

1. Record Nr.	UNISA996587866703316
Autore	Hermanns Holger
Titolo	Dependable Software Engineering. Theories, Tools, and Applications : 9th International Symposium, SETTA 2023, Nanjing, China, November 27-29, 2023, Proceedings
Pubbl/distr/stampa	Singapore : , : Springer, , 2024 ©2024
ISBN	981-9986-64-8
Edizione	[1st ed.]
Descrizione fisica	1 online resource (448 pages)
Collana	Lecture Notes in Computer Science Series ; ; v.14464
Altri autori (Persone)	SunJun BuLei
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Nota di contenuto	Intro -- Preface -- Organization -- Contents -- String Constraints with Regex-Counting and String-Length Solved More Efficiently -- 1 Introduction -- 2 Overview -- 3 Preliminaries -- 4 String Constraints with Regex-Counting and String-Length -- 5 Cost-Enriched Finite Automata -- 6 Solving RECL Constraints -- 6.1 From SATRECL to NELIA (CEFA) -- 6.2 Solving NELIA(CEFA) -- 7 Experiments -- 7.1 Benchmark Suites and Experiment Setup -- 7.2 Performance Evaluation -- 7.3 Evaluation on Problem Instances with Large Bounds -- 7.4 Empirical Justification of the Technical Choices Made in the Decision Procedure -- 8 Conclusion -- References -- Reachability Based Uniform Controllability to Target Set with Evolution Function -- 1 Introduction -- 2 Problem Formulation -- 3 Reachability Based Theoretical Foundation -- 3.1 Approximation of Reachable Set -- 3.2 Hausdorff Semi-distance Based Control Synthesis -- 3.3 Reachability Verification -- 4 Reachability Based Heuristic Framework -- 5 Improvements of Reachability Based Framework -- 5.1 K-Arm Bandit Model Based Improvement -- 5.2 Reference Trajectory Based Further Improvement -- 6 Examples and Discussions -- 7 Conclusion -- References -- Enhancing Branch and Bound for Robustness Verification of Neural Networks via an Effective Branching Strategy -- 1 Introduction -- 2 Preliminaries -- 2.1 Neural Network Verification -- 2.2 Branch and

Bound -- 3 Enhancing BaB with a Better Branching Strategy -- 3.1  
Improvement to Lower Bound by Splitting -- 3.2 Better Splitting  
Decision -- 4 Experimental Evaluation -- 4.1 Benchmarks -- 4.2  
Experiment Results -- 5 Conclusion -- References -- Graph-Based Log  
Anomaly Detection via Adversarial Training -- 1 Introduction -- 2  
Related Work -- 2.1 Log-Based Anomaly Detection -- 2.2 Generative  
Adversarial Network for Anomaly Detection -- 3 Framework -- 3.1  
Problem Statement.  
3.2 Log Preprocessing -- 3.3 Graph Construction -- 3.4 Graph  
Representation Learning -- 3.5 Adversarial Training Model -- 3.6  
Anomaly Detection -- 4 Experiments and Results -- 4.1 Datasets and  
Evaluation Metrics -- 4.2 Baselines and Implementation Details -- 4.3  
RQ1: Comparison with Baseline Models -- 4.4 RQ2: Robustness w.r.t.  
Data Contamination -- 4.5 RQ3: Ablation Study -- 5 Conclusion --  
References -- Formal Verification Based Synthesis for Behavior Trees --  
1 Introduction -- 2 Background -- 2.1 Behavior Trees -- 2.2 Linear  
Temporal Logic -- 2.3 Communicating Sequential Processes -- 3  
Problem Formulation -- 4 Proposed Approach -- 4.1 CSP Modelling  
and Verification -- 4.2 Grammar-Based MCTS -- 5 Demonstration --  
5.1 Experimental Setup -- 5.2 Comparison Experiment -- 5.3 Ablation  
Experiment -- 5.4 Discussion -- 6 Related Work -- 7 Conclusion --  
References -- SeHBPL: Behavioral Semantics-Based Patch Presence Test  
for Binaries -- 1 Introduction -- 2 Overview -- 2.1 Challenges -- 2.2  
Behavioral Semantics -- 2.3 Insights -- 2.4 Running Example -- 3  
Approach Design -- 3.1 Source Code Parsing -- 3.2 Behavioral  
Semantics Extracting -- 3.3 Behavioral Semantics Matching -- 4  
Evaluation -- 4.1 Testing Data Setup -- 4.2 Evaluation Setup -- 4.3  
Evaluation on Effectiveness -- 4.4 Evaluation of Efficiency -- 4.5  
Performance of Behavioral Semantics -- 5 Discussion -- 6 Related Work  
-- 7 Conclusion -- References -- Session Types with Multiple Senders  
Single Receiver -- 1 Introduction -- 1.1 Related Work -- 2 A Motivating  
Example -- 3 MSSR Session Types -- 3.1 Session -Calculus -- 3.2  
Global and Local Session Types -- 3.3 Projection and Well-Formed  
Global Types -- 3.4 Consistency of Global Types and Local Types -- 4  
Type System -- 5 A Communication Type System for Progress -- 5.1  
Typing Rules -- 6 Modelling Rust Multi-threads in MSSR Session Types.  
7 Conclusion and Future Work -- References -- Understanding the  
Reproducibility Issues of Monkey for GUI Testing -- 1 Introduction -- 2  
Empirical Study Methodology -- 2.1 Notations and Definitions -- 2.2  
Experimental Method -- 2.3 Experimental Setup -- 3 Experimental  
Results Analysis -- 3.1 RQ1: REPRODUCIBILITY RATE -- 3.2 RQ2: ROOT  
CAUSE -- 4 Discussions and Implications -- 4.1 How Does Throttle  
Affect Monkey's Reproducibility Rate? -- 4.2 Can R&R Tools  
Improve Monkey's Reproducibility Rate? -- 4.3 Threats to Validity -- 5  
Related Work -- 6 Conclusion and Future Work -- References -- Multi-  
dimensional Abstraction and Decomposition for Separation of Concerns  
-- 1 Introduction -- 2 Linking Formal Methods -- 2.1 Formal Theories  
of Different Aspects -- 2.2 UTP for Linking Formal Theories -- 3 rCOS  
Theory for Component-Based Architecture -- 3.1 rCOS Theory of  
Semantics and Refinement of OO Programming -- 3.2 Model of  
Interface Contracts of Components -- 4 rCOS Support to Separation of  
Concerns in MDE -- 4.1 Use Case Driven Requirements Model Building  
-- 4.2 Component Development Process -- 4.3 System Development  
-- 5 Conclusions -- References -- Solving SMT over Non-linear Real  
Arithmetic via Numerical Sampling and Symbolic Verification -- 1  
Introduction -- 2 Preliminaries -- 2.1 Real Arithmetic Formula -- 2.2  
Zero-Dimensional Systems and Real Zeros -- 3 Numeric Sampling via  
Random Global Optimization -- 4 The Main Algorithm -- 4.1 Using

Numeric Samples to Simplify the Formula -- 4.2 DPLL-Based Splitting Procedure -- 4.3 Model Generation and Verification -- 4.4 Reducing the Equation Set to Zero-Dimensional System -- 4.5 Determine Satisfiability of Inequality -- 5 Experiment -- 5.1 Experiment Preparation -- 5.2 Instances -- 5.3 Comparison to Symbolic Computation Tools -- 5.4 Comparison to State-of-the-Art SMT Solvers -- 6 Conclusion and Future Work.

References -- Leveraging TLA+ Specifications to Improve the Reliability of the ZooKeeperCoordination Service -- 1 Introduction -- 2 ZooKeeper and TLA+ -- 2.1 ZooKeeper and Zab -- 2.2 TLA+ Basics -- 3 Protocol Specification -- 3.1 Development of Protocol Specification -- 3.2 Ensuring Quality of Protocol Specification -- 4 System Specification -- 4.1 Essentials of a Super-Doc in TLA+ -- 4.2 Development of the Super-Doc -- 4.3 Ensuring Quality of the Super-Doc -- 5 Test Specification -- 5.1 Obtaining the Test Specification -- 5.2 Improving the Accuracy of Specification -- 5.3 Test Specification in Action -- 6 Related Work -- 7 Conclusion and Future Work -- References -- Modeling Regex Operators for Solving Regex Crossword Puzzles -- 1 Introduction -- 2 Preliminaries -- 3 Modeling Regex Operators -- 3.1 The Function  $(r,p,l)$  -- 3.2 Refinement -- 4 Solving Regex Crossword Puzzles -- 4.1 The Definitions -- 4.2 Rcps: Regex Crossword Puzzle Solver -- 5 Experiments -- 5.1 Experiment Setup -- 5.2 Effectiveness and Efficiency of Rcps -- 6 Related Work -- 7 Conclusion -- References -- Software Vulnerability Detection Using an Enhanced Generalization Strategy -- 1 Introduction -- 2 Related Works -- 2.1 In-Domain Vulnerability Detection Methods -- 2.2 Cross-Domain Vulnerability Detection Methods -- 3 Approach -- 3.1 Data Preparation -- 3.2 Meta Learning Model -- 3.3 Detection Model -- 4 Evaluation -- 4.1 Experimental Setup -- 4.2 Experiment Results -- 5 Conclusion -- References -- HeatC: A Variable-Grained Coverage Criterion for Deep Learning Systems -- 1 Introduction -- 2 Class Activation Mapping -- 3 HeatC Coverage Criterion -- 3.1 Generation of Heat Feature Buckets -- 3.2 Test Adequacy Evaluation and Test Sample Selection -- 4 Evaluation -- 4.1 Experiment Design -- 4.2 Experiment Results -- 5 Conclusion -- References.

Formalization of Lambda Calculus with Explicit Names as a Nominal Reasoning Framework -- 1 Introduction -- 1.1 Related Work -- 1.2 Our Contributions -- 2 Syntax and Alpha-Equivalence -- 2.1 Names -- 2.2 The Syntax of Lambda Calculus -- 2.3 Alpha Equivalence -- 2.4 Substitution -- 2.5 An Alpha-Structural Induction Principle -- 3 The Church-Rosser Theorem -- 3.1 Rule Induction Principles -- 3.2 Proof of Church-Rosser -- 4 Application: First-Order Logic Extended with Dynamically-Defined Predicates -- 5 Conclusions -- References -- Vulnerability Report Analysis and Vulnerability Reproduction for Web Applications -- 1 Introduction -- 2 Basic Concepts and Related Techniques -- 2.1 Vulnerability Report -- 2.2 Natural Language Processing -- 2.3 Test Scripts and Testing Framework of Web Applications -- 3 Vulnerability Report Analysis and Vulnerability Reproduction Approach for Web Applications -- 3.1 Overview -- 3.2 Syntactic Dependency Patterns Summarising from Vulnerability Reports -- 3.3 Automated Parsing of Vulnerability Reports -- 3.4 Automatic Vulnerability Reproduction Based on Semantic Similarity Between CIS and Event of Web Application -- 4 Experiment -- 4.1 Experimental Subjects and Environment -- 4.2 Analysis of Experimental Results -- 5 Conclusion -- References -- Run-Time Assured Reinforcement Learning for Safe Spacecraft Rendezvous with Obstacle Avoidance -- 1 Introduction and Related Work -- 2 Background -- 2.1 Reinforcement Learning -- 2.2 Run-Time Assurance -- 3 Safe Spacecraft Rendezvous

-- 3.1 Relative Motion Dynamics -- 3.2 Safety Constraints -- 3.3 Run-Time Assured RL Algorithm -- 4 Experiments -- 4.1 Spacecraft Rendezvous Environment -- 4.2 Reward Shaping -- 4.3 Hyperparameters -- 5 Results -- 5.1 Training Performance Analysis -- 5.2 Simulation and Expanded Test -- 6 Conclusions -- References.  
An Abstract Domain of Linear Templates with Disjunctive Right-Hand-Side Intervals.

---