SIGMOD Officers, Committees, and Awardees

Chair
Raghu Ramakrishnan
Yahoo! Research
2821 Mission College
Santa Clara, CA 95054
USA
<First8CharsOfLastName AT yahoo-inc.com>

Vice-Chair
Yannis Ioannidis
University of Athens
Department of Informatics & Telecom
Panepistimioupolis, Informatics Buildings
157 84 Ilissia, Athens
HELLAS
<yannis AT di.uoa.gr>

Secretary/Treasurer
Mary Fernández
ATT Labs - Research
180 Park Ave., Bldg 103, E277
Florham Park, NJ 07932-0971
USA
<mff AT research.att.com>

SIGMOD Officers, Committees, and Awardees (continued)

SIGMOD Edgar F. Codd Innovations Award

For innovative and highly significant contributions of enduring value to the development, understanding, or use of database systems and databases. Until 2003, this award was known as the "SIGMOD Innovations Award." In 2004, SIGMOD, with the unanimous approval of ACM Council, decided to rename the award to honor Dr. E.F. (Ted) Codd (1923 - 2003) who invented the relational data model and was responsible for the significant development of the database field as a scientific discipline. Recipients of the award are the following:

Michael Stonebraker (1992)	Jim Gray (1993)	Philip Bernstein (1994)
David DeWitt (1995)	C. Mohan (1996)	David Maier (1997)
Serge Abiteboul (1998)	Hector Garcia-Molina (1999)	Rakesh Agrawal (2000)
Rudolf Bayer (2001)	Patricia Selinger (2002)	Don Chamberlin (2003)
Ronald Fagin (2004)	Michael Carey (2005)	Jeffrey D. Ullman (2006)
Jennifer Widom (2007)		

SIGMOD Contributions Award

For significant contributions to the field of database systems through research funding, education, and professional services. Recipients of the award are the following:

Maria Zemankova (1992)	Gio Wiederhold (1995)	Yahiko Kambayashi (1995)
Jeffrey Ullman (1996)	Avi Silberschatz (1997)	Won Kim (1998)
Raghu Ramakrishnan (1999)	Michael Carey (2000)	Laura Haas (2000)
Daniel Rosenkrantz (2001)	Richard Snodgrass (2002)	Michael Ley (2003)
Surajit Chaudhuri (2004)	Hongjun Lu (2005)	Tamer Özsu (2006)
Hans-Jörg Schek (2007)		

SIGMOD Doctoral Dissertation Award

The annual ACM SIGMOD Doctoral Dissertation Award, inaugurated in 2006, recognizes excellent research by doctoral candidates in the database field.

- **2006** *Winner*: Gerome Miklau, University of Washington
  Runners-up: Marcelo Arenas, University of Toronto; Yanlei Diao, University of California at Berkeley.
- **2007** *Winner*: Boon Thau Loo, University of California at Berkeley
  Honorable Mentions: Xifeng Yan, University of Illinois at Urbana-Champaign; Martin Theobald, Saarland University

A complete listing of all SIGMOD Awards is available at: **http://www.sigmod.org/awards/**

[Last updated on September 26, 2007]

Editor's Notes

Welcome to my inaugural issue of SIGMOD Record. I am very excited and honored to take over as Editor; SIGMOD Record has a long history (36 volumes) and has consistently maintained a high level of quality as a result of the hard work of many people in our community. Many thanks go to all the previous Editors, Associate Editors, authors, and reviewers for making this possible. As I take over, I would also like to thank the SIGMOD Executive Committee for their trust and encouragement, as well as the outgoing Record Editor, Mario Nascimento, the Associate Editors (Ugur Cetintemel, Brian Cooper, Andrew Eisenberg, Cesar Galindo-Legaria, Denilson, Barbosa, Leonid Libkin, Jim Melton, Len Seligman, and Marianne Winslett) and the staff from ACM HQ (Julie Goetz and Ginger Ignatoff) for their help, advice, and patience during the transition process.

The first thing that you have probably noticed is that the cover of SIGMOD Record has changed! I wish I could get the credit for this, but this is part of an ACM-wide effort with the overall aim of making the ACM brand more recognizable for audiences around the world. The last major redesign of SIGMOD Record was back in the previous millennium, with the September 1999 issue. I hope you agree with me that the new design is much "cleaner" and more compelling, while maintaining the identity of SIGMOD and SIGMOD Record. Many thanks to the ACM Marketing Department for their help in updating the design.

The second thing that you must have observed is that this is still the *June* issue, and not the September issue; the enclosed CD with the proceedings of the SIGMOD 2007 and PODS 2007 conferences was probably a big give-away. The main reason behind the delay is the switch from an email-based to a web-based paper submission system. The new system, *RECESS* (short for SIGMOD RECord Electronic Submission System) was designed, developed, and tested in record time; it went live on June 7, 2007 at `http://db.cs.pitt.edu/recess`. The implementation was done by Julie Pagano, an undergraduate student at the University of Pittsburgh, who did a fantastic job. RECESS greatly simplifies managing the entire review and publication process, but it introduced a small start-up overhead, due to a small backlog of papers, which will dissipate in the next couple of issues. In fact, you should hopefully receive the September issue within a few weeks of receiving this issue; I expect the December issue to be on time.

We start this issue with two regular articles. The first one is on estimating the selectivity of cosine similarity predicates, that is useful in approximate string matching (by Tata and Patel). The second paper is also on estimation, but this time for the cardinality of queries on ontologies (by Shinoroshita et al.). The next article is a contribution to the Database Principles column (edited by Leonid Libkin), discussing the expressivity and algebras for navigational XPath (by Cate and Marx).

We continue with a very lively interview of Georg Gottlob by Marianne Winslett for the *Distinguished Profiles in Data Management* column (formerly known as the Distinguished DB Profiles column). Read it to find out (among other things) why we do not see more XML data on the Web today.

This issue also includes three event reports (edited by Brian Cooper). The first one is for the 3^{rd} International Workshop on XQuery Implementation, Experience and Perspectives (XIME-P 2006), which was held in June 2006, together with SIGMOD. The second one is for the ACM Workshop on Health Information and Knowledge Management (HIKM 2006), which was held in November 2006, together with CIKM. The last report is for the 8^{th} International Conference on Information Integration & Web-Services and Applications (iiWAS 2006) and the 4^{th} International Conference on Advances in Mobile Computing and Multimedia (MoMM 2006) that were held in December 2006. Finally, we have the Call for Papers for SIGMOD 2008 and PODS 2008, along with a message about the experimental repeatability requirements for SIGMOD 2008 and an announcement for the ACM - Infosys Foundation Award.

Before closing, I would like to welcome Magdalena Balazinska from the University of Washington to the Editorial Board and thank her for volunteering her time. Magdalena will serve as the new Associate Editor in charge of the *Systems and Prototypes* column; the goal of the column is to highlight recent and exciting systems

being built by researchers in the database community. I am looking forward to working with Magda and all the Associate Editors.

Even though the *outside* of SIGMOD Record has changed, the *inside* structure remains largely the same. I plan to continue with the same general structure (selection of high quality papers from those submitted plus solicited contributions), relying greatly on the help of the Associate Editors. Having said that, I would also encourage and welcome any suggestions that you have on special topics issues, on new columns, or in other ways to make SIGMOD Record even better.

Alexandros Labrinidis
September 2007

Past SIGMOD Record Editors:

Harrison R. Morse	(1969)
Daniel O'Connell	(1971 – 1973)
Randall Rustin	(1975)
Thomas J. Cook	(1981 – 1983)
Jon D. Clark	(1984 – 1985)
Margaret H. Dunham	(1986 – 1988)
Arie Segev	(1989 – 1995)
Jennifer Widom	(1995 – 1996)
Michael Franklin	(1996 – 2000)
Ling Liu	(2000 – 2004)
Mario Nascimento	(2005 – 2007)

2007 SIGMOD Award Winners

2007 SIGMOD Edgar F. Codd Innovations Award

Jennifer Widom

Professor Jennifer Widom is the recipient of the 2007 SIGMOD Edgar F. Codd Innovations Award for a series of fundamental contributions in several database sub-areas. Her contributions have either brought new structure to existing database research areas, or opened up whole new lines of database research.

Professor Widom has made fundamental contributions in areas including database rule systems, data warehousing and view maintenance, semi-structured data management, stream database systems, and database support for uncertainty and lineage. In the area of database rule systems, her 1990 ACM SIGMOD paper offered the first complete framework for incorporating production rules into a DBMS with well-defined semantics; that paper was later recognized for its contribution via the SIGMOD Test of Time Award in 2000. In her 1991 VLDB paper, Professor Widom applied those results to the problem of view maintenance, providing practical guidelines for determining when materialized views can be efficiently incrementally maintained. This paper won both the Best Paper Award for the 1991 VLDB Conference as well as the Ten Year Paper Award at VLDB 2001. In the area of semi-structured data, Professor Widom's Lore system pioneered techniques for storing, querying, and managing semi-structured data, before the emergence and popularity of XML. This work heavily influenced subsequent work on XML databases and their query languages. Professor Widom's CQL language is widely regarded as having brought structure and well-thought-out semantics to the problem of querying streams; it is proving foundational for the StreamSQL standardization effort by IBM, Oracle, StreamBase, and others. Recently, Professor Widom has turned her attention to uncertainty and data lineage in her TRIO project, and this work appears to be again spearheading a new research thrust for the database community.

2007 SIGMOD Contributions Award

Hans-Jörg Schek

Professor Hans-Jörg Schek is the recipient of the 2007 SIGMOD Contritions Award for his significant service to the database research community, as well as the broader scientific community, as a schola r, an educator, a supervisor, a referee of and advisor for large collaborative research projects in Europe, and as an organizer of conferences, journals, and other community activities. He has also served the community through his pioneering research efforts in nested relational data management and database support for advanced applications such as office automation, engineering information management, and digital libraries.

Professor Schek's contributions are numerous. As one notable example, Professor Schek served as the founding Editor-in-Chief of the VLDB Journal, which according to Thomson's Science Citation Index, currently stands as the computer science journal with the highest impact factor. Professor Schek served as a Trustee of the VLDB Endowment, the organization that sponsors and oversees the annual VLDB Conference and its associated activities, from 1998-2006. As a teacher, Professor Schek graduated several generations of students, many of whom are today highly recognized database professors themselves. He thus played a major role in growing the European database research community. Over the years Professor Schek rang the warning bell of potential irrelevance, urging the database community to "get out of its box", engage with other communities, and move from traditional databases to more general, diverse,

and universal data management. He modeled this philosophy through his research and worked to broaden the scope of the VLDB Conference and the VLDB Journal. He served and continues to serve as a role model for numerous young scientists, both his own students and others in our field, with his dedication to community service, to database research, and to the continued vitality of the field.

2007 SIGMOD Test of Time Award

Online Aggregation
Joseph M. Hellerstein (UC Berkeley), Peter J. Haas (IBM Almaden Research Center), and Helen J. Wang (UC Berkeley)

The paper *Online Aggregation* from the 1997 ACM SIGMOD Conference in Tucson, AZ, was chosen from a number of potential candidates for the lasting impact that it had by opening up new database research directions. In the years since 1997, this frequently cited paper has had a significant influence on subsequent research on approximate query processing, sampling-based data reduction, and more recently, approaches to handling aggregation in stream data management and continuous query processing systems.

2007 SIGMOD Doctoral Dissertation Award

- *Winner*: Boon *Thau Loo* (advisors: Joseph M. Hellerstein and Ion Stoica), University of California at Berkeley
- *Honorable Mentions*: *Xifeng Yan*, University of Illinois at Urbana-Champaign; *Martin Theobald*, Saarland University

2007 SIGMOD Best Paper Award

- Compiling Mappings to Bridge Applications and Databases
  Sergey Melnik, Atul Adya, and Philip Bernstein
- Scalable Approximate Query Processing with the DBO Engine
  Christopher Jermaine, Subramanian Arumugam, Abhijit Pol, and Alin Dobra

2007 SIGMOD Undergraduate Awards

- *Rui Fang*, Hong Kong University of Science & Technology, China
- *Marcin Kwietniewski*, Warsaw University of Technology, Poland, and York University, Canada
- *Yin Yee (Samantha) Leung*, University of British Columbia, Canada
- *Rui Li*, Shanghai Jiao Tong University, China
- *Yinan Li*, Peking University, China
- *Zhongyuan Wang*, Renmin University, China
- *Zhijun Yin*, Fudan University, China

A complete listing of all SIGMOD Awards is available at: **http://www.sigmod.org/awards/**

Estimating the Selectivity of *tf-idf* based Cosine Similarity Predicates

Sandeep Tata Jignesh M. Patel
Department of Electrical Engineering and Computer Science
University of Michigan
2260 Hayward Street, Ann Arbor, Michigan 48109
{tatas,jignesh}@eecs.umich.edu

Abstract

An increasing number of database applications today require sophisticated approximate string matching capabilities. Examples of such application areas include data integration and data cleaning. Cosine similarity has proven to be a robust metric for scoring the similarity between two strings, and it is increasingly being used in complex queries. An immediate challenge faced by current database optimizers is to find accurate and efficient methods for estimating the selectivity of cosine similarity predicates. To the best of our knowledge, there are no known methods for this problem. In this paper, we present the first approach for estimating the selectivity of tf.idf based cosine similarity predicates. We evaluate our approach on three different real datasets and show that our method often produces estimates that are within 40% of the actual selectivity.

1 Introduction

A growing number of database applications require approximate string matching predicates on text attributes. For example, in data scrubbing [4] and data integration applications [5, 6], these predicates are valuable in dealing with spelling errors, typographical errors, and problems with non-uniform data representation. Address fields for instance can refer to the same location, but be written using different conventions ("1301 Beal Ave., Ann Arbor" vs. "1301 Beal Avenue, Ann Arbor"). Another example is the case of item descriptions which vary slightly from vendor to vendor. One might want to search on the description field to find similar items.

For many real world application, the authors in [3, 8] show that the cosine similarity metric can robustly handle spelling errors, rearrangement of words, and other differences in strings. They also demonstrate that cosine similarity searches and joins can be implemented completely in SQL without adding any code to the relational engine. While cosine similarity is a good metric for comparing strings, to the best of our knowledge, there are no known methods for estimating the selectivity of these predicates. As a result, optimizers may often produce inefficient plans for queries involving these predicates. With the increasing use of cosine similarity predicates, there is an urgent need to develop methods that can estimate the selectivity of these predicates.

In this paper, we discuss a technique for estimating the selectivity of tf.idf based cosine similarity predicates. We make use of a statistical summary of the distribution of different tokens in the database. We also make use of the distribution of the dot product of a typical query with a database row's tf.idf vector. We present two techniques that use the data in different ways and compare their performance on different datasets.

The rest of the paper is organized as follows: Section 2 describes related work and briefly reviews cosine similarity. Section 3 describes the summary structure we employ. Section 4 describes the algorithm used to compute the estimates. The experimental evaluation is presented in Section 5. Finally, we make concluding remarks and point to directions of future work in Section 6.

2 Review and Related Work

Cosine similarity is a vector-based measure of the similarity of two strings. The basic idea behind cosine similarity is to transform each string into a vector in some high dimensional space such that similar strings are close to each other. The cosine of the angle between two vectors is a measure of how "similar" they are, which in turn, is a measure of the similarity of these strings. If the vectors are of unit length, the cosine of the angle between them is simply the dot product of the vectors.

There are many ways of transforming a string in the database into a vector. The *tf.idf* vector is a popular choice for this representation. The *tf.idf* vector is composed of the product of a *term frequency* and the *inverse document frequency* for each token that appears in the string. The process of constructing the *tf.idf* vector is described below.

As a first step towards implementing the cosine similarity predicate, we construct a *tf.idf* vector for each row in the relation. If there are multiple string attributes of interest in each row, then we need to compute a vector for each string attribute. To keep the discussion simple, we will assume there is only one string attribute in the relation that is used in a cosine similarity operation.

The length of the *tf.idf* vector is equal to the total number of tokens. A token can be a q-gram or a word. If we are using q-grams, then the length of each vector is the total number of possible q-grams = $|A|^q$, where $|A|$ is the size of the alphabet. The vector stores the *tf.idf* value corresponding to each token for each string. The term *frequency* is the number of times the token appears in the string and is a measure of the importance of that token in the string. The inverse document frequency (inverse of the number of strings in which the token appears) serves to normalize the effect of tokens (like *"the"*) that appear commonly in many strings. The product of *tf* and *idf* is a measure of the importance of the token in the string and the database as a whole. Note that in most real datasets, the strings are very short when compared to the total number of possible tokens, and therefore these vectors tend to be very sparse.

When a query comes in, the normalized *tf.idf* vector corresponding to the query is constructed. The *idf* of each term in the query is just 1. We compute the dot product of this vector with the vector for each row in the database: this is the cosine similarity. If the query and the string share more terms, the dot product is higher. In addition, if they share more "uncommon" terms, that contributes to the score more. The predicate is typically of the form *cosine_similarity (R.s,"Dr. Jekyll")* > 0.5, and is evaluated by selecting all those strings where the dot product exceeds the given threshold [3, 8].

To the best of our knowledge, there is no literature on techniques to estimate the selectivity of a cosine similarity predicate. The work closest to ours is [7] where the authors describe a selectivity estimation technique for a fuzzy string predicate. However, this fuzzy predicate is different from any of the well known predicates and has not been shown to perform like cosine similarity in real world tasks [3, 8].

In this paper, we focus on tf.idf based cosine similarity. Although there are other vector representations where cosine similarity can be used, tf.idf is a popular choice in many applications because of its simplicity and robustness. The techniques in this paper take advantage of some of the properties of tf.idf, and therefore will likely require adaptation to work with other vector representations.

Interestingly, the authors of [1] show that many metric distance measures follow a power law distribution for average number of neighbors with respect to distance. That is, number of neighbors within distance s is proportional to s^d where d is some positive constant. As has been argued in [7], this property does not hold for similarity functions like the edit distance, and in our case, the cosine similarity function because of the large number of pairs of words within the same distance. Furthermore, this approach only estimates the average number of neighbors for a string in a dataset, and does not estimate the number of neighbors for a given query string which could be very different from the average.

3 Summary Structure

The summary structure we describe stores a concise representation of the distribution of the *tf.idf* values for each token. If we think of the *tf.idf* vectors for all tuples in the relation as a matrix, we observe that this matrix is very sparse. Table 1 shows a sketch of such a matrix. Most rows in this table are sparse because a given string

rowID	string	Tok 1	.....	Tok N
1	s_1	w_1^1	...	w_N^1
.	.	.	.	.
.	.	.	.	.
R	s_R	w_1^R	...	w_N^R
		μ_1, σ_1, C_1	...	μ_N, σ_N, C_N

Table 1: A Table and the tf.idf Vectors

is likely to contain only a small number of tokens. In addition, most columns in this table are also sparse, because very few tokens (like "the") are likely to appear in a large number of strings. We have observed empirically that the probability density function of the *tf.idf* weights for a column is characterized by a large mass of probability at zero (most tokens appear only in a few strings). The rest of the probability is distributed around a small positive value. The proposed summary structure (as shown in the last row of Table 1) captures this distribution by storing the following three values for each token:

1. Mean of X for X $\neq$ 0 (μ_i),
2. Standard Deviation for X $\neq$ 0 (σ_i), and
3. Probability that a q-gram is non zero, 1-Prob(X=0) (C_i)

Assume that all the nonzero *tf.idf* values are stored in a table called *Vectors(token,row,value)*. That is, for each token in the original database, the table *Vectors* stores a record for each row in which this token appears with the *tf.idf* value for that token in that row. This is merely a compact way of storing the (sparse) *tf.idf* vector for each row of the database. The summary structure can be generated from *Vectors* by simply using the following SQL query:

```
SELECT token, avg(value),
stddev(value),count(row)/total_rows
FROM Vectors GROUP BY token;
```

Note that the size of this summary structure is bounded by the number of distinct tokens in the language from which the text is drawn. For example, [3, 8] show that for many real applications cosine similarity works well with a token size of three. Assuming an alphabet of size 50 (characters, numbers, punctuation, etc.) the maximum number of tokens (q-grams) is 125K. Also note that the size of the structure is largely independent of the database size, and for a large text database the summary structure is a very small proportion of the total size.

4 The Estimation Algorithm

The cosine similarity is the dot product of two *tf.idf* vectors representing the query and the database string. The key to estimating the selectivity of a cosine similarity predicate is to understand the distribution of the dot product. In other words, the problem at hand is to compute the cumulative distribution function of the dot product given a) the query vector, and b) a distribution characterizing the *tf.idf* vectors in the database. Once we compute this cumulative probability distribution function, calculating the probability that the cosine similarity exceeds a certain threshold becomes fairly simple.

We model the *tf.idf* vector in the database as a vector of random variables $(X_1 X_2 X_3 ... X_n)$ – one for each token. The dot product can now be modeled as:

$$Y = \Sigma_{i=i}^{n} u_i \times X_i \quad (1)$$

where u is the query vector.

A straightforward approach to understanding the distribution of Y is to model the distribution of each of the $X_i's$ and analytically compute the PDF of Y. However, this turns out to be extremely difficult for any non-trivial characterization of X_i. Alternately if we were to evaluate the PDF of Y by sampling the PDF's of $X_i's$, it turns out that the number of samples required to accurately estimate the selectivity is prohibitively high. We therefore choose an alternate technique where we model the distribution of Y and try to determine the parameters of the distribution.

In order to understand how the dot product is distributed, we generated a large set of sample queries by randomly picking strings in the database and introducing one or two errors in the string. We repeated this experiment for a variety of datasets. We observed that the distribution is as shown in Figure 1. The distribution is characterized by a mass of probability close to zero. The rest of the probability is distributed such that it peaks at a small positive value and a long tail tapering off to 0 at Y = 1. After evaluating several well known distributions, we determined that this data was modeled accurately as an inverse normal distribution [9] with a mass of proba-

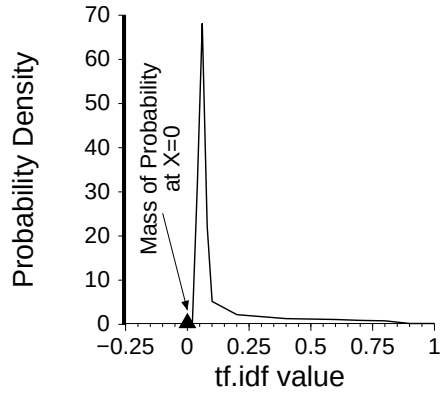


Figure 1: Typical Distribution of the tf.idf Dot Product

bility at 0. We show in the next section, that this simple empirical observation leads to surprisingly good results.

The inverse normal distribution can be completely characterized by its mean and standard deviation. We remind the reader that the probability distribution function of the inverse normal function is:

$$PDF = \sqrt{\frac{B}{2\pi y^3}} exp(-\frac{B}{2y}(\frac{y-A}{A})^2) \qquad (2)$$

where the mean is A, and the variance is $\frac{A^3}{B}$. The cumulative distribution function (CDF) is:

$$CDF = \Phi(\sqrt{\frac{B}{y}}\frac{y-A}{A}) + exp(\frac{2B}{A})\Phi(\sqrt{\frac{B}{y}}\frac{-y-A}{A}) \qquad (3)$$

where $\Phi(x)$ is the CDF of a standard Gaussian.

The problem now reduces to determining the parameters of the inverse normal distribution (mean and variance). We now present two empirical algorithms for estimating the mean and standard deviation of Y using the data at hand. We then show through experiments in Section 5 that these techniques lead to good estimates.

4.1 Algorithm ES

A simple approach to estimate the mean of Y is to use the weighted average of the means of $X_i's$ (in Equation 1) from each column of the *tf.idf* matrix. The mean of each X_i is available in the summary structure. We compute

$$\mu^{ES} = \alpha \times \sum(C_i \times \mu_i \times u_i) \qquad (4)$$

where C_i is the probability that token i assumes a nonzero value. μ_i is the mean of the nonzero values of token i as stored in the summary, and u_i is the *tf.idf* weight of the token i in the query vector.

The standard deviation is also computed similarly:

$$\sigma^{ES} = \beta \times \sum(C_i \times \mu_i \times u_i) \qquad (5)$$

We call this simple approach ES.

In the above equations, α and β are empirically determined scaling constants for a given relation. They are present to accommodate for the fact that the weighted sum of means does not necessarily yield the actual mean of Y. In order to determine α, we first assume $\alpha = 1$. We determine the average value of the ratio $\frac{\mu^{actual}}{\mu^{ES}}$ for a training set of queries and set α to this value. β is determined similarly. Using samples from a real workload for the training set will ensure that these values are more accurate.

Algorithm ES(query,threshold,summary)
1. Construct the *tf.idf* vector u for the query.
2. Compute $\mu_{ES} = \alpha\Sigma_{i=1}^{N}(\mu_i \times C_i \times u_i)$
3. Compute $\sigma_{ES} = \beta\Sigma_{i=1}^{N}(\sigma_i \times C_i \times u_i)$
4. Compute over nonzero u_i :
5. $nz_{ES} = 1 - (\Pi_{i=1}^{N}(1 - C_i))^{1/q}$
6. Compute Estimate =
$nz_{ES} \times inv\_normal\_cdf(threshold, \mu_{ES}, \sigma_{ES})$

Figure 2: Estimation using ES

In order to completely characterize Y, we also need to estimate the mass of probability at $Y = 0$. This is the probability that the dot product is zero. We use:

$$PZ^{ES} = (\Pi_{i=1}^{N}(1 - C_i))^{1/q} \qquad (6)$$

where N is the number of nonzero *tf.idf* weights in the query vector, and q is the length of the tokens used. In effect, we are computing the product of all the values

corresponding to the nonzero entries in the query vector. The exponentiation with $\frac{1}{q}$ is to correct for the fact that q-grams are usually not independent. For instance tokens like 'THA' and 'HAT' are more likely to co-occur because they constitute common words like 'THAT'. This simple approximation leads to some very good estimates.

Once we have μ^{ES}, σ^{ES}, and PZ^{ES}, we calculate the selectivity s of the query as:

$$s = (1 - PZ^{ES}) \times in\_cdf(threshold, \mu^{ES}, \sigma^{ES}) \quad (7)$$

where in_cdf is the CDF for the inverse normal distribution, and $threshold$ is the value obtained from the predicate of the form cosine_similarity(R.a,string) $\geq threshold$.

4.2 Algorithm EL

Although Algorithm ES gives us fairly good estimates, we found that instead of simply learning constants α and β from a training workload, learning a simple function using linear regression can significantly improve the accuracy of the estimate.

Algorithm EL trains functions to estimate the actual mean and the actual standard deviation for the dot product from μ^{ES} and σ^{ES} computed as in ES using $\alpha = 1$ and $\beta = 1$. In the training phase, we use the data from a set of sample queries that is representative of the workload. We train functions f_μ and f_σ to estimate μ^{actual} and σ^{actual} from μ^{ES} and σ^{ES}. We also train a function to better estimate PZ^{actual} using $PZ^{ES}_{corrected}$. If there are changes to the query workload or the data itself, one can retrain these functions to increase their accuracy. (If such retraining is not feasible, then one can resort to the ES algorithm.) For the training function, we empirically tried and evaluated several families of function, including polynomials of various degrees, exponential functions, and combinations of polynomials and exponentials. We found that the following simple family of functions works best for training the estimators:

$$f(x) = c_1 + c_2 x + c_3 e^{-x^2} \quad (8)$$

5 Experimental Evaluation

In this section, we present an experimental evaluation of the estimates produced by the ES and EL algorithms on three datasets. The results are largely representative of many other datasets that we tried. The three dataset that we use are SCH, AUT, and HEAD as described below:

EstimateEL(query,threshold,summary,f_μ, f_σ, f_{nz})
1. Construct the *tf.idf* vector u for the query.
2. Compute $\mu_{eq} = \Sigma_{i=1}^{N}(\mu_i \times C_i \times u_i)$
3. Compute $\sigma_{eq} = \Sigma_{i=1}^{N}(\sigma_i \times C_i \times u_i)$
4. Compute over nonzero u_i :
5. $nz_{eq} = 1 - (\Pi_{i=1}^{N}(1 - C_i))^{1/k}$
6. $\mu_{EL} = f_\mu(\mu_{eq})$
7. $\sigma_{EL} = f_\sigma(\sigma_{eq})$
8. $nz_{EL} = f_{nz}(nz_{eq})$
9. Compute Estimate = $nz_{EL} \times inv\_normal\_cdf(threshold, \mu_{EL}, \sigma_{EL})$

Figure 3: Estimation using EL

1. SCH consists of 99,632 records with high school names and addresses in the USA. The total size of the dataset is 13MB. The school name field was used for cosine similarity.

2. AUT [2] is a set of 371,022 author names from DBLP totaling 8MB.

3. HEAD [10] contains 119,015 article headlines from the Wall Street Journal totaling 7.5MB.

For each dataset, we randomly chose a set of 50 strings from the database itself, and posed 5 queries with each string by varying the cosine similarity threshold from 0.2 to 0.6 in increments of 0.1. Another (different) set was similarly generated to first train ES and EL. Queries were roughly classified as having Low, Medium, or High selectivity based on whether they selected $> 10\%$, $1\% - 10\%$ or $< 1\%$ of the rows respectively.

The size of the summary structure was less than 3% and took less than 3 minutes to construct in each case. Both ES and EL are efficient and take less than 1 millisecond per query to compute an estimate.

We report the average percentage error in Figures 4, 5, and 6. That is, we report $\frac{|estimate - actual|}{actual} \times 100$. The

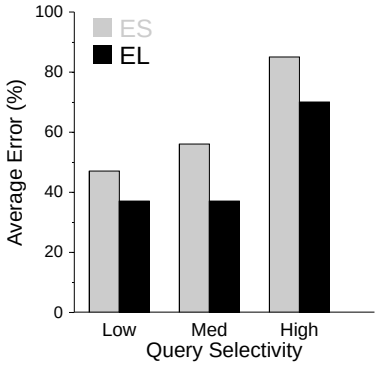


Figure 4: SCH Dataset

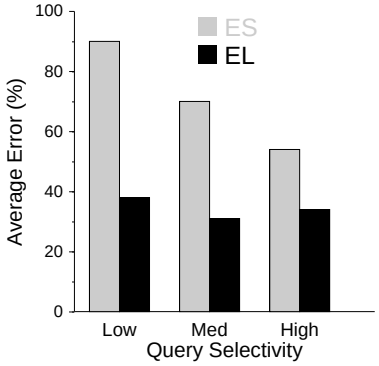


Figure 5: AUT Dataset

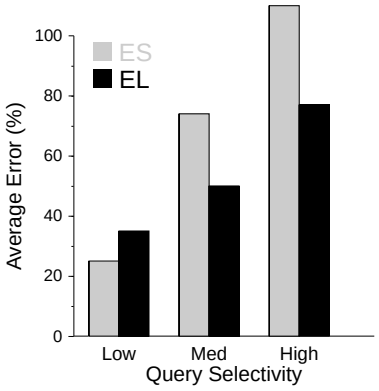


Figure 6: HEAD Dataset

figures show that in each case EL is more accurate than ES by 10 to 50 percentage points. For instance, in Figure 5, in the case of low selectivity queries, ES incurs a 90% error while EL has less than 40% error. Although ES is fairly accurate in many cases, it occasionally has a very large error (eg. high selectivity queries in SCH and HEAD). In all low and medium selectivity cases, the estimates provided by EL have less than 40% error. The error is usually higher in the case of highly selective queries as can be expected. The benefits of using the more complex learning model in EL are evident as they pay off in terms of more accurate estimates.

6 Conclusions and Future Work

In this paper, we have presented the problem of estimating the selectivity of cosine similarity predicates. To our knowledge, this is the first paper to address this problem. We discussed why estimating the selectivity of cosine similarity predicates is a very difficult problem, and proposed a solution based on careful empirical observations about the distribution of the dot product of typical queries. We showed that the approach is space efficient (summaries are small in size) and time efficient (estimation time is also small). We also showed that this technique has reasonably good accuracy in practice.

Directions for future work include exploring analytical modeling for the tf.idf dot product, and alternative approaches that might lead to more accurate estimates.

References

[1] Caetano Traina and Agma J. M. Traina and Christos Faloutsos. Distance Exponent: A New Concept for Selectivity Estimation in Metric Trees. In *ICDE*, pages 195–195, 2000.

[2] Digital Bibliography and Library Project (DBLP), `http://dblp.uni-trier.de/`.

[3] L. Gravano, P. G. Ipeirotis, H. V. Jagadish, N. Koudas, S. Muthukrishnan, L. Pietarinen, and D. Srivastava. Using q-grams in a DBMS for approximate string processing. *IEEE Data Engineering Bulletin*, 24(4):28–34, 2001.

[4] L. Gravano, P. G. Ipeirotis, N. Koudas, and D. Srivastava. Text Joins for Data Cleansing and Integration in an RDBMS. In *ICDE*, pages 729–731, 2003.

[5] L. Gravano, P. G. Ipeirotis, N. Koudas, and D. Srivastava. Text Joins in an RDBMS for Web Data Integration. In *WWW*, pages 90–101, 2003.

[6] Y. Huang and G. Madey. Web Data Integration Using Approximate String Join. In *WWW*, pages 364–365.

[7] L. Jin and C. Li. Selectivity Estimation for Fuzzy String Predicates in Large Data Sets. In *VLDB*, pages 397–408, 2005.

[8] N. Koudas, A. Marathe, and D. Srivastava. Flexible String Matching Against Large Databases in Practice. In *VLDB*, pages 1078–1086, 2004.

[9] A. Papoulis. *Probability, Random Variables and Stochastic Processes*. McGraw-Hill.

[10] The LDC Corpus Catalog, `http://wave.ldc.upenn.edu/Catalog/`.

CARDINALITY ESTIMATION FOR THE OPTIMIZATION OF QUERIES ON ONTOLOGIES

[1]E. Patrick Shironoshita, MSEE
patrick@infotechsoft.com

[1]Michael T. Ryan
mryan@infotechsoft.com

[1,2]Mansur R. Kabuka, Ph.D.
kabuka@infotechsoft.com

[1]INFOTECH Soft, Inc.
9200 S Dadeland Blvd., Suite 620
Miami, FL 33156, USA
+1 (305) 670 5111

[2]University of Miami
Coral Gables, FL 33124

ABSTRACT
An effective, accurate algorithm for cardinality estimation of queries on ontology models of data is presented. The algorithm relies on the decomposition of queries into query pattern paths, where each path produces a set of values for each variable within the result form of the query. In order to estimate the total number of result set parameters for each path, a set of statistics is compiled on the properties of the ontology. Experimental analysis has shown that the algorithm produces estimates with high accuracy and with high correlation to actual values. Thus, this algorithm can be used as the cornerstone of an effective optimization strategy for queries on diverse, heterogeneous data sources modeled as ontologies.

Categories and Subject Descriptors
H.2.4. [**Database Management**]: Systems – *query processing.*
I.2.4. [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods – *Representations*, *Semantic networks.*

General Terms
Algorithms, Performance, Standardization.

Keywords
Ontology, semantic query, query optimization, cardinality estimation, OWL, SPARQL.

1. INTRODUCTION
An ontology – the explicit specification of a conceptualization [5] – is a means of representing semantic knowledge, and includes at least a controlled vocabulary of terms, and some specification of their meaning [7]. The modeling of data sources as ontologies mapped to standardized representations is essential for defining correspondence among entities belonging to different sources, providing a semantically consistent, unified, and evolving view of data regardless of its storage formats and naming conventions, and resolving conflicts among sources [12].

All but the most trivial queries over such distributed, heterogeneous sources can be executed according to different plans, each of which may require vastly different amounts of time and computational resources. Thus, there is a growing need to develop mechanisms for the optimization of query execution. The objective of query optimization is to select an efficient query plan according to a cost function or cost model. The optimization cost function depends on the estimation of properties of the input data (cardinalities and constraints), and of the operating environment (CPU, disk access) [6]. Most query optimization strategies rely heavily on accurate cardinality estimation based upon statistics such as value histograms [9]. There is a substantial body of research that has been dedicated to the optimization problem in relational database management systems [1], and more recently, in XML data sources [4][16][17]. While no algorithms or methods have been found in the literature for query optimization over ontologies, query algebras suitable for performing algebraic optimizations of queries over RDF data models have been proposed [3][11].

In this paper we present an effective, accurate algorithm for cardinality estimation of queries posed against ontologies. Result set cardinality estimation is a critical component of query optimization, especially in the context of distributed data sources, where network traffic times constitute a substantial portion of total query execution time. The ability to decide, with a high degree of certainty, which possible query plan results in the least amount of network traffic is crucial to the performance of any information integration system based on ontologies.

2. ONTOLOGIES AND DATA MODELS
An ontology $\mathcal{O}$ can be conceptualized as the union of four sets: a set of classes C, a set of properties P, a set of individuals I, and a set of literals L:

$$\mathcal{O} = C \cup P \cup I \cup L \,. \quad (1)$$

We define the graph of the ontology, $G(\mathcal{O})$ as a set of triples that relate individuals to each other or to literals through properties, such that

$$G(\mathcal{O}) \subset I \times P \times (I \cup L). \quad (2)$$

Individuals in the ontology are necessarily instances of one or more classes in the ontology. Properties relate individuals to each other, or relate one individual to a literal; the former are called object properties, and the latter datatype properties.

The data model of an ontology is a subset of the total ontology comprised of classes and properties that have a direct relationship with the actual data source. A mechanism to create ontology data models from different types of data sources, including relational databases and XML filesystems, has been devised. In essence, it involves the creation or identification of a set of data classes C_D that correspond to all relevant element types within the structure of the data source, i.e., tables and columns in a relational database or node types in an XML document; and the conceptualization of individuals corresponding to each relevant element, that is, table cells in a database or nodes in XML. A class $c \in C_D$ if and only if there is a non-empty set of individuals I_c that are instances of class c and that are not instances of any subclass of c. Further, all individuals in the ontology must also belong to the set of individuals in the data model, and all individuals $i \in I$ are instances of one and only one class $c \in C_D$. A set of data properties P_D is also created to reflect actual relationships within the underlying data source, such as node edges in XML or table-column links in relational databases.

All individuals that belong to an ontology also belong to its data model. The data model of the ontology, then, can be defined as the union of the set of data classes, data properties, and individuals:

$$\mathcal{O}_D = C_D \cup P_D \cup I. \quad (3)$$

The classes and properties in the ontology that are not part of the data model may represent additional concepts or aggregations; these additional classes and properties are related to those in the data model through equivalence and subsumption relations.

3. QUERIES AND QUERY PATTERNS

The World Wide Web Consortium (W3C) has recently produced a Recommendation for a Web Ontology Language (OWL) [14], as a vocabulary extension to its Resource Description Framework (RDF) [8]. OWL is fast becoming the standard for ontology representation in the Web and beyond. We are particularly concerned with the DL fragment of OWL, where classes, properties, and individuals are disjoint sets. To address the need for querying over information modeled as ontologies, a number of RDF/OWL query languages have been developed [2][10][13], and the W3C has published a Last Call Working Draft for an RDF query language and protocol called SPARQL [11]. SPARQL is a declarative language, designed specifically to obtain information from RDF graphs.

In SPARQL, a query is a tuple consisting of a query pattern, a set of graphs, a set of solution modifiers, and a result form [11],

$$Q = \{QP, GS, SM, RF\}. \quad (4)$$

Queries in SPARQL are posed against a set of RDF graphs by attempting to match query patterns against this set of graphs, processing these matchings through solution modifiers, and preparing a result set according to the result form specified. The size or cardinality of this result set is given by the number of different individuals that match the variables specified by the result form. The objective of the algorithm presented here is to estimate this result set cardinality for the query Q, which we denote as $|Q|$.

For simplification purposes, the remainder of this paper deals with queries against a single graph, without solution modifiers, and with a result form that specifies a set of variable bindings to be returned. Extensions to this model to incorporate multiple graphs and modifiers is left for future work.

The basic query pattern in SPARQL is a set of triple patterns mixed with value constraints. The set of triple patterns is of the form

$$TP = (I \cup V) \times (P \cup V) \times (I \cup L \cup V). \quad (5)$$

where V is a set of variables. In other words, a triple pattern is a member of the graph of the ontology where zero or more of its three components is substituted by a variable. The three elements of a triple pattern are called the *subject*, *property*, and *object*, respectively.

Value constraints are Boolean expressions used to limit the allowable matchings of variables to literals or individuals in the ontology data set. If F denotes the set of possible Boolean expressions on values of the ontology, a conjunctive query pattern QP is a subset of the set of basic query patterns and value constraints, that is,

$$QP \subseteq (TP \cup F). \quad (6)$$

SPARQL defines two operators on query patterns besides conjunction: OPTIONAL and UNION. The cardinality of the result set of query patterns joined together through the UNION operator is the sum of the cardinalities of each pattern. Also, for cardinality estimation purposes, an OPTIONAL pattern can be conservatively substituted by a UNION and a conjunction: given two query patterns QP1 and QP2, the cardinality of QP1 OPTIONAL QP2 will be less than or equal to the cardinality of QP1 UNION (QP1.QP2). The rest of this paper deals only with conjunctive patterns.

A query pattern defined against an ontology $\mathcal{O}$ must first be re-written so that all terms in the query pattern refer to terms contained in the ontology data model $\mathcal{O}_D$. If a triple

pattern *t* in a query pattern contains a term that is not in the data model $\mathcal{O}_D$, and if *t* cannot be transformed into a set of semantically equivalent triple patterns that only contain terms in $\mathcal{O}_D$, then the result set of the evaluation of *t* is empty. The design of algorithms for query transformation from terms in $\mathcal{O}$ to terms in $\mathcal{O}_D$ is a matter for future work and outside the scope of this paper.

4. QUERY PATTERN PATHS

A triple *t = (s, p, o)*, where *o* is a literal or individual, can be substituted by a triple *t' = (s, p, v)* and a value constraint *(v = l)*, where *v* is a variable different from any other variable in the query pattern. A similar substitution can be made if the subject *s* is a literal or individual. Thus, after such substitutions, all triples in a query pattern can be considered to have only variables as subject and object.

A subset of a query pattern *QP* is considered a structural query pattern path for an ordered set of variables $W \subset V$ and an ordered set of properties or variables $R \subset (P \cup V)$, denoted *QPP(W,R)*, if it is of the form

$$QPP(W,R) = \left(v_1\ p_{12}\ v_2\ .\cdots.\ v_{n-1}\ p_{(n-1)n}\ v_n\right) \quad (7)$$

where $W=\{v_1 \ldots v_n\}$, $R=\{p_{12} \ldots p_{(n-1)n}\}$, and $\forall i,j,\ v_i \neq v_j$.

A maximal query pattern path $QPP^M(W,R)$ is one where there do not exist $W' \supset W$ and $R' \supset R$ such that $QPP(W',R') = QPP((W'-W),(R'-R)).QPP^M(W,R)$ is also a valid query pattern path. A literal query pattern path *LQPP(W,R)* is one where the last variable in the path matches with literals rather than individuals.

For a structural query pattern path *QPP(W,R)*, every $v_k \in W$ has a (possibly empty) set of value constraints $F(v_k)$ consisting of every $f(v_i) \in F$ such that there exists a query pattern path from some $v_k \in W$ to v_i which does not contain any triple pattern that is also in *QPP(W,R)*. The total set of value constraints for query pattern *QPP(W,R)* is then

$$F(QPP(W,R)) = \bigcup_{v_k \in W} F(v_k). \quad (8)$$

A complete query pattern path $QPP_C(W,R)$ is the union of a structural query pattern path and its total set of value constraints. The result set of a complete query pattern path, denoted as $RS(QPP_C(W,R))$, is given by the individuals that match the triples in the path, and where every variable $v_i \in W$ fulfills its set of value constraints. A complete maximal query path $QPP^M{}_C(W)$ is a complete query pattern path that contains a maximal query pattern path.

Given a query *Q* with query pattern *QP* and result form *RF*, the total result set according to *QP* for any variable *v* in *RF*, denoted as *RS(v,QP)*, is given by the intersection of the result sets for each complete maximal query path $QPP^M{}_C(W,R)$ where $v \in (W \cup R)$, that is

$$RS(v,QP) = \bigcap_{v \in (W \cup R)} RS(QPP_C^M(W,R)) \quad (9)$$

The cardinality of *RS(v,QP)* is upper-bound by the minimum cardinality of all $QPP^M{}_C(W,R)$.

The result set of a query *Q*, *RS(Q)*, is given by the union of the result sets for each of the variables $v_i \in RF$; the size of this result set, and therefore the cardinality of the query Q, is given by

$$|Q| = |RS(Q)| = \left| \bigcup_{v_i \in RF} RS(v_i, QP) \right|. \quad (10)$$

The problem of estimating the cardinality of a query, then, can be solved by finding accurate estimates of each possible maximal complete query pattern path on every variable *v* in *RF*.

5. CARDINALITY ESTIMATION

5.1 Estimation Function

We define an estimation function for the cardinality of a maximal complete query pattern path as a probability distribution:

$$E(QPP_C^M(W,R)) = e(QPP_C^M(W,R)) \pm k\Delta e(QPP_C^M(W,R)). \quad (11)$$

where *e* is the expected estimation value, Δe is a measure of potential error, and *k* is a tunable error factor. The actual estimate of cardinality is then calculated by choosing a *k* appropriate to the specific application: a *k* of zero chooses the most accurate estimate, but risks under-estimating cardinalities for query paths with large potential errors; a positive or negative *k* give a more conservative or more aggressive estimate, respectively.

We next consider the estimation of cardinality for two specific cases: the case where a maximal complete query pattern path is a structural path, i.e., it has no value constraints, and the case of a maximal complete query pattern path that is a lexical path with a set of constraints only on its last variable.

5.2 Estimation over Structural Query Pattern Paths

Structural query pattern paths define constraints only on the structure of the underlying data sources. To estimate cardinalities based on these structural constraints, statistics are kept on the properties comprising the data model.

For every property $p \in \mathcal{O}_D$, its total property cardinality, denoted as $|p|$, is defined as the total number of triples in the graph of the data model $G(\mathcal{O}_D)$ that contain *p*.

Further, for a given triple $t_{ab} = (i_a, p_{ab}, i_b)$ in the graph, the dependent property cardinality of a property p_{bc} with respect to t_{ab}, denoted $|t_{ab}, p_{bc}|$, is defined as the total number of triples $t_{bc} = (i_b, p_{bc}, i_c)$ in $G(\mathcal{O}_D)$ that follow

$t_{ab} = (i_a,p_{ab},i_b)$, where t_{bc} is said to follow t_{ab} if and only if the object of triple t_{ab} is equal to the subject of triple t_{bc}.

Given two properties p_i, p_j, then, the mean of the dependent property cardinality of p_i with respect to p_j is given by

$$\mu(p_i, p_j) = \frac{1}{|p_i|} \sum_{t_i \in \texttt{extent}(p_i)} |t_i, p_j| . \quad (12)$$

The extent of a property, denoted `extent(p)`, is defined as the set of triples (i_a,p,i_b) in the graph that contain p. The variance of the dependent property cardinality is given by

$$\sigma^2(p_i, p_j) = \texttt{abs}\left(\left(\frac{1}{|p_i|} \sum_{t_i \in \texttt{extent}(p_i)} |t_i, p_j|^2 \right) - \mu^2(p_i, p_j) \right) \quad (13)$$

where `abs` is the absolute value. The standard deviation $\sigma(p_i,p_j)$ of the dependent property cardinality is given by the positive square root of the variance.

The distribution function of p_i with respect to p_j gives a probability estimate of the number of triples containing p_j that follow each specific triple containing p_i, and is defined as

$$D(p_i, p_j) = \mu(p_i, p_j) \pm k\sigma(p_i, p_j) . \quad (14)$$

Given a query pattern path $QPP(W,R)$, if p_i denotes the ith element in R, then the distribution function of the path is the product of the distribution functions of each property in the path,

$$D(QPP(W,R)) = \prod_{p_i, p_{i+1} \in R} D(p_i, p_{i+1}) . \quad (15)$$

If p_1 is the first property in R, then the estimate of the total cardinality of each structural query pattern path, denoted $E(QPP(W,R))$, is

$$E(QPP(W,R)) = |p_1| D(QPP(W,R)) . \quad (16)$$

and thus

$$e(QPP(W,R)) = |p_1| \prod_{p_i, p_{i+1} \in R} \mu(p_i, p_{i+1}) . \quad (17)$$

$$\Delta e(QPP(W,R)) = ke(QPP(W,R)) \left[\sum_{p_i, p_{i+1} \in R} \left(\frac{\sigma^2(p_i, p_{i+1})}{\mu^2(p_i, p_{i+1})} \right) \right]^{1/2} . \quad (18)$$

5.3 Estimation over Value Constraints

Value constraints in SPARQL can be defined against variables that match to literals or individuals, or where the type of object to be matched is unknown. A particularly important case of value constraints on variables matched to individuals is the test of whether an individual is an instance of a class; in SPARQL, this is done through a triple pattern using the pre-defined property `rdf:type`. The cardinality of such a value constraint can be kept directly by calculating class cardinality statistics. Other value constraints defined against matchings to individuals defined in [11] have less incidence on result set cardinality; their incorporation into our estimation model remains for future work.

In a literal query pattern path $LQPP(W,R)$, the last property in the ordered set R, $p_{(n-1)n}$, is by definition a datatype property. To estimate the cardinality of a complete literal query pattern path $LQPP_C(W,R)$ consisting of a literal path $LQPP(W,R)$ and a set of value constraints $F(v_n)$ over the last variable in W, statistics are also kept on the distribution of values for the objects of these datatype properties, in the form of value histograms.

Histograms are widely used in relational database systems to represent the data distribution of values on a table column [1]. In the ontology data model $\mathcal{O}_D$, values are conceptualized as the objects of datatype properties; thus, for each datatype property $p \in \mathcal{O}_D$, an equi-depth histogram is constructed to represent its distribution. The total number of buckets B in each histogram is determined according to the following:

$$B = \texttt{min}(|p|_{unique}, B_{\max}, \frac{|p|}{d}) . \quad (19)$$

where as before $|p|$ is the total property cardinality of p, and $|p|_{\text{unique}}$ represents the total number of unique values in the range of p. Both B_{max} and d are tunable parameters of the histograms; B_{max} indicates the maximum number of buckets that should be constructed for any histogram (to avoid excessive memory and storage space consumption), and d indicates the expected depth of each frequency in the histogram. At a minimum, a number of buckets equal to the total number of unique values is created.

The set of value constraints $F(v_n)$ over a variable v_n is processed against the histograms for the property $p_{(n-1)n}$, resulting in a value ratio $\rho(F(v_n))$ that indicates the estimated proportion of the possible value bindings for variable v_n that satisfy the value constraints. Note that all values satisfy an empty value constraint, and thus, if $F(v_n)$ is the empty set, $\rho(F(v_n)) = 1$.

The total cardinality of a complete literal query pattern path is then given by the product of the estimate of cardinality for the literal query pattern path and the value ratio for the end variable in the query,

$$E(LQPP_C(W,R)) = \rho(F(v_n)) \cdot E(LQPP(W,R)) . \quad (20)$$

5.4 General Cardinality Estimation of Maximal Complete Query Pattern Path

In the general case, a maximal complete query pattern path $QPP^M{}_C(W,R)$ consists of a maximal query pattern path $QPP^M(W,R)$ and a set of value constraints $F(QPP(W,R))$. The total cardinality estimate of such a path is obtained by

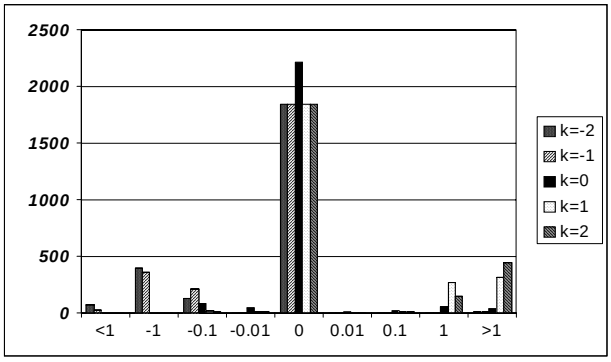


Figure 1. Histograms of estimated vs. real cardinality difference ratio

calculating the product of the cardinality estimate of its maximal query path with all the value ratios for every variable in the query pattern path:

$$E(QPP_C^M(W,R)) = E(QPP^M(W,R)) \cdot \prod_{v_k \in W} \rho(F(v_k)). \quad (21)$$

6. EXPERIMENTAL RESULTS

In order to validate the algorithm for cardinality estimation of semantic queries presented here, we obtained two XML data sets using the `eFetch` utility from the Entrez Gene website maintained by the National Center for Biotechnology Information (NCBI). Both datasets contains data from the `Gene` database; the dataset #1 contains data for a `geneId` of 2, while dataset #2 contains data for a sample set of six ATP-binding cassette transporter genes. Additionally, we also used an XML dataset obtained from a collection form in a mental health assessment system [15].

Each of these three datasets was odeled as an ontology in these cases, the data model of the ontology is equivalent to the ontology itself. Experimental analysis was then carried out over these models, to determine the validity of the cardinality estimation algorithm over both structural query pattern paths and value constraints.

6.1 Estimation over Structural Query Pattern Paths

To analyze the performance of the algorithm over structural constraints, cardinality was estimated for all possible query pattern paths containing between 2 and 4 properties, using error factor *k* of -2, -1, 0, 1, and 2. Each of these estimates was then compared with the real cardinality by taking the difference ratio, that is, the difference divided by the real cardinality value. Histograms of this difference ratio were obtained for each error factor, as shown in Figure 1. Aside from a few outlying values, in most cases this difference ratio is close to zero, indicating the accuracy of the estimation. The deviation of error factor from zero produces less accuracy, resulting in conservative estimates for positive error factors and aggressive estimates for negative factors.

Most importantly, the correlation between the actual and estimated total number of paths is very high, showing that the algorithm identifies, with high accuracy, the relative size between two query pattern paths. The correlation decreases as the error factor deviates from zero, also as expected; correlation also decreases, but is still very high, as the number of properties increases.

The degree of correlation between actual and estimate is highlighted in Table 1, where we present aggregate correlations for all possible query pattern paths in all datasets containing between 2 and 4 properties, and for all possible query plans in all datasets.

Table 1. Correlation between actual and estimated total number of paths

	Number of properties in path			
	2	**3**	**4**	**All**
Dataset #1	1.0000	0.9949	0.9785	0.9921
Dataset #2	1.0000	0.9997	0.9982	0.9994
Dataset #3	1.0000	0.9954	0.9214	0.9646
All datasets	1.0000	0.9994	0.9955	0.9985

6.2 Estimation over Value Constraints

To validate the use of histograms for the estimation of the cardinality of query pattern paths including value constraints, a total sampling of twenty different queries was chosen, including queries over ranges of values as well as queries over value equalities. A maximum bucket size of 5 was used, and in a few cases queries were redone using bucket sizes of 10 and 25. The aggregate results over this sampling are shown in Table 2. As can be observed, even with such a small number of buckets the estimates yield accurate results, and, more importantly, there is a high correlation between the estimates and the actual totals.

7. CONCLUSIONS AND FUTURE WORK

An algorithm for the estimation of cardinality for queries posed against ontology representations of data sources has been presented in this paper. Experimental results show that the proposed algorithm produces accurate estimates of cardinality, and more importantly, that the estimation of relative size among two queries is highly precise.

The estimation of the cardinality of a query is used to approximate the data transfer times of the result set, as part of the estimation of the total cost of executing a query. In a highly distributed architecture where data sources are located at diverse locations connected through the Internet, this is the most critical aspect of query execution time; however, in settings where very high speed connections are

Table 2. Statistics for estimation including value constraints

Number of different paths	20
Actual total number of paths	796
Estimated total number of paths	760
Correlation of estimated to actual	0.9863
Ratio of difference to total	
Average	0.03
Maximum	0.48
0.9 percentile	0.23
0.1 percentile	-0.20
Minimum	-0.42

available, other costs, such as CPU and disk access times, become significant; we are currently investigating the effects that the modeling of data sources as ontologies has on CPU execution times.

The estimation of the total cost of query execution is in turn used to select an optimal (or at least highly efficient) query plan. Algorithms for determining the set of different possible plans to be considered for selection are also currently under development. Work is also underway for the incorporation into our estimation model of multiple graphs and solution modifiers in a query, and of value constraints on matchings of variables to individuals.

A query optimization strategy is crucial to obtain reasonable performance over queries against ontology data models, especially if they are done over a highly distributed architecture. The algorithm proposed here is an important component for the construction of an efficient querying engine over ontology-modeled, distributed, heterogeneous data sources.

ACKNOWLEDGEMENTS

This work is supported by NIH grant R43RR018667. The authors also wish to acknowledge the contribution of Mr. Thomas Taylor and Dr. Akmal Younis of INFOTECHSoft, Inc.

REFERENCES

[1] Chauduri S. An Overview of Query Optimization in Relational Systems. Proc. of the 17th ACM Symp on Principles of Database Systems, Seattle, WA, USA, 1998:34-43.

[2] Chong EI, Das S, Eadon G, Srinivasan J. An efficient SQL-based RDF querying scheme. Proc. of the 31st Intl. Conf. on Very Large Data Bases, Trondheim, Norway. 2005:1216-1227.

[3] Frasincar F, Houben G-J, Vdovjak R, Barna P. RAL: An algebra for querying RDF. World Wide Web 2004;7(1):93-109.

[4] Freire J, Haritsa JR, Ramanath M, Roy P, Siméon J. StatiX: Making XML Count. Proc. ACM SIGMOD, 2002 Jun 4-6, Madison, WI, USA.

[5] Gruber TR. Toward principles for the design of ontologies used for knowledge sharing. Technical report KSL 93-04, Knowledge Systems Laboratory, Stanford University, Available from: ftp.ksl.stanford.edu/pub/KSL_Reports/KSL-93-04.ps.gz.

[6] Ives ZG, Halevy AY, Weld DS. Adapting to source properties in processing data integration queries. Proc ACM SIGMOD Intl Conf on Mgmt of Data. 2004:395-406.

[7] Kohler J, Philippi S, Lange M. SEMEDA: ontology based semantic integration of biological databases. Bioinformatics. 2003 Dec 12;19(18):2420-2427.

[8] Manola F, Miller E, editors. RDF Primer. W3C Recommendation [updated 2004 Feb 10, accessed 2006 Mar 21]. Available from: http://www.w3.org/TR/rdf-primer/.

[9] Markl V, Raman V, Simmen D, Lohman G, Pirahesh H, Cilimdzic M. Robust query processing through progressive optimization. Proc ACM SIGMOD Intl Conf on Mgmt of Data. 2004:659-670.

[10] Pérez de Laborda C, Conrad S. Querying Relational Databases with RDQL. Berliner XML Tage 2005. [Accessed 2006 Mar 21]. Available from: http://dbs.cs.uni-duesseldorf.de/~perezdel/pdf/05PeCob.pdf.

[11] Prud'hommeaux E, Seaborne A, editors. SPARQL Query Language for RDF, W3C Working Draft [updated 2007 Mar 26, accessed 2007 Jun 11]. Available from: http://www.w3.org/TR/rdf-sparql-query/.

[12] Rodriguez MA, Egenhofer MJ. Determining Semantic Similarity among Entity Classes from Different Ontologies. IEEE Trans on Knowledge and Data Eng, 2003 15(2):442-456.

[13] Sattler K-U, Geist I, Schallehn E. Concept-based querying in mediator systems. The VLDB Journal. 2005;14(1):97-111.

[14] Smith MK, Welty C, McGuiness DL editors. OWL Web Ontology Language Guide, W3C Recommendation [updated 2004 Feb 10, accessed 2006 Mar 21]. Available from: http://www.w3.org/TR/owl-guide/.

[15] Taylor TJ, Kabuka MR, Shironoshita EP, Ryan MT, Younis AA, John, NM, et.al. Viability of Mental Health Assessment Software in Diverse Settings. 45th Annual NCDEU (New Clinical Drug Evaluation Unit), Boca Raton, FL, USA. June 6-9, 2005.

[16] Wu Y, Patel JM, Jagadish HV. Structural Join Order Selection for XML Query Optimization. Proc. of the 19th Intl Conf on Data Engineering. 2003 Mar 5-8: 443-454.

[17] Zhang N, Ozsu MT, Aboulnaga A, Ilyas IF. XSeed: accurate and fast cardinality estimation for XPath queries. [Accessed 2006 Mar 21]. Available from: http://www.cs.uwaterloo.ca/~ilyas/papers/synopsis.pdf.

Navigational XPath: calculus and algebra

Balder ten Cate
ISLA – Informatics Institute
Universiteit van Amsterdam
balder.tencate@uva.nl

Maarten Marx
ISLA – Informatics Institute
Universiteit van Amsterdam
marx@science.uva.nl

ABSTRACT

We survey expressivity results for navigational fragments of XPath 1.0 and 2.0, as well as Regular XPath$^{\approx}$. We also investigate algebras for these fragments.

1. INTRODUCTION

XPath is a common fragment of the XML querying and processing languages XQuery and XSLT, used for navigation through XML documents. In this paper we address two foundational issues concerning this language: (1) its *expressivity* in comparison to the first-order logic, and (2) *algebras* for XPath.

We focus on the *navigational* part of XPath: the part that is concerned purely with document navigation, not considering operations involving strings, numbers, or any other types of atomic content. Several navigational fragments of XPath 1.0 and 2.0 have been proposed [10, 26]. All in all, we consider four navigational XPath dialects: Core XPath 1.0, variable-free Core XPath 2.0, Core XPath 2.0 with variables, and Regular XPath$^{\approx}$.

1.1 XML tree navigation using path expressions

Path expressions describe ways of navigating through XML documents, i.e., traveling from one node to another in the tree. This means we can model the meaning of a path expressions by a binary relation on the nodes of the tree. For example, the XPath 1.0 path expression `descendant::p` (abbreviated as `.//p`) denotes in any XML tree T, the set of all pairs (m, n) with n a descendant node of m that has tag name `p`. Of course, binary relations can be defined using many other formalisms, e.g., by means of a first-order formula in two free variables. In the case of this example, the binary relation is equivalently expressed by the conjunctive query

$$\phi(x, y) = \texttt{descendant}(x, y) \wedge \texttt{p}(y) \ .$$

Conversely, the conjunctive query

$$\phi(x, y) = \exists z_1 \dots z_n \bigwedge_{i=1}^{n} \begin{array}{l} \texttt{descendant}(x, z_i) \wedge \texttt{p}_i(z_i) \\ \wedge\, \texttt{descendant}(z_i, y) \wedge \texttt{q}(y) \end{array} \tag{1}$$

defines a binary relation that can be defined in XPath 1.0 by the union of the path expressions

$$\texttt{descendant} :: \texttt{p}_{\rho(1)} / \cdots / \texttt{descendant} :: \texttt{p}_{\rho(\texttt{n})} / \texttt{descendant} :: \texttt{q}$$

for all, exponentially many, permutations ρ of $1 \dots n$.

Next, consider the following first-order binary relation (familiar from temporal logic, and raising children):

$$\phi(x, y) = \texttt{descendant}(x, y) \wedge \texttt{q}(y) \wedge \forall z(\texttt{descendant}(x, z) \wedge \texttt{descendant}(z, y) \rightarrow \texttt{p}(z)) \tag{2}$$

A pair (m, n) stands in this relation if n is a descendant of m with tag name q and all nodes in-between m and n in the tree have tag name p. Can we express this in XPath 1.0?[1]

Questions such as these are hard to answer for languages as rich as full XPath 1.0 (whose technical specification is about 30 pages long). In order to be able to give a mathematically precise answer, in [19] the same question was studied in the context of *Core XPath 1.0* [10]. This is a compact, well defined fragment of XPath 1.0 with a clean logical semantics. It captures the navigational core of XPath 1.0, abstracting away from operations involving strings, numbers, or any other types of atomic content. It was shown in [19] that (2) cannot be defined in Core XPath.

1.2 Ways of extending Core XPath 1.0

Various extensions of XPath 1.0 have been proposed, including the official W3C standard of XPath 2.0. With more expressive power, new binary relations can be defined and sometimes older ones can be defined more succinctly. We give examples of both, starting with the latter.

*** Database Principles Column.** Column editor: Leonid Libkin, School of Informatics, University of Edinburgh, Edinburgh, EH8 9LE, UK. E-mail: libkin@inf.ed.ac.uk.

[1] Note that `.//q[not(ancestor :: *[not(self :: p)])]` does not define the intended relation: it is only correct for pairs (m, n) where m is the root.

XPath 2.0 has an `intersect` operator: `Path1 intersect Path2` denotes the intersection of the binary relations defined by `Path1` and `Path2`. Using intersection, (1) can be expressed without exponential blow-up:

$$\begin{aligned}&\texttt{descendant :: p}_1\texttt{/descendant :: q intersect}\\&\texttt{descendant :: p}_2\texttt{/descendant :: q intersect}\\&\quad\ldots\\&\texttt{descendant :: p}_n\texttt{/descendant :: q}\end{aligned}$$

Similarly, the previously undefinable "until" relation (2) can be defined in various ways using additional operators that have been proposed. A first possibility is to use the Kleene star, inspired by [1]:

$$(\texttt{child :: p})^*\texttt{/child :: q}.$$

Here $(\texttt{Path})^*$ denotes the reflexive transitive closure of the binary relation denoted by `Path`. The Kleene star does not belong to XPath 1.0 or 2.0, but extensions of XPath with this operator have been proposed and implemented [24, 7, 6]. A second solution is to use the *path complementation* operator `except` that was introduced in XPath 2.0:

```
descendant :: q except
  descendant :: *[not(self :: p)]/descendant :: q
```

Finally, a third option is to use quantified variables, which is possible in XPath 2.0 using the `for`-construct. Using `for`, we can write (2) as follows:

```
for $s in . return
  descendant :: q[not(ancestor :: *[not(self :: p)]/
                      ancestor :: *[. is $s])]
```

Notice how the variable `$s` stores the initial node.

1.3 Two main questions of this paper

In this paper, we consider Core XPath 1.0 and three extensions of it, roughly corresponding to XPath 2.0, the variable free fragment of XPath 2.0, and an extension of XPath with transitive closure and path equalities. For each of these, we study two main questions: *what is the expressive power* and *what are suitable algebras*.

Expressivity and Codd completeness.
When a new query language is introduced, it is always useful to compare its expressive power to existing languages. E.F. Codd did this for SQL and relational algebra by showing that they are equally expressive as first-order logic [4]. With the navigational languages for XML we can do the same: given a dialect of XPath, we can ask how it compares to (fragments or extensions of) first-order logic. We explore this in Section 3.

Algebras for navigational XPath.
An important step towards efficient query evaluation is to identify a suitable algebra in which query plans can be formulated. Which algebra are suitable for our XPath dialects? In answering this question, we guide ourselves by the following criteria:

Table 1: Syntax of Core XPath 1.0.

Axis	:=	`self` | `child` | `parent` | right | left
		| `descendant`
		| `ancestor`
		| `following`
		| `preceding`
		| `following_sibling`
		| `preceding_sibling`
NameTest	:=	QName | *
Step	:=	Axis::NameTest .
PathExpr	:=	Step
		| PathExpr/PathExpr
		| PathExpr `union` PathExpr
		| PathExpr[NodeExpr]
NodeExpr	:=	PathExpr
		| `not` NodeExpr
		| NodeExpr `and` NodeExpr
		| NodeExpr `or` NodeExpr.

1. expressions in the XPath dialect should be efficiently translatable to algebraic expressions,
2. the algebra should not be much more expressive than the XPath dialect requires,
3. the algebra should not have much harder query evaluation or equivalence problem than the XPath dialect itself, and
4. there should be a nice set of algebraic equivalence rules for the algebra.

In Section 4, we will consider several candidates, such as Codd's relational algebra (CRA) and Tarski's algebra of binary relations (TRA). For each dialect of navigational XPath, a different algebra turns out to fit best.

2. PRELIMINARIES: FOUR DIALECTS OF NAVIGATIONAL XPATH

In this section, we review the syntax and semantics of Core XPath 1.0 —the navigational fragment of XPath 1.0 introduced in [10]— as well as three extensions.

Core XPath 1.0.
Core XPath 1.0 was introduced in [10] to capture the navigational core of XPath 1.0. The definition we will give here is from [18], which differs from the one of [10] as (1) it include the "one-step sibling axes" left, right (which are definable in XPath 1.0 using numerical predicates), (2) filters can be applied to any expression, and (3) we include the union operator on path expressions.

Table 1 gives the syntax of Core XPath 1.0. Here QName stands for any XML tag name. The primary type of expression is a *path expression* (PathExpr). Table 2 gives the semantics. Expressions are evaluated

Table 2: Semantics of Core XPath 1.0.

$$
\begin{array}{lcl}
[\![\mathsf{Axis} :: N]\!]_{\mathsf{PExpr}} & = & \{(x,y) \mid x\mathsf{Axis}y \text{ holds in the tree, and } y \text{ has tag } N\} \\
[\![\mathsf{Axis} :: *]\!]_{\mathsf{PExpr}} & = & \{(x,y) \mid x\mathsf{Axis}y \text{ holds in the tree}\} \\
[\![R/S]\!]_{\mathsf{PExpr}} & = & [\![R]\!]_{\mathsf{PExpr}} \circ [\![S]\!]_{\mathsf{PExpr}} \\
[\![R \texttt{ union } S]\!]_{\mathsf{PExpr}} & = & [\![R]\!]_{\mathsf{PExpr}} \cup [\![S]\!]_{\mathsf{PExpr}} \\
[\![R[T]]\!]_{\mathsf{PExpr}} & = & \{(x,y) \mid (x,y) \in [\![R]\!]_{\mathsf{PExpr}} \text{ and } y \in [\![\mathtt{T}]\!]_{\mathsf{NExpr}}\} \\
\\
[\![\mathsf{PathExpr}]\!]_{\mathsf{NExpr}} & = & \{x \mid \exists y.(x,y) \in [\![\mathsf{PathExpr}]\!]_{\mathsf{PExpr}}\} \\
[\![\texttt{not } T]\!]_{\mathsf{NExpr}} & = & \{x \mid x \notin [\![T]\!]_{\mathsf{NExpr}}\} \\
[\![T_1 \texttt{ and } T_2]\!]_{\mathsf{NExpr}} & = & [\![T_1]\!]_{\mathsf{NExpr}} \cap [\![T_2]\!]_{\mathsf{NExpr}} \\
[\![T_1 \texttt{ or } T_2]\!]_{\mathsf{NExpr}} & = & [\![T_1]\!]_{\mathsf{NExpr}} \cup [\![T_2]\!]_{\mathsf{NExpr}}
\end{array}
$$

on finite sibling-ordered unranked trees whose nodes are labeled by XML tag names. Given such a tree, the meaning $[\![R]\!]_{\mathsf{PExpr}}$ of a PathExpr R is always a binary relation. This is just another, equivalent, way of specifying a function from nodes to sets of nodes (the answer-set semantics). The meaning $[\![T]\!]_{\mathsf{NExpr}}$ of a node expression T is always a set of nodes.

We will study the complexity of two tasks: query evaluation and query containment. For *query evaluation*, we will consider the *combined complexity* of the following problem: given a path expression, an XML-tree (suitably encoded) and a pair of nodes, determine whether the pair belongs to the relation denoted by the path expression. In the case of the *query containment* problem, the task is to determine, given two path expressions R, S, whether in every tree model, $[\![R]\!]_{\mathsf{PExpr}} \subseteq [\![S]\!]_{\mathsf{PExpr}}$. For Core XPath 1.0, the query evaluation problem for Core XPath 1.0 is in PTIME (in fact, it can be performed in linear time) [10], and the query containment problem is EXPTIME-complete [20, 17].

Core XPath 2.0 without variables.

In [26], Core XPath 2.0 was introduced as a navigational core of XPath 2.0 with a clean, logical semantics. One important simplifying assumption underlies Core XPath 2.0, namely that path expressions still denote *binary relations between nodes*, as they did in Core XPath 1.0. This is not the case in the full XPath 2.0, where they denote functions from nodes to sequences of nodes (not necessarily in document order and possibly containing duplicates). We follow the definition of Core XPath 2.0 from [26].

First, we consider the variable-free fragment of Core XPath 2.0. This is a very simple extension of Core XPath 1.0: it differs from Core XPath 1.0 only in that one can take intersections and complements of path expressions:

$$
\begin{array}{lcl}
[\![R \texttt{ intersect } S]\!]_{\mathsf{PExpr}} & = & [\![R]\!]_{\mathsf{PExpr}} \cap [\![S]\!]_{\mathsf{PExpr}} \\
[\![R \texttt{ except } S]\!]_{\mathsf{PExpr}} & = & [\![R]\!]_{\mathsf{PExpr}} \setminus [\![S]\!]_{\mathsf{PExpr}}.
\end{array}
$$

These operators do not only increase the expressive power of the language (as we will see in the next section), they also greatly increase its complexity. The query evaluation problem for variable free Core XPath 2.0 is still in PTIME (in fact, it can be performed in quadratic time), but the query containment problem is non-elementary (2-EXPTIME-complete for expressions without the complementation operator) [25].

Core XPath 2.0.

Besides the addition of the `intersect` and `except` operators, an important difference between XPath 1.0 and 2.0 is the use of quantified variables by means of the `for` construct. Formally, let a NodeRef expression be an expression of the form $\$i$ or . (where $\$i$ is a variable ranging over nodes in the tree). Then the syntax of full Core XPath 2.0 is obtained by extending the syntax of Core XPath 1.0 with the `intersect` and `except` operators from above, with path expressions of the form $\$i$ and `for` $\$i$ `in` PathExpr `return` PathExpr, and with node expressions of the form NodeRef `is` NodeRef. The latter tests whether the two expressions refer to the same node.

Since the expressions of Core XPath 2.0 can contain variables, the semantic interpretation is relative to an *assignment*, i.e., a function mapping variables to nodes. For g an assignment, $\$i$ a variable, and x a node, $g[\$i \mapsto x]$ denotes the assignment g' which is identical to g except that $g'(i) = x$. Also, for any assignment g, node x, and NodeRef expression a, let $[\![a]\!]^{g,x}$ be $g(a)$ in case a is a variable, or x in case a is '.'. The semantics of the new constructs is as follows:

$$[\![\$i]\!]^g_{\mathsf{PExpr}} = \{(x,y) \mid g(i) = y\}$$

$$[\![\texttt{for } \$i \texttt{ in } R \texttt{ return } S]\!]^g_{\mathsf{PExpr}} = \{(x,y) \mid \exists z((x,z) \in [\![R]\!]^g_{\mathsf{PExpr}} \text{ and } (x,y) \in [\![S]\!]^{g[\$i \mapsto z]}_{\mathsf{PExpr}})\}$$

$$[\![a \texttt{ is } b]\!]_{\mathsf{NExpr}} = \{x \mid [\![a]\!]^{g,x} = [\![b]\!]^{g,x}\}.$$

The query evaluation problem for Core XPath 2.0 is PSPACE-complete, and the query containment problem is non-elementary [25].

Regular XPath$^{\approx}$.

Regular XPath$^{\approx}$ extends Core XPath 1.0 with two operators that are not part of XPath 1.0 or 2.0, and that, as we will see, make it more expressive. The most important of these is the Kleene star, which allows us to take the reflexive transitive closure of arbitrary path expressions. The other is *path equalities* (not to be confused with data value equalities). Formally, the semantics of these operators is as follows [24]:

$$[\![R^*]\!]_{\mathsf{PExpr}} = \text{reflexive transitive closure of } [\![R]\!]_{\mathsf{PExpr}}$$

$$[\![R \approx S]\!]_{\mathsf{NExpr}} = \{x \mid \exists y.(x,y) \in [\![R]\!]_{\mathsf{PExpr}} \cap [\![S]\!]_{\mathsf{PExpr}}\}$$

Regular XPath$^{\approx}$ can be viewed as a mix between Core XPath 1.0 and *regular path expressions* [1]: it has the filter expressions of the former and the Kleene star of the latter. It is still mainly studied in the theoretical community [9, 24, 7].

The query evaluation problem for Regular XPath$^{\approx}$ is

in PTIME (in fact, in quadratic time), and the query containment problem is EXPTIME-complete [25].

3. EXPRESSIVITY OF XPATH DIALECTS

We have defined four XPath fragments. How do they compare in terms of expressivity and succinctness? We will answer this question by mapping each XPath dialect to an equally expressive variant of first-order logic.

Since the data model of an XML document is a finite sibling ordered tree, it is natural to consider first-order logic in the signature with eight atomic binary relations corresponding to the basic axes (`child`, `parent`, `left` and `right`, and their transitive closures `descendant`, `ancestor`, `following-sibling` and `preceding-sibling`) plus a unary predicate for each tag name. We will call the first order language in this signature FO_{tree}. With $FO_{\mathsf{tree}}(x)$ and $FO_{\mathsf{tree}}(x, y)$ we denote the FO_{tree} formulas in one and two free variables, respectively.

Besides looking at expressive power, we will also compare different languages in terms of *succinctness*. As usual, if two languages, L and L', are equally expressive, we say that L is *(at least) exponentially more succinct than* L' if there is a infinite sequence of L-expressions $R_1, R_2, \ldots$ where the length of R_k is polynomial in k, such that for every sequence of equivalent L'-expressions $R'_1, R'_2, \ldots$, the length of R'_k is exponential in k. Similarly, one can say that a language is *non-elementarily more succinct* than another language.

The results from this section are summarized in Table 3. These results hold both for path expressions and for node expressions.

The results discussed in this section naturally build on a existing line of research in temporal logic, which originates in the work of H. Kamp [15] and which studies expressive completeness for various temporal logics on trees. A survey of this area may be found in [13].

Core XPath 1.0

As we have already seen in Section 1, not every FO_{tree}-definable binary relation is definable in Core XPath 1.0. However, we can define a natural fragment of FO_{tree} with respect to which Core XPath 1.0 is complete.

Let $\exists FO^{(\mathrm{mon}\neg)}_{\mathsf{tree}}$ be the fragment of FO_{tree} where negation can only be applied to subformulas with exactly one free variable, and universal quantification is disallowed altogether (thus, the connectives are conjunction, disjunction, and existential quantification, plus negation of formulas with at most one free variable). It can be seen from Table 2 that Core XPath 1.0 path expressions can be translated into this fragment of FO_{tree} (indeed, the only form of negation present in Core XPath 1.0 is negation in filter expressions, which corresponds to negation of a formula in one free variable). A converse translation is possible as well, although it involves an exponential blow-up (recall the example we gave in the introduction):

Theorem 1 (Core XPath 1.0 $\equiv \exists FO^{(\mathbf{mon}\neg)}_{\mathsf{tree}}(x, y)$)

1. *There is a linear translation from Core XPath 1.0 path expressions to $\exists FO^{(mon\neg)}_{\mathsf{tree}}(x, y)$ formulas, and an exponential translation backwards.*
2. *Indeed, $\exists FO^{(mon\neg)}_{\mathsf{tree}}(x, y)$ formulas are exponentially more succinct than Core XPath 1.0 path expressions.*

PROOF. The difficult direction of (1) can be proved by induction on the nesting depth of negation, using the fact that positive existential fist-order formulas can be translated to Core XPath 1.0 path expressions at the cost of an exponential blowup [2, 11]. For the exponential difference in succinctness, see [25, Thm. 26]. □

An alternative characterization of Core XPath 1.0, in terms of conjunctive queries and the two-variable fragment of FO_{tree}, is given in [19].

Core XPath 2.0

In the case of Core XPath 2.0, there is a precise match with FO_{tree}, in terms of expressive power. In fact, this Codd-completeness has been one of the design considerations for XPath 2.0 [16]. Moreover, it turns out to hold already for the variable free fragment. Still, the presence of variables matters for the succinctness of the language.

For simplicity, we consider only path expressions that have no *free variables*. For a discussion of expressive completeness in the presence of free variables, see [8].

Theorem 2 (Core XPath 2.0 $\equiv FO_{\mathsf{tree}}(x, y)$)

1. *There are linear translations between Core XPath 2.0 path expressions and $FO_{\mathsf{tree}}(x, y)$ formulas.*
2. *There is a linear translation from variable free Core XPath 2.0 path expressions to $FO_{\mathsf{tree}}(x, y)$ formulas and a non-elementary translation backwards.*
3. *$FO_{\mathsf{tree}}(x, y)$ formulas are at least exponentially more succinct than variable free Core XPath 2.0 path expressions.*

PROOF. The linear translations are straightforward. A non-elementary translation from FO_{tree} to variable free Core XPath 2.0 is given in [18]. The exponential difference in succinctness between FO_{tree} and variable free Core XPath 2.0 holds already on linear orders (i.e., documents in which each node has at most one child) [12]. □

In fact, it was shown in [18] that a more modest extension of Core XPath 1.0 called *Conditional XPath* is already expressively complete for FO_{tree}. It extends Core XPath 1.0 with "conditional axes" of the form (Axis `while` NodeExpr), with Axis $\in \{\texttt{child}, \texttt{parent}, \texttt{left}, \texttt{right}\}$. Without going into further details, we only mention that (Axis `while` T) :: N can be written in Core XPath 2.0 as

$$\mathsf{Axis}^+ :: N \ \texttt{except}\ (\mathsf{Axis}^+ :: *[\texttt{not}(T)]/\mathsf{Axis}^+ :: *)$$

where Axis^+ is the transitive version of Axis.

Table 3: Expressivity and succinctness of XPath dialects.

XPath dialect	Core XPath 1.0	$\subsetneq$	Variable-free Core XPath 2.0	$\equiv$	Core XPath 2.0	$\subsetneq$	Regular XPath$^{\approx}$
Equivalent FO-dialect	$\exists FO^{\text{mon}\neg}_{\mathsf{tree}}$ (exponential succinctness gap)		FO_{tree} (at least exponential succinctness gap)		FO_{tree} (no succinctness gap: linear translations)		FO^{*}_{tree} (non-elementary succinctness gap)

Regular XPath$^{\approx}$

Since the conditional axes of [18] are definable in Regular XPath$^{\approx}$ using the Kleene star — `(Axis while` T`)::`N is equivalent to `(Axis::`$*$ `[not(`T`)])*/Axis::`N — we already know by [18] that Regular XPath$^{\approx}$ extends FO_{tree} in expressive power. In order to give a precise characterization of the expressive power of Regular XPath$^{\approx}$, we must consider an extension of FO_{tree}.

The simplest option is to simply extend FO_{tree} with a Kleene star (i.e., a transitive closure operator for binary relations). Thus, let $FO^{*}_{\mathsf{tree}}(x, y)$ be the extension of FO_{tree} (x, y) with a transitive closure operator that applies to formulas with exactly two free variables. Then the following is proved in [24] and [25, Thm. 27]:

Theorem 3 (Regular XPath$^{\approx}$ $\equiv FO^{*}_{\mathsf{tree}}$)

1. *There is a linear translation from Regular XPath$^{\approx}$ path expressions to $FO^{*}_{\mathsf{tree}}(x, y)$ formulas, and a non-elementary translation backwards.*
2. *In fact, $FO^{*}_{\mathsf{tree}}(x, y)$ formulas are non-elementarily more succinct than Regular XPath$^{\approx}$ path expressions.*

Incidentally, FO^{*}_{tree} is not the same as $FO_{\mathsf{tree}} + TC^{1}$: the standard unary transitive closure operator TC^{1} can be applied to formulas containing more than two free variables, as long as two of the variables are designated; the others are treated as parameters (cf. for instance [5]). We do not know at present whether FO^{*}_{tree} and $FO_{\mathsf{tree}} + TC^{1}$ have the same expressive power on trees.

4. ALGEBRAS FOR XPATH DIALECTS

The previous section showed that Core XPath 2.0 corresponds in expressive power to exactly first-order logic. The next question is which algebras are appropriate for representing query plans for Core XPath 2.0 expressions. The same question holds for the other dialects we discussed. Codd's relational algebra seems a natural choice because it is again equally expressive as first-order logic. Indeed, we will see that it is a good choice when considering Core XPath 2.0. For other XPath dialects however (including the variable free fragment of Core XPath 2.0), there are better options.

In Section 1.3 we gave criteria for determining whether an algebra is suitable for an XPath dialect. In this section, we discuss four different algebras, and determine which ones match best with each XPath dialect. The results are summarized in Table 5.

To simplify the presentation, we will first consider Core XPath 1.0 and 2.0, and only afterward Regular XPath$^{\approx}$, as the latter requires (a mild form of) recursion in the algebra.

4.1 Four candidate algebras

Codd's relational algebra (CRA) and its fragment CRA($mon\neg$)

We briefly recall Codd's relational algebra. A characteristic feature of this algebra is that it is *many sorted*: each expression has an associated *arity* corresponding to the number of columns of the table it computes. The atomic expressions are simply the names of the relations in the database, and the operations are *selection* (σ), *projection* (π), *cross-product* ($\times$), *union* ($\cup$) and *complementation* ($-$).

The fact that there is no bound on the arity of the expressions has some negative consequences on the complexity of query evaluation: it is PSPACE-complete, whereas it becomes polynomial if there is a bound on the allowed arity of (sub)expressions [3, 28].

Inspired by the results in the previous section, it makes sense to distinguish another restricted fragment of CRA, namely $CRA(mon\neg)$. This fragment is obtained by restricting the use of complementation to unary tables. Note that all $SPCU$-expressions still belong to this fragment.

Tarski's relation algebra (TRA)

Tarski's relation algebra [22, 23] is an algebra of binary relations: each expression denotes a table with precisely two columns. The operations on binary relations considered by Tarski are the Boolean operations (union, intersection and complementation), as well as composition $\circ$ and converse $(\cdot)^{-1}$. There are also two constants (or, 0-ary operations) $\top$ and ϵ, which stand for the total relation and the identity relation (over the given domain). A typical example of an equivalence in this algebra is $\alpha \circ (\beta \cup \gamma) \equiv \alpha \circ \beta \cup \alpha \circ \gamma$.

It was shown in [23] that TRA has the same expressive power as the three-variable fragment of first-order logic in two free variables, over vocabularies consisting of binary relations only.

Although in TRA all expressions denote binary relations, unary relations can be easily dealt with as

well, for instance by treating them as subrelations of the identity relation (e.g., $\{a, b, c\}$ can be treated as $\{(a, a), (b, b), (c, c)\}$).

Dynamic relation algebra (DRA)

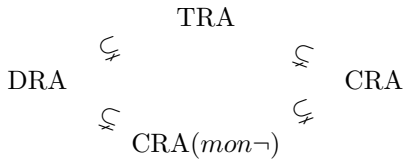
In [27, 14], a reduct of Tarski's relation algebra is studied containing only the operations $\cup$, $\circ$ and $\sim$. The latter of these is called the *counterdomain* operation. It takes a binary relation R and produces a subrelation of the identity relation: $\sim R$ denotes $\{(x, y) \mid x = y$ and $\neg\exists z.(x, z) \in R\}$. In TRA, it can be expressed as $\epsilon - (R \circ \top)$. This operator is quite handy: e.g., $\sim$`child` expresses "I am a leaf node", and $\sim\sim$`child` expresses "I am *not* a leaf node". We call this algebra *dynamic relation algebra* (DRA).

The signature of DRA might seem poor, but it is rich enough to capture all of Core XPath 1.0. If we encode properties of nodes as subrelations of the identity relation (as we already suggested above), then we have the following translation:

$$
\begin{array}{lcl}
\mathsf{TR}_{\mathsf{PExpr}}(\mathsf{Axis} :: *) & = & \mathsf{Axis} \\
\mathsf{TR}_{\mathsf{PExpr}}(\mathsf{Axis} :: N) & = & \mathsf{Axis} \circ N \\
\mathsf{TR}_{\mathsf{PExpr}}(R/S) & = & \mathsf{TR}_{\mathsf{PExpr}}(R) \circ \mathsf{TR}_{\mathsf{PExpr}}(S) \\
\mathsf{TR}_{\mathsf{PExpr}}(R \texttt{ union } S) & = & \mathsf{TR}_{\mathsf{PExpr}}(R) \cup \mathsf{TR}_{\mathsf{PExpr}}(S) \\
\mathsf{TR}_{\mathsf{PExpr}}(R[T]) & = & \mathsf{TR}_{\mathsf{PExpr}}(R) \circ \mathsf{TR}_{\mathsf{NExpr}}(T) \\
\mathsf{TR}_{\mathsf{NExpr}}(\mathsf{PathExpr}) & = & \sim\sim \mathsf{TR}_{\mathsf{PExpr}}(\mathsf{PathExpr}) \\
\mathsf{TR}_{\mathsf{NExpr}}(\texttt{not } T) & = & \sim \mathsf{TR}_{\mathsf{NExpr}}(T) \\
\mathsf{TR}_{\mathsf{NExpr}}(T_1 \texttt{ and } T_2) & = & \mathsf{TR}_{\mathsf{NExpr}}(T_1) \circ \mathsf{TR}_{\mathsf{NExpr}}(T_2) \\
\mathsf{TR}_{\mathsf{NExpr}}(T_1 \texttt{ or } T_2) & = & \mathsf{TR}_{\mathsf{NExpr}}(T_1) \cup \mathsf{TR}_{\mathsf{NExpr}}(T_2)
\end{array}
$$

In [27], an elegant model theoretic characterization of DRA is given in terms of *safety for bisimulations*.

DRA is a fragment of both TRA and CRA($mon\neg$). More precisely, the relationships between the four algebras on arbitrary models are as follows:



When we restrict attention to XML documents (i.e., where the atomic relations are the 8 binary relations corresponding to the different axes, as well a "unary" relation for each of the different tag names), the situation is a bit different: on this restricted class of models TRA and CRA have the same expressive power, as do DRA and CRA($mon\neg$).

4.2 Complexity of these algebras on trees

We will now discuss the complexity of *query evaluation* and *query containment* for the four algebras interpreted on XML-trees (i.e., where the atomic relations are the 8 binary relations corresponding to the different axes, as well a "unary" relation for each of the different tag names). As before, in the case of query evaluation we consider the combined complexity of testing whether a given pair belongs to the relation defined by a given path expression on a given XML document. Table 4 provides a summary of the results.

Table 4: Complexity of evaluation and containment for the algebras on trees.

	Evaluation	*Containment*
CRA	PSPACE-compl.	Non-elementary
$CRA(mon\neg)$	NP-hard, in P^{NP}	2-EXPTIME-compl.
TRA	PTIME (quadratic)	Non-elementary
DRA	PTIME (linear)	EXPTIME-compl.

Containment.

By Rabin's theorem, query containment is decidable for all four algebras. For TRA and CRA, containment is non-elementary, as follows from Stockmeyer's non-elementary lower bound for the non-emptiness problem of star-free expressions [21, 25]. The results for CRA($mon\neg$) and DRA follow from known results about XPath. In particular, the 2-EXPTIME-hardness of CRA($mon\neg$) query containment follows from the same lower bound for Core XPath 1.0 extended with path intersection, as the latter can be linearly translated into CRA($mon\neg$). The upper bound follows from the existence of a singly exponential translation from CRA($mon\neg$)-expressions of arity 2 to Core XPath 1.0, and the fact that Core XPath 1.0 has an EXPTIME-complete query containment problem (the restriction to expressions of arity 2 is not essential: containment of CRA($mon\neg$)-expressions of arity greater than 2 can be linearly reduced to containment of ones of arity 2, in fact to Boolean CRA($mon\neg$)-expressions) [25]. The result for DRA follows from linear translations to Core XPath 1.0.

Evaluation.

The combined complexity of query evaluation for CRA is PSPACE-complete, also when restricted to XML-trees [3]. As TRA corresponds to a fixed variable fragment of first-order logic the complexity drops to PTIME. Using the bottom-up algorithm sketched in [28] it can be shown to be in $O(n^2)$. In [10], it is shown that query evaluation for Core XPath 1.0 can be performed in linear time. Because Core XPath 1.0 and DRA linearly translate to each other, the result transfers to DRA. Recall that we are not talking about the complexity of computing the relation denoted by a path expression (which could be quadratic in the size of the tree), but of the complexity of checking whether a given pair of nodes belongs to the denotation of a given expression in a given tree. Query evaluation for CRA($mon\neg$) is NP-hard: this holds even for positive conjunctive queries with only downward axis relations [11]. For the P^{NP}-upperbound, we use an algorithm that runs in polynomial time and that uses an oracle for testing whether a tuple belongs to the answer set of an SPCU-expression. The algorithm proceeds roughly as follows: given an expression α, it starts by listing all subexpressions whose main connective is a (unary) complementation operator, in order of growing length.

Table 5: Which algebra for which XPath dialect?

	CRA	CRA($mon\neg$)	TRA	DRA
Core XPath 1.0	**Y** (linear translation) **N** (too expressive) **N** (complexity too high)	**Y** (linear translation) **Y** (same expressivity) **N** (complexity too high)	**Y** (linear translation) **N** (too expressive) **N** (complexity too high)	**Y** (linear translation) **Y** (same expressivity) **Y** (same complexity)
Core XPath 2.0 w/o variables	**Y** (linear translation) **Y** (same expressivity) **N** (complexity too high)	**N** (no translation possible) **N** (too little expressivity) **N** (complexity too high)	**Y** (linear translation) **Y** (same expressivity) **Y** (same complexity)	**N** (no translation possible) **N** (too little expressivity) **Y** (lower complexity)
Core XPath 2.0 with variables	**Y** (linear translation) **Y** (same expressivity) **Y** (same complexity)	**N** (no translation possible) **N** (too little expressivity) **Y** (lower complexity)	**N** (no elem. translation) **Y** (same expressivity) **Y** (same complexity)	**N** (no translation possible) **N** (too little expressivity) **Y** (lower complexity)
	CRA($*$)	CRA($mon\neg, *$)	TRA($*$)	DRA($*$, loop)
Regular XPath$^{\approx}$	**Y** (linear translation) **Y** (same expressivity) **N** (complexity too high)	**Y** (linear translation) **Y** (same expressivity) **N** (complexity too high)	**Y** (linear translation) **Y** (same expressivity) **N** (complexity too high)	**Y** (linear translation) **Y** (same expressivity) **Y** (same complexity)

One by one, it computes for each such subexpression α the (polynomially large) answer set, by asking the oracle for each element whether it belongs to the answer set. The occurrences of α within larger expressions are then replaced by the computed answer set. Finally, we are left with a single SPCU-expression, to which the oracle is once more applied.

4.3 Axiomatizations

One of our criteria for being a good algebra was the availability of an axiomatization of the valid equations on XML-trees (finite sibling ordered node-labeled trees). Only a few results are known here. In [2], an axiomatization is given for the $\sim$-free reduct of Dynamic Relation Algebras DRA with only the two downward axis plus atomic label tests. An axiomatization of the full DRA on XML-trees is not known (a complete axiomatization on arbitrary models is given in [14]). In [26], an axiomatization of first-order logic on XML-trees is given, from which an axiomatization for TRA on XML-trees is derived. We believe that in a similar way an axiomatization of CRA on XML-trees can be found. The TRA axiomatization consists of general axioms for the TRA similarity type like $R \circ (S \circ T) = (R \circ S) \circ T$ plus special axioms which are only valid on trees. Two examples are Tr5 and Tr11:

$$\text{Tr5.} \quad \downarrow^+ \circ \uparrow^+ \quad \equiv \quad \downarrow^+ [\downarrow] \cup \epsilon[\downarrow] \cup (\epsilon[\downarrow] \circ \uparrow^+)$$

$$\text{Tr11.} \quad \epsilon \cup \uparrow^+ \cup \downarrow^+ \cup (\uparrow^* \circ \rightarrow^+ \circ \downarrow^*) \cup (\uparrow^* \circ \leftarrow^+ \circ \downarrow^*) \quad \equiv \quad \top$$

Here we abbreviate the steps in the trees by arrows, e.g., $\downarrow$ is the `child` axis, $\uparrow$ is `parent`, etc. E.g., in XPath notation, the left-hand side of Tr5 would be `descendant/ancestor`. Also, we use $R[S]$ as a shorthand for $R \circ \sim\!\sim S$. Tr5 is a natural complexity reducing equivalence when read from left to right. Tr11 states the well known fact that the self, ancestor, descendant, following and preceding axis relations partition each XML-tree from every given node.

4.4 Which algebra for which XPath?

We now have three XPath dialects (Regular XPath$^{\approx}$ will be dealt with in the next subsection) and four candidate algebras. We determine which algebra fits best to which fragment by answering the following questions, corresponding to the first three requirements from Section 1.3:

1. Is there a linear translation from the expressions in XPath dialect to expressions in the algebra?
2. Are the XPath dialect and the algebra equally expressive?
3. Do the XPath dialect and the algebra have the same query containment and evaluation complexities?

(as for the fourth requirement, concerning the existence of nice sets of algebraic equivalence rules for the algebra, we have too little information at present to say much about it).

The answers, based on the results discussed in the previous sections, are given in Table 5. The combinations with only affirmative answers are marked by a gray background.

4.5 Regular XPath$^{\approx}$

For Regular XPath$^{\approx}$, the algebras need to be extended with a transitive closure operator. In the case of TRA and DRA, the semantics of such an operator is clear: the denotation of R^* is the reflexive, transitive closure of the binary relation denoted by R. In the case of CRA and CRA($mon\neg$) a similar proviso needs to be made as for FO^*_{tree} (cf. Section 3): the transitive closure operator may only be applied to expressions that denote tables with precisely two columns. We use

CRA(∗), CRA($mon\neg, *$), TRA(∗) and DRA(∗) to denote the extensions of the respective algebras with the transitive closure operator, conform this restriction.

The path equalities of Regular XPath$^{\approx}$ can be expressed in CRA(∗) and CRA($mon\neg, *$) using intersection and projection, and in TRA(∗) using intersection and $\sim$: $R \approx S$ can be expressed as $\sim\sim(R \cap S)$. On the other hand, in DRA(∗) it is not clear whether path equalities can be expressed. Let DRA(∗,loop) denote the extension of DRA with both the Kleene star and the $(\cdot)^{\mathsf{loop}}$ operator, that has the following semantics [9]: $R^{\mathsf{loop}} = R \cap \epsilon$. Using loop, and given the fact that Regular XPath$^{\approx}$ is closed under taking inverses of path expressions, we can express path equalities: $R \approx S$ translates to $(R \circ S^{-1})^{\mathsf{loop}}$.

It follows from Theorem 3 that Regular XPath$^{\approx}$, DRA(∗,loop), TRA(∗), CRA(∗) and CRA($mon\neg, *$) all have the same expressive power. Of these four, DRA(*,loop) is the most suitable algebra for Regular XPath$^{\approx}$, since its query containment problem is ExpTime-complete (as follows from the fact that there are linear translations from and to Regular XPath$^{\approx}$ [25]). See also Table 5.

5. CONCLUSION AND OPEN PROBLEMS

We have discussed four dialects of Navigational XPath, and we have shown that they correspond, in terms of expressive power, to natural fragments or extensions of first-order logic. Furthermore, we have identified suitable algebras for each of the dialects.

We have not discussed *monadic second-order logic* (MSO) as a target for expressive power. Several dialects of navigational XPath have been proposed in the literature that have the same expressive power as MSO (see for instance [9, 24]), but in our view none has the simplicity and appeal of the dialects we studied here.

We end with four open problems. The first two are related to the DRA and its extension DRA(∗,loop). The second two relate to the strictness of the hierarchy of path languages given in [5].

Problem 1 *Give axiomatizations for DRA and DRA(∗,loop) on XML-trees.*

Problem 2 *Does loop really any add expressive power to DRA(∗,loop)? Or equivalently, do path equalities really contribute to the expressive power of Regular XPath$^{\approx}$?*

Problem 3 *Is Regular XPath$^{\approx}$ (or equivalently FO^{*}_{tree}) less expressive than MSO?*

Problem 4 *Is FO^{*}_{tree} less expressive than $FO_{\mathsf{tree}} + TC^{1}$? If so, identify an XPath dialect that has exactly the same expressive power as the latter.*

Acknowledgments. We are grateful to Loredana Afanasiev and Tadeusz Litak for helpful comments. Balder ten Cate is supported by NWO research grant 639.021.508.

6. REFERENCES

[1] S. Abiteboul, P. Buneman, and D. Suciu. *Data on the web.* Morgan Kaufman, 2000.

[2] M. Benedikt, W. Fan, and G. Kuper. Structural properties of XPath fragments. In *Proc. ICDT 2003*, 2003.

[3] A. Chandra and D. Harel. Structure and complexity of relational queries. *J. Comput. Syst. Sci.*, 25(1):99–128, 1982.

[4] E. Codd. Relational completeness of data base sublanguages. In R. Rustin, editor, *Database Systems*, pages 33–64. Prentice-Hall, 1972.

[5] J. Engelfriet and H. Hoogeboom. Nested pebbles and transitive closure. In *Proc. STACS*, pages 477–488, 2006.

[6] W. Fan, F. Geerts, X. Jia, and A. Kementsietsidis. SMOQE: a system for providing secure access to XML. In *Proc. VLDB'2006*, pages 1227–1230, 2006.

[7] W. Fan, F. Geerts, X. Jia, and A. Kementsietsidis. Rewriting regular XPath queries on XML views. In *Proc. ICDE'2007*, 2007.

[8] E. Filiot, J. Niehren, J.-M. Talbot, and S. Tison. Polynomial time fragments of xpath with variables. In *Proc. PODS'07*, 2007.

[9] E. Goris and M. Marx. Looping caterpillars. In *Proc. LICS 2005*. IEEE Computer Society, 2005.

[10] G. Gottlob, C. Koch, and R. Pichler. Efficient algorithms for processing XPath queries. In *VLDB'02*, 2002.

[11] G. Gottlob, C. Koch, and K. Schulz. Conjunctive queries over trees. In *Proc. PODS'04*, pages 189–200, 2004.

[12] M. Grohe and N. Schweikardt. The succinctness of first-order logic on linear orders. 1(1), 2005.

[13] I. Hodkinson and M. Reynolds. Separation - past, present, and future. In S. Artemov et al., editor, *We will show them! (Essays in honour of Dov Gabbay on his 60th birthday)*, pages 117–142. College Publications, 2005.

[14] M. Hollenberg. An equational axiomatization of dynamic negation and relational composition. *Journal of Logic, Language and Information*, 6(4):381–401, 1997.

[15] J.A.W. Kamp. *Tense Logic and the Theory of Linear Order*. PhD thesis, University of California, Los Angeles, 1968.

[16] M. Kay. *XPath 2.0 Programmer's Reference*. Wrox, 2004.

[17] M. Marx. XPath with conditional axis relations. In *Proc. EDBT'04*, volume 2992 of *LNCS*, pages 477–494, 2004.

[18] M. Marx. Conditional XPath. *ACM Transactions on Database Systems (TODS)*, 30(4):929–959, 2005.

[19] M. Marx and M. de Rijke. Semantic Characterizations of Navigational XPath. *SIGMOD Record*, 34(2):41–46, 2005.

[20] F. Neven and T. Schwentick. XPath containment in the presence of disjunction, DTDs, and variables. In *Proc. ICDT 2003*, 2003.

[21] L. Stockmeyer. *The Complexity of Decision Problems in Automata Theory*. PhD thesis, Dept. Electrical Engineering, MIT, Cambridge, Mass., 1974.

[22] A. Tarski. On the calculus of relations. *Journal of Symbolic Logic*, 6:73–89, 1941.

[23] A. Tarski and S. Givant. *A Formalization of Set Theory without Variables*, volume 41. AMS Colloquium publications, Providence, Rhode Island, 1987.

[24] B. ten Cate. The expressivity of XPath with transitive closure. In *Proc. PODS*, pages 328–337, 2006.

[25] B. ten Cate and C. Lutz. The complexity of query containment in expressive fragments of XPath 2.0. In *Proc. PODS'07*, 2007.

[26] B. ten Cate and M. Marx. Axiomatizing the logical core of XPath 2.0. In *Proc. ICDT'07*, 2007.

[27] J. van Benthem. Program constructions that are safe for bisimulation. *Studia Logica*, 60(2):331–330, 1998.

[28] M. Vardi. On the complexity of bounded–variable queries. In *Proc. PODS'95*, pages 266–276, 1995.

Georg Gottlob Speaks Out

on How Computer Science is a Continuation of Logic by Other Means, How Change Makes You Live Longer, How to Improve the Status of Computer Science, How to Interest Practical People in Complexity Theory and Database Theory, and More

by Marianne Winslett

Georg Gottlob
http://web.comlab.ox.ac.uk/oucl/people/georg.gottlob.html
(see also http://benner.dbai.tuwien.ac.at/staff/gottlob/)

Welcome to this installment of ACM SIGMOD Record's series of interviews with distinguished members of the database community. I'm Marianne Winslett, and today we are at the SIGMOD 2006 conference in Chicago, Illinois. I have here with me Georg Gottlob, who is a professor of computing science at Oxford University. Before that he was a professor at the Technical University of Vienna for many years. His research interests lie in database theory, logic, AI, and complexity. Georg is a co-founder of Lixto Corporation, and until recently he was the editor in chief of AI Communications*. Georg's PhD is from the Technical University of Vienna. So, Georg, welcome!*

Georg, as a young man, you worked as a night train conductor, you wrote a booklet on the catacombs of St Stephen's Cathedral in Vienna, and you were very interested in psychology. How did you end up in computer science?

I was interested in psychology and I considered studying medicine, but out of laziness I started to study mathematics. In high school, mathematics didn't take much time to learn. It was very easy for me, so I thought, why shouldn't I continue and have a nicer life? For medicine, you had to study very big books and do other difficult things. I started mathematics, and I liked analysis, but was more interested in algebra, logic, and discrete mathematics. Eventually I saw that there is more of this kind of mathematics in computer science, such as automata theory, formal languages, and complexity theory. Eventually, after having done almost all the mathematics exams, I switched to computer science.

You typically change your main research focus every few years, moving between databases, AI, complexity, and logic. Would you recommend this form of research nomadism to others?

This form of nomadism is very nice for *me*, at least. I could recommend it to others; it makes life more interesting in some sense. You switch between topics, but you come back, of course, to the same set of research communities. You learn a lot more by switching. When you have been in a field, and you come back two years later, many things have happened there, so it is not boring. You can bring ideas from one area to another, so it is actually rather nice. There is a drawback too: people will ask you to review papers from three different fields, and people will ask you to write letters of recommendation in multiple areas, so that is a drawback.

You are probably as well known in the AI and computational logic communities as in the database theory community. What are the connections and differences between these communities and fields?

The differences are probably the specific goals, the narrow goals. But I see a lot of connections, and the basic thing is that they are all connected through logic. For me, logic is very important, and for me, the entire field of computer science is a continuation of logic by other means. Logic is a very important component of databases. If you answer a query, the answer should be a logical consequence of the database in some sense. And in AI planning, your actions should be in some sense logically related to what you know, to your knowledge. So you need forms of reasoning in both areas, and they are not forms of reasoning that correspond to classical logic. For example, we have the closed-world assumption in databases.

How does chairing IJCAI compare to chairing PODS?

IJCAI is a huge conference, PODS is a smaller conference. In IJCAI we have more than 1,000 submissions. So it's unthinkable that you read all of the very good papers as when you are an IJCAI program chair. As an IJCAI program chair, you have to be a very good organizer. I'm not such a good organizer, but still when I chaired IJCAI in 2003, I tried to do my best and fortunately I succeeded. You must know people, you must organize your program committee, you must try to get many people to cover areas that you don't know, and so on. For PODS, you don't have to organize so much; the community is much smaller, so it is easier to pick a good program committee.

Why isn't computer science considered on a par with the "old" disciplines like math, physics, and chemistry?

I think there are several reasons for that. One reason is certainly that computer science is a very new discipline, compared to disciplines that are two thousand years old. Of course, there is also the grant money. Physicists are very strong in getting lots of funding and they don't want computer science to take that funding away from them. On the other hand, the popular view of computer science, of computing, is not necessarily a view of computing *research*. The icons of computing are not scientists, but are the big entrepreneurs. That is not so in physics or in chemistry. But I see a slight change, and one symptom or sign of the change is that Academies of Science, who never in the past considered computer science to be one of their own disciplines, are starting to include it. For instance, the German Academy of Science Leopoldina starts to have its own section on information processing. And the European Academy of Science is starting to have its own group on computing. So there are good signs, and I think eventually computer science will end up being on par with the other sciences.

Is there some way that we could leverage the fact that we have these icons that are so well known?

We computer scientists have to promote our scientific ideas a little bit more. I think we have to be a little bit more oriented toward the popular audience in presenting our ideas. If you look at journals like *Science* and *Nature* and magazines such as *New Scientist or Scientific American*, you find a lot of articles on physics and biology and so on, but very few featured articles on computing. I think more popularization is needed so that people understand that computer science is not just entrepreneurship, and has a real science behind it.

You were a professor at the Technical University of Vienna for 18 years, with a well-endowed chair there. Many people thought you would stay there forever, so what led you to move to Oxford?

There are several reasons for this. First, Oxford is a very nice place. It is a beautiful place, and it is a great university in my opinion. The students are fantastic. Actually, I have to say that the best students in Oxford are probably at the same level as the best students in Vienna. But the average student in Oxford is so much better than in Vienna, and that is due to the very strict and severe selection they have at Oxford. Of course, we also have fewer students at Oxford. In Vienna, we had classes of 500 students. You probably cannot imagine this as an American.

I certainly can--- we have computer science classes of 700 students at the University of Illinois!

Do you? So you know what it means---teaching a programming class to 500 students. After the class, you are tired, but the students don't let you out of the room, they want you to explain the errors in their programs and so on. At Oxford, we don't have all that. We have small classes, very bright students.

And there are other reasons why I moved to Oxford. I like Vienna very much (in particular, the Technical University, where I now have an adjunct position), but I also like the change. After 18 years, you feel it is time to change. As I get older, time is passing much faster. There is a psychological reason for this, namely, your life is becoming more monotonous. As a child, every year, every month, there are so many changes, you get new teachers every year, you get into another grade and so on. Once you are stable in a job, you do always the same thing or similar things, and in retrospect, it seems like much less has happened. So, if you make a change to your life, then you will also get a stronger feeling of time passing, and you will feel like you live longer. So it is for a longer life that I went to Oxford.

Now you have done research in four countries with completely different university systems: Italy, Austria, the US, and now the UK. How do these university systems differ?

In my opinion, the main difference is between the Anglo-Saxon countries, like England and the US on one hand; and the continental European system like Italy and Austria, which are still different from one another, but are comparable in some sense. The main difference is the entrance to a scientific career, namely, at the level of assistant professor in the US, or at the level of a lecturer in the UK. Once you are an assistant professor or a lecturer, whether you can get promoted or not depends only on your work. It doesn't depend on anything else. You are in a tenure track position. The tenure track doesn't exist in most countries of continental Europe. In Austria, for instance, if you are hired as an assistant professor, you must switch universities in order to be promoted. I think that is not very good. I prefer the Anglo-Saxon system, where it is maybe a little more difficult to get an assistant professor job, but once you have it, then you can plan your life in some sense and you know you can succeed if you work hard.

You are such a proponent of change that I would think that you would enjoy a system that encourages people to work at multiple universities.

Yes, I am personally a proponent of change, but many people are not so. They want to have a certain stability. Some young people in continental Europe don't want to work at a university because they think the jobs there aren't secure enough. I personally am not so much interested in security, but I can understand that others are. We lost several potential assistant professors at the Technical University of Vienna for that reason. Many people don't apply to Vienna who would apply for a junior faculty position in the UK or in the United States, because in the UK or US they can get a secure job.

You are interested in data exchange. Tell us about that.

Data exchange is actually a very old problem, and there is a renewed interest in it. Data exchange is related to data integration. In data integration, the main goal is that you ask a query to heterogeneous databases, and the query is processed in a distributed way, then the results are collected together and you get the answer. In data exchange, the main goal is to take data present in one database and materialize it in another database with a different schema. Data exchange involves slightly different problems from data integration; since you want to materialize the data, but you have columns that don't exist in one of the databases, and you have different schemas, you need to deal very much with null values and how to satisfy integrity constraints when you transfer data. This is a very interesting problem. It's a very theoretically demanding problem. It's a problem that the industry is very interested in now. For instance, the IBM Almaden labs already have a system called CLIO that is used for data exchange. So I think our research on data exchange can influence the industry.

Why does computational complexity matter in the database field?

Complexity is one of the major problems in in querying databases, and in computer science in general. If you have an information system, you want to be able to guarantee that your users will get their answers in reasonable time, without having to wait two years for an answer. If the information system is very good, you can guarantee this. It is not sufficient to just say that some queries may be slow, but in general the system is very fast. That is not a good guarantee, because who knows which query is which kind? You want to guarantee performance, and complexity theory is the right tool to do that. You can say, okay, these algorithms are linear, that is very good; or you can say this problem is, say, NP-hard.

In practice people say that complexity theory doesn't interest them. But I have found a way to convince them that complexity theory is important, just by renaming it! If you go to a database practitioner, a systems person, and tell him or her, "Your system, your query language, doesn't scale," then they will get nervous very quickly. And it is actually the same as saying that their approach has a high computational complexity.

Some of our readers might think, "Well, you just submit the query and you get the answer back in seconds, so what's hard about it?"

Quick query answers are possible in simple cases with well designed databases and very good query optimizers, but generally they are not possible. If you want to solve problems that have many constraints and sub-queries, it is quite hard to get an answer. Just ask the people who deal with constraint processing or constraint databases. There they are exasperated, they don't know how to answer these queries, how to get to a good solution to those kinds of constraint problems.

As long as we are in very classical databases and consider short and simple queries, then things may work well. But if you go slightly to other domains, and want to handle that with database technology, it doesn't succeed.

There are so many hard problems, even just with conjunctive queries. Queries are getting longer and longer because of views. We query a view, and the view itself is a query, so if we resolve that, we get a very long query. And these long queries are very hard to answer, computationally hard.

You also have been working on hypertree decompositions and their applications in database theory and AI. What are hypertree decompositions?

Any query can be described by a hypergraph, and the hypertree width of a query is the measure of the degree of cyclicity in the hypergraph of the query. If the query is acyclic, then its hypertree width is 1. Now, query processing is, in general, NP-complete, but acyclic queries are easy to handle. Unfortunately, in practice, acyclic queries don't occur too often. But we observed that in practice most of the non-acyclic queries are nearly acyclic, with a hypertree width of 2 or 3. With Nicola Leone and Francesco Scarcello, we have developed algorithms for efficiently processing queries which are nearly acyclic. In some sense, we can fight complexity through this notion of hypertree decomposition.

Most of the web is still formatted in HTML. Why don't we see more XML data on the web---is XML a good data model?

That is a very interesting question. I remember that in 1995 some people predicted that in the early 2000s most of the web would be in XML, and everything would be annotated in XML. But what we see is the contrary; people are still producing huge amounts of HTML pages, and you almost never find a page in XML. XML is a very good data interchange format, but it has failed so far as a format that people generally use for putting information on the web.

There are, in my opinion, several reasons for that. One reason is that it is so much easier to write an HTML page and right away see what you get. When my son and my daughter were 13, they perfectly understood HTML. They could put it on the web, they could change HTML code, but they would not have been able to write DTD, or an XML DTD, or an XML schema. HTML is much more user friendly. The other reason is that in XML you have to use tags, but what do these tags mean? For one person, one tag means one thing, for another person it means another thing. There is no universal meaning attached to the tags. There may be some conventions, and some XML styles that two companies can use to exchange data, but there is no general meaning for the tags, and this will not happen until we have the semantic web. Basically, there is not much reason to put information in XML on the web (with the exception of the very restricted RSS format). But XML is a very nice language for exchanging data.

Do you believe in the semantic web?

Yes, I believe very much in the semantic web. I think the semantic web will be the Next Big Thing, the next revolution, and it will change our lives, just like the web did. The web did change our life, you have to admit that, and the semantic web will change our life in a similar way. That is my prediction.

Why hasn't it happened yet?

It hasn't happened yet because it must happen through companies. I think we are almost at the point where it will happen, I believe it will happen very soon. The lead will be taken by the big web players, like Google, Yahoo!, Microsoft. Some day soon, one of these big companies will announce that their search engines will now use ontological search based on semantic annotations attached to web pages, and if you want your web page to be found very efficiently, then please annotate it. Then people will do the annotations and we will have the semantic web. Of course, we will probably have another semantic web after the one which is now standardized, but it will be very similar. People will annotate their pages in a semantic way using nice tools offered by Microsoft, Google, Yahoo!, and other companies, and then this information will be put into the search engine databases and indexed on those annotations.

What do you think is currently hot in database theory?

There are many interesting problems in database theory. Data extraction still poses a lot of important problems. For example, the web is in HTML, but corporate data are needed in a more structured format. So how do you convert large amounts of HTML into structured data? That is what the Lixto tool from our startup company does, but while we were building it we discovered so many interesting problems. For example, can complex extraction patterns be learned? How can you best extract data from PDF? Database theory can address those questions, so they are not just practical questions. How can you recognize hierarchically organized data? How can you recognize table structures in a PDF document? These are not just problems for systems people. There can be a good theory, and that theory can help us to build better systems.

Of course there are other theory problems relevant to databases. Let me give you just one that has to do with privacy. Given a view, can we in some sense describe all the queries over the original database that we can answer using just this view? That is an old problem in some sense, but it is also a new problem because nobody has asked the question in these particular terms and little work has been done on it. Victor Vianu is working on it now, and Alan Nash, and a few other people. I would like to work on that problem.

And there are of course plenty of other interesting problems in database theory. The interest in the PODS conference is growing, so it is a good sign. We have been getting more submissions, and high attendance, so it is a very lively field.

Would you advise a gifted youngster to become a computer scientist today?

Definitely. I have been a computer scientist now for 25 years and I never regretted it for a single second in my life. Computer science is a very beautiful field. One of the big advantages of computer science is that once you study computer science, it does not determine what you have to do in your life, because it is such a rich and diverse field. If you study computer science you may end up as a mathematician; most of the time I do mathematics. But you might also end up as a graphics designer, or an electronics engineer. Or you might end up as a sociologist, because if you do computer supported cooperative work, you have to study how people interact. So if you study computer science, you don't choose too early in life a particular branch of a field that offers so many possibilities. This is helpful especially for young people who don't always know their talents; in computer science, they still have time to develop them.

Do you have any words of advice for fledgling or mid-career database researchers or practitioners?

My advice is to change a little bit. If you are a mid career computer scientist, then you have done something for a long time. I enjoy changes between different areas, and then coming back to one's area with different perspectives, and I think that is very good. Change places; do not be attached too much to one place.

Among all your past research, what is your favorite piece of work?

Apart from hypertree decompositions, which I already described, it is certainly the work on semi-structured data. For instance, finding polynomial algorithms for the full XPath language was a challenge. Developing hypertree decompositions was cool. I also enjoyed building up a theory and algorithms for data extraction, and realizing them in the Lixto system.

If you magically had enough extra time to do one additional thing at work that you are not doing now, what would it be?

I would like to study quantum computing. I have never had the time to study it, I don't know how it works very well. I would have to study first a little calculus, all these things that I have forgotten, and then study quantum computing. I think quantum computing is a challenging area, and there are nice algorithms.

If you could change one thing about yourself as a computer science researcher, what would it be?

It would be to be less distracted by daily business and short term needs. I am very susceptible to them. I have some long-term plans that I haven't realized so far, simply because I get continuously distracted by short-term needs. I should say to myself, "Why should I always be on top of everything for short term things?" It is much better sometimes to work on long-term goals, and I would like to have more concentration on them.

Report on the Third International Workshop on XQuery Implementation, Experience and Perspectives (XIME-P 2006)

Michael Carey
BEA Systems Inc., San Jose, CA, USA
mcarey@bea.com

Torsten Grust
Technische Universität München, Germany
grust@in.tum.de

Summary

On June 30, 2006, *XIME-P 2006*, the International Workshop on *XQuery Implementation, Experience and Perspectives* was held. This workshop marks the third event in a workshop series whose primary aim is to shed light on XQuery systems, specification aspects, foundations of the language, and the many perceivable shapes it may take on in the future. Like the two previous workshops of 2004 (Paris) and 2005 (Baltimore), XIME-P 2006 was held as a co-located workshop of the ACM SIGMOD Conference, which this year was held in Chicago, IL, USA. The 2006 edition was co-chaired by Michael Carey and Torsten Grust. The workshop web site can be found at www.ximep-2006.org.

A Landmark Year for XQuery

After eight years of hard work, in 2006 the XQuery family of standards has entered W3C Candidate Recommendation status and odds are that XQuery will become a Recommendation this very year. XIME-P once more provided *the* forum for XQuery designers, practitioners, and system builders to reflect on the past design process, the current state of the specification, and the XQuery processors implementing the language. Further, the workshop had a distinctive "Now what?" atmosphere to it: several of the talks as well as the panel were addressing language extensions and even discussed efforts to embrace programming models other than the functional approach that XQuery has adopted from the start (see comments on the workshop panel below).

In this landmark year for XQuery, XIME-P 2006 was happy and proud to welcome Don Chamberlin (IBM Almaden, co-editor of the W3C XQuery specifications and co-designer of the original SQL language) as this year's keynote speaker. Don, who clearly seems to devote a significant share of his lifetime to language design, gave a one-hour keynote entitled "XQuery—Where Do We Go From Here?" which started out with an assessment of what went right and wrong with the design of XQuery. Don placed the serious integration with existing XML standards, declarativity and independence of a specific persistence model, and completeness in the presence of an easy-to-learn language core on the "right" side of the fence. On the other side live a partly fragile syntax, side-effecting node constructors, and a number of important omissions from version 1.0 of the standard (*e.g.*, updates, text search, and explicit support for grouping). Next, since the first meeting of the XQuery Working Group took place back in November 1999, Don tried to respond to the oft-heard critical question on why it took so (too?) long to get to where the language is today. One piece of Don's answer related to the sometimes overwhelming number of public comments on draft XQuery standards. Processing these comments thoroughly took time. According to Don, "XQuery is like SQL was in 1986", which, since XML is probably not going away, answers the question of whether XQuery has a future or not. Apart from the version 1.0 omissions mentioned above, Don closed his talk by underlining the need for language extensions that turn XQuery into a stand-alone programming language, a thesis that received both applause and debate over the course of the workshop day.

The slides of the keynote talk are available on the web (see Electronic Proceedings below).

Workshop Sessions

A total of seven presentations promoted different possible extensions to XQuery, dug deep into innova-

tive implementations details of XQuery processors, and also tried to embrace non-(or multi-)hierarchical XML data instances. Michael Kay presented several use cases for a *positional grouping* extension which can turn a flat sequence of items into a hierarchy based on their position in the sequence. XQueryP, a proposal that was co-authored by a number of W3C XQuery Working Group members, adds defined evaluation order and destructive variable assignment (in case `declare execution sequential` is given in the query preamble). XQueryP ultimately aims to evolve XQuery in the direction of an XML application development language.

The XQuery processing model lends itself to a diversity of implementation approaches. A group of three presentations described (1) a pull-based *streaming* processor (XQPull) for XQuery that can cope with recursive queries and backward axis steps, (2) a *native* XPath processor (Natix) that takes advantage of recurring structures in incoming XML messages by shifting parts of the query evaluation effort from run time to compile time, and (3) a university course (taught at U Saarland, Germany) that teaches students how to use *relational* query compilation and processing techniques to process a subset of XQuery.

Two talks convincingly showed how real applications naturally lead to overlapping, non-tree-shaped data instances, which the markup community refers to as *multi-hierarchical* or *standoff-annotated* XML. The talks proposed language extensions designed to access such overlapping data instances and also described how an efficient implementation of "standoff " XPath axes found their way into the MonetDB/XQuery processor.

Caffeine and Code

For the first time, the 2006 edition of XIME-P explicitly called for XQuery system and application demonstrations. In retrospect, this can only be considered a success. The workshop's 90-minute "Caffeine and Code" session featured 10 exhibits, ranging from XQuery development and debugging tools to XQuery processors based on different implementation paradigms (streaming, native, relational) to innovative applications with an XQuery core (search, content assembly and publishing). Additionally, Erik Meijer (Microsoft) presented XLinq, a tight integration of an XML-aware in-memory query language—based on the monad comprehension calculus—with .NET-based programming languages. Different from the first edition of XIME-P, the XQuery community now has clearly reached a stage where there is no lack of mature implementations anymore. This provides the much needed playground to test drive the core language as well as its many perceivable extensions. It is also one of the key prerequisites for XQuery to finally obtain W3C Recommendation status.

Panel "Programming for XML"

The workshop day drew to a close with a panel reviving the aforementioned question of whether (and if so, how) XQuery shall evolve into a full-fledged programming language. Led by Dana Florescu (Oracle), the group of panelists was split in two "opposing" camps. Michael Carey (BEA), co-author of the XQueryP proposal, and Erik Meijer, proposing XLinq embedded into Visual Basic, each argued for a procedural-style approach to XML application development. Michael Kay (Saxonica), on the other hand, voiced his fear that imperative constructs in XQuery would be abused by users unfamiliar with the declarative style of querying: instead, stateful logic should only be used in the glue that couples (almost purely) functional XQuery building blocks. Complex applications *can* be built today using the current specification of XQuery, insisted Ron Avnur (Mark Logic). As if to prove his point in advance, Ron undertook some on-the-fly XQuery coding during the earlier Caffeine and Code session. The notion of *monads*, which came up in the XLinq presentation and which has already helped more than once to bring impure constructs to the world of functional programming languages, received some interest among the workshop participants.

Raw Numbers

XIME-P 2006 received 15 paper submissions plus approximately the same number of proposals for system demonstrations. Each submission was reviewed by at least six eyes from the workshop program committee. The PC had 16 members, five of which are also part of the W3C XQuery standardization effort. The workshop room (hard to find, but worth it) in the Renaissance Chicago Hotel hosted a lively group of 40 attendees, with academia and industry each contributing approximately 50% of the participants.

Electronic Proceedings

All accepted papers have been published in the XIME-P 2006 Electronic Proceedings, ISBN 1-5953-465-0, as part of the ACM Digital Library. Pa-

pers will appear in the SIGMOD Digital Symposium Collection (DiSC). The Electronic Proceedings may also be found on the workshop web site at `www.ximep-2006.org`. This site additionally hosts the slides of Don Chamberlin's keynote talk as well as entries for all demonstrations featured in the Caffeine and Code session.

Acknowledgments

XIME-P 2006 gratefully acknowledges the financical support provided by BEA Systems, Inc., IBM, Oracle, and X-Hive Corp. ACM supported and sponsored the workshop. Our thanks also go to the SIGMOD 2006 team of organizers, in particular to Joanne Martori, Lisa Singh, and Kevin C. Chang for their truly excellent remote as well as on-site support.

The advent of XQuery has led to interesting interconnections between the database, markup, and programming language research and development communities. This year's workshop co-chairs hope that a fourth edition (and further editions) of XIME-P will continue to tighten this bond in 2007 and beyond.

Report on ACM Workshop on Health Information and Knowledge Management (HIKM 2006)

Li Xiong
Department of Mathematics and Computer Science
Emory University
lxiong@mathcs.emory.edu

Yuni Xia
Department of Computer Science
Indiana University and Purdue University at Indianapolis
yxia@cs.iupui.edu

1 Introduction

Health information technology is receiving a tremendous amount of attention as a strategic area that will benefit the society in the 21st century. The continued advances in healthcare such as digitization of medical records, creation of central record systems, development of healthcare data warehouses increasingly pose new challenges to information and knowledge management. The high stakes and unique characteristics of healthcare data such as the long-term value of the data, varied data quality, the complexity of the data, the privacy constraints, as well as the availability requirements in emergent situations require a special treatment of traditional information management techniques.

The inaugural Workshop on Health Information and Knowledge Management is the first in its series for presentation and exchange of research results and experiences on leading edge issues of health information and knowledge management. The mission of the workshop is to provide an open yet focused platform for researchers and practitioners from computer science, medical informatics, as well as healthcare industry to discuss current research challenges and advances and share their perspective in various aspects of health information and knowledge management. The workshop was held on November 11, 2006, in conjunction with the ACM International Conference on Information and Knowledge Management (CIKM) in Arlington, VA.

2 Workshop Themes

The workshop was very interactive, with the audience raising many questions for the speakers and a lively discussion following the technical presentations. Several overall themes emerged from the paper presentations and discussions.

Data Heterogeneity and Data Integration. An overarching complexity associated with health information is data heterogeneity. Data resides in relational databases (such as patient medical records) as well as in unstructured text forms (such as lab and pathology reports). Many healthcare information systems and warehouses need to integrate or pull data from a variety of heterogeneous data sources including electronic medical records, registry data, GenBank, etc. Data integration across the heterogeneous sources remains a challenging issue.

Medical Text Management. The high throughput and data intensive era provides enormous opportunities for decision making and clinical research. It also presents information challenges for clinicians and medical researchers to find relevant information efficiently and to obtain just-in-time information during patient encounters. New factors such as the rich context, user-oriented evaluation, and access to the data archive have to be taken into account in designing and developing information retrieval systems for health care.

Medical Data Mining. Data mining offers promising techniques in finding patterns in medical data for

public health research such as disease prediction, diagnosis, and outcomes research. However, researchers have to cope with a number of challenges in working with medical datasets as they tend to be small, high dimensional, rich in data types, and error prone. It remains an open question to have quality mining results that are easy to search and interpreted by domain experts.

Data Security and Privacy. A major barrier for integration and exchange of medical data is the privacy and security concerns. On one hand, health information need to be integrated and available for better patient care and public health research, on the other hand, health information need to be protected for confidentiality and privacy. Research in data security and privacy is of growing importance for health information. The interesting problems range from traditional ones such as access control and multi-level security to recently emerged ones such as flexible privacy preserving data mining and data sharing.

Data integrity. A major barrier in health information management is that a small but significant amount of data contains error. In some cases, it is a result of a historical rule such as using mother's Social Security Number in a child's medical record. In other cases, it is a mistake during the process of transcribing data from doctor's hand-written notes. Research advances in data verification as well as practical techniques that can be applied in health information are of particular interests.

3 Program

The workshop program includes a keynote speech and 7 paper presentations followed by a discussion. The paper presentations are divided into 3 sessions that cover a variety of topics, including medical document indexing and retrieval, medical data mining and clinical trial management, healthcare data integration and exchange, and security management.

3.1 Keynote Address

The keynote address was given by Dr. Tyrone Grandison from IBM Almaden Research with the title "Enabling the Healthcare Revolution". In his talk, he emphasized that the 21st Century has ushered in new awareness of the need to address the dire state of the Healthcare sector. There is worldwide push to leverage Information Technology to help reduce the current problems in the sector. However, many problems to be faced in this transition are either incorrectly labeled as problems or go unidentified. He gave an overview of the problems in the health care from the business, social and legal perspectives. He illustrated the technology challenges and existing solutions in various areas including modeling, standardization, storage, security and privacy, data analytics, interoperability, remote system and service science. He concluded his talk with a discussion of the areas for future exploration and a call for action. The slides are available at the HIKM workshop website[1].

3.2 Medical Document Indexing and Retrieval

Evangelos Milios first presented a paper titled "Automatic document indexing in large medical collections" [2]. He presented AMTEx, an automatic term extraction method, specifically designed for the automatic indexing of documents in large medical collections such as MEDLINE, the premier bibliographic database of the U.S. National Library of Medicine (NLM). AMTEx combines MeSH, the terminological thesaurus resource of NLM, with a well-established method for extraction of domain terms, the C/NC-value method. The performance of various AMTEx configurations in the indexing task is measured against the current state-of-the-art, the MMTx method, on a subset of MEDLINE documents. While AMTEx achieves better precision and recall than MMTx, it still suggests that term extraction in large medical document collections remains to be a challenging task.

Susan Price then presented a paper titled "Using semantic components to express clinical questions against document collections" [5]. She described a new model for describing the content of documents in domain-specific collections, using document classes and semantic components, that may supplement existing indexing and searching techniques and improve information retrieval. She also presented the results of using the model to represent clinical questions in the medical domain. They manually mapped generic questions from a clinical question taxonomy to two web-based document collections using the document classes and semantic components they identified for each collection. They successfully mapped 36 of 50 question categories in one resource, and 34 of 50 in the other. Based on the frequency of the question

[1] http://www.mathcs.emory.edu/hikm

types in the taxonomy, over 92% of questions were covered by the mappings in both resources.

3.3 Medical Data Mining and Applications

Carlos Ordonez (University of Houston) presented a paper titled "Comparing association rules and decision trees for disease prediction" [4]. He described a decision rule mining method in which search constraints are introduced to find only medically significant association rules and make search more efficient. Association rules are compared to predictive rules mined with decision trees on a radiology dataset. Experiments show that decision trees tend to find few simple rules, most rules have somewhat low reliability, most attribute splits are different from medically common splits, and most rules refer to very small sets of patients. In contrast, association rules generally include simpler predictive rules, they work well with user-binned attributes, rule reliability is higher and rules generally refer to larger sets of patients.

Ravi Shankar (Stanford University) presented a paper titled "Epoch: an ontological framework to support clinical trials management" [6]. The increasing complexity of clinical trials has generated an enormous requirement for knowledge and information specification at all stages of the trials, including planning, documentation, implementation, and analysis. He presented a knowledge-based framework (Epoch) to support the management of clinical trials tailored to the Immune Tolerance Network (ITN), an international research consortium developing new therapeutics in immune-mediated disorders. They currently target two areas that are vital to the successful implementation of a trial: (1) tracking study participants as they advance through the trials, and (2) tracking biological specimens as they are processed at the trial laboratories. The core of the software architecture is a suite of ontologies that conceptualizes relevant clinical trial domain.

3.4 Medical Data Integration and Exchange

Vagelis Hristidis (Florida International University) presented a paper titled "A flexible approach for electronic medical records exchange" [3]. The presented approach allows generating a customized EMR independent of existing healthcare applications and provides an on-demand, secure, efficient, and semantics-agnostic way to exchange EMRs in a collaborative environment using a declarative communication engine, called Communication Virtual Machine (CVM). CVM negotiates the capabilities of the involved parties and underlying networks to guarantee Quality of Service and presentation compatibility. It can also be customized to enforce privacy and security requirements (e.g., HIPAA) by enabling logging, authentication, and so on. A prototype of the EMR exchange approach has been implemented which integrates the i-Rounds medical record system used at Miami Children's Hospital.

Rafae Bhatti (Purdue University) presented a paper titled "Policy-based security management for federated healthcare databases (or RHIOs)" [1]. He described a context-aware policy-based security management system for health informatics. The policies are based on a set of use cases developed for the HL7 Clinical Document Architecture (CDA) standard. The system is designed to adapt well to ubiquitous healthcare services in a non-traditional, pervasive environment using the same infrastructure that enables federated healthcare management for traditional organizational boundaries. Their work also included an enforcement architecture and a demonstration prototype for the policy-based system.

Wai Gen Yee (Illinois Institute of Technology) presented a paper titled "Bridging a gap in the proposed personal health record" [7]. The emerging electronic health record infrastructure is guiding records to be stored in repositories that collectively supply a patient's comprehensive health history. However, he argued that legal and technological constraints may keep such a system from delivering health histories in a timely manner (i.e., when medical attention is needed). He presented a design for a portable personal health record system that complies with HIPAA standards of security and interaction. The authenticity of stored records on this PHR is automatically verifiable, increasing its usefulness to health care providers.

4 Final Note

There have been recent and upcoming workshops focusing on individual information management issues in general or biomedical domains that also apply to health information management, such as Workshop on Database Interoperability (InterDB)[2], Workshop

[2]http://www.fundp.ac.be/eco/interdb/2007/

on Privacy Data Management (PDM)[3], Workshop on Secure Data Management (SDM)[4], Workshop on Quality of Databases (QDB)[5], Workshop on Management of Uncertain Data Workshop (MUD)[6], Workshop on Data Mining in Bioinformatics (BIOKDD)[7], Workshop on Biomedical Data Engineering(BMDE)[8], and Workshop on Data Integration in the Life Sciences (DILS)[9].

The HIKM workshop was the first to bring together the individual information and knowledge management issues unique to health information. The workshop organizers and attendees envision a series of workshops building upon the success of this workshop. A proposal for a second workshop is being prepared.

5 Acknowledgements

The success of HIKM 2006 was due to a team effort. First of all, we would like to thank the authors for providing the quality content of the program and the participants for the lively discussion at the workshop. We would also like to express our gratitude to the program committee and external reviewers, who worked very hard in reviewing papers and providing suggestions for their improvements. Finally, we would like to thank the CIKM'06 conference for providing a venue as well as support for the workshop.

References

[1] R. Bhatti, K. Moidu, and A. Ghafoor. Policy-based security management for federated healthcare databases (or rhios). In *HIKM*, pages 41–48, 2006.

[2] A. Hliaoutakis, K. Zervanou, E. G. M. Petrakis, and E. E. Milios. Automatic document indexing in large medical collections. In *HIKM*, pages 1–8, 2006.

[3] V. Hristidis, P. J. Clarke, N. Prabakar, Y. Deng, J. A. White, and R. P. Burke. A flexible approach for electronic medical records exchange. In *HIKM*, pages 33–40, 2006.

[4] C. Ordonez. Comparing association rules and decision trees for disease prediction. In *HIKM*, pages 17–24, 2006.

[5] S. Price, L. M. L. Delcambre, and M. L. Nielsen. Using semantic components to express clinical questions against document collections. In *HIKM*, pages 9–16, 2006.

[6] R. D. Shankar, S. B. Martins, M. J. O'Connor, D. B. Parrish, and A. K. Das. Epoch: an ontological framework to support clinical trials management. In *HIKM*, pages 25–32, 2006.

[7] W. G. Yee and B. Trockman. Bridging a gap in the proposed personal health record. In *HIKM*, pages 49–56, 2006.

[3] http://www.ceebi.curtin.edu.au/PDM2007/
[4] http://www.hitech-projects.com/sdm-workshop/sdm07.html
[5] http://WWW.hiqiq.de/qdb/
[6] http://mud.cs.utwente.nl/
[7] http://bio.informatics.iupui.edu/biokdd07/
[8] http://www.db.is.kyushu-u.ac.jp/bmde2005/
[9] http://dils07.cis.upenn.edu/

Event Report on iiWAS 2006 and MoMM 2006, Yogyakarta, December 2006

Eric Pardede
Department of Computer Science and Computer Engineering
La Trobe University, Bundoora, VIC 3086 Australia
E.Pardede@latrobe.edu.au

ABSTRACT

In this paper, we report on two events held in Yogyakarta, Indonesia from 4 – 6 December 2006. These conferences are the Eight International Conference on Information Integration & Web-Services and Applications (iiWAS 2006) and the Fourth International Conference on Advances in Mobile Computing and Multimedia (MoMM 2006).

1. HISTORY OF IIWAS AND MOMM

In 1999, the International Organization for Information Integration and Web-Based Application and Services (@WAS) endorsed an international conference in Yogyakarta, Indonesia. The conference, which was called the International Conference on Information Integration and Web Applications & Web Services (iiWAS), had a vision of encouraging researchers from developing countries to submit their works in the emerging area of information integration on the web and its related applications. Throughout the years, the conference has emerged as a major event and has gained a well-respected reputation as an international conference with global participation without losing its original vision of supporting early researchers and researchers from developing countries.

In 2003, iiWAS was held along with a new conference called the International Conference on Advanced in Mobile Multimedia (MoMM). This was a response to the emergence of multimedia research in mobile computing. Since then, the two conferences have been co-located and organized by @WAS. The conference has been renamed the International Conference on Advances in Mobile Computing and Multimedia, to promote opportunities of research in a broader area of mobile computing and multimedia.

In 2006, when iiWAS conference is being held in the city where it began, @WAS expanded the conferences audiences to much broader areas by organizing five (5) workshops along with the two main conferences. The workshops vary from interest in Broadband and Wireless Computing, Communication and Applications (BWCCA), Mobile Multimedia Information Retrieval (MoMIR), Semantic Information Integration on Knowledge Discovery (SIIK), Systems-On-Chips (SoC) to Trustworthy Ubiquitous Computing (TwUC).

2. KEYNOTE TALKS

During the three days of the event, five keynote talks were presented. Gabriele Kotsis from Johannes Kepler University Linz presented a session on the emergence of user-oriented Quality of Service (QoS) as opposed to technical-point-of-view QoS. The talk outlined the associated research challenges and gave an overview on the existing solutions and approaches towards user perceived QoS concepts [1].

Elizabeth Chang presented a talk on digital ecosystem, which is a new-networked architecture and collaborative environment that addresses the weakness of client-service, peer-to-peer, grid and web services [2]. In this talk, the speaker provided a detailed explanation of the system including the architecture, swarm intelligence, design and implementation, its comparison to existing networked architecture, social, cultural and economic impact and also provided practical examples.

In his talk, Andreas Langegger presented the opportunities of ontologies for electronic data integration. While XML family of standard use syntactic rules enables data integration, ontologies provide answers to using a semantics web approach. This talk set down a discussion on existing misconceptions and also the expectations and potentialities of ontologies [3].

In the next keynote talk in MoMM, Omar Boucelma discussed integration issues for spatial data. There are several challenges posed by spatial data integration such as the lack of accepted standard for spatial and geographic data representation, no standard for spatial data repository, and the problems of semantic impairment during the spatial data integration [4].

The final keynote firstly provides the existing techniques used for ad hoc and sensor networks connectivity and shows how the techniques do not assure a connected

topology. The speakers then proposed a new formula for estimating the critical transmitting range [5].

Despite the different issues discussed, these five keynote talks encapsulated the theme of the conferences. The talks cover methodologies, architecture, qualities, issues, and formulas for information integration.

3. TECHNICAL PROGRAM

Both conferences attracted a large number of submissions from various countries. iiWAS attracted 121 submissions from 30 countries, from which 40 papers were accepted as regular papers (33% acceptance rate) . In addition, we also accepted 13 papers as short papers. MoMM attracted 64 submissions from 29 countries, from which 25 papers were accepted as regular papers (39% acceptance rate) and 8 papers were accepted as short papers.

This year, many of the iiWAS contributed papers discussed e-applications in various domains such as e-commerce, e-government, e-learning and e-counselling. Web services, web databases and ontologies were also widely discussed during the conference. As in previous years, human-computer interaction aspects were also represented in this conference. [6] defines a formal approach that models worker behaviors in an electronic environment. The approach is vital for managerial and marketing purposes, resource allocation, and design of efficient and user-friendly web-based platforms and services. This paper was awarded the best paper of iiWAS 2006.

Another area that was extensively discussed during iiWAS 2006 was information integration and retrieval . One of the papers in this area discussed the use of relevance feedback on 3D model similarity retrieval to increase the precision retrieval during objects integration [7]. This paper was awarded best student paper of iiWAS 2006.

For MoMM 2006, the papers were evenly distributed across the area of Mobile Computing and Multimedia, with an increasing tendency for research in wireless sensor networks. One paper in multimedia [8], which discussed the mobility of multimedia flow in heterogeneous networks was awarded best paper in MoMM 2006. Another multimedia paper, which discussed the performance of error detection algorithm of encoded video streams [9] was awarded best student paper.

4. WORKSHOPS

There were five successful workshops held during the event of iiWAS/MoMM 2006. The First International Workshop on Broadband and Wireless Computing, Communication and Applications (BWCCA-2006) embodied the rapid evolution of communication networks. Different kinds of networks with different characteristics and integration exhibited interconnection problems at different levels in the hardware and software design. This workshop provided a forum for disseminating new ideas in broadband and wireless computing.

The First International Workshop on Mobile Multimedia Information Retrieval (MoMIR-2006) discussed the challenges for more creative content retrieval due to the rapid expansion of digital multimedia data delivered on new generation mobile devices, such as PDA, smart phones and portable audiovisual players.

While iiWAS has continuous interest on information integration, the next workshop focused on information integration for knowledge discovery. The First International Workshop on Semantic Information Integration on Knowledge Discovery (SIIK-2006) presented recent works to synergize different views of techniques and policies of future research directions on semantic integration used in knowledge discovery.

The International Workshop on System-On-Chip (SoC-2006) was inspired by the emergence of complex multicore system-on-chips that consist of a large number of IP blocks on the same silicon. The multiple cores increase the challenges for processing loads. This workshop aimed to be the forum for the latest research in the area.

Finally, the First International Workshop on Trustworthy Ubiquitous Computing (TwUC-2006) discussed the dilemma of misuse of information that is supposed to be easily accessible in the pervasive computing era. This workshop covered the security issues that arise to ensure trust in evolving ubiquitous interfaces. In this last workshop, the best workshop paper was chosen. [10] introduced a context authentication proxy for shared devices using spatial reference.

5. MOVING FORWARD

Starting from a small workshop in a developing country with participants from a limited number of regional countries, iiWAS has developed into a reputable global conference. In the last few years, the iiWAS conference has gained a new momentum to move forward. Now, with a sister conference, MoMM, and a list of interesting workshops, the event has served a large scholarly audience.

In the future, following the success in 2006, iiWAS and MoMM will be held in tandem. Workshops will be maintained as they attract a large number of people who have an interest in particular areas of expertise. It is also planned to invite highly-respected speakers to address future conferences. The event should attract more global researchers, while maintaining the same level of support for early researchers and researchers from developing countries.

6. REFERENCES

[1] Kotsis, G. QoS in Next Generation Internet: Putting the User into the Focus. In the Eight International Conference on Information Integration and Web-Based Applications & Services, *books@ocg.at, Band 214,* (2006), pp. 1-2

[2] Chang, E. and West, M. Digital Ecosystems A Next Generation of the Collaborative Environment. In the Eight International Conference on Information Integration and Web-Based Applications & Services, *books@ocg.at, Band 214,* (2006), pp. 3-23

[3] Wagner, R. and Langegger, S. Are Ontologies the Nostrum to Bridge the Semantic Gap?. In the Eight International Conference on Information Integration and Web-Based Applications & Services, *books@ocg.at, Band 214,* (2006), pp. 27-28

[4] Boucelma, O. Spatial Data Integration on the Web: Issues and Solutions. In the Fourth International Conference on Advances in Mobile Computing and Multimedia, *books@ocg.at, Band 215,* (2006), pp.5

[5] De Marco, G. and Barolli, L. A Bound for the Connectivity of Ad Hoc and Sensor Networks with Shadowing-Induced Radio Irregularities. In the Fourth International Conference on Advances in Mobile Computing and Multimedia, *books@ocg.at, Band 215,* (2006), pp.231-239

[6] Geczy, P., Izumi, N., Akaho, S., and Hasida, K. Navigation Space Formalism and Exploration of Knowledge Worker Behavior on Intranet. In the Eight International Conference on Information Integration and Web-Based Applications & Services, *books@ocg.at, Band 214,* (2006), pp. 163-172

[7] Akbar, S., Kung, J., Wagner, R. and Prihatmanto, A.S. Multi-Feature Integration with Relevance Feedback on 3D Model Similarity Retrieval. In the Eight International Conference on Information Integration and Web-Based Applications & Services, *books@ocg.at, Band 214,* (2006), pp. 77-86

[8] Ahlund, C., Brannstrom, R, Andersson, K. and Tjernstrom, O. Multimedia Flow Mobility in Heterogeneous Networks Using Multihomed Mobile IPv6. In the Fourth International Conference on Advances in Mobile Computing and Multimedia, *books@ocg.at, Band 215,* (2006), pp.29-38

[9] Superiori, L., Nemethova, O. and Rupp, M. Performance of a H.264/AVC Error Detection Algorithm Based on Syntax Analysis. In the Fourth International Conference on Advances in Mobile Computing and Multimedia, *books@ocg.at, Band 215,* (2006), pp.49-58

[10] Mayrhofer, R. A Context Authentication Proxy for IpsSec Using Spatial Reference. In Frontiers in Mobile and Web Computing: Proceedings of MoMM2006 and iiWAS 2006 Workshops, *books@ocg.at, Band 216,* (2006), pp.449 - 462

ACM SIGMOD Conference: Vancouver, 2008

Call for Papers

The annual ACM SIGMOD conference is a leading international forum for database researchers, practitioners, developers, and users to exchange ideas and results. We invite submissions of original research contributions, case studies, and industrial papers, as well as proposals for demonstrations and tutorials. We encourage submissions relating to all aspects of data management defined broadly, and particularly encourage work on topics of emerging interest in the research and development communities.

SIGMOD 2008 will be hosted in Vancouver, Canada from June 9 - 12, 2008.

Areas of Interest

Benchmarking and performance evaluation
Case studies that reveal research challenges
Data quality, semantics and integration
Database monitoring and tuning
Data privacy and security
Data mining and machine learning
Database tuning
Decision support on large data sets
Embedded, sensor and mobile databases
Indexing
Managing uncertain and imprecise information
Novel/Advanced applications, systems, and platforms
Peer-to-peer and networked data management
Personalized information systems
Query processing and optimization
Replication, caching, and publish-subscribe systems
Semi-structured data
Storage and transaction management
Text and image databases
Large scale social networks
Web services

Important Dates

November 9, 2007:	Research paper abstracts due (11:59 pm US Eastern Time)
November 16, 2007:	Research and industrial papers, demonstration and tutorial proposals due (11:59 pm US Eastern Time)
Feb 1-15, 2008:	Author feedback
February 16, 2008:	Notification
March 20, 2008:	Submission to proceedings

For more detailed information, please refer to the conference website at http://www.sigmod08.org.

SIGMOD 2008 Experimental Repeatability Requirements

To help published papers achieve an impact and stand as reliable reference-able works for future research, the SIGMOD 2008 reviewing process includes an assessment of the extent to which the presented experiments are repeatable by someone with access to all the required hardware, software, and test data. Thus, we attempt to establish that the code developed by the authors exists, runs correctly on well-defined inputs, and performs in a manner compatible with that presented in the paper.

Papers that are accepted and are verified this way will be eligible to include the following text in the proceedings: "The results in this paper were verified by the SIGMOD repeatability committee". If a subset of the results cannot be verified by the SIGMOD repeatability committee (e.g., for IP reasons), then the phrasing will change to: "The results in Figures <list of figures> and Tables <list of tables> in this paper were verified by the SIGMOD repeatability committee." If verified code and data is also made available for archiving, the following phrase may be added: "And the code and data are available at <a site to be determined>."

If we test your code for repeatability, we will give feedback about any problems we encounter. This will not influence whether your paper is selected or not for publication. Submission of code and data is optional for SIGMOD 2008, that is, submitting code and data will have no influence on whether your paper is accepted for SIGMOD 2008.

If you choose not to submit, however, then you are still required to submit, on December 16, 2007, a note stating: (1) the reasons you feel you could not submit; (2) how much time, if at all feasible, it would take you to satisfy these requirements; and (3) any suggestions you have about how to achieve the goal of scientific repeatability in our field.

If you choose to submit code and data for at least some of your reported experiments, please follow the requirements listed on the SIGMOD 2008 website at http://www.sigmod08.org/sigmod_research.shtml.

Authors are required to upload to a special website (to be announced on the conference website), at the latest on Dec. 16, 2007 (one month after the SIGMOD paper submission deadline):

1. The code needed to run the experiments quoted in the paper.
2. The data sets used in the experiments.
3. A plain-text file named INSTALL, describing the platform used for running the experiments and the installation procedure.
4. A plain-text file named HOWTO**,** describing how figures and tables of experimental results in the paper were produced.

Please, see the above website for further details of the repeatability requirements.

CALL FOR PAPERS

27th ACM SIGMOD–SIGACT–SIGART Symposium on

PRINCIPLES OF DATABASE SYSTEMS (PODS 2008)

June 9–11, 2008, Vancouver, Canada

`http://www.sigmod08.org/`

Program Chair:

Maurizio Lenzerini
Dip. di Informatica e Sistemistica,
University of Rome La Sapienza,
Roma, Italy
`lenzerini@dis.uniroma1.it`

The PODS symposium series, held in conjunction with the SIGMOD conference series, provides a premier annual forum for the communication of new advances in the theoretical foundation of database systems. For the 27th edition, original research papers providing new insights in the specification, design, or implementation of data management tools are called for. Topics that fit the interests of the symposium include the following (as they pertain to databases):

> *algorithms; complexity; computational model theory; concurrency; constraints; data exchange; data integration; data mining; data modeling; data on the Web; data streams; data warehouses; distributed databases; information retrieval; knowledge bases; logic; multimedia; physical design; privacy; quantitative approaches; query languages; query optimization; real-time data; recovery; scientific data; security; semantic Web; semi-structured data; spatial data; temporal data; transactions; updates; views; Web services; workflows; XML.*

Submitted papers should be at most ten pages, using reasonable page layout and font size of at least 10pt (note that the SIGMOD style file does not have to be followed). Additional details may be included in an appendix, which, however, will be read at the discretion of the program committee. *Papers longer than ten pages or in font size smaller than 10pt risk rejection without consideration of their merits.*

The submission process will be through the Web; a link to the submission website will appear on the conference website in due time. Note that, unlike the SIGMOD conference, PODS does not use double-blind reviewing, and therefore PODS submissions should be eponymous (i.e., the names and affiliations of authors should be listed on the paper).

The results must be unpublished and not submitted for publication elsewhere, including the proceedings of other symposia or workshops. All authors of accepted papers will be expected to sign copyright release forms. One author of each accepted paper will be expected to present it at the conference.

Important Dates:

Short abstracts due:	28	November	2007
Paper submission:	5	December	2007
Notification:	26	February	2008
Camera-ready copy:	20	March	2008

Best Paper Award: An award will be given to the best submission, as judged by the program committee.

Best Newcomer Award: There will also be an award for the best submission, as judged by the program committee, written solely by authors who have never published in earlier PODS proceedings.

The program committee reserves the right to give both awards to the same paper, not to give an award, or to split an award among several papers. Papers authored or co-authored by PC members are not eligible for an award.

Nominations Solicited for ACM – Infosys Foundation Award

Nominations are invited for the 2007 ACM – Infosys Foundation Award. The ACM – Infosys Foundation Award in the Computing Sciences recognizes personal contributions by young scientists and system developers to a contemporary innovation that, through its depth, fundamental impact and broad implications, exemplifies the greatest achievements in the discipline. The award is accompanied by a prize of $150,000. Financial support for the award is provided by an endowment from the Infosys Foundation.

Nominations should include:

1. A current vitae, listing the age of the candidate, publications, patents, honors other awards, etc.
2. A nomination letter from the principal nominator, which describes the work of the nominee, and draws particular attention to the contributions which are seen as meriting the award.
3. Short supporting letters from at least three, and no more than five, endorsers. The letters should come from prominent individuals familiar with the nominee's contributions and no more than two from the same institution.

Please visit the ACM Awards site (http://awards.acm.org/html/awards.cfm) for additional information on ACM's award program.

Nominations should be sent to the Chair of the ACM – Infosys Foundation Award by December 31, 2007: *Juris Hartmanis, jh at cs dot cornell dot edu*

In addition to Dr. Hartmanis, the ACM – Infosys Foundation Award Committee members are Susan Graham, Butler Lampson, Kurt Mehlhorn, Amit Singhal, and Andrew Yao.

Press Release: ACM and Infosys Technologies Announce New Award to Recognize Contemporary Computer Research and Innovation

Infosys Foundation to Fund $150,000 Annual Award for Contributions of Young Computer Scientists and System Developers

New York, August 20, 2007 - The Association for Computing Machinery (ACM) has signed an agreement with Infosys Technologies Limited to create a new annual award that recognizes young scientists and system developers whose contemporary innovations are having a dramatic impact on the computing field. The award, to be known as the ACM-Infosys Foundation Award in the Computing Sciences, will seek eligible candidates globally, and will carry a prize of $150,000. The first recipient will be announced as part of the ACM Awards program in early 2008.

"ACM welcomes this new category of recognition to its awards program," said Stuart I. Feldman, president of ACM. "By adding to our portfolio of prestigious awards, we are helping to advance the science and the profession of computing. With the extremely generous support of Infosys, we are able to provide encouragement and support to the best work of our contemporaries in this dynamic field".

Instituted by Infosys through the company's philanthropic arm, the Infosys Foundation, the new award joins the roster of ACM awards that honor outstanding contributors to a range of computing applications, including computer architecture, theory and practice, education, humanitarian initiatives, software, and major technical advances of an enduring nature.

Infosys Technologies Chief Executive Officer S. Gopalakrishnan said Infosys wanted to work with ACM to recognize major research contributions in a timely manner. "We want to honor contemporary innovation to ensure that the award goes to younger individuals whose scientific contributions in the computing sciences are significantly impacting the field. Our goal is to identify breakthroughs that have broad implications well beyond the scope of the innovation itself, and that reflect an underlying scientific or engineering methodology that is remarkable for its rigor or for its sheer audacity," he said. "Even more importantly, we hope the winners of the ACM-Infosys Foundation Award in the Computing Sciences will, by example, inspire students worldwide to consider careers in computing science, and continue the impressive history of innovations and contributions achieved in the field."

A selection committee to administer the award will be formed and led by Juris Hartmanis, the Walter R. Read Professor of Engineering at Cornell University. He is a co-recipient of the 1993 ACM A.M. Turing Award for his contributions in establishing the foundations for computational complexity theory.

About ACM

ACM, the Association for Computing Machinery http://www.acm.org, is an educational and scientific society uniting the world's computing educators, researchers and professionals to inspire dialogue, share resources and address the field's challenges. ACM strengthens the profession's collective voice through strong leadership, promotion of the highest standards, and recognition of technical excellence. ACM supports the professional growth of its members by providing opportunities for life-long learning, career development, and professional networking.

About Infosys Technologies

Infosys Technologies Ltd. (NASDAQ:INFY) defines, designs and delivers IT-enabled business solutions that help Global 2000 companies win in a flat world. These solutions focus on providing strategic differentiation and operational superiority to clients. Infosys creates these solutions for its clients by leveraging its domain and business expertise along with a complete range of services. With Infosys, clients are assured of a transparent business partner, world-class processes, speed of execution and the power to stretch their IT budget by leveraging the Global Delivery Model that Infosys pioneered. Infosys has 75,000 employees in 44 offices worldwide. Infosys is part of the NASDAQ-100 Index. For more information visit http://www.infosys.com

About The Infosys Foundation

Established in 1996, the Infosys Foundation is the philanthropic arm of Infosys Technologies Ltd. and has the sole objective of fulfilling he social responsibility of the company by creating opportunities and working toward a more equitable society. The Infosys Foundation has made effective strides in the areas of healthcare, education, social rehabilitation, and the arts. The company contributes up to one percent of its profit to the foundation each year.