1.

| | |
|---|---|
| Record Nr. | UNINA9910484603403321 |
| Titolo | Automated Technology for Verification and Analysis : 13th International Symposium, ATVA 2015, Shanghai, China, October 12-15, 2015, Proceedings / / edited by Bernd Finkbeiner, Geguang Pu, Lijun Zhang |
| Pubbl/distr/stampa | Cham : , : Springer International Publishing : , : Imprint : Springer, , 2015 |
| ISBN | 3-319-24953-3 |
| Edizione | [1st ed. 2015.] |
| Descrizione fisica | 1 online resource (XIII, 520 p. 96 illus.) |
| Collana | Programming and Software Engineering ; ; 9364 |
| Disciplina | 005.13 |
| Soggetti | Programming languages (Electronic computers)<br>Computer logic<br>Mathematical logic<br>Artificial intelligence<br>Computer programming<br>Programming Languages, Compilers, Interpreters<br>Logics and Meanings of Programs<br>Mathematical Logic and Formal Languages<br>Artificial Intelligence<br>Programming Techniques |
| Lingua di pubblicazione | Inglese |
| Formato | Materiale a stampa |
| Livello bibliografico | Monografia |
| Note generali | Bibliographic Level Mode of Issuance: Monograph |
| Nota di contenuto | Intro -- Preface -- Organization -- Contents -- Probabilistic Programming: A True Verification Challenge -- References -- Machines Reasoning About Machines: 2015 -- 1 Introduction -- 2 A Quick History and Acknowledgments -- 3 1970s -- Simple List Processing -- 4 1980s -- Academic Math and Computer Science -- 5 1990s -- Commercial Breakthrough -- 6 2000s -- Gradual Acceptance -- 7 2010 -- Integrated with the Design Process -- 8 Lessons -- 9 Conclusion -- References -- Using SMT for Solving Fragments of Parameterised Boolean Equation Systems -- 1 Introduction -- 2 Preliminaries -- 3 A Motivating Example -- 4 Disjunctive and Conjunctive PBESs -- 5 Solving Disjunctive and Conjunctive PBESs Using SMT -- 5.1 Finding Lasso-Shaped Proof Graphs -- 5.2 Proving the |

Absence of Proof Graphs -- 5.3 A Pseudo-Decision Procedure -- 6 Experiments -- 7 Closing Remarks -- References -- Unfolding-Based Process Discovery -- 1 Introduction -- 2 Preliminaries -- 3 Independence-Preserving Discovery -- 4 Introducing Generalization -- 4.1 Language-Preserving Generalization -- 4.2 Controlling Generalization via Negative Information -- 5 Computing Folding Equivalences -- 5.1 SMT Encoding -- 5.2 Finding an Optimal Folding Equivalence -- 6 Experiments -- 7 Related Work -- 8 Conclusions -- References -- Improving Interpolants for Linear Arithmetic -- 1 Introduction -- 2 Preliminaries -- 2.1 Notations -- 2.2 Computing Interpolants by Resolution Proofs -- 2.3 Computing Interpolants by DC-Removability Checks -- 2.4 Description of a State Set -- 3 The General Algorithm -- 3.1 Test Whether One Additional Linear Constraint Is Sufficient -- 3.2 Finding Pairs of Indistinguishable Points with SMT -- 3.3 Enlarge Pairs of Indistinguishable Points to Convex Regions -- 3.4 Finding a Linear Constraint Separating Pairs of Convex Regions with LP -- 4 Optimizations -- 5 Experimental Results.

6 Conclusion and Further Research -- References -- A Mechanically Checked Generation of Correlating Programs Directed by Structured Syntactic Differences -- 1 Introduction -- 2 Overview -- 3 Source Language and Its Guarded Form -- 3.1 Input Language -- 3.2 Guarded Language -- 3.3 Translation to Guarded Form -- 4 Structured Differences Between Programs -- 5 Generation Algorithm Directed by Structured Differences -- 6 Implementation and Experiments -- 7 Related Work -- 8 Conclusion -- References -- On Automated Lemma Generation for Separation Logic with Inductive Definitions -- 1 Introduction -- 2 Motivating Example -- 3 Separation Logic with Inductive Definitions -- 4 Composition Lemmas -- 5 Derived Lemmas -- 5.1 The ``Completion'' Lemmas -- 5.2 The ``Stronger'' Lemmas -- 6 A Proof Strategy Based on Lemmas -- 7 Experimental Results -- 8 Related Work -- 9 Conclusion -- References -- Severity Levels of Inconsistent Code -- 1 Introduction -- 2 Overview -- 3 Inconsistent Code -- 4 Severity Levels of Inconsistency -- 5 Algorithm for Categorizing Inconsistent Code -- 6 Evaluation -- 7 Related Work -- 8 Conclusion -- References -- Learning the Language of Error -- 1 Introduction -- 2 Preliminaries -- the L* Algorithm -- 3 The Language We Learn -- 4 L* and the Queries -- 4.1 Membership Queries -- 4.2 Conjecture Queries -- 5 Usage Scenarios for the Learning Framework -- 6 System Description and Empirical Evaluation -- 6.1 Optimisations -- 6.2 Implementation and Evaluation -- 7 Conclusions and Future Work -- References -- Explicit Model Checking of Very Large MDP Using Partitioning and Secondary Storage -- 1 Introduction -- 2 Preliminaries -- 3 Disk-Based State Space Exploration with Partitioning -- 3.1 Representation of MDP in Memory and on Disk -- 3.2 Disk-Based Exploration Using Partitioning -- 4 Disk-Based Partitioned Value Iteration -- 5 Evaluation.

6 Conclusion -- References -- Model Checking Failure-Prone Open Systems Using Probabilistic Automata -- 1 Introduction -- 2 Preliminaries -- 2.1 Hierarchical Probabilistic Automata -- 3 Decision Algorithms and Robustness -- 3.1 Forward Algorithm -- 3.2 Backward Algorithm -- 3.3 Forward vs. Backward -- 3.4 Robustness -- 4 Undecidability for 2-HPA -- 5 Modeling Failure-Prone Open Systems -- 6 Implementation and Experiment -- 6.1 Implementation of the Verification Algorithms -- 6.2 Abstracting Models of Web Applications -- 6.3 Experiment -- 7 Conclusions -- References -- Optimal Continuous Time Markov Decisions -- 1 Introduction -- 2 Preliminaries -- 3 Unif+: Optimal Time-Bounded Reachability Revisited -- 3.1 Uniformisation to C -- 3.2 Lower and Upper Bounds on the Value of C

-- 3.3 Convergence of the Bounds for Increasing -- 3.4 Extracting the Scheduler -- 4 Existing Algorithms -- 5 Empirical Evaluation and Comparison -- 6 Conclusion -- References -- Spanning the Spectrum from Safety to Liveness -- 1 Introduction -- 2 Preliminaries -- 3 Classes of Safety -- 3.1 Co-safety -- 3.2 Finding the Safety Level -- 4 Finding the Safety Class -- 4.1 Observations on Automata -- 4.2 Expressive Power -- 4.3 Decision Procedures -- 5 Discussion and Directions for Future Research -- References -- Marimba: A Tool for Verifying Properties of Hidden Markov Models -- 1 Introduction -- 2 Tool Architecture and Implementation -- 2.1 Marimba's Input and Modules -- 3 Verification of a Human-Robot Interaction -- 4 Conclusions -- References -- ParaVerifier: An Automatic Framework for Proving Parameterized Cache Coherence Protocols -- 1 Introduction -- 2 Consistency Lemma -- 3 Overview of Our Approach -- 4 Validation and Conclusion -- References -- ASSA-PBN: An Approximate Steady-State Analyser of Probabilistic Boolean Networks -- 1 Introduction -- 2 Architecture and Usage.

3 Comparison, Evaluation and Future Developments -- References -- EviCheck: Digital Evidence for Android -- 1 Introduction and Related Work -- 2 EviCheck's Main Ingredients -- 2.1 Policy Language -- 2.2 The Verifier -- 2.3 The Checker -- 2.4 Policy Generator -- 2.5 Technical Discussion -- 3 Implementation and Experiments -- References -- Lazy-CSeq-SP: Boosting Sequentialization-Based Verification of Multi-threaded C Programs via Symbolic Pruning of Redundant Schedules -- 1 Introduction -- 2 Optimized Scheduler Encoding -- 3 Implementation Details -- 4 Experimental Evaluation and Conclusion -- References -- A Contextual Equivalence Checker for IMJ* -- 1 Introduction -- 2 Tool Architecture -- 3 Evaluation -- References -- Trace Diagnostics Using Temporal Implicants -- 1 Introduction -- 2 Propositional Foundations -- 2.1 Problem Statement -- 2.2 Syntactic and Semantic Formulations -- 2.3 Practical Solution -- 3 Temporal Issues -- 3.1 Syntactic Rewritings -- 3.2 Semantic Restrictions -- 3.3 Sub-models of a Formula -- 3.4 Temporal Implicants -- 4 MTL Diagnostics -- 4.1 Non-Standard MTL Semantics -- 4.2 Explanation Operators -- 4.3 Computation of Selection Functions -- 4.4 Discussion -- 5 LTL Ultimately-Periodic Diagnostics -- 5.1 Recurrent LTL Semantics -- 5.2 Explanation Operators -- 6 Conclusion and Perspectives -- References -- Test Case Generation of Actor Systems -- 1 Introduction -- 2 The Language -- 3 Symbolic Execution -- 4 Less Redundant Exploration in Symbolic Execution -- 5 Generation of Test Cases -- 5.1 Coverage and Termination Criteria for Actor Systems -- 5.2 Test Cases for Actor Systems -- 6 Implementation and Experimental Evaluation -- 7 Related Work and Conclusions -- References -- Lattice-Based Semantics for Combinatorial Model Evolution -- 1 Introduction -- 2 Running Example and Overview -- 3 Preliminaries.

4 Lattice-Based Semantics for Combinatorial Model Evolution -- 5 Atomic Operations on Combinatorial Models -- 5.1 Adding or Removing a Parameter and Its Values -- 5.2 Adding or Removing a Value from an Existing Parameter -- 5.3 Adding, Removing, or Changing a Constraint -- 6 Split and Merge Operations on Combinatorial Models -- 7 Related Work -- 8 Summary and Future Work -- References -- Effective Verification of Replicated Data Types Using Later Appearance Records (LAR) -- 1 Introduction -- 2 CRDTs, Traces and Specifications -- 3 CRDT Implementation -- 3.1 Reference Implementation -- 3.2 Details of the Reference Implementation -- 3.3 Correctness of the Reference Implementation -- 3.4 Bounding the Reference Implementation -- 4 Effective Verification Using Bounded

Reference Implementation via CEGAR -- 5 Conclusion -- References -- TSO-to-TSO Linearizability Is Undecidable -- 1 Introduction -- 2 TSO Concurrent Systems -- 2.1 Notations -- 2.2 Libraries and the Most General Clients -- 2.3 TSO Operational Semantics -- 3 TSO-to-TSO Linearizability and Equivalent Characterization -- 3.1 TSO-to-TSO Linearizability -- 3.2 Equivalence Characterization -- 4 Undecidability of TSO-to-TSO Linearizability -- 4.1 Classic-Lossy Single-Channel Systems -- 4.2 Simulation on the TSO Memory Model -- 4.3 Undecidability of History Inclusion -- 4.4 Undecidability of TSO-to-TSO Linearizability -- 5 Conclusion and Future Work -- References -- Formal Verification of Infinite-State BIP Models -- 1 Introduction -- 2 The BIP Model -- 3 ESST for BIP (ESST BIP ) -- 3.1 The ESST Framework -- 3.2 Instantiation of ESST for BIP -- 3.3 Optimizations -- 4 Encoding BIP into Transition System -- 5 Related Work -- 6 Experimental Evaluation -- 7 Conclusions and Future Work -- References -- PBMC: Symbolic Slicing for the Verification of Concurrent Programs -- 1 Introduction.
2 Motivating Example.

Sommario/riassunto

This book constitutes the proceedings of the 13th International Symposium on Automated Technology for Verification and Analysis, ATVA 2015, held in Shanghai, China, in October  2015. The 27 revised papers presented together with 6 tool papers in this volume were carefully reviewed and selected from 95 submissions. They show current research on theoretical and practical aspects of automated analysis, verification and synthesis by providing an international forum for interaction among the researchers in academia and industry.