08171 Abstracts Collection
Beyond the Finite: New Challenges in Verification and Semistructured Data
— Dagstuhl Seminar —

Anca Muscholl[1], Ramaswamy Ramanujam[2], Michaël Rusinowitch[3], Thomas Schwentick[4] and Victor Vianu[5]

[1] LaBRI - Bordeaux, F
anca@labri.fr
[2] IMSc - Chennai, IND
jam@imsc.res.in
[3] INRIA Lorraine, F
Michael.Rusinowitch@loria.fr
[4] Universität Dortmund, D
thomas.schwentick@udo.edu
[5] UC San Diego, US
vianu@cs.ucsd.edu

Abstract. From 20.04. to 25.04.2008, the Dagstuhl Seminar 08171 "Beyond the Finite: New Challenges in Verification and Semistructured Data" was held in the International Conference and Research Center (IBFI), Schloss Dagstuhl. During the seminar, several participants presented their current research, and ongoing work and open problems were discussed. Abstracts of the presentations given during the seminar as well as abstracts of seminar results and ideas are put together in this paper. The first section describes the seminar topics and goals in general. Links to extended abstracts or full papers are provided, if available.

Keywords. Infinite state systems, data values, verification, semistructured data

Summary

Exploring the interaction of model checking and database static analysis techniques in the development of novel approaches to the verification of software systems handling data.

Keywords: Infinite state systems, data values, verification, semistructured data

Joint work of: Muscholl, Anca; Ramanujam, Ramaswamy; Rusinowitch, Michaël; Schwentick, Thomas; Vianu, Victor

Extended Abstract: http://drops.dagstuhl.de/opus/volltexte/2008/1558

Static Analysis of Active XML Systems

Serge Abiteboul (INRIA SaclaY, FR)

Active XML is a high-level specification language tailored to data-intensive, distributed, dynamic Web services. Active XML is based on XML documents with embedded function calls. The state of a document evolves depending on the result of internal function calls (local computations) or external ones (interactions with users or other services). Function calls return documents that may be active, so may activate new subtasks. The focus of the talk is on the verification of temporal properties of runs of Active XML systems, specified in a tree-pattern based temporal logic, Tree-LTL, that allows expressing a rich class of semantic properties of the application.

Keywords: XML, services, verification, workflow, active xml

Joint work of: Abiteboul, Serge; Segoufin, Luc; Vianu, Victor

See also: ACM PODS 2008

Shape Analysis via Monotonic Abstraction

Parosh Aziz Abdulla (Uppsala University, SE)

Several classes of programs can be modelled as infinite transition systems where the transition relation is monotonic with respect to a certain natural ordering on the set of states. Examples include Petri nets, lossy channel systems, timed networks, multiset rewriting systems, broadcast protocols, etc. On the other hand, there are also classes of programs whose behaviours do not satisfy the monotonicity condition. In this talk, we present the notion of "monotonic abstractions" which transforma a non-monotonic transition system into a monotonic one through an over-approximation. This enables us to adapt methods developed for the above models for verifying non-monotonic systems.

We illustrate the method through an application to shape analysis of programs manipulating singly linked lists.

Keywords: Model checking, shape analysis, monotonic abstraction

Shape Analysis via Monotonic Abstraction

Parosh Aziz Abdulla (Uppsala University, SE)

We propose a new formalism for reasoning about dynamic memory heaps, using monotonic abstraction and symbolic backward reachability analysis. We represent the heaps as graphs, and introduce an ordering on these graphs. This enables us to represent the violation of a given safety property as the reachability of a finitely representable set of bad graphs. We also describe how to symbolically compute the reachable states in the transition system induced by a program.

Keywords: Shape analysis, Program verification, Static analysis

Joint work of: Aziz Abdulla, Parosh; Bouajjani, Ahmed ; Cederberg, Jonathan; Haziza, Frédéric; Ji, Ran; Rezine, Ahmed

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2008/1559

Conjunctive Query Containment over Trees

Henrik Björklund (TU Dortmund, DE)

We study the containment, satisfiability, and validity problems for conjunctive queries over trees with respect to a schema. We show that conjunctive query containment and validity are 2exptime-complete w.r.t. a schema (DTD or Relax NG). Furthermore, we consider conjunctive queries that can test for equalities and inequalities of data values. Here, satisfiability and validity are decidable, but containment is undecidable, even without schema information. On the other hand, containment w.r.t. a schema becomes decidable again if the "larger" query is not allowed to use both equalities and inequalities.

Joint work of: Björklund, Henrik; Martens, Wim; Schwentick, Thomas

On the Sets of Real Numbers Recognized by Finite Automata in Multiple Bases

Bernard Boigelot (University of Liège, BE)

This work studies the expressive power of finite automata recognizing sets of real numbers encoded in positional notation. We consider Muller automata as well as the restricted class of weak deterministic automata, used as symbolic set representations in actual applications. In previous work, it has been established that the sets of numbers that are recognizable by weak deterministic automata in two bases that do not share the same set of prime factors are exactly those that are definable in the first order additive theory of real and integer numbers (R, Z, $+$, $<$). This result extends Cobham's theorem, which characterizes the sets of integer numbers that are recognizable by finite automata in multiple bases.

In this work, we first generalize this result to multiplicatively independent bases, which brings it closer to the original statement of Cobham's theorem. Then, we study the sets of reals recognizable by Muller automata in two bases. We show with a counterexample that, in this setting, Cobham's theorem does not generalize to multiplicatively independent bases. Finally, we prove that the sets of reals that are recognizable by Muller automata in two bases that do not share the same set of prime factors are exactly those definable in (R, Z, $+$, $<$).

These sets are thus also recognizable by weak deterministic automata. This result leads to a precise characterization of the sets of real numbers that are recognizable in multiple bases, and provides a theoretical justification to the use of weak automata as symbolic representations of sets.

Keywords: Automata, mixed integer-real arithmetic, Cobham's theorem

Joint work of: Boigelot, Bernard; Brusten, Julien; Bruyère, Véronique

XPath with data

Mikolaj Bojanczyk (University of Warsaw, PL)

I will talk about XPath queries on XML documents. The "data" in question is attribute values in document nodes. Since there is no limit on the text that can appear with attribute values, the document cannot be modeled as tree over a fixed alphabet.

The focus of the talk is on how Simon's factorization forests can be used to handle such documents. A first example is a linear time algorithm for XPath evaluation. The algorithm uses factorization forests to speed up calculations. A second example is an automaton model for trees with data. The factorization forests are used to show that this model captures all XPath queries.

Keywords: Automata XPath algebra

Full Paper:
http://www.mimuw.edu.pl/∼bojan/papers/linearXPath.pdf

An Introduction to Timed Systems

Patricia Bouyer (ENS - Cachan / CNRS - LSV, FR)

In this talk, we present some of the technics that are used to prove the decidability of classes of real-time systems. We start by presenting the classical finite region automaton abstraction, and then go to more recent developments, where an infinite-state well-structured abstraction is proposed.

Keywords: Timed systems

Automatically Refining Abstract Interpretations for Program Analysis

Supratik Chakraborty (IIT - Bombay, IN)

Abstract interpretation techniques prove properties of programs by computing abstract fixpoints. All such analyses suffer from the possibility of false errors.

We present a few techniques to automatically refine such abstract interpretations to reduce false errors. First, we show how precision loss due to *widening* can be recovered by using information from *interpolants*. This gives rise to a new operator called *interpolated widen*. Next, we show how disjunctions that arise due to refinement can be handled without using operations on powerset domains. Finally, we discuss how refinement can be done for abstract analyses that use the join operator to merge abstract states at program join points. This leads to an algorithm that performs abstraction refinement using DAGs instead of trees. All the above techniques have been implemented in a tool for intraprocedural analysis of C programs called DAGGER. We show by means of extensive experiments that DAGGER is able to prove properties of C programs that several other abstraction-refinement tools like SLAM, BLAST, ARMC etc. are unable to analyze.

Keywords: Automatic abstraction refinement, program analysis, interpolation, widening, disjunctive invariants, DAG based refinement

Joint work of: Gulavani, Bhargav; Chakraborty, Supratik; Nori, Aditya; Rajamani, Sriram

Full Paper:
http://www.cse.iitb.ac.in/ supratik/publications/papers/TR-TACAS08.ps.gz

See also: Bhargav S. Gulavani, Supratik Chakraborty, Aditya V. Nori, Sriram K. Rajamani, Automatically Refining Abstract Interpretations, Proceedings of TACAS 2008, pp. 443-458

Conflict-Tolerant Features

Deepak D'Souza (IMSc - Chennai, IN)

We consider systems composed of a base system with multiple "features" or "controllers", each of which independently advise the system on how to react to input events so as to conform to their individual specifications. We propose a methodology for developing such systems in a way that guarantees the "maximal" use of each feature. The methodology is based on the notion of "conflict-tolerant" features that are designed to continue offering advice even when their advice has been overridden in the past. We give a simple priority-based composition scheme for such features, which ensures that each feature is maximally utilized. We also provide a formal framework for specifying, verifying, and synthesizing such features. In particular we obtain a compositional technique for verifying systems developed in this framework. This framework is also extended to a real-time framework based on Alur-Dill timed automata.

Keywords: Feature interaction, controller synthesis, verification

Joint work of: D'Souza, Deepak; Gopinathan, Madhu

Data Tree Patterns

Claire David (LIAFA - Université Paris VII, FR)

In this work we study a formalism to specify XML-documents that can deal with data values of documents. We consider data tree as model for XML document and we consider regular constraints and Boolean combinations of data tree patterns.

Data tree patterns are tree patterns with descendant or child relation between nodes and possibly label constraints and data (in)equality constraints between nodes. Data tree patterns are a simple and natural formalism for expressing properties of XML documents.

First, we consider the model-checking problem. We show that it is DP-complete in general and already NP-complete for only one very simple pattern.

Then, we consider the satisfiability problem in the presence of DTD (regular constraint on the tree). We show that this problem is general undecidable for the general fragment. We prove decidability and exact complexity result for positive fragment, boolean combination of pattern that allow only child or only descendant axis and for the general fragment but when we restrict models to bounded depth data tree.

The results can be extend when allowing order on data value or if we consider data tree where each node carries several data values.

When considering data word instead of data tree we have same decidability results and few differences for complexity of satisfiability problem.

If we try to extend pattern allowing next sibling axis, satisfiability problem becomes undecidable for any fragment allowing negation of patterns (because we can encode words into it).

Keywords: XML , Tree Pattern, Model-Checking, satisfiability

Safely composing security protocols via tagging

Stéphanie Delaune (ENS - Cachan, FR)

Security protocols are small programs that are executed in hostile environments. Many results and tools, especially for a bounded number of sessions, have been developed to formally analyze the security of a protocol.

However, even when a protocol has been proved secure for a bounded number of sessions, there is absolutely no guarantee that it is really: an attack may involve one extra session. Moreover, if we are able to prove the security of the protocol for an unbounded number of sessions, the interactions with some other protocols are still not taken into account and they may dramatically damage the security of the former protocol.

We will see that these problems do not occur when the protocols are well-tagged.

Keywords: Security protocols, composition, verification

Model checking memoryful linear-time logics over one-counter automata

Stéphane Demri (ENS - Cachan, FR)

We study complexity issues related to the model-checking problem for LTL with registers (a.k.a. freeze LTL) over one-counter automata. We consider several classes of one-counter automata (mainly deterministic vs. nondeterministic) and several syntactic fragments (restriction on the number of registers and on the use of propositional variables for control locations). The logic has the ability to store a counter value and to test it later against the current counter value.

We show that model checking LTL with registers over deterministic one-counter automata is PSPACE-complete with infinite accepting runs. By constrast, we prove that model checking LTL with registers over nondeterministic one-counter automata is undecidable in the infinitary and finitary cases even if only one register is used and with no propositional variable. This makes a difference with the facts that several verification problems for one-counter automata are known to be decidable with relatively low complexity, and that finitary satisfiability for LTL with a unique register is decidable.

Finally, we explain how these results can be adapted to similar problems for its sister logic, first-order logic with data equality test.

This is a joint work with Ranko Lazic (University of Warwick) and Arnaud Sangnier (LSV, ENS Cachan)

Joint work of: Demri, Stéphane; Lazic, Ranko; Sangnier, Arnaud

SDSIRep : A reputation system based on SDSI

Javier Esparza (TU München, DE)

We introduce SDSIRep (pronounced sodsi-rep), a reputation system based on the SPKI/SDSI authorization system. It is well-known that a system of SPKI/SDSI certificates corresponds to the formal model of a pushdown system (PDS).

SDSIRep allows principals to express trust and recommendations in the form of so-called certificates with weights, and it corresponds to probabilistic pushdown systems. By interpreting weights as probabilities, we obtain a random-walk model of the reputation of a principal, and an algorithm that computes it. An extension of SDSIRep also provides for so-called intersection certificates, by which, loosely speaking, a principal gains reputation if recommended by all members of a given group of principals. On a formal-methods level, this extension makes SDSIRep correspond to probabilistic alternating PDSs, and we extend the underlying theory of PDSs to handle this case. As an example we sketch a small academic reputation system that combines information from different reputation sources, like conferences, coauthors, and rankings.

Keywords: Reputation, probabilistic systems, pushdown systems

Closure of Hedge-Automata Languages by Hedge Rewriting

Florent Jacquemard (ENS - Cachan, FR)

We consider rewriting systems for unranked ordered terms, i.e. trees where the number of successors of a node is not determined by its label, and is not a priori bounded. The rewriting systems are defined such that variables in the rewrite rules can be substituted by hedges (sequences of terms) instead of just terms. Consequently, this notion of rewriting subsumes both standard term rewriting and word rewriting.

We investigate some preservation properties for two classes of languages of unranked ordered terms under this generalization of term rewriting.

The considered classes include languages of hedge automata (HA) and some extension (called CF-HA) with context-free languages in transitions, instead of regular languages.

In particular, we show that the set of unranked terms reachable from a given HA language, using a so called inverse context-free rewrite system, is a HA language. The proof, based on a HA completion procedure, reuses and combines known techniques with non-trivial adaptations. Moreover, we prove, with different techniques, that the closure of CF-HA languages with respect to restricted context-free rewrite systems, the symmetric case of the above rewrite systems, is a CF-HA language. As a consequence, the problems of ground reachability and regular hedge model checking are decidable in both cases.

We give several counter examples showing that we cannot relax the restrictions.

Keywords: Unranked ordered tree languages, tree automata, rewrite systems

Joint work of: Jacquemard, Florent; Rusinowitch, Michaël

Describing DTD by context-free grammars and tree automata over infinite alphabets

Michael Kaminski (Technion - Haifa, IL)

We study two models of computations which can be used to describe DTD with data values: context-free grammars and tree automata over infinite alphabets. Both models possess desirable decision and closure properties. We also present polynomial parsing algorithms for these models. (Joint work with Tony Tan)

Joint work of: Kaminski, Michael; Tan, Tony

Runtime Monitoring of Metric First-order Temporal Properties

Felix Klaedtke (ETH Zürich, CH)

Runtime monitoring is an approach to verifying system properties at execution time by using an online algorithm to check whether a system trace satisfies a temporal property. Applications of runtime monitoring range from software checking to compliance and business activity monitoring.

In this talk, I will present a novel approach to the runtime monitoring of complex system properties. In particular, I present an online algorithm for a safety fragment of metric first-order temporal logic that is considerably more expressive than the logics offered by competing monitoring methods. The algorithms effectively combines ideas from different, but related areas, including database theory, model checking, and model theory. The presented approach, based on automatic structures, allows the unrestricted use of negation, universal and existential quantification over infinite domains, and the arbitrary nesting of both past and time-bounded future operators.

As an alternative to the automata-based representation of automatic structures, I also show how to use and optimize the online monitoring algorithm for the common case where structures consist of only finite relations, over possibly infinite domains.

This is joint work with David Basin, Samuel Müller, and Birgit Pfitzmann

Keywords: Runtime monitoring, metric first-order temporal logic, automatic structures

Joint work of: Basin, David; Klaedtke, Felix; Mueller, Samuel; Pfitzmann, Birgit

R-automata

Pavel Krcal (Uppsala University, SE)

We introduce R-automata - finite state machines which operate on a finite number of unbounded counters. The values of the counters can be incremented, reset to zero, or left unchanged along the transitions. R-automata can be used to model systems with resources (modeled by the counters) which are consumed in small parts but which can be replenished at once. We define the language accepted by an R-automaton relative to a natural number D as the set of words allowing a run along which no counter value exceeds D. We show decidability of the universality problem, i.e., the problem whether there is a number D such that the corresponding language is universal. The decidability proof is based on a reformulation of the problem in the language of finite monoids and solving it using the factorization forest theorem. This approach extends the way in which the factorization forest theorem was used to solve the limitedness problem for

distance automata in Simon, 1994. We also discuss several extensions of the problem – R-automata with Buechi acceptance conditions, (bounded) universality and limitedness problems for (reset) vector addition systems.

Keywords: Finite automata with resources, (bounded) universality, decidability, factorization forest theorem

Joint work of: Abdulla, Parosh; Krcal, Pavel; Yi, Wang

Tree automata with subtree comparisons

Christof Löding (RWTH Aachen, DE)

This talk is about an extension of tree automata in which the execution of transitions is subject to equality and disequality constraints on the subtrees of the current node. A transition can only be executed if the constraints it specifies are satisfied. For ranked trees such automata in general have an undecidable emptiness problem (Mongy 1981). When restricting the comparisons to direct subtrees of the current node, then one obtains a model that is determinizable, is closed under Boolean operations, and has a decidable emptiness problem (Bogaert,Tison 1992). In the context of unranked trees the comparison of subtrees is a possibility to model data from an infinite domain. In this talk we discuss an extension of automata with equality and disequality constraints to the setting of unranked trees. The central question is how to specify the constraints in the transitions such that the emptiness problem for the resulting automaton model remains decidable.

Keywords: Tree automata, equality constraints

Full Paper:
http://automata.rwth-aachen.de/download/papers/karianto/kalo07.pdf

Communicating systems

Anca Muscholl (LaBRI - Bordeaux, FR)

Automated verification (model-checking) and synthesis of communicating systems with unbounded FIFO channels are challenging problems, since these automata are Turing-expressive. We survey in this talk different techniques used in model-checking (symbolic queue representations, lossy channel systems), as well as a solution for synthesizing closed communicating systems with weak channel bounds.

Some Aspects of Program Analysis Beyond the Finite

Markus Müller-Olm (Universität Münster, DE)

Various aspects of programs give rise to an unbounded or even infinite state space. Examples are data aspects, e.g. infinite number domains and dynamic data structures, as well as control aspects, e.g. recursive procedures and thread creation. To handle such aspects completely in an automatic program analysis is a challenge. The purpose of this talk has been to illustrate some techniques for addressing this challenge in automatic fixpoint-based program analysis. More specifically, I presented the following: (1) a technique for checking or inferring polynomial invariants in polynomial programs based on effective weakest preconditions and techniques from computable algebra; (2) Sharir and Pnueli's classic functional approach to come to grips with recursive procedures in interprocedural data flow analysis; and (3) (complete) generalizations of the functional approach to two classes of parallel programs for gen/kill data flow problems.

Keywords: Program analysis, infinite state

Bisimulation Invariance over Transitive Frames

Martin Otto (TU Darmstadt, DE)

Expressive completeness results are obtained for bisimulation invariant FO and MSO queries on certain (non-elementary) classes of (finite and infinite) transitive frames. Surprisingly, a new modality is needed to capture all bisimulation invaraint first-order properties over the class of all finite transitive frames, and also over the class of all transitive frames without infinite irreversible paths - in marked contrast with the situation over the class of all transitive frames. Moreover the techniques extend to the bisimulation invariant fragment of MSO, which in fact collapses to FO though not to basic modal logic.

Joint work with Anuj Dawar

Back to the future: Revisiting precise program verification using SMT solvers

Shaz Qadeer (Microsoft Corp. - Redmond, US)

This paper takes a fresh look at the problem of precise verification of heap-manipulating programs using first-order Satisfiability-Modulo-Theories (SMT) solvers. We augment the specification logic of such solvers by introducing the Logic of Interpreted Sets and Bounded Quantification for specifying properties of heapmanipulating programs. Our logic is expressive, closed under weakest preconditions, and efficiently implementable on top of existing SMT solvers. We

have created a prototype implementation of our logic over the solvers SIMPLIFY and Z3 and used our prototype to verify many programs. Our preliminary experience is encouraging; the completeness and the efficiency of the decision procedure is clearly evident in practice and has greatly improved the user experience of the verifier.

The complexity of lossy channel systems

Philippe Schnoebelen (ENS - Cachan, FR)

We show that reachability and termination for lossy channel systems is exactly at level $\mathcal{F}_{\omega^\omega}$ in the Fast-Growing Hierarchy of recursive functions, the first level that dominates all multiply-recursive functions.

Keywords: Lossy channel systems

Joint work of: Chambart, Pierre; Schnoebelen, Philippe

Full Paper:
http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/CS-lics08.pdf

Static analysis around XML

Luc Segoufin (ENS - Cachan, FR)

I will talk about static analysis for XML data managment. I will give a couple of examples, survey their solutions and present a possible generic approach.

No Strings Attached: Polyhedral Analysis of C String Buffers

Axel Simon (ENS - Paris, FR)

Finite sets of linear inequalities (polyhedra) have been proposed over three decades ago as a means to infer information on the values of program variables. In this talk I will give an introduction to this technique. I shall also comment on how to implement polyhedral operations efficiently. Finally I shall present how to model other program properties (besides normal variables) as polyhedral variables. Specifically, I show how operations on sting buffers in C can be verified. The idea of modelling properties as polyhedral variables carries over to other problem domains and should therefore be interesting to people concerned with the verification of communication protocols, database queries, and others where correctness hinges on numeric properties.

Keywords: Abstract interpretation, polyhedral analysis

Unbounded data in security protocols

S.P. Suresh (CMI - Chennai, IN)

Verification of security protocols is a very active research area. It is particularly challenging because of the need to handle unbounded data. A four-line protocol, specified as a sequence of communications between two parties, will still give rise to an infinite state system, when we model a set of agents (in the presence of a malicious intruder) engaging in many (perhaps) parallel sessions of the protocol. The protocols might also force agents to use 'fresh' random numbers in each session – a major source of unboundedness. Not surprisingly, even basic problems, like the secrecy problem, are undecidable for such systems.

We have proved decidability for reasonable (syntactic) subclasses of protocols.

In this talk, we focus on handling one source of unboundedness – namely the ability of the intruder to force the honest agents to construct very long messages. We prove some powerful theorems on the structure of the derivations used by the intruder to generate new messages from old. Based on these 'normalization' results, we can show that for some specific kinds of protocols, the intruder cannot learn any more messages when the honest agents can be forced to construct longer and longer messages, than when they cannot be so forced.

Keywords: Security protocols, unbounded data, proof normalization

Joint work of: S. P., Suresh; Ramaswamy, Ramanujam

Full Paper:
http://www.cmi.ac.in/ spsuresh/pdffiles/tcs-sub.pdf

See also: A (restricted) quantifier elimination for security protocols. R. Ramanujam and S.P. Suresh. Theoretical Computer Science, volume 367, 2006, pages 228–256.

A theory of stream queries

Jan Van den Bussche (Hasselt University - Diepenbeek, BE)

Data streams are modeled as infinite or finite sequences of data elements coming from an arbitrary but fixed universe. The uni- verse can have various built-in functions and predicates. Stream queries are modeled as functions from streams to streams. Both timed and un- timed settings are considered. Issues investigated include abstract defini- tions of computability of stream queries; the connection between abstract computability, continuity, monotonicity, and non-blocking operators; and bounded memory computability of stream queries using abstract state machines (ASMs).

Joint work of: Gurevich, Yuri; Leinders, Dirk; Van den Bussche, Jan

Full Paper:
http://dx.doi.org/10.1007/978-3-540-75987-4_11

See also: Database Programming Languages, 11th International Symposium, Lecture Notes in Computer Science, volume 4797, pages 153-168, Springer, 2007.

Single Blind Copying Protocols, and XOR

Kumar Neeraj Verma (TU München, DE)

Single blind copying in cryptographic protocols means that at most one piece of unknown data is blindly copied in each protocol step. These protocols were defined by Comon-Lundh and Cortier, who modeled them using the class of first-order Horn clauses called flat and one-variable clauses, and showed them to be decidable in 3-EXPTIME. We describe automated deduction techniques which allow us to obtain a DEXPTIME upper bound, which is optimal. When an XOR symbol is additionally present in the protocols, we show how these techniques allow us to obtain an elementary upper bound.

Keywords: Cryptographic protocol verification, XOR, first-order logic, Horn clauses, resolution

Joint work of: Seidl, Helmut; Verma, Neeraj, Kumar

Verification of Data-Aware Web Services

Victor Vianu (UC at San Diego - La Jolla, US)

The talk describes models of Web services and their compositions, and results on their automatic verification. After a brief review of finite-state models of Web services, the talk focuses on infinite-state models and verification results taking into account the presence of data. Web services accessing an underlying database are modeled as transducers whose control is specified in first-order logic. Compositions of Web services are modeled by communicating transducers. The properties to be verified are expressed in an extended LTL where propositions are interpreted as FO formulas. The results establish under what conditions automatic verification of such properties is possible and provide the complexity of verification. This brings into play a mix of techniques from logic and model checking.

Much of the talk is based on joint work with Alin Deutsch, Liying Sui, and Dayou Zhou (UC San Diego).

Full Paper:
http://www.cs.ucsd.edu/users/vianu/webservices-refs.pdf