

目录

| | | |
|----------|--------------------|----------|
| 1 | 关于编译程序的基本概念 | 1 |
| 1.1 | 什么是编译程序？ | 2 |
| 1.2 | 编译程序结构 | 3 |
| 1.3 | 编译程序的实现机制 | 5 |
| 1.4 | 编译过程实例 | 5 |
| 1.5 | 本章小结 | 8 |
| 2 | 形式语言基础 | 9 |
| 2.1 | 形式语言是符号串集合 | 9 |
| 2.1.1 | 符号串（集合）的运算 | 10 |
| 2.1.2 | 符号串集合的文法描述 | 12 |
| 2.2 | 形式语言由文法定义的 | 14 |
| 2.2.1 | 什么是文法？ | 14 |
| 2.2.2 | 文法是怎样定义语言的？ | 14 |
| 2.3 | 主要语法成分的定义 | 16 |
| 2.3.1 | 文法的运算问题 | 16 |
| 2.3.2 | 句型、句子和语法树 | 18 |
| 2.3.3 | 短语、简单短语和句柄 | 20 |
| 2.4 | 两种特性文法 | 21 |
| 2.4.1 | 递归文法 | 22 |
| 2.4.2 | 二义性文法 | 22 |
| 2.5 | 文法的等价变换 | 23 |
| 2.5.1 | 文法的等价性 | 23 |
| 2.5.2 | 文法变换方法 | 23 |
| 2.5.3 | 文法变换方法 1 | 24 |

| | | |
|----------|----------------|-----------|
| 2.6 | 形式语言的分类 | 28 |
| 3 | 自动机 | 29 |
| 3.1 | 正规语言及其描述方法 | 29 |
| 3.1.1 | 正规语言的正规式表示法 | 30 |
| 3.1.2 | 正规语言的有限自动机表示法 | 31 |
| 3.2 | 有限自动机的定义与分类 | 33 |
| 3.2.1 | 有限自动机的定义 | 33 |
| 3.2.2 | 有限自动机怎样描述语言 | 33 |
| 3.2.3 | 有限自动机的两种表现形式 | 35 |
| 3.2.4 | 有限自动机的分类 | 38 |
| 3.3 | 有限自动机的等价变换 | 39 |
| 3.3.1 | 有限自动机的确定化 | 39 |
| 3.3.2 | 有限自动机的最小化 | 43 |
| 3.4 | 正规语言描述方法间的相互转换 | 50 |
| 3.5 | 有限状态自动机的实现问题 | 54 |
| 3.5.1 | 控制程序设计 | 55 |
| 3.5.2 | 变换表存贮结构设计 | 56 |
| 4 | 词法分析 | 59 |
| 4.1 | 词法分析的基本概念 | 59 |
| 4.1.1 | 单词的分类与识别 | 59 |
| 4.1.2 | 单词的机内表示 | 61 |
| 4.2 | 词法分析程序的设计 | 62 |
| 4.2.1 | 词法分析程序功能划分 | 62 |
| 4.2.2 | 一个简单词法分析器的实现 | 63 |
| 4.2.3 | 词法分析示例 | 66 |
| 4.3 | 算数常数处理机设计 | 67 |
| 4.3.1 | 识别器设计 | 68 |
| 4.3.2 | 翻译器设计 | 68 |
| 5 | 语法分析 | 73 |
| 5.1 | 语法分析的基本概念 | 73 |
| 5.2 | 递归子程序法 | 75 |

| | | |
|--------|-----------------------|-----|
| 5.2.1 | 递归子程序法示例 | 75 |
| 5.2.2 | 递归子程序构造方法 | 77 |
| 5.2.3 | 递归子程序法适用范围 | 78 |
| 5.3 | LL(1) 文法定义 | 80 |
| 5.4 | LL(1) 分析法的完整流程 | 81 |
| 5.4.1 | 抽象的流程表示 | 86 |
| 5.5 | LL(1) 文法及其判定 | 87 |
| 5.5.1 | 首符号集合、后继符集合与选择符集合 | 87 |
| 5.5.2 | LL(1) 文法及其判定 | 89 |
| 5.6 | LL(1) 分析器设计 (实现) | 91 |
| 5.6.1 | LL(1) 分析表的构造 | 91 |
| 5.7 | LR() 分析法的介绍 | 93 |
| 5.7.1 | LR() 分析法的“统治地位” | 93 |
| 5.7.2 | LR() 分析法的定义 | 94 |
| 5.8 | LR(0) 分析器设计 | 97 |
| 5.8.1 | LR(0) 文法及其判定 | 98 |
| 5.8.2 | LR(0) 分析表构造 | 98 |
| 5.8.3 | LR(0) 控制程序设计 | 99 |
| 5.9 | 项目集和可归约前缀图 | 99 |
| 5.9.1 | 扩展文法 | 100 |
| 5.9.2 | 由扩展文法构造可归约前缀图 (句柄识别器) | 100 |
| 5.9.3 | 由可归约前缀图构造 LR(0) 分析表 | 101 |
| 5.9.4 | LR(0) 分析法过程示例 | 101 |
| 5.9.5 | LR(0) 分析法实例 | 102 |
| 5.10 | LR(0) 分析法的扩展 | 103 |
| 5.11 | SLR(1) 分析法的扩展 | 105 |
| 5.11.1 | 扩展文法 | 106 |
| 5.11.2 | 构造可归约前缀图 | 106 |
| 5.12 | 简单优先分析法基本概念 | 108 |
| 5.12.1 | 什么是简单优先分析法 | 108 |
| 5.12.2 | 简单优先分析过程示例 | 108 |
| 5.12.3 | 文法符号之间的优先关系 | 109 |
| 5.13 | 简单优先分析器设计 | 113 |

| | | |
|----------|-------------------------|------------|
| 5.13.1 | 简单优先文法及其判定 | 114 |
| 5.13.2 | 简单优先分析矩阵分析表构造 | 114 |
| 5.13.3 | 简单优先控制程序设计 | 115 |
| 5.14 | 算符优先分析 | 115 |
| 5.14.1 | 算符文法 | 115 |
| 5.14.2 | 头符号集合和尾符号集合 | 116 |
| 5.14.3 | 算符优先关系定义 | 116 |
| 5.14.4 | 算符优先文法 | 116 |
| 6 | 符号表 | 119 |
| 6.1 | 符号表的地位和作用 | 119 |
| 6.1.1 | 符号表的定义 | 119 |
| 6.1.2 | 标识符的四种语义信息 | 120 |
| 6.1.3 | 符号表的基本功能 | 120 |
| 6.2 | 符号表的组织与管理 | 121 |
| 6.2.1 | 符号表的工作原理 | 121 |
| 6.2.2 | 符号表的查询、访问方式 | 121 |
| 6.2.3 | 符号表的维护、管理方式 | 121 |
| 6.3 | 符号表的结构设计 | 122 |
| 6.3.1 | 符号表总表 (SYNBL) | 123 |
| 6.3.2 | 类型表 (TYPEL) | 124 |
| 6.3.3 | 数组表 (AINFL) | 124 |
| 6.3.4 | 结构表 (RINFL) | 125 |
| 6.3.5 | 函数表 (PFINFL)——过程或函数语义信息 | 125 |
| 6.3.6 | 其他表 (...) | 126 |
| 6.4 | 符号表的构造过程示例 | 126 |
| 6.5 | 运行时刻存储分配 | 131 |
| 6.5.1 | 标识符值单元分配 | 131 |
| 6.5.2 | 活动记录 | 132 |
| 6.5.3 | 简单的栈式存储分配 | 134 |
| 6.5.4 | 嵌套过程语言的栈式存储分配 | 137 |
| 7 | 中间代码生成 | 145 |
| 7.1 | 导入 | 145 |

| | | |
|-------|-----------------------------------|-----|
| 7.1.1 | 为什么要研究语义分析 | 145 |
| 7.1.2 | 为什么需要生成中间代码 | 145 |
| 7.2 | 中间代码生成 | 146 |
| 7.2.1 | 常用的中间代码形式 | 146 |
| 7.2.2 | 各种语法成分的中间代码设计 | 147 |
| 7.3 | 中间代码翻译算法 | 155 |
| 7.3.1 | 属性文法 | 155 |
| 7.3.2 | 语法制导翻译技术 | 156 |
| 7.3.3 | 四元式翻译文法设计扩展 1 | 159 |
| 7.4 | 中间代码翻译的实现 | 163 |
| 7.4.1 | 递归子程序翻译法 | 163 |
| 7.4.2 | LL(1) 翻译法 | 165 |
| 7.4.3 | LR() 翻译法 | 166 |
| 7.4.4 | 算符优先翻译法 | 168 |
| 7.4.5 | 翻译文法的变换问题 | 169 |
| 8 | 优化处理 | 171 |
| 8.1 | 优化的分类 | 171 |
| 8.1.1 | 与机器无关的优化 | 171 |
| 8.1.2 | 与机器有关的优化 | 171 |
| 8.2 | 常见的几种局部优化方法 | 172 |
| 8.2.1 | 常值表达式节省（常数合并） | 172 |
| 8.2.2 | 公共表达式节省（删除多余运算） | 172 |
| 8.2.3 | 删除无用赋值 | 172 |
| 8.2.4 | 不变表达式外提（循环优化之一：把循环不变运算提到循环外） | 172 |
| 8.2.5 | 消减运算强度（循环优化之二：把运算强度大的运算换算成强度小的运算） | 173 |
| 8.3 | 局部优化算法探讨 | 173 |
| 8.3.1 | 基本块划分算法 | 173 |
| 8.3.2 | 局部优化示例 | 175 |
| 8.4 | 基于 DAG 的局部优化方法 | 176 |
| 8.4.1 | 四元式序列的 DAG 表示 | 177 |
| 8.4.2 | 基于 DAG 的局部优化算法 | 180 |
| 9 | 目标代码及其生成 | 185 |

| | | |
|-------|---------------------------|-----|
| 9.1 | 目标代码生成的基本问题 | 185 |
| 9.1.1 | 目标代码选择 | 185 |
| 9.1.2 | 变量的活跃信息 | 189 |
| 9.1.3 | 寄存器的分配问题 | 192 |
| 9.1.4 | 目标代码生成问题 | 195 |
| 9.2 | 目标代码生成算法设计 | 198 |
| 9.2.1 | 目标代码生成要点和生成环境 | 202 |
| 9.2.2 | 表达式四元式目标代码生成算法 | 203 |
| 9.2.3 | 赋值四元式目标代码生成算法 | 203 |
| 9.2.4 | 条件语句四元式目标代码生成算法 | 204 |
| 9.2.5 | 循环语句四元式目标代码生成算法 | 206 |
| 9.3 | 一个简单代码生成器的实现 | 208 |