Theoretical Computer Science 537 (2014) 3–28

Contents lists available at SciVerse ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

TCTL-preserving translations from timed-arc Petri nets to networks of timed automata

Joakim Byg, Morten Jacobsen, Lasse Jacobsen, Kenneth Yrke Jørgensen [1], Mikael Harkjær Møller [1], Jiří Srba [*,1]

Department of Computer Science, Aalborg University, Selma Lagerlöfs Vej 300, 9220 Aalborg East, Denmark

ARTICLE INFO

Keywords:
Formal verification
TCTL
Timed-arc Petri nets
Timed automata

ABSTRACT

We present a framework for TCTL-preserving translations between time-dependent modeling formalisms. The framework guarantees that once the original and the translated system are in one-by-many correspondence relation (a notion of behavioral equivalence between timed transition systems) then TCTL properties of the original system can be transformed too while preserving the verification answers. We demonstrate the usability of the technique on two reductions from bounded timed-arc Petri nets to networks for timed automata, providing unified proofs of the translations implemented in the verification tool TAPAAL. We evaluate the efficiency of the approach on a number of experiments: alternating bit protocol, Fischer's protocol, Lynch–Shavit protocol, MPEG-2 encoder, engine workshop and medical workflow. The results are encouraging and confirm the practical applicability of the approach.

1. Introduction

Formal verification of embedded and hybrid systems is an active research area. Recently, a lot of attention has been devoted to the analysis of systems with quantitative attributes like timing, cost and probability. In particular, several different time-dependent models have been developed over the two last decades or so. These models are often introduced as time extensions of well-studied untimed formalisms and include, among others, networks of timed automata [5,6] and different time extensions of the Petri net model [42]. These formalisms are nowadays supported by a number of tools [8,16,17,32,23,25,11,27] and applied in model-driven system design methodologies.

We shall focus on the Petri net model extended with continuous time. The timing aspects can be associated with different parts of the model in the various time-extended Petri net formalisms. For example, *timed transitions Petri nets* where transitions are annotated with their durations were proposed in [43]. A model in which time parameters are associated with places is called *timed places Petri nets* and it was introduced in [44]. *Time Petri nets* of Merlin and Faber [35,36] were introduced in 1976 and associate time intervals to each transition. The intervals define the earliest and latest firing time of the transition since it became enabled. Yet another model of *Timed-Arc Petri Nets* (TAPN) was first studied around 1990 by Bolognesi, Lucidi, Trigila and Hanisch [12,26]. Here time information is attached to the tokens in the net representing their relative age while arcs from places to transitions contain time intervals that restrict the enabledness of the transitions (see [29] for a recent introduction to TAPNs). For an overview of the different extensions consult e.g. [15,51,40,47]. The TAPN

* Corresponding author.
E-mail addresses: joakim@aptusoft.com (J. Byg), morten.jacobsen.2k@gmail.com (M. Jacobsen), lassejacobsen@gmail.com (L. Jacobsen), kyrke@ist.aau.dk (K.Y. Jørgensen), m@mikael.hm (M.H. Møller), srba@cs.aau.dk (J. Srba).

[1] The authors were supported by VKR Center of Excellence MT-LAB.

0304-3975/$ – see front matter
http://dx.doi.org/10.1016/j.tcs.2013.07.011

model is particularly suitable for modeling of manufacturing systems, workflow management and similar applications [4,2,38,49,50,39] and a recently developed tool TAPAAL [19,23] enables automatic verification of bounded TAPNs extended with transport/inhibitor arcs and age invariants.

One way to verify formal models is via a translation to another formalism with a well-established tool support and indeed, several such translations have already been developed (see e.g. [47] for an overview). Many of the translations utilize similar tricks that allow for the simulation of one system by another. Typically, a single step in one formalism is simulated by a sequence of steps in the other one. Inspired by the translations presented in the literature [21,14,24,30,9] we identify a general class of model transformations that preserve the satisfiability of *Timed Computation Tree Logic* (TCTL) [41], a logic suitable for a practical specification of many useful temporal properties. The main goal of this article is to provide a general proof framework directly applicable to the kinds of translations often implemented by the tool developers, and to demonstrate its applicability by presenting two concrete translations from timed-arc Petri nets to networks of timed automata that proved to be the most efficient and are implemented in the tool TAPAAL.

Unlike much work on TCTL where only infinite alternating runs are considered [41] or the details of the semantics are not fully discussed [21,13], we consider also finite maximal runs that appear in the presence of stuck computations or time invariants (strict or nonstrict) and treat the semantics in its full generality as necessary for the correct semantics of the models used in many tools including e.g. Uppaal [8]. This is particularly important for liveness properties. While some translations in the literature (not necessarily only between timed-arc Petri nets and timed automata) preserve some variant of timed bisimilarity [21,14,24,30], other translations preserve only reachability or trace equivalence [9,18]. Our framework allows us to argue that several such translations preserve the full TCTL or at least its safety fragment.

Further we propose two novel TCTL-preserving translations from timed-arc Petri nets extended with all the additional features implemented in TAPAAL to Uppaal networks of timed automata. Earlier translations either caused exponential blow-up in the size [45,46,14], or where not suitable for implementation in tools due to an inefficient use of clocks and communication primitives [46]. One of the translations from TAPN to Uppaal timed automata presented in this article is the first one to run in polynomial time while preserving the full TCTL. It uses the idea of a controller (like for example in [21] where it was employed for a translation from time Petri nets to timed automata) that is guiding the firing of transitions. We implemented both translations in the tool TAPAAL and the experiments that we shall present confirm their efficiency also in practice. One of the important contributions of this article is the fact that we treat the TAPN models in their full generality, including transport and inhibitor arcs and age invariants, exactly as implemented in the tool TAPAAL. This provides the evidence that the theory behind the translations in the tool is sound.

The work we present here is based on several conference papers [18,28,29,46] where the initial translations and the theory were originally published. This journal article, as presented at ICTAC'11 in a half-day tutorial, unifies the different proofs into a common framework and considerably simplifies the correctness arguments. In Section 2 we present the proof framework for arguing about TCTL-preservation of the translations. We give a formal syntax and semantics of timed-arc Petri nets in Section 3 and of networks of timed automata in Section 4. In Section 5 and Section 6 we present two translations from TAPN to NTA, and prove their correctness using the proof framework from Section 2. Section 7 presents a selection of experiments where we compare the performance of the implemented translations. Finally, Section 8 gives a short conclusion.

2. Proof framework

In this section, we shall present a general framework for arguing when a simulation of one time-dependent system by another one preserves the satisfiability of TCTL formulae. We define the notion of one-by-many correspondence, a relation between two TTSs A and B. If A is in one-by-many correspondence with B then every transition in A can be simulated by a sequence of transitions in B. Further, every TCTL formula φ can be algorithmically translated into a formula $tr(\varphi)$ such that $A \models \varphi$ iff $B \models tr(\varphi)$. Before we can describe the framework, we need to introduce some preliminary definitions.

2.1. Preliminaries

We let $\mathbb{Z}$, $\mathbb{N}$, $\mathbb{N}_0$, $\mathbb{R}$ and $\mathbb{R}_{\geqslant 0}$ denote the sets of integers, natural numbers, non-negative integers, real numbers and non-negative real numbers, respectively.

Definition 1. A *timed transition system* (TTS) is a tuple $T = (S, \longrightarrow, \mathcal{AP}, \mu)$ where S is a set of states (or processes), $\longrightarrow \subseteq (S \times S) \cup (S \times \mathbb{R}_{\geqslant 0} \times S)$ is a transition relation, $\mathcal{AP}$ is a set of atomic propositions, and $\mu : S \longrightarrow 2^{\mathcal{AP}}$ is a function assigning sets of true atomic propositions to states.

We write $s \longrightarrow s'$ whenever $(s, s') \in \longrightarrow$ and call them *discrete transitions*, and $s \xrightarrow{d} s'$ whenever $(s, d, s') \in \longrightarrow$ and call them *delay transitions*. We require that the TTSs we consider satisfy the following standard axioms for delay transitions (see e.g. [10]). For all $d, d' \in \mathbb{R}_{\geqslant 0}$ and $s, s', s'' \in S$:

1. **Time additivity:** if $s \xrightarrow{d} s'$ and $s' \xrightarrow{d'} s''$ then $s \xrightarrow{d+d'} s''$;
2. **Time continuity:** if $s \xrightarrow{d+d'} s''$ then $s \xrightarrow{d} s' \xrightarrow{d'} s''$ for some s';