

1. Record Nr.	UNINA9910485602103321
Titolo	Tests and proofs : 15th international conference, TAP 2021, held as part of STAF 2021, virtual event, June 21-22, 2021 : proceedings // edited by Frederic Louergue and Franz Wotawa
Pubbl/distr/stampa	Cham, Switzerland : , : Springer, , [2021] ©2021
ISBN	3-030-79379-6
Descrizione fisica	1 online resource (117 pages)
Collana	Lecture Notes in Computer Science ; ; v.12740
Disciplina	005.1
Soggetti	Computer software - Testing Software engineering
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Nota di bibliografia	Includes bibliographical references and index.
Nota di contenuto	Intro -- Preface -- Organization -- Abstracts of Invited Events -- Learning About the Change: An Adaptive Approach to Automata Learning -- Testing, Runtime Verification and Automata Learning -- Contents -- Learning, Test Resource Allocation and Benchmarks -- Use Case Testing: A Constrained Active Machine Learning Approach -- 1 Introduction -- 2 Background and Problem Statement -- 2.1 Use Case Modeling -- 2.2 Active Automaton Learning -- 2.3 Problem Statement: Scalable ML -- 3 Constrained Active Machine Learning (CAML) -- 3.1 Use Case Testing: An Example -- 3.2 A Parallel Distributed CAML Architecture -- 4 A Use Case Modeling Language for CAML -- 4.1 Input/Output Declaration -- 4.2 Sequencing, Static and Dynamic Constraints -- 4.3 Automated Test Verdict Construction -- 5 Evaluation and Benchmarking -- 5.1 ROBOTest: A CAML Implementation -- 5.2 Integration of ROBOTest and ASM -- 5.3 ACC Use Case Descriptions -- 5.4 ACC Test Objectives -- 6 Results -- 7 Related Work -- 8 Conclusions and Future Work -- References -- Architecture-Guided Test Resource Allocation via Logic -- 1 Introduction -- 1.1 Related Work -- 2 Quantitative Confidence Logic -- 2.1 Syntax and Proof Rules of QCL -- 2.2 Interpretation as Random Variables -- 3 Translating System Architectures to Proofs -- 4 Solving the Test Resource Allocation Problem -- 4.1 Confidence Functions --

4.2 The Optimisation Problem -- 5 Experimental Results -- 5.1 RQ1: Theoretical Evaluation -- 5.2 RQ2: Empirical Evaluation -- 5.3 Evaluation Results -- 6 Conclusion and Future Work -- References -- A Benchmarks Library for Extended Parametric Timed Automata -- 1 Introduction -- 2 Related Libraries -- 3 Parametric Timed Automata -- 4 The Benchmarks Library -- 4.1 Organization -- 4.2 Distribution -- 4.3 Benchmarks Classification -- 4.4 Properties -- 4.5 Unsolvable Benchmarks -- 4.6 Expected Performances.

5 Perspectives -- References -- Testing -- Generating Timed UI Tests from Counterexamples -- 1 Introduction -- 2 Preliminaries -- 3 Scratch Error Witnesses -- 3.1 User Inputs -- 3.2 Mocks -- 4 Witness Generation -- 4.1 Concrete Program Trace from an Abstract Reachability Graph -- 4.2 Error Witness from a Concrete Program Trace -- 5 Evaluation -- 5.1 Experiment Setup -- 5.2 Witness Replay and Validation (Effectiveness) -- 5.3 Sequential Tool Combination (Efficiency) -- 6 Related Work -- 7 Conclusions -- References -- Using a Guided Fuzzer and Preconditions to Achieve Branch Coverage with Valid Inputs -- 1 Introduction -- 2 Related Work -- 3 JMLKelinci -- 4 Experimental Study -- 5 Conclusion -- References -- FuSeBMC: An Energy-Efficient Test Generator for Finding Security Vulnerabilities in C Programs -- 1 Introduction -- 2 Preliminaries -- 2.1 Fuzzing -- 2.2 Symbolic Execution -- 2.3 Types of Vulnerabilities -- 3 FuSeBMC: An Energy-Efficient Test Generator for Finding Security Vulnerabilities in C Programs -- 3.1 Analyze C Code -- 3.2 Inject Labels -- 3.3 Produce Counterexamples -- 3.4 Create Graphml -- 3.5 Produce Test Cases -- 3.6 Selective Fuzzer -- 3.7 Test Validator -- 4 Evaluation -- 4.1 Description of Benchmarks and Setup -- 4.2 Objectives -- 4.3 Results -- 5 Related Work -- 6 Conclusions and Future Work -- A Appendix -- A.1 Artifact -- A.2 Tool Availability -- A.3 Tool Setup -- References -- Author Index.
