

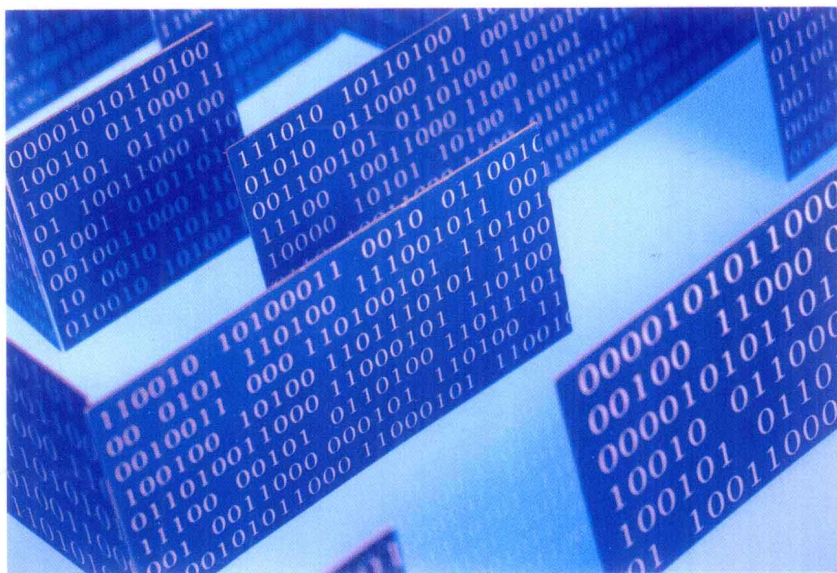


普通高等教育“十二五”规划教材

PUTONG GAODENG JIAOYU "12·5" GUIHUA JIAOCAI

# 编译原理学习指导

莫礼平 虞清 张兆海 唐铭 彭华 编著



冶金工业出版社  
Metallurgical Industry Press



普通高等教育“十二五”规划教材

# 编译原理学习指导

莫礼平 度清 张兆海 唐铭 彭华 编著

北京  
冶金工业出版社  
2012

## 内 容 提 要

本书是计算机技术与软件相关专业编译原理课程的辅助教学教材，主要介绍高级程序设计语言的编译程序的设计及实现的基本原理、基本方法。全书以词法分析和语法分析技术为重点，分13章进行编写。每章从学习要求、知识结构、知识要点、例题分析、习题与习题解答6个部分来指导学生进行学习。

本书可作为高等院校本科学生学习编译原理课程的指导书、教学参考书，也可作为报考相关专业硕士研究生或全国计算机技术与软件专业技术资格（水平）考试的读者的复习参考书，并可供相关专业技术人员参考使用。

### 图书在版编目（CIP）数据

编译原理学习指导 / 莫礼平等编著. —北京：冶金工业出版社，2012. 1

普通高等教育“十二五”规划教材

ISBN 978-7-5024-5845-4

I. ①编… II. ①莫… III. ①编译程序—程序设计—高等学校—教学参考资料 IV. ①TP314

中国版本图书馆 CIP 数据核字（2011）第 271760 号

出 版 人 曹胜利

地 址 北京北河沿大街嵩祝院北巷 39 号，邮编 100009

电 话 (010) 64027926 电子信箱 yjchs@cnmip.com.cn

责任编辑 宋 良 张 晶 美术编辑 李 新 版式设计 孙跃红

责任校对 刘 倩 责任印制 牛晓波

ISBN 978-7-5024-5845-4

北京兴华印刷厂印刷；冶金工业出版社出版发行；各地新华书店经销

2012 年 1 月第 1 版，2012 年 1 月第 1 次印刷

787mm × 1092mm 1/16；14.5 印张；373 千字；224 页

30.00 元

冶金工业出版社投稿电话：(010) 64027932 投稿邮箱：tougao@cnmip.com.cn

冶金工业出版社发行部 电话：(010) 64044283 传真：(010) 64027893

冶金书店 地址：北京东四西大街 46 号(100010) 电话：(010) 65289081(兼传真)

(本书如有印装质量问题，本社发行部负责退换)

# 前 言

编译原理是计算机科学与技术、软件工程、网络工程等本科专业重要的主干课程，在专业的教学计划中占有重要地位。该课程所介绍的高级程序设计语言的设计及实现的基本原理、基本方法和技术知识，在计算机的各个领域中有着广泛的应用，对计算机应用和软件相关专业的学生和计算机软件工程技术人员来说是重要的基础知识，对与计算机相关的其他许多领域的工作也有着一定的启发和指导作用。系统地掌握这些知识，有利于进一步提高学生以及软件相关技术人员的素质和能力。然而，学生普遍反映编译原理课程理论性较强，学习起来难度较大，而市场上大部分教材的内容较为抽象，不便于学生自学。

本书是湖南吉首大学“十二五精品教材”立项教材，它是编者在积累多年教学经验的基础上，根据培养新时代大学生对教学提出的新要求，在总结本校历届学生学习中所遇到的问题的基础上，根据学生学习的实际需要，精心编著而成，旨在帮助学生正确理解教材中的概念和原理，把握重点和难点，掌握解题技巧，进而达到深刻理解程序设计语言的设计及实现原理和技术，真正了解程序设计语言相关理论，在宏观上把握程序设计语言的目标。

本书以清华大学张素琴、吕映芝等编写的《编译原理（第2版）》的结构和内容为主线，每章均按学习要求、知识结构、知识要点、例题分析、习题与习题解答6个部分来编写。学习要求部分根据教学大纲和考核大纲给出了学生学习各章时的具体要求；知识结构部分按照《编译原理（第2版）》教材中对教学内容的安排，给出了知识结构图；知识要点部分简明扼要地归纳了各章的主要内容和需要重点掌握的知识，着重理清其中的概念、原理和方法，为学生理解和掌握课程内容提供指导；例题分析部分主要针对重要原理和算法，特别是针对学生在学习中遇到的重点和疑难问题，以例题形式进行了详尽的分析和讨论，并给出了一些简便的解题方法，以帮助学生拓宽思路，加深对课程内容的理解，提高分析和解决问题的能力；在习题与解答部分，针对各章重点选编了适当数量的各类习题，供学生练习，所有习题均给出了参考解答。

编者的愿望是，通过对本书的阅读，读者能够把握编译原理课程的主线，

加深对基本概念和原理的理解，掌握解题的思路与方法，融会贯通、举一反三，进而提高分析问题、解决问题的能力。

本书是编者多年来从事编译原理课程教学实践的积累成果，得益于张素琴等老师所编写的《编译原理》教科书和全国计算机技术与软件专业技术资格（水平）考试大纲和历届试题。在此，对张素琴等老师表示衷心的感谢。同时，还要感谢我们的学生，是他们强烈的求知欲望和勤学好问的认真学习态度给了我们编著该教材的动力。尽管如此，由于编著者水平有限，书中难免有疏漏之处，诚请读者批评指正。

编著者

2011年9月

# 目 录

|                             |    |
|-----------------------------|----|
| <b>1 引论</b> .....           | 1  |
| 1.1 学习要求 .....              | 1  |
| 1.2 知识结构 .....              | 1  |
| 1.3 知识要点 .....              | 1  |
| 1.4 例题分析 .....              | 5  |
| 1.5 习题 .....                | 7  |
| 1.6 习题解答 .....              | 8  |
| <b>2 PL/0 编译程序的实现</b> ..... | 10 |
| 2.1 学习要求 .....              | 10 |
| 2.2 知识结构 .....              | 10 |
| 2.3 知识要点 .....              | 10 |
| 2.4 例题分析 .....              | 15 |
| 2.5 习题 .....                | 17 |
| 2.6 习题解答 .....              | 19 |
| <b>3 文法和语言</b> .....        | 21 |
| 3.1 学习要求 .....              | 21 |
| 3.2 知识结构 .....              | 21 |
| 3.3 知识要点 .....              | 22 |
| 3.4 例题分析 .....              | 24 |
| 3.5 习题 .....                | 30 |
| 3.6 习题解答 .....              | 34 |
| <b>4 词法分析</b> .....         | 38 |
| 4.1 学习要求 .....              | 38 |
| 4.2 知识结构 .....              | 38 |
| 4.3 知识要点 .....              | 38 |
| 4.4 例题分析 .....              | 44 |
| 4.5 习题 .....                | 51 |
| 4.6 习题解答 .....              | 52 |

|                              |     |
|------------------------------|-----|
| <b>5 自顶向下语法分析</b> .....      | 58  |
| 5.1 学习要求 .....               | 58  |
| 5.2 知识结构 .....               | 58  |
| 5.3 知识要点 .....               | 58  |
| 5.4 例题分析 .....               | 65  |
| 5.5 习题 .....                 | 71  |
| 5.6 习题解答 .....               | 74  |
| <b>6 自底向上优先分析</b> .....      | 84  |
| 6.1 学习要求 .....               | 84  |
| 6.2 知识结构 .....               | 84  |
| 6.3 知识要点 .....               | 84  |
| 6.4 例题分析 .....               | 89  |
| 6.5 习题 .....                 | 95  |
| 6.6 习题解答 .....               | 99  |
| <b>7 LR 分析</b> .....         | 113 |
| 7.1 学习要求 .....               | 113 |
| 7.2 知识结构 .....               | 113 |
| 7.3 知识要点 .....               | 114 |
| 7.4 例题分析 .....               | 119 |
| 7.5 习题 .....                 | 126 |
| 7.6 习题解答 .....               | 131 |
| <b>8 语法制导翻译和中间代码生成</b> ..... | 151 |
| 8.1 学习要求 .....               | 151 |
| 8.2 知识结构 .....               | 151 |
| 8.3 知识要点 .....               | 151 |
| 8.4 例题分析 .....               | 157 |
| 8.5 习题 .....                 | 159 |
| 8.6 习题解答 .....               | 160 |
| <b>9 符号表</b> .....           | 164 |
| 9.1 学习要求 .....               | 164 |
| 9.2 知识结构 .....               | 164 |
| 9.3 知识要点 .....               | 164 |
| 9.4 例题分析 .....               | 168 |
| 9.5 习题 .....                 | 169 |

|                              |            |
|------------------------------|------------|
| 9.6 习题解答 .....               | 170        |
| <b>10 目标程序运行时的存储组织 .....</b> | <b>173</b> |
| 10.1 学习要求 .....              | 173        |
| 10.2 知识结构 .....              | 173        |
| 10.3 知识要点 .....              | 173        |
| 10.4 例题分析 .....              | 177        |
| 10.5 习题 .....                | 184        |
| 10.6 习题解答 .....              | 187        |
| <b>11 代码优化 .....</b>         | <b>190</b> |
| 11.1 学习要求 .....              | 190        |
| 11.2 知识结构 .....              | 190        |
| 11.3 知识要点 .....              | 190        |
| 11.4 例题分析 .....              | 198        |
| 11.5 习题 .....                | 204        |
| 11.6 习题解答 .....              | 208        |
| <b>12 代码生成 .....</b>         | <b>212</b> |
| 12.1 学习要求 .....              | 212        |
| 12.2 知识结构 .....              | 212        |
| 12.3 知识要点 .....              | 212        |
| 12.4 例题分析 .....              | 216        |
| 12.5 习题 .....                | 217        |
| 12.6 习题解答 .....              | 217        |
| <b>13 编译程序的构造 .....</b>      | <b>219</b> |
| 13.1 学习要求 .....              | 219        |
| 13.2 知识结构 .....              | 219        |
| 13.3 知识要点 .....              | 219        |
| 13.4 例题分析 .....              | 221        |
| 13.5 习题 .....                | 222        |
| 13.6 习题解答 .....              | 222        |
| <b>参考文献 .....</b>            | <b>224</b> |



# 1 引 论

## 1.1 学习要求

本章的学习要求如下：

- (1) 明确编译程序的功能及其在计算机系统中的作用。
- (2) 了解源语言程序被编译为目标程序的整个过程。
- (3) 掌握编译程序及相关概念、编译程序的结构及编译过程各阶段的工作任务。
- (4) 了解编译程序各个部分的设计原则，以及对应编译阶段之间的逻辑关系。
- (5) 了解编译阶段按前端与后端或按遍/趟组合的方法及主要参考因素。
- (6) 了解高级程序设计语言解释方式与编译方式的区别。
- (7) 了解常用的几种程序设计语言规范。

## 1.2 知识结构

本章的知识结构如图 1-1 所示。

## 1.3 知识要点

本章的知识要点主要包括以下内容：

### (1) 翻译程序

1) 定义：翻译程序是指这样的一个程序，它能够把某一种语言程序（称为源语言程序或源程序）转换成另一种语言程序（称为目标语言程序或目标程序），而后者与前者在逻辑上是等价的。

2) 分类：翻译程序根据所处理的对象和实现的途径不同又分为汇编程序、编译程序和解释程序。

①汇编程序：如果源语言是某种汇编语言，而目标语言是某种计算机的机器语言，这样的—个翻译程序就称为汇编程序。

②编译程序：如果源语言是某种高级语言，而目标语言是某种低级语言（汇编语言或机器语言），这样的—个翻译程序就称为编译程序。世界上第一个编译程序——FORTRAN 编译程序是 20 世纪 50 年代中期研究成功的。

根据不同的用途和侧重，编译程序还可以进一步分类。专门用于帮助程序开发和调试的编译程序称为诊断编译程序，着重于提高目标代码效率的编译程序称为优化编译程序。

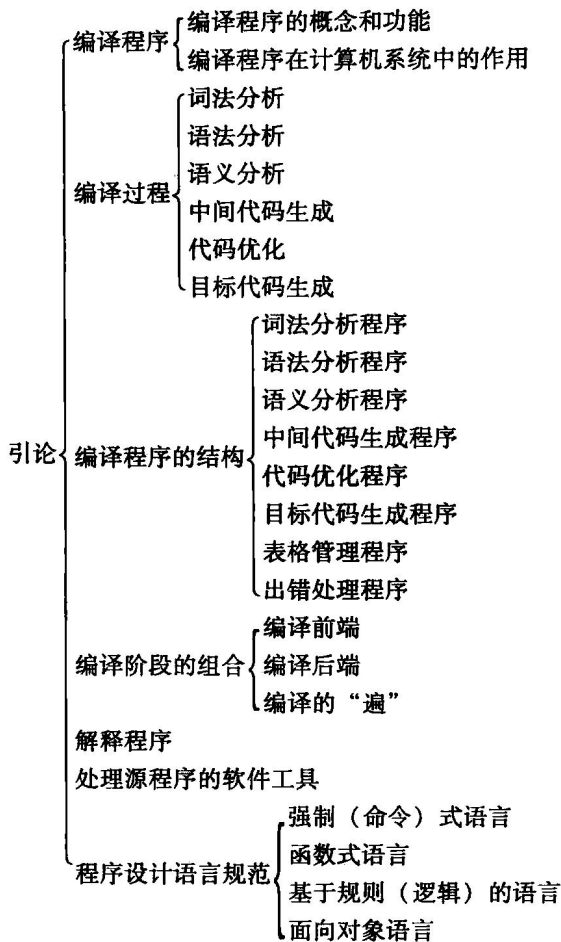


图 1-1 第 1 章知识结构图

现在很多编译程序同时提供调试、优化等多种功能，用户可以通过“开关”进行选择。

③解释程序：在翻译过程中，它按照高级语言源程序在计算机上执行的动态顺序对源程序的语句逐条翻译（解释），边解释边执行，直至结束。它不产生目标程序，其工作结果就是源程序的执行结果。这样的一个翻译程序就称为解释程序。

## (2) 编译过程的阶段划分

一般编译程序的工作过程按阶段进行，每个阶段将源程序从一种表示形式转换成另一种表示形式。典型的阶段划分方法是整个编译过程分为如下 6 个阶段：

### 1) 词法分析阶段

任务：对构成源程序的字符串进行扫描和分解，识别出单词符号。

所谓单词，是指逻辑上紧密相连的一组字符，这些字符具有集体含义。比如标识符用于表示变量名，是由字母字符开头，后跟字母、数字的字符序列组成的一种单词。保留字（关键字或基本字）也是一种单词，此外，还有算符、界符等也是单词。

输入：构成源程序的字符串。

输出：单词符号序列。

#### 2) 语法分析阶段

任务：根据语言的语法规则对单词符号串（符号序列）进行语法分析，识别出各类语法短语（能够表示为语法树的语法单位），判断输入串在语法上是否正确。

输入：单词序列。

输出：语法分析得到的可表示为语法树的语法单位的单词序列。

#### 3) 语义分析阶段

任务：按语义规则对语法分析所得到的语法单位进行语义分析，审查有无语义错误，为代码生成阶段收集类型信息，并进行类型审查以及违背语言规范的报错处理。

输入：语法分析后的单词序列。

输出：语义分析后得到的带有语义信息的单词序列。

#### 4) 中间代码生成阶段（并非所有的编译程序都包含此阶段）

任务：将语义分析得到的源程序变成一种结构简单、含义明确、易生成、易翻译成目标代码的内在代码形式。

输入：语义分析后的单词序列。

输出：中间代码。

#### 5) 代码优化阶段（也可以放到目标代码生成阶段之后）

任务：对中间代码或目标代码进行等价的变换改造等优化处理，使生成的代码更高效。

输入：中间代码或目标代码。

输出：优化后的中间代码或目标代码。

#### 6) 目标代码生成阶段

任务：将语义分析后的单词序列或中间代码生成特定机器上的绝对或可重定位的指令代码或汇编指令代码。

输入：语义分析后的单词序列或优化后的中间代码。

输出：特定机器上的目标代码。

### (3) 编译程序的结构

编译过程的6个阶段的任务，再加上表格管理和出错处理的工作，可由几个模块或程序完成，它们分别称作词法分析程序、语法分析程序、语义分析程序、中间代码生成程序、代码优化程序、目标代码生成程序、表格管理程序和出错处理程序。一个典型的编译程序结构框图如图1-2所示。

### (4) 编译阶段的组合与分遍

#### 1) 前后端组合法

编译前端：与源语言相关、与目标机无关的部分（第1~4阶段）。

编译后端：与目标机相关的部分（第6阶段）。

将第5阶段置于前端或后端都可以。基于前后端的常用组合方式有：

①同一源语言的编译前端 + 不同后端 = 不同机器上同一源语言的编译程序。

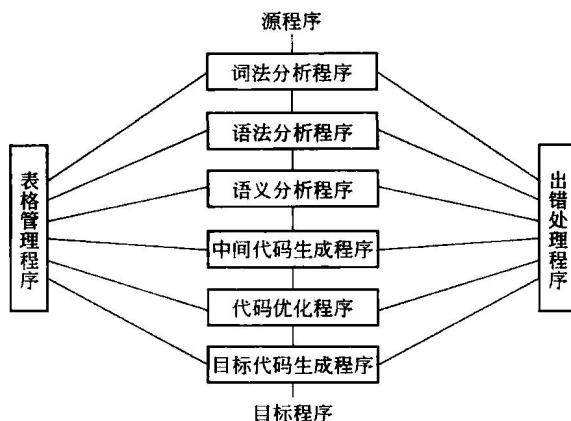


图 1-2 典型的编译程序结构框图

②不同源语言的编译前端生成同一种中间语言 + 共同后端 = 同一机器上不同语言的编译程序。

2) 遍组合法：对源程序或源程序的中间结果从头到尾扫描一次称为一遍或一趟。每一遍扫视完成一个或几个阶段的工作。一个编译程序可由一遍或多遍完成。

编译阶段前后端组合、实际编译程序分遍的主要参考因素是源语言与目标机器的特征。

#### (5) 在计算机上运行高级语言程序的两种途径

在计算机上运行高级语言的程序主要有两个途径：一是把该程序翻译为这个计算机的指令代码序列，这就是编译过程；二是编写一个程序，它解释所遇到的高级语言程序中的语句并且完成这些语句的动作，这样的程序就叫解释程序。从功能上说，一个解释程序能让计算机执行高级语言。它与编译程序的主要不同是它不生成目标代码，每遇到一个语句，就要对这个语句进行分析以决定语句的含义，执行相应的动作。许多编译程序的构造与实现技术同样适用于解释程序。

#### (6) 程序开发通常需要的其他工具

编译程序是实现高级语言的一个最重要的工具，但支持程序设计人员进行程序开发通常还需要一些其他的工具，如编辑程序、连接程序、调试工具等。编译程序与这些程序设计工具一起构成所谓的程序设计环境。在一个程序设计环境中，编译程序起着中心的作用。连接程序、调试程序、程序分析等工具的工作直接依赖于编译程序所产生的结果，而其他工具的构造也常常要用到编译的原理、方法和技术。

#### (7) 程序设计语言（书写计算机程序的高级语言）规范

1) 强制命令式语言，即过程式语言。该型语言中，一个过程可看做是一系列动作，其动作由命令驱动，以语句形式表示，一个语句接一个语句地执行。属于这种规范的语言如 PASCAL、C、FORTRAN 等，C++、Ada、COBOL 等也支持这种模式。本书介绍的编

译技术即针对这型语言。

2) 函数式语言, 即应用式语言。该型语言注重程序所表示的功能, 程序的开发过程是从前面已有函数出发构造出更加复杂的函数, 对初始数据集进行操作, 直到最后形成的函数可以得到最终结果。属于这种规范的语言如 ML、LISP 等。

3) 基于规则(逻辑)的语言。该型语言程序的执行过程是检查一定的使能条件, 满足时, 则执行适当的动作。如逻辑程序设计语言 PROLOG, 其基本的使能条件是某种谓词逻辑表达式; 再如 YACC, 其使能条件是程序的形式语法(BNF)。

4) 面向对象语言。该型语言的主要特点是提供抽象数据类型, 支持封装性、继承性和多态性。C++、Ada、Java 等也支持这种模式。

## 1.4 例题分析

**【例 1-1】** 写出编译程序对如下 C 源程序段进行处理时 6 个编译阶段的返回结果。

$$\text{Sum} = \text{Data}_1 + \text{Data}_2 * 100$$

分析与解答:

(1) 词法分析阶段: 该阶段的任务是从左到右一个字符一个字符地读入源程序, 对构成源程序的字符流进行扫描和分解, 从而识别出一个个单词(也称单词符号或符号)。

题中的 C 源程序段经词法分析后返回:

标识符: Sum

算符: =

标识符: Data<sub>1</sub>

算符: +

标识符: Data<sub>2</sub>

算符: \*

常数: 100

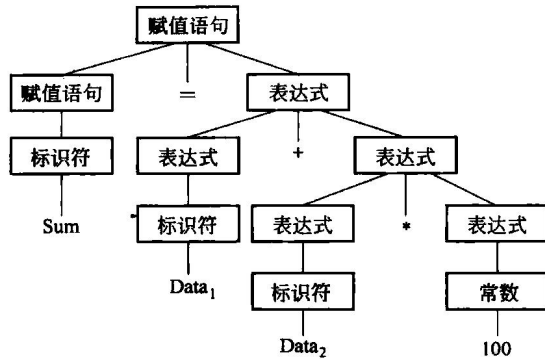
界符: ;

(2) 语法分析阶段: 该阶段的任务是在词法分析的基础上将单词序列分解成各类语法短语, 如“程序”、“语句”、“表达式”等。一般这种语法短语, 也称语法单位, 可表示成语法树。语法分析的功能依据语法规则(即描述程序结构的规则)进行层次分析, 把源程序的单词序列组合成语法短语(表示成语法树), 确定整个输入串是否构成一个语法上正确的程序。

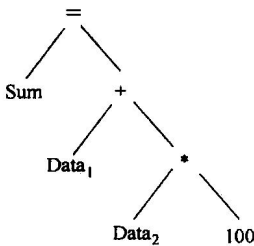
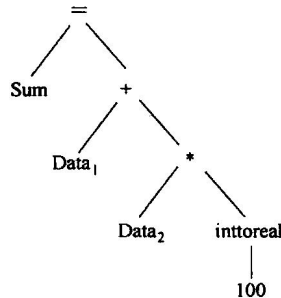
上述 C 语言源程序段中的单词序列经语法分析可得知是 C 语言的“赋值语句”, 可表示成如图 1-3 所示的三叉语法树或如图 1-4 所示的二叉语法树形式。

(3) 语义分析阶段: 该阶段的任务是审查语法上正确的源程序有无语义错误。例如: 源程序中有些语法成分, 按照语法规则去判断, 它是正确的, 但它不符合语义规则。比如使用了没有声明的变量; 或者给一个过程名赋值; 或者调用函数时参数类型不合适或者参加运算的两个变量类型不匹配等。这些都属于语义错误。

题中 C 源程序段语法正确, 在编译程序进行语义分析阶段的类型审查之后, 会将整型

图 1-3 赋值语句  $\text{Sum} = \text{Data}_1 + \text{Data}_2 * 100$  的三叉语法树

量转换为实型量。语义分析的结果可以体现在语法分析所得到的分析树上，在语法树上增加一个一目算符结点，这个结点的名称为“inttoreal”，表示进行将整型量转换成实型量的语义处理。那么，语义分析后的语法树如图 1-5 所示。

图 1-4 赋值语句  $\text{Sum} = \text{Data}_1 + \text{Data}_2 * 100$  的二义语法树图 1-5 赋值语句  $\text{Sum} = \text{Data}_1 + \text{Data}_2 * 100$  语义分析后的语法树

(4) 中间代码生成阶段：在语义分析之后，大部分编译程序将源程序变成一种叫做中间语言或中间代码的内部表示形式。中间代码是一种结构简单、含义明确的记号系统，这种记号系统可以设计为多种多样的形式。重要的设计原则有两点：一是容易生成；二是容易将它翻译成目标代码。很多编译程序采用了一种近似“三地址指令”的“四元式”中间代码，这种四元式的形式为：(运算符，运算对象 1，运算对象 2，结果)。

题中 C 源程序段转换为如下 4 个四元式形式的中间代码：

(inttoreal, 100, -,  $t_1$ )

(\* ,  $\text{Data}_2$ ,  $t_1$ ,  $t_2$ )

(+ ,  $\text{Data}_1$ ,  $t_2$ ,  $t_3$ )

(= ,  $t_3$ , -, Sum)

(5) 代码优化阶段：该阶段的任务是对前阶段产生的中间代码通过删除公共子表达式、强度削弱、循环优化等优化技术进行变换或改造，使生成的目标代码更为高效。代码优化工作会降低编译程序的编译速度，因此编译优化阶段常常作为可选择阶段，编译程序

具有控制机制以允许用户在编译速度和目标代码的质量间进行权衡。

本题中可以进行将 100 转换成实型数的代码优化，同时，由于生成  $t_3$  的四元式只起到值传递的作用，所以，也可以被优化。上述中间代码优化后变换为如下两个四元式：

$$(*, \text{Data}_2, 100.0, t_1)$$

$$(+, \text{Data}_1, t_1, \text{Sum})$$

(6) 目标代码生成阶段：该阶段的任务是把中间代码变换成特定机器上的绝对指令代码或可重定位的指令代码或汇编指令代码。这是编译的最后阶段，该阶段的工作很复杂，与硬件系统功能部件的运用、机器指令的选择、各种数据类型变量的存储空间分配以及寄存器的调度等相关。

本题中，使用两个寄存器 ( $R_1$  和  $R_2$ ) 可将优化后的中间代码生成某种汇编代码，形式如下：

```
MOVF Data2 R2
```

```
MOLF #100.0 R2
```

```
MOVF Data1 R1
```

```
ADDF R1 R2
```

```
MOV R1 Sum
```

第一条指令将  $\text{Data}_2$  的内容送至寄存器  $R_2$ ，第二条指令将  $R_2$  中的值与实常数 100.0 相乘，这里用 # 表明 100.0 处理为常数，第三条指令将  $\text{Data}_1$  移至寄存器  $R_1$ ，第四条指令  $R_1$  中的值加上前面计算出的  $R_2$  中的值存入  $R_1$  中，第五条指令将寄存器  $R_1$  的值移到 Sum 的地址中。

## 1.5 习 题

1-1 \_\_\_\_\_ 和 \_\_\_\_\_ 都是一种语言翻译程序，二者的区别在于它接受源程序后立即运行源程序，不生成 \_\_\_\_\_，直接输出结果。如 BASIC、SQL、Java 的 BYTECODE 解释程序等。

1-2 编译过程 6 个阶段的任务分别由 6 个子程序模块来完成。一个完整的编译程序还应包括 \_\_\_\_\_ 管理和 \_\_\_\_\_ 处理程序。因此，在典型编译程序框图中，从一个源程序翻译成目标程序，需要进行 \_\_\_\_\_ 个子程序模块的调用。

1-3 编译阶段按 \_\_\_\_\_ 可分为编译前端和编译后端，其中与目标机有关的阶段一般属于 \_\_\_\_\_，而与源语言相关的阶段一般属于 \_\_\_\_\_。

1-4 典型高级程序设计语言编译系统的工作过程常分为 6 个阶段，即词法分析、语法分析、语义分析、中间代码生成、\_\_\_\_\_、目标代码生成。编译阶段的两种组合方式是 \_\_\_\_\_ 组合法和按遍组合法，进行组合的主要参考因素都是源语言和 \_\_\_\_\_ 的特征。

1-5 编译阶段按遍组合时，对源程序或源程序的中间结果从头到尾扫描一次称为 \_\_\_\_\_。每一遍扫视完成一个或几个阶段的工作。一个编译程序可由 \_\_\_\_\_ 完成。实际编译程序分遍的主要参考因素是 \_\_\_\_\_ 与目标机器的特征。

1-6 从功能上看, 一个编译程序就是一个\_\_\_\_\_程序。其\_\_\_\_\_是高级语言, 其\_\_\_\_\_程序是低级语言。

1-7 典型高级程序设计语言编译系统的编译阶段按前后端方法进行组合时, 编译前端主要包括词法分析、语法分析等与\_\_\_\_\_有关而与目标机无关的阶段, 编译后端主要是指与目标机相关的阶段, 即\_\_\_\_\_阶段。而\_\_\_\_\_阶段编译前端或后端都可以。

1-8 编译阶段按前后端方法进行组合时, 同一源语言的\_\_\_\_\_使用不同的后端, 得到机器上同一源语言的编译程序。不同源语言的编译前端生成同一种中间语言后使用共同后端得到\_\_\_\_\_机器上不同语言的编译程序。

1-9 从支持的计算模式来看, 程序设计语言规范有: 强制命令式语言, 即\_\_\_\_\_语言; \_\_\_\_\_, 即应用式语言; 基于规则(逻辑)的语言和\_\_\_\_\_语言。

1-10 简述典型编译程序在各个工作阶段所完成的任务。

1-11 简述下列 C 语言源程序在编译程序进行词法分析后的输出结果。

```
main() { int x = 10, y; }
```

## 1.6 习题解答

1-1 编译程序; 解释程序; 目标代码。

1-2 表格; 出错; 8。

1-3 前后端组合; 后端; 前端。

1-4 代码优化; 前后端; 目标机。

1-5 一遍; 一遍或多遍; 源语言。

1-6 语言翻译; 源程序; 目标语言。

1-7 源语言; 目标代码生成; 代码优化。

1-8 编译前端; 不同; 同一。

1-9 过程式; 函数式; 面向对象。

1-10 典型编译程序在各个工作阶段所完成的任务如下:

(1) 词法分析: 对构成源程序的字符串进行扫描和分解, 识别出单词(如标识符等)符号。

(2) 语法分析: 根据语言的语法规则对单词符号串(符号序列)进行语法分析, 识别出各类语法短语(可表示成语法树的语法单位), 判断输入串在语法上是否正确。

(3) 语义分析: 按语义规则对语法分析得到的语法单位进行语义分析, 审查有无语义错误, 为代码生成阶段收集类型信息, 并进行类型审查和违背语言规范的报错处理。

(4) 中间代码生成(并非所有的编译程序都包含此阶段): 将语义分析得到的源程序变成一种结构简单、含义明确、容易生成、容易翻译成目标代码的内在代码形式。

(5) 代码优化(可放到目标代码生成阶段后): 对中间代码或目标代码进行变换改造等优化处理, 使生成的代码更高效。

(6) 目标代码生成: 将中间代码生成特定机器上的绝对或可重定位的指令代码或汇编指令代码。



1-11 题中 C 语言源程序在编译程序进行词法分析后的输出结果如下:

保留字: main  
界符: 左圆括号 (  
界符: 右圆括号 )  
界符: 左大括号 {  
保留字: int  
标识符: x  
运算符: =  
常数: 10  
标识符: y  
界符: ;  
界符: 右大括号 }