Brk 2002rac Guide

Project Factorisations in Partial EvaluationLogic in Computer ScienceSoftware AbstractionsBuilding Web Applications with ErlangIntroduction to Programming Using SMLSpreadsheet Implementation TechnologyRealm of RacketModern Compiler Implementation in MLNonlinear Least Squares for Inverse ProblemsConcurrent Programming in MLThe Fun of ProgrammingML for the Working ProgrammerIntracellular Signaling Mediators in the Circulatory and Ventilatory SystemsSteps in ScalaStructure and Interpretation of Classical MechanicsDrawing Programs: The Theory and Practice of Schematic Functional ProgrammingAn Introduction to Computing with HaskellProgramming MethodologyOCaml from the Very BeginningSMIL 2.0Elements of ML ProgrammingThe Rap Canterbury TalesHigher Order Operational Techniques in SemanticsTrends in Functional Programming Volume 4

Project Factorisations in Partial Evaluation

Written by the members of the IFIP Working Group 2.3 (Programming Methodology) this text constitutes an exciting reference on the front-line of research activity in programming methodology. The range of subjects reflects the current interests of the members, and will offer insightful and controversial opinions on modern programming methods and practice. The material is arranged in thematic sections, each one introduced by a problem which epitomizes the spirit of that topic. The exemplary problem will encourage vigorous discussion and will form the basis for an introduction/tutorial for its section.

Logic in Computer Science

"Working with REST and Web-Sockets on Yaws"--Cover.

Software Abstractions

In OCaml from the Very Beginning John Whitington takes a no-prerequisites approach to teaching a modern general-purpose programming language. Each small, self-contained chapter introduces a new topic, building until the reader can write quite substantial programs. There are plenty of questions and, crucially, worked answers and hints. OCaml from the Very Beginning will appeal both to new programmers, and experienced programmers eager to explore functional languages such as OCaml. It is suitable both for formal use within an undergraduate or graduate curriculum, and for the interested amateur.

Building Web Applications with Erlang

An innovative textbook that emphasizes the development of practical intellectualtools to support the analysis of nonlinear Hamiltonian systems.

Introduction to Programming Using SML

Hip-hop artist Brinkman resurrects Chaucer's brilliant stories into visible and audible contemporary forms.

Spreadsheet Implementation Technology

This new, expanded textbook describes all phases of a modern compiler: lexical analysis, parsing, abstract syntax, semantic actions, intermediate representations, instruction selection via tree matching, dataflow analysis, graph-coloring register allocation, and runtime systems. It includes good coverage of current techniques in code generation and register allocation, as well as functional and object-oriented languages, that are missing from most books. In addition, more advanced chapters are now included so that it can be used as the basis for two-semester or graduate course. The most accepted and successful techniques are described in a concise way, rather than as an exhaustive catalog of every possible variant. Detailed descriptions of the interfaces between modules of a compiler are illustrated with actual C header files. The first part of the book, Fundamentals of Compilation, is suitable for a one-semester first course in compiler design. The second part, Advanced Topics, which includes the advanced chapters, covers the compilation of object-oriented and functional languages, garbage collection, loop optimizations, SSA form, loop scheduling, and optimization for cache-memory hierarchies.

Realm of Racket

This highly accessible introduction to the fundamentals of ML is presented by computer science educator and author, Jeffrey D. Ullman. The primary change in the Second Edition is that it has been thoroughly revised and reorganized to conform to the new language standard called ML97. This is the first book that offers both an accurate step-by-step tutorial to ML programming and a comprehensive reference to advanced features. It is the only book that focuses on the popular SML/NJ implementation. The material is arranged for use in sophomore through graduate level classes or for self-study. This text assumes no previous knowledge of ML or functional programming, and can be used to teach ML as a first programming language. It is also an excellent supplement or reference for programming language concepts, functional programming, or compiler courses.

Modern Compiler Implementation in ML

In this textbook, leading researchers give tutorial expositions on the current state of the art of functional programming. The text is suitable for an undergraduate course immediately following an introduction to functional programming, and also for self-study. All new concepts are illustrated by plentiful examples, as well as exercises. A website gives access to accompanying software.

Nonlinear Least Squares for Inverse Problems

This book collects the latest research developments in the use of functional programming languages. The contents highlight major research goals and engineering concerns in the subject

Concurrent Programming in ML

This new edition of a successful text treats modules in more depth, and covers the revision of ML language.

The Fun of Programming

A guide to innovative spreadsheet implementation technology, accompanied by a free software platform for experimentation.

ML for the Working Programmer

SMIL 2.0 - Interactive multimedia for Web and Mobile Devices gently introduces you to the Web multimedia standard SMIL 2.0. Written by world-renowned SMIL experts who helped to develop the language and software for it, this book covers all aspects of the standard in a knowledgeable yet accessible manner: the overall concepts, the technical details and the many facets of SMIL's current and expected use. It is written to serve as an introduction, a full manual and a detailed technical reference.

Intracellular Signaling Mediators in the Circulatory and Ventilatory Systems

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.

Steps in Scala

Drawing Programs: The Theory and Practice of Schematic Functional Programming describes a diagrammatic (schematic) approach to programming. It introduces a sophisticated tool for programmers who would rather work with diagrams than with text. The language is a complete functional language that has evolved into a representation scheme that is unique. The result is a simple coherent description of the process of modelling with the computer. The experience of using this tool is introduced gradually with examples, small projects and exercises. The new computational theory behind the tool is interspersed between these practical descriptions so that the reasons for the activity can be understood and the activity, in turn, illustrates some elements of the theory Access to the tool, its source code and a set of examples that range from the simple to the complex is free (see www.springer.com/978-1-84882-617-5). A description of the tool's construction and how it may be extended is also given. The authors' experience with undergraduates and graduates who have the understanding and skill of a functional language learnt through using schema have also shown an enhanced ability to program in other computer languages. Readers are provided with a set of concepts that will ensure a good robust program design and, what is more important, a path to error free programming.

Structure and Interpretation of Classical Mechanics

This thesis covers the theory and practice behind practical evaluation. It explores how programmers write programs in a highly interpretive style without paying the price in efficiency.

Drawing Programs: The Theory and Practice of Schematic Functional Programming

This title offers an introduction to computing for the absolute beginner to Haskell language.

An Introduction to Computing with Haskell

A 'how-to' book for programmers and researchers interested in practical applications of Concurrent ML.

Programming Methodology

The volumes in this authoritative series present a multidisciplinary approach to modeling and simulation of flows in the cardiovascular and ventilatory systems, especially multiscale modeling and coupled simulations. The cardiovascular and respiratory systems are tightly coupled, as their primary function is to supply oxygen to and remove carbon dioxide from the body's cells. Because physiological conduits have deformable and reactive walls, macroscopic flow behavior and prediction must be coupled to phenomenological models of nano- and microscopic events in a corrector scheme of regulated mechanisms when the vessel lumen caliber varies markedly. Therefore, investigation of flows of blood and air in physiological conduits requires an understanding of the biology, chemistry, and physics of these systems together with the mathematical tools to describe their functioning. Volume 4 is devoted to major sets of intracellular mediators that transmit signals upon stimulation of cell-surface receptors. Activation of signaling effectors triggers the release of substances stored in cellular organelles and/or gene transcription and protein synthesis. Complex stages of cell signaling can be studied using proper mathematical models, once the role of each component is carefully handled. Volume 4 also reviews various categories of cytosolic and/or nuclear mediators and illustrates some major signal transduction pathways, such as NFkappaB axis, oxygen sensing, and mechanotransduction.

OCaml from the Very Beginning

Racket is a descendant of Lisp, a programming language renowned for its elegance, power, and challenging learning curve. But while Racket retains the functional goodness of Lisp, it was designed with beginning programmers in mind. Realm of Racket is your introduction to the Racket language. In Realm of Racket, you'll learn to program by creating increasingly complex games. Your journey begins with the Guess My Number game and coverage of some basic Racket etiquette. Next you'll dig into syntax and semantics, lists, structures, and conditionals, and learn to work with recursion and the GUI as you build the Robot Snake game. After that it's on to lambda and mutant structs (and an Orc Battle), and fancy loops and the Dice of Doom. Finally, you'll explore laziness, AI, distributed games, and the Hungry Henry game. As you progress through the games, chapter checkpoints and challenges help reinforce what you've learned. Offbeat comics keep things fun along the way. As you travel through the Racket realm, you'll: –Master the quirks of Racket's syntax and semantics –Learn to write concise and elegant functional programs –Create a graphical user interface using the 2htdp/image library –Create a server to handle true multiplayer games Realm of Racket is a lighthearted guide to some serious programming. Read it to see why Racketeers have so much fun!

SMIL 2.0

A 1998 collection of original articles by leading researchers in area of programming languages.

Elements of ML Programming

Scala is a highly expressive, concise and scalable language. It is also the most prominent method of the new and exciting methodology known as object-functional programming. In this book, the authors show how Scala grows to the needs of the programmer, whether professional or hobbyist. They teach Scala with a step-by-step approach and explain how to exploit the full power of the industry-proven JVM technology. Readers can then dive into specially chosen design challenges and implementation problems, inspired by the trials of real-world software engineering. It also helps readers to embrace the power of static typing and automatic type inference. In addition, the book shows how to use the dual-object and functional-oriented natures combined at Scala's core, and so write code that is less 'boilerplate', giving a genuine increase in productivity.

The Rap Canterbury Tales

Previously published in hardcover: 2012.

Higher Order Operational Techniques in Semantics

The domain of inverse problems has experienced a rapid expansion, driven by the increase in computing power and the progress in numerical modeling. When I started working on this domain years ago, I became somehow frtratedtoseethatmyfriendsworkingonmodelingwhereproducingexistence, uniqueness, and stability results for the solution of their equations, but that I was most of the time limited, because of the nonlinearity of the problem, to provethatmyleastsquaresobjectivefunctionwasdi?erentiable.Butwith my experience growing, I became convinced that, after the inverse problem has been properly trimmed, the ?nal least squares problem, the one solved on the computer, should be Quadratically (Q)-wellposed,thatis,both we- posed and optimizable: optimizability ensures that a global minimizer of the least squares function can actually be found using e?cient local optimization algorithms, and wellposedness that this minimizer is stable with respect to perturbation of the data. But the vast majority of inverse problems are nonlinear, and the clas- cal mathematical tools available for their analysis fail to bring answers to these crucial questions: for example, compactness will ensure existence, but provides no uniqueness results, and brings no information on the presence or absenceofparasiticlocalminimaorstationarypoints.

Trends in Functional Programming Volume 4

An introductory programming textbook for students using SML. The text teaches SML program design based on a set of simple, clean and powerful concepts. It emphasizes mathematical structures, modelling and abstraction as a basis for programming.

ROMANCE ACTION & ADVENTURE MYSTERY & THRILLER BIOGRAPHIES & HISTORY CHILDREN'S YOUNG ADULT FANTASY HISTORICAL FICTION HORROR LITERARY FICTION NON-FICTION SCIENCE FICTION