§3.3 有限自动机(FA)

§3.3.1 确定的有限自动机（DFA）

一个DFA有以下五个元素组成：

DFA M=（K，$\sum$，f，S_0，Z）

其中 K：状态的集合（有限个状态）

$\sum$： 允许输入的字符的集合 V_t

f： 状态转换函数，单值函数$K\times\sum\rightarrow K$

$$f(S_i,a)=S_j$$

S_0： 初始状态$S0\in K$

Z： 终止状态集$Z \in K$

f单值函数$K\times\Sigma\to K$

讨论：

(1) ①确定性　f是单值函数

从某一状态读一字符的下一状态唯一确定

②有限的　K集合元素个数有限

(2) f 定义的推广　　$\hat{f}: K\times\Sigma^*\to K$ 单值映射

$K\times\Sigma^*\to K$　　记为 $\hat{f}$　　$\Sigma^*=\Sigma^+\cup\{\varepsilon\}$

① $\hat{f}(S, \varepsilon) = S$

② $\hat{f}(S, a\omega) = \hat{f}(f(S,a), \omega)$　　$a\in\Sigma, \omega\in\Sigma^*$

$= \hat{f}(S_k, \omega)$　　$f(S,a)=S_k$

$=\ldots=S_t$　　$S_t\in K$

③ f ^是在Σ^*上定义,f是在$\sum$上定义,

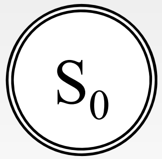
f ^包含f, 将 f ^和f合并为一个f,记为f(推广后)

(3) <u>有限自动机的功能</u>:识别句子

$$L(M)= \{x \mid f(S_0,x) \in Z,\ x \in \Sigma^* \}$$

特别:$f(S_o, \varepsilon) = S_o$ 且 $S_o \in Z$

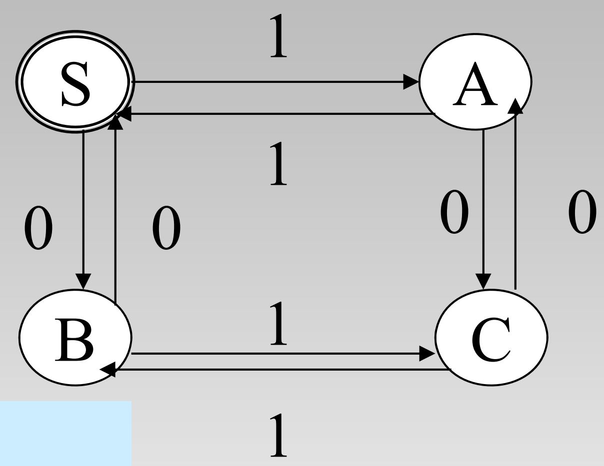
称 ε 可为DFA M识别



(4) DFA可非形式地表示成状态图和状态矩阵

例：DFA M=({S,A,B,C},{0,1},δ,S,{S})

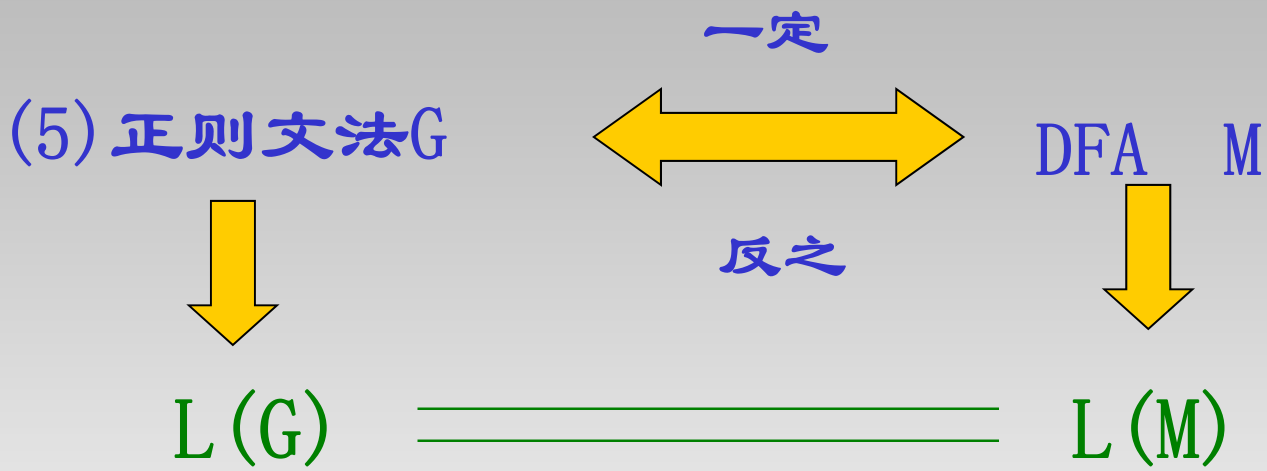
δ(S,0)=B　δ(S,1)=A
δ(A,0)=C　δ(A,1)=S
δ(B,0)=S　δ(B,1)=C
δ(C,0)=A　δ(C,1)=B



101011

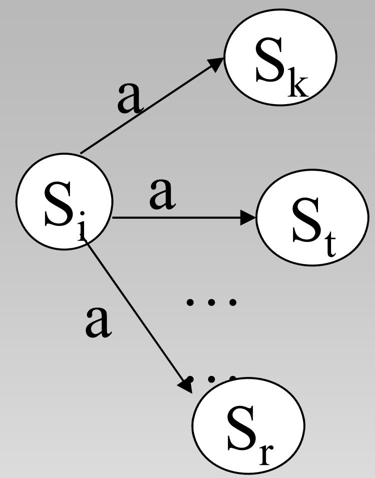
δ(S, 101011)
= δ(δ(S, 1), 01011)
= δ(A, 01011) = δ(C, 1011)
= δ(B, 011) = δ(S, 11)
= δ(A, 1) =S

状态	输入 0	输入 1
S	B	A
A	C	S
B	S	C
C	A	B


(5) 正则文法G
一定
反之
DFA M
L(G)
L(M)

§3.3.2 非确定的有限自动机(NFA M′)

一个NFA M′有以下五个元素组成,



DFA $M'=(K',\Sigma,f',S_0',Z')$

其中： K′:状态的集合(有限状态)

Σ: 允许输入的字符集合

f′:状态转换函数, 多值函数,

$K'\times\Sigma\rightarrow 2^{k'}$(K的所有子集的集合)

$f(S_i,a)=\{S_k,S_t\ldots\}$

S_0:初始状态 $S_0\in K'$

Z: 终止状态集 $Z\in K'$

讨论：

1) ①不确定:f′是多值函数，一对多

②有限: K′有限

2) f′ 定义推广到 Σ^* 上，

$K' \times \Sigma^* \to 2^{k'}$，记为 $\hat{f}'$

① $\hat{f}'(S, \varepsilon) = \{S\}$

② $\hat{f}'(S, aw)$

设： $f'(S, a) = \{S_1, S_2, \ldots, S_k\}$

$= \hat{f}'(f'(S, a), w)$

$= \hat{f}'(\{S_1, S_2, \ldots, S_k\}, w)$

$= \bigcup_{i=1}^{k} \hat{f}'(S_i, w)$

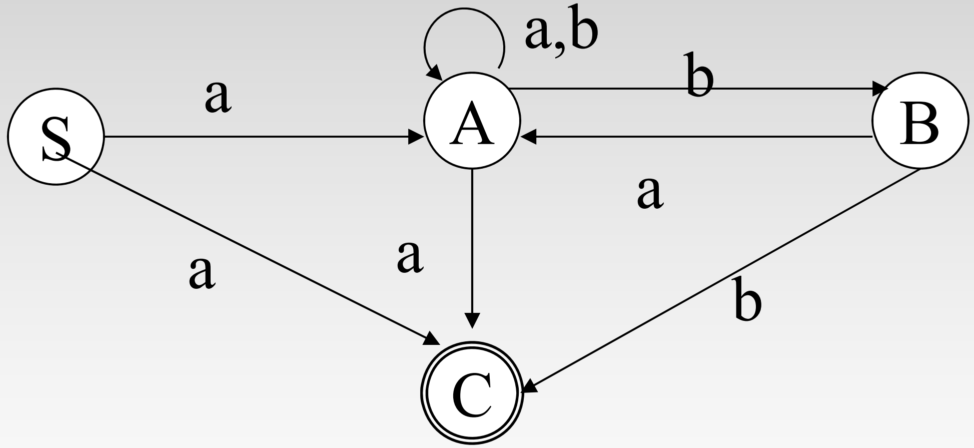
③将 $\hat{f}'$ 和f′合并为一个f′,记为f′

(3) NFA M ′所确定的语言

$L(M') = \{x \mid f'(S_0', x) \cap Z \neq \emptyset, x \in \Sigma^*\}$

(4) NFA M ′ 通常用状态转换图来表示

例：NFA M ′=({S,A,B,C}, {a,b}, f ′, S, {C})



符号串ababb可由此NFA M ′所识别.

§ 3.3.3 DFA M与NFA M′的等价性

一、对于∑上NFA M′ $\xrightarrow{\text{一定}}$ DFA M

$$\begin{array}{ccc} \text{NFA } M' & \xrightarrow{\text{一定}} & \text{DFA } M \\ \downarrow & & \downarrow \\ L(M') & = & L(M) \end{array}$$

二、方法：

<u>确定化</u> 1、读ε不动作的NFA M′

2、读ε动作的NFA M′

<u>最小化</u>

1、 读ε不动作的NFA M′的确定化

$K' \times \Sigma \to 2^{k'}$

问题：设有一NFA $M'=(K',\sum,f',S_0',Z')$

现构造一$\sum$上的DFA $M=(K,\sum,f,S_0,Z)$

使$L(M')=L(M)$

①K由K'的全部子集组成 $K=2^{K'}$ 特别$S_0=[S_0']$

② 映射f的定义

S_i、$R_j \in K'$

当且仅当$f'(\{S_1,S_2,\ldots S_i\},a)=\{R_1,R_2,\ldots R_j\}$时，

则$f(\underbrace{[S_1,S_2,\ldots S_i]}_{q_i},a)=\underbrace{[R_1,R_2,\ldots R_j]}_{q_j}$

$f(q_i, a)=q_j$

思想：状态合并为状态集合

③DFA M的终态集Z的定义是：

M的某一状态$[R_1,R_2,\ldots R_j]$,

其中至少含有一个M′的一个终态

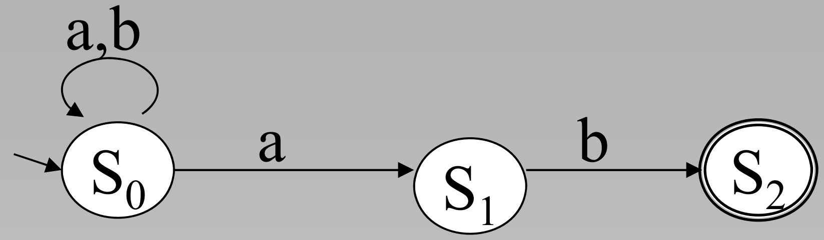
则$[R_1,R_2,\ldots R_j] \in Z$

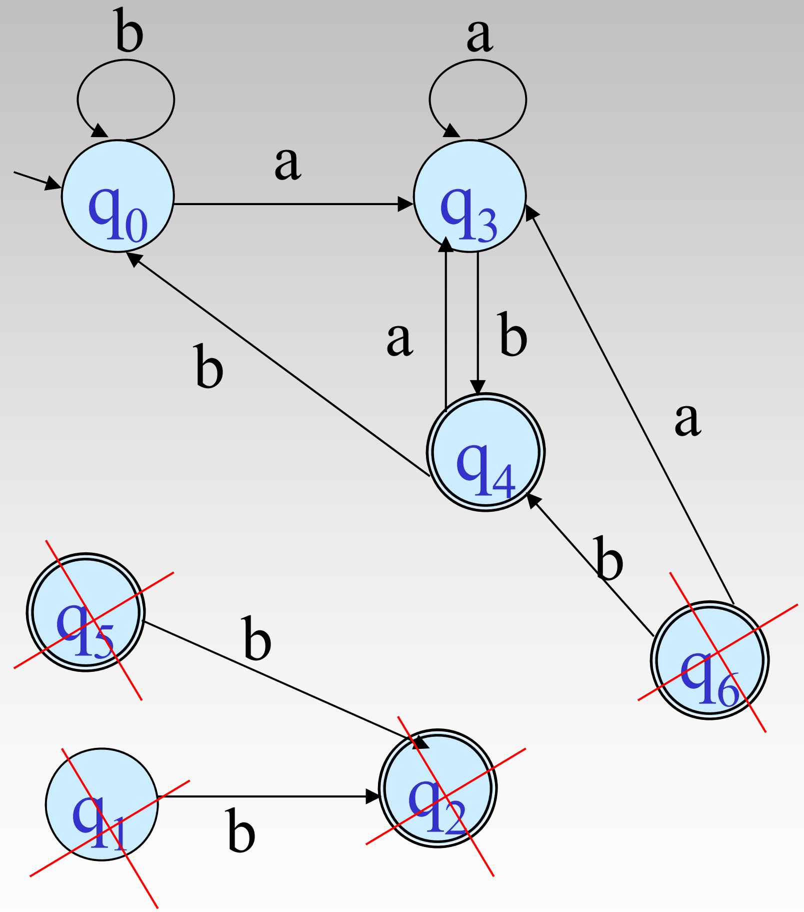
$$Z=\{[R_1,R_2,\ldots,R_j] \mid [R_1,R_2,\ldots,R_j] \in K \text{ 且 } \{R_1,R_2,\ldots,R_j\} \cap Z' \neq \varnothing\}$$

$f'(\{s_0\},a)=\{s_0, s_1\}$

$f(q_0,a)=q_3$

	a	b
$q_0=[s_0]$	q_3	q_0
$q_1=[s_1]$	Ø	q_2
$q_2=[s_2]$	Ø	Ø
$q_3=[s_0, s_1]$	q_3	q_4
$q_4=[s_0, s_2]$	q_3	q_0
$q_5=[s_1, s_2]$	Ø	q_2
$q_6=[s_0, s_1, s_2]$	q_3	q_4
Ø		

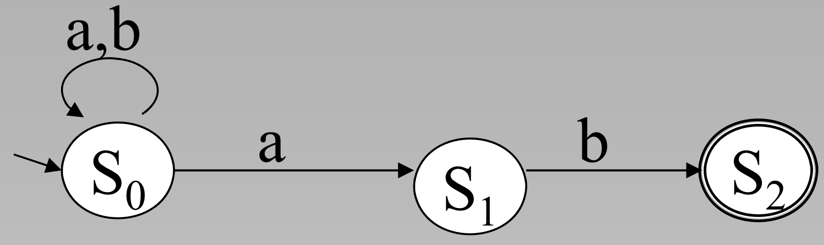




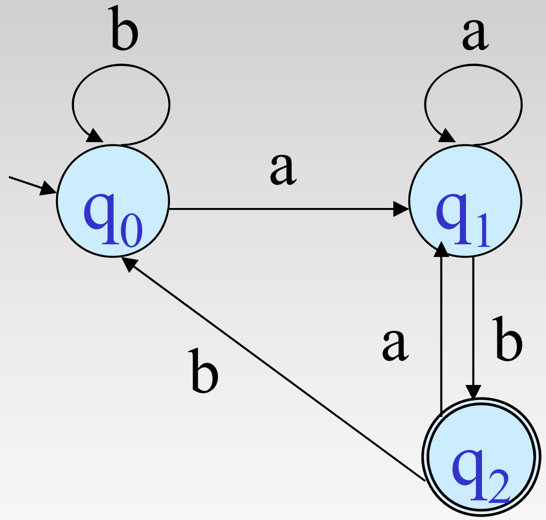
各个状态的全部组合

多余部分删除

	a	b
s_0	$\{S_0, S_1\}$ (q_1)	S_0 (q_0)
s_1	Ø	S_2 (q_2)
s_2	Ø	Ø



	a	b
q_0	q_1 $\{S_0, S_1\}$	q_0 $\{S_0\}$
q_1	q_1 $\{S_0, S_1\}$	q_3 $\{S_0\ S_2\}$
~~q_2~~	~~Ø~~	~~Ø~~
q_3	q_1 $\{S_0, S_1\}$	q_0 $\{S_0\}$



其中q_3包含S_2, 为终端符号，

故可删除表中的 q_2， 并调整q_3为q_2.

2、 读ε动作的NFA M ′的确定化

目的：

$K' \times \Sigma \cup \{\varepsilon\} \rightarrow 2^{k'}$

为了把识别各类单词的DFA用ε链接起来，组成一个单一的NFA，然后把NFA确定化，最后再以此设计词法分析器

定义:

(1) 状态q的ε闭包　　ε_closure(q)　　$q \in K'$

① $q \in \varepsilon\_closure(q)$

② $q \xrightarrow{\varepsilon\{\varepsilon\}}$ 到达的状态均属于ε_closure(q)

(2) 状态集P的ε闭包　　ε_closure(P)　　$P \in K'$

① 若$q \in P$,则 $q \in \varepsilon\_closure(P)$

② 若$q \in P$,

$q \xrightarrow{\varepsilon\{\varepsilon\}}$ 到达的状态均属于 ε_closure(p)

全部合并

（3）重新定义f' :$S \in K', a \in Vt, w \in \Sigma^*$

① $f'(S,\varepsilon) = \varepsilon\_closure(S)$

② $f'(S,aw) = \varepsilon\_closure(p)$

其中，$P = \hat{f}'(S, aw)$

上述定义同样适用读ε不动作的有限自动机

方法：

设有一 NFA $M'=(K',\sum,f',S_0,Z')$

现构造一 $\sum$ 上的 DFA $M=(K,\sum,f,q_0,Z)$

使 $L(M')=L(M)$

(1) $q_0=\varepsilon\_closure(S_0)$ $q_0\in K$

(2) 对K中任一尚未标记的状态 $q_i=\{S_{i1},S_{i2},\ldots,S_{im}\}$

$S_{ik}\in K'$

做①标记 q_i;

②对于每一 $a\in\sum$, 置

$T=f'(\{S_{i1},S_{i2},\ldots,S_{im}\},a)$;

$q_j=\varepsilon\_closure(T)$;

③ 若q_j不在K中，则将q_j作为一未加标记的状态加 入K中，$f(q_i,a)=q_j$，添加到M上

$$q_i \xrightarrow{a} q_j$$

(3) 重复步骤(2)，直到K中不再含有未标记的状态为止.

(4) 若某一$q_j \cap Z' \neq \emptyset$,则$q_j \in Z$

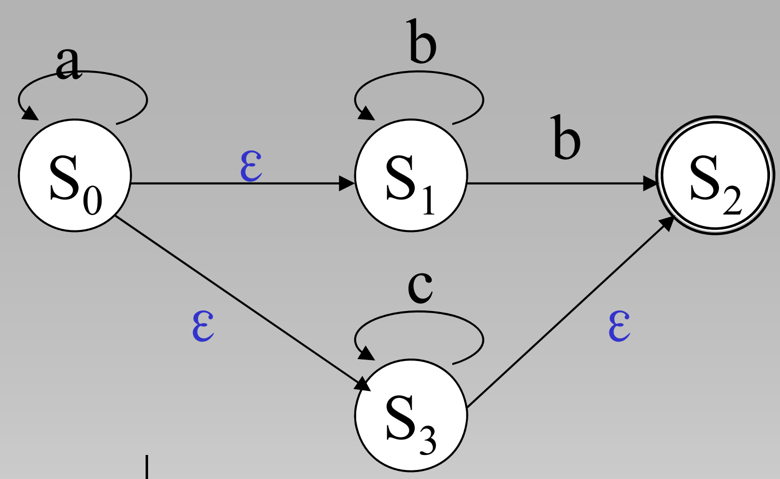
$q_0 = \varepsilon\_closure(S_0)$

$= \{S_0, S_1, S_2, S_3\}$

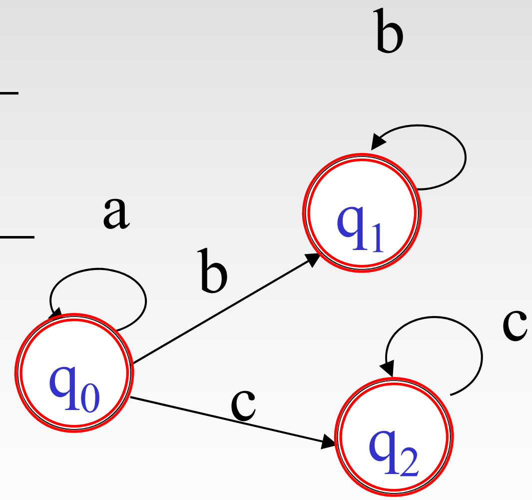
$f(q_0, a) = \varepsilon\_closure(f'(q_0, a))$

$= \{S_1, S_2, S_3, S_0\}$

$= q_0$



	a	b	c
$q_0 = \{S_1, S_2, S_3, S_0\}$	q_0	$\{S_1, S_2\} = q_1$	$\{S_2, S_3\} = q_2$
$q_1 = \{S_1, S_2\}$	Ø	q_1	Ø
$q_2 = \{S_2, S_3\}$	Ø	Ø	q_2



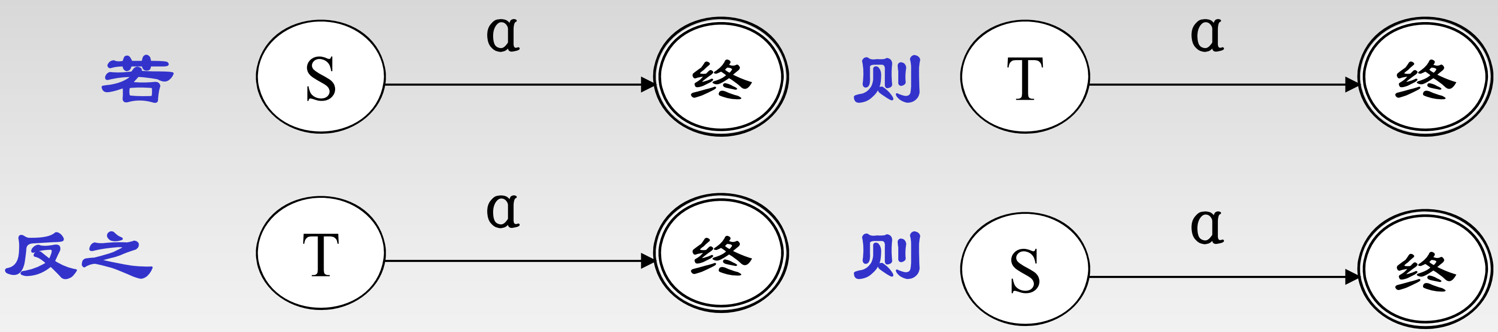
3、DFA状态数的最小化

--对DFA中具有“同一性”的状态进行合并

--寻找状态数比DFA M少的DFA M_1，使$L(M)=L(M_1)$

DFA M的两个不同状态S,T

（1）S和T等价　　$\alpha \in \Sigma^*$



称S和T等价，否则，称S和T不等价，是可区别的。

(2) S和T是可区别的： S和T不等价

如终态和非终态是可区别的。

对DFA终态可读ε， 而非终态不可。

(3) 最小化的思路

- 将DFA M的状态集分割成一些不相交的子集；
- 使任何不同子集的状态是可区别的；
- 同一子集的任何两个状态都是等价的；
- 每一个子集选出一个代表，消去其它等价状态。

(4) 最小化的方法

①将状态集k的终态和非终态分开，分成两个子集；

形成基本划分$\Pi =\{Z, K-Z\}$

②令某时$\Pi =\{I_1, I_2, \ldots, I_m\}$,并且属于不同子集的状态是可区分的

再划分某个$I_i=\{S_1, S_2, \ldots, S_k\}$,若有$a \in \Sigma$，使$f(I_i, a)$不全包含在现行$\Pi$的某一子集中，则将$I_i$一分为二。

如$f(S_i,a)=R_i$, $f(S_j,a)=R_j$,且$R_i \in I_k$，$R_j \in I_t$

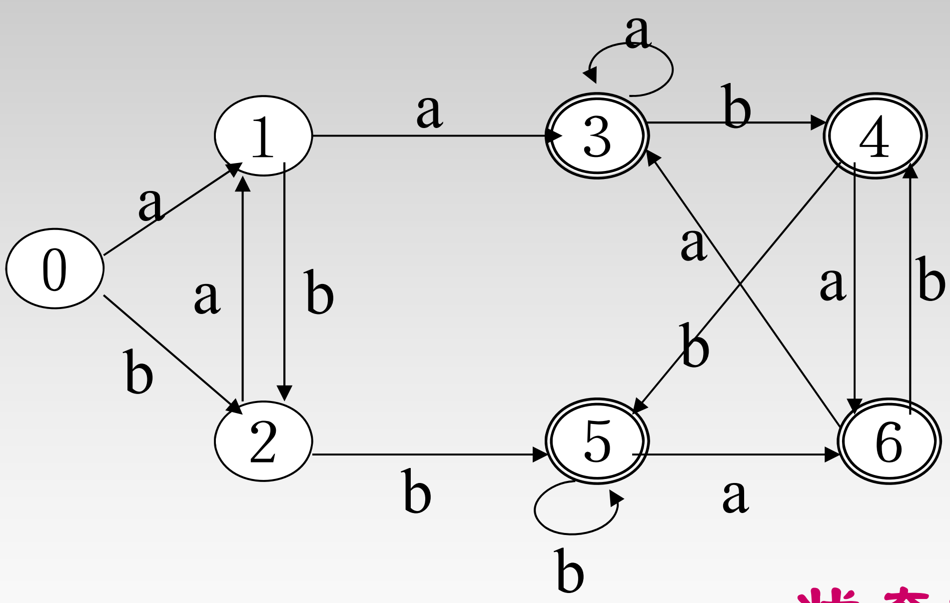
则将I_i划分成 $I_{i1}=\{S_i | S_i \in I_i$,且S经a弧到$R_i\}$

$I_{i2}=I_t - I_{i1}$

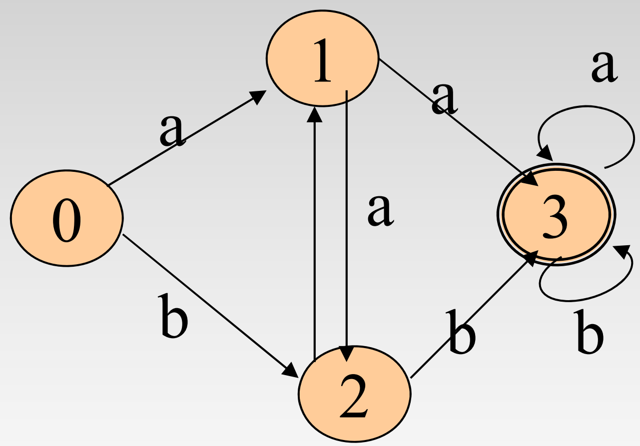
③重复上述过程，直到Π不再增大为止。

每一子集中的状态都是等价的，

不同子集中的状态都是可区分的。



状态数最小化



§3.4正规表达式与正规集

§3.4.1正则表达式与正规集

正则式：描述单词符号

正规集：正规式描述的语言

1、正则式的递归定义

设有字汇表V，则：(V_T就是$\sum$)

(1)ε, Ø, a, $a \in V_t$都是正则表达式,

正规集{ε}, Ø, {a}

(2)如果e_1和e_2是正则式,正规集分别为L_1和L_2

则 $e_1 | e_2$是正则式,正规集为$L_1 \cup L_2$

$e_1 \cdot e_2$是正则式,正规集为$L_1 L_2$

e_1^*是正则式, 正规集为L_1^*

注：*, · , | 的优先级依次降低

例:$a \in V_t$, $b \in V_t$, $V_t=\{a,b\}$

a是正规式 a^*, ba, $a|ba^*$ 均是正规式

b是正规式 $a|b$,$(a|b)^*$, $a(a|b)^*$均是正规式

2、两个正则式相等:

两个正则式表示相同的语言

如$(a|b)^*=(a^*|b^*)^*$,

语言为$\{x \mid x\in\{a,b\}^*\}$

3、正则式的操作

结合律$(ab)c=a(bc)$,$(a|b)|c=a|(b|c)$

交换律$a|b=b|a$

分配律 $a(b|c)=ab|ac$, $(a|b)c=ac|bc$

其他:$r|r=r$, $r^*=(r|\varepsilon)^*=\varepsilon|r r^*$

§ 3.4.2 由正则文法构造相应的正则式

求出以{ }、|、. 表示的文法的语言就是正则式,

用=代替→, 用+代替|求出联立方程组的解

1、文法 $x \to rx|t$ 右线性 $r,t \in V_t^+$

$$L_x = L_r \cdot L_x \cup L_t$$

为求解方便写成 $x=rx+t$ 解方程求x

又 $x \Rightarrow rx \Rightarrow r\ldots rx$,

$L_x = \{t, rt, rrt, \ldots\}$ 即 r^*t

可解得 $x=r^*t$

[论断3.1]:方程组 $x=rx+t$ 有形如 $x=r^*t$ 的解

2、文法 x=xr|t 左线性 $r,t \in Vt^+$

$L_x = L_x \cdot L_r \cup L_t$

为求解方便写成 x=xr+t 解方程求x

又x=>xr=>xr…r,

L_x ={t,tr,trr,…}即 tr*

可解得x=tr^*

[论断3.2]:方程组x=xr+t有形如x=tr^*的解

例:G[S]: S→aA　　求文法对应的正规式

A→bA|aB|b

B→aA

解:改写成 S=aA　　(1)

A=bA+aB+b　　(2)

B=aA　　(3)

(3)代入(2)得　A=bA+aaA+b

A=(b+aa)A+b

A= (b+aa)*b (论断3.1)

S=aA=a(b+aa)*b再用|代替+

即正规式为a(b|aa)*b

§ 3.4. 3 由正则式构造FA M

1、正则式与FA 的等价性

① e为正则式,则存在一NFA　M',它接受的语言为L(e)

$$
\begin{tikzcd}
e \arrow[r] \arrow[d] & \text{NFA } M' \arrow[d] \\
L(e) \arrow[r, equal] & L(M')
\end{tikzcd}
$$

② L(M)能为DFA M所识别,则L(M)能用正则式表示为L(e)

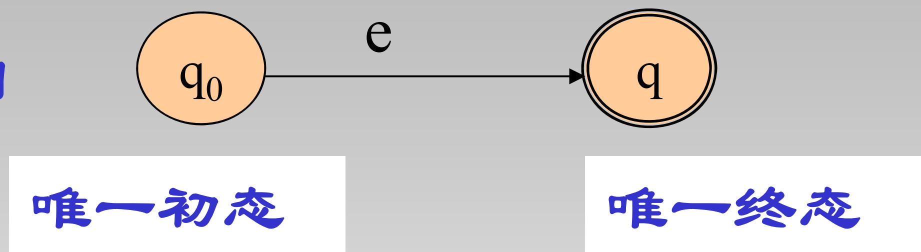
$$
\begin{tikzcd}
\text{DFA } M \arrow[r] \arrow[d] & e \arrow[d] \\
L(M) \arrow[r, equal] & L(e)
\end{tikzcd}
$$

结论： e→NFA M'→DFA M→e

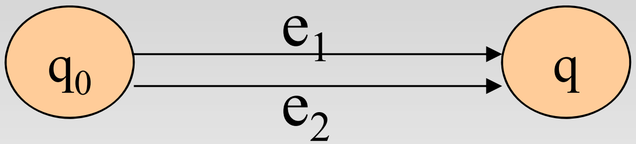
2、正则式与FA的转换

① e→NFA M'

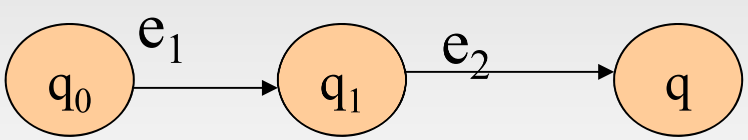
规则：e存在状态图



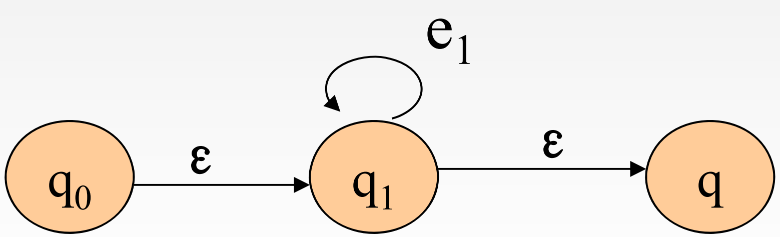
$e_1|e_2$代之以



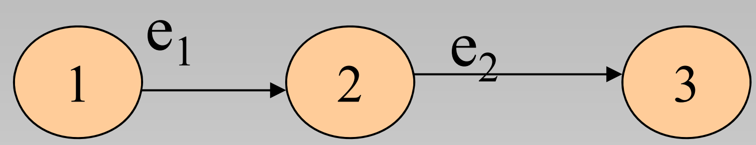
e_1e_2代之以



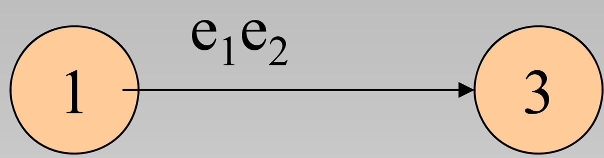
e_1^*代之以

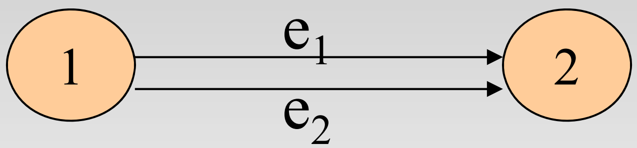


② DFA M ➔e

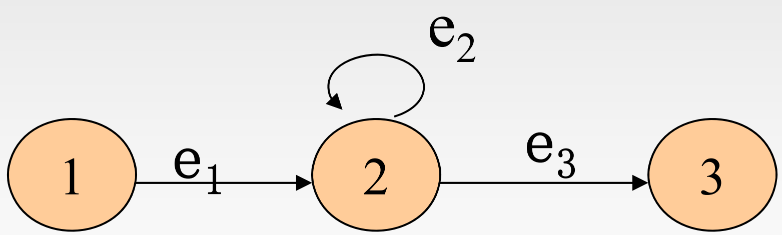


代之以

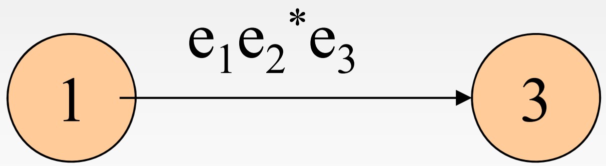




代之以



代之以

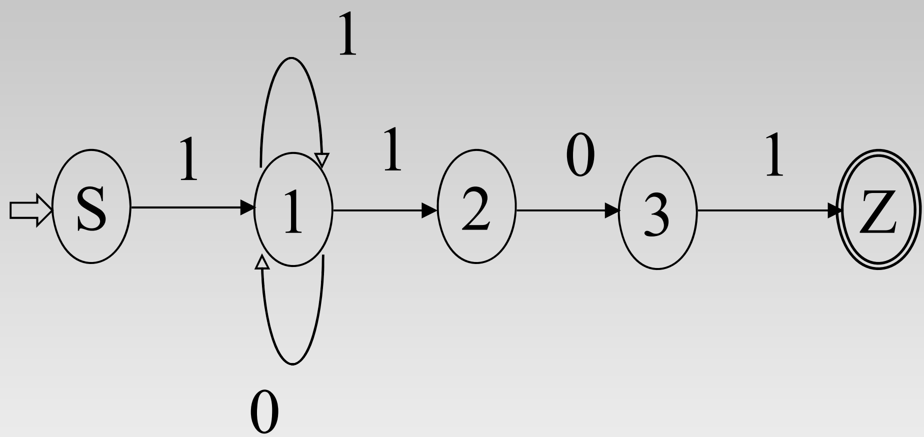


思考题（一）

- 构造与正规式 1(0｜1)*101相应的DFA

解：转化为NFA（Step1）状态转换矩阵

NFA转DFA（Step2）状态图

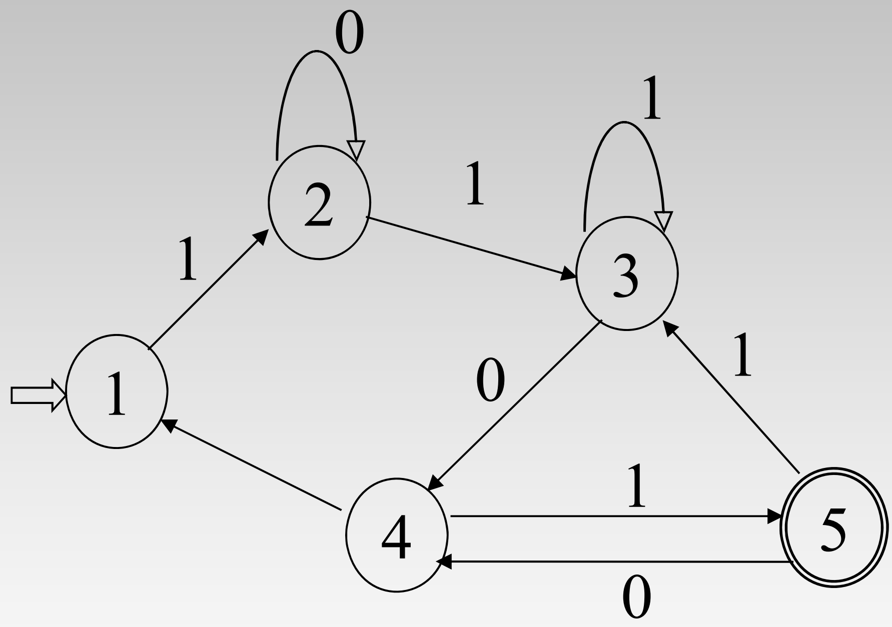


	0	1
① [s]	Ø	② [1]
② [1]	② [1]	③ [1,2]
③ [1,2]	④ [1,3]	③ [1,2]
④ [1,3]	② [1]	⑤ [1,2,Z]
⑤ [1,2,Z]	④ [1,3]	③ [1,2]

思考题

- 构造与正规式 $1(0 \mid 1)^*101$ 相应的DFA

解： DFA图



思考题（二）

- 已知有穷自动机如图：

1. 表示的语言是什么？（可用L={W｜...}来描述）
2. 写出该语言的正规式和正规文法
3. 构造识别该语言的DFA