Marta Kwiatkowska, Ulrich Herzog
Christoph Meinel, Moshe Vardi (editors)

# Probabilistic Methods in Verification

Dagstuhl Seminar Report 2000181
April 30 - May 5, 2000

# Contents

# 1 Preface

The established methodology of engineering computer systems, both hardware and software, involves first building a *model* of the system, and then its detailed *analysis*, before implementation takes place. The motivation for this is to increase the engineers' confidence in the design of the system, with respect to desirable characteristics such as functional correctness and performance requirements. Two related disciplines which are instances of this methodology are *verification* and *performance evaluation*:

- The first of these, **verification** through *model checking*, employs algorithmic methods to provide "Yes/No" answers to qualitative correctness requirements, primarily concerned with *system behaviour over time* (for example, ensuring the delivery of a message or safety of a particular activity). A model of the system is formulated using an appropriate formalism, and then supplied as input to a software tool which automatically checks if a given specification is satisfied. Such model checking tools are used widely in practical applications, particularly for analysing hardware and communication protocols. The term **probabilistic verification** refers to methods in which "Yes/No" answers are replaced with estimates of the *likelihood* of the system satisfying a specification. Two prevailing views of probabilistic verification exist. The first concerns probabilistic models of the system (for example, discrete-time Markov chains or Markov decision processes), and aims to model check these against probabilistic variants of temporal logics. The second is applied in the context of non-probabilistic systems, but those of a size which makes exhaustive model checking impractical or infeasible. The aim is then to establish that the required properties hold with high probability.

- The field of **performance evaluation** involves building a probabilistic model of a system, followed by analysis focused on the calculation of *performance measures*. Typically, the underlying model of system description formalisms in this field is a continuous-time Markov chain, with the desired system requirements (throughput, mean time to failure, etc.) expressed in terms of steady-state probabilities. This relies heavily on the use of numerical methods and tools when analysing the models. Performance evaluation tools have been successfully used to predict the impact of changes to load and arrival characteristics of computer networks.

Though the fields of verification and performance evaluation have historically concentrated on analysing different aspects of the system (*qualitative* correctness requirements versus *quantitative* performance issues), they are complementary and have much in common. For example, both aim to build a representation of the model in the computer memory, and in fact the difficulties and challenges posed by representing very large models have been recognised in both communities (the verification community calls this the 'state explosion problem', whereas to performance evaluation practitioners it is known as 'largeness'). Therefore, the appeal of integration and cross-fertilisation of techniques between

the two fields is immediate. The goal of this Dagstuhl meeting was to bring together researchers representing the different communities, who would not necessarily meet at other conferences or workshops, in order to provoke debate and to facilitate exchange of expertise. In all, 48 researchers from 11 countries participated in the meeting.

In an effort to bring the two distinct, yet closely related, fields together four keynote speakers were invited to give overview lectures. **Luca de Alfaro** gave an introductory talk on the algorithmic verification of probabilistic systems. **Rajeev Alur** then spoke about modular specification and simulation of hybrid systems. An introduction to the field of performance evaluation was presented by **Boudewijn Haverkort**. This was complemented by a talk from **Martin Reiser** giving an overview of performance evaluation of computer and communication systems over the past thirty years.

The remaining 31 presentations given by participants of the meeting covered a range of topics from probabilistic verification and performance evaluation. Some centred on the development of **modelling formalisms** for probabilistic systems, including stochastic variants of process algebras and the $\pi$-calculus, real-time extensions to probabilistic and stochastic systems, and continuous space Markov processes.

Other talks concerned **methods of analysis** for such systems: model checking algorithms for probabilistic and stochastic temporal logics; and equivalences on probabilistic systems and their corresponding decision procedures.

A third group of talks centred on the **implementation** of probabilistic verification, describing recent or ongoing work on the development of efficient tools and techniques. These included symbolic, BDD-based, model checking of probabilistic algorithms and stochastic Petri net tools employing Kronecker-based techniques. While the former focuses on verification and the latter on performance evaluation, what they have in common is the use of BDDs and Kronecker, which should lead to fruitful exchange of ideas. A group of talks introduced BDD-based methods for representing and verifying logical circuits with high probability.

A number of interesting **application areas** were also highlighted, including security and fault-tolerant systems.

The selection of presentations was accompanied by a **panel discussion** chaired by **Moshe Vardi** held towards the end of the meeting. Six prominent researchers

**Boudewijn Haverkort**
**Ulrich Herzog**
**Radha Jagadeesan**
**Joost-Pieter Katoen**
**Marta Kwiatkowska**
**Frits Vandraager**

were invited to answer questions on the present and future relationship between the fields of probabilistic verification and performance evaluation, prompting lively, interesting and productive discussion summarised at `http://www.cs.bham.ac.uk/~mzk/Dagstuhl/`. The

optimism for future cooperation was evident not only here, but in the numerous stimulating discussions between participants during the week.

Of course, the success of the event was only made possible by the excellent facilities and working environment of the venue. On behalf of everyone who attended the seminar, the organisers would like to thank the staff at Schloß Dagstuhl for all their hard work.

Marta Kwiatkowska
Ulrich Herzog
Christoph Meinel
Moshe Vardi

# 2 Workshop Programme

**Monday, May 1, 2000**

9.00 Welcome and Introductions

*Chair: Moshe Vardi*
9.30 Luca de Alfaro: Algorithmic Verification of Probabilistic Systems

10:30 Coffee

*Chair: Michael Huth*
11:00 Prakash Panangaden: Approximation of Markov Processes
11:30 Radha Jagadeesan: Approximate Reasoning on Partial Labeled Markov Processes

12:00 Lunch

*Chair: Holger Hermanns*
14:00 David Parker: Verifying Randomized Distributed Algorithms with Prism
14:30 Peter Kemper: Markov Chain Analysis with Kronecker Representations
15:00 Gianfranco Ciardo: Using Decision Diagrams for the Solution of Large Markov Chains

15:30 Coffee

*Chair: Marta Kwiatkowska*
16:00 Andrei Sabelfeld: Probabilistic Models of Secure Information Flow
16:30 Annabelle McIver: Separation for Probabilistic Programs
17:00 Michael Huth: Model Checking Loosely Specified Probabilistic Systems

**Tuesday, May 2, 2000**

*Chair: Frits Vaandrager*
9:00 Rajeev Alur: CHARON: Modular Specification and Simulation of Hybrid Systems

10:00 Coffee

*Chair: Rajeev Alur*
10:30 Jeremy Sproston: Model Checking Probabilistic Timed Automata
11:00 Daniele Beauquier: Probabilistic Timed Automata
11:30 Pedro D'Argenio: Simulation and Verification of Stochastic Timed Systems

12:00 Lunch

*Chair: Prakash Panangaden*
14:00 David Monniaux: Abstract Interpretation of Probabilistic Semantics
14:30 Oltea Mihaela Herescu: Probabilistic Asynchronous $\pi$-Calculus
15:00 Marco Bernardo: A Theory of Testing for Markovian Processes

15:30 Coffee

*Chair: Marek Karpinski*
16:00 Elena Dubrova: Probabilistic Verification of Multiple-Valued Functions
16:30 Harald Sack: Representation of Discrete Functions with Mod-p-Decision Diagrams
17:00 Jawahar Jain: Probabilistic Verification of Boolean Functions and Partitioned-OBDDs
17:30 Martin Sauerhoff: Randomized BDDs

**Wednesday, May 3, 2000**
*Chair: Ulrich Herzog*
9:00 Boudewijn Haverkort: An Introduction to Performance and Dependability Evaluation

10:00 Coffee

*Chair: Jane Hillston*
10:30 Joost-Pieter Katoen: Analysing Markov Chains by Model Checking
11:00 Joachim Meyer-Kayser: Model Checking Action-Labelled CTMCs
11:30 Gerardo Rubino: Computing Rare Event Probabilities on Large Markov Models

12:00 Lunch

14:00 Excursion

**Thursday, May 4, 2000**

*Chair: Boudewijn Haverkort*
9:00 Martin Reiser: Performance Evaluation of Computer and Communications: An Overview

10:00 Coffee

*Chair: Joost-Pieter Katoen*
10:30 Holger Hermanns: Markov Chain Algebra
11:00 Mario Bravetti: Stochastic Process Algebras with General Distributions
11:30 Purush Iyer: Probabilistic Interpretation of Modal $\mu$-Calculus

12:00 Lunch

*Chair: Radha Jagadeesan*
14:00 Anna Philippou: Weak Bisimulation of Probabilistic Systems
14:30 Roberto Segala: Decidability Results for Probabilistic Bisimulation
15:00 Mariëlle Stoelinga: Normed Simulations and Bisimulations

15:30 Coffee

*Chair: Moshe Vardi*
16:00 Luca de Alfaro: Concurrent Games

19:30 Panel discussion chaired by Moshe Vardi

**Friday, May 5, 2000**

*Chair: Roberto Segala*
9:00 Frits Vaandrager: Testing for Probabilistic Automata
9:30 Suzana Andova: Probabilistic Process Algebra
10:00 Susanna Donatelli: Verification and Evaluation on the EC Project TIRAN

10:30 Coffee

*Chair: Marta Kwiatkoswka*
11:00 Summing up and general discussion

# 3 Collected Abstracts

### CHARON: Modular Specification and Simulation of Hybrid Systems

*Rajeev Alur*
University of Pennsylvania

We describe a language, called CHARON, for modular specification of interacting hybrid systems. For hierarchical description of the system architecture, CHARON supports building complex agents via the operations of instantiation, hiding, and parallel composition. For hierarchical description of the behavior of atomic components, CHARON supports building complex modes via the operations of instantiation, scoping, and encapsulation. Features such as weak preemption, history retention, and externally defined Java functions, facilitate the description of complex discrete behavior. Continuous behavior can be specified using differential as well as algebraic constraints, and invariants restricting the flow spaces, all of which can be declared at various levels of the hierarchy. The modular structure of the language is not merely syntactic, but can be exploited during analysis. We illustrate this aspect by presenting (1) a scheme for modular simulation in which submodes can integrate at a finer time scale than the enclosing modes, (2) a compositional trace semantics for modes with refinement calculus for the discrete subset of CHARON, and (3) heuristics for symbolic model checking for the discrete subset that exploit the hierarchical structure.

---

### Abstraction in Probabilistic Process Algebra

*Suzana Andova*
Eindhoven University of Technology

In this work we treat the problem of abstraction in fully probabilistic process algebra and its semantics based on branching bisimulation. Since the idea of fairness rules together with abstraction (introduced by the abstraction operator $tau_I$ and constant $tau$ meaning an internal action) is central to the verification techniques in process algebra we introduce verification rules in fully probabilistic process algebra that arise rather in a natural way from the one defined in standard process algebra. These rules express the idea that due to a non-zero probability for a system to execute an external action, abstraction from internal step will yield the external step(s) with probability 1 after finitely many repetitions. The new result in our approach is the definition of a probabilistic branching bisimulation that

relates processes with the same branching structure and the same reachability probabilities of their initial states. Moreover, we expect that the extension with non-determinism can be achieved on the basis of the results in this work.

---

**Probabilistic Timed Automata**

*Danièle Beauquier*
Université Paris 12

We introduce a notion of probabilistic timed automaton as a tool to model uncertainty in the behavior of transition systems with continuous time. Firstly we prove that given a Markov Decision Process (MDP) and a fixed subset of states, there is a Markov policy which maximizes everywhere the probability to reach infinitely often. Moreover such a maximum policy is computable in polytime in the size of the MDP. Secondly we apply this result to probabilistic timed automata to prove our main theorem : given a probabilistic timed automaton $A$ and a fixed subset of locations $F$, there is a Markov policy which maximizes everywhere the probability to reach $F$ infinitely often. Moreover such a maximum policy is computable in polytime in the size of the region automaton $R(A)$ associated to $A$.

---

**A Theory of Testing for Markovian Processes**

*Marco Bernardo*
Università di Torino

We present a testing theory for Markovian processes based on a quantification of the probability with which they pass tests within a given amount of time, in order to establish in this setting a notion of process efficiency which may be useful for the analysis of soft real time systems. Our Markovian testing theory is shown to enjoy close connections with the classical testing theory of De Nicola-Hennessy and the probabilistic testing theory of Cleaveland-Smolka et al. The relationship between the induced Markovian testing equivalence and the Markovian bisimulation equivalence is also presented. In order to ease the task of establishing testing related relationships between Markovian processes, a fully abstract alternative characterization of our Markovian testing preorder is developed which is

based on extended traces. A proof technique is derived from such an alternative characterization. It is also demonstrated that our Markovian testing equivalence, which is based on the (easier to work with) probability of executing a successful computation whose average duration is not greater than a given amount of time, coincides with the Markovian testing equivalence based on the (more intuitive) probability of reaching success within a given amount of time. Finally, it is shown that it is not possible to define a Markovian preorder which turns out to be consistent with reward based performance measures. This justifies the fact that a generic notion of efficiency has been considered.

---

**Stochastic Process Algebras with General Distributions**

*Mario Bravetti*
Università di Bologna)

We introduce the model of Interactive Generalized Semi-Markov Processes (IGSMPs) which describes concurrent systems with probabilistic time delays with a general probability distribution. Basically an IGSMP is a combination of a Generalized Semi-Markov Process (GSMP), representing the probabilistic timed behavior of a system through a set of clocks each with a given probabilistic duration, and a transition system labeled with actions, representing the synchronization behavior of the system. From an IGSMP describing a complete system (obtained as the parallel composition of several IGSMPs) we can easily derive either a GSMP to evaluate performance or a timed automata to verify real-time properties via model-checking. We then present the calculus of Interactive Generalized Semi-Markov Processes which is obtained by simply extending CSP with a delay prefix consisting of a general probability distribution. Two main approaches have been considered to derive IGSMPs from algebra terms: one based on static names, where a clock name for a delay is derived at compile time, i.e. according to its syntactical position, and another one based on dynamic names, where a clock name for a delay is derived at run-time, i.e. depending on the order of execution of delays. The approach based on static names turned out to lead to a simple operational semantics, but also to a very complex notion of equivalence that matches names of clocks that cannot be checked using standard tools and cannot be used to reduce models. The approach based on dynamic names turned out to lead to a more complex operational semantics based on a new technique called levelwise renaming, but to a very simple notion of equivalence which is basically standard probabilistic bisimulation and can be used to reduce system models. In the case of the dynamic approach we have also defined a notion of weak bisimulation and we have provided an axiomatization which is complete also on recursive systems.

**Using Decision Diagrams for the Solution of Large Markov Chains**

*Gianfranco Ciardo*
College of William and Mary

The numerical solution of a high-level model having an underlying continuous time Markov chain (CTMC) requires the generation and storage of several objects:

- The state space $S$, a set of structured states (we assume they are vectors of natural numbers of size $K$).
- An indexing function $f$ that maps a state to an index in the range 1,...,$|S|-1$.
- The transition rate matrix $R$, of size $|S| \times |S|$, which specifies the rate at which the process moves from state to state.
- A probability vector $p$, of size $|S|$, where the numerical method will compute the stationary probability of each state (assuming this is our goal).

Traditional explicit methods for storing $S$ and $R$ require memory proportional to the number of states and nonzero entries, respectively, hence they are severely limited by the available RAM.
We consider instead implicit methods based on the idea of decision diagrams. We use multivalued-decision diagrams (MDDs) to store $S$. This allows us to compute the indexing function $f(i_K, \ldots, i_1)$ in time $O(K)$, while the time spent to enumerate $S$ is even better, only $O(1)$ per state. With this technique, we can generate and store enormous state spaces in very little time and memory. For the storage of $R$, the Kronecker approach has been widely used to store a super-matrix $R'$ of the actual matrix $R$, but the problem with that approach is that unreachable entries must be somehow ignored, for example by testing each of them at each iteration and check whether it corresponds to states in $S$. We use instead matrix diagrams (MDs), which combine the idea of the MDD storage of $S$ with the Kronecker approach, resulting in a compact data structure which encodes exactly $R$, not a super-matrix of it. When used in the numerical solution, MDs provide "by-columns" access without additional overhead (unlike the Kronecker representation of $R$), and result in lower per-iteration cost.
By using MDDs and MDs, we can encode exactly and efficiently both $S$ and $R$, allowing the solution of CTMCs at least one order of magnitude larger than with traditional methods. The main remaining obstacle to the solution of even larger model is the probability vector, for which explicit full storage still appears to be required in general, unless we resort to approximations.

**Specification and Analysis of Stochastic Timed Systems**

*Pedro D'Argenio*
University of Twente

The design and analysis of various types of systems, like embedded systems or communication protocols, require insight in not only the functional, but also in the real-time and performance aspects of applications involved.
Traditionally, there was a clear separation between the functional and performance aspects of systems, and as a result different communities have constructed and analysed their own, largely unrelated models for the aspects under their responsibility. Nevertheless, both functional and performance are important features to be studied and analysed in the design stage of the development process of a system. As a consequence, it would be beneficial to be able to check how changes in functionality affect performance issues, and vice versa. In addition, one would like to have a better control over the relation between the models that are used for qualitative and quantitative analysis, and avoid the use of different models for different aspects that seem mutually incompatible. Thus, a single framework where both aspects could be defined would therefore be advantageous for several reasons.
Concretely, we propose a framework in which verification techniques and stochastic techniques can be applied to analyse the functional correctness and the performance and reliability of soft real-time systems. We introduce a stochastic process algebra for discrete event systems (called *spades* and denoted ♤). In order to give semantics to ♤, we also introduce a model which is an extension of traditional automata with clocks which are basically random variables: the *stochastic automata* model. Although stochastic automata are adequate to analyse systems since they are finite objects, they are still too coarse to serve as concrete semantic objects. Hence, we introduce a type of *probabilistic transition system* that can deal with arbitrary probability spaces. Notions of bisimulation-based equivalences are defined both on stochastic automata and probabilistic transition systems. The formal framework is completed by providing an axiomatisation for ♤
Moreover, we provide verification and performance analysis techniques that can be applied to ♤ and the stochastic automata model.

---

**Concurrent Games**

*Luca de Alfaro*
University of California, Berkeley

(joint work with Orna Kupferman and Thomas A. Henzinger)

We consider two-player games which are played on a finite state space for an infinite number of rounds. The games are concurrent, that is, in each round, the two players choose their moves independently and simultaneously; the current state and the two moves determine a successor state. We consider omega-regular winning conditions on the resulting infinite state sequence. To model the independent choice of moves, both players are allowed to use randomization for selecting their moves. This gives rise to the following qualitative modes of winning, which can be studied without numerical considerations concerning probabilities: sure-win (player 1 can ensure winning with certainty), almost-sure-win (player 1 can ensure winning with probability 1), limit-win (player 1 can ensure winning with probability arbitrarily close to 1), bounded-win (player 1 can ensure winning with probability bounded away from 0), positive-win (player 1 can ensure winning with positive probability), and exist-win (player 1 can ensure that at least one possible outcome of the game satisfies the winning condition).
We describe algorithms for computing the sets of winning states for each of these winning modes with respect to general omega-regular winning conditions. For Rabin-chain games, the sets can be computed in $n^{O(m)}$ time, where $n$ is the size of the game structure and $m$ is the number of pairs in the Rabin-chain condition. While this complexity is in line with traditional turn-based games, where in each state only one of the two players has a choice of moves, our algorithms are considerably more involved than those for turn-based games. This is because concurrent games violate two of the most fundamental properties of turn-based games. First, concurrent games are not determined, but rather exhibit a more general duality property which involves multiple modes of winning. Second, winning strategies for concurrent games may require infinite memory. In particular, infinite-memory winning strategies are needed already to limit-win Buechi games, and to almost-sure-win Rabin-chain games.

---

**Algorithmic Verification of Probabilistic Systems**

*Luca de Alfaro*
University of California, Berkeley

In this talk, we provide an overview of algorithms for the analysis of reliability and performance properties of probabilistic systems. We model probabilistic systems as Markov

decision processes, with additional labels describing their timing behavior. This model is closely related to that of Probabilistic I/O Automata of Segala and Lynch (1994). We present algorithms for the computation of reachability probability, reachability time, and performance (expressed as the long-run average outcome duration of specified tasks). We show that all these problems can be solved either by reduction to linear programming, or by iterative algorithms. Furthermore, we show that there is a close relation between these iterative algorithms and the usual mu-calculus algorithms used for the verification of reactive systems.

---

**Verification and Evaluation in the EC Project TIRAN**

*Susanna Donatelli*
Università di Torino

The talk discusses the lessons learned in the modelling of a fault tolerance solution architecture built in an on-going Esprit project called TIRAN. Goal of the project is to devise a portable software solution to the problem of fault tolerance in embedded systems, while the goal of the evaluation is to provide evidence of the efficacy of the proposed solution. Petri nets with stochastic durations are used as the basic modelling formalism, while the requirements of high flexibility and modularity posed by the industrial partner to achieve reuse, have lead to a solution based on compositionality. An example is used to show the intertwining between the verification of logical properties and the evaluation of performance aspects. Since the interest is for assessing both correctness and performance of the proposed solution, we have cared for these two aspects at the same time, and, by means of an example, we show how this was a central aspect of our analysis.

---

**Probabilistic Verification of Multiple-Valued Functions**

*Elena Dubrova*
KTH

This paper describes a probabilistic method for verifying the equivalence of two multiple-valued functions. Each function is hashed to an integer code by transforming it to an

integer-valued polynomial and the equivalence of two polynomials is checked probabilistically. The hash codes for two equivalent functions are always the same. Thus, the equivalence of two functions can be verified with a known probability of error, arising from collisions between inequivalent functions. Such a probabilistic verification can be an attractive alternative for verifying functions that are too large to be handled by deterministic verification methods.

---

### Performance and Dependability Evaluation

*Boudewijn Haverkort*
RWTH Aachen

After an introduction about what the aims and goals of performance and dependability are, Markov chains will be addressed as central model for performance and dependability evaluation. Focussing on continuous-time Markov chains, it will be discussed how they can be analysed for their long-term (steady-state) probabilities and for their transient probabilities (focussing on a method known as uniformisation).
To illustrate current research activities in performance and dependability evaluation, a new fixed-point method to evaluate, in an approximate manner, large networks of complex queueing systems, in which customer losses due to finite buffers are allowed will be presented. For this class of models, no exact solution methods are known. The new method, named FiFiQueues, is fast and provides accurate results.
As a second research activity, the class of quasi-birth-death models will be presented. With recently developed solution techniques, also infinite- state Markov models can be studied/solved. An example will be given, as well as a high-level model specification language based on stochastic Petri nets.

---

### Probabilistic Asynchronous $\pi$-Calculus

*Oltea Mihaela Herescu*
Pennsylvania State University

We propose an extension of the asynchronous $\pi$-calculus with a notion of random choice. We define an operational semantics which distinguishes between probabilistic choice, made

internally by the process, and nondeterministic choice, made externally by an adversary scheduler. This distinction will allow us to reason about the probabilistic correctness of algorithms under certain schedulers. We show that in this language we can solve the electoral problem, which was proved not possible in the asynchronous $\pi$-calculus. Finally, we show an implementation of the probabilistic asynchronous $\pi$-calculus in a Java-like language.

---

**Markov Chain Algebra**

*Holger Hermanns*
Universiteit Twente

Markov chains are widely used as stochastic models to study and estimate performance characteristics of various nature. This talk addresses the issue of compositional specification and analysis of continuous time Markov chains. After introducing the basics of Markov chains and process algebra, we integrate these two antipodes in a single formalism, Interactive Markov Chains (IMC). The ideas behind IMC are substantially different from all other existing approaches to compositional Markov chain generation. Actions and delays are strictly seperated, and their interrelation is governed by the notion of maximal progress. I will discuss why this treatment leads to a compositional formalism with the following distinguishing properties:

- IMC are equipped with substitutive notions of strong and weak equivalence, allowing us to develop an algebra of IMC, where non-stochastic process algebra forms a proper subalgebra. Furthermore, continuous time Markov chains form another, orthogonal subalgebra of IMC. In the latter, both equivalences coincide with the notion of lumpability.

- Sound and complete axiomatisations for strong and weak equivalence are developed. The axiomatic treatment of maximal progress in IMC solves an open problem for timed process calculi in general, and can be adapted to solve similar problems for process calculi with priorities.

- In order to support a compositional specification style, means to specify time constraints in a constraint-oriented style are introduced. As a side result, this enables the smooth incorporation of generally distributed delays into IMC, since each time constraint can be governed by some phase-type distribution.

To conclude the presentation, these results are discussed in the context of Markov chain model checking.

**Model Checking Loosely Specified Probabilistic Systems**

*Michael Huth*
Kansas State University

We re-interpret the category of relations according to VIEWS, pairs of domains $(P, T)$ with interpretations of [] and <> such that $T$ embeds as a set onto the set of maximal elements of $P$. A qualitative view $(3, 2)$ renders the modal transition systems of K. Larsen and B. Thomsen as a partial version, $R : X \times Y \to 3$, of ordinary relations, $R : X \times Y \to 2$. A quantitative view represents relations as fuzzy ($T$ is the unit interval $[0, 1]$) or interval-valued ($P$ is the interval domain $ID$) relations. We specify functors mediating between these categories to provide soundness of these interpretations. As for probability theory, we propose the view ($ID$, $[0, 1]$) to realize the set of probability measures as the set of maximal elements of a space of PARTIAL probability measures. We use these structures to present a framework in which total and partial system description, abstraction, and finite-state model checking all have a uniform presentation across various levels of qualitative and quantitative views together with mediating abstraction and concretization maps. We prove safety results for abstractions within and across such views for the entire modal mu-calculus and show that such abstractions allow for some compositional reasoning à la CCS.

---

**Probabilistic Interpretation of Modal Mu-Calculus**

*Purush Iyer*
North Carolina State University

This paper presents a semantics for alternation-free formulas of the mu-calculus with respect to transition systems in which some states are probabilistic. Using this semantics one may precisely define the probability with which such a system satisfies a formula in the logic. An algorithm for computing these probabilities is also given. Our approach generalizes the results for purely probabilistic systems, and purely non-deterministic systems.

---

**Approximate Reasoning on Partial Labeled Markov Processes**

*Radha Jagadeesan*
Loyola University Chicago

(joint work with J. Desharnais, V. Gupta and P. Pananagaden)

We illustrate the subtlety of the interaction of logical reasoning, and approximations in the presence of continuous probabilities via examples. We tackle these problems by providing two metric space structures on labeled Markov processes. Our first metric permits us to cope with perturbations of probability numbers and supports compositional reasoning on labeled Markov processes via non-expansivity results on process combinators. Our second metric provides a Polish space structure on labeled Markov processes, enabling the approximate calculation (upto epsilon) of numerical observations (such as integrals) on labeled Markov processes.

---

**Probabilistic Verification of Boolean Functions and Partitioned-OBDDs**

*Jawahar Jain*
Fujitsu Labs of America Inc.

(joint work with Kartik Mohanram, Ingo Wegener, Nur Touba, Dinos Moundanos)

We take an overview of an integer-valued arithmetic transform for Boolean functions, first presented in 1991, 1992 by Jain, Bitner, Abraham, Fussell (ICCAD91, Formal Methods in System Design, July 92) and show how it can form basis for efficient probabilistic verification. Then we discuss an open problem in construction of Reduced Ordered Binary Decision Diagrams (ROBDDs) using composition, and prove that the worst case complexity of the construction is truly cubic. With this insight we we show that the process of composition naturally leads to the construction of (even exponentially) compact partitioned-OBDDs (POBDDs). Our algorithm which incorporates dynamic partitioning, leads to the most general (and compact) form of POBDDs - graphs with multiple root variables. **These graphs lead to very efficient probabilistic verification**. To show that our algorithm is robust and practical, we showed very encouraging experimental results on practical industrial circuits which could be hashed (probabilistically verified) using our algorithm. Note, our approach can generate graphs which are even orders of magnitude smaller.

**Analysing Markov chains by Model Checking**

*Joost-Pieter Katoen*
Universiteit Twente

Over the last two decades many techniques have been developed to specify and evaluate Markovian dependability models. Most often, these Markovian models are automatically derived from queueing networks, stochastic Petri nets, stochastic process algebras, or stochastic activity networks. However, whereas the model specification has become very comfortable, the specification of the dependability measures of interest most often has remained fairly cumbersome. In this talk we show that our recently introduced logic CSL (continuous stochastic logic) provides ample means to specify state - as well as path-based dependability measures in a compact and flexible way. Moreover, due to the formal syntax and semantics of CSL, we can exploit the structure of CSL-specified dependability measures in the dependability evaluation process ("measure-driven state space generation"). Typically, the under- lying Markov chains that need to be evaluated can be reduced considerably in size by this structure exploitation.

---

**Markov Chain Analysis with Kronecker Representations**

*Peter Kemper*
University of Dortmund

Performance analysis with Markov chains suffers from the well known state space explosion problem which results in extremly large equation systems. Such equation systems need to be solved, e.g. to compute a steady state distribution of a continuous time Markov chain. Kronecker representations are based on Kronecker algebra, a specific matrix algebra which allows to represent large matrices as sums of Kronecker products over a set of small matrices. The key advantage is an extremely compact representation which allows to perform iterative, numerical solution methods in an efficient manner. The presentation adresses two kinds of Kronecker representations: modular and hierarchical ones. The latter is a generalization with more structural information while a modular Kronecker representation is more simple but needs additional care to handle unreachable, irrelevant states, which are articifically introduced by construction. Markov chains with a Kronecker representations can be analyzed with respect to a steady state or transient probability distribution to obtain performance measures like utilization, throughput etc. but also for functional analysis, e.g. for model checking temporal logic formulas as for computational tree logic (CTL).

### Separation Theorems for Probabilistic Distributed Programs

*Annabelle McIver*
Oxford University

Kozen's calculus of regular expressions has a simple interpretation over a relational model which, in turn, yields a simple calculus for reasoning about distributed protocols. Many separation-style theorems which underly the correctness of many distributed protocols may be proved simply and efficiently in this calculus.
This talk described how a slight weakening of the calculus can be interpreted over a relational-style model general enough to accommodate probabilistic distributed protocols. This suggests that separation theorems are also applicable in the probabilistic context, and some examples were given.

---

### Model Checking Action-Labelled Continuous Time Markov Chains

*Joachim Meyer-Kayser*
Universität Erlangen-Nürnberg

Stochastic Process Algebras (SPA) have become a powerful method for the compositional modelling and performance analysis of concurrent systems. Their main features are composition of components, the abstraction from a component's internal behaviour, and model reduction through the concepts of equivalence and bisimulation. For analysing the behaviour of a SPA model, existing tools generate the underlying stochastic process, i.e. an action-labelled continuous-time Markov chain, which is analysed by standard methods. The results of such an analysis are transient or long-run state probabilities from which user-defined measures of interest can be computed.
Unfortunately, this approach contains a disturbing shift of paradigm: While the model specification via SPA is fully behaviour-oriented, the definition of measures of interest and their calculation is altogether state-oriented. This discrepancy clearly hinders the further proliferation of the SPA approach, since users are forced to provide both the behaviour-oriented model specification and a state-oriented definition of measures of interest.
In order to avoid the above mentioned shift of paradigm, we develop an entirely behaviour-oriented analysis methodology based on Model Checking: A model is specified with the help of a SPA and the measures of interest are formulated in the form of properties specified by the action-based continuous stochastic logic (aCSL), which is strongly inspired by the logics ACTL and CSL. We propose the syntax, semantics and the model checking

algorithm which decides whether a given specification satisfies a particular property. Furthermore, we discuss the relationship between aCSL and CSL.

### Abstract Interpretation of Probabilistic Semantics

*David Monniaux*
Ecole Normale Suprieure

Abstract interpretation is a general framework for the analysis of computer programs and other automatic systems. It gives a formal notion of "safe approximation": the property that is decided by an abstract interpreter, in general, is not equal to the property that the user wants to check (typically: "can the system enter this state?"), which is often undecidable, but implies that property. With adequate heuristics, an abstract interpreter can most often solve the questions automatically. Applications include critical system certification and compiling optimizations (ex: automatic, static, check for array bounds). Our goal is to apply this framework to probabilistic programs and systems. We first define a semantics for probabilistic programs, akin to Kozen's, as continuous linear operators on measures. We then define abstract interpretation using sets of measures instead of sets of points and give a method for the analysis of fixpoints. As a particular case of this framework, we introduce a method to lift a "normal" abstract domain to a probabilistic one and give some experimental results on the precision attained.

---

### Approximating Labelled Markov Processes

*Prakash Panangaden*
McGill University

(joint work with Josee Desharnais, Radhakrishnan Jagadeesan, Vineet Gupta)

Labelled Markov processes are probabilistic versions of labelled transition systems. In general, the state space of a labelled Markov process may be a continuum. We show that the collection of labelled Markov processes carries a Polish-space structure with a countable basis given by finite-state Markov chains with rational probabilities; thus permitting the approximation of quantitative observations (e.g. an integral of a continuous function) of a continuous-state labelled Markov process by the observations on finite-state Markov chains. The primary technical tools that we develop to reach these results are

- A finite model theorem for the modal logic used to characterize bisimulation

- An isomorphism between the category of Markov processes (with simulation morphisms) with the $\omega$-continuous dcpo *Proc* (defined as the solution of the recursive domain equation $Proc = \prod_{\mathtt{Labels}} \mathcal{P}_{\mathtt{Prob}}(Proc)$.

The isomorphism between labelled Markov processes and *Proc* can be independently viewed as a full-abstraction result relating an operational (labelled Markov process) and a denotational model (following Abramsky's "A domain equation for bisimulation"), and yields a logic complete for reasoning about simulation for continuous-state processes.

---

**Verifying Randomized Distributed Algorithms with Prism**

*David Parker*
University of Birmingham

(joint work with Marta Kwiatkowska, Gethin Norman)

This talk introduces Prism, an experimental probabilistic symbolic model checker being developed at Birmingham. The tool is designed to support efficient automatic verification of systems exhibiting both probabilistic and nondeterministic behaviour, in particular randomized distributed algorithms. These are modelled as concurrent probabilistic systems, similar to Markov decision processes. System specifications are given as formulas in the probabilistic temporal logic PCTL. The tool performs symbolic model checking using BDDs and MTBDDs. Prism also has a system description language, based on modular composition. Each module is described as a set of guarded probabilistic commands. Through a direct translation from this language into MTBDDs, an efficient symbolic encoding is obtained, exploiting structure in the system being modelled. This allows us to construct models with up to $10^{30}$ states and, using reachability based precomputation algorithms, verify some liveness properties against them.

---

**Weak Bisimulation for Probabilistic Systems**

*Anna Philippou*
University of Cyprus

(joint work with Insup Lee and Oleg Sokolsky)

In this talk, we introduce weak bisimulation in the framework of Labeled Concurrent Markov Chains, that is, probabilistic transition systems which exhibit both probabilistic and nondeterministic behavior. By resolving the nondeterminism present, these models can be decomposed into a possibly infinite number of computation trees. We show that in order to compute weak bisimulation it is sufficient to restrict attention to only a finite number of these computations. Finally, we present an algorithm for deciding weak bisimulation which has polynomial-time complexity in the number of states of the transition system. This talk is based on a paper to be presented at CONCUR'00.

---

**Performance Evaluation of Computer and Communications: An Overview**

*Martin Reiser*
GMD Bonn

This talk gives an overview of performance evaluation from a historical perspective. Success stories, failures and challenges for the future have been identified.

---

**Computing Rare Event Probabilities on Large Markov Models**

*Gerardo Rubino*
IRISA Rennes

Rare event probabilities arise frequently in performance evaluation (think of loss probabilities in some communications systems) and in dependability evaluation (for instance, the unavailability of a repairable system composed of highly reliable components is in general a small number). To get an idea and very roughly speaking, "small probabilities" means here $10^{-6}$ or less. Their estimation (by means of a Monte Carlo technique) is difficult or

impossible with the natural implementation of the standard estimator. When associated with a model having a large state space, their numerical computation is again difficult, or even impossible, since the needed computatioal effort is proportional to the size of the model. In this presentation, we present two approaches whose aim is to cope with these difficulties. The general framework is Markov models of multicomponent repairable systems. The two techniques use some knowledge of the structure of the considered systems, which is usually available. One consists of importance sampling Monte Carlo schemes that can be very efficient, in particular when the systems belong to specific (and commonly found) classes. The other technique allows to derive tights bounds of stationary metrics (generally speaking, of the asymptotic performability measure). Both approaches have in common the fact that they are particularly efficient if the events of interest are very rare.

---

**Probabilistic Models of Secure Information Flow**

*Andrei Sabelfeld*
Chalmers University of Technology

(joint work with David Sands)

When is an untrusted program safe to use? One aspect of safety is confidentiality. Given you have some confidential (high) data and some public (low) data in your computer, you want to make sure the attacker – the supplier of the untrusted code – will not learn anything about your personal data, despite the fact that the application (e.g., a spreadsheet) may require legitimate access to the confidential data in order to perform its task, and legitimate communication with the supplier of the code (e.g., a registration process for all users).
We assume that the attacker is external to the (trusted) system upon which the program is run. Our aim is to specify when a program is safe to run – from the point of view of its confidentiality properties – with an aim to provide automatic methods for certifying programs prior to execution.
This talk proposes a probability-sensitive confidentiality specification – a form of probabilistic noninterference – for a small multi-threaded programming language with dynamic thread creation. Probabilistic covert channels of information flow arise from a scheduler which is probabilistic. Since scheduling policy is typically outside the language specification for multi-threaded languages, we describe how to generalise the security condition in order to define robust security with respect to a wide class of schedulers, not excluding the possibility of deterministic (e.g., round-robin) schedulers and program-controlled thread priorities. The formulation is based on an adaptation of Larsen and Skou's notion

of probabilistic bisimulation. We show how the security condition satisfies compositionality properties which facilitate straightforward proofs of correctness for, e.g., security type systems. We illustrate this by defining a security type-system and proving it correct.

---

**Representation of Discrete Functions with Mod-p-Decision Diagrams**

*Harald Sack*
Universität Trier

In computer aided design of very large scale integrated circuits (CAD for VLSI) Ordered Binary Decision Diagrams (OBDDs) became an established standard data structure esp. for tasks as formal verification. OBDDs are well suited for the representation of most functions of practical relevance but, unfortunately, not for all. This restriction leads to the investigation of more general, less restrictive data structures. We are going to present Parity-OBDDs, OBDDs that are containing additional functional nodes, here associated with the parity function. For practical application of Parity-OBDDs an efficient equivalence test is of high importance, because the data structure is not canonical.
Many problems in logic design and combinatorial optimization are formulated with functions over discrete domains. To apply OBDDs to these functions a binary encoding step in required, which becomes superfluous if we are extending the decision digram concept to the discrete domain. These Multiple valued Decision Diagrams (MDDs) in the same way can be extended by introducing functional nodes, and thus, instead of using the parity function we can apply the addition-modulo-p operation and we are dealing with Mod-p-Decision Diagrams. As for Parity-OBDDs, Mod-p-DDs also require an efficient equivalence test, because of their property of being not canonical. Up to now, there is no fast deterministic equivalence test. The required time is cubic in the number of nodes and therefore, too slow for practical applications. We present a probabilistic eauivalence test for Mod-p-DDs that can be performed in time linear in the number of input variables.

---

**Randomized BDDs**

*Martin Sauerhoff*
Universität Dortmund

(includes results of F. Ablayev, M. Agrawal, M. Karpinski, and Th. Thierauf)

Randomized BDDs (introduced by Ablayev and Karpinski) are BDDs which may have additional *probabilistic nodes*. During the computation for a given input, the successor of such a node is chosen according to the outcome of a fair coin-toss. Randomized variants of the usual restricted variants of BDDs, like OBDDs and FBDDs (free BDDs) are defined in a straightforward way by requiring that the non-probabilistic nodes fulfill the respective restriction.

In the talk, some of the more practically relevant results for randomized OBDDs which have been obtained in the last years have been reviewed. Especially, it has been discussed how the algorithmic problems which are crucial for formal verification may be solved with randomized OBDDs.

Briefly summarized, the current state of knowledge is as follows. On the positive side, it could be shown that randomized OBDDs allow a succinct representation for several functions for which deterministic OBDDs have large size. Furthermore, randomized OBDDs may be combined by Boolean operations using the well-known "apply" algorithm for the deterministic case. On the negative side, a decisive drawback of randomized OBDDs is that testing satisfiability (more precisely, even its "promise" variant) is NP-complete. Hence, only heuristic solutions for this problem remain. Furthermore, it is an open problem how the power of randomness can be exploited in an automated process for generating randomized OBDDs, e. g., from combinational circuits.

Altogether, it does not seem to be very likely that randomized OBDDs can be used for applications, at least not for formal verification. Apart from these rather disappointing news for practice, the investigation of randomized BDDs has lead to a "more complete" picture of the different types of BDDs in theory, and has stimulated the current development in the area of complexity theory which deals with proving lower bounds for more and more general variants of BDDs.

---

**Decidability Results for Probabilistic Bisimulations**

*Roberto Segala*
Università di Bologna

In this talk we propose decision algorithms for the different notions of strong and weak bisimulations that arise from working in the alternating and non alternating model and from simulating a transition by means of a randomized and non-randomized scheduler. Several algorithms are derived directly from existing literature on the alternating model, showing that in most cases the two models are the same; for bisimulations that use randomized schedulers we show that in the alternating model we obtain the same relations as with deterministic scheduler, while in the non alternating model we obtaine weaker relations whose decidability is equivalent to deciding equivalence between convex hulls of points in an $n$-dimensional space, where $n$ is the number of states.

---

**Model Checking Probabilistic Timed Automata**

*Jeremy Sproston*
University of Birmingham

(joint work with Marta Kwiatkowska, Gethin Norman, and Roberto Segala)

In this talk, we consider the formal description of real-time systems in terms of a model which exhibits both nondeterministic and probabilistic choice. For this purpose, we extend the timed automata of Alur and Dill with discrete probability distributions to define probabilistic timed automata. We then consider three model checking approaches for this class of model with respect to probabilistic, timed properties. The first employs a finitary partition of the state space of a probabilistic timed automaton, using the notion of 'region equivalence', which has been used widely in the analysis of non-probabilistic timed automata. Given this partition, it is then possible to obtain a model checking method for the verification of the model against properties of a probabilistic, timed temporal logic. The second verification technique employs forward exploration through the state space of a probabilistic timed automaton to establish bounds on the likelihood of reachability properties. The final technique makes use of backward reachability to model check a subset of properties of our probabilistic, timed temporal logic.

**Normed Simulations and Bisimulations**

*Mariëlle Stoelinga*
Universiteit Nijmegen

In this talk, I consider action-labelled systems with non-deterministic and probabilistic choice. Using the concept of norm functions, I introduce two types of bisimulations called (strict) normed bisimulation equivalence that allow for delays when simulating a transition and are strictly between strong and weak bisimulation equivalence known from literature. A suitable modification of the prominent splitter/partitioning technique, yields a polynomial-time algorithms that constructs the quotient space of the (strict) normed bisimulation equivalence classes. This talk is based on joint work with Christel Baier (University of Bonn) and has been published in this year's Fossacs proceedings (LNCS 1784).

---

**Testing Probabilistic Automata**

*Frits Vaandrager*
University of Nijmegen

(joint work with Mariëlle Stoelinga)

A basic idea in concurrency theory is that two systems are deemed equivalent if they cannot be distinguished by observation. Depending on the power of an observer, different notions of behavioral equivalence arise. For nondeterministic automata, this idea has been thorougly explored (for an overview see Van Glabbeek's PhD Thesis on Comparative Concurrency Semantics and Refinement of Actions) and a large number of equivalences has been characterized operationally, algebraically, denotationally, logically, and via intuitive "button pushing scenarios". In this talk we start with a brief overview of the various button pushing scenarios that have been proposed for nondeterministic automata.
Recently, also a large number of equivalences for probabilistic automata has been proposed. However, thus far none of these equivalences has been characterized in terms of button pushing scenarios. In this talk we propose an extremely simple and (to our taste) intuitive button pushing scenario for probabilistic automata, based on the obvious idea that one can only observe probabilistic behavior by repeating experiments and by applying statistical methods on the outcomes of these experiments. We show that the equivalence on probabilistic automata induced by our experimental setup coincides with the trace distribution equivalence proposed by Segala.
We argue that any behavioral equivalence should be characterized via some button pushing scenario, or should be stricly finer than such an equivalence and be justified via computational arguments.