Parameter Synthesis for Timed Automata with Clock-Aware LTL Properties

Nikola Beneš

Faculty of Informatics, Masaryk University,
Botanická 68a, 602 00 Brno, Czechia
xbenes3@fi.muni.cz

Abstract. The parameter synthesis problem for timed automata is undecidable in general even for very simple reachability properties. We introduce restrictions on parameter valuations under which the parameter synthesis problem is decidable for Clock-Aware LTL properties. Clock-Aware LTL is an extension of LTL that allows to reason about the clock values in the atomic propositions. We propose a symbolic zone-based method for the problem which can be significantly faster than the naive parameter scan solution. Our technique adapts the ideas of the automata-based approach to Clock-Aware LTL model checking of timed automata.

Keywords: Parameter synthesis · Parametric timed automata · Linear temporal logic · Clock-aware linear temporal logic

Model checking [15] is a formal verification technique applied to check for logical correctness of discrete distributed systems. While it is often used to prove the unreachability of a bad state (such as an assertion violation in a piece of code), with a proper specification formalism, such as the *Linear Temporal Logic* (LTL), it can also check for many interesting liveness properties of systems, such as repeated guaranteed response, eventual stability, live-lock, etc.

Timed automata have been introduced in [2] and have emerged as a useful formalism for modelling time-critical systems as found in many embedded and cyber-physical systems. The formalism is built on top of the standard finite automata enriched with a set of real-time clocks and allowing the system actions to be guarded with respect to the clock valuations. In the general case, such a timed system exhibits infinite-state semantics (the clock domains are continuous). Nevertheless, when the guards are limited to comparing clock values with integers only, there exists a bisimilar finite state representation of the original infinite-state real-time system referred to as the region abstraction [1]. A practically efficient abstraction of the infinite-state space came with the so called zones [16]. The zone-based abstraction is much coarser and the number of zones *reachable* from the initial state can be significantly smaller. This in turn allows for an efficient implementation of verification tools for timed automata, see e.g. UPPAAL [9].

Very often the correctness of a time-critical system relates to proper timing, i.e. it does not only depend on the logical result of the computation, but also on the time at which the results are produced. To that end the designers are not only in the need of tools to verify correctness once the system is fully designed, but also in the need of tools that would help them derive proper time parameters of individual system actions that would make the system as a whole satisfy the required specification. After all this problem of *parameter synthesis* is more urgent in practice than the verification as such.

The problem of the existence of a parameter valuation for a reachability property of a parametric timed automaton in continuous time has been shown to be undecidable in [5, 23] for a parametric timed automaton with as few as 3 clocks. This problem remains undecidable even for integer-valued parameters [10]. A partial solution for the parameter synthesis problem with reachability properties is presented in [19] where the authors provide a semi-decision algorithm which is not guaranteed to terminate in all cases. The authors also introduce a subclass of parametric timed automata, called L/U automata for which the emptiness problem is decidable. Decidability results for the class of L/U automata are further extended in [7, 14]. L/U automata introduce a significant structural restriction to the model of timed automata whereas our goal is to provide a parameter synthesis method for timed automata without any structural restrictions.

To obtain a decidable version of the parameter synthesis problem for timed automata it is sufficient to restrict parameter valuations to bounded integers. When modelling a real-time system, designers can usually provide practical bounds on the time parameters of individual system actions. Therefore, introducing a parameter synthesis method with such a restriction is still reasonable. In [20] the authors show that the problem of the existence of a bounded integer parameter value such that a given property is satisfied is PSPACE-complete for a significant number of properties, which include Timed Computational Tree Logic. They give symbolic algorithms for reachability and unavoidability properties only. While we work with the same model and restrictions as the authors in [20], the fundamental difference is in the supported class of properties. We focus on an extension of the linear temporal logic called Clock-Aware LTL (CA-LTL), which we have defined in [13]. In this logic, the atomic propositions may compare the clocks of the timed automaton against constants.

There is a plethora of other derivatives of linear temporal logics for the specification of properties of real-time systems, timed automata in particular. To name at least some of them, we list TPTL [4], MTL [21], MITL [3], RTTL [24], XCTL [18], CLTL [17], LTLC [22]. These logics employ various ways of expressing time aspects of underlying systems including one global time clock, time-bounded temporal operators, timing variables with quantifiers, and freeze operators. Some logics are defined with the use of time sampling semantics, which has been shown to be counter-intuitive [6]. The key aspect differentiating CA-LTL from the other logics mentioned above is the ability to properly and intuitively reason about clock values in the classical continuous-time semantics while still preserving practical efficiency of the model checking process.

Similar qualities are found in the branching time logic TCTL [8], a subset of which is actually supported by the UPPAAL tool. We stress that CA-LTL is able to reason about values of clocks in timed automata while still being practically simple enough to allow for an efficient model checking procedure as well as parameter synthesis. Note that the inclusion of time-bounded operators, such as the *until* operator of TCTL, would lead to the expressive power of at least MTL, model checking of which is considered computationally infeasible. CA-LTL can thus be seen as a practically motivated extension of LTL, which is powerful enough to express the same properties as the specification language of the world-wide leading timed automata verification tool UPPAAL.

Our main goal is thus to solve the parameter synthesis problem for CA-LTL properties of parametric timed automata. Our solution has three main parts. We first describe a solution to the parameter synthesis problems for standard LTL specifications. The method is based on the concept of parametric zones, which we extend with a new notion of a parametric extrapolation abstraction. The resulting symbolic state-space annotated with symbolic description of parameter sets is then explored using a novel algorithm based on the well known Nested DFS, which we call the *Cumulative Nested DFS* (CNDFS). The implementation of the algorithm employs heuristics to deal with the problem of non-uniqueness of symbolic states representation. The heuristics are based on the concept of an integer hull.

As a next step, we build a model checking procedure for (non-parametric) timed automata with CA-LTL properties. We note that the standard zone-based approach does not work here and introduce a novel concept of the so-called *ultraregions* to solve the problem. Intuitively, the ultraregions keep track of which atomic propositions containing clock comparisons are true in a given state and which can be true in the future. We define a new zone-ultraregion semantics of a timed automaton (with respect to a given set of clock comparisons) and show how model checking can be done with standard automata-based techniques using this semantics.

Finally, we combine the two previous steps into a general CA-LTL parameter synthesis algorithm. We extend the symbolic semantics based on parametric zones to also include ultraregions and show how the CNDFS can be used to deal with the resulting state space. The algorithm is implemented in our proof-of-concept tool available online at https://paradise.fi.muni.cz/parameterSynthesis/.

This presentation is based on work done in cooperation with Peter Bezděk, Ivana Černá, Vojtěch Havel, and Jiří Barnat; the results presented here were published in the papers [11–13].

References

1. Alur, R., Courcoubetis, C., Dill, D.L.: Model-Checking for Real-Time Systems. In: LICS '90). pp. 414–425. IEEE Computer Society (1990)
2. Alur, R., Dill, D.L.: A Theory of Timed Automata. Theor. Comput. Sci. **126**(2), 183–235 (1994)
3. Alur, R., Feder, T., Henzinger, T.A.: The Benefits of Relaxing Punctuality. J. ACM **43**(1), 116–146 (1996)
4. Alur, R., Henzinger, T.A.: A Really Temporal Logic. J. ACM **41**(1), 181–204 (1994)

5. Alur, R., Henzinger, T.A., Vardi, M.Y.: Parametric Real-Time Reasoning. In: Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing. pp. 592–601. ACM (1993)
6. Alur, R., Madhusudan, P.: Decision Problems for Timed Automata: A Survey. In: Formal Methods for the Design of Real-Time Systems. LNCS, vol. 3185, pp. 1–24. Springer (2004)
7. André, É., Lime, D.: Liveness in L/U-parametric timed automata. In: ACSD 2017. pp. 9–18. IEEE Computer Society (2017)
8. Baier, C., Katoen, J.P.: Principles of Model Checking. The MIT Press (2008)
9. Behrmann, G., David, A., Larsen, K.G., Håkansson, J., Pettersson, P., Yi, W., Hendriks, M.: UPPAAL 4.0. In: Third International Conference on the Quantitative Evaluation of Systems (QEST 2006). pp. 125–126. IEEE (2006)
10. Beneš, N., Bezděk, P., Larsen, K.G., Srba, J.: Language Emptiness of Continuous-Time Parametric Timed Automata. In: ICALP 2015. LNCS, vol. 9135, pp. 69–81. Springer (2015)
11. Bezděk, P., Beneš, N., Barnat, J., Černá, I.: LTL Parameter Synthesis of Parametric Timed Automata. In: Software Engineering and Formal Methods. LNCS, vol. 9763, pp. 172–187. Springer (2016)
12. Bezděk, P., Beneš, N., Černá, I., Barnat, J.: On clock-aware LTL parameter synthesis of timed automata. J. Log. Algebr. Meth. Program. **99**, 114–142 (2018)
13. Bezděk, P., Beneš, N., Havel, V., Barnat, J., Černá, I.: On Clock-Aware LTL Properties of Timed Automata. In: Theoretical Aspects of Computing - ICTAC 2014. LNCS, vol. 8687, pp. 43–60. Springer (2014)
14. Bozzelli, L., La Torre, S.: Decision Problems for Lower/Upper Bound Parametric Timed Automata. Formal Methods in System Design **35**(2), 121–151 (2009)
15. Clarke, E., Grumberg, O., Peled, D.: Model Checking. The MIT press (1999)
16. Daws, C., Tripakis, S.: Model Checking of Real-Time Reachability Properties Using Abstractions. In: Tools and Algorithms for Construction and Analysis of Systems. LNCS, vol. 1384, pp. 313–329. Springer (1998)
17. Demri, S., D'Souza, D.: An Automata-Theoretic Approach to Constraint LTL. Information and Computation **205**(3), 380–415 (2007)
18. Harel, E., Lichtenstein, O., Pnueli, A.: Explicit Clock Temporal Logic. In: Proceedings of the Fifth Annual Symposium on Logic in Computer Science (LICS '90). pp. 402–413 (1990)
19. Hune, T., Romijn, J., Stoelinga, M., Vaandrager, F.W.: Linear Parametric Model Checking of Timed Automata. The Journal of Logic and Algebraic Programming **52-53**, 183–220 (2002)
20. Jovanović, A., Lime, D., Roux, O.H.: Integer Parameter Synthesis for Real-Time Systems. IEEE Trans. Software Eng. **41**(5), 445–461 (2015)
21. Koymans, R.: Specifying Real-Time Properties with Metric Temporal Logic. Real-Time Systems **2**(4), 255–299 (1990)
22. Li, G., Tang, Z.: Modelling Real-Time Systems with Continuous-Time Temporal Logic. In: Formal Methods and Software Engineering. LNCS, vol. 2495, pp. 231–236. Springer (2002)
23. Miller, J.S.: Decidability and Complexity Results for Timed Automata and Semi-linear Hybrid Automata. In: Hybrid Systems: Computation and Control. LNCS, vol. 1790, pp. 296–309. Springer (2000)
24. Ostroff, J.S.: Temporal Logic for Real-Time Systems, vol. 40. Research Studies Press Advanced Software Development Series (1989)