I N F O R M A T I K

Symposium on the Effectiveness of Logic in Computer Science in Honour of Moshe Vardi

Witold Charatonik   Harald Ganzinger
(Eds.)

MPI–I–2002–2–007   March 2002

FORSCHUNGSBERICHT   RESEARCH REPORT

M A X - P L A N C K - I N S T I T U T
F Ü R
I N F O R M A T I K

Stuhlsatzenhausweg 85   66123 Saarbrücken   Germany

Author's Address

Max-Planck-Institut für Informatik
Stuhlsatzenhausweg 85
66123 Saarbrücken
Germany

Abstract

This report contains abstracts of the talks given at the Symposium on the Effectiveness of Logic in Computer Science in Saarbrücken on March 4-6, 2002. The symposium was held in honour of Professor Moshe Vardi, Rice University, on the ocasion of awarding him a Honorary Doctoral Degree from the Saarland University for his outstanding contributions to the area of Logic in Computer Science.

Invited Talks

On the Unusual Effectiveness of Logic in Computer Science

Moshe Y. Vardi
Rice University

Abstract: During the past twenty five years there has been extensive, continuous, and growing interaction between logic and computer science. In fact, logic has been called "the calculus of computer science". The argument is that logic plays a fundamental role in computer science, similar to that played by calculus in the physical sciences and traditional engineering disciplines. Indeed, logic plays an important role in areas of computer science as disparate as architecture (logic gates), software engineering (specification and verification), programming languages (semantics, logic programming), databases (relational algebra and SQL), artificial intelligence (automatic theorem proving), algorithms (complexity and expressiveness), and theory of computation (general notions of computability). This talk will provide an overview of the unusual effectiveness of logic in computer science by surveying the history of logic in computer science and presenting some of the areas in which logic played a crucial role.

Model Checking Games for Fixed Point Logics

Erich Graedel
RWTH Aachen

Abstract: Model checking problems ("is a given formula true in a given structure?") for almost every logic can be cast as strategy problems ("which player does have a winning strategy in a given game?") for the appropriate evaluation games (also called Hintikka games).

In first-order games all plays are finite and the strategy problem can be solved in linear time in the size of the game. For fixed point logics, the appropriate evaluation games are parity games, which admit also infinite plays. Each position is assigned a priority, and the winner of an infinite play is determined according to whether the least priority seen infinitely often during the play is even or odd. It is open whether winning sets and winning strategies for parity games can be computed in polynomial time. The best algorithms known today are polynomial in the size of the game graph, but exponential with respect to the number of priorities.

An intuitive reason for the good model checking properties of certain logics is that the associated evaluation games remain small. For instance, the restricted quantification in modal or guarded logics limits the number of possible moves in the evaluation games and thus leads to smaller game graphs. Modal and guarded fixed point logics admit efficient model checking if and only if parity games can be efficiently solved. While we do not know whether this is possible in general, we can analyze the structure of parity games and isolate 'easy' cases that admit efficient solutions. With this analysis, we also try to make precise some of the game theoretic intuitions that underly algorithmic approaches to automata and model checking problems. We link these 'easy games' to logic and thus obtain efficient model checking algorithms for fragments of fixed point logic.

Causes and Explanations: A Structural-Model Approach

Joseph Halpern
Cornell University
(joint work with Judea Pearl)

Abstract: What does it mean that an event C "actually caused" event E? The problem of defining actual causation goes beyond mere philosophical speculation. For example, in many legal arguments, it is precisely what needs to be established in order to determine responsibility. (What exactly was the actual cause of the car accident or the medical problem?) Actual causation is also important in artificial intelligence applications. Whenever we undertake to explain a set of events that unfold in a specific scenario, the explanation produced must acknowledge the actual cause of those events.

The philosophy literature has been struggling with the problem of defining causality since the days of Hume, in the 1700s. Many of the definitions have been couched in terms of counterfactuals. (C is a cause of E if, had C not happened, then E would not have happened.) However, all the previous definitions have been shown (typically by example) to be problematic. We propose here new definitions of actual cause and (causal) explanation, using Pearl's notion of structural equations to model counterfactuals. We show that these definitions yield a plausible and elegant account of causation and explanation that handles well examples which have caused problems for other definitions and resolve major difficulties in the traditional account.

Representing and Querying XML with Incomplete Information

Serge Abiteboul
INRIA-Rocquencourt
(joint work with Luc Segoufin and Victor Vianu)

Abstract: We study the representation and querying of XML with incomplete information. We consider a simple model for XML data and their DTDs, a very simple query language, and a representation system for incomplete information in the spirit of the representations systems developed by Imielinski and Lipski for relational databases. In the scenario we consider, the incomplete information about an XML document is continuously enriched by successive queries to the document. We show that our representation system can represent partial information about the source document acquired by successive queries, and that it can be used to intelligently answer new queries. We also consider the impact on complexity of enriching our representation system or query language with additional features. The results suggest that our approach achieves a practically appealing balance between expressiveness and tractability. The research presented here was motivated by the Xyleme project at INRIA, whose objective it to develop a data warehouse for Web XML documents.

Hypertree Decompositions

Georg Gottlob
Vienna University of Technology
(joint work with N. Leone, F. Scarcello and R. Pichler)

Abstract: Many NP-hard problems become polynomially soblvable if restricted to instances whose underlying graph structure has bounded treewidth. The notion of treewidth can be straightforwardly extended to hypergraphs by simply considering the treewidth of their primal graphs or, alteratively, of their incidence graphs. However, doing so comes along with a loss of information on the structure of a hypergraph with the effect that many polynomially solvable problems cannot be recognized as such because the treewidth of the underlying hypergraphs is unbounded. In particular, the treewidth of the class of acyclic hypergraphs is unbounded. In this talk, I will describe and discuss a decomposition method for hypergraphs called hypertree-decompositions. In particular, I will present in form of a survey the following topics and results: 1.) queries and constraint satisfaction problems of bounded hypertree width can be solved in polynomial time Hypergraphs of bounded hypertree width can be recognized in polynomial time; 2.) a hypertree decomposition of such a hypergraph graph can be computed in polynomial time; 3.) hypertree decompositions generalize other notions of hypergraph decompositions; 4.) the hypertree width of a graph has anice game-theoretic characterization 5.) Queries of bounded hypertree width correspond to an interesting fragment of First Order Logic 6.) hypertree width generalizes clique width (of the incidence graph of a hypergraph).

Handling Real Arithmetic with Infinite Word Automata

Pierre Wolper
Université de Liège

Abstract: It has been known for a long time that integer linear arithmetic can be handled using finite automata. That linear arithmetic over the reals and integers can be similarly handled with automata on infinite words is quite obvious, but had never been exploited due to the difficult and delicate to implement algorithms that are involved. This talk focuses on the problem of turning finite automata into a practical tool for handling real and integer linear arithmetic. It shows, using topological arguments, that only a restricted class of automata on infinite words are necessary for this purpose, and discusses related algorithmic results that have made an effective implemented system possible.

Reflections on Finite Model Theory

Phokion Kolaitis
University of California, Santa Cruz

Abstract: Finite model theory can be succinctly described as the study of logics on classes of finite structures. It is an area of research in the interface between logic, combinatorics, and computational complexity that has been steadily developing during the past twenty five years. In this talk, we trace the early origins of finite model theory, highlight some of the main results in this area, and conclude with certain challenging open problems.

Smart Play-Out: Executing Requirements with the Aid of Model Checking

David Harel
The Weizmann Institute of Science

Abstract: We describe a methodology for executing scenario-based requirements of reactive systems, focusing on "playing-out" the behavior using formal verification techniques for driving the execution. The approach appears to be useful in many stages in the development of reactive systems, and might also pave the way to systems that are constructed directly from their requirements, without the need for inter-object or inter-component modeling or coding.

A Comparison of Automata and Tableau Methods for Modal Satisfiability

Franz Baader
RWTH Aachen

Abstract: Tableaux-based decision procedures for satisfiability of modal and description logics behave quite well in practice, but it is sometimes hard to obtain exact worst-case complexity results using these approaches, especially for EXPTIME-complete logics. In contrast, automata-based approaches often yield algorithms for which optimal worst-case complexity can easily be proved. However, the algorithms obtained this way are usually not only worst-case, but also best-case exponential: they first construct an automaton that is always exponential in the size of the input, and then apply the (polynomial) emptiness test to this large automaton. To overcome this problem, one must try to construct the automaton "on-the-fly" while performing the emptiness test.

In this talk, I will first review both the tableaux-method and the automata method for obtaining satisfiability procedures in modal logics, and give an intuitive comparison of both. As a more technical result, I will show that Voronkov's inverse method for the modal logic K can be seen as an on-the-fly realization of the emptiness test done by the automata approach for K. The benefits of this result are two-fold. First, it shows that Voronkov's implementation of the inverse method, which behaves quite well in practice, is an optimized on-the-fly implementation of the automata-based satisfiability procedure for K. Second, it can be used to give a simpler proof of the fact that Voronkov's optimizations do not destroy completeness of the procedure. We will also show that the inverse method can easily be extended to handle global axioms, and that the correspondence to the automata approach still holds in this setting. In particular, the inverse method yields an EXPTIME-algorithm for satisfiability in K w.r.t. global axioms.

View-Based Query Processing

Maurizio Lenzerini
Università di Roma "La Sapienza"

Abstract: View-based query processing is the problem of computing the answer to a query based on a set of materialized views, rather than on the raw data in the database. It represents an abstraction for various data management problems arising in several contexts, including query optimization, query answering with incomplete information, data warehousing, and data integration. There are two approaches to view-based query processing, called query rewriting and query answering, respectively. In the former approach, we are given a query Q and a set of view definitions, and the goal is to reformulate the query into an expression that refers only to the views, and provides the answer to Q. In the latter approach, besides Q and the view definitions, we are also given the extensions of the views, and the goal is to compute the set of tuples t such that the knowledge on the view extensions logically implies that t is an answer to Q. In this talk, we address view-based query processing in the context of semistructured data. The main difficulty arising in this context is that languages for querying semistructured data enable expressing regular-path queries, and therefore may contain a restricted form of recursion.

Easier Ways to Win Logical Games

Ronald Fagin
IBM Almaden Research Center

Abstract: The "computational complexity" of a problem is the amount of resources, such as time or space, required by a machine that solves the problem. The "descriptive complexity" of a problem is the complexity of describing the problem in some logical formalism. There is an intimate connection between the descriptive complexity and the computational complexity. This connection was first discovered by the speaker, who showed that the complexity class NP coincides with the class of properties of finite structures expressible in existential second-order logic (where we are allowed to existentially quantify over not just points, as in first-order logic, but also over relations). The equivalence of questions in computational and descriptive complexity holds the promise that techniques from one domain can be brought to bear on questions in the other domain.

Essentially the only known technique for proving "hardness" (that is, inexpressibility) results in descriptive complexity is to make use of games on graphs played between two players. The purpose of this talk is to discuss some results and techniques that assist in the use of these games, and in particular that make the task of proving inexpressibility results easier. Thereby, we can prove new and deeper inexpressibility results. Our hope is that we can develop such a powerful toolkit that we can eventually make a serious assault on such fundamental problems as the question of whether NP = co-NP.

Logic as a Query Language: from Frege to XML

Victor Vianu
University of California, San Diego

Abstract: The database area is an important area of computer science concerned with storing, querying and updating large amounts of data. Logic and databases have been intimately connected since the birth of database systems in the early 1970's. Indeed, first-order logic lies at the core of modern database systems, and the standard query languages such as SQL and QBE are syntactic variants of FO. More powerful query languages are based on extensions of FO with recursion, and are reminiscent of the well-known fixpoint queries studied in finite-model theory. The impact of logic on databases is one of the most striking examples of the effectiveness of logic in computer science. This talk will discuss some of the reasons for the success of logic in the database area, from relational systems all the way to XML.

The Early History of Database Theory

Jeff Ullman
Stanford University

Abstract: We review the most important ideas that came out of the theoretical study of databases, including the theory of functional dependencies and normalization, generalized dependency theory, acyclic hypergraphs, universal relations, conjunctive queries, Datalog, and recursive queries. We conclude with some observations on the applications of this venerable theory.

Verifying Liveness Properties of Parameterized Systems

Amir Pnueli
The Weizmann Institute of Science

Reachability problems over automaton-definable graphs

Wolfgang Thomas
RWTH Aachen

Abstract: The subject of this lecture is the solvability of some fundamental algorithmic problems over finitely presented infinite graphs. We consider presentations of graphs in terms of finite automata (where vertex properties are defined by finite-state acceptors, and edge relations by different types of finite-state transducers). We focus on the reachability problem and the alternating reachability problem, and report on recent progress regarding decidability results. Starting from this, more general questions on verification and synthesis of state-based systems become accessible. In a wider perspective, such automata theoretic studies lead to an algorithmic model theory of infinite structures.

Synthesis of Reactive Systems

Orna Kupferman
Hebrew University
(joint work with Moshe Vardi)

Abstract: In system synthesis, we transform a specification into a system that is guaranteed to satisfy the specification. The talk will survey the development of synthesis algorithms. We will start with early work on synthesis of closed systems (which do not interact with an environment, making the synthesis problem similar to the satisfiability problem) and will reach synthesis of open systems (which interact with an environment, making the synthesis problem similar to the problem of generating winning strategies in two-player games), synthesis of open systems with incomplete information (where the system cannot read all the signals that the environment generates), and synthesis of distributed systems.

The Complexity of First-Order and Monadic Second-Order Logic Revisited

Martin Grohe
University of Edinburgh
(joint work with Markus Frick)

Abstract: The model-checking problem for a logic L on a class C of structures asks whether a given L-sentence holds in a given structure in C. We give super-exponential lower bounds for fixed-parameter tractable model-checking problems.

We show that unless PTIME = NP, the model-checking problem for monadic second-order logic on finite words is not solvable in time f(k)p(n), for any elementary function f and any polynomial p. Here k denotes the size of the input sentence and n the size of the input word. We prove the same result for first-order logic under a stronger complexity theoretic assumption from parameterized complexity theory.

Furthermore, we prove that the model-checking problem for first-order logic on structures of degree 2 is not solvable in time $2^{2^{o(k)}} p(n)$, for any polynomial p, again under an assumption from parameterized complexity theory. We match this lower bound by a corresponding upper bound.

Logic and Automata: A Match Made in Heaven

Moshe Y. Vardi
Rice University

Abstract: One of the most fundamental results in mathematical logic is the Buchi-Elgot-Trakhtenbrot Theorem, established in the 1960s, which states that finite-state automata and monadic second-order logic (interpreted over finite words) have the same expressive power, and that the transformations from formulas to automata and vice versa are effective. In this talk, I'll survey the evolution of this beautiful connection and show how provides an algorithmic tool set in the area of model checking.

Poster Presentations