Acces PDF Control Flow Graph Based Framework For Effective

When people should go to the books stores, search opening by shop, shelf by shelf, it is really problematic. This is why we give the books compilations in this website. It will enormously ease you to look guide **Control Flow Graph Based Framework For Effective** as you such as.

By searching the title, publisher, or authors of guide you in fact want, you can discover them rapidly. In the house, workplace, or perhaps in your method can be every best area within net connections. If you objective to download and install the Control Flow Graph Based Framework For Effective, it is entirely easy then, previously currently we extend the join to purchase and make bargains to download and install Control Flow Graph Based Framework For Effective fittingly simple!

TPEQBP - DRAVEN AUBREY

This two-volume-set (LNCS 8384 and 8385) constitutes the refereed proceedings of the 10th International Conference of Parallel Processing and Applied Mathematics, PPAM 2013, held in Warsaw, Poland, in September 2013. The 143 revised full papers presented in both volumes were carefully reviewed and selected from numerous submissions. The papers cover important fields of parallel/distributed/cloud computing and applied mathematics, such as numerical algorithms and parallel scientific computing; parallel non-numerical algorithms; tools and environments for parallel/distributed/cloud computing; applications of parallel computing; applied mathematics, evolutionary computing and metaheuristics.

This book constitutes the proceedings of the 23rd International Conference on Compiler Construction, CC 2014, which was held as part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2014, which took place in Grenoble, France, in April 2014. The 10 full papers and 4 tool papers included in this volume were carefully reviewed and selected from 47 submissions; the book also contains one invited talk. The papers are organized in topical sections named: program analysis and optimization; parallelism and parsing and new trends in compilation.

This book constitutes the refereed proceedings of the 34th International Conference on Computer Safety, Reliability, and Security, SAFECOMP 2015, held in Delft, The Netherlands, in September 2014. The 32 revised full papers presented together with 3 invited talks were carefully reviewed and selected from 104 submissions. The papers are organized in topical sections on flight systems, automotive embedded systems, automotive software, error detection, medical safety cases, medical systems, architecture and testing, safety cases, security attacks, cyber security and integration, and programming and compiling.

The widespread use of object-oriented languages and Internet security concerns are just the beginning. Add embedded systems, multiple memory banks, highly pipelined units operating in parallel, and a host of other advances and it becomes clear that current and future computer architectures pose immense challenges to compiler designers-challenges th

Normal 0 false false false EN-US X-NONE X-NONE This book constitutes the thoroughly refereed post-conference proceedings of the 6th International Conference on Trusted Systems, INTRUST 2014, held in Beijing, China, in December 2014. The conference brings together academic and industrial researchers, designers, and implementers with end-users of trusted systems, in order to foster the exchange of ideas in this challenging and fruitful area. The revised full papers focus on the theory, technologies and applications of trusted systems and cover all aspects of trusted computing systems, including trusted modules, platforms, networks, services and applications, from their fundamental features and functionalities to design principles, architecture and implementation technologies. /* Style Definitions */ table.MsoNormalTable {mso-style-name:"Table Normal"; mso-tstyle-rowband-size:0; mso-tstyle-colband-size:0; mso-style-noshow:yes; mso-style-priority:99; mso-style-qformat:yes; mso-style-parent:""; mso-padding-alt:0in 5.4pt 0in 5.4pt; mso-para-margin:0in; mso-para-margin-bottom:.0001pt; mso-pagination:widow-orphan; font-size:11.0pt; font-family:"Calibri","sans-serif"; mso-ascii-font-family:Calibri; mso-ascii-theme-font:minor-latin; mso-fareast-font-family:"Times New Roman"; mso-fareast-theme-font:minor-fareast; mso-hansi-font--family:Calibri; mso-hansi-theme-font:minor-latin; mso-bidi-font--family:"Times New Roman"; mso-bidi-theme-font:minor-bidi;}

This book constitutes the refereed proceedings of the 21st International Conference on Theorem Proving in Higher Order Logics, TPHOLs 2008, held in Montreal, Canada, in August 2008. The 17 revised full papers presented together with 1 proof pearl (concise and elegant presentations of interesting examples), 5 tool presentations, and 2 invited papers were carefully reviewed and selected from 40 submissions. The papers cover all aspects of theorem proving in higher order logics as well as related topics in theorem proving and verification such as formal semantics of specification, modeling, and programming languages, specification and verification of hardware and software, formalisation of mathematical theories, advances in theorem prover technology, as well as industrial application of theorem provers.

This book constitutes the refereed proceedings of the 8th International Symposium on Engineering Secure Software and Systems, ESSoS 2016, held in London, UK, in April 2016. The 13 full papers presented together with 3 short papers and 1 invited talk were carefully reviewed and selected from 50 submissions. The goal of this symposium, is to bring together researchers and practitioners to advance the states of the art and practice in secure software engineering. The presentations and associated publications at ESSoS 2016 contribute to this goal in several directions: First, by improving methodologies for secure software engineering (such as flow analysis and policycompliance). Second, with results for the detection and analysis of software vulnerabilities and the attacks they enable. Finally, for securing software for specific application domains (such as mobile devices and access control).

Ralf Karrenberg presents Whole-Function Vectorization (WFV), an approach that allows a compiler to automatically create code that exploits data-parallelism using SIMD instructions. Data-parallel applications such as particle simulations, stock option price estimation or video decoding require the same computations to be performed on huge amounts of data. Without WFV, one processor core executes a single instance of a data-parallel function. WFV transforms the function to execute multiple instances at once using SIMD instructions. The author describes an advanced WFV algorithm that includes a variety of analyses and code generation techniques. He shows that this approach improves the perfor-

mance of the generated code in a variety of use cases.

The 17th International Workshop on Languages and Compilers for High Performance Computing was hosted by Purdue University in September 2004 on Purdue campus in West Lafayette, Indiana, USA.

This book describes analytical models and estimation methods to enhance performance estimation of pipelined multiprocessor systems-on-chip (MPSoCs). A framework is introduced for both design-time and run-time optimizations. For design space exploration, several algorithms are presented to minimize the area footprint of a pipelined MPSoC under a latency or a throughput constraint. A novel adaptive pipelined MPSoC architecture is described, where idle processors are transitioned into low-power states at run-time to reduce energy consumption. Multi-mode pipelined MPSoCs are introduced, where multiple pipelined MPSoCs optimized separately are merged into a single pipelined MPSoC, enabling further reduction of the area footprint by sharing the processors and communication buffers. Readers will benefit from the authors' combined use of analytical models, estimation methods and exploration algorithms and will be enabled to explore billions of design points in a few minutes.

This book constitutes the thoroughly revised selected papers from the 13th International Conference on Formal Aspects of Component Software, FACS 2016, held in Besançon, France, in October 2016. The 11 full papers presented together with one tool paper and 3 invited papers were carefully reviewed and selected from 27 submissions. FACS 2016 is concerned with how formal methods can be used to make component-based and service-oriented software development succeed. Formal methods have provided a foundation for component-based software by successfully addressing challenging issues such as mathematical models for components, composition and adaptation, or rigorous approaches to verification, deployment, testing, and certification.

From the basics to the most advanced quality of service (QoS) concepts, this all encompassing, first-of-its-kind book offers an in-depth understanding of the latest technical issues raised by the emergence of new types, classes and qualities of Internet services. The book provides end-to-end QoS guidance for real time multimedia communications over the Internet. It offers you a multiplicity of hands-on examples and simulation script support, and shows you where and when it is preferable to use these techniques for QoS support in networks and Internet traffic with widely varying characteristics and demand profiles. This practical resource discusses key standards and protocols, including real-time transport, resource reservation, and integrated and differentiated service models, policy based management, and mobile/wireless QoS. The book features numerous examples, simulation results and graphs that illustrate important concepts, and pseudo codes are used to explain algorithms. Case studies, based on freely available Linux/FreeBSD systems, are presented to show you how to build networks supporting Quality of Service. Online support material including presentation foils, lab exercises and additional exercises are available to text adopters.

"This book provides a comprehensive assessment of the latest developments in Web services research, focusing on composing and coordinating Web services, XML security, and service oriented architecture, and presenting new and emerging research in the Web services discipline"--Provided by publisher.

In August 1999, the Twelfth Workshop on Languages and Compilers for P- allel Computing (LCPC) was hosted by the Hierarchical Tiling Research group from the Computer Science and Engineering Department at the University of California San Diego (UCSD). The workshop is an annual international forum for leading research groups to present their current research activities and the latest results. It has also been a place for researchers and practitioners to - teract closely and exchange ideas about future directions. Among the topics of interest to the workshop are language features, code generation, debugging, - timization, communication and distributed shared memory libraries, distributed object systems, resource management systems, integration of compiler and r- time systems, irregular and dynamic applications, and performance evaluation. In 1999, the workshop was held at the International Relations/Paci c Studies Auditorium and the San Diego Supercomputer Center at UCSD. Seventy-seven researchers from Australia, England, France, Germany, Korea, Spain, and the United States attended the workshop, an increase of over 50% from 1998.

This book constitutes the refereed proceedings of the 13th International Symposium on Static Analysis, SAS 2006. The book presents 23 revised full papers together with the abstracts of 3 invited talks. The papers address all aspects of static analysis including program and systems verification, shape analysis and logic, termination analysis, bug detection, compiler optimization, software maintenance, security and safety, abstract interpretation and algorithms, abstract domain and data structures and more.

The two volume set LNCS 7133 and LNCS 7134 constitutes the thoroughly refereed post-conference proceedings of the 10th International Conference on Applied Parallel and Scientific Computing, PARA 2010, held in Reykjavík, Iceland, in June 2010. These volumes contain three keynote lectures, 29 revised papers and 45 minisymposia presentations arranged on the following topics: cloud computing, HPC algorithms, HPC programming tools, HPC in meteorology, parallel numerical algorithms, parallel computing in physics, scientific computing tools, HPC software engineering, simulations of atomic scale systems, tools and environments for accelerator based computational biomedicine, GPU computing, high performance computing interval methods, real-time access and processing of large data sets, linear algebra algorithms and software for multicore and hybrid architectures in honor of Fred Gustavson on his 75th birthday, memory and multicore issues in scientific computing - theory and praxis, multicore algorithms and implementations for application problems, fast PDE solvers and a posteriori error estimates, and scalable tools for high performance computing.

This book includes a collection of peer-reviewed best selected research papers presented at the Third International Conference on Advances in Distributed Computing and Machine Learning (ICADCML 2022), organized by Department of Computer Science and Engineering, National Institute of Technology, Warangal, Telangana, India, during 15–16 January 2022. This book presents recent innovations in the field of scalable distributed systems in addition to cutting edge research in the field of Internet of Things (IoT) and blockchain in distributed environments.

This book presents the proceedings of the First International EURO-PAR Conference on Parallel Processing, held in Stockholm, Sweden in August 1995. EURO-PAR is the merger of the former PARLE and CONPAR-VAPP conference series; the aim of this merger is to create the premier annual scientific conference on parallel processing in Europe. The book presents 50 full revised research papers and 11 posters selected from a total of 196 submissions on the basis of 582 reviews. The scope of the contributions spans the full spectrum of parallel processing ranging from theory over design to application; thus the volume is a "must" for anybody interested in the scientific aspects of parallel processing or its advanced applications.

This book constitutes the refereed proceedings of the 8th International Conference on Grid and Pervasive Computing, GPC 2016, held in Seoul, Korea, in May 2016. The 20 revised papers were

carefully reviewed and selected from 94 submissions. The conference contains various aspects including green computing, cloud computing, virtualisation, data and storage, and network security.

This book constitutes the refereed proceedings of the 25th International Conference on Logic Programming, ICLP 2009, held in Pasadena, CA, USA, in July2009. The 29 revised full papers together with 9 short papers, 4 invited talks, 4 invited tutorials, and the abstracts of 18 doctoral consortium articles were carefully reviewed and selected from 69 initial submissions. The papers cover all issues of current research in logic programming, namely semantic foundations, formalisms, nonmonotonic reasoning, knowledge representation, compilation, memory management, virtual machines, parallelism, program analysis, program transformation, validation and verification, debugging, profiling, concurrency, objects, coordination, mobility, higher order, types, modes, programming techniques, abductive logic programming, answer set programming, constraint logic programming, inductive logic programming, alternative inference engines and mechanisms, deductive databases, data integration, software engineering, natural language, web tools, internet agents, artificial intelligence, bioinformatics.

This book presents a modular framework for slicing in the proof assistant Isabelle/HOL which is based on abstract control flow graphs. Building on such abstract structures renders the correctness results language-independent. To prove that they hold for a specific language, it remains to instantiate the framework with this language, which requires a formal semantics of this language in Isabelle/HOL. We show that formal semantics even for sophisticated high-level languages are realizable.

Information obtained from static software analyses is valuable to applications ranging from compiler optimizations to program understanding tools, from software validation to security. One such analysis is flow analysis. Traditionally, program flow analysis has been modeled using directed graphs called control flow graphs. Though they provide natural flow sensitivity, control flow graphs are unable to provide complete context sensitivity. This thesis describes a new representation of control flow information termed path grammars. Path grammars, being inherently both flow and context sensitive, overcome the limitations of standard control flow graphs and permit the incorporation of both intraprocedural and interprocedural analyses into a single comprehensive model. The algorithm for the generation of path grammars from an abstract syntax tree or intermediate representation is described, and numerous examples are presented. Reaching definitions, a classic data flow analysis problem, was implemented in both path grammar and control flow graph frameworks. Empirical tests with both frameworks are compared for a number of SPEC2000 benchmark programs, and numerous statistics are collected. For the benchmarks tested, a thirty fold speedup on average and approximate three fold savings in maximum memory usage was seen using path grammars, demonstrating their practicality in both time and space.

The International Conference on Compiler Construction provides a forum for presentation and discussion of recent developments in the area of compiler construction, language implementation and language design. Its scope ranges from compilation methods and tools to implementation techniques for specific requirements on languages and target architectures. It also includes language design and programming environment issues which are related to language translation. There is an emphasis on practical and efficient techniques. This volume contains the papers selected for presentation at CC '94, the fifth International Conference on Compiler Construction, held in Edinburgh, U.K., in April 1994.

Bachelor Thesis from the year 2019 in the subject Computer Science - IT-Security, grade: 1,0, Technical University of Munich (Fakultät für Informatik), language: English, abstract: The topic of this thesis is to develop a graph-based static analysis framework for Java code that tolerates incomplete or non-compiling source code. For this purpose, the concept of Code Property Graphs (CPGs) is to be researched and extended, in order to provide information about more complex erroneous patterns in Java source code. Additionally, an evaluation of the resulting graph model is to be performed, by searching for cryptographic vulnerabilities in publicly available Java projects. This evaluation needs to show, whether this graph-based analysis approach is capable of finding security issues in Java code, and how feasible the analysis is from a performance point of view. Automatic code analysis is a widely used technique to find and eliminate errors in software projects. Instead of executing the program and verify that its behavior is correct, as dynamic analysis does it, static analysis is applied on its source code. Here, we search for suspicious patterns that are likely to indicate erroneous behavior. A special type of software bugs are those errors, that lead to security vulnerabilities. In this case, attackers may be able to undermine fundamental security aspects, by exfiltrating sensitive user data from server applications or assume control over the machine running the program in question. Security vulnerabilities in the code can have drastic consequences, which is why it is important to identify them as fast as possible and fix them immediately afterwards. This thesis extends the concept of Code Property Graphs (CPGs), which has been proposed for static analysis of C/C++ code, to be applied on programs and incomplete code snippets written in Java. Unifying Abstract Syntax Trees (ASTs), Control Flow Graphs (CFGs) and Data Flow Graphs (DFGs) in a single datastructure, this approach enables searching for vulnerabilities whose code patterns are spread out across the boundaries of single methods and classes. These patterns are identified using the graph query language cypher, which is provided by the graph database Neo4j. In an evaluation run on 100 public repositories on GitHub using cryptography, 135 findings of cryptographic API misuse have been identified using this technique. These include the use of insecure algorithms, like the Data Encryption Standard (DES) or Electronic Code Book mode (ECB), and hardcoded passwords that are used for encryption purposes. This thesis has been created in cooperation with Fraunhofer AISEC

The book constitutes the refereed proceedings of the 10th International Conference on Verification, Model Checking, and Abstract Interpretation, VMCAI 2009, held in Savannah, GA, USA, in January 2009 - co-located with POPL 2009, the 36th Annual Symposium on Principles of Programming Languages. The 24 revised full papers presented together with 3 invited talks and 2 invited tutorials were carefully reviewed and selected from 72 submissions. The papers address all current issues from the communities of verification, model checking, and abstract interpretation, facilitating interaction, cross-fertilization, and advancement of hybrid methods that combine the three areas.

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 derivation, analysis, transformation, software security, safety, verification, concurrency, constraints, domain-specific languages, as well as tools for programming, verification, and implementation.

Software similarity and classification is an emerging topic with wide applications. It is applicable to the areas of malware detection, software theft detection, plagiarism detection, and software clone detection. Extracting program features, processing those features into suitable representations, and constructing distance metrics to define similarity and dissimilarity are the key methods to identify software variants, clones, derivatives, and classes of software. Software Similarity and Classification reviews the literature of those core concepts, in addition to relevant literature in each application and demonstrates that considering these applied problems as a similarity and classification problem enables techniques to be shared between areas. Additionally, the authors present in-depth case studies using the software similarity and classification techniques developed throughout the book.

The proceedings set LNCS 12891, LNCS 12892, LNCS 12893, LNCS 12894 and LNCS 12895 constitute the proceedings of the 30th International Conference on Artificial Neural Networks, ICANN 2021, held in Bratislava, Slovakia, in September 2021.* The total of 265 full papers presented in these proceedings was carefully reviewed and selected from 496 submissions, and organized in 5 volumes. In this volume, the papers focus on topics such as adversarial machine learning, anomaly detection, attention and transformers, audio and multimodal applications, bioinformatics and biosignal analysis, capsule networks and cognitive models. *The conference was held online 2021 due to the COVID-19 pandemic.

As human activities moved to the digital domain, so did all the well-known malicious behaviors including fraud, theft, and other trickery. There is no silver bullet, and each security threat calls for a specific answer. One specific threat is that applications accept malformed inputs, and in many cases it is possible to craft inputs that let an intruder take full control over the target computer system. The nature of systems programming languages lies at the heart of the problem. Rather than rewriting decades of well-tested functionality, this book examines ways to live with the (programming) sins of the past while shoring up security in the most efficient manner possible. We explore a range of different options, each making significant progress towards securing legacy programs from malicious inputs. The solutions explored include enforcement-type defenses, which excludes certain program executions because they never arise during normal operation. Another strand explores the idea of presenting adversaries with a moving target that unpredictably changes its attack surface thanks to randomization. We also cover tandem execution ideas where the compromise of one executing clone causes it to diverge from another thus revealing adversarial activities. The main purpose of this book is to provide readers with some of the most influential works on run-time exploits and defenses. We hope that the material in this book will inspire readers and generate new ideas and paradigms.

Design frameworks have become an important infrastructure for building complex design systems. Electronic Design Automation Frameworks presents a state-of-the-art review of the latest research results covering this topic; results which are also of value for other design frameworks. The book contains the selected proceedings of the Fourth International Working Conference on Electronic Design Frameworks, organized by the International Federation for Information Processing and held in Gramado, Brazil, in November 1994.

This book features high-quality research papers presented at the International Conference on Applications and Techniques in Cyber Security and Digital Forensics (ICCSDF 2021), held at The North-Cap University, Gurugram, Haryana, India, during April 3–4, 2021. This book discusses the topics ranging from information security to cryptography, mobile application attacks to digital forensics, and from cyber security to blockchain. The goal of the book is to provide 360-degree view of cybersecurity to the readers which include cyber security issues, threats, vulnerabilities, novel idea, latest technique and technology, and mitigation of threats and attacks along with demonstration of practical applications. This book also highlights the latest development, challenges, methodologies as well as other emerging areas in this field. It brings current understanding of common Web vulnerabilities while maintaining awareness and knowledge of contemporary standards, practices, procedures, and methods of Open Web Application Security Project. It also expounds how to recover information after a cybercrime.

This book constitutes the refereed proceedings of the 11th IAPR-TC-15 International Workshop on Graph-Based Representation in Pattern Recognition, GbRPR 2017, held in Anacapri, Italy, in May 2017. The 25 full papers and 2 abstracts of invited papers presented in this volume were carefully reviewed and selected from 31 submissions. The papers discuss research results and applications in the intersection of pattern recognition, image analysis, graph theory, and also the application of graphs to pattern recognition problems in other fields like computational topology, graphic recognition systems and bioinformatics.

This book presents the most interesting talks given at ISSE 2015 – the forum for the interdisciplinary discussion of the key European Commission security objectives and policy directions. The topics include: · Encrypted Communication · Trust Services, eID and Cloud Security · Industrial Security and Internet of Things · Cybersecurity, Cybercrime, Critical Infrastructures · BYOD and Mobile Security · Regulation and Policies · Biometric Applications Adequate information security is one of the basic requirements of all electronic business processes. It is crucial for effective solutions that the possibilities offered by security technology can be integrated with the commercial requirements of the applications. The reader may expect state-of-the-art: best papers of the Conference ISSE 2015.

This book constitutes the refereed proceedings of the 20th International Conference on Formal Engineering Methods, ICFEM 2018, held in Gold Coast, QLD, Australia, in November 2018. The 22 revised full papers presented together with 14 short papers were carefully reviewed and selected from 66 submissions. The conference focuses on all areas related to formal engineering methods, such as verification; network systems; type theory; theorem proving; logic and semantics; refinement and transition systems; and emerging applications of formal methods.

The book Intelligent Systems and Applications - Proceedings of the 2020 Intelligent Systems Conference is a remarkable collection of chapters covering a wider range of topics in areas of intelligent systems and artificial intelligence and their applications to the real world. The Conference attracted a total of 545 submissions from many academic pioneering researchers, scientists, industrial engineers, students from all around the world. These submissions underwent a double-blind peer review process. Of those 545 submissions, 177 submissions have been selected to be included in these proceedings. As intelligent systems continue to replace and sometimes outperform human intelligence in decision--making processes, they have enabled a larger number of problems to be tackled more effectively.This branching out of computational intelligence in several directions and use of intelligent systems in everyday applications have created the need for such an international conference which serves as a venue to report on up-to-the-minute innovations and developments. This book collects both theory and application based chapters on all aspects of artificial intelligence, from classical to intelligent

scope. We hope that readers find the volume interesting and valuable; it provides the state of the art intelligent methods and techniques for solving real world problems along with a vision of the future research.

The 3-volume set CCIS 1252 until CCIS 1254 constitutes the refereed proceedings of the 6th International Conference on Artificial Intelligence and Security, ICAIS 2020, which was held in Hohhot, China, in July 2020. The conference was formerly called “International Conference on Cloud Computing and Security” with the acronym ICCCS. The total of 178 full papers and 8 short papers presented in this 3-volume proceedings was carefully reviewed and selected from 1064 submissions. The papers were organized in topical sections as follows: Part I: artificial intelligence; Part II: artificial intelligence; Internet of things; information security; Part III: information security; big data and cloud computing; information processing.

This book constitutes the refereed proceedings of the 17th International Conference on Tools and Algorithms for the Construction and Analysis of Systems, TACAS 2011, held in Saarbrücken, Germany, March 26—April 3, 2011, as part of ETAPS 2011, the European Joint Conferences on Theory and Practice of Software. The 32 revised full papers presented were carefully reviewed and selected from 112 submissions. The papers are organized in topical sections on memory models and consistency, invariants and termination, timed and probabilistic systems, interpolations and SAT-solvers, learning, model checking, games and automata, verification, and probabilistic systems.

Edited in collaboration with FoLLI, the Association of Logic, Language and Information this book constitutes the refereed proceedings of the 26th Workshop on Logic, Language, Information and Communication, WoLLIC 2019, held in Utrecht, The Netherlands, in July 2019. The 41 full papers together with 6 invited lectures presented were fully reviewed and selected from 60 submissions. The idea is to have a forum which is large enough in the number of possible interactions between logic and the sciences related to information and computation, and yet is small enough to allow for concrete and useful interaction among participants.

Static analysis is increasingly recognized as a fundamental reasearch area aimed at studying and developing tools for high performance implementations and v- i cation systems for all programming language paradigms. The last two decades have witnessed substantial developments in this eld, ranging from theoretical frameworks to design, implementation, and application of analyzers in optim- ing compilers. Since 1994, SAS has been the annual conference and forum for researchers in all aspects of static analysis. This volume contains the proceedings of the 6th International Symposium on Static Analysis (SAS’99) which was held in Venice, Italy, on 22{24 September 1999. The previous SAS conferences were held in Namur (Belgium), Glasgow (UK), Aachen (Germany), Paris (France), and Pisa (Italy). The program committee selected 18 papers out of 42 submissions on the basis of at least three reviews. The resulting volume o ers to the reader a complete landscape of the research in this area. The papers contribute to the following topics: foundations of static analysis, abstract domain design, and applications of static analysis to di erent programming paradigms (concurrent, synchronous, imperative, object oriented, logical, and functional). In particular, several papers use static analysis for obtaining state space reduction in concurrent systems. New application elds are also addressed, such as the problems of security and secrecy.

Software Engineering for Science provides an in-depth collection of peer-reviewed chapters that describe experiences with applying software engineering practices to the development of scientific software. It provides a better understanding of how software engineering is and should be practiced, and which software engineering practices are effective for scientific software. The book starts with a detailed overview of the Scientific Software Lifecycle, and a general overview of the scientific software development process. It highlights key issues commonly arising during scientific software development, as well as solutions to these problems. The second part of the book provides examples of the use of testing in scientific software development, including key issues and challenges. The chapters then describe solutions and case studies aimed at applying testing to scientific software development efforts. The final part of the book provides examples of applying software engineering techniques to scientific software, including not only computational modeling, but also software for data management and analysis. The authors describe their experiences and lessons learned from developing complex scientific software in different domains. About the Editors Jeffrey Carver is an Associate Professor in the Department of Computer Science at the University of Alabama. He is one of the primary organizers of the workshop series on Software Engineering for Science (http://www.SE4Science.org/workshops). Neil P. Chue Hong is Director of the Software Sustainability Institute at the University of Edinburgh. His research interests include barriers and incentives in research software ecosystems and the role of software as a research object. George K. Thiruvathukal is Professor of Computer Science at Loyola University Chicago and Visiting Faculty at Argonne National Laboratory. His current research is focused on software metrics in open source mathematical and scientific software.