

Relational Database Technology Monographs In

"Specification and transformation of programs" is short for a methodology of software development where, from a formal specification of a problem to be solved, programs correctly solving that problem are constructed by stepwise application of formal, semantics-preserving transformation rules. The approach considers programming as a formal activity. Consequently, it requires some mathematical maturity and, above all, the will to try something new. A somewhat experienced programmer or a third- or fourth-year student in computer science should be able to master most of this material - at least, this is the level I have aimed at. This book is primarily intended as a general introductory textbook on transformational methodology. As with any methodology, reading and understanding is necessary but not sufficient. Therefore, most of the chapters contain a set of exercises for practising as homework. Solutions to these exercises exist and can, in principle, be obtained at nominal cost from the author upon request on appropriate letterhead. In addition, the book also can be seen as a comprehensive account of the particular transformational methodology developed within the Munich CHIP project.

Fully revised, updated, and expanded, Relational Database Design and Implementation, Third Edition is the most lucid and effective introduction to the subject available for IT/IS professionals interested in honing their skills in database design, implementation, and administration. This book provides the conceptual and practical information necessary to develop a design and management scheme that ensures data accuracy and user satisfaction while optimizing performance, regardless of experience level or choice of DBMS. The book begins by reviewing basic concepts of databases and database design, then briefly reviews the SQL one would use to create databases. Topics such as the relational data model, normalization, data entities and Codd's Rules (and why they are important) are covered clearly and concisely but without resorting to "Dummies"-style talking down to the reader. Supporting the book's step-by-step instruction are three NEW case studies illustrating database planning, analysis, design, and management practices. In addition to these real-world examples, which include object-relational design techniques, an entirely NEW section consisting of three chapters is devoted to database implementation and management issues. * Principles needed to understand the basis of good relational database design and implementation practices. * Examples to illustrate core concepts for enhanced comprehension and to put the book's practical instruction to work. * Methods for tailoring DB design to the environment in which the database will run and the uses to which it will be put. * Design approaches that ensure data accuracy and consistency. * Examples of how design can inhibit or boost database application performance. * Object-relational design techniques, benefits, and examples. * Instructions on how to choose and use a normalization technique. * Guidelines for understanding and applying Codd's rules. * Tools to implement a relational design using SQL. * Techniques for using CASE tools for database design.

Fully revised and updated, Relational Database Design, Second Edition is the most lucid and effective introduction to relational database design available. Here, you'll find the conceptual and practical information you need to develop a design that ensures data accuracy and user satisfaction while optimizing performance, regardless of your experience level or choice of DBMS. Supporting the book's step-by-step instruction are three case studies illustrating the planning, analysis, and design steps involved in arriving at a sound design. These real-world examples include object-relational design techniques, which are addressed in greater detail in a new chapter devoted entirely to this timely subject. * Concepts you need to master to put the book's practical instruction to work. * Methods for tailoring your design to the environment in which the database will run and the uses to which it will be put. * Design approaches that ensure data accuracy and consistency. * Examples of how design can inhibit or boost database application performance. * Object-relational design techniques, benefits, and examples. * Instructions on how to choose and use a normalization technique. * Guidelines for understanding and applying Codd's rules. * Tools to implement a relational design using SQL. * Techniques for using CASE tools for database design.

This book brings all of the elements of database design together in a single volume, saving the reader the time and expense of making multiple purchases. It consolidates both introductory and advanced topics, thereby covering the gamut of database design methodology? from ER and UML techniques, to conceptual data modeling and table transformation, to storing XML and querying moving objects databases. The proposed book expertly combines the finest database design material from the Morgan Kaufmann portfolio. Individual chapters are derived from a select group of MK books authored by the best and brightest in the field. These chapters are combined into one comprehensive volume in a way that allows it to be used as a reference work for those interested in new and developing aspects of database design. This book represents a quick and efficient way to unite valuable content from leading database design experts, thereby creating a definitive, one-stop-shopping opportunity for customers to receive the information they would otherwise need to round up from separate sources. Chapters contributed by various recognized experts in the field let the reader remain up to date and fully informed from multiple viewpoints. Details multiple relational models and modeling languages, enhancing the reader's technical expertise and familiarity with design-related requirements specification. Coverage of both theory and practice brings all of the elements of database design together in a single volume, saving the reader the time and expense of making multiple purchases.

Programming with Specifications

Computational Geometry

Object-Oriented and Object-Relational Design

Beauty Is Our Business

Object-Oriented Database Programming

An Introduction

More than anything else, this book is a tribute to Edsger W. Dijkstra, on the occasion of his sixtieth birthday, by just a few of those fortunate enough to be influenced by him and his work and to be called his friend or relation, his master, colleague, or pupil. This book contains fifty-four technical contributions in different areas of endeavor, although many of them deal with an area of particular concern to Dijkstra: programming. Each contribution is relatively short and could be digested in one sitting. Together, they form a nice cross section of the discipline of programming at the beginning of the nineties. While many know of Dijkstra's technical contributions, they may not be aware of his ultimate goal, the mastery of complexity in mathematics and computing science. He has forcefully argued that beauty and elegance are essential to this mastery. The title of this book, chosen to reflect his ultimate goal, comes from a sentence in an article of his on some beautiful arguments using mathematical induction: "... when we recognize the battle against chaos, mess, and unmastered complexity as one of computing sci- ence's major callings, we must admit that 'Beauty Is Our Business'."

In the 1930s, mathematical logicians studied the notion of "effective comput ability" using such notions as recursive functions, λ -calculus, and Turing machines. The 1940s saw the construction of the first electronic computers, and the next 20 years saw the evolution of higher-level programming languages in which programs could be written in a convenient fashion independent (thanks to compilers and interpreters) of the architecture of any specific machine. The development of such languages led in turn to the general analysis of questions of syntax, structuring strings of symbols which could count as legal programs, and semantics, determining the "meaning" of a program, for example, as the function it computes in transforming input data to output results. An important approach to semantics, pioneered by Floyd, Hoare, and Wirth, is called assertion semantics: given a specification of which assertions (preconditions) on input data should guarantee that the results satisfy desired assertions (postconditions) on output data, one seeks a logical proof that the program satisfies its specification. An alternative approach, pioneered by Scott and Strachey, is called denotational semantics: it offers algebraic techniques for characterizing the denotation of (i , e), the function computed by a program-the properties of the program can then be checked by direct comparison of the denotation with the specification. This book is an introduction to denotational semantics. More specifically, we introduce the reader to two approaches to denotational semantics: the order semantics of Scott and Strachey and our own partially additive semantics.

The Information Systems: The Information Value Chain the authors explain the interrelationships between the decision support, decision modelling, and information systems. The first two parts of the book focus on the interdisciplinary decision support framework, in which mathematical programming (optimization) is taken as the inference engine. The role of business analytics and its relationship with recent developments in organisational theory, decision modelling, information systems and information technology are considered in depth. Part three of the book includes a carefully chosen selection of invited contributions from internationally-known researchers. These contributions are thought-provoking and cover key decision modelling and information systems issues. The final part of the book covers contemporary developments in the related area of business intelligence considered within an organizational context. The topics cover computing delivered across the web, management decision-making, and socio-economic challenges that lie ahead. It is now well accepted that globalisation and the impact of digital economy are profound; and the role of e-business and the delivery of decision models (business analytics) across the net lead to a challenging business environment. In this dynamic setting, decision support is one of the few interdisciplinary frameworks that can be rapidly adopted and deployed to so that businesses can survive and prosper by meeting these new challenges.

There are many kinds of books on formal logic. Some have philosophers as their intended audience, some mathematicians, some computer scientists. Although there is a common core to all such books they will be very dif ferent in emphasis, methods, and even appearance. This book is intended for computer scientists. But even this is not precise withi th computer sci ence formal logic turns up in a number of areas, from program verification to logic programming to artificial intelligence. This book is intended for computer scientists interested in automated theorem proving in classical logic. To be more precise yet, it is essentially a theoretical treatment, not a how-to book, although how-to issues are not neglected. This does not mean, of course, that the book will be of no interest to philosophers or mathematicians. It does contain a thorough presentation of formal logic and many proof techniques, and as such it contains all the material one would expect to find in a course in formal logic covering completeness but not incompleteness issues. The first item to be addressed is, what are we talking about and why are we interested in it. We are primarily talking about truth as used in mathematical discourse, and our interest in it is, or should be, self-evident. Truth is a semantic concept, so we begin with models and their properties. These are used to define our subject.

The Science of Programming

6th International Conference on Extending Database Technology, Valencia, Spain, March 23-27, 1998.

The Synthesizer Generator

String-Rewriting Systems

Theory and Applications

Relational Databases

The creation of the text really began in 1976 with the author being involved with a group of researchers at Stanford University and the Naval Ocean Systems Center, San Diego. At that time, adaptive techniques were more laboratory (and mental) curiosities than the accepted and pervasive categories of signal processing that they have become. Over the last 10 years, adaptive filters have become standard components in telephony, data communications, and signal detection and tracking systems. Their use and consumer acceptance will undoubtedly only increase in the future. The mathematical principles underlying adaptive signal processing were initially fascinating and were my first experience in seeing applied mathematics work for a paycheck. Since that time, the application of even more advanced mathematical techniques have kept the area of adaptive signal processing as exciting as those initial days. The text seeks to be a bridge between the open literature in the professional journals, which is usually quite concentrated, concise, and advanced, and the graduate classroom and research environment where underlying principles are often more important.

This book provides a superb introduction to and overview of the MIT PI System for custom VLSI placement and routing. Alan Sher man has done an excellent job of collecting and clearly presenting material that was previously available only in various theses, confer ence papers, and memoranda. He has provided here a balanced and comprehensive presentation of the key ideas and techniques used in PI, discussing part of his own Ph. D. work (primarily on the place ment problem) in the context of the overall design of PI and the contributions of the many other PI team members. I began the PI Project in 1981 after leaving first-hand how dif ficult it is to manually place modules and route interconnections in a custom VLSI chip. In 1980 Adil Shami, Leonard Adleman, and I designed a custom VLSI chip for performing RSA encryption/decryption [226]. I became fascinated with the combinatorial and algorithmic questions arising in placement and routing, and to gain active research in these areas. The PI Project was started in the belief that many of the most interesting research issues would arise during an actual implementation effort, and secondarily in the hope that a practically useful tool might result. The belief was well-founded, but I had underestimated the difficulty of building a large easily-used software tool for a complex domain: the PI soft ware should be considered as a prototype implementation validating the design choices made.

This book is a detailed account of the Synthesizer Generator, a system for creat ing specialized editors that are customized for editing particular languages. The book is intended for those with an interest in software tools and in methods for building interactive systems. It is a must for people who are using the Syn thesizer Generator to build editors because it provides extensive discussions of how to write editor specifications. The book should also be valuable for people who are building specialized editors "By hand," without using an editor generating tool. The need to manage the development of large software systems is one of the most pressing problems faced by computer programmers. An important aspect of this problem is the design of new tools to aid interactive program develop ment. The Synthesizer Generator permits one to create specialized editors that are tailored for editing a particular language. In program editors built with the Synthesizer Generator, knowledge about the language is used to continuously assess whether a program contains errors and to determine where such errors occur. The information is then displayed on the terminal screen to provide feed back to the programmer as the program is developed and modified.

Algebraic Approaches to Program Semantics The PI Project studies some of the main algorithmic tools of computer algebra, covering such topics as Gr ö bner bases, characteristic sets, resultants and semialgebraic sets. The main purpose of the book is to acquaint advanced undergraduate and graduate students in computer science, engineering and mathematics with the algorithmic ideas in computer algebra so that they could do research in computational algebra or understand the algorithms underlying many popular symbolic computational systems: Mathematica, Maple or Axiom, for instance. Also, researchers in robotics, solid modeling, computational geometry and automated theorem proving community may find it useful as symbolic algebraic techniques have begun to play an important role in these areas. The book, while being self-contained, is written at an advanced level and deals with the subject at an appropriate depth. The book is accessible to computer science students with no previous algebraic training. Some mathematical readers, on the other hand, may find it interesting to see how algorithmic constructions have been used to provide fresh proofs for some classical theorems. The book also contains a large number of exercises with solutions to selected exercises, thus making it ideal as a textbook or for self-study.

Clearly Explained

Algebraic Algebra

Relational Database Technology

First-Order Logic and Automated Theorem Proving

Logic for Applications

NIDA Research Monograph

"Temporal Information Processing Technology and Its Applications" systematically studies temporal information processing technology and its applications. The book covers following subjects: 1) time model, calculus and logic; 2) temporal data models, semantics of temporal variable 'how' temporal database concepts; 3) temporal query language, a typical temporal database management system; TempDB; 4) temporal extension on XML, workflow and knowledge base; and, 5) implementation patterns of temporal applications, a typical example of temporal application. The book is intended for researchers, practitioners and graduate students of databases, data/knowledge management and temporal information processing. Dr. Yong Tang is a professor at the Computer School, South China Normal University, China.

Object-oriented databases were originally developed as an alternative to relational database technology for the representation, storage, and access of non-traditional data forms that were increasingly found in advanced applications of database technology. After much debate regarding object-oriented versus relational database technology, object-oriented extensions were eventually incorporated into relational technology to create object-relational databases. Both object-oriented databases and object-relational databases, collectively known as object databases, provide inherent support for object features, such as object identity, classes, inheritance hierarchies, and associations between classes using object references. This monograph presents the fundamentals of object databases, with a specific focus on conceptual modeling of object database designs. After an introduction to the fundamental concepts of object-oriented data, the monograph provides a review of object-oriented conceptual modeling techniques using side-by-side Enhanced Entity Relationship diagrams and Unified Modeling Language conceptual class diagrams that feature class hierarchies with specialization constraints and object associations. These object-oriented conceptual models provide the basis for introducing case studies that illustrate the use of object features within the design of object-oriented and object-relational databases. For the object-oriented database perspective, the Object Data Management Group data definition language provides a portable, language-independent specification of an object schema, together with an SQL-like object query language, LINGO (Language INtegrated Query) is presented as a case study of an object query language together with its use in the db4o open-source object-oriented database. For the object-relational perspective, the object-relational features of the SQL standard are presented together with an accompanying case study of the object-relational features of Oracle. For completeness of coverage, an appendix provides a mapping of object-oriented conceptual designs to the relational model and its associated constraints. Table of Contents: List of Figures / List of Tables / Introduction to Object Databases / Object-Oriented Databases / Object-Relational Databases

There are several theories of programming. The first useful theory, often called "Hoare's Logic", is still probably the most widely known. In it, a specification is a pair of predicates: a precondition and postcondition (these and all technical terms will be defined in due course). Another popular and closely related theory by Dijkstra uses the weakest precondition predicate transformer, which is a function from programs and postconditions to preconditions. Jones's Vienna Development Method has been used to advantage in some industries; in it, a specification is a pair of predicates (as in Hoare's Logic), but the second predicate is a relation. Temporal Logic is yet another formalism that introduces some special operators and quantifiers to describe some aspects of computation. The theory in this book is simpler than any of those just mentioned. In it, a specification is just a boolean expression. Refinement is just ordinary implication. This theory is also more general than those just mentioned, applying to both terminating and nonterminating computation, to both sequential and parallel computation, to both stand-alone and interactive computation. And it includes time bounds, both for algorithm classification and for tightly constrained real-time applications.

This book constitutes the thoroughly refereed joint post-proceedings of the 6th International Conference on Relational Methods in Computer Science, ReIMICS 2001 and the 1st Workshop of COST Action 274 TARSKI, Theory and Application of Relational Structures as Knowledge Instruments held in Oosterwijk, The Netherlands, in October 2001. The 20 revised full papers presented together with an invited paper were carefully reviewed and selected. The papers are organized in topical sections on algebraic and logical foundations of real world relations, mechanization of relational reasoning, and relational scaling and preferences.

Temporal Information Processing Technology and Its Applications

New Directions for Database Systems

State of the Art Report 14:5

6th International Conference, ReIMICS 2001 and 1st Workshop of COST Action 274 TARSKI Oosterwijk, The Netherlands, October 16–21, 2001 Revised Papers

A Birthday Salute to Edsger W. Dijkstra

Predicate Calculus and Program Semantics

Mathematical Foundations of Computer Science, Volume I is the first of two volumes presenting topics from mathematics (mostly discrete mathematics) which have proven relevant and useful to computer science. This volume treats basic topics, mostly of a set-theoretical nature (sets, functions and relations, partially ordered sets, induction, enumerability, and diagonalization) and illustrates the usefulness of mathematical ideas by presenting applications to computer science. Readers will find useful applications in algorithms, databases, semantics of programming languages, formal languages, theory of computation, and program verification. The material is treated in a straightforward, systematic, and rigorous manner. The volume is organized by mathematical area, making the material easily accessible to the upper-undergraduate students in mathematics as well as in computer science and each chapter contains a large number of exercises. The volume can be used as a textbook, but it will also be useful to researchers and professionals who want a thorough presentation of the mathematical tools they need in a single source. In addition, the book can be used effectively as supplementary reading material in computer science courses, particularly those courses which involve the semantics of programming languages, formal languages and automata, and logic programming.

Algebraic Approaches to Program Semantics studies some of the main algorithmic tools of computer algebra, covering such topics as Gr ö bner bases, characteristic sets, resultants and semialgebraic sets. The main purpose of the book is to acquaint advanced undergraduate and graduate students in computer science, engineering and mathematics with the algorithmic ideas in computer algebra so that they could do research in computational algebra or understand the algorithms underlying many popular symbolic computational systems: Mathematica, Maple or Axiom, for instance. Also, researchers in robotics, solid modeling, computational geometry and automated theorem proving community may find it useful as symbolic algebraic techniques have begun to play an important role in these areas. The book, while being self-contained, is written at an advanced level and deals with the subject at an appropriate depth. The book is accessible to computer science students with no previous algebraic training. Some mathematical readers, on the other hand, may find it interesting to see how algorithmic constructions have been used to provide fresh proofs for some classical theorems. The book also contains a large number of exercises with solutions to selected exercises, thus making it ideal as a textbook or for self-study.

Describes basic programming principles and their step-by- step applications.Numerous examples are included.

Topics • what this book is about, • its intended audience, • what the reader ought to know, • how the book is organized, • acknowledgements. Specifications express information about a program that is not normally part of the program, and often cannot be expressed in a programming lan guage. In the past, the word "specification" has sometimes been used to refer to somewhat vague documentation written in English. But today it indicates a precise statement, written in a machine processable language, about the purpose and behavior of a program. Specifications are written in languages that are just as precise as programming languages, but have additional capabilities that increase their power of expression. The term nology formal specification is sometimes used to emphasize the modern meaning. For us, all specifications are formal. The use of specifications as an integral part of a program opens up a whole new area of programming - programming with specifications. This book describes how to use specifications in the process of building programs, debugging them, and interfacing them with other programs. It deals with a new trend in programming - the evolution of specification languages from the current generation of programming languages. And it describes new strategies and styles of programming that utilize specifications. The trend is just beginning, and the reader, having finished this book, will viii Preface certainly see that there is much yet to be done and to be discovered about programming with specifications.

Algebraic Approaches to Program Semantics

Relational Database Design and Implementation

Fundamentals of Object Databases

Towards a Unified Access Control Model [Elektronische Ressource]]

ECOOP '97 - Object-Oriented Programming

Deductive Databases and Their Applications

This book provides a structured introduction to the verification of sequen tial and concurrent programs. It thus belongs to the area of programming languages but at the same time it is firmly based on mathematical logic. In logic one usually studies fixed syntactic or semantic objects. This is not necessarily the case in the area of program verification. The objects studied here, namely programs, do not have a standard syntax, their semantics can be defined in many different ways, and several approaches to their verification can be contemplated. These differences lead to various difficult design decisions. Even though we restrict our attention here to one programming styl e - imperative pro gramming - we are still confronted with a veritable cornucopia of pro gramming constructs from which an appropriate selection has to be made.

Relational Databases explores the major advances in relational databases and provides a balanced analysis of the state of the art in relational databases. Topics covered include capture and analysis of data placement requirements; distributed relational database systems; data dependency manipulation in database schemata; and relational database support for computer graphics and computer aided design. This book is divided into three sections and begins with an overview of the theory and practice of distributed systems, using the example of INGRES from Relational Technology as illustration. The following chapters focus on whether relational and relational-like systems actually meet business needs; IBM's Structured Query Language/Data System (SQL/DS); tools for database design and programming; and Secondary Access Methods and the problem of secondary index selection. A number of quantitative models for assessing the performance of physical databases are also described. This text concludes by assessing some of the most conspicuous trends in relational database research and development. This monograph will be of interest to database designers.

*Addresses the evolution of database management, technologies and applications along with the progress and endeavors of new research areas. "-P. xiii.

This book constitutes the refereed proceedings of the 6th International Conference on Extending Database Technology, EDBT '98, held in Valencia, Spain, in March 1998. The 32 revised full papers presented together with one invited keynote were selected from a total of 191 submissions. The book is divided in sections on similarity search and indexing, query optimization on the Web, Algorithms for data mining, Mathematical Foundations of Computer Science

A Practical Theory of Programming

Formal Models and Semantics

11th European Conference, Jyväskylä, Finland, June 9 - 13, 1997, Proceedings

An Introduction to ANNA, A Language for Specifying Ada Programs

Database Design: Knowledge

From the reviews: "This book offers a coherent treatment, at the graduate textbook level, of the field that has come to be known in the last decade or so as computational geometry. The book is well organized and lucidly written; a timely contribution by two founders of the field. It clearly demonstrates that computational geometry in the plane is now a fairly well-understood branch of computer science and mathematics. It also points the way to the solution of the more challenging problems in dimensions higher than two." #Mathematical Reviews# ".... This remarkable book is a comprehensive and systematic study on research results obtained especially in the last ten years. The very clear presentation concentrates on basic ideas, fundamental combinatorial structures, and crucial algorithmic techniques. The plenty of results is cleverly organized following these guidelines and within the framework of some detailed case studies. A large number of figures and examples also aid the understanding of the material. Therefore, it can be highly recommended as an early graduate text but it should prove also to be essential to researchers and professionals in applied fields of computer-aided design, computer graphics, and robotics." #Biometrical Journal#2

The second part of this Handbook presents a choice of material on the theory of automata and rewriting systems, the foundations of modern programming languages, logics for program specification and verification, and some chapters on the theoretic modelling of advanced information processing.

This volume is intended for researchers, practitioners, and members of the business community interested in the shape of data management in the years to come. The volume is both retrospective and future oriented and the chapters recapitulate current 1980s database research and applications.

The subject of this book is string-rewriting systems. It is generally accepted that string-rewriting was first introduced by Axel Thue in the early part of this century. In the 1960's and early 1970's, it received renewed attention due to interest in formal language theory. In the 1980's and 1990's, it has received more interest since it can be viewed as a special case of term rewriting, a subject that has become important in the study of automated deduction. Today, string-rewriting is studied by researchers in theoretical computer science and also by researchers interested in the foundations of artificial intelligence. A sketch of the way that the subject has developed is contained in Chapter 0, and the reader is advised to begin with that chapter. Both authors have been active in the field and have lectured on the subject in several universities. Lecture notes have been produced and distributed. This monograph is a result of revising and rewriting those notes. It represents an attempt by the authors to present the concepts that the authors consider to be most fundamental and to gather together the most useful results in such a way that they can be understood and used in studies relating to more general rewriting, to automated deduction, and to algo rithmic problems of algebraic structures. This monograph is written for independent study by researchers in the oretical computer science or in the foundations of artificial intelligence.

Adaptive Signal Processing

Decision Modelling and Information Systems

Encyclopedia of Database Technologies and Applications

A Formal Approach to Software Development

Sets, Relations, and Induction

Specification and Transformation of Programs

Deductive Databases and their Applications is an introductory text aimed at undergraduate students with some knowledge of database and information systems. The text comes complete with exercises and solutions to encourage students to tackle problems practically as well as theoretically. The author presents the origins of deductive databases in Prologue before proceeding to analyse the main deductive database paradigm - the data-log model. The final chapters are dedicated to closely related topics such as prepositional expert systems, integrity constraint specification and evaluation, and update propagation. Particular attention is paid to CASE tool repositories.

Here, the authors strive to change the way logic and discrete math are taught in computer science and mathematics: while many books treat logic simply as another topic of study, this one is unique in its willingness to go one step further. The book traets logic as a basic tool which may be applied in essentially every other area.

"When do the Lebesgue-Bochner function spaces contain a copy or a complemented copy of any of the classical sequence spaces?" This problem and the analogous one for vector- valued continuous function spaces have attracted quite a lot of research activity in the last twenty-five years. The aim of this monograph is to give a detailed exposition of the answers to these questions, providing a unified and self-contained treatment. It presents a great number of results, methods and techniques, which are useful for any researcher in Banach spaces and, in general, in Functional Analysis. This book is written at a graduate student level, assuming the basics in Banach space theory.

In writing this book, our goal was to produce a text suitable for a first course in mathematical logic more attuned than the traditional textbooks to the recent dramatic growth in the applications of logic to computer science. Thus our choice of topics has been heavily influenced by such applications. Of course, we cover the basic traditional topics - syntax, semantics, soundness, completeness and compactness - as well as a few more advanced results such as the theorems of Skolem-Lowenheim and Herbrand. Much of our book, however, deals with other less traditional topics. Resolution theorem proving plays a major role in our treatment of logic, especially in its application to Logic Programming and PROLOG. We deal extensively with the mathematical foundations of all three of these subjects. In addition, we include two chapters on nonclassical logic- modal and intuitionistic - that are becoming increasingly important in computer science. We develop the basic material on the syntax and semantics (via Kripke frames) for each of these logics. In both cases, our approach to formal proofs, soundness and completeness uses modifications of the same tableau method introduced for classical logic. We indicate how it can easily be adapted to various other special types of modal log ics. A number of more advanced topics (including nonmonotonic logic) are also briefly introduced both in the nonclassical logic chapters and in the material on Logic Programming and PROLOG.

Advances in Database Technology - EDBT '98

A Logical Approach to Discrete Math

VLSI Placement and Routing: The PI Project

Historical Social Research

Relational Methods in Computer Science

The Information Value Chain

This book presents a unified collection of concepts, tools, and techniques that constitute the most important technology available today for the design and implementation of information systems. The framework adopted for this integration goal is the one offered by the relational model of data, its applica tions, and implementations in multuser and distributed environments. The topics presented in the book include conceptual modeling of application environments using the relational model, formal properties of that model, and tools such as relational languages which go with it, techniques for the logical and physical design of relational database systems and their imple mentations. The book attempts to develop an integrated methodology for addressing all these issues on the basis of the relational approach and various research and practical developments related to that approach. This book is the only one available today that presents such an inte gration. The diversity of approaches to database data models, to logical and physical database design, to database application programming, and to use and imple mentation of database systems calls for a common framework for all of them. It has become difficult to study modern database technology with out such a unified approach to a diversity of results developed during the vigorous growth of the database area in recent years, let alone to teach a course on the subject. This book provides research communities in Europe and Japan with an overview of scientific results and experiences achieved using innovative methods and approaches in computer science and other disciplines, which have a common interest in understanding and solving problems on information modelling and knowledge bases, as well as applying the results of research to practice. The topics of research covered in this book mainly concentrate on a variety of themes in the domain of theory and practice of information modelling, conceptual modelling, design and specification of information systems, software engineering, databases and knowledge bases. We also aim to recognize and study new areas of modelling and knowledge bases to which more attention should be paid. Therefore philosophy and logic, cognitive science, knowledge management, linguistics and management science are relevant areas, too.

This booklet presents a reasonably self-contained theory of predicate trans former semantics. Predicate transformers were introduced by one of us (EWD) as a means for defining programming language semantics in a way that would directly support the systematic development of programs from their formal specifications. They met their original goal, but as time went on and program derivation became a more and more formal activity, the informal introduction used and the fact that many of their properties had never been proved became more and more unsatisfactory. And so did the original exclusion of unbounded nondeterminism. In 1982 we started to remedy these shortcomings. This little monograph is a result of that work. A possible - and even likely- criticism is that anyone sufficiently versed in lattice theory can easily derive all of our results by itself. This criticism would be correct but somewhat beside the point. The first remark is that the average book on lattice theory is several times fatter (and probably less self contained) than this booklet. The second remark is that the predicate transformer semantics provided only one of the reasons for going through the pains of publication.

Relational Database Design Clearly Explained

A System for Constructing Language-Based Editors

Verification of Sequential and Concurrent Programs

Information Modelling and Knowledge Bases XV

An Access Definition and Query Language