Alon Halevy
Avigdor Gal (Eds.)

LNCS 2382

Next Generation Information Technologies and Systems

5th International Workshop, NGITS 2002
Caesarea, Israel, June 2002
Proceedings

Springer

Alon Halevy Avigdor Gal (Eds.)

Next Generation Information Technologies and Systems

5th International Workshop, NGITS 2002
Caesarea, Israel, June 24-25, 2002
Proceedings

Springer

Volume Editors

Alon Halevy
University of Washington, Department of Computer Science and Engineering
Sieg Hall, Room 310, Mailstop 352350, Seattle, WA, 98195, USA
E-mail:alon@cs.washington.edu

Avigdor Gal
Technion – Israel Institute of Technology
William Davidson Faculty of Industrial Engineering and Management
Technion City 3200, Haifa, Israel
E-mail:avigal@ie.technion.ac.il

Cataloging-in-Publication Data applied for

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Next generation information technologies and systems : 5th international workshop ; proceedings / NGITS 2002, Caesarea, Israel, June 24 - 25, 2002. Alon Halevy ; Avigdor Gal (ed.). - Berlin ; Heidelberg ; New York ; Barcelona ; Hong Kong ; London ; Milan ; Paris ; Tokyo : Springer, 2002
(Lecture notes in computer science ; Vol. 2382)
ISBN 3-540-43819-X

CR Subject Classification (1998): H.4, H.3, H.5, H.2, D.2.12, C.2.4

ISSN 0302-9743
ISBN 3-540-43819-X Springer-Verlag Berlin Heidelberg New York

Springer-Verlag Berlin Heidelberg New York
a member of BertelsmannSpringer Science+Business Media GmbH

http://www.springer.de

Typesetting: Camera-ready by author, data conversion by PTP Berlin, Stefan Sossna e. K.
Printed on acid-free paper SPIN 10870481 06/3142 5 4 3 2 1 0

Lecture Notes in Computer Science 2382
Edited by G. Goos, J. Hartmanis, and J. van Leeuwen

Springer
Berlin
Heidelberg
New York
Barcelona
Hong Kong
London
Milan
Paris
Tokyo

Lecture Notes in Computer Science

For information about Vols. 1–2287
please contact your bookseller or Springer-Verlag

Vol. 2288: K. Kim (Ed.), Information Security and Cryptology – ICISC 2001. Proceedings, 2001. XIII, 457 pages. 2002.

Vol. 2289: C.J. Tomlin, M.R. Greenstreet (Eds.), Hybrid Systems: Computation and Control. Proceedings, 2002. XIII, 480 pages. 2002.

Vol. 2290: F. van der Linden (Ed.), Software Product-Family Engineering. Proceedings, 2001. X, 417 pages. 2002.

Vol. 2291: F. Crestani, M. Girolami, C.J. van Rijsbergen (Eds.), Advances in Information Retrieval. Proceedings, 2002. XIII, 363 pages. 2002.

Vol. 2292: G.B. Khosrovshahi, A. Shokoufandeh, A. Shokrollahi (Eds.), Theoretical Aspects of Computer Science. IX, 221 pages. 2002.

Vol. 2293: J. Renz, Qualitative Spatial Reasoning with Topological Information. XVI, 207 pages. 2002. (Subseries LNAI).

Vol. 2294: A. Cortesi (Ed.), Verification, Model Checking, and Abstract Interpretation. Proceedings, 2002. VIII, 331 pages. 2002.

Vol. 2295: W. Kuich, G. Rozenberg, A. Salomaa (Eds.), Developments in Language Theory. Proceedings, 2001. IX, 389 pages. 2002.

Vol. 2296: B. Dunin-Kęplicz, E. Nawarecki (Eds.), From Theory to Practice in Multi-Agent Systems. Proceedings, 2001. IX, 341 pages. 2002. (Subseries LNAI).

Vol. 2297: R. Backhouse, R. Crole, J. Gibbons (Eds.), Algebraic and Coalgebraic Methods in the Mathematics of Program Construction. Proceedings, 2000. XIV, 387 pages. 2002.

Vol. 2298: I. Wachsmuth, T. Sowa (Eds.), Gesture and Language in Human-Computer Interaction. Proceedings, 2001. XI, 323 pages. 2002. (Subseries LNAI).

Vol. 2299: H. Schmeck, T. Ungerer, L. Wolf (Eds.), Trends in Network and Pervasive Computing – ARCS 2002. Proceedings, 2002. XIV, 287 pages. 2002.

Vol. 2300: W. Brauer, H. Ehrig, J. Karhumäki, A. Salomaa (Eds.), Formal and Natural Computing. XXXVI, 431 pages. 2002.

Vol. 2301: A. Braquelaire, J.-O. Lachaud, A. Vialard (Eds.), Discrete Geometry for Computer Imagery. Proceedings, 2002. XI, 439 pages. 2002.

Vol. 2302: C. Schulte, Programming Constraint Services. XII, 176 pages. 2002. (Subseries LNAI).

Vol. 2303: M. Nielsen, U. Engberg (Eds.), Foundations of Software Science and Computation Structures. Proceedings, 2002. XIII, 435 pages. 2002.

Vol. 2304: R.N. Horspool (Ed.), Compiler Construction. Proceedings, 2002. XI, 343 pages. 2002.

Vol. 2305: D. Le Métayer (Ed.), Programming Languages and Systems. Proceedings, 2002. XII, 331 pages. 2002.

Vol. 2306: R.-D. Kutsche, H. Weber (Eds.), Fundamental Approaches to Software Engineering. Proceedings, 2002. XIII, 341 pages. 2002.

Vol. 2307: C. Zhang, S. Zhang, Association Rule Mining. XII, 238 pages. 2002. (Subseries LNAI).

Vol. 2308: I.P. Vlahavas, C.D. Spyropoulos (Eds.), Methods and Applications of Artificial Intelligence. Proceedings, 2002. XIV, 514 pages. 2002. (Subseries LNAI).

Vol. 2309: A. Armando (Ed.), Frontiers of Combining Systems. Proceedings, 2002. VIII, 255 pages. 2002. (Subseries LNAI).

Vol. 2310: P. Collet, C. Fonlupt, J.-K. Hao, E. Lutton, M. Schoenauer (Eds.), Artificial Evolution. Proceedings, 2001. XI, 375 pages. 2002.

Vol. 2311: D. Bustard, W. Liu, R. Sterritt (Eds.), Soft-Ware 2002: Computing in an Imperfect World. Proceedings, 2002. XI, 359 pages. 2002.

Vol. 2312: T. Arts, M. Mohnen (Eds.), Implementation of Functional Languages. Proceedings, 2001. VII, 187 pages. 2002.

Vol. 2313: C.A. Coello Coello, A. de Albornoz, L.E. Sucar, O.Cairó Battistutti (Eds.), MICAI 2002: Advances in Artificial Intelligence. Proceedings, 2002. XIII, 548 pages. 2002. (Subseries LNAI).

Vol. 2314: S.-K. Chang, Z. Chen, S.-Y. Lee (Eds.), Recent Advances in Visual Information Systems. Proceedings, 2002. XI, 323 pages. 2002.

Vol. 2315: F. Arhab, C. Talcott (Eds.), Coordination Models and Languages. Proceedings, 2002. XI, 406 pages. 2002.

Vol. 2316: J. Domingo-Ferrer (Ed.), Inference Control in Statistical Databases. VIII, 231 pages. 2002.

Vol. 2317: M. Hegarty, B. Meyer, N. Hari Narayanan (Eds.), Diagrammatic Representation and Inference. Proceedings, 2002. XIV, 362 pages. 2002. (Subseries LNAI).

Vol. 2318: D. Bošnački, S. Leue (Eds.), Model Checking Software. Proceedings, 2002. X, 259 pages. 2002.

Vol. 2319: C. Gacek (Ed.), Software Reuse: Methods, Techniques, and Tools. Proceedings, 2002. XI, 353 pages. 2002.

Vol.2320: T. Sander (Ed.), Security and Privacy in Digital Rights Management. Proceedings, 2001. X, 245 pages. 2002.

Vol. 2321: P.L. Lanzi, W. Stolzmann, S.W. Wilson (Eds.), Advances in Learning Classifier Systems. Proceedings, 2002. VIII, 231 pages. 2002. (Subseries LNAI).

Vol. 2322: V. Mařík, O. Štěpánková, H. Krautwurmová, M. Luck (Eds.), Multi-Agent Systems and Applications II. Proceedings, 2001. XII, 377 pages. 2002. (Subseries LNAI).

Vol. 2323: À. Frohner (Ed.), Object-Oriented Technology. Proceedings, 2001. IX, 225 pages. 2002.

Vol. 2324: T. Field, P.G. Harrison, J. Bradley, U. Harder (Eds.), Computer Performance Evaluation. Proceedings, 2002. XI, 349 pages. 2002.

Vol 2326: D. Grigoras, A. Nicolau, B. Toursel, B. Folliot (Eds.), Advanced Environments, Tools, and Applications for Cluster Computing. Proceedings, 2001. XIII, 321 pages. 2002.

Vol. 2327: H.P. Zima, K. Joe, M. Sato, Y. Seo, M. Shimasaki (Eds.), High Performance Computing. Proceedings, 2002. XV, 564 pages. 2002.

Vol. 2328: R. Wyrzykowski, J. Dongarra, M. Paprzycki, J. Waśniewski (Eds.), Parallel Processing and Applied Mathematics. Proceedings, 2001. XIX, 915 pages. 2002.

Vol. 2329: P.M.A. Sloot, C.J.K. Tan, J.J. Dongarra, A.G. Hoekstra (Eds.), Computational Science – ICCS 2002. Proceedings, Part I. XLI, 1095 pages. 2002.

Vol. 2330: P.M.A. Sloot, C.J.K. Tan, J.J. Dongarra, A.G. Hoekstra (Eds.), Computational Science – ICCS 2002. Proceedings, Part II. XLI, 1115 pages. 2002.

Vol. 2331: P.M.A. Sloot, C.J.K. Tan, J.J. Dongarra, A.G. Hoekstra (Eds.), Computational Science – ICCS 2002. Proceedings, Part III. XLI, 1227 pages. 2002.

Vol. 2332: L. Knudsen (Ed.), Advances in Cryptology – EUROCRYPT 2002. Proceedings, 2002. XII, 547 pages. 2002.

Vol. 2334: G. Carle, M. Zitterbart (Eds.), Protocols for High Speed Networks. Proceedings, 2002. X, 267 pages. 2002.

Vol. 2335: M. Butler, L. Petre, K. Sere (Eds.), Integrated Formal Methods. Proceedings, 2002. X, 401 pages. 2002.

Vol. 2336: M.-S. Chen, P.S. Yu, B. Liu (Eds.), Advances in Knowledge Discovery and Data Mining. Proceedings, 2002. XIII, 568 pages. 2002. (Subseries LNAI).

Vol. 2337: W.J. Cook, A.S. Schulz (Eds.), Integer Programming and Combinatorial Optimization. Proceedings, 2002. XI, 487 pages. 2002.

Vol. 2338: R. Cohen, B. Spencer (Eds.), Advances in Artificial Intelligence. Proceedings, 2002. X, 197 pages. 2002. (Subseries LNAI).

Vol. 2340: N. Jonoska, N.C. Seeman (Eds.), DNA Computing. Proceedings, 2001. XI, 392 pages. 2002.

Vol. 2342: I. Horrocks, J. Hendler (Eds.), The Semantic Web – ISCW 2002. Proceedings, 2002. XVI, 476 pages. 2002.

Vol. 2345: E. Gregori, M. Conti, A.T. Campbell, G. Omidyar, M. Zukerman (Eds.), NETWORKING 2002. Proceedings, 2002. XXVI, 1256 pages. 2002.

Vol. 2346: H. Unger, T. Böhme, A. Mikler (Eds.), Innovative Internet Computing Systems. Proceedings, 2002. VIII, 251 pages. 2002.

Vol. 2347: P. De Bra, P. Brusilovsky, R. Conejo (Eds.), Adaptive Hypermedia and Adaptive Web-Based Systems. Proceedings, 2002. XV, 615 pages. 2002.

Vol. 2348: A. Banks Pidduck, J. Mylopoulos, C.C. Woo, M. Tamer Ozsu (Eds.), Advanced Information Systems Engineering. Proceedings, 2002. XIV, 799 pages. 2002.

Vol. 2349: J. Kontio, R. Conradi (Eds.), Software Quality – ECSQ 2002. Proceedings, 2002. XIV, 363 pages. 2002.

Vol. 2350: A. Heyden, G. Sparr, M. Nielsen, P. Johansen (Eds.), Computer Vision – ECCV 2002. Proceedings, Part I. XXVIII, 817 pages. 2002.

Vol. 2351: A. Heyden, G. Sparr, M. Nielsen, P. Johansen (Eds.), Computer Vision – ECCV 2002. Proceedings, Part II. XXVIII, 903 pages. 2002.

Vol. 2352: A. Heyden, G. Sparr, M. Nielsen, P. Johansen (Eds.), Computer Vision – ECCV 2002. Proceedings, Part III. XXVIII, 919 pages. 2002.

Vol. 2353: A. Heyden, G. Sparr, M. Nielsen, P. Johansen (Eds.), Computer Vision – ECCV 2002. Proceedings, Part IV. XXVIII, 841 pages. 2002.

Vol. 2358: T. Hendtlass, M. Ali (Eds.), Developments in Applied Artificial Intelligence. Proceedings, 2002 XIII, 833 pages. 2002. (Subseries LNAI).

Vol. 2359: M. Tistarelli, J. Bigun, A.K. Jain (Eds.), Biometric Authentication. Proceedings, 2002. XII, 373 pages. 2002.

Vol. 2360: J. Esparza, C. Lakos (Eds.), Application and Theory of Petri Nets 2002. Proceedings, 2002. X, 445 pages. 2002.

Vol. 2361: J. Blieberger, A. Strohmeier (Eds.), Reliable Software Technologies – Ada-Europe 2002. Proceedings, 2002 XIII, 367 pages. 2002.

Vol. 2363: S.A. Cerri, G. Gouardères, F. Paraguaçu (Eds.), Intelligent Tutoring Systems. Proceedings, 2002. XXVIII, 1016 pages. 2002.

Vol. 2364: F. Roli, J. Kittler (Eds.), Multiple Classifier Systems. Proceedings, 2002. XI, 337 pages. 2002.

Vol. 2366: M.-S. Hacid, Z.W. Raś, D.A. Zighed, Y. Kodratoff (Eds.), Foundations of Intelligent Systems. Proceedings, 2002. XII, 614 pages. 2002. (Subseries LNAI).

Vol. 2367: J. Fagerholm, J. Haataja, J. Järvinen, M. Lyly. P. Råback, V. Savolainen (Eds.), Applied Parallel Computing. Proceedings, 2002. XIV, 612 pages. 2002.

Vol. 2368: M. Penttonen, E. Meineche Schmidt (Eds.), Algorithm Theory – SWAT 2002. Proceedings, 2002. XIV, 450 pages. 2002.

Vol. 2370: J. Bishop (Ed.), Component Deployment. Proceedings, 2002. XII, 269 pages. 2002.

Vol. 2374: B. Magnusson (Ed.), ECOOP 2002 – Object-Oriented Programming. XI, 637 pages. 2002.

Vol. 2382: A. Halevy, A. Gal (Eds.), Next Generation Information Technologies and Systems. Proceedings, 2002. VIII, 169 pages. 2002.

Vol. 2385: J. Calmet, B. Benhamou, O. Caprotti, L. Henocque, V. Sorge (Eds.), Artificial Intelligence, Automated Reasoning, and Symbolic Computation. Proceedings, 2002. XI, 343 pages. 2002. (Subseries LNAI).

Vol. 2386: E.A. Boiten, B. Möller (Eds.), Mathematics of Program Construction. Proceedings, 2002. X, 263 pages. 2002.

Vol. 2389: E. Ranchhod, N.J. Mamede (Eds.), Advances in Natural Language Processing. Proceedings, 2002. XII, 275 pages. 2002. (Subseries LNAI).

Preface

NGITS 2002 was the fifth workshop of its kind, promoting papers that discuss new technologies in information systems. Following the success of the four previous workshops (1993, 1995, 1997, and 1999), the fifth NGITS Workshop took place on June 24–25, 2002, in the ancient city of Caesarea.

In response to the Call for Papers, 22 papers were submitted. Each paper was evaluated by three Program Committee members. We accepted 11 papers from 3 continents and 5 countries, Israel (5 papers), US (3 papers), Germany, Cyprus, and The Netherlands (1 paper from each).

The workshop program consisted of five paper sessions, two keynote lectures, and one panel discussion. The topics of the paper sessions are: Advanced Query Processing, Web Applications, Moving Objects, Advanced Information Models, and Advanced Software Engineering.

We would like to thank all the authors who submitted papers, the program committee members, the presenters, and everybody who assisted in making NGITS 2002 a reality.

June 2002 Alon Halevy, Avigdor Gal

NGITS 2002 Conference Organization

NGITS 2002 Program Committee

Einat Amitay	IBM Haifa Research Lab, Israel
Remzi Arpaci-Dusseau	University of Wisconsin, USA
Vijay Atluri	Rutgers University, USA
Catriel Beeri	Hebrew University, Israel
Nick Belkin	Rutgers University, USA
Dan Berry	University of Waterloo, Canada
Diego Calvanese	Università di Roma "La Sapienza", Italy
Silvana Castano	Università degli Studi di Milano, Italy
Giuseppe De Giacomo	Università di Roma "La Sapienza", Italy
Asuman Dogac	Middle East Technical University, Turkey
Opher Etzion	IBM Haifa Research Lab, Israel
Elena Ferrari	Università degli Studi dell'Insubria, Italy
Johannes Gehrke	Cornell University, USA
James Geller	NJIT, USA
Lise Getoor	University of Maryland at College Park, USA
Ehud Gudes	Ben-Gurion University, Israel
Joachim Hammer	University of Florida, USA
Zack Ives	University of Washington, USA
Hasan Jamil	Mississippi State University, USA
Christian S. Jensen	Aalborg University, Denmark
Pat Martin	Queen's University, Canada
Alberto Mendelzon	University of Toronto, Canada
Tova Milo	Tel-Aviv University, Israel
Danilo Montesi	Università di Bologna, Italy
Ami Motro	George Mason University, USA
Natasha Noy	Stanford University, USA
Luigi Palopoli	Università di Reggio Calabria, Italy
Avi Pfeffer	Harvard University, USA
Dimitris Plexousakis	University of Crete, Greece
Lucian Popa	IBM Almaden Research Center, USA
Louiqa Raschid	University of Maryland at College Park, USA
Rajeev Rastogi	Bell Laboratories, USA
Marek Rusinkiewicz	Telcordia Technologies, USA
Peter Scheuermann	NorthWestern University, USA
Avi Silberschatz	Bell Laboratories, USA
Peretz Shoval	Ben-Gurion University, Israel
Moshe Tennenholtz	Technion, Israel
Ouri Wolfson	University of Illinois at Chicago, USA
Carlo Zaniolo	UCLA, USA

Table of Contents

The Fifth Workshop on Next Generation Information Technologies and Systems (NGITS'2002)

Enabling Design-Centric eBusiness Applications

Arie Segev

Haas School of Business
University of California, Berkeley, CA 94720-1900
segev@haas.berkeley.edu

This talk discusses a research project on models for supporting of end-to-end eBusiness processes associated with design-centric applications. Design-centric applications are those where various business processes are initiated in the context of a design process. Our focus is on ad-hoc design environments that are characterized by collaborative processes among the initiators of the design and other players involved in moving the conceptual design to an actual implementation. This project is done in collaboration with the department of architecture, and the specific domain chosen for it is office design (either Business-to-Business or Business-to-Consumer). The conceptual results, however, apply to numerous other domains such as contract manufacturing, general construction projects, and designing and building one-of-a-kind complex products. The project examines next-generation eBusiness models and processes that best support collaborative office design, procurement of products and services, negotiations, and implementing (or building) the contacted solutions. A prototype system will be described and various implementation alternatives discussed.

A. Halevy and A. Gal (Eds.): NGITS 2002, LNCS 2382, p. 1, 2002.

Select-Project Queries over XML Documents*

Sara Cohen, Yaron Kanza, and Yehoshua Sagiv

Dept. of Computer Science,
The Hebrew University,
Jerusalem 91904, Israel
{sarina, yarok, sagiv}@cs.huji.ac.il

Abstract. This paper discusses evaluation of select-project (SP) queries over an XML document. A SP query consists of two parts: (1) a conjunction of conditions on values of labels (called the *selection*) and (2) a series of labels whose values should be outputed (called the *projection*). Query evaluation involves finding tuples of nodes that have the labels mentioned in the query and are related to one another other in a meaningful fashion. Several different semantics for query evaluation are given in this paper. Some of these semantics also take into account the possible presence of incomplete information. The complexity of query evaluation is analyzed and evaluation algorithms are described.

1 Introduction

Increasingly large amounts of data are accessible to the general public in the form of XML documents. It is difficult for the naive user to query XML and thus, potentially useful information may not reach its audience. Search engines are currently the only efficient way to query the Web. These engines do not exploit the structure of documents and hence, are not well suited for querying XML.

As a long-term goal, we would like to allow a natural-language interface for querying XML. It has been noted that the universal relation [9,12,13] is a first step towards facilitating natural-language querying of relational databases. This is because of the inherent simplicity of formulating a query against the universal relation. Such queries usually consist of only selection and projection and are called *select-project* or *SP* queries. Evaluating queries over the universal relation was studied in [11,7].

Many languages, such as XQuery [3] and XML-QL [6] have been proposed for querying XML. However, these languages are not suitable for a naive user. They also require a rather extensive knowledge of document structure in order to formulate a query correctly. The language EquiX [4] has been proposed for querying XML by a naive user. However, EquiX queries can only be formulated against a document with a DTD. A query language for XML must also take into consideration incomplete information. This has been studied in [2,8].

* Supported by Grant 96/01-1 from the Israel Science Foundation

A. Halevy and A. Gal (Eds.): NGITS 2002, LNCS 2382, pp. 2–13, 2002.

In this paper we explore the problem of answering an SP query formulated against an XML document. In order to formulate a query, users only need to know the names of the tags appearing in the document being queried. Queries consist of two parts:

- **Select:** boolean conditions on tags of a document (e.g., title = 'Cat in the Hat');
- **Project:** names of tags whose values should appear in the result (e.g., price).

Answering an SP query requires finding elements in a document that are *related* to one another in a *meaningful fashion*. Intuitively, such sets of elements correspond to rows in a universal relation that could be defined over an XML document. However, there are several questions that arise in this context:

- How can we decide when elements are related in a meaningful fashion? This becomes especially difficult when one considers the fact that documents may have varied structure.
- How can we deal with incompleteness in documents? If a document may be missing information, then we may have to discover whether a particular element is meaningfully related to an element that does not even appear in the document.

This paper deals with these questions.

Section 2 presents some necessary definitions and Section 3 present query semantics. In Sections 4 and 5 we discuss the complexity of answering SP queries over XML documents and present algorithms for query evaluation. Section 6 concludes.

2 Definitions

In this section we present some necessary definitions. We specify our data model and describe the syntax of *select-project* or *SP* queries.

Trees. We assume that there is a set $\mathcal{L}$ of labels and a set $\mathcal{A}$ of constants. An XML document is a tree T in which each *interior node* is associated with a *label* from $\mathcal{L}$ and each *leaf node* is associated with *value* from $\mathcal{A}$. We denote the label of an interior node n by $lbl(n)$ and the value of a leaf node n' by $val(n')$. We extend the *val* function to interior nodes n by defining $val(n)$ to be the concatenation of the values of its leaf descendents. In Figure 1 there is an example of such a tree, describing information about books. The nodes are numbered to allow easy reference.

Let T be a tree and let $n_1, \ldots, n_k$ be nodes in T. We denote by $lca\{n_1, \ldots, n_k\}$ the *lowest common ancestor* of $n_1, \ldots, n_k$. Let T_{lca} be the subtree of T rooted at $lca\{n_1, \ldots, n_k\}$. We denote by $T_{n_1,\ldots,n_k}$ the tree obtained by pruning from T_{lca} all nodes that are not ancestors of any of the nodes $n_1, \ldots, n_k$. We call this tree the *relationship tree* of $n_1, \ldots, n_k$. For example, in Figure 1, $lca\{15, 19, 21\}$ is 13. The relationship tree of 15, 19, 21 contains the nodes 13, 14, 15, 19 and 21.

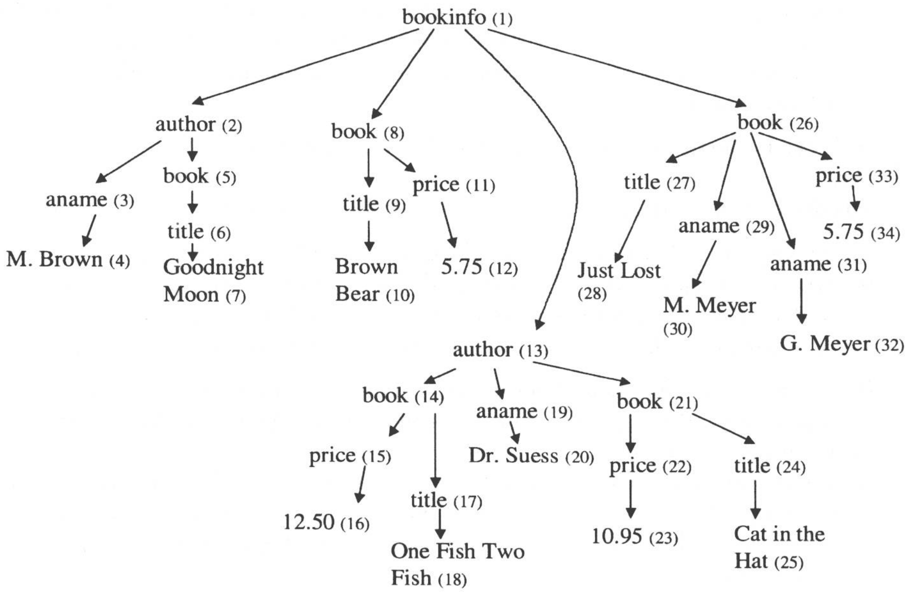

Fig. 1. An XML document describing books for sale

Relations. A *tuple* has the form $t = \{l_1 \colon a_1, \dots, l_k \colon a_k\}$ where l_i and a_i are a *column name* and a *value*, respectively. We will use $l_i(t)$ to denote the value a_i. We call $\{l_1, \dots, l_k\}$ the *signature* of t. A *relation* R is a set of tuples with the same signature, also called the signature of R.

Let N be a set of nodes in which no two nodes have the same label. Let L be the set of labels of nodes in N. The set N naturally gives rise to a tuple denoted t_N with signature L. Formally, if $n \in N$ and $lbl(n) = l$, then $l(t_N) = n$. Given a set of labels L' that contains L, the set N gives rise to a tuple with signature L', denoted $t_{L',N}$ by padding t_N with null values (denoted $\perp$), as necessary.

Select-Project Queries. A *condition* has the form $l\,\theta\,a$, $a\,\theta\,l$, or $l\,\theta\,l'$ where l, l' are labels, a in a constant, and θ is an operator (e.g., $<$, $=$, $\in$). A *query* has the form

$$q(l_1, \dots, l_k) \leftarrow c_1 \wedge \dots \wedge c_n \tag{1}$$

where l_i are labels and c_j are conditions. We do not allow a label to appear more than once among $l_1, \dots, l_k$. We sometimes denote the above query by $q(l_1, \dots, l_k)$ or simply by q. We call the conjunction $c_1 \wedge \dots \wedge c_n$ the *selection* of q and we call the sequence $(l_1, \dots, l_k)$ the *projection* of q. Note that we allow the selection to be an empty conjunction of conditions. We denote the empty conjunction by $\top$. We call queries with empty selections *project queries*. The set of labels appearing in either the selection or the projection of q is denoted $lbl(q)$. We will say that q is defined *over* the set $lbl(q)$.

Example 1. We present a few queries and their intuitive meaning.

- Pairs of titles and their respective prices:

$$\mathsf{q(title, price)} \leftarrow \top$$

- Titles and prices of books written by Dr. Suess that cost less than $12:

$$\mathsf{q(title, price)} \leftarrow \mathsf{(aname = \text{'Dr.Suess'})} \wedge \mathsf{(price < 12)}$$

- Title, author and price of books written by Meyer:

$$\mathsf{q(title, aname, price)} \leftarrow \mathsf{\text{'Meyer'}} \in \mathsf{aname}$$

3 Query Semantics

Consider a query q and a tree T. Suppose that $lbl(q) = \{l_1, \ldots, l_k\}$. Intuitively, we can understand query evaluation as a two-step process. First, compute a relation R which contains tuples of nodes from T with labels $l_1, \ldots, l_k$ that are *related* in a *meaningful fashion*. We call this relation the *relational image* of T with respect to $l_1, \ldots, l_k$ and it is denoted $R(q, T)$. Next, evaluate the selection and projection given in q on $R(q, T)$ to derive the query result.

In order to compute the relational image of a tree with respect to a set of nodes, we must be able to decide which nodes are related in a meaningful fashion in a given tree. We observe that nodes are not meaningfully related if their relationship tree contains two different nodes with the same label. Intuitively, two nodes in a tree that have the same label correspond to different entities in the world. Thus, in Figure 1, nodes 22 and 24 are related. However, nodes 22 and 27 are not since their relationship tree contains the label book twice. This reflects the intuition that 22 is the price of the book with title 24 and not the price of the book with title 27.

We formalize this idea. Let $n_1, \ldots, n_k$ be nodes in T. We say that $n_1, \ldots, n_k$ are *interconnected*, denoted $\approx(n_1, \ldots, n_k)$, if the tree $T_{n_1,\ldots,n_k}$ does not contain any two nodes with the same label. We say that N is *maximally interconnected* with respect to a set of labels L if there is no strict superset N' of N with labels from L that is also interconnected. Now, given a query q over labels L, let $\mathcal{S}$ be the set of all sets of maximally interconnected nodes in T with labels from L. The relational image of T w.r.t. L is defined as follows

$$R(q, T) := \{t_{L,N} \mid N \in \mathcal{S}\}.$$

The relational image of a tree contains nodes that are related to each other. However, some such relationships may be more significant than others. Nodes are more likely to be meaningfully related if their lowest common ancestor is relatively deep in the tree. If their lowest common ancestor is very high, then it is more likely that their relationship is coincidental. Thus, nodes 19 and 24 are more likely to be related then nodes 19 and 27. Note that in both these cases, the relationship trees do not have any repeated labels.

Let N be a set of interconnected nodes. We say that N' is an *improvement* on N, denoted $N \prec N'$ if

- $N \setminus N' = \{n_1\}$, $N' \setminus N = \{n_2\}$, $lbl(n_1) = lbl(n_2)$, i.e., N' is derived from N by replacing n_1 with n_2;
- For all nodes n in $N \cap N'$, the lowest common ancestor of $\{n_2, n\}$ is a descendent of the lowest common ancestor of $\{n_1, n\}$.

If N is maximal w.r.t. $\prec$, we say that N is $\prec$*-maximal*. We can remove some of the tuples in $R(q,T)$ that may be related in a less significant fashion, using the definition above. Let $\mathcal{S}^{\prec}$ be the set of all sets of maximally interconnected nodes in T that are also $\prec$-maximal. We define the $\prec$*-relational image* of T w.r.t. L as

$$R^{\prec}(q,T) := \{t_{L,N} \mid N \in \mathcal{S}^{\prec}\}.$$

Example 2. Consider the query

$$\mathsf{q(title, price) \leftarrow (aname = \text{`Dr. Suess'}) \wedge (price < 12)}.$$

The $\prec$-relational image of q over the tree presented in Figure 1 is

title	aname	price
Goodnight Moon	M. Brown	$\perp$
Brown Bear	$\perp$	5.75
One Fish Two Fish	Dr. Suess	12.50
Cat in the Hat	Dr. Suess	10.95
Just Lost	M. Meyer	5.75
Just Lost	G. Meyer	5.75

We extend the function *val* to sets of tuples of nodes in the natural fashion. Note that the tuple

$$(\text{`One Fish Two Fish'}, \text{`Dr. Suess'}, 10.95) = val(17, 19, 22)$$

is not in $R^{\prec}(q,T)$, since its relationship tree contains the same label twice. The tuple

$$(\text{`Just Lost'}, \text{`Dr. Suess'}, 5.75) = val(27, 19, 33)$$

is also not in $R^{\prec}(q,T)$ since $(27, 19, 33) \prec (27, 31, 33)$. However, ('Just Lost', 'Dr. Suess', 5.75) is in $R(q,T)$.