Providing Evidence of Likely Being on Time
– Counterexample Generation for CTMC Model Checking –

Tingting Han[1,2] and Joost-Pieter Katoen[1,2]

[1] Software Modelling and Verification, RWTH Aachen University, Germany
[2] Formal Methods and Tools, University of Twente, The Netherlands
E-mail: {tingting.han, katoen}@cs.rwth-aachen.de

Abstract. Probabilistic model checkers typically provide a list of individual state probabilities on the refutation of a temporal logic formula. For large state spaces, this information is far too detailed to act as useful diagnostic feedback. For quantitative (constrained) reachability problems, sets of paths that carry enough probability mass are more adequate. We recently have shown that in the context of discrete-time probabilistic processes, such sets of smallest size can be efficiently computed by (hop-constrained) k-shortest path algorithms. This paper considers the problem of generating counterexamples for continuous-time Markov chains. The key contribution is a set of approximate algorithms for computing small sets of paths that indicate the violation of time-bounded (constrained) reachability probabilities.

1 Introduction

A major strength of model checking is the possibility to generate counterexamples in case of a property violation. In fact, it is this facility that makes model checking an effective bug hunting technique. Even if only a fragment of the entire model can be searched, such counterexamples provide useful diagnostic feedback. Efficient algorithms for generating (succinct) counterexamples therefore have received considerable attention by the model checking community, cf. [5, 9, 18]. For probabilistic models, though, counterexample generation is far less developed.

Model checking of probabilistic models is focused on verifying system models in which transitions are equipped with random information. Popular models are discrete- and continuous-time Markov chains (DTMCs and CTMCs, respectively), and variants thereof which exhibit nondeterminism. Most probabilistic model checkers support variants of CTL [3, 4, 11]. For quantitative properties such as "the (maximal) probability to reach a set of goal states by avoiding certain states is at most p", alternative algorithms have to be employed. In case such property is refuted, the idea is to provide a set of paths—such path is called an *evidence*—that all together carry a probability mass that exceeds p. As such sets could be huge, the interest is in generating small sets, possibly the smallest possible. Preferably, the probability mass of such sets deviates significantly from the bound p.

Recently, we have shown [10] that for DTMCs wrt. the quantitative (hop-constrained) until formulas, most probable evidences—thus contributing the most to the violation—can be determined efficiently using either well-known (hop-constrained) shortest path

(SP or HSP) algorithms, or Viterbi's algorithm. In addition, smallest counterexamples—containing the least number of evidences while maximally deviating from p among all counterexamples containing the same number of evidences—can be determined using k-SP (k-HSP) algorithms. Here, k is the size of the counterexample and is determined on-the-fly. Similar results hold for properties where p is a lower bound, where sets of paths are considered that indicate the violation of the "dual" of the formula to be checked; see [10] for details.

This paper considers the generation of evidences and counterexamples for model checking CSL [3, 4] on CTMCs. For (hop-constrained) reachability properties expressed in CSL, the algorithms of [10] can be exploited. Properties that involve time, however, require other strategies. The continuous-time setting is unfortunately different and more complicated than the discrete one. First, an evidence cannot be a single timed path (an alternating sequence of states and time instants) as such paths have zero probability. Instead, we consider *symbolic evidences* for $\Phi \,\mathsf{U}^{\leqslant t} \Psi$, i.e., time-abstract paths—finite state sequences—that satisfy $\Phi \,\mathsf{U}\, \Psi$. A symbolic evidence induces a set of concrete evidences, viz. the set of timed paths on the same state sequence whose duration does not exceed t. Counterexamples are sets of symbolic evidences that exceed probability p. The main contribution of the paper is a set of algorithms for computing informative (symbolic) evidences and counterexamples, i.e., evidences with large probability and small counterexamples. We first indicate how the likelihood of symbolic evidences can be computed, both numerically and analytically. The latter approach exploits the fact that symbolic evidences are in fact acyclic CTMCs for which closed-form solutions exist [15]. We then consider the problem of how to find symbolic evidences such that small counterexamples result. First, we (naively) apply the strategy from [10], i.e., use k-SP algorithms on a discretized CTMC (obtained by uniformization [12]). This yields a simple algorithm, though may result in large counterexamples. A first variant exploits timing information and generates paths in the discretized CTMC that correspond to symbolic evidences. The advantage of this approach is that one can guarantee that counterexamples are obtained that contain the smallest number of evidences wrt. to their probability contribution in the CTMC. As probable paths of this kind usually correspond to probable symbolic evidences, this yields small counterexamples. Finally, we present a heuristic to improve the time and memory efficiency of this algorithm.

Organization of the paper. Section 2 summarizes the main steps of counterexample generation for DTMCs, and defines the main concepts of CTMCs needed for the rest of the paper. Section 3 defines symbolic evidences and counterexamples. Computing probabilities of symbolic evidences is treated in Section 4. Section 5 and 6 present the algorithms for determining symbolic evidences. Section 7 concludes the paper.

2 Preliminaries

Counterexample generation in DTMCs. Let AP denote a fixed, finite set of atomic propositions ranged over by $a, b, c, \ldots$.

Definition 1 (DTMC). *A (labelled)* discrete-time Markov chain *(DTMC) $\mathcal{D}$ is a triple $(S, \mathbf{P}, L)$ with S a finite set of states, $\mathbf{P} : S \times S \to [0,1]$ a stochastic matrix, and $L : S \to 2^{AP}$ a labelling function.*

For a DTMC, $\sum_{s' \in S} \mathbf{P}(s, s') = 1$, i.e. it is *stochastic*. If $\sum_{s' \in S} \mathbf{P}(s, s') \in [0, 1)$, then we call the model a *fully probabilistic system (FPS)* and it is *sub-stochastic*. A state s is absorbing if $\mathbf{P}(s, s) = 1$, i.e., if s only has a self-loop. A path σ in $\mathcal{D}$ is a state sequence $s_0\, s_1\, s_2 ...$ such that $\mathbf{P}(s_i, s_{i+1}) > 0$, for all i. The probability $\Pr\{\sigma\}$ for finite $\sigma = s_0\, s_1 \ldots s_n$ is defined as $\mathbf{P}(s_0, s_1) \cdot \mathbf{P}(s_1, s_2) \cdot \ldots \cdot \mathbf{P}(s_{n-1}, s_n)$. For finite set of paths C, $\Pr(C) = \sum_{\sigma \in C} \Pr\{\sigma\}$. $\sigma[i]$ denotes the $(i+1)$-st state on σ.

For PCTL [11] formula $\mathcal{P}_{\leqslant p}(\phi)$ where ϕ is a path formula, we have:

$$s \not\models \mathcal{P}_{\leqslant p}(\phi) \quad \text{iff} \quad \Pr\{\sigma \mid \sigma[0] = s \text{ and } \sigma \models \phi\} > p.$$

So, $\mathcal{P}_{\leqslant p}(\phi)$ is refuted by state s whenever the total probability mass of all ϕ-paths that start in s exceeds p. This indicates that a counterexample for $\mathcal{P}_{\leqslant p}(\phi)$ is a *set* of paths starting in s and satisfying ϕ. As ϕ is a path formula whose validity can be witnessed by finite state sequences, *finite* paths suffice.

Definition 2 (Evidence). *An* evidence *for $\mathcal{P}_{\leqslant p}(\phi)$ in state s is a finite path σ that starts in s and minimally satisfies ϕ. A* strongest evidence *is an evidence σ^* such that $\Pr\{\sigma^*\} \geqslant \Pr\{\sigma\}$ for any evidence σ.*

A finite path σ minimally satisfies ϕ if it satisfies ϕ, but no proper prefix of σ does so.

Definition 3 (Counterexample). *A* counterexample *for $\mathcal{P}_{\leqslant p}(\phi)$ in state s is a set C of evidences such that $\Pr(C) > p$. C^* is a* smallest (most indicative) counterexample *if $|C^*| \leqslant |C|$ for all counterexamples C and $\Pr(C^*) \geqslant \Pr(C')$ for any counterexample C' with $|C'| = |C^*|$.*

The intuition is that a smallest counterexample is mostly exceeding the required probability bound given that it has the smallest number of paths. To compute the strongest evidence and smallest counterexample, the DTMC $\mathcal{D}$ is transformed to a weighted digraph $\mathcal{G}_D = (V, E, w)$, where V and E are finite sets of vertices and edges, respectively. $V = S$ and $(v, v') \in E$ iff $\mathbf{P}(v, v') > 0$, and $w(v, v') = \log(\mathbf{P}(v, v')^{-1})$. Multiplication of transition probabilities is thus turned into the addition of edge weights along paths. Now:

Lemma 1. *For any path σ from s to t in DTMC $\mathcal{D}$, $k \in \mathbb{N}_{>0}$, and $h \in \mathbb{N} \cup \{\infty\}$: σ is a k-th most probable path of at most h hops in $\mathcal{D}$ iff σ is a k-th shortest path of at most h hops in $\mathcal{G}_D$.*

Consider $\phi = \Phi \,\mathsf{U}^{\leqslant h}\, \Psi$ for PCTL state-formulas Φ, Ψ and hop bound $h \in \mathbb{N} \cup \{\infty\}$. If $s \not\models \mathcal{P}_{\leqslant p}(\phi)$, then a strongest evidence can be found by a shortest path (SP) algorithm once all Ψ-states and all $(\neg\Phi \wedge \neg\Psi)$-states in DTMC $\mathcal{D}$ are made absorbing. Similarly, a smallest counterexample can be determined by k-SP algorithms that allow k to be determined on-the-fly. If $h \neq \infty$, hop-constrained SP and k-SP algorithms need to be employed; they have pseudo-polynomial time complexity in $\mathcal{O}(hm)$ and $\mathcal{O}(hm + hk \log(\frac{m}{n}))$, respectively, where $n = |S|$ and m is the number of non-zero entries in $\mathbf{P}$.

CTMCs.

Definition 4 (CTMC). *A (labelled) continuous-time Markov chain (CTMC) $\mathcal{C}$ is a quadruple $(S, \mathbf{P}, E, L)$ with $(S, \mathbf{P}, L)$ a DTMC and $E : S \to \mathbb{R}_{\geqslant 0}$ a rate vector, assigning exit rates to states.*

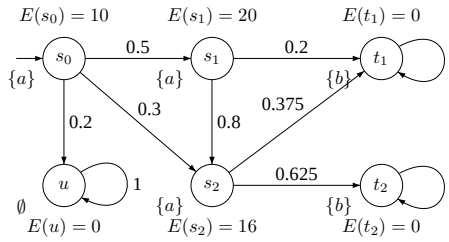


Fig. 1. CTMC $\mathcal{C}$

$(S, \mathbf{P}, L)$ is the *embedded* DTMC of $\mathcal{C}$. $E(s)$ denotes the rate of firing a transition from s, which, in other words, specifies the average delay of transitions. More precisely, with probability $(1 - e^{-E(s)\cdot t})$, a transition is enabled within the next t time units provided that the current state is s. If $\mathbf{P}(s, s') > 0$ for more than one state s', a *race* between the outgoing transitions from s exists. The probability of transition $s \to s'$ winning this race in time interval $[0, t]$ is given by:

$$\mathbf{P}(s, s', t) = \mathbf{P}(s, s') \cdot \big(1 - e^{-E(s)\cdot t}\big).$$

The probability density function is $p(s, s', t) = \mathbf{P}(s, s') \cdot E(s) \cdot e^{-E(s)\cdot t}$. Note that $\mathbf{P}(s, s', t) = \int_0^t p(s, s', t_1) \cdot dt_1$. We sometimes use $\mathbf{R}(s, s') = \mathbf{P}(s, s') \cdot E(s)$ to denote the rate of the transition $s \to s'$.

Remark 1. Except for absorbing states, all states in a CTMC are assumed to have no self-loops. The reason for this assumption will become clear later. Note that this is not a severe restriction as self-loops can be removed without affecting the transient and the steady-state probabilities of the CTMC.

Example 1. An example CTMC $\mathcal{C}$ is shown in Fig. 1. $S = \{s_i, t_1, t_2, u\}$; $L(s_i) = \{a\}$, $L(t_1) = L(t_2) = \{b\}$ and $L(u) = \emptyset$ with $0 \leqslant i \leqslant 2$; $E(s_0) = 10$, $E(s_1) = 20$, and so on. States u, t_1 and t_2 are absorbing.

Paths and probability measure.

Definition 5 (Timed paths in CTMCs). *Let $\mathcal{C} = (S, \mathbf{P}, E, L)$ be a CTMC. An* infinite *timed path σ is a sequence $s_0 \xrightarrow{t_0} s_1 \xrightarrow{t_1} s_2 \xrightarrow{t_2} \ldots$ with $s_i \in S$ and $t_i \in \mathbb{R}_{\geqslant 0}$ such that $\mathbf{P}(s_i, s_{i+1}) > 0$ for $i \geqslant 0$. A* finite *timed path σ is a finite prefix of an infinite path ending in an absorbing state.*

Let $|\sigma|$ denote the length of the path σ, i.e., $|s_0 \xrightarrow{t_0} s_1 \xrightarrow{t_1} \ldots s_{l-1} \xrightarrow{t_{l-1}} s_l| = l$, $|s_0| = 0$ and $|\sigma| = \infty$ for infinite σ. For (finite or infinite) path σ and $i < |\sigma|$, let $\sigma[i] = s_i$ be the $(i{+}1)$-st state of σ, and $\delta(\sigma, i) = t_i$ be the time spent in s_i. For $t \in \mathbb{R}_{\geqslant 0}$ and k the smallest index with $t < \sum_{j=0}^{k} t_j$, let $\sigma@t = \sigma[k]$ denote the state in σ occupied at time t. For finite path σ and $l = |\sigma|$, $\delta(\sigma, l) = \infty$; and for $t \geqslant \sum_{j=0}^{l-1} t_j$, $\sigma@t = s_l$.

A *time-abstract path* is obtained by omitting all timing information from a timed path. The function α performs this, i.e., $\alpha(s_0 \xrightarrow{t_0} s_1 \xrightarrow{t_1} \ldots) = s_0 s_1 \ldots$. Let $Paths^{\mathcal{C}}$ denote

the set of all timed paths in CTMC $\mathcal{C}$ and $Paths^{\mathcal{C}}_{abs}$ all time-abstract paths in $\mathcal{C}$. The superscript is omitted if $\mathcal{C}$ is clear from the context. $Paths(s)$ and $Paths_{abs}(s)$ denote the set of timed and time-abstract paths starting from s, respectively. We use ρ to range over time-abstract paths.

A σ-algebra and probability measure of the timed paths of a CTMC can be defined using the standard cylinder set construction, cf. [4]. It follows that time-convergent paths, i.e., paths on which time does not diverge, have probability 0.

CSL. Continuous Stochastic Logic (CSL) [4] is a variant of the logic originally proposed by Aziz et al. [3] and extends PCTL by path operators that reflect the real-time nature of CTMCs: in particular, a time-bounded until operator.

Syntax. The syntax of CSL state-formulae is defined as follows:

$$\Phi ::= \mathrm{tt} \mid a \mid \neg\Phi \mid \Phi \wedge \Phi \mid \mathcal{P}_{\trianglelefteq p}(\phi),$$

where $p \in [0,1]$ is a probability, $\trianglelefteq \in \{<, \leqslant, >, \geqslant\}$. For t a non-negative real number or $t = \infty$, ϕ is a path-formula defined according to the following grammar:

$$\phi ::= \Phi \,\mathsf{U}^{\leqslant t}\, \Psi \mid \Phi \,\mathsf{W}^{\leqslant t}\, \Psi.$$

The path formula $\Phi \,\mathsf{U}^{\leqslant t} \Psi$ asserts that Ψ is satisfied within t time units and that at all preceding time instants Φ holds. $\mathsf{W}^{\leqslant t}$ is the weak counterpart which does not require Ψ to eventually become true. For the sake of simplicity, the next-operator and the steady-state operator [4] are not considered here.

Semantics. CSL state-formulae are interpreted over the states of a CTMC. Let $\mathcal{C} = (S, \mathbf{P}, E, L)$ with labels in AP, and $Sat(\Phi) = \{s \in S \mid s \models \Phi\}$. The semantics of CSL state-formulae is defined for path-formula ϕ as:

$s \models \mathrm{tt}$	iff	*true*	$s \models \Phi \wedge \Psi$	iff	$s \models \Phi$ and $s \models \Psi$
$s \models a$	iff	$a \in L(s)$	$s \models \mathcal{P}_{\trianglelefteq p}(\phi)$	iff	$\mathit{Prob}(s, \phi) \trianglelefteq p$
$s \models \neg\Phi$	iff	not $(s \models \Phi)$			

$Prob(s, \phi)$ denotes the probability measure of all paths $\sigma \in Paths$ starting in state s and satisfying ϕ, i.e., $Prob(s, \phi) = \Pr\{\sigma \in Paths(s) \mid \sigma \models \phi\}$. For a timed path σ in $\mathcal{C}$, the satisfaction relation for CSL path-formulae is defined as:

$\sigma \models \Phi \,\mathsf{U}^{\leqslant t} \Psi$	iff	$\sigma@x \models \Psi$ for some $x \leqslant t$ and $\sigma@y \models \Phi$ for all $y < x$,
$\sigma \models \Phi \mathsf{W}^{\leqslant t} \Psi$	iff	either $\sigma \models \Phi \,\mathsf{U}^{\leqslant t} \Psi$ or $\sigma@x \models \Phi$ for all $x \leqslant t$.

The until and weak until operators are closely related. This follows from the following equations. For any CSL-formulae Φ and Ψ we have:

$$\mathcal{P}_{\geqslant p}(\Phi \,\mathsf{W}^{\leqslant t} \Psi) \equiv \mathcal{P}_{\leqslant 1-p}\big((\Phi \wedge \neg\Psi) \,\mathsf{U}^{\leqslant t} (\neg\Phi \wedge \neg\Psi)\big)$$
$$\mathcal{P}_{\geqslant p}(\Phi \,\mathsf{U}^{\leqslant t} \Psi) \equiv \mathcal{P}_{\leqslant 1-p}\big((\Phi \wedge \neg\Psi) \mathsf{W}^{\leqslant t} (\neg\Phi \wedge \neg\Psi)\big)$$

Counterexamples for $\mathcal{P}_{\geqslant p}(\Phi \,\mathsf{U}^{\leqslant t} \Psi)$ can be obtained by considering a formula of the form $\mathcal{P}_{\leqslant p'}(\Phi' \,\mathsf{U}^{\leqslant t}\, \Psi')$. This can be seen as follows. Extend the labels of $\mathcal{C}$ with a new

atomic proposition, at_B, say, where at_B is a new atomic proposition such that $s \models at_B$ iff (i) either $s \models \neg\Phi \wedge \neg\Psi$ (ii) or $s \in B$ where B is a bottom strongly connected component (BSCC) such that $B \subseteq Sat(\Phi \wedge \neg\Psi)$, or shortly $B_{\Phi\wedge\neg\Psi}$. A BSCC B is a maximal strong component that has no transitions leaving B. Then:

$$\mathcal{P}_{\geqslant p}(\Phi \,\mathsf{U}^{\leqslant t}\Psi) \equiv \mathcal{P}_{\leqslant 1-p}((\Phi \wedge \neg\Psi)\mathsf{W}^{\leqslant t}(\neg\Phi \wedge \neg\Psi)) \equiv \mathcal{P}_{\leqslant 1-p}((\Phi \wedge \neg\Psi)\,\mathsf{U}^{\leqslant t} at_B)$$

Intuitively, to show that the set of $(\Phi \,\mathsf{U}^{\leqslant t}\Psi)$-paths has probability $\geqslant p$, it is sufficient to show that the paths violating $\Phi \,\mathsf{U}^{\leqslant t}\Psi$ have probability $\leqslant 1-p$.

Note that for $t = \infty$, $\Phi \,\mathsf{U}^{\leqslant t}\Psi$ denotes the standard-until operator. As this operator can be verified on the embedded DTMC, counterexamples can be obtained as for DTMCs. In the sequel, we therefore consider $t \neq \infty$.

3 Evidences and counterexamples

Assume $s \not\models \mathcal{P}_{\leqslant p}(\phi)$ for CSL path-formula ϕ. Unlike in DTMCs, a timed path could not be an evidence since it always has probability 0. Instead, we consider *symbolic* evidences that represent a set of (concrete) finite timed paths satisfying ϕ. For time-abstract path ρ, let $\rho{\downarrow_k}$ denote the prefix of ρ of length k, i.e., $(s_0 s_1 \ldots){\downarrow_k} = s_0 s_1 \ldots s_k$.

Definition 6 (Symbolic evidence). *A* symbolic evidence *for $\mathcal{P}_{\leqslant p}(\phi)$ in state s is a finite time-abstract path that starts in s and minimally satisfies ϕ. Let $Paths_{abs}(s,\phi)$ denote the set of symbolic evidences starting from s for ϕ.*

Actually, a symbolic evidence for $\phi = \Phi \,\mathsf{U}^{\leqslant t}\,\Psi$ is a finite time-abstract path that goes along Φ-states and halts at the first encountered Ψ-state. A symbolic evidence for $\phi = \Phi \,\mathsf{U}^{\leqslant t}\,\Psi$ represents a set of (infinite) timed paths in the CTMC:

$$Paths_{\leqslant t}(\rho) = \{\sigma \in Paths \mid \rho = \alpha(\sigma){\downarrow_l} \wedge \sum_{i=0}^{l-1} \delta(\sigma,i) \leqslant t\} \quad \text{where } l = |\rho|.$$

The timed paths induced by ρ have a common initial state sequence, viz. ρ, and the total duration of this prefix is at most t, i.e., the last state of ρ is reached within t. We define the probability of a symbolic evidence ρ to be $\Pr_{\leqslant t}(\rho)$, and for the set C of symbolic evidences, the probability is $\Pr(C) = \sum_{\rho\in C} \Pr_{\leqslant t}(\rho)$. A *strongest* symbolic evidence is a symbolic evidence of maximal probability.

Lemma 2. *For CTMC $\mathcal{C}$ and $\phi = \Phi \,\mathsf{U}^{\leqslant t}\,\Psi$: $Prob(s,\phi) = \sum_{\rho\in Paths_{abs}(s,\phi)} \Pr_{\leqslant t}(\rho)$.*

For state s in CTMC $\mathcal{C}$ and formula $\mathcal{P}_{\leqslant p}(\phi)$ we now have:

$$s \not\models \mathcal{P}_{\leqslant p}(\phi) \quad \text{iff} \quad Prob(s,\phi) > p \quad \text{iff} \quad \sum_{\rho\in Paths_{abs}(s,\phi)} \Pr_{\leqslant t}(\rho) > p.$$

As $Paths_{abs}(s,\phi)$ only contains finite time-abstract paths, counterexamples are sets of symbolic evidences of sufficient probability mass.

Definition 7 (Symbolic counterexample). *A* symbolic counterexample *for $\mathcal{P}_{\leqslant p}(\phi)$ where $\phi = \Phi \,\mathsf{U}^{\leqslant t}\,\Psi$ is a set C of symbolic evidences for ϕ such that $\Pr(C) > p$.*

Example 2. For the CTMC $\mathcal{C}$ in Fig. 1 and CSL formula $\mathcal{P}_{\leqslant 0.45}(a \,\mathsf{U}^{\leqslant 1} b)$ the symbolic evidences are $\rho_1 = s_0 s_2 t_2$, $\rho_2 = s_0 s_1 s_2 t_2$, $\rho_3 = s_0 s_1 t_1$, and so on. These paths all satisfy $a \,\mathsf{U}\, b$. For instance, $s_0 \overset{0.5}{\rightarrow} s_1 \overset{0.25}{\rightarrow} s_2 \overset{0.05}{\rightarrow} t_2 \in Paths_{\leqslant 1}(\rho_2)$. Without specifying the details (see next section), the probabilities of the symbolic evidences are: $\Pr_{\leqslant 1}(\rho_1) = 0.24998$, $\Pr_{\leqslant 1}(\rho_2) = 0.24994$ and $\Pr_{\leqslant 1}(\rho_3) = 0.16667$. $C = \{\rho_1, \rho_2\}$ is a counterexample since $\Pr(C) > 0.45$, but $C' = \{\rho_1, \rho_3\}$ is not.

The remainder of the paper is concerned with determining (symbolic) counterexamples and symbolic evidences. As in conventional model checking, the intention is to obtain *comprehensible* counterexamples. We interpret this as counterexamples of minimal size, i.e., minimal cardinality. An algorithmic skeleton to generate such counterexamples iteratively is given below:

```
(1)  k := 1; pr := 0;
(2)  while pr ⩽ p do
(3)      determine symbolic evidence ρ^k;
(4)      compute Pr_{⩽t}(ρ^k);
(5)      pr := pr + Pr_{⩽t}(ρ^k);
(6)      k := k + 1;
(7)  od;
(8)  return (ρ^1, ..., ρ^{k-1})
```

The termination of this algorithm is guaranteed as the violation of the property has been already established prior to invoking it. Evidently, the smaller the index k, the more succinct the counterexample. The next section presents a way to determine $\Pr_{\leqslant t}(\rho)$, i.e., the probability of a symbolic evidence (cf. line (4)). In subsequent sections, we present algorithms that aim to finding probable symbolic evidences, cf. line (3) of the algorithm. Stated differently, we aim to terminating with a small value of k.

4 The likelihood of a symbolic evidence

Assume we have symbolic evidence $\rho = s_0\, s_1\, s_2 \ldots s_l$ at our disposal. The probability $\Pr_{\leqslant t}(\rho)$ of this evidence—in fact, the probability of all concrete evidences of ρ up to time t—is given by:

$$\int_0^t \left(p(s_0, s_1, t_0) \cdot \Big(\ldots \big(\int_0^{t - \sum_{i=0}^{l-2} t_i} p(s_{l-1}, s_l, t_{l-1}) \cdot dt_{l-1} \big) \ldots \Big) \right) dt_0 \qquad (1)$$

where $p(s_0, s_1, t_0)$ denotes the probability density function of $s_0 \rightarrow s_1$ winning the race at time instant t_0 in the interval $[0, t]$. The corresponding probability is thus derived by the outermost integral. Suppose the transition $s_0 \rightarrow s_1$ takes place at time instant t_0. Then the possible time instant for the second transition $s_1 \rightarrow s_2$ to take place is in $[0, t - t_0]$. This determines the range of the second outermost integral. The rest is likewise. The innermost integral determines the residence time in state s_{l-1}, the one-but-last state in ρ.

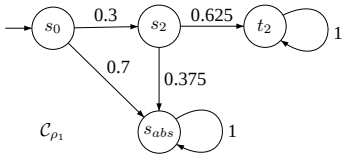
To avoid computing this (somewhat involved) integral directly by numerical techniques we resort to a simpler technique. The main idea is to isolate the time-abstract path ρ from the entire CTMC. This yields a simple acyclic CTMC, i.e., an acyclic phase-type distribution [16] which can be solved either analytically or numerically.

Transformation into an acyclic CTMC. As a first step, we transform the CTMC as suggested in [4]. (The same strategy was applied in Section 2 for DTMCs prior to applying SP algorithms.) Consider CTMC $\mathcal{C}$ and CSL path-formula $\phi = \Phi \,\mathsf{U}^{\leqslant t}\, \Psi$. All Ψ-states as well as all $(\neg\Phi \wedge \neg\Psi)$-states are made absorbing in $\mathcal{C}$, i.e., their outgoing transitions are replaced by a self-loop. It is not difficult to establish that the validity of $\mathcal{P}_{\leqslant p}(\phi)$ remains invariant under this modification. In the rest of the paper, CTMCs are assumed to have been subject to this transformation.

Definition 8 (CTMC induced by symbolic evidence). *Let CTMC $\mathcal{C} = (S, \boldsymbol{P}, E, L)$ and $\rho = s_0\, s_1 \ldots s_l$ a symbolic evidence in CTMC $\mathcal{C}$ in which all states are pairwise distinct* [1]. *The CTMC $\mathcal{C}_\rho$ induced by ρ on $\mathcal{C}$ is defined by: $\mathcal{C}_\rho = (S_\rho, \mathbf{P}_\rho, E_\rho, L_\rho)$ with:*

- $S_\rho = \{s_0, \ldots, s_l, s_{abs}\}$ *with* $s_{abs} \notin S \cup \{s_0, \ldots, s_l\}$,
- $\mathbf{P}_\rho(s_i, s_{i+1}) = \mathbf{P}(s_i, s_{i+1})$, $\mathbf{P}_\rho(s_i, s_{abs}) = 1 - \mathbf{P}_\rho(s_i, s_{i+1})$ *for* $0 \leqslant i < l$ *and* $\mathbf{P}_\rho(s, s) = 1$ *for* $s = s_l$ *or* $s = s_{abs}$
- $E_\rho(s_i) = E(s_i)$ *and* $E_\rho(s_{abs}) = 0$ *and* $L_\rho(s_i) = L(s_i)$ *and* $L_\rho(s_{abs}) = \{\mathit{abs}\}$.

Stated in words, $\mathcal{C}_\rho$ is the CTMC obtained from $\mathcal{C}$ by incorporating all states in ρ, and deleting all outgoing transitions from these states except $s_i \to s_{i+1}$. The total probability mass of these omitted transitions becomes the probability to move to the trap state s_{abs}. It follows directly that $\mathcal{C}_\rho$ is acyclic when ignoring the self-loops of the absorbing states.



Example 3. Consider CTMC $\mathcal{C}$ in Fig. 1 and symbolic evidence $\rho_1 = s_0\, s_2\, t_2$. The induced CTMCs $\mathcal{C}_{\rho_1}$ is shown on the left.

The following result states that computing the probability of symbolic evidence ρ boils down to a (standard) transient analysis of the induced CTMC by ρ.

Lemma 3. *For CTMC $\mathcal{C}$ and symbolic evidence ρ for $\phi = \Phi \,\mathsf{U}^{\leqslant t}\, \Psi$:*

$$\Pr^{\mathcal{C}}_{\leqslant t}(\rho) \;=\; \pi^{\mathcal{C}_\rho}(s, s_l, t)$$

where $\pi^{\mathcal{C}_\rho}(s, s_l, t)$ is the transient probability of state s_l, the last state of ρ, at time t under the condition that $\mathcal{C}_\rho$ started in s.

This result enables us to exploit well-known algorithms for the transient analysis of CTMCs to determine the likelihood of a symbolic evidence. In fact, as CSL model checking of time-bounded until-formulas is reduced to transient analysis (see [4]), the desired likelihood can be determined by verifying the property $\Diamond^{\leqslant t}\, \mathit{at}_{s_l}$ on the CTMC $\mathcal{C}_\rho$. (Here, at_{s_l} is an atomic proposition that only holds in state s_l.) This yields an approximate solution up to an a priori user-defined accuracy and is part of the standard machinery in model checkers such as PRISM [14] and MRMC [13]. Alternatively, we can exploit the fact that $\mathcal{C}_\rho$ is acyclic (ignoring the self-loops at the absorbing states) and use the closed-form expression for transient distributions in acyclic CTMCs as proposed by Marie *et al.* [15]. This yields an exact solution.

[1] This is not a restriction since it is always possible to rename a state along ρ while keeping e.g. its exit rate and its labeling the same.

5 A first attempt to find probable symbolic evidences

It remains to clarify how symbolic evidences can be obtained and how to obtain them in such a way that small counterexamples result. As symbolic evidences are just state sequences, the first idea is to adapt the strategy for DTMCs [10], cf. Section 2. That is, the CTMC under consideration is discretized. This is done using uniformization [12], a technique to transform a CTMC into a DTMC whose transient behaviour is equal (up to some accuracy ε) [2]. k-SP algorithms are then exploited to obtain symbolic evidences in ascending order of likelihood (in the obtained DTMC). k is determined on-the-fly as the minimal natural number such that $\sum_{i=1}^{k} \Pr_{\leqslant t}(\rho^i) > p$ where p is the lower bound of the property that is refuted. Let us first briefly present uniformization.

Uniformization (also known as Jensen's method or randomization) [12] is a well-known method for computing the transient probabilities of a CTMC at specific time t. Its formulation involves construction of a DTMC and Poisson process from an original CTMC. Uniformization is attractive because of its excellent numerical stability and the fact that the computational error is well-controlled and can be specified in advance.

For CTMC $\mathcal{C} = (S, \mathbf{P}, E, L)$, the uniformized DTMC is $\mathcal{U} = unif(\mathcal{C}) = (S, \mathbf{U}, L)$, where $\mathbf{U}$ is defined by $\mathbf{U} = \mathbf{I} + \frac{\mathbf{Q}}{q}$ with $q \geqslant \max_i\{E(s_i)\}$ and $\mathbf{Q} = \mathbf{R} - diag(\underline{E})$. For the special case $q = 0$, $\mathbf{U}(s, s) = 1$ for any $s \in S$. In the rest of the paper, we always use $\mathcal{U}$ to denote $unif(\mathcal{C})$. The uniformization rate q can be chosen to be any value exceeding the shortest mean residence time. All rates in the CTMC are normalized with respect to q. For each state s with $E(s) = q$, one epoch in the uniformized DTMC corresponds to a single exponentially distributed delay with rate q, after which one of its successor states is selected probabilistically. As a result, such states have no additional self-loop in the DTMC. If $E(s) < q$, i.e., state s has, on average, a longer state residence time than $\frac{1}{q}$, one epoch in the DTMC might not be "long enough"; hence, in the next epoch, these states might be revisited with some positive probability. This is represented by equipping these states with a self-loop with probability $1 - \frac{E(s)}{q} + \frac{\mathbf{R}(s,s)}{q}$.

Remark 2 (Self-loops). As a CTMC is assumed to have no self-loops on non-absorbing states, all self-loops in the uniformized DTMC are caused by uniformization.

After uniformization, the vector of state probabilities $\underline{\pi}^{\mathcal{C}}(t)$ at time t, namely the *transient probability* vector, is computed as:

$$\underline{\pi}^{\mathcal{C}}(t) = \alpha_0 \cdot \sum_{i=0}^{\infty} PP(i, qt)\mathbf{U}^i = \sum_{i=0}^{\infty} PP(i, qt)\underline{\pi}^{\mathcal{U}}(i), \tag{2}$$

where $PP(i, qt) = e^{-qt}\frac{(qt)^i}{i!}$ is the ith Poisson probability that i epochs occur in $[0, t]$ when the average rate is $\frac{1}{qt}$ and $\underline{\pi}^{\mathcal{U}}(i)$ is the state probability distribution vector after i epochs in $\mathcal{U}$ with transition matrix $\mathbf{U}$ determined recursively by $\underline{\pi}^{\mathcal{U}}(i) = \underline{\pi}^{\mathcal{U}}(i-1) \cdot \mathbf{U}$ with the initial distribution $\underline{\pi}^{\mathcal{U}}(0) = \alpha_0$.

[2] An alternative discretization is to use the embedded DTMC, but as this does not involve any timing aspects, this is senseless.

The Poisson probabilities can be computed in a stable way with the Fox-Glynn algorithm [8], thus avoiding numerical instability. The infinite summation problem is solved by introducing a required accuracy ε, so that $\|\underline{\pi}^{\mathcal{C}}(t) - \tilde{\underline{\pi}}^{\mathcal{C}}(t)\| \leqslant \varepsilon$, where $\tilde{\underline{\pi}}^{\mathcal{C}}(t) = \sum_{i=0}^{N_\varepsilon(t)} PP(i, qt) \cdot \underline{\pi}^{\mathcal{U}}(i)$ is the approximation of $\underline{\pi}^{\mathcal{C}}(t)$ and $N_\varepsilon(t)$ is the number of terms to be taken in Equation (2) for time t, which is the smallest value satisfying:

$$\sum_{i=0}^{N_\varepsilon(t)} \frac{(qt)^i}{i!} \geqslant \frac{1-\varepsilon}{e^{-qt}} = (1-\varepsilon) \cdot e^{qt}. \tag{3}$$

If qt is larger, $N_\varepsilon(t)$ tends to be of the order $\mathcal{O}(qt)$.

Let θ denote a path in $\mathcal{U}$, $Paths^{\mathcal{U}}$ denote the set of all paths in $\mathcal{U}$ and $Paths^{\mathcal{U}}(s)$ the paths in $\mathcal{U}$ starting in s.

Model transformation. Given a CTMC $\mathcal{C}$ and a CSL formula $\phi = \Phi \,\mathsf{U}^{\leqslant t}\, \Psi$, we take the uniformized DTMC $\mathcal{U}$ of $\mathcal{C}$ and remove all its self-loops. The resulting DTMC is referred to as $\mathcal{U}^{\otimes}$, which is an FPS instead of a DTMC. If $\mathcal{U}^{\otimes}$ would be normalized, we obtain the embedded DTMC of $\mathcal{C}$. The probability in the embedded DTMC only considers the race of transitions after the delay, while the probability in $\mathcal{U}^{\otimes}$ takes delays into consideration. We remove self-loops in $\mathcal{U}$ as many paths in $\mathcal{U}$ correspond to the same time-abstract path in $\mathcal{C}$. Every path in $\mathcal{U}^{\otimes}$ is a time-abstract path in $\mathcal{C}$ and satisfies ϕ. Besides, the information of the self-loops (viz., delays) can be recovered easily by taking the difference between the total probability of a state and one.

Algorithm by pure graph analysis. For $s \not\models \mathcal{P}_{\leqslant p}(\phi)$, a counterexample can be computed as follows: The k most probable paths in $\mathcal{U}^{\otimes}$ are computed, each corresponding to a symbolic evidence in $\mathcal{C}$, i.e., symbolic evidences are computed in such an order $\rho^1, \rho^2, ..., \rho^k$ that $\Pr\{\rho^1\} \geqslant \Pr\{\rho^2\} \geqslant ... \geqslant \Pr\{\rho^k\}$. k is determined on the fly, as the smallest number such that $\sum_{i=1}^{k} \Pr_{\leqslant t}(\rho^i) > p$. The k most probable paths problem can be reduced to k-SP problem by the standard transformation in Section 2 which also applies to FPS $\mathcal{U}^{\otimes}$. The resulting algorithm becomes:

(1) $k := 1$; $pr := 0$;
(2) **while** $pr \leqslant p$ **do**
(3) determine symbolic evidence ρ^k as the k-th most probable path in $\mathcal{U}^{\otimes}$;
(4) compute $\Pr_{\leqslant t}(\rho^k)$;
(5) $pr := pr + \Pr_{\leqslant t}(\rho^k)$;
(6) $k := k + 1$;
(7) **od**;
(8) **return** $(\rho^1, \ldots, \rho^{k-1})$

The time complexity for computing the k most probable paths is as the k-SP problem, cf. [7], $\mathcal{O}(m+n\log n+k)$. The transformation from ρ to $\mathcal{C}_\rho$ takes $\mathcal{O}(|\rho|)$ time. It takes $\mathcal{O}(|\rho|qt)$ to compute the probability of a symbolic evidence ρ, where $\mathcal{O}(qt)$ is the number of terms before truncation (i.e., $N_\varepsilon(t)$, cf. [4]) and $\mathcal{O}(|\rho|)$ time is needed for vector-vector multiplication. There are k symbolic evidences, which gives rise to the total time complexity $\mathcal{O}(m + n\log n + k|\rho|qt)$.

In most of the cases, probable paths in $\mathcal{U}^{\otimes}$ correspond to probable symbolic evidences in $\mathcal{C}$. However, this is not always the case, since the time bound in the property is not considered. In particular, this approach does not guarantee $\Pr_{\leqslant t}(\rho^i) \geqslant \Pr_{\leqslant t}(\rho^j)$ for $i < j$. An example is given as follows:

Example 4. Consider our running example. The uniformized DTMC $\mathcal{U}$ is illustrated on the left. The uniformization rate is chosen as $q = E(s_1) = 20$, since s_1 has the largest exit rate. For symbolic evidences $\rho_2 = s_0 s_1 s_2 t_2$ and $\rho_3 = s_0 s_2 t_1$ of Example 3, the probabilities in $\mathcal{U}^{\otimes}$ are $\Pr\{\rho_2\} = 0.100$ and $\Pr\{\rho_3\} = 0.045$, respectively. For CSL path formula $\phi = a \,\mathsf{U}^{\leqslant 1} b$, $\Pr_{\leqslant 1}(\rho_2) = 0.24994$ and $\Pr_{\leqslant 1}(\rho_3) = 0.16362$. For $\phi' = a \,\mathsf{U}^{\leqslant 0.1} b$, $\Pr_{\leqslant 0.1}(\rho_2) = 0.04478$ and $\Pr_{\leqslant 0.1}(\rho_3) = 0.06838$. Thus, for $t = 1$, $Paths_{\leqslant 1}(\rho_2)$ is more probable than $Paths_{\leqslant 1}(\rho_3)$, whereas for $t = 0.1$, the reverse holds.

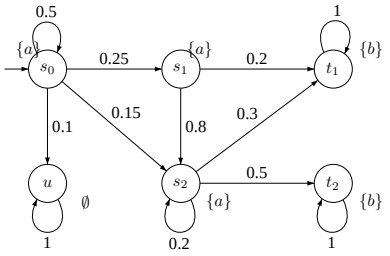


Fig. 2. $unif(\mathcal{C}) = \mathcal{U}$

This implies that for symbolic evidences ρ and ρ' and arbitrary time bound t, $\Pr\{\rho\} > \Pr\{\rho'\}$ cannot guarantee that $\Pr_{\leqslant t}(\rho) > \Pr_{\leqslant t}(\rho')$. A direct consequence is that the algorithm might terminate with a large value of k. The counterexamples may thus be less comprehensive, because evidences with large probability might not be included. The algorithm in the next section attempts to overcome this problem by taking the time bound into account.

6 Involving time bounds

The previous algorithm ignores the time bound t in determining the order of generating the symbolic evidences $\rho^1, \rho^2, ...$ By definition, however, every transition in the uniformized DTMC $\mathcal{U}$ takes $\frac{1}{q}$ time units. In fact, using the Poisson probabilities we can determine the probability of path θ in $\mathcal{U}$ to have a duration of at most t:

Definition 9 ([17]). *Given a CTMC $\mathcal{C}$, for $\theta \in Paths^{\mathcal{U}}$ with $|\theta| = l$ and $t \in \mathbb{R}_{>0}$, the probability of θ occurring in $[0, t]$ with rate q is defined as:*

$$\Pr_{\leqslant t}(\theta, qt) = PP(l, qt) \cdot \Pr^{\mathcal{U}}\{\theta\}.$$

Intuitively, given that $|\theta|$ transitions occur in the interval $[0, t]$, the likelihood of θ occurring in $\mathcal{U}$ is $\Pr^{\mathcal{U}}\{\theta\}$. As $\mathcal{U}$ is a DTMC, $\Pr^{\mathcal{U}}\{\theta\}$ is simply $\prod_{i=0}^{|\theta|-1} \mathbf{U}(s_i, s_{i+1})$ for $\theta = s_0 s_1 s_2 ....$

It remains to establish a connection between $\Pr_{\leqslant t}(\rho)$ and the probabilities obtained in the uniformized DTMC $\mathcal{U}$, i.e., $\Pr_{\leqslant t}(\theta, qt)$, where θ relates to ρ. This can be done as follows. Consider symbolic evidence ρ, say of length l. Paths in $\mathcal{U}$ that correspond to $\rho = s_0 s_1 ... s_l$ visit the same state sequence $s_0 s_1 ... s_l$ but may take the self-loop in s_i zero or more times. Recall that the purpose of this self-loop is to mimic the probability for the CTMC to reside longer in s_i. The set of paths in $\mathcal{U}$ that correspond to (or can mimic) ρ is defined by:

$$mimic(\rho) = \{s_0^{n_0} s_1^{n_1} ... s_l^{n_l} \in Paths^{\mathcal{U}} \mid n_i > 0 \text{ for } 0 \leqslant i \leqslant l\},$$

where $l = |\rho|$ and $s_0^{n_0}$ is short for the n_0-time replication of s_0. Then:

$$\Pr^{\mathcal{C}}_{\leqslant t}(\rho) = \sum_{\theta \in mimic(\rho)} \Pr^{\mathcal{U}}_{\leqslant t}(\theta, qt) = \sum_{i=|\rho|}^{\infty} PP(i, qt) \cdot \sum_{\theta \in mimic(\rho) \wedge i = |\theta|} \Pr^{\mathcal{U}}\{\theta\}$$

Note the similarity to Equation (2). The intuition is also similar: given a symbolic evidence ρ of $\mathcal{C}$, there are paths in $\mathcal{U}$ that can mimic ρ. These paths can have $i(=|\rho|)$ hops, $i+1$ hops, and so forth. The extra hops are self-loops in $\mathcal{U}$ which simulate the longer residence time in a state in $\mathcal{C}$.

To truncate the infinite summation, which lengths i do we need to consider? A natural criterion for this is fortunately provided by the uniformization process. As the probability of any path longer than $N_\varepsilon(t)$ is negligible – given an accuracy ε – this suggests to only consider paths up to length $N_\varepsilon(t)$. By taking this approach, it is guaranteed that the total probability mass of the not considered paths is less than ε.

An algorithm involving time. In the following, we give an algorithm that determines paths in a decreasing order with respect to $\Pr_{\leqslant t}(\theta, qt)$. Since we are interested in paths without self-loops, we consider paths in $\mathcal{U}^{\otimes}$.

Let ϖ_h^j denote the j-th most probable path in $\mathcal{U}^{\otimes}$ of h hops, i.e. $\Pr\{\varpi_h^j\} \geqslant \Pr\{\varpi_h^{j+1}\}$. Since the Poisson probability is fixed for a given h, $\Pr_{\leqslant t}(\varpi_h^j, qt) \geqslant \Pr_{\leqslant t}(\varpi_h^{j+1}, qt)$. Let τ^k denote the path in $\mathcal{U}^{\otimes}$ with k-th largest probability $\Pr_{\leqslant t}(\tau^k, qt)$. Then:

$$\tau^k = \arg\max_{\theta} \Big\{ \Pr_{\leqslant t}(\theta, qt) \mid \theta \in Q^k \Big\}, \tag{4}$$

where Q^k is the candidate path set defined as:

$$Q^k = \begin{cases} \Big\{ \varpi_h^1 \mid 0 \leqslant h \leqslant N_\varepsilon(t) \Big\} & \text{if } k = 1 \\ \Big(Q^{k-1} - \{\tau^{k-1}\} \Big) \cup \Big\{ \varpi_h^{j+1} \Big\} & \text{if } k > 1 \text{ and } \tau^{k-1} = \varpi_h^j \end{cases}$$

where j and h are the index and path length of τ^{k-1}, the previous path computed.

The algorithm starts with a "candidate" path set Q^1 which contains all ϖ_h^1 paths, the most probable path of length h, for $0 \leqslant h \leqslant N_\varepsilon(t)$. τ^1 is picked out as the one with the maximal probability in Q^1, according to Equation (4). To compute the next evidence τ^2, Q^2 is computed on the basis of Q^1. As $\varpi_{l^*}^1$ has been removed from Q^1, where $l^* = |\tau^1|$, another path of exactly l^* hops replaces it. This new path is $\varpi_{l^*}^2$, i.e., the second most probable path with the same length l^* as τ^1. Then τ^2 can be picked from Q^2. Since each path in $\mathcal{U}^{\otimes}$ is an evidence in $\mathcal{C}$, the algorithm will terminate when the probability of the first k evidences exceeds p.

Candidate paths are stored in a priority queue pq sorted by the keys $\Pr_{\leqslant t}(\varpi_h^j, qt)$. The *enqueue* function inserts a new path to its proper position and the *dequeue* function returns the pair (h, j) of the corresponding path with the highest probability in pq. Function $\varpi(h, j, qt)$ computes the j-th h-hop most probable path ϖ_h^j, which can be reduced to computing j-th shortest h-hop path in $\mathcal{G}_{\mathcal{U}^{\otimes}}$ and can be solved by adapted REA, see [10] for more details.

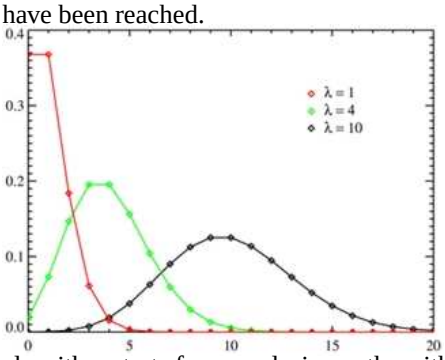
(1) $k := 0$; $pr := 0$; $h := 0$; PriorityQueue pq;
(2) **for** $h := 0$ **to** $N_\varepsilon(t)$ **do** $pq.enqueue(\varpi(h, 1, qt))$; **od**; $\backslash * \; Q^1 \; * \backslash$
(3) **while** $pr \leqslant p$ **do** $\backslash * \; Q^2$ to $Q^k \; * \backslash$
(4) $(h', j') := pq.dequeue()$; $k := k + 1$; $\rho^k := \varpi_{h'}^{j'}$;
(5) $\varpi := \varpi(h', j' + 1, qt)$; $pq.enqueue(\varpi)$; $pr := pr + \Pr_{\leqslant t}(\varpi)$; **od**;
(6) **return** $(\rho^1, \ldots, \rho^{k-1})$;

Note that the resulting evidence sequence $\rho^1, \rho^2, \ldots$ coincides with $\tau^1, \tau^2, \ldots$.

Time complexity. The time complexity for computing Q^1 is $\mathcal{O}(q^2t^2m)$, since there are $N_\varepsilon(t)+1$ most probable paths to compute and the time to compute one probable path is $\mathcal{O}(qtm)$. Note that $N_\varepsilon(t)$ is linear in $\mathcal{O}(qt)$ [4]. To compute ρ^k, there are at most $N_\varepsilon(t)$ paths in Q^k, so it takes $\mathcal{O}(qt\log(qt))$ time [10]. There are $k-1$ such paths (ρ^2 through ρ^k) to be computed. This yields a total time complexity $\mathcal{O}(q^2t^2m + kqt\log(qt))$.

A refined algorithm. The above algorithm will generate a sequence of evidences $\rho^1, ...\rho^k$ by the decreasing order of their probability product $\Pr_{\leqslant t}(\rho^i, qt)$. However, $N_\varepsilon(t)$ is usually large (typically, a few hundred or thousand) yielding a large set Q^1. As a result, the above algorithm is costly. We now suggest a heuristic to improve this strategy. The basic idea is to use the Poisson probability function to obtain smaller counterexamples.

Observation 1: We first notice that for fixed qt, the Poisson probability $PP(h, qt)$ is maximal when $h = \lceil qt \rceil$ or $h = \lfloor qt \rfloor$, which is the expectation of h. If $h < \lceil qt \rceil$, $PP(h, qt)$ is monotonically increasing and if $h > \lfloor qt \rfloor$, $PP(h, qt)$ is monotonically decreasing, cf. the figure below where the horizontal axis is the hop count h. The function is only non-zero at integer values of h. The connecting lines do not indicate continuity. This observation justifies the heuristics that we start from the crest of the function ($h = \lceil qt \rceil$) and proceed in two directions, in which way the hop counts for larger Poisson probabilities are explored first, as a consequence, the probability product will usually be large. This bidirectional increments will stop when the bounds 0 and $N_\varepsilon(t)$ have been reached.



Observation 2: When the value qt is small, the Poisson probability is almost monotonically decreasing, cf. case $\lambda = 1$ in the figure. Then $h = \lceil qt \rceil$ is not suitable as the starting point any more.

Let $\varpi^1_{l^*}$ be the most probable path in $\mathcal{U}^\otimes$. It means that paths with $h \neq l^*$ have less or equal probability than $\varpi^1_{l^*}$. Therefore, $h = l^*$ is also considered as a starting point. Due to Observation 1 and 2, our algorithm starts from exploring paths with $H_0 = \max\{\lceil qt \rceil, l^*\}$.

We use the priority queue pq to keep track of the paths which have been explored but not yet expanded. A path is "explored" when its probability is computed and added to the total counterexample probability. Note that every path that is explored is already taken as an evidence. This is different from the previous algorithm where we might explore many more paths (the huge basic set Q^1) than actually needed. That also partly explains why this algorithm is more efficient. A path is called "expanded" when it is dequeued from pq and its successor is computed. When a path is dequeued, it means that it has the largest probability product among all the paths in the queue; and this fact makes the expansion reasonable.

New path(s) or evidence(s) will be added to the counterexample in each iteration. The increments are in two dimensions. In one dimension, we have to increase the index of some most probable path. More specifically, we dequeue the path ϖ^j_h with the highest probability $\Pr_{\leqslant t}(\varpi^j_h, qt)$ from pq, and add its successor ϖ^{j+1}_h, namely the $(j+1)$-st

most probable path with the same hop count. This happens in each iteration. In the other dimension, the minimal and maximal number of hops of paths are incremented. We use $H_{\min}$ and $H_{\max}$ to denote the minimal and maximal hop counts explored so far. Two more new paths with $H_{\min}-1$ and $H_{\max}+1$ hops are added, namely, $\varpi^1_{H_{\min}-1}$ and $\varpi^1_{H_{\max}+1}$ when the bounds 0 and $N_\varepsilon(t)$ have not yet been reached. The more iterations the algorithm proceeds, the farther away $H_{\min}$ and $H_{\max}$ are from H_0 and the less Poisson probability the path will have.

The sketch of the improved algorithm is shown as follows:

(1) Compute most probable path $\varpi^1_{l^*}$ in $\mathcal{U}^{\otimes}$; * Initialization: *\
(2) $pr := \Pr_{\leqslant t}(\varpi^1_{l^*})$; PriorityQueue $pq.enqueue(\varpi^1_{l^*})$;
(3) $H_{\min} := H_{\max} := \max\{\lceil qt \rceil, l^*\}$; $k := 1$; $\rho^k = \varpi^1_{l^*}$;
(4) **while** $pr \leqslant p$ **do** * Main body: *\
(5) $(h', j') := pq.\,dequeue()$; $\varpi_1 := \varpi(h', j'+1, qt)$; * Increments on j *\
(6) $pq.enqueue(\varpi_1)$; $pr := pr + \Pr_{\leqslant t}(\varpi_3)$; $\rho^k := \varpi_1$; $k := k+1$;
(7) **if** $H_{\min} > 0$ **then** * Decrease of hop count *\
(8) $H_{\min} := H_{\min} - 1$; $\varpi_2 := \varpi(H_{\min}, 1, qt)$; $pq.enqueue(\varpi_2)$;
(9) $pr := pr + \Pr_{\leqslant t}(\varpi_2)$; $\rho^k := \varpi_2$; $k := k+1$;
(10) **if** $H_{\max} < N_\varepsilon(t)$ **then** * Increase of hop count: *\
(11) $H_{\max} := H_{\max} + 1$; $\varpi_3 := \varpi(H_{\max}, 1, qt)$; $pq.enqueue(\varpi_3)$;
(12) $pr := pr + \Pr_{\leqslant t}(\varpi_3)$; $\rho^k := \varpi_3$; $k := k+1$; **od**;
(13) **return** $(\rho^1, ..., \rho^{k-1})$;

Note that $\varpi^1_{l^*}$ in Line (1) can be computed by SP algorithms, say Dijkstra's [6], in $\mathcal{G}_{\mathcal{U}^{\otimes}}$.

7 Conclusion

Comparison of algorithms. This paper presented a set of approximate algorithms for computing small sets of paths that indicate the violation of time-bounded constrained reachability probabilities. The algorithm involving time bounds for computing informative evidences considers Poisson probability besides the probability of paths themselves, which characterizes the significance of the paths in $\mathcal{U}$, thus provides a clue of the significance of the corresponding evidences in $\mathcal{C}$. As we mentioned, the first algorithm of pure graph analysis *cannot* guarantee that for the sequence of paths that computed by our algorithm in order: $\rho^1, ..., \rho^k$, it holds that $\Pr_{\leqslant t}(\rho^1) \geqslant ... \geqslant \Pr_{\leqslant t}(\rho^k)$. Unfortunately, the one involving time also cannot guarantee this, however, it *can* guarantee that $\Pr_{\leqslant t}(\rho^1, qt) \geqslant ... \geqslant \Pr_{\leqslant t}(\rho^k, qt)$ which is usually very close to the target sequence. The refined algorithm exploits the monotonicity of the Poisson probability function to obtain small counterexamples. Experimental research of the proposed algorithms is to be carried out as the future work.

Related work. Aljazzar et al. [1][2] applied directed explicit-state search algorithms to determine a set of diagnostic traces which carry large amount of probability. Their algorithms are guided by heuristics which exploit stochastic information on the traces. In contrast, we have proposed several algorithms according to different levels of knowledge about the CTMCs, which to some extent shows the significant role of probability and time. The uniformization technique discretizes the continuous-time setting and

makes the efficient algorithms for DTMC counterexample-generation [10] adaptable here. Moreover, the analysis and utilization of Poisson probability distribution gives rise to an almost decreasing order of the evidence probabilities, which enables the incremental exploration of the candidate evidence set.

Acknowledgment. Holger Hermanns and Boundewijn R. Haverkort are thanked for useful discussions. This research has been performed as part of the QUPES project that is financed by the Netherlands Organization for Scientific Research (NWO).

References

1. H. Aljazzar, H. Hermanns and S. Leue. Counterexamples for timed probabilistic reachability. FORMATS, LNCS 3829: 177-195, 2005.
2. H. Aljazzar and S. Leue. Extended directed search for probabilistic timed reachability. FORMATS, LNCS 4202: 33-51, 2006.
3. A. Aziz, K. Sanwal, V. Singhal and R.K. Brayton. Model-checking continous-time Markov chains. *ACM Trans. Comput. Log.* 1(1): 162-170 (2000).
4. C. Baier, B.R. Haverkort, H. Hermanns and J.-P. Katoen. Model-checking algorithms for continuous-time Markov chains. *IEEE Trans. Softw. Eng.* 29(6): 524-541 (2003).
5. E.M. Clarke, S. Jha, Y. Lu and H. Veith. Tree-like counterexamples in model checking. LICS: 19-29 (2002).
6. E.W. Dijkstra. A note on two problems in connection with graphs. *Num. Math.*, 1:395-412 (1959).
7. D. Eppstein. Finding the k shortest paths. *SIAM J. Comput.* 28(2): 652-673 (1998).
8. B.L. Fox and P.W. Glynn. Computing Poisson probabilities. *Comm. ACM*, vol.31, no. 4, pp. 440-445, 1988.
9. A. Gurfinkel and M. Chechik.Proof-like counter-examples. TACAS, LNCS 2619: 160-175, 2003.
10. T. Han and J.-P. Katoen. Counterexamples in probabilistic model checking. TACAS, LNCS 4424:72–86, 2007.
11. H. Hansson and B. Jonsson. A logic for reasoning about time and reliability. *Formal Asp. Comput.* 6(5): 512-535 (1994).
12. A. Jensen. Markoff chains as an aid in the study of Markoff processes. Skand. Aktuarietidskrift, vol. 36, pp. 87-91, 1953.
13. J.-P. Katoen, M. Khattri and I. S. Zapreev. A Markov reward model checker. QEST 2005, IEEE Computer Society: 243-244 (2005).
14. M.Z. Kwiatkowska, G. Norman and D. Parker. PRISM 2.0: A tool for probabilistic model checking. QEST 2004, IEEE Computer Society: 322-323 (2004).
15. R.A. Marie, A.L. Reibman and K.S. Trivedi. Transient analysis of acyclic Markov chains. *Perform. Eval.* 7(3): 175-194 (1987).
16. M.F. Neuts. *Matrix-Geometric Solutions in Stochastic Models: An Algorithmic Approach.* The Johns Hopkins Univ. Press, 1981.
17. M. Qureshi and W. Sanders. A new methodology for calculating distributions of reward accumulated during a finite interval. FTCS, IEEE Computer Society: 116-125 (1996).
18. S. Shoham and O. Grumberg. A game-based framework for CTL counterexamples and 3-valued abstraction-refinement. CAV, LNCS 2725: 275-287, 2003.