LNCS 3255

András Benczúr
János Demetrovics
Georg Gottlob (Eds.)

Advances in Databases and Information Systems

8th East European Conference, ADBIS 2004
Budapest, Hungary, September 2004
Proceedings

Springer

András Benczúr   János Demetrovics
Georg Gottlob (Eds.)

Advances in Databases and Information Systems

8th East European Conference, ADBIS 2004
Budapest, Hungary, September 22-25, 2004
Proceedings

Springer

Volume Editors

András Benczúr
Hungarian Academy of Sciences
Computer and Automation Institute, Informatics Laboratory
Lagymanyosi ut 11, 1111, Budapest, Hungary
E-mail: benczur@sztaki.hu

János Demetrovics
Hungarian Academy of Sciences
Computer and Automation Institute
Lagymanyosi ut 11, 1111 Budapest, Hungary
E-mail: demetrovics@sztaki.hu

Georg Gottlob
Technische Universität Wien, Institut für Informationssysteme
Favoritenstraße 9-11, 1040 Wien, Austria
E-mail: gottlob@dbai.tuwien.ac.at

Library of Congress Control Number: 2004112841

CR Subject Classification (1998): H.2, H.3, H.4, H.5, J.1

ISSN 0302-9743
ISBN 3-540-23243-5 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springeronline.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by PTP-Berlin, Protago-TeX-Production GmbH
Printed on acid-free paper SPIN: 11320166 06/3142 5 4 3 2 1 0

Lecture Notes in Computer Science 3255

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Preface

This volume continues the series of proceedings of the annual ADBIS conferences in the field of advances in databases and information systems.

ADBIS, founded by the Moscow ACM SIGMOD Chapter, was extended into a regular East-European conference in 1996 to establish a forum for promoting interaction and collaboration between the database research communities of Eastern and Central Europe and the rest of the world, and thus the ADBIS conferences provide an international platform for the presentation of research on database theory, development of advanced DBMS technologies, and their advanced applications.

The fact that this year eight of the countries that have representatives in the ADBIS Steering Committee became member countries of the European Union gives a new perspective on the role and mission of the ADBIS community.

Following Dresden (Germany, 2003), Bratislava (Slovakia, 2002), Vilnius (Lithuania, 2001), Prague (Czech Republic, as a joint ADBIS-DASFAA conference in 2000), Maribor (Slovenia, 1999), Poznan (Poland, 1998), and the starting place St. Petersburg (Russia, 1997), and preceding Tallin (Estonia, 2005), the next venue, the "tour of Eastern and Central Europe" of ADBIS conferences arrived at Budapest (Hungary, 2004).

The series of East-European Conferences on Advances in Databases and Information Systems continues, with a growing interest that is demonstrated in the number of submitted contributions. This year the international Program Committee had an intensive load in evaluating the 130 submitted papers.

As a tradition since 1998, the best 25–30 contributions were selected to be published in the Springer LNCS series. This year's volume contains 27 submitted papers and one invited paper.

In response to the high activity in paper submission and to fulfil the main objective of the ADBIS conferences, the organizers decided to give more room for presentation, and accepted from the PC ranking the next 25 papers for presentation. Thanks to the support from one of the organizing institutions, the Computer and Automation Research Institute of the Hungarian Academy of Sciences, these contributions will also be published, in a local volume of the proceedings in the series of publications of CARI.

Many individuals and institutions supported the organization of this conference and made this year's publication possible. Our special thanks go first to the authors and invited speaker for their outstanding and strenuous work and contributions. Also, the work of the Program Committee members who accepted the unexpectedly heavy load in reviewing a large number of contributions so that we were able to meet the deadline is gratefully acknowledged.

We received important financial support for the publication of the proceedings from the Faculty of Informatics of the Eötvös Loránd University.

Finally, we should express our thanks to Springer for supporting the publication of the proceedings in their LNCS series.

We do hope that the papers included in the present publication serve as a useful tool for becoming more familiar with recent research results in the field of databases and information systems.

July 2004

András Benczúr
János Demetrovics
Georg Gottlob

Organizing Committee

The 8th East-European Conference on Advances of Databases and Information Systems (ADBIS) was organized by the members of the Department of Information Systems of the Eötvös Loránd University, Budapest and of the Informatics Laboratory and Conference Department of the Computer and Automation Research Institute of the Hungarian Academy of Sciences, in cooperation with the Moscow ACM SIGMOD Chapter.

General Chair

András Benczúr, Head of the Department of Information Systems, Eötvös Loránd University, Budapest

General Co-chair

Georg Gottlob, Vienna University of Technology (TU Wien), Vienna

Local Program Committee Co-chair

János Demetrovics, Computer and Automation Research Institute of the Hungarian Academy of Sciences

Program Committee Chair

András Benczúr, Head of the Department of Information Systems, Eötvös Loránd University, Budapest

Program Committee Co-chairs

Georg Gottlob, Vienna University of Technology (TU Wien), Vienna
János Demetrovics, Computer and Automation Research Institute of the Hungarian Academy of Sciences

Program Committee

Albertas Caplinskas
Béla Uhrin
Bogdan Czejdo
Boris Novikov
Gunter Saake
Heinz Schweppe
Hele-Mai Haav
Jacek Plodzien
Jaroslav Pokorny
Johann Eder
Julius Stuller
Kjetil Norvag
Klaus Meyer-Wegener
Leonid Kalinichenko
Leopoldo Bertossi

Mihhail Matskin
Mikhail Kogalovsky
Nikolay Nikitchenko
Oscar Pastor
Peri Loucopoulos
Rainer Manthey
Remigijus Gustas
Robert Wrembel
Silvio Salza
Theo Härder
Tomas Hruska
Toni Urpí
Vladimir Zadorozhny
Yannis Manolopoulos
Vu Duc Thi
Peter Spyns
Luciano Baresi
Elyar Gasanov
Nghia Vu
V.B. Kudryavtsev
Pavol Navrat
Christelle Vangenot
Torben Bach Pedersen

Local Organizing Committee

Chairman: Gusztáv Hencsey (Computer and Automation Research Institute, Budapest, Hungary)
Secretary: Magdolna Zsivnovszki (Computer and Automation Research Institute, Budapest)
Tibor Márkus, Attila Kiss, Zsolt Hernáth, Zoltán Vinczellér, Balázs Kósa, András Molnár (Eötvös Loránd University, Budapest)
András Lukács, Béla Uhrin, Károly Móczár (Computer and Automation Research Institute, Budapest)

ADBIS Steering Committee Chair

Leonid Kalinichenko, Russian Academy of Sciences

ADBIS Steering Committee

András Benczúr (Hungary)
Radu Bercaru (Romania)
Albertas Caplinskas (Lithuania)
Johann Eder (Austria)
Janis Eiduks (Latvia)
Hele-Mai Haav (Estonia)
Mirjana Ivanovic (Yugoslavia)
Mikhail Kogalovsky (Russia)
Yannis Manolopoulos (Greece)
Rainer Manthey (Germany)
Tadeusz Morzy (Poland)
Pavol Navrat (Slovakia)
Boris Novikov (Russia)
Jaroslav Pokorny (Czech Republic)
Boris Rachev (Bulgaria)
Anatoly Stogny (Ukraine)
Bernhard Thalheim (Germany)
Tatjana Welzer (Slovenia)
Viacheslav Wolfengagen (Russia)

Sponsoring Institutions

Faculty of Informatics of the Eötvös Loránd University

Lecture Notes in Computer Science

For information about Vols. 1–3119

please contact your bookseller or Springer

Vol. 3255: A. Benczúr, J. Demetrovics, G. Gottlob (Eds.), Advances in Databases and Information Systems. XI, 423 pages. 2004.

Vol. 3253: Y. Lakhnech, S. Yovine (Eds.), Formal Techniques in Timed, Real-Time, and Fault-Tolerant Systems. X, 397 pages. 2004.

Vol. 3249: B. Buchberger, J.A. Campbell (Eds.), Artificial Intelligence and Symbolic Computation. X, 285 pages. 2004. (Subseries LNAI).

Vol. 3246: A. Apostolico, M. Melucci (Eds.), String Processing and Information Retrieval. XIV, 316 pages. 2004.

Vol. 3241: D. Kranzlmüller, P. Kacsuk, J.J. Dongarra (Eds.), Recent Advances in Parallel Virtual Machine and Message Passing Interface. XIII, 452 pages. 2004.

Vol. 3240: I. Jonassen, J. Kim (Eds.), Algorithms in Bioinformatics. IX, 476 pages. 2004. (Subseries LNBI).

Vol. 3239: G. Nicosia, V. Cutello, P.J. Bentley, J. Timmis (Eds.), Artificial Immune Systems. XII, 444 pages. 2004.

Vol. 3238: S. Biundo, T. Frühwirth, G. Palm (Eds.), KI 2004: Advances in Artificial Intelligence. XI, 467 pages. 2004. (Subseries LNAI).

Vol. 3232: R. Heery, L. Lyon (Eds.), Research and Advanced Technology for Digital Libraries. XV, 528 pages. 2004.

Vol. 3224: E. Jonsson, A. Valdes, M. Almgren (Eds.), Recent Advances in Intrusion Detection. XII, 315 pages. 2004.

Vol. 3223: K. Slind, A. Bunker, G. Gopalakrishnan (Eds.), Theorem Proving in Higher Order Logics. VIII, 337 pages. 2004.

Vol. 3221: S. Albers, T. Radzik (Eds.), Algorithms – ESA 2004. XVIII, 836 pages. 2004.

Vol. 3220: J.C. Lester, R.M. Vicari, F. Paraguaçu (Eds.), Intelligent Tutoring Systems. XXI, 920 pages. 2004.

Vol. 3210: J. Marcinkowski, A. Tarlecki (Eds.), Computer Science Logic. XI, 520 pages. 2004.

Vol. 3208: H.J. Ohlbach, S. Schaffert (Eds.), Principles and Practice of Semantic Web Reasoning. VII, 165 pages. 2004.

Vol. 3207: L.T. Yang, M. Guo, G.R. Gao, N.K. Jha (Eds.), Embedded and Ubiquitous Computing. XX, 1116 pages. 2004.

Vol. 3206: P. Sojka, I. Kopecek, K. Pala (Eds.), Text, Speech and Dialogue. XIII, 667 pages. 2004. (Subseries LNAI).

Vol. 3205: N. Davies, E. Mynatt, I. Siio (Eds.), UbiComp 2004: Ubiquitous Computing. XVI, 452 pages. 2004.

Vol. 3203: J. Becker, M. Platzner, S. Vernalde (Eds.), Field Programmable Logic and Application. XXX, 1198 pages. 2004.

Vol. 3202: J.-F. Boulicaut, F. Esposito, F. Giannotti, D. Pedreschi (Eds.), Knowledge Discovery in Databases: PKDD 2004. XIX, 560 pages. 2004. (Subseries LNAI).

Vol. 3201: J.-F. Boulicaut, F. Esposito, F. Giannotti, D. Pedreschi (Eds.), Machine Learning: ECML 2004. XVIII, 580 pages. 2004. (Subseries LNAI).

Vol. 3199: H. Schepers (Ed.), Software and Compilers for Embedded Systems. X, 259 pages. 2004.

Vol. 3198: G.-J. de Vreede, L.A. Guerrero, G. Marín Raventós (Eds.), Groupware: Design, Implementation and Use. XI, 378 pages. 2004.

Vol. 3194: R. Camacho, R. King, A. Srinivasan (Eds.), Inductive Logic Programming. XI, 361 pages. 2004. (Subseries LNAI).

Vol. 3193: P. Samarati, P. Ryan, D. Gollmann, R. Molva (Eds.), Computer Security – ESORICS 2004. X, 457 pages. 2004.

Vol. 3192: C. Bussler, D. Fensel (Eds.), Artificial Intelligence: Methodology, Systems, and Applications. XIII, 522 pages. 2004. (Subseries LNAI).

Vol. 3190: Y. Luo (Ed.), Cooperative Design, Visualization, and Engineering. IX, 248 pages. 2004.

Vol. 3189: P.-C. Yew, J. Xue (Eds.), Advances in Computer Systems Architecture. XVII, 598 pages. 2004.

Vol. 3186: Z. Bellahsène, T. Milo, M. Rys, D. Suciu, R. Unland (Eds.), Database and XML Technologies. X, 235 pages. 2004.

Vol. 3185: M. Bernardo, F. Corradini (Eds.), Formal Methods for the Design of Real-Time Systems. VII, 295 pages. 2004.

Vol. 3184: S. Katsikas, J. Lopez, G. Pernul (Eds.), Trust and Privacy in Digital Business. XI, 299 pages. 2004.

Vol. 3183: R. Traunmüller (Ed.), Electronic Government. XIX, 583 pages. 2004.

Vol. 3182: K. Bauknecht, M. Bichler, B. Pröll (Eds.), E-Commerce and Web Technologies. XI, 370 pages. 2004.

Vol. 3181: Y. Kambayashi, M. Mohania, W. Wöß (Eds.), Data Warehousing and Knowledge Discovery. XIV, 412 pages. 2004.

Vol. 3180: F. Galindo, M. Takizawa, R. Traunmüller (Eds.), Database and Expert Systems Applications. XXI, 972 pages. 2004.

Vol. 3179: F.J. Perales, B.A. Draper (Eds.), Articulated Motion and Deformable Objects. XI, 270 pages. 2004.

Vol. 3178: W. Jonker, M. Petkovic (Eds.), Secure Data Management. VIII, 219 pages. 2004.

Vol. 3177: Z.R. Yang, H. Yin, R. Everson (Eds.), Intelligent Data Engineering and Automated Learning – IDEAL 2004. XVIII, 852 pages. 2004.

Vol. 3176: O. Bousquet, U. von Luxburg, G. Rätsch (Eds.), Advanced Lectures on Machine Learning. IX, 241 pages. 2004. (Subseries LNAI).

Vol. 3175: C.E. Rasmussen, H.H. Bülthoff, B. Schölkopf, M.A. Giese (Eds.), Pattern Recognition. XVIII, 581 pages. 2004.

Vol. 3174: F. Yin, J. Wang, C. Guo (Eds.), Advances in Neural Networks - ISNN 2004. XXXV, 1021 pages. 2004.

Vol. 3173: F. Yin, J. Wang, C. Guo (Eds.), Advances in Neural Networks – ISNN 2004. XXXV, 1041 pages. 2004.

Vol. 3172: M. Dorigo, M. Birattari, C. Blum, L. M. Gambardella, F. Mondada, T. Stützle (Eds.), Ant Colony, Optimization and Swarm Intelligence. XII, 434 pages. 2004.

Vol. 3170: P. Gardner, N. Yoshida (Eds.), CONCUR 2004 - Concurrency Theory. XIII, 529 pages. 2004.

Vol. 3166: M. Rauterberg (Ed.), Entertainment Computing – ICEC 2004. XXIII, 617 pages. 2004.

Vol. 3163: S. Marinai, A. Dengel (Eds.), Document Analysis Systems VI. XI, 564 pages. 2004.

Vol. 3162: R. Downey, M. Fellows, F. Dehne (Eds.), Parameterized and Exact Computation. X, 293 pages. 2004.

Vol. 3160: S. Brewster, M. Dunlop (Eds.), Mobile Human-Computer Interaction – MobileHCI 2004. XVII, 541 pages. 2004.

Vol. 3159: U. Visser, Intelligent Information Integration for the Semantic Web. XIV, 150 pages. 2004. (Subseries LNAI).

Vol. 3158: I. Nikolaidis, M. Barbeau, E. Kranakis (Eds.), Ad-Hoc, Mobile, and Wireless Networks. IX, 344 pages. 2004.

Vol. 3157: C. Zhang, H. W. Guesgen, W.K. Yeap (Eds.), PRICAI 2004: Trends in Artificial Intelligence. XX, 1023 pages. 2004. (Subseries LNAI).

Vol. 3156: M. Joye, J.-J. Quisquater (Eds.), Cryptographic Hardware and Embedded Systems - CHES 2004. XIII, 455 pages. 2004.

Vol. 3155: P. Funk, P.A. González Calero (Eds.), Advances in Case-Based Reasoning. XIII, 822 pages. 2004. (Subseries LNAI).

Vol. 3154: R.L. Nord (Ed.), Software Product Lines. XIV, 334 pages. 2004.

Vol. 3153: J. Fiala, V. Koubek, J. Kratochvíl (Eds.), Mathematical Foundations of Computer Science 2004. XIV, 902 pages. 2004.

Vol. 3152: M. Franklin (Ed.), Advances in Cryptology – CRYPTO 2004. XI, 579 pages. 2004.

Vol. 3150: G.-Z. Yang, T. Jiang (Eds.), Medical Imaging and Augmented Reality. XII, 378 pages. 2004.

Vol. 3149: M. Danelutto, M. Vanneschi, D. Laforenza (Eds.), Euro-Par 2004 Parallel Processing. XXXIV, 1081 pages. 2004.

Vol. 3148: R. Giacobazzi (Ed.), Static Analysis. XI, 393 pages. 2004.

Vol. 3146: P. Érdi, A. Esposito, M. Marinaro, S. Scarpetta (Eds.), Computational Neuroscience: Cortical Dynamics. XI, 161 pages. 2004.

Vol. 3144: M. Papatriantafilou, P. Hunel (Eds.), Principles of Distributed Systems. XI, 246 pages. 2004.

Vol. 3143: W. Liu, Y. Shi, Q. Li (Eds.), Advances in Web-Based Learning – ICWL 2004. XIV, 459 pages. 2004.

Vol. 3142: J. Diaz, J. Karhumäki, A. Lepistö, D. Sannella (Eds.), Automata, Languages and Programming. XIX, 1253 pages. 2004.

Vol. 3140: N. Koch, P. Fraternali, M. Wirsing (Eds.), Web Engineering. XXI, 623 pages. 2004.

Vol. 3139: F. Iida, R. Pfeifer, L. Steels, Y. Kuniyoshi (Eds.), Embodied Artificial Intelligence. IX, 331 pages. 2004. (Subseries LNAI).

Vol. 3138: A. Fred, T. Caelli, R.P.W. Duin, A. Campilho, D.d. Ridder (Eds.), Structural, Syntactic, and Statistical Pattern Recognition. XXII, 1168 pages. 2004.

Vol. 3137: P. De Bra, W. Nejdl (Eds.), Adaptive Hypermedia and Adaptive Web-Based Systems. XIV, 442 pages. 2004.

Vol. 3136: F. Meziane, E. Métais (Eds.), Natural Language Processing and Information Systems. XII, 436 pages. 2004.

Vol. 3134: C. Zannier, H. Erdogmus, L. Lindstrom (Eds.), Extreme Programming and Agile Methods - XP/Agile Universe 2004. XIV, 233 pages. 2004.

Vol. 3133: A.D. Pimentel, S. Vassiliadis (Eds.), Computer Systems: Architectures, Modeling, and Simulation. XIII, 562 pages. 2004.

Vol. 3132: B. Demoen, V. Lifschitz (Eds.), Logic Programming. XII, 480 pages. 2004.

Vol. 3131: V. Torra, Y. Narukawa (Eds.), Modeling Decisions for Artificial Intelligence. XI, 327 pages. 2004. (Subseries LNAI).

Vol. 3130: A. Syropoulos, K. Berry, Y. Haralambous, B. Hughes, S. Peter, J. Plaice (Eds.), TeX, XML, and Digital Typography. VIII, 265 pages. 2004.

Vol. 3129: Q. Li, G. Wang, L. Feng (Eds.), Advances in Web-Age Information Management. XVII, 753 pages. 2004.

Vol. 3128: D. Asonov (Ed.), Querying Databases Privately. IX, 115 pages. 2004.

Vol. 3127: K.E. Wolff, H.D. Pfeiffer, H.S. Delugach (Eds.), Conceptual Structures at Work. XI, 403 pages. 2004. (Subseries LNAI).

Vol. 3126: P. Dini, P. Lorenz, J.N.d. Souza (Eds.), Service Assurance with Partial and Intermittent Resources. XI, 312 pages. 2004.

Vol. 3125: D. Kozen (Ed.), Mathematics of Program Construction. X, 401 pages. 2004.

Vol. 3124: J.N. de Souza, P. Dini, P. Lorenz (Eds.), Telecommunications and Networking - ICT 2004. XXVI, 1390 pages. 2004.

Vol. 3123: A. Belz, R. Evans, P. Piwek (Eds.), Natural Language Generation. X, 219 pages. 2004. (Subseries LNAI).

Vol. 3122: K. Jansen, S. Khanna, J.D.P. Rolim, D. Ron (Eds.), Approximation, Randomization, and Combinatorial Optimization. IX, 428 pages. 2004.

Vol. 3121: S. Nikoletseas, J.D.P. Rolim (Eds.), Algorithmic Aspects of Wireless Sensor Networks. X, 201 pages. 2004.

Vol. 3120: J. Shawe-Taylor, Y. Singer (Eds.), Learning Theory. X, 648 pages. 2004. (Subseries LNAI).

Table of Contents

XML and Semistructured Databases

Physical Database Design and Query Evaluation

Transaction Management and Workflow Systems

Query Processing and Data Streams

Spatial Databases

Agents and Mobile Systems

Quantifier-Elimination for the First-Order Theory of Boolean Algebras with Linear Cardinality Constraints*

Peter Revesz

Department of Computer Science and Engineering
University of Nebraska-Lincoln, Lincoln, NE 68588, USA
revesz@cse.unl.edu

Abstract. We present for the first-order theory of atomic Boolean algebras of sets with linear cardinality constraints a quantifier elimination algorithm. In the case of atomic Boolean algebras of sets, this is a new generalization of Boole's well-known variable elimination method for conjunctions of Boolean equality constraints. We also explain the connection of this new logical result with the evaluation of relational calculus queries on constraint databases that contain Boolean linear cardinality constraints.

1 Introduction

Constraint relations, usually in the form of semi-algebraic or semi-linear sets, provide a natural description of data in many problems. What programming language could be designed to incorporate such constraint relations? Jaffar and Lassez [27] proposed in a landmark paper *constraint logic programming*, with the idea of extending Prolog, and its usual top-down evaluation style, with constraint solving (which replaces unification in Prolog). That is, in each rule application after the substitution of subgoals by constraint tuples, the evaluation needs to test the satisfiability of the constraints, and proceed forward or backtrack according to the result.

As an alternative way of incorporating constraint relations, in a database framework, Kanellakis, Kuper, and Revesz [29,30] proposed *constraint query languages* as an extension of relational calculus with the basic insight that the evaluation of relational calculus queries on X-type constraint relations reduces to a quantifier elimination [15,16] in the first-order theory of X.[1] As an example from [29,30], if the constraint relations are semi-algebraic relations, then

* This work was supported in part by USA National Science Foundation grant EIA-0091530.

[1] They also considered various bottom-up methods of evaluating Datalog queries and the computational complexity and expressive power of constraint queries. Since relational calculus is closely tied with practical query languages like SQL, it has captured the most attention in the database area.

A. Benczúr, J. Demetrovics, G. Gottlob (Eds.): ADBIS 2004, LNCS 3255, pp. 1–21, 2004.

the quantifier elimination for real closed fields [2,3,12,13,40,54] can be used to evaluate queries.

There are advantages and disadvantages of both styles of program/query evaluations. Constraint logic programs have the advantage that their implementation can be based on only constraint satisfiability testing, which is usually easier and faster than quantifier elimination required by constraint relational calculus. On the other hand, the termination of constraint logic programs is not guaranteed, except in cases with a limited expressive power. For example, for negation-free Datalog queries with integer (gap)-order constraints the termination of both the tuple-recognition problem [14] and the least fixed point query evaluation [41, 42] can be guaranteed. When either negation or addition constraints are also allowed, then termination cannot be guaranteed. In contrast, the evaluation of constraint relational calculus queries have a guaranteed termination, provided there is a suitable effective quantifier elimination method.

While many other comparisons can be made (see the surveys [28,43] and the books [31,33,47]), these seem to be the most important. Their importance becomes clear when we consider the expected users. Professional programmers can write any software in any programming language and everything could be neatly hidden (usually under some kind of options menu) from the users. In contrast, database systems provide for the users *not ready-made programs but a easy-to-use high-level programming language*, in which they can write their own simple programs. It is unthinkable that this programming language not terminate, and, in fact, run efficiently. Therefore constraint database research focused on the efficient evaluation of simple non-recursive query languages.[2]

The constraint database field made initially a rapid progress by taking off-the-shelf some quantifier elimination methods. Semi-linear sets as constraint relations are allowed in several prototype constraint database systems [8,21,24,25, 48,49] that use Fourier-Motzkin quantifier elimination for linear inequality constraints [18]. The latest version of the DISCO system [9,50] implements Boolean equality constraints using Boole's existential quantifier elimination method for conjunctions of Boolean equality constraints.

Relational algebra queries were considered in [20,42,46]. As in relational databases, the algebraic operators are essential for the efficient evaluation of queries. In fact, in the above systems logical expressions in the form of relational calculus, SQL, and Datalog rules are translated into relational algebra.

There were also deep and interesting questions about the relative expressive power and computational complexity of relational versus constraint query languages. Some results in this area include [4,23,37,45] and a nice survey of these can be found in Chapters 3 and 4 of [31].

[2] Of course, many database-based products also provide menus to the users. However, the users of database-based products are only indirect users of database systems. The direct users of database systems are application developers, who routinely embed SQL expressions into their programs. Thanks to today's database systems, they need to worry less about termination, efficiency, and many other issues than yesterday's programmers needed while developing software products.

After these initial successes, it became clear that further progress may be possible only by extending the quantifier elimination methods. Hence researchers who happily got their hands dirty doing implementations found themselves back at the mathematical drawing table.

The limitations of quantifier elimination seemed to be most poignant for Boolean algebras. It turns out that for conjunctions of Boolean equality *and inequality* constraints (which seems to require just a slight extension of Boole's method) no quantifier elimination is possible. Let us see an example, phrased as a lemma.

Lemma 1. *There is no quantifier-free formula of Boolean equality and inequality constraints that is equivalent in every Boolean algebra to the following formula:*

$$\exists d \ (d \sqcap g \neq \bot) \wedge (\bar{d} \sqcap g \neq \bot)$$

where d and g are variables and $\bot$ is the zero element of the Boolean algebra. □

Consider the Boolean algebra of sets, with the one element being the names of all persons, the zero element being the empty set, the domain being the powerset (set of all subset of the one element).

In the formula variable d may be the set of students who took a database systems class, variable g may be the set of students who graduate this semester. Then the formula expresses the statement that "some graduating student took a database systems class, and some graduating student did not take a database systems class." This formula implies that g has at least two elements, that is, the cardinality of g is at least two, denoted as:

$$|g| \geq 2$$

But this fact can not be expressed by any quantifier-free formula with Boolean equality and inequality constraints and g as the only variable.

Lemma 1 implies that there is no general quantifier elimination method for formulas of Boolean equality and inequality constraints. This negative result was noted by several researchers, who then advocated approximations. For example, Helm et al. [26] approximate the result by a formula of Boolean equality and inequality constraints. Can we do better than just an approximation?

The only hopeful development in the quantifier elimination area was by Marriott and Odersky [32] who showed that formulas with equality and inequality constraints admit quantifier elimination for the special case of *atomless* Boolean algebras. However, many Boolean algebras are not atomless but atomic. How can we deal with those Boolean algebras? Could any subset of atomic Boolean algebras also admit quantifier elimination? In this paper we show that the atomic Boolean algebras of sets, i.e., Boolean algebras where the Boolean algebra operators are interpreted as the usual set operators of union, intersection and complement with respect to the one element, also admit quantifier elimination, in spite of the pessimistic looking result of Lemma 1.

Let us take a closer look at the Lemma. Surprisingly, the condition $|g| \geq 2$ is not only necessary, but it is also sufficient. That is, for any Boolean algebra of sets if G is any set with at least two elements, then we can find a set D such that the above formula holds. Therefore, $|g| \geq 2$ is exactly the quantifier-free formula that we would like to have as a result of the quantifier elimination. However, quantifier elimination techniques are normally required to give back equivalent quantifier-free formulas with the same type of constraints as the input. This condition is commonly called being *closed* under the set of constraints. This raises the interesting question of what happens if we allow cardinality constraints in our formulas.

While cardinality constraints on sets are considered by many authors, and interesting algorithms are developed for testing the satisfiability of a conjunction of cardinality constraints, there were, to our knowledge, no algorithms given for quantifier elimination for atomic Boolean algebras of sets with cardinality constraints.

Calvanese and Lenzerini [11,10] study cardinality constraints that occur in ER-diagrams and ISA hierarchies. They give a method to test the satisfiability of a schema. This is a special case of cardinality constraints, because the ER-diagrams do not contain inequality constraints.

Ohlbach and Koehler [35,36] consider a simple description logic with cardinality constraints. They give methods to test subsumption and satisfiability of their formulas, but they do not consider quantifier elimination.

Seipel and Geske [51] use constraint logic programming to solve conjunctions of cardinality constraints. Their set of constraint logic programming [27] rules is sound but incomplete.

Surprisingly, in this paper, we show that the augmented formulas, called *Boolean linear cardinality constraint formulas*, admit quantifier elimination. It is surprising that by adding to the set of atomic constraints, the problem of quantifier elimination becomes easier, not harder. Indeed, the end result, which is our quantifier elimination method described in this paper, may strike the reader as simple. But the finding of the trick of adding cardinality constraints for the sake of performing quantifier elimination is not obvious as shown by the following history summarized in Figure 1. In the figure the arrows point from less to more expressive Boolean constraint theories, but the labels on them indicate that the Boolean algebra needs to be of a certain type. Let's describe Figure 1 in some detail (please see Section 2 for definitions of unfamiliar terms).

Precedence between variables: A naive elimination of variables from a set of Boolean precedence constraints between variables or constants (in the case of algebras of sets set containment constraints between sets) occurs in syllogistic reasoning. Namely, the syllogistic rule *if all x are y, and all y are z, then all x are z* yields a simple elimination of the y variable. Such syllogisms were described already by Aristotle and developed further in medieval times and can be used as the basis of eliminating variables. Srivastava, Ramakrishnan, and Revesz [52] gave an existential quantifier elimination method for a special subset of the Boolean algebra of sets. They considered existentially quantified formulas with