Beyond Model Checking: Parameters Everywhere

Étienne André[1], Benoît Delahaye[2], Peter Habermehl[3], Claude Jard[2], Didier Lime[4],
Laure Petrucci[1], Olivier H. Roux[4], Tayssir Touili[3]

[1] Université Paris 13, Sorbonne Paris Cité, LIPN, CNRS, UMR 7030, Villetaneuse, France
[2] LINA/Université de Nantes, France
[3] LIAFA, Université Paris Diderot – Paris7, France
[4] IRCCyN, École Centrale de Nantes, France

Keywords: parameter synthesis, parametric model checking, discrete parameters, continuous parameters

1 Context and Motivation

Beyond Model Checking… After many years of academic research on model checking, its impact in industry is mostly limited to critical embedded systems, and thus is somewhat disappointing w.r.t. the expectations. Two major reasons are the binary response to properties satisfaction, which is not informative enough, and the insufficient abstraction to cater for tuning and scalability of systems.

A major challenge is to overcome these limitations by providing parametric formal methods for the verification and automated analysis of systems behaviour.

…are Parameters The challenge is clearly to obtain guarantees on the quality of the systems in operation, quality being evaluated during the design phase. For any given level of abstraction, we want to maintain the formal description of the behaviour of the system together with its expected properties. The current verification techniques ensure that the properties are true for all possible behaviours of a given instance and environment of the system. Hence the utmost importance of a characterisation of the conditions under which the properties are guaranteed to hold, in particular since systems are often incompletely specified or with an environment unknown *a priori*.

In order to broaden the applicability of formal modelling methods within the wide range of digital world that is being built, a key point is the control of abstraction in the models. A main challenge is to develop the theory and implementation of the verification of parametrised models. This area of research is still in its infancy and a significant advance should be performed, by a coordinated study of several types of parameters: discrete (e.g. number of threads, size of counters), timed (deadlines, periods), continuous costs (energy, memory), and probabilistic (redundancy, reliability).

Being able to treat these parametrised models constitutes a scientific breakthrough in two ways:

- It significantly increases the level of abstraction in models. It will be possible to handle a much larger and therefore more *realistic class of models*.

- The existence of parameters can also address more relevant and *realistic verification issues*. Instead of just providing a binary response to the satisfaction of expected properties, constraints on the parameters can be synthesised. These constraints can either ensure satisfaction of the expected properties when this is possible, or provide quantitative information in order to *optimise* satisfaction of *some* properties w.r.t. parameter values. Such information are highly valuable to designers for the proper behaviour of the systems they develop.

Towards a Safe Digital Society With the booming broadening of software and hardware devices in our lives, the need for safe, secure and predictable systems becomes higher and higher. Hence, methods for formally verifying these systems are strongly needed. Model checking techniques used in the design phase of a system prove the system either correct or incorrect, in which case the design phase may have to restart from the beginning, thus implying a high cost. This binary answer is certainly one of the key reasons explaining why formal methods are not as widespread as they could be. Parameter synthesis overcomes this drawback by directly providing the designer with sufficient working conditions, hence allowing to consider systems only partially specified, or with an only partially known environment. Efficient and effective parameter synthesis techniques shall broaden the use of formal methods in the software and hardware industry towards a safe digital society. The modelling and derivation of formal conditions ensuring a good behaviour is a clear step towards a digital and software industry able to guarantee and ensure its products, thus becoming a more mature industry. This is in particular of utmost importance for the development of the open source software industry.

2 Challenges and Agenda

One of the key challenges in the area of parameter synthesis, that we hope to be solved in 2025, is the definition of *decidable* subclasses of existing formalisms and problems. Almost all interesting parameter synthesis problems for formalisms such as parametric timed automata (PTA) [AHV93] or parametric time Petri nets [TLR09] are known to be undecidable in the general case. However, in the past few years, some problems were shown to be decidable, in particular integer parameter synthesis [JLR13a], or characterization of the system robustness (see, e.g., [Mar11]), which are subproblems of the main parameter synthesis issue.

Decidability problems may appear to be disconnected from applications, but they are not: although undecidable problems may yield useful semi-algorithms that can output interesting results, finding decidable subclasses of models is an incentive for scientists to seek efficient algorithms (that always terminate, by definition).

Studying concurrent systems with both discrete and continuous parameters can lead to several types of parameters (discrete, timed, hybrid, probabilistic), and combining them can lead to many different problems. We believe that the ultimate goal would be to combine all kinds of parameters in a single model. This also implies the definition of adequate formalisms, either decidable, or with efficient semi-decidable algorithms.

Discrete Parameters Regular model checking (RMC) techniques [BJNT00,BHH+08,BHRV12] and cut-off based algorithms [CTTV04,BHV08] apply to the analysis of systems where the number of entities is *a priori* unknown, but not to the analysis of all parametrised systems. Indeed, RMC addresses systems with linear or tree-like topologies, and cut-off techniques particular kinds of systems in an *ad hoc* manner. A first goal will be to develop techniques as general as RMC but that apply to general topologies. One way consists in extending the RMC framework to deal with graphs and to develop techniques based on graph automata for the symbolic representation of (infinite) sets of graphs.

A second goal is to deal with timed models with discrete parameters, where some discrete components such as the number of processes are *a priori* unknown. The discrete parametric model checking problem for timed models is likely to be undecidable. A first direction will be to consider subclasses of timed automata (or time Petri Nets) with discrete parameters where the abstract state space of the timed part of the model (zone graph, state class graph) could be handled with an extension of the RMC framework based on their topology properties. Another direction will be consider decidable subclasses of this parameter synthesis problem (e.g. bounded parameters) and propose efficient symbolic synthesis algorithms based on symbolic state space abstractions.

Timing Parameters The parameter synthesis problem is known to be undecidable for PTA [AHV93,BLT09], but decidable for subclasses such as L/U automata [HRSV02], although this model has a strong syntactical restriction for practical purposes. In [JLR13a] an approach based on restricting to integers the possible values of the parameters, leads to decidability. Although extending to rationals this results appears to be possible for non-reachability properties, it remains to be done for more elaborated properties (such as unavoidability, equality of trace sets [ACEF09], games [JLR13b], etc.). These results should be extended so as to exhibit subclasses of PTA for which parameter synthesis (possibly under- or over-approximated) is guaranteed to terminate.

From these results and those related to discrete parameter synthesis, a further goal will be to synthesise constraints of good behaviour based on both these timing parameters and the discrete parameters.

Cost Parameters A challenge is to investigate the use of richer dynamics in models to make them suitable for the modelling of a wider range of applications, such as energy consumption. This leads to the so-called generic *hybrid* setting, in which the continuous variables may have dynamics defined by arbitrary differential equations. The study of this class of models is notoriously difficult and the decidability results are scarce in this area. In terms of parametrisation, two decidable subclasses of hybrid automata seem promising: O-minimal automata [LPS00,BMRT04] and interrupt timed automata (ITA) [BHS12]. First, O-minimal automata feature extremely rich dynamics but each discrete transition must reset all continuous variables. Interrupt timed automata (ITA) have been introduced with the aim to describe timed multitask systems with interruptions in a single processor environment. The accepted language of ITA is incomparable to the one of TA, and reachability is decidable.

Furthermore, weighted or priced models [ALP04,BFH$^+$01], restrict the richer dynamics to *cost* variables that are never tested, only updated, and therefore do not participate in the actual trajectory of the system. Previous results show that the question of knowing if there exists some parameter values such that so location is reachable within T time units, for some given T, is undecidable for PTA.

In both cases, it will then be challenging to extend the obtained decidable parametrised subclasses with parametrised discrete behaviours, much as for continuous parameters.

Probabilistic Parameters In real-life applications, probabilities are often used as a building block that allows abstracting from physical constraints or unknown environments. Hence, a challenge is to extend the models considered above with probabilities, and study parametric probabilistic timed systems where parameters can range over time constraints, cost variables *and* transition probabilities. Obtaining fundamental results in this domain would carry much weight as they would impact many applicative fields.

In another setting, probabilities can also be seen as a tool for synthesising optimal values of parameters: probabilities can be artificially injected on the parameter space of *non-probabilistic* parametrised systems, and Statistical Model Checking (SMC) [LDB10] can then be used in order to identify regions in the parameter space that optimise given properties. Since SMC is still at its early stages, it suffers from many limitations that will have to be overcome in order to produce significant results.

Applications Beyond classical applications (hardware verification, process management, embedded and cyber-physical systems), typical applications in the near future are *smart homes*, in particular catering for elderly or disabled people in a safe manner. Parametrisation there characterises the adaptation of the system to a specific subject, either in a static manner (list of parameters to be instantiated when the managing software is installed for a specific person) or in a dynamic manner (parameters regularly improved following new living conditions). Additionally, the use of costs in parameter synthesis typically addresses the reduction of energy consumption, either by managing the home, or performing medical surveillance through sensor networks. More generally, distributed applications (with a variable number of processes, of local environment) will be a natural application of both discrete and continuous parameter synthesis.

Acknowledgements We would like to thank the anonymous reviewers for their useful comments.

References

ACEF09. É. André, T. Chatain, E. Encrenaz, and L. Fribourg. An inverse method for parametric timed automata. *IJFCS*, 20(5):819–836, 2009.

AHV93. R. Alur, T. A. Henzinger, and M. Y. Vardi. Parametric real-time reasoning. In *STOC*, 1993.

ALP04. R. Alur, S. La Torre, and G. J. Pappas. Optimal paths in weighted timed automata. *Theoretical Computer Science*, 318(3):297–322, 2004.

BFH$^+$01. G. Behrmann, A. Fehnker, T. Hune, K. Larsen, P. Pettersson, J. Romijn, and F. Vaandrager. Minimum-cost reachability for priced timed automata. In *HSCC*, 2001.

BHH$^+$08. A. Bouajjani, P. Habermehl, L. Holík, T. Touili, and T. Vojnar. Antichain-based universality and inclusion testing over nondeterministic finite tree automata. In *CIAA*, 2008.

BHRV12. A. Bouajjani, P. Habermehl, A. Rogalewicz, and T. Vojnar. Abstract regular (tree) model checking. *STTT*, 14(2):167–191, 2012.

BHS12. B. Bérard, S. Haddad, and M. Sassolas. Interrupt timed automata: verification and expressiveness. *Formal Methods in System Design*, 40(1):41–87, 2012.

BHV08. A. Bouajjani, P. Habermehl, and T. Vojnar. Verification of parametric concurrent systems with prioritized FIFO resource management. *FMSD*, 32(2):129–172, 2008.

BJNT00. A. Bouajjani, B. Jonsson, M. Nilsson, and T. Touili. Regular model checking. In *CAV*, 2000.

BLT09. L. Bozzelli and S. La Torre. Decision problems for lower/upper bound parametric timed automata. *Formal Methods in System Design*, 35(2):121–151, 2009.

BMRT04. T. Brihaye, C. Michaux, C. Rivière, and C. Troestler. On O-minimal hybrid systems. In *HSCC*, 2004.

CTTV04. E. M. Clarke, M. Talupur, T. Touili, and H. Veith. Verification by network decomposition. In *CONCUR*, 2004.

HRSV02. T. Hune, J. Romijn, M. Stoelinga, and F. W. Vaandrager. Linear parametric model checking of timed automata. *JLAP*, 52-53, 2002.

JLR13a. A. Jovanović, D. Lime, and O. H. Roux. Integer parameter synthesis for timed automata. In *TACAS*, 2013.

JLR13b. A. Jovanović, D. Lime, and O. H. Roux. Synthesis of bounded integer parameters for parametric timed reachability games. In *ATVA*, volume 8172 of *Lecture Notes in Computer Science*, pages 87–101. Springer, 2013.

LDB10. A. Legay, B. Delahaye, and S. Bensalem. Statistical model checking: An overview. In *RV*, 2010.

LPS00. G. Lafferriere, G. Pappas, and S. Sastry. O-minimal hybrid systems. *MCSS*, 13:1–21, 2000.

Mar11. N. Markey. Robustness in real-time systems. In *SIES*, pages 28–34. IEEE Computer Society Press, 2011.

TLR09. L.-M. Traonouez, D. Lime, and O. H. Roux. Parametric model-checking of stopwatch Petri nets. *Journal of Universal Computer Science*, 15(17):3273–3304, 2009.