

<<编译原理>>

图书基本信息

书名：<<编译原理>>

13位ISBN编号：9787040305777

10位ISBN编号：7040305771

出版时间：2011-6

出版时间：高等教育出版社

作者：苏运霖，颜松远 著

页数：435

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<编译原理>>

内容概要

《编译原理:包含代数方法的新编译方法(英文版)》通过把编程语言的编译同人类对自然语言的理解过程进行类比来阐述编译程序的思想,采用标记法来创建源语言、中间语言和目标语言的符号,生动地描述了多层次编译程序的编译过程;详细地介绍了LL(1)和LR(1)的分析方法,不仅可以帮助读者了解如何做,还使他们知道为什么这样做;同时介绍了编译程序的设计方法,引入了一个重要的方法——代数形式化方法。

《编译原理:包含代数方法的新编译方法(英文版)》适合作为计算机和电子专业本科生和研究生教材,也可供相关学科研究人员参考。

书籍目录

Chapter 1 Introduction

- 1.1 Language and Mankind
- 1.2 Language and Computer
- 1.3 Compilation of Programming Languages
- 1.4 Number of Passes of Compiler
- 1.5 An Example of Compilation of a Statement
- 1.6 Organization of the Book

Problems

References

Chapter 2 Grammars and Languages

- 2.1 Motivation of the Chapter
- 2.2 Preliminary Knowledge
- 2.3 Grammar
- 2.4 Language
- 2.5 Language Generated by a Grammar
- 2.6 Turing Machine
- 2.7 Issues Concerning Grammars and Languages

Problems

References

Chapter 3 Finite State Automata and Regular

Languages

- 3.1 Motivations of the Chapter
- 3.2 Languages, Grammars and Automata
- 3.3 Deterministic Finite Automata
- 3.4 Nondeterministic Finite Automata
- 3.5 Regular Expressions
- 3.6 Regular Grammar
- 3.7 Kleene's and Moore's Theorems
- 3.8 Pumping Theorems and Closure Properties for LREG
- 3.9 Applications of Finite Automata
- 3.10 Variants of Finite Automata

Problems

References

Chapter 4 Lexical Analysis

- 4.1 Motivation of the Chapter
- 4.2 Lexical Analyzer
- 4.3 Output of Lexical Analyzer
- 4.4 Error Handling

Problems

References

Chapter 5 Push-Down Automata and Context-Free Languages

- 5.1 Motivation of the Chapter
- 5.2 Push-Down Automata
- 5.3 Context-Free Languages (LCF)
- 5.4 Pumping Theorems for Context-Free Languages

<<编译原理>>

- 5.5 Push-Down Automata and Context-Free Languages
- 5.6 Applications of Context-Free Languages
- 5.7 Turing Machines
- 5.8 Turing Machines as Language Accepters
- 5.9 Equivalence of Various Turing Machines
- 5.10 Recursively Enumerable Languages (LRE)
- 5.11 Context-Sensitive Languages (LCS)
- 5.12 Hierarchy of Machines, Grammars and Languages
- 5.13 Relations Among Machines, Languages and Grammars
 - 5.12.1 Hierarchy of Machines
- Problems
- References
- Chapter 6 Context-Free Grammars
 - 6.1 Motivation of the Chapter
 - 6.2 Context-Free Grammars
 - 6.3 Characteristics of Context-Free Grammars
- Problems
-
- Chapter 7 Syntax Analysis
- Chapter 8 Attribute Grammars and Analysis
- Chapter 9 Algebraic Method of Compiler Design
- Chapter 10 Generation of Intermediate Code
- Chapter 11 Debugging and Optimization
- Chapter 12 Storage Management
- Chapter 13 Generation of object code
- Chapter 14 Compilation of object-oriented Languages
- Chapter 15 Compilation of parallel Languages
- Chapter 16 Compilation of Grid Computing
- Index

章节摘录

1) The high efficiency of the execution of programs. Initially, when talking about high efficiency, almost no exception, it meant the efficiency of the execution of the programs. And this entails the quality of the compiler that compiles the source programs to object programs with the high efficiency of the execution. Therefore, it may involve the design of the optimal compiler, the efficient register allocation, as well as the mechanism design for supporting the running of programs. Though the efficiency of the execution of programs is intimately related to the design of the language, the quality of the compilation by the compiler decidedly affects the efficiency of the execution of the programs.

2) The efficiency of the compilation. Probably, this point is just consistent with what Wirth expressed. The large-scale productive program means that they frequently run. Hence the saving of even only few minutes is still crucial if the great number of running is taken into account for every day. This is why it becomes an issue which people are concerned with. The issue is related to the quality of the compiler. The other kind of the program is also related to the quality of the compiler. It is the compiler that is in charge of the compilation of student programs or programs used for teaching. Typically, the programs written by students will not be used for production. They were the results of their learning. Only the correctness is concerned. Therefore, we just need to make the compiler working efficiently to compile the program to point out the errors (if any) in the program. Therefore, in this situation the important thing is still the fast translation of the compiler, rather than that compiler can produce the object program with high execution efficiency and optimization.?

<<编译原理>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>