形式语言与自动机

陈文宇 欧 齐 程 炼 编著

人民邮电出版社
POSTS & TELECOM PRESS

形式语言与自动机

陈文宇 欧 齐 程 炼 编著

人民邮电出版社

图书在版编目（CIP）数据

形式语言与自动机/陈文宇，欧齐，程炼编著．
—北京：人民邮电出版社，2005．8

ISBN 7-115-13518-5

Ⅰ．形...　Ⅱ．①陈...②欧...③程...　Ⅲ．①形式语言②自动机理论
Ⅳ．TP301

中国版本图书馆 CIP 数据核字（2005）第 064256 号

内容提要

本书系统论述了形式语言和自动机的基础理论，从语言的产生角度和识别角度对 Chomsky 的短语结构文法（包括右线性文法、上下文无关文法、上下文相关文法和无限制文法）以及自动机（包括有限状态自动机、下推自动机和图灵机）进行讨论。并介绍了文法与自动机之间的等价关系。另外，还介绍了语法分析中一些基本的问题和语言（程序设计语言及自然语言）语法结构的描述方法。

本书以新的思维方式为读者提供一把钥匙，主要培养读者的独立思考能力，使用符号化的系统描述程序设计语言或自然语言的语法结构的能力，对语言进行语法分析的能力，以及构造自动机的能力，以适应计算机科学不断发展的需要。

实际上，形式语言与自动机理论除了在计算机科学与技术领域的直接应用外，更在计算机科学与技术领域人才的计算思维的培养中占有极其重要的地位。

本书可作为高等学校计算机科学应用专业、软件专业本科和研究生的教材或参考书，也可作为计算机应用领域内广大科技人员提高理论素质的参考书。

形式语言与自动机

◆ 编　著　陈文宇　欧　齐　程　炼
责任编辑　邹文波

◆ 人民邮电出版社出版发行　北京市崇文区夕照寺街 14 号
邮编　100061　电子函件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京隆昌伟业印刷有限公司印刷
新华书店总店北京发行所经销

◆ 开本：787×1092　1/16
印张：16.5
字数：398 千字　　2005 年 8 月第 1 版
印数：1-3 000 册　　2005 年 8 月北京第 1 次印刷

ISBN 7-115-13518-5/TP・4715

定价：27.00 元

读者服务热线：(010)67170985　印装质量热线：(010)67129223

前　言

形式语言和自动机的理论是计算机科学的理论基础。这些理论来源于：

（1）Chomsky 对自然语言的研究；

（2）巴科斯和诺尔使用巴科斯-诺尔范式(BNF)对 ALGOL 60 语言的语法规则进行描述；

（3）Kleene 在研究神经细胞时建立的自动机模型。

形式语言理论的研究对象与以前所有语言的研究对象不同，不止是自然语言，而是人类的一切语言：既有自然语言，也有人工语言，包括高级程序设计语言。

形式语言和自动机的理论已经成为计算机科学的理论基础，其应用范围已被扩展到生物工程、自动控制系统、图像处理与模式识别等许多领域。

国内只有为数不多的大学开设该类课程，教材内容也比较分散。电子科技大学早在 1979 年，就为本科生开设了“形式语言与自动机”课程，使用国外的英文教材，将形式语言与自动机理论和计算理论一起介绍，偏重于理论的证明过程，内容过难，后来使用自编讲义进行教学。

本书不注重定理繁琐的证明过程，而强调问题的思考方法和思路的研究，以提高读者的创新思维能力。

全书共分为两部分(共 9 章)。第 1～5 章作为第一部分。第 1 章介绍本书所需要的基本数学知识。第 2～4 章是形式语言的基本内容，包括文法的定义、分类，文法的构造方法，以及语言运算的封闭性的讨论。其中，对于高级程序设计语言所使用的上下文无关文法的性质进行重点的讨论。第 5 章介绍语法分析的理论基础。其余 4 章作为第二部分，是有限自动机的内容。第 6、7 章介绍有限状态自动机的构造方法及其对应的正则语言的性质，第 8 章介绍下推自动机，第 9 章是对图灵机的讨论。

本书力求使各类计算机专业的本科学生，掌握程序设计语言的词法和语法规则的描述方法，掌握语法分析的基本的理论，为“编译原理”的学习奠定基础；使各类计算机专业的研究生掌握有限自动机的构造方法，培养学生的计算思维能力。

本书基本覆盖了形式语言和自动机的所有内容，可以作为计算机专业本科和研究生的教材。在本科阶段作为“编译原理”课程的先修课程，主要介绍形式语言的内容和基本的语法分析技术。在研究生阶段，着重于自动机的模型和构造方法和技巧，以及形式语言与自动机的等价理论。

本书是作者在电子科技大学计算机科学和工程学院的“形式语言与自动机”讲义的基础上，大量阅读关于形式语言和自动机理论方面的文献资料，并结合近 15 年的实际教学经验编著而成的。第 1～4、6～9 章由陈文宇编著，第 5 章由欧齐、程炼编写。

在此，谨对参阅文献的作者和翻译人员表示衷心感谢。

特别感谢龚天富教授在百忙之中抽出时间认真审阅了全稿，并提出了许多宝贵的意见。

本书提供相关的课件，请读者与作者联系索取(E-mail：cwy@uestc. edu. cn)。

由于时间仓促，水平有限，书中难免有错误之处，敬请广大读者批评指正。

作　者

目　录

第1章 预备知识

计算机学科的方法论有 3 个过程:抽象、理论和设计及实现。最根本的问题是:问题如何进行描述?哪些部分能够被自动化?如何进行自动化描述?

问题的计算机求解建立在高度抽象的基础上,问题的符号表示及处理过程的机械化、严格化等固有特性决定了数学是计算机科学与技术学科的重要基础之一。数学及其形式化描述以及严密的表达和计算,是计算机科学与技术学科的重要工具。建立物理符号系统并对其实施变换是计算机科学与技术学科进行问题描述和求解的重要手段。学科所要求的计算机问题求解的"可行性"限定了从问题抽象开始到根据适当理论的指导进行实现的科学过程。

本书内容属于计算机科学的理论范畴,所需的数学基础知识较多。

本章将对形式语言和有限自动机理论中所需的数学基础知识做简要的介绍,内容包括集合及其运算、关系、证明和证明的方法以及图与树的概念。

1.1 集合及其运算

集合理论是计算机理论的重要基础,也是形式语言和自动机理论的基础。

一些没有重复对象的全体称为集合,而这些被包含的对象称为该集合的元素。集合中的元素可以按任意的顺序进行排列。通常,使用大写英文字母表示一个集合。使用 Φ 代表空集,表示该集合没有包含任何元素。

如果集合 A 包含元素 x(也称元素 x 在集合 A 中),记为 $x \in A$。

如果集合 A 没有包含元素 x(也称元素 x 不在集合 A 中),记为 $x \notin A$。

如果一个集合包含的元素是有限的,称该集合为有穷集合。如果一个集合包含的元素是无限的,称该集合为无穷集合。

对于任意的有穷集合 A,使用 $|A|$ 表示该集合包含的元素的个数,显然,$|\Phi| = 0$。

对于具体的集合,可以使用明确的、形式化的方法进行描述。

对于元素个数较少的有穷集合,可以采用列举法,即将集合的所有元素全部列出,放在一对花括号中。例如,集合 $A = \{0,1,2,3,4,5,6,7,8,9\}$,表示集合 A 由 0,1,2,3,4,5,6,7,8,9 共 10 个元素组成。

对于集合元素较多的有穷集合或者是无穷集合,可以使用集合形成模式 $\{x \mid P(x)\}$ 进行描述(也称为命题法),其中,x 表示集合中的任一元素,$P(x)$ 是一个谓词,对 x 进行限定。$\{x \mid P(x)\}$ 表示由满足 $P(x)$ 的一切 x 构成的集合。可以使用自然语言,或者数学表示法来描述谓词 $P(x)$。例如,$\{n \mid n\text{是偶数}\}$,或者,$\{n \mid n \bmod 2 = 0\}$,都表明了由所有偶数组成的集合。

定义 1-1 子集的定义。

对于两个集合 A,B,若集合 A 的元素都是集合 B 的元素,则称集合 A 包含于集合 B 中(或称集合 B 包含集合 A),记为 $A \subseteq B$,并且称集合 A 是集合 B 的子集。

若 $A \subseteq B$,且集合 B 中至少有一个元素不属于集合 A,则称集合 A 真包含于 B 中(或称集合 B 真包含集合 A),记为 $A \subset B$,此时,称集合 A 是集合 B 的真子集。

两个集合相等,当且仅当 $A \subseteq B$ 且 $B \subseteq A$。注意:不是 $A \subset B$ 且 $B \subset A$。

定义 1-2 集合之间的运算。

集合 A 与集合 B 的并(或称为集合 A 与集合 B 的和),记为 $A \cup B$,是由集合 A 的所有元素和集合 B 的所有元素合并在一起组成的集合。

$A \cup B = \{x \mid x \in A$ 或 $x \in B\}$。

集合 A 与集合 B 的交,记为 $A \cap B$,是由集合 A 和集合 B 的所有公共元素组成的集合。

$A \cap B = \{x \mid x \in A$ 且 $x \in B\}$。

集合 A 与集合 B 的差,记为 $A-B$,是由属于集合 A 但不属于集合 B 的所有元素组成的集合。

$A - B = \{x \mid x \in A$ 且 $x \notin B\}$。

如果 $B \subseteq A$,将 $A-B$ 称为集合 B(关于集合 A)的补,集合 A 称为集合 B 的全集或论域。

定义 1-3 幂集的定义。

设 A 为一个集合,那么 A 的幂集为 A 的所有子集组成的集合,记为 2^A,

即 $2^A = \{B \mid B \subseteq A\}$。

例 1-1 幂集的例子。

集合 $A = \{1,2,3\}$,则 A 的幂集为:

$2^A = \{\Phi,\{1\},\{2\},\{3\},\{1,2\},\{1,3\},\{2,3\},\{1,2,3\}\}$。

当集合 A 为有穷集时,如果集合 A 包含的元素个数为 n,那么集合 2^A 的元素个数(集合 A 的所有子集的个数)为 2^n,这就是称 2^A 为集合 A 的幂集的原因。当集合 A 为无穷集时,仍然使用 2^A 表示集合 A 的幂集,它也是无穷集。

注意:

任何集合 A 的幂集 2^A 的元素都是集合。

空集 Φ 是任何集合的子集,也是任何集合 A 的幂集 2^A 的子集。

定义 1-4 笛卡儿乘积的定义。

集合 A 和 B 的笛卡儿乘积使用 $A \times B$ 表示(也简记为 AB),它是集合

$\{(a,b) \mid a \in A$ 且 $b \in B\}$。

$A \times B$ 的元素称为有序偶对 (a,b),总是 A 的元素在前,B 的元素在后。

$A \times B$ 与 $B \times A$ 一般不相等。

例 1-2 设 $A = \{a,b,c\}$,$B = \{0,1\}$;

则

$A \times B = \{(a,0),(a,1),(b,0),(b,1),(c,0),(c,1)\}$。

或记为:

$$A \times B = \{a0,a1,\\ b0,b1,\\ c0,c1\}。$$

而

$$B \times A = \{(0,a),(0,b),(0,c),(1,a),(1,b),(1,c)\}。$$

思考：

什么情况下，$A \times B = B \times A$?

1.2 关　　系

1.2.1 二元关系

定义 1-5 二元关系的定义。

设 A 和 B 为两个集合，则由 A 到 B 的二元关系是 $A \times B$ 的某个子集。

二元关系简称为关系。

若 $A = B$，则称为 A 上的(二元)关系。

若 R 为由 A 到 B 的关系，当(a,b) 在 R 内时，可记为 aRb。

例 1-3 关系的例子。

设 A 为正整数集合，则 A 上的关系“$<$”是集合：

$\{(a,b) \mid a,b \in A，且\ a < b\}$。

即

$\{$

$(1,2),(1,3),\cdots$

$(2,3),(2,4),\cdots$

$(3,4),(3,5),\cdots$

$\cdots$

$\}$

1.2.2 等价关系

设 R 是集合 A 上的关系，那么，

如果对 A 中的任一元素 a，都有 aRa，则称 R 为自反的；

如果对 A 中的任一元素 a，都没有 aRa，则称 R 为反自反的；

如果对 A 中的任何元素 a 和 b，从 aRb 能够推出 bRa，则称 R 为对称的；

如果对 A 中的任何元素 a 和 b，若 aRb 且 bRa，则 $a = b$，则称 R 是反对称的；

如果对 A 中的任何元素 a,b 和 c，从 aRb 和 bRc 能够推出 aRc，则称 R 为传递的。

定义 1-6 等价关系的定义。

如果关系 R 同时是自反的、对称的和传递的，则称之为等价关系。

等价关系的一个重要性质为：集合 A 上的一个等价关系 R 可以将集合 A 划分为若干个互不相交的子集，称为等价类。对 A 中的每个元素 a，使用$[a]$ 表示 a 的等价类，即$[a] = \{b \mid aRb\}$。等价关系 R 将集合 A 划分成的等价类的数目，称为该等价关系的指数。

例 1-4 等价关系的例子。

考虑非负整数集合 **N** 上的模 3 同余关系 R，$R = \{(a,b) \mid a,b \in N，且\ a \bmod 3 = b \bmod$

3}。那么，集合 $\{0,3,6,\cdots,3n,\cdots\}$ 形成一个等价类，记为[0]；集合 $\{1,4,7,\cdots,3n+1,\cdots\}$ 形成一个等价类，使用[1] 表示。集合 $\{2,5,8,\cdots,3n+2,\cdots\}$ 形成另一个等价类，记为[2]。

$\mathbf{N}=[0]\cup[1]\cup[2]$；$R$ 的指数为 3。

1.2.3 关系的合成

关系是可以合成的。

定义 1-7 关系合成的定义。

设 $R_1\subseteq A\times B$ 是集合 A 到 B 的关系，设 $R_2\subseteq B\times C$ 是集合 B 到 C 的关系，则 R_1 和 R_2 的合成是集合 A 到 C 的(二元)关系。

R_1 和 R_2 的合成记为 $R_1\circ R_2$。

$R_1\circ R_2=\{(a,c)\mid (a,b)\in R_1,且(b,c)\in R_2\}$。

例 1-5 关系合成的例子。

设 R_1 和 R_2 的是集合{1,2,3,4}上的关系；

$R_1=\{(1,1),(1,2),(2,3),(3,4)\}$,

$R_2=\{(2,4),(4,1),(4,3),(3,1),(3,4)\}$,

则

$R_1\circ R_2=\{(1,4),(2,1),(2,4),(3,1),(3,3)\}$。

$R_2\circ R_1=\{(4,1),(4,2),(4,4),(3,1),(3,2)\}$。

定义 1-8 关系的 n 次幂的定义。

设 R 是 S 上的二元关系，则关系的 n 次幂 R^n 如下递归定义：

$R^0=\{(a,a)\mid a\in S\}$

$R^i=R^{i-1}R \qquad (i=1,2,3,\cdots)$

定义 1-9 关系的闭包的定义。

设 R 是 S 上的二元关系，R 的正闭包 R^+ 定义如下：

(1) $R\in R^+$；

(2) 如果 $(a,b),(b,c)\in R^+$，则 $(a,c)\in R^+$；

(3) 除(1)，(2)外，R^+ 不再含有其他任何元素。

即

$R^+=R\cup R^2\cup R^3\cup\cdots$

且当 S 为有穷集时，有

$R^+=R\cup R^2\cup R^3\cup\cdots\cup R^{|s|}$

设 R 是 S 上的二元关系，R 的克林包 R^* 定义为

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

例 1-6 关系闭包的例子。

设 R_1 和 R_2 是集合 $\{a,b,c,d,e\}$ 上的二元关系，其中

$R_1=\{(a,b),(c,d),(b,d),(b,b),(d,e)\}$

$R_2=\{(a,a),(b,c),(d,c),(e,d),(c,a)\}$

求 R_1R_2，R_1^+，R_1^*。

(1) $R_1R_2=\{(a,c),(c,c),(b,c),(d,d)\}$

(2) $R_1^+ = \{(a, b), (c, d), (b, d), (b, b), (d, e), (a, d), (a, e), (c, e), (b, e)\}$

(3) $R_1^* = \{(a, a), (b, b), (c, c), (d; d), (e, e)\} \cup R_1^+$

1.3　证明和证明的方法

形式语言和有限自动机，有很强的理论性，许多的论断是以定理的形式给出的，而定理的正确性是需要进行证明的。

形式语言和有限自动机理论中定理的证明大多使用反证法和归纳法进行。

1.3.1　反证法

反证法也称为归谬法。利用反证法证明一个命题时，一般的步骤为：

(1) 假设该命题不成立；

(2) 进行一系列的推理；

(3) 如果在推理的过程中，出现了下列情况之一：

① 与已知条件矛盾；

② 与公理矛盾；

③ 与已证过的定理矛盾；

④ 与临时的假定矛盾；

⑤ 自相矛盾。

(4) 由于上述矛盾的出现，可以断言“假设该命题不成立”的假定是不正确的；

(5) 肯定原来的命题是正确的。

例 1-7　反证法例 1。

有牛、羊和猪三种动物共 10 头，利用反证法证明：在这三种动物中至少有一种动物不会少于 4 头。

证明：

假设该命题不成立，即在这三种动物中没有一种动物多于 4 头；那么，每种动物最多只有 3 头，则三种动物总数≤9，而三种动物共 10 头，矛盾，所以，假设不成立，在这三种动物中至少有一种动物不会少于 4 头。证毕。

例 1-8　反证法例 2。

利用反证法证明：$\sqrt{2}$是无理数。

证明：

假设$\sqrt{2}$不是无理数，那么，$\sqrt{2}$是有理数；则$\sqrt{2}$可以记为 n/m，而且 n、m 的最大公约数为 1。

$$\sqrt{2} = n/m;$$

则

$$2 = n^2/m^2$$

$$n^2 = 2m^2$$

所以，n 是偶数。令 $n = 2k$，

则

$$(2k)^2 = 2m^2$$
$$4k^2 = 2m^2$$
$$2k^2 = m^2$$

所以，m 是偶数。

n 和 m 都是偶数，而 n、m 的最大公约数为 1，矛盾。

所以，假设不成立，$\sqrt{2}$是无理数。

思考：

18 是完全平方数吗？

1.3.2 归纳法

归纳法就是从特殊到一般的推理方法。分为完全归纳法和不完全归纳法两种形式。

完全归纳法是根据一切情况的分析而作出的推理。由于必须考虑所有的情况，所以得出的结论是完全可靠的。

不完全归纳法是根据一部分情况作出的推理，因此，不能作为严格的证明方法。

在形式语言与有限自动机理论中，大量使用数学归纳法证明某个命题。

数学归纳法可以使用“有限”步骤来解决“无限”的问题。

数学归纳法的原理为：

假定对于一切非负整数 n，有一个命题 $M(n)$，假设证明了：

(1) $M(0)$ 为真；

(2) 设对于任意的 $k \geqslant 0$，$M(k)$ 为真，如果能够推出 $M(k+1)$ 为真，则对一切 $n \geqslant 0$，$M(n)$ 为真。

因此，在使用数学归纳法证明某个关于非负整数 n 的命题 $P(n)$ 时，只需要证明(1)、(2)两点即可。第(1)步称为归纳基础，第(2)步称为归纳步骤。第(2)步中“设对于任意的 $k \geqslant 0$，$M(k)$ 为真”，称为归纳假设。

在实际应用中，某些命题 $P(n)$ 并非对 $n \geqslant 0$ 都成立，而是对 $n \geqslant N$(N 为大于 0 的某个自然数)成立，此时，也一样可以使用该归纳法。具体步骤如下。

假定对于一切非负整数 n，有一个命题 $M(n)$，假设证明了：

(1) $M(N)$ 为真；

(2) 设对于任意的 $k \geqslant N$，$M(k)$ 为真，如果能够推出 $M(k+1)$ 为真，则对一切 $n \geqslant N$，$M(n)$ 为真。

1.3.3 递归的定义与归纳证明

递归定义提供了一种集合的良好的定义方式，使得集合中的元素的构造规律较明显，同时给集合性质的归纳证明提供了良好的基础。

归纳法证明递归定义集合性质的步骤如下。

(1) 基础：证明该集合中的最基本元素具有性质 P；而且使得该集合非空；

(2) 归纳：证明如果该集合的元素 $x_1, x_2, x_3, \cdots$，具有性质 P，则使用某种运算、函数或组合方法对这些元素进行处理后所得的元素也具有性质 P；

(3) 由归纳法原理，集合中的所有元素也具有性质 P。

例 1-9　Fibonacci 数组成的集合的定义。

Fibonacci 数组成的集合为：{0,1,1,2,3,5,8,13,21,34,55,…}

如何生成该集合中的所有数呢？

(1) 基础：0 和 1 是该集合的最基本的两个元素；

(2) 归纳：若 m 是第 i 个元素，n 是第 $i+1$ 个元素，则第 $i+2$ 个元素为 $n+m$，其中：$i\geqslant 1$；

(3) 只有满足(1)和(2)的数，才是集合的元素。

例 1-10　括号匹配的串构成的集合的定义。

该集合是指所有的左括号和右括号相匹配的串的集合；如()，(())，()()等都是该集合的合法的串；而)(，())等就不是合法的串。

如何生成该集合中的所有字符串呢？

(1) 基础：()是合法的该集合的最基本的元素；

(2) 归纳：

若 A 是一个合法的元素，则(A) 是一个合法的元素；

若 A 和 B 是合法的元素，则 AB 是一个合法的元素。

(3) 只有满足(1)和(2)的串，才是集合的元素。

1.4　图　与　树

现实世界中，有许多现象可以抽象成图来表示。直观地，图是由一些点和连接两点的边组成的。

定义 1-10　无向图的定义。

设 V 是一个非空的有穷集合，$E\subseteq V\times V$，称 $G=(V,E)$ 为一个无向图。其中，V 称为顶点集，V 中的元素称为顶点，E 称为无向边集，E 中的元素称为无向边。

无向图中的边都没有方向。

(v_i,v_j) 和 (v_j,v_i) 表示的是同一条边。

定义 1-11　有向图的定义。

设 V 是一个非空的有穷集合，$E\subseteq V\times V$，称 $G=(V,E)$ 为一个有向图。其中，V 称为顶点集，V 中的元素称为顶点，E 称为有向边集，E 中的元素称为有向边。

有向图中的边都有方向。

(v_i,v_j) 表示的是从顶点 v_i 出发，到达顶点 v_j 的一条边。其中，v_i 称为 v_j 的前导，v_j 称为 v_i 的后继。

(v_i,v_j) 和 (v_j,v_i) 表示的是不同的边。

定义 1-12　有向路的定义。

设 $G=(V,E)$ 为一个有向图。如果对于 $1\leqslant i\leqslant k$，均有 $(v_i,v_{i+1})\in E$，则称 $v_1,v_2,v_3,\cdots,v_k$ 是 G 的一条有向 v 路。当 $v_1=v_k$ 时，$v_1,v_2,v_3,\cdots,v_k$ 称为一条有向回路。

定义 1-13　树的定义。

设 $G=(V,E)$ 为一个有向图。当 G 满足如下条件时，称 G 为一棵(有向)树：

(1) 存在一个顶点 v 没有前导，且 v 到图中的其他顶点都有一条有向路，该顶点称为树的根；

（2）每一个非根顶点有且仅有一个前导；

（3）每个顶点的后继按其拓扑关系从左到右排序。

通常，树中的顶点称为结点，某个顶点的前导称为该结点的父亲，某个顶点的后继称为该结点的儿子。如果树中有一条从顶点 v_i 到顶点 v_j 的有向路，则称 v_i 是 v_j 的祖先，v_j 是 v_i 的后代。无儿子的结点称为叶子结点，非叶子结点称为中间结点（分支结点）。

1.5 语　　言

任意字符的集合就是一个字母表，最常用的字母表是大小写 26 个英文字母表，10 个阿拉伯数字字母表，24 个希腊字母表以及 0 和 1 的二进制字母表。

字母表具有非空性、有穷性。一般使用 Σ 表示字母表。

字母表中的字母按照某种顺序一个接一个地排列起来，形成字符的序列，称作一个字符串。

一般使用 ε 代表空串。

定义 1-14　语言的定义。

形式语言和自动机理论中的语言是一个广泛的概念，一个字母表上的语言就是该字母表的某些字符串（也称为句子）的集合。

字母表中的字母是句子的最基本元素。字母表中的字母必须具有如下两个特点。

（1）整体性（不可分性），例如，字母表 $\{aa,ab\}$ 中的 aa 和 ab 都是单个字符，不能被拆分为 a,a，和 a,b。

（2）可辨认性（可区分性），即字母表中任意两个字符都是不同的，是可以区分的，也就是说，字母表中的元素是不能重复的

对于语言的研究，实际上包括以下三个方面。

首先，如何给出一个语言的表示。如果该语言是有穷语言，那么，可以使用列举法列举出语言中所包含的所有字符串即可；而如果该语言是无穷语言，对该语言的表示，就成了问题，需要考虑语言的有穷描述。

第二个问题，对于一个给定的语言是否存在有穷描述。并不是所有的语言都存在有穷描述，即对于某些语言，并不存在有穷表示。

第三个问题是具有有穷表示的语言的结构以及结构的特性问题。

1.6 常用术语

下面给出本书常用的术语。

（1）用 ε 代表空串，$\{\varepsilon\}$ 代表仅含有空串的集合。

（2）用 Φ 代表空集，表示一个元素都不包含的集合。

（3）用 Σ 代表一个符号的非空有限集合，称之为字母表，其中的元素称为字母。

（4）用 $\alpha\beta$ 代表两个字符串 α 与 β 的连接。

即若 $\alpha = a_1a_2a_3\cdots a_n, \beta = b_1b_2b_3\cdots b_m;\ m,n \geqslant 0$，

则 $\alpha\beta = a_1a_2a_3\cdots a_nb_1b_2b_3\cdots b_m$。

显然，$\alpha\varepsilon = \varepsilon\alpha = \alpha$。

(5) 用 AB 代表两个集合 A 与 B 的连接。

即若 $A = \{a_1, a_2, a_3, \cdots, a_n\}, B = \{b_1, b_2, b_3, \cdots, b_m\}$，则

$$\begin{aligned} AB = \{ & a_1b_1, a_1b_2, a_1b_3, \cdots, a_1b_m, \\ & a_2b_1, a_2b_2, a_2b_3, \cdots, a_2b_m, \\ & a_3b_1, a_3b_2, a_3b_3, \cdots, a_3b_m, \\ & \cdots \\ & a_nb_1, a_nb_2, a_nb_3, \cdots, a_nb_m\} \end{aligned}$$

注意：

$$A\Phi = \Phi A = \Phi。$$

通常，AB 与 BA 是不相等的。

AB 与 BA 在什么情况下相等？

① 当 $A = B$；

② A 和 B 中有一个为$\{\varepsilon\}$，则 $A\{\varepsilon\} = \{\varepsilon\}A = A$；

③ A 和 B 中有一个为 Φ，则 $A\Phi = \Phi A = \Phi$。

(6) 用 A^n 代表集合 A 的 n 次连接：

$$\begin{aligned} A^0 &= \{\varepsilon\} \\ A^1 &= A \\ A^2 &= AA \\ A^3 &= AA^2 \\ &\cdots \\ A^n &= AA^{n-1} \end{aligned}$$

(7) 用 A^* 代表集合 A 上所有字符串的集合。即表示集合 A 中的所有字符串进行任意次连接而形成的串的集合，也称 A^* 为集合 A 的闭包(克林闭包)。

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

例 1-11　$A = \{0,1\}$，

则　$A^0 = \{\varepsilon\}$　　　即长度为 0 的 0 和 1 组成的串的集合

$A^1 = A = \{0,1\}$　　即长度为 1 的 0 和 1 组成的串的集合

$A^2 = AA = \{00,01,10,11\}$　即长度为 2 的 0 和 1 组成的串的集合

$A^3 = AA^2 = \{000,001,010,011,100,101,110,111\}$ 即长度为 3 的 0 和 1 组成的串的集合

…

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

$= \{$0 和 1 组成的所有的串$\}$

$= \{w \mid w$ 是 0 和 1 组成的串$\}$

例 1-12　若 $A = \{a,b,c\}$，则 $A^* = A^0 \cup A^1 \cup A^2 \cup \cdots \cup A^n$

$= \{a,b,c$ 组成的所有的串$\}$

$= \{w \mid w$ 是 a,b 和 c 组成的串$\}$

如果一个串 ω 是一个集合 A 的闭包中的串，也称 ω 是集合 A 上的串。

对于任何集合 A，有$(A^*)^* = A^*$。