1. Record Nr. UNINA9910465559403321

Titolo	Protecting heritage in the Caribbean [[electronic resource] /] / edited by Peter E. Siegel and Elizabeth Righter
Pubbl/distr/stampa	Tuscaloosa, : University of Alabama Press, 2011
ISBN	0-8173-8390-5
Descrizione fisica	1 online resource (218 p.)
Collana	Caribbean archaeology and ethnohistory
Altri autori (Persone)	SiegelPeter E RighterElizabeth <1937->
Disciplina	972.9/01
Soggetti	Cultural property - Protection - Caribbean Area Historic preservation - Caribbean Area Monuments - Conservation and restoration - Caribbean Area Museums - Caribbean Area Historic preservation - Economic aspects - Caribbean Area Economic development - Caribbean Area Electronic books. Caribbean Area Cultural policy Caribbean Area Antiquities Caribbean Area Economic conditions
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Description based upon print version of record.
Nota di bibliografia	Includes bibliographical references and index.
Nota di contenuto	Preface: Intersecting values in Caribbean heritage preservation / Peter E. Siegel -- The Bahamas / Michael P. Pateman -- Cuba / Daniel Torres Etayo -- United States Naval Station, Guantanamo Bay, Cuba / Bruce J. Larson -- Jamaica / Andrea Richards and Ainsley Henriques -- Dominican Republic / Esteban Prieto Vicioso -- Puerto Rico / Peter E. Siegel -- U.S. Virgin Islands / Elizabeth Righter -- St. Kitts and Nevis / Todd M. Ahlman and Kelley Scudder-Temple -- Antigua and Barbuda / Reg Murphy -- French West Indies / Benoit Berard and Christian Stouvenot -- Saint Lucia / Milton Eric Branford -- St. Vincent and the Grenadines / Paul E. Lewis -- St. Vincent and the Grenadines: recent efforts in protecting heritage / Richard T. Callaghan -- Barbados / Kevin Farmer -- Trinidad and Tobago / Basil A. Reid and Vel Lewis --

	Netherlands Antilles / Jay B. Haviser and R. Grant Gilmore III -- Patrimony or patricide? / William F. Keegan and Winston Phulgence -- Protecting Heritage in the Caribbean / Peter E. Siegel.
Sommario/riassunto	Heritage preservation is a broad term that can include the protection of a wide range of human-mediated material and cultural processes ranging from specific artifacts, ancient rock art, and features of the built environment and modified landscapes. As a region of multiple independent nations and colonial territories, the Caribbean shares a common heritage at some levels, yet at the same time there are vast historical and cultural differences. Likewise, approaches to Caribbean heritage preservation are similarly diverse in range and scope. This volume address

2.

Record Nr.	UNISA996384185603316
Autore	Case Thomas <1598-1682.>
Titolo	Two sermons lately preached at Westminster, before sundry of the Honourable House of Commons, by Thomas Case, minister of Gods Word. Published by order of the said House [[electronic resource]]
Pubbl/distr/stampa	London, : Printed by I. Raworth for Luke Fawne, and are to be sold at his shop, at the signe of the Parrot in Pauls-Churchyard, 1641
Descrizione fisica	[6], 31, [1], 56 p
Soggetti	Sermons, English - 17th century
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	"The second sermon preached before divers of the Honorable House of Commons at Westminster" (caption title) begins new pagination on Â² B1r. Reproduction of the original in the British Library.
Sommario/riassunto	eebo-0018

3. Record Nr. UNINA9910484522803321

Titolo	Verified Software. Theories, Tools, and Experiments : 9th International Conference, VSTTE 2017, Heidelberg, Germany, July 22-23, 2017, Revised Selected Papers / / edited by Andrei Paskevich, Thomas Wies
Pubbl/distr/stampa	Cham : , : Springer International Publishing : , : Imprint : Springer, , 2017
ISBN	3-319-72308-1
Edizione	[1st ed. 2017.]
Descrizione fisica	1 online resource (XIII, 211 p. 69 illus.)
Collana	Programming and Software Engineering, , 2945-9168 ; ; 10712
Disciplina	005.13
Soggetti	Software engineering Computer science Compilers (Computer programs) Computer simulation Artificial intelligence Computers Professions Software Engineering Theory of Computation Compilers and Interpreters Computer Modelling Artificial Intelligence The Computing Profession
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Includes index.
Nota di contenuto	Intro -- Preface -- Organization -- Abstracts of Short Papers -- Everest: A Verified and High-Performance HTTPS Stack -- Design Principles of Automated Reasoning Systems -- Why Verification Cannot Ignore Resource Usage -- Constructing Correct Concurrent Programs Layer by Layer -- Contents -- A Formally Verified Interpreter for a Shell-Like Programming Language -- 1 Introduction -- 2 Language -- 2.1 Elements of Shell -- 2.2 Syntax of CoLiS -- 2.3 Semantics -- 2.4 Mechanised Version -- 3 Interpreter -- 3.1 Proof of Soundness -- 4

Proof of Completeness -- 4.1 Proving (or not Proving) Termination with Heights and Sizes -- 4.2 Proving Termination with Ghosts and Skeletons -- 4.3 Reproducibility -- 5 Related Work -- 6 Conclusion and Future Work -- References -- A Formal Analysis of the Compact Position Reporting Algorithm -- 1 Introduction -- 2 The Compact Position Reporting Algorithm -- 2.1 Number of Longitude Zones -- 2.2 Encoding -- 2.3 Local Decoding -- 2.4 Global Decoding -- 3 Practical Results -- 3.1 Decoding Requirements -- 3.2 Numerical Simplifications -- 4 Animation of the CPR Specification -- 5 Conclusion -- References -- Proving JDK's Dual Pivot Quicksort Correct -- 1 Introduction -- 2 Dual Pivot Quicksort -- 2.1 The Abstract Algorithm -- 2.2 JDK's Implementation -- 3 Background -- 3.1 Java Modeling Language -- 3.2 The Program Verification System KeY -- 4 Specification and Verification -- 4.1 Proof Management by Gentle Problem Adaptation -- 4.2 The Sortedness Property -- 4.3 The Permutation Property -- 4.4 Absence of Integer Overflow -- 4.5 Sorting Pivot Candidates -- 5 Verification Effort -- 6 Invalid Invariant in Single Pivot Quicksort -- 7 Conclusions -- References -- A Semi-automatic Proof of Strong Connectivity -- 1 Introduction -- 2 The Algorithm -- 3 Invariants -- 4 Pre-/Post-conditions -- 5 The Formal Proof -- 6 Conclusion.
References -- Verifying Branch-Free Assembly Code in Why3 -- 1 Introduction -- 1.1 The Why3 Verification Platform -- 1.2 Multiprecision Multiplication -- 2 Verification Approach -- 2.1 Validation of the Formal Model -- 2.2 Logical Partitioning -- 3 Modelling the AVR Instruction Set -- 3.1 Representing Multiprecision Integers -- 3.2 Using Ghost Code to Reduce Annotations -- 3.3 Model Validation -- 3.4 Underspecification -- 4 Verifying AVR Assembly Code -- 4.1 Operand-Scanning Multiplication -- 4.2 Karatsuba Multiplication -- 5 Related Work -- 6 Conclusion -- 6.1 Future Work -- References -- How to Get an Efficient yet Verified Arbitrary-Precision Integer Library -- 1 Introduction -- 2 From WhyML to C -- 2.1 Machine Words and Arithmetic Primitives -- 2.2 A Simple Model for C Pointers and Heap Memory -- 2.3 Extracting to Idiomatic C Code -- 3 Computing with Arbitrary-Precision Integers -- 3.1 Algorithm Specifications -- 3.2 Example of Proved Algorithm: Comparison -- 3.3 Trickier Example: Long Division -- 3.4 Statistics on the Proof Effort -- 4 Benchmarks -- 5 Related Work -- 6 Conclusions -- References -- Automating the Verification of Floating-Point Programs -- 1 Introduction -- 2 Quick Introduction to SPARK and Why3 -- 3 VC Generation for Floating-Point Computations -- 3.1 Signature for a Generic Theory of IEEE FP Arithmetic -- 3.2 Theory Clones and FP Literals -- 3.3 Interpreting FP Computations in Ada -- 3.4 Proving VCs Using Native Support for SMT-LIB FP -- 3.5 Axiomatization for Provers Without Native Support -- 3.6 Consistency and Faithfulness -- 4 Experiments -- 4.1 Small Representative Examples -- 4.2 A Case Study -- 5 Conclusions and Perspectives -- References -- Adaptive Restart and CEGAR-Based Solver for Inverting Cryptographic Hash Functions -- 1 Introduction -- 2 Background on Cryptographic Hash Functions -- 2.1 SHA-1.
3 Architecture of MapleCrypt -- 3.1 SAT Encoding -- 3.2 CEGAR Loop Design -- 3.3 Multi-armed Bandit Restart -- 4 Experimental Results -- 4.1 Experimental Setup -- 4.2 CEGAR and MABR -- 4.3 Partial Preimage -- 5 Related Work -- 6 Conclusion and Future Work -- References -- Practical Void Safety -- 1 Introduction -- 2 Motivating Examples -- 3 Overview -- 3.1 Language Conventions and Terminology -- 3.2 Solution Outline -- 4 Related Work -- 5 Formalization -- 5.1 Initialization State -- 5.2 Safe Uses of Current -- 5.3 Detection of Qualified Feature Calls -- 5.4 Validity Predicate -- 6 Practical Results -- 7 Conclusion -- References -- Memory-Efficient Tactics for

Randomized LTL Model Checking -- 1 Introduction -- 2 Preliminaries -- 3 Monte Carlo Model Checking -- 4 Variants of the GS Tactic -- 5 Token-Based Tactics Without Hashing -- 6 Experimental Results -- 6.1 Dining Philosophers -- 6.2 Shared Memory Consensus Protocol -- 6.3 Pathological Linear Model -- 7 Conclusions and Further Work -- References -- Reordering Control Approaches to State Explosion in Model Checking with Memory Consistency Models -- 1 Introduction -- 2 Preliminaries -- 2.1 Instructions and Concurrent Programs -- 2.2 Reorderings of Instructions Under Relaxed MCMs -- 2.3 Representations of Memory Consistency Models -- 3 Reordering Control -- 3.1 Restriction of the Number of Issued Instructions -- 3.2 Restriction of Separations of Specified Instructions -- 3.3 Reordering-Oriented Explorations -- 4 Implementation for a Model Checker McSPIN -- 4.1 PROMELA Code Generated by McSPIN -- 4.2 Implementation of Local Restrictions -- 4.3 Implementation of Global Restrictions -- 4.4 Implementation of the Increasing and Decreasing Exploration Strategies -- 5 Case Studies of Various Concurrent Programs -- 5.1 MCMs, Programs, and the Experimental Environment.
5.2 How to Read the Tables of Experiments -- 5.3 Experimental Results -- 5.4 Other Results Obtained Besides the Comparisons -- 6 Related Work and Discussion -- 7 Conclusion and Future Work -- References -- An Abstraction Technique for Describing Concurrent Program Behaviour -- 1 Introduction -- 2 Background -- 2.1 Dynamic Locking -- 2.2 Process Algebra Terms -- 3 Motivating Example -- 4 Program Logic -- 4.1 Assertion Language -- 4.2 Proof System -- 5 Applications of the Logic -- 5.1 Concurrent Counting -- 5.2 Lock Specification -- 5.3 Other Verification Examples -- 6 Related Work -- 7 Conclusion -- References -- Author Index.

Sommario/riassunto

This volume constitutes the thoroughly refereed post-conference proceedings of the 9th International Conference on Verified Software: Theories, Tools, and Experiments, VSTTE 2017, held in Heidelberg, Germany, in July 2017. The 12 full papers presented were carefully revised and selected from 20 submissions. The papers describe large-scale verification efforts that involve collaboration, theory unification, tool integration, and formalized domain knowledge as well as novel experiments and case studies evaluating verification techniques and technologies.