σ in $(ab+ac)^\omega$ with infinitely many c 's is rejected by \mathcal{P}' relies on the observation that almost all runs for σ are finite and end in state p_1 , where the next input symbol is c and cannot be consumed in state p_1 . (The formulation “almost all runs have property X ” means that the probability measure of the runs where property X does not hold is 0.) \square

Note that each DBA can be viewed as a PBA (we just have to assign probability 1 to all edges in the DBA and deal with the initial distribution that assigns probability 1 to the unique initial state). As shown in Example 1 there is a PBA for the language $(a+b)^*a^\omega$ which cannot be accepted by DBA. As a consequence we obtain that the class of DBA-recognizable languages is a proper subclass of the class of languages $\mathcal{L}(\mathcal{P})$ for some PBA \mathcal{P} .

³ In fact, for each PBA \mathcal{P} all words in $\mathcal{L}(\mathcal{P})$ have at least one accepting run. Thus, $\mathcal{L}(\mathcal{P})$ is always contained in the accepted language of the NBA that results by ignoring the transition probabilities.

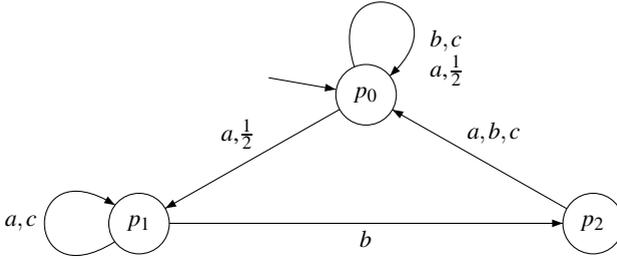


Fig. 2. PSA \mathcal{P}_S with acceptance condition $\text{Acc} = \{\{p_2\}, \{p_1\}\}$

Example 2 (Probabilistic Streett automaton). The PSA \mathcal{P}_S shown in Figure 2 accepts the language of all infinite words σ over $\Sigma = \{a, b, c\}$ such that either σ contains only finitely many a 's or σ contains infinitely many b 's (or both). That is,

$$\mathcal{L}_{\text{Streett}}(\mathcal{P}_S) = (a + b + c)^*(b + c)^\omega + ((a + b + c)^*b)^\omega.$$

Note that each word in $(a + b + c)^*(b + c)^\omega$ has an accepting run π which never leaves state p_0 . This run $\pi = p_0^\omega$ is accepting (as $\text{inf}(\pi) = \{p_0\}$ and therefore $\text{inf}(\pi) \cap \{p_1\} = \emptyset$) and has positive measure. More precisely, the acceptance probability $\text{Pr}^{\mathcal{P}_S}(\sigma)$ for a word $\sigma = xy$ with $x \in \{a, b, c\}^*$ and $y \in \{b, c\}^\omega$ is $\geq 1/2^k$ where k is the number of a 's in x . Hence, $\sigma = xy \in \mathcal{L}_{\text{Streett}}(\mathcal{P}_S)$. For the infinite words σ with infinitely many b 's all runs are accepting, since they either visit p_1, p_2 only finitely many times (in which case $\text{inf}(\pi) \cap \{p_1\} = \emptyset$) or they visit p_2 infinitely often (as σ contains infinitely many b 's). Hence, $\text{Pr}^{\mathcal{P}_S}(\sigma) = 1$ which yields $\sigma \in \mathcal{L}_{\text{Streett}}(\mathcal{P}_S)$.

Vice versa, for each word in $\mathcal{L}_{\text{Streett}}(\mathcal{P}_S)$ which contains infinitely many a 's, almost all runs will enter state p_1 infinitely often. But then an accepting run must also visit state p_2 infinitely often, which is only possible by reading letter b . Thus, all words accepted by \mathcal{P}_S either contain only finitely many a 's or infinitely many b 's. \square

Equivalence of ω -automata means that their accepted languages agree. The notion of the *size*, denoted $|\mathcal{P}|$, of an ω -automaton \mathcal{P} is used here as follows. The size of a PBA is simply the number of states. The size of a probabilistic Rabin or Streett automaton denotes the number of states plus the number of acceptance pairs.

2 Expressiveness of Probabilistic ω -Automata

The three types of probabilistic ω -automata (Büchi, Rabin, Streett) are equally expressive. As the Büchi acceptance condition can be rewritten as a Rabin or Streett acceptance condition, each PBA can be viewed as a PRA or as a PSA with the same accepted language. Vice versa, there are polynomial transformations from PRA and PSA to PBA:

Theorem 1 (Equivalence of PBA, PRA and PSA [BG05])

- (a) For each PBA there is an equivalent PRA and an equivalent PSA of the same size.
- (b) Given a PRA \mathcal{P}_R with ℓ acceptance pairs there exists an equivalent PBA of size $O(\ell|\mathcal{P}_R|)$.

(c) Given a PSA \mathcal{P}_S with ℓ acceptance pairs there exists an equivalent PBA of size $O(\ell^2|\mathcal{P}_S|)$.

The transformation from PRA to PBA is roughly the same as in the nondeterministic case. The construction of a PBA of size $O(\ell^2|\mathcal{P}_S|)$ from a given PSA \mathcal{P}_S , however, crucially relies on the probabilistic semantics. In fact, it is worth noting that in the non-probabilistic case it is known (see [SV89]) that there are families $(L_n)_{n \geq 0}$ of languages $L_n \subseteq \Sigma^\omega$ that are recognizable by nondeterministic Streett automata of size $O(n)$, while each nondeterministic Büchi automaton for L_n has 2^n or more states. Thus, the polynomial transformation from Streett to Büchi acceptance is specific for the probabilistic case.

Let us now discuss the expressiveness of probabilistic ω -automata compared to their nondeterministic counterparts. The first observation is that each NBA can be transformed into an equivalent PBA. Hence, the class of languages that is recognizable by a PBA subsumes the class of ω -regular languages. A transformation from NBA into an equivalent PBA is obtained by using NBA that are *deterministic-in-limit*. These are NBA such that for each state p that is reachable from some accepting state $q \in F$ and each letter $a \in \Sigma$ state p has at most one outgoing a -transition. That is, as soon as an accepting state has been reached the behavior from then on is deterministic. [CY95] presented some kind of powerset construction which turns a given NBA \mathcal{N} into an equivalent NBA \mathcal{N}_{det} that is deterministic-in-limit. If we now resolve the nondeterministic choices in \mathcal{N}_{det} by uniform distributions⁴ then \mathcal{N}_{det} becomes a PBA that accepts the same language as \mathcal{N} (and \mathcal{N}_{det}). This yields:

Lemma 1 (see [BG05]). *For each ω -regular language $L \subseteq \Sigma^\omega$ there exists a PBA \mathcal{P} with the alphabet Σ such that $\mathcal{L}(\mathcal{P}) = L$.*

The powerset construction used in the transformation from \mathcal{N} to \mathcal{N}_{det} can cause an exponential blow-up. In fact, the worst-case exponential blow-up cannot be avoided for the transformation from NBA to PBA since there are examples for families $(L_n)_{n \geq 1}$ of ω -regular languages that are accepted by NBA of linear size, while each PBA for L_n has $\Omega(2^n)$ states [BG05]. An example for such a family of languages is $((a+b)^*a(a+b)^n c)^\omega$.

Vice versa, there are languages L_n that can be recognized by PBA of size $O(n)$, while any NSA for L_n has $\Omega(2^n/n)$ states [BG05].

We now address the question whether each PBA can be transformed into an equivalent NBA. Surprisingly, this is not the case, as there are PBA where the accepted language is not ω -regular. An example is given in Figure 3 which shows a PBA \mathcal{P}_λ where λ is an arbitrary real number in the open interval $]0, 1[$. The PBA \mathcal{P}_λ accepts the language

$$\mathcal{L}(\mathcal{P}_\lambda) = \left\{ a^{k_1} b a^{k_2} b a^{k_3} b \dots : \prod_{i=1}^{\infty} (1 - \lambda^{k_i}) > 0 \right\}$$

⁴ If q is a state in \mathcal{N}_{det} and $a \in \Sigma$ such that q has k a -successors q_1, \dots, q_k then we define $\delta(q, a, q_i) = \frac{1}{k}$ for $1 \leq i \leq k$ and $\delta(q, a, p) = 0$ for all states $p \notin \{q_1, \dots, q_k\}$. Similarly, if Q_0 is the set of initial states in \mathcal{N}_{det} and Q_0 is nonempty then we deal with the initial distribution μ_0 that assigns probability $1/|Q_0|$ to each state in Q_0 .

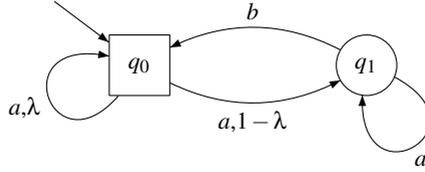


Fig. 3. PBA \mathcal{P}_λ accepts a non- ω -regular language

To see this, let us first observe that all words in $\mathcal{L}(\mathcal{P}_\lambda)$ must contain infinitely many b 's. As \mathcal{P}_λ cannot consume two consecutive b 's, all words in $\mathcal{L}(\mathcal{P}_\lambda)$ have the form $a^{k_1}ba^{k_2}ba^{k_3}b\dots$ where k_1, k_2, \dots is a sequence of positive natural numbers. We now show that

$$\Pr^{\mathcal{P}_\lambda}(a^{k_1}ba^{k_2}ba^{k_3}b\dots) = \prod_{i=1}^{\infty} (1 - \lambda^{k_i})$$

The factors $1 - \lambda^{k_i}$ stand for the probability to move from state q_0 to q_1 when reading the subword a^{k_i} . With the remaining probability λ^{k_i} , the automaton \mathcal{P}_λ stays in state q_0 , but then letter b at position $k_1 + \dots + k_i + i$ of the input word $a^{k_1}ba^{k_2}ba^{k_3}b\dots$ cannot be consumed and \mathcal{P}_λ rejects. Hence, the probability for run fragments of the form $q_0 \dots q_0 q_1 \dots q_1 q_0$ that are generated while reading the subword $a^{k_i}b$ is precisely $1 - \lambda^{k_i}$. This yields that the infinite product of these values agrees with the acceptance probability for the input word $a^{k_1}ba^{k_2}ba^{k_3}b\dots$.

The convergence condition which requires the infinite product over the values $1 - \lambda^{k_i}$ to be positive can easily be shown to be non- ω -regular, i.e., $\mathcal{L}(\mathcal{P}_\lambda)$ cannot be recognized by an NBA. Together with Lemma 1 we get:

Theorem 2 (Expressiveness of PBA, [BG05]). *The class of languages that are accepted by a PBA strictly subsumes the class of ω -regular languages.*

This stands in contrast to the fact that probabilistic finite automata (PFA) with the acceptance criterion “the accepting runs have a positive probability measure” can be viewed as nondeterministic finite automata, and hence, have exactly the power of regular languages.

The PBA \mathcal{P}_λ can also serve to illustrate another interesting property of PBA. Consider two values λ and $\nu \in]0, 1[$ with $\lambda < \nu$. For any sequence $(k_i)_{i \geq 1}$ of natural numbers k_i where the infinite product over the values $1 - \nu^{k_i}$ converges to some positive value, also the infinite product over the values $1 - \lambda^{k_i}$ is positive, as we have $1 - \nu^{k_i} < 1 - \lambda^{k_i}$. Thus, $\mathcal{L}(\mathcal{P}_\nu) \subseteq \mathcal{L}(\mathcal{P}_\lambda)$. In fact, whenever $\lambda < \nu$ then there are sequences $(k_i)_{i \geq 1}$ such that the product of the values $1 - \lambda^{k_i}$ converges to some positive real number, while the product of the values $1 - \nu^{k_i}$ has value 0 [BBG08]. Hence:

Lemma 2 (see [BBG08]). *If $\lambda < \nu$ then $\mathcal{L}(\mathcal{P}_\nu)$ is a proper sublanguge of $\mathcal{L}(\mathcal{P}_\lambda)$.*

Thus, the languages of PBA are sensitive to the distributions for the successor states. That is, if we are given a PBA and modify the nonzero transition probabilities then also the accepted language might change. This property is surprising since the definition of

the accepted language just relies on a qualitative criterion: the acceptance probability must be positive, but might be arbitrarily small. This should be opposed to the verification of finite-state Markov decision processes where it is known that whether or not a given linear time property holds with positive probability just depends on the underlying graph, but not on the concrete transition probabilities.

3 Composition Operators for PBA

The most important composition operators for any class of languages over infinite words are the standard set operations union, intersection and complementation. In fact, the class of PBA-recognizable languages is closed under all three operations.

Theorem 3 (see [BBG08]). *The class of languages $\mathcal{L}(\mathcal{P})$ for some PBA \mathcal{P} is closed under union, intersection and complementation.*

For *union*, this is obvious since given two PBA \mathcal{P}_1 and \mathcal{P}_2 over the same alphabet with initial distributions μ_1 and μ_2 , respectively, we consider the PBA that arises from the disjoint union of \mathcal{P}_1 and \mathcal{P}_2 with the initial distribution $\mu(q) = \frac{1}{2}\mu_i(q)$ if q is a state in \mathcal{P}_i . If F_1 and F_2 are the sets of accepting states in \mathcal{P}_1 and \mathcal{P}_2 , respectively, then \mathcal{P} requires to visit $F_1 \cup F_2$ infinitely often.

For *intersection*, we can reuse the ideas of an intersection operator on NBA. Given two PBA \mathcal{P}_1 and \mathcal{P}_2 over the same alphabet, we use a product construction $\mathcal{P}_1 \times \mathcal{P}_2$ (which runs \mathcal{P}_1 and \mathcal{P}_2 in parallel) and equip $\mathcal{P}_1 \times \mathcal{P}_2$ with the Streett condition consisting of two acceptance pairs. One of the acceptance pairs requires that an accepting state of \mathcal{P}_1 is visited infinitely often, the other one stands for the acceptance condition of \mathcal{P}_2 . This PSA $\mathcal{P}_1 \times \mathcal{P}_2$ can then be transformed into an equivalent PBA (part (c) of Theorem 1).

The most interesting operator is *complementation*. Given a PBA \mathcal{P} with $L = \mathcal{L}(\mathcal{P}) \subseteq \Sigma^\omega$, the idea for the construction of a PBA $\overline{\mathcal{P}}$ for the language $\overline{L} = \Sigma^\omega \setminus L$ is somehow similar to the complementation of NBA via Safra's determinisation operator [Saf88] and relies on the transformations sketched in Figure 4. Here, by a 0/1-PRA we mean a PRA \mathcal{P}_R where the acceptance probabilities for all words are 0 or 1, i.e., $\Pr^{\mathcal{P}_R}(\sigma) \in \{0, 1\}$ for each word $\sigma \in \Sigma^\omega$. Similarly, a 0/1-PSA is a PSA \mathcal{P}_S with $\Pr^{\mathcal{P}_S}(\sigma) \in \{0, 1\}$ for each word $\sigma \in \Sigma^\omega$. Given a 0/1-PRA \mathcal{P}_R , the duality of Rabin and Streett acceptance yields that $\mathcal{L}_{\text{Rabin}}(\mathcal{P}_R)$ and $\mathcal{L}_{\text{Streett}}(\mathcal{P}_R)$ are complements of each other. Thus, in the second step of Figure 4 we may deal with the PSA \mathcal{P}_S which is syntactically the same as \mathcal{P}_R (but \mathcal{P}_S is viewed as a Streett and \mathcal{P}_R as a Rabin automaton). The last step of Figure 4 relies on the transformation from PSA to PBA (part (c) of Theorem 1).

The interesting step is the first one in Figure 4 where we have to construct a 0/1-PRA from a given PBA. The idea for this transformation is to design a 0/1-PRA \mathcal{P}_R that

$$\text{PBA } \mathcal{P} \text{ for } L \rightsquigarrow \text{0/1-PRA } \mathcal{P}_R \text{ for } L \rightsquigarrow \text{0/1-PSA } \mathcal{P}_S \text{ for } \overline{L} \rightsquigarrow \text{PBA } \overline{\mathcal{P}} \text{ for } \overline{L}$$

Fig. 4. Complementation of a PBA

generates up to n sample runs of \mathcal{P} and checks whether at least one of them is accepting, where n is the number of states in \mathcal{P} . If so then \mathcal{P}_R accepts, otherwise it rejects. For the details of this construction we refer to [BBG08, Grö08].

Lemma 3 (From PBA to 0/1-PRA, [BBG08]). *For each PBA there exists an equivalent 0/1-PRA.*

4 Decision Problems for PBA

For many applications of automata-like models, it is important to have (efficient) decision algorithms for some fundamental problems, like checking emptiness or language inclusion. For instance, the automata-based approach [VW86] for verifying ω -regular properties of a nondeterministic finite-state system relies on a reduction to the emptiness problem for NBA. Unfortunately, the emptiness problem and various other classical decision problems for automata cannot be solved algorithmically for PBA:

Theorem 4 (Undecidability of PBA [BBG08]). *The following problems are undecidable:*

- **emptiness:** given a PBA \mathcal{P} , does $\mathcal{L}(\mathcal{P}) = \emptyset$ hold?
- **universality:** given a PBA \mathcal{P} with the alphabet Σ , does $\mathcal{L}(\mathcal{P}) = \Sigma^\omega$ hold?
- **equivalence:** given two PBA \mathcal{P}_1 and \mathcal{P}_2 , does $\mathcal{L}(\mathcal{P}_1) = \mathcal{L}(\mathcal{P}_2)$ hold?

To prove undecidability of the emptiness problem, we provided in [BBG08] a reduction from a variant of the emptiness problem for probabilistic finite automata (PFA) which has been shown to be undecidable [MHC03]. Undecidability of the universality problem then follows by the effectiveness of complementation for PBA. Undecidability of the PBA-equivalence problem is an immediate consequence of the undecidability of the emptiness problem (just consider $\mathcal{P}_1 = \mathcal{P}$ and \mathcal{P}_2 a PBA for the empty language).

A consequence of Theorem 4 is that PBA are not appropriate for verification algorithms. Consider, e.g., a finite-state transition system \mathcal{T} and suppose that a linear-time property p to be verified for \mathcal{T} is specified by a PBA \mathcal{P} in the sense that $\mathcal{L}(\mathcal{P})$ represents all infinite behaviors where property p holds. (Typically p is a language over some alphabet $\Sigma = 2^{\text{AP}}$ where AP is a set of atomic propositions and the states in \mathcal{T} are labeled with subsets of AP.) Then, the question whether all traces of \mathcal{T} have property p is reducible to the universality problem for PBA and therefore undecidable. Similarly, the question whether \mathcal{T} has at least one trace where p holds is reducible to the emptiness problem for PBA and therefore undecidable too.

Another important consequence of Theorem 4 is that it yields the undecidability of the verification problem for partially observable Markov decision processes (POMDPs) against ω -regular properties. POMDPs provide an operational model for stochastic systems with non-observable behaviors. They play a central role in many application areas such as mobile robot navigation, probabilistic planning task, elevator control, and so on. See, e.g., [Son71, Mon82, PT87, Lov91]. The syntax of a POMDP can be defined as for probabilistic ω -automata, except that the acceptance condition has to be replaced with an equivalence relation \sim on the states which formalizes which states cannot be

distinguished from outside. The elements in the alphabet Σ are viewed as action names. The goal is then to design a *scheduler* S that chooses the actions for the current state and ensures that a certain condition X holds when the choices between different enabled actions in the POMDP \mathcal{M} are resolved by S . For his choice the scheduler may use the sequence of equivalence classes that have been passed to reach the equivalence class of the current state. That is, the scheduler is supposed to observe the equivalence classes, but not the specific states. (Such schedulers are sometimes called “partial-information schedulers” or “observation-based schedulers”.)

The emptiness problem for PBA is a special instance for the scheduler-synthesis problem for POMDPs. Given a PBA $\mathcal{P} = (Q, \Sigma, \delta, \mu_0, F)$, we regard the POMDP $\mathcal{M} = (Q, \Sigma, \delta, \mu_0, \sim)$ where \sim identifies all states and ask for the existence of a scheduler that ensures that F will be visited infinitely often with positive probability. We first observe that the infinite words over Σ can be viewed as schedulers for \mathcal{M} , and vice versa. Hence, $\mathcal{L}(\mathcal{P})$ is nonempty if and only if there is a scheduler S such that $\Pr_S^{\mathcal{M}}(\Box\Diamond F) > 0$, where $\Pr_S^{\mathcal{M}}(\Box\Diamond F)$ denotes the probability that \mathcal{M} visits F infinitely often when S is used to schedule the actions in \mathcal{M} . Similarly, the universality problem for PBA can be viewed as a special instance of the problem where we are given a POMDP \mathcal{M} and a set F of states and ask for the existence of a scheduler S such that $\Pr_S^{\mathcal{M}}(\Diamond\Box F) = 1$ where $\Pr_S^{\mathcal{M}}(\Diamond\Box F)$ denotes the probability that \mathcal{M} under scheduler S eventually enters F and never leaves F from this moment on. Thus:

Theorem 5 (Undecidability results for POMDPs, [BBG08]). *The following problems are undecidable:*

- given a POMDP \mathcal{M} and a set F of states, decide whether $\exists S. \Pr_S^{\mathcal{M}}(\Box\Diamond F) > 0$
- given a POMDP \mathcal{M} and a set F of states, decide whether $\exists S. \Pr_S^{\mathcal{M}}(\Diamond\Box F) = 1$

The result of Theorem 5 is remarkable since the corresponding questions for fully observable Markov decision processes (i.e., POMDPs where the \sim -equivalence classes are singletons) are decidable in polynomial time. However, there some other instances of the verification problem for POMDPs which are decidable. This includes the following questions (for further details see [dA99, BBG08, Grö08]):

$$\begin{aligned} \exists S. \Pr_S^{\mathcal{M}}(\Box F) &> 0 \\ \exists S. \Pr_S^{\mathcal{M}}(\Diamond F) &> 0 \\ \exists S. \Pr_S^{\mathcal{M}}(\Box F) &= 1 \\ \exists S. \Pr_S^{\mathcal{M}}(\Diamond F) &= 1 \\ \exists S. \Pr_S^{\mathcal{M}}(\Box\Diamond F) &= 1 \\ \exists S. \Pr_S^{\mathcal{M}}(\Diamond\Box F) &> 0 \end{aligned}$$

From the decidability of the question whether a given POMDP \mathcal{M} and set F of states in \mathcal{M} has a scheduler S such that $\Pr_S^{\mathcal{M}}(\Box\Diamond F) = 1$ we can derive the decidability of the emptiness problem for PBA under an alternative semantics that requires that almost all runs are accepting. Formally, this *almost-sure semantics* for PBA assigns the language

$$\mathcal{L}^1(\mathcal{P}) = \{ \sigma \in \Sigma^\omega : \Pr^{\mathcal{P}}(\sigma) = 1 \}$$

to each PBA \mathcal{P} . Thus, the switch from the standard semantics $\mathcal{L}(\mathcal{P})$ to the almost-sure semantics simplifies algorithmic problems, but the almost-sure semantics has several other disadvantages. PBA with the almost-sure semantics are less expressive. They even do not cover the full class of ω -regular languages. For instance, the ω -regular language $(a + b)^*a^\omega$ cannot be recognized by a PBA with the almost-sure semantics. Since the complement $(a^*b)^\omega$ of this language is recognizable by a deterministic Büchi automaton (and therefore also by a PBA with the almost-sure semantics), PBA with the almost-sure semantics are not closed under complementation. Furthermore, there are PBA where the almost-sure semantics yields a non- ω -regular language. An example is the complement of the language $\mathcal{L}(\mathcal{P}_\lambda)$ which is recognizable by a PBA with the almost-sure semantics, while $\mathcal{L}(\mathcal{P}_\lambda)$ is not.

We finally mention that the differences between the standard semantics $\mathcal{L}(\mathcal{P})$ and the almost-sure semantics $\mathcal{L}^=1(\mathcal{P})$ for PBA do not hold when switching to Rabin or Streett acceptance. Recall that by the results established in Lemma 3 and Theorem 1, PRA with the almost-sure semantics are as expressive as ordinary PRA (and PBA), and checking emptiness, universality or equivalence of PRA with the almost-sure semantics are undecidable.

5 Conclusion

We gave a summary of the fundamental properties of probabilistic acceptors for infinite words formalized by probabilistic ω -automata with Büchi, Rabin or Streett acceptance conditions. The results show some major differences to nondeterministic (or alternating) ω -automata concerning the expressiveness, efficiency and decidability, which makes PBA interesting at least from a theoretical point of view.

So far, just the basic principles of probabilistic ω -automata have been studied and many interesting problems still have to be addressed, such as characterizations of the class of PBA-recognizable languages (e.g., by some temporal or modal logic or some extension of monadic second-order logic or by an extension of ω -regular expressions), or alternative semantics for PBA where checking emptiness is decidable and the full class of ω -regular languages is covered.

The undecidability of the emptiness problem and related problems shows that PBA with the standard semantics are not adequate for algorithmic purposes, e.g., the verification of systems with nondeterministic behaviors. The situation changes if the system to be verified is purely probabilistic (i.e., modelled by a Markov chain). In this case some decidability results for the verification problem against PBA-specifications can be established [BG05]. Being a special instance of POMDPs all negative results for PBA (undecidability) carry over from PBA to POMDP. Vice versa, for many algorithmic problems for POMDPs, algorithmic solutions for PBA can be combined with standard algorithms for (fully observable) Markov decision processes to obtain an algorithm that solves the analogous problem for POMDPs.

Another application of probabilistic ω -automata is run-time verification where special types of PBA can serve as probabilistic monitors [CSV08]. Given the wide range of application areas of probabilistic finite automata, there might be various other applications of probabilistic ω -automata. For instance, the concept of probabilistic

ω -automata is also related to partial-information games with ω -regular winning objectives [CDHR06] or could serve as starting point for studying quantum automata over infinite inputs, in the same way as PFA yield the basis for the definition of quantum finite automata [KW97, AF98].

For these reasons, we argue that the concept of probabilistic ω -automata is an interesting new research field with plenty of open questions that might lead to interesting applications.

References

- [AF98] Ambainis, A., Freivalds, R.: 1-way quantum finite automata: strengths, weaknesses and generalizations. In: Proc. of the 39th Symposium on Foundations of Computer Science (FOCS 1998). IEEE Computer Society Press, Los Alamitos (1998)
- [BBG08] Baier, C., Bertrand, N., Größer, M.: On decision problems for probabilistic Büchi automata. In: Amadio, R. (ed.) FOSSACS 2008. LNCS, vol. 4962, pp. 287–301. Springer, Heidelberg (2008)
- [BC03] Blondel, V., Canterini, V.: Undecidable problems for probabilistic finite automata. *Theory of Computer Systems* 36, 231–245 (2003)
- [BG05] Baier, C., Größer, M.: Recognizing ω -regular languages with probabilistic automata. In: Proc. of the 20th IEEE Symposium on Logic in Computer Science (LICS 2005), pp. 137–146. IEEE Computer Society Press, Los Alamitos (2005)
- [CDHR06] Chatterjee, K., Doyen, L., Henzinger, T.A., Raskin, J.-F.: Algorithms for ω -regular games with imperfect information. In: Ésik, Z. (ed.) CSL 2006. LNCS, vol. 4207, pp. 287–302. Springer, Heidelberg (2006)
- [CSV08] Chadha, R., Sistla, A.P., Viswanathan, M.: On the expressiveness and complexity of randomization in finite state monitors. In: Proc. of the 23rd IEEE Symposium on Logic in Computer Science (LICS 2008), pp. 18–29. IEEE Computer Society Press, Los Alamitos (2008)
- [CY95] Courcoubetis, C., Yannakakis, M.: The complexity of probabilistic verification. *Journal of the ACM* 42(4), 857–907 (1995)
- [dA99] de Alfaro, L.: The verification of probabilistic systems under memoryless partial-information policies is hard. In: Proc. of the 2nd International Workshop on Probabilistic Methods in Verification (ProbMiV 1999), vol. 9, pp. 19–32. Birmingham University, Research Report CSR-99-9 (1999)
- [DS90] Dwork, C., Stockmeyer, L.: A time-complexity gap for two-way probabilistic finite state automata. *SIAM Journal of Computing* 19, 1011–1023 (1990)
- [Fre81] Freivalds, R.: Probabilistic two-way machines. In: Gruska, J., Chytil, M.P. (eds.) MFCS 1981. LNCS, vol. 118, pp. 33–45. Springer, Heidelberg (1981)
- [Grö08] Größer, M.: Reduction Methods for Probabilistic Model Checking. PhD thesis, Technical University Dresden, Faculty for Computer Science (2008)
- [GTW02] Grädel, E., Thomas, W., Wilke, T. (eds.): Automata, Logics, and Infinite Games. LNCS, vol. 2500. Springer, Heidelberg (2002)
- [KW97] Kondacs, A., Watrous, J.: On the power of quantum finite state automata. In: Proc. of the 38th Symposium on Foundations of Computer Science (FOCS 1997), pp. 66–75. IEEE Computer Society Press, Los Alamitos (1997)
- [Lov91] Lovejoy, W.: A survey of algorithmic methods for partially observable Markov decision processes. *Annals of Operations Research* 28(1), 47–65 (1991)

- [MHC03] Madani, O., Hanks, S., Condon, A.: On the undecidability of probabilistic planning and related stochastic optimization problems. *Artificial Intelligence* 147(1-2), 5–34 (2003)
- [Mon82] Monahan, G.: A survey of partially observable Markov decision processes: Theory, models, and algorithms. *Management Science* 28(1), 1–16 (1982)
- [Paz66] Paz, A.: Some aspects of probabilistic automata. *Information and Control* 9 (1966)
- [PT87] Papadimitriou, C., Tsitsiklis, J.: The complexity of Markov decision processes. *Mathematics of Operations Research* 12(3) (1987)
- [Rab63] Rabin, M.O.: Probabilistic automata. *Information and Control* 6(3), 230–245 (1963)
- [Saf88] Safra, S.: On the complexity of ω -automata. In: *Proc. of the 29th Symposium on Foundations of Computer Science (FOCS 1988)*, pp. 319–327. IEEE Computer Society Press, Los Alamitos (1988)
- [Son71] Sondik, E.J.: *The Optimal Control of Partially Observable Markov Processes*. PhD thesis, Stanford University (1971)
- [SV89] Safra, S., Vardi, M.Y.: On ω -automata and temporal logic. In: *Proc. of the 21st ACM Symposium on Theory of Computing (STOC 1989)*, pp. 127–137. ACM Press, New York (1989)
- [Tho97] Thomas, W.: Languages, automata, and logic. *Handbook of formal languages* 3, 389–455 (1997)
- [VW86] Vardi, M.Y., Wolper, P.: An automata-theoretic approach to automatic program verification. In: *Proc. of the 1st IEEE Symposium on Logic in Computer Science (LICS 1986)*, pp. 332–345. IEEE Computer Society Press, Los Alamitos (1986)