Advanced Symbolic Analysis For Compilers New Techniques And Algorithms For Symbolic Program Analysis And Optimization Lecture Notes In Computer Science

This book presents novel symbolic control and data flow techniques as well as symbolic techniques and algorithms for program analysis and program optimization. Program contexts, defining a new symbolic description of program semantics for control and data flow analysis, are at the center of the techniques and methods introduced. The authors develop solutions for a number of problems encountered in program analysis by using program contexts. The solutions proposed are efficient, versatile, unified, and more general than most existing methods. The authors' symbolic analysis framework is implemented as a prototype as part of the Vienna High Performance Compiler.

This book constitutes the thoroughly refereed post-conference proceedings of the 27th International Workshop on Languages and Compilers for Parallel Computing, LCPC 2014, held in Hillsboro, OR, USA, in September 2014. The 25 revised full papers were carefully reviewed and selected from 39 submissions. The papers are organized in topical sections on accelerator programming; algorithms for parallelism; compilers; debugging; vectorization.

ETAPS'99 is the second instance of the EuropeanJoint Conferences on T- ory 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 comprises ?ve conferences (FOSSACS, FASE, ESOP, CC, TACAS), four satellite workshops (CMCS, AS, WAGA, CoFI), seven invited lectures, two invited tutorials, and six contributed tutorials. The events that comprise ETAPS address various aspects of the system - velopment process, including speci?cation, design, implementation, analysis and improvement. The languages, methodologies and tools which support these - tivities are all well within its scope. Di?erent blends of theory and practice are represented, with an inclination towards theory with a practical motivation on one hand and soundly-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.

The CC program committee is pleased to present this volume with the p- ceedings of the 13th International Conference on Compiler Construction (CC 2004). CC continues to provide an exciting forum for researchers, educators, and practitioners to exchange ideas on the latest developments in compiler te- nology, programming language implementation, and language design. The c- ference emphasizes practical and experimental work and invites contributions on methods and tools for all aspects of compiler technology and all language paradigms. This volume serves as the permanent record of the 19 papers accepted for presentation at CC 2004 held in Barcelona, Spain, during April 1–2, 2004. The 19 papers in this volume were selected from 58 submissions. Each paper was assigned to three committee members for review. The program committee met for one day in December 2003 to discuss the papers and the reviews. By the end of the meeting, a consensus emerged to accept the 19 papers presented in this volume. However, there were many other quality submissions that could not be accommodated in the program; hopefully they will be published elsewhere.

ThecontinuedsuccessoftheCCconferenceserieswouldnotbepossiblewi- out the help of the CC community. I would like to gratefully acknowledge and thank all of the authors who submitted papers and the many external reviewers who wrote reviews.

Encyclopedia of Parallel Computing

... Iberian Conference, IbPRIA ... : Proceedings

Parallel and Distributed Processing and Applications

Languages, Compilation Techniques, and Run Time Systems

13th International Euro-Par Conference, Rennes, France, August 28-31, 2007, Proceedings

13th International Workshop, LCPC 2000, Yorktown Heights, NY, USA, August 10-12, 2000, Revised Papers

As Chairmen of HiPEAC 2005, we have the pleasure of welcoming you to the proceedings of the ?rst international conference promoted by the HiPEAC N- work of Excellence. During the last year, HiPEAC has been building its clusters of researchers in computer architecture and advanced compiler techniques for embedded and high-performance computers. Recently, the Summer School has been the seed for a fruitful collaboration of renowned international faculty and young researchers from 23 countries with fresh new ideas. Now, the conference promises to be among the premier forums for discussion and debate on these research topics. Theprestigeofasymposiumismainlydeterminedbythequalityofitstech- cal program. This ?rst programlived up to our high expectations, thanks to the largenumber of strong submissions. The ProgramCommittee received a total of 84 submissions; only 17 were selected for presentation as full-length papers and another one as an invited paper. Each paper was rigorously reviewed by three ProgramCommittee members and at least one external referee. Many reviewers spent a great amount of e?ort to provide detailed feedback. In many cases, such feedback along with constructive shepherding resulted in dramatic improvement in the quality of accepted papers. The names of the Program Committee m- bers and the referees are listed in the proceedings. The net result of this team e?ort is that the symposium proceedings include outstanding contributions by authors from nine countries in three continents. In addition to paper presentations, this ?rst HiPEAC conference featured two keynotes delivered by prominent researchers from industry and academia.

The articles in this volume are revised versions of the best papers presented at the Fifth Workshop on Languages and Compilers for Parallel Computing, held at Yale University, August 1992. The previous workshops in this series were held in Santa Clara (1991), Irvine (1990), Urbana (1989), and Ithaca (1988). As in previous years, a reasonable cross-section of some of the best work in the field is presented. The volume contains 35 papers, mostly by authors working in the U.S. or Canada but also by authors from Austria, Denmark, Israel, Italy, Japan and the U.K.

Scalable parallel systems or, more generally, distributed memory systems offer a challenging model of computing and pose fascinating problems regarding compiler optimization, ranging from language design to run time systems. Research in this area is foundational to many challenges from memory hierarchy optimizations to communication optimization. This unique, handbook-like monograph assesses the state of the art in the area in a systematic and comprehensive way. The 21 coherent chapters by leading

researchers provide complete and competent coverage of all relevant aspects of compiler optimization for scalable parallel systems. The book is divided into five parts on languages, analysis, communication optimizations, code generation, and run time systems. This book will serve as a landmark source for education, information, and reference to students, practitioners, professionals, and researchers interested in updating their knowledge about or active in parallel computing.

The objective of program analysis is to automatically determine the p- perties of a program. Tools of software development, such as compilers, p- formance estimators, debuggers, reverse-engineering tools, program veri?- tion/testing/proving systems, program comprehension systems, and program specializationtoolsarelargelydependentonprogramanalysis. Advancedp- gram analysis can: help to ?nd program errors; detect and tune performan- critical code regions; ensure assumed constraints on data are not violated; tailor a generic program to suit a speci?c application; reverse-engineer so- ware modules, etc. A prominent program analysis technique is symbolic a- lysis, which has attracted substantial attention for many years as it is not dependent on executing a program to examine the semantics of a program, and it can yield very elegant formulations of many analyses. Moreover, the complexity of symbolic analysis can be largely independent of the input data size of a program and of the size of the machine on which the program is being executed. In this book we present novel symbolic control and data ?ow repres- tation techniques as well as symbolic techniques and algorithms to analyze and optimize programs. Program contexts which de?ne a new symbolic - scription of program semantics for control and data ?ow analysis are at the center of our approach. We have solved a number of problems encountered in program analysis by using program contexts. Our solution methods are e?cient, versatile, uni?ed, and more general (they cope with regular and irregular codes) than most existing methods.

International Conference and Exhibition, Amsterdam, The Netherlands, April 21-23, 1998 Proceedings

Languages, Compilers and Run-Time Systems for Scalable Computers

28th International Workshop, LCPC 2015, Raleigh, NC, USA, September 9-11, 2015, Revised Selected Papers

Symbolic Analysis

Pattern Recognition and Image Analysis

Euro-Par 2008 Parallel Processing

The 3rd International Workshop on Software Engineering and Middleware {SEM 2002) was held May 20-21, 2002, in Orlando, Florida, as a co-located event of the 2002 International Conference on Software Engineering. The workshop attracted 30 participants from academic and industrial institutions in many countries. Twenty-seven papers were submitted, of which 15 were accepted to create a broad program covering the topics of architectures, specification, components and adaptations, technologies, and services. The focus of the workshop was on short presentations, with substantial dis cussions afterwards. Thus, we decided to include in this proceedings also a short summary of every technical session, which was written by some of the partici pants at the workshop. The workshop invited one keynote speaker, Bobby Jadhav of CalKey, who presented a talk on the design and use of model-driven architecture and middle ware in industry. We would like to thank all the people who helped organize and run the workshop. In particular, we would like to thank the program committee for their careful reviews of the submitted papers, Wolfgang Emmerich for being an excellent General Chair, and the participants for a lively and interesting workshop.

Advanced Symbolic Analysis for CompilersNew Techniques and Algorithms for Symbolic Program Analysis and OptimizationSpringer Science & Business Media

The 15th Workshop on Languages and Compilers for Parallel Computing was held in July 2002 at the University of Maryland, College Park. It was jointly sponsored by the Department of Computer Science at the University of Ma- land and the University of Maryland Institute for Advanced Computer Studies (UMIACS).LCPC2002broughttogetherover60researchersfromacademiaand research institutions from many countries. The program of 26 papers was selected from 32 submissions. Each paper was reviewed by at least three Program Committee members and sometimes by additional reviewers. Prior to the workshop, revised versions of accepted papers were informally published on the workshop's website and in a paper proceedings that was distributed at the meeting. This year, the workshopwas organizedinto sessions of papers on related topics, and each session consisted of two to three 30-minute presentations.Based on feedback from the workshop,the papers were revised and submitted for inclusion in the formal proceedings published in this volume. Two papers were presented at the workshop but later withdrawn from the ?nal proceedings by their authors. We were very lucky to have Bill Carlson from the Department of Defense give the LCPC 2002 keynote speech on "UPC: A C Language for Shared M- ory Parallel Programming." Bill gave an excellent overview of the features and programming model of the UPC parallel programming language.

Static program analysis aims to determine the dynamic behavior of programs without actually executing them. Symbolic analysis is an advanced static program analysis technique that has been successfully applied to memory leak detection, compilation of parallel programs, detection of superfluous bound checks, variable aliases and task deadlocks, and to worst-case execution time analysis. The symbolic analysis information is invaluable for optimizing compilers, code generators, program verification, testing and debugging. In this book we take a novel algebra-based approach to the symbolic analysis of imperative programming languages. Our approach employs path expression algebra to compute the complete control

and data flow analysis information valid at a given program point. This information is then provided for subsequent domain-specific analyses. Our approach derives solutions for arbitrary (even intra-loop) nodes of reducible and irreducible control flow graphs. We prove the correctness of our analysis method. Experimental results show that the problem sizes arising from real-world applications such as the SPEC95 benchmark suite are tractable for our symbolic analysis method.

14th International Euro-Par Conference, Las Palmas de Gran Canaria, Spain, August 26-29, 2008, Proceedings

Symbolic Analysis of Analog Circuits: Techniques and Applications

A Special Issue of Analog Integrated Circuits and Signal Processing

8th International Conference, CC'99, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS'99, Amsterdam, The Netherlands, March 22-28, 1999, Proceedings

Compiler Optimizations for Scalable Parallel Systems

This book constitutes the refereed proceedings of the 14th International Conference on Parallel Computing, Euro-Par 2008, held in Las Palmas de Gran Canaria, Spain, in August 2008. The 86 revised papers presented were carefully reviewed and selected from 264 submissions. The papers are organized in topical sections on support tools and environments; performance prediction and evaluation; scheduling and load balancing; high performance architectures and compilers; parallel and distributed databases; grid and cluster computing; peer-to-peer computing; distributed systems and algorithms; parallel and distributed programming; parallel numerical algorithms; distributed and high-performance multimedia; theory and algorithms for parallel computation; and high performance networks.

This volume presents revised versions of the 32 papers accepted for the Seventh Annual Workshop on Languages and Compilers for Parallel Computing, held in Ithaca, NY in August 1994. The 32 papers presented report on the leading research activities in languages and compilers for parallel computing and thus reflect the state of the art in the field. The volume is organized in sections on fine-grain parallelism, align- ment and distribution, postlinear loop transformation, parallel structures, program analysis, computer communication, automatic parallelization, languages for parallelism, scheduling and program optimization, and program evaluation.

Computer professionals who need to understand advanced techniques for designing efficient compilers will need this book. It provides complete coverage of advanced issues in the design of compilers, with a major emphasis on creating highly optimizing scalar compilers. It includes interviews and printed documentation from designers and implementors of real-world compilation systems.

In Symbolic Analysis for Parallelizing Compilers the author presents an excellent demonstration of the effectiveness of symbolic analysis in tackling important optimization problems, some of which inhibit loop parallelization. The framework that Haghighat presents has proved extremely successful in induction and wraparound variable analysis, strength reduction, dead code elimination and symbolic constant propagation. The approach can be applied to any program transformation or optimization problem that uses properties and value ranges of program names. Symbolic analysis can be used on any transformational system or optimization problem that relies on compile-time information about program variables. This covers the majority of, if not all optimization and parallelization techniques. The book makes a compelling case for the potential of symbolic analysis, applying it for the first time - and with remarkable results - to a number of classical optimization problems: loop scheduling, static timing or size analysis, and dependence analysis. It demonstrates how symbolic analysis can solve these problems faster and more accurately than existing hybrid techniques.

Proceedings ...

7th International Workshop, Ithaca, NY, USA, August 8 - 10, 1994. Proceedings

5th International Workshop, New Haven, Connecticut, USA, August 3-5, 1992. Proceedings

4th International Symposium, ISPA 2006, Sorrento, Italy, December 4-6, 2006, Proceedings

10th International Conference, CC 2001 Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2001 Genova, Italy, April 2-6, 2001 Proceedings

Half a Century of Inspirational Research

This book constitutes the thoroughly refereed post-conference proceedings of the 28th International Workshop on Languages and Compilers for Parallel Computing, LCPC 2015, held in Raleigh, NC, USA, in September 2015. The 19 revised full papers were carefully reviewed and selected from 44 submissions. The papers are organized in topical sections on programming models, optimizing framework, parallelizing compiler, communication and locality, parallel applications and data structures, and correctness and reliability.

This book constitutes the thoroughly refereed post-conference proceedings of the 29th International Workshop on Languages and Compilers for Parallel Computing, LCPC 2016, held in Rochester, NY, USA, in September 2016. The 20 revised full papers presented together with 4 short papers were carefully reviewed. The papers are organized in topical sections on large scale parallelism, resilience and persistence, compiler analysis and optimization, dynamic computation and languages, GPUs and private memory, and runt-time and performance analysis.

This book brings together important contributions and state-of-the-art research results in the rapidly advancing area of symbolic analysis of analog circuits. It is also of interest to those working in analog CAD. The book is an excellent reference, providing insights into some of the most important issues in the symbolic analysis of analog circuits.

This book constitutes the refereed proceedings of the international Joint Modular

Languages Conference, JMLC 2006. The 23 revised full papers presented together with 2 invited lectures were carefully reviewed and selected from 36 submissions. The papers are organized in topical sections on languages, implementation and linking, formal and modelling, concurrency, components, performance, and case studies.

High-Performance Computing and Networking

New Techniques and Algorithms for Symbolic Program Analysis and Optimization

Developments in Language Theory

17th International Workshop, LCPC 2004, West Lafayette, IN, USA, September 22-24, 2004, Revised Selected Papers

ACM Transactions on Programming Languages and Systems

High Performance Embedded Architectures and Compilers

This volume contains the papers presented at the 13th International Workshop on Languages and Compilers for Parallel Computing. It also contains extended abstracts of submissions that were accepted as posters. The workshop was held at the IBM T. J. Watson Research Center in Yorktown Heights, New York. As in previous years, the workshop focused on issues in optimizing compilers, languages, and software environments for high performance computing. This continues a trend in which languages, compilers, and software environments for high performance computing, and not strictly parallel computing, has been the organizing topic. As in past years, participants came from Asia, North America, and Europe. This workshop re?ected the work of many people. In particular, the members of the steering committee, David Padua, Alex Nicolau, Utpal Banerjee, and David Gelernter, have been instrumental in maintaining the focus and quality of the workshop since it was ?rst held in 1988 in Urbana-Champaign. The assistance of the other members of the program committee – Larry Carter, Sid Chatterjee, Jeanne Ferrante, Jans Prins, Bill Pugh, and Chau-wen Tseng – was crucial. The infrastructure at the IBM T. J. Watson Research Center provided trouble-free logistical support. The IBM T. J. Watson Research Center also provided ?nancial support by underwriting much of the expense of the workshop. Appreciation must also be extended to Marc Snir and Pratap Pattnaik of the IBM T. J. Watson Research Center for their support.

This book constitutes the thoroughly refereed post-proceedings of the 17th International Workshop on Languages and Compilers for High Performance Computing, LCPC 2004, held in West Lafayette, IN, USA in September 2004. The 33 revised full papers presented were carefully selected during two rounds of reviewing and improvement. The papers are organized in topical sections on compiler infrastructures; predicting and reducing memory access; locality, tiling, and partitioning; tools and techniques for parallelism and locality; Java for high-performance computing; high-level languages and optimizations; large-scale data sharing; performance studies; program analysis; and exploiting architectural features.

Proceedings -- Parallel Computing.

Honors Professor Antoni Mazurkiewicz, who during his long scientific career made fundamental contributions to theoretical computer science. This book includes contributions, which span a range of research areas, including the theory of programming, models of concurrent and distributed systems, and (de)composition methods for Pietri nets.

16th International Workshop, LCPC 2003, College Sation, TX, USA, October 2-4, 2003, Revised Papers

Symbolic Analysis Techniques for Effective Automatic Parallelization

29th International Workshop, LCPC 2016, Rochester, NY, USA, September 28-30, 2016, Revised Papers

Testing of Communicating Systems

Advanced Compiler Design Implementation

Euro-Par 2007 Parallel Processing

This book constitutes the refereed proceedings of the 4th International Symposium on Parallel and Distributed Processing and Applications, ISPA 2006, held in Sorrento, Italy in November 2006. The 79 revised full papers presented together with five keynote speeches cover architectures, networks, languages, algorithms, middleware, cooperative computing, software, and applications.

This book constitutes the thoroughly refereed post-proceedings of the 18th International Workshop on Languages and Compilers for Parallel Computing, LCPC 2005, held in Hawthorne, NY, USA in October 2005. The 26 revised full papers and eight short papers presented were carefully selected during two rounds of reviewing and improvement. The papers are organized in topical sections.

This book constitutes the thoroughly refereed post-proceedings of the 16th International Workshop on Languages and Compilers for Parallel Computing, LCPC 2003, held in College Station, Texas, USA, in October 2003. The 35 revised full papers presented were selected from 48 submissions during two rounds of reviewing and improvement upon presentation at the workshop. The papers are organized in topical sections on adaptive optimization, data locality, parallel languages, high-level transformations, embedded systems, distributed systems software, low-level transformations, compiling for novel architectures, and optimization infrastructure.

This volume constitutes the refereed proceedings of the 13th International Conference on Parallel Computing. The papers are organized into topical sections covering support tools and environments, performance prediction and evaluation, scheduling and load balancing, compilers for high performance, parallel and distributed databases, grid and cluster computing, peer-to-peer computing, distributed systems and algorithms, and more.

Combinatorial Pattern Matching

Software Engineering and Middleware

Algorithms and Complexity

15th Workshop, LCPC 2002, College Park, MD, USA, July 25-27, 2002, Revised Papers

Modular Programming Languages

Symbolic Analysis for Parallelizing Compilers

Language, Compilers and Run-time Systems for Scalable Computers contains 20 articles based on presentations given at the third workshop of the same title, and 13 extended abstracts from the poster session. Starting with new developments in classical problems of parallel compiler design, such

as dependence analysis and an exploration of loop parallelism, the book goes on to address the issues of compiler strategy for specific architectures and programming environments. Several chapters investigate support for multi-threading, object orientation, irregular computation, locality enhancement, and communication optimization. Issues of the interface between language and operating system support are also discussed. Finally, the load balance issues are discussed in different contexts, including sparse matrix computation and iteratively balanced adaptive solvers for partial differential equations. Some additional topics are also discussed in the extended abstracts. Each chapter provides a bibliography of relevant papers and the book can thus be used as a reference to the most up-to-date research in parallel software engineering.

Containing over 300 entries in an A-Z format, the Encyclopedia of Parallel Computing provides easy, intuitive access to relevant information for professionals and researchers seeking access to any aspect within the broad field of parallel computing. Topics for this comprehensive reference were selected, written, and peer-reviewed by an international pool of distinguished researchers in the field. The Encyclopedia is broad in scope, covering machine organization, programming languages, algorithms, and applications. Within each area, concepts, designs, and specific implementations are presented. The highly-structured essays in this work comprise synonyms, a definition and discussion of the topic, bibliographies, and links to related literature. Extensive cross-references to other entries within the Encyclopedia support efficient, user-friendly searchers for immediate access to useful information. Key concepts presented in the Encyclopedia of Parallel Computing include; laws and metrics; specific numerical and non-numerical algorithms; asynchronous algorithms; libraries of subroutines; benchmark suites; applications; sequential consistency and cache coherency; machine classes such as clusters, shared-memory multiprocessors, special-purpose machines and dataflow machines; specific machines such as Cray supercomputers, IBM's cell processor and Intel's multicore machines; race detection and auto parallelization; parallel programming languages, synchronization primitives, collective operations, message passing libraries, checkpointing, and operating systems. Topics covered: Speedup, Efficiency, Isoefficiency, Redundancy, Amdahls law, Computer Architecture Concepts, Parallel Machine Designs, Benmarks, Parallel Programming concepts & design, Algorithms, Parallel applications. This authoritative reference will be published in two formats: print and online. The online edition features hyperlinks to cross-references and to additional significant research. Related Subjects: supercomputing, high-performance computing, distributed computing

*To effectively translate real programs written in standard, sequential languages into parallel computer programs, parallelizing compilers need advanced techniques such as powerful dependence tests, array privatization, generalized induction variable substitution, and reduction parallelization. All of these techniques need or can benefit from symbolic analysis. To determine what kinds of symbolic analysis techniques can significantly improve the effectiveness of parallelizing Fortran compilers, we compared the automatically and manually parallelized versions of the Perfect Benchmarks. The techniques identified include: data dependence tests for nonlinear expressions, constraint propagation, interprocedural constant propagation, array summary information, and run time tests. We have developed algorithms for two of these identified symbolic analysis techniques: nonlinear data dependence analysis and constraint propagation. For data dependence analysis nonlinear expressions, (e.g., A(n * i + j), where $1 le j le n), $ we developed a data dependence test called the Range Test. The Range Test proves independence by determining whether certain symbolic inequalities hold for a logical permutation of the loop nest. We use a technique called Range Propagation to prove these symbolic inequalities. For constraint propagation, we developed a technique called Range Propagation. Range Propagation computes the range of values that each variable can take at each point of a program. A range is a symbolic lower and upper bound on the values taken by a variable. Range propagation also includes a facility to compare arbitrary expressions under the constraints imposed by a set of ranges. We have developed both a simple but slow algorithm and a fast and demand-driven but complex algorithm to compute these ranges. The Range Test and Range Propagation have been fully implemented in Polaris, a parallelizing compiler being developed at the University of Illinois. We have found that these techniques significantly improve the effectiveness of automatic parallelization. We have also found that these techniques are reasonably efficient.*

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 ve conferences (FOSSACS, FASE, ESOP, CC, TACAS), ten satellite workshops (CMCS, ETI Day, JOSES, LDTA, MMAABS, PFM, RelMiS, UNIGRA, WADT, WTUML), seven invited lectures, a debate, and ten tutorials. The events that comprise ETAPS address various aspects of the system de- lopment process, including speci cation, design, implementation, analysis, and improvement. The languages, methodologies, and tools which support these - tivities are all well within its scope. Di erent blends of theory and practice are represented, with an inclination towards theory with a practical motivation on one hand and soundly-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.

Honoring the Scientific Influence of Antoni Mazurkiewicz

13th International Conference, CC 2004, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2004, Barcelona, Spain, March 29 - April 2, 2004, Proceedings

... Annual Symposium

7th Joint Modular Languages Conference, JMLC 2006, Oxford, UK, September 13-15, 2006, Proceedings

Advanced Symbolic Analysis for Compilers

Proceedings