1. Record Nr. UNINA9910734886603321

Titolo	Computer Aided Verification : 35th International Conference, CAV 2023, Paris, France, July 17–22, 2023, Proceedings, Part I / / edited by Constantin Enea, Akash Lal
Pubbl/distr/stampa	Cham : , : Springer Nature Switzerland : , : Imprint : Springer, , 2023
ISBN	3-031-37706-0
Edizione	[1st ed. 2023.]
Descrizione fisica	1 online resource (XXXI, 488 p. 160 illus., 121 illus. in color.)
Collana	Lecture Notes in Computer Science, , 1611-3349 ; ; 13964
Disciplina	005.1
Soggetti	Software engineering Artificial intelligence Algorithms Computer engineering Computer networks Software Engineering Artificial Intelligence Design and Analysis of Algorithms Computer Engineering and Networks
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Nota di contenuto	Intro -- Preface -- Organization -- Invited Talks -- Privacy-Preserving Automated Reasoning -- Enhancing Programming Experiences Using AI: Leveraging LLMs as Analogical Reasoning Engines and Beyond -- Verified Software Security Down to Gates -- Contents - Part I -- Contents - Part II -- Contents - Part III -- Automata and Logic -- Active Learning of Deterministic Timed Automata with Myhill-Nerode Style Characterization -- 1 Introduction -- 2 Preliminaries -- 2.1 Timed Words and Timed Automata -- 2.2 Recognizable Timed Languages -- 2.3 Distinguishing Extensions and Active DFA Learning -- 3 A Myhill-Nerode Style Characterization of Recognizable Timed Languages with Elementary Languages -- 4 Active Learning of Deterministic Timed Automata -- 4.1 Successors of Simple Elementary Languages -- 4.2 Timed Observation Table for Active DTA Learning -- 4.3 Counterexample Analysis -- 4.4 L*-Style Learning Algorithm for

DTAs -- 4.5 Learning with a Normal Teacher -- 4.6 Complexity Analysis -- 5 Experiments -- 5.1 RQ1: Scalability with Respect to the Language Complexity -- 5.2 RQ2: Performance on Practical Benchmarks -- 6 Conclusions and Future Work -- References -- Automated Analyses of IOT Event Monitoring Systems -- 1 Introduction -- 2 Overview -- 3 Technique -- 3.1 Well-formedness Properties -- 4 Experiments -- 5 Conclusion -- A Common Issues with Detector Models -- References -- Learning Assumptions for Compositional Verification of Timed Automata -- 1 Introduction -- 2 Preliminaries -- 2.1 Timed Automata -- 2.2 Learning Deterministic One-Clock Timed Automata -- 3 Framework for Learning-Based Compositional Verification of Timed Automata -- 3.1 Verification Framework via Assumption Learning -- 3.2 Model Conversion -- 3.3 Membership Queries -- 3.4 Candidate Queries -- 3.5 Correctness and Termination -- 4 Optimization Methods.
4.1 Using Additional Information -- 4.2 Minimizing the Alphabet of the Assumption -- 5 Experimental Results -- 6 Conclusion -- References -- Online Causation Monitoring of Signal Temporal Logic -- 1 Introduction -- 2 Preliminaries -- 2.1 Signal Temporal Logic -- 2.2 Classic Online Monitoring of STL -- 3 Boolean Causation Online Monitor -- 4 Quantitative Causation Online Monitor -- 5 Experimental Evaluation -- 5.1 Experiment Setting -- 5.2 Evaluation -- 6 Related Work -- 7 Conclusion and Future Work -- References -- Process Equivalence Problems as Energy Games -- 1 Introduction -- 2 Distinctions and Equivalences in Transition Systems -- 2.1 Transition Systems and Hennessy-Milner Logic -- 2.2 Price Spectra of Behavioral Equivalences -- 3 An Energy Game of Distinguishing Capabilities -- 3.1 Energy Games -- 3.2 The Spectroscopy Energy Game -- 3.3 Correctness: Tight Distinctions -- 3.4 Becoming More Clever by Looking One Step Ahead -- 4 Computing Equivalences -- 4.1 Computation of Attacker Winning Budgets -- 4.2 Complexity and How to Flatten It -- 4.3 Equivalences and Distinguishing Formulas from Budgets -- 5 Exploring Minimizations -- 6 Conclusion and Related Work -- References -- Concurrency -- Commutativity for Concurrent Program Termination Proofs -- 1 Introduction -- 2 Preliminaries -- 2.1 Concurrent Programs -- 2.2 Termination -- 2.3 Commutativity and Traces -- 3 Closures and Reductions -- 3.1 The Compromise: A New Proof Rule -- 3.2 Omega Prefix Generalization -- 4 Finite-Word Reductions -- 4.1 Efficient Reduction to Safety -- 4.2 Sound Finite Word Reductions -- 5 Omega Regular Reductions -- 6 Experimental Results -- 7 Related Work -- 8 Conclusion -- References -- Fast Termination and Workflow Nets -- 1 Introduction -- 2 Preliminaries -- 2.1 (Integer) Linear Programs -- 2.2 Petri Nets -- 2.3 Workflow Nets -- 2.4 Termination Complexity.
3 A Dichotomy of Termination Time in Workflow Nets -- 4 Refining Termination Time -- 5 Soundness in Terminating Workflow Nets -- 6 Termination Time and Concurrent Semantics -- 7 Experimental Evaluation -- 7.1 Benchmark Suite -- 7.2 Termination and Deadlocks -- 7.3 aN, MinTimeN(1) and MaxTimeN(1) -- 7.4 1-Soundness -- References -- Lincheck: A Practical Framework for Testing Concurrent Data Structures on JVM -- 1 Introduction -- 2 Lincheck Overview -- 2.1 Phase 1: Scenario Generation -- 2.2 Phase 2: Scenario Running -- 2.3 Phase 3: Verification of Outcome Results -- 3 Evaluation -- 4 Related Work -- 5 Discussion -- References -- nekton: A Linearizability Proof Checker -- 1 Introduction -- 2 Input -- 2.1 Proof Outlines -- 3 Case Study -- 4 Correctness and Implementation -- References -- Overcoming Memory Weakness with Unified Fairness -- 1 Introduction -- 2 Modelling Concurrent Programs -- 2.1 Labelled Transition

Systems -- 2.2 Concurrent Programs -- 2.3 Concurrent Programs as Labelled Transition Systems -- 3 A Unified Framework for Weak Memory Models -- 3.1 Message Structures -- 3.2 Ensuring Consistency of Executions -- 3.3 Instantiating the Framework -- 4 Fairness Properties -- 4.1 Transition and Memory Fairness -- 4.2 Probabilistic Memory Fairness -- 4.3 Relating Fairness Notions -- 5 Applying Fairness Properties to Decision Problems -- 5.1 Deciding Repeated Control State Reachability -- 5.2 Quantitative Control State Repeated Reachability -- 5.3 Adapting Subroutines to Our Memory Framework -- 6 Related Work -- 7 Conclusion, Future Work, and Perspective -- References -- Rely-Guarantee Reasoning for Causally Consistent Shared Memory -- 1 Introduction -- 2 Motivating Example -- 3 Preliminaries: Syntax and Semantics -- 4 Generic Rely-Guarantee Reasoning -- 5 Potential-Based Memory System for SRA -- 6 Program Logic -- 7 Examples.
8 Discussion, Related and Future Work -- References -- Unblocking Dynamic Partial Order Reduction -- 1 Introduction -- 2 DPOR and Blocked Executions -- 2.1 Dynamic Partial Order Reduction -- 2.2 Assume Statements and DPOR -- 3 Key Ideas -- 3.1 Avoiding Blocking Due to Stale Reads -- 3.2 Handling Await Loops with In-Place Revisits -- 3.3 Handling Confirmation CASes with Speculative Revisits -- 4 Await-Aware Model Checking Algorithm -- 4.1 Execution Graphs -- 4.2 Awamoche -- 5 Correctness and Optimality -- 5.1 Approaches to Correctness -- 5.2 Awamoche: Completeness, Optimality, and Strong Optimality -- 6 Evaluation -- 6.1 Results -- 7 Related Work -- 8 Conclusion -- References -- Cyber-Physical and Hybrid Systems -- 3D Environment Modeling for Falsification and Beyond with Scenic 3.0 -- 1 Introduction -- 2 New Features -- 2.1 3D Geometry -- 2.2 Mesh Shapes and Regions -- 2.3 Precise Visibility Model -- 2.4 Temporal Requirements -- 2.5 Rewritten Parser -- 3 Case Studies -- 3.1 Falsification of a Robot Vacuum -- 3.2 Constrained Data Generation for an Autonomous Vehicle -- 4 Conclusion -- References -- A Unified Model for Real-Time Systems: Symbolic Techniques and Implementation -- 1 Introduction -- 2 Generalized timed automata -- 3 Expressivity of GTA and Examples -- 4 The Reachability Problem for GTA -- 5 Symbolic Enumeration -- 6 Computing with GTA Zones Using Distance Graphs -- 7 Finiteness of the Simulation Relation -- 8 Experimental Evaluation -- 9 Conclusion -- References -- Closed-Loop Analysis of Vision-Based Autonomous Systems: A Case Study -- 1 Introduction -- 2 Autonomous Center-Line Tracking with TaxiNet -- 3 Probabilistic Analysis -- 3.1 Probabilistic Abstractions for Perception -- 3.2 DNN Checks as Run-Time Guards -- 3.3 Confidence Analysis -- 4 Experiments -- 5 Conclusion -- References.
Hybrid Controller Synthesis for Nonlinear Systems Subject to Reach-Avoid Constraints -- 1 Introduction -- 1.1 Related Works -- 2 Preliminaries -- 3 Hybrid Polynomial-DNN Controllers Training -- 3.1 Training Well-Performing DNN Controllers Using RL -- 3.2 Polynomial Approximation -- 3.3 Training the Residual Controller -- 4 Reach-Avoid Verification with Lyapunov-Like Functions and Barrier Certificates Generation -- 4.1 Constructing Polynomial Simulations of the Controller Network -- 4.2 Producing Barrier Certificate and Lyapunov-Like Function -- 5 Experiments -- 6 Conclusion -- References -- Safe Environmental Envelopes of Discrete Systems -- 1 Introduction -- 2 Motivating Example -- 3 Modeling Formalism -- 4 Robustness Against Environmental Deviations -- 4.1 Deviations -- 4.2 Comparing Deviations -- 4.3 Robustness -- 4.4 Problem Statement -- 4.5 Comparing Robustness -- 5 Computing Robustness -- 5.1 Brute-Force Algorithm -- 5.2 Controlling the Deviations Without Environmental

Constraints -- 5.3 Controlling the Deviations with Environmental Constraints -- 6 Case Studies -- 6.1 Implementation -- 6.2 Therac-25 -- 6.3 Voting -- 6.4 Oyster -- 6.5 PCA Pump -- 6.6 Results and Discussion -- 7 Related Work -- 8 Conclusion -- References -- Verse: A Python Library for Reasoning About Multi-agent Hybrid System Scenarios -- 1 Introduction -- 2 Overview of Verse -- 3 Scenarios in Verse -- 4 Verse Scenario to Hybrid Verification -- 5 Experiments and Use Cases -- 6 Related Work -- 7 Conclusions and Future Directions -- References -- Synthesis -- Counterexample Guided Knowledge Compilation for Boolean Functional Synthesis -- 1 Introduction -- 2 A Motivating Example -- 3 Preliminaries and Notation -- 4 A New Knowledge Representation for Skolem Functions -- 5 Towards Synthesizing the Skolem Basis Vector -- 6 Counterexample-Guided Rectification.
7 Implementation and Experiments.

Sommario/riassunto

This open access proceedings set constitutes the refereed proceedings of the 35th International Conference on Computer Aided Verification, CAV 2023, which was held in Paris, France, in July 2023. .