Continuous-Time Markov Chain with Alarms analysis and comparaison of tools *

Thomas Mari[1 [0000−1111−2222−3333]], Second Author[2,3 [1111−2222−3333−4444]], and Third Author[3 [2222−−3333−4444−5555]]

[1] Princeton University, Princeton NJ 08544, USA
[2] Springer Heidelberg, Tiergartenstr. 17, 69121 Heidelberg, Germany
lncs@springer.com
http://www.springer.com/gp/computer-science/lncs
[3] ABC Institute, Rupert-Karls-University Heidelberg, Heidelberg, Germany
{abc,lncs}@uni-heidelberg.de

Abstract. Phase-type fitting remains the only way of modeling non-Markovian distributions within PRISM model checker. When analyzing models with phase-type distributions, it is problematic to obtain results of sufficient precision within reasonable time. We ran experimental computations and deduced a reliable way of obtaining precise analysis results for phase-type fitted PRISM CTMCs. Lastly, we hint at an entirely different (and arguably better) approach to handling non-Markovian distributions.
The abstract should briefly summarize the contents of the paper in 150–250 words.

Keywords: PRISM model checker · CTMC · deterministic timeout · phase-type distribution · modeling · analysis · Second keyword · Another keyword.

1 Introduction

PRISM [1] and Storm [2] are popular tools for modeling and analysis of stochastic systems in continuous time. They use efficient algorithms for analysis of continuous time Markov chains (CTMC). This approach suffers from a severe restriction of modeling power — the time between transitions must be exponentially distributed. This restriction can be remedied by the use of phase-type distributions, which can approximate any general distribution with arbitrary accuracy by only using exponential distributions [3]. However, the use of phase-type distributions drastically increases the number of states within the CTMC.

In this paper, we experimentally evaluate the precision of the result and required computation time of various approaches to analysis of continuous time stochastic models with deterministic transitions (timeouts). The obtained results are then compared against results of our extension of PRISM capable of efficiently analyzing CTMC with non-Markovian alarms (ACTMC).

* Supported by organization x.

1.1 Time distribution

The model we use are probabilistic continuous time model, which means that the time passed in a state will depend on a time distribution. A time distribution can be depicted by its cumulative distribution function (CDF) $F : \mathbb{R} \to [0, 1]$, $F(t)$ is the probability that the event will happend before the time t. The exponential distribution of rate λ has a CDF $F(t) = 1 - \exp(-\lambda t)$, it is important to keep in mind that a rate refers to the exponential distribution. The Dirac distribution of timeout τ has a CDF $F(t) = \chi_{[\tau,+\infty[}(t) = \begin{cases} 0 \text{ if } t < \tau \\ 1 \text{ else} \end{cases}$. In this paper, time distribution which are not exponential will be called non-Markovian distribution.

1.2 CTMC

A continuous-time Markov chain (CTMC) is a triple $C = (S, Q, s_{in})$, where

- S is a finite set of states
- $Q : S \times S \to \mathbb{R}_{\geq 0}$ is a matrix of rates such that $\sum_{s' \in S} Q(s, s') > 0$ for each $s \in S$
- $s_{in} \in S$ is an initial state

$Q(s, s')$ is the rate of the transition from s to s'. We define the exit rate of state s as $\lambda_s = \sum_{s' \in S} Q(s, s')$

A run of a CTMC $\mathcal{C}$ is an infinite alternating sequence of states and times $\omega = s_0 t_0 s_1 t_1 ...$ where

- $s_0 = s_{in}$
- s_i is the i-th state
- t_i is the time spent in s_i

For each $i \in \mathbb{N}$

- t_i is choosen randomly according to the exponential distribution with rate λ_{s_i}
- s_{i+1} is choosen according to the discrete distribution $\dfrac{Q(s_i, .)}{\lambda_{s_i}}$

1.3 ACTMC

Continuous-Time Markov Chain with Alarms (ACTMC)s. I won't give the formal semantics which might not fit in this paper. We can see ACTMC as CTMC with some events called alarms that are enabled in disjoints sets of states and those events has their own timer which is active only when inside the enabled set and happend according to some continuous time distribution. Those alarms works concurrently with the exponential transition seen in the CTMC.

$\mathcal{A} = (S, Q, s_{in}, A, \langle S_a \rangle, \langle P_a \rangle, \langle F_a \rangle)$ where :

- (S, Q, s_{in}) is a CTMC

- A is a set of alarms
- $\langle S_a \rangle = (S_a)_{a \in A}$, the set of states where a is enabled
  - if $a \neq a'$ then $S_a \bigcap S_{a'} = \emptyset$
- $\langle P_a \rangle = (P_a)_{a \in A}$ where P_a is a probability matrix
- $\langle F_a \rangle = (F_a)_{a \in A}$ where F_a is a cumulative distribution function (CDF)

Operational Behavior

A run of a ACTMC $\mathcal{A}$ is an infinite alternating sequence of states and times $(s_0, \eta_0)t_0(s_1, \eta_1)t_1...$ where

- $s_0 = s_{in}$
- s_i is the i-th state
- t_i is the delay in s_i, t_i is the time spent in s_i without alarms
- η_i is the value of the timer, the remaining time for the alarm to ring.

We define $S_{off} = S \setminus \bigcup_{a \in A} S_a$ the set of states where no alarms are enabled

For initialization

- $s_0 = sin$
- $\eta_0 = \begin{cases} \infty \text{ if } s_0 \in S_{off} \\ \text{randomly choosen according to } F_a \text{ if } s_0 \in S_a \end{cases}$

For each $i \in \mathbb{N}, t_i$ is choosen randomly according to the exponential distribution λ_{s_i}

Two cases are possible, either the alarm ring ($\eta_i \leq t_i$) or t_i is too short and the alarm doesn't ring.

If $\eta_i \leq t_i$, the alarm ring

- $t_i := \eta_i$ the value of delay is overwritten and match the time spent in s_i
- s_{i+1} is choosen according to the discrete distribution $P_a(s_i, .)$
- $\eta_{i+1} = \begin{cases} \infty \text{ if } s_{i+1} \in S_{off} \\ \text{randomly choosen according to } F_a \text{ if } s_{i+1} \in S_a \end{cases}$

If $\eta_i > t_i$, the alarm does not ring

- t_i the value of delay remain the same and match the time spent in s_i
- s_{i+1} is choosen according to the discrete distribution $\dfrac{Q(s_i, .)}{\lambda_{s_i}}$
- $\eta_{i+1} = \begin{cases} \infty \text{ if } s_{i+1} \in S_{off} \\ \eta_i - t \text{ if } s_{i+1} \in S_a \text{ and } s_i \in S_a \\ \text{randomly choosen according to } F_a \text{ if } s_{i+1} \in S_a \text{ and } s_i \notin S_a \end{cases}$

If the state remain in the set enabled by the alarm, it is updated. Otherwise it is reset according to the new enabled set.

1.4 Phase Type fitting

Phase Type fitting (PH fitting) is approaching a non-Markovian distribution by a sequence of exponential distributions. Consider an infinite sequence of finite sequences of rates of length k $((\lambda_{i,k})_{i=0}^{k})_{k\in\mathbb{N}}$. We want that the distribution of the sequence of exponential transitions to converge to the non-Markovian distribution.

To obtain analysis on ACTMC, we use PH fitting to create CTMC from ACTMC, and analyse those CTMC Then we deduce results on the ACTMC. The reason is that Prism or Storm can use CTMC and not ACTMC. Figure 1 and figure 2 are a toy example of the PH fitting of a simple ACTMC $\mathcal{A} = (\{A, B\}, \begin{bmatrix} 0.5 & 0 \\ 2 & 1 \end{bmatrix}, A, \{A\}, \begin{bmatrix} 1 & 0 \end{bmatrix}, \{d\})$ with constant parameter k. The created CTMC is $\mathcal{C} = (\{A, B, 1, 2, ..., k-1, B\}, \begin{bmatrix} 0 & \lambda_{1,k} & 0 & 0 & ... & 0 \\ 0 & 0 & \lambda_{2,k} & 0 & ... & 0 \\ 0 & 0 & 0 & \lambda_{3,k} & ... & 0 \\ 0 & 0 & 0 & 0 & ... & 0 \\ ... & ... & ... & ... & ... & \lambda_{k,k} \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, A)$

EXIT RATE ¿ 0 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

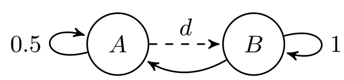

Fig. 1. Model of a simple ACTMC $\mathcal{A}$, d is a non-Markovian distribution.

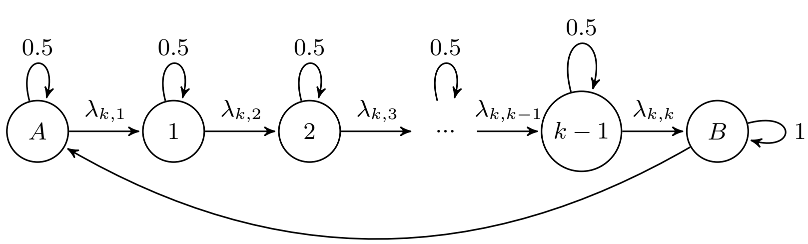

Fig. 2. Model of a CTMC obtain by PH fitting of the ACTMC figure 1.

2 Related Work

Maybe talk about the previous work of vojta and lubosh

3 Contribution

3.1 A Subsection Sample

Please note that the first paragraph of a section or subsection is not indented. The first paragraph that follows a table, figure, equation etc. does not need an indent, either.

Subsequent paragraphs, however, are indented.

Sample Heading (Third Level) Only two levels of headings should be numbered. Lower level headings remain unnumbered; they are formatted as run-in headings.

Sample Heading (Fourth Level) The contribution should contain no more than four levels of headings. Table 1 gives a summary of all heading levels.

Table 1. Table captions should be placed above the tables.

Heading level	Example	Font size and style
Title (centered)	**Lecture Notes**	14 point, bold
1st-level heading	**1 Introduction**	12 point, bold
2nd-level heading	**2.1 Printing Area**	10 point, bold
3rd-level heading	**Run-in Heading in Bold.** Text follows	10 point, bold
4th-level heading	*Lowest Level Heading.* Text follows	10 point, italic

Displayed equations are centered and set on a separate line.

$$x + y = z \tag{1}$$

Please try to avoid rasterized images for line-art diagrams and schemas. Whenever possible, use vector graphics instead (see Fig. 3).

Theorem 1. *This is a sample theorem. The run-in heading is set in bold, while the following text appears in italics. Definitions, lemmas, propositions, and corollaries are styled the same way.*

Proof. Proofs, examples, and remarks have the initial word in italics, while the following text appears in normal font.

For citations of references, we prefer the use of square brackets and consecutive numbers. Citations using labels or the author/year convention are also acceptable. The following bibliography provides a sample reference list with entries for journal articles [**ref˜article1**], an LNCS chapter [**ref˜lncs1**], a book [**ref˜book1**], proceedings without editors [**ref˜proc1**], and a homepage [**ref˜url1**]. Multiple citations are grouped [**ref˜article1**, **ref˜lncs1**, **ref˜book1**], [**ref˜article1**, **ref˜book1**, **ref˜proc1**, **ref˜url1**].

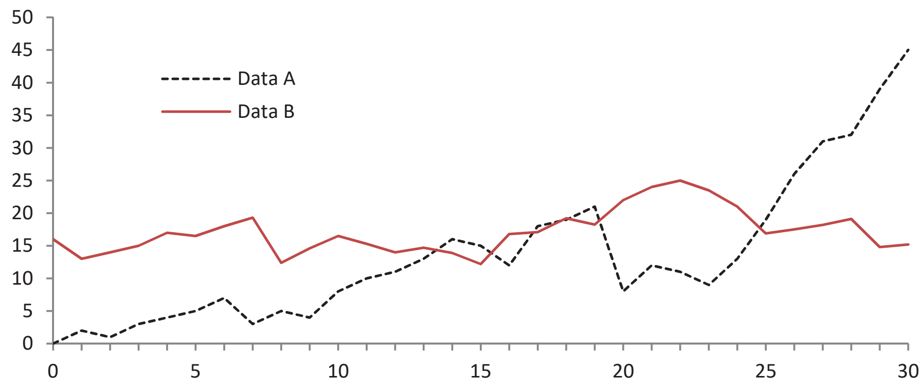

Fig. 3. A figure caption is always placed below the illustration. Please note that short captions are centered, while long ones are justified by the macro package automatically.

4 Conclusion

References

1. Author, F.: Article title. Journal **2**(5), 99–110 (2016)
2. Author, F., Author, S.: Title of a proceedings paper. In: Editor, F., Editor, S. (eds.) CONFERENCE 2016, LNCS, vol. 9999, pp. 1–13. Springer, Heidelberg (2016). https://doi.org/10.10007/1234567890
3. Author, F., Author, S., Author, T.: Book title. 2nd edn. Publisher, Location (1999)
4. Author, A.-B.: Contribution title. In: 9th International Proceedings on Proceedings, pp. 1–2. Publisher, Location (2010)
5. LNCS Homepage, `http://www.springer.com/lncs`. Last accessed 4 Oct 2017

References

[1] M. Kwiatkowska, G. Norman, and D. Parker. "PRISM 4.0: Verification of Probabilistic Real-time Systems". In: *Proc. 23rd International Conference on Computer Aided Verification (CAV'11)*. Ed. by G. Gopalakrishnan and S. Qadeer. Vol. 6806. LNCS. Springer, 2011, pp. 585–591.

[2] Christian Dehnert et al. "A storm is Coming: A Modern Probabilistic Model Checker". In: *CoRR* abs/1702.04311 (2017). arXiv: `1702.04311`. URL: `http://arxiv.org/abs/1702.04311`.

[3] Peter Buchholz, Jan Kriege, and Iryna Felko. *Input Modeling with Phase-Type Distributions and Markov Models: Theory and Applications*. Springer Publishing Company, Incorporated, 2014. ISBN: 3319066730, 9783319066738.