Project no.: ICT-FP7-STREP-214755

Project full title: Quantitative System Properties in Model-Driven Design

Project Acronym: QUASIMODO

Deliverable no.: D2.1

Title of Deliverable: Model-checking of real-time probabilistic models

Contractual Date of Delivery to the CEC:	**Month 12**
Actual Date of Delivery to the CEC:	**Month 12 (February 1, 2009)**
Organisation name of lead contractor for this deliverable:	**P04 RWTH**
Author(s):	**Joost-Pieter Katoen, Holger Hermanns, Kim G Larsen, Jean-Francois Raskin Alexandre David, and Francois Laroussine**
Participants(s):	**P04 RWTH, P01 AAU P03 CNRS, P05 SU, P06 CVF**
Work package contributing to the deliverable:	**WP 2**
Nature:	**R**
Version:	**1.0**
Total number of pages:	**21**
Start date of project:	1 Jan. 2008 **Duration:** 36 month

Project co-funded by the European Commission within the Seventh Framework Programme (2007-2013)		
Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Abstract:

This deliverable describes the first year results of the QUASIMODO project on analysing quantitative systems.

Keyword list: Markov chain, Markov decision process, probabilistic bisimulation, probabilistic simulations; probabilistic timed automata; priced probabilistic timed automata; Continuous Time Markov Chains, inhomogeneous CTMC, Infinite state CTMC, counter-example guided abstraction-refinement

Contents

Abbreviations

AAU: Aalborg University, DK

CFV: Centre Fèdèrè en Vèrification, B

CNRS: National Center for Scientific Research, FR

ESI: Embedded Systems Institute, NL

ESI/RU: Radboud University Nijmegen, NL

LSV: LSV, ENS Cachan, F

RWTH: RWTH Aachen University, D

SU: Saarland University, D

1 Introduction

This deliverable presents an overview of the work that has been done within the Workpackage 2 on Analysis for quantitative models during the first year of the QUASIMODO project. In full agreement with the *Description of Work* emphasis has been on extensions of existing model checking techniques to deal with multiple quantitative aspects, in particular real-time and probabilistic aspects. More precisely, we have worked on the following topics:

1. Analysis of discrete Markov chains and Markov decision processes;
2. Analysis of probabilistic timed automata;
3. Analysis of continuous Markov chains.

The next sections summarize the results that we obtained in these respective areas, and discuss their relevance and future perspectives for the project.

2 Discrete Markov Chains

2.1 Equivalence of Labeled Markov Chains

Participants

- Laurent Doyen (CFV), Thomas A. Henzinger (EPFL) and Jean-Francois Raskin (CFV)

Results

We consider the equivalence problem for probabilistic machines. Two probabilistic machines are equivalent if every finite sequence of observations has the same probability of occurrence in the two machines. We show that deciding equivalence can be done in polynomial time for labeled Markov chains, using a reduction to the equivalence problem for probabilistic automata, which is known to be solvable in polynomial time. We provide an alternative algorithm to solve the equivalence problem, which is based on a new definition of bisimulation for probabilistic machines. We also extend that technique to decide equivalence of weighted probabilistic automata. Then, we consider the equivalence problem for labeled Markov decision processes (LMDPs) which asks given two LMDPs whether for every way of resolving the deci- sions in each of the processes, there exists a way of resolving the decisions in the other process such that the resulting probabilistic machines are equivalent. We show that the strategies used to resolve the decisions can be restricted to be observation-based, but may require infinite memory.

This work is published in [47].

Perspectives

The decidability of the LMDP problem problem remains open, and will be subject for further investigation in the coming periods.

2.2 Flow Faster: Efficient Decision Algorithms for Probabilistic Simulations

Participants

- Lijun Zhang (SU), Holger Hermanns (SU), Friedrich Eisenbrand (SU), David N. Jansen (ESI/RU)

Results

Strong and weak simulation relations have been proposed for Markov chains, while strong simulation and strong probabilistic simulation relations have been proposed for probabilistic automata. However, decision algorithms for strong and weak simulation over Markov chains, and for strong simulation over probabilistic automata are not efficient, which makes it as yet unclear whether they can be used as effectively as their non-probabilistic counterparts. This paper presents drastically improved algorithms to decide whether some (discrete- or continuous-time) Markov chain strongly or weakly simulates another, or whether a probabilistic automaton strongly simulates another. The key innovation is the use of parametric maximum flow techniques to amortize computations. We also present a novel algorithm for deciding strong probabilistic simulation preorders on probabilistic automata, which has polynomial complexity via a reduction to an LP problem. When extending the algorithms for probabilistic automata to their continuous-time counterpart, we retain the same complexity for both strong and strong probabilistic simulations.

This work is published in Logical Methods in Computer Science.

Perspectives

Efficient methods for establishing simulation relationships between Markov chains will be particular useful as a tool for establishing refinements between models operating at different levels of abstraction. Here, the algorithms for weak simulation will be of particular relevance.

3 Probabilistic Timed Automata

3.1 Model checking probabilistic timed automata with one or two clocks.

Participants

- Marcin Jurdzinski (Warwick), Francois Laroussinie (LSV), Jeremy Sproston(Torino)

We have studied model-checking algorithms for *probabilistic timed automata* (PTA) that are a variant of timed automata extended with discrete probability distributions. This formalism has already been used to model many industrial systems (for example the IEEE 1394 root contention protocol). Given a property P stated with the probabilistic timed temporal logic PTCTL, there exist algorithms to decide whether a given PTA A satisfies or not P. Nevertheless this problem has a high complexity. Thus we have studied the case of PTAs containing only one clock. In this case, we have shown that there exist efficient model-checking algorithms when the property is stated either with PCTL (with quantitative probabilities but without timed constraint) or with the sublogic $\mathrm{PTCTL}_{<,>}{}^0/1$ in which (1) punctual timing bounds and (2) comparisons with probability bounds other than 0 or 1, are disallowed. We have also proved that verifying (simple) reachability properties becomes EXPTIME-complete for PTA with two or more clocks.

The work is published in [15]

3.2 A probabilistic semantics for timed automata

Participants

- Christel Baier (Dresden), Nathalie Bertrand (Dresden) , Patricia Bouyer (LSV), Thomas Brihaye (Mons), Marcus Grösser, Nicolas Markey (LSV).

Results

To model systems with probabilities and constraints on time, we defined a probabilistic semantics to timed automata that randomizes both delays and discrete choices. That way we can give a measure to sets of paths satisfying ω-regular properties, and thus how likely a property is satisfied in a timed automaton. We proved that the almost-sure model-checking of ω-regular properties is decidable in PSPACE in single-clock timed automata. Also we showed how to compute an approximation of the probability of ω-regular properties in a subclass of timed automata.

The work is published in [1, 9].

Perspectives

Investigation of the probabilistic semantics in the presence of multi-clock timed automata remains to be done.

Probabilistic semantics for timed automata (as proposed in this work) seem related to robustness of timed automata: in both case “unlikely” behaviour is addressed (e.g. as a consequence of having singleton guards of the type $x = 4$). However, a formal relationship between the two approaches remains unsettled.

3.3 Towards Cost-bounded Reachability in Priced Probabilistic Timed Automata

Participants

- Jasper Berendsen (ESI/RU), Taolue Chen and David N. Jansen (ESI/RU)

Results

Recently the model of Priced Probabilistic Timed Automata (PPTA) has been put forward, which are timed automata extended with price rates in locations and discrete probabilistic branching. The model is a natural combination of Priced Timed Automata and Probabilistic Timed Automata. In this work we focus on cost-bounded probabilistic reachability for PPTA, which determines if the maximal probability to reach a set of goal locations within a given price bound (and time bound) exceeds a threshold $p \in [0, 1]$. We prove undecidability of the problem for simple PPTA with 3 clocks. Simple PPTA, a. o., have strictly positive price rates. Undecidability also holds for PPTA with 2 clocks that allow negative price rates. Our undecidability results show that it is difficult to weaken this assumption.

This work is under submission.

Perspectives

Identification of restrictions (one-clock, strong cost-non zenoness) under which decidability may be obtained is under consideration.

3.4 Efficient Model Checking for Probabilistic Timed Automata

Participants

- Alexandre David (AAU), Robert J. Engdahl (AAU), Kim G. Larsen (AAU), Arild Martin M. Haugstad (AAU)

Results

In this work, we provide an efficient tool implementation – UPPAAL PRO – supporting the analysis of Probabilistic Timed Automata (PTA), an extension of timed automata with discrete probability distributions [?]. Semantically, PTA describe an infinite Markov decision process in which the non-deterministic choice include choice of delay quantities in locations. The key problems supported by UPPAAL PRO are those of determinining infimum and supremum reachability probabilities for a given set of goal states (including possibly an upper time-bound). The implementation is fully integrated with UPPAAL allowing the full use of the UPPAAL modeling formalism including parallel composition with a variety of synchronization mechanisms (e.g. two-way, broadcast, urgency), the use of discrete variables, array and record types as well as user-defined types and functions.

Following the efficient on-the-fly exploration paradigm applied in all branches of UPPAAL's, the tool implementation applies forward state space exploration of the underlying timed automata. The forward exploration is – as in UPPAAL TIGA – combined with backwards stabalization aiming at producing a coarse(st) finite partitioning of the state space. The final partitioning has the properties of a (probabilistic) time-abstracted bisimulation and constitutes a finite-state Markov decision process based on which the desired infimum and supremum probabilities can be obtained using linear programming.

Perspectives

The number of states in the Markov decision process corresponding to the final (stable) partitioning may exceed the capability of current linear programming technology. However – as demonstrated in RaPTURE[1] – it is possible to compute safe approximations of the reachability probabilities based on intermediate partitionings; and, the approximate probability may be sufficient (small or large) with respect to a given threshold and if not further refinement may be performed.

The implemented partitioning algorithm of UPPAAL PRO provides the possibility of obtaining efficient implementations of a number of problems which have so far not been supported. This include:

- efficient computation of optimal infinite schedules for priced timed automata; at present this problem is known to be decidable [?] using so-called corner-point region abstractions, but no efficient zone-based algorithm has been suggested.

[1] RaPTURE [?] is a tool for symbolic analysis of finite-state Markov decision processes using abstraction/refinement.

- full TCTL model checking

Future perspectives included support of a probablistic simulator in UPPAAL. Here the simulation should guided by the discrete probabilistic information in the PTA and possibly probability distributions on delays as given in [1, 9]. The probablistic simulator could be used for performance analysis.

4 Continuous Time Markov Chanis

4.1 Inhomogeneous continuous-time Markov chains

Participants

- Alexandre Mereacre (RWTH), Joost-Pieter Katoen (RWTH)

Results

Continuous-time Markov chains (CTMCs) are applied in a large range of applications, ranging from transportation systems to systems biology, and are a popular model in performance and dependability analysis. These Markov chains are typically homogeneous, i.e., the rates that determine the speed of changing state as well as the probabilistic nature of mode transitions are constant. However, in some situations constant rates do not adequately model real behavior. This applies, e.g., to failure rates of hardware components (that usually depend on the component's age), battery depletion (where the power extraction rate non-linearly depends on the remaining amount of energy), and random phenomena that are subject to environmental influences such as temperature. In these circumstances, Markov models with *inhomogeneous* rates, i.e., rates that are time-varying functions, are more appropriate.

Whereas temporal logics and accompanying model-checking algorithms have been developed for CTMCs [10, 8], the verification of time-inhomogeneous CTMCs (ICTMCs) has not yet been investigated. This paper presents an initial step in that direction by presenting a stochastic variant of the well-known Hennessy-Milner Logic [14] (HML) for ICTMCs. The main ingredient is a simple probabilistic real-time extension of the modal operator $\langle\Phi\rangle$ in (state-based) HML: the formula $\langle\Phi\rangle^{I}_{\geq p}$ asserts that a Φ-state is reachable in the time interval I with likelihood at least p. The adequacy of this extension is justified by the fact that logical equivalence corresponds to strong bisimulation. Opposed to CTMC model checking (where all rate functions are constant), restrictions have to be imposed on the rate functions in order to enable (approximate) model-checking algorithms for ICTMCs. It is shown that verifying our variant of HML for rate functions that are piecewise constant boils down to determining the zeros of an exponential polynomial. Using some standard techniques to solve expolynomials, this results in an approximative verification algorithm for stochastic HML which is exponential in the nesting depth of the formula (i.e., the number of $\langle\Phi\rangle^{I}_{\geq p}$ formulas in sequence), linear in the size of the ICTMC, linear in the number of pieces of a rate function, and logarithmic in the number of bits precision of Newton's method.

This work has bee published in FORMATS 2008.

4.2 Parameter synthesis in CTMCs

Participants

- Alexandre Mereacre (RWTH), Joost-Pieter Katoen (RWTH)

Results

Model checking aims at checking a property, typically stated in some temporal logic, against a given concrete model. A disadvantage of the traditional approaches to model checking is that they can only check the validity of properties under the assumption that all parameter values of the model are known. This means that concrete values of e.g., timing parameters, branching probabilities, costs, and so forth, need to be explicitly given. Although this might be appropriate for the a posteriori verification of concrete system realizations, for design models at a higher level of abstraction this is less adequate.

In practical system design, one is not interested in checking a concrete instance, but rather, often in deriving parameter constraints that can ensure the validity of the property under consideration. Typical examples are failure-repair systems in which components (such as memories or processors) may fail and where only lower- and upper bounds on repair times are known. Rather than determining whether for a certain combination of failure and repair rates, a property holds, one would like to synthesize the set of pairs of rates for which the validity of the property is guaranteed.

We study a parametric version of CTMCs, a novel variant of CTMCs in which rate expressions over variables (with bounded range) indicate the average speed of state changes. They are expressed using the polynomial ring over reals, allowing rate expressions such as $3\alpha{\cdot}\beta$ and $\alpha^3{-}\alpha{\cdot}\beta$. We showed that checking whether time-bounded reachability probabilities meet certain thresholds ($\mathcal{P}_{\bowtie\, p}(\diamondsuit^{\leqslant t} a)$ in CSL formula) amounts to solving a polynomial function over the rate parameters. Our main contribution in this paper [13] is an algorithm that approximates the synthesis region, i.e., the set of rate parameter values for which the validity of an a priori given time-bounded reachability property is guaranteed. Synthesizing these values is done by a (grid) discretization of the initially-given parameter ranges together with a refinement technique. We described the details of this approach for the *two-parameter* setting and showed the polygon approximation of the synthesis region up to any user-defined accuracy that is related to the grid size. The time complexity of our algorithm is quadratic in the number of states in the parametric CTMC, linear in the grid discretization parameter, and polynomial in the expected number of discrete steps taken before reaching the deadline.

This paper appeared in the IEEE Real-Time Systems Symposium 2008.

4.3 Infinite-state continuous time Markov chains

Participants

- Ernst Moritz Hahn (SU), Holger Hermanns (SU), Björn Wachter (SU), and Lijun Zhang (SU).

Results

Since (homogeneous) CTMCs together with their extensions with rewards, are popular means to model performance and dependability of computing systems and the behavior of biological systems, we also consider the model checking problem for CSL over *infinite* CTMCs. In practice, infiniteness occurs in the form of unbounded behavior, such as quantities of substances in biological/chemical models or unbounded queues in queueing systems.

In contrast to earlier work, we consider arbitrarily infinite CTMCs, including rate-unbounded models, and focus on the CSL without the admit steady-state operator. The resulting logic can express (possibly nested) probabilistic timing properties such as: *"is the probability to reach Ψ-states along Φ-states within time interval [6.5,8.5] smaller than* 0.1*"*. For CTMCs, these properties constitute the arguably most important class of CSL formulae. They can express many performance measures, including timed probabilistic reachability, various availability measures like instantaneous availabilities, conditional instantaneous availabilities and interval availabilities.

The technical contribution in this context is a novel CSL model checking algorithm, which works on a finite truncation of the infinite-state model, and this finite truncation is computed on-the fly. The algorithm is extensible to timed reward properties, enables the analysis of infinite (or very large) CTMCs, and applies to arbitrarily structured (finite or infinite) CTMC models. Our truncation-based analysis follows the same general idea as the one in [33] and [34], but (apart from absence of the model checking context) differs significantly in terms of the employed error estimation method.

We have implemented the truncation-based model checker TRUNCATOR, and have assessed the effectiveness of truncation. We also compared different error estimation methods on a number of infinite-state CTMCs, including a protein synthesis model, a Jackson queueing network, and a job processing system. Further, we have also considered finite-state CTMCs with very large state spaces ($> 10^8$ states). In the infinite-state case, truncation proves to be a practical approach to verify models that are not directly amenable to finite-state methods, while, in the finite-state case, significant speedups can be achieved over model checking without truncation.

Remarkably, when applied to a protein synthesis model, the method of [34] turns out to outperform our novel method because, due to its more precise error estimation, it terminates at a shorter truncation depth. However, when applied to other models, that method is slower since the higher cost of the estimation is not amortized by the shorter truncation.

Based on this experimental insight, we investigated different error estimation methods further and proved that they are strictly ordered in terms of precision, and this order is the inverse of the observed runtime order induced by our experimental evaluation. The work is reported in [46] and an invited journal submission (available upon request).

Perspective

The comparison of our approach with other truncation approaches lead us to the insight that in order to obtain a general-purpose infinite-state analysis framework, the error estimation method should be configurable, in order to explore the cost-performance trade-off of error estimation.

4.4 Probabilistic counterexample-guided abstraction-refinement

Participants

- Holger Hermanns (SU), Björn Wachter (SU), and Lijun Zhang (SU).

Results

Model checking of probabilistic behavioral descriptions are an important subject of investigation in the QUASIMODO project. In this work, we consider probabilistic automata [39]. Properties of these systems can be specified by formulas in temporal logics such as PCTL, where for instance quantitative probabilistic reachability ("the probability to reach a set of bad states is at most 3%") is expressible. Model checking algorithms for such logics have been devised mainly for finite-state models, and effective tool support is provided by probabilistic model checkers such as PRISM [17]. Despite its remarkable versatility, the approach is limited by the state explosion problem, aggravated by the cost of numerical computation compared to Boolean CTL model checking.

Predicate abstraction is a method for creating finite abstract models of non-probabilistic systems where symbolic expressions, so-called predicates, induce a partitioning of its (potentially infinite) state space into a finite number of regions. For automation, it is typically coupled with *counterexample-guided abstraction refinement* (CEGAR) where an initially very coarse abstraction is refined using diagnostic information (predicates) derived from abstract counterexamples, until either the property is proved or refuted.

In this work, we investigate how counterexample-guided abstraction refinement can be developed in a probabilistic setting. Predicate abstraction without abstraction refinement has been presented in [44] for a guarded command language strongly inspired by the PRISM input language. Its concrete semantics maps to probabilistic automata, and this is the natural basis for our present work. We restrict our treatment to probabilistic reachability and aim to determine if the probability to reach a set of bad states exceeds a given threshold. We are thus restricting to only determining upper probability bounds.

The core challenge of developing probabilistic CEGAR lies in the notion and analysis of counterexamples. In the traditional setting, an abstract counterexample is a single finite path (to some bad state) and counterexample analysis consists in checking if the concrete model exhibits a corresponding error path. In contrast, a counterexample to a probabilistic reachability property can be viewed as a finite, but generally cyclic, Markov chain. Due to these cycles, probabilistic counterexample analysis is not directly amenable to conventional methods. We circumvent this problem, by preprocessing the abstract counterexample using the strongest evidence idea of [12]: We generate a finite set $\mathcal{S}$ of abstract finite paths that together carry enough abstract probability mass, and formulate the problem of computing the realizable probability mass of $\mathcal{S}$ in terms of a weighted MAX-SMT problem. The set $\mathcal{S}$ is built incrementally in an *on-the-fly* manner, until either enough probability is realizable, or $\mathcal{S}$ cannot be enriched with sufficient probability mass to make the probability threshold realizable, in which case the counterexample is spurious.

These ingredients result in a viable approach to probabilistic CEGAR. We have implemented and evaluated the approach on various case studies. Indeed, CEGAR entirely mechanizes the verification process: predicates are added mechanically on demand based on counterexample analysis. The work is published in [28].

Perspective

We have explored fundamental questions and pragmatic issues of probabilistic counterexample-guided abstraction refinement. While the resulting theory and tool work smoothly, the approach is as yet insufficient in many respects. In particular, we are only providing upper probability bounds, while other abstraction-refinement approaches are tailored to effectively provide also lower probability bounds [18], albeit in technically different settings. In order to achieve the project goals, we are currently trying to incorporate this, and are also aiming at extension of the basic strategy with costs.

Bibliography

[1] Christel Baier, Nathalie Bertrand, Patricia Bouyer, Thomas Brihaye, and Marcus Größer. Almost-sure model checking of infinite paths in one-clock timed automata. In *Proceedings of the 23rd Annual IEEE Symposium on Logic in Computer Science (LICS'08)*, pages 217–226, Pittsburgh, PA, USA, June 2008. IEEE Computer Society Press.

[2] Martin De Wulf, Laurent Doyen, Nicolas Markey, and Jean-François Raskin. Robust Safety of Timed Automata. In *Formal Methods in System Design*, 33(1–3):45–84, Kluwer Academic Publishers, 2008.

[3] Patricia Bouyer, Nicolas Markey, and Pierre-Alain Reynier. Robust Analysis of Timed Automata *via* Channel Machines. In *Proceedings of the 11th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS'08)*, pages 157–171, Lecture Notes in Computer Science, volume 4962, 2008.

[4] François Laroussinie, Nicolas Markey, and Ghassan Oreiby. On the Expressiveness and Complexity of ATL. *Logical Methods in Computer Science*, 4(2:7), 2008.

[5] Thomas Brihaye, Arnaud Da Costa, François Laroussinie, and Nicolas Markey. ATL with Strategy Contexts and Bounded Memory. In *Proceedings of the Symposium on Logical Foundations of Computer Science (LFCS'09)*, pages 92–106, Lecture Notes in Computer Science, 2009.

[6] R. Alur, C. Courcoubetis, and D.L̃. Dill. Model-checking for real-time systems. In *Proceedings of the Fifth Annual IEEE Symposium on Logic in Computer Science*, pages 414–425, 1990.

[7] R. Alur and D.L̃. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, 1994.

[8] A. Aziz, K. Sanwal, V. Singhal, and R. Brayton. Model checking continuous time markov chains. *ACM Trans. on Comp. Logic*, 1(1):162–170, 2000.

[9] Nathalie Bertrand, Patricia Bouyer, Thomas Brihaye, and Nicolas Markey. Quantitative model-checking of one-clock timed automata under probabilistic semantics. In *Proceedings of the 5th International Conference on Quantitative Evaluation of Systems (QEST'08)*, pages 55–64, Saint Malo, France, September 2008. IEEE Computer Society Press.

[10] C. Baier, B.R̃. Haverkort, H. Hermanns, and J.-P. Katoen. Model-checking algorithms for continuous-time markov chains. *IEEE Trans. on Softw. Eng.*, 29(6):524–541, 2003.

[11] I.-R. Chen and I.-L. Yen. Analysis of probabilistic error checking procedures on storage systems. *J. on Comput.*, 38(5):348–354, 1995.

[12] Han, T., Katoen, J.P.: Counterexamples in probabilistic model checking. In: TACAS. Number 4424 in LNCS (2007) 60–75

[13] T. Han, J.-P. Katoen, and A. Mereacre. Approximate parameter synthesis for probabilistic time-bounded reachability. In *The Real-Time Systems Symposium (RTSS'08)*, page to appear, 2008.

[14] M. Hennessy and R. Milner. Algebraic laws for nondeterminism and concurrency. *ACM*, 32(1):137–161, 1985.

[15] Marcin Jurdziński, François Laroussinie, and Jeremy Sproston. Model checking probabilistic timed automata with one or two clocks. *Logical Methods in Computer Science*, 4(3:12), September 2008.

[16] J.-P. Katoen, M. Khattri, and I.S̃. Zapreev. A Markov reward model checker. In *Quantitative Evaluation of Systems (QEST)*, pages 243–245. IEEE CS Press, 2005.

[17] M.Z̃. Kwiatkowska, G. Norman, and D.Ã. Parker. Probabilistic symbolic model checking using PRISM: a hybrid approach. *Journal on Software Tools for Technology Transfer*, 6(2):128–142, 2004.

[18] Kwiatkowska, M., Norman, G., Parker, D.: Game-based Abstraction for Markov Decision Processes. In: QEST. (2006) 157–166

[19] E. Laguerre. Sur la théorie des équations numériques. *J. Math. Pures Appl. (3e série)*, 9:99–146, 1883.

[20] K.G̃. Larsen and A. Skou. Bisimulation through probabilistic testing. *Inf. Comput.*, 94(1):1–28, 1991.

[21] R. Alur, T. A. Henzinger, and M. Y. Vardi. Parametric real-time reasoning. In *STOC*, pages 592–601, 1993.

[22] M. Avriel. *Nonlinear Programming: Analysis and Methods*. Dover Publishing, 2003.

[23] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. The MIT Press and McGraw-Hill Book Company, 2001.

[24] C. Daws. Symbolic and parametric model checking of discrete-time Markov chains. In *ICTAC*, LNCS 3407, pages 280–294, 2004.

[25] R. Fröberg. *An Introduction to Gröbner Bases*. John Wiley & Sons, 1998.

[26] R. W. Hamming. *Numerical Methods for Scientists and Engineers*. McGraw-Hill, 1973.

[27] E. Hansen. *Global Optimization Using Interval Analysis*. Dekker, New York, 1992.

[28] Hermanns, H., Wachter, B., Zhang, L.: Probabilistic CEGAR. In: *CAV*, LNCS 5123, pages 162–175, 2008.

[29] P. Henrici and W. R. Kenan. *Applied & Computational Complex Analysis: Power Series Integration Conformal Mapping Location of Zero*. John Wiley & Sons, 1988.

[30] T. Hune, J. Romijn, M. Stoelinga, and F. W. Vaandrager. Linear parametric model checking of timed automata. *J. Log. Algebr. Program.*, 52-53:183–220, 2002.

[31] A. Jensen. Markoff chains as an aid in the study of Markoff processes. *Skand. Aktuarietidskrift*, 36:87–91, 1953.

[32] R. Lanotte, A. Maggiolo-Schettini, and A. Troina. Parametric probabilistic transition systems for system design and analysis. *Formal Asp. Comput.*, 19(1):93–109, 2007.

[33] van Moorsel, A. P. A.: *Performability Evaluation Concepts and Techniques*, Ph.D. Thesis, Universiteit Twente, 1993.

[34] Munsky, B., Khammash, M.: The Finite State Projection Algorithm for the Solution of the Chemical Master Equation, *Journal of Chemical Physics*, **124**(044104), 2006.

[35] J. Nocedal and S. Wright. *Numerical Optimization*. Springer, New York, 1999.

[36] V. Y. Pan. Approximating complex polynomial zeros: Modified Weyl's quadtree construction and improved Newton's iteration. *J. Complexity*, 16(1):213–264, 2000.

[37] Kim Just a test. *This is a test*.

[38] A. Schrijver. *Theory of Linear and Integer Programming*. John Wiley & Sons, 1998.

[39] Segala, R., Lynch, N.A.: Probabilistic simulations for probabilistic processes. Nord. J. Comput. **2** (1995) 250–273

[40] G. A. Sitton, C. S. Burrus, J. W. Fox, and S. Treitel. Factoring very-high-degree polynomials. *Signal Processing Magazine, IEEE*, 20(6):27–42, 2003.

[41] H. Stetter. *Analysis of Discretization Methods for Ordinary Differential Equations*. Springer-Verlag, 1973.

[42] I. Stewart. *Galois Theory*. Chapman and Hall, 1989.

[43] A. Thümmler, P. Buchholz, and M. Telek. A novel approach for phase-type fitting with the EM algorithm. *IEEE Trans. Dependable Sec. Comput.*, 3(3):245–258, 2006.

[44] Wachter, B., Zhang, L., Hermanns, H.: Probabilistic Model Checking Modulo Theories. In: QEST. (2007) 129–138

[45] D. Zhang and R. Cleaveland. Fast on-the-fly parametric real-time model checking. In *RTSS*, pages 157–166, 2005.

[46] Zhang, L., Hermanns, H., Hahn, E. M., Wachter, B.: Time-Bounded Model Checking of Infinite-State Continuous-Time Markov Chains, *ACSD*, 2008, Xi'an, China.

[47] Laurent Doyen, Thomas A. Henzinger, and Jean-François Raskin Equivalence of Labeled Markov Chains In *Int. J. Found. Comput. Sci.*, 19(3):549–563, 2008.