



有穷自动机

语言的识别

- 推导和归约中的回溯问题将对系统的效率产生极大的影响

$S \rightarrow aA | aB$

$A \rightarrow aA | c$

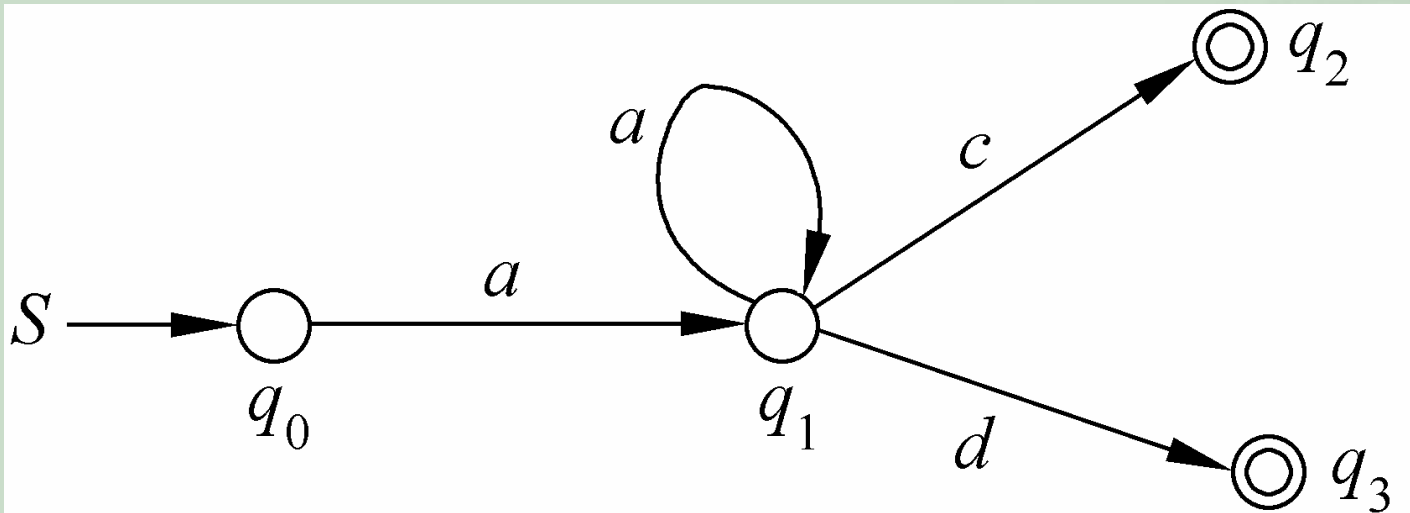
$B \rightarrow aB | d$

分析句子 **aaac** 的过程中可能需要回溯。



语言的识别

- 系统识别语言 $\{a^n c | n \geq 1\} \cup \{a^n d | n \geq 1\}$ 的字符串过程中状态的变化图示如下：



语言的识别

- 识别系统(模型)
- (1) 系统具有有穷个状态，不同的状态代表不同的意义。按照实际的需要，系统可以在不同的状态下完成规定的任务。
- (2) 我们可以将输入字符串中出现的字符汇集在一起构成一个字母表。系统处理的所有字符串都是这个字母表上的字符串。



语言的识别

- (3) 系统在任何一个状态(当前状态)下，从输入字符串中读入一个字符，根据当前状态和读入的这个字符转到新的状态。当前状态和新的状态可以是同一个状态，也可以是不同的状态；当系统从输入字符串中读入一个字符后，它下一次再读时，会读入下一个字符。这就是说，相当于系统维持有一个读写指针，该指针在系统读入一个字符后指向输入串的下一个字符。

语言的识别

- (4) 系统中有一个状态，它是系统的开始状态，系统在这个状态下开始进行某个给定句子的处理。
- (5) 系统中还有一些状态表示它到目前为止所读入的字符构成的字符串是语言的一个句子，把所有将系统从开始状态引导到这种状态的字符串放在一起构成一个语言，该语言就是系统所能识别的语言。



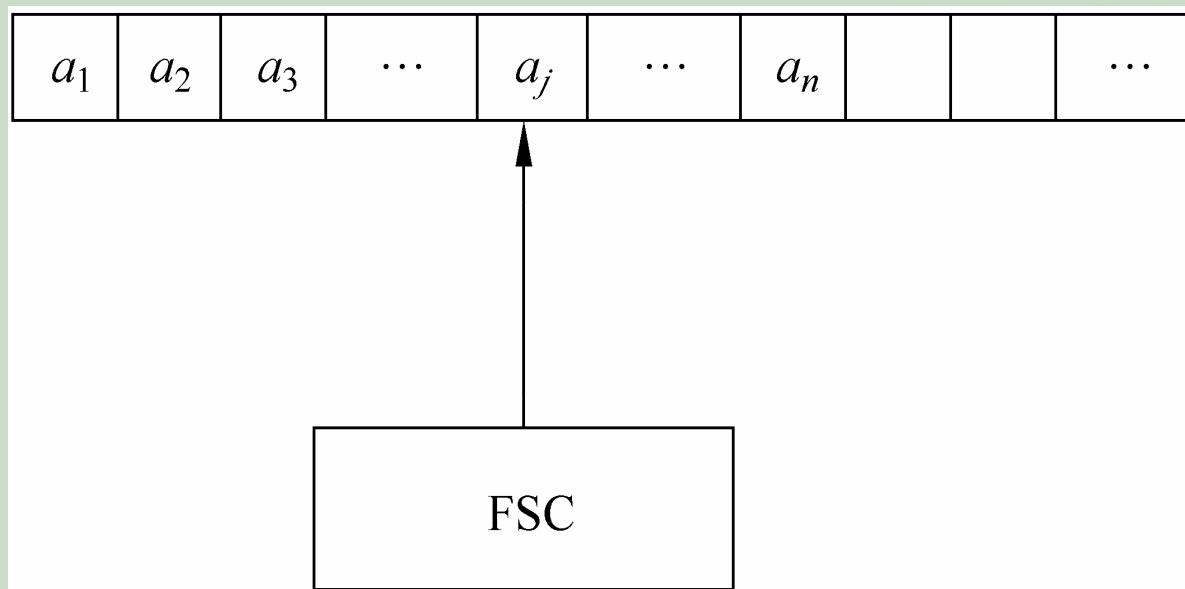
语言的识别

- 相应的物理模型
 - 一个右端无穷的输入带。
 - 一个有穷状态控制器(**finite state control, FSC**)。
 - 一个读头。
- 系统的每一个动作由三个节拍构成：读入读头正注视的字符；根据当前状态和读入的字符改变有穷控制器的状态；将读头向右移动一格。



语言的识别

■ 有穷状态自动机的物理模型



有穷状态自动机

- 有穷状态自动机(finite automaton,FA)

$$M=(Q, \Sigma, \delta, q_0, F)$$

Q ——状态的非空有穷集合。 $\forall q \in Q$, q 称为 M 的一个状态(state)。

Σ ——输入字母表(Input alphabet)。输入字符串都是 Σ 上的字符串。

q_0 —— $q_0 \in Q$, 是 M 的开始状态(initial state), 也可叫做初始状态或者启动状态。

有穷状态自动机

δ ——状态转换函数(transition function), 有时候又叫做状态移动函数。 $\delta : Q \times \Sigma \rightarrow Q$, 对 $\forall (q, a) \in Q \times \Sigma$, $\delta(q, a) = p$ 表示: M 在状态 q 读入字符 a , 将状态变成 p , 并将读头向右移动一个带方格而指向输入字符串的下一个字符。

F —— $F \subseteq Q$, 是 M 的终止状态(final state)集合。 $\forall q \in F$, q 称为 M 的终止状态, 又称为接受状态(accept state)。



有穷状态自动机

- 例：下面是一个有穷状态自动机

$$M_1 = (\{q_0, q_1, q_2\}, \{0\}, \delta_1, q_0, \{q_2\})$$

其中, $\delta_1(q_0, 0) = q_1$, $\delta_1(q_1, 0) = q_2$, $\delta_1(q_2, 0) = q_1$

用下表表示 δ_1 。

状态说明	状态	输入字符
		0
开始状态	q_0	q_1
	q_1	q_2
终止状态	q_2	q_1



有穷状态自动机

$$M_2 = (\{q_0, q_1, q_2, q_3\}, \{0, 1, 2\}, \delta_2, q_0, \{q_2\})$$

$$\delta_2(q_0, 0) = q_1, \quad \delta_2(q_1, 0) = q_2$$

$$\delta_2(q_2, 0) = q_1, \quad \delta_2(q_3, 0) = q_3$$

$$\delta_2(q_0, 1) = q_3, \quad \delta_2(q_1, 1) = q_3$$

$$\delta_2(q_2, 1) = q_3, \quad \delta_2(q_3, 1) = q_3$$

$$\delta_2(q_0, 2) = q_3, \quad \delta_2(q_1, 2) = q_3$$

$$\delta_2(q_2, 2) = q_3, \quad \delta_2(q_3, 2) = q_3$$



有穷状态自动机

δ_2 转换函数

状态说明	状态	输入字符		
		0	1	2
开始状态	q_0	q_1	q_3	q_3
	q_1	q_2	q_3	q_3
终止状态	q_2	q_1	q_3	q_3
	q_3	q_3	q_3	q_3

有穷状态自动机

- 将 δ 扩充为

$$\hat{\delta}: Q \times \Sigma^* \rightarrow Q$$

对任意的 $q \in Q$, $w \in \Sigma^*$, $a \in \Sigma$, 定义

$$(1) \hat{\delta}(q, \varepsilon) = q$$

$$(2) \hat{\delta}(q, wa) = \delta(\hat{\delta}(q, w), a)$$



有穷状态自动机

$$\begin{aligned}\hat{\delta}(q, a) &= \hat{\delta}(q, \varepsilon a) \\ &= \delta(\hat{\delta}(q, \varepsilon), a) \\ &= \delta(q, a)\end{aligned}$$

两值相同，不用区分这两个符号。



有穷状态自动机

■ 确定的有穷状态自动机

∞ 由于对于任意的 $q \in Q$, $a \in \Sigma$, $\delta(q, a)$ 均有确定的值, 所以, 将这种FA称为确定的有穷状态自动机(deterministic finite automaton, DFA)



有穷状态自动机

- **M接受(识别)的语言**

对于 $\forall x \in \Sigma^*$ 如果 $\delta(q_0, x) \in F$, 则称 x 被 M 接受, 如果 $\delta(q_0, x) \notin F$, 则称 M 不接受 x 。

$$L(M) = \{x \mid x \in \Sigma^* \text{ 且 } \delta(q_0, x) \in F\}$$

称为由 M 接受(识别)的语言。

$$L(M_1) = L(M_2) = \{0^{2n} \mid n \geq 1\}$$

- 如果 $L(M_1) = L(M_2)$, 则称 M_1 与 M_2 等价。



有穷自动机与正则语言

- 一种语言 L 称为正则 (**regular**) 语言, 当且仅当存在某个确定型有穷自动机 M 满足 $L=L(M)$ 。



有穷状态自动机

- 例：构造一个DFA，它接受的语言为 $\{x000y \mid x, y \in \{0, 1\}^*\}$

q_0 ——M的启动状态；

q_1 ——M读到了一个0，这个0可能是子串“000”的第1个0；

q_2 ——M在 q_1 后紧接着又读到了一个0，这个0可能是子串“000”的第2个0；

q_3 ——M在 q_2 后紧接着又读到了一个0，发现输入字符串含有子串“000”；因此，这个状态应该是终止状态。

有穷状态自动机

可得到关于M的状态转换函数:

$\delta(q_0, 0) = q_1$ ——M读到了一个0, 这个0可能是子串“000”的第1个0;

$\delta(q_1, 0) = q_2$ ——M又读到了一个0, 这个0可能是子串“000”的第2个0;

$\delta(q_2, 0) = q_3$ ——M找到子串“000”;



有穷状态自动机

$\delta(q_0, 1) = q_0$ ——M在 q_0 读到了一个1，它需要继续留在 q_0 “等待”可能是子串“000”的第1个0的输入字符0；

$\delta(q_1, 1) = q_0$ ——M在刚刚读到了一个0后，读到了一个1，表明在读入这个1之前所读入的0并不是子串“000”的第1个0，因此，M需要重新回到状态 q_0 ，以寻找子串“000”的第1个0；



有穷状态自动机

$\delta(q_2, 1) = q_0$ ——M在刚刚发现了00后，读到了一个1，表明在读入这个1之前所读入的00并不是子串“000”的前两个0，因此，M需要重新回到状态 q_0 ，以寻找子串“000”的第1个0；

$\delta(q_3, 0) = q_3$ ——M找到了子串“000”，只用读入该串的剩余部分。

$\delta(q_3, 1) = q_3$ ——M找到了子串“000”，只用读入该串的剩余部分。



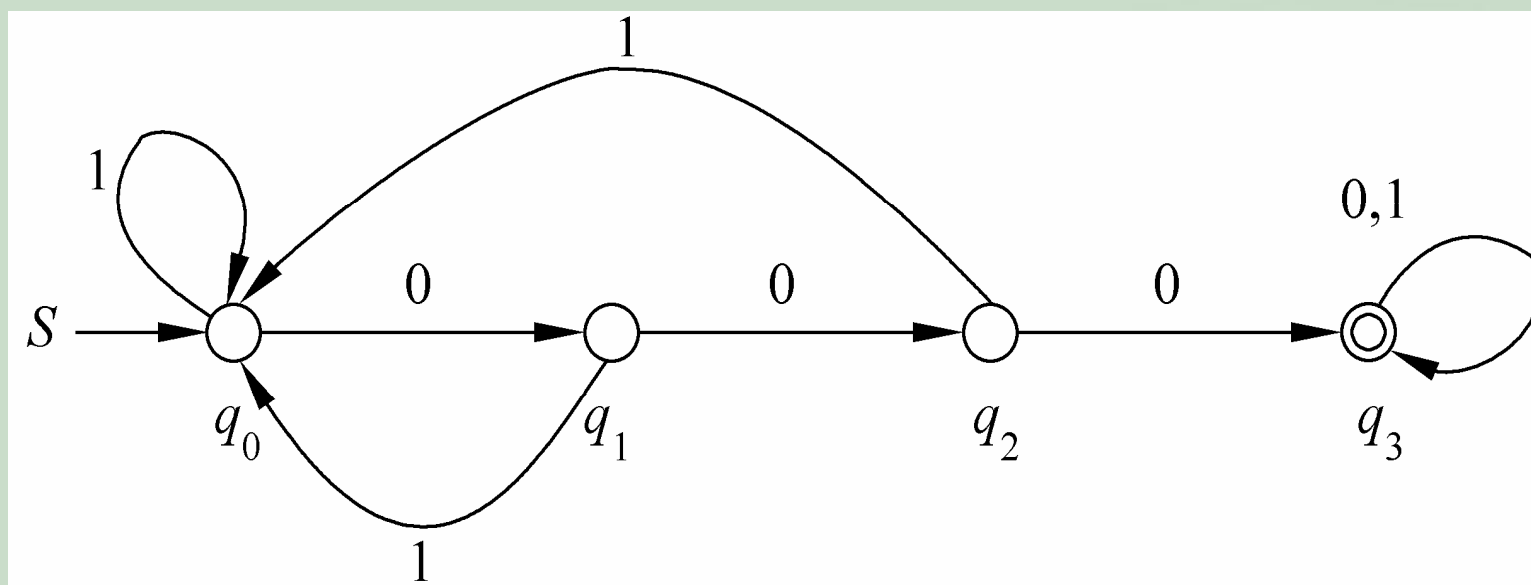
有穷状态自动机

$M = (\{q_0, q_1, q_2, q_3\}, \{0, 1\}, \{(q_0, 0) = q_1,$
 $\delta(q_1, 0) = q_2, \delta(q_2, 0) = q_3, \delta(q_0, 1) = q_0,$
 $\delta(q_1, 1) = q_0, \delta(q_2, 1) = q_0, \delta(q_3, 0) = q_3,$
 $\delta(q_3, 1) = q_3\}, q_0, \{q_3\})$

状态说明	状态	输入字符	
		0	1
开始状态	q_0	q_1	q_0
	q_1	q_2	q_0
	q_2	q_3	q_0
终止状态	q_3	q_3	q_3

有穷状态自动机

- 一种更为直观的表达:



有穷状态自动机

■ 状态转换图(transition diagram)

- (1) $q \in Q \Leftrightarrow q$ 是该有向图中的一个顶点;
- (2) $\delta(q, a) = p \Leftrightarrow$ 图中有一条从顶点 q 到顶点 p 的标记为 a 的弧;
- (3) $q \in F \Leftrightarrow$ 标记为 q 的顶点被用双层圈标出;
- (4) 用标有 S 的箭头指出 M 的开始状态。



有穷状态自动机

- 例：构造一个DFA，它接受的语言为 $\{x000|x \in \{0, 1\}^*\}$ 。



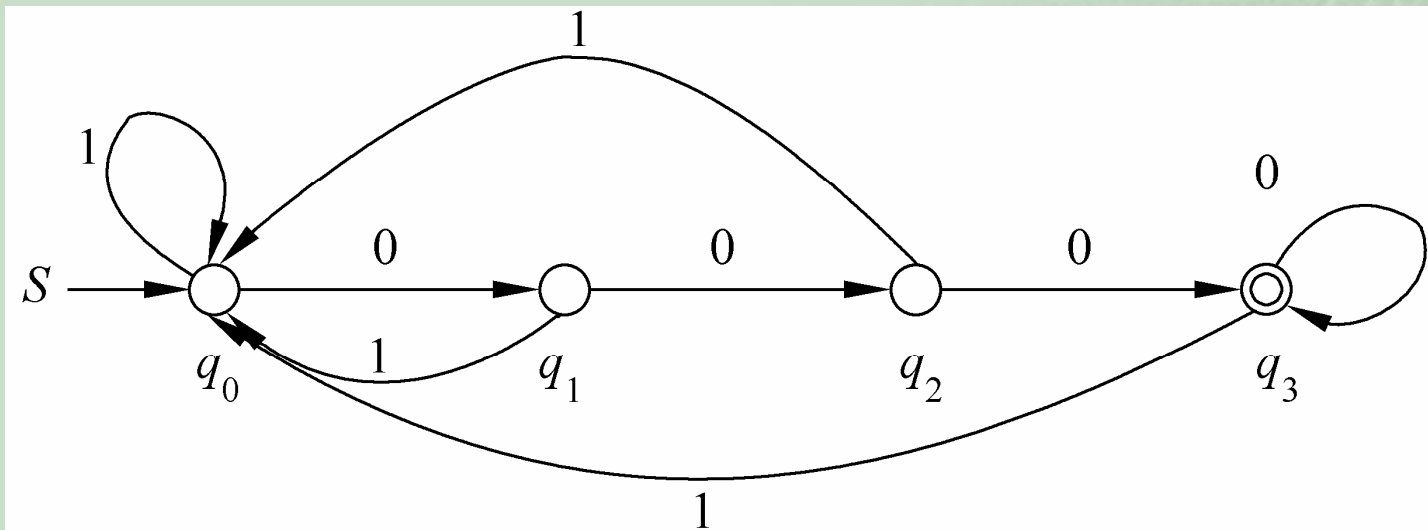
有穷状态自动机

- 例：构造一个DFA，它接受的语言为 $\{x000|x \in \{0, 1\}^*\}$ 。
 - ∞ 状态 q_0 读到的0可能是输入字符串的最后三个0的第1个0；
 - ∞ 在状态 q_1 紧接着读到的0可能是输入字符串的最后三个0的第2个0；
 - ∞ 在状态 q_2 紧接着读到的0可能是输入字符串的最后三个0的第3个0；



有穷状态自动机

- 在状态 q_3 紧接着读到的0也可能是输入字符串的最后三个0的第3个0;
- 如果在状态 q_1 , q_2 , q_3 读到的是1, 则要重新检查输入串是否以三个0结尾。



有穷状态自动机

■ 几点值得注意

- (1) 定义FA时，常常只给出FA相应的状态转换图就可以了。
- (2) 对于DFA来说，并行的弧按其上的标记字符的个数计算，对于每个顶点来说，它的出度恰好等于输入字母表中所含的字符的个数。



有穷状态自动机

- (3) 不难看出，字符串 x 被FA M 接受的充分必要条件是，在 M 的状态转换图中存在一条从开始状态到某一个终止状态的有向路，该有向路上从第1条边到最后一条边的标记依次并置而构成字符串 x 。简称此路的标记为 x 。
- (4) 一个FA可以有多个的终止状态。



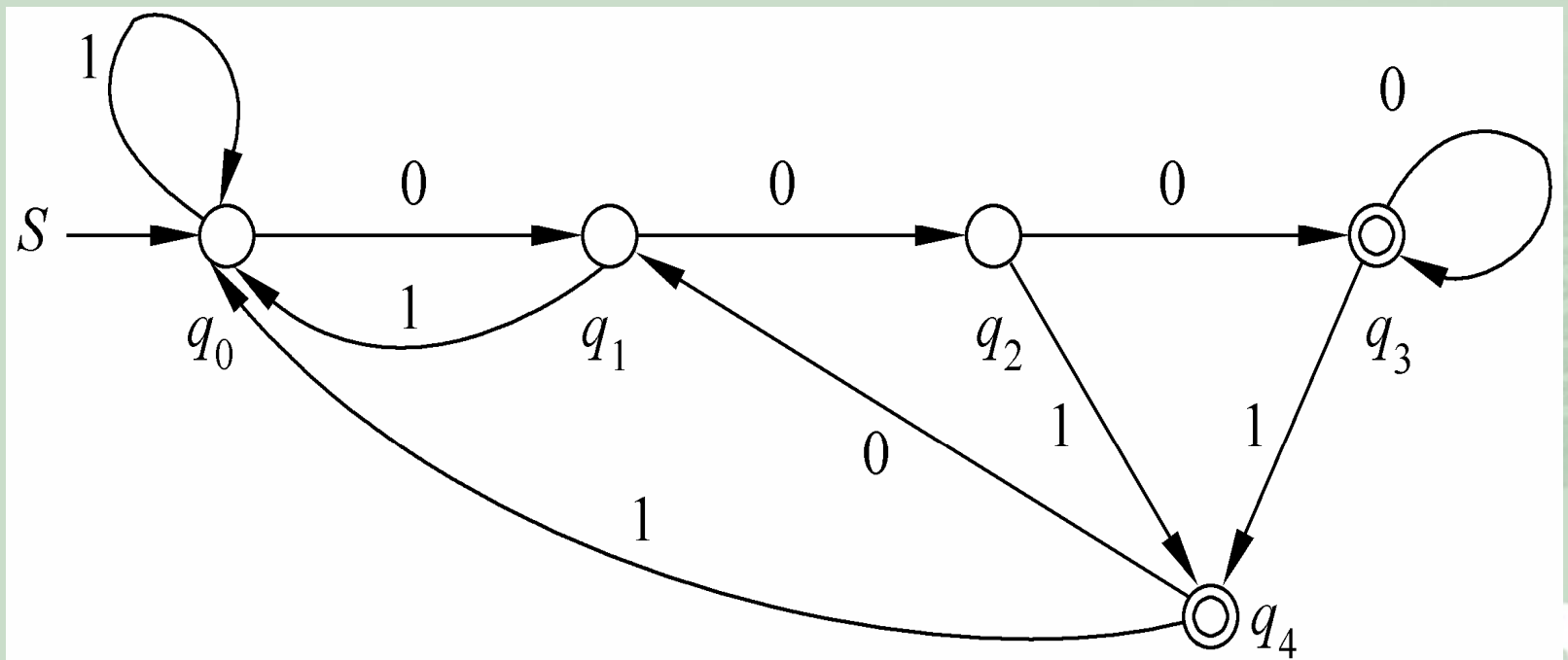
有穷状态自动机

接受语言 $\{x000|x \in \{0, 1\}^*\} \cup \{x001|x \in \{0, 1\}^*\}$ 的FA。



有穷状态自动机

接受语言 $\{x000|x \in \{0, 1\}^*\} \cup \{x001|x \in \{0, 1\}^*\}$ 的FA。



有穷状态自动机

■ 即时描述(instantaneous description, ID)

∞ $x, y \in \Sigma^*$, $\delta(q_0, x) = q$, xqy 称为 M 的一个即时描述, 表示 xy 是 M 正在处理的一个字符串, x 引导 M 从 q_0 启动并到达状态 q , M 当前正注视着 y 的首字符。

∞ 如果 $xqay$ 是 M 的一个即时描述, 且 $\delta(q, a) = p$, 则 $xqay \vdash_M xapy$ 。



有穷状态自动机

- $\alpha \vdash_M^n \beta$: 表示M从即时描述 α 经过n次移动到达即时描述 β , 即
M存在即时描述 $\alpha_1, \alpha_2, \dots, \alpha_{n-1}$, 使得
 $\alpha \vdash_M \alpha_1, \alpha_1 \vdash_M \alpha_2, \dots, \alpha_{n-1} \vdash_M \beta$
- 当n=0时, 有 $\alpha = \beta$ 。即 $\alpha \vdash_M^0 \alpha$ 。
- $\alpha \vdash_M^+ \beta$: 表示M从即时描述 α 经过至少1次移动到达即时描述 β 。
- $\alpha \vdash_M^* \beta$: 表示M从即时描述 α 经过若干步移动到达即时描述 β 。

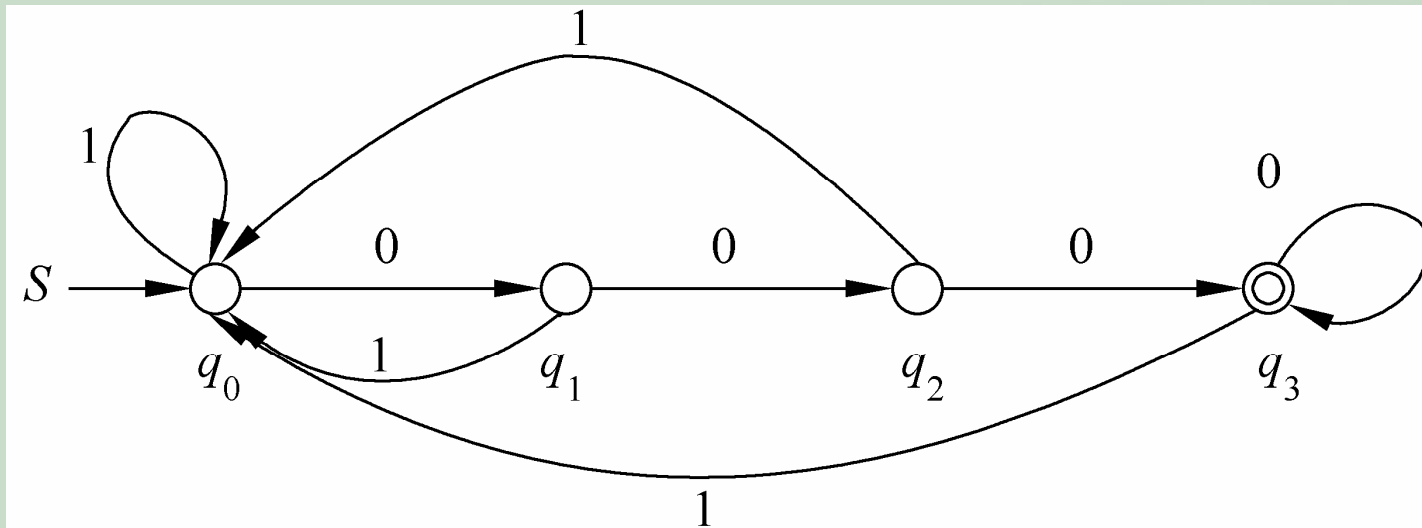
有穷状态自动机

- 当意义清楚时，我们将符号 \vdash_M 、 \vdash_M^n 、 \vdash_M^* 、 \vdash_M^+ 中的 M 省去，分别用 \vdash 、 \vdash^n 、 \vdash^* 、 \vdash^+ 表示。



有穷状态自动机

对下图所示的**DFA**有如下的**ID**转换：



有穷状态自动机

$q_01010010001 \vdash 1q_0010010001$
 $\vdash 10q_110010001$
 $\vdash 101q_00010001$
 $\vdash 1010q_1010001$
 $\vdash 10100q_210001$
 $\vdash 101001q_00001$
 $\vdash 1010010q_1001$
 $\vdash 10100100q_201$



有穷状态自动机

$$\vdash 101001000q_31$$

$$\vdash 1010010001q_0$$

即 $q_01010010001 \vdash^{10} 1010010001q_0$

$$q_01010010001 \vdash^+ 1010010001q_0$$

$$q_01010010001 \vdash^* 1010010001q_0$$



有穷状态自动机

- 对于 $x \in \Sigma^*$,

$$q_0x1 \vdash^+ x1q_0$$

$$q_0x10 \vdash^+ x10q_1$$

$$q_0x100 \vdash^+ x100q_2$$

$$q_0x000 \vdash^+ x000q_3$$

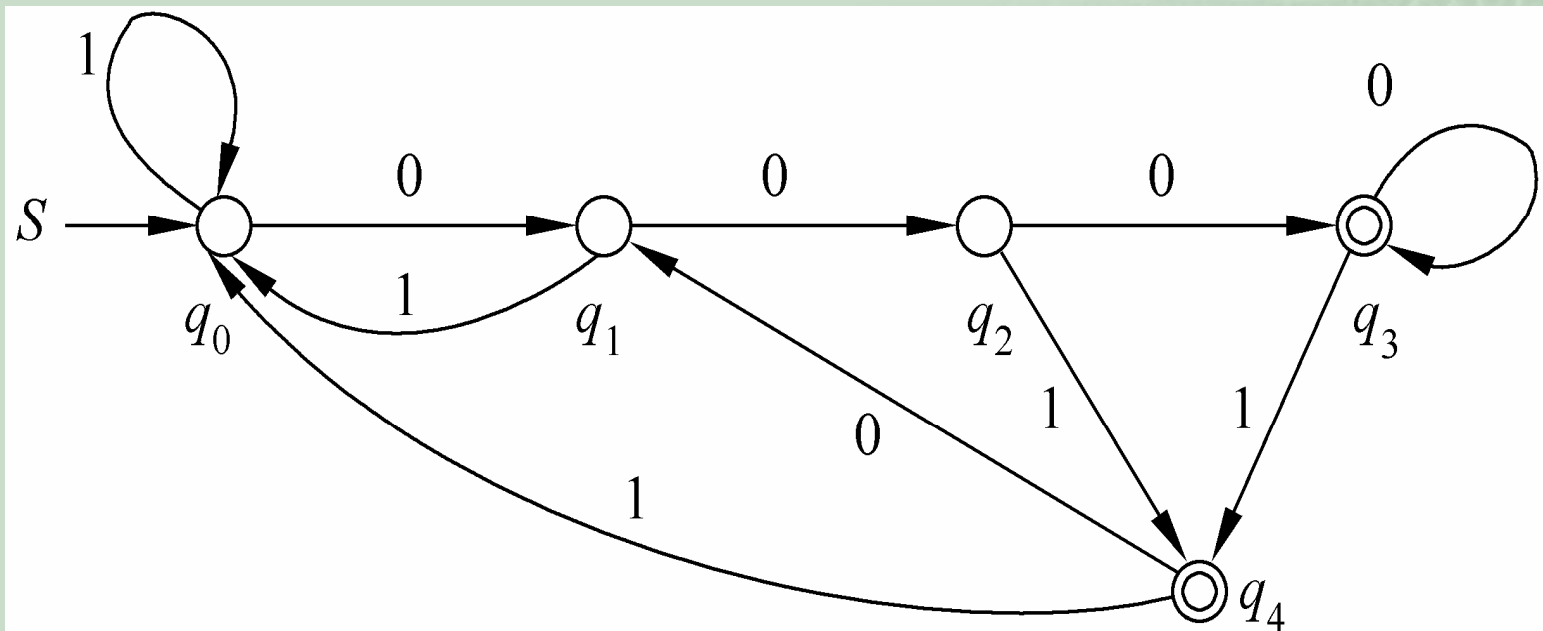


有穷状态自动机

能引导FA从开始状态到达 q 的字符串的集合为：

$$\text{set}(q) = \{x \mid x \in \Sigma^*, \delta(q_0, x) = q\}$$

对下图所给的DFA 中的所有 q ，求 $\text{set}(q)$ 。



$\text{set}(q_0) = \{x \mid x \in \Sigma^*, x = \varepsilon \text{ 或者 } x \text{ 以 } 1 \text{ 但不是 } 001 \text{ 结尾}\}$

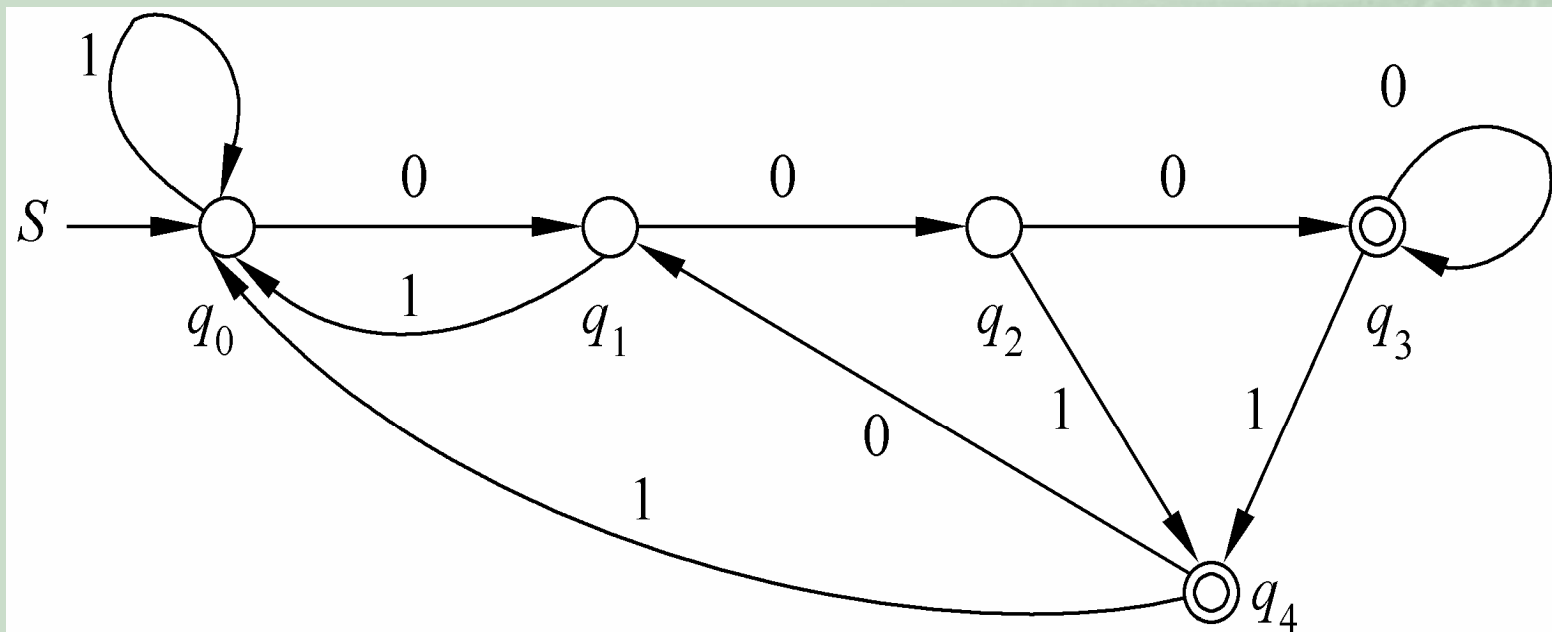
$\text{set}(q_1) = \{x \mid x \in \Sigma^*, x = 0 \text{ 或者 } x \text{ 以 } 10 \text{ 结尾}\}$

$\text{set}(q_2) = \{x \mid x \in \Sigma^*, x = 00 \text{ 或者 } x \text{ 以 } 100 \text{ 结尾}\}$

$\text{set}(q_3) = \{x \mid x \in \Sigma^*, x \text{ 以 } 000 \text{ 结尾}\}$

$\text{set}(q_4) = \{x \mid x \in \Sigma^*, x \text{ 以 } 001 \text{ 结尾}\}$

这5个集合两两互不相交。



有穷状态自动机

- 对于任意一个DFA $M=(Q, \Sigma, \delta, q_0, F)$ 我们可以按照如下方式定义关系 R_M :
- 对 $\forall x, y \in \Sigma^*$, $xR_M y \Leftrightarrow \exists q \in Q$, 使得 $x \in \text{set}(q)$ 和 $y \in \text{set}(q)$ 同时成立。
- 按照这个定义所得到的关系实际上是 Σ^* 上的一个等价关系。利用这个关系, 可以将 Σ^* 划分成不多于 $|Q|$ 个等价类。



有穷状态自动机

- 例：构造一个DFA，它接受的语言为 $\{0^n 1^m 2^k | n, m, k \geq 1\}$ 。

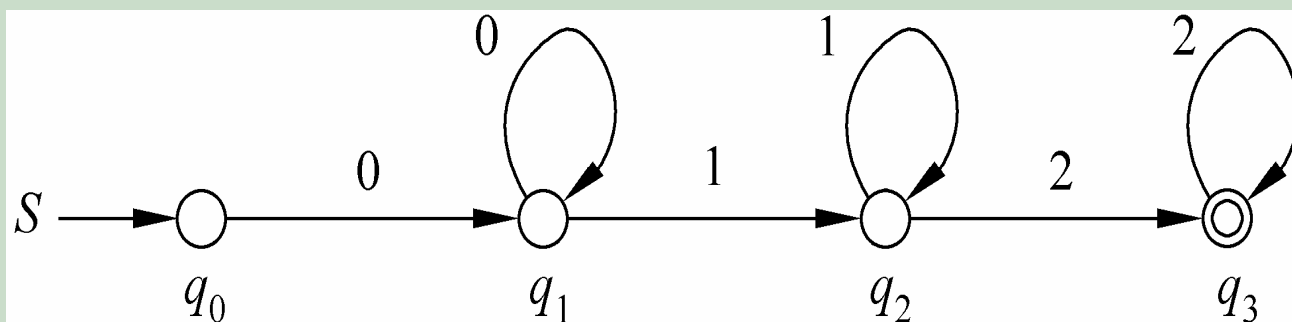


有穷状态自动机

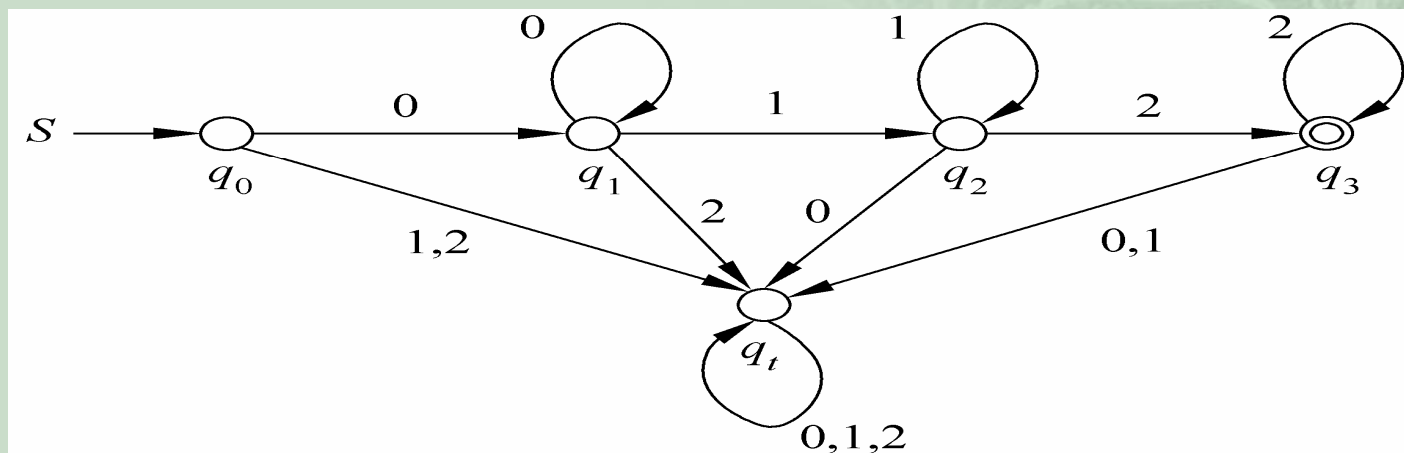
- 例：构造一个DFA，它接受的语言为 $\{0^n 1^m 2^k | n, m, k \geq 1\}$ 。
 - q_0 ——M的启动状态；
 - q_1 ——M读到至少一个0，并等待读更多的0；
 - q_2 ——M读到至少一个0后，读到了至少一个1，并等待读更多的1；
 - q_3 ——M读到至少一个0后跟至少一个1后，并且接着读到了至少一个2。

有穷状态自动机

- 先设计“主体框架”



- 再补充细节



有穷状态自动机

- (1) 当FA一旦进入状态 q_t ，它就无法离开此状态。所以， q_t 相当于一个陷阱状态(trap)。一般地，我们将陷阱状态用作在其他状态下发现输入串不可能是该FA所识别的语言的句子时进入的状态。在此状态下，FA读完输入串中剩余的字符。



有穷状态自动机

- (2) 在构造一个识别给定语言的FA时，用画图的方式比较方便、直观。我们可以先根据语言的主要特征画出该FA的“主体框架”，然后再去考虑画出一些细节要求的内容。



有穷状态自动机

(3) **FA**的状态具有一定的记忆功能：不同的状态对应于不同的情况，由于**FA**只有有穷个状态，所以，在识别一个语言的过程中，如果有无穷种情况需要记忆，我们肯定是无法构造出相应的**FA**的。



有穷状态自动机

- 例：构造一个DFA，它接受的语言为 $\{x | x \in \{0, 1\}^*, \text{且当把} x \text{看成二进制数时, } x \text{模} 3 \text{与} 0 \text{同余}\}$ 。



有穷状态自动机

- 例：构造一个DFA，它接受的语言为 $\{x|x \in \{0, 1\}^*, \text{且当把} x \text{看成二进制数时, } x \text{模} 3 \text{与} 0 \text{同余}\}$ 。
- q_0 ——对应除以3余数为0的 x 组成的等价类；
- q_1 ——对应除以3余数为1的 x 组成的等价类；
- q_2 ——对应除以3余数为2的 x 组成的等价类；
- q_s ——M的开始状态。



有穷状态自动机

- q_s ——在此状态下读入0时，有 $x=0$ ，所以应该进入状态 q_0 ；读入1时，有 $x=1$ ，所以应该进入状态 q_1 。即： $\delta(q_s, 0) = q_0$ ； $\delta(q_s, 1) = q_1$ 。



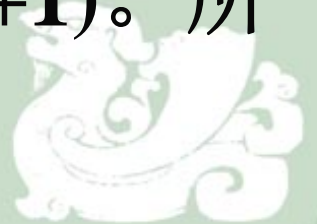
有穷状态自动机

- q_0 ——能引导M到达此状态的x除以3余0，所以有： $x=3*n+0$ 。
- 读入0时，引导M到达下一个状态的字符串为x0， $x0=2*(3*n+0)=3*2*n+0$ 。所以， $\delta(q_0, 0)=q_0$ ；
- 读入1时，M到达下一个状态的字符串为x1， $x1=2*(3*n+0)+1=3*2*n+1$ 。所以， $\delta(q_0, 1)=q_1$ ；



有穷状态自动机

- q_1 ——能引导M到达此状态的x除以3余1，所以有： $x=3*n+1$ 。
- 读入0时，引导M到达下一个状态的字符串为 x_0 ， $x_0=2*(3*n+1)=3*2*n+2$ 。所以即： $\delta(q_1, 0)=q_2$;
- 读入1时，引导M到达下一个状态的字符串为 x_1 ， $x_1=2*(3*n+1)+1=3*2*n+2+1=3*(2*n+1)$ 。所以 $\delta(q_1, 1)=q_0$



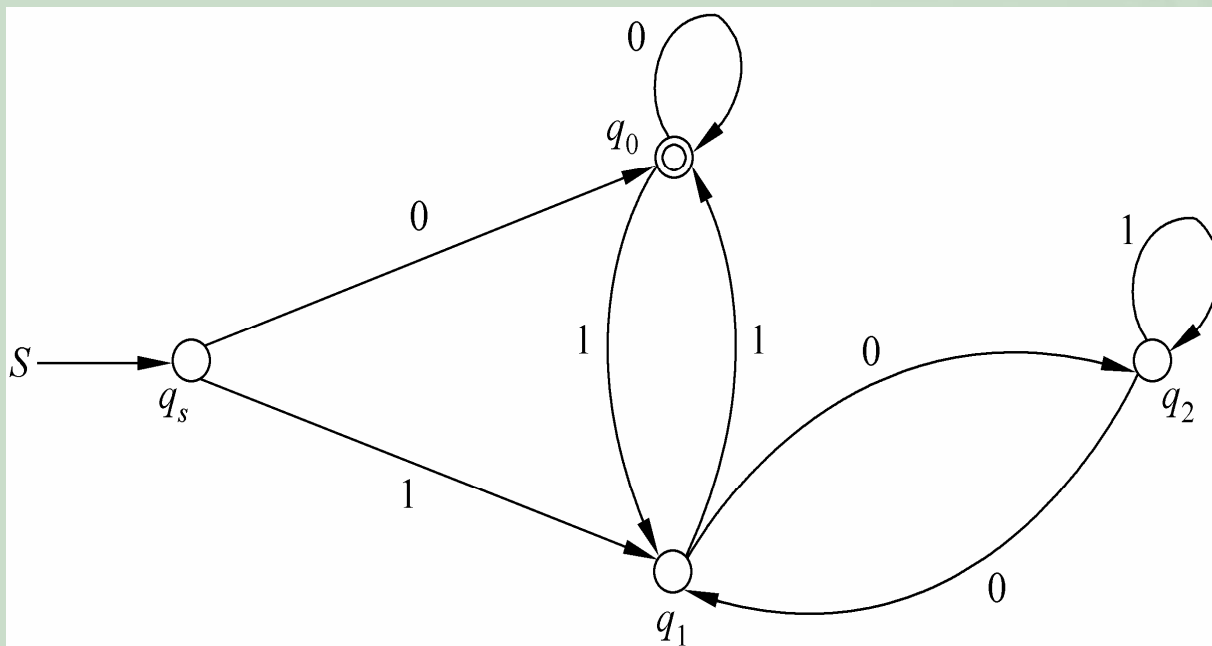
有穷状态自动机

- q_2 ——能引导M到达此状态的x除以3余2，所以：
 $x=3*n+2$ 。
- 读入0时，引导M到达下一个状态的字符串为x0，
 $x0=2*(3*n+2)=3*2*n+4=3*(2*n+1)+1$ 。所以，
 $\delta(q_2, 0)=q_1$ ；
- 读入1时，引导M到达下一个状态的字符串为x1，
 $x1=2*(3*n+2)+1=3*2*n+4+1=3*(2*n+1)+2$ 。所以，
 $\delta(q_2, 1)=q_2$ 。



有穷状态自动机

- 接受语言 $\{x | x \in \{0, 1\}^*, \text{且当把} x \text{看成二进制数时, } x \bmod 3 \text{与} 0 \text{同余}\}$ 的 DFA 如下:



有穷状态自动机

- 例：构造一个DFA，它接受的语言 $L = \{x | x \in \{0, 1\}^*, \text{ 且对 } x \text{ 中任意一个长度不大于5的子串 } a_1a_2\dots a_n, a_1+a_2+\dots+a_n \leq 3, n \leq 5\}$ 。
- 输入串为 $a_1a_2\dots a_i\dots a_{i+4}a_{i+5}\dots a_m$



有穷状态自动机

- 当 $i=1, 2, 3$, 也就是 M 读到输入串的第1、2、3个字符时, 它需要将这些字符记下来。因为 $a_1 \dots a_i$ 可能需要用来判定输入串的最初4~5个字符组成的子串是否满足语言的要求。
- 当 $i=4, 5$, 也就是 M 读到输入串的第4、5个字符时, 在 $a_1 + a_2 + \dots + a_i \leq 3$ 的情况下, M 需要将 $a_1 \dots a_i$ 记下来; 在 $a_1 + a_2 + \dots + a_i > 3$ 时, M 应该进入陷阱状态 q_t 。



有穷状态自动机

- 当 $i=6$ ，也就是M读到输入串的第6个字符，此时，以前读到的第1个字符 a_1 就没有用了，此时它要看 $a_2+a_3+\dots+a_6 \leq 3$ 是否成立，如果成立，M需要将 $a_2\dots a_6$ 记下来；在 $a_2+a_3+\dots+a_i > 3$ 时，M应该进入陷阱状态 q_t 。



有穷状态自动机

- 当M完成对子串 $a_1a_2\dots a_i\dots a_{i+4}$ 的考察，并发现它满足语言的要求时，M记下来的是 $a_i\dots a_{i+4}$ ，此时它读入输入串的第 $i+5$ 个字符 a_{i+5} ，以前读到的第 i 个字符 a_i 就没有用了，此时它要看 $a_{i+1}+a_{i+2}+\dots+a_{i+5}\leq 3$ 是否成立，如果成立，M需要将 $a_{i+1}, a_{i+2}, \dots, a_{i+5}$ 记下来；在 $a_{i+1}+a_{i+2}+\dots+a_{i+5}>3$ 时，M应该进入陷阱状态 q_t 。



有穷状态自动机

- M需要记忆的内容有：
- 什么都未读入—— $2^0=1$ 种；
- 记录有1个字符—— $2^1=2$ 种；
- 记录有2个字符—— $2^2=4$ 种；
- 记录有3个字符—— $2^3=8$ 种；
- 记录有4个字符—— $2^4-1=15$ 种；
- 记录有5个字符—— $2^5-6=26$ 种；
- 记录当前的输入串不是句子——1种。



有穷状态自动机

状态设置:

$q[\varepsilon]$ ——M还未读入任何字符;

q_t ——陷阱状态;

$q[a_1a_2\dots a_i]$ ——M记录有*i*个字符, $1 \leq i \leq 5$ 。 $a_1, a_2, \dots, a_i \in \{0, 1\}$ 。

取DFA $M=(Q, \{0, 1\}, \delta, q[\varepsilon], F)$

$F=\{ q[\varepsilon] \} \cup \{ q[a_1a_2\dots a_i] \mid a_1, a_2, \dots, a_i \in \{0, 1\} \text{ 且 } 1 \leq i \leq 5 \text{ 且 } a_1+a_2+\dots+a_i \leq 3 \}$

$Q=\{q_t\} \cup F$



有穷状态自动机

- $\delta(q[\varepsilon], a_1) = q[a_1]$
 - $\delta(q[a_1], a_2) = q[a_1a_2]$
 - $\delta(q[a_1a_2], a_3) = q[a_1a_2a_3]$
- $$\delta(q[a_1a_2a_3], a) = \begin{cases} q[a_1a_2a_3a] & \text{如果 } a_1 + a_2 + a_3 + a \leq 3 \\ q_t & \text{如果 } a_1 + a_2 + a_3 + a > 3 \end{cases}$$



有穷状态自动机

$$\delta(q[a_1a_2a_3a_4], a) = \begin{cases} q[a_1a_2a_3a_4a] & \text{如果 } a_1+a_2+a_3+a_4+a \leq 3 \\ q_t & \text{如果 } a_1+a_2+a_3+a_4+a > 3 \end{cases}$$

$$\delta(q[a_1a_2a_3a_4a_5], a) = \begin{cases} q[a_2a_3a_4a_5a] & a_2+a_3+a_4+a_5+a \leq 3 \\ q_t & \text{如果 } a_2+a_3+a_4+a_5+a > 3 \end{cases}$$

$$\delta(q_t, a) = q_t$$



有穷状态自动机

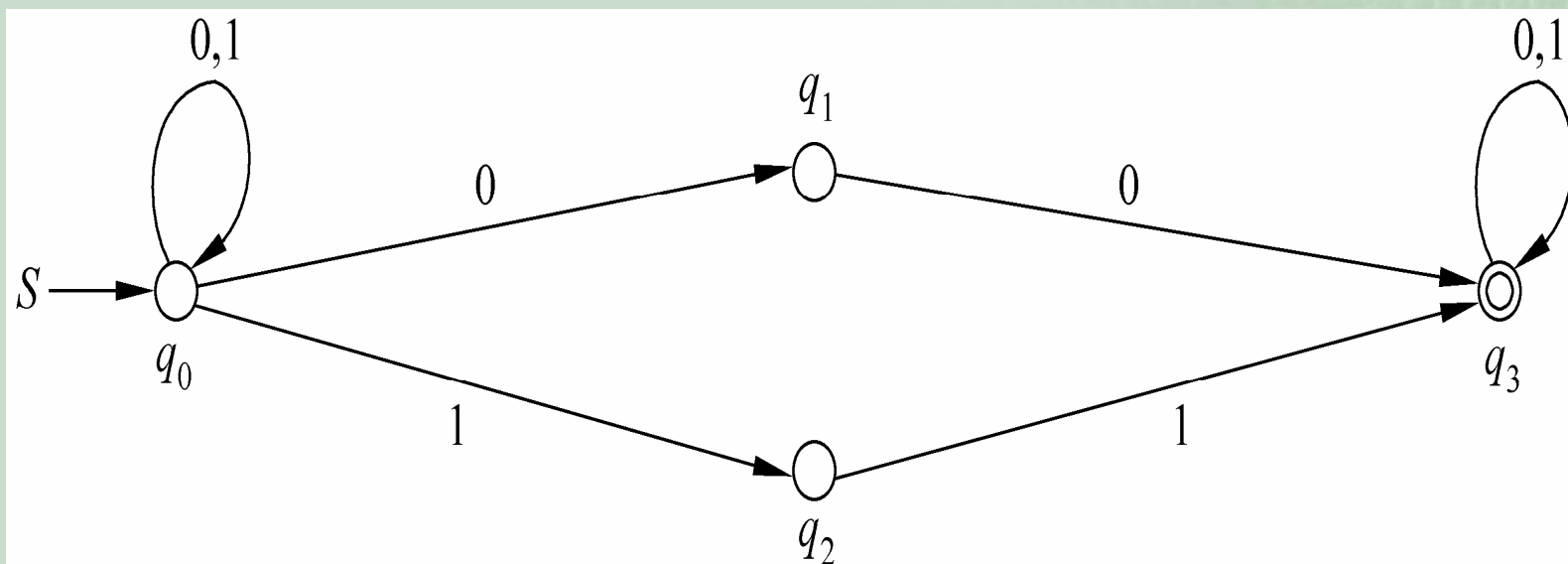
- 可考虑针对不同关注后缀定义状态：
1, 10, 11, 101, 110, 111, 1011, 1101, 1110
- 为什么不考虑100, 1010, 1100, ...



NFA

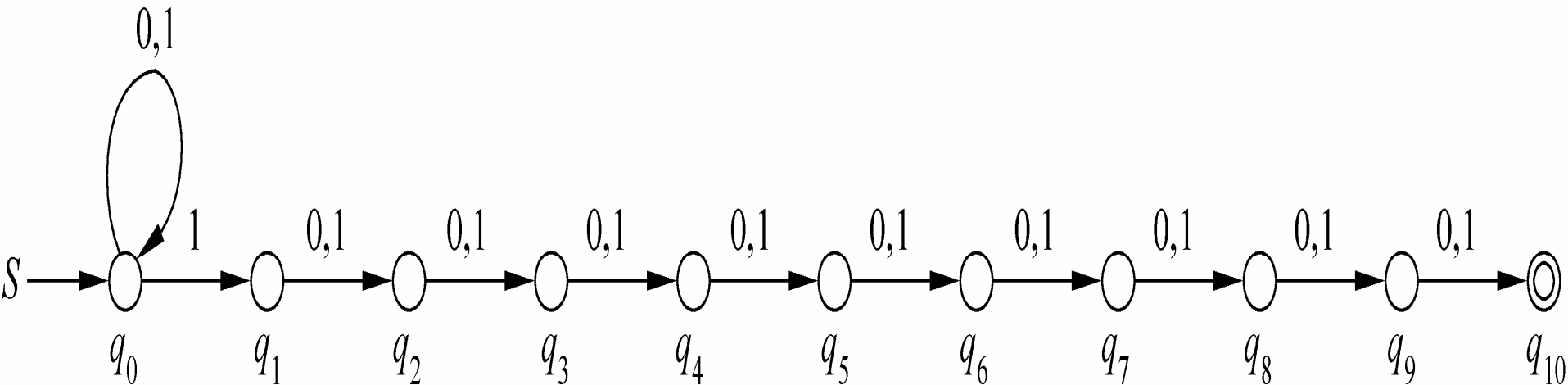
作为对DFA的修改

- 希望是接受 $\{x \mid x \in \{0, 1\}^*, \text{且} x \text{含有子串} 00 \text{或} 11\}$ 的FA如下:



作为对DFA的修改

- 希望是接受 $\{x|x \in \{0, 1\}^*, \text{且} x \text{ 的倒数第10个字符为1}\}$ 的FA如下：



作为对DFA的修改

- 这两个图所给的“FA”与前面我们所定义的FA(即DFA)的区别在于:
 - (1) 并不是对于所有的 $(q, a) \in \Sigma \times Q$, $\delta(q, a)$ 都有一个状态与它对应;
 - (2) 并不是对于所有的 $(q, a) \in \Sigma \times Q$, $\delta(q, a)$ 只对应一个状态。
- “FA”在任意时刻可以处于有穷多个状态。
- “FA”具有“智能”。



NFA的形式定义

- 不确定的有穷状态自动机(non-deterministic finite automaton ,NFA)

M是一个五元组

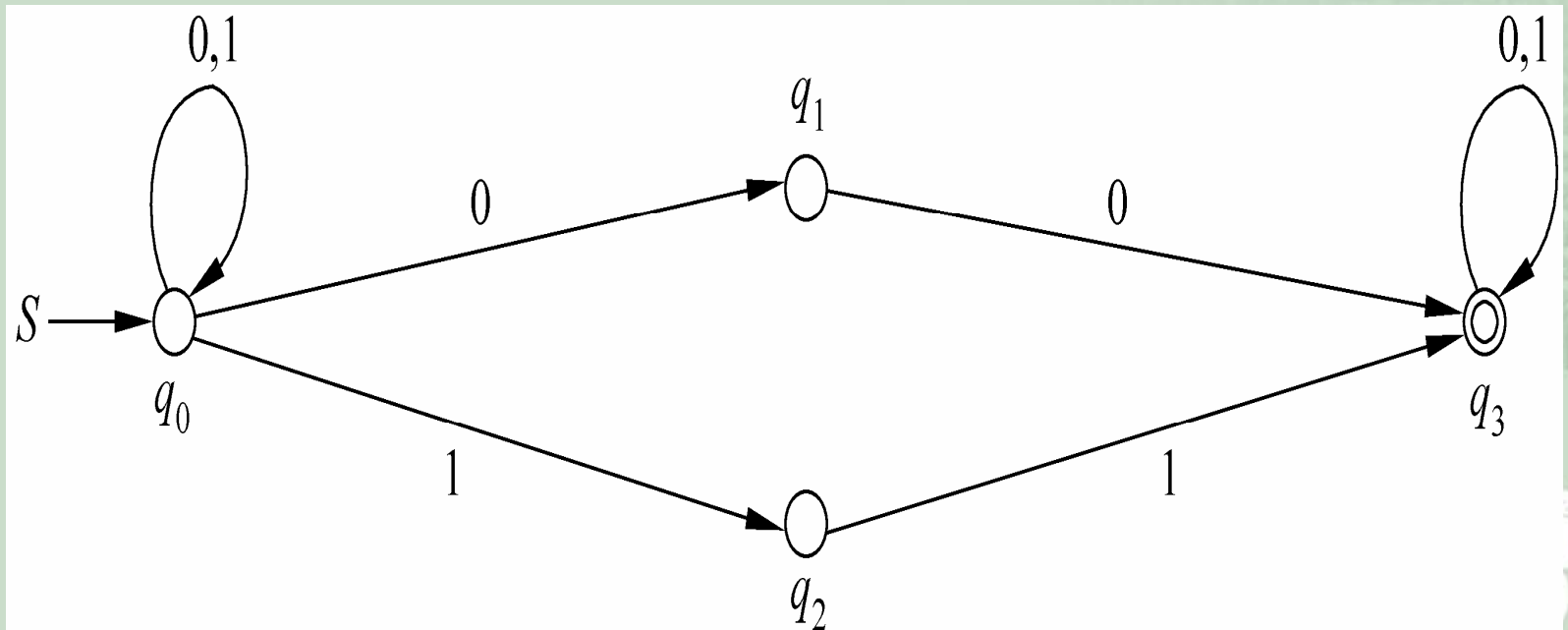
$$M=(Q, \Sigma, \delta, q_0, F)$$

∞ Q、 Σ 、 q_0 、F的意义同DFA。

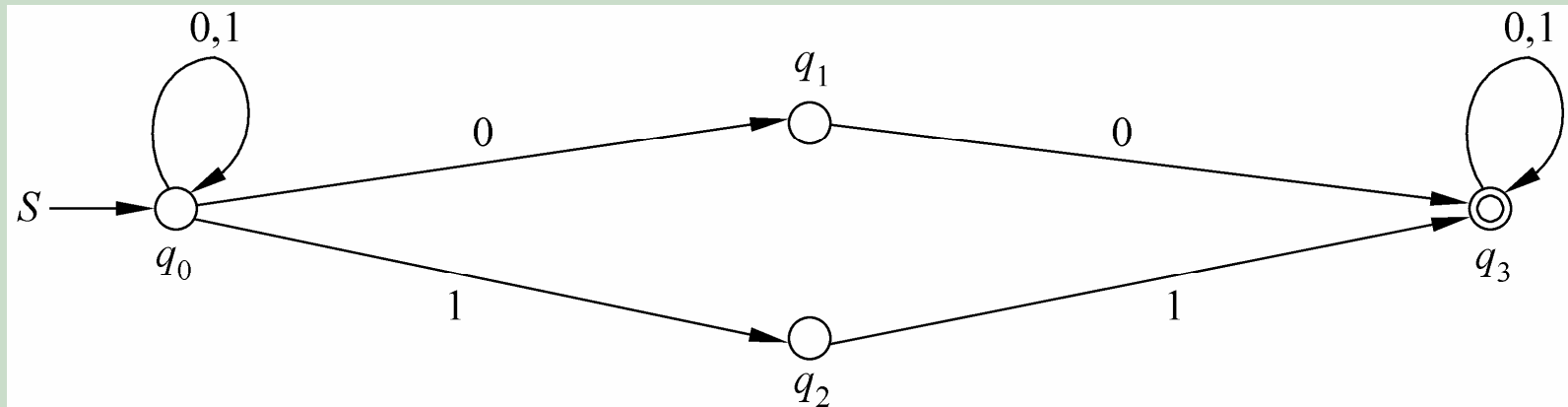
∞ $\delta : Q \times \Sigma \rightarrow 2^Q$, 对 $\forall (q, a) \in Q \times \Sigma$, $\delta(q, a) = \{p_1, p_2, \dots, p_m\}$ 表示M在状态q读入字符a, 可以选择地将状态变成 p_1 、或者 p_2 、...、或者 p_m , 并将读头向右移动一个带方格而指向输入字符串的下一个字符。

NFA的形式定义

- 接受 $\{x|x \in \{0, 1\}^*, \text{且} x \text{含有子串} 00 \text{或} 11\}$ 的FA对应的转换函数定义表。



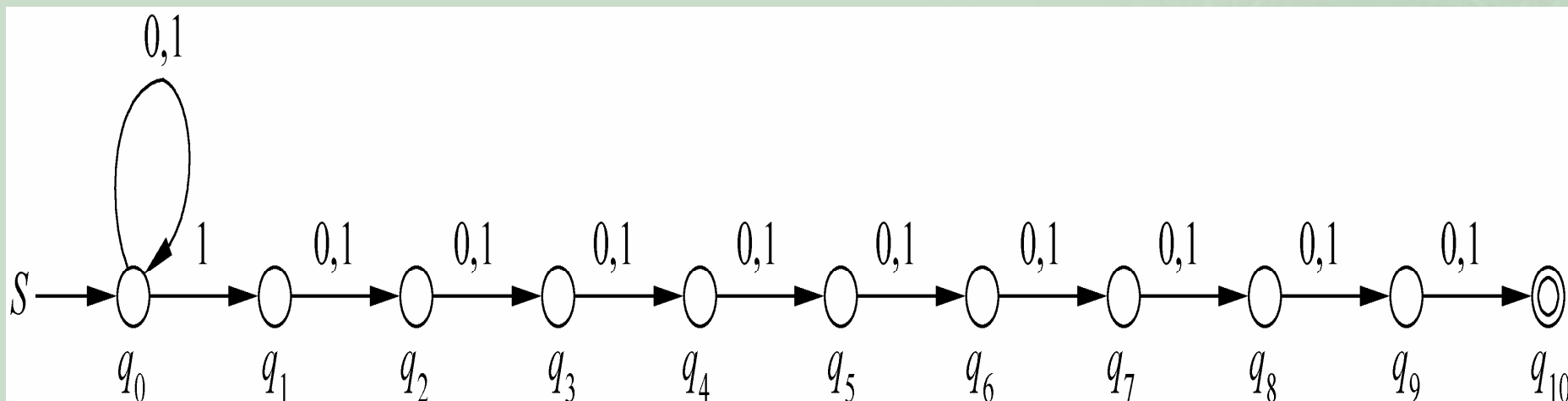
NFA的形式定义



状态说明	状态	输入字符	
		0	1
启动状态	q_0	$\{q_0, q_1\}$	$\{q_0, q_2\}$
	q_1	$\{q_3\}$	Φ
	q_2	Φ	$\{q_3\}$
终止状态	q_3	$\{q_3\}$	$\{q_3\}$

NFA的形式定义

- 接受 $\{x|x \in \{0, 1\}^*, \text{且} x \text{ 的倒数第10个字符为} 1\}$ 的FA对应的转换函数定义表。



NFA的形式定义

状态说明	状态	输入字符	
		0	1
启动状态	q_0	$\{q_0\}$	$\{q_0, q_1\}$
	q_1	$\{q_2\}$	$\{q_2\}$
	q_2	$\{q_3\}$	$\{q_3\}$
	q_3	$\{q_4\}$	$\{q_4\}$
	q_4	$\{q_5\}$	$\{q_5\}$
	q_5	$\{q_6\}$	$\{q_6\}$
	q_6	$\{q_7\}$	$\{q_7\}$
	q_7	$\{q_8\}$	$\{q_8\}$
	q_8	$\{q_9\}$	$\{q_9\}$
	q_9	$\{q_{10}\}$	$\{q_{10}\}$
终止状态	q_{10}	Φ	Φ

NFA的形式定义

- 将 δ 扩充为

$$\hat{\delta}: Q \times \Sigma^* \rightarrow 2^Q$$

对任意的 $q \in Q$, $w \in \Sigma^*$, $a \in \Sigma$, 定义

$$(1) \hat{\delta}(q, \varepsilon) = q$$

$$(2) \hat{\delta}(q, wa) = \{p \mid \exists r \in \hat{\delta}(q, w) \text{ 使得 } p \in \delta(r, a)\}$$

NFA的形式定义

$$\begin{aligned}\hat{\delta}(q, a) &= \hat{\delta}(q, \varepsilon a) \\ &= \{p \mid \exists r \in (q, \varepsilon), p \in \delta(r, a)\} \\ &= \{p \mid \exists r \in \{q\}, p \in \delta(r, a)\} \\ &= \{p \mid p \in \delta(q, a)\} \\ &= \delta(q, a)\end{aligned}$$

和关于DFA的结论一样，两值相同，也不用区分这两个符号。



NFA的形式定义

- 进一步扩充 δ 的定义域: $\delta : 2^Q \times \Sigma^* \rightarrow 2^Q$ 。
对任意的 $P \subseteq Q$, $w \in \Sigma^*$

$$\delta(P, w) = \bigcup_{q \in P} \delta(q, w)$$



NFA的形式定义

- 由于，对 $\forall (q, w) \in Q \times \Sigma^*$

$$\begin{aligned}\delta(\{q\}, w) &= \bigcup_{q \in \{q\}} \delta(q, w) \\ &= \delta(q, w)\end{aligned}$$

所以，不一定严格地区分 δ 的第1个分量是一个状态还是一个含有一个元素的集合。



NFA的形式定义

- 对任意的 $q \in Q$, $w \in \Sigma^*$, $a \in \Sigma$:

$$\delta(q, wa) = \delta(\delta(q, w), a)$$

- 对输入字符串 $a_1a_2\dots a_n$

$$\delta(q, a_1a_2\dots a_n) = \delta(\delta(\dots \delta(\delta(q, a_1), a_2), \dots), a_n) \text{ 。$$



NFA的形式定义

■ M接受(识别)的语言

↻ 对于 $\forall x \in \Sigma^*$, 如果 $\delta(q_0, x) \cap F \neq \emptyset$, 则称 x 被 M 接受, 如果 $\delta(q_0, x) \cap F = \emptyset$, 则称 M 不接受 x 。

↻ $L(M) = \{x \mid x \in \Sigma^* \text{ 且 } \delta(q_0, x) \cap F \neq \emptyset\}$, 称为由 M 接受(识别)的语言。



NFA与DFA等价

- 对于一个输入字符，**NFA**与**DFA**的差异是前者可以进入若干个状态，而后者只能进入一个惟一的状态。虽然从**DFA**看待问题的角度来说，**NFA**在某一时刻同时进入若干个状态，但是，这若干个状态合在一起的“总效果”相当于它处于这些状态对应的一个“综合状态”。因此，我们考虑让**DFA**用一个状态去对应**NFA**的一组状态。



NFA与DFA等价

- NFA $M_1=(Q, \Sigma, \delta_1, q_0, F_1)$ 与DFA $M_2=(Q_2, \Sigma, \delta_2, q_0', F_2)$ 的对应关系:
 - ∞ NFA从开始状态 q_0 启动, 我们就让相应的DFA从状态 $[q_0]$ 启动。所以 $q_0' = [q_0]$ 。
 - ∞ 对于NFA 的一个状态组 $\{q_1, q_2, \dots, q_n\}$, 如果NFA在此状态组时读入字符 a 后可以进入状态组 $\{p_1, p_2, \dots, p_m\}$, 则让相应的DFA在状态 $[q_1, q_2, \dots, q_n]$ 读入字符 a 时, 进入状态 $[p_1, p_2, \dots, p_m]$ 。



NFA与DFA等价

定理 NFA与DFA等价。

证明：

(1) 构造与NFA M_1 等价的DFA M_2 。

$$M_1=(Q, \Sigma, \delta_1, q_0, F_1)$$

$$M_2=(Q_2, \Sigma, \delta_2, [q_0], F_2)$$

$$Q_2=2^Q$$

$$F_2=\{[p_1, p_2, \dots, p_m] \mid \{p_1, p_2, \dots, p_m\} \subseteq Q \& \{p_1, p_2, \dots, p_m\} \cap F_1 \neq \Phi\}$$



NFA与DFA等价

$$\delta_2([q_1, q_2, \dots, q_n], a) = [p_1, p_2, \dots, p_m]$$
$$\Leftrightarrow \delta_1(\{q_1, q_2, \dots, q_n\}, a) = \{p_1, p_2, \dots, p_m\}$$

(2) 证明 $\delta_1(q_0, x) = \{p_1, p_2, \dots, p_m\}$
 $\Leftrightarrow \delta_2([q_0], x) = [p_1, p_2, \dots, p_m]$ 。

设 $x \in \Sigma^*$, 施归纳于 $|x|$

$$x = \varepsilon, \quad \delta_1(q_0, \varepsilon) = \{q_0\}, \quad \delta_2([q_0], \varepsilon) = [q_0]$$

NFA与DFA等价

设当 $|x|=n$ 是结论成立。下面证明当 $|x|=n+1$ 时结论也成立。不妨设 $x=wa$, $|w|=n$, $a \in \Sigma$

$$\begin{aligned}\delta_1(q_0, wa) &= \delta_1(\delta_1(q_0, w), a) \\ &= \delta_1(\{q_1, q_2, \dots, q_n\}, a) \\ &= \{p_1, p_2, \dots, p_m\}\end{aligned}$$

由归纳假设,

$$\begin{aligned}\delta_1(q_0, w) &= \{q_1, q_2, \dots, q_n\} \Leftrightarrow \delta_2([q_0], \\ w) &= [q_1, q_2, \dots, q_n]\end{aligned}$$

NFA与DFA等价

根据 δ_2 的定义,

$$\delta_2([q_1, q_2, \dots, q_n], a) = [p_1, p_2, \dots, p_m] \Leftrightarrow \delta_1(\{q_1, q_2, \dots, q_n\}, a) = \{p_1, p_2, \dots, p_m\}$$

所以,

$$\begin{aligned} \delta_2([q_0], wa) &= \delta_2(\delta_2([q_0], w), a) \\ &= \delta_2([q_1, q_2, \dots, q_n], a) \\ &= [p_1, p_2, \dots, p_m] \end{aligned}$$



NFA与DFA等价

故，如果 $\delta_1(q_0, wa) = \{p_1, p_2, \dots, p_m\}$ 则必有 $\delta_2([q_0], wa) = [p_1, p_2, \dots, p_m]$ 。

由上述推导可知，反向的推导也成立。这就是说，结论对 $|x|=n+1$ 也成立。

由归纳法原理，结论对 $\forall x \in \Sigma^*$ 成立。



NFA与DFA等价

(3) 证明 $L(M_1)=L(M_2)$

设 $x \in L(M_1)$, 且 $\delta_1(q_0, x) = \{p_1, p_2, \dots, p_m\}$,

从而 $\delta_1(q_0, x) \cap F_1 \neq \Phi$,

这就是说, $\{p_1, p_2, \dots, p_m\} \cap F_1 \neq \Phi$,

由 F_2 的定义, $[p_1, p_2, \dots, p_m] \in F_2$ 。



NFA与DFA等价

再由(2)知,

$$\delta_2([q_0], x) = [p_1, p_2, \dots, p_m]$$

所以, $x \in L(M_2)$ 。故 $L(M_1) \subseteq L(M_2)$ 。

反过来推, 可得 $L(M_2) \subseteq L(M_1)$ 。

从而 $L(M_1) = L(M_2)$ 得证。

综上所述, 定理成立。



NFA与DFA等价

例： 下图所示的**NFA** 对应的**DFA**的状态转移函数表中所示。

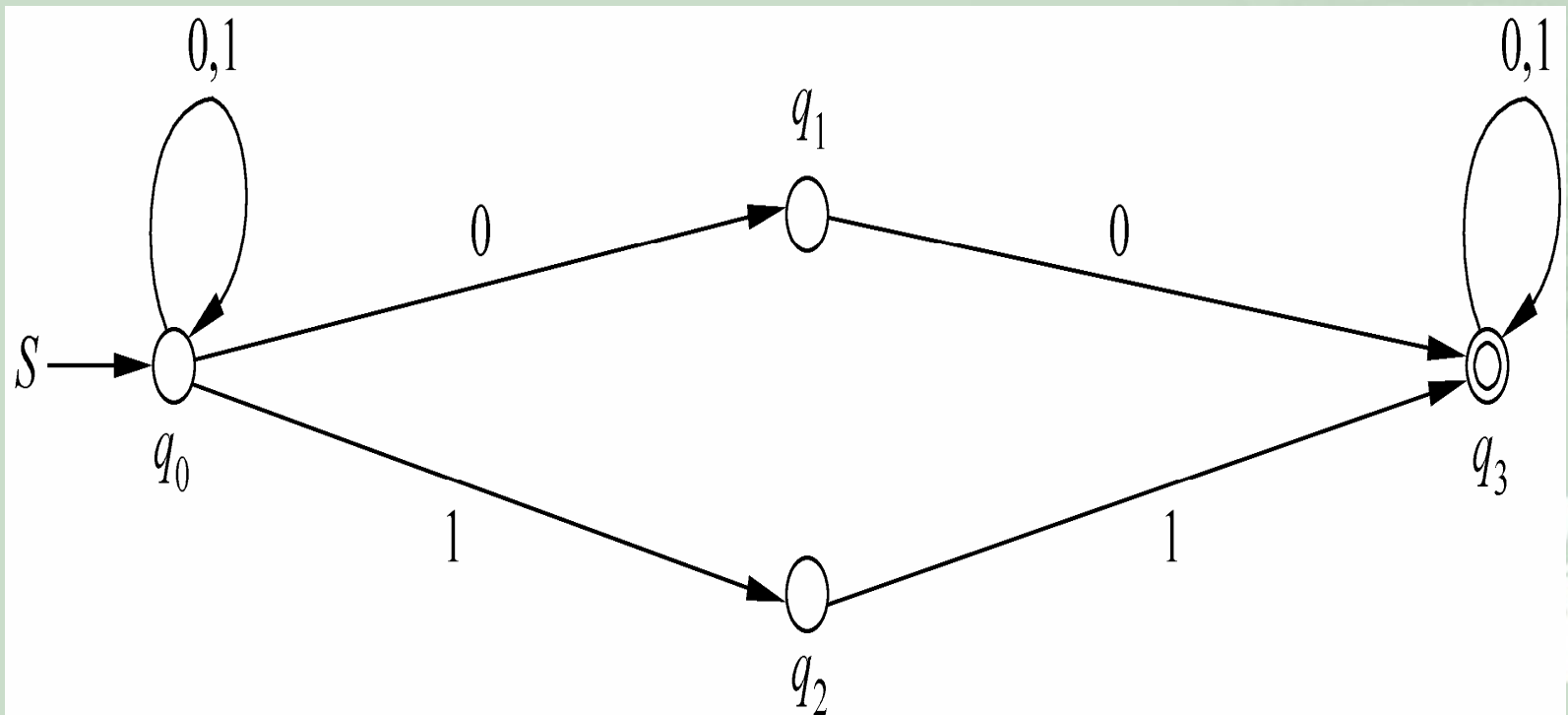


表 状态转移函数

状态说明		状态	输入字符	
			0	1
启动	√	$[q_0]$	$[q_0, q_1]$	$[q_0, q_2]$
		$[q_1]$	$[q_3]$	$[\Phi]$
		$[q_2]$	$[\Phi]$	$[q_3]$
终止		$[q_3]$	$[q_3]$	$[q_3]$
	√	$[q_0, q_1]$	$[q_0, q_1, q_3]$	$[q_0, q_2]$
	√	$[q_0, q_2]$	$[q_0, q_1]$	$[q_0, q_2, q_3]$
终止		$[q_0, q_3]$	$[q_0, q_1, q_3]$	$[q_0, q_2, q_3]$
		$[q_1, q_2]$	$[q_3]$	$[q_3]$
终止		$[q_1, q_3]$	$[q_3]$	$[q_3]$
终止		$[q_2, q_3]$	$[q_3]$	$[q_3]$
		$[q_0, q_1, q_2]$	$[q_0, q_1, q_3]$	$[q_0, q_2, q_3]$
终止	√	$[q_0, q_1, q_3]$	$[q_0, q_1, q_3]$	$[q_0, q_2, q_3]$
终止	√	$[q_0, q_2, q_3]$	$[q_0, q_1, q_3]$	$[q_0, q_2, q_3]$
终止		$[q_1, q_2, q_3]$	$[q_3]$	$[q_3]$
终止		$[q_0, q_1, q_2, q_3]$	$[q_0, q_1, q_3]$	$[q_0, q_2, q_3]$
		$[\Phi]$	$[\Phi]$	$[\Phi]$

NFA与DFA等价

- **不可达状态** (inaccessible state): 不存在从 $[q_0]$ 对应的顶点出发, 到达该状态对应的顶点的路。我们称此状态从开始状态是不可达的。
- 上表中, 所有标记“√”的状态是从开始状态可达的, 其他是不可达的——无用的。



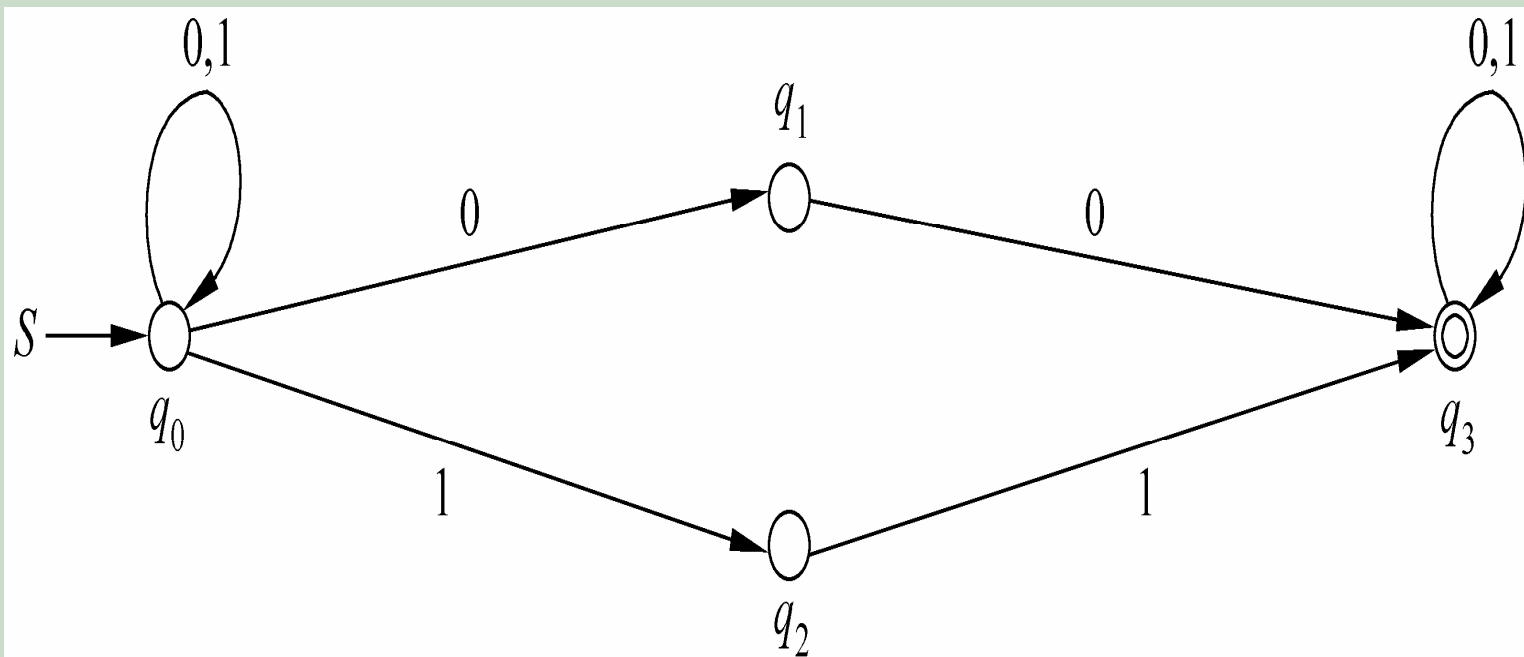
NFA与DFA等价

■ 构造给定NFA等价的DFA策略

- ∞ 先只把开始状态 $[q_0]$ 填入表的状态列中，如果表中所列的状态列有未处理的，则任选一个未处理的状态 $[q_1, q_2, \dots, q_n]$ ，对 Σ 中的每个字符 a ，计算 $\delta([q_1, q_2, \dots, q_n], a)$ ，并填入相应的表项中，如果 $\delta([q_1, q_2, \dots, q_n], a)$ 在表的状态列未出现过，则将它填入表的状态列。如此重复下去，直到表的状态列中不存在未处理的状态。



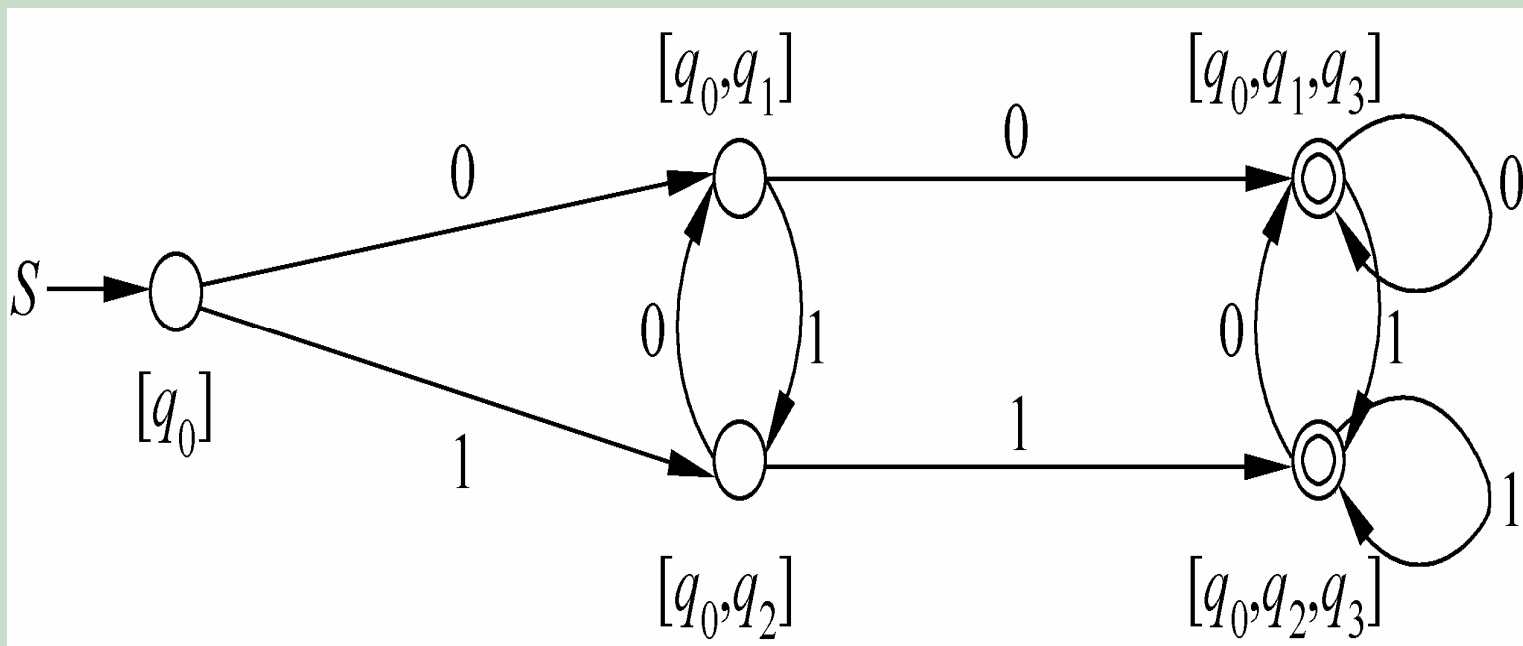
NFA与DFA等价



NFA



NFA与DFA等价

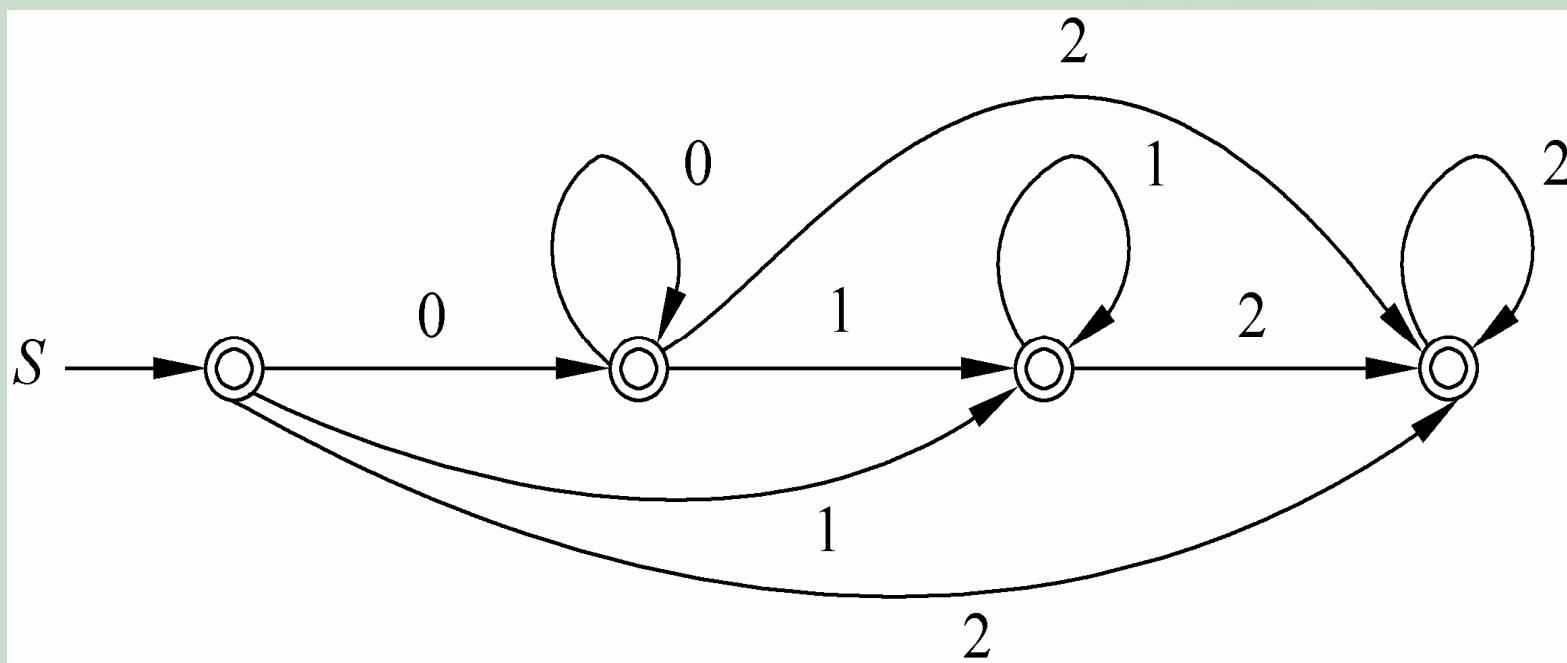


上图所示NFA的等价DFA



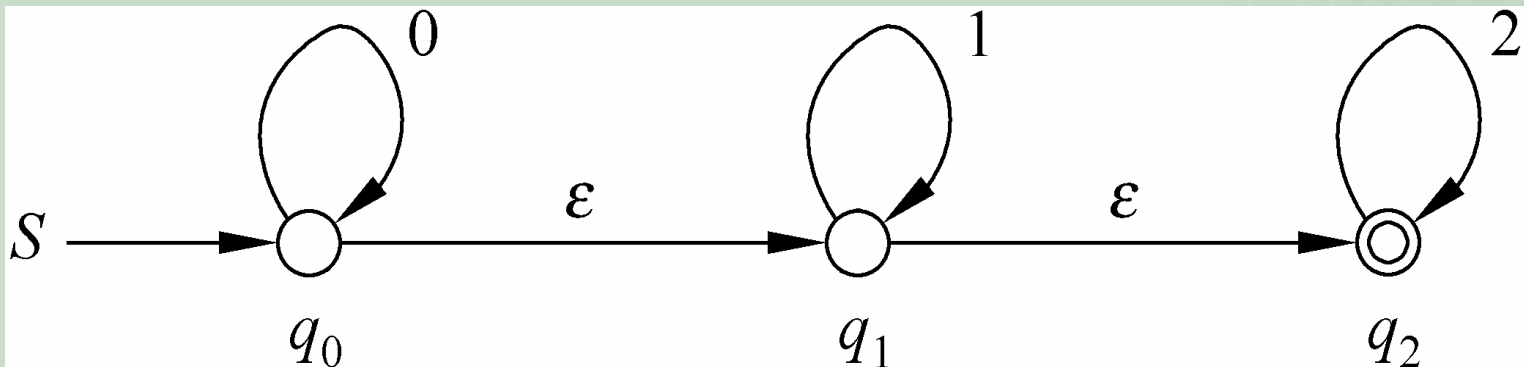
带空移动的有穷状态自动机

- 接受语言 $\{0^n 1^m 2^k \mid n, m, k \geq 0\}$ 的 NFA。



带空移动的有穷状态自动机

- 接受语言 $\{0^n 1^m 2^k \mid n, m, k \geq 0\}$ 的 NFA 是否可以构造成下图所示的“ ϵ -NFA”?



其构造显然比上图所示的 NFA 更容易。当然还希望它们是等价的。

带空移动的有穷状态自动机

- 带空移动的不确定的有穷状态自动机(non-deterministic finite automaton with ε -moves, ε -NFA)

$M=(Q, \Sigma, \delta, q_0, F),$

Q 、 Σ 、 q_0 、 F 的意义同DFA,

$\delta : Q \times (\Sigma \cup \{ \varepsilon \}) \rightarrow 2^Q$ 。



带空移动的有穷状态自动机

■ 非空移动

∞ $\forall (q, a) \in Q \times \Sigma$

∞ $\delta(q, a) = \{p_1, p_2, \dots, p_m\}$

∞ 表示M在状态q读入字符a，可以选择地将状态变成 p_1 、 p_2 、...或者 p_m ，并将读头向右移动一个带方格而指向输入字符串的下一个字符。



带空移动的有穷状态自动机

■ 空移动

∞ $\forall q \in Q$

∞ $\delta(q, \varepsilon) = \{p_1, p_2, \dots, p_m\}$

∞ 表示M在状态q不读入任何字符，可以选择地将状态变成 p_1 、 p_2 、...或者 p_m 。也可以叫做M在状态q做一个空移动(又可以称为 ε 移动)，并且选择地将状态变成 p_1 、 p_2 、...或者 p_m 。



带空移动的有穷状态自动机

- 进一步扩充 δ 的定义域:

$$\hat{\delta} : Q \times \Sigma^* \rightarrow 2^Q.$$

对任意的 $P \subseteq Q$, $q \in Q$, $w \in \Sigma^*$, $a \in \Sigma$:

(1) ε -CLOSURE(q) = $\{p \mid \text{从 } q \text{ 到 } p \text{ 有一条标记为 } \varepsilon \text{ 的路}\}$ 。

即代表从 q 开始, 扫描一个或多个 ε 后能够到达的状态的集合, 或从 q 接收空串 ε 后能够到达的状态集。



带空移动的有穷状态自动机

ε -CLOSURE(q)是由下述规则确定的状态集:

1. $q \in \varepsilon$ -CLOSURE(q); 任意状态 q 接收空串 ε , 至少都能保持状态 q 不变;
2. 如果 $p \in \varepsilon$ -CLOSURE(q), 那么若 $r \in \delta(p, \varepsilon)$, 则 $r \in \varepsilon$ -CLOSURE(q);
3. 重复2, 直到 ε -CLOSURE(q)中状态不再增加为止。



带空移动的有穷状态自动机

$$(2) \varepsilon - CLOSURE (P) = \bigcup_{p \in P} \varepsilon - CLOSURE (p)$$

如果 $P = \{p_1, p_2, p_3, \dots, p_m\}$, 则

$\varepsilon - CLOSURE(P)$

$= \varepsilon - CLOSURE(p_1) \cup$

$\varepsilon - CLOSURE(p_2) \cup$

$\dots \cup$

$\varepsilon - CLOSURE(p_m)$



带空移动的有穷状态自动机

$$(3) \hat{\delta}(q, \varepsilon) = \varepsilon - CLOSURE(q)$$

$$(4) \hat{\delta}(q, wa) = \varepsilon - CLOSURE(P)$$

$$\begin{aligned} P &= \{p \mid \exists r \in \hat{\delta}(q, w) \text{ 使得 } p \in \delta(r, a)\} \\ &= \bigcup_{r \in \hat{\delta}(q, w)} \delta(r, a) \end{aligned}$$



带空移动的有穷状态自动机

- 进一步扩展移动函数： $2^Q \times \Sigma \rightarrow 2^Q$ 。

对任意 $(P, a) \in 2^Q \times \Sigma$ 。

$$(5) \delta(P, a) = \bigcup_{q \in P} \delta(q, a)$$

$$(6) \hat{\delta}(P, w) = \bigcup_{q \in P} \hat{\delta}(q, w)$$



带空移动的有穷状态自动机

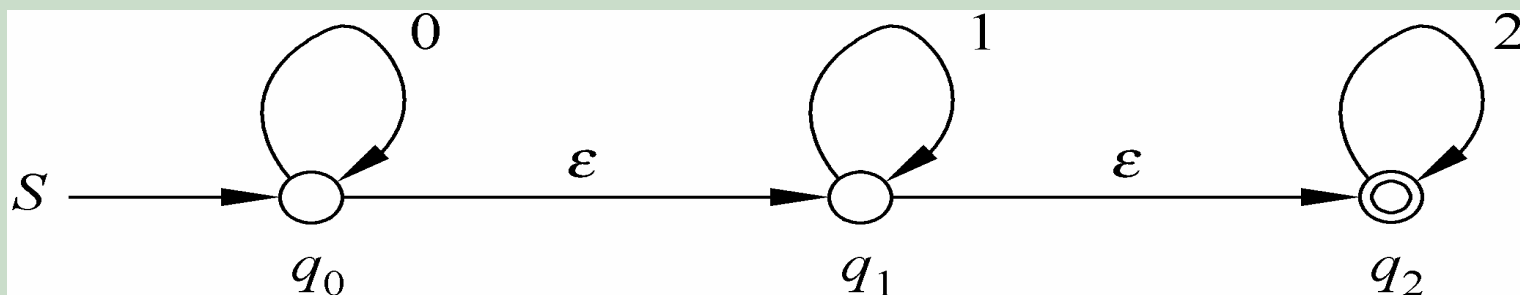
- 在 ε -NFA 中，对任意 $a \in \Sigma$ ， $q \in Q$ ，

$$\hat{\delta}(q, a) \neq \delta(q, a)$$

需要严格区分。



带空移动的有穷状态自动机



状态	δ				$\hat{\delta}$			
	ϵ	0	1	2	ϵ	0	1	2
q_0	$\{q_1\}$	$\{q_0\}$	Φ	Φ	$\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$	$\{q_1, q_2\}$	$\{q_2\}$
q_1	$\{q_2\}$	Φ	$\{q_1\}$	Φ	$\{q_1, q_2\}$	Φ	$\{q_1, q_2\}$	$\{q_2\}$
q_2	Φ	Φ	Φ	$\{q_2\}$	$\{q_2\}$	Φ	Φ	$\{q_2\}$

带空移动的有穷状态自动机

- **M**接受(识别)的语言
对于 $\forall \mathbf{x} \in \Sigma^*$, 仅当

$$\hat{\delta}(q_0, x) \cap F \neq \Phi$$

时, 称 \mathbf{x} 被**M**接受。

$$L(M) = \{x \mid x \in \Sigma^* \text{ 并且 } \hat{\delta}(q_0, x) \cap F \neq \Phi\}$$



带空移动的有穷状态自动机

定理 ε -NFA与NFA等价。

证明：设有 ε -NFA $M_1=(Q, \Sigma, \delta_1, q_0, F)$

(1) 构造与之等价的NFA M_2 。

取NFA $M_2=(Q, \Sigma, \delta_2, q_0, F_2)$

$$F_2 = \begin{cases} F \cup \{q_0\} & \text{如果 } F \cap \varepsilon\text{-CLOSURE}(q_0) \neq \emptyset \\ F & \text{如果 } F \cap \varepsilon\text{-CLOSURE}(q_0) = \emptyset \end{cases}$$

$$\delta_2(q, a) = \hat{\delta}_1(q, a)$$



带空移动的有穷状态自动机

(2) 施归纳于 $|x|$, 证明对 $\forall x \in \Sigma^+$

$$\delta_2(q_0, x) = \hat{\delta}_1(q_0, x)$$

$$\delta_2(q_0, x) = \delta_2(q_0, wa)$$

$$= \delta_2(\delta_2(q_0, w), a)$$

$$= \delta_2(\hat{\delta}_1(q_0, w), a)$$



带空移动的有穷状态自动机

$$= \bigcup_{q \in \hat{\delta}_1(q_0, w)} \delta_2(q, a)$$

$$= \bigcup_{q \in \hat{\delta}_1(q_0, w)} \hat{\delta}_1(q, a)$$

$$= \varepsilon - \text{CLOSURE} \left(\bigcup_{q \in \hat{\delta}_1(q_0, w)} \delta_1(q, a) \right)$$

$$= \varepsilon - \text{CLOSURE} (\{ p \mid \exists q \in \hat{\delta}_1(q_0, w), p \in \delta_1(q, a) \})$$

$$= \hat{\delta}_1(q_0, wa)$$

$$= \hat{\delta}_1(q_0, x)$$



带空移动的有穷状态自动机

(3) 证明, 对 $\forall \mathbf{x} \in \Sigma^+$, $\delta_2(\mathbf{q}_0, \mathbf{x}) \cap F_2 \neq \Phi$ 的充分必要条件是

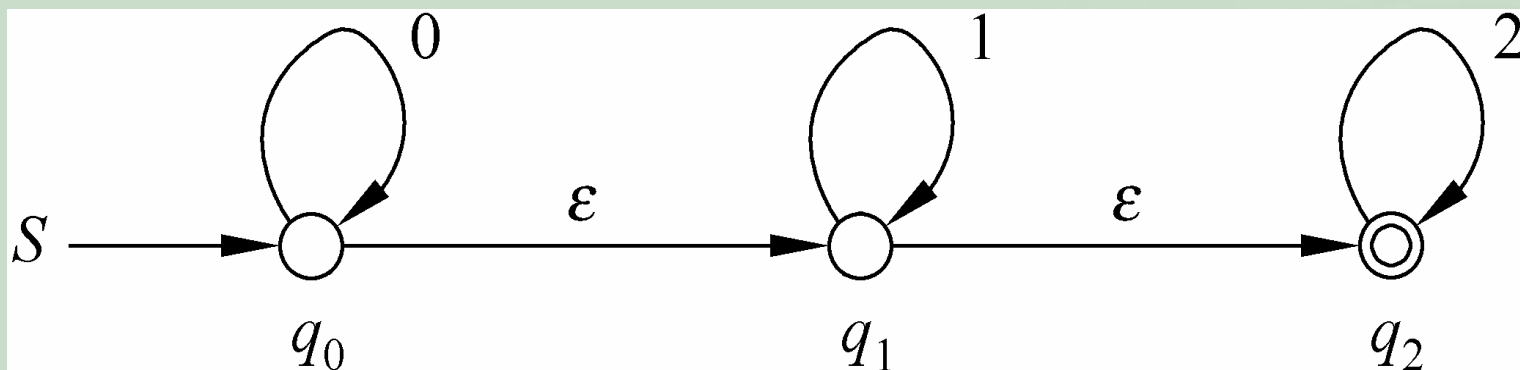
$$\hat{\delta}_1(q_0, x) \cap F \neq \Phi$$

(4) 证明 $\varepsilon \in L(M_1) \Leftrightarrow \varepsilon \in L(M_2)$ 。

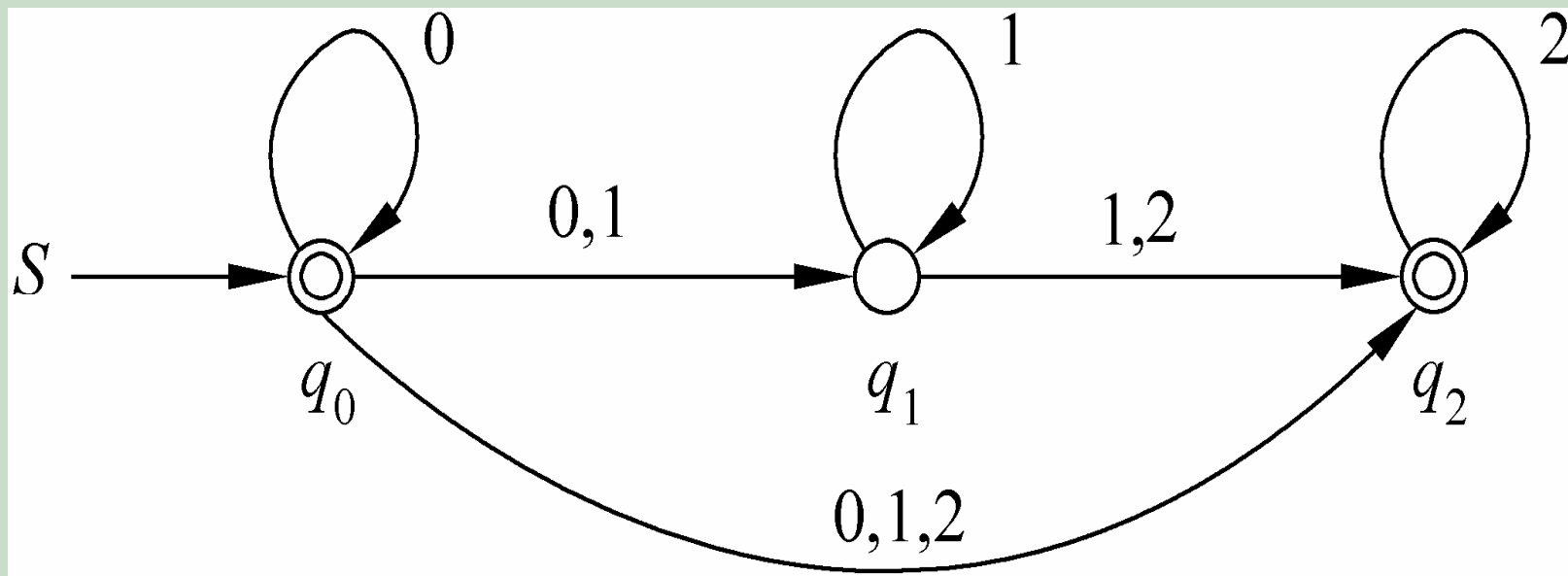


带空移动的有穷状态自动机

- 例：求与下图所示 ε -NFA 等价的 NFA。



带空移动的有穷状态自动机

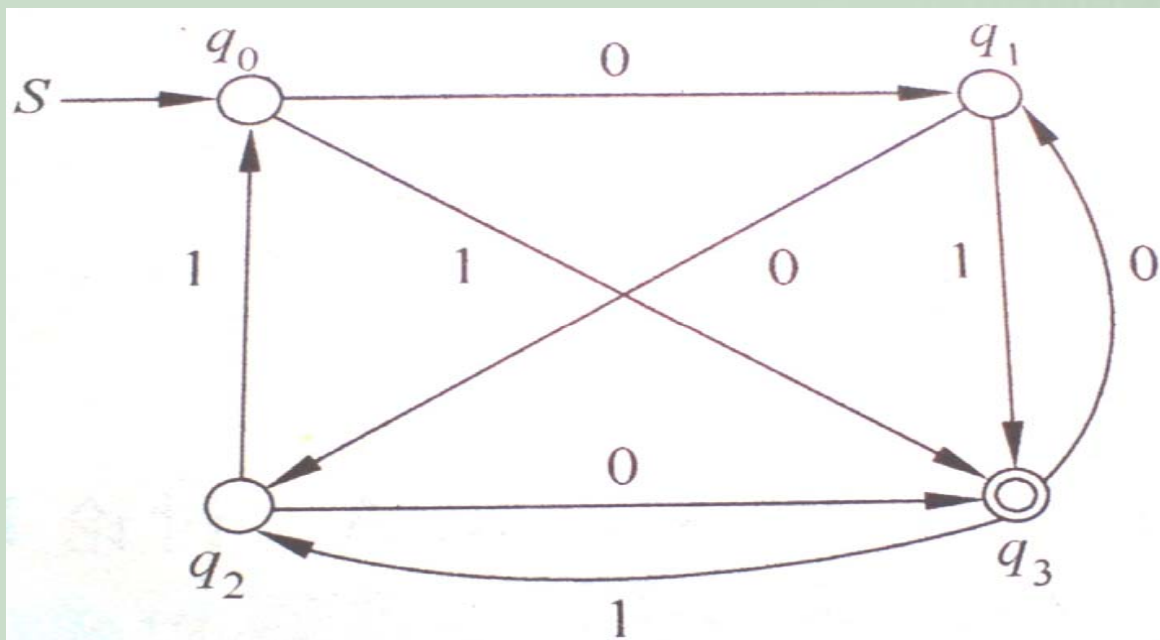


与上图所示 ϵ -NFA 等价的 NFA



习题

- 1 已知DFA M 如下图所示，请给出它的形式描述，并写出它在处理字符串1011001的过程中经过的ID转换序列。



习题

2 构造识别下列语言的DFA。

(1) $\{0, 1\}^*$ 。

(2) $\{x \mid x \in \{0, 1\}^+ \text{ 且 } x \text{ 中不含形如 } 00 \text{ 的子串}\}$ 。

(3) $\{x \mid x \in \{0, 1\}^+ \text{ 且 } x \text{ 中含形如 } 110 \text{ 的子串}\}$ 。

(4) $\{x \mid x \in \{0, 1\}^+ \text{ 且当把 } x \text{ 看作二进制数时, } x \text{ 模 } 5 \text{ 与 } 3 \text{ 同余}\}$ 。

(5) $\{x \mid x \in \{0, 1\}^+ \text{ 且 } x \text{ 以 } 1 \text{ 开头以 } 0 \text{ 结尾}\}$ 。

(6) $\{x \mid x \in \{0, 1\}^+ \text{ 且若 } x \text{ 以 } 0 \text{ 结尾则其长度为偶数; 若 } x \text{ 以 } 1 \text{ 结尾则其长度为奇数}\}$ 。



习题

- 3 证明：对于任意的DFA $M_1 = (Q, \Sigma, \delta, q_0, F_1)$ ，存在DFA $M_2 = (Q, \Sigma, \delta, q_0, F_2)$ ，使得 $L(M_2) = \Sigma^* - L(M_1)$ 。
- 4 构造识别下列语言的NFA。
- (1) $\{x \mid x \in \{0, 1\}^+ \text{ 且 } x \text{ 中不含形如 } 00 \text{ 的子串}\}$ 。
 - (2) $\{x \mid x \in \{0, 1\}^+ \text{ 且 } x \text{ 中含形如 } 110 \text{ 的子串}\}$ 。
 - (3) $\{x \mid x \in \{0, 1\}^+ \text{ 且 } x \text{ 以 } 1 \text{ 开头以 } 0 \text{ 结尾}\}$ 。
 - (4) $\{x \mid x \in \{0, 1\}^+ \text{ 且若 } x \text{ 以 } 0 \text{ 结尾则其长度为偶数；若 } x \text{ 以 } 1 \text{ 结尾则其长度为奇数}\}$ 。



习题

5 根据给定的NFA，构造与之定价的DFA。

	状态	输入字符		
		0	1	2
开始	q_0	$\{q_0, q_1\}$	$\{q_0, q_2\}$	$\{q_0, q_2\}$
	q_1	$\{q_3, q_0\}$	Φ	$\{q_2\}$
	q_2	Φ	$\{q_3, q_1\}$	$\{q_2, q_1\}$
终止	q_3	$\{q_3, q_2\}$	$\{q_3\}$	$\{q_0\}$



习题

6 根据给定的 ε -NFA，构造与之等价的NFA。

	状态	输入字符		
		0	1	ε
开始	q_0	$\{q_0, q_1\}$	$\{q_0, q_2\}$	$\{q_0, q_2\}$
	q_1	$\{q_3, q_0\}$	Φ	$\{q_2\}$
	q_2	Φ	$\{q_3, q_1\}$	$\{q_2, q_1\}$
终止	q_3	$\{q_3, q_2\}$	$\{q_3\}$	$\{q_0\}$



习题

7 证明：对于任意的FA $M_1 = (Q_1, \Sigma_1, \delta_1, q_{01}, F_1)$ ，
FA $M_2 = (Q_2, \Sigma_2, \delta_2, q_{02}, F_2)$ ，存在FA M ，使得
 $L(M) = L(M_1) \cup L(M_2)$ 。

