编译原理

第3版

高等学校教材

计算机系列教材

BIANYIYUANLI

普通高等教育“十五”国家级重点教材

主编 蒋立源 康慕宁

主审 冯博琴

西北工业大学出版社

普通高等教育“十五”国家级重点教材

编 译 原 理

（第3版）

主编 蒋立源 康慕宁
主审 冯博琴
编者 蒋立源 康慕宁 吴 健 邓正宏
林 奕 张延园 薛 贺 郑玉山

西北工业大学出版社

2005年1月 西安

【内容简介】 本书第3版系普通高等教育"十五"国家级重点教材，旨在系统地介绍编译系统的结构、工作流程以及编译程序的设计原理和实现技术。全书共11章，内容包括语言及文法的基本知识、词法分析、语法分析、语义分析及中间代码生成、符号表组织、运行时的存储组织与分配、代码优化、目标代码生成以及面向对象语言的编译技术等。在内容的组织上，本书将编译的基本理论和具体的实现技术有机地结合起来，既准确清楚地阐述相关的概念和原理，又给出典型的实现程序；同时，对目前颇为流行且使用效果良好的分析器自动生成工具（如LEX，LLama，YACC，OCCS及GCC等）的功能和使用方法也作了详细的介绍。本书力求反映编译技术方面的最新成果。书中所列的分析算法、驱动程序及语义动作等，全部用C语言描述，各章之后附有大量的习题和上机实习题目。本书文字简洁易懂，内容循序渐进、深入浅出，便于自学。

本书可作为大学计算机类本科专业的教材，也可作为计算机软件科技人员的参考书。

与本书配套，西北工业大学出版社已出版了《编译原理常见题型解析及模拟题》（世纪精版）一书。该书是为了帮助学生加深对课程基本内容的理解，提高解题能力及满足考研复习需要而编写的，并对本书中的重点习题做了详尽的解答。

图书在版编目(CIP)数据

编译原理/蒋立源，康慕宁主编．—3版．—西安：西北工业大学出版社，2005.1
ISBN 7-5612-1870-2

Ⅰ．编… Ⅱ．①蒋… ②康… Ⅲ．编译程序-程序设计-高等学校-教材 Ⅳ．TP314

中国版本图书馆CIP数据核字(2004)第131769号

出版发行：西北工业大学出版社
通讯地址：西安市友谊西路127号 邮编：710072 电话：(029)88493844
网　　址：www.nwpup.com
印 刷 者：陕西百花印刷有限责任公司
开　　本：787 mm×1 092 mm 1/16
印　　张：25.5
字　　数：610千字
版　　次：2005年1月第3版 2005年1月第13次印刷
印　　数：68 001～76 000册
定　　价：32.00元

第3版前言

本书第2版(普通高等教育"九五"国家级重点教材)自1999年9月出版以来,已陆续重印过5次。在此期间,我们曾对原书进行过多次勘校,对所发现的舛误一一进行了订正,并对书中一些定义、定理的表述方式以及某些章节的词句、符号和程序进行了修改,以便于读者学习。

近来,为了准备普通高等教育"十五"国家级重点教材的书稿,我们又对本书进行了一次较全面的修订,其中较重大的修改有下列几点:

1. 考虑到构造编译程序簇(Compiler Collection)的技术已经广泛在国内外成功应用,因此在本书的一些章节中,我们对GNU CC的结构、工作流程以及相关知识进行了简要的介绍,以扩大学生的知识面。

2. 由于面向对象的程序设计语言已经成为国际上的主流编程语言,故作为讲述"语言处理"知识的编译原理教材,对具有此种特征的语言编译技术也应有所反映。几年前我们就曾有意于此,但由于参考资料收集不易,且对内容的组织也尚未考虑成熟,故只得暂付阙如。近两年来,我们又对此课题进行了较深入的学习、研究,在此基础上由林奕博士对相关内容进行了整理、组织,遂写成本书第11章——面向对象语言的编译技术。由于我们的视野和学术水平有限,上述工作恐不能完全满足教学工作需要,尚祈读者批评指教。

3. 对"抽象语法树"和"寄存器转换语言"的知识作了概括的介绍,因为它们在构造编译器开发工具中有重要的作用。

4. 由于"优先分析法"目前在设计编译器中的重要性已相对降低,因此,为突出重点和压缩篇幅,我们将有关构造优先矩阵的繁复算法予以删节。

此外,其他小的修改还有多处,不再一一列举。

本书第3版由蒋立源和康慕宁担任主编。第1～4章和第6～10章由蒋立源编写(其中,8.1节和8.2节由吴健编写);第5章的5.1～5.9节由康慕宁编写,5.10节由张延园编写;第11章由林奕编写;薛贺负责习题和上机实习题的选编;康慕宁、邓正宏和郑玉山负责本书

电子版的创意、改编和制作。最后由蒋立源对全书进行统稿。

在本书第3版即将付梓之际，编者谨向教育部计算机科学与技术教学指导委员会副主任、西安交通大学冯博琴教授致以深切的谢意，他不仅一直关心和支持我们的工作，还担任本书1～3版的主审，对各版的书稿均进行过仔细的审阅，提出了许多宝贵、中肯的意见。在本书立项、编写、出版和报奖的过程中，我们还得到中国科学院沈绪榜院士，中国工程院崔俊芝院士，西北大学郝克刚、周明全和耿国华教授，西安电子科技大学陈家正、徐甲同、龚杰民和刘坚教授，西安理工大学胡元义教授以及西北工业大学教务处、出版社和计算机学院的许多领导和教授的指导和帮助，在此一并表示衷心的谢忱。

陕西省老年书画学会常务副会长李峰山教授欣然为本书题写了书名。我们谨向李老先生表示崇高的敬意。

最后，我们还要特别向本书的许多读者表示感谢。他们不惮烦劳，对本书内容进行仔细的推敲，并将所发现的问题进行反馈，使我们能及时更正书中存在的疏漏，衷心期望能继续得到读者的帮助和支持。

蒋立源谨识

2004年8月于西安

第 1 版前言

“编译原理”是计算机类专业特别是计算机软件专业的一门重要专业课。设置本课程的目的，在于系统地向学生讲述编译系统的结构、工作流程及编译程序各组成部分的设计原理和实现技术，使学生通过学习本课程，既掌握编译理论和方法方面的基本知识，也具有设计、实现、分析和维护编译程序等方面的初步能力。

根据上述要求，在航空高等学校第二教学指导委员会的组织下，我们参照中国计算机教育研究会向全国推荐的《编译原理教学大纲》，并结合编者多年来讲授本课程的教学实践经验，编写了本书。

全书共分为 10 章，第 1 章对编译过程、编译程序的逻辑结构及编译程序各组成部分的主要功能进行了概括的说明。第 2 章介绍前后文无关文法和语言的基本知识，它为学习后续各章奠定了理论基础。第 3 章以正规文法、正规式和有限自动机为工具，讨论了词法分析程序的设计原理。第 4 章讲述了语法分析程序的设计技术，其中既介绍了传统的算符优先分析方法，也介绍了目前十分流行的递归下降分析法及 LR 分析法，同时还对一种行之有效的语法分析程序自动生成工具 YACC 的使用方法进行了简要的说明。第 5 章以语法制导翻译为模式，介绍了将程序设计语言常见的语法成分翻译为中间代码的方法。第 6 章至第 10 章分别讨论了符号表的构造、目标程序运行时的存储组织与分配、代码优化、目标代码生成及源程序的查错与改错等问题。各章之后均附有一定数量的习题供读者选做。

编译原理是一门理论性和实践性都比较强的课程。在本书的编写过程中，我们力图将其中的基本概念、基本原理和实现方法的思路阐述清楚，因为它们不仅是构造编译程序的依据，而且对开发其它系统软件和应用软件也是很有用的。同时，为了培养学生的实际工作能力，我们在有关的章节之后，还列出了一些上机实习题目，学生通过完成这些作业可进一步加深对课堂教学内容的理解。

本书系航空教材编审委员会 1991—1992 年教材选题计划所列的部委规划教材，可作为计算机类各专业编译原理课的教科书（课堂讲授约需 72～80 学时，此外还应有 25～30 学时的上机时间），也可供有关工程技术人员参考。

本书由西北工业大学蒋立源主编，参加编写工作的还有张延园、石志强、叶军和胡滨等同志。

西安交通大学冯博琴教授对本书进行了仔细的审阅，提出了许多宝贵的意见。在本书编写过程中，我们还得到了西北工业大学出版社的许多同志以及计算机系的张遵濂、徐秋元、韩兆轩、白中英、胡正国、赵政文等教授的支持、关心和帮助，在此一并表示衷心的谢忱。

由于我们学力有限，书中定有不妥之处，恳请读者批评指正。

编　者

1992 年 3 月于西安

目　录

第 1 章　绪论 …… 1

1.1　编译过程概述 …… 3
1.2　编译程序的逻辑结构 …… 4
1.2.1　词法分析程序 …… 5
1.2.2　语法分析程序 …… 6
1.2.3　语义分析程序 …… 6
1.2.4　中间代码生成 …… 7
1.2.5　代码优化程序 …… 7
1.2.6　目标代码生成程序 …… 8
1.2.7　错误检查和处理程序 …… 9
1.2.8　信息表管理程序 …… 10
1.3　编译程序的组织 …… 11
习题 …… 12

第 2 章　前后文无关文法和语言 …… 14

2.1　文法及语言的表示 …… 14
2.2　文法和语言的定义 …… 15
2.2.1　基本概念和术语 …… 16
2.2.2　文法和语言的形式定义 …… 17
2.3　句型的分析 …… 23
2.3.1　规范推导和规范归约 …… 23
2.3.2　语法树和二义性 …… 25
2.3.3　短语和句柄 …… 29
2.4　文法的化简和改造 …… 31
2.4.1　无用符号和无用产生式的删除 …… 31
2.4.2　ε-产生式的消除 …… 33
2.4.3　单产生式的消除 …… 35
2.5　文法和语言的 Chomsky 分类 …… 36

习题 …… 38

第3章 词法分析及词法分析程序 …… 42

3.1 设计扫描器时应考虑的几个问题 …… 42
3.1.1 词法分析阶段的必要性 …… 42
3.1.2 单词符号的内部表示 …… 43
3.1.3 识别标识符的若干约定和策略 …… 44
3.1.4 源程序的输入及预处理 …… 46
3.2 正规文法和状态转换图 …… 49
3.2.1 由正规文法构造状态转换图 …… 49
3.2.2 状态转换图的一种实现——状态矩阵法 …… 53
3.3 有限自动机 …… 59
3.3.1 确定的有限自动机 …… 59
3.3.2 非确定的有限自动机 …… 60
3.3.3 NFA与DFA的等价性 …… 62
3.3.4 具有ε动作的FA …… 64
3.3.5 具有ε动作的NFA的确定化——子集法 …… 66
3.3.6 DFA状态数的最小化 …… 69
3.4 正规表达式与正规集 …… 71
3.4.1 正规表达式与正规集的定义 …… 72
3.4.2 由正规文法构造相应的正规式 …… 73
3.4.3 由正规式构造FA——Thompson法 …… 76
3.5 词法分析程序的实现 …… 78
3.5.1 词法分析程序的编写 …… 79
3.5.2 词法分析程序的自动生成 …… 82
习题 …… 98
上机实习题 …… 104

第4章 语法分析和语法分析程序 …… 106

4.1 自顶向下的语法分析 …… 107
4.1.1 消除文法的左递归 …… 108
4.1.2 回溯的消除及LL(1)文法 …… 111
4.1.3 递归下降分析法 …… 113
4.1.4 预测分析法 …… 119
4.1.5 某些非LL(1)文法的改造 …… 124
4.2 自底向上的语法分析 …… 126
4.2.1 简单优先分析法 …… 127
4.2.2 算符优先分析法 …… 133
4.2.3 优先函数 …… 138

4.2.4 LR 分析法 …… 144
习题 …… 172
上机实习题 …… 178

第 5 章 语法制导翻译及中间代码生成 …… 181

5.1 引言 …… 181
5.2 属性文法与属性翻译文法 …… 183
5.2.1 语义属性与属性文法 …… 184
5.2.2 属性翻译文法 …… 187
5.3 常见中间语言概述 …… 192
5.3.1 逆波兰表示 …… 192
5.3.2 四元式和三元式 …… 194
5.3.3 其它表示法 …… 197
5.4 简单算术表达式和赋值语句的翻译 …… 198
5.5 布尔表达式的翻译 …… 200
5.6 程序流程控制语句的翻译 …… 205
5.6.1 常见控制结构的翻译 …… 205
5.6.2 FOR 循环语句的翻译 …… 211
5.6.3 语句标号及 GOTO 语句的翻译 …… 214
5.6.4 情况语句的翻译 …… 217
5.7 含数组元素的算术表达式及赋值语句的翻译 …… 219
5.7.1 下标变量地址的计算 …… 220
5.7.2 含有下标变量的赋值语句的翻译 …… 222
5.8 过程说明和过程调用的翻译 …… 225
5.8.1 过程说明的翻译 …… 225
5.8.2 实参和形参间的信息传递 …… 226
5.8.3 过程语句的翻译 …… 228
5.8.4 关于形实结合的进一步讨论 …… 230
5.9 说明语句的翻译 …… 231
5.9.1 类型说明(变量及数组定义)语句的翻译 …… 231
5.9.2 数据类型定义语句的翻译 …… 234
5.10 语法分析程序的自动生成工具 …… 237
5.10.1 LALR(1)分析器的自动生成工具——YACC 和 OCCS …… 237
5.10.2 LL(1)语法分析程序自动生成工具 LLama 简介 …… 248
5.10.3 LLGen 简介 …… 249
5.10.4 GCC 概述 …… 251
习题 …… 253
上机实习题 …… 255

第6章 符号表 …… 257

6.1 符号表的组织 …… 257
6.2 分程序结构语言符号表的建立 …… 261
6.3 非分程序结构语言符号表的建立 …… 266
习题 …… 268

第7章 运行时的存储组织与分配 …… 270

7.1 存储组织 …… 271
7.1.1 运行时内存的划分 …… 271
7.1.2 活动记录 …… 271
7.2 运行时的分配策略 …… 272
7.2.1 静态分配 …… 273
7.2.2 栈式分配 …… 276
7.2.3 堆式分配 …… 278
习题 …… 281

第8章 代码优化 …… 284

8.1 语法制导翻译阶段的优化 …… 284
8.2 线性窥孔优化 …… 285
8.2.1 强度削弱 …… 286
8.2.2 常数合并和常数传播 …… 287
8.2.3 无用变量与无用代码的删除 …… 288
8.2.4 窥孔优化实例 …… 291
8.3 基于结构信息的优化 …… 293
8.3.1 基本块及其优化 …… 294
8.3.2 数据流分析方法 …… 300
8.3.3 循环优化 …… 310
习题 …… 326
上机实习题 …… 331

第9章 目标代码生成 …… 332

9.1 目标代码的形式 …… 332
9.2 一种假想的计算机模型 …… 334
9.3 一种代码生成程序的雏型 …… 337
9.3.1 待用信息 …… 337
9.3.2 寄存器描述符与地址描述符 …… 338
9.3.3 生成目标代码的算法 …… 339
9.4 DAG的代码生成 …… 342

9.5 全局寄存器分配 …… 344
习题 …… 348
上机实习题 …… 349

第 10 章 查错与改错 …… 350

10.1 语法错误的校正 …… 351
10.1.1 单词错误的校正 …… 351
10.1.2 自顶向下分析中的错误校正 …… 352
10.1.3 自底向上分析中的错误校正 …… 355
10.2 语义错误的校正 …… 357
10.2.1 遏止株连信息 …… 358
10.2.2 遏止重复信息 …… 358
习题 …… 359

第 11 章 面向对象语言的编译技术 …… 360

11.1 引言 …… 360
11.1.1 面向对象程序设计语言 …… 361
11.1.2 面向对象语言编译器的一些特点 …… 361
11.2 类和对象的基本特征 …… 362
11.3 类的基本定义 …… 366
11.3.1 基本类声明的语法定义 …… 366
11.3.2 基本类声明的抽象语法树表示 …… 367
11.3.3 基本类声明的处理 …… 369
11.4 面向对象程序设计语言的类型系统 …… 377
11.4.1 面向对象程序设计语言对类型系统的影响 …… 377
11.4.2 文法的扩展 …… 378
11.4.3 重载的处理 …… 378
11.4.4 继承的处理 …… 382
习题 …… 388

参考文献 …… 389

第1章 绪 论

程序设计语言是用来编写程序的工具，可分为两大类。第一类称为低级语言，包括机器语言、汇编语言以及其它面向机器的程序设计语言。这类语言对计算机的依赖性强、直观性差、编写程序的工作量大，只有对相应计算机的结构比较熟悉，且经过一定训练的程序人员才能较好地使用。第二类称为高级语言，有几百种之多，但除了一些专用语言之外，得到广泛运用的只有其中少数几种，如 FORTRAN，PASCAL，COBOL，C，ADA，JAVA 等。高级语言不论在算法描述的能力上，还是在编写和调试程序的效率上，都远比低级语言优越。

然而，计算机硬件只懂自己的指令系统，即只能直接执行相应机器语言格式的代码程序，而不能直接执行用高级语言或汇编语言编写的程序。因此，要在计算机上实现除机器语言之外的任一程序设计语言，就首先应使此种语言为计算机所“理解”。解决这一问题的方法有两种：一种是对程序进行翻译；另一种是对程序进行解释。

所谓翻译，是指在计算机中放置一个能由计算机直接执行的翻译程序，它以某一种程序设计语言（源语言）所编写的程序（源程序）作为翻译或加工的对象，当计算机执行翻译程序时，就将它翻译为与之等价的另一种语言（目标语言）的程序（目标程序）。“源”和“目标”这两个术语，总是相对于一类特定的翻译程序和翻译过程而言的。如果一个翻译程序的源语言是某种高级语言，其目标语言是相应于某一计算机的汇编语言或机器语言，则称这种翻译程序为编译程序（或编译器）。汇编程序也是一种翻译程序，它的源语言和目标语言分别是相应的汇编语言和机器语言。

由此可见，欲按编译方式在计算机上执行用高级语言编写的程序，一般须经过两个阶段：第一阶段称为编译阶段，其任务是由编译程序将源程序编译为目标程序，若目标程序不是机器代码，而是汇编语言程序，则尚需汇编程序再行汇编为机器代码程序；第二阶段称为运行阶段，其任务是在目标计算机上执行编译阶段所得到的目标程序。在执行目标程序时，一般还应有一些子程序配合进行工作，例如：常见的数据格式转换子程序、标准函数计算子程序、浮点解释子程序、数组动态存储分配子程序、下标变量地址计算子程序等都属此类。这些子程序

组成一个子程序库，称为运行系统。运行子程序库中的各个子程序，大都按模块化的结构来编制。显然，库中的子程序愈丰富，各子程序的功能愈强，编译程序本身就愈简明紧凑。

编译程序与运行系统合称为编译系统。

源程序的编译（或汇编）和目标程序的执行不一定在同一种计算机上完成。当源程序由另一种计算机进行编译（或汇编）时，我们将此种编译（或汇编）称为交叉编译（或汇编）。

图 1－1 粗略地显示了按编译方式执行一个高级语言程序的主要步骤。

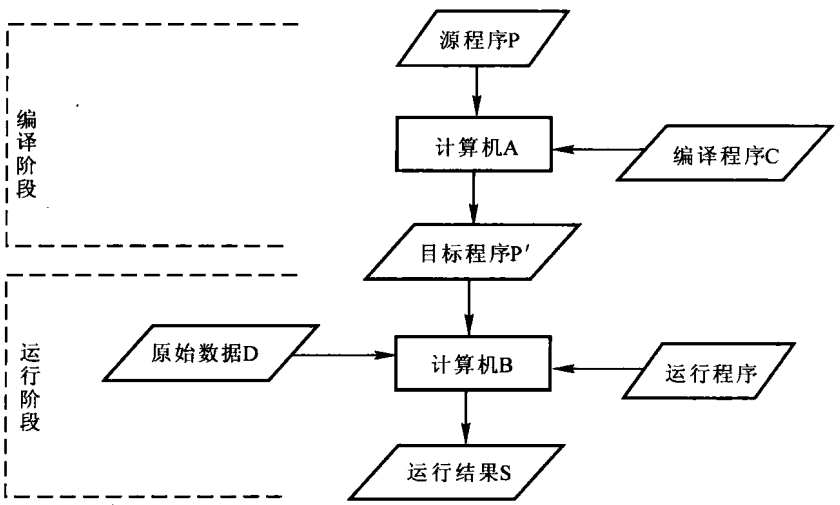

图 1－1 计算机执行高级语言程序的步骤

用高级语言编写的程序也可以通过解释程序来执行。解释程序也以源程序作为它的输入，它与编译程序的主要区别是在解释程序的执行过程中不产生目标程序，而是解释执行源程序本身。这种边翻译边执行的方式工作效率很低，但由于解释程序的结构比编译程序简单，且占用内存较少，在执行过程中也易于在源程序一级对程序进行修改，因此一些规模较小的语言，如BASIC，也常采用此种方式。然而就目前的情况来看，纯粹的解释程序并不多见，通常的做法是把编译和解释作某种程度的结合。例如，有的先将源程序翻译为某种易于进行解释执行的内部中间语言形式，然后再对此中间语言程序进行解释执行；有的甚至在进行上述翻译时，还对一部分出现比较频繁的结构（如算术表达式等）产生目标代码。在采取上述这些措施后，解释程序执行效率不高的缺陷将有可能得到部分弥补。

编译程序已成为现今任何计算机系统的最重要系统程序之一。本课程的目的，在于向读者介绍设计和构造编译程序的基本原理和基本方法，其中许多方法也同样适用于构造解释程序或汇编程序。事实上，任何一个熟悉编译程序构造的人，是不难旁通解释程序或汇编程序的工作原理和实现方法的。因此，限于篇幅，对于有关构造解释程序和汇编程序方面的问题，本书将不再涉及。

1.1 编译过程概述

编译程序的主要功能是把用高级语言编写的源程序翻译为等价的目标程序。既然编译过程是实现一种语言的翻译，那么我们可将编译程序的工作过程与通常外语资料的翻译过程进行类比，这将有助于更直观地了解一个编译程序一般应由哪些部分组成，以及各个组成部分应如何进行工作，等等。

抽象地看，任何一本外文资料都是由字母、标点符号（包括空格和其它符号）按相应语法规则所组成的字符串。因此，任何欲进行外文翻译的人，都应具备如下能力：

(1) 能认识外语的字母及标点符号；

(2) 能识别出文中的各个单词；

(3) 会查字典；

(4) 懂得此种外语的语法；

(5) 具有目标语言的修辞能力。

至于如何进行翻译，概括地讲无非是做两方面的工作：一是进行分析，二是进行综合。所谓分析，就是从第一行的第一个字母开始，依次阅读原文中的各个符号，逐个识别出原文中的各个单词，然后根据语法规则进行语法分析，即分析原文中如何由单词组成短语和句子，以及句子的种类特点等。此外，在识别单词和进行语法分析的过程中，还要不时地查阅字典，做语法正确性的检查，进行相应的语义分析，并做一些必要的信息记录工作等。所谓综合，就是根据上述分析所得到的信息，拟定译稿，进行修辞加工，最后写出译文。

类似地，编译程序在其工作过程中，也须做两方面的工作，即先分析源程序，然后再综合为目标程序。为了便于理解编译程序在此两方面应包括的工作环节，现将源程序的编译和外文资料的翻译这两过程的主要工作进行对比如表 1-1 所示。

表 1-1 翻译和编译工作的比较

	翻译外文书刊	编译源程序
分析	阅读原文	输入并扫视源程序
	识别单词	词法分析
	分析句子	语法分析
综合	修辞加工	代码优化
	写出译文	目标代码生成

尽管编译过程与外文书刊翻译的工作过程比较类似，但由于编译程序所翻译的毕竟不是自然语言，因此，就必然有其自身特有的一些工作。比如中间代码的产生，编译过程中信息表的构造与查询以及运行时存储空间的分配，对语法和语义错误进行必要的处理等杂务工作。诸如此类的工作还有很多，兹不一一列举。

总之，编译程序是计算机的一个十分复杂的系统程序。为便于构造或分析一个编译程序，宜将整个编译程序分解为若干个组成部分，每一部分都用一段相对独立的程序去完成整个编译过程的一部分功能。就一个典型的编译程序而论，一般都含有下面八个部分：

(1) 词法分析程序（也称为扫描器）；

(2) 语法分析程序（有时简称为分析器）；

(3) 语义分析程序；

(4) 中间代码生成程序;

(5) 代码优化程序;

(6) 目标代码生成程序;

(7) 错误检查和处理程序;

(8) 各种信息表格的管理程序。

在下一节中,我们将简要说明上述各个部分的功能,并指出如何将这八个部分组成一个完整的编译程序。

1.2 编译程序的逻辑结构

在 1.1 节中,我们已概括地介绍了编译程序的工作过程,并指出了一个典型的编译程序一般所包含的八个组成部分。图 1-2 表示了这八个部分间的控制流程和信息流程(分别用实线和虚线表示)。

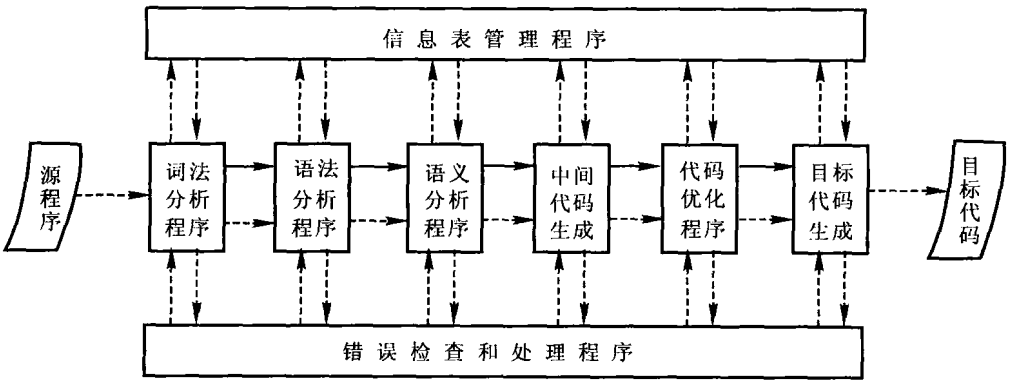

图 1-2 编译程序的逻辑结构

下面,我们用一个微型 PASCAL 语言 (PASCAL/M)所编写的程序为例,分别介绍这八个部分的功能,并分别给出每一部分对此程序进行加工处理可能得到的结果。我们假定此语言只有如下四种语句:

(1) PROGRAM 语句;

(2) 说明语句;

(3) BEGIN - END 语句;

(4) 赋值语句。

每个 PASCAL/M 程序都以一个 PROGRAM 语句开头,此语句中的标识符用来给程序命名;PROGRAM 语句之后是说明语句,用来指明程序中所出现的各个变量的数据类型(假定 PASCAL/M 中只有整型变量);在一系列说明语句之后,再跟以一个 BEGIN - END 语句,在保留字 BEGIN 和 END 之间,应有一个或多个赋值语句。程序 1-1 所示的 PASCAL/M 源程序,对于我们后面的讨论来说,是一个恰当的例子。

程序 1-1 一个 PASCAL 源程序 source

```
1  PROGRAM source;
2    {this little source program is used to
3     illustrate compiling procedure.}
4  VAR x,y,z: integer;
5        a: integer;
6  BEGIN
7   {this program has only four executable statements.}
8   x:=23+5;
9   z:=x DIV -3;
10  y:=z+18*3;
11  a:=x+(y-2) DIV 4;
12  END.
```

1.2.1 词法分析程序

作为编译程序的输入，源程序仅仅是一个长长的字符串，扫描器将把这种形式的源程序转换为便于编译程序其余部分进行处理的内部格式。扫描器的工作任务如下：

(1) 识别出源程序中的各个基本语法单位（也称为单词或语法符号）；

(2) 删除无用的空白字符、回车字符以及其它与输入介质相关的非实质性字符；

(3) 删除注释；

(4) 进行词法检查，报告所发现的错误。

现考虑程序 1-1 所示的源程序 source。扫描器依次查看缓冲区中源程序的各个字符，根据当前正查看之字符的种类，并参考扫描过程中前面所得到的信息，就能准确地判断当前正扫描的字符在源程序中所处的地位。概括而言，不外下述五种情况之一：

(1) 它是正处理的注释中的一个字符；

(2) 它是一个无用的空白字符；

(3) 它是下一个单词的首字符；

(4) 它是正识别的单词中的一个字符；

(5) 它是一个不合词法规则的字符，或是一个不属于本语言字符集的字符。

显然，如果扫描器根据上述不同的情况作不同的处置，并产生预定形式的输出，那么，它就能圆满地完成上面所提到的四项任务。

图 1-3 给出了扫描器对程序 source 进行处理后的一种可采用的输出形式。其中，程序里的各个单词已被一一识别出来，并用一个特定的标志符号“#”将相邻的两个单词加以分隔，程序中的非实质性符号已被全部删除。

```
# PROGRAM # source #; # VAR # x #, # y #, # z # : # integer # ; # a #:
# integer # ; # BEGIN # x # := # 23 # + # 5 #; # z # := # x # DIV # -
# 3 #; # y # := # z # + # 18 # * # 3 #; # a # := # x # + #(# y # -
# 2 #) # DIV # 4 # ; # END #. #
```

图 1-3 source 经扫描器处理后的一种输出