CASPA: A performance evaluation tool based on stochastic process algebra and symbolic data structures

Matthias Kuntz, Markus Siegle, Edith Werner
Friedrich-Alexander-Universität Erlangen-Nürnberg, Institut für Informatik
{mskuntz,siegle,siedwern}@informatik.uni-erlangen.de

***Abstract*— This note describes the tool CASPA, a new performance evaluation tool which is based on a Markovian stochastic process algebra. CASPA uses multi-terminal binary decision diagrams (MTBDD) to represent the transition systems underlying a given process algebraic specification. All phases of modelling, from model construction via numerical analysis to the computation of performance measures, are based entirely on this symbolic data structure.**

I. INTRODUCTION

Symbolic data structures, such as binary decision diagrams (BDD) [1] and variants thereof have proved to be suitable for the efficient generation and compact representation of very large state spaces and transition systems. It has been shown that in the context of a compositional model specification formalism such as process algebra, the size of the symbolic representation can be kept within linear bounds, even if the underlying state space grows exponentially [2]. The key to such compact representation is the exploitation of the compositional structure of a given specification [3], [4], [5]. It is also known that in addition to functional analysis, performance analysis and the verification of performability properties can also be carried out on such symbolic representations [3], [6], [7], [5], [8].

In this note, we describe the new tool CASPA [9] which offers a Markovian stochastic process algebra language for model specification. CASPA generates a symbolic model representation, which is based on multi-terminal binary decision diagrams (MTBDD) [10], [11], directly from the high-level model, without generating transition systems as an intermediate representation. In addition to specifying the model, the CASPA modelling language allows the user to specify performance measures of interest. Numerical analysis and computation of measures are also carried out directly on the symbolic representation of the transition rate matrix of the underlying labelled CTMC.

To our knowledge, CASPA is the first stochastic process algebra tool whose implementation relies completely on symbolic data structures.

II. THE MODELLING LANGUAGE

The modelling language of CASPA is a restricted version of the language supported by TIPPtool [12]. It is a stochastic process algebra where all actions have an exponentially distributed delay. The language provides operators for prefixing, choice, parallel composition and hiding. Infinite (i.e. cyclic) behaviour can be specified with the help of defining equations (instead of employing an explicit recursion operator). The technique used for symbolic model representation (cf. Sec. IV) works only for finite state spaces. Therefore the grammar of the input language is such that recursion over static operators (i.e. parallel composition and hiding) is not allowed, which ensures that the underlying state space is finite.

CASPA allows the specification of parameterised processes, i.e. processes which carry one ore more integer parameters. This feature is very useful for describing the behaviour of queueing, counting, or generally indexed processes. Within a parameterised process, the enabling of actions may be conditioned on the current value of the process parameters.

We demonstrate the use of the CASPA modelling language by means of a small example which is shown in Fig. 1. It is a queueing system, consisting of two arrival processes (inter-arrival times have Erlang-3 and Erlang-2 distribution) and a service center with finite capacity buffer and exponential service times. In the first four lines the rate parameter values and the buffer size are defined.

III. SPECIFICATION OF PERFORMANCE METRICS

CASPA supports the definition and computation of three different types of performance measures:

A *state measure* is defined as the probability of the system being in a specific state or in a well-defined subset of the state space. Such sets of states are defined by referencing to process names, thereby possibly using Boolean operators and conditioning on process parameter ranges.

A *throughput measure* is defined as the mean number of occurences of a specific named action per unit of time.

A *mean value measure* is defined as the expected value of a certain process parameter, taken over all reachable states.

For the queueing system example from Sec. II, the measures describing the probability of the buffer being full, the expected throughput of action `serve`, and the mean number of jobs in the server are defined in the last three lines of Fig. 1.

This work is supported by the German DFG-funded project BDDANA

```
/*** rate and constant definitions ***/
rate lambda1 = 0.5;
rate lambda2 = 0.1;
rate mu = 4.3;
int max = 15;

/*** system specification ***/
System   := ( Arrival1 |[]| Arrival2 ) |[enq]| Server(0)
Arrival1 := (tau,lambda1); (tau,lambda1); (enq,lambda1); Arrival1
Arrival2 := (tau,lambda2); (enq,lambda2); Arrival2
Server(n [max]) := [n < max] -> (enq,1); Server(n+1)
                   [n > 0]   -> (serve,mu); Server(n-1)

/*** measure definition ***/
statemeasure Buffer_full Server(n = max)
throughputmeasure Service_rate serve
meanvalue Occupancy Server(n)
```

Fig. 1. Specification of a small queuing system example

IV. CONSTRUCTION OF THE SYMBOLIC STATE SPACE REPRESENTATION

CASPA translates a given process algebraic specification directly to an MTBDD-based symbolic representation of the underlying state space and transition system. It uses the CUDD library [13] which provides support for the construction and manipulation of BDD-based data structures. The translation implements the denotational semantics described in [14], with some extensions and optimisations. The basic idea of this translation is as follows: In a first step, the parse tree of the process algebraic specification at hand is constructed. Then the MTBDD representation of the underlying transition relation is constructed in a compositional fashion, starting with sequential processes (i.e. processes which do not contain the parallel composition operator) which are located close to the leaves of the parse tree. Finally the MTBDD for the overall process is built from the MTBDDs of its components by applying the rules for symbolic parallel composition (see, e.g, [5], [8]). This construction procedure is completely symbolic and compositional, i.e. each sub-process of the specification is represented by an MTBDD, which is then used as an operand during the construction of the higher-level processes.

Already during its construction, the parse tree is annotated with information concerning the performance measures to be derived. For example, nodes associated with a process name that occurs in a state measure specification are marked, and subsequently a BDD is constructed which encodes exactly the states which are relevant for that state measure. Nodes associated with a parameterised process for which a mean value has to be computed are also marked, and subsequently an MTBDD is constructed which encodes the relevant states and associated parameter value.

V. NUMERICAL ANALYSIS AND COMPUTATION OF PERFORMANCE MEASURES

In its current version, CASPA supports the computation of the steady-state probability vector of the underlying CTMC, from which the performance measures of interest are then calculated. The rate matrix of the CTMC is obtained from the symbolic representation of the transition system by abstracting from the action labels. This abstraction can easily be performed with the help of MTBDD operations, and the result is a symbolic representation of the rate matrix.

For the numerical computation of the steady-state probabilities, CASPA uses the iterative solvers code that was developed within the PRISM project [7], which is available from the PRISM web page [15]. To our knowledge, these algorithms are currently the fastest available implementation of MTBDD-based linear systems solvers. They rely on the advanced concept of hybrid offset-labelled MTBDDs as described in [16].

Once the vector of steady-state probabilities has been obtained, the performance measures of interest are computed. Basically, each type of measures can be computed as the sum or weighted sum of certain state probabilities. These expressions can be computed efficiently on the basis of MTBDD operations, using the MTBDDs which we mentioned at the end of Sec. IV.

VI. CONCLUSION AND FUTURE WORK

The first version of CASPA has just been completed and is currently being tested systematically on the basis of several standard case studies from the literature. While it is still too early for a comprehensive assessment of the tool, we can already state the following: State space construction as realised in CASPA is very efficient, it is several orders of magnitude faster than that of TIPPtool which uses an explicit (and therefore much less space efficient) representation of the state space and transition system. As an example, for a certain instance of the Kanban system from [17] CASPA needs 0.29 seconds to generate the potential state space and another 2.17 seconds to construct the transition relation among the 2,546,432 reachable states (these times were obtained on a SUN UltraSPARC workstation running at 500 MHz). The numerical solution with the algorithms from PRISM achieves an efficiency which is almost comparable to state-of-the-art sparse solvers. Once the state probabilities have been computed, the derivation of performance measures is very fast, in fact it is much faster than for TIPPtool.

For the future we plan to extend CASPA by a transient analysis module, in order to enable the computation of transient performance measures. We also plan to develop a user-friendly graphical user interface to replace the current command-line interface, and to support modelling experiments with automatically varying model parameters and graphical display of results. As a long-term goal, we are hoping to integrate CASPA with our model checker ETMCC [18], a software tool which supports the verification of performability properties of stochastic systems.

REFERENCES

[1] R. Bryant, "Graph-based Algorithms for Boolean Function Manipulation," *IEEE Transactions on Computers*, vol. C-35, no. 8, pp. 677–691, August 1986.

[2] H. Hermanns, M. Kwiatkowska, G. Norman, D. Parker, and M. Siegle, "On the use of MTBDDs for performability analysis and verification of stochastic systems," *Journal of Logic and Algebraic Programming*, vol. 56, no. 1-2, pp. 23–67, 2003.

[3] H. Hermanns, J. Meyer-Kayser, and M. Siegle, "Multi Terminal Binary Decision Diagrams to Represent and Analyse Continuous Time Markov Chains," in *3rd Int. Workshop on the Numerical Solution of Markov Chains*, B. Plateau, W. Stewart, and M. Silva, Eds. Prensas Universitarias de Zaragoza, 1999, pp. 188–207.

[4] L. de Alfaro, M. Kwiatkowska, G. Norman, D. Parker, and R. Segala, "Symbolic Model Checking for Probabilistic Processes using MTBDDs and the Kronecker Representation," in *TACAS'2000*, S. Graf and M. Schwartzbach, Eds. Springer LNCS 1785, 2000, pp. 395–410.

[5] M. Siegle, "Advances in model representation," in *Process Algebra and Probabilistic Methods, Joint Int. Workshop PAPM-PROBMIV 2001*, L. de Alfaro and S. Gilmore, Eds. Springer, LNCS 2165, September 2001, pp. 1–22.

[6] J.-P. Katoen, M. Kwiatkowska, G. Norman, and D. Parker, "Faster and Symbolic CTMC Model Checking," in *PAPM-PROBMIV'01*. Springer, LNCS 2165, 2001, pp. 23–38.

[7] M. Kwiatkowska, G. Norman, and D. Parker, "Probabilistic Symbolic Model Checking with PRISM: A Hybrid Approach," in *TACAS'2002*, J.-P. Katoen and P. Stevens, Eds. Springer LNCS 2280, April 2002, pp. 52–66.

[8] M. Siegle, *Behaviour analysis of communication systems: Compositional modelling, compact representation and analysis of performability properties*. Aachen: Shaker Verlag, 2002.

[9] E. Werner, "Leistungsbewertung mit Multi-Terminalen Binären Entscheidungsdiagrammen," Master's thesis, Universität Erlangen–Nürnberg, Informatik 7, 2003, (in German).

[10] M. Fujita, P. McGeer, and J.-Y. Yang, "Multi-terminal Binary Decision Diagrams: An efficient data structure for matrix representation," *Formal Methods in System Design*, vol. 10, no. 2/3, pp. 149–169, April/May 1997.

[11] R. Bahar, E. Frohm, C. Gaona, G. Hachtel, E. Macii, A. Pardo, and F. Somenzi, "Algebraic Decision Diagrams and their Applications," *Formal Methods in System Design*, vol. 10, no. 2/3, pp. 171–206, April/May 1997.

[12] H. Hermanns, U. Herzog, U. Klehmet, V. Mertsiotakis, and M. Siegle, "Compositional performance modelling with the TIPPtool," *Performance Evaluation*, vol. 39, no. 1-4, pp. 5–35, January 2000.

[13] F. Somenzi, "CUDD: Colorado University Decision Diagram Package, Release 2.3.0," September 1998, user's Manual and Programmer's Manual. [Online]. Available: http://vlsi.colorado.edu/∼fabio.

[14] M. Kuntz and M. Siegle, "Deriving symbolic representations from stochastic process algebras," in *Process Algebra and Probabilistic Methods, Proc. PAPM-PROBMIV'02*, H. Hermanns and R. Segala, Eds. Springer, LNCS 2399, 2002, pp. 188–206.

[15] "PRISM web page," http://www.cs.bham.ac.uk/∼dxp/prism/.

[16] D. Parker, "Implementation of symbolic model checking for probabilistic systems," Ph.D. dissertation, School of Computer Science, Faculty of Science, University of Birmingham, 2002.

[17] G. Ciardo and M. Tilgner, "On the use of Kronecker operators for the solution of generalized stochastic Petri nets," ICASE, Tech. Rep. 96-35, 1996.

[18] H. Hermanns, J.-P. Katoen, J. Meyer-Kayser, and M. Siegle, "A tool for model checking Markov chains," *Software Tools for Technology Transfer (STTT)*, vol. 4, no. 2, pp. 153–172, 2003.