“十一五”国家重点图书　　计算机科学与技术学科前沿丛书
计算机科学与技术学科研究生系列教材（中文版）

形式语言与自动机

朱保平　李千目　编著

清华大学出版社

“十一五”国家重点图书　计算机科学与技术学科前沿丛书

计算机科学与技术学科研究生系列教材（中文版）

形式语言与自动机

朱保平　李千目　编著

清华大学出版社
北京

内容简介

本书介绍了形式语言和自动机的基本理论和基本方法，主要内容包括计算理论基础、文法和语言、上下文无关文法及其语言、有穷状态自动机、正则语言与正则文法、正则语言的性质、下推自动机与上下文无关文法、上下文无关语言的性质、图灵机、图灵机的其他模型和其他计算模型等内容。

本书可作为高等院校计算机科学与技术及相关专业的教材，也可作为教师、研究生或软件技术人员的参考书。

图书在版编目(CIP)数据

形式语言与自动机/朱保平，李千目编著．-北京：清华大学出版社，2015

计算机科学与技术学科前沿丛　计算机科学与技术学科研究生系列教材(中文版)

ISBN 978-7-302-39975-9

Ⅰ．①形…　Ⅱ．①朱…　②李…　Ⅲ．①形式语言-研究生-教材　②自动机理论-研究生-教材　Ⅳ．①TP301

中国版本图书馆CIP数据核字(2015)第087777号

责任编辑：谢　琛　薛　阳

封面设计：傅瑞学

责任校对：时翠兰

责任印制：王静怡

出版发行：清华大学出版社

网　　址：http://www.tup.com.cn，http://www.wqbook.com

地　　址：北京清华大学学研大厦A座　　邮　编：100084

社 总 机：010-62770175　　邮　购：010-62786544

投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn

质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn

课 件 下 载：http://www.tup.com.cn，010-62795954

印 装 者：三河市金元印装有限公司

经　　销：全国新华书店

开　　本：185mm×260mm　　**印　张**：10　　**字　数**：245千字

版　　次：2015年8月第1版　　**印　次**：2015年8月第1次印刷

印　　数：1～1500

定　　价：25.00元

产品编号：061658-01

前言

“形式语言与自动机”是计算机科学与技术专业的重要理论基础课程，它不仅是计算机科学的核心课程，而且已成为电子信息类专业的热门课程。

形式语言与自动机是数学系统应用于计算的模型，形式语言是对语言的语法规则进行描述和分类的形式化方法；自动机是能够识别各种语言的自动化装置。形式语言与自动机在编译理论、人工智能、现代密码协议和通信等领域有着极其广泛的应用。

本书共分 11 章。第 1 章计算理论基础，介绍了离散数学的一些基础知识，主要包括集合及其表示，集合之间的关系，集合的运算，二元关系及其性质，等价关系与等价类，递归定义和递归证明，无向图、有向图、树；第 2 章文法和语言，包括文法的直观意义与形式定义，推导，归约，文法产生句型、句子和语言、文法的构造，乔姆斯基体系等内容；第 3 章上下文无关文法及其语言，主要介绍上下文无关文法的派生、派生树、文法的二义性、句法分析相关算法、上下文无关文法的化简、chomsky 和 Greibach 范式等内容；第 4 章有穷状态自动机，主要包括 DFA 和 NFA 的形式定义、DFA 和 NFA 接受的语言、DFA 和 NFA 的等价性、带空移动的 NFA 与 NFA 的等价性和 DFA 的最小化等内容；第 5 章正则语言与正则文法，包括正则表达式的定义、正则表达式与 FA 的等价性、正则文法与 FA 的等价性等内容；第 6 章正则语言的性质，包括正则语言的泵引理的证明及其应用，正则语言的封闭性，Myhill Nerode 定理与正则语言判定算法。第 7 章下推自动机与上下文无关文法，包括下推自动机的定义及其接受的语言、下推自动机的构造方法及与上下文无关文法的等价性、双栈自动机等；第 8 章上下文无关语言的性质，包括上下文无关语言的泵引理的应用、上下文无关语言的封闭性及判定算法。第 9 章图灵机，包括图灵机作为一个计算模型的基本定义、图灵机接受的语言、图灵机转换器和一些完成复杂任务的图灵机等；第 10 章图灵机的其他模型，包括带不动选择图灵机、多道图灵机、单向无穷带图灵机、离线图灵机、多带(头)图灵机和线性有界自动机等内容；第 11 章其他计算模型，主要包括递归函数的定义、递归函数的扩展理解、波斯特系统、矩阵文法和马尔可夫算法等内容。

本书第 1～第 5 章、第 10、第 11 章由朱保平编著，第 6、第 7 章由徐建编著，第 8、第 9 章由李千目编著。

由于作者水平有限，书中难免存在疏漏和不足之处，恳请读者和同行批评指正。

编著者

目录

第 1 章 计算理论基础

1.1 集合的基本概念

1.1.1 集合的定义

如果一个集合是某些确定的、能够区分的对象的聚合，那么可把其中的对象称为元素。设 a 为一个对象，A 为一个集合。如果 a 是集合 A 的一个元素，那么就称 a 属于集合 A，记为 $a \in A$。如果 a 不是集合 A 中的一个元素，就称 a 不属于 A，记为 $a \notin A$。例如，三种网络信息服务：微信、微博、博客的集合可以表示成

$$S = \{\text{微信}, \text{微博}, \text{博客}\}$$

1.1.2 集合的表示

集合通常有枚举法和抽象法两种表示方法。

1. 枚举法

枚举出集合中的所有元素。例如

$$A = \{1,2,3,4,5,6,7,8,9\}$$

2. 描象法

利用元素所具有的性质来描述集合。例如

$$A = \{x \in \mathbf{N} \mid 1 \leqslant x \leqslant 9\}$$

一般地，将集合中元素的特征用性质 P 来描述，例如

$$S = \{x \mid x \text{ 具有性质 } P\}$$

其意义是集合 S 由且仅由满足性质 P 的对象 x 所组成，即 $x \in S$ 当且仅当 x 具有性质 P。

1.1.3 集合间的关系

1. 子集

A，B 是两个集合，对于任意的 x，若 $x \in A$，则 $x \in B$，就称集合 A 是集合 B 的子集，也称集合 B 包含集合 A，记为 $A \subseteq B$。

若 A 不是 B 的子集，记为 $A \not\subset B$，也称 B 不包含 A。

如果 $A \subseteq B$，且 $\exists x \in B$，但 $x \notin A$，则称 A 是 B 的真子集，记作 $A \subset B$。

如果一个集合中一个元素也没有，称之为空集，记为 ϕ。空集是任何集合的子集。

2. 集合相等

A,B 是两个集合。若 $A\subseteq B$，且 $B\subseteq A$，则称 A 与 B 是相等的两个集合，记为 $A=B$。

1.1.4 集合的运算

1. 并、交、差

设 A 和 B 是两个集合，则

① 存在一个集合，它的元素是所有或者属于集合 A、或者属于集合 B 的元素，称这个集合为集合 A 与集合 B 的并集，记为 $A\cup B$，即

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

② 存在一个集合，它的元素是所有既属于集合 A、又属于集合 B 的元素，称这个集合为集合 A 与集合 B 的交集，记为 $A\cap B$，即

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

③ 存在一个集合，其元素是所有属于集合 A、但不属于集合 B 的元素，称这个集合为集合 A 与集合 B 的差，记之为 $A-B$，即

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

2. 集合的对称差

设 A 和 B 是两个集合，则存在一个集合，它的元素是所有或者属于 A 不属于 B、或者属于 B 不属于 A 的元素，称它为集合 A 和集合 B 的对称差，记为 $A\oplus B$，即

$$A \oplus B = \{x \mid x \in A \text{ 且 } x \notin B \text{ 或 } x \in B \text{ 且 } x \notin A\}$$

由定义不难知

$$A \oplus B = (A - B) \cup (B - A)$$

3. 幂集

设 A 是一个集合，存在一个集合，它是由 A 的所有子集为元素构成的集合，称它为集合 A 的幂集合，记为 $\rho(A)$ 或记为 2^A，即

$$\rho(A) = 2^A = \{x \mid x \subseteq A\}$$

4. 集合的补

在研究某一个具体问题时，往往规定一个集合，使所涉及的集合都是其子集合，称这个集合为全集，记为 U。全集是一个相对性的概念，不同的问题，可以规定不同的全集。

设 A 是一个任意集合，则

$$A \cap U = A$$

$$A \cup \phi = A$$

设 A 是一个集合，U 是全集合，称集合 $U-A$ 为 A 的补集，记为 $\overline{A}$，即

$$\overline{A} = U - A = \{x \mid x \in U, x \notin A\}$$

显然有

$$A \cup \overline{A} = U$$

$$A \cap \overline{A} = \phi$$

5. 集合的笛卡儿积集

设 A 和 B 是两个集合，存在一个集合，它的元素是用 A 中元素为第一元素，B 中元素为第二元素构成的有序二元组，称它为集合 A 和 B 的笛卡儿积集，记为 $A\times B$，即

$$A \times B = \{(a,b) \mid a \in A, b \in B\}$$

例如，$A=\{0,1\}$，$b=\{a,b,c\}$，则

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

1.2　关系

1.2.1　二元关系

设 A，B 是两个集合，R 是 $A\times B$ 的任意一个子集，即

$$R \subseteq A \times B$$

则称 R 为从集合 A 到集合 B 的一个二元关系，简称为从 A 到 B 的一个二元关系。

若 $R=\phi$，称 R 为空关系。

若 $R=A\times B$，称为全关系。

当 $A=B$ 时，称二元关系 $R\subseteq A\times A$ 为 A 上的二元关系。

当 $A=B$ 时，记 $\Delta_A=\{(x,x)\mid x\in A\}$，称之为 A 上的恒等关系。

1.2.2　二元关系的性质

设 R 是集合 A 上的一个二元关系，即 $R\subseteq A\times B$，于是

(1) 若对于 $\forall x\in A$，满足 $(x,x)\in R$，则称关系 R 有自反性，或称 R 是 A 上的自反关系。

(2) 若对于 $\forall x\in A$，满足 $(x,x)\notin R$，则称关系 R 有反自反性，或称 R 是 A 上的反自反关系。

(3) 若对于 $\forall x,y\in A$，满足当 $(x,y)\in R$，有 $(y,x)\in R$，则称关系 R 有对称性，或称 R 是 A 上的对称关系。

(4) 若对于 $\forall x,y\in A$，满足当 $(x,y)\in R$，且 $(y,x)\in R$，有 $x=y$，则称关系 R 有反对称性，或称 R 是 A 上的反对称关系。

(5) 若对于 $\forall x,y,z\in A$，满足当 $(x,y)\in R$，且 $(y,z)\in R$ 时，有 $(x,z)\in R$，则称关系 R 有传递性，或称 R 是 A 上的传递关系。

1.2.3　等价关系

A 是一个非空集，R 是 A 上的一个二元关系，若 R 满足自反性、对称性、传递性，则称 R 是 A 上的等价关系。

若 R 是 A 上的等价关系，a 是 A 中任意一个元素，称集合 $\{x\in A\mid (x,a)\in R\}$ 为集合

A 关于关系 R 的一个等价类，记为$[a]_R$，即

$$[a]_R = \{x \in A \mid (x,a) \in R\}$$

例如

设 $A=\{1,2,3\}$，$R=\{(1,1),(2,2),(3,3),(1,2),(2,1)\}$，显然 R 是 A 上的一个等价关系，于是有

$$[1]_R = \{1,2\}$$
$$[2]_R = \{1,2\}$$
$$[3]_R = \{3\}$$

可以看出$[1]_R=[2]_R$，说明同一个等价类可以选取不同的代表元。

1.2.4 递归定义与归纳证明

递归定义又称为归纳定义，它由三部分组成。

(1) 基础条款：用来定义该集合的最基本的元素；

(2) 归纳条款：用来构造集合的新元素的规则；

(3) 最小性条款：指出一个对象是所定义集合中的元素的充要条件是它可以通过有限次的使用基础条款和归纳条款中所给的规定构造出来。

例 1.1 用递归定义定义$\{a,b\}$上所有以 a 打头，长度为偶数的所有字符串的集合。

设 L 是满足以上条件的语言，L 中的元素满足以下条件：

(1) $aa,ab \in L$；

(2) 若 $u \in L$，则 $uaa,uab,uba,ubb \in L$；

(3) L 中的每个字符串是有限使用(1)(2)所得的。

例 1.2 斐波那契(Fibonacci)数列的定义。

(1) 基础条款：0 是第一个斐波那契数，1 是第二个斐波那契数；

(2) 归纳条款：如果 n 是第 i 个斐波那契数，m 是第 $i+1$ 个斐波那契数，则 $n+m$ 是第 $i+2$ 个斐波那契数，其中 i 为大于等于 1 的正整数；

(3) 最小性条款：只有满足(1)和(2)的数才是斐波那契数。

根据以上定义可以给出部分斐波那契数如下：

$$0,1,1,2,3,5,8,13,21,34,55,\cdots$$

对递归定义三要素(基础条款、归纳条款、最小性条款)的充分理解，可以解决递归子程序的形式化描述问题以及程序开发过程中软件设计的完备性问题。

下面以程序设计中的实际问题加以说明。

例 1.3 构造函数 $f(n)=n!$ 的递归子程序。

多数程序设计教材在描述该递归子程序时一般会如下描述：

```
int f(int n){
    if(n==0||n==1)return 1;
    else return n*f(n-1);
}
```

该程序对于一个正确的输入($n\geqslant 0$)，会有一个正确的输出。但对于一个错误的输入($n<0$)，程序未能给出合理的反应。其理由是该程序未能满足递归定义的最小性条款。一

种完备的处理方式如下：

```
int f(int n){
    if(n==0||n==1)return 1;              //基础条款
    else if(n>0)return n*f(n-1);         //归纳条款
    else exit(0);                        //最小性条款,非正常退出,具体问题可做相应处理
}
```

例 1.4　已知二叉链表的定义如下。

```
typedef struct Bitnode{
    char data;
    Bitnode *lchild,*rchild;
}Bitnode;
```

试构造一递归子程序,计算二叉树叶子的个数。

```
int countleaf(Bitnode *t){
    if(strcmp(typeid(t).name(),"Bitnode *")==0)
        if(t==NULL)return 0;                 //基础条款
        else{
            int m=countleaf(t->lchild);
            int n=countleaf(t->rchild);
            if(m+n==0)return 1;
            else return m+n;
        }                                    //归纳条款
    else exit(0);                            //最小性条款,非正常退出,具体问题可做相应处理
}
```

递归定义提供了一种良好的定义方式,使得集合中的元素由构造规律明确地表现出来,这给集合性质的归纳证明提供了良好的基础。归纳证明与归纳定义相对应,也由三步组成。

(1) 基础：证明该集合中最基本的元素具有性质 P。

(2) 归纳：证明如果被定义集合中的元素都具有性质 P,则用某种运算、函数或组合方法对这些元素处理后所得的结果也有性质 P。

(3) 由归纳法原理,集合中的所有元素具有性质 P。

例 1.5　对于有穷集合 A,证明 $|2^A|=2^{|A|}$。

证明：对 $|A|$ 进行归纳证明。

(1) 基础：当 $|A|=0$ 时,即 $A=\phi$,于是 $|A|=0$,$|2^A|=|\{\phi\}|=1=2^{|A|}=2^0=1$,所以命题成立。

(2) 归纳：假设 $|A|=n$ 时结论成立,要证 $|A|=n+1$ 结论成立。

不失一般性,设 $A=B\cup\{a\}$,其中 $a\notin B$,于是 $|A|=|B\cup\{a\}|=|B|+1$。

根据幂集的定义可知,$2^A=2^B\cup\{C\cup\{a\}|C\in 2^B\}$。由于 $a\notin B$,所以 $2^B\cap\{C\cup\{a\}|C\in 2^B\}=\phi$。显然,可以构造一个双射 $f:\{C\cup\{a\}|C\in 2^B\}\rightarrow 2^B$,使得 $f(C\cup\{a\})=C$。所以 $|\{C\cup\{a\}|C\in 2^B\}|=|2^B|$。于是

$$|2^A|=|2^B\cup\{C\cup\{a\}|C\in 2^B\}|$$

$$
\begin{aligned}
&= \mid 2^B \mid + \mid \{C \cup \{a\} \mid C \in 2^B\} \mid \\
&= \mid 2^B \mid + \mid 2^B \mid \\
&= 2 \mid 2^B \mid
\end{aligned}
$$

显然，$|B| = n$。根据归纳假设知，$|2^B| = 2^{|B|}$。从而有

$$
\mid 2^A \mid = 2 \mid 2^B \mid = 2 \times 2^{|B|} = 2^{|B|+1} = 2^{|A|}
$$

所以，$|A| = n+1$ 时，命题成立。

(3) 由归纳法原理知，结论对任意有穷集合成立。

例 1.6 证明 $\sum_{i=1}^{n} \frac{1}{i^2} \leqslant 2 - \frac{1}{n}$。

证明：

(1) 基础：$n=1$ 时，$\sum_{i=1}^{1} \frac{1}{i^2} = \frac{1}{1^2} = 1 \leqslant 2 - \frac{1}{1} = 1$，命题成立。

(2) 归纳：假设 $n=k$ 时，命题成立，即 $\sum_{i=1}^{k} \frac{1}{i^2} \leqslant 2 - \frac{1}{k}$，要证 $n=k+1$ 时命题成立。

$$
\begin{aligned}
\sum_{i=1}^{k+1} \frac{1}{i^2} &= \sum_{i=1}^{k} \frac{1}{k^2} + \frac{1}{(k+1)^2} \\
&\leqslant 2 - \frac{1}{k} + \frac{1}{(k+1)^2} \\
&= 2 + \frac{k-(k+1)^2}{k(k+1)^2} \\
&= 2 + \frac{k-k^2-2k-1}{k(k+1)^2} \\
&= 2 - \frac{k^2+k+1}{k(k+1)^2} \\
&= 2 - \frac{k(k+1)}{k(k+1)^2} - \frac{1}{k(k+1)^2} \\
&= 2 - \frac{1}{k+1} - \frac{1}{k(k+1)^2} \\
&\leqslant 2 - \frac{1}{k+1}
\end{aligned}
$$

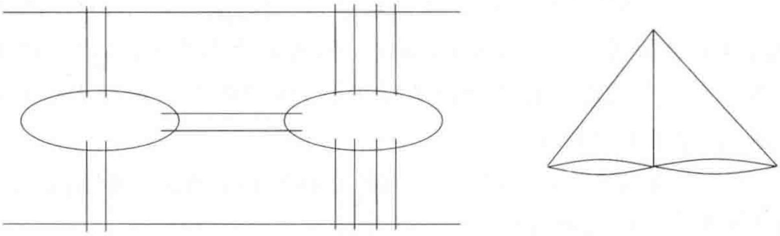
所以，当 $n=k+1$ 时命题成立。

(3) 由归纳法原理知，结论对任意大于 0 的自然数均成立。

1.3 图与树

18 世纪初，在当时德国哥德尼斯堡(Konigsberg)城的普雷格尔(Pregel)河上有 7 座桥，哥德尼斯堡的地图如图 1.1(a)表示。它也可以表示为图 1.1(b)，图中的边表示桥，而顶点表示岛和河的两岸。当地人经常在桥上散步，有人提出，从岛和河岸的某一处出发是否能找到一条通过每一座桥一次且仅一次的通路。这个问题也相当于在图 1.1(b)中找一条通过每条边一次且仅一次的通路。1736 年，欧拉(L. Euler)解决了这个问题。从此，欧拉成为图

论之父。



(a)　　(b)

图 1.1　哥德尼斯堡城图

由上例不难看出，在讨论离散对象和二元关系中的许多问题时，用图来表示这些对象以及关于它们的二元关系是十分方便的。这就很自然地导致人们对图的理论进行研究。

1.3.1　无向图

一个二元组 $G=(V,E)$是一个无向图，若其中 V 是一个非空有限集合，E 是 V 上的二元关系，称 V 为顶点集，E 为边集。

$G=(V,E)$是一个无向图，$\{v,u\}\in E$，称$\{v,u\}$是 G 中的一条边。设 $e=\{v,u\}$是 G 中的一条边，v 和 u 称为边 e 的两个端点，称边 e 关联 v 和 u，也称 v 邻接 u，或 u 邻接 v。若 $u=v$，称$\{v,u\}$为 G 中的自环。对于任意的 $u\in V$，若不存在任何边关联 u，称顶点 u 是孤立点。

一个简单图，若每一对不同的顶点之间都有一条边相连，这样的图称为完全图。一个有 $n(\in\mathbf{N})$个顶点的完全图在同构的意义下是唯一的，记为 K_n。图 1.2 给出了 K_1，K_2，K_3，K_4 和 K_5。

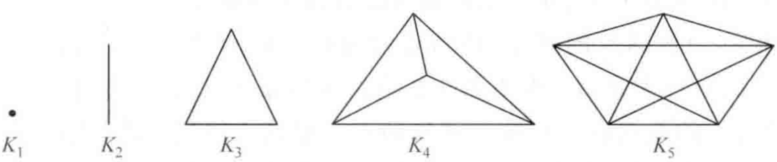


图 1.2　若干完全图

$G=(V,E)$是一个图，对于每一个 $v\in V$，称关联顶点 v 的边数为顶点 v 的度数，记为 $d(v)$。对于任何一个无向图，图中所有顶点的度数之和为图中边数的 2 倍，即

$$\sum_{v\in V} d(v) = 2\mid E\mid$$

设 $G=(V,E)$是一个无向图，称一个顶点序列$(v_{i_1},v_{i_2},\cdots,v_{i_k})$为图 G 的一条通路。其中，$v_{i_j}\in V$，$1\leqslant j\leqslant s$，且$(v_{i_j},v_{i_{j+1}})\in E$，其中 $1\leqslant j\leqslant s-1$。

设$(v_{i_1},v_{i_2},\cdots,v_{i_k})$是 G 中的一条通路，若 $v_{i_1}=v_{i_s}$，则这条通路为 G 中的一条回路，若一个回路中边不重复出现，则称为简单回路，若顶点不重复出现，则称为初等回路，初等回路又称为圈。一条回路的长度，就是这条回路经过的边的条数。

1.3.2　有向图

称有序二元组 $G=(V,E)$是一个有向图，其中 V 是一个非空有限集合，E 是 V 上的二

元关系，称 V 为顶点集，E 为边集。

设 $G=(V,E)$ 是一个有向图，其中 V 为顶点集，E 为边集。若 $(a,b)\in E$，称 (a,b) 为图 G 中的一条边，称边 (a,b) 关联于顶点 a 和 b，顶点 a 称为该边的始点，顶点 b 称为该边的终点，并称 a 和 b 相邻。若一个顶点没有任何边关联于它，称该顶点为孤立点。若一条边的始点和终点是同一顶点，称该边为自环。

$G=(V,E)$ 是一个有向图，$v\in V$，称以 v 为始点的边的条数为 v 的出度，记为 $d_{出}(v)$；以 v 为终点的边的条数为 v 的入度，记为 $d_{入}(v)$，且有

$$\sum_{v\in V} d_{出}(v) = \sum_{v\in V} d_{入}(v) = |E|$$

1.3.3 树

树结构是一类重要的非线性结构。树状结构是结点之间有分支，并且具有层次关系的结构，它非常类似于自然界中的树。树结构在客观世界是大量存在的，例如家谱、行政组织机构都可用树形象地表示。树在计算机领域中也有着广泛的应用，例如在编译程序中，用树来表示源程序的语法结构；在数据库系统中，可用树来组织信息；在分析算法的行为时，可用树来描述其执行过程等。

树是 $n(n\geqslant 0)$ 个结点的有限集 T，T 为空时称为空树，否则它满足以下两个条件：

(1) 有且仅有一个特定的称为根的结点；

(2) 其余的结点可分为 $m(m\geqslant 0)$ 个互不相交的子集 $T_1,T_2,\cdots,T_m$，其中每个子集又是一棵树，并称其为子树。

树结点包含一个数据元素及若干指向子树的分支。结点拥有子树的个数称为结点的度，树中结点的最大度数称为树的度。结点的子树的根称为该结点的孩子，该结点称为孩子的双亲。同一双亲的孩子结点称为兄弟。度不为 0 的结点称为分支点或非终端结点，度为 0 的结点称为树叶或终端结点。根结点的层次定义为 1，根的孩子为第二层结点，以此类推。树中结点的最大层次称为树的深度或称树高。图 1.3 给出了一棵树的示例。

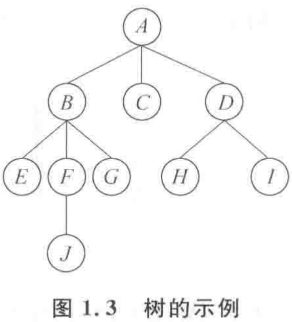


图 1.3 树的示例

树中结点 D 的度数为 2；树的度为 3；结点 E、F、G 为结点 B 的孩子；结点 B、C、D 为兄弟；结点 A、B、D、F 为分支点；结点 C、E、G、H、I、J 为树叶。

1.3.4 二叉树

二叉树在树结构的应用中起着非常重要的作用，因为对二叉树的许多操作算法简单，而任何树都可以与二叉树相互转换，这样就解决了树的存储结构及其运算中存在的复杂性。

二叉树是由 $n(n\geqslant 0)$ 个结点的有限集合构成的，此集合或者为空集，或者由一个根结点及两棵互不相交的左右子树组成，并且左右子树都是二叉树。

1.4 三个重要概念

语言、文法和自动机是本书的三个重要主题，本书将讨论这些概念的相关结论以及它们间的关系。

语言学家 Chomsky 最初从定义语言的角度研究语言。1956 年，他将语言 L 定义为字母表 Σ 中的字符组成的一些串的集合。

1. 语言研究的三个方面

表示：无穷语言的表示；

有穷描述：研究的语言要么是有穷的，要么是可数无穷的，这里主要研究可数无穷语言的有穷描述；

结构：语言的结构特征。

2. 基本概念

1）字母表

字母表是一个非空有穷集合，字母表中的元素称为该字母表中的符号。

例如，$\Sigma=\{0,1\}$等。

2）字母表的积运算

$$\Sigma_1 \times \Sigma_2 = \{ab \mid a \in \Sigma_1, b \in \Sigma_2\}$$

例如，$\Sigma_1=\{0,1\}$，$\Sigma_2=\{a,b\}$，则有

$$\Sigma_1 \times \Sigma_2 = \{0a, 0b, 1a, 1b\}$$

3）字母表 Σ 的 n 次幂

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

其中 ε 称为空串。

$$\Sigma^n = \Sigma^{n-1} \times \Sigma$$

$$\Sigma^+ = \Sigma \cup \Sigma^1 \cup \Sigma^2 \cup \cdots$$

称为 Σ 的正闭包。

$$\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \cdots$$

称为 Σ 的克林闭包。

直观理解，

$$\Sigma^* = \{x \mid x \text{ 是 } \Sigma \text{ 中若干个字符连接而成的字符串}\}$$

$$\Sigma^+ = \{x \mid x \text{ 是 } \Sigma \text{ 中至少一个字符连接而成的字符串}\}$$

例如，已知 $\Sigma=\{0,1\}$，则有

$$\Sigma^* = \{\varepsilon, 0, 1, 00, 11, 000, 001, \cdots\}$$

$$\Sigma^+ = \{0, 1, 00, 11, 000, 001, \cdots\}$$

4）句子

Σ 是一个字母表，对于 Σ^* 上的任何元素 x，x 叫作 Σ 上的一个句子。