[1] G. Agha, C. Gunter, M. Greenwald, S. Khanna, J. Meseguer, K. Sen, and P. Thati. Formal modeling and analysis of DoS using probabilistic rewrite theories. In *Workshop on Foundations of Computer Security (FCS)*, June 2005. [bib | .pdf]

Existing models for analyzing the integrity and confidentiality of protocols need to be extended to enable the analysis of availability. Prior work on such extensions shows promising applications to the development of new DoS countermeasures. Ideally it should be possible to apply these countermeasures systematically in a way that preserves desirable properties already established. This paper investigates a step toward achieving this ideal by describing a way to expand term rewriting theories to include probabilitic aspects that can show the effectiveness of DoS countermeasures. In particular, we consider the shared channel model, in which adversaries and valid participants share communication bandwidth according to a probabilistic interleaving model, and a countermeasure known as selective verification applied to the handshake steps of the TCP reliable transport protocol. These concepts are formulated in a probabilistic extension of the Maude term rewriting sytem and automated techniques are used to demonstrate the effectiveness of the countermeasures.

[2] R. Alur, K. Etessami, S. L. Torre, and D. Peled. Parametric temporal logic for “model measuring”. *ACM Trans. Comput. Logic*, 2(3):388-407, 2001. [bib | DOI]

We extend the standard model checking paradigm of linear temporal logic, LTL, to a “model measuring” paradigm where one can obtain more quantitative information beyond a “Yes/No” answer. For this purpose, we define a parametric temporal logic, PLTL, which allows statements such as “a request p is followed in at most x steps by a response q,” where x is a free variable. We show how one can, given a formula ***(x1...,xk) of PLTL and a system model K satisfies the property ***, but if so find valuations which satisfy various optimality criteria. In particular, we present algorithms for finding valuations which minimize (or maximize) the maximum (or minimum) of all parameters. Theses algorithms exhibit the same PSPACE complexity as LTL model checking. We show that our choice of syntax for PLTL lies at the threshold of decidability for parametric temporal logics, in that several natural extensions have undecidable “model measuring” problems.

[3] R. I. Brafman, C. Domshlak, and S. E. Shimony. Qualitative decision making in adaptive presentation of structured information. *ACM Trans. Inf. Syst.*, 22(4):503-539, 2004. [bib | DOI]

[4] J. Bryans, H. Bowman, and J. Derrick. Model checking stochastic automata. *ACM Trans. Comput. Logic*, 4(4):452-492, 2003. [bib | DOI]

Modern distributed systems include a class of applications in which non-functional requirements are important. In particular, these applications include multimedia facilities where real time constraints are crucial to their correct functioning. In order to specify such systems it is necessary to describe that events occur at times given by probability distributions; stochastic automata have emerged as a useful technique by which such systems can be specified and verified.However, stochastic descriptions are very general, in particular they allow the use of general probability distribution functions, and therefore their verification can be complex. In the last few years, model checking has emerged as a useful verification tool for large systems. In this article we describe two model checking algorithms for stochastic automata. These algorithms consider how properties written in a simple probabilistic real-time logic can be checked against a given stochastic automaton.

[5] J. Meseguer. Conditional rewriting logic as a unified model of concurrency. *Theoretical Computer Science*, 96(1):73-155, April 1992. [bib | DOI]

Rewriting with conditional rewrite rules modulo a set *E* of structural axioms provides a general framework for unifying a wide variety of models of concurrency. Concurrent rewriting coincides with logical deduction in *conditional rewriting logic*, a logic of actions whose models are concurrent systems. This logic is sound and complete and has initial models. In addition to general models interpreted as concurrent systems which provide a more operational style of semantics, more restricted semantics with an incresingly denotational flavor such as preorder, poset, cpo, and standard algebraic models appear as special cases of the model theory. This permits dealing with operational and denotational issues within the same model theory and logic. A programming language called Maude whose modules are rewriting logic theories is defined and given denotational and

operational semantics. Maude provides a simple unification of concurrent programming with functional and object-oriented programming and supports high level declarative programming of concurrent systems.

Keywords: concurrency conditional rewriting

[6] NASA Software Assurance Technology Center web site http://satc.gsfc.nasa.gov/assure/agbsec5.txt. [bib | .txt]

[7] D. Parker. *Implementation of Symbolic Model Checking for Probabilistic Systems*. PhD thesis, University of Birmingham, August 2002. [bib | .pdf]

In this thesis, we present efficient implementation techniques for probabilistic model checking, a method which can be used to analyse probabilistic systems such as randomised distributed algorithms, fault-tolerant processes and communication networks. A probabilistic model checker inputs a probabilistic model and a specification, such as ”the message will be delivered with probability 1”, ”the probability of shutdown occurring is at most 0.02” or ”the probability of a leader being elected within 5 rounds is at least 0.98”, and can automatically verify if the specification is true in the model.

Motivated by the success of symbolic approaches to non-probabilistic model checking, which are based on a data structure called binary decision diagrams (BDDs), we present an extension to the probabilistic case, using multi-terminal binary decision diagrams (MTBDDs). We demonstrate that MTBDDs can be used to perform probabilistic analysis of large, structured models with more than 7.5 billion states, way out of the reach of conventional, explicit techniques, based on sparse matrices. We also propose a novel, hybrid approach, combining features of both symbolic and explicit implementations and show, using results from a wide range of case studies, that this technique can almost match the speed of sparse matrix based implementations, but uses significantly less memory. This increases, by approximately an order of magnitude, the size of model which can be handled on a typical workstation.

[8] K. Sen, M. Viswanathan, and G. Agha. Statistical model checking of black-box probabilistic systems. In *16th conference on Computer Aided Verification (CAV)*, volume 3114 of *Lecture Notes in Computer Science*, pages 202-215. Springer, July 2004. [bib | .html | .pdf]

We propose a new statistical approach to analyzing stochastic systems against specifications given in a sublogic of continuous stochastic logic (CSL). Unlike past numerical and statistical analysis methods, we assume that the system under investigation is an unknown, deployed black-box that can be passively observed to obtain sample traces, but cannot be controlled. Given a set of executions (obtained by Monte Carlo simulation) and a property, our algorithm checks, based on statistical hypothesis testing, whether the sample provides evidence to conclude the satisfaction or violation of a property, and computes a quantitative measure (p-value of the tests) of confidence in its answer; if the sample does not provide statistical evidence to conclude the satisfaction or violation of the property, the algorithm may respond with a ”don't know” answer. We implemented our algorithm in a Java-based prototype tool called VeStA, and experimented with the tool using case studies analyzed in [15]. Our empirical results show that our approach may, at least in some cases, be faster than previous analysis methods.

[9] SPIN web site http://netlib.bell-labs.com/netlib/spin/whatispin.html. [bib | .html]

[10] Telelogic web site http://www.telelogic.com/. [bib | http]

[11] Verification & Validation web site http://ase.arc.nasa.gov/docs/vandv.html. [bib | .html]

This file was generated by bibtex2html 1.96.