Dagstuhl Seminar on Logic, Algebra, and Formal Verification of Concurrent Systems

26.11. – 1.12.2000

organized by

V. Diekert, M. Droste, A. Muscholl and D. Peled

The seminar brought together researchers developping formal methods and tools for the analysis of concurrent systems with those working on algebraic and logical theories that can potentially enhance the capabilities of such tools.

The interaction of different approaches and methodologies has resulted in new verification techniques and has improved existing ones. For instance, automata on infinite words are simple mathematical models that have been used as the theoretical basis for state-based automatic verification of temporal logic specifications. A recent example of using theoretical models for improving the efficiency of algorithmical analysis is the ITU standard of message sequence charts. The theoretical study of this partial-order-based notation resulted in algorithms that are used to detect errors in preliminary designs of communication softwares.

A basic concern in formal methods is to design and analyze appropriate logical and algebraic formalisms used for describing properties of systems and for checking these properties efficiently for interesting classes of models. In logic as well as in algebra, formalisms with partial-order semantics offer special challenges. For instance, both the expressivity and the algorithmic properties of temporal logic frameworks depend to a large extent upon the kind of concurrency expressed in the model. Even for the rather simple partial-order model of Mazurkiewicz traces, the existing temporal logics still do not offer a good trade-off between expressivity and complexity, in spite of the considerable progress that has been achieved over the past few years. It is not by coincidence that some of the progress was based on applying algebraic methods, which often offer in a natural way the right way to formalize properties related to concurrency.

The seminar was attended by 35 researchers and 30 talks were presented. They presented ongoing work on generalized models of concurrent systems, expressivity of temporal logics in discrete and timed partial-order models and the verification of infinite-state systems. In a special evening session Yuri Matiyasevich presented the main results and open problems related to Hilbert's 10^{th} problem, a century after its presentation to the 2^{nd} International Congress of Mathematicians.

The discussions during the seminar were very stimulating and brought an intense exchange of ideas. They showed that promising solutions for achieving a good understanding and application of concurrent behavior can result from combining several existing techniques and specification formalisms, with algebra as a guiding formalism.

As a result of the seminar, we think that a natural challenge is the question whether there is a partial-order temporal logics for Mazurkiewicz traces (or other meaningful partial-order based models) which is expressively complete but which has an elementary complexity.

The Dagstuhl team provided as always an excellent organization, for which we would like to thank on behalf of all participants. Our thank goes also to Denis Oddoux, who prepared the seminar report.

V. Diekert (Stuttgart)
M. Droste (Dresden)
A. Muscholl (Paris)
D. Peled (Murray Hill)

Contents

Abstracts of the Talks

Recognizability and regularity for languages of series-parallel pomsets
Pascal Weil

Finite series-parallel (sp) pomsets are defined either as the N-free pomsets, or the pomsets which can be constructed from the singletons using only sequential and parallel products. In this algebraic setting,we first consider the natural class of recognizable sets of sp-pomsets (sp-languages): those which are unions of classes in a finite-index congruence.

In the case of bounded-width sp-languages, recognizability is equivalent to series-rationality (constructibility from the singletons using unions, sequential and parallel products, and sequential iterations), to MSO-definability, and to regularity – defined here by means of a branching automaton model (results of Lodaya and Weil, and of Kuske).

Recognizability is not naturally restricted to bounded-width, and neither is the branching automaton model. It is shown that in arbitrary width, every recognizable sp-language is regular but the converse is not true. The regular sp-languages are exactly those described by rational expressions using unions, sequential and parallel products, sequential iteraitons, substitutions of the form $K \circ_a L$ (a is a letter and $K \circ_a L$ is the set of elements of L where all occurrences of letter a have been replaced non uniformly with elements of K) and iterations of the form $L^{+\mathsf{a}}$ (an iteration of substitution, $L^{+\mathsf{a}} = a \ \cup \ L \ \cup \ (L \circ_a L) \ \cup \ (L \circ_a (L \circ_a L)) \cup \cdots$) when L contains no sequential element, that is, no element which can be written as a sequential product.

The proof uses a detour via freer algebras where the sequential product is assumed to be associative, but the parallel product is neither associative nor commutative. In these algebras, intermediary between the term algebras and the free sp-algebras, recognizability, regularity and rationality (as defined above) coincide.

Automatic Deductive Verification with Invisible Invariants
Lenore Zuck

The talk presents a method for the automatic verification of a certain class of parameterized systems. These are bounded data systems consisting of N processes (N being the parameter), where each process is finite state. First, it is shown that, using the standard deductive INV rule for proving invariance properties, all the generated verification conditions can be automatically resolved by finite state BDD-based methods with no need for interactive theorem proving.

Next, it is shown how to use model checking techniques over finite (and small) instances of the parameterized system in order to derive candidates for invariant assertions. Combining this automatic computation of invariants with the previously mentioned resolution of the VCs (verification conditions) yields a (necessarily) incomplete but fully automatic sound method for verifying bounded data parameterized systems. The generated invariants can be transferred to the VC-validation phase without ever been examined by the user, which explains why they are termed "invisible".

The method is illustrated on a non-trivial example of a cache protocol, provided by Steve German.

Joint work with Amir Pnueli and Sitvanit Ruah.

Process Cost Functions for Concurrent Systems
Manfred Droste

We consider recognizable formal power series over semirings K and in partially commuting variables, i.e. over trace monoids. These series may be viewed as the cost functions associated to processes of a concurrent system where the transitions have costs taken in K. We show that each such function can be constructed from polynomials using the operations $+, \cdot$ (Cauchy-product), and a restricted version of $*$. Conversely, these operations preserve recognizability if the semiring is commutative resp. commutative and idempotent, depending on the version of $*$ permitted. We also define the concepts of aperiodic and of starfree formal power series, and we show that they coincide if the semiring K is idempotent and the matrix monoids $(K^{n\times n}, \cdot)$ have a Burnside property (satisfied, e.g., by the tropical semiring). This generalizes results of Schützenberger, Ochmanski, and of Guaiana, Restivo and Salemi.

Joint work with Paul Gastin.

Black Box Checking
Doron Peled

Two main approaches are used for increasing the quality of systems: in model checking, one checks properties of a known design of a system; in testing, one usually checks whether a given implementation, whose internal structure is often unknown, conforms with an abstract design. We are interested in the combination of these techniques. Namely, we would like to be able to test whether an implementation with unknown structure satisfies some given properties. We propose and formalize this problem of black box checking and suggest several algorithms. Since the input to black box checking is not given initially, as is the case in the classical model of computation, but is learned through experiments, we propose a computational model based on games with incomplete information. We use this model to analyze the complexity of the problem. We also address the more practical question of finding an approach that can detect errors in the implementation before completing an exhaustive search.

This is a joint work with Moshe Vardi and Mihalis Yannakakis.

On Rational Message Sequence Chart Languages and Relationships with Mazurkiewicz Trace Theory
Rémi Morin

Hierarchical Message Sequence Charts are a well-established formalism to specify communication protocols. In this model, numerous undecidability results were obtained recently through algebraic approaches or relationships to Mazurkiewicz trace theory. We show first the decidability of the channel-boundedness property for HMSCs, that is, one can check whether a rational language of MSCs requires only channels of finite capacity. We also explain why the proof differs from a somewhat similar study by Ben-Abdallah and Leue (TACAS 97). This first step enables us to prove our main result: one can decide whether the iteration of a given regular language of MSCs is regular if, and only if, the Star Problem in trace monoids is decidable too. This relationship justifies the restriction to strongly connected HMSCs which describe all regular finitely generated languages. We also show in this talk how this last result, known from papers by Muscholl & Peled (MFCS 99), Alur, Yannakakis (CONCUR 99), Henriksen et al. (ICALP 2000), can actually be infered from Ochmanski's theorem.

Word Problems for Mazurkiewicz Traces
Markus Lohrey

The talk presents an overview on known results about trace rewriting systems and their word problems. A trace rewriting system is a finite set of rules where the left and right hand side of each rule are Mazurkiewicz traces. These systems generalize both semi-Thue systems and vector replacement systems. The word problem for a trace rewriting system asks whether two given traces can be transformed into each other by applications of the rules, where the rules may be applied both forward and backward. It is well-known that there exist trace rewriting systems with undecidable word problems. This motivates the question for restricted subclasses of trace rewriting systems that give rise to decidable word problems. One such class is the class of confluent and terminating trace rewriting systems. Whereas for terminating semi-Thue systems as well as for general vector replacement systems confluence is known to be decidable, even for length-reducing trace rewriting systems confluence is undecidable if the underlying trace monoid is neither free nor free commutative. Finally the talk presents several complexity results for different variants of word problems for length-reducing and confluent trace rewriting systems.

Executing High-Level Message Sequence Charts
Claude Jard

The talk has presented a solution to the question of simulation of scenario languages like HMSCs. The declarative aspect of HMSCs and their relative theoretical expressiveness (due to the weak sequential composition) put several interesting questions.

The approach is to preserve the visual aspect of the formalism (in contrast to a formal language view). We use event structures to represent the semantics of HMSCs and graph grammars to encode it in a finite manner. The computation of a normal form of grammars is a pre-compilation step which allows a fast simulation of scenarios.

This work identifies also some strange HMSCs which are not simulable.

This work is currently being done in the Pampa research group at Irisa, with the collaboration of L. Hélouet (now working for France Telecom) and B. Caillaud. The algorithms have been implemented in a prototype called SLIM.

(Un-)Decidable Problems Concerning Asynchronous Cellular Automata

Dietrich Kuske

Asynchronous cellular automata (ACA) were first considered in the realm of Mazurkiewicz traces. Droste and Gastin generalized this computing device in such a way that it can run on arbitrary Hasse-diagrams of partially ordered sets without autoconcurrency (pomsets). They could in particular show that it is decidable whether an ACA accepts all/some CROW-pomset. Later, I proved the same result for a larger class of pomsets, called k-pomsets.

Here, we consider ACAs that can accept Σ-dags, slightly more general structures than Hasse-diagrams of pomsets. It is shown that the universality is undecidable. The proof (obtained in collaboration with Paul Gastin) uses a reduction of the tiling problem to Σ-dags. This undecidability proof can be used to show that several other problems are undecidable as well: The question whether a given ACA is equivalent to some deterministic one, whether two ACAs accept the same Σ-dags, whether a set of Σ-dags defined in monadic second order logic can be accepted by an ACA, and whether the complement of a recognizable set of Σ-dags is recognizable. Since ACAs in general cannot be complemented, this list of undecidable questions does not contain the emptiness. On the contrary, we show that it is decidable whether an ACA accepts some Σ-dag. This is obtained using well-structured transition systems considered by Finkel and Schnoebelen.

Quantitative Aspects of Bisimulation Collapse

Richard Mayr

Bisimulation equivalence preserves all properties expressible in the modal μ-calculus. Thus, for model checking a finite labeled Kripke structure F, it suffices to consider the quotient graph F/e modulo bisimilarity.

The step from F to F/e is called bisimulation collapse, and F/e can be efficiently computed by partition refinement.

Given a random directed labeled graph F with M arcs, N nodes and K actions, what is the expected number of nodes of the quotient graph F/e ?

We show that for sparse graphs F bisimulation collapse is very unlikely and with high probability F/e is not much smaller than F.

However, in dense graphs with few actions i.e. $M > N \log N$ and $M/N \gg K$, complete bisimulation collapse to a single state is very likely.

LTL is Expressively Complete for Mazurkiewicz Traces

Volker Diekert

It was a long standing open problem whether an analogue of Kamp's Theorem holds for (Mazurkiewicz) traces: Every first order definable language of real traces is definable in linear time logic using only Next- and Until-operators.

We have shown that this is true using algebraic techniques. The starting point was a new proof for Kamp's Theorem on finite words which can be found in the Habilitationsschrift of Thomas Wilke, see also Wilke (STACS'99). The main contribution is an extension of this new proof in order to overcome the additional difficulties due to concurrency. This has been worked out in Diekert & Gastin (ICALP'00).

As a corollary of our work we can, e.g., describe safety and liveness properties over traces very much alike as it is known for infinite sequences.

Joint work with Paul Gastin.

From Concrete Domains to Concrete Data Structures

Ingmar Meinecke

We study concrete data structures (cds) and their associated partial orders, concrete domains. Both structures were introduced by Kahn and Plotkin within the theory of denotational semantics of programming languages.

By defining a labelling of the prime intervals of a concrete domain we are able to associate labelled concrete domains and regular cds to each other. This leads to the construction of the categories RCDS of the regular cds and LCD of the labelled concrete domains. We show that these two categories give rise to a coreflexion. Moreover, the coreflection cuts down to an equivalence between LCD and the category FCDS of fully regular cds.

We show that there always exists a greatest cds generating a given concrete domain. We also obtain that each cds can be reduced to a minimal fully regular cds generating the same concrete domain. However, these minimal structures are not unique up to isomorphism as we show by an example.

Model checking CTL$^+$ and FCTL is hard
Philippe Schnoebelen

CTL$^+$ (Emerson & Halpern, JCSS 1985) extends CTL in that it allows boolean combinations of path formulae under the scope of a path quantifier. E.g. the following is a CTL$^+$ formula:

$$E[(Fa \to Fb) \wedge (cUd)]$$

FCTL (Emerson & Lei, Sci. Comp. Prog. 1987) combines CTL formulae with a general fairness assumption expressed as a boolean combination of GFa_i where the a_i's are propositional formulae. All CTL paths quantifiers are then implicitly relativised to fair paths. E.g. the following is an FCTL formula:

$$AFc \text{ under the assumption } (GFa \to GFb) \wedge (GFc \vee GFd)$$

The precise complexity of the model checking problem for these logics was not known, beyond the facts that it is NP-hard and coNP-hard, and is in Δ_2^p (the class of problems solvable by a deterministic polynomial-time Turing machine that can use a SAT oracle).

In this talk we show the above-mentioned problems are all Δ_2^p-complete. This result extends to ECTL$^+$ (from Emerson & Halpern, JACM 1986) and related but more obscure fragments of CTL*. They are the first examples of Δ_2^p-complete the field of temporal model checking.

See LSV Research Reports 2000-7 & 2000-8 for more details.

Partial Order Characterization of a Process Algebra with Iteration
Angelika Votintseva

We consider process algebra PBPA*, which is a subclass of so called PA-processes, being obtained as an extension of well-known algebra BPP (Basic Parallel Processes) with enriching its prefix operation and embedding an iterative composition into the definition of algebraic terms. For PBPA* we consider three kinds of operational semantics: interleaving, step and pomset. Besides, for these kinds of semantics we establish correspondences between algebraic bisimulations and behavioral ones, defined over event structures.

Design for Verification or How to Make a Software Specification to Look like a Hardware Specification to the Verifier

Natasha Sharygina

This talk presents and applies a methodology for integration of formal verification by automata-based model-checking into a commercially supported object-oriented software development process. We define and illustrate a set of design rules for OOA models with executable semantics, which lead to automata models with tractable spaces. The design rules yield OOA models with functionally structured designs similar to those of hardware systems, which have enabled successful application of model-checking to verification of hardware systems. The design rules are incorporated into an extended object-oriented development process for software systems. The methodology, including the design rules was applied to a NASA robot control software. The complex robot control system was decomposed into several functional subsystems prior to the verification stage. Evaluation by model checking of one control intensive subsystem is demonstrated. Results including identification of significant errors in the original robotic control system are presented.

Symbolic Analysis of Infinite-State Systems Using Extended Automata

Ahmed Bouajjani

We show that infinite-state systems like lossy/perfect FIFO channel systems, and parametric networks of processes (e.g. mutual exclusion protocols), can be modeled naturally as rewriting systems, and that they can be analyzed using (extended) automata or regular expressions as symbolic representations for their sets of reachable configurations. Then, the essential problem is to identify classes of rewriting system (relations on words), and classes of representations (languages) such that, for any rewriting system R and language L in these classes, $R^*(L)$ is effectively computable.

The talk surveys results obtained by the speaker and several co-authors :
P. Abdulla, B. Jonsson (Uppsala); A. Annichini (Verimag, Grenoble);
L. Boasson, P. Habermehl, A. Muscholl, T. Touili (LIAFA, Paris7).

On the Expressivity and Complexity of Quantitative Branching-Time Temporal Logics
Francois Laroussinie

We investigate extensions of CTL allowing to express quantitative requirements about an abstract notion of time in a simple discrete-time framework, and study the expressive power of several relevant logics (with subscripted modalities or freeze quantifiers).

When only subscripted modalities are used, polynomial-time model checking is possible even for the largest logic we consider, while introducing freeze quantifiers leads to a complexity blow-up.

A paper is available at the following web address:
`http://www.lsv.ens-cachan.fr/Publis/PAPERS/LST-latin2000.ps`

Joint work with Ph. Schnœbelen and M. Turuani.

Compositionality of Message Sequence Charts
Anca Muscholl

This talk considers some questions related to the conversion of regular MSC languages into high level MSCs (HMSCs). It is known that it is undecidable whether the set of linearizations of an HMSC is regular. However, the reverse translation (from automata to HMSCs) is decidable, as shown by Henriksen et al. (ICALP'00).

We show that the test for the conversion of automata into HMSCs is feasible in polynomial time (NLOG-complete) provided that the given automaton satisfies some strong structural conditions, namely the diamond property and fixed buffer capacities. In the more general setting of bounded automata, this test becomes co-NP-complete.

We consider then Compositional MSCs and HMSCs (CMSC, HCMSC) as a natural extension of MSCs, where we do not impose that nodes are labeled by full charts. We show that basic properties for HCMSCs (e.g., does a message get ever received) are undecidable. Other questions, for example whether every path of an HCMSC defines an execution., can be checked efficiently. Moreover, characterizations obtained for bounded HMSCs can be easily extended to HCMSCs.

Joint work with D. Peled, Bell Labs.

Logic of Fixed Points with Applications to Concurrency
Zoltàn Ésik

The general theory of fixed points is used to obtain axiomatization results for concurrent systems. It is shown that both simulation equivalence and bisimulation equivalence are finitelyt axiomatizable over iteration algebras. The equational theory of simulation equivalence agrees with the theory of the binary supremum operation on monotonic functions on complete lattices.

Distributed System Design with Message Sequence Charts
Ingolf Krüger

The methodical mastery of interaction scenarios is a key factor for capturing and modeling system requirements of distributed, reactive systems. Message Sequence Charts (MSCs) and variants thereof are well-accepted as a graphical description technique for interaction scenarios. MSCs emphasize the inter-component coordination aspect of typically partial system executions; this complements the usually complete behavior description for individual components, as given by state-oriented automaton specifications.

The topic of this presentation is the seamless, methodically founded integration of MSCs into the development process for distributed, reactive systems.

The starting point for this investigation is the definition of a precise, stream-based model for the system class under consideration. This model enables the integrated consideration of interaction-oriented and state-oriented system specifications; it also serves as the basis for the introduction of effective refinement notions for MSCs. Next, different MSC interpretations – in the range from scenario specification to complete behavior descriptions – are introduced. In addition, the application of MSCs for the description of safety and liveness properties is analyzed.

Finally, two transformation procedures, supporting the transition from interaction scenarios to complete behavior specifications for individual components, are presented. The first one schematically extracts relational assumption/commitment specifications from MSCs. The second one turns MSCs syntactically into corresponding state automata. On the one hand this makes the component properties defined by MSCs accessible to formal analysis; on the other hand this constructively bridges the gap between interaction requirements and component implementations.

Regular MSC Languages
Madhavan Mukund

Message Sequence Charts (MSCs) are an attractive visual formalism widely used to capture system requirements during the early design stages in domains such as telecommunication software. It is fruitful to have mechanisms for specifying and reasoning about collections of MSCs so that errors can be detected even at the requirements level. We propose, accordingly, a notion of regularity for collections of MSCs and explore its basic properties. In particular, we provide an automata-theoretic characterization of regular MSC languages in terms of finite-state distributed automata called bounded message-passing automata. These automata consist of a set of sequential processes that communicate with each other by sending and receiving messages over bounded FIFO channels. We also provide a logical characterization in terms of a natural monadic second-order logic interpreted over MSCs.

A commonly used technique to generate a collection of MSCs is to use a High-level Message Sequence Chart (HMSC). We show that the class of languages arising from the so-called locally synchronized HMSCs constitute a proper subclass of the languages which are regular in our sense. In fact, we characterize the locally synchronized HMSC languages as the subclass of regular MSC languages that are finitely generated.

Joint work with Jesper G Henriksen (BRICS, Aarhus, Denmark), K Narayan Kumar (CMI, Madras, India), Milind Sohoni (IIT, Bombay, India) and P S Thiagarajan (CMI, Madras, India).

Towards Model-Checking Trace Logics
Igor Walukiewicz

We discuss several logical formalisms for describing trace languages. First, we show that every first-order formula is equivalent over traces to a first-order formula using only three distinct variables. This way we obtain a local temporal logic which is expressively complete with respect to first-order logic.

Next, we present a variant of the μ-calculus over traces. This variant is PSPACE-complete and expressively complete with respect to monadic second-order logic. It has also some other properties which seem to be relevant for model checking concurrent systems. Finally, we discuss the possibilities of extending the Petri net unfolding method from reachability checking to model checking trace logics.

The Power of Algebra in Logic
Denis Thérien

First-order logic and temporal logic each provide powerful models to describe subclasses of regular languages. It is in general a difficult question to decide if a given language can be described by a formula of a given class. Quite often, the decision procedure can be given in terms of algebraic invariants of the language. We survey several examples where this is the case, and we suggest that there may be general principles that explain when such algebraic characterizations of logical formulas are possible.

Bisimulation Invariance in the Levels of the Polynomial Alternation Depth Hierarchy
David Janin

Bisimulation invariance in mathematical logics (e.g. first order logic (FOL), monadic second order logic (MSOL), ...) is a central notion (both for expressiveness and succinctness issues) in the study of logic of programs as soon as bisimulation equivalence is considered as the appropriate notion of behavioral equivalence over processes modeled as transition systems with a source. We recall here a panorama of various results investigating bisimulation invariance in this logical settings.

From [Van Benthem 76] it is known that the bisimulation invariant fragment of FOL is as expressive as modal logic. From [Janin - Walukiewicz 96] it is known that the bisimulation invariant fragment of MSOL is as expressive as Kozen's propositionnal μ-calculus. An immediate question is : does the bisimulation invariant fragment of MSOL collapses to some level of the monadic alternation depth hierarchy ? Yet we fail to answer this question.

However, investigating this (more general) question, we [Giacomo Lenzi and myself] have at least shown that the bisimulation invariant fragment of monadic Σ_2 equals the Buchi level of the monadic hierarchy (fixpoint formulas with greatest fixpoints nested by least fixpoints). Moreover, an encoding of binary parity games into some particular (bisimulation closed) class graphs over which winning positions can be expressed by a monadic Σ_3 formula shows (together with Arnold's works on the infiniteness of the μ-calculus hierarchy over binary games) that the bisimulation invariant fragment of monadic Σ_k for any $k > 2$ is not included into any level of the μ-calculus (fixpoint alternation depth) hierarchy.

Hilbert's Tenth Problem Today : Main Results and Open Problems

Yuri Matiyasevich

This was one of 23 problems which David Hilbert stated in 1900 in his address *Mathematische Probleme* to the Second International Congress of Mathematicians. The 10th problem (the only one among the 23 problems) can also be viewed as a decision problem, that is, as a problem in computer science.

It took 70 years before the problem was shown to be undecidable. The main technical result (which is sometimes referred to as *DPRM-theorem* after Davis-Putnam-Robinson-Matiyasevich) states that *every recursively enumerable set M of natural numbers is Diophantine, that is, it has a representastion of the form*

$$a \in M \Longleftrightarrow \exists x_1 \dots x_m \{D(a, x_1, \dots, x_m) = 0\}$$

where D is a polynomial with integer coefficients. This theorem serves as a bridge allowing one to transfer ideas and methods from number theory to computability theory as well as in the opposite direction.

There is quite a few problems closely related to Hilbert's 10th problem which are still waiting for their solutions.

URL: `http://logic.pdmi.ras.ru/Hilbert10` is a WWW site devoted to Hilbert's 10th problem.

Updatable Timed Automata

Antoine Petit

In classical timed automata, as defined by Alur and Dill, the only operation allowed to modify the clocks is the reset operation. We present a model that allows extended updates on clocks: for instance a clock can be set to some non-null constant or to the value of an other clock, or, in a non-deterministic way, to some value lower or higher than a given constant. Our main results are the following. First, we prove the (un)decidability of many classes of this model. Second, we propose a generalization of the region automaton to prove decidability of some interesting classes. And finally, we explore the expressiveness of all decidable classes.

Joint work with Patricia Bouyer, Catherine Dufourd, Emmanuel Fleury.

Abstractions and Partial Order Reductions for Checking Branching Properties of Time Petri Nets

Wojciech Penczek

Model checking is one of the most popular methods of automated verification of concurrent systems, e.g., hardware circuits, communication protocols, and distributed programs. However, the practical applicability of this method is strongly restricted by the state explosion problem, which is mainly caused by representing concurrency of operations by their interleaving. Therefore, many different reduction techniques have been introduced in order to alleviate the state explosion. The major methods include application of partial order reductions, symmetry reductions, abstraction techniques, BDD-based symbolic storage methods, and SAT-related algorithms.

Recently, the interest in automated verification is moving towards concurrent real-time systems. Two main models for representing such systems are usually exploited: timed automata and time Petri Nets. The properties to be verified are expressed in either a standard temporal logic like LTL and CTL*, or in its timed versions MITL, and TCTL.

Most of the efficient reduction techniques exist for linear time formalisms. The talk deals with verification of untimed branching time temporal properties of Time Petri Nets. Since Time Petri Nets have usually infinite state spaces, abstraction techniques are used to represent these by finite ones. To reduce the sizes of abstract state spaces partial order reductions are used. The main contribution of the paper relies on:

- improving the variant of the geometric region method for defining abstract state spaces preserving properties of CTL* and ACTLS* such that the structure of a verified formula is exploited,
- showing, **for the first time**, how to extend the po-reduction methods to deal with next-time free branching properties of time Petri Nets,
- combining the above two results offering an efficient method for model checking of ACTL*-X and CTL*-X properties of time Petri Nets.

Derivations of Rational Expressions with Multiplicity
Jacques Sakarovitch

In this talk, we have introduced a generalization of the *partial derivatives* of rational expressions, due to Antimirov, to rational expressions with multiplicity. We define the derivation of a rational expression with multiplicity in such a way that the result is a *polynomial of expressions.* This amounts to interpreting the addition symbol at the upper level in the semiring of coefficients. Former results of Brzozowski and of Antimirov are then expressed in that framework that allows to deal with rational power series, and automata and expressions with multiplicity as well.

This is a joint work with Sylvain Lombardy.

The Star Problem in Trace Monoids : Reductions Beyond C4
Daniel Kirsten

In the talk, we deal with the star problem in free, partially commutative monoids, also called trace monoids. It was raised by Ochmański in 1985 and means to decide whether the iteration of a given recognizable trace language is recognizable. Closely related to the star problem is the finite power problem (FPP), which means to decide whether for some given recognizable trace language L, there is some integer n such that $L^* = L^0 \cup L^1 \cup \ldots \cup L^n$. It was raised by Brzozowski in 1966 for free monoids. Due to a theorem by Richomme from 1994, both problems are decidable in trace monoids without a submonoid of the form $\{a, b\}^* \times \{c, d\}^*$ which is called C4.

In the talk, we show a new reduction. We consider trace monoids

$$\rightarrow_{n} \cong \{a_1, b_1\}^* \times \ldots \times \{a_n, b_n\}^*.$$

The main result of the talk asserts that if the star problem is decidable in $\rightarrow_{n}$ for some n, then it is also decidable in every trace monoid without a submonoid $\rightarrow_{n+1}$. The case $n = 1$ gives Richomme's theorem, above.

Consequently, to show the decidability of the star problem in a trace monoid $\triangle$, it suffices to show its decidability in the biggest submonoid $\rightarrow_{n}$ in $\triangle$. This strictly subsumes all previously know reduction steps.

The recently shown decidability equivalence between the star problem and the FPP due to Kirsten and Richomme plays a crucial role in the main proof.

An Eilenberg Type Correspondence for Recognizable Trace Languages

Mikhail Volkov

We present a version of Eilenberg's variety theorem restricted to a particular alphabet.

Let A be a fixed finite alphabet. By a *map* we mean any mapping $\mu : A \to M$ where M is a finite monoid provided that the set $A\mu$ generates M. A *morphism* $\varphi : \mu \to \nu$ between the maps $\mu : A \to M$ and $\nu : A \to N$ is a homomorphism between the monoids M and N such that $a\mu\varphi = a\nu$ for all $a \in A$. The *product* $\mu \times \nu$ of the maps $\mu : A \to M$ and $\nu : A \to N$ is the map $\pi : A \to D$ defined via $a\pi = (a\mu, a\nu)$ where D stands for the submonoid of the direct product $M \times N$ generated by the set $\{(a\mu, a\nu) \mid a \in A\}$. A *variety* of maps is any class of maps closed under the taking morphic images and products.

Every map $\mu : A \to M$ uniquely extends to a monoid homomorphism $\tilde{\mu} : A^* \to M$. We say that μ *recognizes* a language $L \subseteq A^*$ if $L = P\tilde{\mu}^{-1}$ for some $P \subseteq M$. For a variety $\Rrightarrow$, let $\mathcal{V}$ denote the class of all languages over A that can be recognized by a map in $\Rrightarrow$.

Theorem 1 *The correspondence $\Rrightarrow \to \mathcal{V}$ preserves inclusions; in particular, it is one-to-one.*

A class of regular languages over A is said to be an A-*variety* if it is closed under the boolean operations and the taking quotients.

Theorem 2 *If $\Rrightarrow$ is a variety of maps and $\Rrightarrow \to \mathcal{V}$, then $\mathcal{V}$ is an A-variety of languages. Conversely, if $\mathcal{W}$ is and A-variety, then there exists a variety $\Lleftarrow$ of maps such that $\Lleftarrow \to \mathcal{W}$.*

Theorems 1 and 2 hold also if we consider an independence aplhabet (A, I) instead of A and the trace monoid $\vartriangle(A, I)$ instead of the free monoid A^* provided that our maps $\mu : A \to M$ preserve the independence relation I in the following sense: if aIb then $a\mu b\mu = b\mu a\mu$ in the monoid M.

An Efficient Translator from LTL to Automata
Paul Gastin and Denis Oddoux

We present a new algorithm which efficiently constructs a Büchi automaton $\mathcal{A}_\varphi$ accepting exactly the models of an LTL formula φ. The theoretical complexity of this problem is well-known (PSPACE-complete) but it is still a crucial issue in model checking to come with implementations which are very efficient in practice. Several papers attacked this issue recently (CAV'99, CAV'00, CONCUR'00). The aims are to get a fast generation of a Büchi automaton $\mathcal{A}_\varphi$ that is as small as possible.

Most properties of distributed systems are valid only under the assumption of some fairness conditions. For instance, a response property which depends on n fairness conditions may be described by the formula

$$\theta_n = \neg((GFp1 \wedge \ldots \wedge GFp_n) \rightarrow G(q \rightarrow Fr)).$$

The algorithm which is used in the Spin model checker, developped by Holtzmann at Bell Labs, though giving small automata, is not time and memory efficient on such very natural formulas. Our algorithm on the other hand outperforms Spin's algorithm as shown in Table 1.

	Spin		LTL2BA	
θ_1	0.18 s	460 Ko	$<$ 0.01 s	9 Ko
θ_2	4.6 s	4,2 Mo	$<$ 0.01 s	11 Ko
θ_3	170 s	52 Mo	$<$ 0.01 s	19 Ko
θ_4	2 h 40	970 Mo	0.06 s	38 Ko
θ_5			0.37 s	48 Ko
θ_6			4.00 s	88 Ko
θ_7			32 s	175 Ko
θ_8			6 mn	250 Ko
θ_9			50 mn	490 Ko
θ_{10}			10 h	570 Ko

Table 1: Comparison between LTL2BA and SPIN on the formulas θ_n, $n \leq 10$

The existing algorithms for building the Büchi automaton from the LTL formula consist of three phases. First, the formula is rewritten into some canonical form. Second a Büchi automaton is generated. Third, the automaton is simplified in order to get fewer states and fewer transitions.

The main improvement of our algorithm is in the second phase. We first generate an alternating automaton, then a generalized Büchi automaton and finally the classical Büchi automaton. The generation of an alternating automaton from an LTL formula is classical.

Going from the alternating automaton to the generalized Büchi automaton is one of the main innovations. We are using a Büchi automaton with accepting conditions on transitions. An accepting run uses infinitely many 'good' transitions. We use the fact that the alternating automaton $\mathcal{A}$ obtained from an LTL formula is *very weak* to build a generalized Büchi automaton $\mathcal{B}$ with at most 2^n states where n is the number of states of $\mathcal{A}$. Then we construct a classical Büchi automaton $\mathcal{C}$ with at most $(m+1)2^n$ states where m is the number of accepting conditions of $\mathcal{B}$. This is a major improvement on the classical construction which yields at most $2^n \cdot 2^n$ states.

The other major improvement is to simplify all the automata $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ on the fly during their construction. This does not yield fewer states than an *a posteriori* simplification but it reduces impressively both the running time of the algorithm and the memory needed. The simplification basically removes redundant states and redundant transitions.

One may use the algorithm through a web page that can be accessed from `http://www.liafa.jussieu.fr/~gastin/Activites/LTL2BA` or `http://verif.liafa.jussieu.fr/~oddoux`

HMSCs as Specifications with Petri Nets as Completions
Philippe Darondeau

We present ongoing work aiming at understanding the nature of specifications given by High Level Message Sequence Charts and the ways in which they may be put inti effective use. A series of undecidability results on unrestricted HMSCs, following from corresponding results on rational subsets of product monoids, seems to indicate that they should be handled as minimal languages to be approximated from above in any realization. We present an effective closure operation that yilds the least realization of a HMSC language by an (injectively labeled) distributable Petri net, relying on the semilinearity of the commutative images of HMSC languages. Distributable Petri nets may in turn be transformed into equivalent systems of automata that communicate by asynchronous message passing.

Joint work with B.Caillaud, L.Helouet, G.Lesventes.

Program of the Seminar

Monday, November 27$^{\text{th}}$ 2000

09:00 – 09:40	Lenore Zuck, Courant Institute - New York Automatic Deductive Verification with Invisible Invariants
09:50 – 10:30	Doron Peled, Bell Labs - Murray Hill Black Box Checking
11:00 – 11:40	Markus Lohrey, Universität Stuttgart Word Problems for Mazurkiewicz Traces
13:50 – 14:30	Pascal Weil, LaBRI - Bordeaux Recognizability and Regularity of Series-Parallel Pomsets
14:40 – 15:20	Dietrich Kuske, TU Dresden (Un-)Decidable Problems Concerning Asynchronous Cellular Automata
16:00 – 16:40	Remi Morin, TU Dresden On Rational MSC Languages and Relationships with Mazurkiewicz Trace Theory
16:50 – 17:30	Philippe Darondeau, IRISA - CNRS HMSCs as Specifications with Petri Nets as Completions

Tuesday, November 28th 2000

09:00 – 09:35	Philippe Schnoebelen, ENS - Cachan Model Checking of CTL$^+$ and FCTL is Hard
09:40 – 10:15	Natasha Sharygina, Bell Labs - Murray Hill Design for Verification...
10:40 – 11:15	Francois Laroussinie, ENS - Cachan On the Expressivity of Quantitative Branching-Time Temporal Logics
11:20 – 11.55	Zoltàn Ésik, University of Szeged Logic of Fixed Points with Applications to Concurrency
14:10 – 14:45	Madhavan Mukund, Chennai Math. Inst., Madras Regular MSC Languages
14:50 – 15:25	Ingolf Krüger, TU München Distributed System Design with Message Sequence Charts
16:00 – 16:35	Angelika Votintseva, Universität Oldenburg Partial Order Characterization of a Process Algebra with Iteration
20:00	Yuri Matiyasevich, Steklov Inst. Math. St. Petersburg Hilbert's Tenth Problem Today : Main Results and Open Problems

Wednesday, November 29th 2000

09:00 – 09:35	Denis Thérien, McGill University The Power of Algebra in Logic
09:40 – 10:15	Richard Mayr, LIAFA - Université Paris VII Quantitative Aspects of Bisimulation Collapse
10:40 – 11:15	Claude Jard, IRISA Rennes Executing High-Level Message Sequence Charts
11:20 – 11.55	Anca Muscholl, LIAFA - Université Paris VII Compositionality of Message Sequence Charts

Thursday, November 30th 2000

09:00 – 09:35	Wojciech Penczek, Polish Academy of Science Abstractions and Partial Order Reductions for Branching Properties of Time Petri Nets
09:40 – 10:15	Antoine Petit, ENS - Cachan Updatable Timed Automata
10:40 – 11:15	Jacques Sakarovitch, ENST - Paris Derivations of Rational Expressions with Multiplicity
11:20 – 11:55	Manfred Droste, TU Dresden Process Cost Functions for Concurrent Systems
14:10 – 14:45	Mikhail Volkov, Ural State Univ. - Ekatarinenburg An Eilenberg Type Correspondence for Recognizable Trace Languages
14:50 – 15:25	Igor Walukiewicz, University of Warsaw Towards Model-Checking Trace Logics
16:00 – 16:35	Ahmed Bouajjani, LIAFA - Université Paris VII Symbolic Analysis of Infinite-State Systems Using Extended Automata
16:40 – 17:15	Paul Gastin and Denis Oddoux, LIAFA - Université Paris VII An Efficient Translator from LTL to Automata

Friday, December 1st 2000

09:00 – 09:35	David Janin, LaBRI - Bordeaux Bisimulation Invariance in the Levels of the Polynomial Alternation Depth Hierarchy
09:40 – 10:15	Daniel Kirsten, TU Dresden The Star Problem in Trace Monoids: Reductions Beyond C4
10:40 – 11:15	Ingmar Meinecke, TU Dresden From Concrete Domains to Concrete Data Structures
11:20 – 11.55	Volker Diekert, Universität Stuttgart LTL is Expressively Complete for Mazurkiewicz Traces