1. Record Nr. UNINA9910620200703321

Titolo	Automated technology for verification and analysis : 20th International Symposium, ATVA 2022, Beijing, China, October 25-28, 2022, proceedings / / edited by Ahmed Bouajjani, Lukas Holik, and Zhilin Wu
Pubbl/distr/stampa	Cham, Switzerland : , : Springer, , [2022] ©2022
ISBN	3-031-19992-8
Descrizione fisica	1 online resource (442 pages)
Collana	Lecture Notes in Computer Science ; ; v.13505
Disciplina	511.3
Soggetti	Automatic theorem proving
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Nota di contenuto	Intro -- Preface -- Organization -- Abstracts of Invited Talks -- Compositional Reasoning about Concurrent Randomized Programs (Extended Abstract) -- Flattening String Constraints -- Runtime Assurance for Verified AI-Based Autonomy -- The Civl Verifier -- Subgame Perfect Equilibrium with an Algorithmic Perspective -- Contents -- Invited Paper -- Learning Monitorable Operational Design Domains for Assured Autonomy -- 1 Introduction -- 2 Motivating Example: Autonomous Lane Keeping -- 3 Optimal Monitors for Operational Design Domains -- 3.1 Learning Monitors for ODDs -- 3.2 Challenges in Learning Monitorable ODDs -- 3.3 Quantitative Monitor Learning -- 3.4 Black-Box vs. White-Box Settings -- 4 Framework -- 4.1 Main Workflow -- 4.2 Simulation-Based Analysis Using VerifAI and Scenic -- 4.3 Data Generation -- 4.4 Conformance Testing -- 5 Experiments -- 5.1 Experimental Setup -- 5.2 Results -- 6 Related Work -- 7 Conclusion -- References -- Reinforcement Learning -- Dynamic Shielding for Reinforcement Learning in Black-Box Environments -- 1 Introduction -- 1.1 Related Works -- 2 Preliminaries -- 2.1 Automata and Games for System Modeling -- 2.2 Safety Automata for Specifications -- 2.3 Shielding for Safe Reinforcement Learning -- 2.4 The RPNI Algorithm for Passive Automata Learning -- 3 Dynamic Shielding with Online Automata Inference -- 3.1 Dynamic Shielding Scheme -- 3.2 Challenge 1: Incompleteness of the Learned

FSRS -- 3.3 Challenge 2: Precision in Automata Learning -- 3.4 Theoretical Validity of Our Dynamic Shielding -- 4 Experimental Evaluation -- 4.1 Implementation and Experiments -- 4.2 Benchmarks -- 4.3 RQ1: Safety by Dynamic Shielding in the Training Phase -- 4.4 RQ2: Performance of the Resulting Controller -- 4.5 RQ3: Time Efficiency of Dynamic Shielding -- 5 Conclusions and Perspectives -- References.
An Impossibility Result in Automata-Theoretic Reinforcement Learning -- 1 Introduction -- 2 Omega-Automata -- 3 Closure Properties of Acceptance Conditions -- 4 Markov Decision Processes -- 4.1 Optimal Strategies Against Omega-Automata -- 4.2 Optimal Strategies Against Scalar Rewards -- 5 Memoryless Reward Translations for RL -- 5.1 Memoryless Reward Translation -- 5.2 Conditions for Memoryless Reductions -- 6 Conclusion -- References -- Reusable Contracts for Safe Integration of Reinforcement Learning in Hybrid Systems -- 1 Introduction -- 2 Preliminaries -- 2.1 Reinforcement Learning -- 2.2 Simulink and the RL Toolbox -- 2.3 Deductive Verification with the Differential Dynamic Logic -- 2.4 Transformation from Simulink to dL -- 3 Related Work -- 4 Reusable Hybrid Contracts -- 4.1 Illustrating Examples -- 4.2 Recurring Elements -- 4.3 Threshold Pattern -- 4.4 Range Pattern -- 4.5 Range Recovery Pattern -- 4.6 Resilience Contracts -- 4.7 Deductive Verification in KeYmaera X -- 5 Conclusion -- References -- Program Analysis and Verification -- SISL: Concolic Testing of Structured Binary Input Formats via Partial Specification -- 1 Introduction -- 2 Scheme-Based Input Specification Language -- 3 Overview and Implementation -- 4 Experiments and Conclusion -- References -- Fence Synthesis Under the C11 Memory Model -- 1 Introduction -- 2 Overview of FenSying and fFenSying -- 3 Preliminaries -- 4 Background: C11 Memory Model -- 5 Invalidating Buggy Traces with C11 Fences -- 6 Methodology -- 7 Implementation and Results -- 8 Related Work -- 9 Conclusion and Future Work -- References -- Checking Scheduling-Induced Violations of Control Safety Properties -- 1 Introduction -- 2 System Model and Encoding -- 2.1 Control System Model and Evolution -- 2.2 Task Specification -- 2.3 An Abstraction for Task Runs -- 2.4 Control Action Update Modeling.
2.5 Composing Control and Scheduling Models -- 3 Refining the Abstraction -- 3.1 Overlapping Jobs -- 3.2 Schedule Violation -- 3.3 Work Conservation Violation -- 3.4 Unconstrained Control Updates -- 3.5 Correctness of Refinement -- 4 Tool Design -- 5 Case Study 1: DC Motor Control Model -- 6 Case Study 2: RC Network Control Model -- 7 Case Study 3: F1Tenth Car Model -- 8 Conclusions and Future Work -- References -- Symbolic Runtime Verification for Monitoring Under Uncertainties and Assumptions -- 1 Introduction -- 2 Preliminaries -- 3 A Framework for Symbolic Runtime Verification -- 3.1 Symbolic Expressions -- 3.2 Symbolic Monitor Semantics -- 3.3 A Symbolic Runtime Verification Algorithm -- 4 Symbolic Runtime Verification at Work -- 4.1 Application to Lola Fragments -- 4.2 Temporal Assumptions -- 5 Implementation and Empirical Evaluation -- 6 Conclusion -- References -- SMT and Verification -- Handling Polynomial and Transcendental Functions in SMT via Unconstrained Optimisation and Topological Degree Test -- 1 Introduction -- 2 Background -- 2.1 Unconstrained Global Optimisation -- 2.2 Interval Arithmetic -- 2.3 Robustness and Quasi-decidability -- 2.4 Topological Degree Test -- 3 Local Search Using Unconstrained Global Optimisation -- 4 Solving Bounded Instances with the Topological Degree Test and Interval Arithmetic -- 4.1 Quasi-decidability Procedure -- 4.2 From a Formula with nm to Quasi-dec -- 4.3 A General Procedure -- 5 From

Constraint Sets to Formulas -- 5.1 An Eager Approach -- 5.2 A Lazy Approach -- 6 Experimental Evaluation -- 7 Conclusions and Future Work -- References -- Verification of SMT Systems with Quantifiers -- 1 Introduction -- 2 Preliminaries -- 3 Verification of Quantified SMT Systems -- 3.1 Symbolic Formalism -- 3.2 Overview -- 3.3 Ground Instances -- 3.4 Generalizing Invariants from Instances -- 3.5 Invariant Checking.
3.6 Termination -- 4 Related Work -- 5 Experimental Evaluation -- 6 Conclusions and Future Work -- References -- Projected Model Counting: Beyond Independent Support -- 1 Introduction -- 2 Notation and Preliminaries -- 3 Related Work -- 4 Technical Contribution -- 4.1 Extremal Properties of GIS and UBS -- 4.2 Algorithm to Compute Projected Count Using UBS -- 5 Experimental Evaluation -- 6 Conclusion -- References -- Automata and Applications -- Minimization of Automata for Liveness Languages -- 1 Introduction -- 2 Preliminaries -- 2.1 Automata -- 2.2 Liveness Languages -- 2.3 Graphs, Nice Graphs, and the Vertex-Cover Problem -- 3 Live1 Languages -- 3.1 Minimizing Automata for Live1 and Doom1 Languages -- 4 Live2 Languages -- 4.1 Minimizing DBWs and GFG-NBWs for Live2 Languages -- 4.2 Minimizing DCWs and GFG-NCWs for Doom2 Languages -- 4.3 Minimizing Automata with Transition-Based Acceptance for Live2 Languages -- 5 Live3 Languages -- References -- Temporal Causality in Reactive Systems -- 1 Introduction -- 2 Preliminaries -- 3 Motivating Example -- 4 Property Causality -- 4.1 Interventions -- 4.2 Contingencies -- 4.3 Actual Causality for Trace Properties -- 5 Checking -Regular Causality -- 5.1 Interventions -- 5.2 Contingencies -- 5.3 Minimality -- 5.4 Deciding -Regular Causality -- 6 Related Work -- 7 Conclusion -- References -- PDAAAL: A Library for Reachability Analysis of Weighted Pushdown Systems -- 1 Introduction -- 2 Weighted Pushdown Systems and Reachability -- 3 Implemented Algorithms and PDAAAL Architecture -- 4 Comparison with State-of-the-Art -- 5 Applications -- 6 Conclusion -- References -- Active Learning -- Learning Deterministic One-Clock Timed Automata via Mutation Testing -- 1 Introduction -- 2 Preliminaries -- 2.1 Deterministic One-Clock Timed Automata -- 2.2 Active Learning Algorithm for DOTAs -- 2.3 Model-Based Mutation Testing.
3 Mutation-Based Testing for DOTAs -- 3.1 The Process Overview -- 3.2 Heuristic Test-Case Generation -- 3.3 Mutation and Score-Based Test-Case Selection -- 4 Learning-Friendly Mutation Operators for DOTAs -- 4.1 Timed Mutation Operator -- 4.2 Split-Location Mutation Operator -- 5 Implementation and Experiments -- 5.1 Case Studies -- 5.2 Evaluation of Improvements -- 6 Conclusion -- References -- Active Learning of One-Clock Timed Automata Using Constraint Solving -- 1 Introduction -- 2 Preliminaries -- 3 Learning Algorithm -- 3.1 Alignment and Comparison of Timed Words -- 3.2 Timed Observation Table -- 3.3 Encoding of Readiness Constraints -- 3.4 Hypothesis Construction -- 3.5 Main Algorithm and Correctness -- 4 Extension to Deterministic Timed Mealy Machines -- 5 Implementation and Experiments -- 5.1 Experiments on DOTAs -- 5.2 Experiments on TMMs -- 6 Conclusion -- References -- Learning and Characterizing Fully-Ordered Lattice Automata -- 1 Introduction -- 2 Preliminaries -- 3 A Myhill-Nerode Characterization for FOLAs -- 3.1 No Unique Minimal FOLA -- 3.2 Difficulties in Defining f -- 3.3 Defining the Equivalence Relation -- 3.4 The Correspondence Between f and a Minimal FOLA -- 4 The Learning Algorithm -- 5 Empirical Results -- 6 Conclusions -- References -- Probabilistic and Stochastic Systems -- Optimistic and Topological Value Iteration for Simple Stochastic Games -- 1 Introduction -- 2 Preliminaries -- 2.1 Simple Stochastic Games --

2.2 Value Iteration and Bounded Value Iteration -- 3 Optimistic Value Iteration -- 4 Precise Topological Value Iteration -- 5 Random Generation of Simple Stochastic Games -- 6 Experiments -- 6.1 Experimental Setup -- 6.2 Overview -- 6.3 Detailed Analysis of Precise Algorithms -- 6.4 Detailed Analysis of Approximate (-Precise) Algorithms -- 7 Conclusion -- References -- Alternating Good-for-MDPs Automata.
1 Introduction.