Model Checking Markov Chains Against Unambiguous Büchi Automata

Michael Benedikt, Rastislav Lenhardt, and James Worrell

Department of Computer Science, University of Oxford, UK

1 Erratum

The authors would like to withdraw the claimed proof of Theorem 2 in the note below. The problems with the proof (and the overall approach, based on the notion of recurrent states) are detailed in a counterexample that can be found at `https://s3-eu-west-1.amazonaws.com/cav16-uba/counterexample.pdf`. In particular, the above document gives an example of a Markov chain M and automaton A such that $P_M(L(A))$ is strictly positive but for which the product $M \otimes A$ has no recurrent state. This shows that Equation (6) does not hold. The problem here stems from Equation (5), which is invalid.

2 Introduction

An automaton is *unambiguous* if each word has at most one accepting run and *separated* if no word is accepted from two distinct states. The classical translation of LTL formulas to Büchi automata [5] produces unambiguous separated automata since the states of such automata correspond to complete subformula types. Motivated by this observation, Couvreur *et al.* [3] present a polynomial-time algorithm to model check Markov chains against separated unambiguous Büchi automata

In this note we give a polynomial-time algorithm for model checking Markov chains against Büchi automata that are unambiguous but not necessarily separated. Apart from the extra generality of this procedure, our main motivation is the fact that the *build-by-need* translation from LTL to Büchi automata described in [1]—adapting the construction of [4]—produces automata that are unambiguous but which may not be separated.

3 Definitions

We briefly recall the main definitions. See [2,3] for more details.

A *Markov chain* $M = (S, P, \pi)$ consists of a set S of *states*, a *transition probability function* $P : S \times S \to [0, 1]$ such that $\sum_{t \in S} P(s, t) = 1$ for each state $s \in S$, and an *initial probability distribution* π on S. We assume that all numerical data are rational.

We denote by $\mathrm{Pr}_M(L)$ the probability that M performs a trajectory in a given measurable set $L \subseteq S^\omega$. We extend this notation to sets of finite words $L \subseteq S^*$, writing $P_M(L)$ as shorthand for $P_M(LS^\omega)$.

A *non-deterministic automaton* $A = (\Sigma, Q, Q_0, \delta, F)$ comprises a finite *alphabet* Σ, a finite set of *states* Q, set of *initial states* $Q_0 \subseteq Q$, *transition function* $\delta : Q \times \Sigma \to 2^Q$, and set of *accepting states* F. We extend δ to a function $\delta : Q \times \Sigma^+ \to 2^Q$ by inductively defining $\delta(q, w\sigma) = \bigcup\{\delta(q', \sigma) : q' \in \delta(q, w)\}$, where $w \in \Sigma^+$. We consider automata alternatively as acceptors of finite words and acceptors of infinite words (via the Büchi acceptance condition). In the former case we speak of *non-deterministic finite automata (NFA)* and in the latter case of *non-deterministic Büchi automata (NBA)*. In either case we write $L(A)$ for the language accepted by A.

4 Main Result

Let $M = (S, P, \pi)$ be a Markov chain, A an unambiguous NBW with alphabet Σ, and $\lambda : S \to \Sigma$ a function labelling the states of M with letters from the alphabet of A. Write $||M||$ and $||A||$ for the respective lengths of the representations of M and A, assuming that integers are encoded in binary. We show how to compute $\mathrm{Pr}_M\{s_1 s_2 \ldots \in S^\omega : \lambda(s_1)\lambda(s_2)\ldots \in L(A)\}$—the probability that a trajectory of M is accepted by A—in time polynomial in $||M||$ and $||A||$.

Without loss of generality, by first applying an existential renaming to A along λ, we assume that the alphabet of A is the set of states of M, i.e., $\Sigma = S$, and the state-labelling map λ is the identity. Note that unambiguous automata are preserved under existential renaming. Our task is now to compute $\mathrm{Pr}_M(L(A))$. We first consider the task of determining whether $\mathrm{Pr}_M(L(A)) > 0$.

Lemma 1. *Let* $M = (S, P, \pi)$ *be a Markov chain and* $A = (S, Q, Q_0, \delta, F)$ *an unambiguous NFA. Then* $\mathrm{Pr}_M(L(A))$ *is computable in time polynomial in* $||A||$ *and* $||M||$.

Proof. Let $L(A, q) \subseteq S^*$ denote the set of words accepted by A with $q \in Q$ as initial state. Similarly let $\mathrm{Pr}_{M,s}$ denote the probability distribution on S^ω induced by M with initial state distribution $P(s, -)$. Without loss of generality, assume that S contains a state s_0 with $P(s_0, s) = \pi(s)$ for each $s \in S$. Let us also assume that every state in A is reachable from Q_0 and can reach F.

Define a directed graph $G_{M \otimes A} = (V, E)$, with set of vertices $V = S \times Q$ and $(s, q)\ E\ (s', q')$ if and only if $P(s, s') > 0$ and $q' \in \delta(q, s')$. Say that a vertex $(s, q) \in V$ is *accepting* if $q \in F$ and *dead* if it cannot reach an accepting vertex. Write V^{acc} and V^{dead} for the respective sets of accepting and dead vertices, and write $V^? = V \setminus (V^{acc} \cup V^{dead})$.

Introduce a real-valued variable $\xi_{s,q}$ to represent $\mathrm{Pr}_{M,s}(L(A, q))$, so that $\sum_{q \in Q_0} \xi_{s_0,q}$ represents $\mathrm{Pr}_M(L(A))$. We claim that the following system of equations uniquely defines $\xi_{s,q}$:

$$\begin{aligned}
\xi_{s,q} &= 0 && (s,q) \in V^{dead} && (1)\\
\xi_{s,q} &= 1 && (s,q) \in V^{acc} && (2)\\
\xi_{s,q} &= \sum_{s' \in S} \sum_{q' \in \delta(q,s')} P(s,s') \cdot \xi_{s',q'} && (s,q) \in V^{?} && (3)
\end{aligned}$$

The correctness of (1) and (2) is self-evident. Correctness of (3) follows from the following calculation:

$$\begin{aligned}
\xi_{s,q} &= \mathrm{Pr}_{M,s}(L(A,q))\\
&= \sum_{s' \in S} P(s,s') \cdot \mathrm{Pr}_{M,s'}\Big[\bigcup_{q' \in \delta(q,s')} L(A,q')\Big]\\
&= \sum_{s' \in S} \sum_{q' \in \delta(q,s')} P(s,s') \cdot \mathrm{Pr}_{M,s'}(L(A,q')) \qquad A \text{ is unambiguous}\\
&= \sum_{s' \in S} \sum_{q' \in \delta(q,s')} P(s,s') \cdot \xi_{s',q'}\,.
\end{aligned}$$

To see that the solution of (3) is unique, write the equation system in matrix form as $\boldsymbol{\xi} = C\boldsymbol{\xi} + \boldsymbol{d}$, where $\boldsymbol{\xi} = \{\xi_{(s,q)} : (s,q) \in V^{?}\}$,

$$C_{(s,q),(s',q')} = \begin{cases} P(s,s') & q' \in \delta(q,s') \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad d_{(s,q)} = \sum_{s' : \delta(q,s') \cap F \neq \emptyset} P(s,s')\,.$$

Given two solutions $\boldsymbol{\xi}$ and $\boldsymbol{\xi'}$, we have $\boldsymbol{\xi} - \boldsymbol{\xi'} = C^n(\boldsymbol{\xi} - \boldsymbol{\xi'})$ for all n. We will show that $\lim_n C^n = 0$, which proves uniqueness.

The entry of index (s,q) in $(I + C + \cdots + C^n)\boldsymbol{d}$ is $\mathrm{Pr}_{M,s}(L(A,q) \cap S^{\leq n})$, which converges to $\mathrm{Pr}_{M,s}(L(A,q))$ as n tends to infinity. It follows that $\lim_n C^n(I + C + \cdots + C^m)\boldsymbol{d} = 0$ for any fixed $m \in \mathbb{N}$. But, since all vertices in $V^{?}$ can reach V^{acc}, there exists some m such that $(I + C + \cdots + C^m)\boldsymbol{d}$ is strictly positive in every entry. We conclude that $\lim_n C^n = 0$.

Since systems of linear equations can be solved in polynomial time, the result follows. □

We now use Lemma 1 to handle the case of automata over infinite words. In particular we use the lemma to classify states of the product $M \otimes A$ as *recurrent* or not.

Theorem 2. *Let $M = (S, P, \pi)$ be a Markov chain and $A = (S, Q, Q_0, \delta, F)$ an unambiguous NBA. Then $P_M(L(A))$ is computable in time polynomial in $||M||$ and $||A||$.*

Proof. Given $(s,q) \in S \times F$, define $G_{s,q}, H_{s,q} \subseteq S^+$ by

$$\begin{aligned}
G_{s,q} &= \{s_1 \dots s_k \in S^+ : s_k = s \text{ and } q \in \textstyle\bigcup_{p \in Q_0} \delta(p, s_1 \dots s_k)\}\\
H_{s,q} &= \{s_1 \dots s_k \in S^+ : s_k = s \text{ and } q \in \delta(q, s_1 \dots s_k)\}\,.
\end{aligned}$$

Thus $G_{s,q}$ is the set of finite trajectories of M that end in state s and which lead A from an initial location to q, while $H_{s,q}$ is the set of finite trajectories of M that end in state s and that lead A from location q back to itself.

Clearly we can express $L(A)$ as the following ω-regular expression:

$$L(A) = \bigcup_{(s,q)\in S\times F} G_{s,q} H^{\omega}_{s,q} \,. \tag{4}$$

Define $(s,q) \in S \times F$ to be *recurrent* if $\mathrm{Pr}_{M,s}(H_{s,q}) = 1$. We claim that if (s,q) is recurrent then $\mathrm{Pr}_{M,s}(H^{\omega}_{s,q}) = 1$, and if (s,q) is not recurrent then $\mathrm{Pr}_{M,s}(H^{\omega}_{s,q}) = 0$.

Suppose first that (s,q) is recurrent. Consider the set of trajectories S^{ω} under the measure $\mathrm{Pr}_{M,s}$. Inductively define a sequence of random variables $\{h_n\}_{n\in\mathbb{N}}$ on S^{ω} with values in $\mathbb{N}\cup\{\infty\}$ by writing $h_0 = 0$, and

$$h_{n+1} = \begin{cases} \min\{k : s_{h_n+1}\ldots s_k \in H_{s,q}\} & \text{if } h_n < \infty \\ \infty & \text{otherwise} \end{cases}$$

Then $\mathrm{Pr}(h_{n+1} < \infty \mid h_n < \infty) = \mathrm{Pr}_{M,s}(H_{s,q}) = 1$. It follows that $\mathrm{Pr}(\bigcap_n h_n < \infty) = 1$ and, *a fortiori*, that $\mathrm{Pr}_{M,s}(H^{\omega}_{s,q}) = 1$. On the other hand, if (s,q) is not recurrent then

$$\mathrm{Pr}_{M,s}(H^{\omega}_{s,q}) = \lim_{n<\omega} \mathrm{Pr}_{M,s}(H^{n}_{s,q}) = \lim_{n<\omega} (\mathrm{Pr}_{M,s}(H_{s,q}))^n = 0 \tag{5}$$

and the claim is established.

From Equation (4), we conclude that

$$\mathrm{Pr}_M(L(A)) = \mathrm{Pr}_M\left(\bigcup_{(s,q)\ \text{recurrent}} G_{s,q} \right) . \tag{6}$$

Now $H_{s,q}$ is the language of an unambiguous NFA. The automaton in question is obtained from A by making q the initial state, adding a new sink state q_{acc}, for every transition $p \xrightarrow{s} q$ adding a transition $p \xrightarrow{s} q_{acc}$, and making q_{acc} the unique accepting state. Thus, by Lemma 1, we can determine whether (s,q) is recurrent in time polynomial in $||M||$ and $||A||$.

The language appearing on the right-hand side of (6) is likewise expressible by an unambiguous NFA. Applying Lemma 1 once again, we can calculate $\mathrm{Pr}_M(L(A))$ in time polynomial in $||A||$ and $||M||$. □

References

1. Michael Benedikt, Rastislav Lenhardt, and James Worrell. LTL model checking of interval markov chains. In *TACAS*, volume 7795 of *Lecture Notes in Computer Science*, pages 32–46. Springer, 2013.
2. C. Courcoubetis and M. Yannakakis. The complexity of probabilistic verification. *J. ACM*, 42(4):857–907, 1995.

3. Jean-Michel Couvreur, Nasser Saheb, and Grégoire Sutre. An optimal automata approach to LTL model checking of probabilistic systems. In *LPAR*, volume 2850 of *Lecture Notes in Computer Science*, pages 361–375. Springer, 2003.
4. Rob Gerth, Doron Peled, Moshe Y. Vardi, and Pierre Wolper. Simple on-the-fly automatic verification of linear temporal logic. In *PSTV*, volume 38 of *IFIP Conference Proceedings*, pages 3–18. Chapman & Hall, 1996.
5. Moshe Y. Vardi and Pierre Wolper. An automata-theoretic approach to automatic program verification (preliminary report). In *LICS*, pages 332–344. IEEE Computer Society, 1986.