Markov Chains and
Unambiguous Büchi Automata

Christel Baier[1], Stefan Kiefer[2], Joachim Klein[1],
Sascha Klüppelholz[1], David Müller[1], and James Worrell[2]

[1] Technische Universität Dresden, Germany [*]
[2] University of Oxford, Great Britain [**]

Abstract. Unambiguous automata, i.e., nondeterministic automata with the restriction of having at most one accepting run over a word, have the potential to be used instead of deterministic automata in settings where nondeterministic automata can not be applied in general. In this paper, we provide a polynomially time-bounded algorithm for probabilistic model checking of discrete-time Markov chains against unambiguous Büchi automata specifications and report on our implementation and experiments.

1 Introduction

Unambiguity is a widely studied generalization of determinism with many important applications in automata-theoretic approaches, see e.g. [12,13]. A nondetermistic automaton is said to be unambiguous if each word has at most one accepting run. In this paper we consider unambiguous Büchi automata (UBA) over infinite words. Not only are UBA as expressive as the full class of nondeterministic Büchi automata (NBA) [2], they can also be exponentially more succinct than deterministic automata. For example, the language "eventually b occurs and a appears k steps before the first b" over the alphabet $\{a, b, c\}$ is recognizable by a UBA with $k+1$ states (see the UBA on the left of Fig. 1), while a deterministic automaton requires at least 2^k states, regardless of the acceptance condition, as it needs to store the positions of the a's among the last k input symbols. Languages of this type arise in a number of contexts, e.g., absence of unsolicited response in a communication protocol – if a message is received, then it has been sent in the recent past.

Furthermore, the NBA for linear temporal logic (LTL) formulas obtained by applying the classical closure algorithm of [41,40] are unambiguous. The generated automata moreover enjoy the separation property: the languages of the states
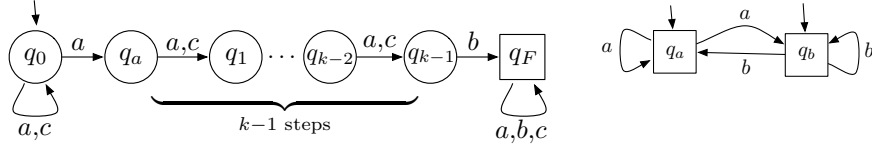
Fig. 1. Two UBA (where final states are depicted as boxes)

are pairwise disjoint. Thus, while the generation of deterministic ω-automata from LTL formulas involves a double exponential blow-up in the worst case, the translation of LTL formulas into separated UBA incurs only a single exponential blow-up. This fact has been observed by several authors, see e.g. [16,35], and recently adapted for LTL with step parameters [42,11].

These nice properties make UBA a potentially attractive alternative to deterministic ω-automata in those applications for which general nondeterministic automata are not suitable. However reasoning about UBA is surprisingly difficult. While many decision problems for unambiguous finite automata (UFA) are known to be solvable in polynomial time [38], the complexity of several fundamental problems for unambiguous automata over infinite words is unknown. This, for instance, applies to the universality problem, which is known to be in P for deterministic Büchi automata (DBA) and PSPACE-complete for NBA. However, the complexity of the universality problem for UBA is a long-standing open problem. Polynomial-time solutions are only known for separated UBA and other subclasses of UBA [8,26].

In the context of probabilistic model checking, UFA provide an elegant approach to compute the probability for a regular safety or co-safety property in finite-state Markov chains [6]. The use of separated UBA for a single exponential-time algorithm that computes the probability for an LTL formula in a Markov chain has been presented in [16]. However, separation is a rather strong condition and non-separated UBA (and even DBA) can be exponentially more succinct than separated UBA, see [8]. This motivates the design of algorithms that operate with general UBA rather than the subclass of separated UBA. Algorithms for the generation of (possibly non-separated) UBA from LTL formulas that are more compact than the separated UBA generated by the classical closure-algorithm have been realized in the tool `Tulip` [34,33] and the automata library `SPOT` [17].

The main theoretical contribution of this paper is a polynomial-time algorithm to compute the probability measure $\Pr^{\mathcal{M}}(\mathcal{L}_\omega(\mathcal{U}))$ of the set of infinite paths generated by a finite-state Markov chain $\mathcal{M}$ that satisfy an ω-regular property given by a (not necessarily separated) UBA $\mathcal{U}$. The existence of such an algorithm has previously been claimed in [6,5,34] (see also [33]). However these previous works share a common fundamental error. Specifically they rely on the claim that if $\Pr^{\mathcal{M}}(\mathcal{L}_\omega(\mathcal{U})) > 0$ then there exists a state s of the Markov chain $\mathcal{M}$ and a state q of the automaton $\mathcal{U}$ such that q accepts almost all trajectories emanating

from s (see [6, Lemma 7.1], [5, Theorem 2][3] , and [34, Section 3.3.1]). While this claim is true in case $\mathcal{U}$ is deterministic [14], it need not hold when $\mathcal{U}$ is merely unambiguous. Indeed, as we explain in Remark 3, a counterexample is obtained by taking $\mathcal{U}$ to be the automaton on the right in Fig. 1 and $\mathcal{M}$ the Markov chain that generates the uniform distribution on $\{a, b\}^\omega$. Appendix A gives a more detailed analysis of the issue, describing precisely the nature of the errors in the proofs of [6,34,5]. To the best of our knowledge these errors are not easily fixable, and the present paper takes a substantially different approach.

Our algorithm involves a two-phase method that first analyzes the strongly connected components (SCCs) of a graph obtained from the product of $\mathcal{M}$ and $\mathcal{U}$, and then computes the value $\mathrm{Pr}^{\mathcal{M}}(\mathcal{L}_\omega(\mathcal{U}))$ using linear equation systems. The main challenge is the treatment of the individual SCCs. For a given SCC we have an equation system comprising a single variable and equation for each vertex (s, q), with s a state of $\mathcal{M}$ and q a state of $\mathcal{U}$. We use results in the spectral theory of non-negative matrices to argue that this equation system has a non-zero solution just in case the SCC makes a non-zero contribution to $\mathrm{Pr}^{\mathcal{M}}(\mathcal{L}_\omega(\mathcal{U}))$. In order to compute the exact value of $\mathrm{Pr}^{\mathcal{M}}(\mathcal{L}_\omega(\mathcal{U}))$ the key idea is to introduce an additional normalization equation. To obtain the latter we identify a pair (s, R), where s is a state of the Markov chain $\mathcal{M}$ and R a set of states of automaton $\mathcal{U}$ such that almost all paths starting in s have an accepting run in $\mathcal{U}$ when the states in R are declared to be initial. The crux of establishing a polynomial bound on the running time of our algorithm is to find such a pair (s, R) efficiently (in particular, without determinizing $\mathcal{U}$) by exploiting structural properties of unambiguous automata.

As a consequence of our main result, we obtain that the *almost universality* problem for UBA, which can be seen as probabilistic variant of the universality problem for UBA and which asks whether a given UBA accepts almost all infinite words, is solvable in polynomial time.

The second contribution of the paper is an implementation of the new algorithm as an extension of the model checker `PRISM`, using the automata library `SPOT` [17] for the generation of UBA from LTL formulas and the `COLT` library [25] for various linear algebra algorithms. We evaluate our approach using the bounded retransmission protocol case study from the PRISM benchmark suite [32] as well as specific aspects of our algorithm using particularly "challenging" UBA.

Outline. Section 2 summarizes our notations for Büchi automata and Markov chains. The theoretical contribution will be presented in Section 3. Section 4 reports on the implementation and experimental results. Section 5 contains concluding remarks. The appendix contains the counterexamples for the previous approaches, proofs and further details on the implementation and results of experimental studies. Further information is available on the website [1] as well.

---

[3] As the flaw is in the handling of the infinite behavior, the claim and proof of Lemma 1 in [5], dealing with unambiguous automata over finite words, remain unaffected.

2 Preliminaries

We suppose the reader to be familiar with the basic notions of ω-automata and Markov chains, see e.g. [22,30]. In what follows, we provide a brief summary of our notations for languages and the uniform probability measure on infinite words, Büchi automata as well as Markov chains.

Prefixes, cylinder sets and uniform probability measure for infinite words.
Throughout the document, we suppose Σ is a finite alphabet with two or more elements. If $w = a_1\, a_2\, a_3 \ldots \in \Sigma^\omega$ is an infinite word then $Pref(w)$ denotes the set of finite prefixes of w, i.e., $Pref(w)$ consists of the empty word and all finite words $a_1\, a_2 \ldots a_n$ where $n \geqslant 1$. Given a finite word $x = a_1\, a_2 \ldots a_n \in \Sigma^*$, the cylinder set of x, denoted $Cyl(x)$, is the set of infinite words $w \in \Sigma^\omega$ such that $x \in Pref(w)$. The set Σ^ω of infinite words over Σ is supposed to be equipped with the σ-algebra generated by the cylinder sets of the finite words and the probability measure given by $\Pr\big(\, Cyl(a_1\, a_2 \ldots a_n)\,\big) \;=\; 1/|\Sigma|^n$ where $a_1, \ldots, a_n \in \Sigma$. Note that all ω-regular languages over Σ are measurable. We often make use of the following lemma (see Appendix B for its proof):

Lemma 1. *If $L \subseteq \Sigma^\omega$ is ω-regular and $\Pr(L) > 0$ then there exists $x \in \Sigma^*$ such that $\Pr\big\{\, w \in \Sigma^\omega \,:\, xw \in L \,\big\} = 1$.*

Büchi automata.
A nondeterministic Büchi automaton is a tuple $\mathcal{A} = (Q, \Sigma, \delta, Q_0, F)$ where Q is a finite set of states, $Q_0 \subseteq Q$ is a set of initial states, Σ denotes the alphabet, $\delta : Q \times \Sigma \to 2^Q$ denotes the transition function, and F is a set of accepting states. We extend the transition function $\delta : 2^Q \times \Sigma \to 2^Q$ in the standard way for subsets of Q and finite words over Σ. Given states $q, p \in Q$ and a finite word $x = a_1\, a_2 \ldots a_n \in \Sigma^*$ then a run for x from q to p is a sequence $q_0\, q_1 \ldots q_n \in Q^+$ with $q_0 = q$, $q_n = p$ and $q_{i+1} \in \delta(q_i, a_{i+1})$ for $0 \leqslant i < n$. A run in $\mathcal{A}$ for an infinite word $w = a_1\, a_2\, a_3 \ldots \in \Sigma^\omega$ is an infinite sequence $\rho = q_0\, q_1 \ldots \in Q^\omega$ such that $q_{i+1} \in \delta(q_i, a_{i+1})$ for all $i \in \mathbb{N}$ and $q_0 \in Q_0$. Run ρ is called accepting, if $q_i \in F$ for infinitely many $i \in \mathbb{N}$. The language $\mathcal{L}_\omega(\mathcal{A})$ of accepted words consists of all infinite words $w \in \Sigma^\omega$ that have at least one accepting run. If $R \subseteq Q$ then $\mathcal{A}[R]$ denotes the automaton $\mathcal{A}$ with R as set of initial states. For $q \in Q$, $\mathcal{A}[q] = \mathcal{A}[\{q\}]$. If $\mathcal{A}$ is understood from the context, then we write $\mathcal{L}_\omega(R)$ rather than $\mathcal{L}_\omega(\mathcal{A}[R])$ and $\mathcal{L}_\omega(q)$ rather than $\mathcal{L}_\omega(\mathcal{A}[q])$. $\mathcal{A}$ is called deterministic if Q_0 is a singleton and $|\delta(q,a)| \leqslant 1$ for all states q and symbols $a \in \Sigma$ and unambiguous if each word $w \in \Sigma^\omega$ has at most one accepting run in $\mathcal{A}$. Clearly, each deterministic automaton is unambiguous. We use the shortform notations NBA, DBA and UBA for nondeterministic, deterministic and unambiguous Büchi automata, respectively.

Markov chains. In this paper we only consider finite-state discrete-time Markov chains. Formally, a Markov chain is a triple $\mathcal{M} = (S, P, \iota)$ where S is a finite set of states, $P : S \times S \to [0,1]$ is the transition probability function satisfying

$\sum_{s' \in S} P(s, s') = 1$ for all states $s \in S$ and ι an initial distribution on S. We write $\mathrm{Pr}^{\mathcal{M}}$ to denote the standard probability measure on the infinite paths of $\mathcal{M}$. For $s \in S$, the notation $\mathrm{Pr}_s^{\mathcal{M}}$ will be used for $\mathrm{Pr}^{\mathcal{M}_s}$ where $\mathcal{M}_s = (S, P, \mathrm{Dirac}[s])$ and $\mathrm{Dirac}[s] : S \to [0, 1]$ denotes the Dirac distribution that assigns probability 1 to state s and 0 to all other states. If $L \subseteq S^\omega$ is measurable then $\mathrm{Pr}^{\mathcal{M}}(L)$ is a shortform notation for the probability for $\mathcal{M}$ to generate an infinite path π with $\pi \in L$.

Occasionally, we also consider Markov chains with transition labels in some alphabet Σ. These are defined as triples $\mathcal{M} = (S, P, \iota)$ where S and ι are as above and the transition probability function is of the type $P : S \times \Sigma \times S \to [0, 1]$ such that $\sum_{(a,s') \in \Sigma \times S} P(s, a, s') = 1$ for all states $s \in S$. If $L \subseteq \Sigma^\omega$ is measurable then $\mathrm{Pr}^{\mathcal{M}}(L)$ denotes the probability measure of the set of infinite paths π where the projection to the transition labels constitutes a word in L. Furthermore, if $\mathcal{M}[\Sigma] = (S, P, \iota)$ is a transition-labeled Markov chain where $S = \{s\}$ is a singleton and $P(s, a, s) = 1/|\Sigma|$ for all symbols $a \in \Sigma$, then $\mathrm{Pr}^{\mathcal{M}[\Sigma]}(L) = \mathrm{Pr}(L)$ for all measurable languages L.

3 Analysis of Markov chains against UBA-specifications

The task of the *probabilistic model-checking problem* for a given Markov chain $\mathcal{M}$ and NBA $\mathcal{A}$ is to compute $\mathrm{Pr}^{\mathcal{M}}(\mathcal{L}_\omega(\mathcal{A}))$ where $\mathcal{M}$ is either a plain Markov chain and the alphabet of $\mathcal{A}$ is the state space of $\mathcal{M}$ or the transitions of $\mathcal{M}$ are labeled by symbols of the alphabet of $\mathcal{A}$. The *positive model-checking problem* for $\mathcal{M}$ and $\mathcal{A}$ asks whether $\mathrm{Pr}^{\mathcal{M}}(\mathcal{L}_\omega(\mathcal{A})) > 0$. Likewise, the *almost-sure model-checking problem* for $\mathcal{M}$ and $\mathcal{A}$ denotes the task to check whether $\mathrm{Pr}^{\mathcal{M}}(\mathcal{L}_\omega(\mathcal{A})) = 1$. While the positive and the almost-sure probabilistic model-checking problems for Markov chains and NBA are both known to be PSPACE-complete [39,14], the analysis of Markov chains against UBA-specification can be carried out efficiently as stated in the following theorem:

Theorem 2. *Given a Markov chain $\mathcal{M}$ and a UBA $\mathcal{U}$, the value $\mathrm{Pr}^{\mathcal{M}}(\mathcal{L}_\omega(\mathcal{U}))$ is computable in time polynomial in the sizes of $\mathcal{M}$ and $\mathcal{U}$.*

Remark 3. The statement of Theorem 2 has already been presented in [4] (see also [34,5]). However, the presented algorithm to compute $\mathrm{Pr}^{\mathcal{M}}(\mathcal{L}_\omega(\mathcal{U}))$ is flawed. This approach, rephrased for the special case where the task is to compute $\mathrm{Pr}(\mathcal{L}_\omega(\mathcal{U}))$ for a given positive UBA $\mathcal{U}$ (which means a UBA where $\mathrm{Pr}(\mathcal{L}_\omega(\mathcal{U})) > 0$) relies on the mistaken belief that there is at least one state q in $\mathcal{U}$ such that $\mathrm{Pr}(\mathcal{L}_\omega(\mathcal{U}[q])) = 1$. However, such states need not exist. To illustrate this, we consider the UBA $\mathcal{U}$ with two states q_a and q_b and $\delta(q_a, a) = \delta(q_b, b) = \{q_a, q_b\}$ and $\delta(q_a, b) = \delta(q_b, a) = \varnothing$. (See the UBA on the right of Fig. 1.) Both states are initial and final. Clearly, $\mathcal{L}_\omega(\mathcal{U}[q_a]) = a\Sigma^\omega$ and $\mathcal{L}_\omega(\mathcal{U}[q_b]) = b\Sigma^\omega$. Thus, $\mathcal{U}$ is universal and $\mathrm{Pr}(\mathcal{L}_\omega(\mathcal{U})) = 1$, while $\mathrm{Pr}(\mathcal{L}_\omega(\mathcal{U}[q_a])) = \mathrm{Pr}(\mathcal{L}_\omega(\mathcal{U}[q_b])) = \frac{1}{2}$.

Outline of Section 3. The remainder of Section 3 is devoted to the proof of Theorem 2. We first assume that the Markov chain $\mathcal{M}$ generates all words according to a uniform distribution and explain how to compute the value $\Pr(\mathcal{L}_\omega(\mathcal{U}))$ for a given UBA $\mathcal{U}$ in polynomial time. For this, we first address the case of strongly connected UBA (Section 3.1) and then lift the result to the general case (Section 3.2). The central idea of the algorithm relies on the observation that each positive, strongly connected UBA has "recurrent sets" of states, called *cuts*. We exploit structural properties of unambiguous automata for the efficient construction of a cut and show how to compute the values $\Pr(\mathcal{L}_\omega(\mathcal{U}[q]))$ for the states of $\mathcal{U}$ by a linear equation system with one equation per state and one equation for the generated cut. Furthermore, positivity of a UBA $\mathcal{U}$ (i.e., $\Pr(\mathcal{L}_\omega(\mathcal{U})) > 0$) is shown to be equivalent to the existence of a positive solution of the system of linear equations for the states. Finally, we explain how to adapt these techniques to general Markov chains (Section 3.3).

3.1 Strongly connected UBA

We start with some general observations about strongly connected Büchi automata under the probabilistic semantics. For this, we suppose $\mathcal{A} = (Q, \Sigma, \delta, Q_0, F)$ is a strongly connected NBA where Q_0 and F are nonempty. Clearly, $\mathcal{L}_\omega(q) \neq \varnothing$ for all states q and

$$\Pr(\mathcal{L}_\omega(\mathcal{A})) > 0 \quad \text{iff} \quad \Pr(\mathcal{L}_\omega(q)) > 0 \text{ for some state } q$$
$$\text{iff} \quad \Pr(\mathcal{L}_\omega(q)) > 0 \text{ for all states } q$$

Moreover, almost all words $w \in \Sigma^\omega \setminus \mathcal{L}_\omega(\mathcal{A})$ have a finite prefix x with $\delta(Q_0, x) = \varnothing$ (for the proof see Lemma 18 in Appendix C):

Lemma 4 (Measure of strongly connected NBA). *For each strongly connected NBA $\mathcal{A}$ with at least one final state, we have:*

$$\Pr(\mathcal{L}_\omega(\mathcal{A})) \;=\; 1 - \Pr\big\{\, w \in \Sigma^\omega \,:\, w \text{ has a finite prefix } x \text{ with } \delta(Q_0, x) = \varnothing \,\big\}$$

In particular, $\mathcal{A}$ is almost universal if and only if $\delta(Q_0, x) \neq \varnothing$ for all finite words $x \in \Sigma^*$. This observation will be crucial at several places in the soundness proof of our algorithm for UBA, but can also be used to establish PSPACE-hardness of the positivity (probabilistic nonemptiness) and almost universality problem for strongly connected NBA (see Theorem 27 in Appendix C). For computing $\Pr(\mathcal{L}_\omega(\mathcal{U}))$ given a UBA $\mathcal{U}$, it suffices to compute the values $\Pr(\mathcal{L}_\omega(q))$ for the (initial) states of $\mathcal{U}$ as we have

$$\Pr(\mathcal{L}_\omega(\mathcal{U})) \;=\; \sum_{q \in Q_0} \Pr(\mathcal{L}_\omega(q))$$

Furthermore, in each strongly connected UBA, the accepting runs of almost all words $w \in \mathcal{L}_\omega(\mathcal{U})$ visit each state of $\mathcal{U}$ infinitely often (see Theorem 29 in Appendix D.1).

Deciding positivity for strongly connected UBA. The following lemma provides a criterion to check positivity of a strongly connected UBA in polynomial time using standard linear algebra techniques.

Lemma 5. *Let $\mathcal{U}$ be a strongly connected UBA with at least one initial and one final state, and*

$$(*) \qquad \zeta_q \;\; = \;\; \frac{1}{|\Sigma|} \cdot \sum_{a \in \Sigma} \sum_{p \in \delta(q,a)} \zeta_p \qquad \text{for all } q \in Q$$

Then, the following statements are equivalent:

(1) $\Pr(\mathcal{L}_\omega(\mathcal{U})) > 0$,
(2) *the linear equation system (*) has a strictly positive solution, i.e., a solution $(\zeta_q^*)_{q \in Q}$ with $\zeta_q^* > 0$ for all $q \in Q$,*
(3) *the linear equation system (*) has a non-zero solution.*

Given the strongly connected UBA $\mathcal{U}$ with at least one final state, we define a matrix $M \in [0,1]^{Q \times Q}$ by $M_{p,q} = \frac{1}{|\Sigma|}|\{a \in \Sigma : q \in \delta(p,a)\}|$ for all $p, q \in Q$. Since $\mathcal{U}$ is strongly connected, M is irreducible. Write $\rho(M)$ for the spectral radius of M. We will use the following Lemma in the proof of Lemma 5.

Lemma 6. *We have $\rho(M) \leq 1$. Moreover $\rho(M) = 1$ if and only if $\Pr(\mathcal{L}_\omega(\mathcal{U})) > 0$.*

Proof. For $p, q \in Q$ and $n \in \mathbb{N}$, let $E_{p,n,q} \subseteq \Sigma^\omega$ denote the event of all words $w = a_1 a_2 \ldots$ such that $q \in \delta(p, a_1 a_2 \ldots a_n)$. Its probability under the uniform distribution on Σ^ω is an entry in the n-th power of M:

$$\Pr(E_{p,n,q}) = (M^n)_{p,q} \tag{1}$$

In particular, $M_{p,q}^n \leq 1$ for all n. From the boundedness of M^n it follows (e.g., by [24, Corollary 8.1.33]) that $\rho(M) \leq 1$. The same result implies that

$$
\begin{aligned}
\rho(M) = 1 \quad &\Longleftrightarrow \quad \limsup_{n \to \infty}(M^n)_{p,q} > 0 \quad \text{for all } p, q \in Q \\
&\Longleftrightarrow \quad \limsup_{n \to \infty}(M^n)_{p,q} > 0 \quad \text{for some } p, q \in Q
\end{aligned}
\tag{2}
$$

For the rest of the proof, fix some state $p \in Q$. By the observations from the beginning of Section 3.1 it suffices to show that $\Pr(\mathcal{L}_\omega(p)) > 0$ if and only if $\rho(M) = 1$. To this end, consider the event $E_{p,n} := \bigcup_{q \in Q} E_{p,n,q}$. Notice that $(E_{p,n})_{n \in \mathbb{N}}$ forms a decreasing family of sets. We have:

$$
\begin{aligned}
\Pr(\mathcal{L}_\omega(p)) &= \lim_{n \to \infty} \Pr(E_{p,n}) & \text{by Lemma 4} \\
&= \lim_{n \to \infty} \Pr\left(\bigcup_{q \in Q} E_{p,n,q} \right) & \text{definition of } E_{p,n}
\end{aligned}
\tag{3}
$$

Assuming that $\rho(M) = 1$, we show that $\Pr(\mathcal{L}_\omega(p)) > 0$. Let $q \in Q$. We have:

$$
\begin{aligned}
\Pr(\mathcal{L}_\omega(p)) &\geq \limsup_{n\to\infty} \Pr(E_{p,n,q}) && \text{by (3)} \\
&= \limsup_{n\to\infty} (M^n)_{p,q} && \text{by (1)} \\
&> 0 && \text{by (2)}
\end{aligned}
$$

Conversely, assuming that $\rho(M) < 1$, we show that $\Pr(\mathcal{L}_\omega(p)) = 0$.

$$
\begin{aligned}
\Pr(\mathcal{L}_\omega(p)) &= \lim_{n\to\infty} \Pr\left(\bigcup_{q\in Q} E_{p,n,q}\right) && \text{by (3)} \\
&\leq \limsup_{n\to\infty} \sum_{q\in Q} \Pr(E_{p,n,q}) && \text{union bound} \\
&= \limsup_{n\to\infty} \sum_{q\in Q} (M^n)_{p,q} && \text{by (1)} \\
&= 0 && \text{by (2)}
\end{aligned}
$$

This concludes the proof. $\qquad\square$

Proof (of Lemma 5). "(1) $\Longrightarrow$ (2)": Suppose $\Pr(\mathcal{L}_\omega(\mathcal{U})) > 0$. Define the vector $(\zeta_q^*)_{q\in Q}$ with $\zeta_q^* = \Pr(\mathcal{L}_\omega(q))$. It holds that

$$
\mathcal{L}_\omega(q) = \bigcup_{a\in\Sigma} \bigcup_{p\in\delta(q,a)} \{aw : w \in \mathcal{L}_\omega(p)\}
$$

Since $\mathcal{U}$ is unambiguous, the sets $\{aw : w \in \mathcal{L}_\omega(p)\}$ are pairwise disjoint. So, the vector $(\zeta_q^*)_{q\in Q}$ is a solution to the equation system.

As $\Pr(\mathcal{L}_\omega(\mathcal{U})) > 0$ and $\mathcal{U}$ is strongly connected, the observation at the beginning of Section 3.1 yields that $\Pr(\mathcal{L}_\omega(q)) > 0$ for all states q. Thus, the vector $(\zeta_q^*)_{q\in Q}$ is strictly positive.

"(2) $\Longrightarrow$ (3)" holds trivially.

"(3) $\Longrightarrow$ (1)": Suppose ζ^* is a non-zero solution of the linear equation system. Then, $M\zeta^* = \zeta^*$. Thus, 1 is an eigenvalue of M. This yields $\rho(M) \geqslant 1$. But then $\rho(M) = 1$ and $\Pr(\mathcal{L}_\omega(\mathcal{U})) > 0$ by Lemma 6. $\qquad\square$

Computing pure cuts for positive, strongly connected UBA. The key observation to compute the values $\Pr(\mathcal{L}_\omega(q))$ for the states q of a positive, strongly connected UBA $\mathcal{U}$ is the existence of so-called cuts. These are sets C of states with pairwise disjoint languages such that almost all words have an accepting run starting in some state $q \in C$. More precisely:

Definition 7 ((Pure) cut). *Let $\mathcal{U}$ be a UBA and $C \subseteq Q$. C is called a* cut *for $\mathcal{U}$ if $\mathcal{L}_\omega(q) \cap \mathcal{L}_\omega(p) = \varnothing$ for all $p, q \in C$ with $p \neq q$ and $\mathcal{U}[C]$ is almost universal. A cut is called* pure *if it has the form $\delta(q, z)$ for some state q and some finite word $z \in \Sigma^*$.*

Obviously, $\mathcal{U}$ is almost universal iff Q_0 is a cut. If $q \in Q$ and K_q denotes the set of finite words $z \in \Sigma^*$ such that $\delta(q, z)$ is a cut then $\Pr(\mathcal{L}_\omega(q))$ equals the probability measure of the language L_q consisting of all infinite words $w \in \Sigma^\omega$ that have a prefix in K_q. See Lemma 40 in Appendix D.2.

Lemma 8 (Characterization of pure cuts). *Let $\mathcal{U}$ be a strongly connected UBA. For all $q \in Q$ and $z \in \Sigma^*$ we have: $\delta(q, z)$ is a cut iff $\delta(q, zy) \neq \varnothing$ for each word $y \in \Sigma^*$. Furthermore, if $\mathcal{U}$ is positive then for each cut C:*

    C is pure, i.e., $C = \delta(q, z)$ for some state-word pair $(q, z) \in Q \times \Sigma^*$

  iff for each state $q \in Q$ there is some word $z \in \Sigma^*$ with $C = \delta(q, z)$

  iff for each cut C' there is some word $y \in \Sigma^*$ with $C = \delta(C', y)$

The proof of Lemma 8 is provided in Appendix D.2 (see Corollary 34 and Lemma 35). By Lemma 1 and Lemma 8 we get:

Corollary 9. *If $\mathcal{U}$ is a strongly connected UBA then $\Pr(\mathcal{L}_\omega(\mathcal{U})) > 0$ iff $\mathcal{U}$ has a pure cut.*

For the rest of Section 3.1, we suppose that $\mathcal{U}$ is positive and strongly connected. The second part of Lemma 8 yields that the pure cuts constitute a bottom strongly connected component of the automaton obtained from $\mathcal{U}$ using the standard powerset construction. The goal is now to design an efficient (polynomially time-bounded) algorithm for the generation of a pure cut. For this, we observe that if $q, p \in Q$, $q \neq p$, then $\{q, p\} \subseteq C$ for some pure cut C iff there exists a word y such that $\{q, p\} \subseteq \delta(q, y)$. See Lemma 37 in Appendix D.2.

Definition 10 (Extension). *A word $y \in \Sigma^*$ is an extension for a state-word pair $(q, z) \in Q \times \Sigma^*$ iff there exists a state $p \in Q$ such that $q \neq p$, $\delta(p, z) \neq \varnothing$ and $\{q, p\} \subseteq \delta(q, y)$.*

It is easy to see that if y is an extension of (q, z), then $\delta(q, yz)$ is a proper superset of $\delta(q, z)$ (see Lemma 41 in Appendix D.2). Furthermore, for all state-word pairs $(q, z) \in Q \times \Sigma^*$ (see Lemma 42 in Appendix D.2):

$$\delta(q, z) \text{ is a cut} \quad \text{iff} \quad \text{there is no extension for } (q, z)$$

These observations lead to the following algorithm for the construction of a pure cut. We pick an arbitrary state q in the UBA and start with the empty word $z_0 = \varepsilon$. The algorithm iteratively seeks for an extension for the state-word pair (q, z_i). If an extension y_i for (q, z_i) has been found then we switch to the word $z_{i+1} = y_i z_i$. If no extension exists then (q, z_i) is a pure cut. In this way, the algorithm generates an increasing sequence of subsets of Q,

$$\delta(q, z_0) \subsetneq \delta(q, z_1) \subsetneq \delta(q, z_2) \subsetneq \ldots \subsetneq \delta(q, z_k),$$

which terminates after at most $|Q|$ steps and yields a pure cut $\delta(q, z_k)$.

It remains to explain an efficient realization of the search for an extension of the state-word pairs (q, z_i). The idea is to store the sets $Q_i[p] = \delta(p, z_i)$ for all states p. The sets $Q_i[p]$ can be computed iteratively by:

$$Q_0[p] = \{p\} \quad \text{and} \quad Q_{i+1}[p] = \bigcup_{r \in \delta(p,y_i)} Q_i[r]$$

To check whether (q, z_i) has an extension we apply standard techniques for the intersection problem for the languages $H_{q,q} = \{y \in \Sigma^* : q \in \delta(q, y)\}$ and $H_{q,F_i} = \{y \in \Sigma^* : \delta(q, y) \cap F_i \neq \varnothing\}$ where $F_i = \{p \in Q \setminus \{q\} : Q_i[p] \neq \varnothing\}$. Then, for each word $y \in \Sigma^*$ we have: $y \in H_{q,q} \cap H_{q,F_i}$ if and only if y is an extension of (q, z_i). The languages $H_{q,q}$ and H_{q,F_i} are recognized by the NFA $\mathcal{U}_{q,q} = (Q, \Sigma, \delta, q, q)$ and $\mathcal{U}_{q,F_i} = (Q, \Sigma, \delta, q, F_i)$. Thus, to check the existence of an extension and to compute an extension y (if existent) where the word y has length at most $|Q|^2$, we may run an emptiness check for the product-NFA $\mathcal{U}[q, q] \otimes \mathcal{U}[q, F_i]$. We conclude:

Corollary 11. *Given a positive, strongly connected UBA $\mathcal{U}$, a pure cut can be computed in time polynomial in the size of $\mathcal{U}$.*

Computing the measure of positive, strongly connected UBA. We suppose that $\mathcal{U} = (Q, \Sigma, \delta, Q_0, F)$ is a positive, strongly connected UBA and C is a cut. (C might be a pure cut that has been computed by the techniques explained above. However, in Theorem 12 C can be any cut.) Consider the linear equation system of Lemma 5 with variables ζ_q for all states $q \in Q$ and add the constraint that the variables ζ_q for $q \in C$ sum up to 1.

Theorem 12. *Let $\mathcal{U}$ be a positive, strongly connected UBA and C a cut. Then, the probability vector $(\Pr(\mathcal{L}_\omega(q)))_{q \in Q}$ is the unique solution of the following linear equation system:*

$$(1) \quad \zeta_q = \frac{1}{|\Sigma|} \cdot \sum_{a \in \Sigma} \sum_{p \in \delta(q,a)} \zeta_p \quad \text{for all states } q \in Q$$

$$(2) \quad \sum_{q \in C} \zeta_q = 1$$

Proof. Let $n = |Q|$. Define a matrix $M \in [0,1]^{Q \times Q}$ by $M_{q,p} = |\{a \in \Sigma : p \in \delta(q, a)\}|/|\Sigma|$ for all $q, p \in Q$. Then, the n equations (1) can be written as $\zeta = M\zeta$, where $\zeta = (\zeta_q)_{q \in Q}$ is a vector of n variables. It is easy to see that the values $\zeta_q^* = \Pr(\mathcal{L}_\omega(q))$ for $q \in Q$ satisfy the equations (1). That is, defining $\zeta^* = (\zeta_q^*)_{q \in Q}$ we have $\zeta^* = M\zeta^*$. By the definition of a cut, those values also satisfy equation (2).

It remains to show uniqueness. We employ Perron-Frobenius theory as follows. Since $\zeta^* = M\zeta^*$, the vector ζ^* is an eigenvector of M with eigenvalue 1. Since ζ^* is strictly positive (i.e., positive in all components), it follows from [7, Corollary 2.1.12] that $\rho = 1$ for the spectral radius ρ of M. Since $\mathcal{U}$ is strongly connected, matrix M is irreducible. By [7, Theorem 2.1.4 (b)] the spectral radius $\rho = 1$ is a simple eigenvalue of M, i.e., all solutions of $\zeta = M\zeta$ are scalar multiples of ζ^*. Among those multiples, only ζ^* satisfies equation (2). Uniqueness follows. ∎

Together with the criterion of Lemma 5 to check whether a given strongly connected UBA is positive, we obtain a polynomially time-bounded computation scheme for the values $\Pr(\mathcal{L}_\omega(q))$ for the states q of a given strongly connected UBA. The next section shows how to lift these results for arbitrary UBA.

3.2 Computing the measure of arbitrary UBA

In what follows, let $\mathcal{U} = (Q, \Sigma, \delta, Q_0, F)$ be a (possibly not strongly connected) UBA. We assume that all states are reachable from Q_0 and that F is reachable from all states. Thus, $\mathcal{L}_\omega(q) \neq \varnothing$ for all states q.

Let $\mathcal{C}$ be a strongly connected component (SCC) of $\mathcal{U}$. $\mathcal{C}$ is called non-trivial if $\mathcal{C}$ viewed as a direct graph contains at least one edge, i.e., if $\mathcal{C}$ is cyclic. $\mathcal{C}$ is called bottom if $\delta(q, a) \subseteq \mathcal{C}$ for all $q \in \mathcal{C}$ and all $a \in \Sigma$. We define Q_{BSCC} to be the set of all states $q \in Q$ that belong to some bottom SCC (BSCC) of $\mathcal{U}$. If $\mathcal{C}$ is a non-trivial SCC of $\mathcal{U}$ and $p \in \mathcal{C}$ then the sub-NBA

$$\mathcal{U}\big|_{\mathcal{C},p} \;\; = \;\; (\mathcal{C}, \Sigma, \delta|_{\mathcal{C}}, \{p\}, \mathcal{C} \cap F)$$

of $\mathcal{U}$ with state space $\mathcal{C}$, initial state p and the transition function $\delta|_{\mathcal{C}}$ given by $\delta|_{\mathcal{C}}(q, a) = \delta(q, a) \cap \mathcal{C}$ is strongly connected and unambiguous. Let L_p be the accepted language, i.e., $L_p = \mathcal{L}_\omega(\mathcal{U}\big|_{\mathcal{C},p})$. The values $\Pr(L_p)$, $p \in \mathcal{C}$, can be computed using the techniques for strongly connected UBA presented in Section 3.1. A non-trivial SCC $\mathcal{C}$ is said to be positive if $\Pr(L_p) > 0$ for all/some state(s) p in $\mathcal{C}$.

We perform the following preprocessing. As before, for any $p \in Q$ we write $\mathcal{L}_\omega(p)$ for $\mathcal{L}_\omega(\mathcal{U}[p])$, and call p zero if $\Pr(\mathcal{L}_\omega(p)) = 0$. First we remove all states that are not reachable from any initial state. Then we run standard graph algorithms to compute the directed acyclic graph (DAG) of SCCs of $\mathcal{U}$. By processing the DAG bottom-up we can remove all zero states by running the following loop: If all BSCCs are marked (initially, all SCCs are unmarked) then exit the loop; otherwise pick an unmarked BSCC $\mathcal{C}$.

- If $\mathcal{C}$ is trivial or does not contain any final state then we remove it: more precisely, we remove it from the DAG of SCCs, and we modify $\mathcal{U}$ by deleting all transitions $p \xrightarrow{a} q$ where $q \in \mathcal{C}$.
- Otherwise, $\mathcal{C}$ is a non-trivial BSCC with at least one final state. We check whether $\mathcal{C}$ is positive by applying the techniques of Section 3.1. If it is positive, we mark it; otherwise we remove it as described above.

Note that this loop does not change $\Pr(\mathcal{L}_\omega(p))$ for any state p.

Let Q_{BSCC} denote the set of states in $\mathcal{U}$ that belong to some BSCC. The values $\Pr(\mathcal{L}_\omega(p))$ for the states $p \in Q_{BSCC}$ can be computed using the techniques of Section 3.1. The remaining task is to compute the values $\Pr(\mathcal{L}_\omega(q))$ for the states $q \in Q \setminus Q_{BSCC}$. For $q \in Q \setminus Q_{BSCC}$, let $\beta_q = 0$ if $\delta(q, a) \cap Q_{BSCC} = \varnothing$ for all $a \in \Sigma$. Otherwise:

$$\beta_q \;\; = \;\; \frac{1}{|\Sigma|} \cdot \sum_{a \in \Sigma} \sum_{p \in \delta(q,a) \cap Q_{BSCC}} \Pr(\mathcal{L}_\omega(p))$$

In Appendix D.4 (see Theorem 43) we show:

Theorem 13. *If all BSCCs of $\mathcal{U}$ are non-trivial and positive, then the linear equation system*

$$\zeta_q \;=\; \frac{1}{|\Sigma|} \;\cdot\; \sum_{a \in \Sigma} \; \sum_{\substack{r \,\in\, \delta(q,a) \\ r \,\notin\, Q_{BSCC}}} \zeta_r \;\;+\;\; \beta_q \qquad \text{for } q \in Q \setminus Q_{BSCC}$$

has a unique solution, namely $\zeta_q^* = \Pr(\,\mathcal{L}_\omega(q)\,)$.

This yields that the value $\Pr(\mathcal{L}_\omega(\mathcal{U}))$ for given UBA $\mathcal{U}$ is computable in polynomial time.

Remark 14. For the special case where $\delta(q,a) = \{q\}$ for all $q \in F$ and $a \in \Sigma$, the language of $\mathcal{U}$ is a co-safety property and $\Pr(\mathcal{L}_\omega(q)) = 1$ if $q \in F = Q_{BSCC}$, when we assume that all BSCCs are non-trivial and positive. In this case, the linear equation system in Theorem 13 coincides with the linear equation system presented in [6] for computing the probability measure of the language of $\mathcal{U}$ viewed as an UFA.

Remark 15. As a consequence of our results, the positivity problem ("does $\Pr(\mathcal{L}_\omega(\mathcal{U})) > 0$ hold?") and the almost universality problem ("does $\Pr(\mathcal{L}_\omega(\mathcal{U})) = 1$ hold?") for UBA are solvable in polynomial time. This should be contrasted with the standard (non-probabilistic) semantics of UBA and the corresponding results for NBA. The non-emptiness problem for UBA is in P (this already holds for NBA), while the complexity-theoretic status of the universality problem for UBA is a long-standing open problem. For standard NBA, it is well known that the non-emptiness problem is in P and the universality problem is PSPACE-complete. However, the picture changes when switching to NBA with the probabilistic semantics as both the positivity problem and the almost universality problem for NBA are PSPACE-complete, even for strongly connected NBA (see Theorem 27 in Appendix C).

3.3 Probabilistic model checking of Markov chains against UBA

To complete the proof of Theorem 2, we show how the results of the previous section can be adapted to compute the value $\Pr^{\mathcal{M}}(\mathcal{L}_\omega(\mathcal{U}))$ for a Markov chain $\mathcal{M} = (S, P, \iota)$ and a UBA $\mathcal{U} = (Q, \Sigma, \delta, Q_0, F)$ with alphabet $\Sigma = S$. [4] The necessary adaptions to the proofs are detailed in Appendix D.5.

---

[4] In practice, e.g., when the UBA is obtained from an LTL formula, the alphabet of the UBA is often defined as $\Sigma = 2^{AP}$ over a set of atomic propositions AP and the Markov chain is equipped with a labeling function from states to the atomic propositions that hold in each state. Clearly, unambiguity w.r.t. the alphabet 2^{AP} implies unambiguity w.r.t. the alphabet S when switching from the original transition function $\delta : Q \times 2^{AP} \to 2^Q$ to the transition function $\delta_S : Q \times S \to 2^Q$ given by $\delta_S(q, s) = \delta(q, L(s))$, where $L : S \to 2^{AP}$ denotes the labeling function of $\mathcal{M}$.

If $\mathcal{A}$ is an NBA over the alphabet S and $s \in S$, then $\mathrm{Pr}_s^{\mathcal{M}}(\mathcal{A})$ denotes the probability $\mathrm{Pr}_s^{\mathcal{M}}(\Pi)$ with Π being the set of infinite paths $\pi = s_0 s_1 \ldots \in S^\omega$ starting with $s_0 = s$ and such that $s_1 s_2 \ldots \in \mathcal{L}_\omega(\mathcal{A})$. Our algorithm relies on the observation that

$$\mathrm{Pr}^{\mathcal{M}}(\mathcal{L}_\omega(\mathcal{U})) \;\; = \;\; \sum_{s \in S} \iota(s) \cdot \mathrm{Pr}_s^{\mathcal{M}}(\mathcal{U}[\delta(Q_0, s)])$$

As the languages of the UBA $\mathcal{U}[q]$ for $q \in \delta(Q_0, s)$ are pairwise distinct (by the unambiguity of $\mathcal{U}$), we have $\mathrm{Pr}_s^{\mathcal{M}}(\mathcal{U}[\delta(Q_0, s)]) \;=\; \sum_{q \in \delta(Q_0, s)} \mathrm{Pr}_s^{\mathcal{M}}(\mathcal{U}[q])$.

Thus, the task is to compute the values $\mathrm{Pr}_s^{\mathcal{M}}(\mathcal{U}[q])$ for $s \in S$ and $q \in Q$. As a first step, we build a UBA $\mathcal{P} = \mathcal{M} \otimes \mathcal{U}$ that arises from the synchronous product of the UBA $\mathcal{U}$ with the underlying graph of the Markov chain $\mathcal{M}$. Formally, $\mathcal{P} = (S \times Q, \Sigma, \Delta, Q_0', S \times F)$ where Q_0' consists of all pairs $\langle s, q \rangle \in S \times Q$ where $\iota(s) > 0$ and $q \in \delta(Q_0, s)$. Let $s, t \in S$ and $q \in Q$. If $P(s, t) = 0$ then $\Delta(\langle s, q \rangle, t) = \varnothing$, while for $P(s, t) > 0$, the set $\Delta(\langle s, q \rangle, t)$ consists of all pairs $\langle t, p \rangle$ where $p \in \delta(q, s)$. We are only concerned with the reachable fragment of the product.

Given that $\mathcal{M}$ viewed as an automaton over the alphabet S behaves deterministically and we started with an unambiguous automaton $\mathcal{U}$, the product $\mathcal{P}$ is unambiguous as well. Let $\mathcal{P}[s, q]$ denote the UBA resulting from $\mathcal{P}$ by declaring $\langle s, q \rangle$ to be initial. It is easy to see that $\mathrm{Pr}_s^{\mathcal{M}}(\mathcal{P}[s, q]) = \mathrm{Pr}_s^{\mathcal{M}}(\mathcal{U}[q])$ for all states $\langle s, q \rangle$ of $\mathcal{P}$, as the product construction only removes transitions in $\mathcal{U}$ that can not occur in the Markov chain. Our goal is thus to compute the values $\mathrm{Pr}_s^{\mathcal{M}}(\mathcal{P}[s, q])$. For this, we remove all states $\langle s, q \rangle$ from $\mathcal{P}$ that can not reach a state in $S \times F$. Then, we determine the non-trivial SCCs of $\mathcal{P}$ and, for each such SCC $\mathcal{C}$, we analyze the sub-UBA $\mathcal{P}\big|_{\mathcal{C}}$ obtained by restricting to the states in $\mathcal{C}$. An SCC $\mathcal{C}$ of $\mathcal{P}$ is called positive if $\mathrm{Pr}_s^{\mathcal{M}}(\mathcal{P}\big|_{\mathcal{C}}[s, q]) > 0$ for all/any $\langle s, q \rangle \in \mathcal{C}$. As in Section 3.2, we treat the SCCs in a bottom-up manner, starting with the BSCCs and removing them if they are non-positive. Clearly, if a BSCC $\mathcal{C}$ of $\mathcal{P}$ does not contain a final state or is trivial, then $\mathcal{C}$ is not positive. Analogously to Lemma 5, a non-trivial BSCC $\mathcal{C}$ in $\mathcal{P}$ containing at least one final state is positive if and only if the linear equation system

$$(*) \qquad \zeta_{s,q} \;\; = \;\; \sum_{t \in \mathrm{Post}(s)} \sum_{p \in \delta_{\mathcal{C}}(q,t)} P(s, t) \cdot \zeta_{t,p} \quad \text{for all } \langle s, q \rangle \in \mathcal{C}$$

has a strictly positive solution if and only if $(*)$ has a non-zero solution. Here, $\mathrm{Post}(s) = \{t \in S : P(s, t) > 0\}$ denotes the set of successors of state s in $\mathcal{M}$ and $\delta_{\mathcal{C}}(q, t) = \{p \in \delta(q, t) : \langle t, p \rangle \in \mathcal{C}\}$.

We now explain how to adapt the cut-based approach of Section 3.1 for computing the probabilities in a positive BSCC $\mathcal{C}$ of $\mathcal{P}$. For $\langle s, q \rangle \in \mathcal{C}$ and $t \in S$, let $\Delta_{\mathcal{C}}(\langle s, q \rangle, t) = \Delta(\langle s, q \rangle, t) \cap \mathcal{C}$. A pure cut in $\mathcal{C}$ denotes a set $C \subseteq \mathcal{C}$ such that $\mathrm{Pr}_s^{\mathcal{M}}(\mathcal{P}[C]) = 1$ and $C = \Delta_{\mathcal{C}}(\langle s, q \rangle, z)$ for some $\langle s, q \rangle \in \mathcal{C}$ and some finite word $z \in S^*$ such that $s\, z$ is a cycle in $\mathcal{M}$. (In particular, the last symbol of z is s, and therefore $C \subseteq \{\langle s, p \rangle \in \mathcal{C} : p \in Q\}$.) To compute a pure cut in $\mathcal{C}$,

we pick an arbitrary state $\langle s, q \rangle$ in $\mathcal{C}$ and successively generate path fragments $z_0, z_1, \ldots, z_k \in S^*$ in $\mathcal{M}$ by adding prefixes. More precisely, $z_0 = \varepsilon$ and z_{i+1} has the form yz_i for some $y \in S^+$ such that (1) $s\,y$ is a cycle in $\mathcal{M}$ and (2) there exists a state $p \in Q \setminus \{q\}$ in $\mathcal{U}$ with $\Delta_{\mathcal{C}}(\langle s, p \rangle, z_i) \neq \varnothing$ and $\{\langle s, q \rangle, \langle s, p \rangle\} \subseteq \Delta_{\mathcal{C}}(\langle s, q \rangle, y)$. Each such word y is called an extension of $(\langle s, q \rangle, z_i)$, and $\Delta_{\mathcal{C}}(\langle s, q \rangle, z_{i+1}) = \Delta_{\mathcal{C}}(\langle s, q \rangle, yz_i)$ is a proper superset of $\Delta_{\mathcal{C}}(\langle s, q \rangle, z_i)$. The set $C = \Delta_{\mathcal{C}}(\langle s, q \rangle, z)$ is a pure cut if and only if $(\langle s, q \rangle, z_i)$ has no extension. The search for an extension can be realized efficiently using a technique similar to the one presented in Section 3.1. Thus, after at most $\min\{|\mathcal{C}|, |Q|\}$ iterations, we obtain a pure cut C.

Having computed a pure cut C of $\mathcal{C}$, the values $\mathrm{Pr}_s^{\mathcal{M}}(\mathcal{P}[s, q])$ for $\langle s, q \rangle \in \mathcal{C}$ are then computable as the unique solution of the linear equation system consisting of equations (*) and the additional equation $\sum_{\langle s, q \rangle \in C} \zeta_{s,q} = 1$.

In this way we adapt Theorem 12 to obtain the values $\mathrm{Pr}_s^{\mathcal{M}}(\mathcal{P}[s, q])$ for the states $\langle s, q \rangle$ belonging to some positive BSCC of $\mathcal{P}$. It remains to explain how to adapt the equation system of Theorem 13. Let Q_{BSCC} be the set of BSCC states of $\mathcal{P}$ and $Q_?$ be the states of $\mathcal{P}$ not contained in Q_{BSCC}. For $\langle s, q \rangle \in Q_?$, let $\beta_{s,q} = 0$ if $\Delta(\langle s, q \rangle, t) \cap Q_{BSCC} = \varnothing$ for all $t \in S$. Otherwise:

$$\beta_{s,q} \;=\; \sum_{t \in \mathrm{Post}(s)} \; \sum_{\substack{p \,\in\, \delta(q, t) \text{ s.t.} \\ \langle t, p \rangle \,\in\, Q_{BSCC}}} P(s, t) \cdot \mathrm{Pr}_t^{\mathcal{M}}(\,\mathcal{P}[t, p]\,)$$

Then, the vector $(\mathrm{Pr}_s^{\mathcal{M}}(\mathcal{P}[s, q]))_{\langle s, q \rangle \in Q_?}$ is the unique solution of the linear equation system

$$\zeta_{s,q} \;=\; \sum_{t \in \mathrm{Post}(s)} \; \sum_{\substack{p \,\in\, \delta(q, t) \text{ s.t.} \\ \langle t, p \rangle \,\notin\, Q_{BSCC}}} P(s, t) \cdot \zeta_{t,p} \;\;+\;\; \beta_{s,q} \qquad \text{for } \langle s, q \rangle \in Q_?$$

This completes the proof of Theorem 2.

4 Implementation and Experiments

We have implemented a probabilistic model checking procedure for Markov chains and UBA specifications using the algorithm detailed in Section 3 as an extension to the probabilistic model checker PRISM [31,36]. [5] Our implementation is based on the explicit engine of PRISM, where the Markov chain is represented explicitly. An implementation for the symbolic, MTBDD-based engines of PRISM is planned as future work.

Our implementation supports UBA-based model checking for handling the LTL fragment of PRISM's PCTL*-like specification language as well as direct verification against a path specification given by a UBA provided in the HOA format [3]. For LTL formulas, we rely on external LTL-to-UBA translators. For

---

[5] More details are available at [1]. All experiments were carried out on a computer with two Intel E5-2680 8-core CPUs at 2.70 GHz with 384GB of RAM running Linux.

the purpose of the benchmarks we employ the `ltl2tgba` tool from `SPOT` [18] to generate UBA for a given LTL formula.

For the linear algebra parts of the algorithms, we rely on the `COLT` library [25]. We considered two different variants for the SCC computations as detailed in Appendix F. The first variant (Appendix F.1) relies on `COLT` to perform a QR decomposition of the matrix for the SCC to compute the rank, which allows for deciding the positivity of the SCC. The second approach (Appendix F.2) relies on a variant of the power iteration method for iteratively computing an eigenvector. This method has the benefit that, in addition to deciding the positivity, the computed eigenvector can be directly used to compute the values for a positive SCC, once a cut has been found. (As the proof of Theorem 12 shows: $\Pr(\mathcal{L}_\omega(q)) = \zeta_q^* / \sum_{p \in C} \zeta_p^*$ if ζ^* is an eigenvector of the matrix M for eigenvalue 1.) We have evaluated the performance and scalability of the cut generation algorithm together with both approaches for treating SCCs with selected automata specifications that are challenging for our UBA-based model checking approach (Appendix F.3). As the power iteration method performed better, our benchmark results presented in this section use this method for the SCC handling.

We report here on benchmarks using the bounded retransmission protocol (BRP) case study of the `PRISM` benchmark suite [32]. The model from the benchmark suite covers a single message transmission, retrying for a bounded number of times in case of an error. We have slightly modified the model to allow the transmission of an infinite number of messages by restarting the protocol once a message has been successfully delivered or the bound for retransmissions has been reached. We consider the LTL property

$$\varphi^k = (\neg\texttt{sender\_ok})\,\mathcal{U}\,\big((\texttt{retransmit} \wedge (\neg\texttt{sender\_ok}\;\mathcal{U}^{=k}\;\texttt{sender\_ok})\big),$$

ensuring that k steps before an acknowledgment the message was retransmitted. To remove the effect of selecting specific tools for the LTL to automaton translation (`ltl2tgba` for UBA, the Java-based `PRISM` reimplementation of `ltl2dstar` [29] to obtain a deterministic Rabin automaton (DRA) for the standard `PRISM` approach), we also consider direct model checking against automata specifications. As the language of φ^k is equivalent to the UBA depicted in Figure 1 (on the left) when $a = \texttt{retransmit}\wedge\neg\texttt{sender\_ok}$, $b = \texttt{sender\_ok}$ and $c = \neg\texttt{retransmit}\wedge\neg\texttt{send\_ok}$, we use this automaton and the minimal DBA for the language (this case is denoted by $\mathcal{A}$). We additionally consider the UBA and DBA obtained by replacing the self-loop in the last state with a switch back to the initial state (denoted by $\mathcal{B}$), i.e., roughly applying the ω-operator to $\mathcal{A}$.

Table 1 shows results for selected k (with a timeout 30 minutes), demonstrating that for this case study and properties our UBA-based implementation is generally competitive with the standard approach of `PRISM` relying on deterministic automata. For φ and $\mathcal{A}$, our implementation detects that the UBA has a special shape where all final states have a true-self loop, which allows for skipping the SCC handling. Without this optimization, t_{Cut} and t_{Pos} are in the sub-second range for φ and $\mathcal{A}$ for all considered k. At a certain point, the

Table 1. Statistics for DBA/DRA- and UBA-based model checking of the BRP case study (parameters $N = 16$, $MAX = 128$), a DTMC with 29358 states, depicting the number of states for the automata and the product and the time for model checking (t_{MC}). For φ, t_{MC} includes the translation to the automaton, for $\mathcal{B}$ the time for checking positivity (t_{Pos}) and cut generation (t_{Cut}) are included in t_{MC}. The mark $-$ stands for "not available" or timeout (30 minutes).

	PRISM standard			PRISM UBA				
	DRA	product	t_{MC}	UBA	product	t_{MC}	t_{Pos}	t_{Cut}
$k=4$, φ	118	62,162	0.8 s	6	34,118	0.6 s		
$\mathcal{A}$	33	61,025	0.8 s	6	34,118	0.5 s		
$\mathcal{B}$	33	75,026	0.7 s	6	68,474	1.9 s	1.1 s	<0.1 s
$k=6$, φ	4,596	72,313	3.2 s	8	36,164	0.9 s		
$\mathcal{A}$	129	62,428	1.1 s	8	36,164	0.9 s		
$\mathcal{B}$	129	97,754	1.1 s	8	99,460	3.1 s	1.5 s	<0.1 s
$k=8$, φ	297,204	−	−	10	38,207	0.8 s		
$\mathcal{A}$	513	64,715	1.1 s	10	38,207	0.7 s		
$\mathcal{B}$	513	134,943	1.3 s	10	136,427	4.5 s	2.5 s	<0.1 s
$k=14$, φ	−	−	−	16	44,340	12.8 s	0.0 s	0.0 s
$\mathcal{A}$	32,769	83,845	5.3 s	16	44,340	1.0 s		
$\mathcal{B}$	32,769	444,653	6.0 s	16	246,346	10.2 s	6.5 s	<0.1 s
$k=16$, φ	−	−	−	18	46,390	115.0 s		
$\mathcal{A}$	131,073	−	−	18	46,390	1.0 s		
$\mathcal{B}$	131,073	−	−	18	282,699	12.3 s	8.6 s	<0.1 s
$k=48$, $\mathcal{A}$	−	−	−	50	79,206	1.8 s		
$\mathcal{B}$	−	−	−	50	843,414	88.4 s	71.1 s	<0.1 s

implementation of the standard approach in `PRISM` becomes unsuccessful, either due to timeouts in the DRA construction (φ: $k \geq 10$) or `PRISM` size limitations in the deterministic product construction (φ: $k \geq 8$, $\mathcal{A}/\mathcal{B}$: $k \geq 16$). For $k \geq 18$, `ltl2tgba` was unable to construct the UBA for φ within the given time limit, for $k = 16$, 114.4 s of the 115.0 s were spent on constructing the UBA. As can be seen, using the UBA approach we were able to successfully scale the parameter k beyond 48 when dealing directly with the automata-based specifications ($\mathcal{A}/\mathcal{B}$) and within reasonable time required for model checking.

5   Conclusion

The main contribution of the paper is a polynomial-time algorithm for the quantitative analysis of Markov chains against UBA-specifications, and an implementation thereof. This yields a single exponential-time algorithm for the probabilistic model-checking problem for Markov chains and LTL formulas, and thus an alternative to the double exponential-time classical approach with deterministic automata that has been implemented in `PRISM` and other tools. Other single exponential algorithms for Markov chains and LTL are known, such as the automata-less method of [14] and the approaches with weak alternating automata

[9] or separated UBA [16]. To the best of our knowledge, no implementations of these algorithms are available.[6]

The efficiency of the proposed UBA-based analysis of Markov chains against LTL-specifications crucially depends on sophisticated techniques for the generation of UBA from LTL formulas. Compared to the numerous approaches for the generation of compact nondeterministic or deterministic automata, research on for efficient LTL-to-UBA translators is rare. The tool `Tulip` [34] uses a variant of the LTL-to-NBA algorithm by Gerth et al. [21] for the direct construction of UBA from LTL formulas, while `SPOT`'s LTL-to-UBA generator relies on an adaption of the Couvreur approach [15]. A comparison of NBA versus UBA sizes for LTL benchmark formulas from [20,37,19] (see Appendix F.4) using `SPOT` suggests that requiring unambiguity does not necessarily lead to a major increase in NBA size. An alternative to the direct translation of LTL formulas into UBA are standard LTL-to-NBA translators combined with disambiguation approaches for NBA (e.g. of [27]). However, we are not aware of tool support for these techniques.

Besides the design of efficient LTL-to-UBA translators that exploit the additional flexibility of unambiguous automata compared to deterministic ones, our future work will include a symbolic implementation of our algorithm and more experiments to evaluate the UBA-based approach against the classical approach with deterministic automata (e.g. realized in `PRISM` [31,36] and `IscasMC` [23] and using state-of-the-art generators for deterministic automata such as `Rabinizer`) or other single exponential-time algorithms [14,16,9], addressing the complex interplay between automata sizes, automata generation time, size of the (reachable fragment of the) product and the cost of the analysis algorithms that all influence the overall model checking time.

Acknowledgments. The authors would like to thank Théodore Lopez for his comments on a draft of this paper.

References

1. `http://wwwtcs.inf.tu-dresden.de/ALGI/PUB/CAV16/`. Website with additional material for this paper.
2. André Arnold. Deterministic and non ambiguous rational omega-languages. In *Automata on Infinite Words, Ecole de Printemps d'Informatique Théorique, Le Mont Dore, May 1984*, volume 192 of *Lecture Notes in Computer Science*, pages 18–27. Springer, 1985.
3. Tomáš Babiak, František Blahoudek, Alexandre Duret-Lutz, Joachim Klein, Jan Křetínský, David Müller, David Parker, and Jan Strejček. The Hanoi omega-automata format. In *27th International Conference on Computer Aided Verification (CAV)*, volume 9206 of *Lecture Notes in Computer Science*, pages 479–486. Springer, 2015.

---

[6] The paper [16] reports on experiments with a prototype implementation, but this implementation seems not to be available anymore. Appendix E briefly explains that our algorithm can be seen as a generalization of the approach of [16] with separated UBA. The tool Tulip [34] also has an engine for analysis of Markov chains against UBA-specifications, but it relies on the flawed algorithm of [6].

4. Michael Benedikt, Rastislav Lenhardt, and James Worrell. LTL model checking of interval Markov chains. In *19th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, volume 7795 of *Lecture Notes in Computer Science*, pages 32–46. Springer, 2013.

5. Michael Benedikt, Rastislav Lenhardt, and James Worrell. Model checking Markov chains against unambiguous Büchi automata. *CoRR*, abs/1405.4560, 2014.

6. Michael Benedikt, Ratislav Lenhardt, and James Worrell. Two variable vs. linear temporal logic in model checking and games. *Logical Methods in Computer Science*, 9(2), 2013.

7. Abraham Berman and Robert J. Plemmons. *Nonnegative matrices in the mathematical sciences*. Academic Press, 1979.

8. Nicolas Bousquet and Christof Löding. Equivalence and inclusion problem for strongly unambiguous Büchi automata. In *4th International Conference on Language and Automata Theory and Applications (LATA)*, volume 6031 of *Lecture Notes in Computer Science*, pages 118–129. Springer, 2010.

9. Doron Bustan, Sasha Rubin, and Moshe Y. Vardi. Verifying ω-regular properties of Markov chains. In *16th International Conference on Computer Aided Verification (CAV)*, volume 3114 of *Lecture Notes in Computer Science*, pages 189–201. Springer, 2004.

10. Olivier Carton and Max Michel. Unambiguous Büchi automata. *Theoretical Computer Science*, 297(1–3):37 – 81, 2003.

11. Souymodip Chakraborty and Joost-Pieter Katoen. Parametric LTL on Markov chains. In *8th IFIP TC 1/WG 2.2 International Conference on Theoretical Computer Science*, volume 8705 of *Lecture Notes in Computer Science*, pages 207–221. Springer, 2014.

12. Thomas Colcombet. Forms of determinism for automata (invited talk). In *29th International Symposium on Theoretical Aspects of Computer Science, (STACS)*, volume 14 of *LIPIcs*, pages 1–23. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2012.

13. Thomas Colcombet. Unambiguity in automata theory. In *17th International Workshop on Descriptional Complexity of Formal Systems (DCFS)*, volume 9118 of *Lecture Notes in Computer Science*, pages 3–18. Springer, 2015.

14. Costas Courcoubetis and Mihalis Yannakakis. The complexity of probabilistic verification. *Journal of the ACM*, 42(4):857–907, 1995.

15. Jean-Michel Couvreur. On-the-fly verification of linear temporal logic. In *World Congress on Formal Methods in the Development of Computing Systems (FM)*, volume 1708 of *Lecture Notes in Computer Science*, pages 253–271. Springer, 1999.

16. Jean-Michel Couvreur, Nasser Saheb, and Grégoire Sutre. An optimal automata approach to LTL model checking of probabilistic systems. In *10th International Conference on Logic for Programming Artificial Intelligence and Reasoning (LPAR)*, volume 2850 of *Lecture Notes in Computer Science*, pages 361–375. Springer, 2003.

17. Alexandre Duret-Lutz. Manipulating LTL formulas using Spot 1.0. In *11th International Symposium on Automated Technology for Verification and Analysis (ATVA)*, volume 8172 of *Lecture Notes in Computer Science*, pages 442–445. Springer, 2013.

18. Alexandre Duret-Lutz. LTL translation improvements in Spot 1.0. *International Journal of Critical Computer-Based Systems*, 5(1/2):31–54, 2014.

19. Matthew B. Dwyer, George S. Avrunin, and James C. Corbett. Patterns in property specifications for finite-state verification. In *21th International Conference on Software Engineering (ICSE)*, pages 411–420. ACM, 1999.

20. Kousha Etessami and Gerard Holzmann. Optimizing Büchi automata. In *11th International Conference on Concurrency Theory (CONCUR)*, volume 1877 of *Lecture Notes in Computer Science*, pages 153–167. Springer, 2000.

21. Rob Gerth, Doron Peled, Moshe Y. Vardi, and Pierre Wolper. Simple on-the-fly automatic verification of linear temporal logic. In *Fifteenth IFIP WG6.1 International Symposium on Protocol Specification (PSTV)*, volume 38 of *IFIP Conference Proceedings*, pages 3–18. Chapman & Hall, 1995.

22. Erich Grädel, Wolfgang Thomas, and Thomas Wilke, editors. *Automata, Logics, and Infinite Games: A Guide to Current Research*, volume 2500 of *Lecture Notes in Computer Science*. Springer, 2002.

23. Ernst Moritz Hahn, Yi Li, Sven Schewe, Andrea Turrini, and Lijun Zhang. IS-CASMC: A web-based probabilistic model checker. In *19th International Symposium on Formal Methods (FM)*, volume 8442 of *Lecture Notes in Computer Science*, pages 312–317. Springer, 2014.

24. Roger A. Horn and Charles R. Johnson. *Matrix analysis*. Cambridge University Press, 2nd edition, 2013.

25. Wolfgang Hoschek. The Colt distribution: Open source libraries for high performance scientific and technical computing in Java, 2004.

26. Dimitri Isaak and Christof Löding. Efficient inclusion testing for simple classes of unambiguous ω-automata. *Information Processing Letters*, 112(14-15):578–582, 2012.

27. Detlef Kähler and Thomas Wilke. Complementation, disambiguation, and determinization of Büchi automata unified. In *35th International Colloquium on Automata, Languages and Programming (ICALP)*, volume 5125 of *Lecture Notes in Computer Science*, pages 724–735. Springer, 2008.

28. Jui-Yi Kao, Narad Rampersad, and Jeffrey Shallit. On NFAs where all states are final, initial, or both. *Theoretical Computer Science*, 410(47–49):5010 – 5021, 2009.

29. Joachim Klein and Christel Baier. Experiments with deterministic ω-automata for formulas of linear temporal logic. *Theoretical Computer Science*, 363(2):182–195, 2006.

30. Vidyadhar G. Kulkarni. *Modeling and Analysis of Stochastic Systems*. Chapman & Hall, 1995.

31. Marta Z. Kwiatkowska, Gethin Norman, and David Parker. PRISM 4.0: Verification of probabilistic real-time systems. In *23rd International Conference on Computer Aided Verification (CAV)*, volume 6806 of *Lecture Notes in Computer Science*, pages 585–591. Springer, 2011.

32. Marta Z. Kwiatkowska, Gethin Norman, and David Parker. The PRISM benchmark suite. In *9th International Conference on Quantitative Evaluation of SysTems (QEST)*, pages 203–204. IEEE Computer Society, 2012.

33. Rastislav Lenhardt. Tulip: Model checking probabilistic systems using expectation maximisation algorithm. In *10th International Conference on Quantitative Evaluation of Systems (QEST)*, volume 8054 of *Lecture Notes in Computer Science*, pages 155–159. Springer, 2013.

34. Rastislav Lenhardt. *Two Variable and Linear Temporal Logic in Model Checking and Games*. PhD thesis, University of Oxford, 2013.

35. Andreas Morgenstern. *Symbolic Controller Synthesis for LTL Specifications*. PhD thesis, Technische Universität Kaiserslautern, 2010.

36. The PRISM model checker. `http://www.prismmodelchecker.org/`.

37. Fabio Somenzi and Roderick Bloem. Efficient Büchi automata from LTL formulae. In *12th International Conference on Computer Aided Verification (CAV)*, volume 1855 of *Lecture Notes in Computer Science*, pages 248–263. Springer, 2000.

38. Richard E. Stearns and Harry B. Hunt. On the equivalence and containment problem for unambiguous regular expressions, grammars, and automata. *SIAM Journal on Computing*, pages 598–611, 1985.

39. Moshe Y. Vardi. Automatic verification of probabilistic concurrent finite-state programs. In *26th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 327–338. IEEE Computer Society, 1985.

40. Moshe Y. Vardi and Pierre Wolper. An automata-theoretic approach to automatic program verification (preliminary report). In *1st Symposium on Logic in Computer Science (LICS)*, pages 332–344. IEEE Computer Society Press, 1986.

41. Pierre Wolper, Moshe Y. Vardi, and A. Prasad Sistla. Reasoning about infinite computation paths (extended abstract). In *24th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 185–194. IEEE Computer Society, 1983.

42. Martin Zimmermann. Optimal bounds in parametric LTL games. *Theoretical Computer Science*, 493:30–45, 2013.

Appendix

Section A provides counterexamples that illustrate the flaw in the previously proposed approaches of [6] and [5]. The proof of Lemma 1 is provided in Section B. Section C provides a proof for the PSPACE-completeness of the positivity (probabilistic non-emptiness) and the almost universality problem for NBA. It also presents some observations on strongly connected NBA that will be applied for reasoning about UBA. Detailed proofs for the results of Section 3 can be found in Section D. Section E explains the connection between our approach for (possibly non-separated) UBA and the approach presented in [16] for separated UBA. Section F provides details on our implementation and further benchmark results.

Throughout the appendix, we will use some additional notations, as detailed below. We often use arrow-notation such as $q \xrightarrow{x} p$ to indicate that $p \in \delta(q, x)$ for the transition relation δ of an NBA.

If $K \subseteq \Sigma^*$ is a regular language then $\mathrm{Pr}(K)$ stands shortly for $\mathrm{Pr}(L_K)$ where L_K is the set of infinite words $w \in \Sigma^\omega$ with $\mathit{Pref}(w) \cap K \neq \varnothing$. The syntax of nondeterministic finite automata (NFA) is the same as for NBA. Given an NFA $\mathcal{A} = (Q, \Sigma, \delta, Q_0, F)$, we write $\mathcal{L}_{\mathit{fin}}(\mathcal{A})$ for the accepted language over finite words, i.e., $\mathcal{L}_{\mathit{fin}}(\mathcal{A}) = \{x \in \Sigma^* : \delta(Q_0, x) \cap F \neq \varnothing\}$.

Note that for the special case of the Markov chain $\mathcal{M} = (S, P, \iota)$ with state space $S = \Sigma$ and $P(a, b) = 1/|S|$ for all $a, b \in \Sigma$ and uniform initial distribution (i.e., $\iota(a) = 1/|\Sigma|$ for all $a \in \Sigma$), we have $\mathrm{Pr}^{\mathcal{M}}(L) = \mathrm{Pr}(L)$ for all measurable languages $L \subseteq \Sigma^\omega$.

A   Counterexamples for the UBA-based quantitative analysis of Markov chains proposed in [6] and [5]

In this section, we provide details about the flaw in the approach to the quantitative analysis of Markov chains using unambiguous automata that has been proposed by Benedikt, Lenhardt and Worrell [6,34]. They first present a technique for computing the probability of a Markov chain to satisfy a (co-)safety specification given by an unambiguous finite automaton (UFA) using a linear equation system with variables for pairs of states in the Markov chain and the UFA. This approach can be seen as an elegant variant of the universality test for UFA using difference equations [38]. Furthermore, [6,34] presents an algorithm for the quantitative analysis of Markov chains against ω-regular properties specified by unambiguous Büchi automata (UBA). As observed by the authors, this approach is flawed. An attempt to repair the proposed UBA-based analysis of Markov chains has been presented by the authors in the arXiv document [5]. However, the approach of [5] is flawed again.

Section A.1 provides an example illustrating the faultiness of the algorithms for the analysis of Markov chains against UBA-specifications presented in [5], while Section A.2 does the same for the algorithm presented in [6], which is the same as in [34].

A.1 Counterexample for the approach of [5]

In what follows, let $\mathcal{M} = (S, P, \iota)$ be a Markov chain and $\mathcal{U} = (Q, \Sigma, \delta, Q_0, F)$ an UBA with the alphabet $\Sigma = S$. In this section, we use the following additional notations: If $\mu : S \to [0,1]$ is a distribution then we write $\mathrm{Pr}_\mu^{\mathcal{M}}$ for $\mathrm{Pr}^{\mathcal{M}_\mu}$ where $\mathcal{M}_\mu$ is the Markov chain (S, P, μ). This notation will be used for the distributions $P(s, \cdot)$ where s is a state in $\mathcal{M}$.[7] Thus,

$$\mathrm{Pr}_{P(s,\cdot)}^{\mathcal{M}} \;\; = \;\; \sum_{t \in S} P(s,t) \cdot \mathrm{Pr}_t^{\mathcal{M}} \qquad \text{where} \qquad \mathrm{Pr}_t^{\mathcal{M}} = \mathrm{Pr}_{\mathrm{Dirac}[t]}^{\mathcal{M}}.$$

The task addressed in [5] is to compute $\mathrm{Pr}^{\mathcal{M}}(\mathcal{L}_\omega(\mathcal{U}))$. The algorithm proposed in [5] relies on the mistaken belief that if the Markov chain $\mathcal{M}$ generates words accepted by the given UBA $\mathcal{U}$ with positive probability then the product-graph $\mathcal{M} \otimes \mathcal{U}$ contains *recurrent pairs*. These are pairs $\langle s, q \rangle$ consisting of a state s in $\mathcal{M}$ and a state q of $\mathcal{U}$ such that almost all paths in $\mathcal{M}$ starting in a successor of s can be written as the infinite concatenation of cycles around s that have a run in $\mathcal{U}$ starting and ending in q. (The formal definition of recurrent pairs will be given below.) This claim, however, is wrong as there exist UBA that continuously need a look-ahead for the paths starting in a fixed state of the Markov chain.

Before presenting a counterexample illustrating this phenomenon and the faultiness of [5], we recall some notations of [5]. Given a state $s \in S$ of the Markov chain $\mathcal{M}$ and a state $q \in Q$ of the UBA $\mathcal{U}$, the regular languages $G_{s,q}, H_{s,q} \subseteq S^+$ are defined as follows:

$$G_{s,q} = \left\{ s_1 s_2 \ldots s_k \in S^+ \; : \; s_k = s \text{ and } p \xrightarrow{s_1 s_2 \ldots s_k} q \text{ for some } p \in Q_0 \right\}$$

$$H_{s,q} = \left\{ s_1 s_2 \ldots s_k \in S^+ \; : \; s_k = s \text{ and } q \xrightarrow{s_1 s_2 \ldots s_k} q \right\}$$

A pair $\langle s, q \rangle \in S \times F$ is called *recurrent* if $\mathrm{Pr}_{P(s,\cdot)}^{\mathcal{M}}\big(H_{s,q}^\omega\big) = 1$.

The accepted language of $\mathcal{U}$ can then be written as:

$$\mathcal{L}_\omega(\mathcal{U}) \;\; = \;\; \bigcup_{(s,q) \in S \times F} G_{s,q} \cdot H_{s,q}^\omega$$

The idea of [5] is to reduce the task to compute $\mathrm{Pr}^{\mathcal{M}}(\mathcal{L}_\omega(\mathcal{U}))$ to the task of computing the probability for $\mathcal{M}$ to generate a finite word accepted by an UFA for the language given by the UFA resulting from the union of the regular languages $G_{s,q}$ where $\langle s, q \rangle$ is recurrent. To show the correctness of this approach, [5] claims that for each pair $\langle s, q \rangle \in S \times Q$:

$$\mathrm{Pr}_{P(s,\cdot)}^{\mathcal{M}}\big(H_{s,q}^\omega\big) \;\; \in \;\; \{0,1\}$$

and thus $\mathrm{Pr}_{P(s,\cdot)}^{\mathcal{M}}\big(H_{s,q}^\omega\big) = 0$ for the non-recurrent pairs $\langle s, q \rangle \in S \times F$. To prove this claim, the authors conjecture (in Equation (5) of [5]) that:

$$\mathrm{Pr}_{P(s,\cdot)}^{\mathcal{M}}\big((H_{s,q})^\omega\big) \;\; = \;\; \lim_{n \to \infty} \mathrm{Pr}_{P(s,\cdot)}^{\mathcal{M}}\big((H_{s,q})^n\big) \;\; \overset{(*)}{=} \;\; \lim_{n \to \infty} \mathrm{Pr}_{P(s,\cdot)}^{\mathcal{M}}\big(H_{s,q}\big)^n$$

---

[7] We depart here from the notations of [5] where the notation $\mathrm{Pr}_{\mathcal{M},t}$ has been used as a shortform for $\mathrm{Pr}_{P(t,\cdot)}^{\mathcal{M}}$.

The following example shows that equality $(*)$ is wrong, and recurrent pairs need not to exist, even if $\mathcal{U}$ is universal.

Example. We consider the Markov chain $\mathcal{M} = (S, P, \iota)$ with two states, say $S = \{a, b\}$, and the transition probabilities

$$P(a,a) \;=\; P(a,b) \;=\; P(b,a) \;=\; P(b,b) \;=\; \tfrac{1}{2}$$

and the uniform initial distribution, i.e., $\iota(a) = \iota(b) = \tfrac{1}{2}$ (depicted in Figure 2 on the left). Thus:

$$\Pr^{\mathcal{M}}_{P(a,\cdot)} \;=\; \Pr^{\mathcal{M}}_{P(b,\cdot)} \;=\; \tfrac{1}{2} \cdot \Pr^{\mathcal{M}}_a + \tfrac{1}{2} \cdot \Pr^{\mathcal{M}}_b$$

From state a, the Markov chain $\mathcal{M}$ schedules almost surely an infinite word w starting with a and containing both symbols a and b infinitely often. The analogous statement holds for state b of $\mathcal{M}$.
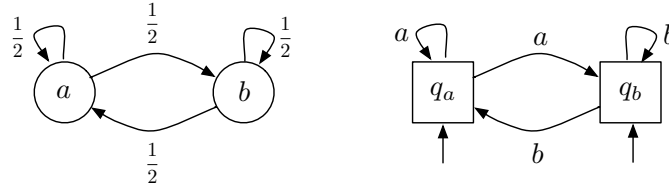

Fig. 2. Markov chain $\mathcal{M}$ (left) and universal UBA $\mathcal{U}$ (right)

Let $\mathcal{U} = (Q, \{a, b\}, \delta, Q, Q)$ be the UBA with state space $Q = \{q_a, q_b\}$ where both states are initial and final and

$$\delta(q_a, a) \;=\; \delta(q_b, b) \;=\; \{q_a, q_b\}$$

while $\delta(q_a, b) = \delta(q_b, a) = \varnothing$ (depicted in Figure 2 on the right). Then, $\mathcal{U}$ is universal as $\mathcal{U}$ can use a one-letter look-ahead to generate an infinite run for each infinite word over $\{a, b\}$. More precisely, for doing so, $\mathcal{U}$ moves to state q_a if the next letter is a and to state q_b is the next letter is b. As both states are final, each word has an accepting run. Thus, $\mathcal{L}_\omega(\mathcal{U}) = \{a, b\}^\omega$ and therefore $\Pr^{\mathcal{M}}(\mathcal{L}_\omega(\mathcal{U})) = 1$.

The language H_{a,q_a} is given by the regular expression $a(a + b)^*$. Thus, for $n \geqslant 2$, the language H^n_{a,q_a} consists of all finite words $x \in \{a, b\}^*$ that start with letter a and contain at least n occurrences of letter a. Likewise, the language H^ω_{a,q_a} consists of all infinite words over $\{a, b\}$ with infinitely many a's and where the first letter is a. Hence:

$$\Pr^{\mathcal{M}}_a\big(H^\omega_{a,q_a}\big) = \Pr^{\mathcal{M}}_a\big(H^n_{a,q_a}\big) = 1$$

$$\Pr^{\mathcal{M}}_b\big(H^\omega_{a,q_a}\big) = \Pr^{\mathcal{M}}_b\big(H^n_{a,q_a}\big) = 0$$

for all $n \in \mathbb{N}$ with $n \geqslant 1$. This yields:

$$\mathrm{Pr}^{\mathcal{M}}_{P(a,\cdot)}\big(H^{\omega}_{a,q_a}\big) \quad = \quad \mathrm{Pr}^{\mathcal{M}}_{P(a,\cdot)}\big(H^{n}_{a,q_a}\big) \quad = \quad \tfrac{1}{2}$$

for all $n \in \mathbb{N}$ with $n \geqslant 1$. On the other hand:

$$\lim_{n\to\infty} \mathrm{Pr}^{\mathcal{M}}_{P(a,\cdot)}(H_{a,q_a})^n \quad = \quad \lim_{n\to\infty} \big(\tfrac{1}{2}\big)^n \quad = \quad 0$$

Thus, equality $(*)$ is wrong. In this example, none of the pairs $\langle a, q_a \rangle$, $\langle a, q_b \rangle$, $\langle b, q_a \rangle$, $\langle b, q_b \rangle$ is recurrent. Note that the languages H_{a,q_b} and H_{b,q_a} are empty and that an analogous calculation yields:

$$\mathrm{Pr}^{\mathcal{M}}_{P(b,\cdot)}\big(H^{\omega}_{b,q_b}\big) \quad = \quad \mathrm{Pr}^{\mathcal{M}}_{P(b,\cdot)}\big(H^{n}_{b,q_b}\big) \quad = \quad \tfrac{1}{2}$$

and $\lim\limits_{n\to\infty} \mathrm{Pr}^{\mathcal{M}}_{P(b,\cdot)}(H_{b,q_b})^n = 0$.

A.2 Counterexample for the approach of [6]

A similar counterexample can be constructed for Lemma 7.1 of [6] (p.22), i.e., the original proposal for using UBA for model checking of DTMCs.

In [6], the Büchi automata are state-labeled, usually with an alphabet over some atomic propositions. For presentational simplicity, we use here a fixed alphabet $\{a, b, c, d\}$, corresponding to the states of the Markov chain, omitting the atomic proposition based labeling functions. The accepting condition in [6] is a generalized Büchi condition, i.e., of set of Büchi conditions. A run is accepting, if it satisfy every Büchi condition in the generalized Büchi condition.

The paper [6] assumes a product graph out of a Markov chain $\mathcal{M}$ and a (generalized) UBA $\mathcal{U}$. The nodes are pairs of Markov chain states and UBA states with agreeing labels. There is an edge between two nodes, if and only if there is an edge between the two corresponding Markov chain states and an edge between the two corresponding UBA states. It defines an SCC $\mathcal{C}$ to be accepting if

(i) for every acceptance set $F \in \mathcal{F}$ there exists a node $\langle s, q_F \rangle$ with $q_F \in F$, and
(ii) for every node $\langle s, p \rangle$ and every transition $s \longrightarrow t$ in the Markov chain $\mathcal{M}$ there exists a transition $p \longrightarrow q$ in $\mathcal{U}$ such that $\langle t, q \rangle$ is contained in $\mathcal{C}$.

Example. Our example will give a Markov chain and a UBA, for which $\mathrm{Pr}^{\mathcal{M}}(\mathcal{L}(\mathcal{U})) = 1$, holds, but the product will not contain an accepting SCC according to the definition of [6]. Consider the (state-labeled) UBA $\mathcal{U}$ and the Markov chain $\mathcal{M}$ of Figure 3.

The UBA accepts all words of the form

$$\big((dab) + (dac)\big)^{\omega}$$

and consequently $\mathrm{Pr}^{\mathcal{M}}(\mathcal{L}(\mathcal{U})) = 1$.

The product graph $\mathcal{M} \otimes \mathcal{U}$ that arises from the construction in the proof of Lemma 7.1 of [6] is depicted in Figure 4.

The product graph is strongly connected, but it is not accepting as condition (ii) is violated: Consider the vertex (a, q_a) in the product graph. There exists a
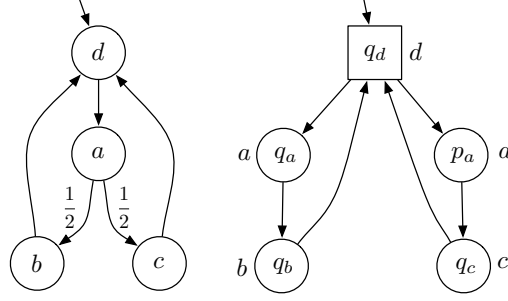
Fig. 3. Markov chain $\mathcal{M}$ (left) and UBA $\mathcal{U}$ (right, state labels from alphabet $\{a, b, c, d\}$)
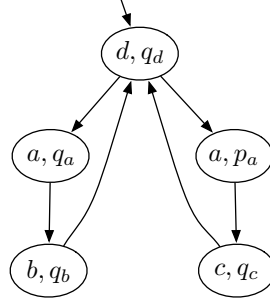

Fig. 4. Product graph according to [6]

transition $a \to c$ in the Markov chain, however there is no successor (c, t) in the SCC of the product graph, with t being a successor of q_a in the UBA (c can not be consumed from the state q_a in the UBA). As the (only) SCC in the product is not accepting, all vertices in the product graph are assigned value 0 in the linear equation system, yielding that $\mathrm{Pr}^{\mathcal{M}}(\mathcal{L}(\mathcal{U})) = 0$. However, as stated above, $\mathrm{Pr}^{\mathcal{M}}(\mathcal{L}(\mathcal{U})) = 1$.

B Almost-sure residuals of positive languages

We now provide the proof for Lemma 1. We first recall the statement:

Lemma 16 (See Lemma 1). *If $L \subseteq \Sigma^\omega$ is ω-regular and $\mathrm{Pr}(L) > 0$ then there exists $x \in \Sigma^*$ such that $\mathrm{Pr}\big\{\, w \in \Sigma^\omega \,:\, xw \in L \,\big\} = 1$.*

Proof. Pick a deterministic ω-automaton $\mathcal{D} = (Q, \Sigma, \delta, q_{init}, Acc)$ for L, for instance, with a Rabin acceptance condition. W.l.o.g. all states are reachable from q_{init} and $\mathcal{D}$ is complete. Let $\mathcal{M}_{\mathcal{D}} = (Q, P)$ be the transition-labeled Markov chain resulting from $\mathcal{D}$ by turning all branchings in $\mathcal{D}$ into uniform probabilistic choices, i.e., for each state q and each letter a, $P(q, a, q') = 1/|\Sigma|$ if $\delta(q, a) = q'$ and $P(q, a, q') = 0$ otherwise. Clearly, the underlying graph of $\mathcal{D}$ and $\mathcal{M}_D$ is the

same. If $\mathcal{C}$ is a bottom strongly connected component (BSCC) of $\mathcal{D}$ resp. $\mathcal{M}_D$ then $\mathcal{C}$ is said to satisfy $\mathcal{D}$'s acceptance condition Acc, denoted $\mathcal{C} \models Acc$, iff all infinite paths π with $inf(\pi) = \mathcal{C}$ meet the condition imposed by Acc, where $inf(\pi)$ denotes the set of states that appear infinitely often in π. For example, if Acc is a Rabin condition, say

$$Acc = \bigvee_{1 \leqslant i \leqslant k} (\Diamond\square A_i \wedge \square\Diamond B_i) \qquad \text{where} \qquad A_i, B_i \subseteq Q,$$

and $\mathcal{C} \subseteq Q$ a BSCC, then $\mathcal{C} \models Acc$ iff there is at least one Rabin pair (A_i, B_i) in Acc with $\mathcal{C} \subseteq A_i$ and $\mathcal{C} \cap B_i \neq \varnothing$. As almost all paths in $\mathcal{M}_\mathcal{D}$ eventually enter a BSCC and visit all its states infinitely often, we get:

$$\Pr(L) > 0 \quad \text{iff} \quad \Pr^{\mathcal{M}_\mathcal{D}}(Acc) > 0$$
$$\text{iff} \quad \mathcal{D} \text{ has at least one BSCC } \mathcal{C} \text{ with } \mathcal{C} \models Acc$$

In this case and if x is a finite word such that $\delta(q_{init}, x) \cap \mathcal{C} \neq \varnothing$, then $\Pr\{w \in \Sigma^\omega : xw \in L\} = 1$. ∎

C Positivity and almost universality for NBA

For the remainder of Section C, we consider the case where $\mathcal{A} = (Q, \Sigma, \delta, Q_0, F)$ is a strongly connected NBA with at least one initial and one final state. As before, we briefly write $\mathcal{L}_\omega(q)$ instead of $\mathcal{L}_\omega(\mathcal{A}[q])$ where $\mathcal{A}[q]$ is the NBA $(Q, \Sigma, \delta, q, F)$. The assumption yields that $\mathcal{L}_\omega(\mathcal{A}[q])$ is nonempty for all states q.

Fact 17 *Suppose $\mathcal{A}$ is a strongly connected NBA. Then, the following statements are equivalent:*

(1) $\Pr(\mathcal{L}_\omega(\mathcal{A})) > 0$
(2) $\Pr(\mathcal{L}_\omega(q)) > 0$ *for some state q*
(3) $\Pr(\mathcal{L}_\omega(p)) > 0$ *for all states p*

Proof. The implications $(1) \implies (2)$ and $(3) \implies (1)$ are trivial. We now show that $(2) \implies (3)$. Since $\mathcal{A}$ is strongly connected, there exists a finite word x with $p \xrightarrow{x} q$, i.e., $q \in \delta(p, x)$. But then

$$\Pr(\mathcal{L}_\omega(p)) \quad \geqslant \quad \frac{1}{2^{|x|}} \cdot \Pr(\mathcal{L}_\omega(q)) \quad > \quad 0$$

Note that $1/2^{|x|}$ is the probability for (the cylinder set spanned by) the finite word x. ∎

Lemma 18 (see Lemma 4). *For each strongly connected NBA $\mathcal{A}$ with at least one final state, we have:*

$$\Pr(\mathcal{L}_\omega(\mathcal{A})) \quad = \quad 1 - \Pr\{w \in \Sigma^\omega : w \text{ has a finite prefix } x \text{ with } \delta(Q_0, x) = \varnothing\}$$

To prove Lemma 18, we show that the language consisting of all words $w \in \Sigma^\omega \setminus \mathcal{L}_\omega(\mathcal{A})$ such that $\delta(Q_0, x) \neq \varnothing$ for all $x \in \mathit{Pref}(w)$ is a null set. If $\mathcal{L}_\omega(\mathcal{A})$ has positive measure, this statement is a simple consequence of Lemma 1 (see Lemma 19 below). The general case will be shown in Lemma 25 using known results for the positive probabilistic model-checking problem.

Lemma 19. *Suppose $\mathcal{A}$ is strongly connected and $\mathrm{Pr}(\mathcal{L}_\omega(\mathcal{A})) > 0$. Let L denote the set of infinite words $w \in \Sigma^\omega \setminus \mathcal{L}_\omega(\mathcal{A})$ such that $\delta(Q_0, x) \neq \varnothing$ for all $x \in \mathit{Pref}(w)$. Then, $\mathrm{Pr}(L) = 0$.*

Proof. Suppose by contradiction that $\mathrm{Pr}(L)$ is positive. Obviously, L is ω-regular. Lemma 1 yields the existence of a finite word x such that

$$\mathrm{Pr}\big\{\, v \in \Sigma^\omega \;:\; xv \in L \,\big\} \;\; = \;\; 1$$

Let $R = \delta(Q_0, x)$. Then, R is nonempty and $\mathrm{Pr}(\mathcal{L}_\omega(\mathcal{A}[R])) = 0$, i.e., $\mathrm{Pr}(\mathcal{L}_\omega(q)) = 0$ for all states $q \in R$. This is impossible by Fact 17. ∎

To prove the analogous result for the general case (possibly $\mathrm{Pr}(\mathcal{L}_\omega(\mathcal{A})) = 0$), we rely on results by Courcoubetis and Yannakakis [14] for the positive probabilistic model-checking problem. These results rephrased for our purposes yield the following. Let $\mathcal{A}_{\mathrm{det}}$ denote the standard powerset construction of $\mathcal{A}$. That is, the states of $\mathcal{A}_{\mathrm{det}}$ are the subsets of Q and the transitions in $\mathcal{A}_{\mathrm{det}}$ are given by $R \xrightarrow{a} R'$ iff $R' = \delta(R, a)$. The initial state of $\mathcal{A}_{\mathrm{det}}$ is Q_0. $\mathcal{A}_{\mathrm{det}}$ is viewed here just as a pointed labeled graph rather than an automaton over words.

Recall that $\mathcal{A}$ might be incomplete. Thus, $\varnothing$ is a trap state of $\mathcal{A}_{\mathrm{det}}$ that is reached via the a-transition from any state $R \subseteq Q$ where $\delta(R, a)$ is empty. Hence, $\{\varnothing\}$ is a BSCC of $\mathcal{A}_{\mathrm{det}}$ that might or might not be reachable from Q_0. We refer to $\{\varnothing\}$ as the *trap-BSCC* of $\mathcal{A}_{\mathrm{det}}$. All other BSCCs of $\mathcal{A}_{\mathrm{det}}$ are called *non-trap*.

A state $q \in Q$ of $\mathcal{A}$ is said to be *recurrent* if there is some BSCC $\mathcal{C}$ of $\mathcal{A}_{\mathrm{det}}$ that contains a state $R \subseteq Q$ of $\mathcal{A}_{\mathrm{det}}$ with $q \in R$ and that is reachable from the singleton $\{q\}$ viewed as a state of $\mathcal{A}_{\mathrm{det}}$. That is, q is recurrent iff there exists a finite word x such that $q \xrightarrow{x} q$ and the set $\delta(q, x)$ belongs to a BSCC of $\mathcal{A}_{\mathrm{det}}$.

Fact 20 (Proposition 4.1.4 in [14]) *For each NBA $\mathcal{A}$ (not necessarily strongly connected):*

$$\mathrm{Pr}\big(\mathcal{L}_\omega(\mathcal{A})\big) > 0 \quad \textit{iff} \quad \begin{cases} \textit{there exists a finite word } x \in \Sigma^* \textit{ such that} \\ \delta(Q_0, x) \cap F \textit{ contains a recurrent state} \end{cases}$$

Example 21. We consider the strongly connected NBA shown in Figure 5. Then, $\mathcal{A}_{\mathrm{det}}$ has two BSCCs, namely the trap-BSCC $\{\varnothing\}$ and the non-trap BSCC $\{Q\}$. The singletons $\{q_a\}$ and $\{q_b\}$, viewed as states of $\mathcal{A}_{\mathrm{det}}$, can reach $\{Q\}$. Hence, both states q_a and q_b are recurrent. To verify the statement of Fact 20, we observe that $\mathrm{Pr}(\mathcal{L}_\omega(\mathcal{A})) = 1/2 > 0$ and $\delta(\{q_a\}, a) \cap F = \{q_b\}$ contains a recurrent state. ∎
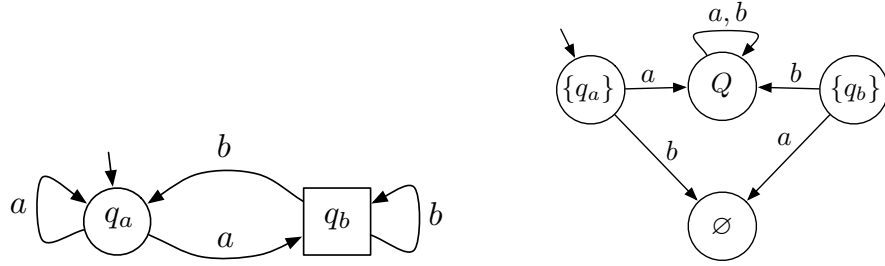
Fig. 5. Automata from Example 21, NBA $\mathcal{A}$ (left) and corresponding $\mathcal{A}_{\mathrm{det}}$ (right)

Suppose now that $\mathcal{A}$ is strongly connected. Then, for each non-trap BSCC $\mathcal{C}$ of $\mathcal{A}_{\mathrm{det}}$ (i.e., $\mathcal{C} \neq \{\varnothing\}$) and each state p of $\mathcal{A}$ there exists some $R \subseteq Q$ with $p \in R \in \mathcal{C}$. Moreover, whenever $R \in \mathcal{C}$ and $x \in \Sigma^*$ then $\delta(R, x) \in \mathcal{C}$.

Lemma 22. *If $\mathcal{A}$ is strongly connected and $\mathcal{A}_{\mathrm{det}}$ has a non-trap BSCC that is reachable from some singleton $\{q\}$ then all states $p \in Q$ are recurrent.*

Proof. Let $\mathcal{C}$ be a non-trap BSCC of $\mathcal{A}_{\mathrm{det}}$ that is reachable from $\{q\}$ and let p be a state of $\mathcal{A}$. We pick finite words $x, y \in \Sigma^*$ such that $q \in \delta(p, x)$ and $\delta(q, y) \in \mathcal{C}$. Then, for all finite words z, $\delta(q, yz) \in \mathcal{C}$ and therefore:

$$\varnothing \quad \neq \quad \delta(q, yz) \quad \subseteq \quad \delta(p, xyz)$$

Hence, $\delta(p, xyz) = \delta(\delta(p, xy), z)$ is nonempty for all words z. Thus, there is a non-trap BSCC $\mathcal{C}'$ (possibly different from $\mathcal{C}$) that is reachable from p via some finite word of the form xyz. As stated above, $\mathcal{C}'$ contains some $R \subseteq Q$ with $p \in R$. Hence, p is recurrent. ∎

Corollary 23 (Probabilistic emptiness of strongly connected NBA). *Let $\mathcal{A}$ be a strongly connected NBA with at least one final state. Then, the following statements are equivalent:*

(1) $\Pr\big(\mathcal{L}_\omega(\mathcal{A})\big) = 0$
(2) $\mathcal{A}_{\mathrm{det}}$ *has no non-trap BSCC that is reachable from some singleton $\{q\}$*
(3) *there is no recurrent state in $\mathcal{A}$*

Example 24. We consider the strongly connected NBA shown in Figure 6. Then, $\mathcal{A}_{\mathrm{det}}$ has a non-trap BSCC consisting of the states $\{p_1, p_2\}$ and Q that is not accessible from any singleton. However, there is another non-trap BSCC in $\mathcal{A}_{\mathrm{det}}$ consisting of the three states $\{p_1\}$, $\{p_2\}$ and $\{q_0, p_2\}$. Indeed, $\mathcal{A}$ accepts almost all words starting with letter c and therefore $\Pr(\mathcal{L}_\omega(\mathcal{A})) = 1/3$. ∎

We are now ready to complete the proof of Lemma 4 by proving the following lemma:
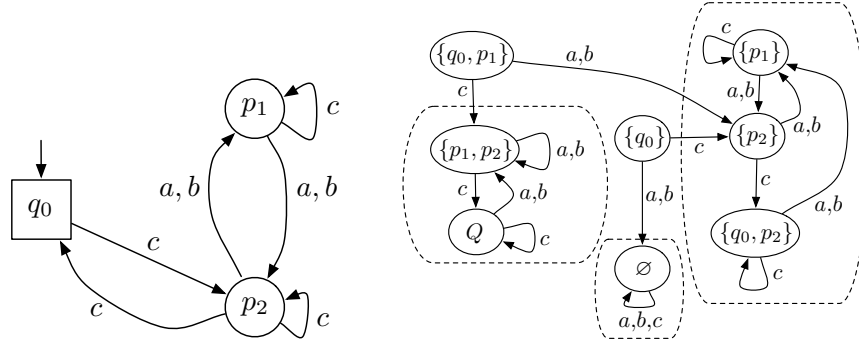
Fig. 6. Automata from Example 24, NBA $\mathcal{A}$ (left) and $\mathcal{A}_{\mathrm{det}}$ (right, with BSCCs)

Lemma 25. *Suppose $\mathcal{A}$ is strongly connected with at least one final state. Let L be the set of infinite words $w \in \Sigma^\omega \setminus \mathcal{L}_\omega(\mathcal{A})$ such that $\delta(Q_0, x) \neq \varnothing$ for all $x \in Pref(w)$. Then, $\mathrm{Pr}(L) = 0$.*

Proof. We consider first the case where $\mathcal{A}$ has a single initial state, say $Q_0 = \{q_0\}$. Suppose by contradiction that $\mathrm{Pr}(L)$ is positive. Then, there is some finite word z such that $zv \in L$ for almost all words $v \in \Sigma^\omega$. Let $R = \delta(q_0, z)$. By definition of L, the set R is nonempty and $\delta(R, x) \neq \varnothing$ for all finite words x. Hence, the state $\varnothing$ is not reachable from R in $\mathcal{A}_{\mathrm{det}}$. Therefore, there is a non-trap BSCC of $\mathcal{A}_{\mathrm{det}}$ that is reachable from the singleton $\{q_0\}$. Hence, $\mathrm{Pr}(\mathcal{L}_\omega(\mathcal{A})) > 0$ by Corollary 23. But then $\mathrm{Pr}(L) = 0$ by Lemma 19. Contradiction.

The argument for the general case is as follows. Suppose by contradiction that L has positive measure. We consider the labeled Markov chain $\mathcal{M} = (2^Q, P, \mathrm{Dirac}[Q_0])$ that arises from the deterministic automaton $\mathcal{A}_{\mathrm{det}}$ with initial state Q_0 by attaching uniform distributions. That is, if $R, R' \subseteq Q$ and $a \in \Sigma$ then $P(R, a, R') = 1/|\Sigma|$ if $R' = \delta(R, a)$ and $P(R, a, R') = 0$ otherwise. For almost all words w in L, the corresponding path π_w in $\mathcal{M}$ eventually visits some BSCC $\mathcal{C}$ of $\mathcal{M}$ resp. $\mathcal{A}_{\mathrm{det}}$ and visits all its states infinitely often. By assumption $\mathcal{C}$ is non-trap.

The goal is to show that a non-trap BSCC is accessible from some singleton. Let L' denote the set of all words $w = a_1\, a_2\, a_3 \ldots \in L$ such that π_w eventually enters some BSCC of $\mathcal{M}$ and visits all its states infinitely often. Then, $\mathrm{Pr}(L') = \mathrm{Pr}(L)$. If $w = a_1\, a_2\, a_3 \ldots \in L'$ and $\pi_w = R_0\, R_1\, R_2 \ldots$ then $R_0 = Q_0$ and $R_n = \delta(Q_0, a_1 \ldots a_n)$. By König's Lemma there is an infinite run $q_0\, q_1\, q_2 \ldots$ for w in $\mathcal{A}$ such that $q_i \in R_i$ for all $i \in \mathbb{N}$. We write $Runs(w)$ to denote the set of all these runs. Pick some run $\rho = q_0\, q_1\, q_2 \ldots \in Runs(w)$ and define $U_0 = \{q_0\}$ and $U_i = \delta(q_0, a_1\, a_2 \ldots a_i) = \delta(U_{i-1}, a_i)$ for $i \geqslant 1$. Clearly, $q_i \in U_i \subseteq R_i$ for all indices i and $\pi_\rho = U_0\, U_1\, U_2 \ldots$ is a path in $\mathcal{M}$ and the unique run for w in $\mathcal{A}_{\mathrm{det}}$ starting in $\{q_0\}$.

Consider the set $\mathfrak{U}$ of all sets $U \subseteq Q$ such that $U \in inf(\pi_\rho)$ for some word $w \in L'$ and some $\rho \in Runs(w)$. (The notation $inf(\cdot)$ is used to denote the set of

elements that appear infinitely often in $(\cdot)$.) We now show that the subgraph of $\mathcal{A}_{\mathrm{det}}$ consisting of the nodes $U \in \mathfrak{U}$ contains a BSCC. For each subset $\mathcal{V}$ of $\mathfrak{U}$ and each state $q \in Q$, let

$$L_{q,\mathcal{V}} \;\; = \;\; \big\{\, w \in L' \;:\; \exists \rho \in \mathit{Runs}(w) \text{ s.t. } q = \mathit{first}(\pi_\rho) \wedge \mathit{inf}(\pi_\rho) = \mathcal{V} \,\big\}$$

Then, L' is the union of all sets $L_{q,\mathcal{V}}$ with $(q, \mathcal{V}) \in Q_0 \times \mathfrak{U}$. As $\Pr(L') = \Pr(L) > 0$ and $Q_0 \times \mathfrak{U}$ is finite, there is some pair $(q, \mathcal{V}) \subseteq Q_0 \times \mathfrak{U}$ with $\Pr(L_{q,\mathcal{V}}) > 0$. Clearly, $L_{q,\mathcal{V}}$ is ω-regular. Hence, there is some finite word z such that

$$\Pr\big\{\, v \in \Sigma^\omega \;:\; zv \in L_{q,\mathcal{V}} \,\big\} \;\; = \;\; 1$$

Let $R = \delta(q, z)$. We now regard the fragment of $\mathcal{A}_{\mathrm{det}}$ that is reachable from R. Let $\mathcal{M}[R]$ be the corresponding Markov chain (i.e., the sub Markov chain of $\mathcal{M}$ with initial state R). Since $zv \in L_{q,\mathcal{V}}$ for almost all words $v \in \Sigma^\omega$, $\mathit{inf}(\pi) = \mathcal{V}$ for almost all paths in $\mathcal{M}[R]$. But then $\mathcal{V}$ constitutes a non-trap BSCC of $\mathcal{M}[R]$, and therefore of $\mathcal{M}$ and $\mathcal{A}_{\mathrm{det}}$. Since $\mathcal{V}$ is reachable from $\{q\}$ in $\mathcal{A}_{\mathrm{det}}$, we obtain $\Pr(\mathcal{L}_\omega(\mathcal{A})) > 0$ by Corollary 23. Lemma 19 yields $\Pr(L) = 0$, which contradicts the assumption $\Pr(L) > 0$. ∎

Remark 26. Clearly, $\Pr(\mathcal{L}_\omega(\mathcal{A}))$ depends on Q_0 as there might be state-letter pairs (q, a) where $\delta(q, a)$ is empty. However, Lemma 4 implies that if $\mathcal{A}$ is strongly connected with at least one final state then $\Pr(\mathcal{L}_\omega(\mathcal{A}))$ does not depend on F. ∎

Theorem 27. *The positivity and the almost universality problem for strongly connected NBA are PSPACE-complete.*

Proof. Membership to PSPACE follows from the results of [39,14]. PSPACE-hardness of the positivity and almost universality problem for strongly connected NBA can be established using a polynomial reduction from the universality problem for nondeterministic finite automata (NFA) where all states are final. The latter problem is known to be PSPACE-complete [28].

Let $\mathcal{B} = (Q, \Gamma, \delta_\mathcal{B}, Q_0, Q)$ be an NFA. We can safely assume that Q_0 is nonempty and all states are reachable from Q_0. We define an NBA $\mathcal{A} = (Q, \Sigma, \delta_\mathcal{A}, Q_0, Q)$ over the alphabet $\Sigma = \Gamma \cup \{\#\}$ as follows. If $q \in Q$ and $a \in \Gamma$ then $\delta_\mathcal{A}(q, a) = \delta_\mathcal{B}(q, a)$ and $\delta_\mathcal{A}(q, \#) = Q_0$. Clearly, $\mathcal{A}$ is strongly connected. Furthermore, $\delta_\mathcal{A}(R, \#) = Q_0$ for all nonempty subsets R of Q and $\delta_\mathcal{A}(q, x\#) = Q_0$ for all states q and all words $x \in \Sigma^*$ where $\delta_\mathcal{A}(q, x)$ is nonempty.

$$\begin{aligned} \mathcal{B} \text{ is universal} \quad &\text{iff} \quad \mathcal{L}_{\mathit{fin}}(\mathcal{B}) = \Gamma^* \\ &\text{iff} \quad \delta_\mathcal{B}(Q_0, y) \neq \varnothing \text{ for all } y \in \Gamma^* \\ &\text{iff} \quad \delta_\mathcal{A}(Q_0, x) \neq \varnothing \text{ for all } x \in \Sigma^* \end{aligned}$$

By Lemma 4, $\mathcal{B}$ is universal iff $\Pr(\mathcal{L}_\omega(\mathcal{A})) = 1$. This yields the PSPACE-hardness of the almost universality problem for strongly connected NBA. Moreover, all singletons $\{q\}$ can reach Q_0, and if there is a non-trap BSCC $\mathcal{C}$ of $\mathcal{A}_{\mathrm{det}}$ then $Q_0 \in \mathcal{C}$. Using Corollary 23, we obtain:

$\mathcal{B}$ is universal   iff Q_0 is contained in some non-trap BSCC of $\mathcal{A}_{\mathrm{det}}$

iff $\Pr(\mathcal{L}_\omega(\mathcal{A})) > 0$

This yields the PSPACE-hardness of the positivity problem for strongly connected NBA. ∎

D   Proofs for Section 3

D.1   Some technical statements about UBA

In what follows, let $\mathcal{U} = (Q, \Sigma, \delta, Q_0, F)$ be a UBA such that all states are reachable from some initial state and $\mathcal{L}_\omega(q) \neq \varnothing$ for all states q.

We start with some simple observations. If q_1, q_2 are states in $\mathcal{U}$ with $q_1 \neq q_2$ such that $\{q_1, q_2\} \subseteq \delta(Q_0, x)$ for some finite word x then $\mathcal{L}_\omega(q_1) \cap \mathcal{L}_\omega(q_2) = \varnothing$. In particular, the languages of the initial states are pairwise disjoint, and therefore:

$$\Pr(\mathcal{L}_\omega(\mathcal{U})) \;\; = \;\; \sum_{q \in Q_0} \Pr(\mathcal{L}_\omega(q)).$$

The following simple fact will be used at several places:

Fact 28 *If $\mathcal{U}$ is a UBA then for each state $p \in Q$, each nonempty subset R of Q of the form $R = \delta(Q_0, y)$ for some word $y \in \Sigma^*$ and each finite word $x = a_1 \, a_2 \ldots a_n \in \Sigma^*$ there exists at most one state $q \in R$ and at most one run*

$$q = q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} \; \ldots \; \xrightarrow{a_n} q_n = p$$

for x starting in q and ending in p. In particular, the NFA $(Q, \Sigma, \delta, \delta(Q_0, y), \{p\})$ is unambiguous for each $y \in \Sigma^*$ and each state $p \in Q$.

Proof. If there would be two runs $q_0 \xrightarrow{y} q \xrightarrow{x} p$ and $q_0' \xrightarrow{y} q' \xrightarrow{x} p$ where $q, q' \in R$ and $q_0, q_0' \in Q_0$ then each word yxw with $w \in \mathcal{L}_\omega(p)$ would have two accepting runs. This is impossible by the unambiguity of $\mathcal{U}$ and the assumption that $\mathcal{L}_\omega(p)$ is nonempty for all states $p \in Q$. ∎

Fact 28 will often be used in form of the statement that, for all states $q, p \in Q$ and all finite words x, there is at most one run for x from q to p.

We now suppose that $\mathcal{U}$ is a strongly connected UBA. Note that $\mathcal{L}_\omega(p) \neq \varnothing$ and $\Pr(\mathcal{L}_\omega(p)) = 0$ is possible. However, in this case $\Pr(\mathcal{L}_\omega(q)) = 0$ for all states q. The following theorem states that for strongly connected UBA $\mathcal{U}$, the accepting runs of almost all words in $\mathcal{L}_\omega(\mathcal{U})$ visit each state of $\mathcal{U}$ infinitely often. Although irrelevant for the soundness of our algorithm, we find that it reveals an interesting structural property of strongly connected UBA.

Theorem 29 (Measure of strongly connected UBA). *If $\mathcal{U}$ is a strongly connected UBA with at least one final state then:*

$$\Pr\big\{\, w \in \mathcal{L}_\omega(\mathcal{U}) \,:\, \mathit{inf}(\mathit{accrun}(w)) = Q \,\big\} \;\; = \;\; \Pr\big(\, \mathcal{L}_\omega(\mathcal{U}) \,\big)$$

In Theorem 29 we use the following notations. For $w \in \mathcal{L}_\omega(\mathcal{U})$, we write $accrun(w)$ to denote the unique accepting run for w in $\mathcal{U}$. For $q_0 \, q_1 \, q_2 \ldots \in Q^\omega$, the set $inf(q_0 \, q_1 \, q_2 \ldots)$ denotes the set of all states $q \in Q$ such that $q = q_i$ for infinitely many indices i. Thus, if $w \in \mathcal{L}_\omega(\mathcal{U})$ then $inf(accrun(w))$ collects all states $q \in Q$ that appear infinitely often in the accepting run for w.

Proof. To prove Theorem 29, we can rely on the following facts that hold for all events (measurable sets) L_i, L in each probability space:

- $\Pr(L_1) = \Pr(L_2) = 1$ iff $\Pr(L_1 \cap L_2) = 1$
  Hence, if $L_1, \ldots, L_n \subseteq L$ and $\Pr(L_1) = \ldots = \Pr(L_n) = \Pr(L)$ then $\Pr(L_1 \cap \ldots \cap L_n) = \Pr(L)$ as $\Pr(L_i \mid L) = 1$ for all $i = 1, \ldots, n$ implies $\Pr(L_1 \cap \ldots \cap L_n \mid L) = 1$.
- if $(L_n)_{n \in \mathbb{N}}$ is a countable family of measurable sets with $L_0 \supseteq L_1 \supseteq L_2 \supseteq \ldots$ then
$$\Pr(L) = \lim_{n \to \infty} \Pr(L_n) \qquad \text{where} \qquad L = \bigcap_{n \in \mathbb{N}} L_n$$

Hence, for the proof of Theorem 29 it suffices to show that the accepting runs for almost all words in $\mathcal{L}_\omega(\mathcal{U})$ contain a fixed state q that appears infinitely often. Thus, the goal is to show:

$$\Pr\{\, w \in \mathcal{L}_\omega(\mathcal{U}) \,:\, q \in inf(accrun(w))\,\} \;=\; \Pr\big(\mathcal{L}_\omega(\mathcal{U})\big)$$

The claim is obvious if $\Pr(\mathcal{L}_\omega(\mathcal{U})) = 0$. Suppose now that $\Pr(\mathcal{L}_\omega(\mathcal{U})) > 0$. Hence, $\Pr(\mathcal{L}_\omega(p)) > 0$ for all states p (Fact 17). We first show that the accepting runs for almost all words in $\mathcal{L}_\omega(\mathcal{U})$ eventually visit q.

Claim: $\Pr\{\, w \in \mathcal{L}_\omega(\mathcal{U}) \,:\, accrun(w) \models \Diamond q \,\} \;=\; \Pr\big(\mathcal{L}_\omega(\mathcal{U})\big)$
    Proof of the claim: Suppose by contradiction that there is some state $q \in Q$ such that the set of words $w \in \mathcal{L}_\omega(\mathcal{U})$ whose run does not visit q has positive measure, i.e.,

$$\Pr\{\, w \in \mathcal{L}_\omega(\mathcal{U}) \,:\, accrun(w) \not\models \Diamond q \,\} \;>\; 0$$

Lemma 1 yields the existence of a finite word $x \in \Sigma^*$ such that:

$$\Pr\{\, w \in \Sigma^\omega \,:\, xw \in \mathcal{L}_\omega(\mathcal{U}),\; accrun(xw) \not\models \Diamond q \,\} \;=\; 1$$

In particular, $\delta(Q_0, x)$ is nonempty. Pick some state $p \in \delta(Q_0, x)$, say $p \in \delta(q_0, x)$ where $q_0 \in Q_0$. As $\mathcal{U}$ is strongly connected, there is some nonempty finite word y with $q \in \delta(p, y)$. Let

$$L \;=\; \{\, xyv \,:\, v \in \mathcal{L}_\omega(q) \,\} \quad \text{and} \quad L' \;=\; \{\, yv \,:\, v \in \mathcal{L}_\omega(q) \,\}.$$

As $\Pr(\mathcal{L}_\omega(q)) > 0$, we get:

$$\Pr(L') \;\geqslant\; \Pr(L) \;=\; \frac{1}{2^n} \cdot \Pr(\mathcal{L}_\omega(q)) \;>\; 0$$

where $n = |x| + |y|$. The words $xyv \in L$ have accepting runs starting with the prefix:

$$q_0 \xrightarrow{x} p \xrightarrow{y} q$$

In particular:

$$L' \ \subseteq \ \{\ w \in \Sigma^\omega \ : \ xw \in \mathcal{L}_\omega(\mathcal{U}), \ accrun(xw) \models \Diamond q \ \}$$

and therefore:

$$\Pr\{\ w \in \Sigma^\omega \ : \ xw \in \mathcal{L}_\omega(\mathcal{U}), \ accrun(xw) \models \Diamond q \ \} \ > \ 0$$

But then:

$$\Pr\{\ w \in \Sigma^\omega \ : \ xw \in \mathcal{L}_\omega(\mathcal{U}), \ accrun(xw) \not\models \Diamond q \ \} \ < \ 1$$

Contradiction. This completes the proof of the claim.

With an analogous argument we get that for each state $q \in Q$, $\Pr(\mathcal{L}_\omega(\mathcal{U})) = \Pr(L_n)$ where L_n is the set of all infinite words $w \in \mathcal{L}_\omega(\mathcal{U})$ such that $accrun(w)$ visits state q at least n times. Then,

$$\mathcal{L}_\omega(\mathcal{U}) \ = \ L_0 \ \supseteq \ L_1 \ \supseteq \ L_2 \ \supseteq \ \dots \ \supseteq \ \bigcap_{n\in\mathbb{N}} L_n \ \overset{\text{def}}{=} \ L$$

Then, L is the set of all words $w \in \mathcal{L}_\omega(\mathcal{U})$ such that $q \in inf(accrun(w))$. And we have:

$$\Pr(L) \ = \ \lim_{n\to\infty} \Pr(L_n) \ = \ \Pr(\mathcal{L}_\omega(\mathcal{U}))$$

This completes the proof of Theorem 29. ∎

Remark 30. The only place where the proof of Theorem 29 uses the unambiguity of $\mathcal{U}$ is in the statement that L agrees with the set of infinite words w such that $q \in inf(accrun(w))$. ∎

Remark 31. With analogous arguments one can show that the accepting runs of almost all words in $\mathcal{L}_\omega(\mathcal{U})$ contain each finite path of $\mathcal{U}$ infinitely often. ∎

D.2 Properties of cuts

In what follows, we suppose $\mathcal{U} = (Q, \Sigma, \delta, Q_0, F)$ is a strongly connected UBA where Q_0 and F are nonempty. Hence, $\mathcal{L}_\omega(q) \neq \varnothing$ for all states q. Moreover, by the strong connectivity of $\mathcal{U}$ we get $\Pr(\mathcal{L}_\omega(\mathcal{U})) > 0$ iff $\Pr(\mathcal{L}_\omega(q)) > 0$ for some state q.

The following facts are simple observations that will be used at various places:

Fact 32 *Suppose $\mathcal{U}$ is a UBA (possibly not strongly connected).*

– *If C is a cut then so are the sets $\delta(C, y)$ for all finite words $y \in \Sigma^*$.*

    – If $\Pr(\mathcal{L}_\omega(\mathcal{U})) > 0$ then there exists some finite word $x \in \Sigma^*$ such that $\delta(Q_0, xy)$ is a cut for all words $y \in \Sigma^*$.

Proof. The first statement is obvious. The argument for the second statement is as follows. Suppose $\Pr(\mathcal{L}_\omega(\mathcal{U})) > 0$. Lemma 1 asserts the existence of a finite word x such that

$$\Pr\big\{\, w \in \Sigma^\omega \ : \ xw \in \mathcal{L}_\omega(\mathcal{U})\, \big\} \ = \ 1$$

Hence, the $\delta(Q_0, x)$ is a cut, and so are the sets $\delta(Q_0, xy) = \delta(\delta(Q_0, x), y)$ for all $y \in \Sigma^*$ by the first statement. ∎

    Fact 32 yields the following characterization of almost universal UBA:

$$\mathcal{U} \text{ is almost universal} \quad \text{iff} \quad Q_0 \text{ is a cut}$$
$$\text{iff} \quad \delta(Q_0, x) \text{ is a cut for all } x \in \Sigma^*$$

The above characterization holds in any (possibly not strongly connected) UBA. An analogous characterization of positivity for strongly connected UBA is obtained using the fact that $\Pr(\mathcal{L}_\omega(\mathcal{U})) > 0$ iff $\Pr(\mathcal{L}_\omega(q)) > 0$ for some state q:

$$\mathcal{U} \text{ is positive, i.e., } \Pr(\mathcal{L}_\omega(\mathcal{U})) > 0$$
$$\text{iff} \quad \mathcal{U} \text{ has a reachable cut, i.e., a cut of the form } \delta(Q_0, x)$$
$$\text{iff} \quad \mathcal{U} \text{ has a pure cut, i.e., a cut of the form } \delta(q, x)$$

    We now elaborate the notion of cuts in strongly connected UBA.

Fact 33 *Suppose $\mathcal{U}$ is a strongly connected UBA and $\Pr(\mathcal{L}_\omega(\mathcal{U})) > 0$.*

    – *Let C be a cut and C' a subset of Q with $\mathcal{L}_\omega(q) \cap \mathcal{L}_\omega(p) = \varnothing$ for all states $q, p \in C'$ with $q \neq p$. Then, $C \subseteq C'$ implies $C = C'$.*
    – *If $\delta(q, x)$ is a cut and $y \in \Sigma^*$ such that $q \in \delta(q, xy)$ then $\delta(q, x) = \delta(q, xyx)$.*

Proof. The first statement is obvious as $\Pr(\mathcal{L}_\omega(r)) > 0$ for all states r. For the second statement we suppose $C = \delta(q, x)$ is a cut and $r \xrightarrow{y} q$ for some state $r \in C$. Then:

$$\delta(q, xyx) \ = \ \delta(C, yx) \ \supseteq \ \delta(r, yx) \ = \ \delta(\,\delta(r, y),\ x) \ \supseteq \ \delta(q, x) \ = \ C$$

is a cut that subsumes R. Hence, $C = \delta(q, xyx)$ by the first statement. ∎

    As a consequence of Lemma 4, we get that if $C \subseteq Q$ such that $\mathcal{L}_\omega(q) \cap \mathcal{L}_\omega(p) = \varnothing$ for all states $q, p \in C$ with $q \neq p$ then:

$$C \text{ is a cut} \quad \text{iff} \quad \delta(C, y) \neq \varnothing \text{ for all } y \in \Sigma^*$$

Corollary 34 (Pure cuts in strongly connected UBA; see first part of Lemma 8). *Let $\mathcal{U}$ be a strongly connected UBA with at least one final state. Then for each state q and each finite word x:*

$$\delta(q, x) \text{ is a cut} \quad \text{iff} \quad \delta(q, xy) \neq \varnothing \text{ for all } y \in \Sigma^*$$

Proof. The languages of the states in $\delta(q, x)$ are pairwise disjoint by the unambiguity of $\mathcal{U}$. Hence, $\delta(q, x)$ is a (pure) cut if and only if $\mathcal{U}[\delta(q, x)]$ is almost universal. By Lemma 4, this is equivalent to the statement that $\delta(q, xy) \neq \varnothing$ for all $y \in \Sigma^*$. ∎

The following lemma yields that for positive, strongly connected UBA, the pure cuts constitute a non-trap BSCC in the deterministic automaton $\mathcal{U}_{\mathrm{det}}$ obtained by applying the standard powerset construction.

Lemma 35 (See second part of Lemma 8). *Suppose* $\Pr(\mathcal{L}_\omega(\mathcal{U})) > 0$ *and* $\mathcal{U}$ *is strongly connected. Then, for each cut $C \subseteq Q$ the following statements are equivalent:*

(1) *C is pure, i.e., $C = \delta(q, x)$ for some state $q \in Q$ and some word $x \in \Sigma^*$*
(2) *for each state $p \in Q$ there exists a word $z \in \Sigma^*$ such that $C = \delta(p, z)$*
(3) *C is reachable from any other cut, i.e., if C' is a cut then there exists a finite word y with $C = \delta(C', y)$.*

Proof. Obviously, (1) is a consequence of (2). Let us prove the implication (1) $\Longrightarrow$ (2). Suppose $\delta(q, x)$ is a cut and let $p \in Q$ be an arbitrary state. Pick some finite word y with $p \xrightarrow{y} q$. Then, $\delta(p, yx) \supseteq \delta(q, x)$. We get $\delta(p, yx) = \delta(q, x)$ by the second statement of Fact 33.

We now show the implication (2) $\Longrightarrow$ (3). Let C' be a cut and $p \in C'$. By assumption (2), there is some word z such that $C = \delta(p, z)$. Then, $\delta(C', z)$ is a cut as well and we have:

$$C = \delta(p, z) \subseteq \delta(C', z)$$

and therefore $C = \delta(C', z)$ by the first statement of Fact 33.

For the implication (3) $\Longrightarrow$ (1) we pick a cut C' of the form $C' = \delta(p, z)$ for some state $p \in C$ and word z. (Such a cut exists by the second statement in Fact 32.) By assumption (3) there is a word y with $C = \delta(C', y)$. But then $C = \delta(p, yz)$. ∎

The above lemma shows that if $\mathcal{U}$ is positive then $\mathcal{U}_{\mathrm{det}}$ has exactly one non-trap BSCC consisting of the cuts of $\mathcal{U}$ that are reachable from some resp. all singleton(s). More precisely, if $\Pr(\mathcal{L}_\omega(\mathcal{U})) > 0$ and $\mathcal{U}$ is strongly connected then the cuts of $\mathcal{U}$ that are reachable from some singleton constitute a BSCC $\mathcal{C}$ of $\mathcal{U}_{\mathrm{det}}$. This is the only non-trap BSCC of $\mathcal{U}_{\mathrm{det}}$ and $\mathcal{C}$ is reachable from each cut. Additionally, the trap BSCC $\{\varnothing\}$ is reachable if there are states q in $\mathcal{U}$ with $\Pr(\mathcal{L}_\omega(q)) < 1$.

Remark 36. As the results above show: there is no cut C that is reachable from some singleton and that is not contained in the non-trap BSCC $\mathcal{C}$. However, there might be cuts outside $\mathcal{C}$. For example, let $\mathcal{U} = (Q, \{a, b\}, \delta, Q_0, F)$ where

$$Q = \{q_a, q_b, p_a, p_b\}, \qquad Q_0 = \{q_a, p_b\}, \qquad F = \{q_a\}$$

The transition function δ is given by:

$$\delta(q_a, a) = \{q_a, q_b\} \qquad \delta(p_a, a) = \{p_a, p_b\}$$
$$\delta(q_b, b) = \{p_a, p_b\} \qquad \delta(p_b, b) = \{q_a, q_b\}$$

and $\delta(\cdot) = \varnothing$ in all remaining cases. The unambiguity of $\mathcal{U}$ is clear since the switch between the q- and p-states are deterministic and since

$$\mathcal{L}_\omega(q_a) = \mathcal{L}_\omega(p_a) = \{ aw \ : \ w \in \{a,b\}^\omega \}$$
$$\mathcal{L}_\omega(q_b) = \mathcal{L}_\omega(p_b) = \{ bw \ : \ w \in \{a,b\}^\omega \}$$

Thus, $\mathcal{U}$ is universal. The sets $\{q_a, q_b\}$, $\{p_a, p_b\}$ constitute the non-trap BSCC consisting of the cuts that are reachable from the four singletons. The set $C = \{q_a, p_b\}$ is a cut too, but C is not reachable from any singleton. However, $\delta(C, a) = \delta(C, b) = \{q_a, q_b\}$. ∎

Lemma 37. *Suppose* $\Pr(\mathcal{L}_\omega(\mathcal{U})) > 0$ *and* $\mathcal{U}$ *is strongly connected. Let* q, p *be states in* $\mathcal{U}$ *with* $q \neq p$*. Then:*

$$\{q, p\} \subseteq C \text{ for some pure cut } C$$

$$\text{iff} \quad \text{there is some finite word } z \text{ with } q \xrightarrow{z} q \xrightarrow{z} p$$

Proof. Let C be a pure cut that contains q and p. By Lemma 35 there is a word z with $C = \delta(q, z)$. Then, $C = \delta(C, z)$ and $\delta(p, z) = \varnothing$ by the unambiguity of $\mathcal{U}$ (see Fact 28). But then $q \xrightarrow{z} q \xrightarrow{z} p$. Vice versa, $q \xrightarrow{z} q \xrightarrow{z} p$ implies $\{q, p\} \subseteq \delta(C, z)$ for each cut C with $q \in C$. ∎

Remark 38. Lemma 37 yields that for all states q and p of a strongly connected UBA $\mathcal{U}$:

- If there is no word z with $q \xrightarrow{z} q \xrightarrow{z} p$ then there is no cut C with $\{q, p\} \subseteq C$.
- If $\Pr(\mathcal{L}_\omega(\mathcal{U})) > 0$ and $q \xrightarrow{z} q \xrightarrow{z} p$ then there is a pure cut C that contains q and p.

The existence of some finite word z with $q \xrightarrow{z} q \xrightarrow{z} p$ can be checked efficiently using standard algorithms for NFA. Note that the first case applies (i.e., no such word z exists) if and only if the accepted languages of the NFA $\mathcal{B}_{q,q} = (Q, \Sigma, \delta, \{q\}, \{q\})$ and $\mathcal{B}_{q,p} = (Q, \Sigma, \delta, \{q\}, \{p\})$ are disjoint. The latter can be checked by running an emptiness check to the product-NFA of $\mathcal{B}_{q,q}$ and $\mathcal{B}_{q,p}$.

If the languages of $\mathcal{B}_{q,q}$ and $\mathcal{B}_{q,p}$ are not disjoint then we can generate a finite word z of length at most $|Q|^2$ such that $q \xrightarrow{z} q \xrightarrow{z} p$ by searching an accepted word of the product of $\mathcal{B}_{q,q}$ and $\mathcal{B}_{q,p}$. ∎

The following concept of cut languages is irrelevant for the soundness proof of our algorithm to compute $\Pr(\mathcal{L}_\omega(\mathcal{U}))$. However, we find that Lemma 40 (see below) is an interesting observation.

Definition 39 (Cut languages). *For each state q in $\mathcal{U}$, let K_q be the set of finite words x such that $\delta(q,x)$ is a cut. We refer to K_q as the cut language for state q.*

The cut languages K_q are upward-closed (i.e., if $x \in K_q$ then $xy \in K_q$ for all finite words y) by the first statement of Fact 32. The second statement of Fact 32 yields that K_q is nonempty if $\Pr(\mathcal{L}_\omega(q))$ is positive. Vice versa, if $x \in K_q$ then almost all infinite words w with $x \in \mathit{Pref}(w)$ belong to $\mathcal{L}_\omega(q)$. Hence, the cut language K_q is nonempty if and only if $\Pr(\mathcal{L}_\omega(q)) > 0$. Moreover, the cut language K_q is regular since $K_q = \mathcal{L}_{\mathit{fin}}(\mathcal{U}_{\mathrm{det}}[q])$ where $\mathcal{U}_{\mathrm{det}}[q]$ denotes the DFA that results from the powerset construction of $\mathcal{U}$ by declaring state $\{q\}$ to be initial and the states that cannot reach the trap BSCC to be final (in particular, the states in the non-trap BSCCs are final). The unambiguity of $\mathcal{U}$ yields that $K_q \cap K_p = \varnothing$ for all states $q, p \in Q$ such that $\{q,p\} \subseteq \delta(Q_0, x)$ for some finite word $x \in \Sigma^*$.

We use $\Pr(K_q)$ as a shortform notation for $\Pr(L)$ where L consists of all infinite words that have some prefix in K_q. Recall that the cut language K_q is empty if $\Pr(\mathcal{L}_\omega(q)) = 0$, in which case $\Pr(\mathcal{L}_\omega(q)) = \Pr(K_q) = 0$.

Lemma 40. $\Pr(\mathcal{L}_\omega(q)) \ = \ \Pr(K_q)$

Proof. Obviously, almost all infinite words that have a prefix in K_q belong to $\mathcal{L}_\omega(q)$. This yields $\Pr(\mathcal{L}_\omega(q)) \geqslant \Pr(K_q)$. Suppose by contradiction that $\Pr(\mathcal{L}_\omega(q)) > \Pr(K_q)$. Then, the regular language $\{w \in \mathcal{L}_\omega(q) : \mathit{Pref}(w) \cap K_q = \varnothing\}$ is positive. Lemma 1 yields the existence of some finite word x such that:

$$\Pr\big\{\, v \in \Sigma^\omega \,:\, xv \in \mathcal{L}_\omega(q), \ \mathit{Pref}(xv) \cap K_q = \varnothing \,\big\} \ = \ 1$$

In particular, $x \notin K_q$ and $\Pr\big\{\, v \in \Sigma^\omega \,:\, xv \in \mathcal{L}_\omega(q) \,\big\} = 1$. The latter yields that $\delta(q,x)$ is a cut. But then $x \in K_q$ (by definition of K_q). Contradiction. ∎

D.3   Properties of extensions

Recall that an extension for a state-word pair $(q,z) \in Q \times \Sigma^*$ is a word $y \in \Sigma^*$ such that there is some state $p \in Q$ $q \neq p$, $\delta(p,z) \neq \varnothing$ and $\{q,p\} \subseteq \delta(q,y)$. Occasionally, we refer to the pair (p,y) as an extension of (q,z).

The idea for the iterative generation of a pure cut in positive, strongly connected UBA sketched in Section 3.1 is to pick an arbitrary state q, define $z_0 = \varepsilon$ and to seek successively for an extension y for (q, z_i). If such an extension exists then we define z_{i+1} as $y\, z_i$. In this way, we obtain a strictly increasing sequence

$$\delta(q, z_0) \ \subsetneq \ \delta(q, z_1) \ \subsetneq \ \delta(q, z_2) \ \subsetneq \ \ldots$$

of subsets of Q as shown in the following lemma:

Lemma 41. *Let $\mathcal{U}$ be a (possibly non-positive, possibly not strongly connected) UBA, and let q, p be states in $\mathcal{U}$ and y, z finite words. If y is an extension of (q, z), then $\delta(q, yz)$ is a proper superset of $\delta(q, z)$.*

Proof. Since $\{q, p\} \subseteq \delta(q, y)$ we have:

$$\delta(q, z) \cup \delta(p, z) \;=\; \delta(\{q, p\}, z) \;\subseteq\; \delta(\delta(q, y), z) \;=\; \delta(q, yz)$$

In particular, $\delta(q, z) \subseteq \delta(q, yz)$. The set $\delta(p, z)$ is nonempty (by the definition of extensions). Let r be an arbitrary state in $\delta(p, z)$. Then, $r \notin \delta(q, z)$, since otherwise the word yz would have two runs from q to r:

$$q \xrightarrow{y} q \xrightarrow{z} r \quad \text{and} \quad q \xrightarrow{y} p \xrightarrow{z} r,$$

which is impossible by the unambiguity of $\mathcal{U}$ (see Fact 28). ∎

Lemma 42. *Let $\mathcal{U}$ be a positive, strongly connected UBA. Then for each state-word pair $(q, z) \in Q \times \Sigma^*$:*

$$\delta(q, z) \text{ is a cut} \quad \text{iff} \quad \text{the pair } (q, z) \text{ has no extension}$$

Proof. "$\Longrightarrow$": Let $\delta(q, z)$ be a cut and suppose by contradiction that there exists an extension y for (q, z). Let $p \in Q$ such that $q \neq p$, $\delta(p, z) \neq \varnothing$ and $\{q, p\} \subseteq \delta(q, y)$. The set $\delta(q, yz) \setminus \delta(q, z)$ is nonempty (Lemma 41). As $\Pr(\mathcal{L}_\omega(r)) > 0$ for all states r, we get:

$$\Pr\big(\mathcal{L}_\omega(\delta(q, yz))\big) \;=\; \Pr\big(\mathcal{L}_\omega(\delta(q, z))\big) + \sum_{\substack{r \,\in\, \delta(q, yz) \\ r \,\notin\, \delta(q, z)}} \Pr\big(\mathcal{L}_\omega(r)\big) \;>\; \Pr\big(\mathcal{L}_\omega(\delta(q, z))\big)$$

Hence, $\Pr(\mathcal{L}_\omega(\delta(q, z))) < 1$. But then $\delta(q, z)$ cannot be a cut as $\Pr(\mathcal{L}_\omega(R)) = 1$ for all cuts R. Contradiction.

"$\Longleftarrow$": Suppose now that (q, z) has no extension. We have to show that $C = \delta(q, z)$ is a cut.

We first observe that there is some cut R that contains C. For this, we may pick any cut R' with $q \in R'$. Then, $R = \delta(R', z)$ is a cut with $C \subseteq R$. For each state $p \in R \setminus \{q\}$, there is some finite word y with $q \xrightarrow{y} q \xrightarrow{y} p$ (see Lemma 37). Since there exists no extension for (q, z), we have $\delta(p, z) = \varnothing$ for all states $p \in R \setminus C$. This yields:

$$C \;=\; \delta(q, z) \;=\; \delta(C, z) \;=\; \delta(R, z)$$

By Fact 32, as R is a cut, $\delta(R, z)$ is a cut as well and hence so is C.

∎

D.4   Proofs for Section 3.2

We now turn to the soundness of the linear equation system that has been presented for the computation of the values $\Pr(\mathcal{L}_\omega(q))$ for $q \in Q \setminus Q_{BSCC}$ where we suppose that all BSCCs of $\mathcal{U}$ are non-trivial and positive. Recall that for $q \in Q \setminus Q_{BSCC}$:

$$\beta_q \;=\; \frac{1}{|\Sigma|} \cdot \sum_{a \in \Sigma} \sum_{p \in \delta(q, a) \cap Q_{BSCC}} \Pr(\mathcal{L}_\omega(p))$$

provided that $\delta(q, a) \cap Q_{BSCC} \neq \varnothing$ for some $a \in \Sigma$. Otherwise, $\beta_q = 0$.

We now show that the probabilities $\Pr(\mathcal{L}_\omega(q))$ for $q \in Q \setminus Q_{BSCC}$ are computable by the linear equation system shown in Figure 7 with $|Q \setminus Q_{BSCC}|$ equations and variables ζ_q for $q \in Q \setminus Q_{BSCC}$.

$$\zeta_q \;=\; \frac{1}{|\Sigma|} \cdot \sum_{a \in \Sigma} \sum_{\substack{r \,\in\, \delta(q,\,a) \\ r \,\notin\, Q_{BSCC}}} \zeta_r \;+\; \beta_q \qquad \text{for } q \in Q \setminus Q_{BSCC}$$

Fig. 7. Linear equation system for computing $\Pr(\mathcal{L}_\omega(q))$ in UBA

Theorem 43 (See Theorem 13). *If all BSCCs of $\mathcal{U}$ are non-trivial and positive, then the linear equation system in Figure 7 has a unique solution, namely $\zeta_q^* = \Pr(\mathcal{L}_\omega(q))$.*

Proof. Write $\overline{Q}$ for $Q \setminus Q_{BSCC}$. Define a matrix $M \in [0,1]^{\overline{Q} \times \overline{Q}}$ by

$$M_{q,p} \;=\; \frac{|\{a \in \Sigma : p \in \delta(q,a)\}|}{|\Sigma|} \quad \text{for all } q, p \in \overline{Q}.$$

Further, define a vector $\beta \in [0,1]^{\overline{Q}}$ with $\beta = (\beta_q)_{q \in \overline{Q}}$. Then the equation system in Figure 7 can be written as

$$\zeta \;=\; M\zeta + \beta,$$

where $\zeta = (\zeta_q)_{q \in \overline{Q}}$ is a vector of variables. By reasoning similarly as at the beginning of the proof of Lemma 5 one can show that the values $\zeta_q^* = \Pr(\mathcal{L}_\omega(q))$ for $q \in \overline{Q}$ satisfy this equation system. That is, defining $\zeta^* = (\zeta_q^*)_{q \in \overline{Q}}$ we have $\zeta^* = M\zeta^* + \beta$.

It remains to show uniqueness. Since M and β are nonnegative, it follows from monotonicity that we have

$$\zeta^* \;\geqslant\; M\zeta^* \;\geqslant\; M^2\zeta^* \;\geqslant\; \ldots, \tag{4}$$

where the inequalities hold componentwise. For $i \geqslant 1$, let Q_i be the set of states in $\overline{Q}$ that have a path in $\mathcal{U}$ of length i or shorter to a BSCC of $\mathcal{U}$. We prove by induction on i that for all $i \geq 1$:

$$\left(M^i\zeta^*\right)_q \;<\; \zeta_q^* \qquad \text{for all } q \in Q_i. \tag{5}$$

For $i = 1$, note that $\beta_q > 0$ for all $q \in Q_1$, hence $(M\zeta^*)_q < (M\zeta^* + \beta)_q = \zeta_q^*$ for all $q \in Q_1$. For the step of induction, let $q \in Q_{i+1}$ for $i \geqslant 1$. Then, there exist

$a \in \Sigma$ and $p \in \delta(q,a) \cap Q_i$. Hence, $M_{q,p} > 0$. By induction hypothesis we have $\left(M^i \zeta^*\right)_p < \zeta_p^*$. This yields:

$$
\begin{aligned}
\left(M^{i+1}\zeta^*\right)_q &= \sum_{r \in \overline{Q}} M_{q,r}\left(M^i\zeta^*\right)_r \\
&= \underbrace{M_{q,p}}_{>0}\underbrace{\left(M^i\zeta^*\right)_p}_{<\zeta_p^*} + \sum_{r \in \overline{Q}\setminus\{p\}} M_{q,r}\big(\underbrace{M^i\zeta^*}_{\leqslant \zeta^* \text{ by } (4)}\big)_r \\
&< M_{q,p} \cdot \zeta_p^* + \sum_{r \in \overline{Q}\setminus\{p\}} M_{q,r}\zeta^* \\
&= (M\zeta^*)_q \\
&\leqslant \zeta_q^* \qquad \text{by } (4)
\end{aligned}
$$

This shows (5). Since $Q_{|\overline{Q}|} = \overline{Q}$ it follows

$$
M^{|\overline{Q}|}\zeta^* \;<\; \zeta^*\,, \tag{6}
$$

where the equality is strict in all components. So there is $0 < a < 1$ with $M^{|\overline{Q}|}\zeta^* \leq a\zeta^*$. By induction, it follows $M^{|\overline{Q}|\cdot i}\zeta^* \leqslant a^i\zeta^*$ for all $i \geqslant 0$. Thus, $\lim_{i\to\infty} M^i\zeta^* = 0$, where 0 denotes the zero vector. By (6) the vector ζ^* is strictly positive. It follows that $\lim_{i\to\infty} M^i = [0]$, where $[0]$ denotes the zero matrix.

Let $x \in \mathbb{R}^{\overline{Q}}$ be an arbitrary solution of the equation system in Figure 7, i.e., $x = Mx + \beta$. Since $\zeta^* = M\zeta^* + \beta$, subtracting the two solutions yields

$$
x - \zeta^* = M(x-\zeta^*) \;=\; M^2(x-\zeta^*) \;=\; \ldots \;=\; \lim_{i\to\infty} M^i(x-\zeta^*) \;=\; [0](x-\zeta^*) = 0\,,
$$

where 0 denotes the zero vector. Hence $x = \zeta^*$, which proves uniqueness of the solution. ∎

D.5   Proofs for Section 3.3

The techniques and proofs presented for reasoning about $\Pr(\mathcal{L}_\omega(\mathcal{U}))$ in a UBA can be adapted for the case where a Markov chain $\mathcal{M} = (S, P, \iota)$ and a UBA $\mathcal{U} = (Q, \Sigma, \delta, Q_0, F)$ over the alphabet $\Sigma = S$ is given. For this we consider the product automaton $\mathcal{P} = \mathcal{M} \otimes \mathcal{U}$ and adapt the techniques presented in Section 3.3 as outlined in Section 3.3.

Recall that for $\mathcal{A}$ to be an NBA over the alphabet S, we write $\Pr_s^{\mathcal{M}}(\mathcal{A})$ for $\Pr_s^{\mathcal{M}}(\Pi)$ where Π denotes the set of infinite paths $s_0\, s_1\, s_2\, s_3 \ldots$ in the Markov chain $\mathcal{M}$ such that $s_0 = s$ and $s_1\, s_2\, s_3 \ldots \in \mathcal{L}_\omega(\mathcal{A})$. Thus, $\Pr_s^{\mathcal{M}}(\mathcal{A}) = \Pr_s^{\mathcal{M}}(L)$ where $L = \{sw : w \in \mathcal{L}_\omega(\mathcal{A})\}$. In contrast, $\Pr_s^{\mathcal{M}}(\mathcal{L}_\omega(\mathcal{A}))$ denotes the probability of the set of infinite paths w in $\mathcal{M}$ that start in s and are accepted by $\mathcal{A}$. Thus, $\Pr_s^{\mathcal{M}}(\mathcal{L}_\omega(\mathcal{A}))$ and $\Pr^{\mathcal{M}}(\mathcal{A})$ might be different.

As explained in Section 3.3:

$$\Pr^{\mathcal{M}}(\,\mathcal{L}_\omega(\mathcal{U})\,) \;\; = \;\; \sum_{s \in S} \iota(s) \;\cdot\; \sum_{q \in \delta(Q_0, s)} \Pr^{\mathcal{M}}_s(\,\mathcal{P}[s, q]\,)$$

where $\mathcal{P}[s, q]$ denotes the UBA $(S \times Q, S, \Delta, Q'_0, S \times F)$. We then adapted the algorithm for computing $\Pr(\mathcal{U}[q])$ to the case where the task is to compute the values $\Pr^{\mathcal{M}}_s(\,\mathcal{P}[s, q]\,)$. To adapt the soundness proofs accordingly we need to "relativize" the probabilities for words in S^* according to the paths in $\mathcal{M}$. That is, we have to switch from the uniform probability measure $\Pr$ over the sigma-algebra spanned by the cylinder sets of the finite words $Cyl(x) = \{xw : w \in \Sigma^\omega\}$ to the measures $\Pr^{\mathcal{M}}_s$ induced by the states of $\mathcal{M}$.

We now illustrate how the proofs can be adapted by a few central statements.

Lemma 44 (cf. Lemma 1 = Lemma 16). *Let $s \in S$ be a state of $\mathcal{M}$ and $L \subseteq S^\omega$ be an ω-regular language with $\Pr^{\mathcal{M}}_s(L) > 0$. Then, there exists a finite path $x \in S^*$ starting in state s such that almost all extensions of x belong to L according to the measure $\Pr^{\mathcal{M}}_s$, i.e.,*

$$\Pr^{\mathcal{M}}_s\{\, w \in L \,:\, x \in Pref(w) \,\} \;\; = \;\; \Pr^{\mathcal{M}}_s(\, Cyl(x)\,)$$

where $Pref(w)$ denotes the set of finite prefixes of w.

Proof. The argument is fairly the same as in the proof of Lemma 16. We regard a deterministic automaton $\mathcal{D}$ for L and consider the product Markov chain $\mathcal{M} \otimes \mathcal{D}$. In this context, we consider $\mathcal{M}$ as a transition-labeled Markov chain where all outgoing transitions of state s are labeled with s. If $\Pr^{\mathcal{M}}_s(L)$ is positive then there is a finite path x from s such that the lifting of x from $\langle s, q_{init}\rangle$ in the product ends in a state that belongs to some BSCC where the acceptance condition of $\mathcal{D}$ holds. But then, $w \in L$ for almost all infinite paths w in $\mathcal{M}$ with $x \in Pref(w)$. ∎

The product construction presented in Section 3.3 can be adapted for NBA and Markov chains. Formally, if $\mathcal{M} = (S, P, \iota)$ is a Markov chain $\mathcal{M}$ and $\mathcal{A} = (Q, S, \delta, Q_0, F)$ an NBA with the alphabet S then

$$\mathcal{M} \otimes \mathcal{A} \;\; = \;\; (S \times Q, S, \Delta, Q'_0, S \times F)$$

where

$$Q'_0 \;\; = \;\; \{\, \langle s, q\rangle \in S \times Q \,:\, \iota(s) > 0,\; q \in \delta(Q_0, s) \,\}$$

and for $s, t \in S$ with $P(s, t) > 0$ and $q \in Q$:

$$\Delta(\langle s, q\rangle, t) \;\; = \;\; \{\, \langle t, p\rangle \,:\, p \in \delta(q, t) \,\}$$

If $P(s, t) = 0$ then $\Delta(\langle s, q\rangle, t) = \varnothing$. Given that $\mathcal{M}$ viewed as an automaton over the alphabet S behaves deterministically, the NBA $\mathcal{M} \otimes \mathcal{A}$ is unambiguous if $\mathcal{A}$ is a UBA.

Lemma 45 (Generalization of Lemma 4). *For each NBA $\mathcal{A}$ over the alphabet S where $\mathcal{M} \otimes \mathcal{A}$ is strongly connected we have:*

$$\mathrm{Pr}^{\mathcal{M}}\big(\mathcal{L}_\omega(\mathcal{A})\big) = 1 - \mathrm{Pr}^{\mathcal{M}}\big\{ w \in S^\omega \: : \: \delta(Q_0, x) = \varnothing \text{ for some } x \in \mathit{Pref}(w) \big\}$$

In particular, $\mathrm{Pr}^{\mathcal{M}}(\mathcal{L}_\omega(\mathcal{A})) = 1$ if and only if $\delta(Q_0, x) \neq \varnothing$ for all finite words $x \in S^+$ where x is a finite path in $\mathcal{M}$ starting in s.

Proof. The arguments are fairly the same as in the proof of Lemma 4. Instead of Fact 20, we can rely on the original result of [14] for checking whether $\mathrm{Pr}_{\mathcal{M}}(\mathcal{L}_\omega(\mathcal{A})) > 0$ for a given Markov chain $\mathcal{M}$. Note that the only reference to the probability measure is in the form "almost all words in L enjoy property XY" which simply means "the words in L not satisfying XY constitute a null set". At a few places we used $1/2^{|x|}$ as the measure of (the cylinder set spanned by) the finite word x as label of some path from state p to q. The value $1/2^{|x|}$ has to be replaced with $\mathrm{Pr}_s(Cyl(x))$ for the corresponding state s in $\mathcal{M}$. Note if $\langle s, p \rangle \overset{x}{\longrightarrow} \langle t, q \rangle$ in $\mathcal{M} \otimes \mathcal{A}$ then $\mathrm{Pr}_s(Cyl(x))$ is positive. ∎

We now turn the soundness of the positivity check of the BSCCs in the product-UBA $\mathcal{P} = \mathcal{M} \otimes \mathcal{U}$. Recall that $\delta_{\mathcal{C}}(q, t) = \{p \in Q : \langle t, p \rangle \in \mathcal{C}\}$. Then, for $\langle s, q \rangle \in \mathcal{C}$ and $t \in S$:

$$\Delta_{\mathcal{C}}(\langle s, q \rangle, t) \quad = \quad \big\{ \langle t, p \rangle \: : \: t \in \mathrm{Post}(s), \; p \in \delta_{\mathcal{C}}(q, t) \big\}$$

and $\Delta_{\mathcal{C}}(\langle s, q \rangle, t) = \varnothing$ if $\langle s, q \rangle \notin \mathcal{C}$. The extension $\Delta_{\mathcal{C}} : (S \times Q) \times S^* \to 2^{\mathcal{C}}$ is defined in the standard way. Then, for $\langle s, q \rangle \in \mathcal{C}$ and $x \in S^*$ we have: $\Delta_{\mathcal{C}}(\langle s, q \rangle, x) \neq \varnothing$ if and only if there exists a finite path $\langle s_0, p_0 \rangle \langle s_1, p_1 \rangle \ldots \langle s_m, p_m \rangle$ in $\mathcal{C}$ with $\langle s_0, p_0 \rangle = \langle s, q \rangle$ and $s_1 s_2 \ldots s_m = x$. Recall that a BSCC $\mathcal{C}$ of $\mathcal{P}$ is said to be positive if $\mathrm{Pr}_s^{\mathcal{M}}(\mathcal{P}[s, q]) > 0$ for some/any pair $\langle s, q \rangle \in \mathcal{C}$.

Lemma 46 (cf. Lemma 5). *Let $\mathcal{C}$ be a BSCC of $\mathcal{P}$ and*

$$(*) \qquad \zeta_{s,q} \quad = \quad \sum_{t \in \mathrm{Post}(s)} \sum_{p \in \delta_{\mathcal{C}}(q,t)} P(s, t) \cdot \zeta_{t,p} \quad \textit{for all } \langle s, q \rangle \in \mathcal{C}$$

Then, the following statements are equivalent:

(1) *$\mathcal{C}$ is positive,*
(2) *the linear equation system $(*)$ has a positive solution,*
(3) *the linear equation system $(*)$ has a non-zero solution.*

Proof. The proof of Lemma 5 presented in Section 3.1 is directly applicable here as well by considering the $|\mathcal{C}| \times |\mathcal{C}|$-matrix M with

$$M_{\langle s,q \rangle, \langle t,p \rangle} \quad = \quad \begin{cases} P(s, t) : \text{if } p \in \delta(q, t) \\ 0 \qquad : \text{otherwise} \end{cases}$$

Note that the matrix M is non-negative and irreducible and the entry in the n-th power of M for state $\langle s, q \rangle$ and $\langle t, p \rangle$ is the probability with respect to $\mathrm{Pr}_s^{\mathcal{M}}$ of all infinite paths $w = s_0 s_1 s_2 \ldots$ in $\mathcal{M}$ with $s_0 = s$ such that $p \in \Delta_{\mathcal{C}}(q, s_1 \ldots s_n)$. (Recall that $\Delta_{\mathcal{C}}$ denotes the transition relation of $\mathcal{P}$ restricted to $\mathcal{C}$.) ∎

Analogously we can adapt the proof of Theorem 12 to obtain:

Theorem 47 (cf. Theorem 12). *Let $\mathcal{C}$ be a positive BSCC of $\mathcal{P}$ and C a pure cut for $\mathcal{C}$. Then, the probability vector $(\Pr_s^{\mathcal{M}}(\mathcal{L}_\omega(\mathcal{P}[s,q])))_{\langle s,q \rangle \in \mathcal{C}}$ is the unique solution of the following linear equation system:*

$$(1) \quad \zeta_{s,q} \;=\; \sum_{t \in \mathrm{Post}(s)} \sum_{p \in \delta_{\mathcal{C}}(q,t)} P(s,t) \cdot \zeta_{t,p} \quad \textit{for all } \langle s,q \rangle \in \mathcal{C}$$

$$(2) \quad \sum_{\langle s,q \rangle \in C} \zeta_{s,q} \;=\; 1$$

Proof. It is easy to see that the vector $(\zeta^*_{\langle s,q \rangle})_{\langle s,q \rangle \in \mathcal{C}}$ with

$$\zeta^*_{\langle s,q \rangle} \;=\; \Pr_s^{\mathcal{M}}(\mathcal{L}_\omega(\mathcal{P}[s,q]))$$

is indeed a solution of (1) and (2). For the uniqueness, we rely on the proof presented for Theorem 12 in Section 3.1 with the $|\mathcal{C}| \times |\mathcal{C}|$-matrix M defined by $M_{\langle s,q \rangle,\langle t,p \rangle} = P(s,t)$ if $p \in \delta(q,t)$ and $M_{\langle s,q \rangle,\langle t,p \rangle} = 0$ otherwise. ∎

Using Lemma 45 we can now adapt Corollary 34 and obtain:

Corollary 48 (Pure cuts in BSCCs of the product; cf. Corollary 34). *Let $\mathcal{C}$ be a positive BSCC of $\mathcal{P}$ with at least one final state. Then, for each state $\langle s,q \rangle \in \mathcal{C}$ and each finite word x, the following two statements are equivalent:*

(a) $\{\langle s,p \rangle : p \in \delta_{\mathcal{C}}(q,x)\}$ is a pure cut.
(b) For each $y \in S^*$, there is some $p \in \delta_{\mathcal{C}}(q,x)$ such that $\Delta_{\mathcal{C}}(\langle s,p \rangle, y) \neq \varnothing$.

As before, let $\mathcal{C}$ be a positive BSCC of the product-UBA $\mathcal{P}$. Given a state $\langle s,q \rangle$ in $\mathcal{C}$ and a word $z \in S^*$, a word $y \in S^*$ is said to be an *extension* of $(\langle s,q \rangle, z)$ if the following two conditions hold:

(1) $s\,y$ is a cycle in the Markov chain $\mathcal{M}$
(2) there exists a state $p \in Q \setminus \{q\}$ in $\mathcal{U}$ such that $\Delta_{\mathcal{C}}(\langle s,p \rangle, z) \neq \varnothing$ and $\{\langle s,q \rangle, \langle s,p \rangle\} \subseteq \Delta_{\mathcal{C}}(\langle s,q \rangle, y)$.

Note that (1) implies, if $y = t_0\,t_1 \ldots t_m$ then $t_0 \in \mathrm{Post}(s)$ and $t_m = s$. In what follows, for $s,t \in S$, $q,p \in Q$ and $x \in S^*$, we often write

$$\langle s,q \rangle \xrightarrow{\;x\;}_{\mathcal{C}} \langle t,p \rangle$$

to indicate that $\langle t,p \rangle \in \Delta_{\mathcal{C}}(\langle s,q \rangle, x)$.

Lemma 49 (cf. Lemma 41). *If y is an extension of $(\langle s,q \rangle, z)$ then $\Delta_{\mathcal{C}}(\langle s,q \rangle, yz)$ is a proper superset of $\Delta_{\mathcal{C}}(\langle s,q \rangle, z)$.*

Proof. Let $p \in Q \setminus \{q\}$ in $\mathcal{U}$ such that $\Delta_{\mathcal{C}}(\langle s, p \rangle, z) \neq \varnothing$ and $\{\langle s, q \rangle, \langle s, p \rangle\} \subseteq \Delta_{\mathcal{C}}(\langle s, q \rangle, y)$ (see condition (2)).

We first show that $\Delta_{\mathcal{C}}(\langle s, q \rangle, z) \subseteq \Delta_{\mathcal{C}}(\langle s, q \rangle, yz)$. For this, we pick a pair $\langle t, r \rangle \in \Delta_{\mathcal{C}}(\langle s, q \rangle, z)$. By condition (2) of extensions, we have $\langle s, q \rangle \in \Delta_{\mathcal{C}}(\langle s, q \rangle, z)$. Then:

$$\langle s, q \rangle \xrightarrow{y}_{\mathcal{C}} \langle s, q \rangle \xrightarrow{z}_{\mathcal{C}} \langle t, r \rangle$$

and therefore $\langle t, r \rangle \in \Delta_{\mathcal{C}}(\langle s, q \rangle, yz)$.

To show that the inclusion is strict we prove that $\Delta_{\mathcal{C}}(\langle s, p \rangle, z) \cap \Delta_{\mathcal{C}}(\langle s, q \rangle, z) = \varnothing$. We suppose by contradiction that $\langle t, r \rangle \in \Delta_{\mathcal{C}}(\langle s, p \rangle, z) \cap \Delta_{\mathcal{C}}(\langle s, q \rangle, z)$. Then:

$$\langle s, q \rangle \xrightarrow{y}_{\mathcal{C}} \langle s, q \rangle \xrightarrow{z}_{\mathcal{C}} \langle t, r \rangle \qquad \text{and} \qquad \langle s, q \rangle \xrightarrow{y}_{\mathcal{C}} \langle s, p \rangle \xrightarrow{z}_{\mathcal{C}} \langle t, r \rangle$$

But then $q \xrightarrow{y} q \xrightarrow{z} t$ and $q \xrightarrow{y} p \xrightarrow{z} t$ in $\mathcal{U}$. This is impossible by the unambiguity of $\mathcal{U}$. ∎

Lemma 50 (cf. Lemma 42). *As before, let $\mathcal{C}$ be a positive BSCC of $\mathcal{P}$. Then for each $\langle s, q \rangle \in \mathcal{C}$ and word $z \in S^+$ where the last symbol is s:*

$$\Delta_{\mathcal{C}}(\langle s, q \rangle, z) \text{ is a pure cut} \quad \text{iff} \quad \text{the pair } (\langle s, q \rangle, z) \text{ has no extension}$$

Proof. The proof is fairly similar to the proof of Lemma 42.

"$\Longrightarrow$": Let $C = \Delta_{\mathcal{C}}(\langle s, q \rangle, z)$ be a pure cut. Suppose by contradiction that there exists an extension y for $(\langle s, q \rangle, z)$. Let $p \in Q$ such that $q \neq p$, $\Delta_{\mathcal{C}}(\langle s, p \rangle, z) \neq \varnothing$ and $\{\langle s, q \rangle, \langle s, p \rangle\} \subseteq \Delta_{\mathcal{C}}(\langle s, q \rangle, y)$. By Lemma 49, the set $\Delta_{\mathcal{C}}(\langle s, q \rangle, yz)$ is a proper superset of $\Delta(\langle s, q \rangle, z)$. As $\mathcal{C}$ is positive, we have $\mathrm{Pr}_u^{\mathcal{M}}(\mathcal{P}[u, r]) > 0$ for all $\langle u, r \rangle \in \mathcal{C}$. With an argument as in the proof of Lemma 42 we obtain:

$$\mathrm{Pr}_s^{\mathcal{M}}\big(\mathcal{P}[\Delta_{\mathcal{C}}(\langle s, q \rangle, yz)]\big) \quad > \quad \mathrm{Pr}_s^{\mathcal{M}}\big(\mathcal{P}[\Delta_{\mathcal{C}}(\langle s, q \rangle, z)]\big)$$

Hence, $\mathrm{Pr}_s^{\mathcal{M}}\big(\mathcal{P}[\Delta_{\mathcal{C}}(\langle s, q \rangle, z)]\big) < 1$. But then $\Delta_{\mathcal{C}}(\langle s, q \rangle, z)$ cannot be a pure cut. Contradiction.

"$\Longleftarrow$": Suppose now that $(\langle s, q \rangle, z)$ has no extension. To prove that $C = \Delta_{\mathcal{C}}(\langle s, q \rangle, z)$ is a cut we rely on the second part of Lemma 45. Thus, the task is to show that $\Delta_{\mathcal{C}}(C, x) \neq \varnothing$ for all finite words $x \in S^+$ where x is a finite path in $\mathcal{M}$ starting in s. To prove this, one can first show that C is contained in some pure cut R of the form $\Delta_{\mathcal{C}}(\langle s, r \rangle, z)$ for some state $r \in Q$. As the last symbol of z is s, the elements of R have the form $\langle s, r \rangle$ for some $r \in Q$. For each state $p \in Q \setminus \{q\}$ where $\langle s, p \rangle \notin R$, there is some finite word y with

$$\langle s, q \rangle \xrightarrow{y}_{\mathcal{C}} \langle s, q \rangle \xrightarrow{y}_{\mathcal{C}} \langle s, p \rangle$$

For this we can rely on an adaption of Lemma 37. Since there exists no extension for $(\langle s, q \rangle, z)$, we have $\Delta_{\mathcal{C}}(\langle s, p \rangle, z) = \varnothing$ for all states $p \in Q$ where $\langle s, p \rangle \notin C$. This yields:

$$C \quad = \quad \Delta_{\mathcal{C}}(\langle s, q \rangle, z) \quad = \quad \Delta_{\mathcal{C}}(C, z) \quad = \quad \Delta_{\mathcal{C}}(R, z)$$

Using that R is a pure cut, we can now apply Lemma 45 to obtain the claim. ∎

Given a non-trivial, non-bottom SCC of $\mathcal{P}$ and a state $\langle s,q \rangle \in \mathcal{C}$, we write $\mathcal{C}[s,q]$ to denote the sub-UBA of $\mathcal{P}$ that arises when declaring $\langle s,q \rangle$ as initial state and restricting the transitions of $\mathcal{P}$ to those inside $\mathcal{C}$. That is, $\mathcal{C}[t,p] = (\mathcal{C}, S, \Delta_{\mathcal{C}}, \langle s,q \rangle, F \cap \mathcal{C})$. Then, $\mathcal{C}$ is positive iff $\mathrm{Pr}_s^{\mathcal{M}}(\mathcal{C}[s,q]) > 0$ for some state $\langle s,q \rangle \in \mathcal{C}$ iff $\mathrm{Pr}_s^{\mathcal{M}}(\mathcal{C}[s,q]) > 0$ for all states $\langle s,q \rangle \in \mathcal{C}$.

Recall that we assume a preprocessing that treats the SCCs of $\mathcal{P}$ in a bottom-up manner and turns $\mathcal{P}$ into a UBA where all BSCCs are non-trivial and positive. Q_{BSCC} denotes the set of states that are contained in some BSCC of $\mathcal{P}$ and $Q_?$ be the states of (the modified UBA) $\mathcal{P}$ not contained in Q_{BSCC}. For $\langle s,q \rangle \in Q_?$, let $\beta_{s,q} = 0$ if $\Delta(\langle s,q \rangle, t) \cap Q_{BSCC} = \varnothing$ for all $t \in S$. Otherwise:

$$\beta_{s,q} = \sum_{t \in \mathrm{Post}(s)} \sum_{\substack{p \,\in\, \delta(q,t) \text{ s.t.} \\ \langle t,p \rangle \,\in\, Q_{BSCC}}} P(s,t) \cdot \mathrm{Pr}_t^{\mathcal{M}}(\mathcal{P}[t,p])$$

Theorem 51 (cf. Theorem 13 = Theorem 43). *Notations and assumptions as before. Then, the vector $(\mathrm{Pr}_s^{\mathcal{M}}(\mathcal{P}[s,q]))_{\langle s,q \rangle \in Q_?}$ is the unique solution of the following linear equation system:*

$$\zeta_{s,q} = \sum_{t \in \mathrm{Post}(s)} \sum_{\substack{p \,\in\, \delta(q,t) \text{ s.t.} \\ \langle t,p \rangle \,\notin\, Q_{BSCC}}} P(s,t) \cdot \zeta_{t,p} \;\; + \;\; \beta_{s,q} \qquad \text{for } \langle s,q \rangle \in Q_?$$

Proof. It is easy to see that the vector $(\mathrm{Pr}_s^{\mathcal{M}}(\mathcal{P}[s,q]))_{\langle s,q \rangle \in Q_?}$ indeed solves the linear equation system. For the uniqueness of the solution, we can apply the same arguments as in the proof of Theorem 43, but now for the $|Q_?| \times |Q_?|$-matrix M given by $M_{\langle s,q \rangle, \langle t,p \rangle} = P(s,t)$ if $p \in \delta(q,t)$ and $M_{\langle s,q \rangle, \langle t,p \rangle} = 0$ otherwise, where $\langle s,q \rangle$ and $\langle t,p \rangle$ range over all states in $Q_?$. ∎

E   Separated Büchi automata

Recall that an NBA is called separated if the languages of its states are pairwise disjoint. Obviously, each separated NBA is unambiguous. Although separated Büchi automata are as powerful as the full class of NBA [10] and translations of LTL formulas into separated UBA of (single) exponential time complexity exist [41,16], non-separated UBA and even deterministic automata can be exponentially more succinct than separated UBA [8].

We now explain in which sense our algorithm for (possibly non-separated) UBA can be seen as a conservative extension of the approach presented by Couvreur, Saheb and Sutre [16]. To keep the presentation simple, we consider the techniques to compute $\mathrm{Pr}(\mathcal{L}_\omega(\mathcal{U}))$. Analogous statements hold for the computation of $\mathrm{Pr}^{\mathcal{M}}(\mathcal{L}_\omega(\mathcal{U}))$ for a given Markov chain $\mathcal{M}$.

Given a separated, strongly connected UBA $\mathcal{U} = (Q, \Sigma, \delta, Q_0, F)$ with at least one initial and one final state, we have:

$$\mathrm{Pr}(\mathcal{L}_\omega(\mathcal{U})) > 0 \quad \text{iff} \quad Q \text{ is a reachable cut}$$

Furthermore, Q is a cut iff $\delta(Q, a) = Q$ for all $a \in \Sigma$ (Lemma 4). Thus, for the special case where the given UBA is separated and positive, there is no need for the inductive construction of a cut as outlined in Section 3.1. Instead, we can deal with $C = Q$. The linear equations in Theorem 12 can be derived from the results presented in [16]. More precisely, equation (1) corresponds to the equation system in Proposition 5.1 of [16], while equation (2) can be rephrased to $\sum_{q \in Q} \zeta_q = 1$, which corresponds to the equation used in Proposition 5.2 of [16].

To check whether Q is a cut for a given (possibly non-positive) separated, strongly connected UBA, [16] presents a simple criterion that is based on a counting argument. Lemma 4.14 in [16] yields that for separated, strongly connected UBA we have:

$$Q \text{ is a cut} \quad \text{iff} \quad |\Sigma| \cdot |Q| \;=\; |\delta|$$

where $|\delta|$ is the total number of transitions in $\mathcal{U}$ given by $\sum_{q \in Q} \sum_{a \in \Sigma} |\delta(q, a)|$.

F   Implementation and Experiments

We have considered and implemented two variants for the handling of SCCs. The first variant relies on an explicit rank computation using the `COLT` library (via QR decomposition) to determine whether the equation system for checking the positivity of an SCC has a strictly positive solution (explained in Section F.1). If this is the case, then subsequently we solve the equation system with the added cut constraint to actually obtain the probabilities for the SCC states.

Another method combines both steps by an iterative eigenvector algorithm that allows simultaneously to check whether the SCC is positive and to compute the probabilities as well. In Section F.2, we explain that approach. Section F.3 reports on experiments.

F.1   Positivity check via rank computation

Let $\mathcal{U}$ be a strongly connected UBA with at least one final state and let M be the $n \times n$-matrix from the proof of Theorem 12, where $n = |Q|$ is the number of states in $\mathcal{U}$. We compute $\text{rank}(M')$ of the matrix $M' = M - I$, where I is the identity matrix. If M' has full rank, i.e., $\text{rank}(M') = n$, then $\mathcal{U}$ is non-positive.

If M' does not have full rank, we know that $\mathcal{U}$ is positive by the following argument. As M' does not have full rank n, there is a vector v such that $(M - I)v$ is the zero vector; in other words, v is an eigenvector of M with eigenvalue 1. So the spectral radius of M is at least 1. But by Lemma 6 the spectral radius of M is at most 1, so it is equal to 1. Since M is irreducible, it follows from the Perron-Frobenius theorem [7, Theorem 2.1.4 (b)] that M has a strictly positive eigenvector v' with eigenvalue 1, i.e., $Mv' = v'$. Lemma 5 then yields that $\mathcal{U}$ is positive.

F.2 Foundations of the eigenvalue algorithm

Let $\mathcal{U}$ be a strongly connected UBA with at least one final state. Let M be the matrix from the proof of Theorem 12. Define $\overline{M} = (I + M)/2$ where I denotes the $Q \times Q$ identity matrix. Denote by $\mathbf{1} = (1)_{q \in Q}$ the column vector all whose components are 1. Define $\mathbf{0}$ similarly. For $i \geq 0$ define $v(i) = \overline{M}^i \mathbf{1}$. Our algorithm is as follows. Exploiting the recurrence $v(i+1) = \overline{M}v(i)$ compute the sequence $v(0), v(1), \ldots$ until we find $i > 0$ with either $v(i+1) < v(i)$ (by this inequality we mean strict inequality in all components) or $v(i+1) \approx v(i)$. In the first case we conclude that $\Pr(\mathcal{L}_\omega(\mathcal{U})) = 0$. In the second case we compute a cut C and multiply $v(i)$ by a scalar $c > 0$ so that $c \cdot \sum_{q \in C} v(i)_q = 1$, and conclude that $(\Pr(\mathcal{L}_\omega(q)))_{q \in Q} \approx c \cdot v(i)$. This algorithm is justified by the following two lemmas.

Lemma 52. *We have $\Pr(\mathcal{L}_\omega(\mathcal{U})) = 0$ if and only if there is $i \geq 0$ with $v(i+1) < v(i)$.*

Lemma 53. *If $\Pr(\mathcal{L}_\omega(\mathcal{U})) > 0$ then $v(\infty) := \lim_{i \to \infty} v(i) > \mathbf{0}$ exists, and $Mv(\infty) = v(\infty)$, and $v(\infty)$ is a scalar multiple of $(\Pr(\mathcal{L}_\omega(q)))_{q \in Q}$.*

For the proofs we need the following two auxiliary lemmas:

Lemma 54. *For any $v \in \mathbb{C}^Q$ and any $c \in \mathbb{C}$ we have $Mv = cv$ if and only if $\overline{M}v = \frac{1+c}{2}v$. In particular, M and $\overline{M}$ have the same eigenvectors with eigenvalue 1.*

Proof. Immediate. $\qquad\square$

Lemma 55. *Let $\rho > 0$ denote the spectral radius of $\overline{M}$. Then the matrix limit $\lim_{i \to \infty} (\overline{M}/\rho)^i$ exists and is strictly positive in all entries.*

Proof. Since M is irreducible, $\overline{M}^{|Q|}$ is strictly positive (in all entries). Then it follows from [24, Theorem 8.2.7] that the matrix limit

$$\lim_{i \to \infty} (\overline{M}/\rho)^i = \lim_{i \to \infty} \left((\overline{M}/\rho)^{|Q|} \right)^i$$

exists and is strictly positive. $\qquad\square$

Proof (of Lemma 52). Let $i \geq 0$ with $v(i+1) < v(i)$. With [7, Theorem 2.1.11] it follows that the spectral radius of $\overline{M}$ is < 1, hence by Lemma 54 the spectral radius of M is < 1 as well. By Lemma 6 it follows $\Pr(\mathcal{L}_\omega(\mathcal{U})) = 0$.

For the converse, let $\Pr(\mathcal{L}_\omega(\mathcal{U})) = 0$. By Lemma 6, the spectral radius of M is < 1. Let ρ denote the spectral radius of $\overline{M}$. By Lemma 54, we have $\rho < 1$. If $\rho = 0$ then $\overline{M}$ is the zero matrix and we have $v(1) = \mathbf{0} < \mathbf{1} = v(0)$. Let $\rho > 0$. It follows from Lemma 55 that there is $i \geq 0$ such that $\rho (\overline{M}/\rho)^{i+1} < (\overline{M}/\rho)^i$ (with the inequality strict in all components). Hence $\overline{M}^{i+1} < \overline{M}^i$ and $v(i+1) = \overline{M}^{i+1}\mathbf{1} < \overline{M}^i\mathbf{1} = v(i)$. $\qquad\square$
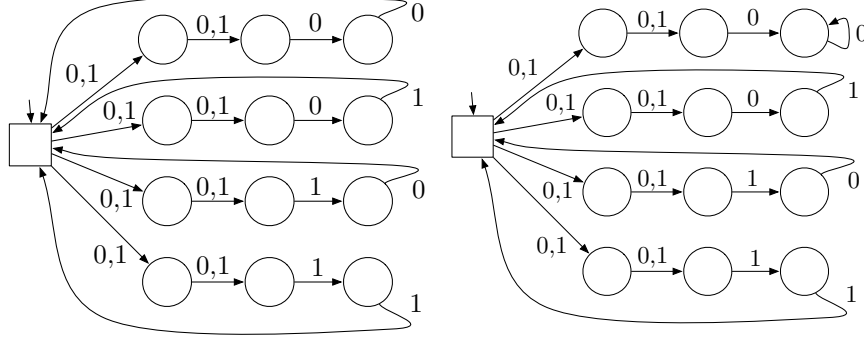
Fig. 8. UBA "complete automaton" (left) and "nearly complete automaton" (right) for $k = 2$

Proof (of Lemma 53). Let $\Pr(\mathcal{L}_\omega(\mathcal{U})) > 0$. Then, by Lemma 6, the spectral radius of M is 1. So, with Lemma 54 the spectral radius of $\overline{M}$ is 1. By Lemma 55 the limit $v(\infty) = \lim_{i\to\infty} \overline{M}^i \mathbf{1}$ exists and is positive. From the definition of $v(\infty)$ we have $\overline{M}v(\infty) = v(\infty)$. By Lemma 54 also $Mv(\infty) = v(\infty)$. So $v(\infty)$ solves equation (1) from Theorem 12. There is a scalar $c > 0$ so that $cv(\infty)$ satisfies both equations (1) and (2) from Theorem 12. By the uniqueness statement of Theorem 12 it follows that $cv(\infty) = (\Pr(\mathcal{L}_\omega(q)))_{q\in Q}$. □

F.3 Experiments

To assess the scalability of our implementation in the face of particularly difficult UBA, we have considered two families of parametrized UBA. Both have an alphabet defined over a single atomic proposition, resulting in a two-element alphabet which we use to represent either a 0 or a 1 bit. The first automaton ("complete automaton"), depicted in Figure 8 on the left for $k = 2$, is a complete automaton, i.e., recognizes the full Σ^ω. It consists of a single, accepting starting state that non-deterministically branches to one of 2^k states, each one leading after a further step to a gadget that only lets one particular of the k-bit bitstrings pass, subsequently returning to the initial state. As all the k-bit bitstrings that can occur have a gadget, the automaton is complete. Likewise, the automaton is unambiguous as each of the bitstrings can only pass via one of the gadgets.

Our second automaton ("nearly complete automaton"), depicted in Figure 8 on the right for $k = 2$, arises from the first automaton by a modification to the gadget for the "all zero" bitstring, inhibiting the return to the initial state. Clearly, the automaton is not complete.

We use both kinds of automata in an experiment using our extension of `PRISM` against a simple, two-state DTMC that encodes a uniform distribution between the two "bits", i.e., allowing us to determine whether the given automaton is almost universal. As the `PRISM` implementation requires the explicit specification of a DTMC, we end up with a product that is slightly larger than the UBA, even

though we are essentially performing the UBA computations for the uniform probability distribution. In particular, this experiment serves to investigate the scalability of our implementation in practice for determining whether an SCC is positive, for the cut generation and for computing the probabilities for the SCC states. It should be noted that equivalent deterministic automata, e.g., obtained by determinizing the UBA using the `ltl2dstar` tool are significantly smaller (in the range of tens of states) due to the fact that the UBA in question are explicitly constructed inefficiently.

Table 2. Benchmark results for "complete automaton" with parameter k

| k | UBA size | SCC size | t_{cut} | cut generation | | power iteration | | rank-based | |
				ext. checks	cut size	t_{eigen}	iter.	$t_{positive}$	t_{values}
5	193	258	0.1 s	10124	32	< 0.1 s	215	0.5 s	0.3 s
6	449	578	0.3 s	40717	64	0.1 s	282	4.1 s	4.0 s
7	1025	1282	0.6 s	163342	128	0.1 s	358	57.4 s	55.4 s
8	2305	2818	1.6 s	654351	256	0.1 s	443	900.1 s	905.7 s
9	5121	6146	9.0 s	2619408	512	0.3 s	537	-	-

Table 2 presents statistics for our experiments with the "complete automaton" with various parameter values k, resulting in increasing sizes of the UBA and the SCC (number of states). We list the time spent for generating a cut (t_{cut}), the number of checks whether a given word is an extension during the cut generation algorithm and the size of the cut. In all cases, the cut generation required 2 iterations. We then compare the SCC handling based on the power iteration with the SCC handling relying on a rank computation for determining positivity of the SCC and a subsequent computation of the values. For the power iteration method, we provide the time spent for iteratively computing an eigenvector (t_{eigen}) and the number of iterations (iters.). For the other method, we provide the time spent for the positivity check by a rank computation with a QR decomposition from the `COLT` library ($t_{positive}$) and for the subsequent computation of the values via solving the linear equation system (t_{values}). We used an overall timeout of 60 minutes for each `PRISM` invocation and an epsilon value of 10^{-10} as the convergence threshold.

As can be seen, the power iteration method for the numeric SCC handling performed well, with a modest increase in the number of iterations for rising k until converging on an eigenvector, as it can fully exploit the sparseness of the matrix. In contrast, the QR decomposition for rank computation performs worse. The time for cut generation exhibits a super-linear relation with k, which is reflected in the higher number of words that where checked to determine that they are a extension. Note that our example was chosen in particular to put stress on the cut generation.

The results for the "nearly complete automaton", depicted in Table 3, focus on the computation in the "dominant SCC", i.e., the one containing all the gadgets that return to the initial state. For the other SCC, containing the self-

Table 3. Benchmark results for "nearly complete automaton" with parameter k

k	UBA size	SCC size	power iteration		rank-based
			t_{eigen}	iter.	t_{positive}
5	193	250	$<0.1\,\text{s}$	52	$0.4\,\text{s}$
6	449	569	$<0.1\,\text{s}$	78	$3.9\,\text{s}$
7	1025	1272	$0.1\,\text{s}$	112	$53.6\,\text{s}$
8	2305	2807	$0.1\,\text{s}$	155	$878.1\,\text{s}$
9	5121	6134	$0.3\,\text{s}$	205	-

loop, non-positivity is immediately clear as it does not contain a final state. In contrast to the "complete automaton", no cut generation takes place, as the SCC is not positive. The results roughly mirror the ones for the "complete automata", i.e., the power iteration method is quite efficient in determining that the SCC is not positive, while the QR decomposition for the rank computation needs significantly more time and scales worse.

F.4 NBA vs UBA

To gain some understanding on the cost of requiring unambiguity for an NBA, we have compared the sizes of NBA and UBA generated by the `ltl2tgba` tool of SPOT for the formulas of [20,37,19] used for benchmarking, e.g., in [29]. We consider both the "normal" formula and the negated formula, yielding 188 formulas.

Table 4. Number of formulas where the (standard) NBA and UBA has a number of states $\leq x$

Number of states $\leq x$	≤ 1	≤ 2	≤ 3	≤ 4	≤ 5	≤ 7	≤ 10	≤ 12	≤ 15	≤ 20	≤ 25	≥ 25
`ltl2tgba` NBA	12	49	103	145	158	176	181	185	187	188	188	0
`ltl2tgba` UBA	12	42	74	108	123	153	168	173	176	180	181	7

As can be seen in Table 4, both the NBA and UBA tend to be of quite reasonable size. Most of the generated UBA (102) have the same size as the NBA and for 166 of the formulas the UBA is at most twice the size as the corresponding NBA. The largest UBA has 112 states, the second largest has 45 states.