1. Record Nr. UNISA996383521703316

Autore	Camden William <1551-1623.>
Titolo	Annales rerum Anglicarum, et Hibernicarum, regnante Elizabetha, ad annum salutis M.D.LXXXIX. Guilielmo Camdeno authore [[electronic resource]]
Pubbl/distr/stampa	Londini, : Typis Guilielmi Stansbij, impensis Simonis Watersoni, ad insigne Coronæ in Cœmeterio Paulino, M.DC.XV. [1615]
Descrizione fisica	[8], 499, [23] p
Soggetti	Great Britain History Elizabeth, 1558-1603 Early works to 1800
Lingua di pubblicazione	Latino
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Books 1-3 only. Includes index. Running title reads: Elizabetha AngliÃ¦ Regina. B1 is a cancel. With a final errata leaf. Identified as STC 4496a on UMI microfilm reel 589. Reproductions of the originals in the Folger Shakespeare Library and the Henry E. Huntington Library and Art Gallery. Appears at reel 589 (Folger Shakespeare Library copy) and at reel 1372 (Henry E. Huntington Library and Art Gallery copy).
Sommario/riassunto	eebo-0216

2. Record Nr. UNINA9910140621603321

Autore	Safonov V. O (Vladimir Olegovich)
Titolo	Trustworthy compilers / / Vladimir O. Safonov
Pubbl/distr/stampa	Hoboken, N.J., : Wiley, 2010
ISBN	9786612551468 9781282551466 1282551469 9780470593387 0470593385 9780470593349 0470593342
Descrizione fisica	1 online resource (317 p.)
Collana	Quantitative Software Engineering Series ; ; v.8
Disciplina	005.4/53
Soggetti	Compilers (Computer programs) Interpreters (Computer programs)
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	Trustworthy Compilers; Contents; Preface; Acknowledgments; 1. Introduction; 1.1. The Concept of a Trustworthy Compiler; 1.2. Kinds of Compilers; 1.3. Evolution of Java Compilers; 1.4. Compilation for .NET; 1.5. Phases of Compilation; 1.6. Overview of Compiler Development Principles and Technologies; 1.7. History of Compiler Development in the U.S.S.R. and in Russia; Exercises to Chapter 1; 2. Theoretical Foundations and Principles of Trustworthy Compilers; 2.1. The Trustworthy Computing (TWC) Initiative; 2.2. TWC and Trustworthy Compilers; 2.3. Verified Compilers 2.4. Spec#: Microsoft's Approach to Verifying Compilers2.5. Perspectives of Verified and Verifying Compilation; Exercises to Chapter 2; 3. Lexical Analysis and Its Trustworthiness Principles; 3.1. Token Classes; 3.2. The Output of the Lexical Analyzer; 3.3. Processing White Spaces, Comments, and New Lines; 3.4. Theoretical Models of Lexical Analysis; 3.5. Lexical Errors, Error Diagnostics, and Recovery; 3.6. Processing Identifiers and Keywords; 3.7. The Architecture of a Lexical Analyzer and the Principles of Its Implementation; 3.8. The Lexical

Analyzer Generator Lex

3.9. Lexical Analyzer Generation in ANTLRExercises to Chapter 3; 4. Parsing and Trustworthy Methods of Syntax Error Recovery; 4.1. Basic Concepts and Principles of Parsing; 4.2. Recursive Descent and Simple Lookahead Mechanism; 4.3. Overview of Error Recovery in Parsing: Error Recovery for Recursive Descent; 4.4. LR(1) and LALR(1) Parsing; 4.5. Error Recovery in LR Parsing; 4.6. The Yacc Parser Generator; 4.7. The Bison Parser Generator: Generalized LR Parsing; 4.8. The Yacc++, JavaCC, SableCC, ANTLR, and CoCo/R Object-Oriented Parser Generators; Exercises to Chapter 4

5. Semantic Analysis and Typing: Efficient and Trustworthy Techniques5.1. Basic Concepts and Principles of Semantic Analysis; 5.2. Formal Model of Semantic Analysis: Attributed Grammars; 5.3. Definition Systems with Forward References and the Algorithm of Their One-Pass Analysis; 5.4. Commonly Used Semantic Attributes for Program Constructs; 5.5. Design Flaws of the Semantic Attribute Evaluation and Our Efficient Methods to Speed It Up; 5.6. Lookup-Traditional and Novel Techniques; 5.7. Typing and Type-Checking: Basic Concepts; 5.8. Representing Types at Compile Time

5.9. Efficient Method and Algorithm to Represent and Handle Types with Structural Identity5.10. Type Identity and Type Compatibility; 5.11. Type-Checking, Typing Error Diagnostics, and Recovery; 5.12. Code Trustworthiness Checks During Semantic Analysis; 5.13. Checks for Context Restrictions in Semantic Analysis; 5.14. Intermediate Code Generation-Principles and Architectural Models; 5.15. Postfix (Reverse Polish) Notation; 5.16. PCC Trees; 5.17. Triples; 5.18. Summary of the Chapter; Exercises to Chapter 5; 6. Trustworthy Optimizations

6.1. Basic Concepts and Trustworthiness of Optimizations

Sommario/riassunto

The Most Complete, Real-World Guide to Compiler Development-and the Principles of Trustworthy Compilers Drawing on the author's over thirty years of expertise in compiler development, research, and instruction, Trustworthy Compilers introduces and analyzes the concept of trustworthy compilers and the principles of trustworthy compiler development, and provides analytical overview of other promising research works in this area. Vladimir Safonov shares the benefit of his long experience as a teacher and compiler development professional to explain that-even in such a well-studied area

3. Record Nr. UNINA9911020225903321

Titolo	Industrial used of formal method : formal verification / / edited by Jean-Louis Boulanger
Pubbl/distr/stampa	London, : ISTE Hoboken, N.J., : Wiley, c2012
ISBN	9781118587843 1118587847 9781118561829 1118561821 9781299187078 1299187072 9781118587904 1118587901
Descrizione fisica	1 online resource (307 p.)
Collana	ISTE
Altri autori (Persone)	BoulangerJean-Louis
Disciplina	005.101
Soggetti	Systems engineering - Data processing Computer simulation Formal methods (Computer science) Computer software - Verification Nondestructive testing
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	Cover; Title; Copyright; Table of Contents; Introduction; Chapter 1. SPARK - A Language and Tool-Set for High-Integrity Software Development; 1.1. Introduction; 1.2. An overview of SPARK; 1.2.1. What is SPARK?; 1.3. The rationale behind SPARK; 1.3.1. Flow analysis; 1.3.2. Code proof; 1.3.3. Correctness by construction; 1.4. Industrial applications of SPARK; 1.4.1. SHOLIS; 1.4.2. Lockheed-Martin C-130J mission computer; 1.4.3. MULTOS CA; 1.4.4. Tokeneer; 1.4.5. Aircraft monitoring software; 1.4.6. iFACTS; 1.4.7. SPARK Skein; 1.5. Conclusion; 1.6. Bibliography Chapter 2. Model-Based Testing Automatic Generationof Test Cases

Using the Markov Chain Model2.1. Preliminaries on the test process; 2.1.1. Findings; 2.1.2. Test optimization; 2.1.3. The statistical usage test; 2.1.4. Generating test cases; 2.2. Modeling using Markov chains; 2.2.1. Origin; 2.2.2. Mathematical formalization; 2.2.3. Principles of generation; 2.2.4. Some indicators; 2.2.5. Calculating reliability; 2.3. The MaTeLo tool; 2.3.1. Engineering tests directed by models,model-based testing; 2.3.2. A chain of tools; 2.3.3. The usage model; 2.3.4. Configuration of test strategies
2.3.5. Generating test campaigns2.3.6. Analysis of the results and indicators; 2.4. Examples of industrial applications; 2.4.1. AUDI; 2.4.2. Magneti marelli; 2.4.3. Other industrial applications; 2.4.4. Industrialization of the tests; 2.5. Conclusion; 2.6. Bibliography; Chapter 3. Safety Analysis of the Embedded Systems with the AltaRica Approach; 3.1. Introduction; 3.2. Safety analysis of embedded systems; 3.3. AltaRica language and tools; 3.3.1. The AltaRica language; 3.3.2. Modeling the propagation of failures with AltaRica; 3.3.3. Tools associated with AltaRica
3.4. Examples of modeling and safety analysis3.4.1. Integrated modular avionics architecture; 3.4.2. System of electric power generation and distribution; 3.5. Comparison with other approaches; 3.5.1. Some precursors; 3.5.2. Tools for preexisting formal languages; 3.5.3. Languages for physical systems; 3.5.4. Injecting faults in nominal models; 3.6. Conclusion; 3.6.1. An approach to assess the safetyof systems tested in aeronautics; 3.6.2. Clarification of the system architectureand horizontal exploration of the failure propagation:impacts on the scope of analyses
3.6.3. Clarification of the nominal system characteristics:impacts on the generic definitions of the failure modes3.6.4. Compositional models of failure propagation:impacts on the overall safety process; 3.7. Special thanks; 3.8. Bibliography; Chapter 4. Polyspace®; 4.1. Overview; 4.2. Introduction to software quality and verification procedures; 4.3. Static analysis; 4.4. Dynamic tests; 4.5. Abstract interpretation; 4.6. Code verification; 4.7. Robustness verification or contextual verification; 4.7.1. Robustness verification; 4.7.2. Contextual verification
4.8. Examples of Polyspace® results

Sommario/riassunto

"At present the literature gives students and researchers of the very general books on the formal technics. The purpose of this book is to present in a single book, a return of experience on the used of the "formal technics" (such proof and model-checking) on industrial examples for the transportation domain.This book is based on the experience of people which are completely involved in the realization and the evaluation of safety critical system software based. The implication of the industrialists allows to raise the problems of confidentiality which could appear and so allow to supply new useful information (photos, plan of architecture, real example)"--

4. Record Nr. UNINA9910893136203321

Titolo	Ponta de lanca: revista eletronica de historia, memoria & cultura
Pubbl/distr/stampa	Sao Cristovao, Sergipe, : Universidade Federal de Sergipe, Programa de Pos-Graduacao em Historia, [2007]-
ISSN	1982-193X
Descrizione fisica	Online-Ressource
Disciplina	900
Soggetti	Zeitschrift
Lingua di pubblicazione	Portoghese
Formato	Materiale a stampa
Livello bibliografico	Periodico
Note generali	Gesehen am 29. August 2018