Compositional Theories of Qualitative and Quantitative Behaviour

Ed Brinksma

Chair of Formal Methods and Tools
Department of Computer Science, University of Twente
PO Box 217, 7500AE Enschede, Netherlands
brinksma@cs.utwente.nl
http:/home.cs.utwente.nl/~brinksma

Extended Abstract

The integrated modelling and analysis of functional and non-functional aspects of system behaviour is one of the important challenges in the field of formal methods today. Our ever-increasing dependence upon of all sorts of critical applications of networked and/or embedded systems, often including sophisticated multi-media features, lends this intellectual challenge also great practical relevance. In this talk we will report on work in this area in the past decade or so on the use of techniques from so-called formal methods in the area of performance modelling and analysis, and in particular on the theory of stochastic process algebra (SPA) and its application.

Traditional performance models like Markov chains and queueing networks are widely accepted as simple but effective models in different areas, yet they lack the notion of hierarchical system (de)composition that has proved so useful for conquering the complexity of systems in the domain of funtional system properties. Compositional, hierarchical description and analysis of functional system behaviour is the domain *process algebra* [24,3,17]. It offers a mathematically well-elaborated framework for reasoning about the structure and behaviour of reactive and distributed systems in a compositional way, including abstraction mechanisms that allow for the treatment of system components as black boxes, encapsulating their internal structure. Process algebras are typically equipped with a formally defined structured operational semantics (SOS [26]) that maps process algebra terms onto *labelled transition systems* in a compositional manner. Such labelled transition systems consist of a set of states and a transition relation that describes how the system evolves from one state to another. These transitions are labelled with action names that represent the (inter)actions that may cause the transitions to occur. Such transition systems can be visualised by drawing states as nodes of a graph and transitions as directed edges (labelled with action names) between them.

The labelled transition model is very close to the usual representation of Markov chains as transition systems or automata. Also there system states are connected

W.M.P. van der Aalst and E. Best (Eds.): ICATPN 2003, LNCS 2679, pp. 37–42, 2003.

by directed transition arcs that are labelled. In the case of discrete time Markov chains the labels are probabilities, and in the case of continuous time Markocv chains the labels are the *rates* that correspond to the (nagative) exponential distributions that represent the stochastic delays associated with the state transitions. This structural correspondence between the two models motivated the beginning of research in the early 1990's on *stochastic process algebras* [4,16,15], which sought to integrate performance modelling with Markov chains with functional analysis, and to transfer the process algebraic notion of (de)composition and hierarchy to Markov chain theory.

This marriage of process algebra with performance modelling requires careful re-examinination of the interpretation of some classical process algebraic concepts, in particular that of choice, concurrent composition, and synchronization. In standard process algebra's choice operator '+' offers a qualitative selection between alternative behaviours. Its idempotency law $B + B = B$ has a natural interpretation as a poor man's choice: identical choices are as good as no choice at all. In most performance models, however, branching behaviour implies a race condition between the alternatives that have a quantitative effect. For example, if a is an action offered with rate μ then the choice between two such actions offers a with a rate of 2μ. This gives rise to additive laws such as $(a,\mu).B+(a,\lambda).B = (a,\mu+\lambda).B$ (see [15]), or $(\mu).B+(\lambda).B = (\mu+\lambda).B$ (see [22], where (μ) denotes an exponentially distributed delay with rate μ). The standard interleaving interpretation of concurrent composition in process algebra matches well with the memorylessness of exponential (or geometric) distributions in continuous (discrete) Markov chains, allowing for so-called expansion laws to remove explicit parallelism from system descriptions. This becomes more problematic if general distributions are allowed, such as, e.g., in semi-Markov chains. An elegant solution here is to treat stochastic delays in a way similar to clocks in timed automata, with separate operators to set them and test for their expiration [18]. Synchronization, finally, in the setting of stochastically delayed actions, poses the question of what is the (distribution of the) delay of the synchronizations between them. Here, different strategies have been proposed, ranging from (normalized) products of rates [15], synchronizations only between passive (no rates) and active action occurences (determining the synchronized rate) [4], to apparent rates [16]. Elegant is the solution to have rates associated only to pure delay actions that do not synchronize, but interleave, as in IMC. This induces a delay of synchronized actions with a distribution of the maximum of the delays preceeding the synchronizing actions, which is an intuitively appealing choice. A more complete overview of Markovian stochastic algebras can be found in [10]; an account of the non-Markovian case is given in [18].

The fruitfulness of the process algebraic approach to the specification and generation of Markov chains has been demonstrated by a number results. In the stochastic setting, *bisimulation* equivalence [25], a central notion of equivalence for comparing labelled transition systems, has been shown to coincide with *lumpability*, a key concept for the aggregation of Markov chains [16]. Moreover, as bisimulation can be shown to be preserved under system composition operators (algebraically: bisimulation is a *congruence*), Markov chain aggregation can be carried out compositionally, i.e. component-wise. Case studies have shown the practicality of this compositional approach, and important progress has been made in exploiting the syntactic structure of specifications for performance analysis purposes.

A second area where approaches from formal methods are being put to use successfully is the evaluation of Markov chain models. Once a continuous-time Markov chain (or CTMC) has been generated, the next step is to evaluate the measure(s) of interest such as time to failure, system throughput or utilisation, with the required accuracy. Whereas various techniques have been developed for the specification of CTMCs, such as stochastic process algebras and Petri nets, the specification of measures of interest has remained fairly cumbersome and is typically done in a rather informal, ad-hoc manner. In particular, usually only simple state-based performance measures – such as long-run and transient probabilities – can be defined and analysed with relative ease.

In contrast, in the area of formal methods powerful means have been developed to express temporal properties of systems, e.g., based on temporal logics. Logics such as CTL (Computation Tree Logic) [14] allow one to express state-based properties as well as properties over paths, i.e., state sequences through transition systems. One thus may express, for instance, that along all (or some) paths a certain set of goal states can eventually be reached while visiting only states of a particular kind before reaching one of these goal states. The validity of CTL-formulas over finite automata can be established by automated techniques such as *model checking* [13]. These techniques are based on a systematic, usually exhaustive, state-space exploration to check whether a property is satisfied in each state, thereby using effective methods to combat the state-space explosion problem. Model checking is supported by software tools such as SMV [12] and SPIN [21] and has been successfully used in various industrial case studies.

Model-checking of CTL formulas is usually done by a recursive descent over the construction of the logical property to be checked, exploiting compositionality on the level of the logical operators. CTL has recently been extended with ample means to specify state- as well as path-based performance and dependability measures for CTMCs in a compact and unambiguous way. Besides the standard steady-state and transient measures, the logic CSL (Continuous Stochastic Logic) [2,8] allows for the specification of (constraints over) probabilistic measures over paths through CTMCs. For instance, it can be expressed what the probability is, that starting from a particular state, within t time units a set of goal-states is reached, thereby avoiding or deliberately visiting particular intermediate states before. This is a useful feature for dependability analysis that goes beyond the standard measures. Other types of non-standard, but practically interesting measures that can conveniently be expressed are, for example, response times that are conditioned on the equilibrium state of a CTMC, properties that are typically analysed using dedicated and rather involved techniques. An indication of the adequacy of CSL is that logical equivalence coincides with the lumpability equivalence over Markov Chains mentioned above. Lumping-equivalent CTMCs thus satisfy the same formulas, and there is no formula that can distinguish between lumping-equivalent CTMCs.

Given a finite CTMC and a performance measure specified in CSL, an automated procedure can be applied – á la model checking – to establish the validity of the (constraint over the) measure [7]. To that purpose, the traditional model-checking algorithms are extended with numerical methods such as matrix-vector multiplication, techniques for solving systems of linear equations, and uniformization (or solvers for Volterra integral equation systems). For path-formulas, measure-driven transformations are employed: for a given CTMC M and state s in M, the probability for s to satisfy path-formula φ is calculated by means of a transient analysis of another (smaller) CTMC M', which can easily be derived from M using φ. The time and space complexity of the model-checking algorithms is polynomial in the size of the model and linear in the length of the formula. Tool-implementations such as $\mathsf{E} \vdash \mathsf{MC}^2$ [20], Prism [23] and the APNNToolBox [9] are available.

Outlook. The work on stochastic process algebras has, apart from the technical achievements, brought the formal methods and performance analysis communities closer together, with now at least a qualified group of people being active in both communities. Markov chains, and CTMCs in particular, have received scant attention in concurrency theory for a long time: whenever probabilities have been considered, they were mostly of a purely discrete nature. Verification of discrete-time Markov chains, for instance, dates back to the early nineties [19]. The scientific interest in verifying CTMCs is steadily increasing. Its main advantages are that it offers a flexible and precise means to succinctly specify standard and complex performance and dependability measures, complemented by an automated technique to compute these measures in a uniform way. Specialized algorithms are thus hidden from the performance engineer. Main current research topics are: extensions to Markov reward models, and development of techniques to deal with very large state spaces, e.g., symbolic techniques, Kronecker algebra, and abstraction techniques.

Stochastic process algebraic specifications mostly do not yield Markov chain models directly, but mixed transition systems that involve both system actions and stochastic distributions. Abstracting the actions away is generally insufficient to obtain the embedded Markov models, as the resulting transition system, known as a Markov decision process, may still contain non-stochastic elements in the form of nondeterministic transitions. To obtain a Markov chain such nondeterminism must first be resolved using, for example, reduction or scheduling techniques using adversaries. Another option is to study the properties of the Markov decison processes directly. Some verification theory for discrete Markov decision processes is available [6,5], as well as for the stationary behaviour of discrete semi-Markov decision processes [1]. A theory for the continuous-time case is yet to be formulated, however. Such a theory and effective model-checking algorithms for continuous-time Markov decision processes remain formidable open problems to be challenged.

A third direction of work is the further elaboration of the stochastic process algebraic theory for the non-Markovian case, which is closely linked to the performance model of generalized semi-Markov processes (GSMP [27]. They are usually analysed by discrete-event simulation techniques. But so far little has been done to categorize interesting subcases for which more powerful analytic techniques exist, and which could be exploited in the form of interesting equational laws on the process algebraic level.

Acknowledgements. For more than a decade the author has had the privilege to discuss and work on stochastic process algebraic topics with a number of great colleagues from whom he has learned a lot. In particular, he wished to acknowledge his interactions with the enthousiastic members of the Twente team Pedro D'Argenio, Holger Hermanns, Joost-Pieter Katoen and Rom Langerak.

References

1. L. de Alfaro. How to specify and verify the long-run average behavior of probabilistic systems. In *IEEE 13th Symp. on Logic in Comp. Sc.*, pp. 174–183, IEEE CS Press, 1998.
2. A. Aziz, K. Sanwal, V. Singhal and R. Brayton. Model checking continuous time Markov chains. *ACM Transactions on Computational Logic*, **1**(1): 162–170, 2000.
3. J.A. Bergstra, A. Ponse, and S.A. Smolka, editors. *Handbook of Process Algebra.* Elsevier Science Publishers, 2001.
4. M. Bernardo and R. Gorrieri. Extended Markovian Process Algebra. In Ugo Montanari and Vladimiro Sassone, editors, *CONCUR '96: Concurrency Theory (7th International Conference, Pisa, Italy, August 1996)*, volume 1119 of *Lecture Notes in Computer Science*. Springer, 1996.
5. A. Bianco and L. de Alfaro. Model checking of probabilistic and nondeterministic systems. In *Found. of Softw. Technology and Th. Comp. Sc.*, LNCS 1026: 499–513, Springer, 1995.
6. C. Baier and M.Z. Kwiatkowska. Model checking for a probabilistic branching time logic with fairness. *Distr. Comp.*, **11**: 125–155, 1998.
7. C. Baier, B.R. Haverkort, H. Hermanns and J.-P. Katoen. Model checking continuous-time Markov chains by transient analysis. In: E.A. Emerson and A.P. Sistla, *Computer-Aided Verification*,LNCS 1855: 358–372, 2000.
8. C. Baier, J.-P. Katoen and H. Hermanns. Approximate symbolic model checking of continuous-time Markov chains. In J.C.M. Baeten and S. Mauw, *Concurrency Theory*, LNCS 1664: 146–162, 1999.
9. P. Buchholz, J.-P. Katoen, P. Kemper and C. Tepper. Model-checking large structured Markov chains. *Journal of Logic and Algebraic Programming*, 2003 (to appear).
10. E. Brinksma and H. Hermanns. Process Algebra and Markov Chains. In [11], LNCS 2090: 183–231, 2001.
11. E. Brinksma and H. Hermanns and J.-P. Katoen. Lectures on Formal Methods and Performance Analysis. LNCS 2090, 2001.
12. A. Cimatti, E. Clarke, F. Giunchiglia and M. Roveri. NuSMV: a new symbolic model checker. *J. on Software Tools for Technology Transfer*, **2**: 410–425, 2000.

13. E. Clarke, O. Grumberg and D. Peled. *Model Checking.* MIT Press, 1999.
14. E.M. Clarke and E.A. Emerson. Design and synthesis of synchronisation skeletons using branching time temporal logic. In *Logic of Programs*, LNCS 131: 52–71, 1981.
15. N. Götz, U. Herzog, and M. Rettelbach. Multiprocessor and Distributed System Design: The Integration of Functional Specification and Performance Analysis using Stochastic Process Algebras. In *Performance'93*, 1993.
16. J. Hillston. *A Compositional Approach to Performance Modelling.* PhD thesis, University of Edinburgh, 1994.
17. C.A.R. Hoare. *Communicating Sequential Processes.* Prentice-Hall, Englewood Cliffs, NJ, 1985.
18. J.-P. Katoen and P.R. D'Argenio. General Distributions in Process Algebra. In [11], LNCS 2090: 375–430, 2001.
19. H. Hansson and B. Jonsson. A logic for reasoning about time and reliability. *Formal Aspects of Computing* **6**: 512–535, 1994.
20. H. Hermanns, J.-P. Katoen, J. Meyer-Kayser and M. Siegle. A Markov chain model checker. In S. Graf and M.I. Schwartzbach, *Tools and Algorithms for the Construction and Analysis of Systems*, LNCS 1785: 347–362, 2000.
21. G.J. Holzmann. The model checker SPIN. *IEEE Tr. on Softw. Eng.*, **23**(5): 279–295, 1997.
22. H. Hermanns. Interactive Markov Chains. LNCS 2428, 2002.
23. M.Z. Kwiatkowska, G. Norman, and D. Parker. Probabilistic symbolic model checking with prism: A hybrid approach. In J-P. Katoen and P. Stevens (eds), *Tools and Algorithms for the Construction and Analysis of Algorithms*, LNCS 2280: 52–66, 2002
24. R. Milner. Calculi for Synchrony and Asynchrony. *Theoretical Computer Science*, 25:269–310, 1983.
25. R. Milner. *Communication and Concurrency.* Prentice Hall, London, 1989.
26. G.D. Plotkin. A Structured Approach to Operational Semantics. Technical Report DAIMI FM-19, Computer Science Department, Aarhus University, 1981.
27. G.S. Shedler. *Regenerative Stochastic Simulation.* Academic Press, 1993.