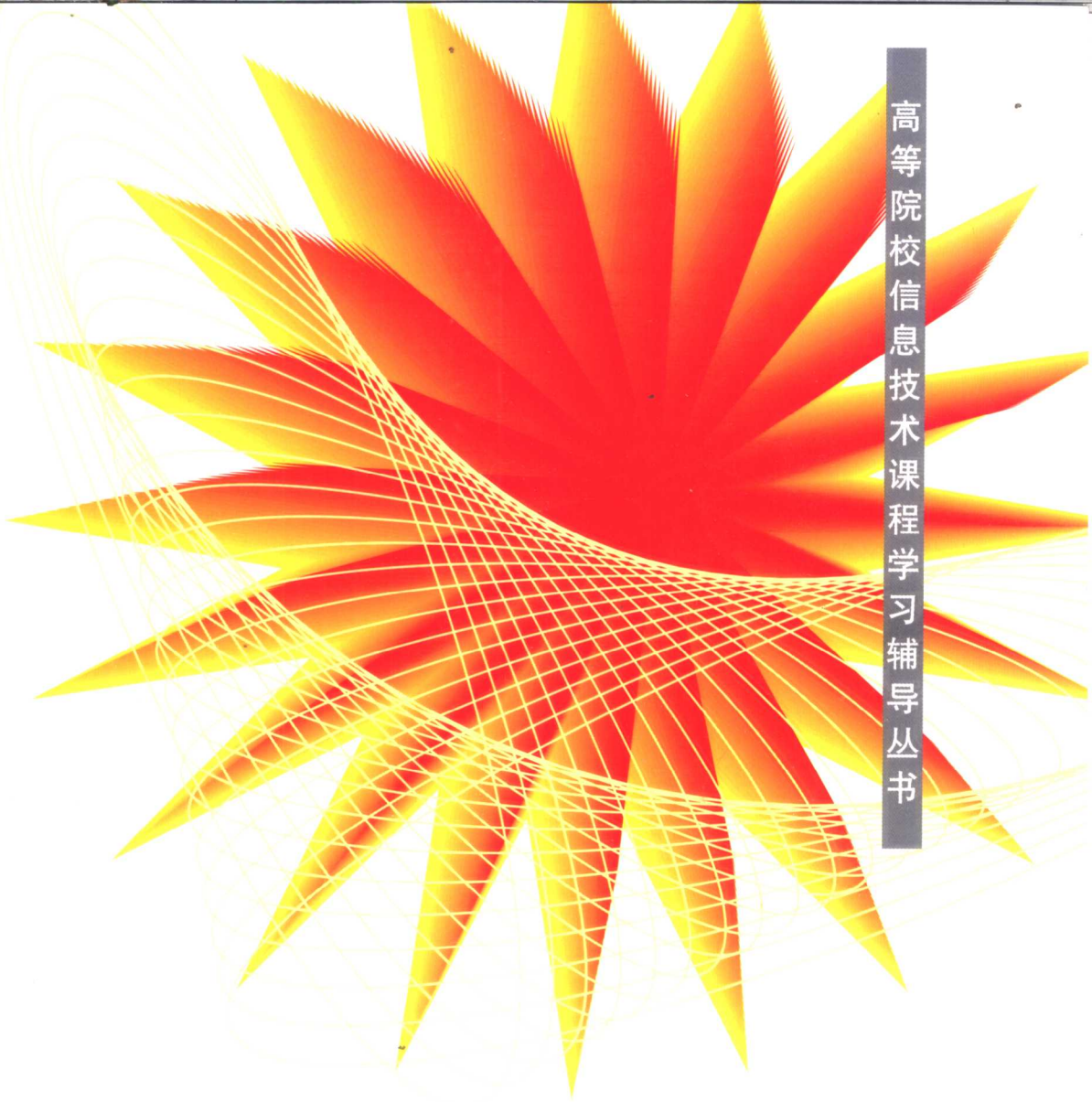

高等院校信息技术课程学习辅导丛书

编译原理学习辅导

张 伟 编著

清华大学出版社

高等院校信息技术课程学习辅导丛书

编译原理学习辅导

张 伟 编著

清华大学出版社
北京

内 容 简 介

本书针对“编译原理”课程理论性和实践性较强的特点，依据编者多年来教学实践的积累，选取了大量题目，并进行了分析解答。全书共分 9 章，基本覆盖了编译原理课程的主要内容，每章包括“知识要点”、“例题解析”、“习题及部分参考答案”三大部分，力求引导读者从理论到实践全面掌握编译技术的原理、概念和方法。

本书可作为计算机专业本科生的学习辅导用书，也可用于研究生入学考试的复习指导，还可供计算机软件开发人员参考阅读。

图书在版编目(CIP)数据

编译原理学习辅导/张伟编著. —北京：清华大学出版社，2005.2
(高等院校信息技术课程学习辅导丛书)
ISBN 7-302-10268-6

Ⅰ. 编… Ⅱ. 张… 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(2004)第 142637 号

出 版 者：清华大学出版社　　**地　　址**：北京清华大学学研大厦
http://www.tup.com.cn　　**邮　　编**：100084
社 总 机：010-62770175　　**客户服务**：010-62776969
组稿编辑：张　龙
文稿编辑：王冰飞
印 刷 者：北京市世界知识印刷厂
装 订 者：三河市新茂装订有限公司
发 行 者：新华书店总店北京发行所
开　　本：185×260　**印张**：17　**字数**：401 千字
版　　次：2005 年 2 月第 1 版　2005 年 7 月第2 次印刷
书　　号：ISBN 7-302-10268-6/TP · 6997
印　　数：4001～6000
定　　价：22.00 元

FOREWORD

前言

“编译原理”是计算机专业的一门主要专业课。通过对本课程的学习，不仅可以掌握编译程序本身的基本实现原理和技术，同时也有助于提高对程序设计语言的理解，提高语言设计及程序设计的能力。该课程涉及内容较广，具有很强的理论性与实践性，且内容抽象、较难理解，掌握起来有一定的难度。本书根据编者多年的教学实践，精选了该课程主要内容的典型习题，通过课程辅导与习题解析的方式帮助读者理解编译程序构造的基本原理和概念，掌握编译的相关技术，提高解题能力。

本书共分 9 章。第 1 章简要介绍程序设计语言的特点与编译的基本概念。第 2 章介绍词法分析器的设计，主要涉及正则表达式与有限自动机相关理论方法。第 3 章概要介绍了形式语言的基本概念、文法的定义及变换方法。第 4 章主要介绍语法分析的相关内容，包括自顶向下和自底向上两种语法分析方法；自顶向下分析法介绍了递归下降法和 LL 方法；自底向上分析法介绍了简单优先方法、算符优先方法和 LR 方法。第 5 章介绍符号表的组织，主要涉及语义信息的提取、存放和使用方法。第 6 章介绍语法制导翻译与中间代码生成的有关内容，给出了如何利用语法分析算法控制语义加工生成中间代码的方法。第 7 章重点介绍程序运行时存储空间组织及分配的相关内容。第 8 章主要介绍代码优化内容。第 9 章提供了本科期末考试试题和硕士研究生入学考试试题供读者参考。

为了便于读者学习，本书每一章内容分为三部分：第一部分介绍本章包含的内容和需要重点掌握的知识要点；第二部分对本章的典型例题进行较详细的分析和解答；第三部分提供了自测习题和部分自测习题的参考答案，目的是提高读者独立分析和解决问题的能力。

本书提供的习题难易程度较为适当，既可作为高校计算机专业本科生的学习辅导书，也可用于计算机相关专业研究生入学考试的复习指导。

本书的完成得益于编者多年来编译原理教学实践的积累，同时也与张基温教授大力支持和帮助密不可分，在此致以真诚的谢意。本书主要由张伟编写，康辉、张利华两位老师也参加了部分编写工作。由于编者水平有限，书中难免有错漏和不妥之处，敬请广大读者批评指正。

编　者

2004 年 10 月

CONTENTS

目 录

CHAPTER 1

第1章 编译程序概论

1.1 知识要点

1.1.1 程序设计语言

程序设计语言是人和计算机之间进行信息通信的载体，是用来编写程序的工具。程序设计语言从结构上可分为两大类。一类称为低级语言，包括机器语言和汇编语言。机器语言是机器的指令系统，它由二进制数字0、1序列组成，是计算机可以直接接受的语言。汇编语言是符号形式的指令系统，它用某些便于记忆的符号替代机器语言中不便记忆的二进制数字序列，这就大大方便了书写与阅读。另一类称为高级程序设计语言，包括FORTRAN、PASCAL、Java和C语言等数百种。大多数高级程序设计语言是用英文单词和数学公式按某种语法和语义规则构建而成，因此它是最接近自然语言的程序设计语言。

汇编语言虽然摆脱了机器语言中二进制编码的束缚，但在结构上并没有什么本质的改变，且与具体的机器有关。因此，汇编语言同机器语言一样都是非通用语言，而且机器不能够直接接受汇编语言，必须通过汇编程序将汇编语言程序转换成机器语言程序，才为计算机所接受。

高级程序设计语言是当今普遍采用的一类编程语言，它彻底摆脱了对计算机硬件的依赖性，是通用的程序设计语言。同自然语言一样，高级语言也是由词法、语法和语义三组规则定义的。词法规则定义了由语言的最小语法单位“符号”构成单词的规则，语法规则定义了由单词构成一个合法程序的规则，而语义规则定义了语言中单词符号和语法单位具有的意义。语言的词法规则和语法规则定义了一个程序的形式结构，语义规则定义了一个程序所代表的意义。

然而，高级语言程序也不能够直接在计算机上运行，必须通过编译程序将高级语言程序转换成机器语言程序，才能够在计算机上运行。

1.1.2 翻译程序和编译程序

翻译是指把某一种结构形式的语言转换成与之等价的另一种结构形式的语言。所谓等价是指两种语言的结构不同，但它们所代表的意义相同。而通常所说的翻译程序指的是，把某一种语言程序（即源程序）转换成与之等价的另一种语言程序（即目标程序）的程序。编译程序是把源程序（高级语言程序）转换（加工）成等价的目标程序（低级语言程序）

的程序，如图 1.1 所示。

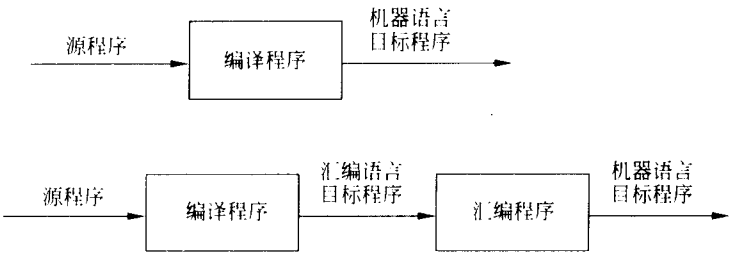

图 1.1　编译程序

1.1.3　解释方式和解释程序

解释方式是指在接受源程序的一条语句后，立即对该语句进行分析得到其包含的操作要求，并按其操作要求运行相应的操作例程，得到该语句的执行结果，然后再接受下一条语句，重复上述过程直至源程序结束，最后得到源程序的执行结果。解释程序就是按解释方式加工处理源程序的程序，即解释执行源程序的程序，如图 1.2 所示。

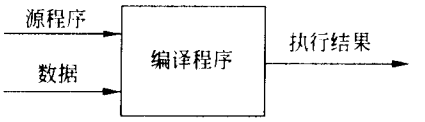

图 1.2　解释程序

解释程序在概念上与编译程序有明显的区别，解释程序是源程序的一个执行系统，而编译程序是源程序的一个转换系统。两者的加工对象相同，但其加工结果不同。解释程序的加工结果是源程序的执行结果，而编译程序的加工结果是与源程序等价的目标程序。

尽管解释程序和编译程序在功能上有明显的区别，但它们的实现技术并没有多大的差别，它们都要完成词法分析、语法分析和语义分析等项工作。如果说有差别的话，其差别仅在于一个是执行分析的结果，一个是把分析的结果转换成目标程序。

1.1.4　编译程序的功能结构

不同的编译程序其内部结构和组织方式是有差异的，它们都是根据源语言的具体特点和对目标程序的具体要求来决定和设计的。因此，并没有一种固定的编译程序结构模式，很难说哪种结构和组织方式是最好的，但从功能结构的角度讲都是一致的。所谓功能结构是指编译程序内部都完成哪些具体的任务，以及各项任务彼此之间的关系，如图 1.3 所示。

作为编译程序的输入，源程序只是一个较长的字符串，词法分析的主要任务是从源程序中识别出单词，并把识别出的单词转换成一种机内表达形式(TOKEN)，与此同时还要进行词法检查；语法分析以词法分析的输出(TOKEN)为输入，检查源程序中的语法错误，当发现错误时输出出错的位置和性质等信息；语义分析的主要任务是进行语义检查并

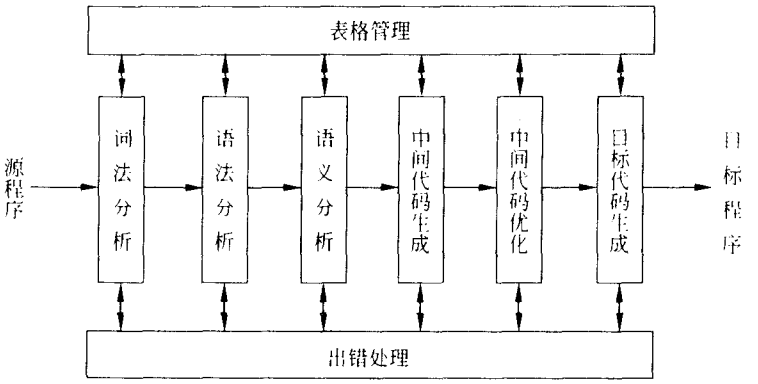

图 1.3 编译程序功能结构

从各种语法成分中提取语义信息且将其存放到相应的信息表中；中间代码生成的任务是生成介于源语言和目标语言之间的中间语言代码，便于生成目标代码和便于优化；中间代码优化的任务是进行不依赖于目标语言的优化，以产生高质量的目标代码；目标代码生成的任务是将中间代码翻译成机器语言程序或汇编语言程序。如果不设置中间代码生成阶段，则在语义分析之后将直接产生目标程序。

在编译程序的整个工作过程中，每个阶段都必须进行造表、查表和更新信息表工作，这些工作将占去相当大的一部分编译时间，因此，合理地组织编译程序中的各种信息表，恰当地选用造表和查表技术，有利于提高编译程序的工作性能。

编译过程的每个阶段都有检查错误的任务，其中，绝大多数错误可在编译的前三个阶段检查出来。源程序中的错误通常分为语法错误和语义错误两大类。语法错误是指源程序中不符合语法(或词法)规则的错误，它们可在词法分析或语法分析时检查出来。语义错误是指不符合语义规则的错误，又分为静态语义错误和动态语义错误。静态语义错误是在编译的语义分析阶段可查出的错误，动态语义错误是在目标程序运行时才能查出的错误。

1.1.5 编译程序的组织

前面介绍了编译程序的功能结构，描述的是编译程序内部的功能以及这些功能之间的关系，这里所说的关系是指各项任务之间的逻辑关系，而不一定是执行时间上的先后关系。事实上一个编译器的体系结构，可以按功能结构方式组织，也可按别的方式组织上述各项任务的工作，这在很大程度上依赖于编译过程中对编译的对象扫描的遍数，以及如何划分各遍扫描所进行的工作。所谓扫描是指对源程序或对其内部表示从头到尾扫视一遍，并进行有关的加工处理工作。

按扫描的遍数可把编译程序分为一遍扫描和多遍扫描两种。一遍扫描的编译程序是通过一遍扫描直接从源程序生成目标程序。对于一遍扫描的编译程序，通常以语法分析为中心来组织它的体系结构，如图 1.4 所示。

一遍扫描的优点是，由于不必产生中间代码，可以避免重复性工作，编译的速度快；缺

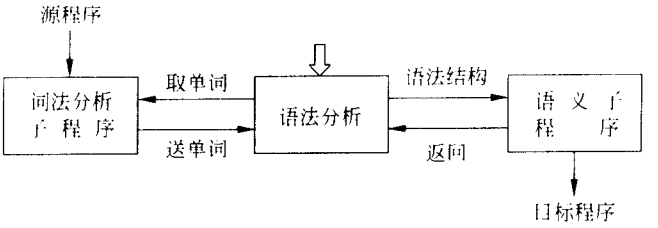

图 1.4 一遍扫描编译程序

点是当出现语法或语义错误时，前面所做的工作可能要半途而废，而且不便于分工和优化，其结构和算法不够清晰。

多遍扫描的编译程序是通过多遍扫描后才生成目标程序，每一遍扫描的输出是下一遍扫描的输入。多遍扫描的优点是结构和算法清晰，便于分工和优化；缺点是编译的速度慢，其原因是每一遍扫描都要从外存读取对象（源程序或中间代码），加工后产生的结果（中间代码或目标代码）还要写入外存中，所以遍数越多速度越慢。

1.1.6 编译程序的设计与实现

在某台计算机上为某种语言设计和实现一个高性能的编译程序，必须掌握三方面的技术：精通源语言、精通目标语言和精通编译技术，每一方面都非常重要。例如，设计者如果没有完全理解源语言的真正含义，那么构造出的编译程序不可能正确地反映语言的正确语义。

设计编译程序不但要考虑它的功能要求，还应考虑其性能要求和其他方面要求。性能主要指编译程序本身的质量和标准，包括编译的可靠性、编译的速度、目标程序的运行速度和空间的节省等。每个编译程序都有不同的特点，即性能要求是有所不同的，因此在分析阶段，应该明确待开发编译程序的主要性能要求以及用途，然后根据要求确定编译程序的组织结构和所要使用的技术。

确定编译程序的组织结构和各个部分的功能、方案后，就可以着手实现（编写）编译程序的各组成部分，这时涉及采用什么工具来编程的问题。最初级的方法使用汇编语言编写；其次用高级语言（如 C 语言）编写；更进步的方法是采用自动化系统（如 Lex、YACC 等）编写。而采用何种工具来实现，应根据现有的资源环境来决定。

1.2 例题解析

一、单项选择题

1. 高级程序设计语言是根据_____定义的。

A. 词法规则　　B. 语法规则　　C. 语义规则　　D. 以上三项都是

答案：D

2. 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

答案：B

3. 编译程序将源程序加工成目标程序是_____之间的转换。

A. 词法　　B. 语法　　C. 语义　　D. 规则

答案：C

4. 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. 是否生成目标程序

答案：D

5. 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

答案：D

6. 编译程序不能够检查、处理的错误是程序中的_____。

A. 静态语义错误　B. 动态语义错误　C. 语法错误　　D. 词法错误

答案：B

解析：编译程序在词法分析阶段检查和处理源程序中存在的词法错误；在语法分析阶段检查和处理源程序中存在的语法错误；在语义分析阶段检查和处理源程序中存在的语义错误，其中的语义错误指的是静态语义错误，动态语义错误只能在目标运行时才能够检查。故选 B。

二、判断题

1. 用高级语言编写的源程序都必须通过编译，产生目标程序后才能运行。

答案：错

解析：采用解释性方式处理的源程序，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. 源程序同目标程序是等价关系。

答案：正确

解析：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. 高级语言程序到低级语言程序的转换是结构上的变换。

答案：错

解析：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，而一个高级语言程序到另一个高级语言程序之间的变换，一般采取结构变换方法，如 PASCAL 语言程序到 C 语言程序之间的变换。

4. 多遍扫描的编译程序的多遍是指多次重复读源程序。

答案：错

解析：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扫描程序的加工结果，即某种形式的中间语言代码。

5. 解释程序虽然不产生目标程序，但它可能产生中间代码。

答案：正确

解析：尽管编译程序和解释程序在功能上有明显的区别，但从结构上看，好的解释程序和编译程序并没有过大的差别，它们都有词法分析、语法分析、语义分析和中间代码生成等工作。

三、问答题

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？

解析：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等。

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。解释方式一般分为两种：一种方式是，源程序功能的实现完全由解释程序承担和完成，即每读出源程序的一条语句的第一个单词，则依据这个单词把控制转移到实现这条语句功能的程序部分，该部分负责完成这条语句的功能的实现，完成后返回到解释程序的总控部分再读入下一条语句继续进行解释、执行，如此反复；另一种方式是，一边翻译一边执行，即每读出源程序的一条语句，解释程序就将其翻译成一段机器指令并执行之，然后再读入下一条语句继续进行解释、执行，如此反复。无论是哪种方式，其加工结果都是源程序的执行结果。目前很多解释程序采取上述两种方式的综合实现方案，即先把源程序翻译成较容易解释执行的某种中间代码程序，然后集中解释执行中间代码程序，最后得到运行结果。

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，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，而目标程序的执行任务由操作系统来完成，即只翻译不执行。

2. 一个典型的编译程序通常由哪些部分组成？各部分的主要功能是什么？

解析：一个典型的编译程序通常包含 8 个组成部分，它们是词法分析程序、语法分析程序、语义分析程序、中间代码生成程序、中间代码优化程序、目标代码生成程序、表格管理程序和错误处理程序。其各部分的主要功能简述如下。

词法分析程序：输入源程序，拼单词、检查单词和分析单词，输出单词的机内表达形式。

语法分析程序：检查源程序中存在的形式语法错误，输出错误处理信息。

语义分析程序：进行语义检查和分析语义信息，并把分析的结果保存到各类语义信息表中。

中间代码生成程序：按照语义规则，将语法分析程序分析出的语法单位转换成一定形式的中间语言代码，如三元式或四元式。

中间代码优化程序：为了产生高质量的目标代码，对中间代码进行等价变换处理。

目标代码生成程序：将优化后的中间代码程序转换成目标代码程序。

表格管理程序：负责建立、填写和查找等一系列表格工作。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信息多数都从表格中读取，

产生的中间结果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进行分析和处理。

1.3 习题及部分参考答案

一、单项选择题

1. 开发一个编译程序应掌握_____。
   A. 源语言　　B. 目标语言　　C. 编译技术　　D. 以上三项都是
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. 测试一个编译程序时使用的测试数据是_____。
   A. 源程序　　B. 中间代码　　C. 目标程序　　D. 任意数据
4. 可以作为目标代码的语言是_____。
   A. 高级语言　　B. 中间语言　　C. 低级语言　　D. 程序设计语言
5. 编译程序检查、处理源程序中的错误具体指的是_____。
   A. 词法错误　　B. 语法错误　　C. 语义错误　　D. 以上三项都是

二、判断题

1. 目标程序一定是机器语言程序。
2. 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。
3. 无论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。
4. 因为编译程序和解释程序具有不同的功能，所以它们的实现技术也完全不同。
5. 编译程序中错误处理的任务是对检查出的错误进行修改。

三、问答题

1. 何谓源程序、中间代码和目标程序？它们三者之间有何种关系？
2. 在计算机上运行高级语言源程序有哪些途径？它们之间的主要区别是什么？
3. 何谓一遍扫描和多遍扫描的编译程序？它们各自有什么特点？

参考答案

一、单项选择题

1. D　2. C　3. A　4. C　5. D

二、判断题

1. 错。也可以是汇编语言程序。

2. 正确。

3. 正确。

4. 错。尽管编译程序和解释程序在功能上有明显的区别，但其实现技术是基本相同的。具体地讲，解释方式也要使用词法分析、语法分析、语义分析甚至中间代码生成等编译技术。

5. 错。其任务是输出出错信息，包括错误的位置和性质等，同时进行适当的校正，以便继续下面的词法分析工作。

三、问答题

1. **解析**：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达形式，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。

2. **解析**：在计算机上运行高级语言源程序主要有两种途径：编译和解释。

解释方式一般分为两种：一种方式是，源程序功能的实现完全由解释程序承担和完成，即每读入源程序的一条语句的第一个单词，则依据这个单词把控制转移到实现这条语句功能的解释程序的程序部分，该部分负责完成这条语句的功能的实现，完成后返回到解释程序的总控部分再读入下一条语句继续进行解释、执行，如此反复；另一种方式是，一边翻译一边执行，即每读出源程序的一条语句，解释程序就将其翻译成一段机器指令并执行之，然后再读入下一条语句继续进行解释、执行，如此反复。无论是哪种方式，其加工结果都是源程序的执行结果。目前很多解释程序采取上述两种方式的综合实现方案，即先把源程序翻译成较容易解释执行的某种中间代码程序，然后集中解释执行中间代码程序，最后得到运行结果。

在编译方式下，编译程序只负责把源程序翻译成目标程序的工作，输出与源程序等价的目标程序，即只翻译不执行；而目标程序的执行任务由操作系统来完成。

两种途径的主要区别是其加工的结果不同，解释程序输出的是源程序的运行结果，而编译程序输出的是与源程序等价的目标程序。

3. **解析**：一遍扫描的编译程序是指只通过一遍扫描直接从源程序生成目标程序。

多遍扫描的编译程序是指通过多遍扫描后才生成目标程序，前一遍扫描的输出结果是后一遍扫描的输入，最后一遍扫描程序生成目标程序。

一遍扫描的特点是，编译过程中一般不产生中间代码，各项任务具有子程序结构，而子程序调用多数情况是在内存中直接实现。由于不产生中间代码程序，减少了编译过程中的读写外存（磁盘）的工作。因此，一遍扫描的编译程序编译的速度快。

多遍扫描的特点是，每一遍扫描程序各自完成的工作都是独立的，因此整个编译程序的逻辑结构和各遍扫描程序的算法都较为清晰，便于分工。而除最后一遍扫描程序外，每一遍扫描程序的输出均是中间代码，而优化是在中间代码一级上进行的，因此便于优化。

CHAPTER 2

第2章 有限自动机与词法分析

2.1 知识要点

2.1.1 词法分析的任务

词法分析程序亦称为词法分析器或扫描器，它的主要任务是从以字符为序列的源程序中识别出一个个具有独立意义的单词，并对识别出的单词进行词法和词义分析，得到各类单词的相关属性值，且把属性值加工成具有统一特定格式的表示形式——属性字，最终输出属性字，供语法和语义等其他加工程序使用。

词法分析器除完成上述主要任务之外，还应承担词法错误检查和校正的任务。但词法分析器所能发现的错误是极为有限的，实际上它只能检查出语言所不允许的错误符号，偶尔可发现某些单词后继字符的错误。

词法错误处理过程中遇到的主要问题是，当发现一个词法错误时不应立即停止词法分析，而是要想办法采取一定的补救措施，把词法分析过程继续下去。完成此项任务的工作称之为词法错误校正。

2.1.2 正则表达式与正则集

1. 基本概念

一个语言是由一些最小的不可再分的符号(字符)构成，字母表是包含构成一个语言的所有符号的非空有穷集合。字母表习惯上用Σ表示。符号串(也称为字)是由Σ上的符号构成的有穷序列。不包含任何符号的符号串称为空符号串，记为ε。符号串长度指符号串所包含的符号个数。

设α和β是符号串，则称$\alpha\beta$为符号串α和β的连接。特别对任一符号串β都有

$\beta\varepsilon=\varepsilon\beta=\beta$

设A和B是符号串集合，则称AB为符号串集合A和B的乘积。具体定义如下：

$AB=\{\alpha\beta \mid \alpha\in A,\beta\in B\}$

即乘积AB仍然表示符号串的集合，集合AB中的符号串是由A和B中的符号串通过连接运算而生成。设$\varnothing$为空集，由$\beta\varepsilon=\varepsilon\beta=\beta$，则对任一符号串集合A都有

$A\varnothing=\varnothing A=\varnothing$

$A\{\beta\}=\{\beta\}A=A$

符号串集合 A 的方幂记为 A^k，k 是非负整数。具体定义如下：

$A^0=\{\varepsilon\}$，$A^1=A$，$A^2=AA$，…，$A^k=AA\cdots A$（k 个 A 的乘积）

符号串集合 A 的正闭包记为 A^+，具体定义如下：

$A^+=A^1\cup A^2\cup A^3\cup\cdots$

符号串集合 A 的星闭包记为 A^*，具体定义如下：

$A^*=A^0\cup A^1\cup A^2\cup A^3\cup\cdots$

2. 正则表达式与正则集

符号串集合的乘积、方幂和闭包等运算给出了构造符号串集合的最基本的概念和方法。正则表达式（简称正则式）是指把基本运算按一定的运算规则组织而成的数学化的描述符号串集合方法，它所描述的集合称之为正则集，具体定义如下：

① ϕ 是定义在 Σ 上的正则表达式，它所代表的正则集为 $L(\phi)=\varnothing$。

② ε 是定义在 Σ 上的正则表达式，它所代表的正则集为 $L(\varepsilon)=\{\varepsilon\}$。

③ $a\in\Sigma$ 是定义在 Σ 上的正则表达式，它所代表的正则集为 $L(a)=\{a\}$。

④ 设 P 和 Q 都是定义在 Σ 上的正则表达式，它所代表的正则集分别为 L(P) 和 L(Q)，则有(P)、P | Q、PQ 和 P^+ 也都是定义在 Σ 上的正则表达式，它们所代表的正则集分别为 L((P)) = L(P)、L(P | Q) = L(P)∪L(Q)、L(PQ) = L(P) L(Q) 和 $L(P^+)=L(P)^+$。

综上所述，所谓正则表达式就是有限次应用上述四步骤定义构成的式子，由正则表达式所表示的集合才是字母表上的正则集。

设 P、Q 和 R 均是定义在 Σ 上的正则表达式，则下述定律成立：

① P|Q = Q|P	“或”运算的交换律
② P|(Q|R) = (P|Q)|R	“或”运算的结合律
③ P(QR) = (PQ)R	“乘积”运算的结合律
④ P(Q|R) = PQ|PR	“乘积”运算的分配律
⑤ (P|Q)R = PR|QR	“乘积”运算的分配律
⑥ $P^+=PP^*$	“闭包”运算的等价性

如果两个正则表达式 P 和 Q 描述的正则集相同，即 L(P)=L(Q)，则称这两个正则表达式 P 和 Q 等价。例如 $a^*=a^{**}$，$a^+=aa^*$，$(0|1)^*=(0^*1^*)^*$ 等。

2.1.3 有限自动机

有限自动机(FA)分为确定有限自动机(DFA)和非确定有限自动机(NFA)，而 DFA 是 NFA 的特例。可以通过构造有限自动机把正则表达式编译成词法分析程序。

1. 确定有限自动机(DFA)

一个确定有限自动机 DFA A 是一个五元组：$A=(S,\Sigma,f,S_0,F)$，其中：

S 是一个非空有限状态集合；

Σ 是有限输入字母表；