Proofs In Computer Science

Mathematics of Discrete Structures for Computer Science
Concepts of Proof in Mathematics, Philosophy, and Computer Science
Types for Proofs and Programs
Proof And Computation: Digitization In Mathematics, Computer Science And Philosophy
Mechanizing Proof
Computability and Complexity
Discrete Mathematics
A Logical Approach to Discrete Math
Proofs and Algorithms
Isabelle/HOL
Proofs and Computations
Discrete Mathematics with Proof
People, Problems, and Proofs
Essential Discrete Mathematics for Computer Science
Fundamental Proof Methods in Computer Science
Basic Proof Theory
Essential Logic for Computer Science
Logic for Computer Scientists
A Proof Theory for Description Logics
Proof, Language, and Interaction
Proofs and Algorithms
Java Programming: From The Ground Up
Elements of Programming
Philosophical Perceptions on Logic and Order
First-Order Logic and Automated Theorem Proving
Discrete Mathematics for Computer Science
Proof Technology in Mathematics Research and Teaching
How to Prove It
Types for Proofs and Programs
Logic for Computer Science
Handbook of Logic and Proof Techniques for Computer Science
Applied Logic for Computer Scientists
Mathematical Logic for Computer Science
Concepts of Proof in Mathematics, Philosophy, and Computer Science
Certified Programming with Dependent Types
Proofs and Types
Adapting Proofs-as-Programs
Mathematics for Computer Science
Verified Software: Theories, Tools, Experiments

HAILEY JAMARCUS

Mathematics of Discrete Structures for Computer Science Princeton University Press
Computability and complexity theory should be of central concern to practitioners as well as theorists. Unfortunately, however, the field is known for its impenetrability. Neil Jones's goal as an educator and author is to build a bridge between computability and complexity theory and other areas of computer science, especially programming. In a shift away from the Turing machine- and G�del number-oriented classical approaches, Jones uses concepts familiar from programming languages to make computability and complexity more accessible to computer scientists and more applicable to practical programming problems. According to Jones, the fields of computability and complexity theory, as well as programming languages and semantics, have a great deal to offer each other. Computability and complexity theory have a breadth, depth, and generality not often seen in programming languages. The programming language community, meanwhile, has a firm grasp of algorithm design, presentation, and implementation. In addition, programming languages sometimes provide computational models that are more realistic in certain crucial aspects than traditional models. New results in the book include a proof that constant time factors do matter for its programming-oriented model of computation. (In contrast, Turing machines have a counterintuitive "constant speedup" property: that almost any program can be made to run faster, by any amount. Its proof involves techniques irrelevant to practice.) Further results include simple characterizations in programming terms of the central complexity classes PTIME and LOGSPACE, and a new approach to complete problems for NLOGSPACE, PTIME, NPTIME, and PSPACE, uniformly based on Boolean programs. Foundations of Computing series
Concepts of Proof in Mathematics, Philosophy, and Computer Science Springer Science & Business Media
Logic is a branch of philosophy, mathematics and computer science. It studies the required methods to determine whether a statement is true, such as reasoning and computation. Proofs and Algorithms: Introduction to Logic and Computability is an introduction to the fundamental concepts of contemporary logic - those of a proof, a computable function, a model and a set. It presents a series of results, both positive and negative, - Church's undecidability theorem, Gödel’s incompleteness theorem, the theorem asserting the semi-decidability of provability - that have profoundly changed our vision of reasoning, computation, and finally truth itself. Designed for undergraduate students, this book presents all that philosophers, mathematicians and computer scientists should know about logic.
Types for Proofs and Programs Springer
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.
Proof And Computation: Digitization In Mathematics, Computer Science And Philosophy Springer
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.
Mechanizing Proof MIT Press
Fundamental Proof Methods in Computer ScienceMIT Press
Computability and Complexity Springer
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.
Discrete Mathematics MIT Press
A handbook to the Coq software for writing and checking mathematical proofs, with a practical engineering focus. The technology of mechanized program verification can play a supporting role in many kinds of research projects in computer science, and related tools for formal proof-checking are seeing increasing adoption in mathematics and engineering. This book provides an introduction to the Coq software for writing and checking mathematical proofs. It takes a practical engineering focus throughout, emphasizing techniques that will help users to build, understand, and maintain large Coq developments and minimize the cost of code change over time. Two topics, rarely discussed elsewhere, are covered in detail: effective dependently typed programming (making productive use of a feature at the heart of the Coq system) and construction of domain-specific proof tactics. Almost every subject covered is also relevant to interactive computer theorem proving in general, not just program verification, demonstrated through examples of verified programs applied in many different sorts of formalizations. The book develops a unique automated proof style and applies it throughout; even experienced Coq users may benefit from reading about basic Coq concepts from this novel perspective. The book also offers a library of tactics, or programs that find proofs, designed for use with examples in the book. Readers will acquire the necessary skills to reimplement these tactics in other settings by the end of the book. All of the code appearing in the book is freely available online.
A Logical Approach to Discrete Math MIT Press
Elements of Programming provides a different understanding of programming than is presented elsewhere. Its major premise is that practical programming, like other areas of science and engineering, must be based on a solid mathematical foundation. The book shows that algorithms implemented in a real programming language, such as C++, can operate in the most general mathematical setting. For example, the fast

exponentiation algorithm is defined to work with any associative operation. Using abstract algorithms leads to efficient, reliable, secure, and economical software.
Proofs and Algorithms MIT Press
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.
Isabelle/HOL Cambridge University Press
Logic is a branch of philosophy, mathematics and computer science. It studies the required methods to determine whether a statement is true, such as reasoning and computation. Proofs and Algorithms: Introduction to Logic and Computability is an introduction to the fundamental concepts of contemporary logic - those of a proof, a computable function, a model and a set. It presents a series of results, both positive and negative, - Church's undecidability theorem, Gödel’s incompleteness theorem, the theorem asserting the semi-decidability of provability - that have profoundly changed our vision of reasoning, computation, and finally truth itself. Designed for undergraduate students, this book presents all that philosophers, mathematicians and computer scientists should know about logic.
Proofs and Computations Brooks/Cole Publishing Company
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.
Discrete Mathematics with Proof IGI Global
A Trusted Guide to Discrete Mathematics with Proof?Now in a Newly Revised Edition Discrete mathematics has become increasingly popular in recent years due to its growing applications in the field of computer science. Discrete Mathematics with Proof, Second Edition continues to facilitate an up-to-date understanding of this important topic, exposing readers to a wide range of modern and technological applications. The book begins with an introductory chapter that provides an accessible explanation of discrete mathematics. Subsequent chapters explore additional related topics including counting, finite probability theory, recursion, formal models in computer science, graph theory, trees, the concepts of functions, and relations. Additional features of the Second Edition include: An intense focus on the formal settings of proofs and their techniques, such as constructive proofs, proof by contradiction, and combinatorial proofs New sections on applications of elementary number theory, multidimensional induction, counting tulips, and the binomial distribution Important examples from the field of computer science presented as applications including the Halting problem, Shannon's mathematical model of information, regular expressions, XML, and Normal Forms in relational databases Numerous examples that are not often found in books on discrete mathematics including the deferred acceptance algorithm, the Boyer-Moore algorithm for pattern matching, Sierpinski curves, adaptive quadrature, the Josephus problem, and the five-color theorem Extensive appendices that outline supplemental material on analyzing claims and writing mathematics, along with solutions to selected chapter exercises Combinatorics receives a full chapter treatment that extends beyond the combinations and permutations material by delving into non-standard topics such as Latin squares, finite projective planes, balanced incomplete block designs, coding theory, partitions, occupancy problems, Stirling numbers, Ramsey numbers, and systems of distinct representatives. A related Web site features animations and visualizations of combinatorial proofs that assist readers with comprehension. In addition, approximately 500 examples and over 2,800 exercises are presented throughout the book to motivate ideas and illustrate the proofs and conclusions of theorems. Assuming only a basic background in calculus, Discrete Mathematics with Proof, Second Edition is an excellent book for mathematics and computer science courses at the undergraduate level. It is also a valuable resource for professionals in various technical fields who would like an introduction to discrete mathematics.
People, Problems, and Proofs Springer Science & Business Media
This introduction to the basic ideas of structural proof theory contains a thorough discussion and comparison of various types of formalization of first-order logic. Examples are given of several areas of application, namely: the metamathematics of pure first-order logic (intuitionistic as well as classical); the theory of logic programming; category theory; modal logic; linear logic; first-order arithmetic and second-order logic. In each case the aim is to illustrate the methods in relatively simple situations and then apply them elsewhere in much more complex settings. There are numerous exercises throughout the text. In general, the only prerequisite is a standard course in first-order logic, making the book ideal for graduate students and beginning researchers in mathematical logic, theoretical computer science and artificial intelligence. For the new edition, many sections have been rewritten to improve clarity, new sections have been added on cut elimination, and solutions to selected exercises have been included.
Essential Discrete Mathematics for Computer Science Springer Science & Business Media
This monograph details several important advances in the direction of a practical proofs-as-programs paradigm, which constitutes a set of approaches to developing programs from proofs in constructive logic with applications to industrial-scale, complex software engineering problems. One of the books central themes is a general, abstract framework for developing new systems of programs synthesis by adapting proofs-as-programs to new contexts.
Fundamental Proof Methods in Computer Science World Scientific
This collection of essays reflects the breadth of research in computer science. Following a biography of Robin Milner it contains sections on semantic foundations; programming logic; programming languages; concurrency; and mobility.
Basic Proof Theory Springer Science & Business Media
Master the fundamentals of discrete mathematics with DISCRETE MATHEMATICS FOR COMPUTER SCIENCE with Student Solutions Manual CD-ROM! An increasing number of computer scientists from diverse areas are using discrete mathematical structures to explain concepts and problems and this mathematics text shows you how to express precise ideas in clear mathematical language. Through a wealth of exercises and examples, you will learn how mastering discrete mathematics will help you develop important reasoning skills that will continue to be useful throughout your career.
Essential Logic for Computer Science Springer Science & Business Media
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.
Logic for Computer Scientists Springer Science & Business Media
Description Logics (DLs) is a family of formalisms used to represent knowledge of a domain. They are equipped with a formal logic-based semantics. Knowledge representation systems based on description logics provide various inference capabilities that deduce implicit knowledge from the explicitly represented knowledge. A Proof Theory for Description Logics introduces Sequent Calculi and Natural Deduction for some DLs (ALC, ALCQ). Cut-elimination and Normalization are proved for the calculi. The author argues that such systems can improve the extraction of computational content from DLs proofs for explanation purposes.
A Proof Theory for Description Logics Cambridge University Press
The refereed post-proceedings of the International Workshop of the Types Working Group are presented in this volume. The 17 papers address all current issues in formal reasoning and computer programming based on type theory, including languages and computerized tools for reasoning; applications in several domains, such as analysis of programming languages; certified software; formalization of mathematics; and mathematics education.
Proof, Language, and Interaction Cambridge University Press
Aimed at undergraduate mathematics and computer science students, this book is an excellent introduction to a lot of problems of discrete mathematics. It discusses a number of selected results and methods, mostly from areas of combinatorics and graph theory, and it uses proofs and problem solving to help students understand the solutions to problems. Numerous examples, figures, and exercises are spread throughout the book.