形式语言与自动机期末速通

3. DFA与NFA

3.1 DFA的形式定义

[定义3.1.1] 有穷状态自动机(Finite Automaton, FA)由五元组(Q,Σ,δ,q_0,F)定义,即$M=(Q,\Sigma,\delta,q_0,F)$,其中:

(1)Q为状态的非空有穷集,$\forall q\in Q$称为M的一个**状态**.

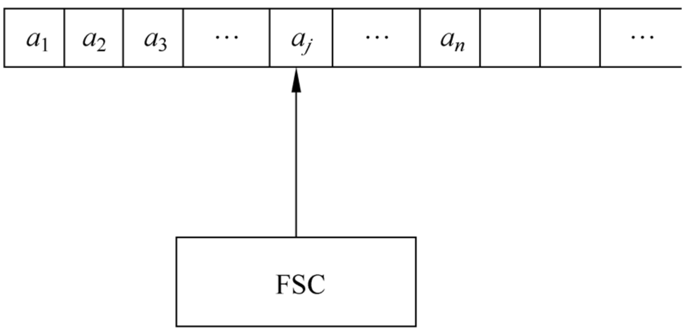
(2)Σ为**输入字母表**,输入的字符串都是Σ上的字符串.

(3)δ为**状态转移函数**,又称**状态转换函数**或**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$a$,将状态变为$p$,并将读头右移一个带方格,指向输入字符串的下一个字符.

(4)q_0为M的**开始状态**,又称**初始状态**、**启动状态**.$q_0\in Q$.

(5)F为终止状态的集合,$F\subseteq Q$.$\forall q\in F$称为M的**终止状态**或**接受状态**.

[注1] FA的物理模型如下图,包含一个右端无穷的输入带、一个有穷状态控制器(Finite State Control, FSC)和一个读头.



[注2] 将δ扩充为$\hat{\delta}:Q\times\Sigma^*\rightarrow Q$.对$\forall q\in Q,w\in\Sigma^*,a\in\Sigma$,定义:

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

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

因$\hat{\delta}(q,a)=\hat{\delta}(q,\varepsilon a)=\delta(\hat{\delta}(q,\varepsilon),a)=\delta(q,a)$,故不区分$\delta$和$\hat{\delta}$.

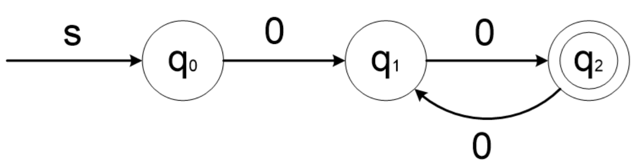
[注3] FA的状态有记忆功能,即不同的状态对应于不同的情况.因FA只有有穷个状态,故在识别语言的过程中,若有无穷种情况需要记忆,则无法构造出相应的FA.

[例3.1.1]

(1)下图是一个FA $M_1=(\{q_0,q_1,q_2\},\{0\},\delta_1,q_0,\{q_2\})$,

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

状态转移图:



(2)$M_2=(\{q_0,q_1,q_2,q_3\},\{0,1,2\},\delta_2,q_0,\{q_2\})$是一个FA,

其中$\delta_2(q_0,0)=q_1,\delta_2(q_1,0)=q_2,\delta_2(q_2,0)=q_1,\delta_2(q_3,0)=q_3,$

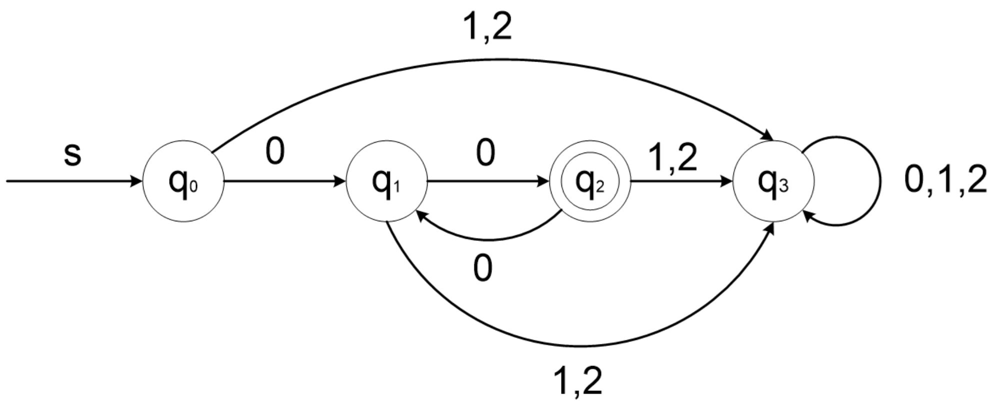
$\delta_2(q_0,1)=q_3,\delta_2(q_1,1)=q_3,\delta_2(q_2,1)=q_3,\delta_2(q_3,1)=q_3,$

$\delta_2(q_0,2)=q_3,\delta_2(q_1,2)=q_3,\delta_2(q_2,2)=q_3,\delta_2(q_3,2)=q_3.$

状态转移表:

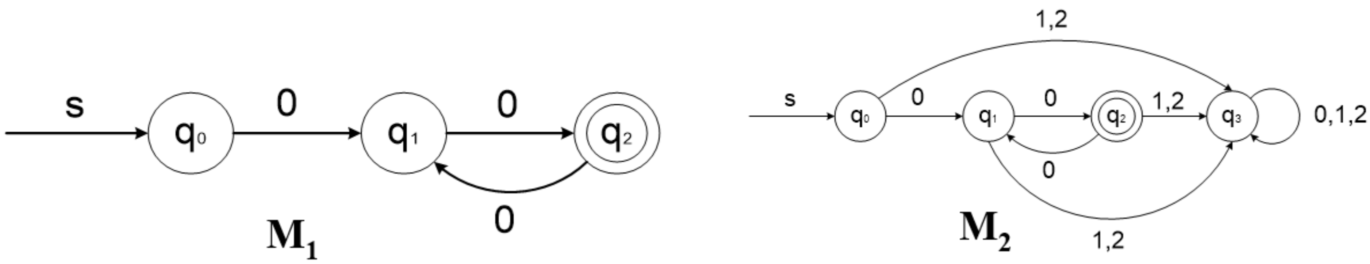
状态说明	状态	输入字符		
		0	1	2
开始状态	q_0	q_1	q_3	q_3
	q_1	q_2	q_3	q_3
终止状态	q_2	q_1	q_3	q_3
	q_3	q_3	q_3	q_3

状态转移图:



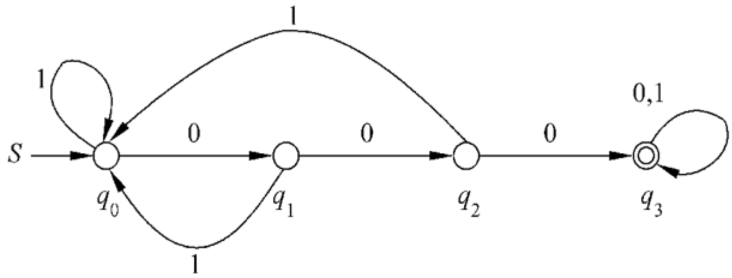
[**定义3.1.2**] 设FA $M=(Q,\Sigma,\delta,q_0,F)$.对$\forall w\in\Sigma^*$,若$\delta(q,w)\in F$,则称$M$**接受**串$w$,否则称$M$**不接受**$w$.集合$L(M)=\{w\mid w\in\Sigma^*\wedge\delta(q,w)\in F\}$称为$M$**接受的语言**或**识别的语言**.称FA M_1与FA M_2等价,如果$L(M_1)=L(M_2)$.

[**例3.1.2**] 下图所示的两个FA等价,它们识别的语言都是$\{0^{2n}\mid n\geq 1\}$.



[**定义3.1.3**] 设FA $M=(Q,\Sigma,\delta,q_0,F)$.若对$\forall q\in Q,a\in\Sigma$,$\delta(q,a)$都有确定的值,则称这样的FA为确定的**有穷状态自动机**(Deterministic Finite Automaton, DFA).

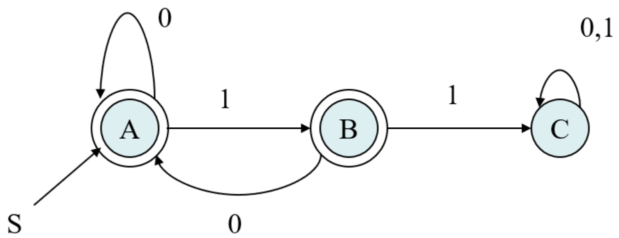
[**注1**] 下图是一个DFA.



[**注2**] 一个DFA可有多个接受状态.

[**注3**] 串x被DFA M接受的充要条件是:在M的状态转移图中,存在一条从起始状态S到某一终止状态的路径,该路径从第一条边到最后一条边的标记依次并置构成字符串x,简称该路径标记为x.

[**例3.1.3**] 指出并证明下面的DFA识别的语言.



[**解**] $T=\{w\mid w\in\{0,1\}^*\wedge w\text{不含连续的两个}1\}$,下面证明该DFA识别的语言$S$与$T$等价.

(1)下证$S\subseteq T$,即证若串w能被该DFA识别,则w不含连续的两个1.

设①若$\delta(A,w)=A$,则w无连续的两个1且不以1结尾;②$\delta(A,w)=B$,则w无连续的两个1且以单个1结尾.

下面对w的长度归纳,证明①和②.

(i)$|w|=0$,即w为空串时,①和②显成立.

(ii)对串x,设$|x|=n$时①和②成立.

(iii)下证串$w=xa$,即$|w|=n+1$时,①和②成立.

i)若$\delta(A,w)=A$,因$w=xa$,由DFA的定义:$\delta(A,x)$为A或B,且$a=0$.

则x不包含子串11,进而w不包含子串11,也不以1结尾,①成立.

ii)若$\delta(A,w)=B$,因$w=xa$,由DFA的定义:$\delta(A,x)$为A或B,且$a=1$.

则x不包含子串11且不以1结尾,进而w不包含子串11且以单个1结尾,②成立.

(2)下证$T\subseteq S$,即证若w不包含子串11,则w被该DFA接受.

考虑逆否命题,即证若w不被该DFA接受,则它包含子串11.

由状态转移图:w不被该DFA接受的充要条件是:$\delta(A,w)=C$,

则w有形式$w=x1y$,即前缀x引导DFA到状态B,字符1引导DFA到状态C,y是到达状态C后的后缀.

若$\delta(A,x)=B$,由(1):x不包含子串11且以单个1结尾,即x有形式$x=z1$,

进而$w=z11y$包含子串11,故证.

3.2 DFA的构造

[**例3.2.1**] 构造一个DFA,接受的语言为$\{x000y \mid x, y \in \{0,1\}^*\}$.

[**解**]

(1)考虑该DFA应包含哪些状态.

①q_0为该DFA的起始状态.

②q_1状态表示M读到了一个0,该0可能是子串000的第一个0.

③q_2状态表示M在q_1状态时又读到了一个0,该0可能是子串000的第二个0.

④q_3状态表示M在q_2状态时又读到了一个0,该0可能是子串000的第三个0,故该状态是接受状态.

(2)考虑状态转移.

①$\delta(q_0,1)=q_0$,表示M在q_0状态时读到了一个1,需继续在q_0状态等待可能是000的第一个0的输入.

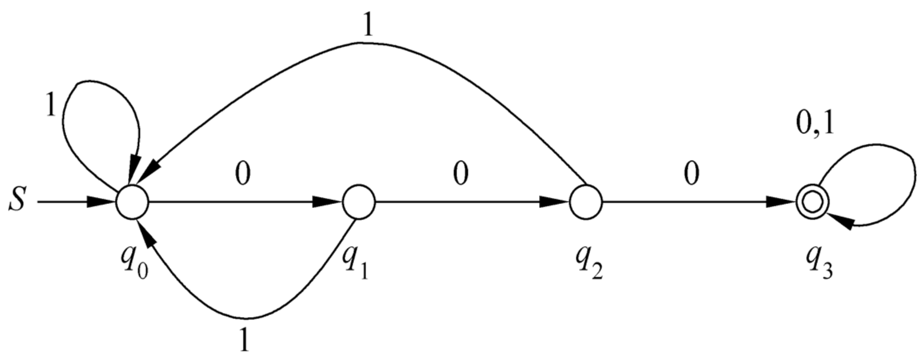
②$\delta(q_1,1)=q_0$,表示M刚读到一个0后又读到了一个1,这表明:之前读到的0并非000的第一个0,

故M回到q_0状态,等待可能是000的第一个0的输入.

③$\delta(q_2,1)=q_0$.

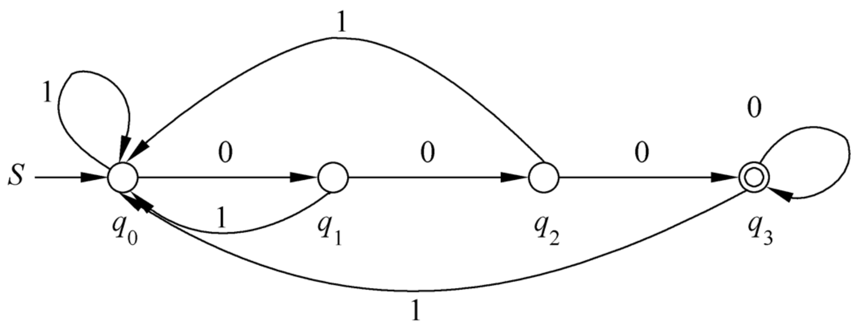
④$\delta(q_3,0)=q_3, \delta(q_3,1)=q_3$,表示$M$以找到子串000,继续读入该串的剩余部分.

(3)整理为状态转移表或状态转移图:



[**例3.2.2**] 构造一个接受的语言为$\{x000 \mid x \in \{0,1\}^*\}$的DFA.

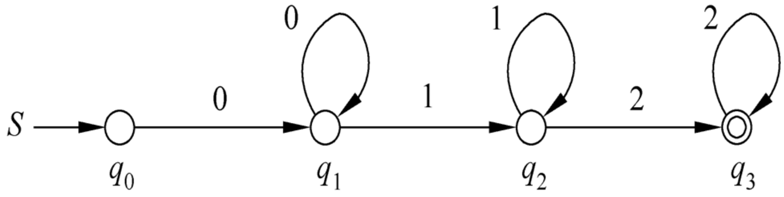
[**解**] 类似于**例3.2.1**,只需将q_3状态输入1的转移改为q_0即可:



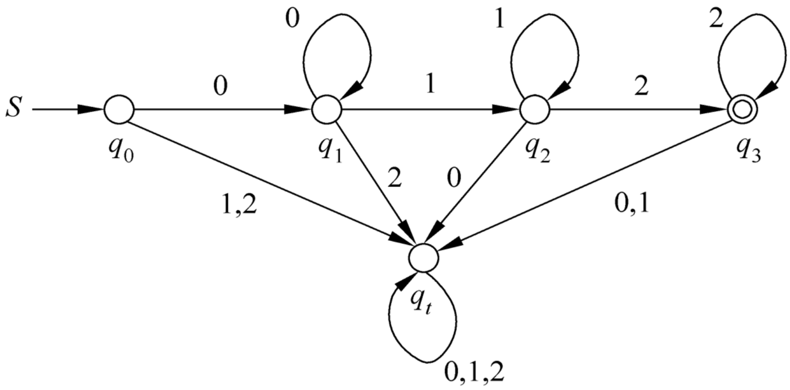
[**例3.2.3**] 构造一个接受的语言$\{0^n 1^m 2^k \mid n, m, k \geq 1\}$的DFA.

[**解**]

(1)先构建接受的串的形式的主体框架:



(2)将状态转移中错误的输入引导到陷阱状态q_t:



[**例3.2.4**] 构造一个接受的语言为$\{x \mid x \in \{0,1\}^*, \text{且将}x\text{视为二进制数时},x\text{是}3\text{的倍数}\}$的DFA.

[**解**]

(1)考虑按二进制数模3的余数分类,因DFA的开始状态可能与余数为0的状态不等价,故分开定义.

①q_0状态表示模3余0的二进制串的等价类.

②q_1状态表示模3余1的二进制串的等价类.

③q_2状态表示模3余2的二进制串的等价类.

④q_s为DFA的起始状态.

(2)考虑状态转移.

①q_s状态:

(i)读入0时,有$x=0$,则转移到q_0状态,即$\delta(q_s,0)=q_0$.

(ii)读入1时,有$x=1$,则转移到q_1状态,即$\delta(q_s,1)=q_1$.

②q_0状态,此时二进制数x可表示为$x=3k+0$:

(i)读入0时,有$x=2(3k+0)=6k+0$,则转移到q_0状态,即$\delta(q_0,0)=q_0$.

(ii)读入1时,有$x=2(3k+0)+1=6k+1$,则转移到q_1状态,即$\delta(q_0,1)=q_1$.

③q_1状态,此时二进制数x可表示为$x=3k+1$:

(i)读入0时,有$x=2(3k+1)=6k+2$,则$\delta(q_1,0)=q_2$.

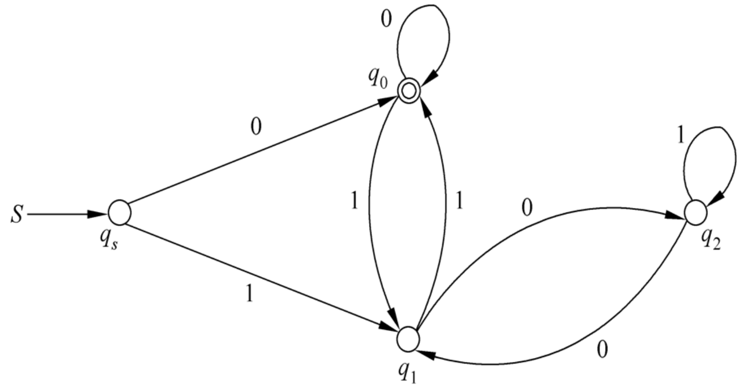
(ii)读入1时,有$x=2(3k+1)+1=3(2k+1)$,则$\delta(q_1,0)=q_0$.

④q_2状态,此时二进制数x可表示为$x=3k+2$:

(i)读入0时,有$x=2(3k+2)=3(2k+1)+1$,则$\delta(q_2,0)=q_1$.

(ii)读入1时,有$x=2(3k+2)+1=3*(2k+1)+2$,则$\delta(q_2,1)=q_2$.

(3)整理为状态转移图:



[**例3.2.5**] 构造一个接受的与语言为$\{x \mid x \in \{0,1\}^*, \text{且对}x\text{中任一长度不大于}5\text{的子串}a_1 \cdots a_n, \text{有}a_1 + \cdots + a_n \leq 3\ \ (n \leq 5)\}$的DFA.

[**解**] 下面过程类似于滑动窗口.

(1)考察该DFA应包含哪些状态.设输入串为$a_1 \cdots a_i \cdots a_{i+5} \cdots a_m$.

①$i=1,2,3$,即M读到输入串的第$1,2,3$个字符时,需记录这些字符,因为它们可能用于判定输入串最初的4或5个字符构成的子串是否满足要求.

②$i=4,5$,即M读到输入串的第$4,5$个字符时,

(i)若$a_1 + \cdots + a_i \leq 3$,则记录$a_1 \cdots a_i$.

(ii)若$a_1 + \cdots + a_i > 3$,则引导到陷阱状态q_t.

③$i=6$,即M读到输入串的第6个字符时,只需考察$a_2 + \cdots + a_6 \leq 3$是否成立,转移与②类似.

同理M读完子串$a_1 \cdots a_{i+4}$并发现它满足语言的要求时,M记录$a_1 \cdots a_{i+4}$,

此时M再读入输入串的第$(i+5)$个字符时,之前读到的第i个字符已无作用.

综上,该DFA需记录的内容有:

①未读入字符时的状态,有$2^0=1$种.

②记录了1个字符的状态,有$2^1=2$种.

③记录了2个字符的状态,有$2^2=4$种.

③记录了3个字符的状态,有$2^3=8$种.

⑤记录了4个字符的状态,有$2^4-1=15$种,其中-1是减去首位为1的1种非法状态,下同.

⑥记录了5个字符的状态,有$2^5-6=26$种.

⑦当前输入串非句子的状态,1种.

故可设置下列状态:

①q_ε表示DFA未读入字符的状态.

②q_t为DFA的陷阱状态.

③$q_{a_1 \cdots a_i}$表示DFA记录i个字符的状态,其中$a_1, \cdots, a_i \in \{0,1\}\ \ (1 \leq i \leq 5)$.

(2)设DFA $M=\{Q,\{0,1\},\delta,q_\varepsilon,F\}$,

其中$F=\{q_\varepsilon\} \bigcup \{q_{a_1 \cdots a_i} \mid a_1, \cdots, a_i \in \{0,1\}\ \ (1 \leq i \leq 5) \wedge a_1 + \cdots + a_i \leq 3\}$,$Q=\{q_t\} \bigcup F$,

状态转移函数:

①$\delta(q_\varepsilon, a_1) = q_{a_1}$.

②$\delta(q_{a_1}, a_2) = q_{a_1a_2}$.

③$\delta(q_{a_1a_2}, a_3) = q_{a_1a_2a_3}$.

④$\delta(q_{a_1a_2a_3}, a) = \begin{cases} q_{a_1a_2a_3a} & if\ a_1+a_2+a_3+a\le 3 \\ q_t & otherwise \end{cases}$.

⑤$\delta(q_{q_1q_2q_3q_4}, a) = \begin{cases} q_{a_1a_2a_3a_4a} & if\ a_1+a_2+a_3+a_4+a\le 3 \\ q_t & otherwise \end{cases}$.

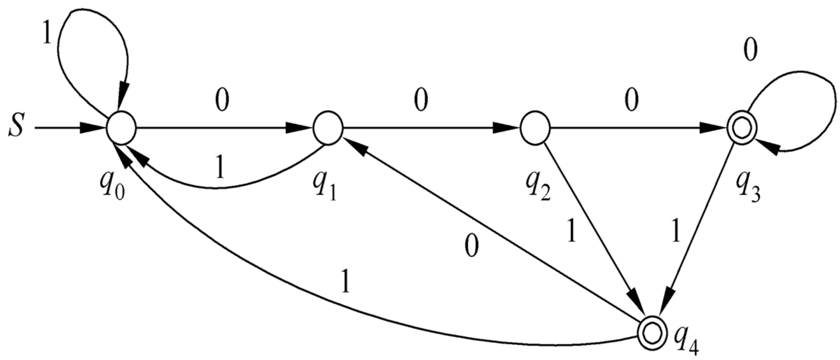
⑥$\delta(q_{a_1a_2a_3a_4a_5}, a) = \begin{cases} q_{a_2a_3a_4a_5a} & if\ a_2+a_3+a_4+a_5+a\le 3 \\ q_t & otherwise \end{cases}$.

⑦$\delta(q_t, a) = q_t$.

[**例3.2.6**] (1)构造接受的语言为$\{x000 \mid x\in\{0,1\}^*\}\bigcup\{x0001 \mid x\in\{0,1\}^*\}$的DFA

(2)设能引导FA到状态q的字符串的集合为$set(q)=\{x\mid x\in\Sigma^*,\delta(q_0,x)=q\}$.对上述DFA的所有$q$,求$set(q)$.

[**解**] (1)



(2)$set(q_0)=\{x\mid x\in\Sigma^*,x=\varepsilon\text{或}x\text{以}1\text{但非}001\text{结尾}\}$.

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

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

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

$set(q_5)=\{x\mid x\in\Sigma^*,x\text{以}001\text{结尾}\}$.

上述5个集合两两不交.

3.3 FA的即时描述

[**定义3.3.1**] 对FA $M=(Q,\Sigma,\delta,q_0,F)$,设能引导$M$到状态$q$的字符串的集合为$set(q)=\{x\mid x\in\Sigma^*,\delta(q_0,x)=q\}$.可如下定义一个二元关系$R_M$:对$\forall x,y\in\Sigma^*$,$xR_My$的充要条件是:$\exists q\in Q\ s.t.\ x\in set(q)\wedge y\in set(q)$.可以证明该关系是$\Sigma^*$上的等价关系,它将$\Sigma^*$划分为不多于$|Q|$个等价类.

[**定义3.3.2**] 设FA $M=(Q,\Sigma,\delta,q_0,F)$.对$x,y\in\Sigma^*$, $\delta(q_0,x)=q$,称xqy为M的一个**即时描述**,其中xy为M正在处理的字符串,子串x引导M从q_0状态启动并到达状态q,此时M正注视着子串y的首字符.

对$a\in\Sigma$,若$xqay$是M的一个即时描述,且$\delta(q,a)=p$,则记作$xqay\vdash_M xapy$.

(1)$\alpha\vdash_M^n\beta$表示M从即时描述α经n次移动到达即时描述β,

即存在M的$(n-1)$个即时描述$\alpha_1,\cdots,\alpha_{n-1}\ s.t.\ \alpha\vdash_M\alpha_1,\alpha_1\vdash_M\alpha_2,\cdots,\alpha_{n-1}\vdash_M\beta$.

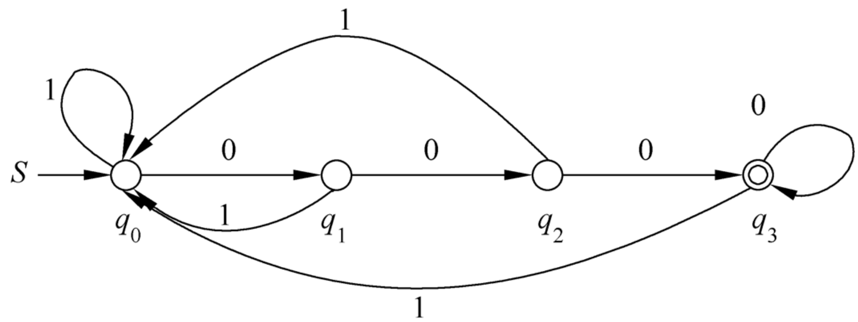
特别地,$n=0$时,有$\alpha=\beta$,即$\alpha\vdash_M^0\alpha$.

(2)$\alpha\vdash_M^+\beta$表示M从即时描述α经至少1次移动到达即时描述β.

(3)$\alpha\vdash_M^*\beta$表示M从即时描述α经若干次移动到达即时描述β.

[**注**] 实际应用中常省略$\vdash_M$中的M,即用$\vdash,\vdash^n,\vdash^+,\vdash^*$分别代替$\vdash_M,\vdash_M^n,\vdash_M^+,\vdash_M^*$.

[**例3.3.1**] 写出下面的DFA识别串1010010001的即时描述.



[**解**] $q_0 1010010001$

$\vdash 1q_0 010010001$

$\vdash 10q_1 10010001$

$\vdash 101q_0 0010001$

$\vdash 1010q_1 010001$

$\vdash 10100q_2 10001$

$\vdash 101001q_2 0001$

$\vdash 1010010q_1 001$

$\vdash 10100100q_2 01$

$\vdash 101001000q_3 1$

$\vdash 1010010001q_0$

即
$q_0 1010010001\vdash^{10}1010010001q_0, q_0 1010010001\vdash^+1010010001q_0, q_0 1010010001\vdash^*1010010001q_0$.

3.4 NFA的形式定义

[**定义3.4.1**] **不确定的有穷状态自动机**(Non-deterministic FInite Automaton, NFA)定义为五元组$M=(Q,\Sigma,\delta,q_0,F)$,其中$Q,\Sigma,q_0,F$的含义同DFA,$\delta:Q\times\Sigma\rightarrow 2^Q$.对$\forall(q,a)\in Q\times\Sigma$, $\delta(q,a)=\{p_1,\cdots,p_m\}$表示$M$在$q$状态读入字符$a$,可选择地将状态变为$p_1,\cdots,p_m$,并将读头右移一个带方格而指向输入字符串的下一个字符.

[**注1**] FA的状态转移图、状态对应的等价类、即时描述对NFA都有效.

[**注2**] 在NFA中,不是对所有的$(q,a)\in\Sigma\times Q$,$\delta(q,a)$都有一个状态与之对应.NFA在任一时刻可处于有穷多个状态,NFA比DFA更智能.

[**注3**] 类似于DFA,将δ扩充为$\hat{\delta}:Q\times\Sigma^*\rightarrow 2^Q$.

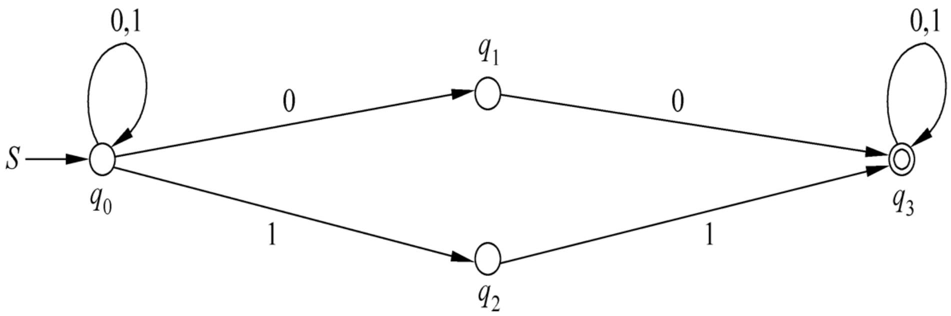
对$\forall q \in Q, w \in \Sigma^*, a \in \Sigma$,定义:

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

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

[**定义3.4.2**] 设NFA $M = (Q, \Sigma, \delta, q_0, F)$.对$\forall$串$w \in \Sigma^*$,若$\delta(q_0, w) \bigcap F \neq \varnothing$,则称$M$**接受**$w$;否则称$M$**不接受**$w$.称集合$L(M) = \{w \mid w \in \Sigma^* \wedge \delta(q_0, w) \bigcap F \neq \varnothing\}$为$M$**接受的语言**或**识别的语言**.

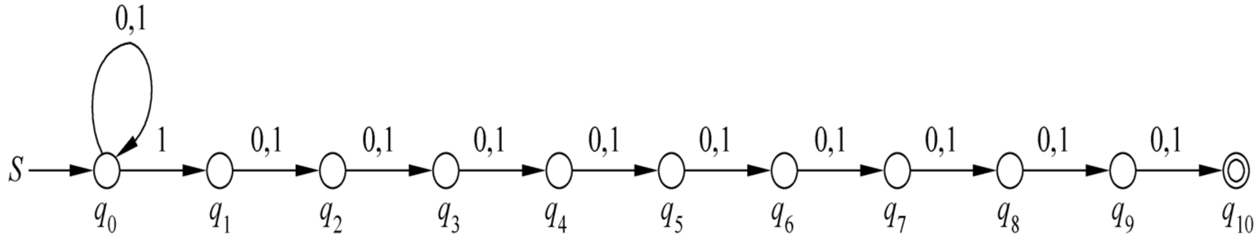
[**例3.4.1**] 识别的语言为$\{x \mid x \in \{0,1\}^*, \text{且}x\text{含有子串}00\text{或}11\}$的NFA之一如下:



其状态转移表:

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

[**例3.4.2**] 识别的语言为$\{x \mid x \in \{0,1\}^*, \text{且}x\text{的导数第}10\text{个字符为}1\}$的NFA之一如下:



3.5 NFA与DFA的等价性

[定理3.5.1] DFA与NFA等价,即:

(1)任一DFA可转化为接受相同语言的NFA.

(2)对任一NFA,存在与其接受相同语言的DFA.

[证]

(1)构造与NFA M_1等价的DFA M_2.

设$M_1=(Q,\Sigma,\delta_1,q_0,F_1),M_2=(Q_2,\Sigma,\delta_2,[q_0],F_2)$,

其中$Q_2=2^Q,F=\{[p_1,\cdots,p_m]\mid\{p_1,\cdots,p_m\}\subseteq Q\wedge\{p_1,\cdots,p_m\}\bigcap F_1\neq\varnothing\}$.

$\delta_2([q_1,\cdots,q_n],a)=[p_1,\cdots,p_m]\Leftrightarrow\delta_1(\{q_1,\cdots,q_n\},a)=\{p_1,\cdots,p_m\}$.

(2)下证$\delta_1(q_0,x)=\{p_1,\cdots,p_m\}\Leftrightarrow\delta_2([q_0],x)=[p_1,\cdots,p_m]$.

设串$x\in\Sigma^*$,对串长$|x|$归纳:

①$x=\varepsilon$时,$\delta_1(q_0,\varepsilon)=\{q_0\},\delta_2([q_0],\varepsilon)=[q_0]$,结论成立.

②设$|x|=n$时结论成立,即$\delta_1(q_0,w)=\{q_1,\cdots,q_n\}\Leftrightarrow\delta_2([q_0],w)=[q_1,\cdots,q_n]$.

③下证$x=wa$,即$|x|=n+1$时结论成立,其中$|w|=n,a\in\Sigma$.

$\delta_1(q_0,wa)=\delta_1(\delta_1(q_0,w),a)=\delta(\{q_1,\cdots q_n\},a)=\{p_1,\cdots,p_m\}$.

由δ_2的定义:$\delta_2([q_1,\cdots,q_n],a)=[p_1,\cdots,p_m]\Leftrightarrow\delta_2(\{q_1,\cdots,q_n\},a)=\{p_1,\cdots,p_m\}$.

故$\delta_2([q_0],wa)=\delta_2(\delta_2([q_0],w),a)=\delta_2([q_1,\cdots,q_n],a)=[p_1,\cdots,p_m]$.

注意到上述推导的反向也成立,故证.

(3)下证$L(M_1)=L(M_2)$.易证$L(M_2)\subseteq L(M_1)$,下证$L(M_1)\subset L(M_2)$.

设$x\in L(M_1)$,且$\delta_1(q_0,x)=\{p_1,\cdots,p_m\}$,则$\delta_1(q_0,x)\bigcap F\neq\varnothing$,这表明:$\{p_1,\cdots,p_m\}\bigcap F_1\neq\varnothing$.

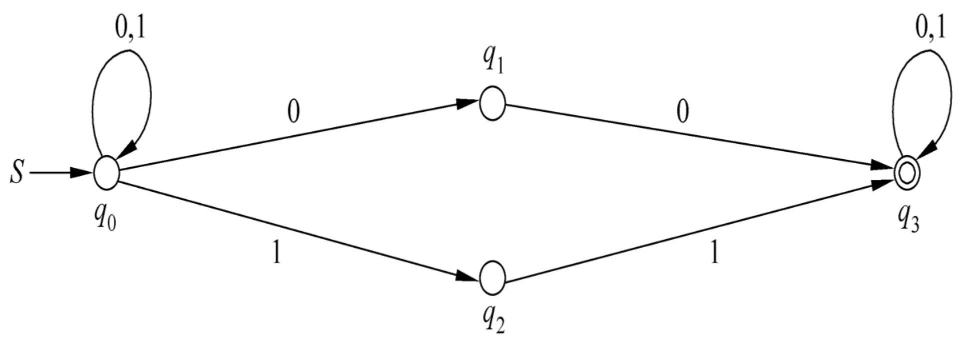
由F_2的定义:$[p_1,\cdots,p_m]\in F_2$.

由(2):$\delta_2([q_0],x)=[p_1,\cdots,p_m]$,则$x\in L(M_2)$,进而$L(M_1)\subseteq L(M_2)$.

[注1] 若不存在从$[q_0]$对应的节点出发到达状态q对应的节点的路径,则称q为**不可达状态**.在状态转移表中,用√标记从开始状态可达的状态,其余状态是不可达的.

[注2] 给定一个NFA,构造一个与之等价的DFA的方法:先将开始状态$[q_0]$填入表的状态列中,若表中有未处理的状态列,则人选一个未处理的状态$[q_1,\cdots,q_n]$,对Σ中的每个字符a,求$\delta([q_1,\cdots,q_n],a)$并填入相应的表格中.若$\delta([q_1,\cdots,q_n],a)$在状态列中未出现过,则将其填入表的状态列.重复该过程,直至表中无未处理的状态列.

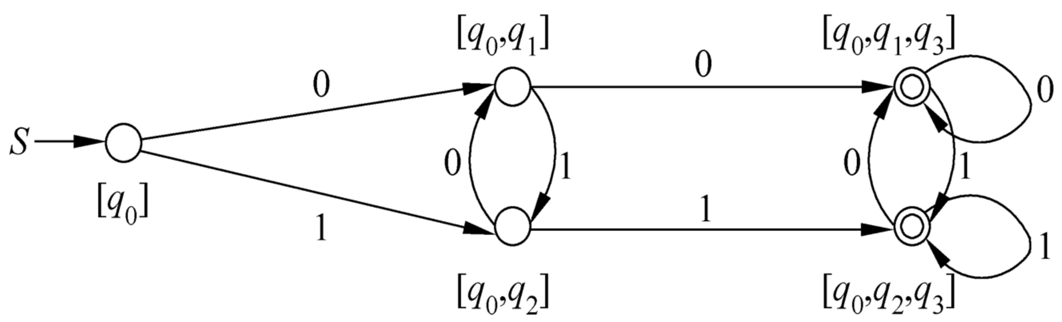
[例3.5.1] 用子集构造的方法,构造一个与下面的NFA等价的DFA.



[**解**] 状态转移表:

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		$[q_0,q_1,q_2,q_3]$	$[q_0,q_1,q_3]$	$[q_0,q_2,q_3]$
		$[\Phi]$	$[\Phi]$	$[\Phi]$

状态转移图:



[**例3.5.2**] 用子集构造的方法,构造一个与下面的NFA等价的DFA.

State	Input Symbols	
	0	1
→ A	A,B	A
B	C	B
C	D	D
D		C

[**解**] 状态转移表:

State	Input Symbols	
	0	1
[A]	[A,B]	[A]
[A,B]	[A,B,C]	[A,B]
[A,B,C]	[A,B,C,D]	[A,B,D]
[A,B,D]	[A,B,C]	[A,B,C]
[A,B,C,D]	[A,B,C,D]	[A,B,C,D]