

<<编译原理>>

图书基本信息

书名：<<编译原理>>

13位ISBN编号：9787811332353

10位ISBN编号：7811332353

出版时间：2008-7

出版时间：哈尔滨工程大学出版社

作者：张素莉 等著

页数：250

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>>

前言

编译技术是计算机语言发展的支柱，也是计算机科学中发展最迅速、最成熟的一个分支。

“编译原理”是一门研究设计和构造编译程序原理和方法的课程，是计算机及其相关专业的一门核心课程，在教学中占有极其重要的地位。

该课程蕴含着计算机学科中解决问题的思路及形式化问题和解决问题的方法。

编译程序是计算机系统软件的重要组成部分，其基本原理和实现技术也适用于一般软件的设计和实现，因此对应用软件和系统软件的设计与开发有一定的启发和指导作用，在软件工程、软件自动化、软件再工程、语言转换、程序分析等领域具有广泛的应用。

本书主要介绍设计和构造编译程序的一般原理、基本方法和主要实现技术。

通过该课程的学习，学生能够掌握编译系统的结构、工作流程及编译程序各组成部分的设计原理和常用的编译技术和方法，为今后从事应用软件和系统软件的开发奠定一定的理论和实践基础。

本书共分9章。

第1章介绍了编译程序的基础知识，包括编译工作的基本过程及各阶段的基本任务；第2章介绍了文法及语言的基本概念、文法分类、短语、简单短语与句柄，语法树和二义性等；第3章介绍了词法分析程序的设计原理与构造方法；第4章介绍了语法分析的基本思想和分析技术，包括语法分析的任务、自顶向下分析法和自底向上分析法等；第5章介绍了语义分析与中间代码的生成；第6章介绍了符号表的组织与管理，包括符号表的作用、符号表的组织和使用方法；第7章介绍了程序运行时的存储组织与分配技术；第8章介绍了代码优化的基本概念、基本块的划分、局部优化和循环优化方法等；第9章介绍了目标代码生成的基本技术。

本书系统性强，概念清晰，内容简明易懂，每章章首配有本章的学习目标和学习要求，章末配有本章小结和适量的习题，使学习者可以快速掌握书中的内容。

该书根据编者多年的教学经验编写而成，在成书的过程中编者参考了有关文献，在此向这些书籍的作者一并表示感谢。

由于编者水平有限，时间仓促，书中难免存在一些缺点和不足，敬请读者多提宝贵意见，以便再版时修订补充。

<<编译原理>>

内容概要

《编译原理》是一门研究设计和构造编译程序原理和方法的课程，是计算机专业的一门重要的核心课程。

《“十一五”重点规划教材·高等院校计算机及其应用系列：编译原理》系统介绍了编译程序构造的一般原理和基本实现方法，主要内容包括词法分析、语法分析、语义分析与中间代码生成、代码优化、目标代码生成、符号表的组织和管理和运行时的存储组织与分配技术等。

《“十一五”重点规划教材·高等院校计算机及其应用系列：编译原理》内容丰富，系统性强，内容循序渐进，实例丰富，文字简练，通俗易懂，而且每章都配有各种类型的习题。

<<编译原理>>

书籍目录

第1章 编译概述1.1 程序的翻译1.2 编译程序的组成1.3 编译程序的构造1.4 小结第2章 形式语言的基本知识2.1 字母表和符号串的基本概念2.2 文法和语言的形式定义2.3 语法树与文法二义性2.4 文法和语言的分类2.5 小结第3章 词法分析3.1 词法分析的任务3.2 词法分析程序的输出形式3.3 词法分析程序的设计与实现3.4 正规式与有穷自动机3.5 词法分析程序的自动生成工具嘞3.6 实例语言的词法分析程序3.7 小结第4章 语法分析4.1 语法分析的任务4.2 自顶向下分析法4.3 自底向上分析法4.4 LR分析法4.5 语法分析程序的自动生成工具YACC4.6 实例语言编译程序的语法分析4.7 小结第5章 语义分析与中间代码的生成5.1 语义分析的任务5.2 语法制导翻译5.3 中间代码5.4 说明语句的翻译5.5 赋值语句的翻译5.6 布尔表达式的翻译5.7 控制流语句的翻译5.8 过程调用的翻译5.9 实例编译程序的语义分析5.10 小结第6章 符号表管理6.1 符号表的作用6.2 符号表的主要内容和作用6.3 符号表的组织6.4 符号表的管理6.5 小结第7章 运行时的存储组织与分配7.1 存储组织概述7.2 静态存储分配7.3 栈式动态存储分配7.4 堆式动态存储分配7.5 小结第8章 代码优化8.1 局部优化8.2 循环优化8.3 小结第9章 目标代码生成9.1 目标代码的形式9.2 假想的计算机模型9.3 一个简单的代码生成程序9.4 小结附录A C语言实现的实例语言编译程序附录B YACC语言实现的实例语言编译程序参考文献

<<编译原理>>

章节摘录

(1) 类型检查确保程序的每一部分在语言的类型规则的作用下有意义。

例如, 参与运算的操作数的类型应相容, 否则编译器会报错。

(2) 控制流检查用以保证控制语句拥有合法的转向点, 即引起控制流从某个结构中跳转出来的语句必须能够决定控制流转向的目标地址。

如C语言中的break语句将导致控制流离开包含其最小的while, for和switch语句, 如果找不到这样的语句, 将导致错误。

(3) 一致性检查有些情况下一个对象只能被定义一次。

如相同作用域内标识符只能说明一次, case语句中的标签也应该是唯一的。

(4) 相关名称检查有时候同样的名字会多次出现。

如Ada中, 循环或块中都将有一个名字同时出现在构造器的开始和结束。

编译器将检查同样的名字可以在两端被使用。

5.1.2语义分析的任务 语义分析的根本任务是对源程序在进行词法分析和语法分析的基础上, 进一步分析其含义, 在理解其含义的基础上为生成相应的目标代码作准备或者直接生成目标代码。

静态语义分析包括执行分析的描述和使用合适的算法对分析的实现, 它和词法分析及语法分析相类似。

例如, 在语法分析中使用上下文无关文法描述语法结构, 并用自顶向下或自底向上的分析算法实现语法结构。

在语义分析中, 不能用正规文法或者上下文无关文法进行描述, 一个原因是没有标准的方法来说明语言的静态语义; 另一个原因是对于各种语言, 静态语义分析的种类和总量的变化范围很大。

因此, 语义的形式化描述是相当困难的, 目前常常使用的且实现得很好的描述语义分析方法是利用属性文法描述程序设计语言的语义, 然后采用语法制导翻译的方法完成对语法单位的翻译工作。

5.2语法制导翻译 语法制导翻译方法就是在语法分析中依随分析的过程, 根据每个产生式添加的语义动作进行翻译的方法。

本节主要介绍属性文法及其语法制导翻译的概念。

5.2.1属性文法 属性文法是Knuth于1968年提出来的, 也被称为属性翻译文法。

属性文法以上下文无关文法为基础, 且为每个文法符号配备了一些属性。

属性代表着上下文无关文法中每个文法符号的语义, 这种语义符号可能由符号的类型、值、符号表的内容, 甚至一段代码序列构成, 而且一个符号可以有多个属性。

属性同变量一样, 可以进行计算和传递, 属性加工的过程就是语义处理的过程。

为文法的每个产生式配备的计算属性规则称为语义规则。

通常情况下, 文法的属性分为继承属性和综合属性两种。

继承属性用于“自上而下”传递信息, 即文法产生式右部符号的某些属性是根据其左部符号的属性和(或)右部的其他符号的某些属性计算得到的。

在语法树中, 一个节点的继承属性由此节点的父节点和(或)兄弟节点的某些属性计算得到, 即沿语法树向下传递, 由根节点到分枝节点, 反映了其对上下文依赖的特性。

因而继承属性可以很方便地表示程序设计语言结构中的上下文依赖关系。

.....

<<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>