计算机科学丛书

自动机理论及其应用

何成武 著

科学出版社

计算机科学丛书

自动机理论及其应用

何成武 著

科学出版社

1990

内 容 简 介

本书系统、扼要地介绍自动机的基本理论及其在形式语言、数字系统结构设计和自动综合、微程序设计自动化、可编程逻辑阵列、收缩阵列(SYSTOLIC)设计、模式识别等领域中的应用。本书取材广泛，反映了自动机理论应用的若干新的方向和进展。

本书适于自动控制、系统工程、计算技术、人工智能及生物医学工程等领域的工程技术人员、研究人员以及相应专业的大学生、研究生阅读。

计算机科学丛书

自动机理论及其应用

何成武 著

责任编辑 那莉莉

科学出版社出版

北京东黄城根北街16号

邮政编码：100707

中国科学院印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

1990年8月第一版 开本：850×1168 1/32

1990年8月第一次印刷 印张：10 1/4

印数：0001—2 100 字数：266 000

ISBN 7-03-001561-4

TP·107

定价：10.40元

计算机科学丛书
编 委 会

序　言

自动机理论作为离散自动装置和计算机的理论模型建立以来，在其发展过程中，以各种数学工具揭示各类离散自动机的内在规律，为其行为描述和结构设计提供理论依据和严格论证，成为信息技术和计算机科学的理论学科之一.

随着微电子技术和信息科学的发展，自动机理论不断向信息技术的各个应用领域渗透，为它们提供理论模型、设计技术和运行算法. 掌握自动机的基本理论及其应用技术，有助于工程技术人员在实际工程的系统分析、结构综合、工程实现和模拟检测方面提高其设计技巧和理论分析能力. 近年来人们对自动机理论的兴趣有了很大的增长，要求有一本综合介绍自动机的基本理论在若干重要领域应用的书籍. 作者根据自己多年来给研究生上课的讲义，纳入若干应用技术写成此书，期望能对感兴趣的读者有所裨益.

本书原则上不假定读者具备什么专门的准备知识，但学过离散数学或具有计算机软、硬件工程的实践经验，会有助于加深对本书内容的理解.

本书在内容的陈述上，采取从一般到特殊的方法，从图灵机及其等价语言讲起，对图灵机施加不同的限制，得到不同计算能力的自动机. 先介绍自动机的基本原理及理论性质，再介绍实际应用. 在自动机的表示形式和证明方面，本书有一定特色.

本书的主要内容可以分成四部分：(1) 导引和第一章介绍作为计算模型的图灵机基本理论，这是讨论各类自动机问题的基础. (2) 第二章—第四章论述有关各类自动机与抽象程序语言类的关系. 数理语言学是软件科学及人工智能的重要基础，这部分介绍了自动机用于形式语言研究的主要结果. (3) 第五章—第八章讨论

自动机在数字系统CAD方面的应用.自动机理论在自动综合、微程序设计、可编程序逻辑阵列PLA、收缩阵列SYSTOLIC方面的应用日渐深入，辅助设计技术已发展到不仅对数字系统的零部件进行自动设计，而且对整个系统进行整体化的自动设计.这几章着重介绍数学系统自动设计的基本思想方法和算法基础，而不涉及具体的实现问题.(4) 第九章—第十一章介绍自动机在模式识别、概率语言和学习自动机等人工智能领域中的应用. 本部分用句法方法处理模式识别问题，从描述模式的结构出发，将模式的结构和语言的句法两者之间的相似性加以引伸，使得自动机和数理语言学的方法得以运用于模式识别.这部分还扼要地介绍了概率语言、学习自动机等方面的基本概念和有代表性的模型. 这些概念在人工智能、控制论和系统工程等领域有着重要的应用，这些研究仍有待进一步深入.

在本书的撰写过程中，得到我的学生窦如静、郭广慧和汤元元的大力协助，他们在整理讲义和全书的文字加工、校对方面付出了大量的劳动；陶仁骥研究员对本书提出了很多宝贵意见；作者在此深表感谢. 限于本人的学识水平，书中的问题和缺点定然不少，敬祈专家和读者指正.

作者

1989年3月1日

于华东计算技术研究所

目　录

导　引

计算机科学的几个基本概念是由数理逻辑学家提出的. 会做机械运算的自动机是这类基本概念之一. Pascal, Babage 和Leibniz 构作了几乎不会犯错误的计算机, 但是这些机器没有坚实的理论背景. 1936 年, Turing 把计算机的理论大大推进, 给予自动机以今天具有的技术含义. Turing 定义了一类简单的机器——图灵机, 它能够计算任何可以定义的实数. 图灵机是现代计算机最早的数学模型. 它在输入纸带上每次读一个符号, 再按照读出的符号改变"状态", 并改变带上这个被读的符号, 然后左/右移一格, 再注视下一个符号, 如此继续. 对这种机器施加不同的限制就产生了不同能力的自动机, 如有限自动机、下推自动机、线性有界自动机等. 对这些机器进行研究的理论叫做自动机理论, 它是计算机科学的一个分支.

同自动机理论密切相关的是形式语言理论. 形式语言是用数学的方法从自然语言抽象出形式文法, 研究它们的结构和性质. 各种形式语言以及它们的文法是研究计算机程序的工具. 一个形式文法 G 由一个四元组(V_N, V_T, P, S)定义, 其中

V_N 是有限的变量字母集合

V_T 是有限的终极字母集合 (V_T 与 V_N 不含公共元素)

P 是有限的产生式集合

S 是开始符号$(S \in V_N)$

产生式由 $\alpha \to \beta$ 形式的表达式组成. 其中 α 和 β 是由变量字母和终极字母组成的字符串, α 至少含有一个字符.

一个字母集合(也称字母表)上的句子是由该集合中的字母组成的有限长度的字符串、行和字是句子的同义词、空句子 ε 是不含符号的句子. 我们用 V 表示 V_N 和 V_T 的并集, V^* 表示字母表

V 中字母组成的所有字的集合(包括空句子)，V^+ 表示集合 $V^*-\{\varepsilon\}$. 我们把一个字母表上句子的任何集合称为语言.

如果 $\alpha\to\beta$ 是一个产生式，γ 和 δ 是 V^* 中的任意字，则在文法 G 中，从 $\gamma\alpha\delta$ 可直接导出 $\gamma\beta\delta$，记为

$$\gamma\alpha\delta \underset{G}{\Rightarrow} \gamma\beta\delta$$

假设有一个 V^* 中的字的序列 $\alpha_1, \alpha_2, \cdots, \alpha_m$，且有直接导出关系

$$\alpha_1 \underset{G}{\Rightarrow} \alpha_2, \cdots, \alpha_{m-1} \underset{G}{\Rightarrow} \alpha_m$$

则称从 α_1 导出 α_m，记为 $\alpha_1 \overset{*}{\underset{G}{\Rightarrow}} \alpha_m$. 我们约定，对任何的 $\alpha\in V^*$，均有 $\alpha \overset{*}{\underset{G}{\Rightarrow}} \alpha$. 我们把由文法 G 产生的语言定义为如下集合：

$$\{w \mid w\in V_T^* \quad 且 \quad S \overset{*}{\underset{G}{\Rightarrow}} w\}$$

即从 S 出发、经有限次使用产生式 P 进行替换而导出的、由终极字母组成的句子的集合.

【例 1】 英文句子 *That stupid student gives me a pain* 可以看成是由图 1 所示的过程产生的.

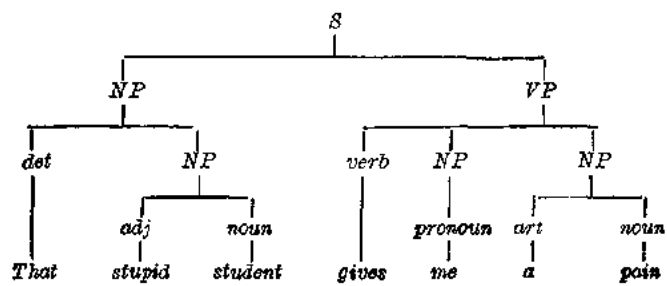

图 1 一个英文句子的导出

其产生式集合为

$S\to NPVP$　　$verb\to gives$

$NP\to det\ NP$　　$NP\to pronoun$

$det \to That$　　$pronoun \to me$

$NP \to adj\ noun$　　$NP \to art\ noun$

$adj \to stupid$　　$art \to a$

$noun \to student$　　$noun \to pain$

$VP \to verb\ NP\ NP$

上面的英文句子就是从 S 出发，经过一系列替换而产生的终极符号串.

形式文法可以根据其产生式的性质分成若干类，其中特别有趣的一类是其产生式为 $A \to \beta$ 形式的文法. 这里 A 是单个变量字母，而 β 是终极字母和变量字母组成的字符串. 这种文法称为上下文无关文法，它产生的语言称为上下文无关语言. 因为它由简单的语法规则产生，又可以表示大部分实际的程序语言.

为方便起见，我们用指数 n 表示一个字母或若干个字母的 n 次连续出现，例如 $a^3 = aa$，$(ab)^3 = ababab$.

【例 2】 产生上下文无关语言 $L = \{a^n b^n \mid n \geqslant 1\}$ 的上下文无关文法是 $G = (V_N, V_T, P, S)$，其中

$$V_N = \{S\},\quad V_T = \{a, b\}$$

$$P \text{ 为 } S \to a\,S\,b,\ S \to ab$$

从 S 开始，运用第一个产生式 $n-1$ 次，然后再用第二个产生式一次，就可以得到字 $a^n b^n$：

$$S \underset{G}{\Rightarrow} a\,S\,b \underset{G}{\Rightarrow} aa\,S\,bb \underset{G}{\Rightarrow} a^3\,S\,b^3$$

$$\underset{G}{\Rightarrow} \cdots \underset{G}{\Rightarrow} a^{n-1}\,S\,b^{n-1} \underset{G}{\Rightarrow} a^n b^n$$

下面的随机访问机器可以看成是现代计算机的抽象模型.

图 2 所示的随机访问机器 (RAM) 是一个累加器的计算机模型，其程序的指令不允许修改. 它由一个只读输入带、一个只写输出带、一个程序和存储器组成. 输入带被分成格子，每格放一个整数，每读入一个数时，读头右移一格. 输出带也被分成格子，开始启动时，全部格子均为空. 每执行一条写指令，就对写入头下的格子写入一个数，然后写入头右移一格. 存储器由一系列的寄存器

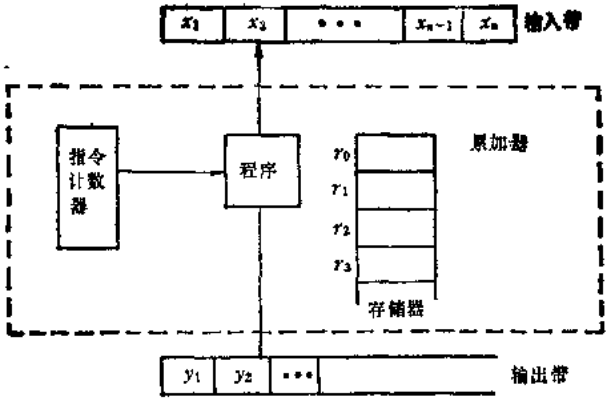

图2

$r_0, r_1, r_2, \cdots$ 组成，每个寄存器可以存放一个数. 我们假定寄存器的字长及个数对于所解决的问题是足够的，程序不存放在存储器里（因而在执行过程中不能修改程序本身）. 程序是指令的序列，各条指令依次执行，由一个指令计数器控制，只有转移指令可以改变执行次序. 寄存器 r_0 是累加寄存器. 指令类型可参见下面一段用类似 ALGOL 的语言编写的程序：

```
begin
        read  r1;
        if   r1≤0  then  write  0  else
            begin
                    r2 ← r1;
                    r3 ← r1 − 1;
                    while  r3>0  do
                          begin
                                r2 ← r2*r1;
                                r3 ← r3 − 1;
                            end
                  write  r2;
             end
        end
```

上述程序中 r_2 存放计算结果 n^n，r_3 记录 n 自乘的次数，r_3 的初值

为 $n-1$.

还可以进一步抽象，例如把输入输出带结合起来，用一根既可读又可写的带子表示，磁带就是这样的带子的体现. 各个寄存器可以用机器的状态表示. 图灵机就是这种机器的高度抽象的模型. 可以证明图灵机与前面的随机访问机器是等价的，即它所计算的函数恰好是所有的"部分递归函数"(详见第一章). 对图灵机施加不同的限制，就得到不同的自动机.

自动机理论研究的各类机器同 Chomsky 等人所定义的抽象程序语言类（如上下文无关语言类）之间存在着直接的对应关系. 各类语言均可由其对应的自动机接受. 被自动机 A 接受的语言 L 是 V_T 中这样的符号串 w 的集合：如果把 w 记录在机器的带子上，自动机从开始状态开始工作，最终会停机于结束状态.

各类文法和自动机的对应关系见下表：

语言和文法的类型	对产生式的限制	自动机类型
0 型（递归可枚举语言）	$\alpha \to \beta$, $\alpha \in V^+$, $\beta \in V^*$	图灵机
1 型（上下文有关语言）	$\alpha \to \beta$, $\alpha \in V^+$, $\beta \in V^+$, $\lvert\alpha\rvert \leqslant \lvert\beta\rvert$	线性有界自动机
2 型（上下文无关语言）	$A \to w$, $A \in V_N$, $w \in V^*$	下推自动机
3 型（有限状态语言）	$A \to aQ$, $A \to a$, A, $Q \in V_N$, $a \in V_T$	有限自动机

注：$|\alpha|$和$|\beta|$分别表示字符串α和β的长度，即各自所含字符的个数.

与其他讲述自动机理论的书不同，我们把图灵机提到前面来讲述. 这不仅因为历史上图灵机的出现先于其他自动机若干年，许多自动机模型只是图灵机的变型，而且因为图灵机理论是讨论自动机问题(例如判定问题)的基础.

第一章　图灵机和递归可枚举语言

1.1　图灵机的基本概念

1.1.1　图灵机的定义

最基本的图灵机由控制器、带子和读写头(或称带头)组成,其结构和作用简述如下:

1) 控制器有有限个状态,在工作过程中它可以改变状态.

2) 带子的左端是有界的(它可以从左端起记录输入符号),但其右端则可以无限延伸. 带子被分成许多格子(即单元),每格可以记录一个符号.输入符号记录在从左端开始的连续单元上,其余单元为空格,记为 B.

3) 读写头可以左右移动,既可扫描符号,也可写下符号.

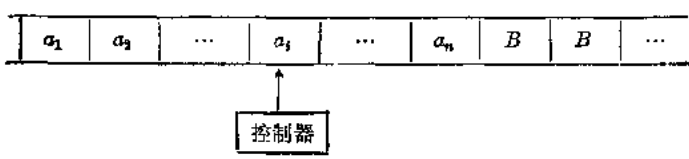

图 1.1　基本图灵机

一个图灵机可定义为六元组,即

$$T=(K,\ \Sigma,\ \Gamma,\ \delta,\ q_1,\ F)$$

其中,K 是状态的有限集合;Σ 是输入字母表(字母的有限集合);Γ 是可在带子上记录的符号的有限集合,包括空白符号,因此 $\Sigma\subseteq\Gamma$;q_1 是 K 中的初始状态;$F\subseteq K$ 是结束状态(或称接受状态)集合.

δ 是 $K\times\Gamma$ 到 $K\times(\Gamma-\{B\})\times D$ 的映射(也可称为指令),其一般形式为 $\delta(q,\ a)=(p,\ b,\ D)$,其中 $D=\{L,\ R,\ S\}$.注意,$\delta(q_i,\ a)$ 的象是唯一的.

这个映射的意思是,机器在状态 q 并注视符号 a 时,把 q 变成状态 p, 并在 a 所占单元中用 b 取代 a, 同时向左或向右移动一个单元(转向注视下一个单元)或不移动.

1.1.2 图灵机的格局

图灵机的格局是一个三元组 $(q,\ \alpha,\ i)$, 其中 q 是 K 中的状态; α 是带子上的字,且 $\alpha\in(\Gamma-\{B\})^*$; i 是一个整数,它表示从读写头所处的位置到带子左端(也就是 α 左端)的距离.

图灵机(即 Turing machine, 简记为 T_m 或 T)根据指令对格局进行变换操作.

设 $(q,\ A_1A_2\cdots A_n,\ i)$ 是 T 的一个格局,T 有一条指令为

$$\delta(q,\ A_i)=(p,\ A,\ D),\quad i=1,\ 2,\ \cdots,\ n+1$$

1) 对于 $i\leqslant n$ 的情形:

a) 若 $D=R$, 则有

$$(q,\ A_1A_2\cdots A_n,\ i)\vdash_T(p,\ A_1A_2\cdots A_{i-1}AA_{i+1}\cdots A_n,\ i+1)$$

即 T 写下符号 A, 转到状态 p, , 并右移一格.

b) 若 $D=L$, 因它不能超出最左单元, 即当 $q=q_1$ 时, $D\neq L$. 所以 $2\leqslant i$, 这时有

$$(q,\ A_1A_2\cdots A_n,\ i)\vdash_T(p,\ A_1A_2\cdots A_{i-1}AA_{i+1}\cdots A_n,\ i-1)$$

即 T 写下符号 A, 转到状态 p, 并左移一格.

2) 对于 $i=n+1$ 的情形,读写头注视着右端空白 B:

a) 若 $D=R$, 有

$$(q,\ A_1A_2\cdots A_n,\ n+1)\vdash_T(p,\ A_1A_2\cdots A_nA,\ n+2)$$

b) 若 $D=L$, 有

$$(q,\ A_1A_2\cdots A_n,\ n+1)\vdash_T(p,\ A_1A_2\cdots A_nA,\ n)$$

两个格局以 $\vdash_T$ 相联系的意思是前一格局 C_1 经一次(或一步)操作改造成为后一格局 C_2. 如果前一格局经过有限次(包括零次)操作改造成后一格局, 我们就用 $\vdash_T^*$ 连接它们, 记为 $C_1\vdash_T^* C_2$.

1.1.3 图灵机接受和产生语言

被图灵机 T 接受的语言 L 是 Σ^* 中这样的符号串 w 的集合：如果把该符号串 w 记录在 T 的带子上，开始工作时 T 处于初始状态 q_1，读写头处于最左端，则经过有限次操作，T 把输入符号串 w 变成 Γ^* 中的符号串 α，并进入 F 中的某个结束状态 q. 不要求 q 是停机状态. 因此，被 T 接受的语言 L 可表示为

$$L=\{w|w\text{ 在 }\Sigma^*\text{ 中，并且对于 }F\text{ 中的某个 }q,\ \Gamma^*\text{ 中的 }\alpha\text{ 以及整数 }i,\text{ 有 }(q_1,\ w,\ 1)\overset{*}{\underset{T}{\vdash}}(q,\ \alpha,\ i)\}$$

下面给出接受上下文无关语言 $L=\{a^nb^n|n\geqslant 1\}$ 的图灵机 T_0：

$$T_0=(K,\ \Sigma,\ \Gamma,\ \delta,\ q,\ F)$$

其中

$$K=\{q,\ q_1,\ q_2,\ q_3,\ q_4,\ q_5\},\ \Sigma=\{a,\ b\}$$

$$\Gamma=\{a,\ b,\ B,\ X,\ Y\},\ F=\{q_5\},\quad \text{开始状态为 } q$$

映射 δ 有如下几条：

$$qa|Xq_1R,\ q_1a|aq_1R,\ q_1Y|Yq_1R$$

$$q_1b|Yq_2L,\ q_2Y|Yq_2L,\ q_2X|Xq_3R$$

$$q_2a|aq_4L,\ q_4a|aq_4L,\ q_4X|XqR$$

$$q_3Y|Yq_3R,\ q_3B|Yq_5R$$

这里为简便起见，把 $\delta(q,\ a)=(p,\ A,\ D)$ 写成 $qa|APD$ 的形式. 例如 T 接收字 $aabb$ 的过程如下：

$$\overset{q}{a}abb\to X\overset{q_1}{a}bb\to Xa\overset{q_1}{b}b\to Xa\overset{q_2}{Y}b\to X\overset{q_4}{a}Yb\to$$

$$X\overset{q}{a}Yb\to XX\overset{q_1}{Y}b\to XXY\overset{q_1}{b}\to X\overset{q_2}{X}YY\to$$

$$X\overset{q_2}{X}YY\to XX\overset{q_3}{Y}Y\to XXY\overset{q_3}{Y}\to XXYY\overset{q_3}{B}\to$$

$$XXYYY\overset{q_5}{B}$$

这个过程的要点如下：

1) T 处于状态 q，扫描到 a 时，将 a 换成 X，然后转到状态 q_1.