

编译技术学习辅导

主编/张守志

上海市高等教育自学考试辅导教材 计算机软件专业(独立本科段)

高等教育出版社

ISBN 7-04-014240-6



9 787040 142402 >

定价：22.30元

上海市高等教育自学考试辅导教材
计算机软件专业(独立本科段)

编译技术学习辅导

张守志 主编

高等教育出版社

内 容 简 介

本书是与上海市高等教育自学考试计算机软件专业(独立本科段)指定教材《编译技术》一书的配套辅助教材,它给出了教材中全部习题的参考解答,总结了每章中涉及到的概念、方法和原理,并增加了大量的练习。

本书不仅可以作为计算机类和信息类专业本科生的编译技术的参考书,也适合作为报考计算机专业研究生的学生的复习指导书,对于参加计算机软件专业水平考试的人员,也有一定的参考价值。

图书在版编目(CIP)数据

编译技术学习辅导/张守志主编. —北京:高等教育出版社, 2003. 12

(上海市高等教育自学考试计算机软件专业(独立本科段)指定系列丛书)

ISBN 7-04-014240-6

I. 编… II. 张… III. 编译程序-程序设计-高等教育-自学考试-自学参考资料 IV. TP314

中国版本图书馆CIP数据核字(2003)第126013号

责任编辑 司马镭 特约编辑 吴燕萍

封面设计 吴 昊 责任印制 蔡敏燕

书 名 编译技术学习辅导

主 编 张守志

出版发行 高等教育出版社

社 址 北京市西城区德外大街4号

邮政编码 100011

总 机 010-82028899

传 真 021-56965341

购书热线 010-64054588

021-56964871

免费咨询 800-810-0598

网 址 <http://www.hep.edu.cn>

<http://www.hep.com.cn>

<http://www.hepsh.com>

排版校对 展望文化发展有限公司

印 刷 江苏如皋市印刷有限公司

开 本 787×1092 1/16

印 张 16

字 数 380 000

版 次 2004年1月第1版

印 次 2004年1月第1次

定 价 22.30元

凡购买高等教育出版社图书,如有缺页、倒页、脱页等质量问题,请在所购图书销售部门联系调换。

版权所有 侵权必究

前 言

高等教育出版社出版的《编译技术》一书是上海市高等教育自学考试软件专业(独立本科段)的指定教材。《编译技术学习辅导》是与此教材配套的辅助教材,它给出了教材中全部习题的参考解答,总结了每章中涉及到的概念、方法和原理,并增加了大量的练习。

编译技术是一门理论性较强的课程,通过学习这门课程,能够对计算机许多问题的理解起到融会贯通的作用,我们希望通过对习题的分析与解答来帮助读者充分理解编译的基本原理,了解和掌握基本的编译方法,提高解决问题的能力。本书中许多习题选自各高校、科研院所的研究生入学试题和计算机软件专业水平考试试题,有些有一定的难度。

本书不仅可以作为计算机类和信息类专业本科生学习编译原理的参考书,也适合作为报考计算机专业研究生的学生的复习指导书,对于参加计算机软件专业水平考试的人员,也有一定的参考价值。

本书由张守志主编,参与本书编写的还有苏朋柿、王欢和许彦。

由于编者水平有限、时间仓促,书中难免会有缺点和错误,恳请读者及同行们批评指正。

编 者

2003 年 11 月

目 录

第 1 章 概论	1
1.1 内容辅导	1
1.2 难点分析	3
1.3 教材中习题的分析与解答	3
1.4 习题	4
1.5 习题答案	5
第 2 章 文法和语言	6
2.1 内容辅导	6
2.2 难点分析	10
2.3 教材中习题的分析与解答	10
2.4 习题	14
2.5 习题答案	26
第 3 章 词法分析	36
3.1 内容辅导	36
3.2 难点分析	42
3.3 教材中习题的分析与解答	42
3.4 习题	48
3.5 习题答案	53
第 4 章 语法分析	69
4.1 内容辅导	69
4.2 难点分析	81
4.3 教材中习题的分析与解答	83
4.4 习题	102
4.5 习题答案	115
第 5 章 语法制导翻译和中间代码生成	147
5.1 内容辅导	147
5.2 难点分析	164



5.3 教材中习题的分析与解答	165
5.4 习题	170
5.5 习题答案	176
第6章 符号表	183
6.1 内容辅导	183
6.2 难点分析	185
6.3 教材中习题的分析与解答	186
6.4 习题	187
6.5 习题答案	188
第7章 运行时的存储组织	189
7.1 内容辅导	189
7.2 难点分析	193
7.3 教材中习题的分析与解答	193
7.4 习题	199
7.5 习题答案	203
第8章 代码优化	212
8.1 内容辅导	212
8.2 难点分析	224
8.3 教材中习题的分析与解答	224
8.4 习题	228
8.5 习题答案	233
第9章 代码生成	242
9.1 内容辅导	242
9.2 难点分析	245
9.3 教材中习题的分析与解答	245
9.4 习题	247
9.5 习题答案	248

第 1 章

概 论

本章目的是介绍编译程序的概念、程序编译的过程及编译程序的结构,要求在熟悉这些内容的基础上,重点了解程序编译过程的各个阶段的任务。

1.1 内 容 辅 导

1.1.1 编译程序的概念

在计算机上执行一个高级语言程序一般要分为两步:第一步,用一个编译程序把高级语言程序翻译成机器语言程序;第二步,运行所得的机器语言程序求得计算结果。

通常所说的翻译程序是指这样的—个程序,它能够把某一种语言程序(称为源语言程序)改造成另一种语言程序(称为目标语言程序),而后者与前者在逻辑上是等价的。如果源语言是诸如 FORTRAN、Pascal、ALGOL 或 COBOL 这样的“高级语言”,而目标语言是诸如汇编语言或机器语言之类的“低级语言”,这样的—个翻译程序就称为**编译程序**。

编译程序所执行的转换工作可以用图 1.1 来说明。

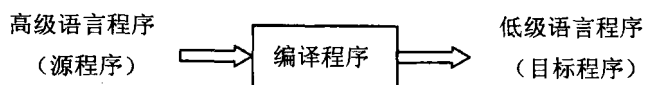


图 1.1 编译程序的功能

1.1.2 编译过程概述

编译程序的工作,从输入源程序开始到输出目标程序为止的整个过程,是非常复杂的。一般来说,这整个过程可以划分成 5 个阶段:词法分析、语法分析、中间代码生成、代码优化和目标代码生成。

第一阶段:词法分析。词法分析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)。这里所谓的单词是指逻辑上紧密相连的一组字符,这些字符具有具体含义。比如标识符是由字母字符开头,后跟字母、数字字符的字符序列组成的一种单词,此外还有算符、界符等等。

第二阶段:语法分析。语法分析的任务是在词法分析的基础上,根据语言的语法规则(文法规则),把单词符号串分解成各类语法单位(语法范畴),如“短语”、“句子”、“程序”等



等。通过语法分解,确定整个输入串是否构成一个语法上正确的“程序”。语法分析遵循的是语言的语法规则。

第三阶段:中间代码生成。在进行了上述的语法分析工作之后,有的编译程序按语言的语义将源程序变成一种内部表示形式,这种内部表示形式叫做中间语言或中间代码。所谓“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条: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)。

第四阶段:代码优化。在此阶段的任务是对前阶段产生的中间代码进行变换或改造,目的是使生成的目标代码更为高效,即节省时间和空间。

第五阶段:目标代码生成。这一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、可重定位的指令代码和汇编指令代码中的一种。这是编译的最后阶段,它的工作与硬件系统结构和指令含义有关,这个阶段的工作很复杂,涉及到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和后缓寄存器的调度等。

1.1.3 编译程序的结构

上述编译过程的5个阶段是编译程序工作时的动态特征。编译程序的结构可以按照这5个阶段的任务分模块进行设计。编译程序的结构设计可从图1.2所示的总框开始,这个总框是一个数据转换图,由它能够比较容易地设计出程序结构的框图。

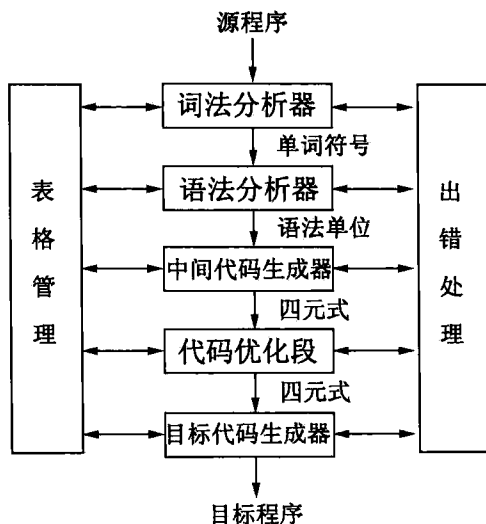


图 1.2 编译程序总框

图1.2中的词法分析器、语法分析器、中间代码生成器、代码优化段和目标代码生成器将分别完成上述5个阶段的编译任务。每个阶段的输出作为下一阶段的输入,第一阶段的输入是源程序,最后阶段的输出是目标程序。每个阶段的工作都与表格管理和出错处理这两部分功能模块相关。



上述编译过程的阶段划分是一种典型的处理模式,事实上并非所有的编译程序都分成这样几个阶段。有些编译程序对优化没有什么要求,优化阶段就可省去;在某些情况下,为了加快编译速度,中间代码产生阶段也可以省去;有些最简单的编译程序在语法分析的同时产生目标指令代码。但是,多数实用的编译程序都采用上述几个阶段的工作过程。

1.1.4 学习构造编译程序

要在某一台机器上为某种语言构造一个编译程序,必须掌握下述三方面的内容。

(1) 源语言:对被编译的源语言(如 FORTRAN 或 Pascal),要深刻理解其结构(语法)和含义(语义);

(2) 目标语言:假定目标语言是机器指令,那么,就必须搞清楚硬件的系统结构和操作系统的功能;

(3) 编译方法:把一种语言程序翻译成另一种语言程序的方法很多,但必须准确地掌握一二。

由于编译程序是一个极其复杂的系统,本书在讨论时是一部分一部分地进行研究的。因此,在学习过程中应注意前后联系,切忌用静止的、孤立的观点看待问题。作为一门技术课程,学习时务必注意理论联系实际,多做练习,多多实践。

1.2 难点分析

1. 编译过程各个阶段的划分:编译过程可以划分成 5 个阶段,即词法分析、语法分析、中间代码生成、代码优化和目标代码生成。

2. 编译过程各个阶段的任务。

1.3 教材中习题的分析与解答

1.1 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?

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,翻译程序并不是把高级语言程序彻底翻译成机器代码,而是每读入一条语句,就解释其含义并执行,然后再读入下一条语句,再解释执行,即按源程序中语句的动态顺序逐句地进行分析解释,并立即予以执行。

在编译方式下,翻译程序先对高级语言程序进行彻底翻译并生成目标代码,然后再对目标代码进行处理,即对源程序的处理是先翻译后执行的。

从速度上看,编译方式下,源程序的执行比解释方式下快,但解释方式有利于程序的调试。

1.2 有人认为编译程序的 5 个组成部分缺一不可,这种看法正确吗?

解答:

不正确。编译程序的 5 个组成部分中,词法分析、语法分析、语义分析和代码生成是必



需的,而代码优化是为了提高目标程序的质量,它不是必需的,没有优化部分的编译程序也能生成目标代码。

1.4 习 题

(一) 填空题

1. 编译程序的工作过程一般可以划分为_____等几个基本阶段,同时还会伴有_____和_____。
2. 编译方式与解释方式的根本区别在于_____。
3. 若源程序是用高级语言编写的,目标程序是_____,则其翻译程序称为编译程序。
4. 对编译程序而言,输入数据是_____,输出结果是_____。
5. 如果编译程序生成的目标程序是机器代码程序,则源程序的执行分为两大阶段:_____和_____。

(二) 单项选择题

1. 一个编译程序中,不仅包含词法分析、语法分析、中间代码生成、代码优化和目标代码生成等5个部分,还应包括_____。其中,_____和代码优化部分不是每个编译程序都必需的。词法分析器用于识别_____,语法分析器则可以发现源程序中的_____。

可选项有:

- | | | | |
|--------------|------------|--------------|-----------|
| (1) a. 模拟执行器 | b. 解释器 | c. 表格处理和出错处理 | d. 符号执行器 |
| (2) a. 语法分析 | b. 中间代码生成 | c. 词法分析 | d. 目标代码生成 |
| (3) a. 字符串 | b. 语句 | c. 单词 | d. 标识符 |
| (4) 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. 要在某一台机器上为某种语言构造一个编译程序,必须掌握下述三方面的内容:

_____, _____和_____。

可选项有:

- | | | | |
|------------|---------|---------|---------|
| a. 汇编语言 | b. 高级语言 | c. 源语言 | d. 目标语言 |
| e. 程序设计方法学 | f. 编译方法 | g. 测试方法 | h. 机器语言 |

3. 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语句和说明的
- ③ 分析语句和说明是如何构成程序的
- ④ 分析程序的结构

可选项有:

- | | | | |
|-------|--------|--------|---------|
| a. ②③ | b. ②③④ | c. ①②③ | d. ①②③④ |
|-------|--------|--------|---------|

4. 编译程序必须完成的工作有_____。

- | | | |
|----------|----------|----------|
| (1) 词法分析 | (2) 语法分析 | (3) 语义分析 |
|----------|----------|----------|



(4) 代码生成

(5) 中间代码生成

(6) 代码优化

可选项有:

a. (1)(2)(3)(4)

b. (1)(2)(3)(4)(5)

c. (1)(2)(3)(4)(5)(6)

d. (1)(2)(3)(4)(6)

e. (1)(2)(3)(5)(6)

5. “用高级语言书写的源程序都必须通过编译,产生目标代码后才能投入运行。”这种说法_____。

可选项有:

a. 不正确

b. 正确

1.5 习题答案

(一) 填空题

1. 词法分析、语法分析、语义分析、中间代码生成、代码优化 表格处理 出错处理
2. 是否生成目标代码
3. 机器语言程序或汇编语言程序
4. 源程序 目标程序
5. 编译阶段 运行阶段

(二) 单项选择题

1. c b c d 2. c d f 3. b 4. a 5. a

第2章

文法和语言

本章介绍了文法和语言的形式定义以及文法的类型。要求在熟悉这些内容的基础上,重点掌握上下文无关文法及其语法树和句型的分析。

2.1 内容辅导

2.1.1 符号和符号串

1. 字母表与符号串的定义

定义 2.1 字母表是元素的有穷非空集合,字母表中的元素称为符号,因此字母表也称为符号集。

定义 2.2 符号串(字)是由字母表中的符号所组成的有穷序列。

符号串总是建立在某个特定字母表上的,且只由字母表上的有穷多个符号组成。需要注意的是,符号串中符号的出现顺序是很重要的。

允许有不包含任何符号的符号串,这种符号串称为空符号串,简称空串(空字),用 ϵ 表示。

定义 2.3 若集合 A 中的一切元素都是某字母表上的符号串,则称 A 为该字母表上的符号串集合。

2. 有关符号串与符号串集合的运算

(1) 符号串的长度。符号串 x 中所包含符号的个数称为符号串 x 的长度,用 $|x|$ 表示。

(2) 符号串的头尾,固有头和固有尾。如果 $z = xy$ 是一符号串,那么 x 是 z 的头, y 是 z 的尾,如果 x 是非空的,那么 y 是固有尾;同样如果 y 是非空的,那么 x 是固有头。

当只对符号 $z = xy$ 的头感兴趣而对其余部分不感兴趣时,可以采用省略写法: $z = x\cdots$; 如果只是为了强调 x 在符号串 z 中的某处出现,则可表示为 $z = \cdots x\cdots$; 若符号 t 是符号串 z 的第一个符号,则表示为 $z = t\cdots$ 。

(3) 符号串的连接。设 x 和 y 是符号串,它们的连接 xy 是把 y 的符号写在 x 的符号之后得到的符号串。

(4) 符号串的方幂。设 x 是符号串,把 x 自身连接 n 次得到符号串 z , 即 $z = xx\cdots xx$, 称为符号串 x 的方幂,写作 $z = x^n$, 也即符号串 x 相继地重复写 n 次。

(5) 符号串集合的乘积。设两个符号串集合 A 和 B , 它们的乘积定义为:



$$AB = \{xy \mid x \in A \text{ 且 } y \in B\}$$

即 AB 是满足 x 属于 A , y 属于 B 的所有符号串 xy 所组成的集合。

(6) 指定字母表 Σ 之后, 可用 Σ^* 表示 Σ 上所有有穷长的串的集合, 也可表示为字母表的方幂形式:

$$\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \cup \dots$$

Σ^* 称为集合 Σ 的闭包。而 $\Sigma^+ = \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \cup \dots$ 称为 Σ 的正闭包。显然有:

$$\Sigma^* = \Sigma^0 \cup \Sigma^+$$

$$\Sigma^+ = \Sigma \Sigma^* = \Sigma^* \Sigma$$

Σ^* 具有可数的无穷数量的元素。使用一般集合论的表示符号来表示元素与集合的关系, 可有: 若 x 是 Σ^* 中的元素, 则表示为 $x \in \Sigma^*$, 否则 $x \notin \Sigma^*$ 。对于所有的 Σ , 有 $\epsilon \in \Sigma^*$ 。

2.1.2 文法和语言的形式定义

1. 文法的形式定义

规则, 也称重写规则、产生式或生成式, 是形如 $\alpha \rightarrow \beta$ 或 $\alpha ::= \beta$ 的 (α, β) 有序对, 其中 α 是某字母表 V 的正闭包 V^+ 中的一个符号, β 是 V^* 中的一个符号。 α 称为规则的左部, β 称为规则的右部。

定义 2.4 文法 G 定义为四元组 (V_N, V_T, P, S) 。

其中, V_N 为非终结符号集; V_T 为终结符号集; P 为产生式的集合; V_N, V_T 和 P 是非空有穷集; S 称作识别符号或开始符号, 它是一个非终结符, 至少在一条规则中作为左部出现。

V_N 和 V_T 不含公共的元素, 即 $V_N \cap V_T = \emptyset$ 。

通常用 V 表示 $V_N \cup V_T$, V 称为文法 G 的字母表或字汇表。

2. 语言的形式定义

为定义文法所产生的语言, 我们还需引入推导的概念, 即定义 V^* 中的符号之间的关系。

定义 2.5 设 $\alpha \rightarrow \beta$ 是文法 $G[S]$ 的规则 (或说是 P 中的一产生式), γ 和 δ 是 V^* 中的任意符号, 若有符号串 v, w 满足:

$$v = \gamma\alpha\delta, w = \gamma\beta\delta$$

则说 v (应用规则 $\alpha \rightarrow \beta$) 直接产生 w , 或说 w 是 v 的直接推导, 或说 w 直接归约到 v , 记作 $v \Rightarrow w$ 。

定义 2.6 如果存在直接推导的序列:

$$v = w_0 \Rightarrow w_1 \Rightarrow w_2 \cdots \Rightarrow w_n = w, (n > 0)$$

则称 v 推导出 w (推导长度为 n), 或称 w 归约到 v , 记作 $v \xRightarrow{+} w$ 。

定义 2.7 若有 $v \xRightarrow{+} w$, 或 $v = w$, 则记作 $v \xRightarrow{*} w$ 。



定义 2.8 设 $G[S]$ 是一文法, 如果符号串 x 是从开始符号推导出来的, 即 $S \Rightarrow^* x$, 则称 x 是文法 $G[S]$ 的句型。若 x 仅由终结符组成, 即 $S \Rightarrow^* x, x \in V_T^*$, 则称 x 为 $G[S]$ 的句子。

定义 2.9 文法 $G[S]$ 所产生的语言定义为集合:

$$L(G) = \{x \mid S \Rightarrow^* x, \text{ 且 } x \in V_T^*\}$$

定义 2.10 若 $L(G_1) = L(G_2)$, 则称文法 G_1 和 G_2 是等价的。

2.1.3 文法的类型

乔姆斯基把文法分成 4 种类型, 即 0 型、1 型、2 型和 3 型。这几类文法的差别在于对产生式施加不同的限制。

设 $G = (V_N, V_T, P, S)$, 若它的每个产生式 $\alpha \rightarrow \beta$ 是这样一种结构: $\alpha \in (V_N \cup V_T)^*$ 且至少含有一个非终结符, 而 $\beta \in (V_N \cup V_T)^*$, 则 G 是一个 **0 型文法**。

(1) 0 型文法。0 型文法也称**短语文法**。一个非常重要的理论结果是 0 型文法的能力相当于图灵机。或者说, 任何 0 型语言都是递归可枚举的; 反之, 递归可枚举的语言必定是一个 0 型语言。

如果对 0 型文法产生式的形式作某些限制, 则可以给出 1、2 和 3 型文法的定义。

(2) 1 型文法。设 $G = (V_N, V_T, P, S)$ 为一文法, 若 P 中的每一个产生式 $\alpha \rightarrow \beta$ 均满足 $|\beta| \geq |\alpha|$, 仅仅 $S \rightarrow \epsilon$ 除外, 则文法 G 是 **1 型文法**或**上下文有关文法**。

(3) 2 型文法。设 $G = (V_N, V_T, P, S)$ 为一文法, 若 P 中的每一个产生式 $\alpha \rightarrow \beta$ 均满足: α 是一非终结符, $\beta \in (V_N \cup V_T)^*$, 则此文法称为 **2 型文法**或**上下文无关文法**。

(4) 3 型文法。设文法 $G = (V_N, V_T, P, S)$, 若 P 中的每一个产生式的形式都是 $A \rightarrow aB$ 或 $A \rightarrow a$, 其中 A 和 B 都是非终结符, a 是终结符或 ϵ , 则称 G 是 **3 型文法**或**正规文法**。这种形式的正规文法也称为**右线性文法**。相应地, 若 P 中的每一个产生式的形式都是 $A \rightarrow Ba$ 或 $A \rightarrow a$, 其中 A 和 B 都是非终结符, a 是终结符或 ϵ , 则称这种形式的正规文法为**左线性文法**。

4 个文法类的定义是逐渐增加限制的, 因此每一种正规文法都是上下文无关的, 每一种上下文无关文法都是上下文有关的, 而每一种上下文有关文法都是 0 型文法。0 型文法产生的语言称为 0 型语言。上下文有关文法、上下文无关文法和正规文法产生的语言分别称为上下文有关语言、上下文无关语言和正规语言。

2.1.4 上下文无关文法及其语法树

1. 语法树的定义

给定文法 $G = (V_N, V_T, P, S)$, 对于 G 的任何句型都能构造与之关联的语法树, 这棵树满足下列 4 个条件:

- (1) 每个结点都有一个标记, 此标记是 V 的一个符号。
- (2) 根的标记是 S 。
- (3) 若一结点 n 至少有一个它自己除外的子孙, 并且有标记 A , 则 A 肯定在 V_N 中。
- (4) 如果结点 n 的直接子孙, 从左到右的次序是结点 $n_1, n_2, \dots, n_k$, 其标记分别为 $A_1,$



$A_2, \dots, A_k$, 那么 $A \rightarrow A_1 A_2 \dots A_k$ 一定是 P 中的一个产生式。

2. 规范推导和规范句型

定义 2.11 如果在推导的任何一步 $\alpha \Rightarrow \beta$, 其中 α, β 是句型, 都是对 α 中的最左(最右)非终结符进行替换, 则称这种推导为最左(最右)推导。

在形式语言中, 最右推导常被称为规范推导。由规范推导所得的句型称为规范句型。

3. 语法树与推导

从语法树构造推导是从推导构造语法树的逆过程。构造语法树是逐次依直接推导增添分支直到推导结束的过程, 那么其逆过程自然是从分支建立直接推导, 然后从语法树中剪去这个分支直到无分支可剪的过程。

概括地说, 从语法树构造推导也就是不断地重复构造最后直接推导, 并剪去相应分支, 直到无分支可剪的过程。按此构造法, 对于每棵语法树必定至少存在一个推导, 当改变构造最后直接推导和剪去相应分支的顺序时, 将得到不同的推导。

4. 文法的二义性

定义 2.12 如果一个文法存在某个句子对应两棵不同的语法树, 则说这个文法是二义的。或者说, 若一个文法中存在某个句子, 它有两个不同的最左(最右)推导, 则这个文法是二义的。

注意, 文法的二义性和语言的二义性是两个不同的概念。因为可能有两个不同的文法 G 和 G' , 其中 G 是二义的, G' 是无二义的, 但是却有 $L(G) = L(G')$, 也就是说, 这两个文法所产生的语言是相同的。如果产生上下文无关语言的每一个文法都是二义的, 则说此语言是先天二义的。

已经证明, 要判定任意的一个上下文无关文法是否为二义的, 或它是否产生一个先天二义的上下文无关语言, 这两个问题是递归不可解的。即, 不存在一个算法, 它能在有限步骤内, 确切判定任给的一个文法是否为二义的。我们所能做的事是为无二义性寻找一组充分条件(当然它们未必都是必要的)。

2.1.5 句型的分析

句型的分析就是识别一个符号串是否为某文法的句型, 是某个推导的构造过程。进一步说, 也就是当给定一个符号串时, 试图按照某文法的规则为该符号串构造推导或语法树, 以此识别出它是不是该文法的一个句型; 当符号串全部由终结符组成时, 识别它是不是某文法的句子。

分析技术可分成两大类: 自上而下的和自下而上的。

自上而下分析法就是从文法的开始符号出发, 反复使用各种产生式, 寻找“匹配”于输入符号串的推导。

自下而上分析法就是从输入符号串出发, 逐步进行“归约”, 直至归约到文法的开始符号。



2.2 难点分析

1. 推导、直接推导、句型和句子的概念。
2. 语法树、规范推导和规范句型的概念。
3. 语法树的构造和应用。

2.3 教材中习题的分析与解答

2.1 名词解释:

文法、直接推导、推导、句型、句子、语言、等价文法、语法树、规范推导、规范句型、二义文法。

解答:

(1) 文法

文法 G 定义为四元组 (V_N, V_T, P, S) 。

其中, V_N 为非终结符号集; V_T 为终结符号集; P 为产生式的集合; V_N, V_T 和 P 是非空有穷集; S 称作识别符号或开始符号, 它是一个非终结符, 至少在一条规则中作为左部出现。 V_N 和 V_T 不含公共的元素, 即 $V_N \cap V_T = \emptyset$ 。

(2) 直接推导

设 $\alpha \rightarrow \beta$ 是文法 $G[S]$ 的规则(或说是 P 中的一产生式), γ 和 δ 是 V^* 中的任意符号, 若有符号串 v, w 满足:

$$v = \gamma\alpha\delta, w = \gamma\beta\delta$$

则说 v (应用规则 $\alpha \rightarrow \beta$) 直接产生 w , 或说, w 是 v 的直接推导, 或说, w 直接归约到 v , 记作 $v \Rightarrow w$ 。

(3) 推导

如果存在直接推导的序列:

$$v = w_0 \Rightarrow w_1 \Rightarrow w_2 \cdots \Rightarrow w_n = w, (n > 0)$$

则称 v 推导出 w (推导长度为 n), 或称 w 归约到 v , 记作 $v \xRightarrow{+} w$ 。

(4) 句型

设 $G[S]$ 是一文法, 如果符号串 x 是从开始符号推导出来的, 即有 $S \xRightarrow{*} x$, 则称 x 是文法 $G[S]$ 的句型。

(5) 句子

设 $G[S]$ 是一文法, 如果符号串 x 是从开始符号推导出来的, 且 x 仅由终结符组成, 即有 $S \xRightarrow{*} x, x \in V_T^*$, 则称 x 为 $G[S]$ 的句子。

(6) 语言

文法 $G[S]$ 所产生的语言定义为集合: