

高等院校实践类系列教材

编译原理

重点难点分析·习题解析·实验指导

黄贤英 王柯柯 刘洁 曹琼 编著



机械工业出版社
CHINA MACHINE PRESS

TP314/73

2008

高等院校实践类系列教材

编译原理

重点难点分析·习题解析·实验指导

黄贤英 王柯柯 刘洁 曹琼 编著

机械工业出版社

本书是编译原理课程的教学辅导书,内容包括:习题解析和实验指导。习题解析部分共6章,基本覆盖了编译原理课程的全部内容,每章包括重点与难点、典型例题解析、习题及参考答案3部分,有助于读者抓住重点,突破难点;实验指导部分以自定义的Sample语言编译器的实现过程为主线组织编排,共5章,各章结合Sample语言编译器某个阶段的设计,分步实施,由浅入深,直到完成整个编译器。

本书可作为计算机及相关专业编译原理课程的辅助教材,也可作为研究生入学考试的复习参考书,还可作为软件工程技术人員或程序设计爱好者的参考书。

图书在版编目(CIP)数据

编译原理:重点难点分析·习题解析·实验指导/黄贤英等编著. —北京:机械工业出版社,2008.1

(高等院校实践类系列教材)

ISBN 978-7-111-23068-7

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

中国版本图书馆CIP数据核字(2007)第199356号

机械工业出版社(北京市百万庄大街22号 邮政编码 100037)

责任编辑:刘亚军

责任印制:李妍

北京蓝海印刷有限公司印刷

2008年3月第1版·第1次印刷

184mm×260mm·12.25印张·296千字

0001—5000册

标准书号:ISBN 978-7-111-23068-7

定价:20.00元

凡购本书,如有缺页,倒页,脱页,由本社发行部调换

销售服务热线电话:(010)68326294

购书热线电话:(010)88379639 88379641 88379643

编辑热线电话:(010)88379753 (010)88379739

封面无防伪标均为盗版

前 言

“编译原理”是计算机专业的一门核心课程，在计算机本科教学中占有十分重要的地位。该课程理论性与实践性都很强，学生在学习时普遍感到内容抽象，不易理解，掌握起来难度较大。学习编译原理课程的最好方法就是自己动手写一个小型的编译程序。然而，编译程序构造比较复杂，涉及的知识面较广，目前普遍缺少与教学内容配套且行之有效的实践教材，造成了学生在学过编译原理课程之后，难以完整、全面地掌握编译原理的有关知识，更不用说灵活运用编译原理知识从事相关设计或应用于其他领域。

本书分两篇，其中：

第一篇习题篇，主要内容为编译原理重点、难点分析和习题解析。本篇作为编译原理课程的教学辅导，旨在指导、启发和帮助学生正确理解编译程序的有关概念和原理，更好地掌握主要的编译技术和方法，提高分析和解决问题的能力。针对编译程序构造的一般原理、基本设计方法、主要实现技术和自动构造工具等重点内容，精选一些典型例题，分析问题涉及的知识点、解决思路及参考答案。每章后面都给出了相应的习题，便于读者掌握相关的重要知识点，复习重要理论和方法。本篇按照编译程序的总体架构，分6章讲解，即：编译原理概述、文法基本知识、词法分析、语法分析、语义分析与中间代码生成，以及运行时存储空间的分配、代码优化和目标代码生成。

第二篇实验篇为编译原理实践指导。本篇根据编译程序的阶段划分来实现一个PASCAL语言的裁剪版本（Sample语言）的编译程序。为了帮助读者更好地理解编译程序的构造，突出重点，简化设计，省略了一些复杂的高级语言成分。本篇利用软件工程的方法，由浅入深地阐述了通过编译理论具体指导编译程序设计的过程，同时给出了程序框架和部分伪代码。为了方便编译程序的教学，给出了各章节主要涉及到的算法的具体实现。

本书可以与目前各种编译原理教材配合使用，起到辅助学习和实践教学指导的作用。同时，本书进一步开拓了学习与应用的视野，特别是实践篇，给出了如何开发一个大型软件的工程化应用实例，可作为计算机软件工程人员较好的参考资料。书中带有*号的习题有一定难度，可能超过大纲要求。

本书提供Sample语言编译器的运行环境，需要者可到机械工业出版社网站（<http://www.cmpedu.com>）免费下载。

由于水平有限，书中难免存在疏漏和不妥之处，敬请读者批评指正。

作 者

目 录

前言

第一篇 习题篇

第1章 编译原理概述	1
1.1 重点与难点	1
1.1.1 编译程序	1
1.1.2 编译程序各阶段的任务	1
1.1.3 编译程序的结构	2
1.2 典型例题解析	2
1.3 习题及答案	4
1.3.1 习题	4
1.3.2 习题答案	6
第2章 形式语言基本知识	8
2.1 重点与难点	8
2.1.1 文法的基本概念	8
2.1.2 文法的二义性	8
2.2 典型例题解析	8
2.3 习题及答案	12
2.3.1 习题	12
2.3.2 习题答案	16
第3章 词法分析	21
3.1 重点与难点	21
3.1.1 词法分析器的功能及设计	21
3.1.2 正规式与有穷自动机	21
3.1.3 正规文法与有穷自动机的等价性	23
3.1.4 正规式与有穷自动机的等价性	23
3.1.5 符号表	24
3.2 典型例题解析	25
3.3 习题及答案	29
3.3.1 习题	29
3.3.2 习题答案	31
第4章 语法分析	37
4.1 重点与难点	37
4.1.1 语法分析器的功能	37
4.1.2 自上而下分析法	37
4.1.3 自下而上分析法	38
4.2 典型例题解析	43

IV

4.3	习题及答案	51
4.3.1	习题	51
4.3.2	习题答案	56
第5章	语义分析和中间代码生成	70
5.1	重点与难点	70
5.1.1	语义分析的功能	70
5.1.2	属性文法	70
5.1.3	中间代码的形式	70
5.1.4	语法制导的翻译方法	70
5.1.5	典型语句的翻译	71
5.2	典型例题解析	75
5.3	习题及答案	77
5.3.1	习题	77
5.3.2	习题答案	79
第6章	代码优化和目标代码生成	85
6.1	重点与难点	85
6.1.1	运行时的存储分配	85
6.1.2	局部优化	86
6.1.3	循环优化	86
6.1.4	目标代码生成	87
6.2	典型例题解析	88
6.3	习题及答案	91
6.3.1	习题	91
6.3.2	习题答案	95

第二篇 实验篇

第7章	实验预备知识	98
7.1	实验介绍	98
7.2	Sample 语言编译程序的实现方法	98
7.3	Sample 语言的定义	99
7.4	符合 Sample 语言定义的源程序示例	101
第8章	词法分析器及算法的设计与实现	102
8.1	词法分析器的设计	102
8.1.1	目的与要求	102
8.1.2	词法分析器实现中的约定	102
8.1.3	词法分析器的总体设计	102
8.1.4	词法分析程序的详细设计	104
8.2	词法分析器的实现	105
8.2.1	词法分析器总控程序的实现	105
8.2.2	定义符号表, 编写查找和插入函数	106
8.2.3	单词识别函数的编写	110

8.2.4	调试和运行结果	113
8.3	词法分析核心算法的设计	114
8.3.1	正规式转换为不确定的有穷自动机算法	114
8.3.2	不确定的有穷自动机的确定化	116
8.3.3	确定的有穷自动机的化简	118
8.4	词法分析器的自动生成	120
8.4.1	目的与要求	120
8.4.2	Lex 概述	120
8.4.3	Lex 源文件的格式	121
8.4.4	Lex 源文件中正规式的约定	122
8.4.5	Lex 源文件的翻译规则	123
8.4.6	Lex 源程序设计举例	124
8.4.7	Lex 使用中的一些注意事项	127
8.4.8	使用 Lex 自动生成 Sample 语言的词法分析器	128
8.4.9	实验内容	129
第 9 章	语法分析器及算法的设计与实现	130
9.1	语法分析器的设计	130
9.1.1	目的与要求	130
9.1.2	语法分析器的结构	130
9.1.3	语法分析的方法	131
9.2	语法分析器的实现	132
9.2.1	语法分析总控程序的编写	132
9.2.2	编写识别算术表达式和布尔表达式的函数	134
9.2.3	编写识别各种语句的函数	135
9.3	语法分析核心算法的设计	137
9.3.1	LL (1) 预测分析	137
9.3.2	算符优先分析	144
9.3.3	LR 分析方法	149
9.4	语法分析器的自动生成	156
9.4.1	目的与要求	156
9.4.2	YACC 的相关概念	156
9.4.3	实验内容	158
第 10 章	语义分析和中间代码的生成	159
10.1	语义分析与中间代码生成概述	159
10.1.1	目的与要求	159
10.1.2	语义分析与中间代码生成方法概述	159
10.1.3	语义分析与中间代码生成程序的设计方法	160
10.2	语义分析与中间代码生成器的实现	161
10.2.1	说明语句的翻译	161
10.2.2	可执行语句翻译的预备知识	162
10.2.3	简单赋值语句的翻译	163
10.2.4	布尔表达式的翻译	165

10.2.5	if 语句的翻译	168
10.2.6	repeat 语句的翻译	169
10.2.7	for 语句的翻译	169
10.2.8	程序开始和结束语句的翻译	170
10.2.9	运行结果	170
第 11 章	代码优化	172
11.1	目的与要求	172
11.2	代码优化器的作用	172
11.3	代码优化器的设计与实现	172
11.3.1	合并已知量	172
11.3.2	删除公共子表达式和无用赋值	176
第 12 章	目标代码生成	178
12.1	目的与要求	178
12.2	目标代码生成器的作用	178
12.3	目标代码生成器的设计	178
12.4	目标代码生成的相关算法	180
12.5	目标代码生成器的实现	182
12.5.1	汇编程序的结构	182
12.5.2	一个简单程序的源程序和汇编程序的对照	182
12.5.3	常见的四元式的翻译方法	183
12.5.4	实现的数据结构	184
12.5.5	实现的程序结构	184
参考文献	185

第一篇 习题篇

第 1 章 编译原理概述

1.1 重点与难点

1.1.1 编译程序

翻译程序能把一种语言书写的程序(称为源语言程序)翻译成另一种语言书写的程序(称为目标语言程序),而且后者与前者在逻辑上是等价的。

如果翻译程序是将高级语言书写的源程序翻译成与之等价的某计算机上的汇编语言程序或机器语言程序,则这种翻译程序称为编译程序。如果翻译程序是将汇编语言程序翻译成某台计算机上的机器语言程序,则这种翻译程序称为汇编程序。

用编译的方式运行程序是分阶段进行的。一般来说,首先进行“翻译”,把用高级语言编写的程序翻译成与之等价的机器语言程序,然后对翻译出来的程序进行运行计算。

1.1.2 编译程序各阶段的任务

编译程序的工作过程一般可以划分为五个阶段:词法分析、语法分析、语义分析与中间代码生成、代码优化和目标代码生成。

1. 词法分析

对输入的源程序进行从左到右的扫描和分解,识别出一个个单词,以单词的内部表示形式输出,并对源程序做一些简单的处理,如滤掉空格、去掉注释、报告错误等。

2. 语法分析

在词法分析的基础上,将单词符号组成各类语法短语,通过分析确定整个输入串是否构成语法上正确的程序。

3. 语义分析与中间代码生成

对语法分析所识别出的各类语法单位,分析其含义,进行初步的翻译。该阶段完成两项工作:对每个语法单位进行静态语义审查;若无语义错误,则进行中间代码的翻译。

4. 代码优化

对已生成的中间代码进行加工变换,使生成的目标代码更为高效。

5. 目标代码生成

把中间代码变换成特定机器上的低级语言程序。

1.1.3 编译程序的结构

编译程序的五个阶段可分别对应5个模块来完成相应功能,即词法分析器、语法分析器、语义分析与中间代码生成器、优化器和目标代码生成器。此外,一个完整的编译程序还必须包括“表格管理”和“出错处理”两部分,因此一个典型的编译程序的总体结构如图 1-1 所示。

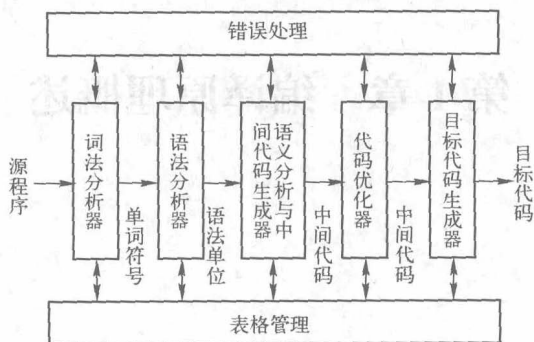


图 1-1 编译程序结构框图

1.2 典型例题解析

例 1 单项选择题:

- (1) 编译程序各阶段工作都涉及_____。
A. 词法分析
B. 表格管理
C. 语法分析
D. 语义分析
- (2) 解释程序和编译程序的区别在于_____。
A. 是否生成中间代码
B. 加工的对象不同
C. 使用的实现技术不同
D. 是否生成目标代码
- (3) 一遍扫描的编译程序的优点是_____。
A. 算法清晰
B. 便于分工
C. 便于优化
D. 编译速度快
- (4) 下列哪个程序不是编译程序的组成部分? _____
A. 词法分析程序
B. 代码读入程序
C. 代码生成程序
D. 语法分析程序

【解答】

(1) B。源程序各类信息和编译各阶段的进展情况都登记在一系列的表格中,其中最重要的是符号表。当扫描器识别出一个名字后将其填入符号表,但其各种属性需要到后续阶段才能填入,如类型的确定是在语义分析阶段,地址的确定是在目标代码生成阶段,因此,编译程序各阶段工作都涉及有关表格的构造、查找或更新。

(2) D。编译和解释两种翻译方法都对源程序进行了翻译,其中前者会生成可多次使用的目标代码,而后者没有生成目标代码,每一次执行源程序都需要重新翻译。

(3) D. 若一个编译程序分为多遍扫描,虽然其逻辑结构会比较清晰,但遍数增加势必会增加输入/输出所消耗的时间,因此,一遍扫描的编译程序的编译速度比多遍扫描要快。

(4) B. 代码读入程序不是编译程序的组成部分。构成编译程序的部分一般包括词法分析程序、语法分析程序、语义分析与中间代码生成程序、代码优化程序和目标代码生成程序。

例 2 判断题:

(1) 高级语言编写的源程序都必须通过编译,产生目标代码后才能运行。 ()

(2) 源程序与目标程序是等价关系。 ()

(3) 多遍扫描的编译程序的多遍是指多次重复读源程序。 ()

【解答】

(1) 错误。采用解释方式处理的源程序,其执行源程序的任务由解释程序来实现,大多数解释程序没有产生目标程序这一环节。

(2) 正确。源程序到目标程序的变换是等价变换,即两者结构不同,但语义是一致的。

(3) 错误。多遍扫描的编译程序的第一遍扫描的对象是源程序,其他各遍扫描的对象均是前一阶段加工的结果。

例 3 简答题:

(1) 一个典型的编译程序由哪几部分组成?各部分的主要功能是什么?

【解答】

一个典型的编译程序通常包含 7 个组成部分,它们是词法分析程序、语法分析程序、语义分析与中间代码生成程序、代码优化程序、目标代码生成程序、表格管理程序和错误处理程序。各部分的主要功能如下:

词法分析程序:对输入的源程序进行从左到右的扫描和分解,识别出一个个单词(或称单词符号),并以单词的内部表示形式输出。在词法分析的过程中,还要对源程序做一些简单的处理,如滤掉空格、去掉注释、报告错误等。

语法分析程序:在词法分析的基础上,将单词符号组成各类语法短语,如程序、语句、表达式等,通过分析确定整个输入串是否构成语法上正确的程序。这种语法短语,亦可称为语法单位、语法范畴,可以表示成语法树。

语义分析与中间代码生成程序:对语法分析所识别出的各类语法短语,分析其含义,进行初步的翻译(产生中间代码)。这一阶段有两项工作:首先对每个语法短语进行静态语义审查,如标识符是否定义,类型是否匹配等。若无语义错误,则进行中间代码的翻译。

代码优化程序:对已产生的中间代码进行加工变换,使生成的目标代码更为高效(时间和空间)。

目标代码生成程序:把中间代码(或经优化的中间代码)变换成特定机器上的低级语言代码(绝对指令代码、可重定位的指令代码或汇编指令代码)。

表格管理程序:负责建立、填写和查找等一系列表格管理工作。表格的作用是记录源程序的各类信息和编译各阶段的进展情况,编译的每个阶段所需的信息多数都从表格中读取,产生的中间结果也都记录在相应的表格中。可以说,整个编译过程就是造表、查表的工作过程。需要指出的是,这里的“表格管理程序”并不意味着它就是一个独立的表格管理模块,而是指编译程序所具有的表格管理功能。

错误处理程序:处理和校正源程序中存在的词法、语法和语义错误。当编译程序发现源程

序中的错误时,错误处理程序负责报告出错的位置和错误性质等信息,同时对发现的错误进行适当的校正(修复),目的是使编译程序能够继续向下分析和处理。

(2) 请指出下列错误信息可能是编译的哪个阶段报告的?

- ① else 没有匹配的 if;
- ② 数组下标越界;
- ③ 使用的函数没有定义;
- ④ 在数中出现了非数字信息。

【解答】

- ① 语法分析阶段;
- ② 语义分析与中间代码生成阶段;
- ③ 语义分析与中间代码生成阶段;
- ④ 词法分析阶段。

1.3 习题及答案

1.3.1 习题

1. 单项选择题

- (1) 构造编译程序应掌握_____方面的知识。
 - A. 源程序
 - B. 目标程序
 - C. 编译方法
 - D. 以上3项都是
- (2) 中间代码生成所依据的是语言的_____。
 - A. 词法规则
 - B. 语法规则
 - C. 语义规则
 - D. 产生规则
- (3) 作为编译程序的源语言不能是_____。
 - A. 高级语言
 - B. C语言
 - C. 低级语言
 - D. Pascal语言
- (4) 编译程序有5个阶段。其中,_____的目的是使最后阶段产生的目标代码更为高效。
 - A. 代码优化
 - B. 目标代码生成
 - C. 词法分析
 - D. 语法分析
- (5) 编译系统一般可分成 ①, ②, ③, ④ 和目标代码生成等5大部分,其中 ①, ② 和目标代码生成3部分是每个编译程序必不可少的,而 ③, ④ 则是可有可无的。以上这5个部分在工作过程中都会涉及到表格处理和 ⑤。

供选择的答案

- A. 词法分析
- B. 变量分析
- C. 数据分析
- D. 语法分析
- E. 过程分析
- F. 出错处理
- G. 优化处理
- H. 表达式处理
- I. 中间代码生成

- (6) 在使用高级语言编程时,首先可通过编译程序发现源程序的全部 ① 错误和部分

② 错误。

- A. 语法
- B. 语义
- C. 语用
- D. 运行

(7) 由于受到具体机器主存容量的限制,编译程序几个不同阶段的工作往往被组合成_____。

- A. 过程
- B. 程序
- C. 批量
- D. 遍

(8) 编译程序各阶段的工作往往是_____进行的。

- A. 顺序
- B. 并行
- C. 成批
- D. 穿插

(9) 编译过程中,语法分析器的任务是_____。

- ① 分析单词是怎样构成的
- ② 分析单词串是如何构成各种语句的
- ③ 分析语句是如何构成程序的
- ④ 分析程序的结构

- A. ②③
- B. ②③④
- C. ①②③
- D. ①②③④

(10) 汇编程序是将 ① 翻译成机器语言程序;编译程序是将 ② 翻译成 ③ 。

- A. 汇编语言程序
- B. 机器语言程序
- C. 高级语言程序
- D. 汇编语言程序或机器语言程序

(11) 高级语言源程序经编译后产生的程序叫_____。

- A. 源程序
- B. 目标程序
- C. 函数
- D. 过程

(12) 编译程序绝大多数时间花在_____上。

- A. 出错处理
- B. 词法分析
- C. 目标代码生成
- D. 管理表格

(13) 词法分析所依据的是_____。

- A. 语义规则
- B. 构词规则
- C. 语法规则
- D. 等价变换规则

(14) 编译程序的词法分析器用于识别_____。

- A. 字符串
- B. 语句
- C. 单词
- D. 标识符

(15) 编译程序的语法分析器可以发现源程序中的_____。

- A. 语义错误
- B. 语法和语义错误
- C. 错误并校正
- D. 语法错误

2. 填空题

(1) 扫描器的任务是从_____中识别出一个个_____。

(2) 高级语言源程序有两种执行方式,即_____和_____。

(3) 翻译程序是这样一种程序,它能够将_____转换成与其等价的_____。

(4) 对编译程序而言,输入数据是_____,输出结果是_____。

3. 判断题

(1) 编译程序是指把用高级语言书写的计算机程序翻译成面向计算机的目标程序而使用的计算机程序。()

(2) 就执行速度而言,编译后再执行程序比解释执行程序慢。()

(3) 解释程序是用来逐句分析执行源程序语句的计算机程序。()

(4) 使用编译程序时,因为是逐句地翻译执行源程序的语句,所以可逐条语句排错。()

(5) 任何一种汇编语言的每一语句都能对应一条机器语言指令。()

(6) 连接装配程序可把经编译程序产生的目标程序变成可执行的机器语言程序。()

(7) 目标程序一定是机器语言程序。()

(8) 高级语言程序到低级语言程序的转换是基于语义的等价变换。()

(9) 无论是一遍扫描的编译器还是多遍扫描的编译器都要对源程序扫描一遍。()

(10) 编译程序中错误处理的任务是对检查出的错误进行修改。()

4. 简答题

(1) 何谓翻译程序、编译程序和解释程序? 它们三者之间有何种关系?

(2) 何谓源程序、中间代码和目标代码? 它们三者之间有何种关系?

(3) 计算机执行用高级语言编写的程序有哪些途径? 它们之间的主要区别是什么?

1.3.2 习题答案

1. 单项选择题

(1) D (2) C (3) C (4) A (5) ① A, ② D, ③ I, ④ G, ⑤ F

(6) ① A, ② B (7) D (8) D (9) B (10) ① A, ② C, ③ D

(11) B (12) D (13) B (14) C (15) D

2. 填空题

(1) 源程序、单词符号;

(2) 解释、编译;

(3) 用甲语言书写的程序、用乙语言书写的程序;

(4) 源程序、目标程序。

3. 判断题

(1) √ (2) × (3) √ (4) × (5) ×

(6) √ (7) × (8) √ (9) √ (10) ×

4. 简答题

(1) 【解答】

翻译程序是指将用某种语言编写的程序转换成另一种语言形式的程序的程序,如编译程序和汇编程序。

编译程序是把用高级语言编写的源程序转换(加工)成与之等价的另一种用低级语言编写的目标程序的翻译程序。

解释程序是解释、执行高级语言源程序的程序。

广义上讲,编译程序和解释程序都属于翻译程序,但它们的翻译方式不同,解释程序是边翻译(解释)边执行,不产生目标代码,直接输出源程序的运行结果。而编译程序只负责把源程序翻译成目标程序,输出与源程序等价的目标程序,而目标代码的执行任务由操作系统来完成,即只翻译不执行。

(2)【解答】

所谓源程序是指用某种高级语言编写的程序,它是编译程序的加工对象。目标程序是指用低级语言(机器语言或汇编语言)编写的程序,它是编译程序的加工结果。中间代码是其结构介于源程序和目标程序之间的一种机内表示形式,它是编译程序产生的中间临时结果。它们三者之间的关系是等价关系,即结构不同,但语义相同。

(3)【解答】

计算机执行用高级语言编写的程序主要有两种途径:解释和编译。

在解释方式下,翻译程序事先并不采用将高级语言程序全部翻译成机器代码程序,然后执行这个机器代码程序的方法,而是每读入一条源程序的语句,就将其解释(翻译)成对应功能的机器代码语句串并执行,而所翻译的机器代码语句串在该语句执行后并不保留,然后再读入下一条源程序语句,并解释执行。这种方法是按源程序中语句的动态执行顺序逐句解释(翻译)执行的,如果一语句处于一循环体中,则每次循环执行到该语句时,都要将其翻译成机器代码后才执行。

在编译方式下,高级语言程序的执行是分两步进行的:第一步将高级语言程序全部翻译成机器代码程序,第二步再执行这个机器代码程序。因此,编译对源程序的处理是先翻译,后执行。

从执行速度上看,高级语言程序经编译后执行比解释执行要快,但解释方式下的人机界面比编译型好,便于程序调试。

这两种途径的主要区别在于:解释方式下不生成目标代码程序,而编译方式下生成目标代码程序。

第2章 形式语言基本知识

2.1 重点与难点

2.1.1 文法的基本概念

1. 文法

文法是一个四元组 $G=(V_N, V_T, P, S)$, 其中: V_N 为非空有穷的非终结符号集; V_T 为非空有穷的终结符号集; P 为产生式(也称规则、生成式)的集合; S 为开始符号。

2. 推导

从文法的开始符号出发, 反复连续使用所有可能的产生式, 将符号串中的非终结符用产生式右部进行替换和展开的过程。

最左推导: 在整个推导中, 每一步都替换句型中最左边的非终结符。

最右推导: 在整个推导中, 每一步都替换句型中最右边的非终结符, 又称为规范推导。

3. 句型

从开始符号 S 出发推导出的符号串称为文法 G 的一个句型。

4. 句子

若 X 是文法 G 的一个句型, 且 $X \in V_T^*$, 则称 X 是文法 G 的一个句子, 即仅含终结符的句型是一个句子。

5. 语言

文法 G 推导出的句子的全体称为文法 G 产生的语言, 记为 $L(G)$ 。

6. 文法等价

如果文法 G_1 与文法 G_2 产生的语言相同, 即 $L(G_1) = L(G_2)$, 则称这两个文法是等价的。

2.1.2 文法的二义性

语法树就是用一棵树来表示一个句型的推导过程。

一个文法, 如果存在某个句子对应两棵或两棵以上的语法树, 则称这个文法是二义文法。也就是说, 如果一个文法存在某个句子, 而这个句子有两个或两个以上不同的最左(最右)推导, 则这个文法是二义文法。文法的二义性并不代表语言是二义的。

2.2 典型例题解析

例1 何谓句型、句子和语言? 它们三者之间有何关系?

【解答】

句型是指由文法的开始符号出发, 通过若干步推导(包括0步)产生的符号串。

句子是指由文法开始符号出发,通过若干步推导产生的终结符号串。

某文法产生的所有句子的集合称为该文法产生的语言。

设 S 是文法 G 的开始符号,句型、句子和语言的形式定义如下:

(1) 称 x 是句型,如果 $S \xRightarrow{*} x$ 且 $x \in (V_N \cup V_T)^*$;

(2) 称 x 是句子,如果 $S \xRightarrow{*} x$ 且 $x \in V_T^*$;

(3) 由文法 G 所描述的语言定义为 $L(G) = \{x | S \xRightarrow{*} x, x \in V_T^*\}$ 。

设 x 是句子,若用 $SF(G)$ 表示文法 G 的所有句型的集合,则句型、句子和语言三者之间的关系可描述如下:

$L(G) \subset SF(G)$ 且 $L(G) = SF(G) \cap V_T^*$

若 $x \in L(G)$, 则 $x \in SF(G)$ 。

例 2 令文法 G 为:

$N \rightarrow D | ND$

$D \rightarrow 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9$

(1) 文法 G 定义的语言是什么?

(2) 给出句子 34、568 和 0127 的最左和最右推导。

【分析】

根据产生式 $N \rightarrow D | ND$ 可以看出, N 最终可以推导出 1 个或多个(可以是无穷多个) D , 根据产生式 $D \rightarrow 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9$ 可以看出, 每个 D 可以推导出 0~9 中的某一个数字。因此, N 最终推导出的就是由 0~9 这 10 个数字组成的任意字符串。

【解答】

(1) G 的语言是任意的数字串: $L(G) = \{a_1 a_2 \cdots a_n | a_i \in [0, 1, 2, \dots, 9]\}$ 。

(2) 句子 34、568 和 0127 的最左推导:

$N \Rightarrow ND \Rightarrow DD \Rightarrow 3D \Rightarrow 34$

$N \Rightarrow ND \Rightarrow NDD \Rightarrow DDD \Rightarrow 5DD \Rightarrow 56D \Rightarrow 568$

$N \Rightarrow ND \Rightarrow NDD \Rightarrow NDDD \Rightarrow DDDD \Rightarrow 0DDD \Rightarrow 01DD$

$\Rightarrow 012D \Rightarrow 0127$

句子 34、568 和 0127 的最右推导:

$N \Rightarrow ND \Rightarrow N4 \Rightarrow D4 \Rightarrow 34$

$N \Rightarrow ND \Rightarrow N8 \Rightarrow ND8 \Rightarrow N68 \Rightarrow D68 \Rightarrow 568$

$N \Rightarrow ND \Rightarrow N7 \Rightarrow ND7 \Rightarrow N27 \Rightarrow ND27 \Rightarrow N127 \Rightarrow D127 \Rightarrow 0127$

例 3 写一个文法,使其语言是奇数集,且每个奇数不以 0 开头。

【分析】

该题是希望构造一个文法,由它产生的句子是奇数,并且不以 0 开头。也就是说,它的每个句子都是以 1、3、5、7、9 中的某个数结尾。如果数字只有一位,则满足要求,如果有多位,则要求第一位不能是 0,而中间有多位、每位是什么数字(必须是数字)则没有要求。

因此,我们可以把这个文法分 3 部分来完成。分别用 3 个非终结符产生句子的第一位、中间部分和最后一位。引入几个非终结符,其中,一个用来产生句子的开头,可以是 1~9 之间的数,不包括 0;一个用来产生句子的结尾,为奇数;另一个则用来产生以 0~9 之间的任意整数