Mathematical Logic For Computer Science 2nd Edition

At the intersection of mathematics, computer science, and philosophy, mathematical logic examines the power and limitations of formal mathematical thinking. In this expansion of Leary's user-friendly 1st edition, readers with no previous study in the field are introduced to the basics of model theory, proof theory, and computability theory. The text is designed to be used either in an upper division undergraduate classroom, or for self study. Updating the 1st Edition's treatment of languages, structures, and deductions, leading to rigorous proofs of Godel's First and Second Incompleteness Theorems, the expanded 2nd Edition includes a new introduction to incompleteness through computability as well as solutions to selected exercises.

In the recent decades mathematical logic has become more and more important in computer science and, in general, in system engineering. In fact, by definition, it is the wey of expressing our reasoning in terms of mathematical formalism, thus supplying it with the typical rigor and precision of mathematics. Not by chance, automatic information processing is now pervasive and we find it practically in any human activity and artefact, from embedded, safety-critical systems, to e-commerce, to social networks, etc. Such a pervasiveness and the consequent heterogeneity of the involved systems mandate much more generality in the formalism supporting the engineering activity than traditional specialized models such as, e.g., those for electric circuits and mechanical engines: mathematical logic, paired with computer applications, provides such generality.

Demonstrating the different roles that logic plays in the disciplines of computer science, mathematics, and philosophy, this concise undergraduate textbook covers select topics from three different areas of logic: proof theory, computability theory, and nonclassical logic. The book balances accessibility, breadth, and rigor, and is designed so that its materials will fit into a single semester. Its distinctive presentation of traditional logic material will enhance readers' capabilities and mathematical maturity. The proof theory portion presents classical propositional logic and first-order logic using a computer-oriented (resolution) formal system. Linear resolution and its connection to the programming language Prolog are also treated. The computability component offers a machine model and mathematical model for computation, proves the equivalence of the two approaches, and includes famous decision problems unsolvable by an algorithm. The section on nonclassical logic discusses the shortcomings of classical logic in its treatment of implication and an alternate approach that improves upon it: Anderson and Belnap's relevance logic. Applications are included in each section. The material on a four-valued semantics for relevance logic is presented in textbook form for the first time. Aimed at upper-level undergraduates of moderate analytical background, Three Views of Logic will be useful in a variety of classroom settings. Gives an exceptionally broad view of logic Treats traditional logic in a modern format Presents relevance logic with applications Provides an ideal text for a variety of one-semester upper-level undergraduate courses

This introduction to first-order logic clearly works out the role of first-order logic in the foundations of mathematics, particularly the two basic questions of the range of the axiomatic method and of theorem-proving by machines. It covers several advanced topics not commonly treated in introductory texts, such as Fraïssé's characterization of elementary equivalence, Lindström's theorem on the maximality of first-order logic, and the fundamentals of logic programming.

An introduction to applying predicate logic to testing and verification of software and digital circuits that focuses on applications rather than theory. Computer scientists use logic for testing and verification of software and digital circuits, but many computer science students study logic only in the context of traditional mathematics, encountering the subject in a few lectures and a handful of problem sets in a discrete math course. This book offers a more substantive and rigorous approach to logic that focuses on applications in computer science. Topics covered include predicate logic, equation-based software, automated testing and theorem proving, and large-scale computation. Formalism is emphasized, and the book employs three formal notations: traditional algebraic formulas of propositional and predicate logic; digital circuit diagrams; and the widely used partially automated theorem prover, ACL2, which provides an accessible introduction to mechanized formalism. For readers who want to see formalization in action, the text presents examples using Proof Pad, a lightweight ACL2 environment. Readers will not become ALC2 experts, but will learn how mechanized logic can benefit software and hardware engineers. In addition, 180 exercises, some of them extremely challenging, offer opportunities for problem solving. There are no prerequisites beyond high school algebra. Programming experience is not required to understand the book's equation-based approach. The book can be used in undergraduate courses in logic for computer science and introduction to computer science and in math courses for computer science students.

Fascinating study of the origin and nature of mathematical thought, including relation of mathematics and science, 20th-century developments, impact of computers, and more.Includes 34 illustrations. 1968 edition."

Logic is, and should be, the core subject area of modern mathemat ics. The blueprint for twentieth century mathematical thought, thanks to Hilbert and Bourbaki, is the axiomatic development of the subject. As a result, logic plays a central conceptual role. At the same time, mathematical logic has grown into one of the most recondite areas of mathematics. Most of modern logic is inaccessible to all but the special ist. Yet there is a need for many mathematical scientists-not just those engaged in mathematical research-to become conversant with the key ideas of logic. The Handbook of Mathematical Logic, edited by Jon Bar wise, is in point of fact a handbook written by logicians for other mathe maticians. It was, at the time of its writing, encyclopedic, authoritative, and up-to-the-moment. But it was, and remains, a comprehensive and authoritative book for the cognoscenti. The encyclopedic Handbook of Logic in Computer Science by Abramsky, Gabbay, and Maibaum is a wonderful resource for the professional. But it is overwhelming for the casual user. There is need for a book that introduces important logic terminology and concepts to the working mathematical scientist who has only a passing acquaintance with logic. Thus the present work has a different target audience. The intent of this handbook is to present the elements of modern logic, including many current topics, to the reader having only basic mathe matical literacy.

In the recent decades mathematical logic has become more and more important in computer science and, in general, in system engineering. In fact, by definition, it is the wey of expressing our reasoning in terms of mathematical formalism, thus supplying it with the typical rigor and precision of mathematics. Not by chance, automatic information processing is now pervasive and we find it practically in any human activity and artefact, from embedded, safety-critical systems, to e-commerce, to social networks, etc. Such a pervasiveness and the consequent heterogeneity of the involved systems mandate much more generality in the formalism supporting the engineering activity than traditional specialized models such as, e.g., those for electric circuits and mechanical engines: mathematical logic, paired with computer applications, provides such generality

This modern introduction to the foundations of logic and mathematics not only takes theory into account, but also treats in some detail applications that have a substantial impact on everyday life (loans and mortgages, bar codes, public-key cryptography). A first college-level

introduction to logic, proofs, sets, number theory, and graph theory, and an excellent self-study reference and resource for instructors.
Classical and Fuzzy Concepts in Mathematical Logic and Applications provides a broad, thorough coverage of the fundamentals of two-valued logic, multivalued logic, and fuzzy logic. Exploring the parallels between classical and fuzzy mathematical logic, the book examines the use of logic in computer science, addresses questions in automatic deduction, and describes efficient computer implementation of proof techniques. Specific issues discussed include: Propositional and predicate logic Logic networks Logic programming Proof of correctness Semantics Syntax Completenesss Non-contradiction Theorems of Herbrand and Kalman The authors consider that the teaching of logic for computer science is biased by the absence of motivations, comments, relevant and convincing examples, graphic aids, and the use of color to distinguish language and metalanguage. Classical and Fuzzy Concepts in Mathematical Logic and Applications discusses how the presence of these facts trigger a stirring, decisive insight into the understanding process. This view shapes this work, reflecting the authors' subjective balance between the scientific and pedagogic components of the textbook. Usually, problems in logic lack relevance, creating a gap between classroom learning and applications to real-life problems. The book includes a variety of application-oriented problems at the end of almost every section, including programming problems in PROLOG III. With the possibility of carrying out proofs with PROLOG III and other software packages, readers will gain a first-hand experience and thus a deeper understanding of the idea of formal proof.

This text provides a practical, modern approach to teaching logic and set theory, equipping students with the necessary mathematical understanding and skills required for the mathematical specification of software. It covers all the areas of mathematics that are considered essential to computer science including logic, set theory, modern algebra (group theory), graph theory and combinatorics, whilst taking into account the diverse mathematical background of the students taking the course. In line with current undergraduate curricula this book uses logic extensively, together with set theory, in mathematical specification of software. Languages such as Z and VDM are used for this purpose. Features Particular emphasis is placed on the application of logic in the fields of software engineering, artificial intelligence and natural language processing 0201179571B04062001

Recent years have seen the development of powerful tools for verifying hardware and software systems, as companies worldwide realise the need for improved means of validating their products. There is increasing demand for training in basic methods in formal reasoning so that students can gain proficiency in logic-based verification methods. The second edition of this successful textbook addresses both those requirements, by continuing to provide a clear introduction to formal reasoning which is both relevant to the needs of modern computer science and rigorous enough for practical application. Improvements to the first edition have been made throughout, with extra and expanded sections on SAT solvers, existential/universal second-order logic, micro-models, programming by contract and total correctness. The coverage of model-checking has been substantially updated. Further exercises have been added. Internet support for the book includes worked solutions for all exercises for teachers, and model solutions to some exercises for students.

This book introduces the notions and methods of formal logic from a computer science standpoint, covering propositional logic, predicate logic, and foundations of logic programming. The classic text is replete with illustrative examples and exercises. It presents applications and themes of computer science research such as resolution, automated deduction, and logic programming in a rigorous but readable way. The style and scope of the work, rounded out by the inclusion of exercises, make this an excellent textbook for an advanced undergraduate course in logic for computer scientists.

An understanding of logic is essential to computer science. This book provides a highly accessible account of the logical basis required for reasoning about computer programs and applying logic in fields like artificial intelligence. The text contains extended examples, algorithms, and programs written in Standard ML and Prolog. No prior knowledge of either language is required. The book contains a clear account of classical first-order logic, one of the basic tools for program verification, as well as an introductory survey of modal and temporal logics and possible world semantics. An introduction to intuitionistic logic as a basis for an important style of program specification is also featured in the book.

This book covers elementary discrete mathematics for computer science and engineering. It emphasizes mathematical definitions and proofs as well as applicable methods. Topics include formal logic notation, proof methods; induction, well-ordering; sets, relations; elementary graph theory; integer congruences; asymptotic notation and growth of functions; permutations and combinations, counting principles; discrete probability. Further selected topics may also be covered, such as recursive definition and structural induction; state machines and invariants; recurrences; generating functions.

This easy-to-follow textbook introduces the mathematical language, knowledge and problem-solving skills that undergraduates need to study computing. The language is in part qualitative, with concepts such as set, relation, function and recursion/induction; but it is also partly quantitative, with principles of counting and finite probability. Entwined with both are the fundamental notions of logic and their use for representation and proof. Features: teaches finite math as a language for thinking, as much as knowledge and skills to be acquired; uses an intuitive approach with a focus on examples for all general concepts; brings out the interplay between the qualitative and the quantitative in all areas covered, particularly in the treatment of recursion and induction; balances carefully the abstract and concrete, principles and proofs, specific facts and general perspectives; includes highlight boxes that raise common queries and clear confusions; provides numerous exercises, with selected solutions.

Mathematical Logic is a collection of the works of one of the leading figures in 20th-century science. This collection of A.M. Turing's works is intended to include all his mature scientific writing, including a substantial quantity of unpublished material. His work in pure mathematics and mathematical logic extended considerably further; the work of his last years, on morphogenesis in plants, is also of the greatest originality and of permanent importance. This book is divided into three parts. The first part focuses on computability and ordinal logics and covers Turing's work between 1937 and 1938. The second part covers type theory; it provides a general introduction to Turing's work on type theory and covers his published and unpublished works between 1941 and 1948. Finally, the third part focuses on enigmas, mysteries, and loose ends. This concluding section of the book discusses Turing's Treatise on the Enigma, with excerpts from the Enigma Paper. It also delves into Turing's papers on programming and on minimum cost sequential analysis, featuring an excerpt from the unpublished manuscript. This book will be of interest to mathematicians, logicians, and computer scientists.

Mathematical logic is a branch of mathematics that takes axiom systems and mathematical proofs as its objects of study. This book shows how it can also provide a foundation for the development of information science and technology. The first five chapters systematically present the core topics of classical mathematical logic, including the syntax and models of first-order languages, formal inference systems, computability and representability, and Gödel’s theorems. The last five chapters present extensions and developments of classical mathematical logic, particularly the concepts of version sequences of formal theories and their limits, the system of revision calculus, proschemes (formal descriptions of proof methods and strategies) and their properties, and the theory of inductive inference. All of these themes contribute to a formal theory of axiomatization and its application to the process of developing information technology and scientific theories. The book also describes the paradigm of three kinds of language environments for theories and it presents the basic properties required of a meta-language environment. Finally, the book brings these themes together by describing a workflow for scientific research in the information era in which formal methods, interactive software and human invention are all used to their advantage. This book represents a valuable reference for graduate and undergraduate students and researchers in mathematics, information science and technology, and other relevant areas of natural sciences. Its first five chapters serve as an undergraduate text in mathematical logic and the last five chapters are addressed to graduate students in relevant disciplines.

Providing an in-depth introduction to fundamental classical and non-classical logics, this textbook offers a comprehensive survey of logics for computer scientists. Logics for Computer Science contains intuitive introductory chapters explaining the need for logical investigations, motivations for different types of logics and some of their history. They are followed by strict formal approach chapters. All chapters contain many detailed examples explaining each of the introduced notions and definitions, well chosen sets of exercises with carefully written solutions, and sets of homework. While many logic books are available, they were written by logicians for logicians, not for computer scientists. They usually choose one particular way of presenting the material and use a specialized language. Logics for Computer Science discusses Gentzen as well as Hilbert formalizations, first order theories, the Hilbert Program, Godel's first and second incompleteness theorems and their proofs. It also introduces and discusses some many valued logics, modal logics and introduces algebraic models for classical, intuitionistic, and modal S4 and S5 logics. The theory of computation is based on concepts defined by logicians and mathematicians. Logic plays a fundamental role in computer science, and this book explains the basic theorems, as well as different techniques of proving them in classical and some non-classical logics. Important applications derived from concepts of logic for computer technology include Artificial Intelligence and Software Engineering. In addition to Computer Science, this book may also find an audience in mathematics and philosophy courses, and some of the chapters are also useful for a course in Artificial Intelligence.

This comprehensive overview ofmathematical logic is designedprimarily for advanced undergraduatesand graduate studentsof mathematics. The treatmentalso contains much of interest toadvanced students in computerscience and philosophy. Topics include propositional logic;first-order languages and logic; incompleteness, undecidability,and indefinability; recursive functions; computability;and Hilbert’s Tenth Problem.Reprint of the PWS Publishing Company, Boston, 1995edition.

This text for the first or second year undergraduate in mathematics, logic, computer science, or social sciences, introduces the reader to logic, proofs, sets, and number theory. It also serves as an excellent independent study reference and resource for instructors. Adapted from Foundations of Logic and Mathematics: Applications to Science and Cryptography © 2002 Birkh?user, this second edition provides a modern introduction to the foundations of logic, mathematics, and computers science, developing the theory that demonstrates construction of all mathematics and theoretical computer science from logic and set theory. The focuses is on foundations, with specific statements of all the associated axioms and rules of logic and set theory, and provides complete details and derivations of formal proofs. Copious references to literature that document historical development is also provided. Answers are found to many questions that usually remain unanswered: Why is the truth table for logical implication so unintuitive? Why are there no recipes to design proofs? Where do these numerous mathematical rules come from? What issues in logic, mathematics, and computer science still remain unresolved? And the perennial question: In what ways are we going to use this material? Additionally, the selection of topics presented reflects many major accomplishments from the twentieth century and includes applications in game theory and Nash's equilibrium, Gale and Shapley's match making algorithms, Arrow's Impossibility Theorem in voting, to name a few. From the reviews of the first edition: "...All the results are proved in full detail from first principles...remarkably, the arithmetic laws on the rational numbers are proved, step after step, starting from the very definitions!...This is a valuable reference text and a useful companion for anybody wondering how basic mathematical concepts can be rigorously developed within set theory." —MATHEMATICAL REVIEWS "Rigorous and modern in its theoretical aspect, attractive as a detective novel in its applied aspects, this paper book deserves the attention of both beginners and advanced students in mathematics, logic and computer sciences as well as in social sciences." —Zentralblatt MATH

A comprehensive and user-friendly guide to the use of logic inmathematical reasoning Mathematical Logic presents a comprehensive introductionto formal methods of logic and their use as a reliable tool fordeductive reasoning. With its user-friendly approach, this booksuccessfully equips readers with the key concepts and methods forformulating valid mathematical arguments that can be used touncover truths across diverse areas of study such as mathematics,computer science, and philosophy. The book develops the logical tools for writing proofs byguiding readers through both the established "Hilbert" style ofproof writing, as well as the "equational" style that is emergingin computer science and engineering applications. Chapters havebeen organized into the two topical areas of Boolean logic andpredicate logic. Techniques situated outside formal logic areapplied to illustrate and demonstrate significant facts regardingthe power and limitations of logic, such as: Logic can certify truths and only truths. Logic can certify all absolute truths (completeness theorems ofPost and Gödel). Logic cannot certify all "conditional" truths, such as thosethat are specific to the Peano arithmetic. Therefore, logic hassome serious limitations, as shown through Gödel'sincompleteness theorem. Numerous examples and problem sets are provided throughout thetext, further facilitating readers' understanding of thecapabilities of logic to discover mathematical truths. In addition,an extensive appendix introduces Tarski semantics and proceeds withdetailed proofs of completeness and first incompleteness theorems,while also providing a self-contained introduction to the theory ofcomputability. With its thorough scope of coverage and accessible style,Mathematical Logic is an ideal book for courses inmathematics, computer science, and philosophy at theupper-undergraduate and graduate levels. It is also a valuablereference for researchers and practitioners who wish to learn howto use logic in their everyday work.

Mathematical logic is essentially related to computer science. This book describes the aspects of mathematical logic that are closely related to each other, including classical logic, constructive logic, and modal logic. This book is intended to attend to both the peculiarities of logical systems and the requirements of computer science. In this edition, the revisions essentially involve rewriting the proofs, increasing the explanations, and adopting new terms and notations. Contents:Prerequisites:SetsInductive Definitions and ProofsNotationsClassical Propositional Logic:Propositions and ConnectivesPropositional LanguageStructure of FormulasSemanticsTautological ConsequenceFormal DeductionDisjunctive and Conjunctive Normal FormsAdequate Sets of ConnectivesClassical First-Order Logic:Proposition Functions and QuantifiersFirst-Order LanguageSemanticsLogical ConsequenceFormal DeductionPrenex Normal FormAxiomatic Deduction System:Axiomatic Deduction SystemRelation between the Two Deduction SystemsSoundness and Completeness:Satisfiability and ValiditySoundnessCompleteness of Propositional LogicCompleteness of First-Order LogicCompleteness of First-Order Logic with EqualityIndependenceCompactness, Löwenheim–Skolem, and Herbrand Theorems:CompactnessLöwenheim-Skolem's TheoremHerbrand's TheoremConstructive Logic:Constructivity of ProofsSemanticsFormal DeductionSoundnessCompletenessModal Propositional Logic:Modal Propositional LanguageSemanticsFormal DeductionSoundnessCompleteness of TCompleteness of S4, B, S5Modal First-Order Logic:Modal First-Order LanguageSemanticsFormal DeductionSoundnessCompletenessEquality Readership: Computer scientists. keywords:
This is a mathematics textbook with theorems and proofs. The choice of topics has been guided by the needs of computer science students. The method of semantic tableaux provides an elegant way to teach logic that is both theoretically sound and yet sufficiently elementary for undergraduates. In order to provide a balanced treatment of logic, tableaux are related to deductive proof systems. The book presents various logical systems and contains exercises. Still further, Prolog source code is available on an accompanying Web site. The author is an Associate Professor at the Department of Science Teaching, Weizmann Institute of Science.

Mathematical Logic and Theoretical Computer Science covers various topics ranging from recursion theory to Zariski topoi. Leading international authorities discuss selected topics in a number of areas, including denotational semanitcs, reccuriosn theoretic aspects fo computer science, model theory and algebra, Automath and automated reasoning, stability theory, topoi and mathematics, and topoi and logic. The most up-to-date review available in its field, Mathematical Logic and Theoretical Computer Science will be of interest to mathematical logicians, computer scientists, algebraists, algebraic geometers, differential geometers, differential topologists, and graduate students in mathematics and computer science.

This advanced text for undergraduate and graduate students introduces mathematical logic with an emphasis on proof theory and procedures for algorithmic construction of formal proofs. The self-contained treatment is also useful for computer scientists and mathematically inclined readers interested in the formalization of proofs and basics of automatic theorem proving. Topics include propositional logic and its resolution, first-order logic, Gentzen's cut elimination theorem and applications, and Gentzen's sharpened Hauptsatz and Herbrand's theorem. Additional subjects include resolution in first-order logic; SLD-resolution, logic programming, and the foundations of PROLOG; and many-sorted first-order logic. Numerous problems appear throughout the book, and two Appendixes provide practical background information.

A proof is a successful demonstration that a conclusion necessarily follows by logical reasoning from axioms which are considered evident for the given context and agreed upon by the community. It is this concept that sets mathematics apart from other disciplines and distinguishes it as the prototype of a deductive science. Proofs thus are utterly relevant for research, teaching and communication in mathematics and of particular interest for the philosophy of mathematics. In computer science, moreover, proofs have proved to be a rich source for already certified algorithms. This book provides the reader with a collection of articles covering relevant current research topics circled around the concept 'proof'. It tries to give due consideration to the depth and breadth of the subject by discussing its philosophical and methodological aspects, addressing foundational issues induced by Hilbert's Programme and the benefits of the arising formal notions of proof, without neglecting reasoning in natural language proofs and applications in computer science such as program extraction.

This is a compact mtroduction to some of the pnncipal tOpICS of mathematical logic . In the belief that beginners should be exposed to the most natural and easiest proofs, I have used free-swinging set-theoretic methods. The significance of a demand for constructive proofs can be evaluated only after a certain amount of experience with mathematical logic has been obtained. If we are to be expelled from "Cantor's paradise" (as nonconstructive set theory was called by Hilbert), at least we should know what we are missing. The major changes in this new edition are the following. (1) In Chapter 5, Effective Computability, Turing-computabIlity IS now the central notion, and diagrams (flow-charts) are used to construct Turing machines. There are also treatments of Markov algorithms, Herbrand-Godel-computability, register machines, and random access machines. Recursion theory is gone into a little more deeply, including the s-m-n theorem, the recursion theorem, and Rice's Theorem. (2) The proofs of the Incompleteness Theorems are now based upon the Diagonalization Lemma. Lob's Theorem and its connection with Godel's Second Theorem are also studied. (3) In Chapter 2, Quantification Theory, Henkin's proof of the completeness theorem has been postponed until the reader has gained more experience in proof techniques. The exposition of the proof itself has been improved by breaking it down into smaller pieces and using the notion of a scapegoat theory. There is also an entirely new section on semantic trees.

Mathematical logic developed into a broad discipline with many applications in mathematics, informatics, linguistics and philosophy. This text introduces the fundamentals of this field, and this new edition has been thoroughly expanded and revised.

Mathematical logic is essentially related to computer science. This book describes the aspects of mathematical logic that are closely related to each other, including classical logic, constructive logic, and modal logic. This book is intended to attend to both the peculiarities of logical systems and the requirements of computer science.In this edition, the revisions essentially involve rewriting the proofs, increasing the explanations, and adopting new terms and notations.

Mathematics plays a key role in computer science, some researchers would consider computers as nothing but the physical embodiment of mathematical systems. And whether you are designing a digital circuit, a computer program or a new programming language, you need mathematics to be able to reason about the design -- its correctness, robustness and dependability. This book covers the foundational mathematics necessary for courses in computer science. The common approach to presenting mathematical concepts and operators is to define them in terms of properties they satisfy, and then based on these definitions develop ways of computing the result of applying the operators and prove them correct. This book is mainly written for computer science students, so here the author takes a different approach: he starts by defining ways of calculating the results of applying the operators and then proves that they satisfy various properties. After justifying his underlying approach the author offers detailed chapters covering propositional logic, predicate calculus, sets, relations, discrete structures, structured types, numbers, and reasoning about programs. The book contains chapter and section summaries, detailed proofs and many end-of-section exercises -- key to the learning process. The book is suitable for undergraduate and graduate students, and although the treatment focuses on areas with frequent applications in computer science, the book is also suitable for students of mathematics and engineering.

Covers propositional calculus, predicate calculus, resolution and logic programming, temporal logic, and formalization of programs

Using a unique pedagogical approach, this text introduces mathematical logic by guiding students in implementing the underlying logical concepts and mathematical proofs via Python programming. This approach, tailored to the unique intuitions and strengths of the ever-growing population of programming-savvy students, brings mathematical logic into the comfort zone of these students and provides clarity that can only be achieved by a deep hands-on understanding and the satisfaction of having created working code. While the approach is unique, the text follows the same set of topics typically covered in a one-semester undergraduate course, including propositional logic and first-order predicate logic, culminating in a proof of Gödel's completeness theorem. A sneak peek to Gödel's incompleteness theorem is also provided. The textbook is accompanied by an extensive collection of programming tasks, code skeletons, and unit tests. Familiarity with proofs and basic proficiency in Python is assumed.

This text is intended for one semester courses in Logic, it can also be applied to a two semester course, in either Computer Science or Mathematics Departments. Unlike other texts on mathematical logic that are either too advanced, too sparse in examples or exercises, too traditional in coverage, or too philosophical in approach, this text provides an elementary "hands-on" presentation of important mathematical logic topics, new and old, that is readily accessible and relevant to all students of the mathematical sciences -- not just those in traditional pure mathematics.

"This book offers a high interdisciplinary exchange of ideas pertaining to the philosophy of computer science, from philosophical and mathematical logic to epistemology, engineering, ethics or neuroscience experts and outlines new problems that arise with new tools"--Provided by publisher.

This book provides an introduction to logic and mathematical induction which are the basis of any deductive computational framework. A strong mathematical foundation of the logical engines available in modern proof assistants, such as the PVS verification system, is essential for computer scientists, mathematicians and engineers to increment their capabilities to provide formal proofs of theorems and to certify the robustness of software and hardware systems. The authors present a concise overview of the necessary computational and mathematical aspects of ‘logic’, placing emphasis on both natural deduction and sequent calculus. Differences between constructive and classical logic are highlighted through several examples and exercises. Without neglecting classical aspects of computational logic, the authors also highlight the connections between logical deduction rules and proof commands in proof assistants, presenting simple examples of formalizations of the correctness of algebraic functions and algorithms in PVS. Applied Logic for Computer Scientists will not only benefit students of computer science and mathematics but also software, hardware, automation, electrical and mechatronic engineers who are interested in the application of formal methods and the related computational tools to provide mathematical certificates of the quality and accuracy of their products and technologies.

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. Within 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.