An Alternative Semantics for Timed Automata with Independent Clocks

Ortiz Vega, James Jerson; Schobbens, Pierre-Yves

Publication date:
2023

Document Version
Early version, also known as pre-print

Link to publication

Citation for pulished version (HARVARD):
Ortiz Vega, JJ & Schobbens, P-Y 2023, 'An Alternative Semantics for Timed Automata with Independent Clocks', Recent Advances in Concurrency and Logic (RADICAL 2023), Antwerp, Belgium, 18/09/23 - 18/09/23.

An Alternative Semantics for Timed Automata with Independent Clocks

James Ortiz and Pierre-Yves Schobbens

University of Namur, Namur, Belgium
{james.ortizvega, pierre-yves.schobbens}@unamur.be

Abstract

With the rapid growth of distributed computing and networking, the demand for large-scale, complex distributed applications is increasing significantly. Distributed Real-Time Applications (DRTA) are used to control and monitor a wide variety of distributed real-time systems, including aerospace, robotics, and nuclear power plants. DRTA often operate on heterogeneous computer networks with multiple interconnected components, each equipped with its own local independent clock. These independent clocks have their own rate behavior without synchronization. Timed Automata (TA) with independent clocks can be used to model DRTA with independent clocks. However, it is important to recognize that in certain scenarios there may be indirect interactions or dependencies between independent clocks. This may be the case in larger systems where different components or subsystems interact or depend on each other for timing information. In this talk, we will propose a derivative-based alternative semantics for TA with independent clocks. This approach has been implemented in a tool called MIMETIC. We will also talk about the problem of bisimulation with our alternative semantics for TA with independent clocks.

1 Introduction

Distributed Real-Time Applications (DRTA) play an increasingly important role in everyday life, from traffic light controllers to airplanes, and from telecommunication networks to medical systems. In the last decades, several formal methods and real-time formalisms have been proposed to formalize and prove properties of DRTA. However, traditional real-time formalisms are not always adequate for reasoning about DRTA because they assume a unique, perfectly synchronous (Newtonian) measure of time. A successful technique for modeling DRTA is Timed Automata (TA)[2]. A TA is a finite automaton augmented with real-valued clocks, where all clocks have infinite precision and are perfectly synchronized. There are other variants of TA called Distributed Timed Automata (DTA) [8], Timed Automata with Independent Clocks (icTA) [1] and Distributed Event Clock Automata (DECA) [12] where the clocks are not necessarily synchronized. A notable feature of these independent clocks in DTA, icTA and DECA is their autonomous operation, without any direct relationship or dependency on other clocks in the automaton. Each clock operates independently and maintains its own rate behavior without the need to synchronize with other clocks. This autonomy provides flexibility and allows each automaton (or component) to follow its own timing constraints without being influenced by the behavior of other clocks. However, it is important to recognize that although the clocks are independent, there may still be indirect interactions or dependencies between them. These interactions can arise due to the interconnected nature of the DRTA or due to dependencies between components. For example, the behavior of one component, driven by its independent clock, may indirectly affect the timing or behavior of other components in the system. Consequently, understanding and managing these dependencies is critical to ensuring proper synchronization, coordination, and overall system performance.

In order to model Distributed Real-Time Applications (DRTA) with greater flexibility and accuracy, we propose an alternative semantics for Timed Automata with Independent Clocks (icTA, DTA, and DECA). This alternative semantics allows for the inclusion of rate constraints on clock derivatives, allowing for a more detailed representation of distributed real-time behavior. Under this proposed semantics, rate constraints are associated with the locations of the automaton and the clock derivatives correspond to the first derivative of the original clocks. Each clock can proceed at its own rate, and these rates can be real values. Rate constraints are expressed as a conjunction of comparisons between clock derivative values, or between a clock derivative value and a natural constant of 1. If no rate constraints are specified at a location, the derivative of the clock is assumed to be constant and equal to 1, similar to a traditional TA.

2 An Alternative Semantics for Timed Automata

Here, we propose a derivative-based alternative semantics for TA with independent clocks, called Timed Automata with independent clocks and clock derivatives (idTA). In our approach, clock derivatives represent the first derivative of independent clocks associated with different processes. This allows us to capture the rate at which the clocks are advancing. The main advantages of our derivative-based alternative semantics are: (1) Our alternative semantics can accommodate multi-timed words [11], which provide a more expressive representation of timing behavior in distributed systems. In addition, rate constraints allow us to precisely define the rates at which clocks evolve. (2) With our alternative semantics, it becomes possible to analyze the local behavior of individual components independently.

Definition 1 (**Rates** [1]). *A rate is a tuple $\tau = (\tau_q)_{q \in Proc}$ of local time functions. Each local time function τ_q maps the reference time to the time of process q, i.e, $\tau_q : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$. The functions τ_q must be continuous, strictly increasing, divergent, and satisfy $\tau_q(0) = 0$.*

Definition 2 (**Rate Constraints**). *Let Proc be a non-empty set of processes. Let X be a finite set of clocks. Let X be a finite set of clocks and $\pi : X \to Proc$. Let $\dot{X}$ be a finite set of time clock derivatives. The set $\Psi(X)$ of rate constraints over the set of clocks X is given by the following grammar:*

$$\psi := true \mid \dot{x} \sim 1 \mid \dot{x} \sim \dot{y} \mid \psi_1 \ \wedge \ \psi_2$$

where $\dot{x}, \dot{y} \in \dot{X}$, $x, y \in X$ and $\sim \in \{<, >, \leq, \geq, =\}$. A rate constraint ψ is a conjunction of comparisons of two clock derivative values or a clock derivative value with a natural constant 1. A classical clock (as in TA) will be described by $\dot{x} = 1$.

Definition 3 (**idTA**). *An idTA is a tuple $\mathcal{A} = (S, s_0, \Sigma, X, \to_{dmta}, Inv, R, F, \pi)$ over Proc where: (1) S is a finite set of locations, (2) $s_0 \in S$ is the initial location, (3) Σ is a finite alphabet, (4) X is a finite set of clock names, (5) $\to_{dmta} \subseteq S \times \Sigma \times \Phi(X) \times 2^X \times S$ is the finite transition relation, (6) $Inv : S \to \Delta(X)$ associates to each location with a clock invariant, (7) $R : S \to \Psi(X)$ associates to each location with a rate constraint, (8) $F \subseteq S$ is a finite set of final locations, (8) $\pi : X \to Proc$ maps each clock to a process.*

Definition 4. *Let Proc be a non-empty set of processes. Let X be a finite set of clocks and $\pi : X \to Proc$. Given a rate constraint $\psi \in \Psi(X)$ and a tuple of functions $\tau \in Rates$, we note τ satisfies ψ at time t, as $(\tau, t) \models \psi$. In particular, the formal definition is as follows:*

$$(\tau, t) \models \dot{x} \sim 1 \Longleftrightarrow \tau_x \text{ is derivable at } t \text{ and } d\tau_x/dt(t) \sim 1$$

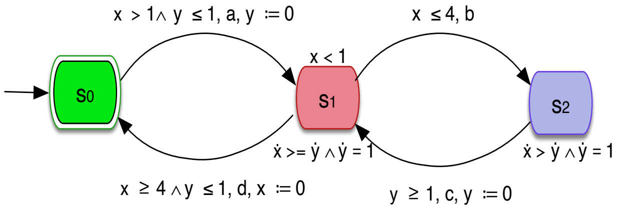

Figure 1: An idTA $\mathcal{M}$

$$(\tau, t) \models \dot{x} \sim \dot{y} \Longleftrightarrow \tau_x \textit{ is derivable at } t \textit{ and}$$
$$\tau_y \textit{ is derivable at } t \textit{ and } d\tau_x/dt(t) \sim d\tau_y/dt(t)$$
$$(\tau, t) \models \psi_1 \ \wedge \ \psi_2 \Longleftrightarrow (\tau, t) \models \psi_1 \textit{ and } (\tau, t) \models \psi_2$$
$$(\tau, t) \models \textit{true} \Longleftrightarrow \textit{true}$$

The semantics of an idTA is given by a Multi-timed Labeled Transition Systems (MLTS) [11].

Example 1. *Figure 1 above shows an idTA $\mathcal{M}$ with the finite alphabet $\Sigma = \{a, b, c, d\}$, the set of independent clocks $X = \{x, y\}$, and rates constraints $\dot{x} \geq \dot{y}$ and $\dot{y} = 1$ in location s_1, and s_2.*

3 Decidability

We present a fundamental decidable problem used to reason about behavioral equivalence between different components of a DRTA using idTA. Therefore, we will extend the zone-based abstraction [9] to represent multi-timed zone graphs with clock derivatives. We show that our multi-timed bisimulation is decidable over a multi-timed zone graph with clock derivatives.

4 Implementation

The above derivative-based alternative semantics is implemented in a tool called MIMETIC. We have used *Java* 8 version in the implementation of our tool. MIMETIC provides a graphical user interface with two main parts: select the XML file (UPPAAL file) and writing a L_ν formula (text box) [9]. We have used the ANTLR parser generator [13] to read the XML file generated by UPPAAL [14] and a L_ν formula. After parsing the xml input files and the L_ν formula, our tool performs a model checking algorithm to verify that a idTA model satisfies a L_ν formula.

5 Related Work

There are several formalisms based on Timed Labeled Transition Systems (TLTS), such as Timed Input/Output Automata (TIOA) [7], Hybrid Automata (HA) [4], Hybrid Input/Output Automata (HIOA) [10], Multi-Rate Timed Automata (MRTA) [5], Rectangular Hybrid Automata (RHA) [5] which are often used for modeling DRTA [6][3]. Consequently, TIOA, HA, MRTA, RHA and HIOA can be used to analyze distributed algorithms, such as clock synchronization algorithms that use independent clocks [6]. HIOA are an extension of TIOA, where external variables model the continuous information flowing into and out of the system [6]. However, the reachability problem and simulation (and bisimulation) are undecidable for TIOA (and HA, RHA, MRTA and HIOA) [6], but decidable for TA and Network of TA (NTA).

References

[1] S. Akshay, Benedikt Bollig, Paul Gastin, Madhavan Mukund, and K. Narayan Kumar. Distributed timed automata with independently evolving clocks. In *19th International Conference, CONCUR 2008*, volume 5201 of *Lecture Notes in Computer Science*, pages 82–97. Springer, 2008.

[2] Rajeev Alur and David L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, 1994.

[3] Alexandre David, Kim G. Larsen, Axel Legay, Ulrik Nyman, and Andrzej Wasowski. Timed i/o automata: A complete specification theory for real-time systems. In *Proceedings of the 13th ACM International Conference on Hybrid Systems: Computation and Control*, HSCC '10, New York, NY, USA, 2010. ACM.

[4] Thomas A. Henzinger. The theory of hybrid automata. In *Proceedings, 11th Annual IEEE Symposium on Logic in Computer Science, New Brunswick, New Jersey, USA, July 27-30, 1996*, pages 278–292. IEEE Computer Society, 1996.

[5] Thomas A. Henzinger, Peter W. Kopke, Anuj Puri, and Pravin Varaiya. What's decidable about hybrid automata? *J. Comput. Syst. Sci.*, 57(1):94–124, 1998.

[6] Dilsun Kaynar, Nancy Lynch, Roberto Segala, and Frits Vaandrager. *The Theory of Timed I/O Automata, Second Edition.* Morgan & Claypool Publishers, 2nd edition, 2010.

[7] Dilsun K. Kaynar, Nancy Lynch, Roberto Segala, and Frits Vaandrager. Timed i/o automata: A mathematical framework for modeling and analyzing real-time systems. In *Proceedings of the 24th IEEE International Real-Time Systems Symposium*, RTSS '03, pages 166–, Washington, DC, USA, 2003. IEEE Computer Society.

[8] Padmanabhan Krishnan. Distributed timed automata. *In Electr. Notes Theor. Comput. Sci.*, pages 185–200, 1999.

[9] François Laroussinie, Kim Guldstrand Larsen, and Carsten Weise. From timed automata to logic - and back. In *MFCS'95*, pages 529–539, 1995.

[10] Nancy Lynch, Roberto Segala, and Frits Vaandrager. Hybrid i/o automata. *Inf. Comput.*, pages 105–157, 2003.

[11] James Jerson Ortiz, Moussa Amrani, and Pierre-Yves Schobbens. Multi-timed bisimulation for distributed timed automata. In Clark Barrett, Misty Davies, and Temesghen Kahsai, editors, *NASA Formal Methods - 9th International Symposium, NFM 2017, Moffett Field, CA, USA, May 16-18, 2017, Proceedings*, volume 10227 of *Lecture Notes in Computer Science*, 2017.

[12] James Jerson Ortiz, Axel Legay, and Pierre-Yves Schobbens. Distributed event clock automata - extended abstract. In *CIAA'11*, pages 250–263, 2011.

[13] Terence Parr. *The Definitive ANTLR 4 Reference.* Pragmatic Bookshelf, Raleigh, NC, 2 edition, 2013.

[14] The UPPAAL tool. Available at `http://www.uppaal.com/`.