有限自动机理论

YOUXIAN ZIDONGJI LILUN

陈文宇 编著

电子科技大学出版社

电子科技大学研究生系列教材建设项目

有限自动机理论

陈文宇 编著

电子科技大学出版社

图书在版编目（CIP）数据

有限自动机理论 / 陈文宇编著. —成都：电子科技大学出版社，2007.3

ISBN 978-7-81114-415-4

Ⅰ. 有… Ⅱ. 陈… Ⅲ. 有限自动机—自动机理论—研究生—教材 Ⅳ. TP301.1

中国版本图书馆 CIP 数据核字（2007）第 024732 号

内容简介

本书简述了形式语言的基本内容，包括文法的分类和语言间运算的封闭性，有限自动机（包括有限状态自动机、下推自动机和图灵机）的基础理论，从构造文法产生语言的角度和构造自动机识别语言的角度对语言进行讨论，并介绍文法与自动机之间等价的转换方法。

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，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，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的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。

有限自动机理论

陈文宇　编著

出　　版：电子科技大学出版社（成都市一环路东一段 159 号电子信息产业大厦　邮编：610051）
策划编辑：杜　倩
责任编辑：杜　倩
主　　页：www.uestcp.com.cn
电子邮箱：uestcp@uestcp.com.cn
发　　行：新华书店经销
印　　刷：成都中铁二局永经堂印务有限责任公司
成品尺寸：185mm×260mm　印张　12.875　字数　310 千字
版　　次：2007 年 3 月第一版
印　　次：2007 年 3 月第一次印刷
书　　号：ISBN 978-7-81114-415-4
定　　价：19.00 元

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了自动机模型。

形式语言和自动机的理论的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。

研究生的适应能力以及创新能力在很大程度上取决于坚实的理论基础和专业基础知识，这是高质量研究生教育的重要特征之一。在当今这个计算机科学技术突飞猛进，专业知识日新月异的时代，只有扎实掌握专业的计算机理论基础，才有可能从事这个专业的科研、教学和其他专业技术工作，才能打好进行创造性研究的基础，因此理论课程的学习就显得尤为重要。研究生理论课程教学，必须立足于提高研究生的学术水平和科研能力，是实现研究生的培养目标、保证研究生质量的重要环节。为此，电子科技大学于 2002 年开始，在研究生阶段开设《有限自动机理论》课程，为配套教学，特编写本书。

本书不注重繁琐的定理证明过程，而是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 6 章，第一章回顾本书所需要的基本数学知识；第二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；第三、四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；第五章介绍下推自动机；第六章是对图灵机的讨论。

本书的目标是力求使各类计算机专业的研究生掌握各类有限自动机的模型、构造方法和技巧，培养学生的计算思维能力。

本书涵盖了形式语言和自动机的基本内容，适合作计算机专业研究生的教材。本书是作者在电子科技大学科学与工程学院的《形式语言与自动机》讲义的基础上，阅读了关于形式语言和自动机理论方面的文献资料，并结合 12 年的实际教学的经验编著而成的。

在此，谨对参阅文献的作者和翻译人员表示衷心感谢。

特别感谢北京工业大学的蒋宗礼教授，为本书提出了大量宝贵的建议和意见；本书的习题也都来源于蒋教授的《形式语言与自动机理论》一书。

感谢龚天富教授在百忙之中抽出时间认真审阅了全稿。还要感谢电子科技大学出版社的杜倩编辑，为本书的出版做了大量工作，才使得本书得以同广大读者见面。屈鸿、陈昆、欧齐、程炼同志查找了大量资料，在此，一并表示感谢。

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。

本书受电子科技大学研究生教材建设基金资助出版。

作者

2006.6

目 录

第一章 基础知识

计算机科学与技术学科强调 4 个方面的专业能力：计算思维能力，算法设计与分析能力，程序设计与实现能力，计算机系统的认知、分析、设计和运用能力。这也是计算机科学与其他学科的重要区别。相关的理论是计算机学科的基础。理论方面的知识是计算机的真正灵魂。理论是从计算机应用当中抽象出来的，目的在于使用抽象出的理论去更好地指导实践。

在本科阶段的学习过程中，学生以观察、描述、比较、分类、推断、应用、创造思维等科学思维过程为主，强调自学的能力再培养。研究生阶段，需要对学生进一步进行抽象思维、逻辑思维、创造思维能力的培养，本科生与研究生的根本区别在于研究生的需要宽厚、坚实的理论基础。

建立物理符号系统并对其实施等价变换是计算机学科进行问题描述和求解的重要手段。“可行性”所要求的“形式化”及其“离散特征”使得数学成为重要的工具，而计算模型无论从方法还是工具等方面，都表现出它在计算机上科学中的重要作用。

计算机科学与技术学科要求从业者具有形式化描述和抽象思维能力，掌握逻辑思维方法。这种能力就是计算思维能力或计算机思维能力[1]。

计算机学科系统地研究信息描述和变换算法，重要包括信息描述和变换算法的理论、分析、效率、实现和运用。学科的根本问题在于：什么能被（有效地）自动化？学科的重要内容之一是研究计算领域中的一些普遍规律，描述算法的基本概念与模型。

计算思维能力的培养主要是通过基础理论系列课程实现的，该系列是由数学和抽象度较高的理论课程组成，包括数学分析、集合和图论、近世代数、数学建模、计算理论等课程。它们构成了对学生计算思维能力培养的一个阶梯训练系统，如图 1-1 所示[1]。

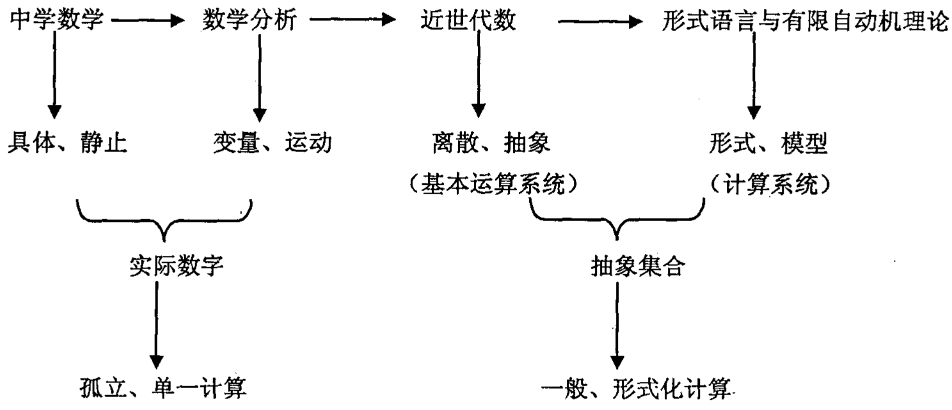

图 1-1 计算思维阶梯训练系统

在此系统中，连续数学、离散数学、计算模型三部分内容按阶段分开。三个阶段对应计算机学科的学生在本科和研究生学习期间的思维方式和能力变化的三个步骤，使得计算思维能力不断提高。

在中学阶段所学的数学，研究的是具体的、静止对象的计算。到了数学分析阶段，通过连续变量和函数，将运动带入了问题考虑的范围中，到离散数学和近世代数阶段，开始考虑基本运算系统，该系统更加抽象，更具一般性，所考虑的计算对象具有更明显的抽象和离散的特征。到形式语言与自动机理论阶段，所考虑的运算对象是更高级别上抽象出来的形式化特征，而运算也呈现出模型化的特征。计算从孤立的、单一的计算演变成为一般、形式化的计算，而一般、形式化的计算，正是计算机上科学所要研究的计算。

考虑的计算对象不同，所需要的思维方式和能力就有所不同，正是通过系列课程的学习，在不断升华的过程中，才能逐步培养出抽象思维能力和逻辑思维方法，同时，创新意识的建立和创新能力的培养也在学习过程中不断加强。

计算理论是研究使用计算机解决计算问题的数学理论。有三个核心领域：自动机理论、可计算性理论和计算的复杂性理论。研究生阶段，有三门课程与之分别对应：《有限自动机理论》、《可计算理论》和《算法分析》。

可计算性理论的中心问题是建立计算的数学模型，进而研究哪些是可计算的，哪些是不可计算的。计算的复杂性理论研究算法的时间复杂性和空间复杂性。在可计算性理论中，将问题分成可计算的和不可计算的；在复杂性理论中，目标是把可计算的问题分成简单的和困难的。

自动机理论论述计算的数学模型的定义和性质，这些模型在计算机科学的若干应用领域中起着重要作用，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。

自动机理论是学习计算理论的良好起点，不仅能提高感知能力，同时也能提高思维的敏捷性，考虑问题仔细、严谨、周密、有理有据；可以由具体形象思维逐渐向抽象思维过渡，从而促进逻辑思维和创造力的发展；可以使逻辑思维过程清晰化、条理化、整体化，有利于培养计算表达能力，提高推理、判断、分析问题和解决问题的能力。

计算机专业的研究生学习计算理论，必须知其然更要知其所以然。学习的目的应该是将抽象的理论应用于实践，不但要掌握题目的解题方法，更要掌握解题思想。对于定理的学习不是简单的应用，而是掌握证明过程即掌握定理的由来，训练自己的推理能力。只有这样才达到了学习这门科学的目的，同时也缩小了与数学专业的同学之间思维上的差距。

计算理论并不神秘，也不令人厌烦，而是容易理解，甚至是有趣的。计算理论中包括有许多迷人而重要的思想，要体会、感悟思想的闪光，并且体会大师们当初发现这些思想而获得的极大的乐趣；同时，它也有许多细小的、有时是枯燥乏味的细节。理论课程的学习是一件艰苦的工作，应该抓住思想，而不要陷入细节的单调乏味之中。

计算机学科的方法论有三个过程：抽象、理论和设计及实现。最根本的问题在于：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和严密的表达、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世界过程。

本书内容属于理论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作扼要的介绍，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。

一些没有重复的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。集合中元素可以按任意的顺序进行排列。一般使用大写英文字母表示一个集合，使用Φ代表空集，即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 $x! \in A$。

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（也称为集合 A 的基数），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由无穷多个元素组成的集合，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|n 是偶数}，或者，$\{n|n \bmod 2 = 0\}$，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 真包含于集合 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。

$A \cup B=\{x|x \in A$ 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。

$A \cap B=\{x|x \in A$ 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。

$A-B=\{x|x\in A$ 且 $x!\in B\}$。

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 $2^A=\{B|B\subseteq A\}$。

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×B 表示（也简记为 AB），它是集合

$$\{(a,b)\mid a\in A \text{ 且 } b\in B\}$$

A×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×B 与 B×A 一般不相等。

例：设 A={a，b，c},B={0，1}；

则

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

而

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

思　考

什么情况下，A×B=B×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×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 1-2 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

$$
\begin{aligned}
&\{\\
&\quad (1, 2), (1, 3), \ldots\\
&\quad (2, 3), (2, 4), \ldots\\
&\quad (3, 4), (3, 5), \ldots\\
&\quad \ldots\\
&\}
\end{aligned}
$$

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

$N=[0] \cup [1] \cup [2]$；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 1-4 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：

$R^0=\{(a，a)|a\in S\}$

$R^i=R^{i-1}\circ R$　　　(i =1，2，3，...)

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^+ 定义为

(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

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

且当 S 为有穷集时，有

$$R^+=R\cup R^2\cup R^3\cup\ldots\cup R^{|S|}$$

设 R 是 S 上的二元关系，R 的克林包 R^* 定义为

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

例子 1-14 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) $R_1\circ R_2=\{(a，c),(c，c),(b，c),(d，d)\}$

(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。在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，得出的结论出现了下列情况之一：

(1) 与已知条件矛盾；

(2) 与公理矛盾；

(3) 与已证明过的定理矛盾；

(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 1-5　反证法例子。

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 $\frac{n}{m}$，而且 n、m 的最大公约数为 1。

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

则

$$2=\frac{n^2}{m^2}$$

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

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。

完全归纳法是根据所有情况的分析而作出的推理。由于必须考虑了所有的情况，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>=0，M(k)为真；

如果能够推出 $M(k+1)$为真，则对一切 n>=0，$M(n)$为真。

因此，在使用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>=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>=0 都成立，而是对 n>=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) $M(N)$为真；

(2) 设对于任意的 k>=N，$M(k)$为真；

如果能够推出 $M(k+1)$为真，则对一切 n>=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。

递归定义集合步骤：

(1) 基础：首先定义该语言中的最基本的元素。

(2) 递归：如果该集合的元素 x_1，x_2，x_3，…，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新元素也在该集合中。

(3) 有限性：只有满足(1)和(2)的元素才包含在语言中。

用归纳方法证明递归定义集合的性质的步骤：

(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，…，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) 根据归纳法原理，集合中的所有元素也具有性质 P。

例 1-6　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>=1。

(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。

直观地看，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（前导）出发，到达顶点 v_j（后继）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，…，v_k 是 G 的一条有向路。当 $v_1=v_k$ 时，v_1，v_2，v_3，…，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

任意字符的非空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，所形成的字符的序列称为一个字符串；一般使用 ε 代表空串。

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。

对于语言的研究，实际上包括三个方面：

首先是如何给出一个语言的表示。如果该语言是有穷语言，那么，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。

第二个问题是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，{ε}代表仅含有空串的集合。

(2) 用Φ代表空集，表示一个元素都没有的集合。

(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αβ代表两个字符串α与β的连接。

即若$\alpha=a_1a_2a_3\ldots a_n$，$\beta=b_1b_2b_3\ldots b_m$；$m,n>=0$,

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

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

用α^n代表字符串α的次连接，$\alpha^0=\varepsilon$，$\alpha^n=\alpha\alpha^{n-1}$，$n>0$。

(5) 用 AB 代表两个集合 A 与 B 的连接。

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

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

注意：

$A\Phi=\Phi A=\Phi$。

一般，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{ε}，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：

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

$A^1=A$

$A^2=AA$

$A^3=AA^2$

…

$A^n=AA^{n-1}$

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）。

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