Theoretical Computer Science 611 (2016) 116–135

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Multiphase until formulas over Markov reward models: An algebraic approach

Ming Xu [a,b], Lijun Zhang [c,*], David N. Jansen [d], Huibiao Zhu [e], Zongyuan Yang [a,b]

[a] *Shanghai Key Laboratory of Multidimensional Information Processing, East China Normal University, Shanghai, China*
[b] *Department of Computer Science and Technology, East China Normal University, Shanghai, China*
[c] *State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing, China*
[d] *Model-Based System Development, Radboud Universiteit, Nijmegen, The Netherlands*
[e] *Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, Shanghai, China*

ARTICLE INFO

Article history:
Received 5 April 2015
Received in revised form 20 July 2015
Accepted 25 July 2015
Available online 30 July 2015
Communicated by P. Aziz Abdulla

Keywords:
Probabilistic model checking
Markov reward model
Continuous stochastic logic
Cylindrical algebraic decomposition
Transcendental number

ABSTRACT

We consider the probabilistic model checking problem of continuous-time Markov chains with rewards. We first extend multiphase until formulas in continuous stochastic logic (CSL) with reward constraints. Then we present an effective integral-style algorithm to compute the probability under the assumption of harmony, and give upper and lower bounds of the probability without this assumption. Furthermore, the resulting probability value (or its upper and lower bounds) is shown to be a real number of a well-formed structure, with which we can successfully (or partially) decide whether the constraints in the CSL formula are satisfied. Our method is entirely based on algebraic manipulations and number theory. Finally, to show the practical usefulness, we apply the results to evaluate the performance of a small multi-processor system.

1. Introduction

Nowadays, computing systems deeply influence all aspects of our daily life. This created a surge of interest in validating their correctness (as well as reachability, safety, reliability, performance, etc.) in the last decades. Model checking [15] is a popular method for verifying *qualitative* properties, e.g., "The aircraft communication system never loses contact with the air-traffic control center." However, many *quantitative* properties are equally worth concerning, e.g., "With probability at least $\frac{999}{1000}$, no message is missed under a specific network protocol." To verify these quantitative requirements, probabilistic model checking came into being, which can be viewed as a supplement to traditional model checking. It usually is based on Markov models. Probabilistic model checking can be used for discrete-time and continuous-time Markov models.

For discrete-time Markov models, early work dates back to 1980s. Based on computation tree logic (CTL) [14], Hansson and Jonsson introduced probabilistic CTL (PCTL) by adding the probabilistic quantifier $[\,\cdot\,]_{>c}$ with $c \in [0,1]$ (we will write this operator as $\mathcal{P}_{>c}(\,\cdot\,)$), and further gave an algorithm for checking the validity of PCTL formulas over discrete-time Markov chains (DTMCs) [24]. Like CTL, PCTL consists of state formulas and path formulas. The syntax of PCTL path formulas allows neither Boolean combinations of path formulas nor nesting them. The former can be used to express a conditional probability [2], while the latter can generate something similar to multiphase until formulas. Both operations on path formulas are allowed in linear temporal logic (LTL) [30]. A natural extension of PCTL is PCTL*, introduced by Aziz et al. [5],

[*] Corresponding author.
E-mail addresses: mxu@cs.ecnu.edu.cn (M. Xu), zhanglj@ios.ac.cn (L. Zhang), dnjansen@cs.ru.nl (D.N. Jansen), hbzhu@sei.ecnu.edu.cn (H. Zhu), yzyuan@cs.ecnu.edu.cn (Z. Yang).

http://dx.doi.org/10.1016/j.tcs.2015.07.047
0304-3975/© 2015 Elsevier B.V. All rights reserved.

which subsumes both PCTL and LTL. The decidability of PCTL* formulas over DTMCs follows from the fact in [21] that a set of paths satisfying a formula in probabilistic LTL is measurable. Furthermore, Bianco and de Alfaro presented model checking algorithms for PCTL and PCTL* formulas over discrete-time Markov decision processes, in which the probabilistic behavior coexists with nondeterminism [13]. For DTMCs with a reward structure, Andova et al. studied PCTL extended with instantaneous and cumulative reward properties [1].

For continuous-time Markov models, the foundational work is Aziz et al.'s paper [4]. The authors introduced continuous stochastic logic (CSL) for continuous-time Markov chains (CTMCs). Roughly speaking, the syntax of CSL amounts to that of PCTL plus the multiphase until formula. Such a formula has the form $\Phi_1 \, \mathrm{U}^{\mathcal{I}_1} \, \Phi_2 \cdots \mathrm{U}^{\mathcal{I}_{K-1}} \, \Phi_K$, for some $K > 1$. Intuitively, it requires that a path passes through Φ_1-, Φ_2-, ..., Φ_K-states (in this order), and each transition from a Φ_k- to a Φ_{k+1}-state happens at a time instant in interval $\mathcal{I}_k$—all times measured from the start of the path.[1] Because [4] restricts probabilities in $\mathcal{P}_{>c}$ to $c \in \mathbb{Q}$, they can show decidability of model checking for CSL using number-theoretic analysis. However, besides some semantical problems pointed out in the paper [26], the method turned out to be inefficient in practice, a.o. because the handling of overlapping intervals ($\mathcal{I}_{k-1} \cap \mathcal{I}_k \neq \emptyset$) was difficult. An approximate model checking algorithm for a reduced version of CSL was provided by Baier et al. [9], who restrict path formulas to binary until: $\Phi_1 \, \mathrm{U}^{\mathcal{I}} \, \Phi_2$. Under this logic, they successfully applied efficient numerical techniques for *transient analysis* [7] using *uniformisation* [32]. Recently the approximate algorithms have been extended for multiphase until formulas using *stratification* [34,35]. Furthermore, Gao et al. constructed a series of stratified CTMCs under a parameter, and proposed an approximate algorithm to compute the probability of a conjunction of path formulas [22]. As a result, requirements on conditional probabilities on CTMCs can be formulated, analogous to that on DTMCs [2,10]. For CTMCs with a reward structure, Baier et al. [8] introduced the continuous stochastic reward logic (CSRL), based on their restricted version of CSL [9], to specify time- and reward-bounded properties. A number of numerical algorithms were given to compute the probability of binary until formulas [25]. Later, Cloth et al. [16,6] derived a hyperbolic system of *partial differential equations* for the probability $\Pr(\Phi_1 \, \mathrm{U}^{\mathcal{I}}_{\mathcal{J}} \, \Phi_2)$. Its solution is a system of *integral equations*, which can be numerically solved by an iteration.

Many of the above algorithms have been implemented in probabilistic and reward model checkers like *PRISM* [28], *MRMC* [27], and *iscasMc* [23]. The technical core lies in the probability computation of binary until formulas. However, most implemented algorithms are based on numerical computation with little regard to safety: If the error interval overlaps c, the tools sometimes report incorrect results. Additionally, their input languages do not allow to express multiphase until formulas. We now aim to fill this gap by means of algebraic methods.

In this paper, we study the probabilistic model checking problem of CTMCs with a reward structure. First we extend CSRL, based on the original version of CSL [4], with time- and reward-bounded multiphase until formulas:

$$\phi = \Phi_1 \mathrm{U}^{\mathcal{I}_1}_{\mathcal{J}_1} \Phi_2 \cdots \mathrm{U}^{\mathcal{I}_{K-1}}_{\mathcal{J}_{K-1}} \Phi_K .$$

This formula requires that paths accumulate a reward bounded in $\mathcal{J}_k$ when they jump from Φ_k-states to Φ_{k+1}-states, and that the jump time instants are in $\mathcal{I}_k$—all times and rewards measured from the start of the path.

The main method of this article can be summarized briefly as follows. We manipulate algebraic terms as long as possible—even the integration step can be executed symbolically—and only start numerical computations once the terms have reached a standard form. We present the assumption of *harmony* that allows us to get the algebraic description of the sojourn times of the satisfying paths. In principle, algebraic manipulations are exact, so we do not suffer from the problem that numerical errors may accumulate throughout a calculation. Only the numerical error at the end of the process needs to be controlled to get a *safe* decision on the truth value of a probabilistic formula in CSRL. Contrary to traditional numerical integration, we propose to use symbolic integration and number-theoretic results, which ensure termination and correctness. To show the practical usefulness, we apply the above results to evaluate the performance of a multi-processor system. However, without the assumption of harmony, we can only get over- and under-approximations of the sojourn times of the satisfying paths. They result in upper and lower bounds of the probability of a multiphase until formula, respectively.

Naturally, if one starts with estimated transition probabilities, even exact manipulations will not turn them into exact reachability probabilities. In such cases, the advantage that our method avoids numerical errors is of little importance, so one may use a quicker approximation method, e.g., one of the methods proposed by [25].

The factors in our calculations are mostly matrices, but in the end, we will get a single scalar value by using the matrix as a bilinear form: $\alpha \cdot \mathbf{A}_1 \cdot \mathbf{A}_2 \cdot \cdots \cdot \mathbf{A}_K \cdot \mathbf{1}$, where α is a row vector of the initial distribution and $\mathbf{1}$ is a column vector of ones. We construct the matrix from left to right; if one immediately multiplies out the intermediary terms $\alpha \cdot \mathbf{A}_1$ etc., one only needs to store a vector of algebraic terms. About half of the matrices $\mathbf{A}_k$ are diagonal matrices that contain only ones and zeroes, which effectively erase some elements in the vector and so simplify the next computation step.

Organization. In Section 2 we review some basic notions of Markov reward models. We introduce our extension of continuous stochastic reward logic in Section 3. Then we present algebraic algorithms for the probability computation of multiphase until formulas in Section 4 and give a number-theoretic method to determine the truth of the probability state

[1] Note that this is a monolithic path formula constructor with K subformulas. The nested formula $\Phi_1 \, \mathrm{U}^{\mathcal{I}_1} \, (\Phi_2 \, \mathrm{U}^{\mathcal{I}_2} \, (\Phi_3 \cdots (\Phi_{K-1} \, \mathrm{U}^{\mathcal{I}_{K-1}} \, \Phi_K) \cdots))$ looks similar, but interprets the time bounds differently: here, the time to remain in Φ_k-states has to be in $\mathcal{I}_k$, independent from earlier sojourn times.

A multiphase until formula (e.g., $\exists(a \, \mathrm{U} \, b \, \mathrm{U} \, c)$) is not needed in bare CTL because a combination of binary until formulas is equivalent, in this example $\exists(a \, \mathrm{U} \, \exists(b \, \mathrm{U} \, c))$.