

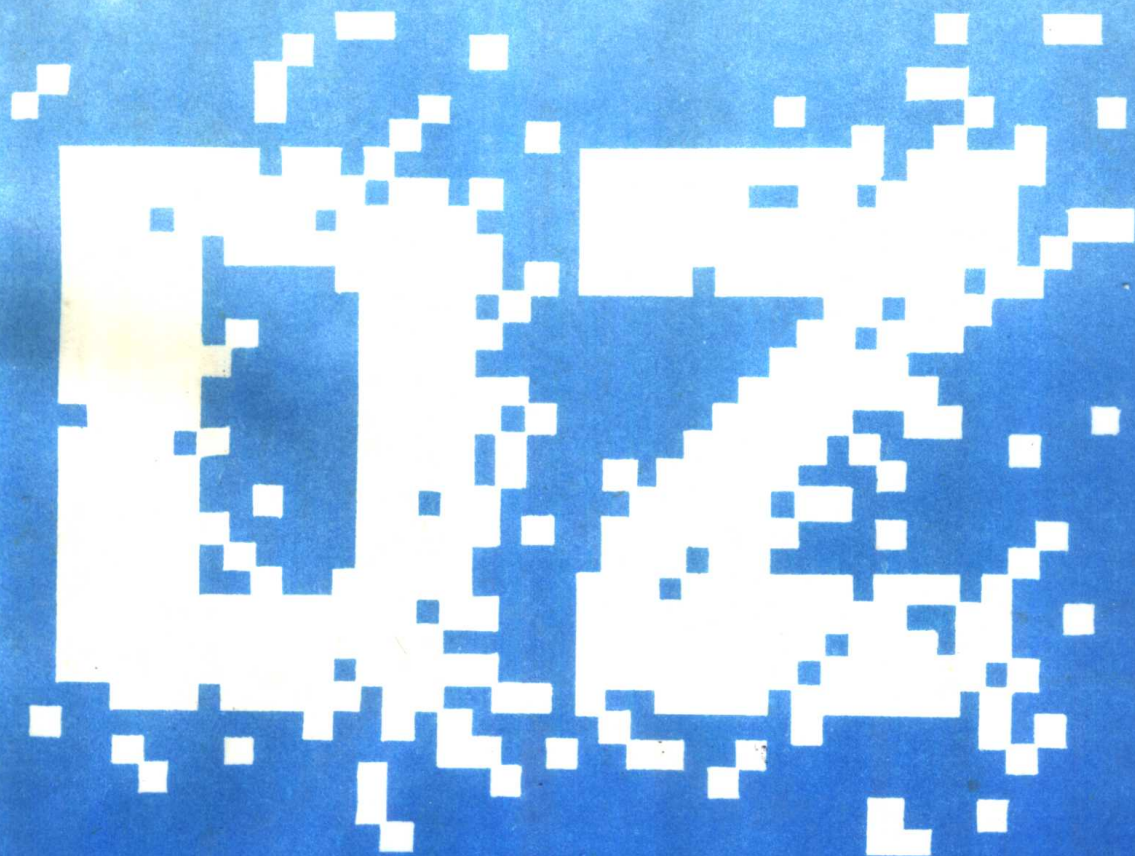


高等学校
工科电子类

规划教材

编译技术

钱焕延



东南大学出版社

编 译 技 术

钱 焕 延

东南大学出版社

(苏)新登字第 012 号

内 容 提 要

本书系统、全面地阐述了编译程序的基本结构、编译技术的一般理论和常用的有效方法与技术。其主要内容包括:文法和形式语言、自动机理论、词法分析、语法分析、语义分析、中间语言、代码生成、优化、存贮组织与分配和程序的查错与处理等。在分析方法中介绍了 LL(k)方法、运算符优先数法、状态矩阵法、递归子程序方法和 LR(k)方法等。

本书叙述由浅入深、循序渐进、通俗易懂。各章均附有习题,有关部分配有上机实习题。

本书系电子工业部八五规划教材,可供各类高等院校计算机专业作为教材,也可供从事计算机工作与研究的科技人员参考。

责任编辑 张克

编译技术

钱焕延

东南大学出版社出版发行

南京四牌楼 2 号 邮编 210018

江苏省新华书店经销

南京雄州印刷厂印刷

开本 787×1092 毫米 1/16 印张: 19.5 字数: 471 千

1995 年 3 月第 1 版 1997 年 3 月第 3 次印刷

印数: 4001—7000 册

ISBN 7—81023—206—1/TP·65

定价: 22.00 元

(凡因印装质量问题,可直接向承印厂调换)

出版说明

根据国务院关于高等学校教材工作的规定,我部承担了全国高等学校和中等专业学校工科电子类专业教材的编审、出版的组织工作。由于各有关院校及参与编审工作的广大教师共同努力,有关出版社的紧密配合,从1978—1990年,已编审、出版了三个轮次教材,及时供给高等学校和中等专业学校教学使用。

为了使工科电子类专业教材能更好地适应“三个面向”的需要,贯彻国家教委《高等教育“八五”期间教材建设规划纲要》的精神,“以全面提高教材质量水平为中心,保证重点教材,保持教材相对稳定,适当扩大教材品种,逐步完善教材配套”,作为“八五”期间工科电子类专业教材建设工作的指导思想,组织我部所属的九个高等学校教材编审委员会和四个中等专业学校专业教学指导委员会,在总结前三轮教材工作的基础上,根据教育形势的发展和教学改革的需要,制订了1991~1995年的“八五”(第四轮)教材编审出版规划。列入规划的,以主要专业主干课程教材及其辅助教材为主的教材约300多种。这批教材的评选推荐和编审工作,由各编委会或教学指导委员会组织进行。

这批教材的书稿,其一是从通过教学实践、师生反映较好的讲义中经院校推荐,由编审委员会(小组)评选择优产生出来的,其二是在认真遴选主编人的条件下进行约编的,其三是经过质量调查在前几轮组织编写出版的教材中修编的。广大编审者、各编审委员会(小组)、教学指导委员会和有关出版社,为保证教材的出版和提高教材的质量,作出了不懈的努力。

限于水平和经验,这批教材的编审、出版工作还可能有缺点和不足之处,希望使用教材的单位、广大教师和同学积极提出批评和建议,共同为不断提高工科电子类专业教材的质量而努力。

电子工业部电子类专业教材办公室

前 言

本教材系按电子工业部制定的工科电子类专业教材 1991~1995 年编审出版规划,由计算机教材编审委员会组织征稿并推荐出版。责任编委为福州大学吴文钰教授。

本教材由南京理工大学钱焕延编著,南京大学郑国梁教授担任主审。

本课程的参考学时数为 60~80 学时。其主要内容为:文法和形式语言、自动机理论、词法、语法、语义分析、中间语言、代码生成、优化、存贮组织与分配和程序的查错与处理等。本教材既重视最经典最广泛应用的基本编译技术,又力求反映近年来的一些最重要的新成果。在分析方法中,重点介绍 LL(k)方法、运算符优先数法、状态矩阵法、递归子程序方法和 LR(k)方法等。

编者根据多年的教学实践,在具体编写过程中力求由浅入深、循序渐近、通俗易懂。书中对各种编译方法技术适当配有相应的处理步骤或流程图,以便读者理解和接受。此外,各章均按序配有一定数量的习题,有关部分配有上机实习题,读者通过练习和上机实习调试,能进一步理解和掌握本书各章节的基本内容。

为使读者对编译程序的全貌有一个比较系统的了解,本书第 7 章以 PL/0 语言为例,对其编译的全过程运用递归子程序方法予以分块详细介绍。并将完整的 PL/0 编译程序附于书后,可供读者参阅或上机实习调试参考。

参加审阅工作的还有西安电子科技大学陈家正教授和福州大学何天牧教授,他们都为本书提出了许多宝贵的意见,这里表示诚挚的感谢。特别还要感谢南京理工大学张志凌同志的协作,她对本书的编写给予了巨大的关心和支持。

由于编者水平有限,书中难免还存在一些缺点和错误,殷切希望广大读者批评指正。

编 者

1994 年 10 月于南京

0598/02

目 录

1 概论	
1.1 程序设计语言	1
1.2 编译程序	3
1.3 源程序执行的途径	4
1.4 编译程序的结构	5
1.5 编译程序的开发	11
习题一	14
2 文法和形式语言	
2.1 符号和符号串	15
2.2 文法和语言	19
2.3 语法树和二义性	28
2.4 文法的实用限制	34
2.5 扩充的 BNF 表示法	43
2.6 文法和语言分类	44
2.7 正则表达式与正则集	49
习题二	51
3 自动机	
3.1 转换图	54
3.2 确定有限自动机(DFA)	55
3.3 不确定有限自动机(NFA)	58
3.4 ϵ -自动机	63
3.5 自动机的简化	64
3.6 语法图与自动机	67
3.7 下推自动机	72
习题三	75
4 符号表	
4.1 符号表及其应用	77
4.2 符号表的内容	79
4.3 存贮分配与符号表的组织	81
4.4 标识符的处理	89
4.5 符号表的结构和存取	90
习题四	95

5 词法分析	
5.1 引言	96
5.2 取单词	98
5.3 读标识符	101
5.4 取无正负号数	102
5.5 词法分析器的自动生成	103
习题五	108
6 语法和语义分析	
6.1 常用的终结符号集	110
6.2 句子的分析	112
6.3 虚拟机	116
6.4 递归子程序方法	119
6.5 LL(k)分析方法	124
6.6 运算符优先数法	129
6.7 状态矩阵法	147
习题六	153
7 递归子程序方法	
7.1 PL/0 程序设计语言	156
7.2 PL/0 处理机	159
7.3 代码生成形式	163
7.4 PL/0 编译程序的子程序	167
习题七	186
8 LR(k)分析方法	
8.1 LR 分析方法的逻辑结构及分析过程	187
8.2 LR(0)分析表的构造	194
8.3 SLR(1)分析表的构造	200
8.4 LR(1)分析表的构造	205
8.5 LALR(1)分析表的构造	211
习题八	216
9 中间语言	
9.1 逆波兰表示	219
9.2 四元式	222
9.3 三元式	225
9.4 树表示	228
习题九	229
10 优化	
10.1 引言	230
10.2 有关概念	230

10.3	优化处理概述	232
10.4	表达式的优化	235
10.5	循环优化	237
	习题十	241
11	语法制导翻译与代码生成	
11.1	语法制导翻译	243
11.2	表达式的翻译	244
11.3	布尔表达式的翻译	246
11.4	语句的语法制导翻译	247
11.5	过程调用语句的翻译	251
11.6	说明部分的翻译	254
11.7	属性文法技术	256
11.8	目标代码的生成	260
11.9	寄存器的分配	261
	习题十一	262
12	存贮组织与分配	
12.1	静态存贮分配	263
12.2	栈式存贮分配	265
12.3	堆式存贮分析	270
12.4	参数传递	270
	习题十二	273
13	程序的查错与处理	
13.1	引言	274
13.2	词法分析阶段的查错处理	275
13.3	语法分析阶段的查错处理	275
13.4	语义错误的处理	277
13.5	错误处理程序	277
	习题十三	278
附录 PL/0 编译程序		
附 I	PL/0 语言的文法规则	279
附 II	PL/0 编译程序的功能	280
附 III	PL/0 编译程序的有关过程及函数	280
附 IV	编译步骤	281
附 V	PL/0 编译程序及主要参数	282
附 VI	上机运行及说明	297
	参考文献	299

1 概 论

电子计算机的诞生是科学发展史上的一个里程碑,它开创了人类脑力劳动的自动化,丰富了人类精神财富和认识能力。科学技术迅猛发展的今天,计算机的发展水平和普及程度已成为衡量一个国家科学技术发展水平的一个重要标志,也是一个国家现代化程度的一个重要标志。

众所周知,计算机系统包括**硬件和软件**两大部分,软件的功能与质量很大程度上左右着整个计算机系统的功能。计算机系统没有好的软件就相当于一个人患聋哑病,软件在计算机和使用者之间架起了桥梁。软件大体上又分为**系统软件和应用软件**,**操作系统、编译系统和诊断程序**等均属于系统软件。

要使用电子计算机,就得编写程序。用来编写程序的语言称为**程序设计语言**。就以高级程序设计语言而论,近年来在我国广为流行的有 FORTRAN、ALGOL、COBOL、BASIC、PASCAL、LISP、PROLOG、C 语言等。计算机只能接受用机器语言编写的程序,为了使高级程序设计语言能被机器所接受,必须将高级语言编写的程序翻译成具体的机器语言,这一翻译任务是由存放在机器中的编译程序来承担的。这有些像我们阅读外文资料或听外国客人讲话一样,如果不懂外文,就必须请一位翻译来将外文译成中文。

通过各种编译程序系统(**编译程序、解释程序**等)将高级语言根据其语法和语义翻译成机器能够接受的机器语言。给机器配上一个“翻译”,它就懂得一种语言,给它配上的“翻译”越多,机器懂得的语言也就越多,从而机器的功能亦就越强。

从本质上来说,**程序设计语言**是按一定规则排列的符号集合,**编译程序**就是把这些符号集合变成机器指令的**转换器**,编译程序又称**编译器**。编译程序是计算机中的重要系统软件,是我们从事电子计算机科学学习和工作的所有同志应该熟悉和掌握的重要知识。本书将在介绍编译程序一般原理的同时重点介绍如何设计和构造编译程序,也即讨论编译程序构造的一般方法技术等。

1.1 程序设计语言

在计算机系统中,程序设计语言大体上可分为**机器语言、汇编语言和高级语言**,人们常常又把机器语言和汇编语言统称为**低级语言**。

1) 机器语言

每一台计算机都配备一套机器指令,每一条指令让计算机执行一个简单的特定动作,如把两个数相加、相减、相除或取送一个数等。这一套机器指令,对于不同的计算机来说,一般是互不相同的,这些机器指令亦称为机器语言。一般来说,计算机可以直接执行的代码形式的指令系统称为**机器语言**。

例如,在国产 DJS-21 计算机中,取、除、减、送的指令代码分别为:

002, 00E, 009, 004

假定 a、b、c、x 四个单元的地址码分别为:

0123, 1234, 2103, 0576

那么要计算

$x := a/b - c$

其机器语言程序为:

002 0123

00E 1234

009 2103

004 0576

2) 汇编语言

为了便于使用计算机,发挥计算机的作用,将计算机工作者从浩繁的计算机代码编程中解放出来,在 50 年代初人们发明了汇编语言。所谓汇编语言,即是计算机符号形式的指令系统。它是机器语言的符号化表示,与机器语言存在着一一对应的关系。它不再是令人讨厌的数字代码,而是采用了容易辨认、容易记忆的符号。如对应于取、除、减、送采用下列符号(当然也可以采用别的符号):

CLA, DIV, SUB, STO

从而对上述程序有如下汇编语言程序:

CLA a

DIV b

SUB c

STO x

与机器语言相比,汇编语言书写与阅读都比较方便、直观,也便于记忆和查找。

3) 高级语言

机器语言和汇编语言都依赖于具体的计算机,使用起来既烦琐又容易出错,程序不便于阅读和交流,实用性差,因此相继出现了很多高级语言。常用的高级语言有 FORTRAN、ALGOL、COBOL、BASIC、PASCAL 等。高级语言由表达各种不同意义的“关键字”和“表达式”按一定的语法规则所组成,它与人们习惯的数学语言非常接近,如上例可用 BASIC 语言写为赋值语句:

$x = a/b - c$

也可以用 PASCAL 语言写为赋值语句:

$x := a/b - c$

其中“=”和“:=”是不同高级语言中定义的赋值号,其含义是把赋值号右部表达式计算的结果赋给左部变量。

此外,高级语言不象机器语言或汇编语言那样只适用于某一台具体的计算机,而是适用于各种不同型号的计算机。

又例如,对于求半径为 r 的圆的面积和周长的问题,用我们熟悉的 PASCAL 语言编

写的程序如下:

```
program scr(input, output);  
  const pi=3.1416;  
  var s,c,r: real;  
  begin read(r);  
    s:=pi * sqr(r);  
    c:=2 * pi * r;  
    write(r,s,c)  
  end.
```

可见,用高级语言编写出来的程序简明清晰,便于阅读和交流,而对于上述问题若用机器语言或汇编语言来编写程序就不是那么简单明了。

总之,高级语言与机器无关,更加接近数学语言表示,简化了编程工作,程序便于阅读,有利于交流和移植。

1.2 编译程序

大家知道,用高级语言编写程序比用汇编语言或机器语言编写程序方便容易得多,但是计算机不懂得高级语言,也不懂本台机器上的汇编语言,而只懂本台机器上的机器语言,所以计算机无法直接执行用高级语言或汇编语言编写的程序。如果要在计算机上执行用高级语言或汇编语言编写的程序,则首先要进行翻译,即把用高级语言或汇编语言编写的程序翻译成与之等价的用机器语言书写的程序,然后对翻译出来的程序进行计算。这一工作由某一个翻译程序来承担。

所谓**翻译程序**,是指能够把 A 语言程序翻译成与之等价的 B 语言程序的程序,翻译程序又称**翻译器**。其中我们把 A 语言称为该程序的**源语言**,B 语言称为翻译程序的**目标语言**(或结果语言);源语言编写的程序称为**源程序**,目标语言书写的程序称为**目标程序**(或结果程序)。见图 1-1。

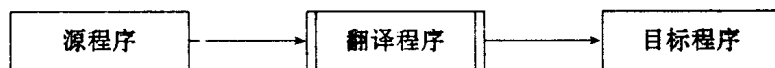


图 1-1

翻译程序大致可分为三类:**汇编程序**、**编译程序**和**解释程序**。

如果源语言为汇编语言,目标语言为机器语言,则这种翻译程序称为**汇编程序**,而逆向的转换称为**反汇编程序**。

如果源语言为高级语言,目标语言为某台计算机上的汇编语言或机器语言,则此翻译程序称为**编译程序**。

解释程序也是一种翻译程序,它能够按源程序的动态顺序逐句进行分析解释,根据语句功能翻译成与该语句相应的机器指令序列,并立即执行,直至结束。解释程序对源程序进行翻译,不产生目标程序。这种对源程序进行边解释边执行的方式便于人机对话,目前

BASIC 语言的翻译一般都是采用解释程序来实现。

此外,用机器语言构成的目标程序又称为**目标代码程序**或简称为**代码程序**,有时也称之**为目标代码或结果代码**。

运行目标代码所需要的所有子程序的全体称为**运行程序**。如存贮管理子程序,越界检查子程序,中断处理子程序等等。

1.3 源程序执行的途径

源程序的执行有两种途径:一种是编译途径,另一种是解释途径,这里分别予以介绍。

1.3.1 编译途径

编译途径即是将一份源程序从头至尾翻译成某台计算机上的机器语言让机器接受,然后执行之,并允许重复执行若干次。其好似自然语言的笔译者一样,将一篇外文资料译成中文让人们阅读,一次译完后可让人们阅读任意次。

编译程序与源语言以及计算机是紧密相关的,任何一个具体的编译程序都是某一特定机器上的关于某一特定语言的编译程序。

对于编译途径,源程序的执行需要分阶段,有些分为两大阶段,即编译阶段和运行阶段。如图 1-2 所示。

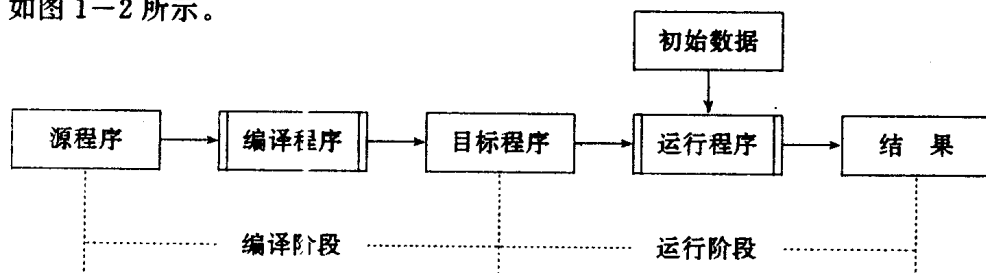


图 1-2

如果编译阶段生成的目标程序不是机器代码程序,而是汇编语言程序,则程序的执行要分三大阶段,即编译阶段、汇编阶段和运行阶段。如图 1-3 所示。

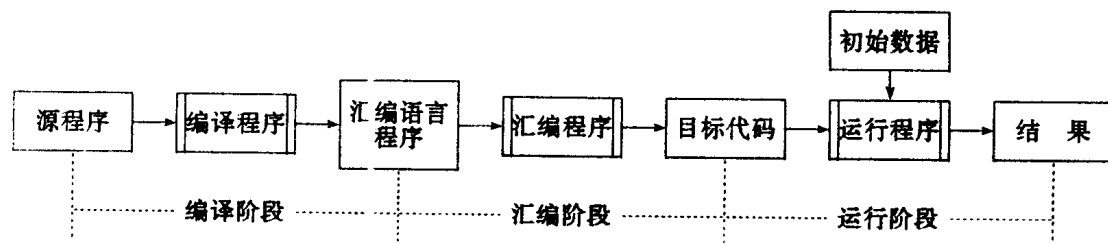


图 1-3

1.3.2 解释途径

解释途径就是对于源程序的一个语句,把它翻译成相应的机器语言,并让计算机立即执行。如果需要数据时,则提示用户输入初始数据,并立即进行处理。解释途径就是边解

释边执行,直至源程序动态处理完毕为止。其好似自然语言的口译者,口译者每遇到一句话便翻译一句,使听者接受。一些所谓“会话语言”(如 BASIC)的翻译就是由解释来实现的。其解释途径如图 1-4 所示。显然解释途径只有一个阶段,即解释执行阶段。

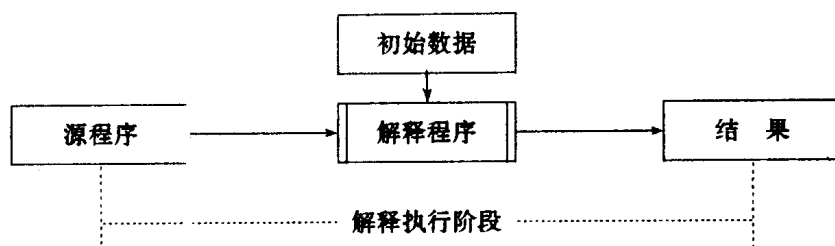


图 1-4

最后还得指出的是,解释程序对源程序的解释过程也有两种方式:一种是直接解释执行;另一种是把源程序先转换成以词为单位的中间代码程序,然后对中间代码程序逐句解释执行。中间代码有多种形式,这在后面将要介绍。解释方法不管采用哪种方式均不产生目标程序,这是与编译方法的不同之处。总之解释途径是对源程序逐字逐句地解释执行,最终得到运行结果。

1.4 编译程序的结构

1.4.1 编译程序的组成

1) 单词和属性字

为了介绍编译程序,这里首先介绍单词和单词的属性等有关概念。

(1) 单词

单词是指语言中具有独立意义的最小语法单位。

如在 PASCAL 语言中,字母和数字不是单词。这是因为在 PASCAL 语言中已明确规定它们没有单独的含义,而是用它们来组成标识符、数或字符串等。

例如,字母 a 不是单词,若在 PASCAL 语言程序的说明部分一旦对其进行了说明(如说明其是实型变量),从而字母 a 为标识符,就有了单独的含义,此时 a 即为单词。

此外,表达式也不是单词,它虽然具有独立含义,但不是最小的语法单位。

从而可见,在 PASCAL 语言中,单词是指所有的保留字、标识符、常数和特定符号等等,它们都是具有独立意义的最小语法单位。

最后还要指出的是,单词与语言密切相关,不同的语言有不同的单词。

(2) 属性和属性字

单词的**属性**是指单词特征和特性的有关信息。单词的最基本属性就是单词的名字,再根据单词的种类还有其它属性。如常数有类型和值;变量有类型和存放单元地址;数组有类型和描述数组的各种信息。

属性字是指单词的一种机内表示。单词的属性字有两个特点:一是长度统一。例如,可把所有的单词统一成一个机器字形式,如 32 位;二是刻划了单词的属性。它非常类似于

学生花名册和职工登记表等。属性字一般包括两个部分：一是单词的内部表示部分，即单词属于哪一类（具体实现中往往包括关于单词的更多信息）；二是单词的属性值（单词本身值或者指向单词本身的指针），如可表示为下列形式：

单词类别	单词属性值
------	-------

2) 编译程序的组成

前面已经讲过，编译程序是把应用高级语言编写的源程序翻译成与之等价的目标程序的一种翻译程序。其工作过程非常类似于自然语言之间的翻译。例如要把英文翻译成中文，最起码的条件是要能认识英文字母和标点符号等基本符号。然后是要能从由基本符号组成的有序串（即文章）中分离出一个一个具有独立意义的语法单位的能力，但是即使认识了它们的含义也并不说明已经具备翻译外文的能力。初学外语时经常会有这样的体会，即使把一个句子中的所有单词都从字典中查到了含义，但就是译不出一个句子的正确意思来，显然这是由于不懂这种句子的文法（即语法）的缘故。此外，即使能译出一个个句子的正确含义了，若不会进行辞句的修饰，那么所得的译文也必然是一篇蹩脚的文章。

编译程序也是一种翻译工作，它从某种意义上来说也应具有上述这些类似的功能。

为了把源程序翻译成与之等价的目标程序，编译程序一般要做词法分析、语法分析、语义分析、代码生成和优化等五个方面的工作。从而编译程序常由词法分析程序、语法分析程序、语义分析程序、代码优化程序和目标代码生成程序等五个主要部分组成。

(1) 词法分析

词法分析程序亦常称为词法分析器或扫描器。其主要任务是从左到右扫描源程序，识别单词及其有关属性，并转换成属性字。

例如，对于 PASCAL 语言中的赋值语句

$x := 2 * a + b$

其词法分析程序识别出 7 个单词：

$x, :=, 2, *, a, +, b$

而且指出：

x, a, b 是标识符；

2 是常数；

$*, +$ 是运算符；

$:=$ 是赋值号。

从而单词进行了分类（共性为一类），分类可根据不同的高级语言和习惯进行。

(2) 语法分析

语法分析程序又称语法分析器或分析器。其主要任务是根据语言的语法规则，逐一分析词法分析时得到的属性字，检查语法错误，若没有语法错误，则给出正确的语法结构。

如对于上述赋值语句，根据 PASCAL 语言中赋值语句的语法规则，可见其语法结构是正确的，即赋值号左部是标识符，右部是表达式。

(3) 语义分析

语义分析亦称语义处理。其主要任务是根据语法结构分析其含义，并用某中间语言

(如三元式、四元式、逆波兰式等)表示出来,亦就生成中间代码,或者直接生成目标代码。

如上述赋值语句,根据赋值语句的含义,即是将赋值号右边表达式的值赋给左部变量。这里引进三元式来具体说明。三元式定义为如下形式:

(op, a_1, a_2)

其中 op 为操作码或称运算符, a_1 和 a_2 为操作数或称为运算分量。

对于表达式

$a + b * c / d$

则有三元式:

(1) $(*, b, c)$

(2) $(/, (1), d)$

(3) $(+, a, (2))$

其中三元式中的①表示第 i 个三元式的结果。对于三元式在第 9 章中将作详细介绍。

于是上述赋值语句的中间代码形式为:

(1) $(*, 2, a)$

(2) $(+, (1), b)$

(3) $(:=, (2), x)$

若某台计算机相应于取、乘、加、送有如下机器指令:

CLA, MPY, ADD, STO

则根据赋值语句的含义,上述赋值语句经语义分析直接生成的目标代码可表示为:

CLA 2

MPY a

ADD b

STO x

(4) 代码优化

代码优化的主要任务是为了提高目标程序的质量而进行的工作。所谓目标程序的“优”,有时间和空间两方面的含义,即尽量缩短运行时间和尽量最少占用存储空间。优化工作往往是在中间代码这一级上进行。

(5) 代码生成

代码生成的主要任务是完成从中间代码到目标代码的生成工作。如果在语义分析时已直接生成了目标代码,则无需做此项工作。

3) 编译程序简单模型

一个编译程序一般由如上五个部分组成,各个部分如何组织起来,各个编译程序均不同。现以一个简单例子具体介绍编译程序的各方面工作过程,以便对编译程序各个部分有个大致的了解。

例如,计算半径为 r 的圆的面积和周长的问题,其 PASCAL 程序如下:

```
program scr(input,output);    {程序首部}
var r,s,c:real;               {说明 r,s,c 是实型变量}
begin s:=3.14*r*r;            {计算圆的面积 s}
```

$c := 2 * 3.14 * r$ {计算圆的周长 c}

end. {程序结束}

此程序不为完整程序,不含有输入和输出语句,这里用来只为说明问题方便而已。

(1) 词法分析

由前面我们知道,高级语言的单词是该语言中有实际意义的最小语法单位,都定义在各语言的字符集上。

词法分析程序对上述输入的源程序字符流中逐个地把这些单词识别出来,并把它们转换成机内单词形式(具体语法单位)。为简单直观,这里用下方划横线表示一个单词,从而对于上述源程序经词法分析后有如下具体语法单位:

```
program scr ( input , output ) ;  
var r , s , c ; real ;  
begin s := 3.14 * r * r ;  
      c := 2 * 3.14 * r  
end .
```

并识别出与单词有关的属性。

经词法分析后,滤除了源程序中的不必要的符号,如注解和多余空格等。此外,词法分析程序还要指出源程序中的单词错误,有时还要求词法分析程序把识别出来的各种标识符,如上例中的scr、r、s和c等填入符号表,以便后阶段查用。符号表可根据具体情况进行分类,常常包括保留字表、标识符表、特定符号表、运算符表和常量表等。这将在第4章中作详细介绍。

(2) 语法分析

根据PASCAL语言的文法规则,将词法分析得到的单词构成更大的语法单位,如表达式、各类语句、程序首部、程序体等,以及最终是怎样组成程序的,并在分析过程中指出程序中的各种语法错误。

如上述程序中求周长c的赋值语句的分析过程,如图1-5所示。

用同样的方法对程序的其它部分进行分析。不难发现,上述程序符合PASCAL语言的语法规则。语法结构是正确的,没有语法错误。此外,若在分析过程中发现语法错误则指出其错误,并将出错的信息记下,填入有关的表中以便错误处理来处理出错。

(3) 语义分析

根据语法分析,上述程序的语法结构正确,进而分析程序中各种语句或语法单位的含义。若是说明部分,则把变量的类型等属性填入表中,若是表达式和其它可执行语句,则把它们翻译成统一格式的中间代码形式或目标代码。经过语义分析阶段后,源程序加工成整齐和标准形式的中间代码程序。如上述源程序经语义分析得到下列中间代码程序:

- (1) (*, 3.14, r)
- (2) (*, (1), r)
- (3) (:=, (2), s)
- (4) (*, 2, 3.14)
- (5) (*, (4), r)

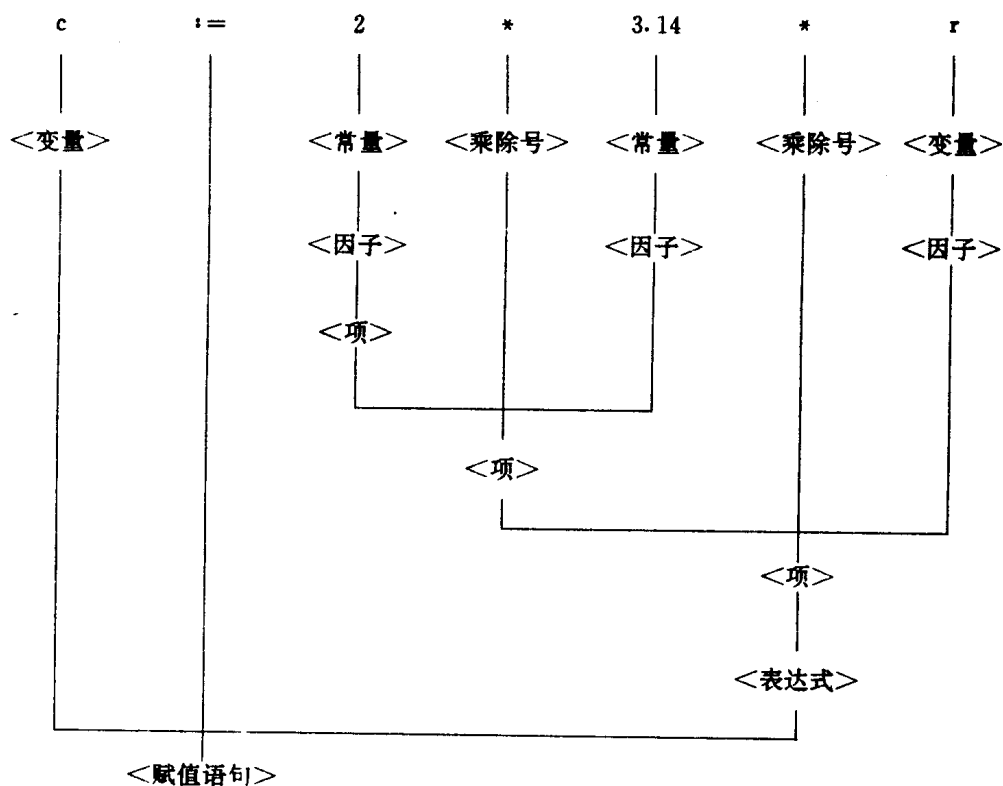


图 1-5

(6) (: = , (5), c)

(4) 代码优化

由前述中间代码程序可见,其三元式序列中对于常量计算 $2 * 3.14$, 最好在编译阶段就计算出来,免得在运行阶段计算,这样就可以减少一条乘法三元式。经优化后得到如下中间代码程序:

- (1) (* , 3.14, r)
- (2) (* , (1), r)
- (3) (: = , (2), s)
- (4) (* , 6.28, r)
- (5) (: = , (4), c)

此外,由求 s 和 c 的两条赋值语句还可以看到,它们的表达式中均要计算 $3.14 * r$, 若将其先计算出来,这样也可以减少一条乘法三元式,从而有优化后的中间代码程序:

- (1) (* , 3.14, r)
- (2) (* , (1), r)
- (3) (: = , (2), s)
- (4) (* , 2, (1))
- (5) (: = , (4), c)

(5) 目标代码生成