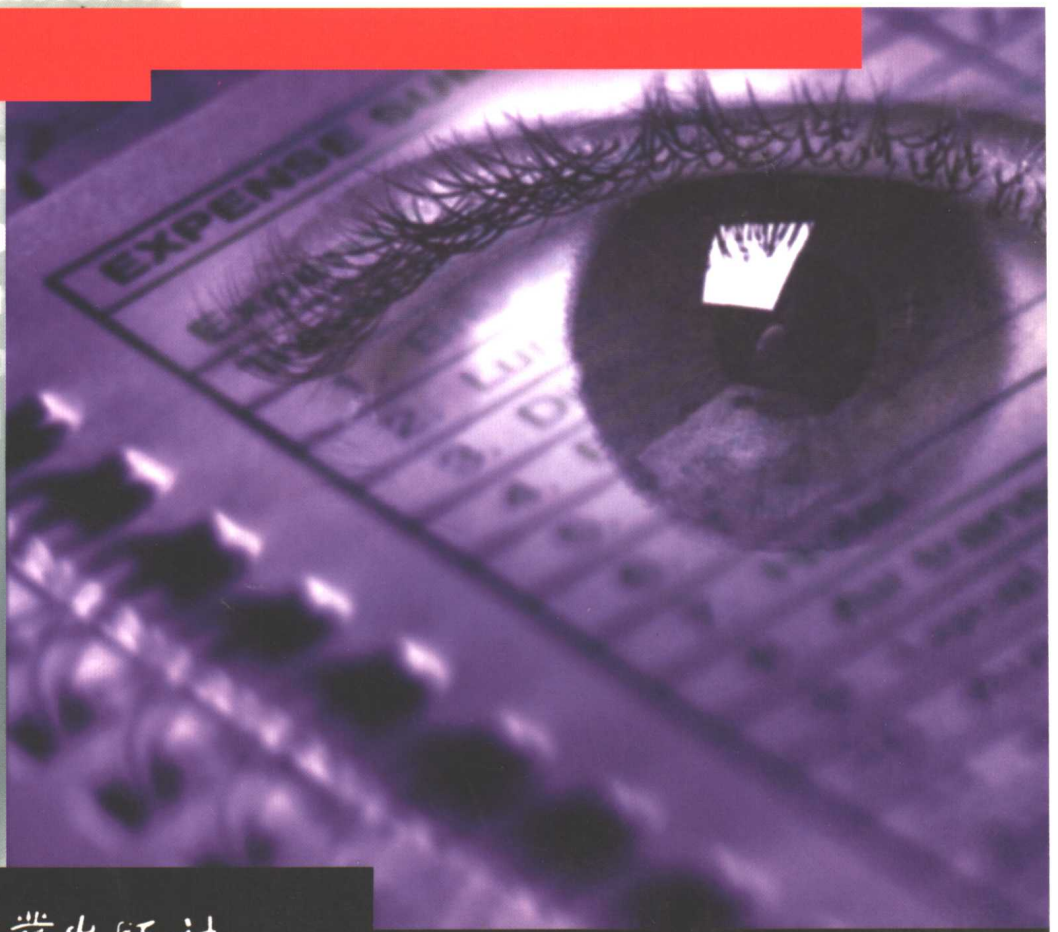
编译原理

胡伦骏　徐兰芳　刘建农　编
卢炎生　主审



電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http: //www.phei.com.cn

编译原理

胡伦骏　徐兰芳　刘建农　编

卢炎生　主审

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内容简介

本书系统介绍了编译程序的一般构造原理、基本设计方法和主要实现技术。内容包括语言的基础知识、词法分析程序的设计原理和构造方法、各种语法分析技术、属性文法的基本概念和中间代码生成、符号表的构造、代码优化、目标代码生成、并行编译技术常识及运行时存储空间的组织等。

本书系统性较强，基本概念阐述清晰，通俗易懂，便于自学。在各章之后均附有本章小结及习题，书后附有习题参考答案。

本书可作为高等院校计算机专业教材，也可作为成人高等教育计算机专业本科生和专科起点本科生的教材，对相关工程技术人员也有参考价值。

图书在版编目(CIP)数据

编译原理/胡伦骏，徐兰芳，刘建农编. —北京：电子工业出版社，2002.3
ISBN 7-5053-7509-1

Ⅰ.编… Ⅱ.①胡… ②徐… ③刘… Ⅲ.编译程序－程序设计－高等学校－教材 Ⅳ.TP314

中国版本图书馆 CIP 数据核字(2002)第 013166 号

责任编辑：童占梅　　特约编辑：逄积仁
印　　刷：北京大中印刷厂
出版发行：电子工业出版社　http://www.phei.com.cn
　　北京市海淀区万寿路 173 信箱　邮编 100036
经　　销：各地新华书店
开　　本：787×1092　1/16 印张：15.25　字数：390.4 千字
版　　次：2002 年 3 月第 1 版　2002 年 3 月第 1 次印刷
印　　数：5000 册　定价：20.00 元

凡购买电子工业出版社的图书，如有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系。
联系电话：(010)68279077

前　言

21世纪是信息时代，各行各业对计算机应用人才的需求越来越大，为了适应高等计算机教育迅猛发展的需要，在华中科技大学计算机学院和成人教育学院的组织下，编者根据多年来讲授本课程的教学经验，编写出适合高等教育计算机专业学生的“编译原理”教材。

“编译原理”课程是计算机专业的一门重要专业必修课。本书主要介绍设计和构造编译程序的一般原理、基本方法和主要实现技术。

在本书编写过程中，充分考虑到成人高等教育的特点，力求将基本概念、基本原理和实现方法的思路阐述清楚，条理清晰，通俗易懂，便于自学，为了帮助学生掌握每章的重点和难点，每章附有小结和习题，书末附有习题答案。

本教材参考学时数为60～70学时，书中主要算法、例题、习题均以C语言为背景，其主要内容包括：编译程序结构及各部分功能、文法和语言的基本概念和表示、词法分析、语法分析、属性文法与语法制导翻译技术、符号表、运行时存储空间的组织、代码优化与目标代码生成、并行编译技术概述等。

“编译原理”是一门实践性较强的课程，为了理论联系实际，本书在附录中拟定了实验内容、实验要求及实验参考算法。

本书的第1～4章由胡伦骏编写，第5～7章由徐兰芳编写，第8～10章以及附录由刘建农编写，全书由胡伦骏统稿。卢炎生教授、博导担任主审，在本书出版前，卢炎生教授仔细审阅了书稿，提出了很多宝贵意见，在此表示诚挚的谢意。在本书编写过程中还得到华中科技大学计算机学院、成人教育学院以及电子工业出版社的领导和有关同志的支持、关心和帮助，对此表示衷心的感谢。在成书过程中，编者参考了书末所列出的有关文献，在此也向这些书籍的作者一并表示感谢。

由于作者水平有限，错误与不妥之处在所难免，恳请各位读者不吝赐教。

作　者

目　录

第1章　编译概述

1.1　翻译程序与编译程序

语言是人与人之间传递信息的媒介和手段。世界上存在着多种语言，人们为了通信方便，建立了各种语言之间的翻译。人与计算机之间的信息交流，同样需要翻译。我们知道，每种计算机只懂得自己独特的指令系统，即它只能直接执行用机器语言编写的程序，这对人们来说很不方便，其原因是机器语言对计算机依赖性强、直观性差、编写程序工作量大、程序的结构也欠清晰。因此使用过现代计算机的人们多数都是用接近自然语言的高级程序设计语言来编写程序，但是计算机不能够直接接受和执行用高级语言编写的程序，需要通过一个翻译程序将它翻译成等价的机器语言程序才能执行。

所谓翻译程序是指这样一个程序，它把一种语言（称做源语言）所写的程序（源程序）翻译成与之等价的另一种语言（称做目标语言）的程序（目标程序），其功能如图 1.1 所示。

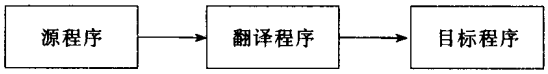


图 1.1　翻译程序功能

如果源语言是高级语言，如 Pascal，C，Ada，Java 语言等，目标语言是诸如汇编语言或机器语言之类的低级语言，那么称这样的翻译程序为编译程序，它所执行的转换工作如图 1.2 所示。

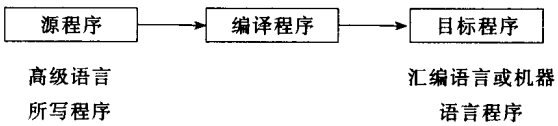


图 1.2　编译程序功能

可见，编译程序是一种翻译程序，它将高级语言所写的源程序翻译成等价的机器语言或汇编语言的目标程序。

采用编译方式在计算机上执行用高级语言编写的程序，需分阶段进行，一般分为两大阶段，即编译阶段和运行阶段，其过程如图 1.3 所示。

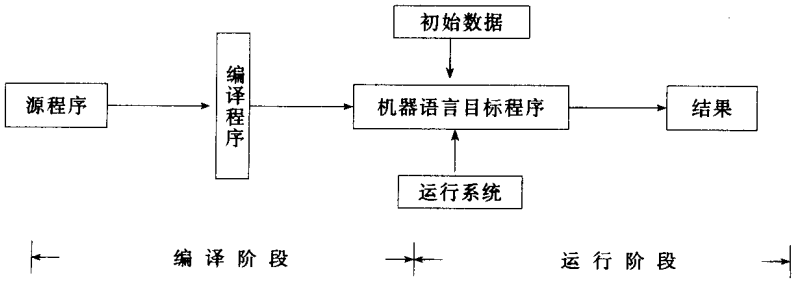


图 1.3　源程序的编译和运行阶段

如果编译阶段生成的目标程序不是机器语言程序，而是汇编语言程序，则程序的执行需分三个阶段，即编译阶段、汇编阶段和运行阶段，如图 1.4 所示。

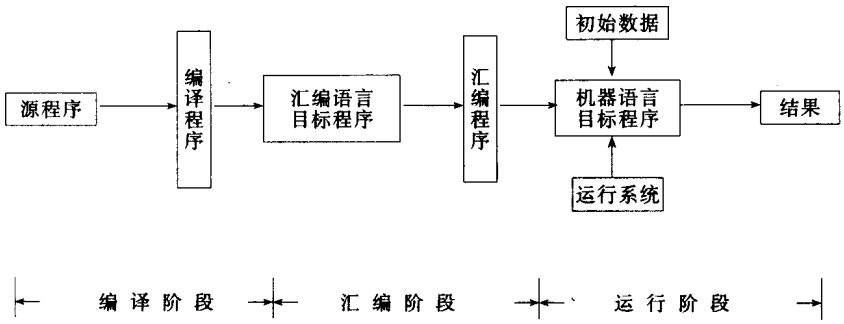


图 1.4　源程序的编译、汇编和运行阶段

用高级语言编写的程序也可通过解释程序来执行。解释程序也是一种翻译程序，它将源程序作为输入并执行之，即边解释边执行。它与编译程序的主要区别是在解释程序的执行过程中不产生目标程序，而是按照源语言的定义解释执行源程序本身。

本书主要介绍设计和构造编译程序的基本原理和方法。

1.2　编译过程和编译程序的基本结构

编译程序的功能是把用高级语言编写的源程序翻译成等价的机器语言或汇编语言表示的目标程序。既然编译过程是一种语言的翻译过程，那么它的工作过程就类似于外文的翻译过程。例如，我们要将英文句子“I wish you success ”翻译成中文句子，其翻译的大致过程是：

(1) 词法分析。根据英语的词法规则，从由字母、空格字符和各种标点符号所组成的字符串中识别出一个一个的英文单词。

(2) 语法分析。根据英语的语法规则，对词法分析后的单词串进行分析、识别，并做语法正确性的检查，看其是否组成一个符合英语语法的句子。

(3) 语义分析。对正确的英文句子分析其含义，并用汉语表示出来。

(4) 根据上下文的关系以及汉语语法的有关规则对词句作必要的修饰工作。

(5) 最后翻译成中文。

类似地，编译程序是将一种语言形式翻译成另一种语言形式，因此其工作过程一般可划分为下列五个阶段：词法分析、语法分析、语义分析及中间代码生成、代码优化和目标代码生成。

下面，我们以一个简单的程序段为例，分别介绍这五个阶段所完成的任务。

例如，计算圆柱体表面积的程序段如下：

```
float r,h,s;
s = 2*3.1416*r*(h+r)
```

第 1 阶段　词法分析

词法分析阶段的任务是对构成源程序的字符串从左到右进行扫描和分解，根据语言的词法规则，识别出一个一个具有独立意义的单词(也称单词符号，简称符号)。

词法规则是单词符号的形成规则，它规定了哪些字符串构成一个单词符号。上述源程序通过词法分析识别出如下单词符号：

基本字　float

标识符　r,h,s

常数　　3.1416,2

运算符　*,+

界符　(、)、;、,、=

第 2 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从单词符号串中识别出各种语法单位(如表达式、说明、语句等)并进行语法检查,即检查各种语法单位在语法结构上的正确性。

语言的语法规则规定了如何从单词符号形成语法单位,换言之,语法规则是语法单位的形成规则。

上述源程序,通过语法分解,根据语言的语法规则识别单词符号串 s=2*3.1416*r*(h+r),其中"s" 是〈变量〉,单词符号串"2*3.1416*r*(h+r)"组合成〈表达式〉这样的语法单位,则由〈变量〉=〈表达式〉构成〈赋值语句〉这样的语法单位。在识别各类语法单位的同时进行语法检查,可以看到,上述源程序是一个语法上正确的程序。

第 3 阶段　语义分析及中间代码生成

定义一种语言除要求定义语法外,还要求定义其语义,即对语言的各种语法单位赋予具体的意义。语义分析的任务是首先对每种语法单位进行静态的语义审查,然后分析其含义,并用另一种语言形式(比源语言更接近于目标语言的一种中间代码或直接用目标语言)来描述这种语义。例如,上述源程序中,赋值语句的语义为:计算赋值号右边表达式的值,并把它送到赋值号左边的变量所确定的内存单元中。语义分析时,先检查赋值号右边表达式和左边变量的类型是否一致,然后再根据赋值语句的语义,对它进行翻译可得到如下形式的四元式中间代码:

(1)　(*,　2,　3.1416,　T_1)

(2)　(*,　T_1,　r,　T_2)

(3)　(+,　h,　r,　T_3)

(4)　(*,　T_2,　T_3,　T_4)

(5)　(=,　T_4,　—,　s)

其中,T_1,T_2,T_3,T_4 是编译程序引进的临时变量,存放每条指令的运算结果。上述每一个四元式所表示的语义为:

2　*　3.1416　⇒　T_1

T_1　*　r　⇒　T_2

h　+　r　⇒　T_3

T_2　*　T_3　⇒　T_4

T_4　⇒　s

这样,我们将源语言形式的赋值语句翻译为四元式表示的另一种语言形式,这两种语言在结构形式上是不同的,但在语义上是等价的。

第 4 阶段　代码优化

代码优化的任务是对前阶段产生的中间代码进行等价变换或改造,以期获得更为高效的,即省时间和空间的目标代码。优化主要包括局部优化和循环优化等,例如上述四元式经局部

优化后得：

(1) (*, 6.28, r, T_2)
(2) (+, h, r, T_3)
(3) (*, T_2, T_3, T_4)
(4) (=, T_4, —, s)

其中，2 和 3.1416 两个运算对象都是编译时的已知量，在编译时就可计算出它的值 6.28，而不必等到程序运行时再计算，即不必生成(*，2，3.1416，T_1)的运算指令。

第 5 阶段　目标代码生成

目标代码生成的任务是将中间代码变换成特定机器上的绝对指令代码或可重定位的指令代码或汇编指令代码。

在编译程序的各个阶段中，都要涉及到表格管理和错误处理。编译程序在工作过程中需要建立一些表格，以登记源程序中所提供的或在编译过程中所产生的一些信息，编译各个阶段的工作都涉及到构造、查找、修改或存取有关表格中的信息，因此在编译程序中必须有一组管理各种表格的程序。一个好的编译程序在编译过程中，应具有广泛的程序查错能力，并能准确地报告错误的种类及出错位置，以便用户查找和纠正，因此在编译程序中还必须有一个出错处理程序。

编译过程的这五个阶段的任务分别由五个程序完成，这五个程序分别称为词法分析程序、语法分析程序、语义分析及中间代码生成程序、代码优化程序和目标代码生成程序，另外再加上表格管理程序和出错处理程序。这些程序便是编译程序的主要组成部分，一个典型的编译程序结构如图 1.5 所示。

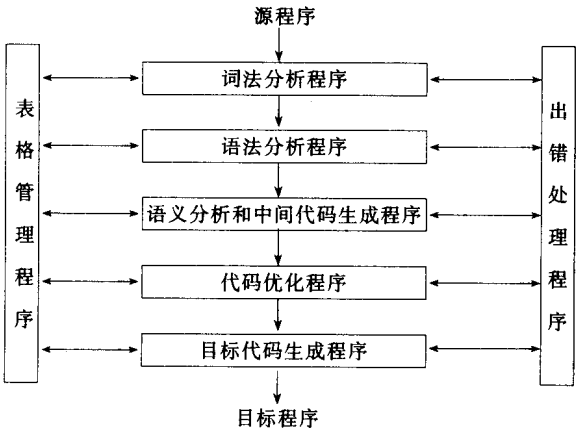


图 1.5　编译程序结构框图

需要注意的是，图中所给出的各个阶段之间的关系是指它们之间的逻辑关系，不一定是执行时间上的先后关系。实际上，可按不同的执行流程来组织上述各阶段的工作，这在很大程度上依赖于编译过程中对源程序扫描的遍数以及如何划分各遍扫描所进行的工作。此处所说的“遍”，是指对源程序或其等价的中间语言程序从头到尾扫描一遍，并完成规定加工处理工作的过程。例如，可以将前述五个阶段的工作结合在一起，对源程序从头到尾扫描一遍来完成编译的各项工作，这种编译程序称为一遍扫描的编译程序。对于某些程序设计语言，用一遍扫描的编译程序去实现比较困难，可采用多遍扫描的编译程序结构，每遍可完成上述某个阶段的一部

分、全部或多个阶段的工作，且每一遍的工作是从前一遍获得的工作结果开始的，最后一遍的工作结果是目标语言程序，第一遍的输入则是用户书写的源程序。

多遍扫描的编译程序较一遍扫描的编译程序少占存储空间，遍数多一些，可使各遍所要完成的功能独立而单纯，其编译程序逻辑结构清晰，但遍数多势必增加输入输出开销，这将降低编译效率。一个编译程序究竟应分成几遍和它所面临的源语言的特征、机器规模、设计的目标等因素有关，很难统一划定。一般在主存可能的前提下，还是遍数少一点为好。

1.3 编译程序的生成方法

编译程序是一个复杂的系统程序，要生成一个编译程序，一般要考虑下面几个方面。

1. 对源语言和目标语言认真分析

编译程序的功能是把某语言的源程序翻译成某台计算机上的目标程序。因此，我们首先要熟悉源语言(如C语言)，要正确理解它的语法和语义；其次要搞清楚目标语言和目标机的性质，在此基础上，确定编译程序的结构和所采用的具体策略。

2. 设计编译算法

设计编译算法是构造编译程序过程中最关键的一步。把一种语言程序翻译成另一种语言程序的方法很多，但在算法设计中要着重考虑如何使编译程序具有易读性、易改性和易扩充性。

3. 选择语言编制程序

根据所设计的算法选用某种语言(如机器语言、汇编语言、Pascal或C语言等)编写出编译程序。早期人们使用机器语言或汇编语言并用手工方式编写编译程序，虽然目标程序效率高但可靠性差，不便于阅读、修改和移植，从20世纪80年代开始，几乎所有编译程序都用高级语言来编写，这样可以提高开发效率，而且构造出来的编译程序增加了易读性、易修改和可移植性。

4. 调试编译程序

通过大量实例对编写好的编译程序进行调试，调试过程中不断修改、完善编译程序。

5. 提交相关文档资料

为方便用户，需提交一份有关编译程序的文档资料，内容包括源语言的文法、目标机指令系统、编译程序结构和所采用的具体策略、错误信息表及使用说明等。

我们希望能有一个自动生成编译程序的软件工具，只要把源程序的定义以及机器语言的描述输入到这个软件工具中去，就能自动生成该语言的编译程序。

随着编译技术和自动机理论的发展，近年来已研制出了一些编译程序的自动生成系统。如目前已广泛使用的词法分析程序自动生成系统LEX和语法分析程序自动生成系统YACC等，此外，还有可用来自动产生整个编译程序的软件工具——编译程序产生器，它的功能是将任一语言的词法规则、语法规则和语义解释的描述作为输入，自动生成该语言的编译程序。

生成编译程序的方法还常采用自编译方式和移植方式。采用自编译方式生成编译程序的思想是先用目标机的汇编语言或机器语言对源语言的核心部分构造一个小小的编译程序(可用手工实现)，再以它为工具构造一个能够编译更多语言成份的较大编译程序。如此扩展下去，就像滚雪球一样，越滚越大，最后生成人们所期望的整个源语言的编译程序。

通过移植生成编译程序的思想是把某机器上已有的编译程序移植到另一台机器上去。使

用交叉编译技术也可生成编译程序，所谓交叉编译是指一个源语言在宿主机(运行编译程序的计算机)上经过编译产生目标机的机器语言或汇编语言代码。

随着并行技术和并行语言的发展，处理并行语言的并行编译技术和将串行程序转换成并行程序的自动并行编译技术正在深入研究之中。

1.4 编译技术在软件开发中的应用

虽然只有少数人从事构造或维护程序语言编译程序的工作，但是大部分系统软件和应用软件的开发，通常要用到编译原理和技术。例如，设计词法分析器的串匹配技术已用于正文编辑器、信息检索系统和模式识别程序，上下文无关文法和语法制导定义已用于创建诸如排版、绘图系统和语言结构化编辑器，代码优化技术已用于程序验证器和从非结构化的程序产生结构化程序的编程之中。通常，在软件开发过程中，我们需要将某种语言开发的程序转换成另一种语言程序，这种转换过程和编译程序的工作过程是类似的，需要对被转换的语言进行词法分析和语法分析，只不过生成的目标语言不一定是可执行的机器语言或汇编语言。

由于编译原理和技术在软件工程的许多领域中被广泛地应用，因此编译原理和技术是一切从事计算机软件开发和研究的科学家和工程技术人员必须具备的专业基础知识。

本章小结

本章重点介绍了什么是编译程序及编译程序的结构。

编译程序是一种翻译程序，它将高级语言所写的源程序翻译成等价的机器语言或汇编语言的目标程序。

整个编译过程可以划分为五个阶段：词法分析、语法分析、语义分析及中间代码生成、代码优化和目标代码生成。

编译程序的结构按上述五个阶段的任务分模块进行设计。一个典型编译程序的结构框图如图 1.5 所示。

习　题

1.1　什么是编译程序？

1.2　编译过程的五个阶段是什么？

1.3　请给出编译程序的结构框图。

第 2 章　文法和语言的基本知识

2.1　概　　述

在第 1 章我们介绍过编译程序，它的功能是将高级语言所写的源程序翻译成与之等价的机器语言或汇编语言的目标程序。也就是说，我们所要构造的编译程序是针对某种程序设计语言的，编译程序要对它进行正确地翻译，首先要对程序设计语言本身进行精确地定义和描述。对程序设计语言的描述是从语法、语义和语用三个因素来考虑的。所谓语法是对语言结构的定义；语义是描述了语言的含义；语用则是从使用的角度去描述语言。

例如，对于赋值语句 s=2 * 3.1416 * r * (r+h)的非形式化的描述如下：

语法——赋值语句由一个变量、一个后随赋值号“=”、及其后跟一个表达式构成。

语义——首先计算语句右部表达式的值，然后把所得结果送入左部变量中。

语用——赋值语句可用来计算和保存表达式的值。

这种非形式化的描述，不够清晰和准确，为了精确定义和描述程序设计语言，需采用形式化的方法。所谓形式化的方法，是用一整套带有严格规定的符号体系来描述问题的方法。这种方法正是著名的语言学家 Noam Chomsky 在 1956 年提出的形式语言理论中所研究的问题，也就是说，形式语言理论是编译的理论基础，因此在这一章将介绍编译理论中用到的有关形式语言的某些基本概念和知识。

2.2　字母表和符号串的基本概念

2.2.1　字母表和符号串

1. 字母表

字母表是元素的非空有穷集合。

例如，$\Sigma = \{a,b,c\}$。

根据字母表的定义，Σ 是字母表，它由 a,b,c 三个元素组成。

需要注意的是，字母表中至少包含一个元素。字母表中的元素，可以是字母、数字或其他符号。

例如，$\Sigma' = \{0,1\}$ 是一个字母表，由 0 和 1 两个元素组成。

不同的语言有不同的字母表，如英文的字母表是 26 个字母、数字和标点符号的集合，C 语言的字母表是由字母、数字和若干专用符号组成。任何语言的字母表指出了该语言中允许出现的一切符号。

2. 符号(字符)

字母表中的元素称为符号，或称为字符。

例如，前述例子中 a,b,c 是字母表 Σ 中的符号；0 和 1 是字母表 Σ' 中的符号。

3. 符号串(字)

符号的有穷序列称为符号串。

例如,设有字母表 $\Sigma = \{a, b, c\}$,则有符号串 $a, b, ab, ba, cba, abc\cdots$

符号串总是建立在某个特定字母表上的且只能由字母表上的有穷多个符号组成。需要指出的是,符号串中符号的顺序是很重要的,如 ab 和 ba 是字母表 Σ 上的两个不同的符号串。不包含任何符号的符号串,称为空符号串,用 ε 表示,即空符号串由 0 个符号组成,其长度 $|\varepsilon| = 0$。

2.2.2 符号串的运算

1. 符号串的连结

设 x 和 y 是符号串,则串 xy 称为它们的连结。即 xy 是将 y 符号串写在 x 符号串之后得到的符号串。

例如,设 $x = abc, y = 10a$,则 $xy = abc10a, yx = 10aabc$。

注意:对任意一个符号串 x,我们有 $\varepsilon x = x\varepsilon = x$。

2. 集合的乘积

设 A 和 B 是符号串的集合,则 A 和 B 的乘积定义为:

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

例如,设 $A = \{a, b\}, B = \{c, d\}$,则 $AB = \{ac, ad, bc, bd\}$。

集合的乘积是满足于 $x \in A, y \in B$ 的所有符号串 xy 所构成的集合。

由于对任意的符号串 x,总有 $\varepsilon x = x\varepsilon = x$,所以,对任意集合 A,有

$$\{\varepsilon\}A = A\{\varepsilon\} = A$$

特别指出的是,ε 是符号串,不是集合,而 $\{\varepsilon\}$ 表示由空符号串 ε 所组成的集合,但这样的集合不是空集合 $\varnothing = \{\}$。

3. 符号串的幂运算

设 x 是符号串,则 x 的幂运算定义为

$$x^0 = \varepsilon$$

$$x^1 = x$$

$$x^2 = xx$$

……

$$x^n = \underbrace{xx\cdots x}_{n} = xx^{n-1} \quad (n > 0)$$

例如,设 $x = abc$,则

$$x^0 = \varepsilon$$

$$x^1 = abc$$

$$x^2 = xx = abcabc$$

……

4. 集合的幂运算

设 A 是符号串的集合,则集合 A 的幂运算定义为

$$A^0 = \{\varepsilon\}$$

$$A^1 = A$$

$$A^2 = AA$$