Extended Abstract: Multi-Timed Bisimulation for Distributed Timed Automata

Ortiz Vega, James Jerson; Schobbens, Pierre-Yves

Published in:
Open problems in real-time computing

Publication date:
2014

Document Version
Early version, also known as pre-print

Link to publication
Citation for published version (HARVARD):

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal?

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 22. Oct. 2018
Multi-Timed Bisimulation for Distributed Timed Automata

James Ortiz and Pierre-Yves Schobbens

Computer Science Faculty, University of Namur
jor@info.unamur.ac.be, pys@info.unamur.edu.be

1 Extended Abstract

Distributed and reactive systems can be modeled in a modular manner using operations such as sequential or parallel composition in order to build larger modules. Each component of a modular system can be replaced by behaviorally equivalent components without changing the properties of the modules. The properties can be preserved by means of semantic equivalences. Branching bisimulation preserves Computation Tree Logic (CTL)[4] and language inclusion preserves Linear Temporal Logic (LTL)[11]. The idea behind bisimulation is to regard two states as equivalent if they can engage in the same observable transitions (transitions with identical labels) and after performing similar transitions end up in equivalent states again. If one chooses to think of every transition as observable, the relation is called a strong bisimulation. If one abstracts from internal transitions, it is called weak bisimulation or branching bisimulation.

The timed properties can be preserved by means of timed semantic equivalences. Timed bisimulation preserves timed properties expressed in timed modal logics such as Timed Computation Tree Logic (TCTL) [16], $L_\nu$ [12]. Timed bisimulation was shown decidable for Timed Automata (TA) [3]. Efficient algorithms checking for timed bisimilarity have been discovered [12] and implemented in tools for automatic verification [6]. TA are used to model real-time systems and have become popular as modeling language for several model checkers [17][13] because of their simplicity and tractability. A timed automaton is a finite automaton augmented with real-valued clocks. Clocks are variables that increase at the same rate in order to register time progress. Constraints on these clocks are used to restrict the behaviors of the automaton. The model of TA assumes perfect clocks: all clocks have infinite precision and are perfectly synchronized. Transitions are labelled with constraints on clocks, called guards, that indicate when such transition may take place. Usually TA are used to model real-time systems with hard constraints. In this case, TA are equipped with an invariant, which is a constraint on clocks that limits time progress in each control state [9]. A established bridge has been provided between TA and timed modal logic, by the notion of characteristic formula. A characteristic formula for TA is a formula in a timed temporal logic $L_\nu$ that completely characterizes the behavior of an automaton modulo some chosen relation [12]. A solution has first been proposed in [12] for TA, providing formulas in the timed modal logic $L_\nu$. Then, these results have been improved in [1], yielding linear constructions.

Some variants of TA called Distributed Timed Automata DTA [10] and icTA [2][15] are used to model real-time distributed systems, where the clocks are not necessarily synchronized. Constraints on the clocks are used to restrict the behaviors of the automaton in the different processes. The clocks belonging to
one process can be read by another process, but a clock can only be reset by its owner process. The clocks in iCTA can be in different processes and we cannot assume that they are perfectly synchronized. DTA and iCTA can be used to model systems such as the Controller Area Network (CAN) [14], WirelessHART Networks [5], and the ARINC-659 protocol [7].

Our first contribution is to extend the definition of Timed Labelled Transition Systems (TLTS), iCTA [2], timed languages and timed bisimulation removing the assumption of clock synchronization, the idea that the clocks can advance independently if they are in different processes, yielding the multi-timed bisimulation. We will show that multi-timed bisimulation is decidable (more exactly, \textsc{Exptime}-complete). Our second contribution is a logical characterization of multi-timed bisimulation for iCTA, based on timed modal logic $\textsc{ML}_\nu$, an extension of $L_\nu$ [12] and Hennessy-Milner logic [8].

References
13. The Liège Automata-based Symbolic Handler (LASH). Available at \url{http://www.montefiore.ulg.ac.be/~boigelot/research/lash/}.
17. The UPPAAL tool. Available at \url{http://www.uppaal.com/}.