Science of Computer Programming 153 (2018) 1–29

Contents lists available at ScienceDirect

Science of Computer Programming

www.elsevier.com/locate/scico

An efficient TCTL model checking algorithm and a reduction technique for verification of timed actor models

Ehsan Khamespanah [a,b,**], Ramtin Khosravi [a,*], Marjan Sirjani [c,b,*]

[a] School of Electrical and Computer Engineering, University of Tehran, Tehran, Iran
[b] School of Computer Science, Reykjavik University, Reykjavik, Iceland
[c] Mälardalen University, School of IDT, Västerås, Sweden

ARTICLE INFO

Article history:
Received 7 February 2016
Received in revised form 10 November 2017
Accepted 10 November 2017
Available online xxxx

Keywords:
Actor model
Timed Rebeca
Model checking
TCTL
Durational transition graph

ABSTRACT

NP-hard time complexity of model checking algorithms for TCTL properties in dense time is one of the obstacles against using model checking for the analysis of real-time systems. Alternatively, a polynomial time algorithm is suggested for model checking of discrete time models against $\text{TCTL}_{\leq,\geq}$ properties (i.e. TCTL properties without $\mathbf{U}^{=c}$ modalities). The algorithm performs model checking against a given formula Φ for a state space with V states and E transitions in $O(V(V+E)\cdot|\Phi|)$. In this work, we improve the model checking algorithm of $\text{TCTL}_{\leq,\geq}$ properties, obtaining time complexity of $O((V\lg V+E)\cdot|\Phi|)$. We tackle the model checking of discrete timed actors as an application of the proposed algorithms. We show how the result of the fine-grained semantics of discrete timed actors can be model checked efficiently against $\text{TCTL}_{\leq,\geq}$ properties using the proposed algorithm. This is illustrated using the timed actor modeling language Timed Rebeca. In addition to introducing a new efficient model checking algorithm, we propose a reduction technique which safely eliminates instantaneous transitions of transition systems (i.e. transition with zero time duration). We show that the reduction can be applied on-the-fly during the generation of the original timed transition system without a significant cost. We demonstrate the effectiveness of the reduction technique via a set of case studies selected from various application domains. Besides, while $\text{TCTL}_{\leq,\geq}$ can be model checked in polynomial time, model checking of TCTL properties with $\mathbf{U}^{=c}$ modalities is an NP-complete problem. Using the proposed reduction technique, we provide an efficient algorithm for model checking of complete TCTL properties over the reduced transition systems.

© 2017 Elsevier B.V. All rights reserved.

1. Introduction

As a basic computational model for modeling of real-time systems, Timed Transition System (TTS) generalizes the basic computational model of transition systems by associating an interval with each transition to indicate how long a transition takes [1]. TTS is expressive enough for modeling the behavior of the majority of real-time distributed systems; however, the formal verification of TTS is PSPACE-complete [1]. Therefore, currently there is no polynomial time algorithm for the verification of TTSs. Another option for analysis of real-time systems is to use Alur and Dill's *Timed Automata* [2]. There

* Corresponding authors.
** Principal corresponding author.
E-mail addresses: e.khamespanah@ut.ac.ir, ehsan13@ru.is (E. Khamespanah), r.khosravi@ut.ac.ir (R. Khosravi), marjan.sirjani@mdh.se, marjan@ru.is (M. Sirjani).

https://doi.org/10.1016/j.scico.2017.11.004
0167-6423/© 2017 Elsevier B.V. All rights reserved.

exists a large amount of theoretical knowledge and practical experiences about timed automata which all agree on the main drawback of using timed automata being the inefficient analysis techniques which are at least PSPACE-hard [3]. The most widely used model checking toolset for timed automata, UPPAAL, only supports a limited subset of Timed Computation Tree Logic (TCTL) which can be model checked efficiently [4]. The source of this inefficiency in the analysis of TTS and timed automata is in how the passage of time is modeled. The model of time in TTS and timed automata is *dense time*, i.e. the passage of time from a state to another state is a nondeterministically chosen real number from an interval.

On the other hand, a wider range of properties can be analyzed for simpler families of timed models in polynomial time. The simplicity of these models lies in the discretization of the passage of time. In these models, the passage of time is modeled by a natural number which is chosen nondeterministically from an interval. The basic approach of such simplifications is proposed in [5,6] by assuming that each transition takes exactly one time unit. Later, a minor extension has been added to this work by allowing existence of instantaneous transitions (zero time transitions) in [7]. Finally, Timed Transition Graph (TTG) [8] and Durational Transition Graph (DTG) [9] extended the former works by associating discrete time duration with transitions. Although TTG and DTG are less expressive than TTS and timed automata, they can be model checked in polynomial time for a wide range of properties. For example, there is a polynomial time algorithm for model checking of DTGs against $\text{TCTL}_{\leq,\geq}$ properties (i.e. TCTL properties without any sub-formula of the form $\Phi\ \mathbf{U}^{=c}\ \Psi$). The algorithm performs model checking against formula Φ for a transition system with V states and E transitions in time $O(V \cdot (V+E) \cdot |\Phi|)$ [9]. The details of these model checking algorithms are reviewed in Section 2. Note that, while $\text{TCTL}_{\leq,\geq}$ can be model checked for DTGs in polynomial time, the model checking against $\text{TCTL}_{=}$ properties (i.e. TCTL properties with sub-formulas of the form $\Phi\ \mathbf{U}^{=c}\ \Psi$) is a NP-hard problem. In this work, we improve the running time of the algorithm of [9] from $O(V \cdot (V+E) \cdot |\Phi|)$ to $O((V \lg V + E) \cdot |\Phi|)$. The newly proposed algorithm is worst-case optimal for model checking of $\text{TCTL}_{\leq,\geq}$ properties, since its running time is the same as the tight running time of the CTL model checking algorithm [3]. This algorithm is presented in detail in Section 3.

In addition to improving the running time of $\text{TCTL}_{\leq,\geq}$ model checking algorithm, we propose a reduction technique which safely eliminates instantaneous transitions of timed transition systems. Applying this technique, a new transition system is created, called folded timed transition system (FTS). As discussed in Section 4, in addition to reducing the size of transition systems, the algorithm of *exact path search* in graphs can be used for model checking of FTS against $\text{TCTL}_{=}$ properties. Having instantaneous transitions, the problem of model checking against $\text{TCTL}_{=}$ properties is reducible to the Subset Sum problem which is well-known to be NP-complete [9]. By eliminating instantaneous transitions, FTS can be model checked against TCTL properties efficiently. In the proposed algorithm, for a given TCTL formula, if small values are used as timed quantifiers of TCTL modalities, FTS can be model checked in $O((V \lg V + E) \cdot |\Phi|)$.

We tackle the problem of analyzing discrete timed actors[1] to illustrate the applicability of the proposed approaches. The actor model is a well-established paradigm for modeling the functional behavior of distributed systems with asynchronous message passing. This model was originally introduced by Hewitt [10] and then elaborated by Agha [11,12] and Talcott [13]. Actors are attracting more and more attention both in academia and industry; whoever, little work has been done on timed actors and even less on analyzing timed actor models. To the best of our knowledge, only a few timed actor modeling languages such as RT-synchronizer [14], real-time Creol [15], and Timed Rebeca [16] exist. Although there are some studies on verification of timed actors [17,18], the lack of efficient model checking algorithms has limited the use of model checking for this purpose. As DTG is expressive enough to be used as the semantics of discrete timed actors, we show how it can be used for efficient model checking of timed actors. We develop this approach for Timed Rebeca models. Timed Rebeca [19] has been proposed as an extension of the Rebeca language [20,21] with time constraints and analysis support. Timed Rebeca is an actor-based modeling language which can be used in model-driven methodologies. Using Timed Rebeca a designer builds an abstract model in which each component is a reactive object communicating with other objects through non-blocking asynchronous message passing. In Section 5, we show how the transition systems which are generated based on the fine-grained semantics of Timed Rebeca can be assumed as DTGs to be efficiently model checked against $\text{TCTL}_{\leq,\geq}$ properties. Although we demonstrate our approach on Timed Rebeca, it can be easily generalized to other timed actor models.

We also elaborate on the execution cost of generating FTSs from DTGs. Using the approach of this paper, the FTS of a Timed Rebeca model is generated without a significant cost, in parallel with the generation of its original transition system and checking for Zeno freedom of models. In Section 4 we show how the algorithm of transition system generation and checking for Zeno behavior are modified to generate FTSs. We have developed a tool (added to the rich Rebeca toolset [22]), to illustrate the impact of using these techniques by applying them on a set of case studies in different application domains (Section 6).

This paper is an extended version of the workshop paper [23]. In [23], we showed how the TCTL model checking algorithm of [9] can be used for the model checking of Timed Rebeca models. We also introduced FTS in that paper. Apart from adding more detail about the proposed approaches, this paper extends [23] as follows:

- We propose a new model checking algorithm with an asymptotically smaller running time in comparison with the existing model checking algorithms of discrete time systems (Sections 3.1 and 3.2).

[1] We use "timed actor" and "discrete timed actor" in this paper interchangeably.

Download English Version:

https://daneshyari.com/en/article/6875291

Download Persian Version:

https://daneshyari.com/article/6875291

Daneshyari.com