Lecture Notes in Computer Science 1358
Edited by G. Goos, J. Hartmanis and J. van Leeuwen

Bernhard Thalheim Leonid Libkin (Eds.)

Semantics in Databases

Springer

Volume Editors

Leonid Libkin
Bell Laboratories
600 Mountain Avenue, Murray Hill, NJ 07974, USA
E-mail: libkin@research.bell-labs.com

Bernhard Thalheim
Brandenburg Technical University at Cottbus
Computer Science Department
PO Box 101344, D-03013 Cottbus, Germany
E-mail: thalheim@informatik.tu-cottbus.de

Cataloging-in-Publication data applied for

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Semantics in databases / Bernhard Thalheim ; Leonid Libkin (ed.). - Berlin ; Heidelberg ; New York ; Barcelona ; Budapest ; Hong Kong ; London ; Milan ; Paris ; Santa Clara ; Singapore ; Tokyo : Springer, 1998
(Lecture notes in computer science ; 1358)
ISBN 3-540-64199-8

CR Subject Classification (1991): H.2, F3.1, H.3.3

ISSN 0302-9743
ISBN 3-540-64199-8 Springer-Verlag Berlin Heidelberg New York

Typesetting: Camera-ready by author
SPIN 10661361 06/3142 – 5 4 3 2 1 0 Printed on acid-free paper

FOREWORD

In the early days of database research, issues related to database semantics played a prominent role, and papers discussing database models, conceptual design, integrity constraints and normalization often dominated major database conferences. This began to change more than a decade ago, and nowadays those issues do not appear to be part of the mainstream research. Why is this so? Has the field been exhausted? Or perhaps the focus of research on semantics in databases has shifted to new areas as the field of databases itself matured and began branching out in new directions?

As an attempt to see where the work on databases semantics is now and where it is heading, Bernhard Thalheim organized a workshop on *Semantics in Databases*. It was held in Rez near Prague, in January 1995, following the 5th International Conference on Database Theory. The three-day workshop was informal and did not have published proceedings. It featured more than a dozen one-hour talks, and plenty of time was left for informal discussions. At the conclusion of the workshop, the participants decided to prepare a volume containing full versions of papers presented at the workshop. At that time, Leonid Libkin joined Bernhard Thalheim as a co-editor.

We immediately agreed on the following. First, it was clear that two important areas in database research – spatial databases and database transformations – have not been represented at the workshop. We extended our invitation, to write papers about semantic issues in spatial databases and database transformations, to two groups that were unable to attend the workshop. Second, we decided that all the papers must be reviewed according to a very high standard. Thus, every paper was assigned to two reviewers who were asked to consider it to be closer to journal-style reviewing. At the end, several submissions were rejected. We also asked the authors to write papers that present at least a partial survey of a respective area. We hope that all the papers in this volume are self-contained and require only some basic knowledge of database fundamentals.

Most submissions arrived during the second half of 1995. All papers were reviewed by the end of 1996, and at that time all acceptance and rejection decisions were made. By this time, we collected all the revised versions. Each of them went through the second round of refereeing.

This volume contains nine papers dealing with various aspects of semantics in databases. We hope that these papers will demonstrate that there are new and interesting developments in the field – both in a more traditional branch, dealing with integrity constraints and conceptual modeling, and in new areas of database theory, such as constraint and spatial databases. Several papers show how formal semantics helps understand some of the classical issues – object-orientation, incomplete information, and database transformations.

The first group consists of four papers dealing with traditional aspects of database semantics. The paper *Achievements of Relational Database Schema Design Theory Revisited* by Joachim Biskup surveys some of the best known achievements of design theory from the point of view of four main heuristics that typically guide the design process. When certain aspects of an application are enumerated, they are represented by the formats which are statically declared in the schema. The Separation of Aspects heuristic postulates that each declared format enumerates exactly one aspect, and the Separation of Specializations heuristic postulates that each declared format conforms to exactly one specialization of an aspect. The schema must be complete in the sense that every meaningful aspect that is not enumerated by a declared format must be inferable (typically in a given query language); this is called Inferential Completeness. Finally, the Unique Flavor heuristic is used to express the requirement that all meaningful aspects be understood by expressing their basic attributes. The paper argues that normal forms assure the first requirement, Separation of Aspects, while Inferential Completeness is formalized as view support (which in turn can be formalized in a variety of ways). The Unique Flavor heuristic is formalized by considering the hypergraph structure of a database schema, and corresponds to the familiar notions of acyclicity. The paper surveys some of the major results related to those concepts.

Another paper dealing with design theory is *Redundancy Elimination and a New Normal Form for Relational Database Design* by Millist W. Vincent. This paper contributes to understanding and justifying the use of normal forms from a semantic perspective. It gives a formal definition of redundancy and redundancy-free normal form, and connects this definition with the known normal forms in the case when constraints contain functional and join dependencies. It is first shown that the redundancy-free normal form implies the fourth normal form. Then the paper introduces a new normal form, called key-complete, and shows that it is equivalent to redundancy-free normal form when constraints include functional and join dependencies. This new normal form turns out to be weaker than the projection-join normal form, which is a normal form proposed for join dependencies almost 20 years ago.

The third paper in this group is *An Informal and Efficient Approach for Obtaining Semantic Constraints Using Sample Data and Natural Language Processing* by Meike Albrecht, Edith Buchholz, Antje Düsterhöft, and Bernhard Thalheim. The efficiency of a database depends crucially on the correctness of the design, and hence on the knowledge of database semantics. Thus, acquisition of semantics constraints is critical for the database performance; at the same time, it is a very difficult task because many database designers may have difficulty with the formal definitions of database constraints. The paper describes the system called RADD (Rapid Application and Database Development) that is designed to overcome these problems. The system conducts a natural language dialog with the designer, in an attempt to produce a conceptual design and acquire semantic constraints. The acquisition stage is followed by validation, and finally by the selection of candidates for constraints.

The last paper in the group is *The Additivity Problem for Data Dependencies in*

Incomplete Relational Databases by Mark Levene and George Loizou. If a relational database does not contain null values, then satisfying a set of constraints Φ is the same as satisfying every single constraint $\varphi \in \Phi$ – this is called the additivity property. When null values are present, the definition of satisfaction is usually replaced by satisfaction in a possible world for an incomplete database. The paper shows that when constraints include functional and/or inclusion dependencies, the additivity property fails if null values are present. The paper characterizes additivity for databases with incomplete information in three scenarios: when constraints include only functional dependencies, when constraints include only unary inclusion dependencies, and when both functional and unary inclusion dependencies are allowed. Furthermore, properties of constraints that ensure additivity in these cases are tractable.

The second group consists of three papers whose major theme is understanding the semantics in well established database areas: object-oriented databases, partial information, and database transformations.

The paper *The Evolving Algebra Semantics of Class and Role Hierarchies* by Georg Gottlob, Gerti Kappel, and Michael Schrefl explains that several features of the object-oriented model, such as methods and inheritance, cannot be handled in first-order logic. It suggests using evolving algebras to provide semantics of object-oriented data models. Evolving algebras were introduced by Y. Gurevich as a framework for defining operational semantics of programming languages. The paper introduces an object-oriented data definition and manipulation language called EasyOBL. It provides support for many object-oriented features, including encapsulation and message passing, dynamic object creation, inheritance and dynamic binding; it can also be statically typechecked. The paper presents an evolving algebra semantics of the main EasyOBL features.

The paper *A Semantics-Based Approach to Design of Query Languages for Partial Information* by Leonid Libkin argues that while most of work on partial information in databases asks which operations of standard languages can be performed correctly in the presence of nulls, it is perhaps worthwhile to understand the semantics of partiality and use it as a guide to design languages specifically tailored to deal with partial information. The paper identifies several sources of partiality, such as missing or disjunctive information and conflicts. It develops a common semantic framework for them, that is based on ideas used in the semantics of programming languages, and represents partiality via orderings on the domains of types. The paper describes the basic principles of an approach that turns operations naturally associated with the datatypes into programming syntax. This approach is applied to partial information: the analysis of semantic domains of types reveals what the main programming primitives should be. The paper shows how resulting languages subsume some of those known in the literature. It also discusses constraints within the ordered semantics framework and extensions to recursive types.

The paper *Semantics of Database Transformations* by Peter Buneman, Susan Davidson, and Anthony Kosky starts by surveying a number of approaches to transforming

instances of one or more source schema into instance of some target schema. By assessing their strengths and weaknesses, the paper develops formal requirements for specifying database transformation. In particular, it concentrates on ensuring correctness of transformations. In order to be able to reason about both transformation and constraints, one needs a unifying framework for them. The paper presents such a framework. It describes the language called WOL that specifies both transformations and constraints. Its data model supports object identities, classes, and complex objects. A formal semantics of database schema, instances and keys is presented. The language itself is based on Horn clause expressions, and has a small number of primitives; at the same time, it is powerful enough to express most constraints typically found in established data models. The language is also capable of expressing constraints that span multiple databases, and can be used to specify transformations and ensure their correctness. The approach of WOL differs from a large body of research on transforming schemas, as it deals with transforming both schemas and the underlying data.

The other two papers deal with constraint databases and spatial databases. The constraint model, introduced by Kanellakis, Kuper, and Revesz in 1990, is designed to deal with finite representations of potentially infinite sets. A typical example is in spatial databases, where a region can be represented by constraints defining it; for example, $x^2 + y^2 \leq 1$ is a finite representation of the infinite set of points on the real plane that satisfy this constraint.

The paper *Constraint Databases: A Survey* by Peter Revesz gives a comprehensive survey of the area. It shows how to extend standard relational languages – relational calculus, datalog, and stratified datalog – to the constraint setting, where databases are finite sets of constraints. The paper explains how queries are evaluated, and gives a survey of complexity results for a variety of constraints, including dense order constraints, linear and polynomial (in)equality constraints, and integer gap-order constraints. It also briefly discusses optimizations, surveys results on expressive power of constraint query languages, and gives pointers to several prototype systems.

The paper *Semantics in Spatial Databases* by Bart Kuijpers, Jan Paredaens, and Luc Vandeurzen discusses two models for spatial applications – the linear model and the topological model – and languages to query databases in both models. The linear model is essentially the constraint model with linear inequality constraints, such as $2x + 5y \leq 3$. This model is well suited for geographical and CAD/CAM applications. While often polynomial constraints give a better representation, spatial operations on curved data are hard to implement efficiently, and approximation with linear functions is often sufficient and easy to understand. As the language for this model, the authors propose the linear spatial calculus, which is essentially relational calculus with linear constraints. The second model is topological; a typical application is finding a route on a railway or highway map. The exact geographical information is not important, but the information about connections (e.g., there is a link from city A to city B) is. The paper defines a model for representing this kind of information and a first-order language for querying spatial databases in the topological data model.

We would like to thank all the attendees at the workshop for very productive discussions that led to this volume. We thank the referees, Catriel Beeri, Joachim Biskup, Francois Bry, Wolfram Clauss, Michael Doherty, Antje Düsterhöft, Stacy Finkelstein, Georg Gottlob, Tim Griffin, Rick Hull, Achim Jung, Gyula Katona, Anthony Kosky, Mark Levene, Rainer Manthey, Rona Machlin, Heikki Mannila, Doron Peled, Peter Revesz, Klaus-Dieter Schewe, Dan Suciu, Ramesh Viswanathan, and Limsoon Wong for their efforts.

November 1997

Leonid Libkin
Bell Labs
Bernhard Thalheim
Cottbus University

Contents