有限自动机的可逆性

陶 仁 骥 著

科 学 出 版 社

有限自动机的可逆性

陶仁骥 著

科学出版社

内 容 简 介

有限自动机是存贮量有限的离散数字系统(例如数字电路)的抽象数学模型. 具有可逆性质的有限自动机，由其输出序列(和附加信息，如初始状态)可决定其输入序列.

本书围绕有限自动机的可逆性这一主题进行讨论. 第一章介绍有限自动机的基本概念. 第二章讨论有限自动机的可逆性的基本问题. 第三章讨论自治有限自动机的输出序列.

本书可供数字通信、计算机和数学专业的科技工作者、大学教师和研究生参考.

有限自动机的可逆性

陶仁骥 著

*

科学出版社出版

北京朝阳门内大街137号

石家庄地区印刷厂印刷

新华书店北京发行所发行　各地新华书店经售

*

1979年 8 月第 一 版　开本：787×1092 1/16

1979年 8 月第一次印刷　印张：19

印数：0001—15,120　字数：446,000

统一书号：15031·225

本社书号：1360•15—8

定 价: 1.95元

前　言

自动机理论是研究离散数字系统的功能、结构及其两者关系的数学理论. 数字电路和数学中的算法,就是离散数字系统的两个典型例子. 一个具体的数字电路,无论它多么复杂,其存贮量(记忆元件个数)总是有限的. 但是,一个算法执行起来所需要的存贮量,则往往是潜在无穷的. 有限自动机(又称时序机、时序电路或时序网络)这个数学概念,是用来描述那些存贮量有限的离散数字系统,作为它们的抽象数学模型的.

在三十年代,命题演算(布尔代数)应用于继电器结点网络的分析和综合,形成了开关电路理论. 有限自动机理论是开关电路理论的自然发展. 在五十年代初,形成了有限自动机的数学概念. 二十多年来,受电子计算机等新技术的推动,有限自动机理论已经有了很大的发展.

由于数字通信技术的发展,要求对有限自动机的可逆性问题进行研究. 本书围绕着有限自动机的可逆性这一主题进行讨论. 第一章介绍有限自动机的基本概念. 第二章讨论有限自动机的可逆性问题,侧重点在延迟 τ 步可逆(或弱可逆,或前馈可逆)有限自动机和延迟 τ 步逆(或弱逆,或前馈逆)有限自动机的结构. 第三章讨论自治有限自动机(包括自治移位寄存器)的输出序列,主要是线性的情形. 序列问题的研究和弱可逆密切相关.

本书中使用了代数方面的许多术语和结果. 属于线性代数方面的,请读者参阅«矩阵论»上册(Ф. Р. 甘特马赫尔著,柯召译,高等教育出版社,1955)中有关部分. 属于有限域方面的,请参阅本书附录 I. 本书中所使用的图论方面的术语,都已在脚注中加以说明了,读者也可参阅«图的理论及其应用»(C. 贝尔热著,李修睦译,上海科学技术出版社,1963)一书.

在本书写作过程中,得到黄祖良等同志的宝贵支持和帮助,在此谨致谢意!

由于作者的水平有限,书中一定还存在许多不完善的地方和缺点错误,请读者批评指正.

陶仁骥

1976年2月

目　　录

第一章 绪　论

1.1 有限自动机的逻辑网络实现

有限自动机是一种数学动态系统，其示意图如图 1.1. 有限自动机具有下述特点：第一，系统的输入、输出和表征系统特征的变量都只取有限种值；第二，时间坐标系统是离散的，由某一个严格上升的非负实数无穷序列 $t_0, t_1, \cdots$ 来规定，只考虑系统的有关变量在这些时刻上的值；第三，在 t_i 时刻系统的输入值和表征系统特征的变量的值完全决定了 t_i 时刻系统的输出值和 t_{i+1} 时刻表征系统特征的变量的值，$i=0, 1, \cdots$. 有限自动机用于描述许多具体的离散数字系统，作为它们的抽象数学模型. 数字电路就是一种离散数字系统，这在数字通信、自动控制和计算技术等工程领域中经常遇到.

图 1.1

下面给出有限自动机的严格数学定义.

设 X, Y 和 S 是三个非空有限集，δ 是笛卡儿积 $S\times X$ 到 S 的单值映射[1]，λ 是 $S\times X$ 到 Y 的单值映射，则称系统 $\langle X, Y, S, \delta, \lambda\rangle$ 为一个有限自动机.

记 $\langle X, Y, S, \delta, \lambda\rangle$ 为 M. 我们称 X 为 M 的输入字母表，Y 为 M 的输出字母表，S 为 M 的状态字母表，δ 为 M 的下一状态函数，λ 为 M 的输出函数，称 X, Y 和 S 中元素分别为 M 的输入、M 的输出和 M 的状态.

给定一个离散时间坐标系统 $t_0, t_1, \cdots$（本书均采用此时间坐标系统）.

记 t_i 时刻 M 的输入值为 $x(i)$，M 的输出值为 $y(i)$，M 的状态值为 $s(i)$，$i=0, 1, \cdots$. 作为一个动态系统，M 的工作方式由下述方程组规定：

$$\begin{aligned} s(i+1) &= \delta(s(i), x(i)) \\ y(i) &= \lambda(s(i), x(i)) \\ i &= 0, 1, \cdots \end{aligned} \tag{1.1}$$

这种工作方式通常又称为同步工作方式；非同步即异步工作方式不在本书讨论之列. 很容易看出，一旦给定 t_0 时刻 M 的状态 $s(0)$，此 $s(0)$ 称为 M 的初始状态，则 M 的输出序列 $y(0), \cdots, y(i)$ 由等长的输入序列 $x(0), \cdots, x(i)$ 唯一决定，从而输出无穷序列 $y(0), y(1), \cdots, y(i), \cdots$ 由输入无穷序列 $x(0), x(1), \cdots, x(i), \cdots$ 唯一决定. 设 α 为序列 $a_0, \cdots, a_i$，β 为序列 $b_0, b_1, \cdots$，则称序列 $a_0, \cdots, a_i, b_0, b_1, \cdots$ 为 α 与 β 的并，记作 $\alpha\beta$. 以 $\emptyset$ 表示空序列即长为 0 的序列. 约定并序列 $\alpha\emptyset=\alpha$, $\emptyset\beta=\beta$, $\emptyset\emptyset=\emptyset$.（由于引入了序列的并这一概念，我们可将序列 $a_0, a_1, \cdots, a_i$ 表示为 $a_0a_1\cdots a_i$. 为方便起见，也将无穷序列 $a_0, a_1, \cdots$ 表示为 $a_0a_1\cdots$. 有时又将序列用括弧括起来.）将函数 δ 和 λ 的定义域从输入扩充到输入序列：

---

1) 设 $S_1, \cdots, S_n$ 是 n 个非空集合，$n\geq 0$，则我们称集合 $\{[s_1, \cdots, s_n] | s_i\in S_i, i=1, \cdots, n\}$ 为 $S_1, \cdots, S_n$ 的笛卡儿积，并记作 $S_1\times\cdots\times S_n$ 或 $\prod_{i=1}^{n} S_i$，当 $S_1=\cdots=S_n=S$ 时又记作 S^n.

$$\delta(s, \emptyset)=s \quad \delta(s, \alpha\beta)=\delta(\delta(s, \alpha), \beta)$$

$$\lambda(s, \emptyset)=\emptyset \quad \lambda(s, \alpha\gamma)=\lambda(s, \alpha)\lambda(\delta(s, \alpha), \gamma) \tag{1.2}$$

上式中 $s\in S$, α 和 β 为有限长输入序列, γ 为有限或无穷长输入序列. 显然, 当 M 的初始状态为 s 时, 输入序列 α 决定的输出序列为 $\lambda(s, \alpha)$.

有限自动机 $M=\langle X, Y, S, \delta, \lambda\rangle$ 可用一个带赋值的有向图给出[1]. 这个有向图以 M 的状态为结点, 任何 $s\in S$ 和 $x\in X$, 从结点 s 到结点 $\delta(s, x)$ 有一条弧, 带有赋值 $\lambda(s, x)/x$. 这个有向图称为 M 的状态图.

例 1 串行二进制加法器. 设 $X=\left\{\begin{bmatrix} i \\ j \end{bmatrix} \middle| i, j=0,1\right\}$, $Y=S=\{0, 1\}$. 又设

$$\delta\left(s, \begin{bmatrix} a \\ b \end{bmatrix}\right)=\begin{cases} 1 & \text{当 } s, a, b \text{ 中至少有两个为 } 1 \\ 0 & \text{其它} \end{cases}$$

$$\lambda\left(s, \begin{bmatrix} a \\ b \end{bmatrix}\right)=\begin{cases} 1 & \text{当 } s, a, b \text{ 中 } 1 \text{ 的个数为奇数} \\ 0 & \text{其它} \end{cases}$$

则 $M=\langle X, Y, S, \delta, \lambda\rangle$ 是一个有限自动机. 很容易验证, 当初始状态为 s_0 且输入序列为 $\alpha=\left(\begin{bmatrix} a_0 \\ b_0 \end{bmatrix}, \cdots, \begin{bmatrix} a_i \\ b_i \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \end{bmatrix}\right)$时, 输出序列 $\lambda(s, \alpha)=[y_0, \cdots, y_{i+1}]$ 由下式决定:

$$\sum_{j=0}^{i+1} y_j 2^j=\sum_{j=0}^{i+1} a_j 2^j+\sum_{j=0}^{i+1} b_j 2^j+s_0$$

M 的状态图如图 1.2 所示.

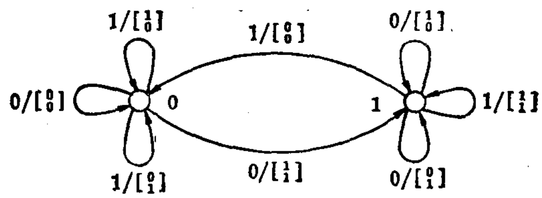


图 1.2

例 2 q 元 n 级移位寄存器, $q\geqslant 2$. 任给集合 F_q, 它共有 q 个元素. 记[2]

$$R_k=\{a \mid a=[a_1, \cdots, a_k]^T, a_1, \cdots, a_k\in F_q\}.$$

设 $X=R_l$, $Y=R_m$, $S=R_n$, $l\geqslant 0$, $m>0$, $n\geqslant 0$. 设

$$\delta\left(\begin{bmatrix} s_1 \\ \vdots \\ s_n \end{bmatrix}, \begin{bmatrix} x_1 \\ \vdots \\ x_l \end{bmatrix}\right)=\begin{bmatrix} s_2 \\ \vdots \\ s_n \\ \delta_n(s_1, \cdots, s_n, x_1, \cdots, x_l) \end{bmatrix}$$

---

1) 设 S 和 U 是两个集合, 且 U 中任一元素 u 都对应 $S\times S$ 的一个元素, 则称 (S, U) 是一个有向图. S 中的元素称为图 (S, U) 的结点, U 中元素称为图 (S, U) 的弧, 且当 $u\in U$ 对应 $S\times S$ 的元素为 $[x, y]$ 时, 称 x 为弧 u 的起点, y 为弧 u 的终点, 并称 u 为从结点 x(发出并进入)到结点 y 的一条弧. 如有可能, 我们总是将 S 中元素用平面上的点表示出来, 将 U 中元素 u 用一个箭头表示, 箭头的起点为弧 u 的起点, 箭头的终点为弧 u 的终点. 将有向图 G 的弧的起点和终点交换后所得的有向图称为 G 的反图. 在有向图的一部分或全部结点或弧上, 可带有某种赋值(图的一个结点或一条弧的赋值可以取任一集合中的任一元素), 称这种有向图为带赋值有向图. 当 S 和 U 都有限时, 称有向图 (S, U) 为有限的. 当 S 和 U 都为空集时, 称 (S, U) 为空有向图或空图.

2) 以 A^T 表示矩阵 A 的转置矩阵. 行(列)向量的转置为列(行)向量.

$$\lambda\left(\begin{bmatrix} s_1 \\ \vdots \\ s_n \end{bmatrix}, \begin{bmatrix} x_1 \\ \vdots \\ x_l \end{bmatrix}\right)=\begin{bmatrix} \lambda_1(s_1, \cdots, s_n, x_1, \cdots, x_l) \\ \vdots \\ \lambda_m(s_1, \cdots, s_n, x_1, \cdots, x_l) \end{bmatrix}$$

其中 δ_n, λ_1, $\cdots$, λ_m 都为 F_q 上 $n+l$ 元函数，则 $M=\langle X, Y, S, \delta, \lambda\rangle$为有限自动机，称它为 q 元 n 级移位寄存器.

逻 辑 网 络

给定集合 F_q，它的元素个数为 $q\geqslant 2$. 仍以 $x(i)$, $y(i)$, $\cdots$ 分别表示变量 x, y, $\cdots$ 在 t_i 时刻的值.

任何一个系统，其输入端为 x，输出端为 y，若它的输入、输出都在 F_q 中取值，且满足方程组

$$y(i+1)=x(i) \quad i=0, 1, \cdots \tag{1.3}$$

则称这个系统为 q 值记忆元件，并用图 1.3 的图形表示. 方程组(1.3)称为 q 值记忆元件的功能运算方程组.

任何一个系统，其输入端为 x_1, $\cdots$, x_k，输出端为 y，若它的输入、输出都在 F_q 中取值，且满足方程组

$$y(i)=f(x_1(i), \cdots, x_k(i)) \quad i=0, 1, \cdots \tag{1.4}$$

其中 f 是一个 F_q 上 k 元函数，则称这个系统为功能函数为 f 的 q 值逻辑元件，简称为 f 门，并用图 1.4 中的图形表示. 方程组(1.4)称为 f 门的功能运算方程组. 请注意，上述 k 可以为 0，这时 f 门没有输入端且输出取常值，其图形如图 1.5 所示.

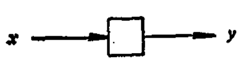


图 1.3

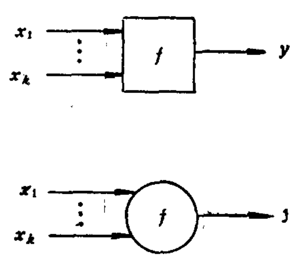


图 1.4

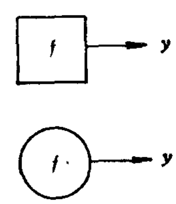


图 1.5

任何一个图形，它由有限个 q 值逻辑元件或 q 值记忆元件所组成，元件的输入端和输出端之间按照某一种方式互相连接在一起. 若这种连接中没有不同元件的输出端相连接，并且在输入端和输出端中，任二端连接在一起当且仅当它们的变元标号相同，则称此图形为一个 q 值逻辑网络.

设 N 是一个 q 值逻辑网络. N 的每一个元件都有一个功能运算方程组，所有这些功能运算方程组的联立方程组称为 N 的结构方程组，用记号 $E(N)$ 表示. 容易证明，N 和 $E(N)$是互相唯一决定的.

逻辑网络 N 的变元中，那些不是逻辑元件或记忆元件的输出的变元，称为 N(或 $E(N)$) 的输入变元；所有记忆元件的输出变元称为 N (或$E(N)$)的状态变元. N(或 $E(N)$) 的输出变元可由人们随意指定为变元的某一部分或全部.

由逻辑网络 N 唯一决定一个有限有向图，它的部分结点具有赋值. 记这个图为

$G(N)$. $G(N)$ 的结点为 N 的变元. 若 x 是 N 的输入变元,则 $G(N)$ 的结点 x 无弧进入,结点 x 也没有赋值; 若 x_0 是 N 的非输入变元, N 中以 x_0 为输出的元件的功能运算方程组为 $\mathrm{eq}(x_0)$,其输入变元为 $x_1, \cdots, x_k$, $k \geqslant 0$($x_0, \cdots, x_k$ 中可能有相同者),则以 $\mathrm{eq}(x_0)$ 为 $G(N)$ 中结点 x_0 的赋值, 且对应每一 x_i, 在 $G(N)$ 中从结点 x_i 到结点 x_0 有一条弧, $i=1, \cdots, k$(结点 x_0 共有 k 条弧进入). 直观地说,将 N 中相同标号的输入输出端在同一个结点上相连, 再将 N 中的元件缩小然后并进元件输出端所连的结点, 并且以该元件的功能运算方程组为该结点赋值则得到 $G(N)$.

所谓有限有向图 G 是 F_q 上赋值合式的, 就是说 G 的部分结点具有赋值且适合下述条件: 对任何结点 y,若进入结点 y 的弧共有 k 条, $k \geqslant 0$, 且这 k 条弧的起点为 $x_1, \cdots, x_k$, 则当 $k>0$ 或 y 是一孤立结点[1]时结点 y 有赋值, 又当结点 y 有赋值时其赋值形式为 $y(i)=f(x_1(i), \cdots, x_k(i))$, $i=0, 1, \cdots$ (当 $k \geqslant 0$) 或 $y(i+1)=x_1(i)$, $i=0, 1, \cdots$ (当 $k=1$), 其中 f 为 F_q 上 k 元函数. 容易证明: 对任何 q 值逻辑网络 N, $G(N)$ 是 F_q 上赋值合式的; 反之, 任何 F_q 上赋值合式的有向图 G, 都唯一存在 q 值逻辑网络 N 使得 $G(N)=G$. 因此,从图论的观点看来, q 值逻辑网络就是 F_q 上赋值合式的有向图.

设 N 是一 q 值逻辑网络. 若 $G(N)$ 去掉全部进入状态结点的弧后所得的部分图无回路[2],则称 N 为 q 值合式网络. 不包含记忆元件的 q 值合式网络称为 q 值组合网络.

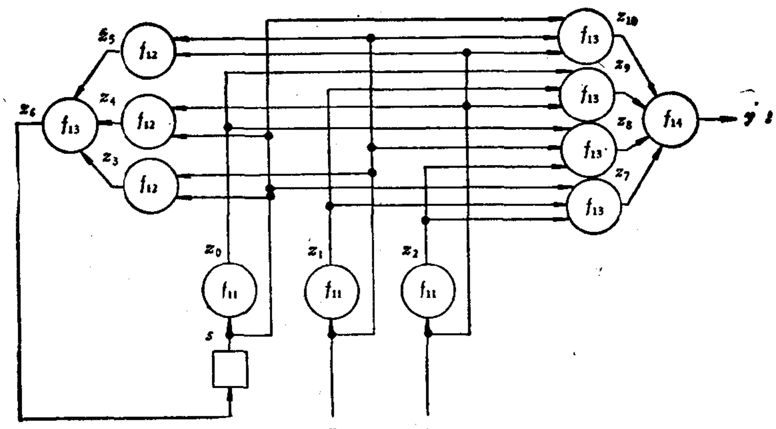
若有向图 $G(N)$ 中结点 y 无层次[2],则称 N 中变元 y 无层次; 若 $G(N)$ 中结点 y 有层次 i,则称 N 中变元 y 有层次且层次为 i. 若 $G(N)$ 能分层, 则称 N 能分层或有层次,且称 $G(N)$ 的层次为 N 的层次. 显然, N 中变元的最大层次为 N 的层次. 因为有向图无回路当且仅当它有层次,故得: q 值逻辑网络 N 是 q 值合式网络的充分必要条件为 N 有层次; 从而 q 值逻辑网络 N 是 q 值组合网络的充分必要条件为 N 有层次且 N 中无记忆元件.

例 3 设 $F_2=\{0, 1\}$. 设 F_2 上函数 $f_{1n}(x_1, \cdots, x_n)$ 在 $x_1=\cdots=x_n=1$ 处的值为 0, 在其它处的值为 1, 习惯上称 f_{1n} 门为 n 端与非门. 图 1.6 中给出了一个 2 值逻辑网络 N,其输入变元为 x_1, x_2, 状态变元为 s, 输出变元为 y. 图 1.7 中画出了与 N 对应的有向图 $G(N)$, $G(N)$ 中有赋值的结点及其赋值如下.

$$s\text{:}\quad s(i+1)=z_6(i),\ i=0, 1, \cdots;$$
$$z_0\text{:}\quad z_0(i)=f_{11}(s(i)),\ i=0, 1, \cdots;$$
$$z_1\text{:}\quad z_1(i)=f_{11}(x_1(i)),\ i=0, 1, \cdots;$$
$$z_2\text{:}\quad z_2(i)=f_{11}(x_2(i)),\ i=0, 1, \cdots;$$
$$z_3\text{:}\quad z_3(i)=f_{12}(s(i), x_1(i)),\ i=0, 1, \cdots;$$
$$z_4\text{:}\quad z_4(i)=f_{12}(s(i), x_2(i)),\ i=0, 1, \cdots;$$

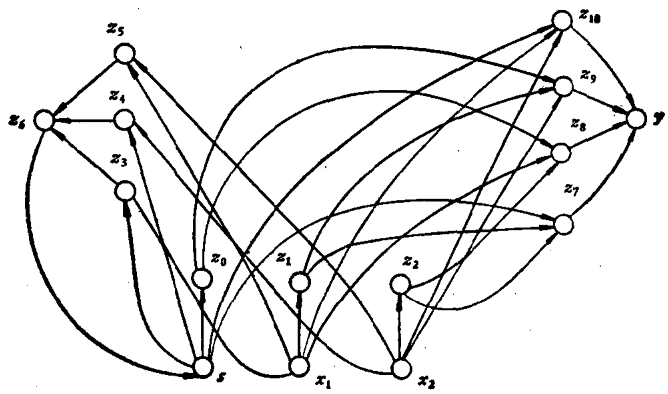
1) 有向图 G 中既无弧进入又无弧发出的结点, 称为 G 的孤立结点.

2) 设 $G=(S, U)$ 是一个有向图, 若 $U' \subseteq U$,则称有向图 (S, U') 为 G 的部分图. 设 $u_1, u_2, \cdots$ 是有向图 G 的弧的序列,若 u_i 的终点和 u_{i+1} 的起点相同, $i=1, 2, \cdots$, 则称 $u_1, u_2, \cdots$ 为 G 中的一条路,并称弧 u_1 的起点为这条路的起点. 若路只有 l 条弧,则称 l 为它的路长,并称弧 u_l 的终点为路的终点. 若一条路的起点和终点相同,则称它是一条回路. 为方便起见, 我们也说有一条起点和终点为 s 的长 0 的路(但不看作回路). 定义 G 中结点的层次如下: 若结点 s 无弧进入,则定义 s 的层次为 0; 设结点 s 有弧进入,进入 s 的弧的起点为 $s_1, s_2, \cdots$,若 $s_1, s_2, \cdots$ 皆已定义了层次,且这些层次的最大值为 i,则定义 s 的层次为 $i+1$. 若 G 的结点皆可定义层次,则称有向图 G 能分层或有层次,并称 G 中结点的最大层次为 G 的层次. 约定空图的层次为 -1. 容易证明, G 有层次的充分必要条件为 G 无回路.



N

图 1.6



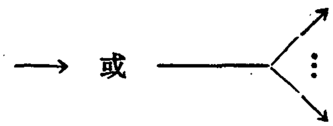
$G(N)$

图 1.7

$$z_5:\quad z_5(i)=f_{12}(x_1(i),\ x_2(i)),\ i=0,\ 1,\ \cdots;$$
$$z_6:\quad z_6(i)=f_{13}(z_3(i),\ z_4(i),\ z_5(i)),\ i=0,\ 1,\ \cdots;$$
$$z_7:\quad z_7(i)=f_{13}(s(i),\ z_1(i),\ z_2(i)),\ i=0,\ 1,\ \cdots;$$
$$z_8:\quad z_8(i)=f_{13}(z_0(i),\ x_1(i),\ z_2(i)),\ i=0,\ 1,\ \cdots;$$
$$z_9:\quad z_9(i)=f_{13}(z_0(i),\ z_1(i),\ x_2(i)),\ i=0,\ 1,\ \cdots;$$
$$z_{10}:\quad z_{10}(i)=f_{13}(s(i),\ x_1(i),\ x_2(i)),\ i=0,\ 1,\ \cdots;$$
$$y:\quad y(i)=f_{14}(z_7(i),\ z_8(i),\ z_9(i),\ z_{10}(i)),\ i=0,\ 1,\ \cdots.$$

很容易验证，$G(N)$ 去掉从 z_6 到 s 的弧所得的有向图无回路. 所以 N 是一个 2 值合式网络. 很容易定出 $G(N)$ 的结点或 N 的变元的层次如下：s, x_1, x_2 为 0 层；z_0, z_1, z_2, z_3, z_4, z_5 为 1 层；z_6, z_7, z_8, z_9, z_{10} 为 2 层；y 为 3 层. 故 $G(N)$ 确有层次.

因为含有逻辑元件的 q 值合式网络去掉记忆元件后所得的 q 值逻辑网络是 q 值组合网络，所以 q 值合式网络可以画成图 1.8 的形式. 这里，约定图 1.8 中 q 值组合网络的部分除包含前面严格定义的一个 q 值组合网络外，还可以包含若干条孤立的线，它们的形状为



每当合式网络中存在着不是逻辑元件的输入的状态变元或输入变元，就出现这种孤立线．当合式网络只有记忆元件时，图 1.8 中 q 值组合网络部分只包含这种孤立线．

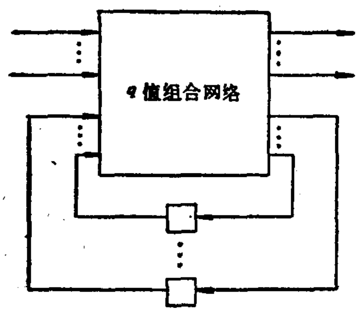


图 1.8

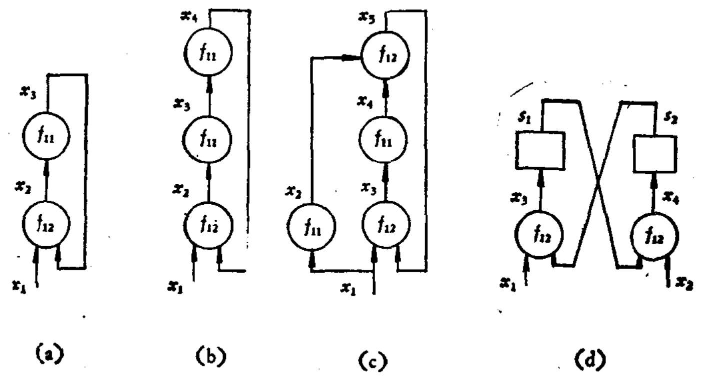


图 1.9

设 N 是一个 q 值逻辑网络，它的输入变元为 $x_1, \cdots, x_l$，状态变元为 $s_1, \cdots, s_n$．若对任意 $a_j(i)\in F_q$，$j=1, \cdots, l$，$i=0, 1, \cdots$，和 $b_j\in F_q$，$j=1, \cdots, n$，方程组

$$\left.\begin{array}{l} E(N) \\ s_j(0)=b_j \quad j=1, \cdots, n \\ x_j(i)=a_j(i) \quad j=1, \cdots, l \quad i=0, 1, \cdots \end{array}\right\} \tag{1.5}$$

都有唯一解，则称 N 为 q 值良性网络．例如，在图 1.9 中，(a) 和 (b) 都不是 2 值良性网络，因为对应于它们的方程组 (1.5) 或者不是唯一解或者无解，但是 (c) 和 (d) 是 2 值良性网络．

设 N 是 q 值合式网络，其输入变元为 $x_1, \cdots, x_l$，状态变元为 $x_{l+1}, \cdots, x_{l+n}$，其它变元为 $x_{l+n+1}, \cdots, x_k$．将 N 的结构方程组 $E(N)$ 分为两部分，第一部分为记忆元件的功能运算方程组

$$x_j(i+1)=x_{r_j}(i) \quad i=0, 1, \cdots \quad j=l+1, \cdots, l+n \tag{1.6}$$

记作 $E_1(N)$，第二部分为逻辑元件的功能运算方程组

$$x_j(i)=f_j(x_{j_1}(i), \cdots, x_{j_{r(j)}}(i)) \quad i=0, 1, \cdots \quad j=l+n+1, \cdots, k \tag{1.7}$$

记作 $E_2(N)$．由于 N 是合式网络，故变元 x_{j_r} 的层次小于变元 x_j 的层次，$j=l+n+1, \cdots, k$，$r=1, \cdots, r(j)$．我们按变元的层次递归地定义函数 $f^{(j)}(x_1, \cdots, x_{l+n})$，$j=1, \cdots, k$，

$$\left.\begin{array}{l} f^{(j)}(x_1, \cdots, x_{l+n})=x_j \quad j=1, \cdots, l+n \\ f^{(j)}(x_1, \cdots, x_{l+n})=f_j(f^{(j_1)}(x_1, \cdots, x_{l+n}), \cdots, f^{(j_{r(j)})}(x_1, \cdots, x_{l+n})) \\ \qquad\qquad j=l+n+1, \cdots, k \end{array}\right\} \tag{1.8}$$

并称 $f^{(j)}(x_1, \cdots, x_{l+n})$ 为变元 x_j 的功能函数，$j=1, \cdots, k$；称 $f^{(r_j)}(x_1, \cdots, x_{l+n})$ 为变元 x_j 的激励函数，$j=l+1, \cdots, l+n$．设 N 的层次为 ρ．对任何 $0\leqslant\tau\leqslant\rho$，记方程组

$$x_j(i)=f_j(x_{j_1}(i),\ \cdots,\ x_{j_{r(j)}}(i))\quad i=0,\ 1,\ \cdots$$
$$\rho\geqslant x_j \text{ 的层次}>\tau$$
$$x_j(i)=f^{(j)}(x_1(i),\ \cdots,\ x_{l+n}(i))\quad i=0,\ 1,\ \cdots$$
$$\tau\geqslant x_j \text{ 的层次}>0$$

为 $E_2^{\tau}(N)$，$\tau=0,\ \cdots,\ \rho$. 容易证明，方程组 $E_2^{\tau}(N)$ 和方程组 $E_2^{(\tau+1)}(N)$ 等价，$\tau=0,\ \cdots,\ \rho-1$. 因为 $E_2^0(N)$ 就是 $E_2(N)$，故方程组 $E_2(N)$ 和方程组 $E_2^{\rho}(N)$ 等价. 因此，$E_1(N)$ 与 $E_2^{\rho}(N)$ 的联立方程组和方程组 $E(N)$ 等价. 故方程组 $E(N)$ 等价于方程组

$$\left.\begin{aligned} x_j(i+1)&=f^{(r_j)}(x_1(i),\ \cdots,\ x_{l+n}(i))\quad i=0,\ 1,\ \cdots\quad j=l+1,\ \cdots,\ l+n\\ x_j(i)&=f^{(j)}(x_1(i),\ \cdots,\ x_{l+n}(i))\quad i=0,\ 1,\ \cdots\quad j=l+n+1,\ \cdots,\ k\end{aligned}\right\}\tag{1.9}$$

因此，方程组(1.5)有唯一解. 所以，N 是 q 值良性网络. 这就证明了，q 值合式网络是 q 值良性网络.

设合式网络 N 的输出变元为 $x_{p_1},\ \cdots,\ x_{p_m}$. 由方程组(1.9)得

$$\left.\begin{aligned} x_j(i+1)&=f^{(r_j)}(x_1(i),\ \cdots,\ x_{l+n}(i))\quad i=0,\ 1,\ \cdots\quad j=l+1,\ \cdots,\ l+n\\ x_{p_j}(i)&=f^{(p_j)}(x_1(i),\ \cdots,\ x_{l+n}(i))\quad i=0,\ 1,\ \cdots\quad j=1,\ \cdots,\ m\end{aligned}\right\}\tag{1.10}$$

取 X 为 F_q^l，Y 为 F_q^m，S 为 F_q^n，定义 $S\times X$ 到 S 的映射 δ 和 $S\times X$ 到 Y 的映射 λ 为

$$\left.\begin{aligned}\delta\left(\begin{bmatrix}x_{l+1}\\ \vdots\\ x_{l+n}\end{bmatrix},\begin{bmatrix}x_1\\ \vdots\\ x_l\end{bmatrix}\right)&=\begin{bmatrix}f^{(r_{l+1})}(x_1,\ \cdots,\ x_{l+n})\\ \vdots\\ f^{(r_{l+n})}(x_1,\ \cdots,\ x_{l+n})\end{bmatrix}\\ \lambda\left(\begin{bmatrix}x_{l+1}\\ \vdots\\ x_{l+n}\end{bmatrix},\begin{bmatrix}x_1\\ \vdots\\ x_l\end{bmatrix}\right)&=\begin{bmatrix}f^{(p_1)}(x_1,\ \cdots,\ x_{l+n})\\ \vdots\\ f^{(p_m)}(x_1,\ \cdots,\ x_{l+n})\end{bmatrix}\end{aligned}\right\}\tag{1.11}$$

记有限自动机 $\langle X,\ Y,\ S,\ \delta,\ \lambda\rangle$ 为 $M(N)$. 显然，$M(N)$ 由 N 和输入、输出及状态变元的排列次序唯一决定. 比较(1.1)和(1.10)式，立刻知道(1.10)式为有限自动机 $M(N)$ 的工作方式方程组. 因此，N 和 $M(N)$ 的功能相同，于是我们说 q 值合式网络 N 是 F_q 上有限自动机 $M(N)$ 的一个实现，或者简单地说 N 是一个有限自动机.

对两个合式网络 N 和 N'，若它们的输入变元、状态变元和输出变元彼此相同，且状态变元的激励函数和输出变元的功能函数彼此相同，则称 N 和 N' 功能等价. 由定义可知，若 N 和 N' 功能等价，则 $M(N)$ 和 $M(N')$ 相同（假设变元的排列两者相同）. 这就是说，用有限自动机来研究合式网络的功能时，抛开了网络的结构细节. 从功能等价的合式网络中，求出一个按照某种标准来看是最好的一个合式网络的问题，自然是与有限自动机的实现相联系的一个重要问题. 状态赋值问题、组合网络的化简问题也属于这个范畴. 这些问题的研究必然涉及到逻辑网络的结构细节，不是本书讨论的主题.

对任何 F_q 上函数集 C，若每个 F_q 上函数皆可由 C 中函数经过变元代入和函数代入而得，则称 C 为完备的. 以 $\mathfrak{F}_C$ 表示 F_q 上函数的集合，且满足条件：(1) $C\subseteq\mathfrak{F}_C$；(2) 若 $f(x_1,\ \cdots,\ x_n),\ g(y_1,\ \cdots,\ y_m)\in\mathfrak{F}_C$，$n\geqslant1$，则 $f(x_1,\ \cdots,\ x_{i-1},\ g(y_1,\ \cdots,\ y_m),\ x_{i+1},\ \cdots,\ x_n)\in\mathfrak{F}_C$，$1\leqslant i\leqslant n$；(3) 若 $f(x_1,\ \cdots,\ x_n)\in\mathfrak{F}_C$，则 $f(x_1,\ \cdots,\ x_n)\big|_{x_i=y}\in\mathfrak{F}_C$[1]；(4) 仅由条件

1) 记号 $f(x_1,\ \cdots,\ x_n)\big|_{x_i=y}$ 表示将 $f(x_1,\ \cdots,\ x_n)$ 的 x_i 代入以 y 的结果. 一般地说，以 $f(x_1,\ \cdots,\ x_n)\Big|_{\substack{x_i=s_i\\ i\in I}}$ 表示将 $f(x_1,\ \cdots,\ x_n)$ 中的 x_i 代入以 s_i，$i\in I$，所得到的结果，I 是 $\{x_1,\ \cdots,\ x_n\}$ 的一个非空子集，s_i 是变元或函数；与此类似，f 的地方也可换为方程组或其它式子，记号所表示的涵义相同.

(1)至(3)定义的函数属于 $\mathfrak{F}_C$. 易知, C 完备的充分必要条件为 $\mathfrak{F}_C$ 是所有 F_q 上函数的集合. 由数理逻辑中命题演算的结果, 我们知道 $\{f_{12}\}$ 是 $F_2=\{0, 1\}$ 上函数的完备集. 令 F_2 上函数 $f_{2n}(x_1, \cdots, x_n)$ 在 $x_1=\cdots=x_n=0$ 处的值为 1, 其余各处为 0, 则 $\{f_{22}\}$ 也是完备的. 其它常用的 F_2 上函数完备集为 $\{x\wedge y, x\vee y, \bar{x}\}$, $\{x+y, x\cdot y, 1\}$, 其中 $x\wedge y$ 或 $x\cdot y$ 表示逻辑乘, $x\vee y$ 表示逻辑或, $x+y$ 表示逻辑不等价或模 2 加, $\bar{x}$ 表示逻辑非, 1 表示取值常为 1 的函数. 当 F_q 为有限域时, 函数集 $\{a, a\in F_q, x+y, x\cdot y\}$ 是完备的(参看附录 II).

考虑 q 值组合网络. 设 q 值组合网络 N 的输入变元为 $x_1, \cdots, x_n$, 输出变元为 z, 变元 z 的功能函数为 $f(x_1, \cdots, x_n)$. 设 y 是 $x_1, \cdots, x_n$ 中某一个变元或 y 是 N 的变元以外的一个新变元. 显然, 将 $E(N)$ 中变元 x_i 替换为变元 y 所得的方程组, 仍然为一个组合网络的结构方程组. 记这个组合网络为 N_1. 则 N_1 的图形为将 N 的图形中的变元 x_i 改记为变元 y, 并将变元相同的线连接在一起. 易知, N_1 中变元 z 的功能函数为 $f(x_1, \cdots, x_n)|x_i=y$. 又

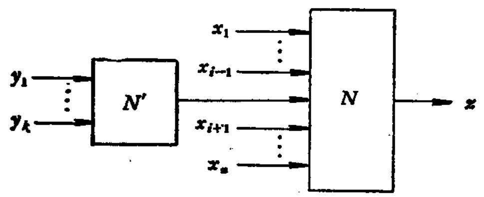


图 1.10

设 N' 也是一个 q 值组合网络, 其输入变元为 $y_1, \cdots, y_k$, 输出变元为 y, y 的功能函数为 $g(y_1, \cdots, y_k)$, 且 N' 和 N 的变元不同. 设 $n\geqslant 1$ 即 N 真有输入. 易知以 $E(N')$ 和 $E(N)|x_i=y$ 的联立方程组为结构方程组的逻辑网络也是 q 值组合网络, 且变元 z 的功能函数为 $f(x_1, \cdots, x_n)|x_i=g(y_1, \cdots, y_k)$; 我们称这个网络为 N 和 N' 的串联, 用图 1.10 表示.

设 C 是 F_q 上非空函数集. 根据上段的讨论, 易知任何 $\mathfrak{F}_C$ 中函数 $f(x_1, \cdots, x_l)$, 都存在逻辑元件的功能函数属于 C 的 q 值组合网络, 使得它的一个输出变元的功能函数为 $f(x_1, \cdots, x_l)$. 反之, 由(1.8)式知道, 逻辑元件的功能函数属于 C 的 q 值组合网络中, 任何变元的功能函数都属于 $\mathfrak{F}_C$.

设 C 是 F_q 上函数的完备集. 设有限自动机 $M=\langle X, Y, S, \delta, \lambda\rangle$ 且 X, Y, S 都是 F_q 的笛卡儿积. (这种有限自动机称为 F_q 上有限自动机.) 显然, 存在一个逻辑元件的功能函数属于 C 的 q 值组合网络实现 δ 和 λ, 即该组合网络的输出变元的功能函数分别为 δ 和 λ 的分量函数. 因此, 存在一个 q 值合式网络 N 使得 $M(N)=M$ 且 N 中逻辑元件的功能函数属于 C.

综上所述, 在实现的意义下, 逻辑元件的功能函数属于某一个 F_q 上函数的完备集的 q 值合式网络和 F_q 上有限自动机两者是一样的.

例 4 很容易验证, 例 3 中逻辑网络 N 是例 1 中有限自动机 M (二进制加法器) 的实现, 即 $M(N)=M$.

实现例 2 中 q 元 n 级移位寄存器的 q 值合式网络的框图, 可表示为图 1.11 的形式.

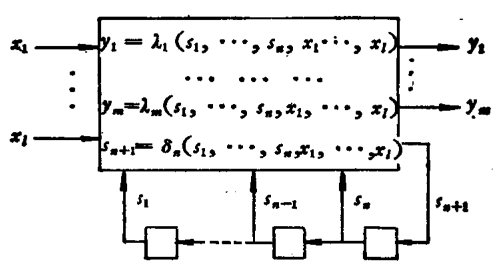


图 1.11

设有限自动机 $M_i=\langle X_i, Y_i, S_i, \delta_i, \lambda_i\rangle$, $i=1, 2$. 若存在 X_1 到 X_2 上的一一映射 φ_1, Y_1 到 Y_2 上的一一映射 φ_2 和 S_1 到 S_2 上的一一映射 φ_3, 使得任何 $s\in S_1$ 和 $x\in X_1$ 都有

$$\varphi_3(\delta_1(s, x)) = \delta_2(\varphi_3(s), \varphi_1(x))$$
$$\varphi_2(\lambda_1(s, x)) = \lambda_2(\varphi_3(s), \varphi_1(x))$$

则称 M_1 与 M_2 弱同构. 直观地说，弱同构有限自动机只是字母表中字母的记号不一样. 显然，弱同构关系是反身、对称和传递的.

若 $X_1 \subseteq X_2$, $Y_1 \subseteq Y_2$, $S_1 \subseteq S_2$, 且当 $s \in S_1$ 和 $x \in X_1$ 时 $\delta_1(s, x) = \delta_2(s_2, x)$ 和 $\lambda_1(s, x) = \lambda_2(s, x)$, 则称 M_1 是 M_2 的子有限自动机，又称 M_2 是 M_1 的扩有限自动机.

若 M_1 与 M_2 的某一个子有限自动机弱同构，则称 M_2 表示 M_1 或 M_1 可由 M_2 表示.

显然，子有限自动机和表示关系都是反身和传递的.

设 $q \geqslant 2$. 设有限自动机 $M = \langle X, Y, S, \delta, \lambda \rangle$, X 的字母个数为 a, Y 的字母个数为 b, S 的字母个数为 c. 取 l, m 和 n 为适合下述条件的最小非负整数，$a \leqslant q^l$, $b \leqslant q^m$, $c \leqslant q^n$. 令 $X' = F_q^l$, $Y' = F_q^m$, $S' = F_q^n$. 任取 X 到 X' 的一一映射 φ_1, Y 到 Y' 的一一映射 φ_2, S 到 S' 的一一映射 φ_3. 记 $X'' = \varphi_1(X)$, $Y'' = \varphi_2(Y)$, $S'' = \varphi_3(S)$. 这里，$\varphi(T)$ 表示集合 $\{\varphi(t) \mid t \in T\}$. 定义 $S'' \times X''$ 到 S'' 的映射 δ'' 和 $S'' \times X''$ 到 Y'' 的映射 λ'' 如下：

$$\left.\begin{aligned} \delta''(s, x) &= \varphi_3(\delta(\varphi_3^{-1}(s), \varphi_1^{-1}(x))) \\ \lambda''(s, x) &= \varphi_2(\lambda(\varphi_3^{-1}(s), \varphi_1^{-1}(x))) \\ s \in S'' & \quad x \in X'' \end{aligned}\right\} \tag{1.12}$$

将 δ'' 任意扩充为 $S' \times X'$ 到 S' 的单值映射，记作 δ'; 将 λ'' 任意扩充为 $S' \times X'$ 到 Y' 的单值映射，记作 λ'. 令有限自动机 $M' = \langle X', Y', S', \delta', \lambda' \rangle$. 很容易验证，$M'' = \langle X'', Y'', S'', \delta'', \lambda'' \rangle$ 是 M' 的子有限自动机且 M'' 与 M 弱同构. 所以，M 可由 M' 表示. 设 C 是任一个 F_q 上函数的完备集. 由于 M' 可由一个逻辑元件的功能函数属于 C 的 q 值合式网络实现，故存在逻辑元件的功能函数属于 C 的 q 值合式网络 N 使得 N 实现 M' 即 $M(N) = M'$. 故 M 可由 $M(N)$ 表示. 这时，我们也说 q 值合式网络 N 是有限自动机 M 的一个实现.（当 M 为 F_q 上有限自动机时，可取 φ_1, φ_2 和 φ_3 都为恒同映射. 这时，上面定义的 M' 与 M 相同. 因此，这里所说的 N 是 M 的一个实现，即 $M(N)$ 表示 M, 和前面所说的 N 是 M 的一个实现，即 $M(N) = M$, 两者是一致的.）这就证明了，任何 F_q 上函数的完备集 C, $q \geqslant 2$ 和有限自动机 M, 都存在逻辑元件的功能函数属于 C 的 q 值合式网络 N 为 M 的一个实现，即 $M(N)$ 表示 M.

设 $M = \langle X, Y, S, \delta, \lambda \rangle$. 若 $\delta(s, x)$ 和 $\lambda(s, x)$ 都不依赖 x, 则称 M 为自治的. 当 M 是自治有限自动机时，输入已不起作用，故可将 M 简记作 $\langle Y, S, \delta, \lambda \rangle$, 将 $\delta(s, x)$ 简记作 $\delta(s)$, 将 $\lambda(s, x)$ 简记作 $\lambda(s)$. 显然，当 X 只有一个字母时，M 是自治的.

对于自治有限自动机 M, 我们称初始状态为 s 时的无穷输出序列为 s 的输出序列，称 s 的输出序列的前 k 位为 s 的长 k 输出序列. 显然，s 的输出序列为 $\lambda(s)$, $\lambda(\delta(s))$, $\cdots$, $\lambda(\delta^i(s))$, $\cdots$, 其中 $\delta^{i+1}(s) = \delta(\delta^i(s))$. 我们也称无穷状态序列 s, $\delta(s)$, $\cdots$, $\delta^i(s)$, $\cdots$ 为 s 产生的状态序列.

设 X 和 Y 是非空有限集，f 是 $Y^k \times X^{h+1}$ 到 Y 的单值映射，k, $h \geqslant 0$. 若 M 由下式决定：

$$\left.\begin{aligned} y(i) = f(y(i-k), \cdots, y(i-1), x(i-h), \cdots, x(i)) \\ i = 0, 1, \cdots \end{aligned}\right\} \tag{1.13}$$

则称 M 为 $(h,\ k)$ 阶存贮有限自动机．详细地说，$M=\langle X,\ Y,\ S,\ \delta,\ \lambda\rangle$，其中

$$
\left.\begin{aligned}
&S=\left\{\begin{bmatrix} y_{-k}\\ \vdots\\ y_{-1}\\ x_{-h}\\ \vdots\\ x_{-1}\end{bmatrix}\middle|\, y_{-k},\ \cdots,\ y_{-1}\in Y,\ x_{-h},\ \cdots,\ x_{-1}\in X\right\}\\
&\delta\left(\begin{bmatrix} y_{-k}\\ \vdots\\ y_{-1}\\ x_{-h}\\ \vdots\\ x_{-1}\end{bmatrix},\ x\right)=\begin{bmatrix} y_{-k+1}\\ \vdots\\ y_{-1}\\ f(y_{-k},\ \cdots,\ y_{-1},\ x_{-h},\ \cdots,\ x_{-1},\ x)\\ x_{-h+1}\\ \vdots\\ x_{-1}\\ x\end{bmatrix}\\
&\lambda\left(\begin{bmatrix} y_{-k}\\ \vdots\\ y_{-1}\\ x_{-h}\\ \vdots\\ x_{-1}\end{bmatrix},\ x\right)=f(y_{-k},\ \cdots,\ y_{-1},\ x_{-h},\ \cdots,\ x_{-1},\ x)\\
&\qquad y_{-k},\ \cdots,\ y_{-1}\in Y\quad x_{-h},\ \cdots,\ x_{-1}\in X
\end{aligned}\right\}\tag{1.14}
$$

M 在时刻 t_i 的状态 $s(i)$ 为 $[y(i-k),\ \cdots,\ y(i-1),\ x(i-h),\ \cdots,\ x(i-1)]^T$．图 1.12 是 M 的示意图．

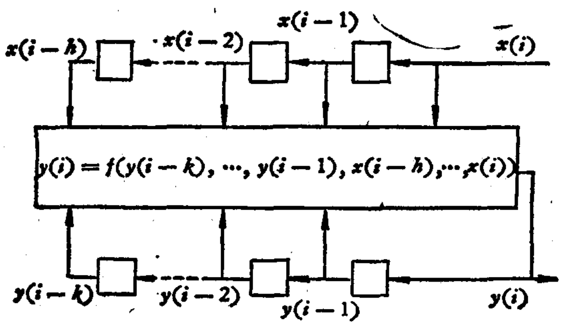


图 1.12

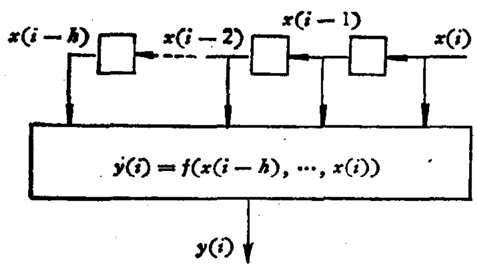


图 1.13

设 M' 也是一个 $(h',\ k')$ 阶存贮有限自动机，由

$$y(i)=f'(y(i-k'),\ \cdots,\ y(i-1),\ x(i-h'),\ \cdots,\ x(i))\quad i=0,\ 1,\ \cdots$$

定义．若 M' 和 M 的输入字母表和输出字母表彼此相同，且对任何 $y_{-j}\in Y$，$j=1,\ \cdots,\ \max(k,\ k')$ 和 $x_{-j}\in X$，$j=0,\ \cdots,\ \max(h,\ h')$，都有

$$f(y_{-k},\ \cdots,\ y_{-1},\ x_{-h},\ \cdots,\ x_{-1},\ x_0)=f'(y_{-k'},\ \cdots,\ y_{-1},\ x_{-h'},\ \cdots,\ x_{-1},\ x_0),$$

则称 M 和 M' 实质相同．显然，当 M 和 M' 实质相同时，它们的工程实现相同．

若 M 是 $(h,\ 0)$ 阶存贮有限自动机，则称 M 为 h 阶输入存贮有限自动机，其示意图如图 1.13 所示．

若 M 由下式决定:

$$\left.\begin{aligned} y(i)=f(y(i-k), \cdots, y(i-1), x(i-k-h), \cdots, x(i-k)) \\ i=k, k+1, \cdots \end{aligned}\right\} \tag{1.15}$$

$h, k \geqslant 0$, f 是 $Y^k \times X^{h+1}$ 到 Y 的单值映射，则称 M 为拟(h, k)阶存贮有限自动机. 详细地说，$M=\langle X, Y, S, \delta, \lambda\rangle$，其中

$$\left.\begin{aligned}
&S=\left\{\begin{bmatrix} y_0 \\ \vdots \\ y_{k-1} \\ x_{-h} \\ \vdots \\ x_{-1} \end{bmatrix} \middle| y_0, \cdots, y_{k-1} \in Y, x_{-h}, \cdots, x_{-1} \in X \right\} \\
&\delta\left(\begin{bmatrix} y_0 \\ \vdots \\ y_{k-1} \\ x_{-h} \\ \vdots \\ x_{-1} \end{bmatrix}, x\right)=\begin{bmatrix} y_1 \\ \vdots \\ y_{k-1} \\ f(y_0, \cdots, y_{k-1}, x_{-h}, \cdots, x_{-1}, x) \\ x_{-h+1} \\ \vdots \\ x_{-1} \\ x \end{bmatrix} \\
&\lambda\left(\begin{bmatrix} y_0 \\ \vdots \\ y_{k-1} \\ x_{-h} \\ \vdots \\ x_{-1} \end{bmatrix}, x\right)=y_0 \\
&\qquad y_0, \cdots, y_{k-1} \in Y \quad x_{-h}, \cdots, x_{-1}, x \in X
\end{aligned}\right\} \tag{1.16}$$

M 在时刻 t_i 的状态 $s(i)$ 为 $[y(i), \cdots, y(i+k-1), x(i-h), \cdots, x(i-1)]^T$. M 的示意图如图 1.14 所示.

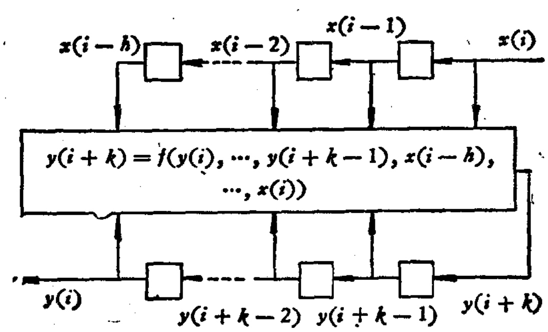


图 1.14

1.2 状 态 等 价

设 $M=\langle X, Y, S, \delta, \lambda\rangle$ 是一个有限自动机. 任何一个状态 s，都决定一个映射 λ_s，它将(有限或无穷长)输入序列 α 转换为同长的输出序列 $\lambda_s(\alpha)=\lambda(s, \alpha)$. 因此，给定一个

有限自动机 M 和它的一个状态 s，就给出了这样一个序列转换器．从序列转换的功能这一角度来看，怎样的 M 和 s 是无关紧要的，只要所决定的映射相同，对它们都可以不加区别地同等看待；自然，从其它的角度（例如结构）看，这些有限自动机可能是很不相同、决不能同等看待的．这样，在数学上很自然地就引出下述概念．

设 $M'=\langle X', Y', S', \delta', \lambda'\rangle$ 也是一个有限自动机．若 $X=X'$ 且对任何有限长输入序列 α 都有 $\lambda(s, \alpha)=\lambda'(s', \alpha)$，则称 M 的状态 s 与 M' 的状态 s' 等价，记作 $s\sim s'$，否则称 s 与 s' 不等价或可分．若对任何 $s\in S$ 都存在 $s'\in S'$ 使得 $s\sim s'$，则称 M' 强于 M 或 M 弱于 M'，记作 $M\prec M'$．若 $M\prec M'$ 且 $M'\prec M$，则称 M 与 M' 等价，记作 $M\sim M'$．

显然，状态的等价关系是反身、对称和传递的；有限自动机的强于关系是反身和传递的；有限自动机的等价关系是反身、对称和传递的．

很容易证明，若 $\lambda(s, \alpha)=\lambda'(s', \alpha)$ 对所有有限长输入序列 α 成立，则它对所有无穷输入序列 α 也成立；反之也对．因此，在状态等价的定义中，将有限长输入序列 α 改为无穷输入序列 α 也是可以的，从而当 $s\sim s'$ 时，映射 λ_s 与 $\lambda'_{s'}$ 相同．

显然，若 $s\in S$ 与 $s'\in S'$ 等价，则对任何 $x\in X$，$\delta(s, x)$ 与 $\delta'(s', x)$ 等价．

若有限自动机 M 的不同状态两两可分，则称 M 极小．

定理 1 设 M 是极小有限自动机．若有限自动机 M' 与 M 等价，则 M' 的状态数 n' 不小于 M 的状态数 n．

证明 假设 $n'<n$．因为 $M'\sim M$，故存在 S' 到 S 的单值映射 φ 使得任何 $s'\in S'$ 都有 $\varphi(s')\sim s'$．因为映射象 $\varphi(S')$ 的元素个数 $\leqslant n'<n$，故存在 S 中状态 $s\notin\varphi(S')$．因为 $M\sim M'$，故存在 S' 中状态 s' 与 s 等价．因为 $\varphi(s')\sim s'$ 和 $s'\sim s$，故得 $s\sim\varphi(s')$．因为 M 极小，故 $s=\varphi(s')$ 从而 $s\in\varphi(S')$，与 $s\notin\varphi(S')$ 矛盾．所以假设 $n'<n$ 不成立，即 $n\leqslant n'$．

定理 2 对任何有限自动机 M，都存在一个极小有限自动机 M_1 使得 $M\sim M_1$．

证明 我们由 $M=\langle X, Y, S, \delta, \lambda\rangle$ 构造有限自动机 $M_1=\langle X_1, Y_1, S_1, \delta_1, \lambda_1\rangle$ 如下．取 $X_1=X$，$Y_1=Y$．因为状态等价关系是 S 上的一个数学等价关系[1)]，所以按这个关系将 S 划分所得的划分块称为 M 的状态等价类．取 S_1 为 M 的所有状态等价类的集合．以 $\varphi(s)$ 表示包含 M 的状态 s 的状态等价类．令

$$\left.\begin{array}{c}\delta_1(\varphi(s), x)=\varphi(\delta(s, x)) \\ \lambda_1(\varphi(s), x)=\lambda(s, x) \\ s\in S \quad x\in X\end{array}\right\} \tag{1.17}$$

因为当 $s\sim s'$ 时，$\delta(s, x)\sim\delta(s', x)$，所以当 $s\sim s'$ 时，$\delta_1(\varphi(s), x)=\varphi(\delta(s, x))=\varphi(\delta(s', x))=\delta_1(\varphi(s'), x)$，$\lambda_1(\varphi(s), x)=\lambda(s, x)=\lambda(s', x)=\lambda_1(\varphi(s'), x)$．因此，由 (1.17) 式定义的 δ_1 和 λ_1 是单值的．所以 $M_1=\langle X_1, Y_1, S_1, \delta_1, \lambda_1\rangle$ 是一个有限自动机．

我们来证明，任何 $s\in S$ 都有 $\lambda(s, \alpha)=\lambda_1(\varphi(s), \alpha)$，$\alpha$ 是任一输入序列．对 α 的长度 $l(\alpha)$ 进行归纳．当 $l(\alpha)=0$ 时，$\alpha=\phi$．显然，$\lambda(s, \alpha)=\phi=\lambda_1(\varphi(s), \alpha)$．假设已证明命题

1) 设 $R(x, y)$（或 xRy）是集合 S 上的一个二元关系．如果关系 R 是反身、对称和传递的，则称 R 是一个数学等价关系．设 S_1, S_2, … 都是 S 的非空子集，如果它们的和集 $\bigcup S_i=S$ 且彼此不相交，即 $S_i\cap S_j=\phi$（空集），当 $i\neq j$，则称 S_1, S_2, … 为 S 的一个划分，称 S_i 为划分块，$i=1, 2, \cdots$．S 上的一个数学等价关系和 S 的一个划分相对应：任何 S 中元素 x 和 y，x 和 y 同属某一个划分块的充分必要条件为 $R(x, y)$．这个对应是一一对应．由 R 决定的划分的划分块又叫做等价类．