1. Record Nr. UNISA996383938203316

Autore	Payne William <1650-1696.>
Titolo	The three grand corruptions of the Eucharist in the Church of Rome [[electronic resource]] : Viz. the adoration of the Host, communion in one kind, sacrifice of the Mass. In three discourses
Pubbl/distr/stampa	London, : printed for Brabazon Aylmer, at the Three Pidgeons over against the Royal Exchange in Cornhil, MDCLXXXVIII. [1688]
Descrizione fisica	[6], 66, [8], 138, [6], 96 p
Soggetti	Lord's Supper
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	By William Payne. A reissue, with collective half-title and title page, of: A discourse concerning the adoration of the Host, 1685 (cf. Wing P898); A discourse of the communion in one kind, 1687 (cf. Wing P900); A discourse of the sacrifice of the Mass, 1688 (cf. Wing P901). Each has a separate dated title page, pagination and register. Reproduction of the original in the Cambridge University Library.
Sommario/riassunto	eebo-0021

2. Record Nr. UNISA996587866003316

Autore	Dimitrova Rayna
Titolo	Verification, Model Checking, and Abstract Interpretation : 25th International Conference, VMCAI 2024, London, United Kingdom, January 15-16, 2024, Proceedings, Part II
Pubbl/distr/stampa	Cham : , : Springer International Publishing AG, , 2024 ©2024
ISBN	3-031-50521-2
Edizione	[1st ed.]
Descrizione fisica	1 online resource (349 pages)
Collana	Lecture Notes in Computer Science Series ; ; v.14500
Altri autori (Persone)	LahavOri WolffSebastian
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Nota di contenuto	Intro -- Preface -- Organization -- Invited Keynote Talks -- Two Projects on Human Interaction with AI -- Automating Relational Verification of Infinite-State Programs -- Contents - Part II -- Contents - Part I -- Concurrency -- Petrification: Software Model Checking for Programs with Dynamic Thread Management -- 1 Introduction -- 2 A Language for Dynamic Thread Creation -- 3 Petrification -- 3.1 Petri Programs -- 3.2 Transformation to Petri Programs -- 3.3 Properties of the Petrified Program -- 4 Verifying Programs Through Repeated Petrification -- 5 Application: Verification of C Programs -- 5.1 Translation from Pthreads to Conc Programs -- 5.2 Practical Performance of the Approach -- 6 Conclusion -- References -- A Fully Verified Persistency Library -- 1 Introduction -- 2 Motivation -- 3 Using IOAs to Verify (Durable) Linearizability -- 4 Abstract Persistency Library and Its Correctness -- 5 The Library FliT and Memory Models -- 6 Proving Correctness of FliT -- 6.1 Proof of FliT [wPSC] PLibIR -- 6.2 Proof of FliT [PTSO] FliT [wPSC] -- 7 Related Work -- 8 Conclusion -- References -- A Navigation Logic for Recursive Programs with Dynamic Thread Creation -- 1 Introduction -- 2 Preliminaries -- 3 Graph Semantics of Dynamic Pushdown Networks -- 4 A Navigation Logic for Dynamic Pushdown Networks -- 5 Example Properties -- 6 From Graph Semantics to Tree Semantics -- 7 Model Checking and Satisfiability --

8 Conclusion -- References -- Neural Networks -- Taming Reachability Analysis of DNN-Controlled Systems via Abstraction-Based Training -- 1 Introduction -- 2 Preliminaries -- 2.1 DNN-Controlled Systems -- 2.2 Reachability Problem of DNN-Controlled Systems -- 3 Motivation -- 4 Abstraction-Based Training -- 4.1 Approach Overview -- 4.2 Interval-Based State Abstraction -- 5 Abstraction-Based Reachability Analysis -- 5.1 Approach Overview.
5.2 Key Operations in Algorithm 1 -- 5.3 The Soundness -- 6 Implementation and Experiments -- 6.1 Implementation and Benchmarks -- 6.2 Performance of Trained Neural Networks -- 6.3 Tightness and Efficiency -- 6.4 Differential and Decomposing Analysis -- 7 Related Work -- 8 Conclusion and Future Work -- References -- Verification of Neural Networks' Local Differential Classification Privacy -- 1 Introduction -- 2 Preliminaries -- 3 Local Differential Classification Privacy -- 4 Our Approach -- 5 Sphynx: Safety Privacy Analyzer via Hyper-networks -- 5.1 Prediction of an Interval Hyper-network -- 5.2 Verification of a Hyper-network -- 5.3 Analysis of Sphynx -- 6 Evaluation -- 7 Related Work -- 8 Conclusion -- References -- AGNES: Abstraction-Guided Framework for Deep Neural Networks Security -- 1 Introduction -- 2 Preliminaries -- 2.1 Deep Neural Networks -- 2.2 Abstraction -- 2.3 Stimulation -- 3 Methodology -- 3.1 Identifying Cluster Representatives -- 3.2 Stimulation Techniques -- 4 Tool Architecture -- 5 Experiments -- 5.1 Backdoor Identification and Runtime Reduction -- 5.2 Performance on Various Triggers -- 6 Conclusion -- References -- Probabilistic and Quantum Programs -- Guaranteed Inference for Probabilistic Programs: A Parallelisable, Small-Step Operational Approach -- 1 Introduction -- 2 Preliminaries on Measure Theory -- 3 Probabilistic Programs -- 4 Observable Semantics -- 5 Inference via Vectorized Monte Carlo Sampling -- 6 Implementation and Evaluation -- 6.1 TSI: A TensorFlow Based Implementation -- 6.2 Evaluation of TSI -- 7 Conclusion -- References -- Local Reasoning About Probabilistic Behaviour for Classical-Quantum Programs -- 1 Introduction -- 2 Preliminaries -- 3 A Classical-Quantum Language -- 3.1 Syntax -- 3.2 Denotational Semantics -- 4 Proof System -- 4.1 Assertion Language -- 4.2 Proof System -- 5 Examples.
5.1 HHL Algorithm -- 5.2 Shor's Algorithm -- 6 Conclusion and Future Work -- References -- Program and System Verification -- Deductive Verification of Parameterized Embedded Systems Modeled in SystemC -- 1 Introduction -- 2 Background -- 2.1 SystemC -- 2.2 VerCors -- 3 Encoding the SystemC Semantics in PVL -- 3.1 Assumptions -- 3.2 Encoding of Events and Concurrent Processes -- 3.3 Encoding of SystemC Designs -- 4 Deductive Verification of SystemC Designs -- 4.1 Property Specification and Verification -- 4.2 Reachable Abstract States Invariant -- 4.3 Variables in the Abstract State -- 4.4 RASI Generation and Soundness -- 4.5 Verifying Global Properties with the RASI -- 5 Evaluation -- 5.1 Case Studies -- 5.2 Experimental Results -- 5.3 Scalability -- 5.4 Discussion -- 6 Related Work -- 7 Conclusion -- References -- Automatically Enforcing Rust Trait Properties -- 1 Introduction -- 2 Background and Related Work -- 2.1 Rust Traits -- 2.2 Kani -- 2.3 Related Work -- 3 Verifying Rust Trait Invariants -- 3.1 Workflow for Verifying Trait Invariants -- 3.2 Harnesses for Trait Invariants -- 4 Implementation and Evaluation -- 4.1 Implementation -- 4.2 Evaluation -- 4.3 Results -- 5 Limitations and Future Work -- 6 Conclusion -- References -- Borrowable Fractional Ownership Types for Verification -- 1 Introduction -- 2 Source Language -- 2.1 Syntax -- 2.2 Operational Semantics -- 3 Type System -- 3.1 Syntax of Types -- 3.2 Typing Rules -- 3.3 Type Preservation -- 4 Translation -- 4.1