

21世纪高等学校规划教材 | 计算机科学与技术

编译原理及实现

姜淑娟 张辰 刘兵 编著



清华大学出版社

21世纪高等学校规划教材 | 计算机科学与技术

编译原理及实现

姜淑娟 张辰 刘兵 编著

清华大学出版社

北京

内 容 简 介

本书介绍编译理论的基础及实现方法,强调语言的形式化定义、编译技术和各种概念及实现过程的具体方法。全书共9章,内容包括编译引论、形式语言和有限自动机理论、词法分析、语法分析、语法制导翻译与中间代码生成、运行时的存储组织与分配、符号表、代码优化、目标代码生成。介绍过程以算法为核心,力求简单明了地反映编译的基础知识,为计算机软件工作者开发大型软件打下良好的基础。

本书可作为高等工科院校计算机专业的本科或专科教材,也可作为硕士研究生及计算机软件技术人员的参考书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

编译原理及实现/姜淑娟,张辰,刘兵编著.--北京:清华大学出版社,2016

21世纪高等学校规划教材·计算机科学与技术

ISBN 978-7-302-43553-2

I. ①编… II. ①姜… ②张… ③刘… III. ①编译程序—程序设计—高等学校—教材 IV. ①TP314

中国版本图书馆CIP数据核字(2016)第081981号

责任编辑:闫红梅 战晓雷

封面设计:傅瑞学

责任校对:时翠兰

责任印制:何 芊

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦A座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 刷 者:三河市君旺印务有限公司

装 订 者:三河市新茂装订有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:15.5 字 数:372千字

版 次:2016年9月第1版 印 次:2016年9月第1次印刷

印 数:1~2000

定 价:34.50元

出版说明

随着我国改革开放的进一步深化,高等教育也得到了快速发展,各地高校紧密结合地方经济建设发展需要,科学运用市场调节机制,加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度,通过教育改革合理调整和配置了教育资源,优化了传统学科专业,积极为地方经济建设输送人才,为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是,高等教育质量还需要进一步提高以适应经济社会发展的需要,不少高校的专业设置和结构不尽合理,教师队伍整体素质亟待提高,人才培养模式、教学内容和方法需要进一步转变,学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007年1月,教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》,计划实施“高等学校本科教学质量与教学改革工程”(简称“质量工程”),通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容,进一步深化高等学校教学改革,提高人才培养的能力和水平,更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中,各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势,对其特色专业及特色课程(群)加以规划、整理和总结,更新教学内容、改革课程体系,建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上,经教育部相关教学指导委员会专家的指导和建议,清华大学出版社在多个领域精选各高校的特色课程,分别规划出版系列教材,以配合“质量工程”的实施,满足各高校教学质量和教学改革的需要。

为了深入贯彻落实教育部《关于加强高等学校本科教学工作,提高教学质量的若干意见》精神,紧密配合教育部已经启动的“高等学校教学质量与教学改革工程精品课程建设工作”,在有关专家、教授的倡议和有关部门的大力支持下,我们组织并成立了“清华大学出版社教材编审委员会”(以下简称“编委会”),旨在配合教育部制定精品课程教材的出版规划,讨论并实施精品课程教材的编写与出版工作。“编委会”成员皆来自全国各类高等学校教学与科研第一线的骨干教师,其中许多教师为各校相关院、系主管教学的院长或系主任。

按照教育部的要求,“编委会”一致认为,精品课程的建设工作从开始就要坚持高标准、严要求,处于一个比较高的起点上。精品课程教材应该能够反映各高校教学改革与课程建设的需要,要有特色风格、有创新性(新体系、新内容、新手段、新思路,教材的内容体系有较高的科学创新、技术创新和理念创新的含量)、先进性(对原有的学科体系有实质性的改革和发展,顺应并符合21世纪教学发展的规律,代表并引领课程发展的趋势和方向)、示范性(教材所体现的课程体系具有较广泛的辐射性和示范性)和一定的前瞻性。教材由个人申报或各校推荐(通过所在高校的“编委会”成员推荐),经“编委会”认真评审,最后由清华大学出版

社审定出版。

目前,针对计算机类和电子信息类相关专业成立了两个“编委会”,即“清华大学出版社计算机教材编审委员会”和“清华大学出版社电子信息教材编审委员会”。推出的特色精品教材包括:

(1) 21 世纪高等学校规划教材·计算机应用——高等学校各类专业,特别是非计算机专业的计算机应用类教材。

(2) 21 世纪高等学校规划教材·计算机科学与技术——高等学校计算机相关专业的教材。

(3) 21 世纪高等学校规划教材·电子信息——高等学校电子信息相关专业的教材。

(4) 21 世纪高等学校规划教材·软件工程——高等学校软件工程相关专业的教材。

(5) 21 世纪高等学校规划教材·信息管理与信息系统。

(6) 21 世纪高等学校规划教材·财经管理与应用。

(7) 21 世纪高等学校规划教材·电子商务。

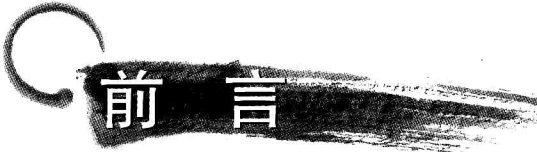
(8) 21 世纪高等学校规划教材·物联网。

清华大学出版社经过三十多年的努力,在教材尤其是计算机和电子信息类专业教材出版方面树立了权威品牌,为我国的高等教育事业做出了重要贡献。清华版教材形成了技术准确、内容严谨的独特风格,这种风格将延续并反映在特色精品教材的建设中。

清华大学出版社教材编审委员会

联系人:魏江江

E-mail:weijj@tup.tsinghua.edu.cn



前言

编译技术是计算机专业必修的一门重要的专业基础课程,也是计算机系统软件中非常重要的一个分支,任何计算机语言的实现都离不开编译技术,而且,编译技术在语言处理、软件工程、软件自动化、计算机体系结构设计及优化等诸多领域有着广泛的应用。因而,作为计算机专业的学生,学习和掌握编译程序的基本构造原理和实现技术,对今后进一步的学习、研究和工作奠定了坚实的专业理论基础。

本书共分为9章,第1章介绍编译程序的有关概念,编译过程、编译程序的组织和结构等要点。第2章作为后续各章的理论基础,主要介绍描述语言的两大工具——文法和有限自动机。第3章以正规式作为单词识别工具,从手工和自动两个角度讨论词法分析器的设计及实现。第4章对上下文无关文法的自上而下和自下而上的语法分析方法,以及几种经典方法,如递归子程序法、LL(1)分析法、算法优先分析法和LR分析法进行详细讲解。第5章介绍语义分析,涉及语法制导翻译与中间代码生成,重点讲解属性文法、翻译模式、中间代码表现形式和流行的高级程序设计语言中典型语句的翻译。第6章介绍编译程序运行时环境的有关概念和存储组织与分配技术。第7章介绍整个编译过程都要涉及的数据结构——符号表。第8章介绍代码优化,讨论优化的基本概念、优化涉及的数据流分析技术和控制流分析技术。第9章简要介绍代码生成的有关知识点。附录给出基于前端的编译程序算法的程序文本。

本书编写工作的分工是:张辰编写第1~3章,刘兵编写第4~6章,姜淑娟编写第7~9章。本书参考了国内外的一些专著、论文和资料,借鉴了一些专家学者的研究成果,对这些前辈和同行的引导和帮助表示衷心的感谢。

限于作者水平,本书难免存在疏误,请读者批评指正。

作 者

2016年1月

目 录

第 1 章 引论	1
1.1 什么是编译程序	1
1.1.1 编译程序与高级程序设计语言的关系	1
1.1.2 高级语言源程序的执行过程	2
1.1.3 与编译器有关的程序	2
1.2 编译过程与编译程序的组织结构	3
1.2.1 编译过程概述	3
1.2.2 编译程序的组织结构	7
1.2.3 编译阶段的组合	8
1.3 编译程序的构造与实现	8
1.3.1 如何构造一个编译程序	8
1.3.2 编译程序的开发	9
1.3.3 编译程序的自动构造工具	9
习题	10
第 2 章 形式语言和有限自动机理论	12
2.1 文法和语言	12
2.1.1 字母表和符号串	12
2.1.2 文法和语言的形式化定义	14
2.1.3 语法分析树与文法二义性	17
2.1.4 文法和语言的分类	18
2.2 有限自动机	19
2.2.1 确定的有限自动机(DFA)	19
2.2.2 非确定的有限自动机(NFA)	21
2.2.3 NFA 转换为等价的 DFA	23
2.2.4 确定的有限自动机的化简	25
习题	27
第 3 章 词法分析	30
3.1 词法分析基本思想	30
3.1.1 词法分析需求分析	30
3.1.2 词法分析任务	30

3.1.3	词法分析方式	31
3.2	单词的描述工具	31
3.2.1	正规文法和正规式	31
3.2.2	正规式与有限自动机的等价性	32
3.2.3	正规文法和有限自动机的等价性	36
3.3	单词的识别	37
3.3.1	单词分类	37
3.3.2	单词的内部表示	38
3.3.3	单词的形式化描述	38
3.4	词法分析程序的设计及实现	39
3.4.1	词法分析程序的输入	39
3.4.2	扫描缓冲区及其预处理	39
3.4.3	由词法规则画出状态转换图	41
3.4.4	单词对应状态转换图的实现	42
3.5	词法分析程序的自动实现	43
3.5.1	Lex 介绍	43
3.5.2	Lex 语法基础	44
3.5.3	词法分析器自动构造	45
习题		48

第 4 章 语法分析 50

4.1	自顶向下的语法分析方法	50
4.1.1	包含回溯的自顶向下语法分析	50
4.1.2	回溯产生的原因与解决方法	52
4.1.3	递归下降分析法	55
4.1.4	LL(1)分析法与 LL(1)分析器	57
4.2	自底向上的语法分析方法	62
4.2.1	“移进-归约”分析	62
4.2.2	规范归约与句柄	63
4.2.3	算符优先分析法	66
4.2.4	LR 分析法	74
4.2.5	语法分析程序自动生成器 YACC	96
习题		100

第 5 章 语法制导翻译与中间代码生成 102

5.1	两种翻译方法简介	102
5.2	属性文法	103
5.2.1	综合属性	104
5.2.2	继承属性	104

5.3	依赖图	105
5.4	语法制导翻译	107
5.4.1	S-属性文法与自底向上翻译	107
5.4.2	L-属性文法与自顶向下翻译	108
5.4.3	翻译模式	109
5.5	中间代码的形式	112
5.5.1	逆波兰表示法	112
5.5.2	三元式表示法	113
5.5.3	四元式表示法	114
5.5.4	图表示法	114
5.6	中间代码生成	115
5.6.1	说明语句的翻译	115
5.6.2	赋值语句的翻译	116
5.6.3	赋值语句中的布尔表达式的翻译	117
5.6.4	控制流语句中的布尔表达式的翻译	119
5.6.5	控制流语句的翻译	122
5.6.6	数组元素的翻译	128
5.6.7	函数调用的翻译	131
习题		133
第 6 章	运行时的存储组织与分配	135
6.1	概述	135
6.1.1	关于存储组织	135
6.1.2	函数(或过程)的活动记录	136
6.1.3	存储分配策略	137
6.2	静态存储分配	138
6.3	基于栈的运行时动态存储分配	139
6.3.1	简单栈式存储分配的实现	139
6.3.2	嵌套过程语言的栈式存储分配的实现	143
6.4	基于堆的动态存储分配的实现	147
6.5	参数传递	149
6.5.1	传值	149
6.5.2	传地址	150
习题		151
第 7 章	符号表	154
7.1	符号表的作用	154
7.2	符号表的内容	155
7.3	符号表的组织	157



7.3.1	符号表的数据结构	157
7.3.2	关键字域的组织	157
7.3.3	其他域的组织	158
7.4	符号表举例	162
7.4.1	无序表	162
7.4.2	有序表	162
7.4.3	散列符号表	162
7.4.4	栈式符号表	163
	习题	164
第 8 章	代码优化	166
8.1	概述	166
8.2	局部优化	166
8.2.1	基本块的划分	167
8.2.2	基本块的优化	167
8.2.3	基本块的有向图表示	169
8.3	循环优化	175
8.3.1	控制流图	175
8.3.2	基本属性	176
8.3.3	支配结点和后必经结点	176
8.3.4	循环的查找	179
8.3.5	循环优化	180
8.4	全局优化	183
8.4.1	相关概念及数据流方程	183
8.4.2	可到达定义	184
8.4.3	结构化程序的数据流分析	186
8.4.4	数据流方程的迭代解	187
8.4.5	活跃变量分析	189
	习题	192
第 9 章	目标代码生成	193
9.1	目标代码的形式	193
9.2	目标代码生成的主要问题	194
9.2.1	目标程序	194
9.2.2	指令选择	194
9.2.3	寄存器分配	195
9.2.4	计算次序选择	195
9.3	目标机器	196
9.3.1	目标机器的指令系统	196

9.3.2	指令开销	197
9.4	一个简单的代码生成器	198
9.4.1	寄存器描述和地址描述	198
9.4.2	代码生成算法	199
9.4.3	寄存器选择函数	199
9.4.4	为变址和指针语句产生代码	201
9.4.5	条件语句	201
9.5	寄存器分配的原则	202
	习题	203
附录 A	基于前端的编译程序算法	205
A.1	词法分析器的实现	205
A.2	递归下降语法分析器的实现	209
A.3	算法优先语法分析器的实现	222
A.4	LR 语法分析器的实现	226
	参考文献	232

1.1 什么是编译程序

1.1.1 编译程序与高级程序设计语言的关系

程序设计语言是向人及计算机描述计算机过程的记号。半个多世纪以来,程序设计语言经历了由低级向高级的发展,从最初的机器语言、汇编语言发展到面向过程语言和面向对象的语言,使得高级语言越来越倾向于以人的语言表达,并且成为人和计算机交互的媒介。但是,实际上计算机硬件仅能识别的是机器语言,根本不懂 Pascal、C、C++、Ada 和 Java 等高级语言,换言之,用高级语言编写的程序不能直接运行在计算机硬件上。那么,如何使一个高级语言编写的程序能够在只识别机器语言的计算机上执行呢?这就需要借助翻译程序,作为沟通计算机硬件与用户的渠道,通过翻译程序的翻译处理工作,计算机才能执行高级语言编写的程序。翻译程序的功能如图 1-1 所示。

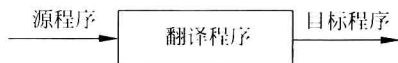


图 1-1 翻译程序的功能

简单地说,把用某一种程序设计语言的程序(源程序)翻译成等价的另一种语言的程序(目标程序)的程序称为翻译程序。被翻译的程序称为源程序。源语言是用来编写源程序的语言,一般是汇编语言或高级程序设计语言。源程序经过翻译程序翻译后生成的程序称为目标程序。目标程序可以用机器语言、汇编语言甚至高级语言或用户自定义的某类中间语言来描述。

翻译程序从源语言类型或实现机制的角度一般可以分为汇编程序、编译程序和解释程序。

汇编程序(Assembler):若源程序用汇编语言编写,经翻译生成机器语言表示的目标程序,该翻译程序称为汇编程序。通常,编译程序会生成汇编语言程序作为其目标语言,然后再由汇编程序将它翻译成目标代码。

编译程序(Compiler):若源程序用高级语言编写,经翻译加工生成某个机器的汇编语言程序或二进制代码程序,该翻译程序称为编译程序。将需要处理的数据送到生成的目标程序,就可以得到程序的结果。但是在把整个程序全部编译完成之前,这个程序是不能开始运行,也不能产生任何结果的。本书重点介绍编译程序的基本原理和主要构造方法。

解释程序(Interpreter):解释程序的工作模式是逐条语句分析执行,一旦第一条语句分析结束,源程序便开始运行并且生成结果。解释执行的过程如图 1-2 所示。解释执行的优

点是易于查错,在程序执行过程中可以修改程序。它特别适合程序员交互方式的工作情况,即希望在获取下一个语句之前了解每个语句的执行结果,允许执行时修改程序。典型的解释程序有 Basic 语言解释程序、UNIX 命令语言解释程序(shell)和数据库查询语言 SQL 解释程序等。

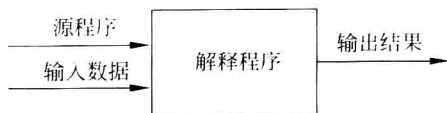


图 1-2 源程序的解释执行

1.1.2 高级语言源程序的执行过程

一个源程序编写后要投入运行,需要编译程序支持的执行过程分为两个阶段:编译阶段和运行阶段,如图 1-3(a)和(b)所示。编译阶段对整个源程序进行分析,翻译成等价的目标程序;运行阶段则由所生成的目标程序连同运行系统(数据空间分配子程序、标准函数程序等)接收程序的初始数据作为输入,运行后输出计算结果。

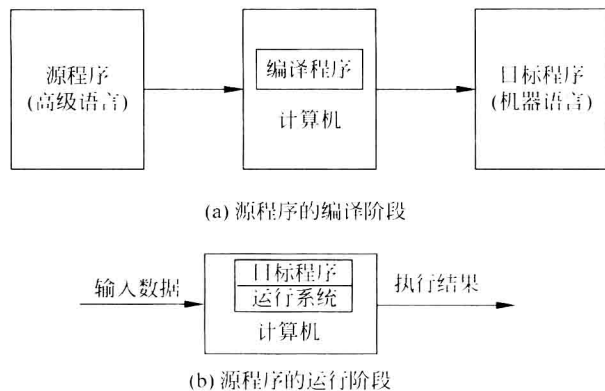


图 1-3 高级语言源程序的执行过程

1.1.3 与编译器有关的程序

除了编译器之外,创建一个可执行的目标程序还需要一些其他程序,如图 1-4 所示。

1. 编辑器(editor)

编译器通常接受由任何生成标准文件(如 ASCII 文件)的编辑器编写的源程序。

2. 预处理器(preprocessor)

预处理器是在真正的翻译之前由编译器调用的独立程序。预处理器可以删除注释、空白符、执行宏等。一个源程序甚至可能被分割成多个模块,分别存放于独立的文件中。用预处理器可以把源程序聚合在一起。

3. 汇编器(assembler)

将经过预处理的源程序作为输入传递给一个编译器,编译器可以产生一个汇编语言程序作为输出,因为汇编语言程序比较容易输出和调试。接着,这个汇编语言程序由称为汇编

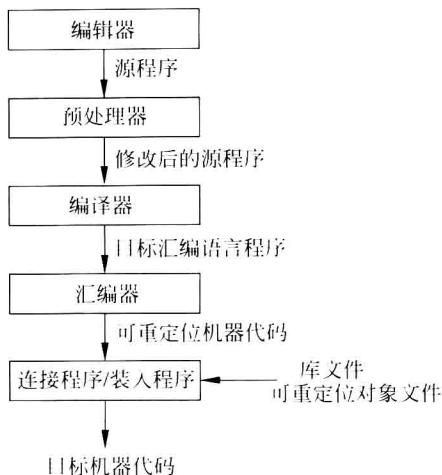


图 1-4 一个语言处理系统

器的程序进行处理,并产生可重定位的机器代码。

4. 连接程序(linker)

连接程序将分别在不同的目标文件中编译或汇编的代码收集到一个可直接执行的文件中。连接程序还直接连接目标程序和用于标准库函数的代码,以及连接目标程序和由计算机操作系统提供的资源。

5. 装入程序(loader)

装入程序可以处理所有与指定的基地址或起始地址有关的可重定位的机器代码。装入程序使得可执行代码更加灵活,但是装入通常是在后台(作为操作环境的一部分)或与连接程序相联合才发生,装入程序极少是实际的独立程序。

1.2 编译过程与编译程序的组织结构

1.2.1 编译过程概述

编译程序是比较复杂、庞大的系统软件。它所涉及的处理对象——源语言程序,从通用语言到计算机应用的各个领域的专用语言有成百上千种,它所涉及的处理结果——目标程序,其形式既可以是另一种程序设计语言或特定目标表示,又可以从微型计算机到超大型计算机的某种机器语言,可见不同源语言需要不同的编译程序。现在比较流行,使用比较广泛的一些编译器,如 Turbo 系列、Visual 系列等,已不仅仅是一个语言翻译工具,更是一个包括编译器、连接器、调试器等功能的庞大的集成开发环境。尽管编译程序的处理过程相当复杂,且不同的编译程序实现方法千差万别,构造原理各异,但任何编译程序要完成的基本任务都是类似的,其基本逻辑功能及必须完成的处理任务的分模块具有共同点。图 1-5 给出了编译程序总体结构的典型表示,也反映了编译程序的概貌与组成。

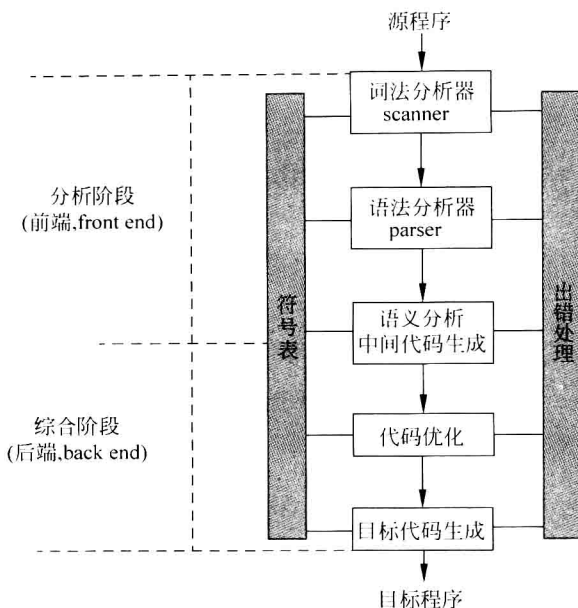


图 1-5 编译程序的体系结构

图 1-5 所示的编译程序总体结构图中,其中间位置的纵向 5 个矩形框表示编译程序工作过程的阶段或完成编译程序某阶段特定功能的模块,各模块间有密切的逻辑联系。图中两边的灰色矩形框是编译程序的辅助模块,可在编译的任何阶段被调用,辅助完成编译功能。

如图 1-5 所示,编译程序的工作过程是:从输入源程序开始到输出目标程序为止,经过词法分析、语法分析、语义分析与中间代码生成、代码优化及目标代码生成 5 个阶段,反映了一般编译器的动态编译过程。

1. 词法分析

词法分析(lexical analysis)阶段的任务是对输入符号串形式的源程序进行最初的加工处理。它依次扫描读入的源程序中的每个字符,识别出源程序中有独立意义的源语言单词,用某种特定的数据结构对它的属性予以表示和标注。词法分析实际上是一种线性分析,词法分析阶段工作依据的是源语言的词法规则。例如,有如下 C 语言表达式: $a=2+3$,经过词法分析识别出 5 个单词并输出每个单词的单词符号表示,如表 1-1 所示。

表 1-1 语句 $a=2+3$ 的单词符号

序号	单词类型	单词值
1	标识符	a
2	赋值	=
3	整常数	2
4	整常数	3
5	加号	+

在表 1-1 中,“单词类型”和“单词值”表示单词符号,通常也称为属性字或记号(token)。单词属性字的数据结构可根据不同语言及编译程序实现方案来设计,但一般由单词类型标识及单词值两部分构成。通俗地讲,单词的属性字实际是单词机器内部表示的一种记号。

2. 语法分析

语法分析(syntax analysis)阶段的任务是:在词法分析的基础上,依据源语言的语法规

则,对词法分析的结果进行语法检查,并识别出单词符号串所对应的语法范畴,类似于自然语言中对短语、句子的识别和分析。通常将语法分析的结果表示为抽象的分析树(parser tree)或称语法树(syntax tree)。例如,上述 C 语言表达式的语法分析树如图 1-6 所示。

3. 语义分析与中间代码生成

语义分析(semantic analysis)阶段的任务是:依据源语言限定的语义规则对语法分析所识别的语法范畴进行语义检查并分析其含义,初步翻译成与其等价的中间代码。语义分析是整个编译程序完成的最具实质性的翻译任务。

4. 代码优化

代码优化(code optimization)是为了改进目标代码的质量而在编译过程中进行的工作。代码优化可以在中间代码级或目标代码级进行,其实质是在不改变源程序语义的基础上对其进行加工变换,以期获得更高效的目标代码。而高效一般是对所产生的目标程序在运行时间上的缩短和存储空间上的节省而言的。

在前述的例子中,C 语言表达式 $a=2+3$ 的中间代码经过常量合并这样的优化后,不生成 $2+3$ 的中间代码,仅产生将 $2+3$ 的结构值 5 赋给标识符 a 的中间代码,即 $a=5$,这是在中间代码上的代码优化。

5. 目标代码生成

目标代码生成(code generation)作为编译程序的最后阶段,其任务是:根据中间代码及编译过程中产生的各种表格的有关信息,最终生成所期望的目标代码程序,一般为特定机器的机器语言代码或汇编语言代码。这个阶段实现了最后的翻译工作,处理过程较烦琐,需要充分考虑计算机硬件和软件所提供的资源,以生成高质量的目标代码。

例如,对上面的示例在代码生成时,设使用寄存器 $R0$ 和 $R1$,考虑怎样存储整型数来为数组元素的引用生成目标代码。表 1-2 给出了用汇编语言描述的目标代码。

表 1-2 用汇编语言描述的 C 语言表达式 $a=2+3$ 的目标代码

操作数	目标操作数	源操作数	说 明
MOV	R0	index	索引值赋给寄存器 R0
MUL	R0	2	存储按字节编址,整型数占 2 个字节
MOV	R1	&a	&a 表示变量 a 的地址
ADD	R1	R0	计算 a 的地址
MOV	R1	2	$R1=2$
MUL	R1	3	计算 $2+3$
MOV	R1	5	$a=5$

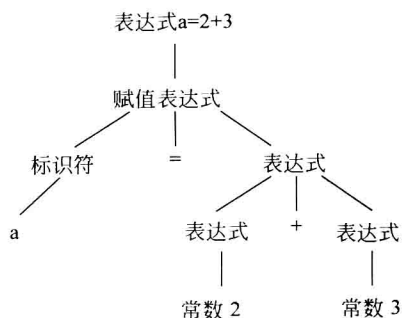


图 1-6 C 语言表达式 $a=2+3$ 的语法分析树

作为对编译程序的编译过程和各工作阶段的小结,下面给出例 1-1。

【例 1-1】设有 C 源语句如下:

$z = x - y * 15;$

//设 x, y, z 为 float 型变量

按照一般编译程序对源程序分析、处理的 5 个阶段,图 1-7 给出了对该语句的编译过程和各阶段的接口。

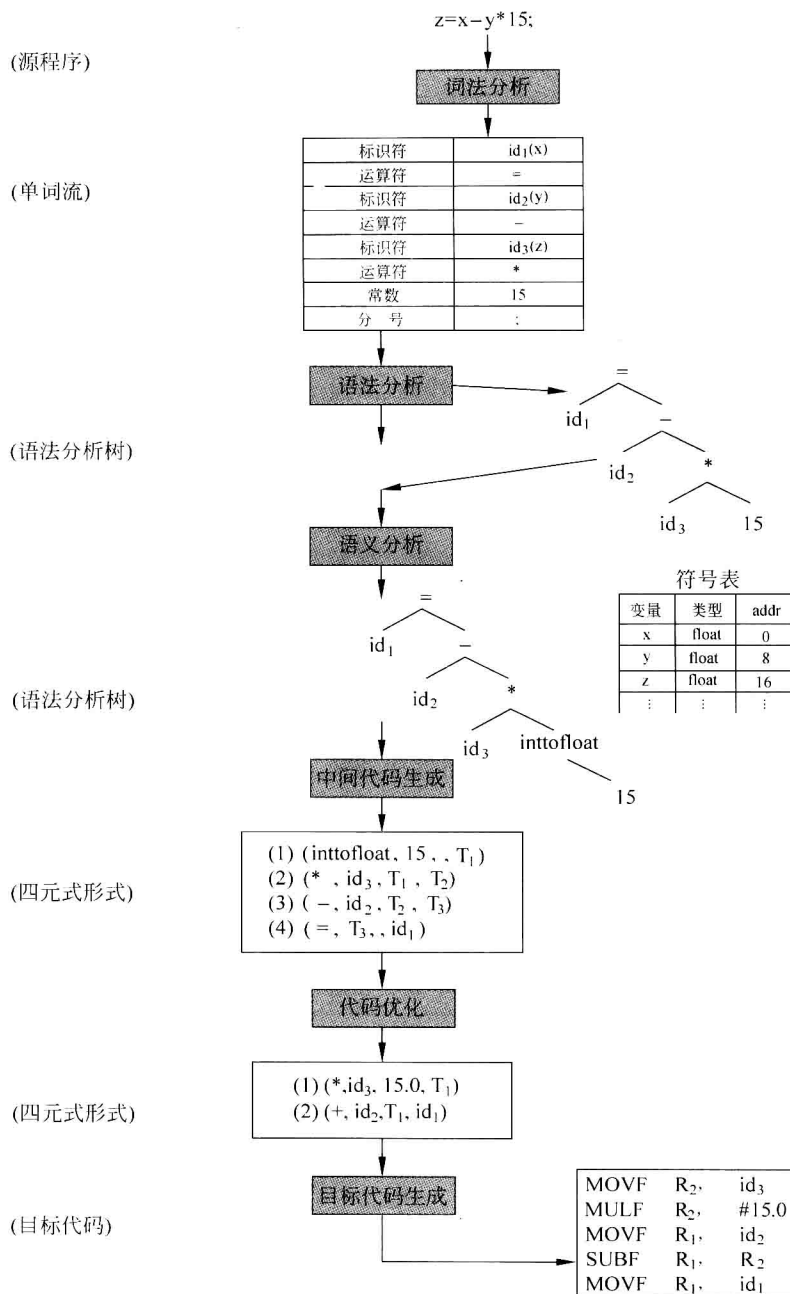


图 1-7 编译 C 语言语句 $z = x - y * 15;$ 的过程