05061 Abstracts Collection
Foundations of Semistructured Data
— Dagstuhl Seminar —

Frank Neven[1], Thomas Schwentick[2] and Dan Suciu[3]

[1] Univ. of Limburg, BE
frank.neven@luc.ac.be
[2] Univ. Marburg, DE
tick@informatik.uni-marburg.de
[3] Univ. of Washington, US
suciu@cs.washington.edu

Abstract. From 06.02.05 to 11.02.05, the Dagstuhl Seminar 05061 "Foundations of 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. Semistructured data, XML, database theory, document processing

05061 Summary – "Foundations of Semistructured Data"

As in the first seminar on this topic, the aim o the workshop was to bring together people from the areas related to semi-structured data. However, besides the presentation of recent work, this time the main goal was to identify the main lines of a common framework for future foundational work on semi-structured data. These lines of research are summarized below. The workshop was of a very interdisciplinary nature with invitees from databases, structured documents, programming languages, information retrieval and formal language theory. Several of the lectures were presented by PhD students. We had four invited speakers and a panel on research evaluation. Due to strong connections between topics treated at this workshop, many of the participants initiated new cooperations and research projects.

Keywords: Report, summary

Joint work of: Neven, Frank; Schwentick, Thomas; Suciu, Dan

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2005/227

Active XML and distributed data management

Serge Abiteboul (INRIA Futurs - Orsay, F)

We study query processing for Active XML (AXML) documents that are XML documents where some of the XML data is given explicitly while other parts are given only intentionally by means of calls to Web services. We focus on query evaluation for such documents. The goal is to dramatically reduce data materialization and communication via the management of selective service invocation, service refinement, and sideways information passing. We show a connection between query optimization for AXML and a well-known datalog optimization technique, namely Query-sub-query (QSQ).

Keywords: XML, distributed data management, web services, query optimization

Joint work of: Abiteboul, Serge; Benjelloun, Omar; Milo, Tova

Extending XMark benchmark with XPath 1.0 queries

Loredana Afanasiev (University of Amsterdam, NL)

XMark is a popular benchmark for XML data management. It consists of a scalable document database modelling an Internet auction website and a concise and comprehensive set of XQuery queries which covers the major aspects of XML query processing.

XQuery is much larger than XPath, and the list of queries provided in the XMark benchmark mostly focuses on XQuery features (joins, construction of complex results, grouping) and provides little insight about XPath characteristics. In particular, only child and descendant XPath axes are exploited.

We have developed a set of XPath queries which covers the main aspects of the language. This helps the evaluation of XPath processors, which are the kernel of any XQuery or XSLT implementation.

The issues that we want to target with the benchmark are completeness and scalability.

With completeness we mean the ability to support all the features offered by XPath. With scalability we mean the ability to process queries on documents of increasing sizes.

In the talk we will discuss the motivation, the actual benchmark and some first result obtained with three popular XML engines.

XML Data Exchange: Consistency and Query Answering

Marcelo Arenas (University of Toronto, CDN)

Data exchange is the problem of finding an instance of a target schema, given an instance of a source schema and a specification of the relationship between the source and the target. Theoretical foundations of data exchange have recently been investigated for relational data.

In this talk, we start looking into the basic properties of XML data exchange, that is, restructuring of XML documents that conform to a source DTD under a target DTD, and answering queries written over the target schema. We define XML data exchange settings in which source-to-target dependencies refer to the hierarchical structure of the data. Combining DTDs and dependencies makes some XML data exchange settings inconsistent. We investigate the consistency problem and determine its exact complexity.

We then move to query answering, and prove a dichotomy theorem that classifies data exchange settings into those over which query answering is tractable, and those over which it is coNP-complete, depending on classes of regular expressions used in DTDs. Furthermore, for all tractable cases we give polynomial-time algorithms that compute target XML documents over which queries can be answered.

Joint work of: Arenas, Marcelo; Libkin, Leonid

Semantics and Optimization of XML Updates

Michael Benedikt (Bell Labs - Lisle, USA)

Current XML query languages do not provide the ability to express updates. Although there are some obvious extensions to the syntax of these languages to support updates, the semantics and evaluation of these extensions is not clear.

I'll present a simple update language for XML, and discuss a) possible semantics of programs, and the implications for complexity and expressiveness and b) results, algorithms, and experiments concerning when the semantics of programs agree.

Joint work of: Benedikt, Michael; Bonifati, Angela; Flesca, Sergio; Vyas, Avinash

Types and Patterns for Querying XML

Giuseppe Castagna (ENS - Paris, F)

I will present a rich type system and a set of patterns derived from functional languages, and show by several examples their use in querying semi-structured data.

Adding XML types to ML.

Alain Frisch (INRIA Rocquencourt, F)

Regular expression types have been proposed in the functional programming community to deal with native XML. New languages built around them have been developed previously.

We present an experimental integration of regular expression types into the Objective Caml language. Implicit subtyping of regular expression types does not mix well with automatic type inference in ML, and we have had to develop a three-pass type-checking algorithm to get the best of both world.

The Complexity of Nonrecursive XQuery

Christoph Koch (TU Wien, A)

This talk studies the complexity of the recursion-free fragment of XQuery. We introduce a fragment of XQuery, Core XQuery, that seems to incorporate all the features of a query language on complex values that are traditionally deemed essential. A close connection between monad algebra on lists and Core XQuery (with “child” as the only axis) is exhibited, and it is shown that these languages are actually expressively equivalent up to representation issues. It follows from the conservativity of complex-value algebra without powerset over relational algebra that Core XQuery in LOGSPACE w.r.t. data complexity. We strengthen this to a TC0 upper bound for data complexity.

Regarding combined complexity, Core XQuery is proven complete for the complexity class $TA[2^{O(n)}, O(n)]$ of problems solvable in linear exponential time with a linear number of alternations if equality testing is restricted to atomic values. For Core XQuery with deep equality, a $TA[2^{O(n)}, O(n)]$ lower and an exponential-space upper bound are given.

The monotone fragments of our languages – excluding both negation and deep equality – are complete for nondeterministic exponential time.

Finally, we also study the complexity of Core XQuery without composition. For this practically very important fragment of XQuery, query evaluation is only PSPACE-complete. It is NP-complete for the monotone composition-free fragment of Core XQuery that excludes negation.

Keywords: Complexity, queries, XML, XQuery

Logic Programming, (Automata,) and XML – or – Queries on Trees: A Journey from Dagstuhl to Vienna and Back

Christoph Koch (TU Wien, A)

This talk surveys the main results on the complexity and expressiveness of query languages on tree data obtained by TU Vienna's database group since the time of the first installment of this Dagstuhl workshop. The main topics covered are the complexity and expressiveness of monadic datalog on trees, cyclic and acyclic conjunctive queries, and XPath.

Keywords: Queries, complexity, expressiveness, succinctness, XML, datalog, XPath

A Word-based Query-aware Compressor for XML Documents

Alberto Laender (Federal University of Minas Gerais, Brazil, BR)

XML has become a de facto standard for data exchanging over the Internet. However, efficiently storing and querying XML data is still an open problem. Thus, several recent efforts have been made to deploy techniques to directly query over compressed XML data. In this talk we present YAQCX, Yet Another Query-aware Compressor for XML.

YAQCX adopts word-based modeling combined with byte-coding to provide a very efficient approach to compressing, decompressing and querying XML data. YAQCX addresses part of XPath with a powerful pattern matching extension that allows regular expressions, range queries, and partial matching. Additionally, when processing queries, it accesses the actual compressed data as few as possible, for example to solve predicates on contents or to show results. Based on our experiments, we show that YAQCX compression ratios are reasonably close to XMill's and, for some cases, better than those of other query-aware compressors, such as XQzip and XGrind. We also show that YAQCX decompresses faster than XMill, compresses and decompresses faster than XGrind, and outperforms XGrind regarding query processing.

Keywords: XML, Query-aware Compressor, XPath, YAQCX

Joint work of: Laender, Alberto H. F.; Lage, Juliano P.; Moura, Edleno S.

Which XML Schemas Admit 1-Pass Preorder Typing?

Wim Martens (University of Limburg, B)

It is shown that the class of regular tree languages admitting one-pass preorder typing is exactly the class defined by restrained competition tree grammars introduced by Murata et al., 2001. In a streaming context, the former is the largest class of XSDs where every element in a document can be typed when its opening tag is met.

The main technical machinery consists of semantical characterizations of restrained competition grammars and their subclasses. In particular, they can be characterized in terms of the context of nodes, closure properties, allowed patterns and guarded DTDs. It is further shown that deciding whether a schema is restrained competition is tractable. Deciding whether a schema is equivalent to a restrained competition tree grammar, or one of its subclasses, is much more difficult: it is complete for exptime. We show that our semantical characterizations allow for easy optimization and minimization algorithms.

Finally, we relate the notion of one-pass preorder typing to the existing XML Schema standard.

Marrying XPath to Regular Tree Queries: Looping Caterpillars.

Maarten Marx (University of Amsterdam, NL)

There are two main paradigms for querying semi structured data: regular path queries and XPath. The aim of this paper is to provide a synthesis between these two.

This synthesis is given by a small addition to tree walk automata and the corresponding caterpillar expressions. These are evaluated on unbounded finite sibling ordered trees. At the expression level we add an operator whose meaning is intersection with the identity relation. This language can express every first order definable relation and its expressive power is characterized by pebble tree walk automata that cannot inspect pebbles. In passing we define an expansion of the caterpillar expressions whose expressive power is characterized by ordinary pebble tree walk automata. Combining results from Bloem-Engelfriet and Gottlob-Koch, we also define an XPath like query language which is complete for all MSO definable binary relations.

Exploiting Structural Similarity For Effective Web Information Extraction

Elio Masciari (ICAR - CNR, Arcavacata di Rende, I)

In this paper we propose an architecture that exploit web pages stuctural information for the extraction of relevant information from them. In this architecture, a primary role played by a distance-based classification methodology is devised.

Such a methodology is based on an efficient and effective technique for detecting structural similarities among semistructured documents, which significantly differs from standard methods based on graph-matching algorithms.

The technique is based on the idea of representing the structure of a document as a time series in which each occurrence of a tag corresponds to a given impulse. By analyzing the frequencies of the corresponding Fourier transform, we can hence state the degree of similarity between documents.

Experiments on real data show the effectiveness of the proposed technique.

Keywords: DFT, Web Document Structural Similarity

Joint work of: Masciari, Elio; Flesca, Sergio; Manco, Giuseppe; Pontieri, Luigi; Pugliese, Andrea

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2005/230

Querying Business Processes (and other software specs) with BP-QL

Tova Milo (Tel Aviv University, IL)

The recently emerging BPEL standard supports a declarative specification of both interfaces and full operational logic of business processes. Organizations that use such specifications may greatly benefit from the ability to query them, to analyze and understand the details and requirments of process components. Allowing cooperating organizations to query their combined processes will enhance the benefits of a query facility. In this talk we present BP-QL, a data model and a query language for possibly distributed BPEL specifications. The data model abstracts on BPEL XML-based representation, and supports graphical query formulation. The query language takes full advantage of features like zoom-in/zoom-out that are available in statecharts, the formalism used for BPEL operational logic specification. A prototype implementation illustrates that queries may be efficiently answered.

Keywords: Business processes, BPEL, Query Language

Boosting first-order logic with data

Anca Muscholl (LIAFA - Université Paris VII, F)

In a data word each position carries a label from a finite alphabet and a data value from some infinite domain. It is shown that two-variable logic (with successor and order relation) on such strings is decidable. The complexity is as hard as Petri net reachability.

Joint work of: Muscholl, Anca; Bojanczyk, Mikolaj; David, Claire; Segoufin, Luc; Schwentick, Thomas

N-ary Queries by Tree Automata

Joachim Niehren (INRIA Futurs, F)

Information extraction from semi-structured documents requires to find n-ary queries in trees that define appropriate sets of n-tuples of nodes. We propose new representation formalisms for n-ary queries by tree automata that we prove to capture MSO. We then investigate n-ary queries by unambiguous tree automata which are relevant for query induction in multi-slot information extraction.

We show that this representation formalism captures the class of n-ary queries that are finite unions of Cartesian closed queries, a property we prove decidable.

Keywords: Information extraction, semistructured documents, node selecting queries in trees

Joint work of: Niehren, Joachim; Planque, Laurent; Talbot, Jean-Marc; Tison, Sophie

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2005/226

Forward XPath-like Queries Revisited

Dan-Alexandru Olteanu (Universität München, D)

This talk reports on rewriting XPath-like queries into forward equivalents, i.e., queries without reverse predicates, using the theory of term rewriting systems. We give three such systems and discuss their properties like soundness and completeness, and confluence.

We show then how the applications of these rewriting systems shed also light on query language properties like the expressivity of some of its fragments, the query minimization, or the complexity of query evaluation.

Schema-based Scheduling of Event Processors and Buffer Minimization for Queries on Structured Data Streams

Stefanie Scherzinger (TU Wien, A)

We introduce an extension of the XQuery language, FluX, that supports event-based query processing and the conscious handling of main memory buffers.

Purely event-based queries of this language can be executed on streaming XML data in a very direct way. We then develop an algorithm that allows to efficiently rewrite XQueries into the event-based FluX language.

This algorithm uses order constraints from a DTD to schedule event handlers and to thus minimize the amount of buffering required for evaluating a query. We discuss the various technical aspects of query optimization and query evaluation within our framework. This is complemented with an experimental evaluation of our approach.

Joint work of: Koch, Christoph; Scherzinger, Stefanie; Schweikardt, Nicole; Stegmaier, Bernhard

See also: Christoph Koch, Stefanie Scherzinger, Nicole Schweikardt, Bernhard Stegmaier: Schema-based Scheduling of Event Processors and Buffer Minimization for Queries on Structured Data Streams. VLDB 2004

See also: Christoph Koch, Stefanie Scherzinger, Nicole Schweikardt, Bernhard Stegmaier: FluXQuery: An Optimizing XQuery Processor for Streaming XML Data. Demo at VLDB 2004

Tight Lower Bounds for Query Processing on Streaming and External Memory Data

Nicole Schweikardt (HU Berlin, D)

We consider a scenario where we want to query a large dataset that is stored in external memory and does not fit into main memory. The most constrained resources in such a situation are the size of the main memory and the number of random accesses to external memory. We note that sequentially streaming data from external memory through main memory is much less prohibitive.

We propose an abstract model of this scenario in which we restrict the size of the main memory and the number of random accesses to external memory, but do not restrict sequential reads.

A distinguishing feature of our model is that it admits the usage of unlimited external memory for storing intermediate results, such as a hard disk or even several hard disks that can be accessed in parallel.

In the scenario with a single hard disk, one can use results from communication complexity to obtain lower bounds. For example, we prove a hierarchy based on the number of sequential scans of the hard disk, and we obtain tight lower bounds for processing XPath queries.

In the scenario where several hard disks can be accessed in parallel,communication complexity (apparently) does not help for proving lower bounds. To prove a lower bound for the sorting problem in this scenario, we simulate our model by a non-uniform computation model (so-called "list machines") for which we can establish the lower bounds by combinatorial means.

Keywords: Lower bounds, PXath processing, sorting problem, external memory data

Joint work of: Grohe, Martin; Koch, Christoph; Schweikardt, Nicole

Regular tree languages definable in FO

Luc Segoufin (INRIA Futurs - Orsay, F)

We consider regular languages of ranked labeled trees. We give an algebraic characterization of the regular languages over such trees that are definable in first-order logic in the language of labeled graphs. These languages are the analog on ranked trees of the “locally threshold testable” languages on strings. We show that this characterization yields a decision procedure for determining whether a regular collection of trees is first-order definable: the procedure is polynomial time in the minimal automaton presenting the regular language.

Joint work of: Segoufin, Luc; Benedikt, Michael

Type-Checking XML Transformers with Macro Tree Transducers

Helmut Seidl (TU München, D)

MSO logic on unranked trees has been identified as a convenient theoretical framework for reasoning about expressiveness and implementations of practical XML query languages.

As a corresponding theoretical foundation of XML transformation languages,we propose here the language TL. Our proposal is based on the language DTL from (Maneth,Neven 1999)which incorporates full MSO pattern matching, arbitrary navigation in the input tree using also MSO patterns, and named procedures. The new language generalizes DTL by additionally allowing procedures to accumulate intermediate results in parameters. We prove that TL – and thus in particular DTL - despite their expressiveness still allow for effective inverse type inference. This result is obtained by means of a translation of TL programs into compositions of top-down finite state tree transductions with parameters, also called (stay) *macro tree transducers.*

Keywords: XML Transformations, Macro Tree Transducers, Type-Checking

Joint work of: Seidl, Helmut; Berlea, Alex; Maneth, Sebastian; Perst, Thomas

What's next ?

Dan Suciu (University of Washington, USA)

The original goal of semistructured data in the mid 90's was to handle the imprecision in the structure of the data, as found in data integration and in some "new kinds of data".

It solved the problem by allowing more flexible structure, and it lead to a new paradigm with tremendous success in research and in industry. Today, however, we find a much richer variety of imprecise information in data management: misspellings, missing data values, measurement errors, multiple representations of the same object, imperfect alignment between schemas, constraint violations, etc. I will argue in this talk that a new query paradigm is possible to emerge, based on probabilistic databases, which can treat in a uniform way different kinds of imprecise information. The new paradigm is enabled by several convergent technologies, and is theoretically rooted in probabilities and logic. I will describe a wish list for a theory of probabilistic data management.

Automata on Unranked Trees: Restrictions and Extensions

Wolfgang Thomas (RWTH Aachen, D)

Two results are presented:

1) Over unranked tress, a model of deterministic top-down automaton is introduced, and the accepted tree languages are characterized by the property "path-closed", leading to an effective characterization of deterministically top-down recognizable regular tree languages.

2) The Parikh automata (introduced by Klaedtke and Ruess) are discussed as a tool to generalize the Presburger tree automata of Seidl, Schwentick and Muscholl. The non-emptiness problem for Parikh automata is treated in a still further generalized setting, where "semi-polynomial sets" are considered in place of semi-linear sets. As a partial result in this direction, it is shown that the intersection of a semi-polynomial set with a componentwise semi-linear set is again semi-polynomial (and can be tested for non-emptiness).

Keywords: Automata, unranked trees, parikh automata

Joint work of: Thomas, Wolfgang; Christau, Julien; Löding, Christof; Karianto, W.; Krieg, A.

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2005/228

Deciding Well-Definedness of XQuery Fragments

Stijn Vansummeren (University of Limburg, B)

Unlike in traditional query languages, expressions in XQuery can have an undefined meaning (i.e., these expressions produce a run-time error). It is hence natural to ask whether we can solve the well-definedness problem for XQuery: given an expression and an input type, check whether the semantics of the expression is defined for all inputs adhering to the input type. In this paper we investigate the well-definedness problem for non-recursive fragments of XQuery under a bounded-depth type system. We identify properties of base operations which can make the problem undecidable, and give conditions which are sufficient to ensure decidability.

Keywords: XQuery, well-definedness

Expressive Power of XQuery Fragments

Roel Vercammen (University of Antwerp, B)

XQuery is known to be a powerful XML query language with many bells and whistles. From a researcher's point of view many of these features often unnecesarilly complicate the study of the language and its properties.

LiXQuery is a stripped-down version of XQuery with a concise and formal semantics that is consistent with the XQuery semantics. However, for many common queries we do not need all the expressive power of LiXQuery. We investigate the effect of omitting certain features of LiXQuery on the expressive power of the language. We start from a base fragment that can express many commonly used features such as some built-in functions, arithmetic, boolean operators, node and value comparisons, path expressions, simple for-loops, XPath set operations, and universal and existential quantifaction. This base fragment can be extended by several optional features being count and sum aggregation functions, sequence generation, node construction, position information in for loops, and recursion. In this way we obtain 64 different XQuery fragments which can be divided into 17 different equivalence classes such that two fragments can express the same functions iff they are in the same equivalence class.

Keywords: XQuery, Expressive Power

Joint work of: Hidders, Jan; Marrara, Stefania; Paredaens, Jan; Vercammen, Roel

Node Identification Schemes for Efficient XML Retrieval

Felix Weigel (CIS - Universität München, D)

Node identifiers (IDs) encoding part of the tree structure in XML documents can save I/O for table look-ups, thus speeding up the evaluation of path and tree queries on large persistent document collections. In particular, binary tree relations such as the extended XPath axes can be either decided for a given pair of node IDs, or reconstructed for a single node ID, without access to secondary storage. Several ID schemes have been proposed so far, which differ with respect to (1) expressiveness, i.e. which relations can be decided or reconstructed from IDs, (2) the runtime performance and asymptotic behaviour of decision and reconstruction operations, (3) the storage overhead for the IDs, and (4) robustness, i.e. behaviour in the presence of updates. First we review five ID schemes, positioning them in the trade-off between these four comparison criteria. Then a new ID scheme called BIRD, for Balanced Index-based ID scheme for Reconstruction and Decision, is introduced and illustrated throughout several examples of decision and reconstruction operations on IDs. We argue that emphasizing runtime performance and expressive power, BIRD's strategy in the above trade-off is best for many applications, especially where storage minimization is not the primary goal and updates occur in a bulk-fashion rather than in realtime. Our experimental results on document collections of up to one gigabyte prove BIRD to be most efficient in terms of expressiveness and runtime performance. Most notably, BIRD is the only scheme to support both decision and reconstruction of many relations in constant time. But also in terms of storage and robustness BIRD is highly competitive.

Keywords: Node identification scheme, labelling scheme, numbering scheme, naming scheme, tree encoding, BIRD

Joint work of: Weigel, Felix; Schulz, Klaus U.; Meuss, Holger

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2005/229