

Model Checking C Programs With Dynamic Memory Allocation

This book constitutes the thoroughly refereed workshop proceedings of the 8th International Workshop on Structured Object-Oriented Formal Language and Method, SOFL+MSVL 2018, held in Gold Coast, QLD, Australia, in November 2018. The 11 revised full papers included in the volume were carefully reviewed and selected from 21 submissions. They are organized in the following topical sections: programming and testing; verification and validation; semantics; and blockchain.

This book constitutes the thoroughly refereed post-conference proceedings of the Second International Workshop on Structured Object-Oriented Formal Language, SOFL 2012, held in Kyoto, Japan, in November 2012. The 10 full papers presented were carefully reviewed and selected for inclusion in this book and address the following topics of interest: testing and tools; tools for specification; model checking; and application and prototyping.

This book constitutes the refereed proceedings of the Third International Workshop on Formal Techniques for Safety-Critical Systems, FTSCS 2014, held in Luxembourg, in November 2014. The 14 revised full papers presented together with two invited talks were carefully reviewed and selected from 40 submissions. The papers address various topics related to the application of formal and semi-formal methods to improve the quality of safety-critical computer systems.

Get Free Model Checking C Programs With Dynamic Memory Allocation

Model checking is a computer-assisted method for the analysis of dynamical systems that can be modeled by state-transition systems. Drawing from research traditions in mathematical logic, programming languages, hardware design, and theoretical computer science, model checking is now widely used for the verification of hardware and software in industry. The editors and authors of this handbook are among the world's leading researchers in this domain, and the 32 contributed chapters present a thorough view of the origin, theory, and application of model checking. In particular, the editors classify the advances in this domain and the chapters of the handbook in terms of two recurrent themes that have driven much of the research agenda: the algorithmic challenge, that is, designing model-checking algorithms that scale to real-life problems; and the modeling challenge, that is, extending the formalism beyond Kripke structures and temporal logic. The book will be valuable for researchers and graduate students engaged with the development of formal methods and verification tools.

This book contains the proceedings of VMCAI 2007. It features current research from the communities of verification, program certification, model checking, debugging techniques, abstract interpretation, abstract domains, and advancement of hybrid methods.

This book constitutes the refereed proceedings of the 13th International Conference on Software Engineering and Formal Methods, SEFM 2015, held in York, UK, in September 2015. The 17 full papers presented together with 2 invited and 6 short papers were carefully reviewed

Get Free Model Checking C Programs With Dynamic Memory Allocation

and selected from 96 submissions. The topics of interest included the following aspects of software engineering and formal methods: program verification, testing, certification, formal specification and proof, testing and model checking, planning, modelling, and model transformation.

This book constitutes the proceedings of the 22nd International Conference on Verification, Model Checking, and Abstract Interpretation, VMCAI 2021, which was held virtually during January 17-19, 2021. The conference was planned to take place in Copenhagen, Denmark, but changed to an online event due to the COVID-19 pandemic. The 23 papers presented in this volume were carefully reviewed from 48 submissions. VMCAI provides a forum for researchers working on verification, model checking, and abstract interpretation and facilitates interaction, cross-fertilization, and advancement of hybrid methods that combine these and related areas. The papers presented in this volume were organized in the following topical sections: hyperproperties and infinite-state systems; concurrent and distributed systems; checking; synthesis and repair; applications; and decision procedures.

This book constitutes the refereed proceedings of the 13th International Conference on Verification, Model Checking, and Abstract Interpretation, VMCAI 2012, held in Philadelphia, PA, USA, in January 2012, co-located with the Symposium on Principles of Programming Languages, POPL 2012. The 26 revised full papers presented were carefully reviewed and selected from 70 submissions. The papers cover a wide range of topics

Get Free Model Checking C Programs With Dynamic Memory Allocation

including program verification, model checking, abstract interpretation, static analysis, deductive methods, program certification, debugging techniques, abstract domains, type systems, and optimization.

This book constitutes the refereed proceedings of the 17th International Conference on Verification, Model Checking, and Abstract Interpretation, VMCAI 2016, held in St. Petersburg, FL, USA, in January 2016. The 24 full papers together with 2 invited talks and 1 abstract presented were carefully reviewed and selected from 67 submissions. VMCAI provides topics including: program verification, model checking, abstract interpretation and abstract domains, program synthesis, static analysis, type systems, deductive methods, program certification, debugging techniques, program transformation, optimization, hybrid and cyber-physical systems.

This book constitutes the refereed proceedings of the 14th International Conference on Verification, Model Checking, and Abstract Interpretation, VMCAI 2013, held in Rome, Italy, in January 2013, co-located with the Symposium on Principles of Programming Languages, POPL 2013. The 27 revised full papers presented were carefully reviewed and selected from 72 submissions.

The papers cover a wide range of topics including program verification, model checking, abstract interpretation and abstract domains, program synthesis, static analysis, type system, deductive methods, program certification, debugging techniques, program transformation, optimization, hybrid and cyber-physical systems.

This book constitutes the refereed proceedings of the 7th

Get Free Model Checking C Programs With Dynamic Memory Allocation

International Conference on Tools and Algorithms for the Construction and Analysis of Systems, TACAS 2001.

The 36 revised full papers presented together with an invited contribution were carefully reviewed and selected from a total of 125 submissions. The papers are organized in sections on symbolic verification, infinite state systems - deduction and abstraction, application of model checking techniques, timed and probabilistic systems, hardware - design and verification, software verification, testing - techniques and tools, implementation techniques, semantics and compositional verification, logics and model checking, and ETAPS tool demonstration.

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.

This volume contains the proceedings of the 14th International Conference on Principles and Practice of Constraint Programming (CP 2008) held in Sydney, Australia, September 14–18, 2008. The conference was held in conjunction with the International Conference on

Get Free Model Checking C Programs With Dynamic Memory Allocation

Automated Planning and Scheduling (ICAPS 2008) and the International Conference on Knowledge Representation and Reasoning (KR 2008). Information about the conference can be found at the website <http://www.unimelb.edu.au/cp2008/>. Held annually, the CP conference series is the premier international conference on constraint programming. The conference focuses on all aspects of computing with constraints. The CP conference series is organized by the Association for Constraint Programming (ACP). Information about the conferences in the series can be found on the Web at <http://www.cs.ualberta.ca/~ai/cp/>. Information about ACP can be found at <http://www.a4cp.org/>. CP 2008 included two calls for contributions: a call for research papers, describing novel contributions in the field, and a call for application papers, describing applications of constraint technology. For the first time authors could directly submit short papers for consideration by the committee. The research track received 84 long submissions and 21 short submissions and the application track received 15 long submissions. Each paper received at least three reviews, which the authors had the opportunity to see and to react to, before the papers and their reviews were discussed extensively by the members of the Program Committee.

This book constitutes the refereed proceedings of the 5th International Symposium on NASA Formal Methods, NFM 2013, held in Moffett Field, CA, USA, in May 2013. The 28 revised regular papers presented together with 9 short papers talks were carefully reviewed and selected from 99 submissions. The topics are organized in topical

Get Free Model Checking C Programs With Dynamic Memory Allocation

sections on model checking; applications of formal methods; complex systems; static analysis; symbolic execution; requirements and specifications; probabilistic and statistical analysis; and theorem proving.

Propositional logic has been recognized throughout the centuries as one of the cornerstones of reasoning in philosophy and mathematics. Over time, its formalization into Boolean algebra was accompanied by the recognition that a wide range of combinatorial problems can be expressed as propositional satisfiability (SAT) problems. Because of this dual role, SAT developed into a mature, multi-faceted scientific discipline, and from the earliest days of computing a search was underway to discover how to solve SAT problems in an automated fashion. This book, the Handbook of Satisfiability, is the second, updated and revised edition of the book first published in 2009 under the same name. The handbook aims to capture the full breadth and depth of SAT and to bring together significant progress and advances in automated solving. Topics covered span practical and theoretical research on SAT and its applications and include search algorithms, heuristics, analysis of algorithms, hard instances, randomized formulae, problem encodings, industrial applications, solvers, simplifiers, tools, case studies and empirical results. SAT is interpreted in a broad sense, so as well as propositional satisfiability, there are chapters covering the domain of quantified Boolean formulae (QBF), constraints programming techniques (CSP) for word-level problems and their propositional encoding, and satisfiability modulo theories (SMT). An extensive

Get Free Model Checking C Programs With Dynamic Memory Allocation

bibliography completes each chapter. This second edition of the handbook will be of interest to researchers, graduate students, final-year undergraduates, and practitioners using or contributing to SAT, and will provide both an inspiration and a rich resource for their work. Edmund Clarke, 2007 ACM Turing Award Recipient: "SAT solving is a key technology for 21st century computer science." Donald Knuth, 1974 ACM Turing Award Recipient: "SAT is evidently a killer app, because it is key to the solution of so many other problems." Stephen Cook, 1982 ACM Turing Award Recipient: "The SAT problem is at the core of arguably the most fundamental question in computer science: What makes a problem hard?"

This book constitutes the refereed proceedings of the 27th International Symposium on Model Checking Software, SPIN 2021, held virtually in July 2021. The 3 full papers, 4 tool papers, and 1 case study presented together with 2 invited talks were carefully reviewed and selected from 20 submissions. Topics covered include formal verification techniques for automated analysis of software; formal analysis for modeling languages, such as UML/state charts; formal specification languages, temporal logic, design-by-contract; model checking, automated theorem proving, including SAT and SMT; verifying compilers; abstraction and symbolic execution techniques; and much more.

This book constitutes the refereed proceedings of the 16th International SPIN workshop on Model Checking Software, SPIN 2009, held in Grenoble, France, in June 2009. The 15 revised full papers presented together with 3 tool papers and

Get Free Model Checking C Programs With Dynamic Memory Allocation

4 invited talks were carefully reviewed and selected from 41 submissions. The papers cover theoretical and algorithmic foundations as well as tools for software model checking by addressing theoretical advances and empirical evaluations related to state-space and path exploration techniques, as implemented in software verification tools.

This book constitutes the refereed proceedings of the 26th International Symposium on Model Checking Software, SPIN 2019, held in Beijing, China, in July 2019. The 11 full papers presented and 2 demo-tool papers, were carefully reviewed and selected from 29 submissions. Topics covered include formal verification techniques for automated analysis of software; formal analysis for modeling languages, such as UML/state charts; formal specification languages, temporal logic, design-by-contract; model checking, automated theorem proving, including SAT and SMT; verifying compilers; abstraction and symbolic execution techniques; and much more.

An expanded and updated edition of a comprehensive presentation of the theory and practice of model checking, a technology that automates the analysis of complex systems. Model checking is a verification technology that provides an algorithmic means of determining whether an abstract model—representing, for example, a hardware or software design—satisfies a formal specification expressed as a temporal logic formula. If the specification is not satisfied, the method identifies a counterexample execution that shows the source of the problem. Today, many major hardware and software companies use model checking in practice, for verification of VLSI circuits, communication protocols, software device drivers, real-time embedded systems, and security algorithms. This book offers a comprehensive presentation of the theory and practice of model checking, covering the foundations of the key algorithms in depth. The

Get Free Model Checking C Programs With Dynamic Memory Allocation

field of model checking has grown dramatically since the publication of the first edition in 1999, and this second edition reflects the advances in the field. Reorganized, expanded, and updated, the new edition retains the focus on the foundations of temporal logic model while offering new chapters that cover topics that did not exist in 1999: propositional satisfiability, SAT-based model checking, counterexample-guided abstraction refinement, and software model checking. The book serves as an introduction to the field suitable for classroom use and as an essential guide for researchers.

This book provides comprehensive coverage of verification and debugging techniques for embedded software, which is frequently used in safety critical applications (e.g., automotive), where failures are unacceptable. Since the verification of complex systems needs to encompass the verification of both hardware and embedded software modules, this book focuses on verification and debugging approaches for embedded software with hardware dependencies. Coverage includes the entire flow of design, verification and debugging of embedded software and all key approaches to debugging, dynamic, static, and hybrid verification. This book discusses the current, industrial embedded software verification flow, as well as emerging trends with focus on formal and hybrid verification and debugging approaches.

This book constitutes the refereed proceedings of the 8th International SPIN Workshop held in Toronto, Canada, in May 2001. The SPIN model checker is one of the most powerful and popular systems for the analysis and verification of distributed and concurrent systems. The 13 revised full papers presented together with one invited survey paper and three invited industrial experience reports were carefully reviewed and selected from 26 submissions. Besides

Get Free Model Checking C Programs With Dynamic Memory Allocation

foundational issues of program analysis and formal verification, the papers focus on tools for model checking and practical applications in a variety of fields.

This book constitutes the refereed proceedings of the 5th International Conference on Verification, Model Checking, and Abstract Interpretation, VMCAI 2004, held in Venice, Italy in January 2004. The 22 revised full papers presented together with 4 invited contributions were carefully reviewed and selected from 68 submissions. The papers are organized in topical sections on security, formal methods, model checking, software checking, liveness and completeness, and miscellaneous.

This book constitutes the refereed proceedings of the 18th International Conference on Verification, Model Checking, and Abstract Interpretation, VMCAI 2017, held in Paris, France, in January 2017. The 27 full papers together with 3 invited keynotes presented were carefully reviewed and selected from 60 submissions. VMCAI provides topics including: program verification, model checking, abstract interpretation and abstract domains, program synthesis, static analysis, type systems, deductive methods, program certification, debugging techniques, program transformation, optimization, hybrid and cyber-physical systems.

This book constitutes the refereed proceedings of the 12th Asian Symposium on Programming Languages and Systems, APLAS 2014, held in Singapore, Singapore in November 2014. The 20 regular papers presented together with the abstracts of 3 invited talks were carefully reviewed and selected from 57 submissions. The papers cover a variety of foundational and practical issues in programming languages and systems - ranging from foundational to practical issues. The papers focus on topics such as semantics, logics, foundational theory; design of languages, type systems and foundational calculi; domain-specific languages; compilers,

Get Free Model Checking C Programs With Dynamic Memory Allocation

interpreters, abstract machines; program derivation, synthesis and transformation; program analysis, verification, model-checking; logic, constraint, probabilistic and quantum programming; software security; concurrency and parallelism; as well as tools and environments for programming and implementation.

The two-volume set LNCS 9206 and LNCS 9207 constitutes the refereed proceedings of the 27th International Conference on Computer Aided Verification, CAV 2015, held in San Francisco, CA, USA, in July 2015. The total of 58 full and 11 short papers presented in the proceedings was carefully reviewed and selected from 252 submissions. The papers were organized in topical sections named: model checking and refinements; quantitative reasoning; software analysis; lightning talks; interpolation, IC3/PDR, and Invariants; SMT techniques and applications; HW verification; synthesis; termination; and concurrency.

This book constitutes the refereed proceedings of the 18th International Conference on Computer Aided Verification, CAV 2006, held in Seattle, WA, USA in August 2006 as part of the 4th Federated Logic Conference, FLoC 2006. The 35 revised full papers presented together with 10 tool papers and 4 invited papers were carefully reviewed and selected from 144 submissions addressing all current issues in computer aided verification and model checking - from foundational and methodological issues ranging to the evaluation of major tools and systems. The

Get Free Model Checking C Programs With Dynamic Memory Allocation

papers are organized in topical sections on automata, arithmetic, SAT and bounded model checking, abstraction/refinement, symbolic trajectory evaluation, property specification and verification, time, concurrency, trees, pushdown systems and boolean programs, termination, abstract interpretation, memory consistency, and shape analysis.

Annotation. This book constitutes the refereed proceedings of the 17th International SPIN workshop on Model Checking Software, SPIN 2010, held at the University of Twente, in Enschede, The Netherlands, in September 2010. The 13 revised full papers presented together with 2 tool papers and 3 invited talks were carefully reviewed and selected from 33 submissions. The papers are organized in topical sections on satisfiability modulo theories for model checking, model checking in context (simulation, testing, UML), implementation and performance of model checking, LTL and Bchi automata, extensions to infinite-state systems, and concurrent software.

A comprehensive introduction to the foundations of model checking, a fully automated technique for finding flaws in hardware and software; with extensive examples and both practical and theoretical exercises. Our growing dependence on increasingly complex computer and software systems necessitates the development of

Get Free Model Checking C Programs With Dynamic Memory Allocation

formalisms, techniques, and tools for assessing functional properties of these systems. One such technique that has emerged in the last twenty years is model checking, which systematically (and automatically) checks whether a model of a given system satisfies a desired property such as deadlock freedom, invariants, and request-response properties. This automated technique for verification and debugging has developed into a mature and widely used approach with many applications.

Principles of Model Checking offers a comprehensive introduction to model checking that is not only a text suitable for classroom use but also a valuable reference for researchers and practitioners in the field. The book begins with the basic principles for modeling concurrent and communicating systems, introduces different classes of properties (including safety and liveness), presents the notion of fairness, and provides automata-based algorithms for these properties. It introduces the temporal logics LTL and CTL, compares them, and covers algorithms for verifying these logics, discussing real-time systems as well as systems subject to random phenomena. Separate chapters treat such efficiency-improving techniques as abstraction and symbolic manipulation. The book includes an extensive set of examples (most of which run through several chapters) and a complete set of basic results accompanied by detailed proofs.

Get Free Model Checking C Programs With Dynamic Memory Allocation

Each chapter concludes with a summary, bibliographic notes, and an extensive list of exercises of both practical and theoretical nature. This book constitutes the proceedings of the 21st International Conference on Verification, Model Checking, and Abstract Interpretation, VMCAI 2020. The 21 papers presented in this volume were carefully reviewed from 44 submissions. VMCAI provides a forum for researchers from the communities of verification, model checking, and abstract Interpretation, facilitating interaction, cross-fertilization, and advancement of hybrid methods that combine these and related areas.

Model checking is a branch of software and hardware veri?cation that involves developingalgorithmsfortheautomaticveri?cationofsystems. Originating from mathematical logic, “model checking” stands for the process of determining whether or not a formula of some logic is satis?ed by a model for the logic. Initiated two and a half decades ago, with papers that have gained their authors the 2007 Turing award, this active research area has resulted in rich theory, and the developmentof a number ofwidely used model-checkingtools. Theseinclude Carnegie-Mellon's SMV, Cadence-SMV, and Bell Laboratories' SPIN. Some of the main activities in model checking involve development of expressive spec- cation formalisms, in particular, temporal logics, the modeling of systems, and ?nding e?cient

Get Free Model Checking C Programs With Dynamic Memory Allocation

algorithms for automatically checking that a model of a system satisfies its temporal specification. The success of model checking in the computer-aided verification community has led to a growth of interest in the use of model checking in AI. One common interest between these two fields is verification of autonomous systems. Logics for autonomous systems can express properties that are not commonly used for reactive systems, expressing properties related to the knowledge and belief of components (agents) of the system about other components. New model-checking algorithms, for such specification properties, are challenging and useful for various applications, including online auction mechanisms, which are embedded in various Internet services, and autonomous robots.

The IFIP TC6 WG 6.1 Joint International Conference on Formal Techniques for Networked and Distributed Systems, FORTE 2002, was held this year at Rice University, Houston, Texas, on November 11–14.

This annual conference provides a forum for researchers and practitioners from universities and industry to meet and advance technologies in areas of specification, testing, and verification of distributed systems and communication protocols.

The main topics are: – FDT-based system and protocol engineering. – Semantical foundations. – Extensions of FDTs. – Formal approaches to concurrent/distributed object-oriented systems. –

Get Free Model Checking C Programs With Dynamic Memory Allocation

Real-time and probability aspects. – Performance modeling and analysis. – Quality of service modeling and analysis. – Verification and validation. – Relations between informal and formal specification. – FDT-based protocol implementation. – Software tools and support environments. – FDT application to distributed systems. – Protocol testing, including conformance testing, interoperability testing, and performance testing. – Test generation, selection, and coverage. – Practical experience and case studies. – Corporate strategic and financial consequences of using formal methods. A total of 61 papers were submitted to FORTE 2002, and reviewed by members of the program committee and additional reviewers. The program committee selected 22 regular papers, two tool papers, and two posters for presentation at the conference. The program also included three tutorials and five invited talks.

This book presents the proceedings of the 14th International SPIN workshop on Model Checking Software, held in Berlin, Germany. Fourteen full papers are presented, together with four tool presentation papers and the abstracts of two invited talks. The papers are organized into topical sections covering directed model checking, partial order reduction, program analysis, exploration advances, modeling and case studies, and tool demonstrations. The SPIN workshop is a forum for researchers interested

Get Free Model Checking C Programs With Dynamic Memory Allocation

in the subject of automata-based, explicit-state model checking technologies for the analysis and verification of asynchronous concurrent and distributed systems. The SPIN - del checker (<http://netlib.bell-labs.com/netlib/spin/whatispin.html>), developed by Gerard Holzmann, is one of the best known systems of this kind, and has attracted a large user community. This can likely be attributed to its efficient state exploration algorithms. The fact that SPIN's modeling language, Promela, resembles a programming language has probably also contributed to its success. Traditionally, the SPIN workshops present papers on extensions and uses of SPIN. As an experiment, this year's workshop was broadened to have a slightly wider focus than previous workshops in that papers on software verification were encouraged. Consequently, a small collection of papers describe attempts to analyze and verify programs written in conventional programming languages. Solutions include translations from source code to Promela, as well as specially designed model checkers that accept source code. We believe that this is an interesting research direction for the formal methods community, and that it will result in a new set of challenges and solutions. Of course, abstraction becomes the key solution to deal with very large state spaces. However, we also see potential for integrating model checking with techniques such as static program analysis and testing. Papers on these issues have therefore been included in the proceedings. This book constitutes the refereed proceedings of the 22nd International Symposium on Model Checking

Get Free Model Checking C Programs With Dynamic Memory Allocation

Software, SPIN 2015, held in Stellenbosch, South Africa, in August 2015. The 18 papers presented – 14 regular papers and 4 tool or new idea papers – were carefully reviewed and selected from 27 submissions. They cover the field between theoretical advances and practical considerations and are organized in topical sections such as abstraction, refinement, translation; Büchi automata and hashing; embedded systems; heuristics and benchmarks; SAT/SMT- based approaches; software validation and verification.

This book constitutes the thoroughly refereed post-proceedings of the 11th International Conference on Computer Aided Systems Theory, EUROCAST 2007. Coverage in the 144 revised full papers presented includes formal approaches, computation and simulation in modeling biological systems, intelligent information processing, heuristic problem solving, signal processing architectures, robotics and robotic soccer, cybercars and intelligent vehicles and artificial intelligence components. Since 1995, when the SPIN workshop series was instigated, SPIN workshops have been held on an annual basis in Montréal (1995), New Brunswick (1996), Enschede (1997), Paris (1998), Trento (1999), Toulouse (1999), Stanford (2000), Toronto (2001), Grenoble (2002) and Portland (2003). All but the first SPIN workshop were organized as satellite events of larger conferences, in particular of CAV (1996), TACAS (1997), FORTE/PSTV (1998), FLOC (1999), the World Congress on Formal Methods (1999), FMOODS (2000), ICSE (2001, 2003) and ETAPS (2002). This year again, SPIN was held as a satellite event of ETAPS 2004. The

Get Free Model Checking C Programs With Dynamic Memory Allocation

co-location of SPIN workshops with conferences has proven to be very successful and has helped to disseminate SPIN model checking technology to wider audiences. Since 1999, the proceedings of the SPIN workshops have appeared in Springer-Verlag's Lecture Notes in Computer Science series. The history of successful SPIN workshops is evidence for the maturing of model checking technology, not only in the hardware domain, but increasingly also in the software area. While in earlier years algorithms and tool development around the SPIN model checker were the focus of this workshop series, for several years now the scope has been widened to include more general approaches to software model checking techniques and tools as well as applications. The SPIN workshop has become a forum for all practitioners and researchers interested in model checking based techniques for the validation and analysis of communication protocols and software systems.

This book constitutes the refereed proceedings of the 12th International SPIN workshop on Model Checking Software, SPIN 2005, held in San Francisco, USA in August 2005. The 15 revised full papers presented were carefully reviewed and selected from 45 submissions; in addition there are 4 tool presentation papers selected from 6 submissions. The papers are organized in topical sections on state representation and abstraction, dealing with concurrency, dealing with complex data, checking temporal properties, and checking security and real-time properties.

This volume contains the proceedings of the 11th

Get Free Model Checking C Programs With Dynamic Memory Allocation

International Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI 2010), held in Madrid, Spain, January 17–19, 2010. VMCAI 2010 was the 11th in a series of meetings. Previous meetings were held in Port Jefferson (1997), Pisa (1998), Venice (2002), New York (2003), Venice (2004), Paris (2005), Charleston (2006), Nice (2007), San Francisco (2008), and Savannah (2009). VMCAI centers on state-of-the-art research relevant to analysis of programs and systems and drawn from three research communities: verification, model checking, and abstract interpretation. A goal is to facilitate interaction, cross-fertilization, and the advance of hybrid methods that combine two or all three areas. Topics covered by VMCAI include program verification, program certification, model checking, debugging techniques, abstract interpretation, abstract domains, static analysis, type systems, deductive methods, and optimization. The Program Committee selected 21 papers out of 57 submissions based on anonymous reviews and discussions in an electronic Program Committee meeting. The principal selection criteria were relevance and quality.

This book constitutes the refereed proceedings of the 15th International SPIN workshop on Model Checking Software, SPIN 2008, held in Los Angeles, CA, USA, in August 2008. The 17 revised full papers presented together with 1 tool paper and 4 invited talks were carefully reviewed and selected from 41 submissions. The main focus of the workshop series is software systems, including models and programs. The papers cover theoretical and algorithmic foundations as well as

Get Free Model Checking C Programs With Dynamic Memory Allocation

tools for software model checking and foster interactions and exchanges of ideas with related areas in software engineering, such as static analysis, dynamic analysis, and testing.

This volume contains the proceedings of the second joint PAMP-PROBMIV Workshop, held at the University of Copenhagen, Denmark, July 25–26, 2002 as part of the Federated Logic Conference (FLoC 2002). The PAMP-PROBMIV workshop results from the combination of two workshops: PAMP (Process Algebras and Performance Modeling) and PROBMIV (Probabilistic Methods in Verification). The aim of the joint workshop is to bring together the researchers working across the whole spectrum of techniques for the modeling, specification, analysis, and verification of probabilistic systems.

Probability is widely used in the design and analysis of software and hardware systems, as a means to derive efficient algorithms (e.g. randomization), as a model for unreliable or unpredictable behavior (as in the study of fault-tolerant systems and computer networks), and as a tool to study performance and dependability properties. The topics of the workshop include specification, models, and semantics of probabilistic systems, analysis and verification techniques, probabilistic methods for the verification of non-probabilistic systems, and tools and case studies. The first PAMP workshop was held in Edinburgh in 1993; the following ones were held in Regensburg (1994), Edinburgh (1995), Turin (1996), Enschede (1997), Nice (1998), Zaragoza (1999), and Geneva (2000). The first PROBMIV workshop was held in Indianapolis, Indiana (1998); the next one took place

Get Free Model Checking C Programs With Dynamic Memory Allocation

in Eindhoven (1999). In 2000, PROBMIV was replaced by a Dagstuhl seminar on Probabilistic Methods in Verification.

[Copyright: c139411a23653462a6f3a2aa2aae2b25](https://www.dagstuhl.de/~c139411a23653462a6f3a2aa2aae2b25)