编译原理学习与实践指导

金登男 何高奇 杨建国◎编著

華東理工大學出版社
EAST CHINA UNIVERSITY OF SCIENCE AND TECHNOLOGY PRESS

编译原理学习与实践指导

金登男　何高奇　杨建国　编著

華東理工大學出版社
EAST CHINA UNIVERSITY OF SCIENCE AND TECHNOLOGY PRESS
·上海·

图书在版编目(CIP)数据

编译原理学习与实践指导/金登男等编著.—上海:华东理工大学出版社,2013.11
ISBN 978-7-5628-3668-1

Ⅰ.①编… Ⅱ.①金… Ⅲ.①编译程序 Ⅳ.①TP314

中国版本图书馆CIP数据核字(2013)第238746号

编译原理学习与实践指导

编　　著/金登男　何高奇　杨建国
责任编辑/李国平
责任校对/张　波
封面设计/裘幼华
出版发行/华东理工大学出版社有限公司
地　址:上海市梅陇路130号,200237
电　话:(021)64250306(营销部)
(021)64252712(编辑室)
传　真:(021)64252707
网　址:press.ecust.edu.cn
印　　刷/上海展强印刷有限公司
开　　本/787 mm×1092 mm　1/16
印　　张/12.25
字　　数/310千字
版　　次/2013年11月第1版
印　　次/2013年11月第1次
书　　号/ISBN 978-7-5628-3668-1
定　　价/38.00元

联系我们:电子邮箱 press@ecust.edu.cn
官方微博 e.weibo.com/ecustpress
淘宝官网 http://shop61951206.taobao.com

前　言

编译原理是计算机专业的一门核心课程，内涵极其丰富，在计算机教学中占有十分重要的地位，通常作为计算机专业研究生的入学考试科目之一。它包含了词法、语法、代码生成和代码优化等方面的理论和技术，也涉及自动机理论、数据结构等众多领域的知识。

由于编译原理涉及的知识面广，理论和实践性都很强，许多学生在学习编译原理时感到内容非常抽象、不易理解、难以掌握，解答习题时也往往需要花费大量的时间，甚至还不得要领，尤其是上机实践时感到困难重重。为了帮助学生正确理解编译原理的基本概念，掌握解题技巧，顺利通过上机实践，理解编译原理的过程和方法，编者根据自己在编译原理课程中的长期教学经验，针对学生在学习编译原理课程中所遇到的困难，特地编写了本书。希望本书能帮助学生充分理解编译的基本方法，提高分析问题、解决问题的能力，既能为学生提供一本合适的编译原理学习参考书，同时也为考研复习者提供一本有价值的参考资料。

全书分为两篇。第1篇是“编译原理”学习指导，共12章，第1至第6章由何高奇完成，第7至第12章由杨建国完成；第2篇是“编译原理”实践指导，由金登男完成；附录是“编译原理”系统软件课程设计及试卷，由杨建国完成。

在学习指导的每章中，首先总结了本章知识结构图，然后进行了疑难解惑，重点讨论了典型例题，最后给出针对性的习题。在实践指导的每章中主要叙述了词法分析、语法分析和代码生成三大实践环节中的实践任务、编译程序的代码、实验步骤、实验要求和实验报告等。

本书选择的习题参考了目前市场上的主流教材和习题辅导，相关书目见参考文献，在此向相关作者表示诚挚的感谢。由于作者水平有限，书中还存在一些缺点和错误，恳请广大读者批评指正。

编者

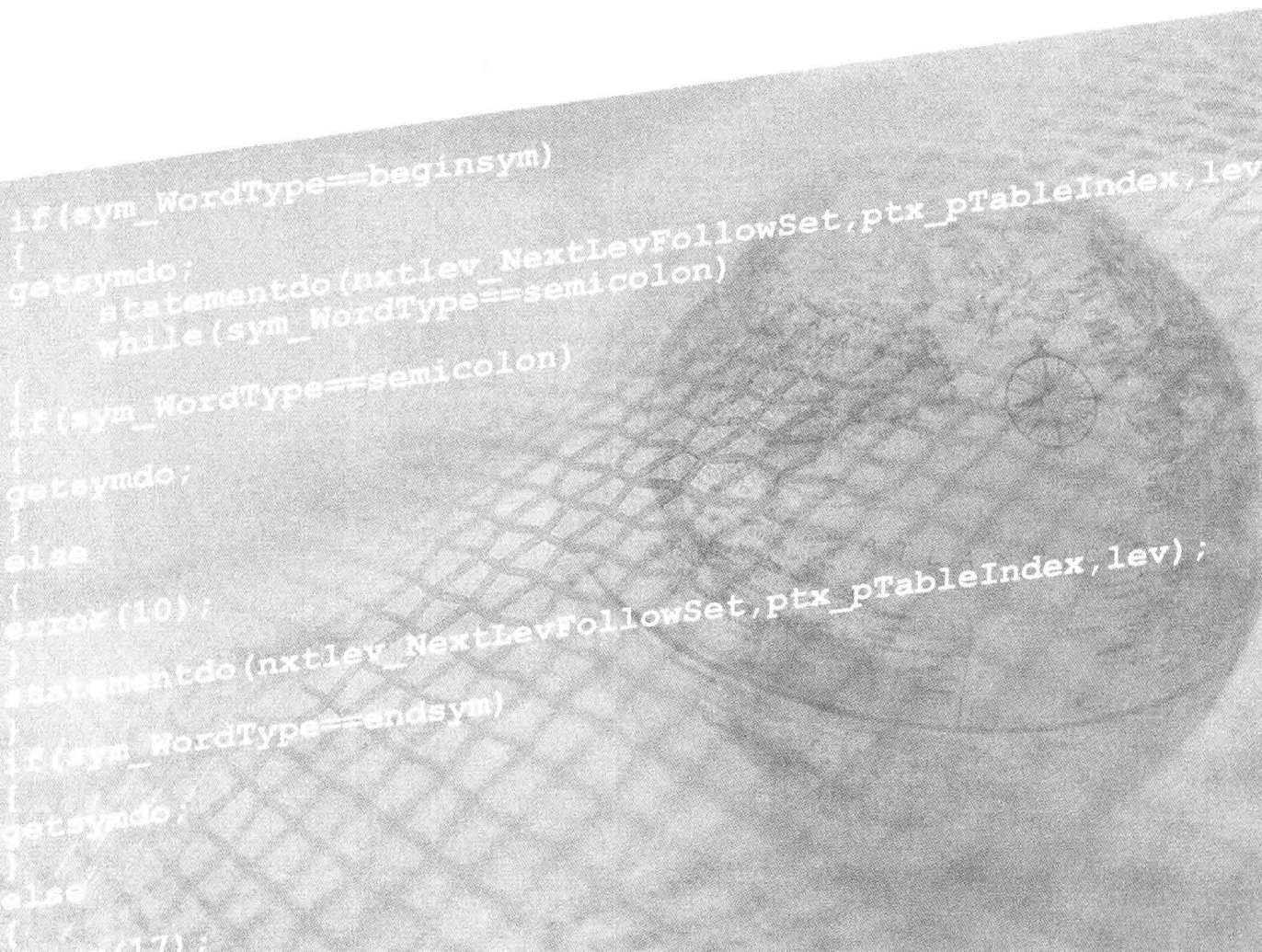
2013年9月

目　录

第 1 篇　编译原理学习指导

第 2 篇 编译原理实践指导

第 1 篇
编译原理学习指导



第 1 章　编译程序概述

1.1　本章知识结构图

本章知识结构图见图 1－1。

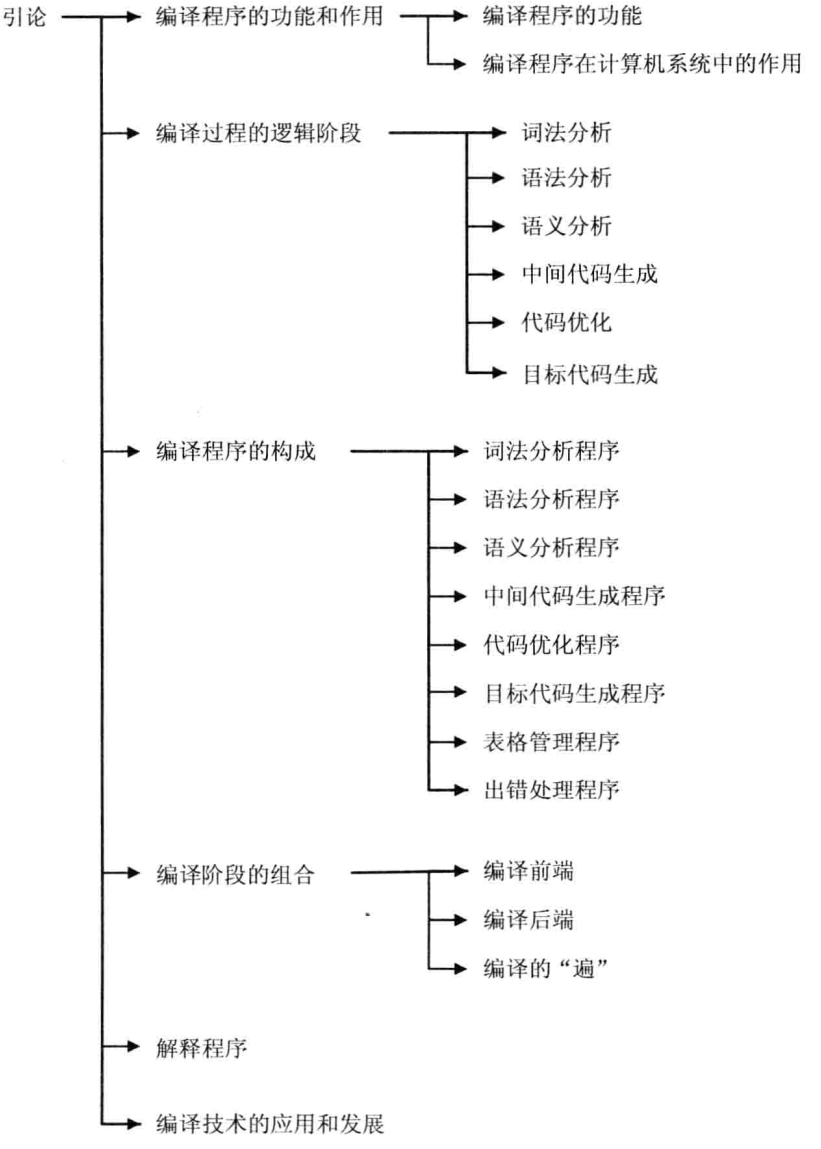


图 1－1　编译原理知识结构图

1.2 疑难解惑

1. 关于编译程序的理解

(1) 编译程序的基本任务是将源语言程序翻译成等价的目标语言程序。其中,输入数据是类似于 FORTRAN 或 C 的高级语言,输出结果则为类似于汇编语言或机器语言的低级语言。

(2) 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个阶段:编译阶段和运行阶段。

如果编译程序生成的目标程序是汇编语言程序,则源程序的执行分为三个阶段:编译阶段,汇编阶段和运行阶段。

2. 编译程序与解释程序的比较

(1) 编译方式与解释方式都是语言处理程序。对编译方式而言,编译和运行是两个相互独立的阶段。解释方式则不需要将这两个阶段分隔开,比较适用于交互式语言处理环境中。

(2) 编译方式与解释方式的根本区别在于是否生成目标代码。编译程序产生机器能识别的汇编或二进制代码,而解释程序则通过分析和执行语句后直接生成运行结果。

(3) 编译程序和解释程序的存储组织方式显著不同。对编译程序而言,在源语言程序被编译阶段,存储区中要为源程序(中间形式)和目标代码开辟空间,存放编译程序需要使用的各种各样表格,如符号表;在目标代码运行阶段,存储区中主要是目标代码和数据,编译所用的任何信息都不再需要。而对解释程序而言,在整个工作过程中,源程序、符号表等内容始终存放在存储区中。

(4) 编译程序执行效率比解释程序高。因为解释程序需要逐行进行翻译,循环体需重复翻译;每遇变量,须从头开始检索变量表;若遇转向语句(Goto),需从头开始检索符号表等。而编译程序不会出现上述三种情况,它是一次性翻译,可多次执行;编译过程中可向用户报告它检测到的一切错误。

3. 编译程序的工作过程

编译程序的工作过程一般可以划分为词法分析、语法分析、语义分析、中间代码生成、代码优化和目标代码生成六个部分,同时还伴有表格处理和出错处理。但事实上,并非所有的编译程序都分成这样几个阶段。有些编译程序并不需要生成中间代码,有些编译程序不进行代码优化,此时只有词法分析、语法分析、语义分析和目标代码生成是必需的。

4. 编译过程的前端、后端和遍

(1) 编译过程中阶段的划分是编译程序的逻辑组织。根据这些阶段所依赖对象的不同,常把编译的过程分为前端和后端。前端工作主要依赖于源语言,与目标机无关,后端工作则依赖于目标机而一般不依赖源语言。前端包括词法分析、语法分析、语义分析和中间代码生成,某些优化工作也可在前端做;与前端每个阶段相关的出错处理工作和符号表管理工作也属于前端。后端则包括目标代码生成以及相关出错处理和符号表操作。

按照这种组合方式，可以实现为不同的机器构成同一个源语言的编译程序（前端相同，后端不同），也可以为同一机器生成几个语言的编译程序（前端不同，后端相同）。这样使得编程工作量大大减少。

（2）遍：遍是指对源程序或等价的中间代码进行处理，完成规定任务的过程。每一遍可以完成上述编译程序六个阶段中的一个或多个阶段的工作，所以一个完整的编译程序可以通过多遍来完成。

编译程序应该分成几遍？主要参考两个因素：源语言特征；机器（目标机）的特征。

多遍编译程序的优点是少占内存，逻辑结构清晰；缺点是增加中间文件的读写次数，速度较慢。

1.3　典型例题

例 1.1 试问“解释程序对源程序直接运行并得到结果，所以并没有真正进行翻译”这种说法是否正确？

【相关知识】 两种程序翻译方式的特点及区别

【例题分析】 编译方式和解释方式对源程序实际上都进行了翻译，只是前者要产生目标代码，然后执行目标代码得到运行结果；而后者是在源程序或与源程序等价的中间代码上直接翻译，不生成目标代码。所以编译方式与解释方式的根本区别在于是否生成目标代码。

【例题答案】 这种说法是错误的。

例 1.2 在使用高级语言编程时，可通过编译程序发现源程序的全部_____错误和部分_____错误。

【相关知识】 编译程序各阶段的功能

【例题分析】 源程序中所有的语法错误都可以在编译阶段被发现，但有些语义错误是编译程序不能发现的。

【例题答案】 语法，语义

例 1.3 “数组下标越界”可能是编译的哪个阶段（词法分析、语法分析、语义分析和代码生成）报告的？

【相关知识】 编译程序各阶段的功能

【例题分析】 在检查数组下标是否越界时，源程序已经分解成一系列的单词，单词序列也组成相应的语法短语（数组），所以词法分析和语法分析是正确的。但运算对象（数组下标）超过了语言规定的范围，所以该错误可能是语义分析阶段报告的。

【例题答案】 语义分析阶段

1.4　习题

1. 填空题

（1）编译程序是一种常用的_____软件。

（2）一般的程序设计语言的定义都涉及语法、_____和语用三个方面。

（3）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_____，诸阶段的工作往往是_____进行的。

(4) 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_____、_____、_____三方面的内容。

2. 选择题

(1) (多选)编译程序必须完成的工作有_____。

A. 词法分析　　B. 语法分析

C. 语义分析　　D. 代码生成

E. 中间代码生成　　F. 代码优化

(2) 与编译系统相比，解释系统_____。

A. 比较简单，可移植性好，执行速度快

B. 比较复杂，可移植性好，执行速度快

C. 比较简单，可移植性差，执行速度慢

D. 比较简单，可移植性好，执行速度慢

(3) 编译程序生成的目标代码通常有三种形式：_____、_____、_____。

A. 可立即执行的机器语言代码　　B. 汇编语言代码

C. 待装配的机器语言代码模块　　D. 高级语言代码

(4) 无符号常数的识别和拼数工作通常都在_____阶段完成。

A. 词法分析　　B. 语法分析　　C. 语义分析　　D. 代码生成

3. 判断题

(1) 一个程序是正确的是指该程序的语法是完全正确的。　(　)

(2) 有的编译程序可以没有目标代码生成部分。　(　)

(3) 编译程序生成的目标程序不一定是机器语言的程序。　(　)

(4) 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。　(　)

(5) 多遍扫描的编译程序的多遍是指多次重复读源程序。　(　)

(6) 多遍扫描的编译程序优于单遍扫描的编译程序。　(　)

第 2 章　文法和语言

2.1　本章知识结构图

本章知识结构图见图 2-1。

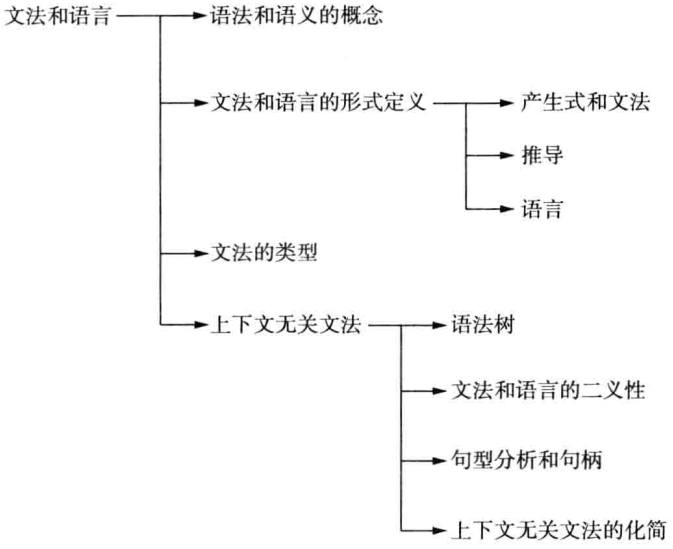


图 2-1　文法语言结构图

2.2　疑难解惑

1. 语法和语义的概念

语法和语义是程序设计语言的两个重要概念。语法是指一组规则，用来定义什么样的符号序列是合法的。语义则进一步从符号的意义上来判断，合法的符号序列是否构成正确的程序。类型匹配、变量作用域等在语法分析阶段无法用上下文无关手段检查，但可以利用语义分析来完成判断。

2. 乔姆斯基文法分类

乔姆斯基(Chomsky)于 1956 年建立了形式语言的描述，把文法分成四种类型，即 0 型(短语文法)，1 型(上下文有关文法)，2 型(上下文无关文法)，3 型(正规文法)。这几类文法的差别在于对产生式施加不同的限制。各文法的描述见表 2-1。

表 2-1　乔姆斯基文法分类

文法类别	产生式形式	产生语言	说明
0 型文法 （短语文法）	$\alpha\rightarrow\beta,\alpha\in V^+$ 且 α 至少含有一个非终结符，$\beta\in V^+$	0 型语言 （短语语言）	产生式左边至少含一个非终结符，其余基本无限制
1 型文法 （上下文有关文法）	$\alpha\rightarrow\beta$， $\lvert\alpha\rvert\geqslant\lvert\beta\rvert$，$S\rightarrow\varepsilon$ 除外	1 型语言 （上下文有关语言）	文法右部符号长度≥左部符号长度，也可描述为 $\alpha_1 A\alpha_2\rightarrow\alpha_1\beta\alpha_2$，则 A 与 α_1，α_2 上下文有关，α_1，α_2，$\beta\in V^*$，$\beta\neq\varepsilon$，$A\in V_N$
2 型文法 （上下文无关文法）	$\alpha\rightarrow\beta$， $\alpha\in V_N,\beta\in V^*$	2 型语言 （上下文无关语言）	也可描述为 $A\rightarrow\beta$，可看出 A 与上下文无关
3 型文法 （正规文法）	$A\rightarrow aB$ 或 $A\rightarrow a$， $A,B\in V_N,a\in V_T$	3 型语言 （正规语言）	文法中的产生式只有这两种情况

3. 语法树与推导

语法树是上下文无关文法句型推导过程的几何表示和直观工具，它表示了在推导过程中针对哪个非终结符使用了哪个产生式，但是并没有表明使用产生式的顺序。

对推导过程中每次替换的非终结符的顺序进行规定，就产生了最左（最右）推导。其中最右推导常称为规范推导，这和后续介绍的规范规约（最左规约）相对应。

虽然语法树没有表明推导的顺序，但与最左（最右）推导相联的语法树是一致的。也就是说，一棵语法树表示了一个句型的种种可能的（但未必是所有的）不同推导过程。

4. 文法二义性和语言二义性

一个句型不一定对应唯一的一棵语法树。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。

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，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。

消除文法的二义性通常有两种方法：

（1）在语义上增加一些限制。

（2）重新构造一个等价的无二义性的文法。

5. 文法的构造方法

为给定语言构造文法是判断一个文法所产生的语言的逆过程，一般来说没有确定的、通用的办法。通常的解题策略是“凑规则”，其具体步骤为：

（1）找出语言的若干句子；

（2）分析句子的特点；

（3）根据句子的特点凑规则；

(4) 形成文法；

(5) 检验文法。

若文法满足以下两点，则该文法就是与给定语言对应的文法：

(1) 语言的所有句子都能由文法的识别符号推导得到；

(2) 文法推导出的所有终结符号串都是语言的句子。

6. 语法分析方法

语法分析方法分为两大类：自顶向下分析法和自底向上分析法。

(1) 自顶向下分析法

思想：从文法的开始符号出发，反复使用各种产生式，逐步向下推导，试图推导出句子。

存在的问题：在推导中如何选择规则？采用回溯法可行，但是代价太高，还可能陷入死循环。

(2) 自底向上分析法

思想：从输入符号串开始，逐步进行归约，直到规约到文法的开始符号。

存在的问题：每次应归约当前句型的句柄，但如何找句柄，以及句柄是否唯一？

对一个句型的短语、直接短语和句柄的判断，常用的方法是：

(1) 查看语法树的叶子结点(终结符或非终结符)，如果叶子结点的父结点还有其他子结点，就将该父结点的所有子结点作为一个字符串集合来判断；

(2) 对子结点的判断要从左向右处理；向上判断短语是相对哪个非终结符的短语时，要一直处理到祖先结点。

2.3　典型例题

例 2.1 设有文法 G[S]：

S→a|ε|(T)

T→T,S|S

(1) 请给出句子(a,(a,a))的最左、最右推导和语法树；

(2) 句子(a,(a,a))的短语、直接短语、句柄；

(3) 试问(a,(a),)是不是 L(G[S])的句子？若是，请给出该句子的语法树；若不是，请给出理由。

【相关知识】 主要考查最左推导、最右推导、短语、直接短语、句柄和语法树的概念。

【例题分析】 如果在推导过程中的任何一步 $\alpha \Rightarrow \beta$，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。

如果 $S \overset{*}{\Rightarrow} \alpha A \delta$ 且 $A \overset{+}{\Rightarrow} \beta$，则称 β 是句型 αβδ 相对非终结符 A 的短语。特别地，若 $A \Rightarrow \beta$，则称 β 是句型 αβδ 相对于规则 A→β 的直接短语，一个句型的最左直接短语称为该句型的句柄。了解了这些概念，我们就可以进行答题了。

【例题答案】

(1) 最左推导：$S \Rightarrow (T) \Rightarrow (T,S) \Rightarrow (S,S) \Rightarrow (a,S) \Rightarrow (a,(T)) \Rightarrow (a,(T,S)) \Rightarrow (a,(S,S)) \Rightarrow (a,(a,S)) \Rightarrow (a,(a,a))$

最右推导：S⇒(T)⇒(T,S)⇒(T,(T))⇒(T,(T,S))⇒(T,(T,a))⇒(T,(S,a))⇒(T,(a,a))⇒(S,(a,a))⇒(a,(a,a))

语法树如图 2-2 所示：

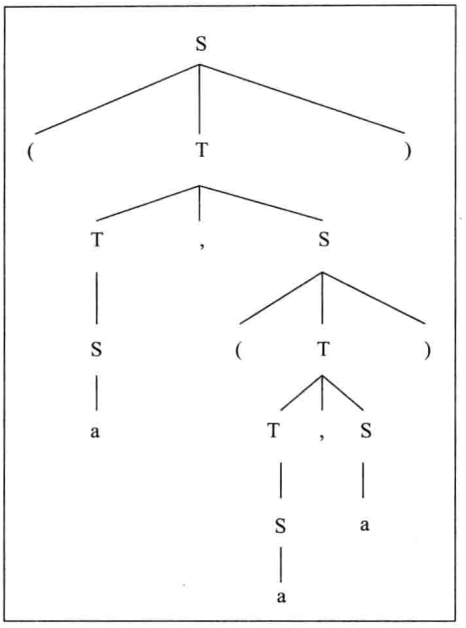


图 2-2 (a,(a,a))的语法树

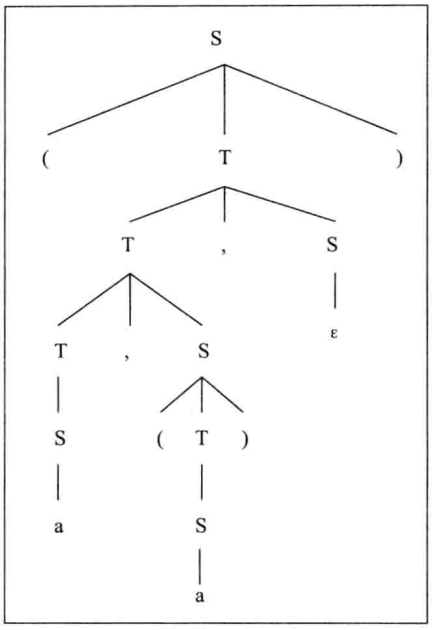


图 2-3 (a,(a),)的语法树

(2) 由语法树图 2-2 可知，

a 是句子(a,(a,a))相对于非终结符 S 的短语，也是相对于规则 S→a 的直接短语。

a 是句子(a,(a,a))相对于非终结符 T 的短语。

a,a 是句子(a,(a,a))相对于非终结符 T 的短语。

(a,a)是句子(a,(a,a))相对于非终结符 S 的短语。

a,(a,a)是句子(a,(a,a))相对于非终结符 T 的短语。

(a,(a,a))是句子(a,(a,a))相对于非终结符 S 的短语。

因此 a、a,a、(a,a)、a,(a,a)、(a,(a,a))都是句子(a,(a,a))的短语，而且 a 是直接短语，其中 a 是最左直接短语，即句柄。

(3) (a,(a),)是 L(G[S])的句子，语法树见图 2-3。

例 2.2 已知语言 $L=\{a^nbb^n \mid n\geq 1\}$

(1) 请构造产生该语言的相应的文法。

(2) 根据所求文法证明句子 aabbb 是否是语言 L 的句子，写出推导。

【相关知识】 主要考查从语言到正规式的凑规则。

【例题分析】 可以按照凑规则的 5 个步骤依次对该语言进行分析。

(1) 本例以语言的定义方式给出语言，所以找出语言的若干句子就比较简单。

(2) 对该语言进行分析可得句子 a^nbb^n 的特点是：句子中只含有 a、b 两种符号，且所有 a 在句子开头部分，所有 b 在句子尾部，且 a 和 b 以中间的一个 b 为中心对称出现。

(3) 根据句子的特点凑规则。

由于语言的句子中符号的个数可以有任意多个，所以必然要用到递归规则。同时，a 和 b 以中间的一个 b 为中心对称出现，因此只要让句子两边产生相同个数的 a、b，再最后在句子中