The Objective Caml System Release 3 | 5efb5aae57574b33b9c272053fadb2fb4


Computer Performance Evaluation This book presents the principles and techniques of program specialization — a general method to make programs faster (and possibly smaller) when some inputs can be known in advance. As an illustration, it describes the architecture of Tempo, an offline program optimizer for C that can also specialize code at runtime, and provides figures for concrete applications in various domains. Technical details address issues related to program analysis, precision verification, incomplete program specialization, strategies to exploit specialized program, incremental specialization, and data specialization. The book, that targets both researchers and software engineers, also opens scientific and industrial perspectives.

Programming Languages and Systems This book constitutes the refereed proceedings of the 7th Asian Symposium on Programming Languages and Systems, APLAS 2009, held in Seoul, Korea, in December 2009. The 21 papers presented in this volume together with 3 invited talks were carefully reviewed and selected from 56 submissions. The papers are divided into topical sections on program analysis, transformation and optimization, type system, separation logic, logic and foundation theory, software security and verification, and software security and verification.

Symbolic Computation and Automated Reasoning This practical book gives a comprehensive introduction to the concepts and languages of the new standard IEC 61131 used to program industrial control systems. A summary of the special requirements in programming industrial automation systems and the corresponding features in the IEC 61131-3 standard makes it suitable for students as well as PLC experts. The material is presented in an easy-to-understand form using numerous examples, illustrations and summary tables. There is also a helpful guide and a CD-ROM containing two reduced but functional versions of programming systems. These increase the value of the book for PLC programmers and for those in charge of purchasing software in industrial companies.

Types for Proofs and Programs This pleasure to present the proceedings of the 22nd European Conference on Object-Oriented Programming (ECOOP 2008) held in Paphos, Cyprus. The conference continues to serve a broad object-oriented community with a tech-cal program spanning theory and practice and a healthy mix of industrial and academic participants. This year a strong workshop and tutorial program complemented the main technical track. We had 3 workshops and 2 tutorials as well as the co-located Dynamic Language Symposium (DLS). Finally, the program was rounded out with a keynote by Rachid Guerraoui and a banquet speech by James Noble. As in previous years, two Dahl-Nygaard awards were selected by AITO, and for the first time, the ECOOP Program Committee gave a best paper award. The proceedings include 27 papers selected from 138 submissions. The papers were reviewed in a single-blind process with three to five reviews per paper. P-finalversionsofthereviewswereavailabletomembersoftheau thors' week before the PC meeting to allow for short (500 words or less) author responses. The sponors were discussed at the PC meeting and were instrumental in reaching decisions. The PC discussions followed Oscar Nierstrasz Champion pattern. PC papers had 3 reviews and were held at a higher standard.

ECOOP 2008 - Object-Oriented Programming This book constitutes the refereed proceedings of the 16th European Symposium on Programming, ESOP 2007, held in Braga, Portugal in March/April 2007. It covers models and languages for Web services, verification, term rewriting, language based security, logics and correctness proofs, static analysis and abstract interpretation, semantic theories for object oriented languages, programming paradigms, algebraic programming, system programming, and for those systems properties.

Tools and Algorithms for the Construction of Analysis of Systems Over time, basic research tends to lead to specialization – increasingly narrow t- ics are addressed by increasingly focussed communities, publishing in increasingly condensed workshops and conferences, discussing increasingly narrow aspects of work – already the community of programming languages is split into various s- communities addressing different aspects and paradigms (functional, imperative, relational, and object- oriented). Only a few people manage to maintain a broader view, and even fewer step back in order to gain an understanding about the basic principles, their interrelation, and their impact in a larger context. The pattern calculus is the result of a profound re-examination of a 50-year development. It attempts to provide a unifying approach, bridging the gaps between different programming styles and paradigms according to a new slogan – computation is pattern matching. It is the contribution of this book to systematically and elegantly present and evaluate the power of pattern matching as the guiding paradigm of programming. Patterns are dynamically generated, discovered, passed, applied, and automatically adapted, based on the environment and re-writing techniques, which allows one to elegantly relate and manipulate things as disparate as functions and data structures. Of course, pattern matching is not new. It underlies term rewriting – it is, for example, incorporated in, typically functional, programming languages, like Standard ML – but it has never been pursued as the basis of a unifying framework for programming.

Types for Proofs and Programs This book presents the proceedings of the 15th IFIP WG 6.1 International Conference on Formal Methods for Open-Based Distributed Systems, FMODDS 2013, and the 33rd IFIP WG 6.1 International Conference on Formal Techniques for Networked and Distributed Systems, FORTE 2013, held in Florence, Italy, in June 2013, as part of the 8th International Federated Conference on Distributed Computing Techniques, DsCToC 2013. The 20 revised full papers presented were carefully reviewed and selected from 39 submissions. The papers present a wide range of topics related to the theory and practice of open systems. They cover distributed computing models and formal specification, testing, and verification, as well as applications such as distributed systems, telecommunication systems, Internet, embedded and real-time systems, and multimedia communication and security reliability.

Rigorous Software Development Following the success of the International Symposium on Software Security 2002 (ISSS 2002), held in Keio University, Tokyo, November 2002, ISSS 2003 was held in the Tokyo Institute of Technology, Tokyo, on November 4-6, 2003. This volume is the collection of the papers that were presented in ISSS 2003. The papers that were presented cover a wide range of topics related to software security. They cover the areas of security and reliability of software systems, security and reliability of software systems, security and reliability of software systems, and security and reliability of software systems. The papers are organized into topical sections on program analysis, transformation and optimization, type system, separation logic, logic and foundation theory, software security and verification, and software security and verification.

Programming Languages and Systems This book constitutes the thoroughly refereed post-workshop proceedings of the Third International Workshop, TYPES’99, organized by the ESPRIT Working Group 21900, in Løkken, Sweden, in June 1999. The 11 revised full papers presented in this volume were carefully reviewed and selected during two rounds of refereeing. All current issues on type theory and type systems and their applications to programming and proof theory are addressed.
Distributed and Parallel Systems These proceedings contain a refereed selection of papers presented at the Second Annual Workshop of the Types Working Group (Computer- Assisted Reasoning based on Type Theory, EUiST project 29001), which was held April 24-28, 2002 in Hotel Erica, Berg en Dal (close to Nijmegen), The Netherlands. The workshop was attended by about 90 researchers. On April 27, there was a special afternoon celebrating the 60th birthday of Per Martin-Löf, one of the founding fathers of the Types community. The afternoon consisted of the following three invited talks: “Constructive Validity Revisited” by Dana Scott. “From the Rules of Logic to the Logic of Rules” by Jean-Yves Girard, and “And The Varieties of Type Theory” by Peter Aczel. The contents of these talks are not discussed in these proceedings, but the videos of the talks and the slides used by the speakers are available at http://www.cs.kun.nl/~nif/hd/PSG/Day7/Day7Talks.htm The preceding workshop of the Types Working Group under EUiST project 29001 was held in 2000 in Durham, UK. The workshops Types 2000 and Types 2002 followed a series of meetings organized in the period 1993 - 1999 within previous types projects (ESPRIT BRA 8435 and ESPRIT Working Group 21900). The proceedings of these earlier Types workshops were also published in the LNCS series, as volumes 806, 996, 1158, 1521, 1657, 1956 and 2277. ESPRIT BRA 8435 was a continuation of ESPRIT Action 3245, Logical Frameworks - Sign, Implementation and Experiments.

Software Security - Theories and Systems ETAPS 2001 was the fourth instance of the European Joint Conferences on Theory and Practice of Software. ETAPS is an annual federated conference that was established in 1998 by combining a number of existing and new conferences. This year it comprised 8 conferences (FOSSACS, FASE, ESOP, CC, TACAS), ten satellite workshops (CMCS, ETI Day, IJES, LTDE, MMAA, PL, ReMLs, UNIGRA, WADT, WUTEL), seven invited lectures, a debate, and ten tutorials. The events that comprise ETAPS address various aspects of the system development process, including specification, design, implementation, analysis and improvement. The languages, methodologies, and tools which support these activities are all within its scope. The two themes of theory and practice are represented, with an inclination towards theory with a practical motivation on one hand and sound-based practice on the other. Many of the issues involved in software design apply to systems in general, including hardware systems, and the emphasis on software is not intended to be exclusive.

Logic for Programming, Artificial Intelligence, and Reasoning An introduction to fundamental theories of concurrent computation and associated programming languages for developing distributed and mobile computing systems. Starting from the premise that understanding the foundations of concurrent programming is key to developing distributed systems, this book presents the fundamentals of concurrent computing and then introduces the programming languages that help develop distributed computing systems at a high level of abstraction. The major theories of concurrent computation—including the π-calculus, the actor model, the join calculus, and mobile ambients—are explained with a focus on how they help design and reason about distributed and mobile computing systems. The book then presents programming languages that follow the theoretical models already described, including Pic, SALSA, and JoCam. The parallel structure of the chapters in both part one (theory) and part two (practice) enable the reader not only to compare the different theories but also to see clearly how a programming language supports a theoretical model. The book is unique in bridging the gap between the theory and the practice of developing distributed computing systems. It can be used as a textbook for graduate and advanced undergraduate students in computer science or as a reference for researchers in the area of programming techniques. The book is organized into two parts. In the first part, the basic principles of concurrency, distribution, and mobility without getting bogged down in syntactic details of specific programming languages. Once the theory is understood, the practical part of implementing a system in an actual programming language becomes much easier.

Tools and Algorithms for the Construction and Analysis of Systems The 23 papers presented together with 4 invited papers 2 system and tool presentations and 1 tutorial lecture were carefully reviewed and selected from 95 initial submissions. The papers are devoted to both foundational and practical issues in programming languages and systems and feature current research in the following areas: semantics, logics, foundational theory, design of languages and foundational calculi, type systems, compilers, interpreters, abstract machines, program analysis, derivation, transformation, software security, safety, verification, concurrency, constraints, domain-specific languages, as well as tools for programming, verification, and implementation.

Formal Techniques for Distributed Systems ETAPS’99 is the second instance of the European Joint Conferences on Theory and Practice of Software. ETAPS is an annual federated conference that was established in 1998 by combining a number of existing and new conferences. This year it comprised 8 conferences (CMCS, A5, ESOP, CACSD, ETI Day, IJES, LTDE, MMAA), and six contributed tutorials.

Tools and Algorithms for the Construction and Analysis of Systems The 23 papers presented together with 4 invited papers 2 system and tool presentations and 1 tutorial lecture were carefully reviewed and selected from 95 initial submissions. The papers are devoted to both foundational and practical issues in programming languages and systems and feature current research in the following areas: semantics, logics, foundational theory, design of languages and foundational calculi, type systems, compilers, interpreters, abstract machines, program analysis, derivation, transformation, software security, safety, verification, concurrency, constraints, domain-specific languages, as well as tools for programming, verification, and implementation.

Programming Languages and Systems This book constitutes the thoroughly refereed post-proceedings of the Third International Workshop on Types in Compilation, TIC 2000, held in Montreal, Canada in September 2000. The seven revised full papers presented have been carefully reviewed and selected from the workshop papers for inclusion in the book. The book focuses on the application of types in the implementation of programming languages. Among the topics addressed are intersection and union types, elimination, Java dynamic linking and loading, typed Assembly language, dynamic linking of native code, and type for recursive data structures.

Programmed Languages and Systems These proceedings contain a refereed selection of papers presented at the Second Annual Workshop of the Types Working Group (Computer-Assisted Reasoning based on Type Theory, EUiST project 29001), which was held April 24-28, 2002 in Hotel Erica, Berg en Dal (close to Nijmegen), The Netherlands. The workshop was attended by about 90 researchers. On April 27, there was a special afternoon celebrating the 60th birthday of Per Martin-Löf, one of the founding fathers of the Types community. The afternoon consisted of the following three invited talks: “Constructive Validity Revisited” by Dana Scott. “From the Rules of Logic to the Logic of Rules” by Jean-Yves Girard, and “And The Varieties of Type Theory” by Peter Aczel. The contents of these talks are not discussed in these proceedings, but the videos of the talks and the slides used by the speakers are available at http://www.cs.kun.nl/~nif/hd/PSG/Day7/Day7Talks.htm The preceding workshop of the Types Working Group under EUiST project 29001 was held in 2000 in Durham, UK. The workshops Types 2000 and Types 2002 followed a series of meetings organized in the period 1993 - 1999 within previous types projects (ESPRIT BRA 8435 and ESPRIT Working Group 21900). The proceedings of these earlier Types workshops were also published in the LNCS series, as volumes 806, 996, 1158, 1521, 1657, 1956 and 2277. ESPRIT BRA 8435 was a continuation of ESPRIT Action 3245, Logical Frameworks - Sign, Implementation and Experiments.

Software Security - Theories and Systems ETAPS 2001 was the fourth instance of the European Joint Conferences on Theory and Practice of Software. ETAPS is an annual federated conference that was established in 1998 by combining a number of existing and new conferences. This year it comprised 8 conferences (FOSSACS, FASE, ESOP, CC, TACAS), ten satellite workshops (CMCS, ETI Day, IJES, LTDE, MMAA, PL, ReMLs, UNIGRA, WADT, WUTEL), seven invited lectures, a debate, and ten tutorials. The events that comprise ETAPS address various aspects of the system development process, including specification, design, implementation, analysis and improvement. The languages, methodologies, and tools which support these activities are all within its scope. Di...
Programming Distributed Computing Systems This book constitutes a refereed post-workshop selection of papers presented at the 6th International Workshop on Computer-Aided Systems Theory, EURoCAST'97, held in Las Palmas de Gran Canaria, Spain, in February 1997. The 50 revised full papers presented were carefully selected for inclusion in the volume. The book is divided into sections on design environments and tools, theory and methods, engineering systems, intelligent systems, signal processing, and specific methods and applications.

ECOOP 2013 -- Object-Oriented Programming This volume contains the papers presented at the Eighth International Conference on Object-Oriented Programming, Artificial Intelligence and Reasoning (ECOOP 2001), held on December 3-7, 2001, at the University of Havana (Cuba), together with the Second International Workshop on Implementation of Logics. There were 112 submissions, of which 19 belonged to the special submissions category of experimental papers, intended to describe implementations or comparisons of systems, or experiments with systems. Each submission was reviewed by at least three program committee members and an electronic program committee meeting was held via the Internet. The high number of submissions caused a large amount of work, and we are very grateful to the other 31 PC members for their efficiency and for the quality of their reviews and discussions. Finally, the committee decided to accept 40 papers in the theoretical category, 9 experimental papers. In addition to the refereed papers, this volume contains an extended abstract of the invited talk by Frank Wolter. Two other invited lectures were given by Matthias Baaz and Manuel Hermenegildo. Apart from the program committee, we would also like to thank the other people who made LPAR 2001 possible: the additional referees, the local arrangements chair Luciano Garcia, Andrés Navarro and Oscar Guell, who ran the internet-based submission software and the program committee discussion software at the ISI Department lab in Barcelona, and Bill McCune, whose program committee management software was used.

Central European Functional Programming School This book constitutes the refereed proceedings of the 18th European Symposium on Programming, ESOP 2009, held in York, UK, in March 2009, as part of ETAPS 2009, the European Joint Conferences on Theory and Practice of Software. The 26 revised full papers presented together with abstracts of invited talks were carefully reviewed and selected from 98 full paper submissions. The topics addressed are typed functional programming, computational effects, types for object-oriented languages, verification, security, concurrency, service-oriented computing, parallel and concurrent programming.

Programming Languages and Systems The use of mathematical methods in the development of software is essential when reliable systems are sought, in particular they are now strongly recommended by the official norms adopted in the production of critical software. Program Verification is the area of computer science that studies mathematical methods for checking that a program conforms to its specification. This text is a self-contained introduction to program verification using logic-based methods, presented in the broader context of formal methods for software engineering. The idea of specifying the behaviour of individual software components by attaching contracts to them is now a widely followed approach in program development, which has given rise notably to the development of a number of behavioural interface specification languages and program verification tools. A foundation for the static verification of programs based on contract-annotated routines is laid out in the book. These can be independently verified, which provides a modular approach to the verification of software. The text assumes only basic knowledge of standard mathematical concepts that should be familiar to any computer science student. It includes a self-contained introduction to propositional logic and first-order reasoning with theories, followed by a study of program verification that combines theoretical and practical aspects - from a program logic (a variant of Hoare logic for programs containing user-provided annotations) to the use of a realistic tool for the verification of C programs (annotated using the ACSL specification language), through the generation of verification conditions and the static verification of runtime errors.

Practical Aspects of Declarative Languages This book constitutes the refereed proceedings of the Third International Conference on Generative Programming and Component Engineering, GPCE 2004, held in Vancouver, Canada in October 2004. The 25 revised full papers presented together with abstracts of 2 invited talks were carefully reviewed and selected from 75 submissions. The papers are organized in topical sections on aspect-orientation, staged programming, types for meta-programming, meta-programming, model-driven approaches, product lines, and domain-specific languages and generation.

Active Networks Distributed and Parallel Systems: From Cluster to Grid Computing, is an edited volume based on DAPSYS 2006, the 6th Austrian-Hungarian Workshop on Distributed and Parallel Systems, which is dedicated to all aspects of distributed and parallel computing. The workshop was held in conjunction with the 2nd Austrian Grid Symposium in Innsbruck, Austria in September 2006. This book is designed for a professional audience composed of practitioners and researchers in industry. It is also suitable for advanced-level students in computer science.

Copyright code: Softsares7574d33b9c2728534d2b7d4