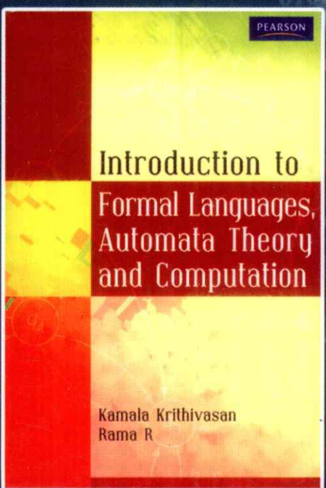
国外计算机科学教材系列

PEARSON

形式语言，自动机理论与计算导论

Introduction to Formal Languages, Automata Theory and Computation



[印] Kamala Krithivasan
Rama R 著

孟宇龙 李健利 王宇华 译

冯晓宁 审校

電子工業出版社
PHEI PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

http://www.phei.com.cn

国外计算机科学教材系列

形式语言，自动机理论与计算导论

Introduction to Formal Languages, Automata Theory and Computation

[印] Kamala Krithivasan
Rama R 著

孟宇龙 李健利 王宇华 译
冯晓宁 审校

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

内容简介

本书主要介绍形式语言、自动机理论及可计算性，主要内容包括：本学科的基础知识、文法、有限状态自动机理论、有限状态自动机变形、下推自动机、图灵机和可计算理论、复杂性原理、时空复杂度、NP完全问题、DNA计算、膜计算以及本学科一些前沿理论等。全书包含大量的例子和图表，便于读者对定义、定理的理解。最后，由浅入深地给出了两套选择题并附有答案，方便教学。

本书适合作为计算机专业高年级本科生或研究生计算理论课程的教材和参考书，也可作为计算机研究人员的参考书。

图书在版编目（CIP）数据

形式语言，自动机理论与计算导论/（印）卡马拉（Kamala，K.），（印）拉玛（Rama，R.）著；孟宇龙，李健利，王宇华译．—北京：电子工业出版社，2012.2

书名原文：Introduction to Formal Languages, Automata Theory and Computation

国外计算机科学教材系列

ISBN 978-7-121-15394-5

Ⅰ．①形…　Ⅱ．①卡…　②拉…　③孟…　④李…　⑤王…　Ⅲ．①形式语言-高等学校-教材②自动机理论-高等学校-教材　Ⅳ．①TP301

中国版本图书馆 CIP 数据核字（2011）第252620号

策划编辑：冯小贝
责任编辑：许菊芳
印　　刷：涿州市京南印刷厂
装　　订：涿州市桃园装订有限公司
出版发行：电子工业出版社
　　　　　北京市海淀区万寿路173信箱　邮编　100036
开　　本：787×1092　1/16　印张：20.5　字数：525千字
印　　次：2012年2月第1次印刷
定　　价：59.00元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

译　者　序

任何一门科学都有其自身的理论基础,计算机科学也不例外。理论计算机科学是推动计算机技术向前发展的强大动力,形式语言、自动机是将数学系统应用于计算的模型,其应用范围已被扩展到生物工程、自动控制、图像处理等许多领域。二三十年以前,计算机科学基本上还是数学的一个分支, 而现在,计算机科学拥有广泛的研究领域和众多的研究人员,在很多方面反过来推动数学的发展。但不管怎样,理论计算这个"孩子"的身上始终流淌着数学"母亲"的血液,它就是计算机科学的数学基础,即理论计算机科学。

本书是由印度理工学院的马德拉斯分校 Kamala Krithivasan 和 Rama R 教授合作编写的。Kamala 教授是 1986 年福尔布莱特学术奖金的获得者,同时还是印度国家工程学会的会员,Rama 教授是印度工业教育学会的终身会员。该书用简洁清晰的方式阐述了相关的理论概念,并深入涵盖了形式文法及基本的自动机类型。除此之外,还介绍了一些本学科前沿领域的研究成果,如 DNA 算法和膜计算。可以说本书是目前市场上自动机理论方面比较优秀的书籍之一。

很幸运能够翻译此书。从接到翻译任务至今,译者颇感该书翻译的艰辛。其部分内容涉及领域的新和广,大大增加了翻译的难度。虽然译者在教学和科研工作中也在讲授和使用相关知识,但仍有一些新思想、新问题从未遇到过,加之本书涉及面广、词汇量大, 令译者颇感费力。通过该书的翻译,也丰富了译者的知识结构。因为教学任务和科研任务的繁重,精心翻译、文字的反复推敲已经成为一种奢望。一拖再拖,终于在 2011 年年底完成了此稿。在此,我要深深地感谢我的妻子李娜同志对我的支持和理解。

本书由孟宇龙、李健利、王宇华翻译。全书由孟宇龙、冯晓宁审校。初译过程中,韩飞、杨歌、张冰玉、李林桧、王路、徐醒涛、刘博、赵风姬、刘凯、姚远、于群、王莹杰、王金娜等人对照英文原稿进行了认真的校对,将译文中讹误和欠妥之处进行了订正。由于时间、能力有限,译稿难免存在错误和疏漏。敬请读者批评指正并不吝赐教。

译者

2011 年 12 月于哈尔滨工程大学计算机学院

前　言

《形式语言，自动机理论与计算导论》这本书从解决相关问题的角度向学生们阐述形式语言、自动机和计算理论的基本概念。

形式语言与自动机理论出现于20世纪50年代，并由诺姆·乔姆斯基(麻省理工学院语言学和哲学教授)给出了形式语言文法的数学定义。其中一个定义与巴科斯诺尔范式一致，并且被应用于ALGOL 60编译器中。同时，人们也提出了有限状态自动机的概念。在研究编译器应用的时候，人们也定义和研究了下推自动机。

计算理论形成于1936年图灵的一篇关于可计算性的论文。他用一个模型计算器定义了图灵机，此定义被认为是计算领域的基石。经过时间的验证，他的可计算性概念被证明可用于解决很多问题。而他在不可判定性上的研究被认为是20世纪前叶数学界中最重要的突破，因为它解决了可计算理论中的诸多难题。

继图灵机被提出后的几十年间，产生了很多的重要理论，其中包括很多新的自动机模型。一个突破性进展是关于算法效率这个难于解决的问题。这些概念在许多计算机科学的分支学科，包括编译原理、计算机网络、数据库以及图形处理等方面都有着广泛的应用。最近，随着DNA计算模型的提出，越来越多的理论计算模型开始展开研究。

几乎所有大学的计算机科学专业都把自动机和语言理论做为本科教育的核心课程。这是由MCA、B. Tech. (IT)和M. Sc. (CS)课程规定的学科，并且也适合正在读本科和研究生的数学专业的学生。本学科更深层次的研究理论可在研究生选修课中进行选择性学习。

目前，市场上有很多关于本学科理论的书籍，但关于基础知识和高级理论的书目却又十分有限。本书从基础知识入手，通过14章来搭建整个学科的知识体系，并在最后两章涉及一些对做研究的学生来说可能更感兴趣的概念和理论。

希望我们这些年积累的经验能够帮助同学们解决在学习这一学科的过程中经常出现的困难。写这本书时，我们认为同学们至少应该具备了学习该学科的基础知识，因此书中提供了更多的实例细节，以便于尽可能清楚地阐述概念。对于从事这个领域研究工作的人们，可能很难决定他们的基础工作应该从哪里开始。第6章、第13章和第14章中介绍了一些简化形式的前沿理论概念，希望读者可以顺利地开始这一领域的研究。

第1章主要涵盖这一学科的基础知识，我们试图让学生领悟一些该学科的宗旨理论。第2章介绍文法的概念，并且解释了上下文无关文法的思想。第3章至第5章阐述有限状态自动机理论。第6章深入探索关于有限状态自动机的其他变型。第7章介绍下推自动机。第8章讨论上下文无关文法的附加理论。第9章至第12章讲述图灵机和可计算理论。其中的前两章讨论图灵机的基本模型、原理和变型，第11章描述可判定性的思想，第12章则主要介绍复杂性原理、时空复杂度的概念和NP完全问题。

第13章和第14章更适合研究生和研究人员阅读。第13章介绍了一些前沿领域的概述，例如正则重写系统，上下文有关语法和语法系统。第14章也是本书的最后一章，介绍了目前研究的DNA算法和膜计算。

本书几乎在每一章中都包含了丰富而广泛的知识。我们始终相信，实践是学好自动机理论课程的最佳途径。因此，在每一章的结尾(除了第 13 章与第 14 章)都提供了附有答案的习题。本书最后有两套按题目难度排列的选择题，第一套的题目比较简单，第二套相对较难，其中有些题目曾经出现在 GATE 考试中。

每一章的重点、难易程度和用途如下表所示，其中的星号表示适合于这一层次的学生学习。

章号	UG 导论	核心 UG/PG	高级 UG/PG/研究
1	*		
2	2.1, 2.2	*	
3	*	*	
4		*	
5		*	
6		*	*
7	*	*	
8		8.1, 8.2	8.3 ~ 8.9
9	*	*	
10		*	
11		*	*
12		*	*
13			*
14			*

我们希望同学们能够充分利用这本书完成本学科的学习，也相信本书能够帮助讲授这门课程的教师更好地讲授本学科知识。我们致力于让这本书尽可能没有错误，同时也希望读者给予意见和回馈。

致谢

首先要对印度理工学院马德拉斯分校的领导们致以最诚挚的感谢，感谢他们对这本在学院 50 周年纪念日出版的书所给予的大力支持。要感谢那些勤奋研究的学生们，尤其是为本书做出贡献的 P. Harsha、L. Jeganathan、S. N. Krisha、M. madhu、S. V. Ramasubramanian、Rita Brata、M. Shakthibalan 和 Y. Sivasubramanian。感激我们各自院系的领导和给予我们鼓励与支持的同事们。这本书的底稿得益于 E. Boopal 的帮助。感谢出版单位，印度 Pearson Education 出版社，它将这本书以现在这样的形式呈现给了我们。特别要感谢 Thomas Mathew Rajesh、Sojan Jose、M. E. Sethurajan 和 M. R. Ramesh 在编辑方面所给予的帮助。最后，更要感谢我们的家人给予的爱，没有他们的无私理解、支持、鼓励和耐心，这本书是不可能成功完成的。

Kamala Krithivasan

Rama R

目　　录

第1章 基础知识

这一章将复习一些有助于理解本书内容的基本概念。

1.1 集合，关系和函数

集合

集合是由一些具有明确定义的对象所组成的总体。通常，集合中的元素具有共同的性质。例如，所有选修了“可计算性”课程的学生可以构成一个集合。集合的定义如下所述。

定义 1.1 集合是由对象组成的无序总体。

集合的定义从自然语义理解比较直观。1895 年，德国数学家康托尔(Cantor)提出了集合的概念。集合中的对象被称为元素或成员。

例 1.1 由小于 20 的正偶数组成的集合 E 可以表示为：

$$E = \{2, 4, 6, 8, 10, 12, 14, 16, 18\}$$

或

$$E = \{x \mid x \text{ 是偶数且 } 0 < x < 20\}$$

包含有限个元素的集合称为有限集，反之则称为无限集。不包含任何元素的集合称为空集，记为 ϕ。集合 A 中含有的元素个数称为该集合的基数，记为 $\#A$ 或 $|A|$。例如，如果 $A = \{2, 4, 6, 8\}$，那么 $\#A = 4$。注意，$\#\phi = 0$。无限集合不能列举出其所有元素。集合也可以通过其所有元素都满足的特有性质进行描述。

例如，$A = \{x \mid x$ 是一个可以被 3 整除的正整数$\}$。这种描述方式的一般形式为 $A = \{x \mid P(x)$ 是所有 x 都满足的性质$\}$。

集合可以通过构造法或列举法进行表示，也可以通过归纳法来定义。集合的归纳定义包含三个步骤。分别是：

1. 定义出集合中的一些基本元素。
2. 利用集合中已经存在的元素归纳生成新的元素。
3. 重复第 1 步和第 2 步，直到所有新生成的元素都存在于集合中。

例 1.2 令 W 为一个完备定义的集合。可以进行如下的归纳定义。

基本子句：$[\ ] \in W$。

归纳子句：如果 $x, y \in W$，那么 $xy \in W$ 且 $[x] \in W$。

外部子句：如果 W 没有遵循有限的基本子句和归纳子句，那么它是空集。

如果集合 A 中的任意元素都属于集合 B，那么集合 A 称为集合 B 的子集，记为 $A \subseteq B$。也可以称为集合 A 包含于集合 B 中。如果集合 A 是集合 B 的子集并且集合 A 不等于集合 B 中，那么集合 A 是集合 B 的真子集，记为 $A \subset B$。空集是任何集合的子集。如果集合 A 中所有元

素都属于集合 B，同时集合 B 中所有元素都属于集合 A，则称集合 A 和集合 B 相等。两个集合相等记为 $A=B$。

两个集合可以通过各种集合操作形成一个新的集合，包括并、交和差。对于两个集合的并集，该集合中的元素要么是集合 A 中的元素，要么是集合 B 中的元素，或者同时包含集合 A 和集合 B 中的元素。并集用符号∪表示。对两个集合 A 和 B，有 $A\cup B=\{x\mid x\in A$ 或 $x\in B\}$。

两个集合的交集是指两个集合所有相同元素的集合，记作∩。对于两个集合 A 和 B，有 $A\cap B=\{x\mid x\in A$ 且 $x\in B\}$。

集合 A 与集合 B 的差集记为 $A-B$，对于集合 A 和 B，$A-B=\{x\mid x\in A$ 且 $x\notin B\}$。

用符号 U 表示全集，且 $A\subseteq U$。集合 A 对于集合 U 的补集定义为 $\overline{A}=U-A$。

设 S 是一个集合，S 的所有子集组成的集合称为集合 S 的幂集，记为$\mathbb{P}(S)$ 。如果 $S=\{a, b, c\}$，那么，

$$\mathbb{P}(S)=\{\phi, \{a\}, \{b\}, \{c\},\{a, b\},\{b, c\}, \{a, c\}, \{a, b, c\}\}$$

序列和元组

序列是排成一列的对象。例如，序列 7，14，21 可以写成(7，14，21)。集合中的元素可以无序，但对于序列，元素之间的顺序非常重要。并且，集合中的元素不允许重复，序列中的元素可以重复。例如，(7，7，14，21，21)是一个序列，而{7，14，21}是一个集合。序列和集合一样，可以是有限的，也可以是无限的。有限序列被称为元组。由 k 个元素组成的元组称为 k 元组。例如，(2，4)是一个二元组，(7，14，21)是一个三元组。

如果 A 和 B 是两个集合，那么 A 和 B 的笛卡儿积(或向量积)记为 $A\times B$，笛卡儿积是有序数对的集合，在每一个有序数对中，第一个元素属于集合 A，第二个元素属于集合 B。例如，如果 $A=\{0, 1\}$，$B=\{x, y, z\}$，则

$$A\times B=\{(0, x), (0, y), (0, z), (1, x), (1, y), (1, z)\}$$

同理，可求出 k 个集合 $A_1, A_2, \cdots, A_k$ 的笛卡儿积。

关系和函数

设 A 和 B 是两个集合，则 A 和 B 的二元关系是 $A\times B$ 的一个子集。例如，如果集合 $A=\{1, 3, 9\}$，集合 $B=\{x, y\}$，那么$\{(1, x), (3, y), (9, x)\}$是集合 A 和集合 B 的一种二元关系。k 个集合 $A_1, A_2, \cdots, A_k$ 之间的二元关系也可以这样确定。

关于集合的另一个基本概念是函数。函数表明了输入和输出之间关系。即函数会接收一个输入值，并产生相应的输出。对于函数 f，如果输入为 x，输出为 y，则记为 $f(x)=y$。也称 f 是从 x 到 y 的映射。例如，求和是一个求出所有输入数值总和的函数。函数输入值的集合称作这个函数的定义域，输出值的集合称作这个函数的值域。对于函数 f，令 D 表示定义域，R 表示值域。可表示为 $f:D\rightarrow R$。例如，函数 f 的定义域和值域都是集合 Z，那么该函数记为 $f:Z\rightarrow Z$。对于一个函数，如果值域里的任意一个元素都能在定义域内找到它所对应的值，那么说这个函数是满射函数。对于函数 $f:D\rightarrow R$，如果定义域 D 中任意两个不相等的元素 a 和 b 都满足 $f(a)$ 不等于 $f(b)$，那么称这个函数是单射函数。如果一个函数既是单射的，又是满射的，那么称这个函数是双射函数，或一一映射函数。

对于一个二元关系 $K \subseteq A \times B$，$K^{-1} \subseteq B \times A$ 是它的逆关系，定义为：$(b, a) \in K^{-1}$ 当且仅当 $(a, b) \in K$。例如，

$K = \{(x, y) \mid x \in S, y \in T, x$ 是 y 的学生$\}$

$K^{-1} = \{(y, x) \mid x \in S, y \in T, y$ 是 x 的老师$\}$

注意：函数的逆关系不一定是函数。对于函数 $f:A \to B$，如果存在 $b \in B$ 使得 $f(a) = b$，但是不存在 $a \in A$，则该函数没有逆关系。但是每个双射函数 f 都有一个逆关系 $f^{-1}(x)$，并且对任意 $a \in A$，都有 $f^{-1}(f(a)) = a$。

如果一个二元关系满足以下条件，那么称 R 为等价关系：

- R 是自反的，即对于任意 x，都满足 $(x, x) \in R$。
- R 是对称的，即对于任意 x 和 y，如果 $(x, y) \in R$，那么 $(y, x) \in R$。
- R 是传递的，即对于任意 x，y 和 z，如果 $(x, y) \in R$ 且 $(y, z) \in R$，那么 $(x, z) \in R$。

从集合 A 的等价关系角度分析，可以将集合 A 划分为若干个等价类。等价类的个数称为这个等价关系的秩。秩可以是有限的，也可以是无限的。

例 1.3 设 N 代表非负整数集。关系 $\equiv$ 定义如下：$a \equiv b$ 当且仅当 a 和 b 分别被 3 整除时余数相同。显然，这是一个秩为 3 的等价关系。

一个等价关系产生一个基本集合的划分。在上面的例子中，非负整数集被划分为 3 个等价类：

$$E_{11}=\{0, 3, 6, \cdots\}$$
$$E_{12}=\{1, 4, 7, \cdots\}$$
$$E_{13}=\{2, 5, 8, \cdots\}$$

对于两个等价关系 E_1 和 E_2，如果 E_2 的每个等价类都包含于 E_1 的等价类中，那么 E_2 是 E_1 的细化。例如，设 E_1 为例 1.3 定义的模 3 的等价关系，E_2 是建立在非负整数集上的一个等价关系，aE_2b 表示任意元素 a 和 b 满足：a 和 b 分别被 6 整除得到相同的余数。那么，在这种情况下会有 6 个等价类：

$$E_{21}=\{0, 6, 12, \cdots\}$$
$$E_{22}=\{1, 7, 13, \cdots\}$$
$$E_{23}=\{2, 8, 14, \cdots\}$$
$$E_{24}=\{3, 9, 15, \cdots\}$$
$$E_{25}=\{4, 10, 16, \cdots\}$$
$$E_{26}=\{5, 11, 17, \cdots\}$$

显然，每个 E_{2j} 都完全包含于一个 E_{1k}，其中 $1 \leqslant j \leqslant 6$，$1 \leqslant k \leqslant 3$。因此，$E_2$ 是 E_1 的一个细化。

有序对

自然数的有序对可以用一个单独的自然数表示。也就是说，不仅要考虑如何把有序对译成自然数，而且还要考虑把自然数译成有序对，即要深入探究从 N^2 到 N 的一一映射。一个最简单的从 N^2 到 N 的双映射如下所述。

康托尔数法：令 $\pi^2 : N^2 \to N$ 表示如下：

$$\pi^2(x,y)=\frac{(x+y)(x+y+1)}{2}+x$$

$\pi^2(x, y)$称为有序数对(x, y)的康托尔数。例如，(2，4)的康托尔数是23。下面这个表格列举了一些有序数对的康托尔数。

$x \mid y$	0	1	2	3	4	5
0	0	1	3	6	10	15
1	2	4	7	11	16	22
2	5	8	12	17	23	30
3	9	13	18	-	-	-
4	14	19	-	-	-	-
5	20	26	-	-	-	-

这种双映射在计算机科学的应用中非常重要。此方法可被用于列举三元组的有序数列，例如，$\pi^3(x_1, x_2, x_3)$可以看做$\pi^2(\pi^2(x_1, x_2), x_3)$，也可用于更高阶的元组。

闭包

闭包是集合之间的一个很重要的关系，而且这是一个处理集合和关系的常用方法。

设R是集合A的一个二元关系。R的自反(对称，传递)闭包R'遵循如下性质：

1. R'是自反的(对称的，可传递的)。
2. $R' \supseteq R$。
3. 如果R''是一个包含R的自反(对称，传递)关系，那么$R'' \supseteq R'$。

R的自反、传递闭包记为R^*。一个二元关系R的自反、传递闭包是几个可能闭包中的一个。R的传递闭包R^+不一定自反，除非R本身自反。

有限集与无限集

由有限个元素构成的集合称为有限集。集合的大小是集合的基本属性。如果集合不是有限的，则称它是无限集。两个具有相同元素数目的集合A和B，如果存在一个一一映射的函数$f:A \rightarrow B$，则称两个集合等势。通常来讲，如果存在一个集合，它和集合$\{1, 2, 3, \cdots, n\}$等势(n是自然数)，那么这个集合是有限的。如果集合A与自然数集合N等势，那么集合A称为可数无限集。如果集合A中的元素是不可数的，那么该集合称为不可数集。

1.2 证明方法

这一节介绍三种基本的证明方法：(i)数学归纳法；(ii)鸽巢原理；(iii)对角化。

数学归纳法

设A是自然数的集合，那么：

i. $0 \in A$。

ii. 对每个自然数n，如果$\{0, 1, 2, \cdots, n\} \in A$，且$n+1 \in A$，则$A=N$。归纳法尤其适合证明“对于所有$n \in N$，均满足属性$P$”的这种命题，证明方法如下。

1. 首先，确认$P(0)$为真。也就是说，对于0，属性P为真。

2. 假设对于任意 n，P 为真。

3. 然后证明 P 对于 $n+1$ 为真。

例 1.4　所有符合 $n\geqslant 0$ 的数，均满足属性 P：$1+2+3+\cdots+n=[n(n+1)]/2$。用数学归纳法证明：$P$ 对于所有 $n\geqslant 0$ 均成立。

i. 当 $n=0$ 时，等式左右两边相等，都为 0，P 为真。

ii. 假设对于某个 $n\geqslant 0$，P 为真，均满足 $1+2+3+\cdots+n=[n(n+1)]/2$。

iii. $P(n+1)$ 为

$$\begin{aligned}1+2+3+\cdots n+(n+1)&=(1+2+3+\cdots+n)+(n+1)\\&=\frac{n(n+1)}{2}+n+1\\&=\frac{(n+1)(n+2)}{2}\end{aligned}$$

有时需要证明当 $n\geqslant t$ 时，$P(x)$ 为真，那么应该首先证明 $P(t)$ 为真。

强数学归纳法

另一种针对自然数进行归纳证明的方法是强数学归纳法。假设想要证明对于所有的 $n\geqslant t$，$P(n)$ 为真。那么，在归纳步骤中应该假定存在一个 j，且对于所有 $t\leqslant j<k$，均满足 $P(j)$ 为真。然后，可以通过这种方法证明 $P(k)$ 为真。在运用普通归纳法（也称弱归纳法）的证明过程中，可以通过假设 $P(k-1)$ 为真来证明 $P(k)$ 为真。有一些例子，若用强归纳法是极为简单的。甚至在某些情况下，只有运用强化归纳法才能进行证明。

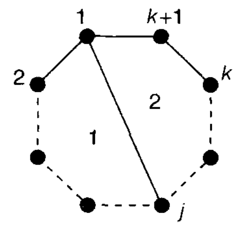
下面给出一些例子。

例 1.5　$P(n)$：n 边形的内角和为 $(2n-4)\pi/2$。

基本步骤：$P(3)$：三角形内角和为 180°，即 $(2\times 3-4)\pi/2$。

归纳步骤：令所有 n 边形均成立，$3\leqslant n\leqslant k$。

证明 $P(k+1)$：计算 $(k+1)$ 边形的内角和，如右图所示。



要计算内角和，连接顶点 1 和 j（$j\neq 2$，$k+1$）。此时，多边形的内角和就等于两个多边形 1 和 2 的内角和之和。多边形 1 有 j 条边，多边形 2 有 $k+3-j$ 条边。多边形 1 的内角和为 $(2j-4)\pi/2$，多边形 2 的内角和为 $(2(k+3-j)-4)\pi/2$。

因此，$(k+1)$ 边形的内角和是 $(2(k+1)-4)\pi/2$。

鸽巢原理

如果 A 和 B 为两个非空有限集合，且 $\#A>\#B$，那么不存在 A 到 B 的一一对应函数关系。例如，假如将集合 A 中的元素和集合 B 中的元素配对，那么集合 B 中至少有一个元素超过一次与集合 A 中的元素进行配对。

例 1.6　任选 367 人，一定存在至少两人有相同的生日，因为一年中最多只有 366 个生日。

对角化原理

令 R 是集合 A 中的一个二元关系，设 $D=\{a|a\in A$，且$(a,a)\notin R\}$。对于任意 $a\in R$，令 $R_a=\{b|b\in A$，且$(a,b)\in R\}$。那么，对于所有 $a\in A$，集合 D 与 R_a 相异。

例 1.7 令 $A=\{a,b,c,d\}$，$R=\{(a,b),(a,d),(b,b),(b,c),(c,c),(b,d)\}$。$R$ 可以表示为如下的矩阵：

	a	b	c	d
a		X		X
b		X	X	
c			X	
d		X		

$R_a=\{b,d\}$，$R_b=\{b,c\}$，$R_c=\{c\}$，$R_d=\{b\}$，$D=\{a,d\}$

显然，$D\neq R_a$，R_b，R_c，R_d。

注意 对角化原理同样适用于无限集合。

1.3 图

通过线连接起来的点的集合，称为无向图或简单图。这些点被称为节点或者顶点，连接点的线称为边。

例 1.8 与一个顶点相连的边的数目称为顶点的度。在图 1.1 中，顶点 1 的度为 3。两个顶点间最多允许存在一条边相连。方便起见，我们将顶点进行标号，并将其称为标号图。

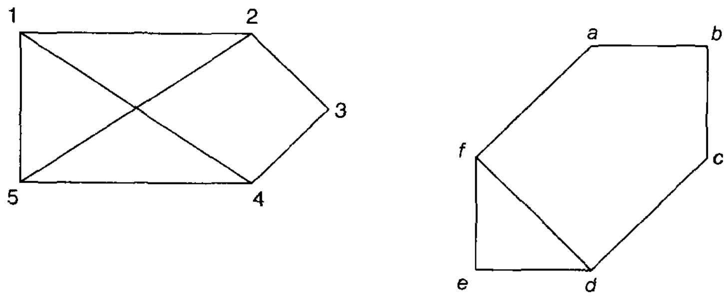


图 1.1 无向图示例

如果图 H 的顶点集合是图 G 顶点集合的子集，那么图 H 称为图 G 的一个诱导子图，而且图 H 中的边对应于图 G 中对应顶点所在的边。图的路径是由边相连的一系列顶点。不重复经过任何顶点的路径称为简单路径。如果图中任意两个顶点之间都有一条路径，那么称这个图是连通的。如果一条路径的起点和终点为同一顶点，那么称这条路为回路。如果一条回路上没有重复的顶点(除起点和终点)，那么称这条回路为简单回路。没有任何回路的连通图称为树。树中度数为 1 的点称为树的叶子节点。

如果图中两个顶点之间的边有方向，则称为有向图。在有向图中，以该顶点为起点(离开该顶点)的边的数目称为该顶点的出度，以该顶点为终点(指向该顶点)的边的数目称为该顶点的入度。

例 1.9 右边的有向图中(见图 1.2),点 2 的入度为 3,出度为 1。

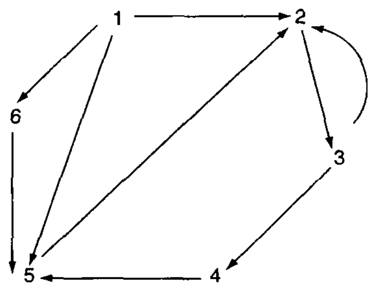


图 1.2 有向图示例

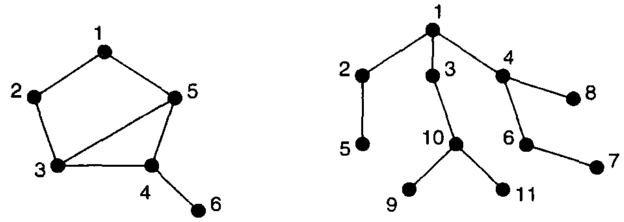
定义 1.2

1. 无向图中,如果任意一对顶点之间存在一条路径,那么称这个图是连通的。顶点之间的连续序列称为图中的路径。

2. 在图 G 中,如果一条路径的起点与终点相同,则称这条路径为回路。

3. 连通的、无回路的无向图是树。

例 1.10 考虑如下图例。



(i) (ii)

图(i)和图(ii)都是连通的。

1-2-3-4-6 是图(i)的路径。

图(i)包含一个回路 1-2-3-4-5-1。

图(ii)是一棵树。

下面是关于树的一些说明。

令 $G = (V, E)$ 为一个无向图,则下列阐述等价。

1. G 是一棵树。
2. G 中任意两点都有一条唯一的简单路径相连。
3. G 是连通的,但如果将任意边从 E 中删除,该图就不连通。
4. G 是连通的,且 $|E| = |V| - 1$。
5. G 没有回路,且 $|E| = |V| - 1$。
6. G 没有回路,但如果加入任意一条边,则新图包含一个回路。

以上性质均可证明,留作练习。

定义 1.3 在一棵树中,如果某一顶点不同于其他顶点,这个点称为树的根,这棵树称为有根树。

有根树也可以看成是一个有向图,该图只有一个点的入度为 0(根),其他点的入度均为 1。如果一个点的出度为 0,那么该点称为叶子节点。

1.4 语言:基本概念

对文法或自动机而言,其输入或基本数据结构是串。串通过一个有限的字母表定义。字母表根据应用的不同而变化。字母表中的元素称为符号。通常,将基本的字母表集合记为 Σ 或 T。下面是一些关于字母表的例子。

$$\Sigma_1 = \{a, b\}$$
$$\Sigma_2 = \{0, 1, 2\}$$
$$\Sigma_3 = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

串或词是一个由字母表中的符号组成的有限序列，在书写过程中，字符之间没有空格或逗号。例如，如果 $\Sigma = \{a, b\}$，$abbab$ 就是一个由 Σ 生成的串或词。如果 w 是 Σ 上的一个串，那么 w 的长度记为 $len(w)$ 或 $|w|$，表示 w 所包含的字符数。如果 $|w| = 0$，那么 w 被称为空串，记为 λ 或 ε。

对任意词 w，有 $w\varepsilon = \varepsilon w$。对于任意长度为 n 的串 $w = a_1 \cdots a_n$，w 的逆记为 w^R，表示为 $a_n a_{n-1} \cdots a_1$，每一个字符 a_i 都属于基本字母表 Σ。对于一个连续的串 z，如果其在串 w 内出现，则称串 z 是串 w 的子串。例如，aab 是 $baabb$ 的一个子串。

字母表 Σ 上所有串的集合记为 Σ^*，其中包括空串 ε。例如，对于 $\Sigma = \{0, 1\}$，$\Sigma^* = \{\varepsilon, 0, 1, 00, 01, 10, 11, \cdots\}$。需要注意的是，$\Sigma^*$ 是一个可数无限集。此外，Σ^n 表示 Σ 上所有长度为 n 的串的集合。因此，$\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \cdots$，$\Sigma^+ = \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \cdots$，$\Sigma^*$ 的子集称为语言。

例如，如果 $\Sigma = \{a, b\}$，那么 Σ 上的语言如下。

$L_1 = \{\varepsilon, a, b\}$

$L_2 = \{ab, aabb, aaabbb, \cdots\}$

$L_3 = \{w\varepsilon\Sigma^* \mid w$ 中 a' 的数目和 b' 的数目相等$\}$

上面几个例子中，L_1 是一个有限语言，L_2 和 L_3 是无限语言。ϕ 表示一个空语言。

有如下一些关于 Σ^+ 与 Σ^* 的归纳定义，其中，Σ 是一个基础字母表集合。

定义 1.4 设 Σ 是一个字母表集合，Σ^+ 是 Σ 上的一个满足下列定义的非空串。

1. 基础：如果 $\alpha \in \Sigma$，那么 $a \in \Sigma^+$。
2. 归纳：如果 $\alpha \in \Sigma^+$ 且 $a \in \Sigma$，αa，$a\alpha$ 那 α 属于 Σ^+。
3. 除(1)和(2)外，Σ^+ 中不含其他元素。

显然，集合 Σ^+ 包含了所有长度为 n 的串，$n \geq 1$。

例 1.11 设 $\Sigma = \{0, 1, 2\}$，则

$$\Sigma^+ = \{0, 1, 2, 00, 01, 02, 10, 11, 12, 20, 21, 22, \cdots\}$$

如果希望 Σ^+ 中包含 ε，那么需要把上述定义做如下改动。

定义 1.5 设 Σ 为任意字母表集合，Σ^* 的定义如下。

1. 基础：$\varepsilon \in \Sigma^*$。
2. 归纳：如果 $\alpha \in \Sigma^*$，$\alpha \in \Sigma$，那么 $a\alpha$，$\alpha a \in \Sigma^*$。
3. 除(1)和(2)外，Σ^* 中不含其他元素。

既然语言是一种集合，那么就可以运用集合论来定义语言的一些常规运算，如并、交、差和补等操作。

还有一些操作也是为语言定义的。如果 $x = a_1 \cdots a_n$，$y = b_1 \cdots b_m$，那么 x 和 y 的连接定义为 $xy = a_1 \cdots a_n b_1 \cdots b_m$。两个语言 L_1，L_2 的连接记为 $L_1 L_2 = \{w_1 w_2 \mid w_1 \in L_1$ 且 $w_2 \in L_2\}$。