

1. 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 Compilers 2.5. 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
