

<<编译原理教程>>

图书基本信息

书名：<<编译原理教程>>

13位ISBN编号：9787560624631

10位ISBN编号：7560624634

出版时间：2010-10

出版时间：西安电子科技大学出版社

作者：胡元义

页数：236

字数：359000

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>

<<编译原理教程>>

内容概要

胡元义主编的这本《编译原理教程(第3版)》系统地介绍了编译程序的设计原理及实现技术，主要内容包括：绪论、词法分析、语法分析、语义分析和中间代码生成、代码优化、目标程序运行时存储空间的组织、目标代码生成、符号表与错误处理等。

在内容的组织上，本书强调知识的实用性，将编译的基本理论与具体的实现技术有机地结合起来，既注重了理论的完整性，化繁为简，又将理论融于具体的实例中，化难为易，以达到准确、清楚地阐述相关概念和原理的目的。

本书注重各章节对理论阐述的条理性，书中给出的例子也具有较强的实用性与连贯性，使读者对编译的各个阶段有一个全面、直观的认识。

本书采用的算法全部由C语言描述，各章均附有习题。

《编译原理教程(第3版)》可作为计算机本科专业的教材，也可作为计算机软件工程人员的参考资料。

<<编译原理教程>>

书籍目录

第一章 绪论

- 1.1 程序设计语言和编译程序
 - 1.2 编译程序的历史及发展
 - 1.3 编译过程和编译程序结构
 - 1.4 编译程序的开发
 - 1.5 构造编译程序所应具备的知识内容
- 习题

第二章 词法分析

- 2.1 词法分析器的设计方法
 - 2.1.1 单词符号的分类与输出形式
 - 2.1.2 状态转换图
 - 2.2 一个简单的词法分析器示例
 - 2.2.1 C语言子集的单词符号表示
 - 2.2.2 C语言子集对应的状态转换图
 - 2.2.3 状态转换图的实现
 - 2.3 正规表达式与有限自动机简介
 - 2.3.1 正规表达式与正规集
 - 2.3.2 有限自动机
 - 2.4 正规表达式到有限自动机的构造
 - 2.4.1 由正规表达式构造等价的非确定有限自动机(NFA)
 - 2.4.2 NFA的确定化
 - 2.4.3 确定有限自动机(DFA)的化简
 - 2.4.4 正规表达式到有限自动机构造示例
 - 2.5 词法分析器的自动生成
- 习题二

第三章 语法分析

- 3.1 文法和语言
 - 3.1.1 文法和语言的基本概念
 - 3.1.2 形式语言分类
 - 3.1.3 正规表达式与上下文无关文法
- 3.2 推导与语法树
 - 3.2.1 推导与短语
 - 3.2.2 语法树与二义性
- 3.3 自顶向下的语法分析
 - 3.3.1 递归下降分析法
 - 3.3.2 LL(1)分析法
- 3.4 自底向上的语法分析
 - 3.4.1 自底向上分析原理
 - 3.4.2 算符优先分析法
- 3.5 规范归约的自底向上语法分析方法
 - 3.5.1 LR分析器的工作原理
 - 3.5.2 LR(0)分析器
 - 3.5.3 SLR(1)分析器
 - 3.5.4 LR(1)分析器
 - 3.5.5 LALR分析器

<<编译原理教程>>

3.5.6 二义文法的应用

3.5.7 LR分析器应用与拓展

习题三

第四章 语义分析和中间代码生成

4.1 概述

4.1.1 语义分析的概念

4.1.2 语法制导翻译方法

4.2 属性文法

4.2.1 文法的属性

4.2.2 属性文法

4.3 几种常见的中间语言

4.3.1 抽象语法树

4.3.2 逆波兰表示法

4.3.3 三地址代码

4.4 表达式及赋值语句的翻译

4.4.1 简单算术表达式和赋值语句的翻译

4.4.2 布尔表达式的翻译

4.5 控制语句的翻译

4.5.1 条件语句if的翻译

4.5.2 条件循环语句while的翻译

4.5.3 三种基本控制结构的翻译

4.5.4 多分支控制语句case的翻译

4.5.5 语句标号和转移语句的翻译

4.6 数组元素的翻译

4.6.1 数组元素的地址计算及中间代码形式

4.6.2 赋值语句中数组元素的翻译

4.6.3 数组元素翻译示例

4.7 过程或函数调用语句的翻译

4.7.1 过程或函数调用的方法

4.7.2 过程或函数调用语句的四元式生成

4.8 说明语句的翻译

4.8.1 变量说明的翻译

4.8.2 数组说明的翻译

4.9 递归下降语法制导翻译方法简介

习题四

第五章 代码优化

5.1 局部优化

5.1.1 基本块的划分方法

5.1.2 基本块的DAG方法

5.1.3 用DAG进行基本块的优化处理

5.1.4 DAG构造算法的进一步讨论

5.2 循环优化

5.2.1 程序流图与循环

5.2.2 循环的查找

5.2.3 循环优化

*5.3 全局优化概述

5.3.1 到达一定值与引用一定值链

<<编译原理教程>>

5.3.2 定值—引用链(du链)

5.3.3 写传播

*5.4 代码优化示例

习题五

第六章 目标程序运行时存储空间的组织

6.1 静态存储分配

6.2 简单的栈式存储分配

6.2.1 栈式存储分配与活动记录

6.2.2 过程的执行

6.3 嵌套过程语言的栈式实现

6.3.1 嵌套层次显示(DISPLAY)表和活动记录

6.3.2 嵌套过程的执行

6.3.3 访问非局部名的另一种实现方法

6.4 堆式动态存储分配

6.4.1 堆式存储的概念

6.4.2 堆式存储的管理方法

*6.5 参数传递补遗

6.5.1 参数传递的方法

6.5.2 不同参数传递方法比较

习题六

第七章 目标代码生成

7.1 简单代码生成器

7.1.1 待用信息与活跃信息

7.1.2 代码生成算法

7.1.3 寄存器分配

7.1.4 源程序到目标代码生成示例

*7.2 汇编指令到机器代码翻译概述

习题七

第八章 符号表与错误处理

8.1 符号表

8.1.1 符号表的作用

8.1.2 符号表的组织

8.1.3 分程序结构语言符号表建立

8.1.4 非分程序结构语言符号表建立

8.1.5 常用符号表结构

8.1.6 符号表内容

8.2 错误处理

8.2.1 语法错误校正

8.2.2 语义错误校正

习题八

附录1 8086/8088指令码汇总表

附录2 8086/8088指令编码空间表

参考文献

章节摘录

LR分析器的推理过程是依据“历史”来展望“未来”，因此LR分析器同样具有智能性，但是这种智能却是有限的。

作为归约过程的“历史”材料的积累虽不困难（已保存于栈中），但是“展望”材料的汇集却很不容易。

因为根据历史推测未来，即使是推测未来的一个符号，也常常存在着非常多的不同可能性，所以在把“历史”和“展望”材料综合在一起时，复杂性就大为增加了。

因此，具体实现LR分析器的功能时通常采用某种限制性措施，尽可能使出现的状态减少，只有这样才能提高效率并易于实现。

如简单LR——极有使用价值的SLR(1)方法，它只在栈中保留已扫描过的那段输入符号的部分信息，即“历史”，并根据这些信息和未来的一串符号决定下一步的操作。

SLR(1)的这种做法使得状态数大为减少，因此可以高效率地实现。

我们可以将上述方法和思想运用于人工智能领域。

通过分析可以发现，人工智能的推理按目标可分为两类：一类是对未知目标的推理；一类是对已知目标的推理。

对事先无法知道的目标的推理一般采用常规的智能搜索方法——试探法或穷举法；而对已知目标的推理，虽然仍可采用试探法或穷举法，但能否采用其它更加简单有效的方法呢？

LR分析器中的“历史”类同于人工智能中已搜索过的路径，而LR分析器对未来的“展望”恰好就是对已知目标的“展望”。

LR分析器是在“目标已知”这个特定条件下对人工智能常规搜索方法的一种简化。

由于LR分析器推理的目标确定，故其效率远远高于常规的人工智能搜索方法。

对那些事先可以分析出目标的推理问题，如智能化教学系统、智能化管理系统以及智能控制机床等，都可以采用LR分析器方法。

LR分析器是一种基于目标的智能推理方法，它适用于目标确定的智能化推理问题。

.....

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>