

Librería
Bonilla y Asociados
desde 1950



Título:

Autor:

Precio: \$1305.00

Editorial:

Año: 2002

Tema:

Edición: 2ª

Sinopsis

ISBN: 9780521820608

This textbook describes all phases of a compiler: lexical analysis, parsing, abstract syntax, semantic actions, intermediate representations, instruction selection via tree matching, dataflow analysis, graph-coloring register allocation, and runtime systems. It includes good coverage of current techniques in code generation and register allocation, as well as the compilation of functional and object-oriented languages, that is missing from most books. The most accepted and successful techniques are described concisely, rather than as an exhaustive catalog of every possible variant, and illustrated with actual Java classes. The first part of the book, Fundamentals of Compilation, is suitable for a one-semester first course in compiler design. The second part, Advanced Topics, which includes the compilation of object-oriented and functional languages, garbage collection, loop optimization, SSA form, instruction scheduling, and optimization for cache-memory hierarchies, can be used for a second-semester or graduate course. This new edition has been extensively rewritten to include more discussion of Java and object-oriented programming concepts, such as visitor patterns. A unique feature is the newly redesigned compiler project in Java, for a subset of Java itself. The project includes both front-end and back-end phases, so that students can build a complete working compiler in one semester.

! The new edition features a newly designed, easy-to-use software project in Java for a subset of Java itself ! Expanded coverage of object-oriented concepts; also includes coverage of imperative and functional languages ! Keeps up with the revolution in computer architecture since 1985, including practical back-end issues not available in most texts, with current techniques in instruction selection, code generation and register allocation

Contents

Part I. Fundamentals of Compilation: 1. Introduction; 2. Lexical analysis; 3. Parsing; 4. Abstract syntax; 5. Semantic analysis; 6. Activation records; 7. Translation to intermediate code; 8. Basic blocks and traces; 9. Instruction selection; 10. Liveness analysis; 11. Register allocation; 12. Putting it all together; Part II. Advanced Topics: 13. Garbage collection; 14. Object-oriented languages; 15. Functional programming languages; 16. Polymorphic types; 17. Dataflow

Teléfonos: 55 44 73 40 y 55 44 72 91

www.libreriabonilla.com.mx

Librería
Bonilla y Asociados
desde 1950



analysis; 18. Loop optimizations; 19. Static single-assignment form; 20. Pipelining and scheduling; 21. The memory hierarchy; Appendix: Mini-Java reference manual.