On Model-Checking Probabilistic Timed Automata against Probabilistic Duration Properties

Dang Van Hung1, Miaomiao Zhang2, Pham Dinh Chinh1

1 University of Engineering and Technology, VNU, Hanoi, Vietnam
2 School of Software Engineering, Tongji University, Shanghai, China

Abstract

In this paper, we consider a subclass of Probabilistic Duration Calculus formulas called Simple Probabilistic Duration Calculus (SPDC) as a language for specifying dependability requirements for real-time systems, and address the two problems: to decide if a probabilistic timed automaton satisfies a SPDC formula and to decide if there is a strategy to choose an execution of a given automaton that satisfies a SPDC formula. We prove that the two problems are decidable for a class of SPDC called probabilistic linear duration invariants, and provide a model checking algorithm for solving these problems.

Introduction

In this paper, we introduced a simple probabilistic extension of DC called Probabilistic Duration Calculus for specifying dependability requirements of real-time systems. We use the behavioral model proposed by Kwiatkowska et al to define the semantics of our logic. Since probabilistic timed CT and PDC are not comparable, and since for many probabilistic properties PDC is more convenient to specify, a probabilistic timed automaton satisfies a PDC formula of a certain form. Then, we generalize this technique to achieve our goal with a model-checking algorithm.

The first version of this paper was published in [2]. In this extended version, in addition to the problem of verification, we formulate also the problem of strategy synthesis, i.e., to decide if there is a strategy for a probabilistic timed automaton that satisfies a probabilistic linear duration invariant and show that this problem is also solvable. We provide all proof details and algorithms for doing model-check.

Main Objectives

1. present the Probabilistic Timed Automata model.
2. present syntax and semantics of our PDC.
3. presented in Section 2.1, we formulate our model checking problem and give our solution to it.

Materials and Methods

In 1992, Chaochen Zhou, Hoare C.A.R and Anders Ravn introduced Duration Calculus [1] as a logic for reasoning about real-time systems. A version with a proof system of Probabilistic Duration Calculus with infinite interval was then developed by Dimitar Guelev [3], and in [4] we have shown that the calculus is useful for reasoning about QoS contracts in component-based real-time systems.

For Duration Calculus, some techniques for checking if a timed automaton satisfies a duration calculus formula written in the form of linear duration invariants have been developed. However, to our knowledge, not many works have been done for checking if a probabilistic real-time system satisfies a PDC formula.

Kwiatkowska et al in [5] proposed a variant of probabilistic timed automata that allows probabilistic choice only at discrete transitions.

Results

Probabilistic Timed Automata

Definition 1. A probabilistic timed automaton (PTA) is a tuple \( G = (S, L, s_0, C, \text{inv}, \text{prob}, \{r|s|_G}) \) consisting of:

- a finite set \( S \) of nodes, a start node \( s_0 \in S \), a finite set \( C \) of clocks,
- a function \( L : S \rightarrow 2^AP \) assigning to each node of the automaton a set of atomic propositions that are supposed to be those that are true in that node, a function \( \text{inv} : S \rightarrow \mathbb{Z}_c \) assigning to each node an invariant condition,
- a function \( \text{prob} : S \rightarrow 2^{(S \times S)} \) assigning to each node a set of discrete probability distributions on \( S \times S \),
- a family of functions \( (\tau_i|s|_G) \), where, for any \( s \in S \), \( \tau_i|s|_G \rightarrow Z_c \) assigns to each \( p \in \text{prob}(s) \) an enabling condition.

Figure 1: A probabilistic timed automaton for a simple gas burner

Probabilistic Duration Calculus

In this section, we introduce a simple form of Probabilistic Duration Calculus.

Definition 3. Let \( \Box \) stand for relations (e.g., \( \leq \)), and \( \mathcal{F} \) stand for functions (e.g., \( + \)). The syntax of Probabilistic Duration Calculus is defined as follows:

\[
\begin{align*}
\Psi &::= 0 \mid 1 \mid \mathcal{F}(\Psi) \mid \mathcal{F}(\Psi) \mathcal{F}(\Psi) \mid \Box \mathcal{F}(\Psi) \mathcal{F}(\Psi) \\
\end{align*}
\]

where \( \mathcal{F} \) stands for Probabilistic Duration Calculus formulas, \( \mathcal{F}(\Psi) \) stands for Duration Calculus formulas, \( \Box \) stands for duration terms, \( \mathcal{F} \) stands for state expressions, and \( \mathcal{F}(\Psi) \) is a symbol in the set of atomic proposition \( \Delta \).

Figure 2: A part of a strategy \( \sigma \) for the simple gas burner

Conclusion

This paper has presented the problem of checking probabilistic timed automata against probabilistic duration calculus formulas. The problem is decidable for a class of PDC formulas of the form \( [\Psi]_t \), where \( \Psi \) is a linear duration invariant, or a DC formula for bounded liveness. The technique for model checking is an extension of our techniques for checking if a timed automaton satisfies a linear duration invariant using a searching method in the interval region graph of the timed automaton. The complexity of the decision procedure is high in general. Since the problem possesses a potential high complexity, we have not implemented the technique yet. Hope that with the increasing computing power in the future, we can develop an effective tool for model-checking based on the technique.

Forthcoming Research

We are looking for some special cases of the problem which are simpler and still useful for which our technique can work well, and then implement it as a tool to assist checking the dependability for embedded systems.

References


Acknowledgements

This research was funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.03-2014.23.