Computer Aided Verification [35th International Conference, CAV 2023, Paris, France, July 17-22, 2023, Proceedings, Part III /

Enea, Constantin.,
editor.
https://orcid.org/0000-0003-2727-8865.
edt.
http://id.loc.gov/vocabulary/relators/edt
Lal, Akash.,
editor.
https://orcid.org/0009-0002-4359-9378.
edt.
http://id.loc.gov/vocabulary/relators/edt

Monografía

The open access proceedings set LNCS 13964, 13965, 13966 constitutes the refereed proceedings of the 35th International Conference on Computer Aided Verification, CAV 2023, which was held in Paris, France, in July 2023

Título: Computer Aided Verification electronic resource] 35th International Conference, CAV 2023, Paris, France, July 17-22, 2023, Proceedings, Part III edited by Constantin Enea, Akash Lal

Edición: 1st ed. 2023

Editorial: Cham Springer Nature Switzerland Imprint: Springer 2023

Descripción física: 1 online resource (XVII, 502 p. 140 illus., 88 illus. in color.)

Mención de serie: Lecture Notes in Computer Science 1611-3349 13966

Mascot-SDS -- References -- Automated Tail Bound Analysis for Probabilistic Recurrence Relations -- 1 Introduction -- 2 Preliminaries -- 2.1 Probabilistic Recurrence Relations -- 3 Exponential Tail Bounds via Markov's Inequality -- 4 An Algorithmic Approach -- 4.1 The Guess Procedure Guess(f,t) -- 4.2 The Check Procedure CheckCond(cf,ct) -- 5 Experimental Results -- 6 Related Work -- References -- Compositional Probabilistic Model Checking with String Diagrams of MDPs -- 1 Introduction -- 2 String Diagrams of MDPs -- 2.1 Outline -- 2.2 Open MDPs -- 2.3 Rightward Open MDPs and Traced Monoidal String Diagrams -- 2.4 TSMC Equations Between roMDPs -- 2.5 Open MDPs and ``Compact Closed'' String Diagrams -- 3 Decomposition Equalities for Open Markov Chains -- 4 Semantic Categories and Solution Functors -- 4.1 Semantic Category for Rightward Open MCs -- 4.2 Semantic Category of Rightward Open MDPs -- 4.3 Semantic Category of MDPs -- 5 Implementation and Experiments -- References -- Efficient Sensitivity Analysis for Parametric Robust Markov Chains -- 1 Introduction -- 2 Overview -- 3 Formal Problem Statement -- 4 Differentiating Solution Functions for pMCs -- 4.1 Computing Derivatives Explicitly -- 4.2 Computing k-Highest Derivatives -- 5 Differentiating Solution Functions for prMCs 5.1 Computing Derivatives via pMCs (and When It Does Not Work) -- 5.2 Computing Derivatives Explicitly -- 5.3 Computing k-Highest Derivatives -- 6 Numerical Experiments -- 7 Related Work -- 8 Concluding Remarks -- References -- MDPs as Distribution Transformers: Affine Invariant Synthesis for Safety Objectives -- 1 Introduction -- 1.1 Related Work -- 2 Preliminaries -- 2.1 Markov Systems -- 2.2 MDPs as Distribution Transformers -- 3 Problem Statement and Examples -- 4 Proving Safety by Invariants -- 4.1 Distribution Strategies -- 4.2 Distributional Invariants for MDP Safety -- 5 Algorithms for Distributional Invariant Synthesis -- 5.1 Synthesis of Affine Invariants and Memoryless Strategies -- 5.2 Synthesis of Affine Invariants and General Strategies -- 6 Discussion, Extensions, and Variants -- 7 Implementation and Evaluation -- 8 Conclusion -- References -- Search and Explore: Symbiotic Policy Synthesis in POMDPs -- 1 Introduction -- 2 Motivating Examples -- 3 Preliminaries and Problem Statement -- 4 FSCs for and from Belief Exploration -- 4.1 Belief Exploration with Explicit FSC Construction -- 4.2 Using FSCs for Cut-Off Values -- 4.3 Extracting FSC from Belief Exploration -- 5 Accelerated Inductive Synthesis -- 5.1 Inductive Synthesis with k-FSCs -- 5.2 Using Reference Policies to Accelerate Inductive Synthesis -- 5.3 Inductive Synthesis with Adequate FSCs -- 6 Integrating Belief Exploration with Inductive Synthesis -- 7 Experiments -- 8 Conclusion and Future Work -- References -- Security andQuantum Systems -- AutoQ: An Automata-Based Quantum Circuit Verifier -- 1 Introduction -- 2 Tree Automata-Based Verification of Quantum Circuits -- 2.1 High-Level Specification Language -- 2.2 Complex Number Representation -- 2.3 Precise Semantics of the Specification -- 3 Entailment Checking -- 4 Architecture -- 5 Use Cases 5.1 Hadamard Square is Identity -- 5.2 Zero Imaginary Part of Amplitudes -- 5.3 Probability of Measuring the Correct Answer -- 5.4 Increasing Amplitude of the Correct Answer -- 6 Conclusion -- References -- Bounded Verification for Finite-Field-Blasting -- 1 Introduction -- 1.1 Related Work -- 2 Background -- 2.1 Logic -- 2.2 Zero Knowledge Proofs -- 2.3 Compilation Targeting Zero Knowledge Proofs -- 3 Overview and Example -- 3.1 An Example of Field-Blasting -- 3.2 Key Ideas -- 4 Architecture -- 4.1 Encodings -- 4.2 Encoding Rules -- 4.3 Calculus -- 5 Verification Conditions -- 5.1 Correctness Definition -- 5.2 Rule VCs -- 5.3 A Correct Field-Blasting Calculus -- 6 Case Study: A Verifiable Field-Blaster for CirC -- 6.1 Verification Evaluation -- 6.2 Performance and Output Quality Evaluation -- 7 Discussion -- A Zero-Knowledge Proofs and Compilers -- B Compiler Correctness Proofs -- C CirC-IR -- D Optimizations to the CirC Field-Blaster -- E Verified Field-Blaster Performance Details -- F Verifier Performance Details -- G Bugs Found in the CirC Field Blaster -- References -- Formally Verified EVM Block-Optimizations -- 1 Introduction -- 2 Background -- 3 EVM Semantics in Coq -- 4 Formal Verification of EVM-Optimizations in Coq -- 4.1 EVM Symbolic Execution in Coq -- 4.2 Simplification Rules -- 4.3 Stacks Equivalence Modulo Commutativity -- 5 Implementation and Experimental Evaluation -- 6 Conclusions, Related and Future Work -- References -- SR-SFLL: Structurally Robust Stripped Functionality Logic Locking -- 1 Introduction -- 2 Background -- 2.1 Stripped Functionality Logic Locking (SFLL) -- 2.2 SFLL Attacks -- 2.3 Analysis of the Structural Attacks on SFLL -- 3 Overview -- 3.1 Preliminaries -- 3.2 Approach -- 4 SR-SFLL -- 4.1 Problem Statement -- 4.2 Intuition: SR-SFLL -- 4.3 Methodology: SR-SFLL -- 5 SyntAk -- 6 Evaluation 6.1 Robustness of SR-SELL(0) and SR-SELL on Existing Attacks -- 6.2 Robustness of SR-SELL(0) and SR-SELL on SyntAk -- 6.3 Overhead of SR-SELL(0) and SR-SELL -- 7 Related Work -- 8 Conclusions -- References -- Symbolic Quantum Simulation with Quasimodo -- 1 Introduction -- 2 Background on Quantum Simulation -- 3 Quasimodo's Programming and Analysis Interface -- 3.1 Extending Quasimodo -- 4 The Internals of Quasimodo -- 5 Experiments -- 6 Conclusion -- References -- Verifying the Verifier: eBPF Range Analysis Verification -- 1 Introduction -- 2 Background on Abstract Interpretation -- 3 Abstract Interpretation in the Linux Kernel -- 4 Automatic Verification of the Kernel's Algorithms -- 4.1 Soundness Specification for Abstraction/Reduction

Operators -- 4.2 Refining Soundness Specification with Input Preconditioning -- 4.3 Automatically Producing Programs Exercising Soundness Bugs -- 5 C to Logic for Kernel's Abstract Operators -- 6 Experimental Evaluation -- 7 Limitations and Caveats -- 8 Related Work -- 9 Conclusion -- References -- Software Verification -- Automated Verification of Correctness for Masked Arithmetic Programs -- 1 Introduction -- 2 Preliminaries -- 3 The Core Language -- 4 Overview of the Approach -- 4.1 Our Approach -- 5 Term Rewriting System -- 6 Algorithmic Verification -- 6.1 Term Normalization Algorithm -- 6.2 Computing Affine Constants -- 6.3 Verification Algorithm -- 6.4 Implementation Remarks -- 7 Evaluation -- 7.1 Evaluation for Computing Affine Constants -- 7.2 Evaluation for Correctness Verification -- 7.3 Scalability of FISCHER -- 7.4 Evaluation for More Boolean Masking Schemes -- 7.5 Evaluation for Arithmetic/Boolean Masking Conversions -- 8 Conclusion -- References -- Automatic Program Instrumentation for Automatic Verification -- 1 Introduction -- 2 Instrumentation Framework -- 2.1 The Core Language 2.2 Instrumentation Operators -- 2.3 Instrumentation Correctness -- 3 Instrumentation Application Strategies -- 4 Instrumentation Operators for Arrays -- 4.1 Instrumentation Operators for Quantification over Arrays -- 4.2 Instrumentation Operators for Aggregation over Arrays -- 5 Evaluation -- 5.1 Implementation -- 5.2 Experiments and Comparisons -- 6 Related Work -- 7 Conclusion -- References -- Boolean Abstractions for Realizability Modulo Theories -- 1 Introduction -- 2 Preliminaries -- 3 Boolean Abstraction -- 3.1 Notation -- 3.2 The Boolean Abstraction Algorithm -- 3.3 From Local Simulation to Equi-Realizability -- 4 Efficient Algorithms for Boolean Abstraction -- 4.1 Quasi-reactions -- 4.2 Quasi-reaction-based Optimizations -- 4.3 A Single Model-Loop Algorithm (Algorithm 2) -- 4.4 A Nested-SAT Algorithm (Algorithm 3) -- 5 Empirical Evaluation -- 6 Related Work and Conclusions -- References -- Certified Verification for Algebraic Abstraction -- 1 Introduction -- 2 Preliminaries -- 3 ToyLang -- 3.1 Syntax and Semantics -- 4 Algebraic Abstraction -- 4.1 Soundness Conditions -- 4.2 Polynomial Program Verification -- 5 Certified Verification -- 5.1 Verified Abstraction Algorithm -- 5.2 Verification through Certification -- 5.3 Optimization -- 6 Evaluation -- 6.1 Field and Group Operation in Elliptic Curves -- 6.2 Number-Theoretic Transform in Kyber -- 7 Conclusion -- References -- Complete Multiparty Session Type Projection with Automata -- 1 Introduction -- 2 Motivation and Overview -- 3 Preliminaries -- 4 Synthesizing Implementations -- 5 Checking Implementability -- 6 Soundness -- 7 Completeness -- 8 Complexity -- 9 Evaluation -- 10 Discussion -- 11 Related Work -- References -- Early Verification of Legal Compliance via Bounded Satisfiability Checking -- 1 Introduction -- 2 Preliminaries -- 3 Bounded Satisfiability Checking Problem 4 Checking Bounded Satisfiability

Restricciones de acceso: Open Access

ISBN: 3-031-37709-5

Materia: Software engineering Artificial intelligence Algorithms Computer engineering Computer networks Software Engineering Artificial Intelligence Design and Analysis of Algorithms Computer Engineering and Networks

Autores: Enea, Constantin., editor. https://orcid.org/0000-0003-2727-8865. edt. http://id.loc.gov/vocabulary/relators/edt Lal, Akash., editor. https://orcid.org/0009-0002-4359-9378. edt. http://id.loc.gov/vocabulary/relators/edt

Enlace a formato físico adicional: 3-031-37708-7

Punto acceso adicional serie-Título: Lecture Notes in Computer Science 1611-3349 13966
