TEMPORAL LOGICS FOR DISTRIBUTED SYSTEMS

—

PARADIGMS AND ALGORITHMS

EDITORS
Edmund M. Clarke
Ursula Goltz
Peter Niebert
Wojciech Penczek

Contents

1 Preface

Distributed systems, i.e. systems characterised by the concurrent operation and interaction of several components, occur throughout information technology; from microprocessors to computer networks. Since the design and development process of distributed systems is very sensitive to errors, it is an accepted fact in both science and industry that formal approaches to specification and automatic verification and debugging are needed. An important formal framework in this line is the family of temporal logics.

Originally, temporal logics were directed to describe a sequentialised (interleaved) and global view of the behaviour of distributed systems. Several problems resulting from this approach have been identified. On the specification level, the global view of the system makes it difficult or impossible to intuitively specify behavioural aspects of selected parts of the whole system. On the algorithmic front, this sequentialised semantics leads to the well known state explosion problem, which is often the reason for automatic verification to fail and makes it neccessary to develop heuristic workarounds. These problems have led in the past to the following investigations:

- On the one hand, various semantic models capturing aspects of distribution and causality of the behaviour of distributed systems have been developed, in particular partial order models and event structures. On some of these models, several extensions of standard temporal logics with differing modes of expressiveness have been defined. The logics have been investigated under several aspects: Axiomatisations, theoretic and pragmatic expressiveness, complexity of the satisfiability and satisfaction problems.
- On the other hand, research directed towards efficient model checking has focussed on heuristic improvements of model checking algorithms for interleaving logics, which are based on state space exploration. Techniques, which have been evolving in this domain include modular model checking, symbolic model checking (BDDs), partial order reductions, abstraction, and others. Many of these approaches heavily exploit the distributed structure of the system, but do not explicitly rely on a distributed logical framework.

While a lot of research in both directions has happened separately, the natural connection between them has not gone unnoticed: some logics tailored towards distribution allow new verification algorithms, and conversely the heuristics discovered in model checking algorithms influence the design of logics. However, dedicated research is needed in order to achieve practically useful results here.

The goal of this workshop was to bring people from these areas of research together. 35 participants from 12 countries accepted the invitation. They presented their current research in the field of interest in 28 presentations.

The following main topics were addressed:

Partial order logics for linear time. Several talks presented the recent developments on the theory of these logics. Two talks addressed the logic LTrL, which is known to have the same expressive power as the first order logic. Albeit non-elementary complexity, automata constructions have been shown to be possible (and thus open a way to do model checking). Moreover, the theory has been cleaned up and new proofs as well as improved syntax have been presented. Two other talks addressed causality based logics, one giving a separation result and proof technique to distinguish some logics wrt. expressive power, the other introducing an elegant μ-calculus on traces, which is expressively complete. Finally, the theoretic foundations for generalizing trace semantics to systems with state dependent independency of actions were presented.

Partial order logics for branching semantics. Three talks introduced logics based on branching partial order semantics. Two talks addressed an interesting ontology based on intuitions of partial knowledge in a distributed setting, and addressed questions of decidability and model checking. In the third talk, a Petri net oriented logic was proposed.

Proof techniques for distributed systems. In seven talks, most of the spectrum of methods to exploit structural knowledge about systems for proving properties were addressed: Unfolding based model checking for reachability and for linear time, partial order reduction techniques, theorem proving exploiting commutativity of actions, symbolic model checking with and without BDDs, and compositional model checking. The presentations were a fair mix between presentations of established results as well as recent developments.

Probabilistic and real time model checking. Three presentations addressed attempts to apply partial order reductions to timed systems, but with very different approaches, advantages and problems. Two presentations introduced model checking of discrete and continuous stochastic systems. On this line, up to now only symbolic model checking methods with generalisations of BDDs are known.

Verification of Message Passing Systems. Three talks addressed issues of automatic verification in message passing systems. In particular, verification and dedicated logics for message sequence charts were presented.

General aspects of temporal logics. The five presentations in this section are a mix of complexity considerations in search of both simple (yet useful) and very powerful temporal logics, which are of course of importance to the distributed case also.

In addition to the talks, two sessions were used to discuss open problems and controversial issues. Particularly interesting and lively were discussions on

- unfolding methods vs. partial order reductions for model checking,
- usability and pragmatics of logics for practical specification.

Summarizing, the seminar was intense and stimulating. Inspite of the full "official" programme, concentrated work in smaller groups continued into the evenings, using the excellent facilities and working atmosphere of Dagstuhl.

Acknowledgements

Many thanks to the Dagstuhl team and to Thomas Firley for their help and the preparation and organisation of the seminar. On behalf of all the participants, we would also like to thank the staff of Schloß Dagstuhl for providing an excellent environment.

Edmund M. Clarke
Ursula Goltz
Peter Niebert
Wojciech Penczek

2 Workshop Programme

Monday 11

Chair: Ursula Goltz

9.00 Edmund M. Clarke: Symbolic Model Checking without BDDs
9.40 Armin Biere: Combining Local and Global Model Checking

10.20 COFFEE BREAK

Chair: Peter Niebert

10.50 Antoine Petit: A (Non-Elementary) Modular Decision Procedure for LTrL
11.30 Paul Gastin: An Expressively Complete Temporal Logic without Past Tense Operators for Mazurkiewicz Traces

12.10 LUNCH BREAK

Chair: Doron Peled

14.10 Shmuel Katz: Do We Need More Temporal Logic? (A Case Study Using Convenient Computations)
14.50 Peter Niebert: Local First Search

15.30 COFFEE BREAK

Chair: Ed Clarke

16.10 Marta Kwiatkowska: Symbolic Probabilistic Model Checking
16.50 Joost-Pieter Katoen: Model Checking Continuous-Time Markov Chains

---

Tuesday 12

Chair: Bengt Jonsson

9.00 Doron Peled: Specification and Verification of Message Sequence Charts
9.40 Anca Muscholl: On the Complexity of Verification of Message Sequence Graphs

10.20 COFFEE BREAK

Chair: Volker Diekert

10.50 Manfred Droste: An Automaton Model for Concurrent Processes
11.30 Igor Walukiewicz: Local Logics for Traces

12.10 LUNCH BREAK

Chair: Javier Esparza

14.10 Jesper Gulman Henriksen: Can't Temporal Logics over Traces be Separated?
14.50 Tiziana Margaria: On Modelling Feature Interaction in Telecommunications

15.30 COFFEE BREAK

Chair: Shmuel Katz

16.10 DISCUSSION

Wednesday 13

Chair: Bernhard Steffen

9.00 Javier Esparza: Verification with Unfoldings
9.40 Frank Wallner: Model Checking LTL using Net Unfoldings

10.20 COFFEE BREAK

Chair: Marta Kwiatkowska

10.50 David Janin: On the Monadic Complexity of the mu-Calculus
11.30 Philippe Schnoebelen: The Complexity of Temporal Logic in Simple Cases

12.10 LUNCH BREAK

14.00 EXCURSION

Thursday 14

Chair: Manfred Droste

9.00 Bengt Jonsson: Local Time Semantics for Networks of Timed Automata
9.40 Ruurd Kuiper: Partial-Order Reduction Techniques for Real-Time Model Checking

10.30 COFFEE BREAK

Chair: Paul Gastin

10.50 Helmut Veith: Linear Time Datalog for Branching Time Logics
11.30 Mikko Tiusanen: How to Apply Process-Algebraic Algorithms to State-Based (Linear) Temporal Logic Verification

12.15 LUNCH BREAK

Chair: Bengt Jonsson

14.30 DISCUSSION

15.30 COFFEE BREAK

Chair: Antoine Petit

16.10 Bernd-Holger Schlingloff: Verification of Real-Time Systems with Partial Orders
16.50 Madhavan Mukund: Towards a Characterization of Finite-State Message-Passing Systems

---

Friday 15

Chair: Wojciech Penczek

9.00 Julian Bradfield: A Causal Fixpoint Logic
9.40 Sophie Pinchinat: Polyniomial Methods for Verification and Control Synthesis

10.20 COFFEE BREAK

Chair: Anca Muscholl

10.50 Michaela Huhn: Model Checking Distributed Temporal Logics on Net Unfoldings
11.30 Wojciech Penczek: Temporal Approach to Causal Knowledge

12.10 LUNCH

END

3 Collected Abstracts

Combining Local and Global Model Checking

Armin Biere
University of Karlsruhe

The verification process of reactive systems in local model checking and in explicit model checking is on-the-fly. Therefore only those states of a system have to be traversed that are necessary to prove a property. In addition, if the property does not hold, than often only a small subset of the state space has to be traversed to produce a counterexample. Global model checking and, in particular, symbolic model checking can utilize compact representations of the states space, e. g. BDDs, to handle much larger designs than what is possible with local and explicit model checking. We present a new model checking algorithm for LTL that combines both approaches. In essence, it is a generalization of the tableau construnction of Bhat et al., that enables the use of BDDs but still is on-the-fly.

Joint work with Edmund Clarke and Yunshan Zhu.

A Causal Fixpoint Logic

Julian Bradfield
University of Edinburgh

We describe some experiments with expressing causal or true concurrent properties in a logic without a real partial order semantics. Our logic is defined on an extended state space of Petri nets, or any other model with locality. Consequently, for finite safe nets the state space is still finite, unlike partial order or event structure based logics. Surprisingly, this simple idea seems to give strong distinguishing power, and many 'natural' causality properties are expressible.

Joint work with Angelika Mader, Javier Esparza and Michaela Huhn.

Symbolic Model Checking without BDDs

Edmund M. Clarke
CMU – Pittsburgh

Symbolic Model Checking has proven to be a powerful technique for the verification of reactive systems. BDDs have traditionally been used as a symbolic representation of the system. In this paper we show how boolean decision procedures, like Staalmarck's Method or the Davis & Putnam Procedure, can replace BDDs. This new technique avoids the space blow up of BDDs, generates counterexamples much faster, and sometimes speeds up the verification. In addition, it produces counterexamples of minimal length. We introduce a bounded model checking procedure for LTL which reduces model checking to propositional satisfiability. We show that bounded LTL model checking can be done without a tableau construction. We have implemented a model checker BMC, based on bounded model checking, and preliminary results are presented.

Joint work with A. Biere, A. Cimatti and Yunshan Zhu.

An Automaton Model for Concurrent Processes

Manfred Droste
TU Dresden

Automata A with concurrency relations, which occurred in formal verification methods for concurrent programs, are labelled transition systems with a collection of binary relations describing when two actions in a given state of the automaton can occur independently of each other. The concurrency monoid $M(A)$ comprises all finite computation sequences of A modulo a canonical congruence induced by the concurrency relations, with concatenation as monoid operation. Its elements can be represented as labelled partially ordered sets. Under suitable assumptions on A we show that a language L in $M(A)$ is recognizable iff it is definable by a formula of monadic second order logic iff it can be constructed from finite languages using c-rational expressions. We also investigate the relationship between languages which are aperiodic, starfree, first order definable, or definable in some temporal logic. This generalizes various recent results of trace theory.

Joint work with Dietrich Kuske (TU Dresden).

Verification with Unfoldings

Javier Esparza

Technical University of Munich

The talk introduces the unfolding technique for automatic verification of finite state systems. The technique exploits the concurrency of the system to obtain a compact representation of the set of reachable states.

An Expressively Complete Temporal Logic without Past Tense Operators for Mazurkiewicz Traces

Paul Gastin

LIAFA, Université Paris 7

Mazurkiewicz traces are a widely accepted model of concurrent systems. They have been extensively used in the context of model checking in the last years. A major problem is to find temporal logics for traces with the same expressive power as the first order theory FO($<$) of finite (infinite resp.) traces. For finite traces, Ebinger [2] proposed such a temporal logic with both past and future modalities, but the proof failed for infinite traces. Then, Thiagarajan and Walukiewicz [3] have introduced a temporal logic LTrL for traces with the usual future modalities and also past tense modalities in the weak form of *previous* constants. We present here a linear time temporal logic $\mathrm{LTL_f}$ which uses future tense modalities only and is still expressively complete [1]. Contrary to all previous cases, our proof is direct and uses no reduction to words. As a formal consequence Kamp's theorem for both finite and infinite words becomes a corollary. This direct approach became possible due to a new proof technique of Wilke [4] developed for the case of finite words.

Joint work with Volker Diekert (Inst. für Informatik, Universität Stuttgart)

References

[1] V. Diekert and P. Gastin. An expressively complete temporal logic without past tense operators for mazurkiewicz traces. In J. Flum and M. Rodríguez-Artalejo, editors, *Proceedings of the Annual Conference of the European Association for Computer Science Logic (CSL'99)*, number 1683 in Lecture Notes in Computer Science, pages 188–203. Springer, 1999.

[2] W. Ebinger. *Charakterisierung von Sprachklassen unendlicher Spuren durch Logiken.* Dissertation, Institut für Informatik, Universität Stuttgart, 1994.

[3] P. S. Thiagarajan and I. Walukiewicz. An expressively complete linear time temporal logic for Mazurkiewicz traces. In *Proceedings of the 12th Annual IEEE Symposium on Logic in Computer Science (LICS'97)*, 1997.

[4] Th. Wilke. Classifying discrete temporal properties. In Chr. Meinel and S. Tison, editors, *Proceedings of the 16th Annual Symposium on Theoretical Aspects of Computer Science (STACS'99), Trier 1999*, number 1443 in Lecture Notes in Computer Science, pages 32–46, Berlin-Heidelberg-New York, 1999. Springer. Invited Lecture.

Can't Temporal Logics over Traces be Separated?

Jesper Gulmann Henriksen
University of Aarhus

In the past years a multitude of temporal logics for Mazurkiewicz traces has been defined by researchers to express properties of concurrency and causality in a direct fashion. While several elegant and nontrivial expressive completeness proofs with respect to first- and monadic second-order logic are now known, only very trivial separation results for such logics are known. Usually, these results follow by specializing to the degenerate sequential case. We suggest extending the Ehrenfeucht-Fraïssé games for temporal logic of Etessami and Wilke to the richer setting of traces. In this way we obtain a proof principle for separation results not relying on the sequential case. We introduce TLC*, which is a strengthened version of TLC introduced by Alur, Peled, and Penczek, and apply the technique in this setting. More specifically, we show that TLC* adds to the expressive power of TLC. We do so by defining an Ehrenfeucht-Fraïssé game to capture the expressive power of TLC. We then exhibit a property and by means of this game prove that the chosen property is not definable in TLC. We then show that the same property is definable in TLC*. We prove in fact the stronger result that TLC* is expressively stronger than TLC exactly when the dependency relation associated with the underlying trace alphabet is not transitive.

Model Checking Distributed Temporal Logics on Net Unfoldings

Michaela Huhn
TU Braunschweig

Distributed temporal logics allow to specify the behaviour of a system from the local point of view of one component. Representatives of this class of logics are the "temporal logic

for communicating agents" by Lodaya, Ramanujam and Thiagarajan or the distributed μ-calculus. We reduce the model checking problem for such logics to standard model checking algorithms for interleaving logics.

We consider Petri net unfoldings as semantic models and define quotient structures which are multi-modal finite state transitions systems. In case of the distributed μ-calculus the underlying equivalence relation can be defined for the complete logic, for the temporal logic of communicating agents an equivalence that is finer than the discriminating power of the logic would have infinite index, i. e. die quotient structure would remain infinite state. However, for a given formula the number of gossip and past modalities is limited and an equivalence with finite index can be found.

Starting with extensions of the McMillan prefix of a Petri net as the state space, we show how to calculate the transitions for these multi-modal transitions systems, on which then standard model checkers can be applied.

On the Monadic Complexity of the mu-Calculus

David Janin

LaBRI – Bordeaux

We consider the propositional mu-calculus and its relationship with monadic second order logic (MSOL) over graphs.

From J. Bradfield [CONCUR'96] result, we know that the hierarchy induced by the alternation of least and greatest fixpoints in mu-formulae is strict. We also know that the hierarchy induced by the alternation of existential and universal quantifiers in MS-formulae is strict. Then the question we is: how these hierarchies relate?

Given the set $M_0 = N_0$ of properties definable with no fixpoint, given, for each k, the set M_{k+1} (resp. N_{k+1}) of properties defined as the closure of N_k (resp. M_k) under disjunction, conjunction, modalities and least fixpoint (resp. greatest fixpoint) computation, we observe that:

1. $M_k \subseteq$ monadic $\Pi_k \quad (k \geq 0)$,
2. provided quantification over graph edges can be simulated in MSOL without increase of (alternating) complexity, e. g. over trees, over graphs of bounded in or out degree, over graphs of bounded tree-width:
   (a) $M_k \subseteq$ monadic $\Pi_3 \cap$ monadic $\Sigma_3 \quad (k \geq 0)$ this translation being optimal as shown by D. Janin and G. Lenzi in [MFCS'99],
   (b) $M_k \subseteq$ monadic $\Pi_2^c \cap$ monadic $\Sigma_2^c \quad (k \geq 0)$,

where Σ_k^c (resp. Π_k^c) refers to the level of the monadic hierarchy induced by alternation of existential and universal quantifier *regardless of the position of first order quantifiers*.

Conversely, after recalling the result of D. Janin and I. Walukiewicz in [CONCUR'96] that the bisimulation invariant fragment of MSOL exactly coïncides with the mu-calculus, we show that a similar result hold for monadic Σ_2 [unpublished result with. G. Lenzi to be checked], namely, the bisimulation invariant fragment of Σ_2 equals N_2 (which in turn is known to capture properties definable by Buchi tree-automata).

Several chancy conjectures are also stated (available on the http address `http://www.labri.u-bordeaux.fr/~janin/index.html` or on request by e-mail).

Local Time Semantics for Networks of Timed Automata

Bengt Jonsson

University of Uppsala

We present a new semantics for networks of timed automata, which is intended to make such networks amenable to partial order reduction techniques in verification. The main idea is to avoid the implicit clock synchronization between processes in a network, which is caused by the fact that in the standard semantics of networks of timed automata, all clocks in all process advance synchronously. We propose to let local clocks in each process advance independently of clocks in other processes, and by requiring that two processes resynchronize their local time scales whenever they communicate. In this semantics, two transitions are independent if they correspond to disjoint transitions in different processes. Thus we can apply standard partial order reduction techniques to the problem of checking reachability for timed systems, in order to avoid exploration of unnecessary interleavings of independent transitions. The overhead of the semantics is that in verification, we must maintain additional information about the relationship between the local times of different processes. Finally, we present some problems in the semantics that concerns the notion of urgency, for which it appears that synchronization between some clocks (as in the standard semantics) is necessary.

Joint work with Johan Bengtsson, Johan Lilius, and Wang Yi.

Model Checking Continuous-Time Markov Chains

Joost-Pieter Katoen
University of Twente

The verification of continuous-time Markov chains (CTMCs) against continuous stochastic logic (CSL), a stochastic branching-time temporal logic, is considered. CSL facilitates amongst others the specification of steady-state properties and the specification of probabilistic timing properties of the form $\mathcal{P}_{\bowtie p}(\Phi_1 \mathcal{U}^I \Phi_2)$, for state formulas Φ_1 and Φ_2, comparison operator $\bowtie$, probability p, and real interval I. We show how the model checking of these probabilistic timing properties boils down to solving a set of Volterra integral equations. To compute these integrals either numerical integration can be exploited, and this can be performed in a symbolic fashion using a variant of MTBDDs. An alternative method is to reduce the model checking of probabilistic timing properties to the problem of computing transient state probabilities in CTMCs. This approach allows us to verify such properties by using the efficient uniformisation technique for transient analysis of CTMCs. We show that a variant of (ordinary) lumping equivalence, a well-known notion for aggregating CTMCs, preserves all CSL-properties. This facilitates a reduction of the state space while preserving correctness.

Joint work with: Christel Baier (University of Bonn), Holger Hermanns (Twente), Boudewijn Haverkort (RWTH Aachen)

Do We Need More Temporal Logic? (A Case Study Using Convenient Computations)

Shmuel Katz
The Technion, Haifa

The issue is raised of how to best formalize any new idea about how to specify or verify distributed systems, in a temporal logic framework. The options considered are (1) to create a new temporal logic with its own modalities and semantic model, (2) to incorporate the idea into an existing but particular temporal logic framework, or (3) to use a general logic, i. e. higher order predicate logic. The advantages of a new temporal logic are that a clear statement can be derived about the expressive power, semantic comparisons can often be made with other logics, and specially optimized tools (e. g., model checkers) can be derived. On the other hand, there are disadvantages: the idea may then be less accessible, there may be overspecification of aspects not related to the new idea, and the development of tools may require extraneous effort. The convenient computations method of specifying and verifying is used to illustrate. In that method verification is divided into (a) showing that certain convenient computations satisfy the desired properties and (b)

show that every other computation can be reduced to a convenient one on a way that preserves the desired properties. The proof is done using mapping to well-founded sets from th computations. The method was first presented on a new temporal logic, LTL* (in works by Katz and Peled), and much later, without a particular logic, as a proof environment built over PVS (see CAV'99, Ghesmen and Katz). The advantages and disadvantages noted above are shown for this case study. In this case it seems that the logic has some value, but the presentation within PVS may be potentialy more useful.

Partial-Order Reduction Techniques for Real-Time Model Checking.

Ruurd Kuiper

Technische Universiteit Eindhoven

Partial order techniques are used to alleviate state explosion in model checking, successfully so for untimed model checkers. Model-checking itself has been extended to also apply to real-time. The addition of timing information changes the independence relation that forms the cornerstone of the partial-order techniques (a direct treatment of time as an extra variable turns out not to lead to substantial reductions).

Here a generalization of the independence relation called covering, is proposed that aims at a better compatibility with real-time systems. Rather than being based on comparing individual states using a symmetric relation, as with classical independence, covering considered sets of states and uses the asymmetric set inclusion for comparison. Furthermore, covering uses the specific properties of time variables to achieve better reductions. Both a theoretical description of covering as well as a set of criteria that allow implementation in the sense that they are locally checkable are provided.

A extra benefit of the Dagstuhl seminar was that this approach was seen to be closely related to the work of Holger Schlinghof – which I had not realised that was the case before.

Joint work with Dennis Dams, Rots Gerth, Bart Knaack.

Symbolic Probabilistic Model Checking

Marta Kwiatkowska
University of Birmingham

Probabilistic analysis can establish that certain properties hold (in some meaningful probabilistic sense) where conventional model checkers fail, either because the property simply is not true in the state (but holds in that state with some acceptable probability), or because exhaustive search of only a portion of the system is feasible. Models used for such analyses are variants of probabilistic automata (such as labelled Markov chains), in which the usual (boolean) transition relation is replaced with its probabilistic version given in the form of a transition probability matrix. We consider the probabilistic temporal logic PCTL of Hansson & Jonsson, based on CTL, which allows one to express properties such as "the probability of the message being delivered is at least 0.98". Model checking procedure for this logic over fully probabilistic systems can be reduced to solving a linear equation system, and over concurrent probabilistic systems to a linear optimization problem. We report on some experimental results with MTBDD-based symbolic probabilistic model checking using Colorado University Decision Diagram package (CUDD).

On Modelling Feature Interaction in Telecommunications

Tiziana Margaria
Uppsala University (Sweden) and Universität Dortmund (Germany)

Over the past five years the interest in overcoming misbehaviours in telecommunication applications has grown in intensity and scope, as we are quickly evolving towards a deeply service-oriented society with a high degree of heterogeneity. Mastering the behaviour of services in the current technological jungle is therefore a vital, but at the same time a most difficult task. Avoidance, recovery, and prevention of misbehaviours, usually characterized in terms of 'interaction' modalities between services, or functional units thereof called features, all depend on the ability to detect misbehaviours, which in its turn requires a precise characterization of the intended behaviours of services, both in isolation and in a context.

We develop a method for specifying the intent of services which abstracts from and is independent of implementation-specific details. Our approach is based on supporting and organizing the dynamics of 'service evolution' rather than on trying to establish a universal formal setting capturing all the potential future needs. It is based on developing an extensible, predicate-based, temporal application-language for capturing a growing set of requirements and features. New predicates, features (feature specifications) and global requirements can be added at need. This leads to an evolving modelling of the domain. The envisaged result is a "dictionary" of abstract logical concepts, which are defined in terms of each other, and which may be refined as new services and aspects are introduced.

We use standard temporal logic to formulate abstract properties and refinements. We can also connect temporal logic descriptions to more concrete automata level descriptions, meant to reflect the 'small-step' operational behaviour of the service/features. Consistency between automata models and temporal logic specifications can be checked via model checking, using the established connection between temporal logic and automata.

Joint work with Bengt Jonsson, Gustaf Naeser, Jan Nyström, and Bernhard Steffen.

Towards a Characterization of Finite-State Message-Passing Systems

Madhavan Mukund

Chennai Mathematical Institute

A message-passing network consists of a collection of finite-state machines which communicate with each other by sending and receiving messages. Channels are assumed to be reliable but not fifo—delays may cause messages to be received in an order different from the one in which they were sent. Since there is no a priori bound on the number of messages which may be in transit in each channel, the system as a whole is potentially infinite-state.

A robust network is one whose interactions with the environment are insensitive to the relative speeds of local nodes, reordering of messages caused by transmission delays and other sources of nondeterminism. We show that robust networks always admit an equivalent finite-state representation. This means, among other things, that robust message-passing systems are amenable to automated verification.

To establish our claim, we introduce an automaton model for message-passing systems in which each node is a finite-state automaton which has a local input tape. We analyze the behaviour of such systems from a language-theoretic point of view. We show that in our model, every robust message-passing network can be transformed into an equivalent finite-state automaton whose sequential language represents, in a precise sense, the distributed inputs accepted by the original network.

Joint work with K Narayan Kumar (CMI, Chennai), Jaikumar Radhakrishnan (TIFR, Mumbai) and Milind Sohoni (IIT Bombay, Mumbai).

On the Complexity of Verification of Message Sequence Graphs

Anca Muscholl
LIAFA, Université Paris 7

Message sequence charts (MSC) are a graphical specification language used for specifying message exchange between distributed processes. The starting point consists of two decision problems concerning the correctness and the consistency of a design based by MSC graphs. Both problems are shown to be undecidable, in general. Under a natural restriction of the iteration known from Mazurkiewicz trace theory we show both problems to be EXPSPACE-complete. LTL model-checking for this restricted class is PSPACE-complete. The results are based on new complexity results for star-connected rational trace languages.

The results are based on joint work with D. Peled (Bell Labs, USA) and were presented at MFCS'99.

Local First Search

Peter Niebert
Verimag

Partial order reductions are an approved heuristic method to cope with the state explosion problem resulting from building the interleaving transition system in algorithms for the automatic verification of parallel systems. The reductions work by providing sufficient criteria for building only a part of the full transition on which the verification algorithms still compute the correct result. We introduce a new method in this line, with a justification very different from preceeding approaches.

The method requires as prequisite parallel systems with rather local communication (e.g. binary communication, even broadcast, but no multi party rendez vous) and where concern is about equally local (properties of a single or a few processes only). For such systems we show the that the restriction of the system behaviour in such a way that only a logarithmic number of processes are allowed to be active concurrently preserves such local properties. The argument is based on an analysis of the structure of partial order executions of such systems.

On the one hand, this observation can be used to limit the state space used in model checking (proving the correctness of a property), on the other hand it can be used as a strategy to quickly find counter examples (heuristic: Try out behaviour with few active processes first), *local first search*.

Specification and Verification of Message Sequence Charts

Doron Peled
Bell Laboratories

The use of message sequence charts (MSCs) is popular in designing and documenting communication protocols. A recent growth of interest in MSCs has led to various algorithms for automatically analyzing systems of MSCs, e. g., finding race conditions or pattern matching. In this paper we adopt a causality based temporal logic to specify properties of MSCs. This prevents two problems which arise when specifying properties of MSCs using the traditional interleaving-based linear temporal logic: systems of MSCs are not necessarily finite state systems, and even when they are, their set of linearizations can easily generate an exponential state space explosion. We provide an efficient model checking algorithm for systems of MSCs. The model checking environment was implemented as an extension to the Spin model checking system.

Temporal Approach to Causal Knowledge

Wojciech Penczek
Polish Academy of Science

Temporal logic of causal knowledge over general partially ordered structures of local states is defined. The definition of knowledge captures the change of state due to action executions. The structures are a variant of flow event structures including prime event structures and branching processes of Petri Nets. Modalities corresponding to the causality, concurrency, and indistinguishability relations are used. Formulas are interpreted over local state occurrences. The logic is proved to be decidable and a complete axiomatization is provided.

A (Non-Elementary) Modular Decision Procedure for LTrL

Antoine Petit
ENS de Cachan

Thiagarajan and Walukiewicz [1] have defined a temporal logic LTrL on Mazurkiewicz traces, patterned on the famous propositional temporal logic of linear time LTL defined

by Pnueli. They have shown that this logic is equal in expressive power to the first order theory of finite and infinite traces.

The hopes to get an "easy" decision procedure for LTrL, as it is the case for LTL, vanished very recently due to a result of Walukiewicz [2] who showed that the decision procedure for LTrL is non-elementary.

However, tools like Mona [4] or Mosel [3] show that it is possible to handle non-elementary logics on significant examples.

Therefore, it appears worthwhile to have a direct decision procedure for LTrL; in this paper we propose such a decision procedure, in a modular way. Since the logic LTrL is not pure future, our algorithm constructs by induction a finite family of Büchi automata for each LTrL-formula. As expected by the results of [2], the main difficulty comes from the "Until" operator.

Joint work with Paul Gastin (LIAFA, Université Paris 7) and Raphaël Meyer (ENS de Cachan)

References

[1] P. S. Thiagarajan and I. Walukiewicz. An Expressively Complete Linear Time Temporal Logic for Mazurkiewicz Traces. In *Proceedings of LICS'97, LNCS*, 1997.

[2] I. Walukiewicz. Difficult configurations - on the complexity of LTrL. In *Proceedings of ICALP'98, LNCS*, 1998.

[3] P. Kelb and T. Margaria and M. Mendler and C. Gsottberger. Mosel: a flexible toolset for Monadic Second-order Logic. In *Proceedings of CAV'97, LNCS*, 1997.

[4] N. Klarlund. Mona & Fido: The Logic-Automaton Connection in Practice. In *Proceedings of CSL'97, LNCS*, 1998.

Polynomial Methods for Verification and Control Synthesis

Sophie Pinchinat

INRIA – Rennes

In this talk, we present a general framework to perform symbolic mu-calculus model-checking over finite state systems, whenever the systems specification is given in terms of polynomials which coefficients range over a finite fields (of characteristic p). Special instances are BDDs approaches ($p = 2$), but also the SIGALI Tool Box ($p = 3$), which is designed for the multiclocks synchronous equational data-flow language SIGNAL.

Next we explore controller synthesis issues: by means of the logic Alternating-time Temporal Logic (but also the mu-calculus), we show how controller synthesis algorithms,

as made in the SIGALI Tool Box, can be understood. We show how classical control objectives (e. g. invariance, attractivity, persistence, ...) can be expressed in this framework, and for thoses cases, we illustrate how the Supervisory Control Problem decision reduces to model-checking a formula.

Verification of Real-Time Systems with Partial Orders

Bernd-Holger Schlingloff
TZI Universität Bremen

In the talk I presented joint work with Tomohiro Yoneda, Tokyo, on the verification of real-time properties of asynchronous circuits. Our methods are based on the reachability analysis of time Petri nets. In order to improve the efficiency of the verification, we define a partial order search strategy which effectively reduces the number of regions reached in the search. In the first approach to verification, we define a temporal logic TNL over execution sequences for specifying properties of asynchronous circuits. The peculiarity of our logic is that it only refers to differences in the timing of event occurrences. We show that model checking for this logic can be reduced to ordinary LTL model checking. In the second approach, we extend Dills notion of conformance between modules, where both specification and implementation are given as time Petri nets. This gives a hierarchical method for verification. We present various notions of failures, each of which gives rise to a different notion of correctness. The relevant publications can be found at `http://www.tzi.org/~hs/Publikationen`.

The Complexity of Temporal Logic in Simple Cases

Philippe Schnoebelen
ENS de Cachan

We argue that the classical results on complexity of model checking must be refined, strengthened and deepened. With this goal in mind, Demri and I looked for fragments of LTL where model checking is not PSPACE-hard. (The full results will appear in Inf. & Comp., and partial results can be found in Proc. STACS'98, LNCS 1373.)

In this talk we focus on one threshold result with a nice and simple proof: model-checking LTL formulas with only Untils (no Next) and modal depth 2 is already PSPACE-hard. The proof even applies to flat-Untils (D. Dams, Logic J. of the IGPL 7(1), 1999) that is, Untils where the left-hand sides only contain atomic propositions.

How to Apply Process-Algebraic Algorithms to State-Based (Linear) Temporal Logic Verification

Mikko Tiusanen

Tampere University of Technology

We consider how some process-algebraic concepts of equivalence, congruence, and preorder of systems can be used to bring compositionality to verifying temporal logic properties. We will specifically consider two congruences and their related preorders that have been shown to be *weakest* (least restrictive) while preserving relevant properties to be verified: NDFD- and CFFD-equivalences (Non-Divergent Failures Divergences and Chaos-Free Failures Divergences). As a particular problem we will consider ways of handling state-based logical formulae in a process algebra, inherently an action-based formalism. The effect of these considerations is shown at work in analysing a demand-driven token-ring mutual exclusion protocol. The analysis is finite but gives results that are valid for rings of arbitrary length.

Joint work with Antti Valmari.

Linear Time Datalog for Branching Time Logics

Helmut Veith

TU Wien/CMU

We introduce Datalog Lite, a variant of Datalog with linear time model checking. The expressive power of Datalog Lite is characterized by the alternation-free fragment of the guarded fixpoint logic μGF. In addition, fragments of Datalog Lite are shown to capture

ML, CTL, alternation-free μ-calculus, and the guarded fragment GF. Moreover, Datalog Lite can be viewed as a fragment of well-founded Datalog.

Joint work with Georg Gottlieb (TU Wien) and Erich Grödel (RWTH Aachen)

Model Checking LTL using Net Unfoldings

Frank Wallner
Technische Universität München

Net unfoldings provide a nice partial order semantics for Petri nets. In this talk, we indicate how to use the *finite prefix* of an unfolding, introduced by McMillan, for model checking linear-time temporal properties. The method is an adaptation of the "automata-theoretic approach" to model checking:

1. construct a Büchi-automaton for the negation of the property
2. define an adequate product of the system (given as a safe Petri net) and this automaton
3. check if this product (Büchi net) is empty; if so, the property holds for the system.

We propose an algorithm for solving 3. efficiently. (And how 2. has to be done such that the solution is valid.) The basic idea is to extract a finite graph from the prefix, with some, hopefully few, configurations of the prefix as set of nodes. An edge is included from configuration c to another one c' iff the global state represented by c' is reachable from the global state represented by c.

Precisely, we propose to begin with the *local configurations* of the cutoff events of the prefix, to compute the (partial) reachability relation among them as far as possible, and to check if this graph is accepting any word. If so, the corresponding counterexample is returned, else successively new (global) configurations are added to the graph, in order to complete the partial reachability relation of the so far constructed partial graph step by step, until either a counterexample is detected, or the reachability relation is guaranteed to be complete.

Local Logics for Traces

Igor Walukiewicz
Warsaw University

Dependence graphs are transition systems representing traces. A temporal logic over dependence graphs is called local if the meaning of a formula is a set of nodes in a dependence graph. A logic is called global if the meaning of a formula is a set of configurations (finite downwards closed subsets of nodes). Because dependence graphs are just transition systems it is possible to evaluate mu-calculus formulas over them. In this setting mu-calculus can be considered as a local logic for traces.

It turns out that plain mu-calculus is not able to express even some first order properties of dependence graphs. We suggest several extensions of the mu-calculus. Among them is an extension with $\langle \mathrm{co}(b) \rangle$true formulas which meaning is that there is a concurrent event labelled by b. We show that this extension captures the power of monadic second order logic on dependence graphs.
