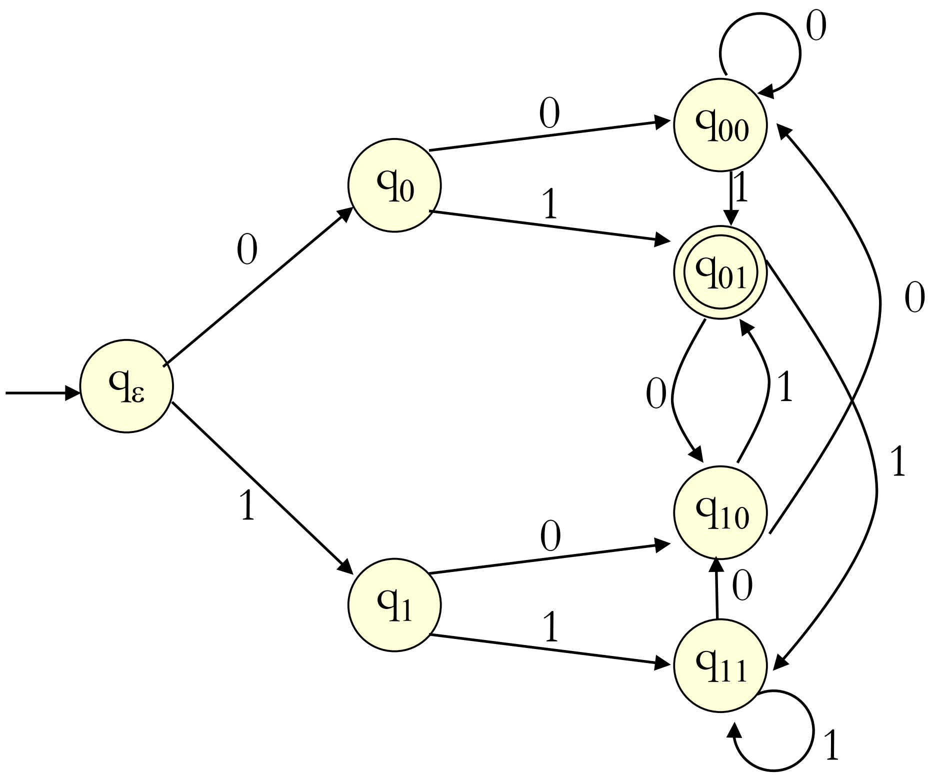
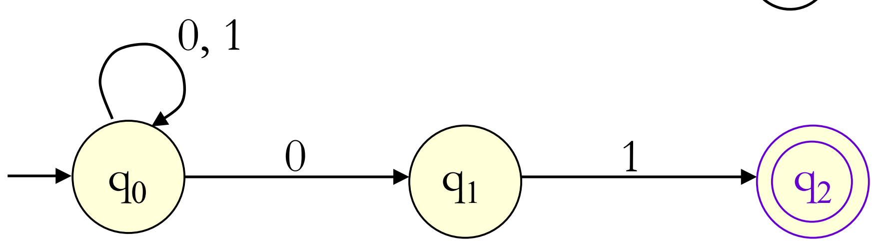
第二章（NFA）2025

NFA与DFA在定义语言能力上是否等价?

- 图示是否定义了同样的语言?





定义语言*L*的DFA只有一个还是多个?

定义语言*L*的NFA只有一个还是多个?

结论是：不管多少，定义语言*L*的DFA和NFA都是彼此相互等价的。

DFA与NFA等价性

定理2.1：语言*L*为某个DFA接受当且仅当它为某个NFA接受

- 二者的字母表相同，设为Σ。
- 无论$w\in\Sigma$是为哪一个FA所接受，当且仅当它们都有一条标记为w的、始端为初始状态、末端为接受状态的路径存在。

- 证明：
  - （当）构造性证明任一个NFA都能被等价地转换为DFA。
  - （仅当）显然，任一DFA都是一个NFA特例。

状态幂集视觉下的构造性证明思路

已知NFA $(Q, \Sigma, \upsilon, q_0, F)$，构造与其等价的一个DFA。

思路：将Q 的子集作为DFA的状态，求υ'（**子集构造法**）

对于$S, T \subseteq Q$ ，如果 $\forall p \in T \cdot \exists q \in S \cdot p \in \upsilon(q, a)$，那么$T = \upsilon'(S, a)$，意味着$S, T$ 是DFA的状态，υ' 是DFA的转移函数。从而，

$$\upsilon' = \{((S, a), T) \mid S \subseteq Q, a \in \Sigma, T = \cup q \in S \cdot \upsilon(q, a)\}$$

$$F' = \{S \subseteq Q \mid S \cap F \neq \emptyset\}$$

得到完全形DFA $(2^Q, \Sigma, \upsilon', \{q_0\}, F')$

最后，去除无用状态得到Q_D。令$\mathbb{q}_0 = \{q_0\}$，即有，

$$Q_D = \{\mathbb{q} \in 2^Q \mid \mathrm{PATH}[\mathbb{q}_0, \mathbb{q}, x] \wedge \mathrm{PATH}[\mathbb{q}, \mathbb{q}_1, y], x, y \in \Sigma^*, \mathbb{q}_1 \in F'\}$$

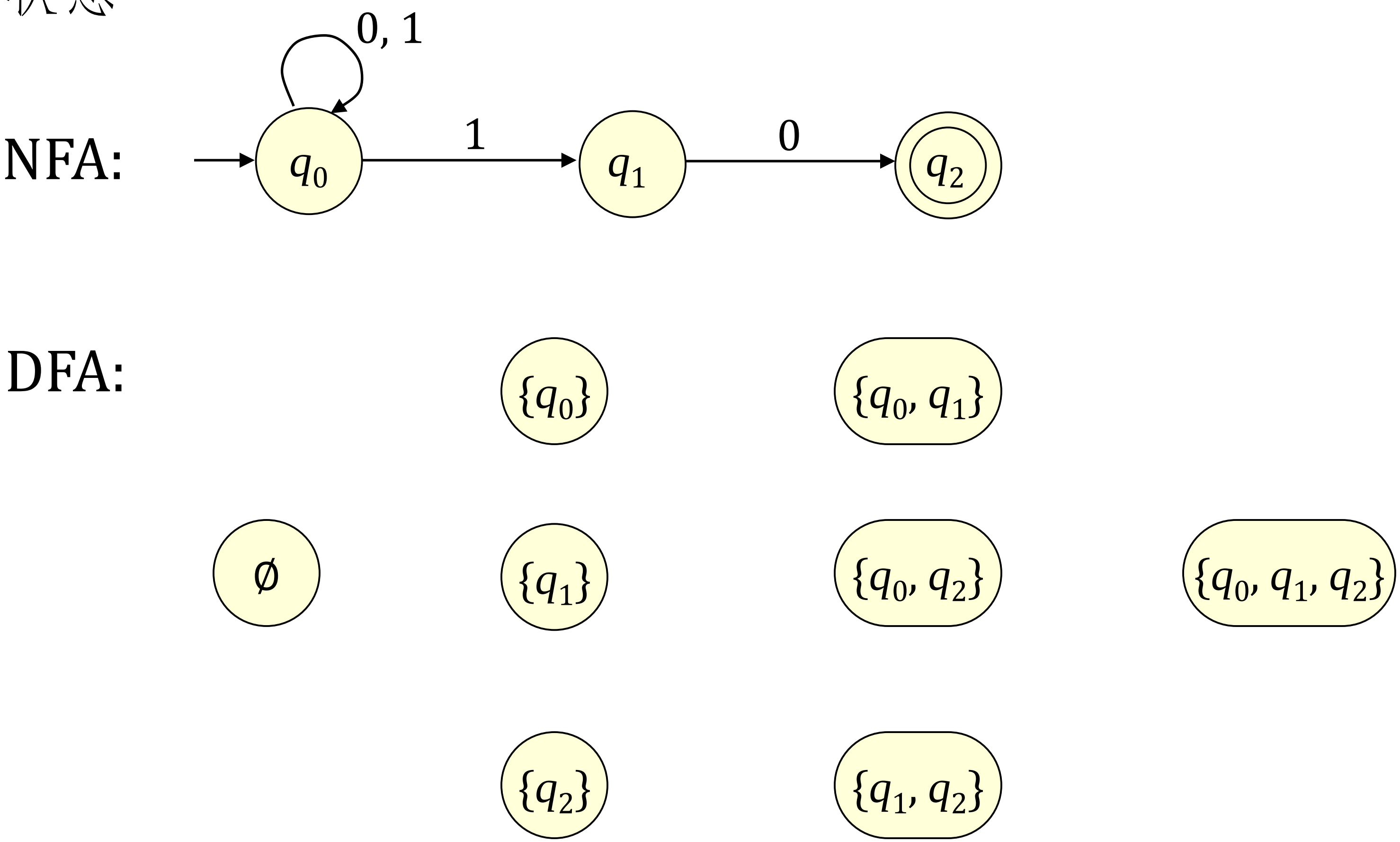
$$\upsilon_D = \{((S, a), T) \in \upsilon' \mid S, T \in Q_D\}$$

$$F_D = \{S \in Q_D \mid S \cap F \neq \emptyset\}$$

最终构造出最简形DFA $(Q_D, \Sigma, \upsilon_D, \{q_0\}, F_D)$。

NFA到DFA子集构造法

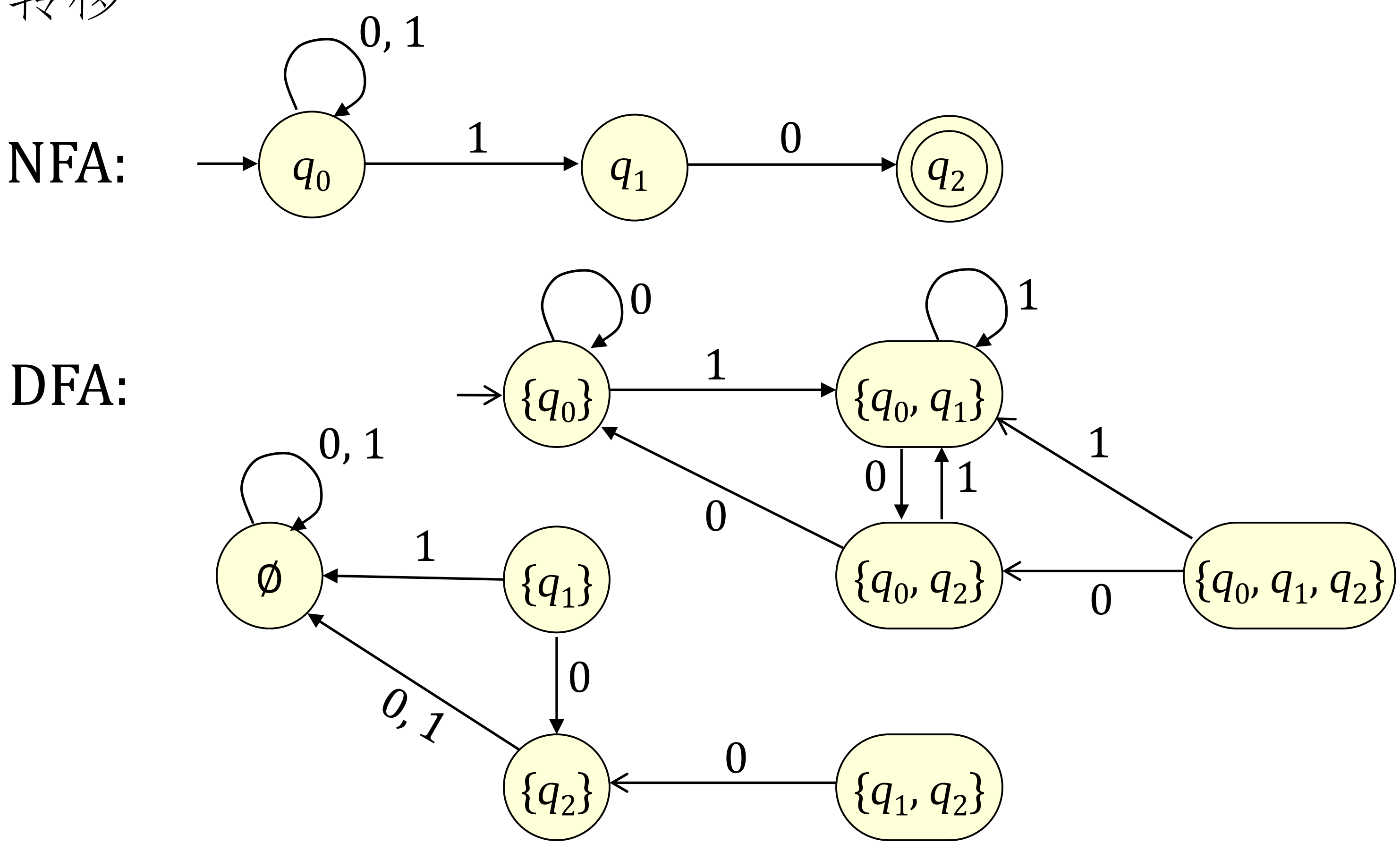
- 状态



对每一个NFA状态的子集DFA都有一个状态与之对应。

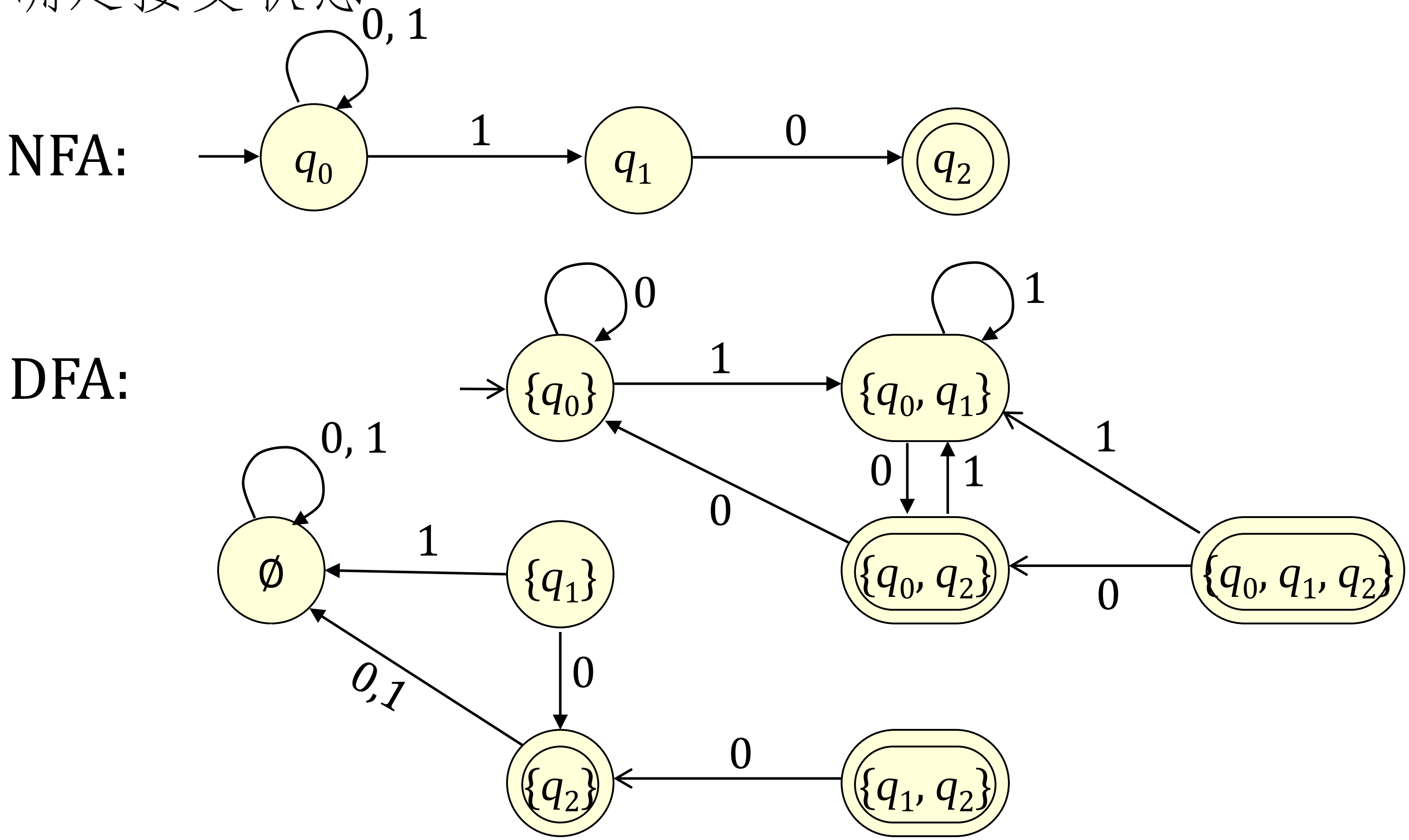
NFA到DFA的子集构造法

- 转移



NFA到DFA的子集构造法

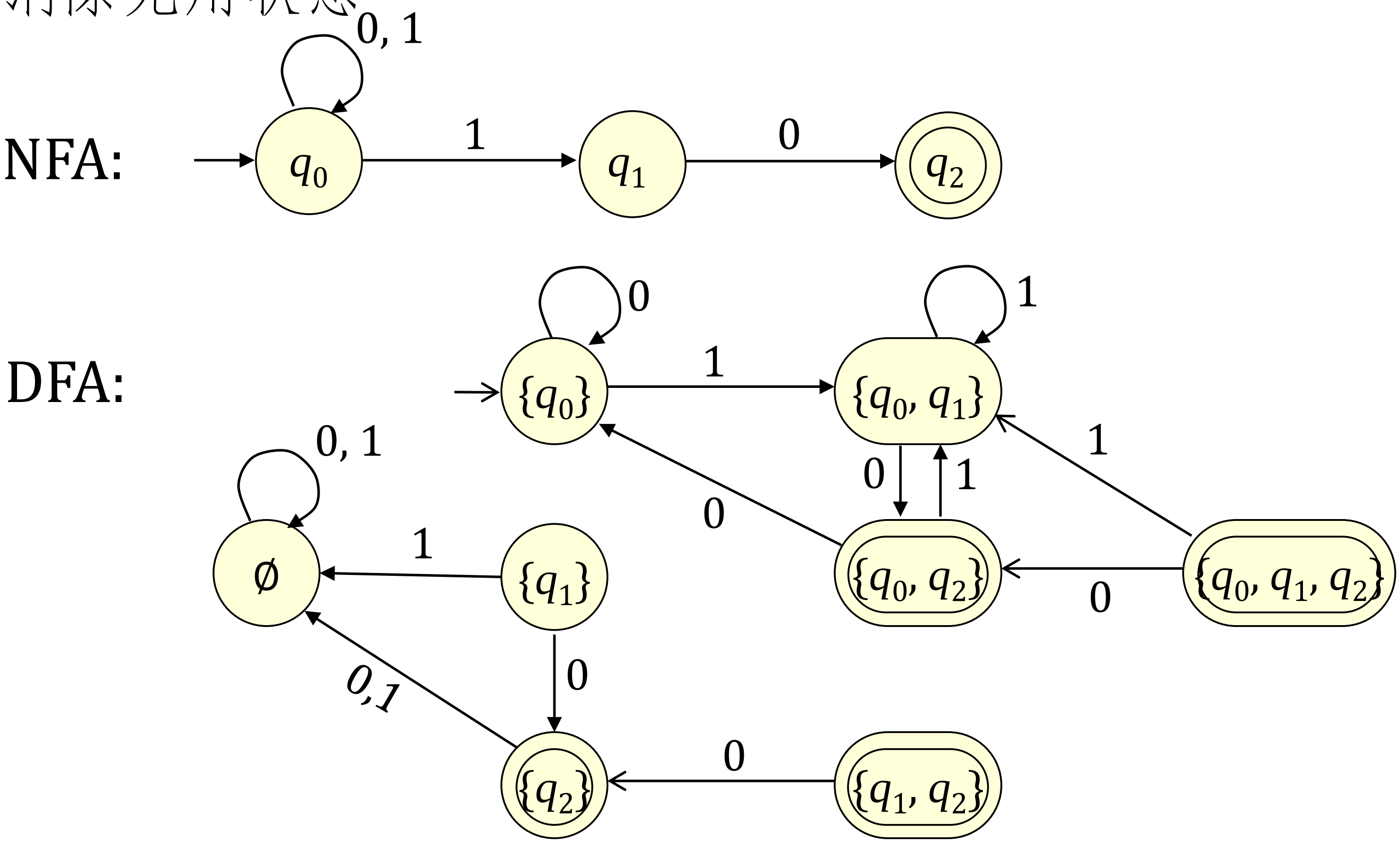
- 确定接受状态



含有NFA接受状态的集合作为DFA接受状态。

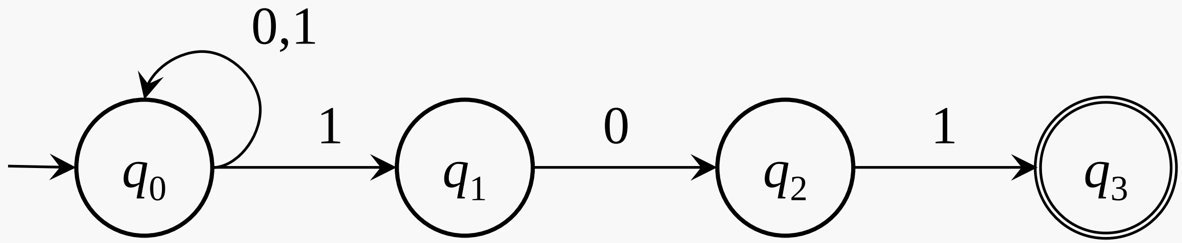
NFA到DFA的子集构造法

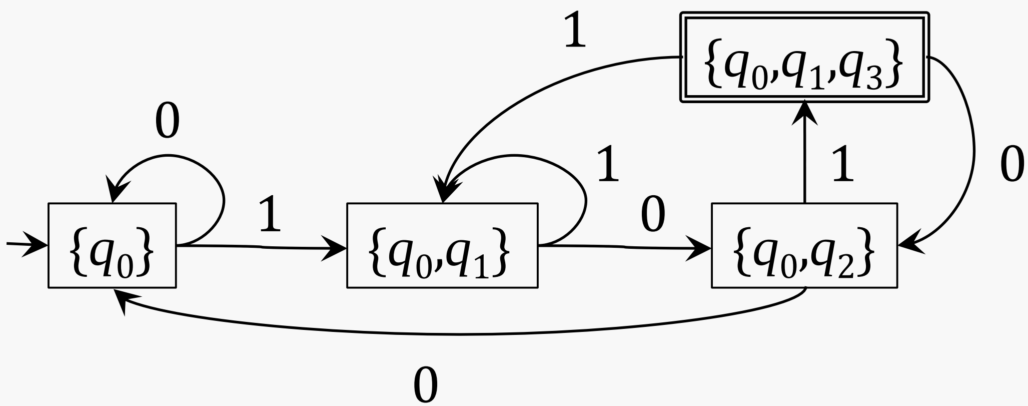
- 消除无用状态



最后，删除不可达和不能到接受状态的那些状态。

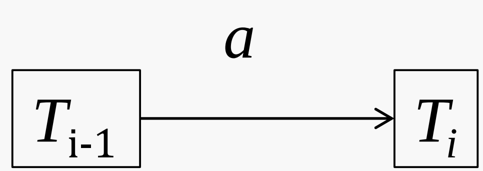
基于活动状态集的惰性构造方法





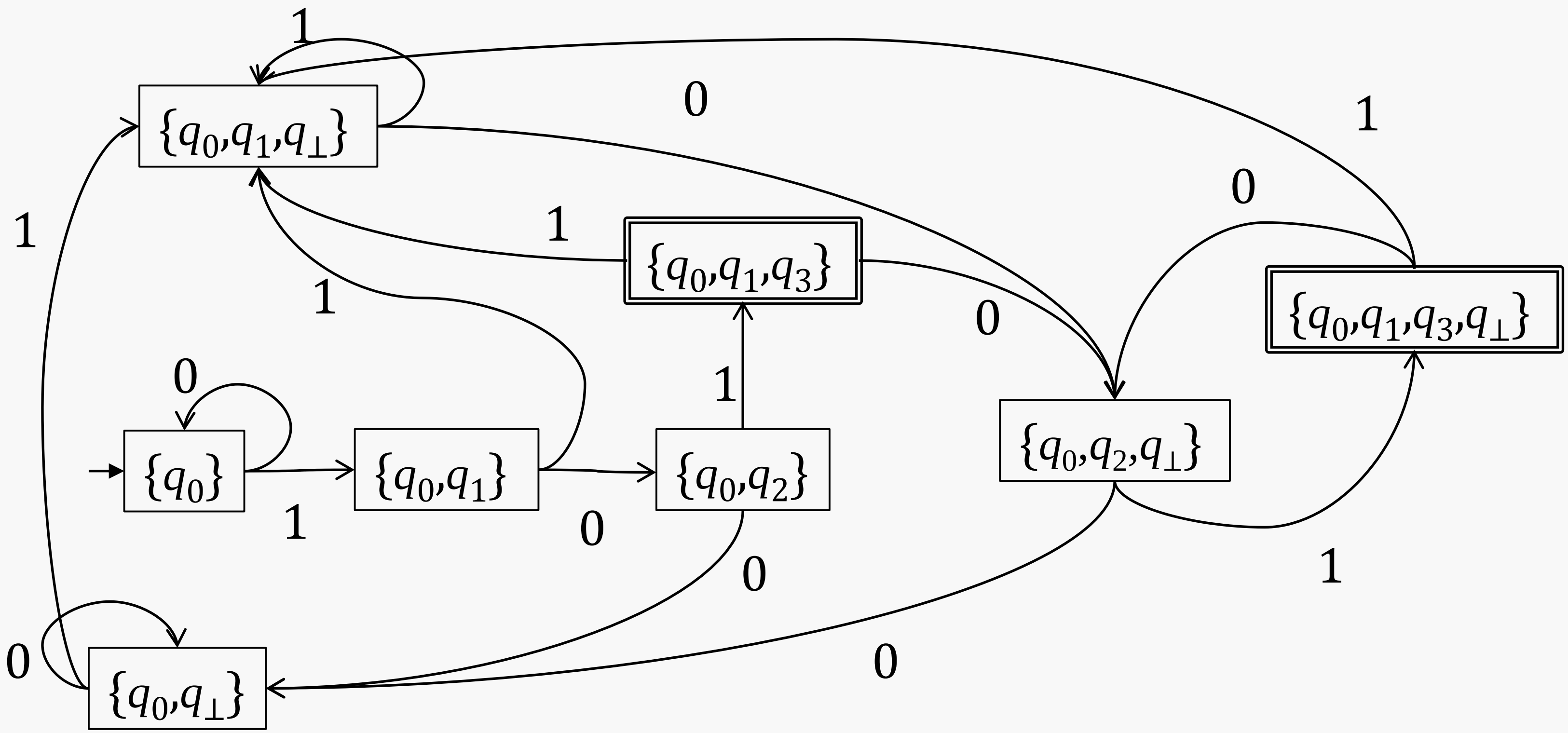
已读前缀x，
剩余串ay；
当前输入符号a；
当前状态集合$T_{i\text{-}1}$；
$T_{i\text{-}1}=\tilde{\delta}(q_0, x)$

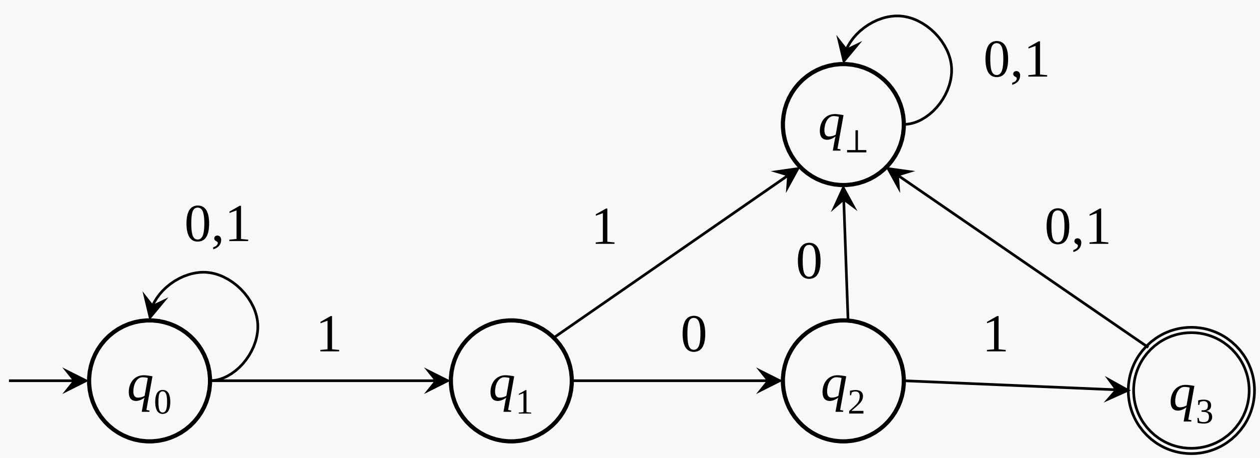
输入串xay



已读前缀xa，
剩余串y；
当前状态集合T_i；
$T_i=\tilde{\delta}(q_0, xa)$

例：基于活动状态集的NFA转DFA





NFA 到 DFA 的子集构造算法

输入： NFA $(Q, \Sigma, \upsilon, q_0, F)$

输出： DFA $(\mathbb{Q}, \Sigma, \text{move}[], \{q_0\}, \{S\in\mathbb{Q} \mid S\cap F\neq\emptyset\})$

$\mathbb{Q} = \emptyset$;

move[] = NIL；

将$\{q_0\}$加入$\mathbb{Q}$ 且未标记；

while ($\mathbb{Q}$ 中存在一个未标记元素S) {

标记S；

for $(a\in\Sigma)$\{

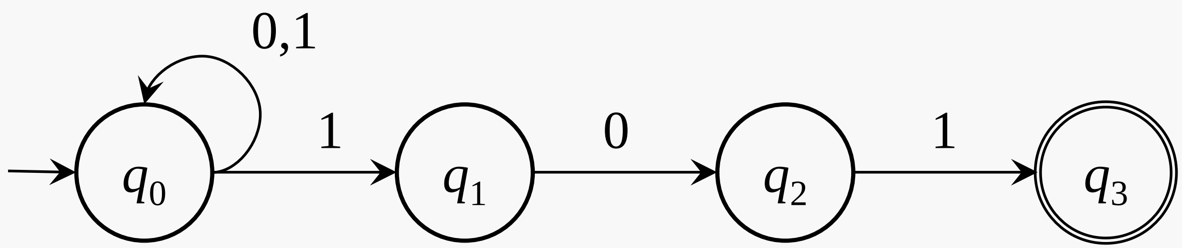
$T = \bigcup q\in S \cdot \upsilon(q, a)$;

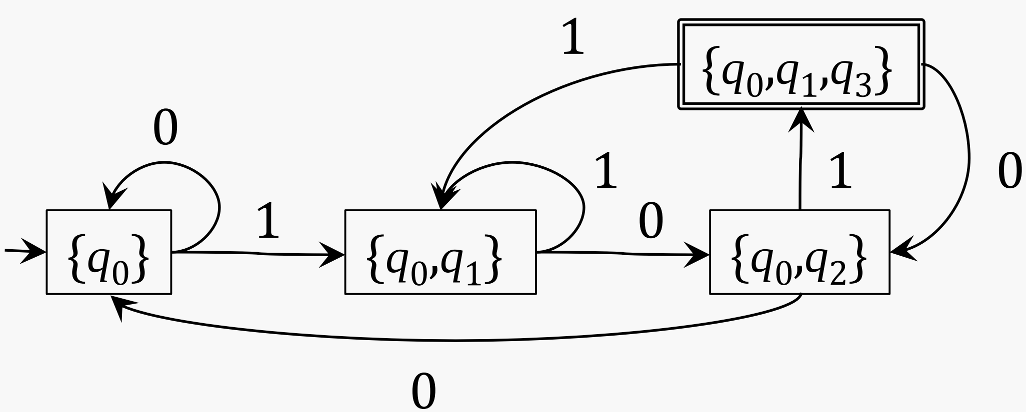
if $(T\notin\mathbb{Q})$ 将T 加入$\mathbb{Q}$ 中且未标记；

move$[S, a] = T$ }

}

子集法示例

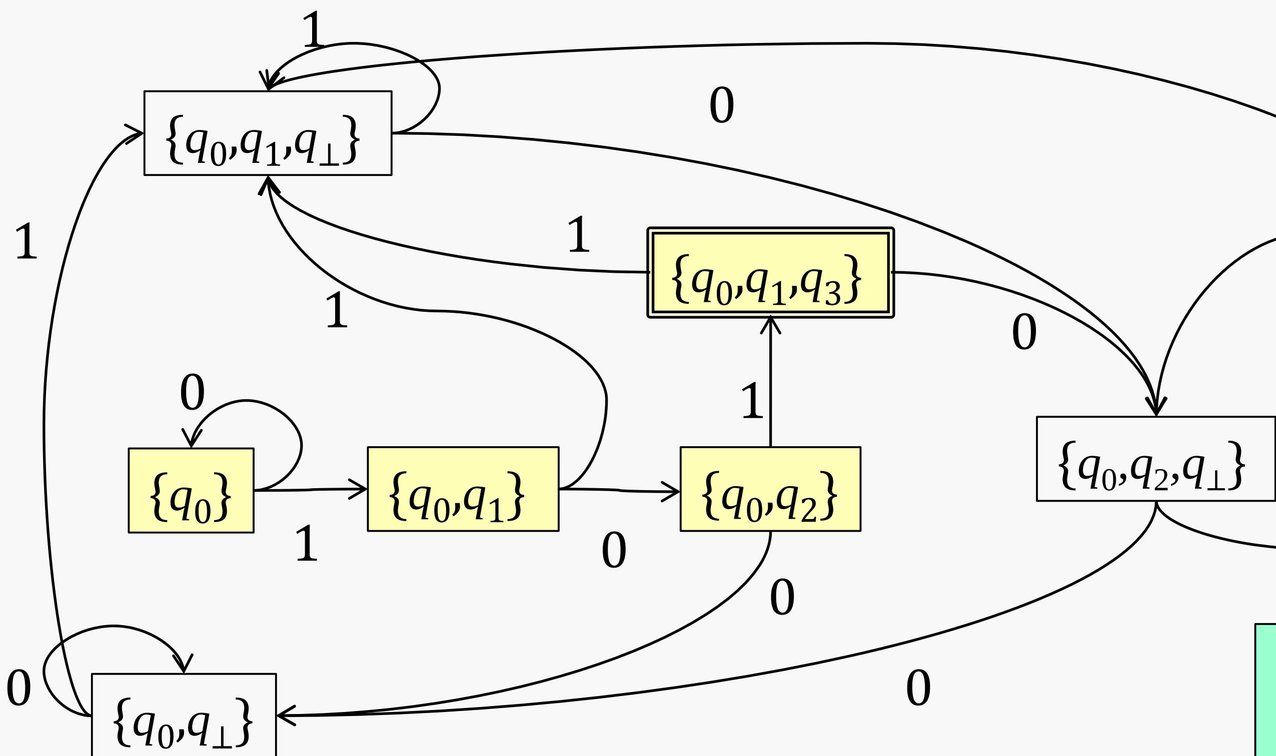




	0	1
{0}	{0}	{0,1}
{0,1}	{0,2}	{0,1}
{0,2}	{0}	{0,1,3}
{0,1,3}	{0,2}	{0,1}

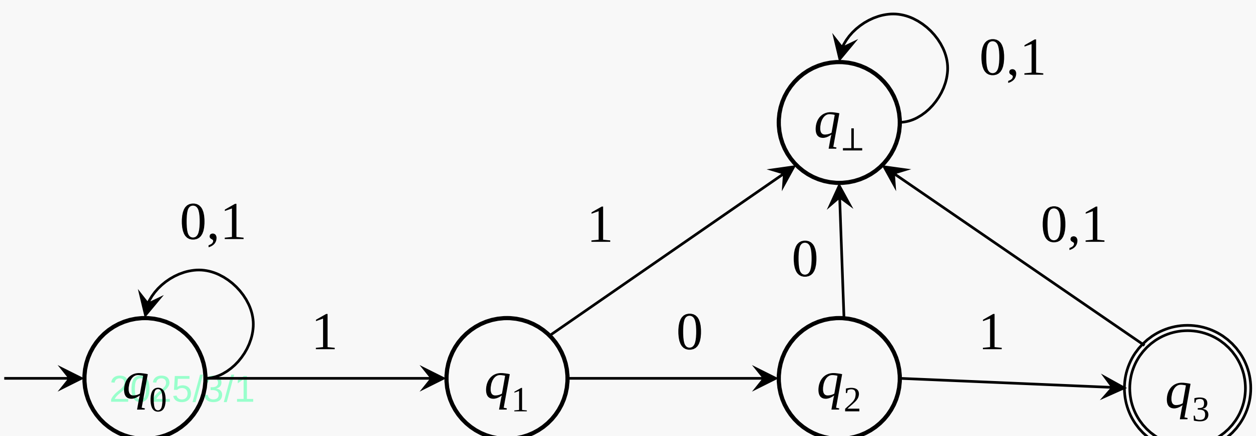
括号中数字i代表状态q_i

子集法示例



括号中符号i代表状态q_i

	0	1
{0}	{0}	{0,1}
{0,1}	{0,2}	{0,1,⊥}
{0,2}	{0,⊥}	{0,1,3}
{0,1,⊥}	{0,2,⊥}	{0,1,⊥}
{0,⊥}	{0,⊥}	{0,1,⊥}
{0,1,3}	{0,2,⊥}	{0,1,⊥}
{0,2,⊥}	{0,⊥}	{0,1,3,⊥}
{0,1,3,⊥}	{0,2,⊥}	{0,1,⊥}

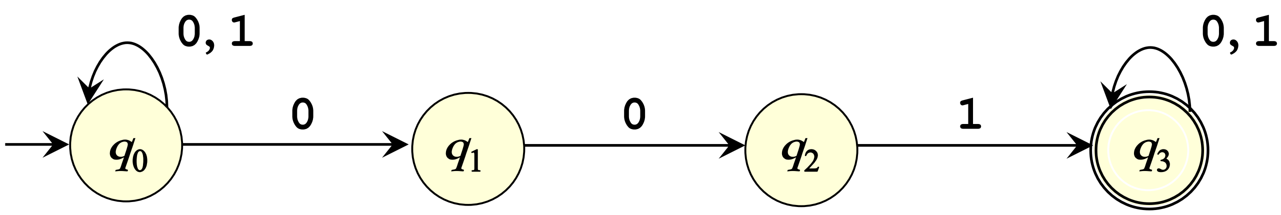


子集构造法的正确性

- 定理2.2: 若 DFA D 是从NFA N 通过子集构造法构造而成，则 $L(D) = L(N)$。
- 依照w长度归纳 $\tilde{\upsilon}_N(q_0, w) = \tilde{\upsilon}_D(\{q_0\}, w)$
  - 基础：对于$w = \varepsilon$，$\tilde{\upsilon}_N(q_0, \varepsilon) = \tilde{\upsilon}_D(\{q_0\}, \varepsilon) = \{q_0\}$。
  - 归纳步：假定归纳假设（IH）是对短于w的串成立。令$w = xa$，则IH对于x成立。
  - 那么，令$\tilde{\upsilon}_N(q_0, x) = \tilde{\upsilon}_D(\{q_0\}, x) = S$
  - 令$T = \bigcup_{p \in S} \cdot \upsilon_N(p, a)$
  - 则根据子集构造法知$\tilde{\upsilon}_D(\{q_0\}, w) = \upsilon_D(\tilde{\upsilon}_D(\{q_0\}, x), a) = \upsilon_D(S, a) = \bigcup_{p \in S} \cdot \upsilon_N(p, a) = T$
  - 同时根据定义知$\tilde{\upsilon}_N(q_0, w) = \bigcup_{p \in \tilde{\upsilon}_N(q_0, x)} \cdot \upsilon_N(p, a) = T$
  - 得证。

子集法举例

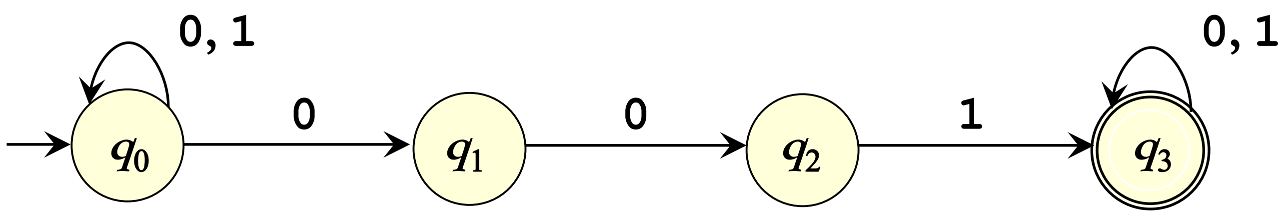
- 识别包含11子串的0-1串，NFA转换为DFA



	0	1
$\{q_0\}$		

子集法举例

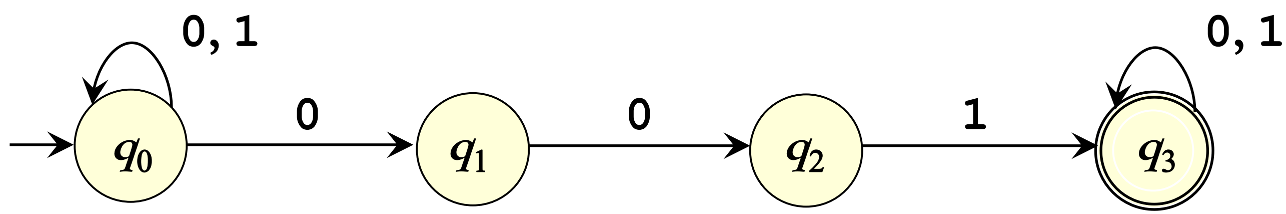
- 识别包含001子串的0-1串，NFA转换为DFA



	0	1
$\{q_0\}$	$\{q_0,q_1\}$	$\{q_0\}$

子集法举例

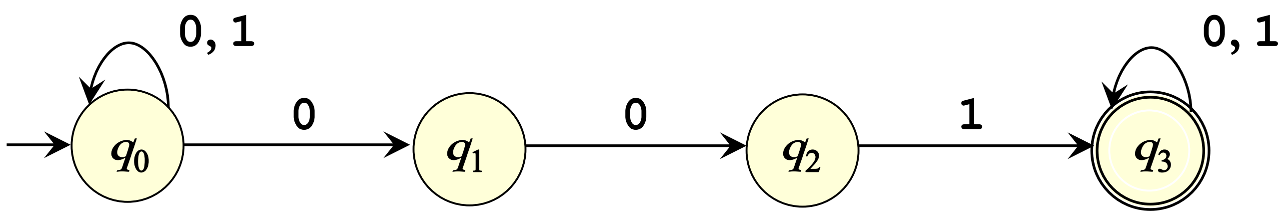
- 识别包含001子串的0-1串，NFA转换为DFA



	0	1
$\{q_0\}$	$\{q_0,q_1\}$	$\{q_0\}$
$\{q_0,q_1\}$		

子集法举例

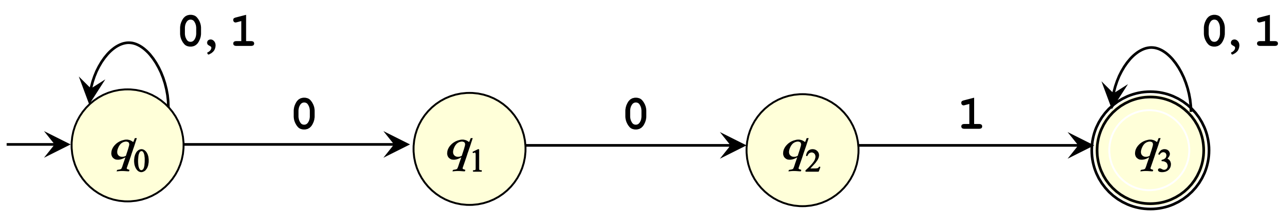
- 识别包含001子串的0-1串，NFA转换为DFA



	0	1
$\{q_0\}$	$\{q_0,q_1\}$	$\{q_0\}$
$\{q_0,q_1\}$	$\{q_0,q_1,q_2\}$	$\{q_0\}$

子集法举例

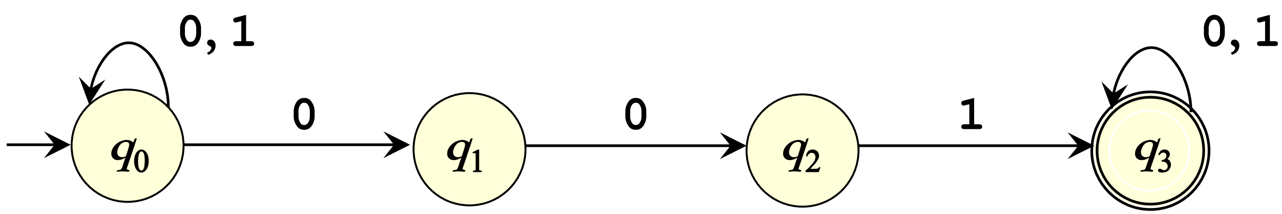
- 识别包含001子串的0-1串，NFA转换为DFA



	0	1
$\{q_0\}$	$\{q_0,q_1\}$	$\{q_0\}$
$\{q_0,q_1\}$	$\{q_0,q_1,q_2\}$	$\{q_0\}$
$\{q_0,q_1,q_2\}$		

子集法举例

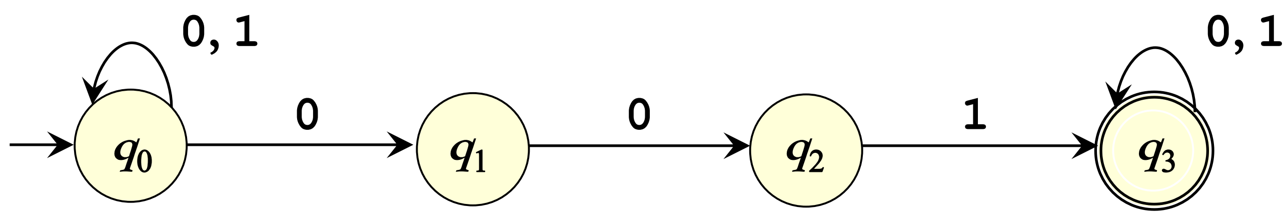
- 识别包含001子串的0-1串，NFA转换为DFA



	0	1
$\{q_0\}$	$\{q_0,q_1\}$	$\{q_0\}$
$\{q_0,q_1\}$	$\{q_0,q_1,q_2\}$	$\{q_0\}$
$\{q_0,q_1,q_2\}$	$\{q_0,q_1,q_2\}$	$\{q_0,q_3\}$
$\{q_0,q_3\}$	$\{q_0,q_1,q_3\}$	$\{q_0,q_3\}$
$\{q_0,q_1,q_3\}$	$\{q_0,q_1,q_2,q_3\}$	$\{q_0,q_3\}$
$\{q_0,q_1,q_2,q_3\}$	$\{q_0,q_1,q_2,q_3\}$	$\{q_0,q_3\}$

子集法举例

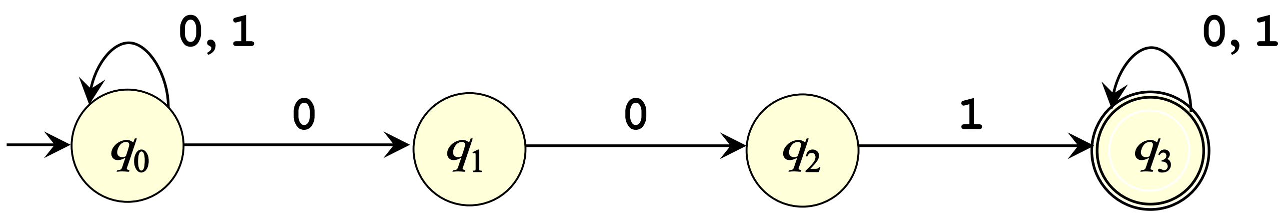
- 识别包含001子串的0-1串，NFA转换为DFA



	0	1
$>\{q_0\}$	$\{q_0,q_1\}$	$\{q_0\}$
$\{q_0,q_1\}$	$\{q_0,q_1,q_2\}$	$\{q_0\}$
$\{q_0,q_1,q_2\}$	$\{q_0,q_1,q_2\}$	$\{q_0,q_3\}$
$*\{q_0,q_3\}$	$\{q_0,q_1,q_3\}$	$\{q_0,q_3\}$
$*\{q_0,q_1,q_3\}$	$\{q_0,q_1,q_2,q_3\}$	$\{q_0,q_3\}$
$*\{q_0,q_1,q_2,q_3\}$	$\{q_0,q_1,q_2,q_3\}$	$\{q_0,q_3\}$

子集法举例

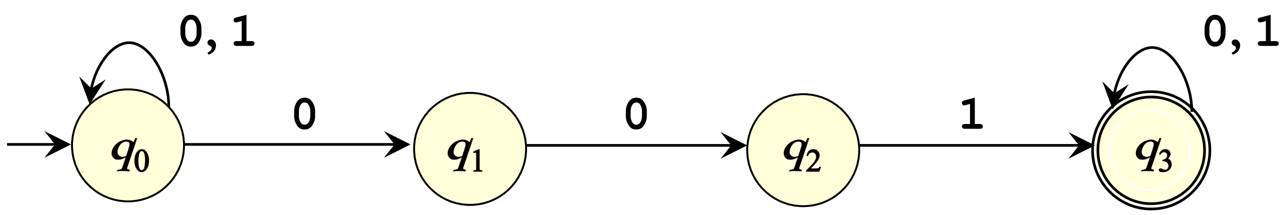
- 识别包含001子串的0-1串，NFA转换为DFA



	0	1
>$\{q_0\}/p_0$	$\{q_0,q_1\}\ /p_1$	$\{q_0\}\ /p_0$
$\{q_0,q_1\}/p_1$	$\{q_0,q_1,q_2\}\ /p_2$	$\{q_0\}\ /p_0$
$\{q_0,q_1,q_2\}/p_2$	$\{q_0,q_1,q_2\}\ /p_2$	$\{q_0,q_3\}\ /p_3$
*$\{q_0,q_3\}/p_3$	$\{q_0,q_1,q_3\}\ /p_4$	$\{q_0,q_3\}\ /p_3$
*$\{q_0,q_1,q_3\}/p_4$	$\{q_0,q_1,q_2,q_3\}\ /p_5$	$\{q_0,q_3\}\ /p_3$
*$\{q_0,q_1,q_2,q_3\}/p_5$	$\{q_0,q_1,q_2,q_3\}\ /p_5$	$\{q_0,q_3\}\ /p_3$

子集法NFA转DFA结果

- 识别包含001子串的0-1串，NFA转换为DFA



Command:

List of States:

State[0]
State[1]
State[2]
State[3] - Final state
State[4] - Final state
State[5] - Final state

List of Transitions:

State[0] --> 0 --> State[1]
State[0] --> 1 --> State[0]

State[1] --> 0 --> State[2]
State[1] --> 1 --> State[0]

State[2] --> 0 --> State[2]
State[2] --> 1 --> State[3]

State[3] --> 0 --> State[4]
State[3] --> 1 --> State[3]

State[4] --> 0 --> State[5]
State[4] --> 1 --> State[3]

	0	1
$\rightarrow p_0$	p_1	p_0
p_1	p_2	p_0
p_2	p_2	p_3
$*p_3$	p_4	p_3
$*p_4$	p_5	p_3
$*p_5$	p_5	p_3

Automata Design

Input sentence: 010011 Execute

> Input= 010011
> Executed Transitions:

State[0] --> 0 --> State[1]
State[1] --> 1 --> State[0]
State[0] --> 0 --> State[1]
State[1] --> 0 --> State[2]
State[2] --> 1 --> State[3]
State[3] --> 1 --> State[3]

> Results:

- State[3] is a Final State
- Input ACCEPTED

Input sentence: 0101011 Execute

> Input= 0101011
> Executed Transitions:

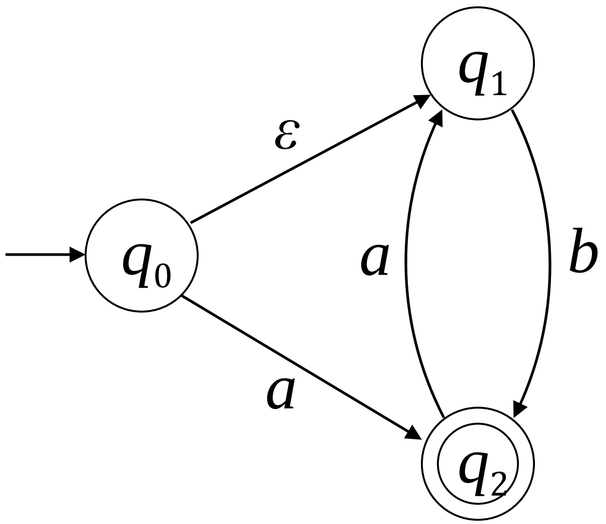
State[0] --> 0 --> State[1]
State[1] --> 1 --> State[0]
State[0] --> 0 --> State[1]
State[1] --> 1 --> State[0]
State[0] --> 0 --> State[1]
State[1] --> 1 --> State[0]
State[0] --> 1 --> State[0]

> Results:

- State[0] is NOT a Final State
- Input REJECTED

ε转移与ε-NFA

- ε转移：不消耗输入符号发生状态转移
- 用符号ε标记ε转移。含有ε转移的NFA就是ε-NFA



接受：

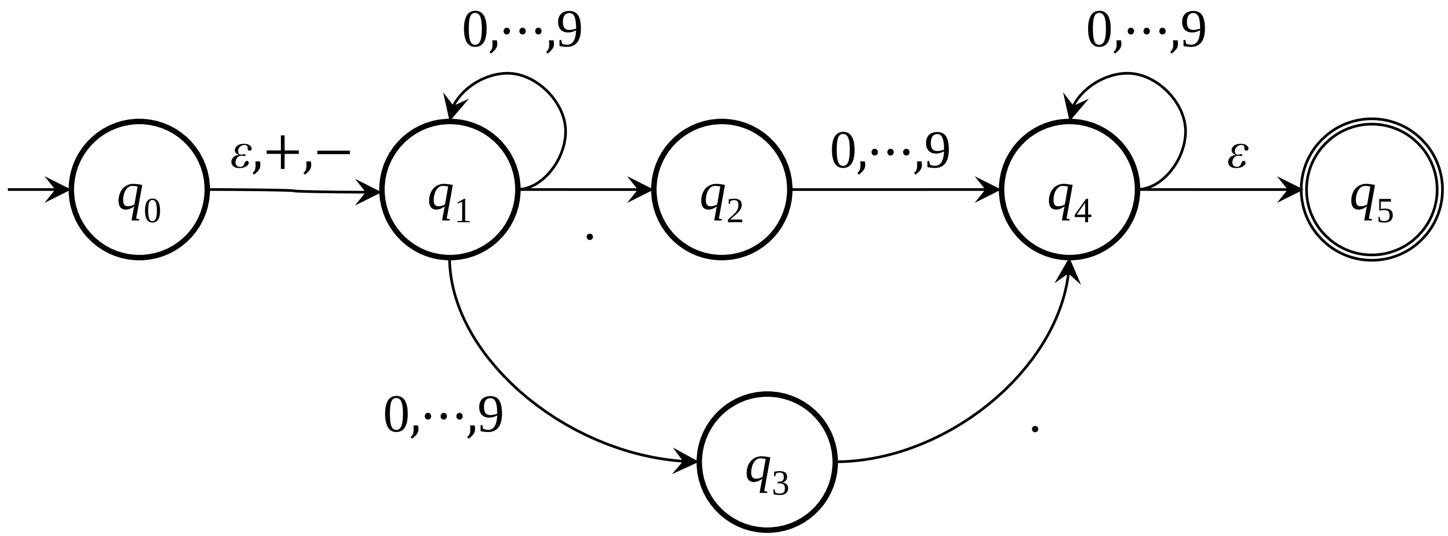
$a, b, aab, bab, aabab, \ldots$

拒绝：

$\varepsilon, aa, ba, bb, \ldots$

观察活动状态集变化情况：初始为 $\{q_0, q_1\}$；遇到a或b都转移到$\{q_2\}$；$\{q_2\}$遇到a转移到$\{q_1\}$。

例：有符号十进制定点数



由四部分依次组成：

（1）+号，−号或空；

（2）数字串或空；

（3）小数点；

（4）数字串或空。

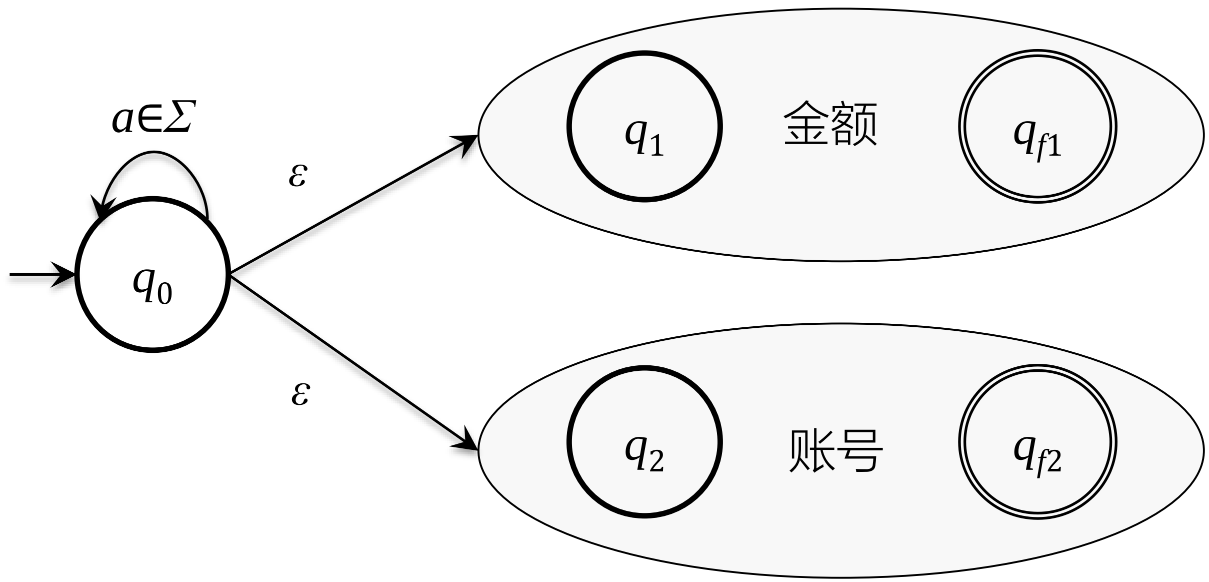
限定（2）和（4）不能同时为空。

精化：

1）无前0、后0

2）可无小数点

例：识别金额和账号的NFA

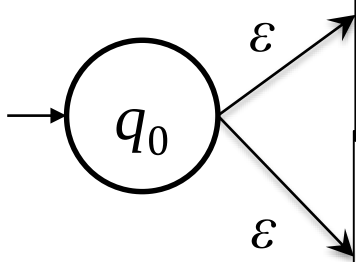


识别的这个符号串可能是金额，也可能是账号。类似地：识别这样的0-1符号串，它或者包含偶数个0或者包含奇数个1

	0	1
$\rightarrow{}^*e_0$	e	e_0
e	e_0	e

	0	1
$\rightarrow o$	o	o_1
$^*o_1$	o_1	o

例：识别串含有偶数个0或者奇数个1



	0	1
$\rightarrow * e_0$	e	e_0
e	e_0	e

	0	1
$\rightarrow o$	o	o_1
$* o_1$	o_1	o

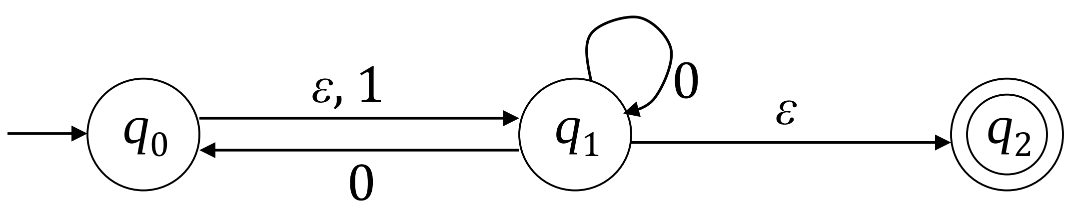
	0	1	ε
$\rightarrow q_0$	$\emptyset$	$\emptyset$	$\{e_0, o\}$
$* e_0$	$\{e\}$	$\{e_0\}$	$\emptyset$
e	$\{e_0\}$	$\{e\}$	$\emptyset$
o	$\{o\}$	$\{o_1\}$	$\emptyset$
$* o_1$	$\{o_1\}$	$\{o\}$	$\emptyset$

状态的ε闭包

- 状态q的ε闭包，记为$\omega(q)$，指自身以及经过连续ε转移所能到达的状态的集合（不消耗输入）

	0	1	ε
$\rightarrow q_0$	$\emptyset$	$\{q_1\}$	$\{q_1\}$
q_1	$\{q_0,q_1\}$	$\emptyset$	$\{q_2\}$
$*q_2$	$\emptyset$	$\emptyset$	$\emptyset$

	$\omega()$
q_0	$\{q_0,q_1,q_2\}$
q_1	$\{q_1,q_2\}$
q_2	$\{q_2\}$



ε-NFA的ε闭包与ε闭集

- 状态q的ε闭包：$\omega(q)$
- 状态集合S的ε闭包：$\omega(S)=\cup q\in S\cdot\omega(q)$
  - 状态集合S为ε闭集当且仅当$S=\omega(S)$
  - 对任意状态集合S，$\omega(S)$是ε闭集
- ε闭集概念扩展了活动状态集概念用于实现ε-NFA判定性质

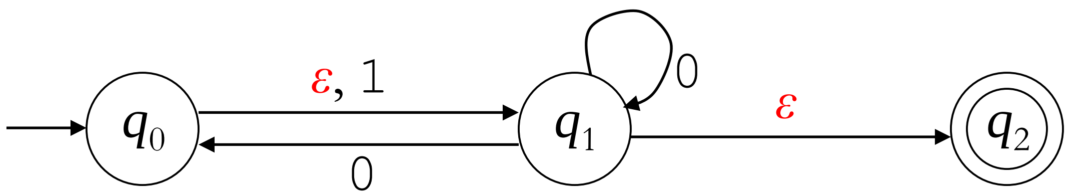
例：

	0	1	ε
$\rightarrow q_0$	$\emptyset$	$\{q_1\}$	$\{q_1\}$
q_1	$\{q_0,q_1\}$	$\emptyset$	$\{q_2\}$
$*q_2$	$\emptyset$	$\emptyset$	$\emptyset$

S	$\omega(S)$
$\emptyset$	$\emptyset$
$\{q_0\}$	$\{q_0,q_1,q_2\}$
$\{q_1\}$	$\{q_1,q_2\}$
$\{q_2\}$	$\{q_2\}$
$\{q_0,q_1\}$	$\{q_0,q_1,q_2\}$
$\{q_0,q_2\}$	$\{q_0,q_1,q_2\}$
$\{q_1,q_2\}$	$\{q_1,q_2\}$
$\{q_0,q_1,q_2\}$	$\{q_0,q_1,q_2\}$

ε闭集与ε-NFA判定性质

输入串：ε；00；001；101；11



	0	1	ε
$\rightarrow q_0$	$\emptyset$	$\{q_1\}$	$\{q_1\}$
q_1	$\{q_0,q_1\}$	$\emptyset$	$\{q_2\}$
$*q_2$	$\emptyset$	$\emptyset$	$\emptyset$

	$\omega()$
$\emptyset$	$\emptyset$
$\{q_0\}$	$\{q_0,q_1,q_2\}$
$\{q_1\}$	$\{q_1,q_2\}$
$\{q_2\}$	$\{q_2\}$
$\{q_0,q_1\}$	$\{q_0,q_1,q_2\}$
$\{q_0,q_2\}$	$\{q_0,q_1,q_2\}$
$\{q_1,q_2\}$	$\{q_1,q_2\}$
$\{q_0,q_1,q_2\}$	$\{q_0,q_1,q_2\}$

已读前缀x；剩余串ay；
当前输入符号a；
当前状态集合$S=\tilde{\upsilon}(q_0, x)$

输入串xay

$$\boxed{S} \xrightarrow{a} \boxed{T}$$

已读前缀xa；剩余串y；
转移状态集合：
$T=\omega(\cup p\in S\cdot \upsilon(p,a))$

扩展的转移函数

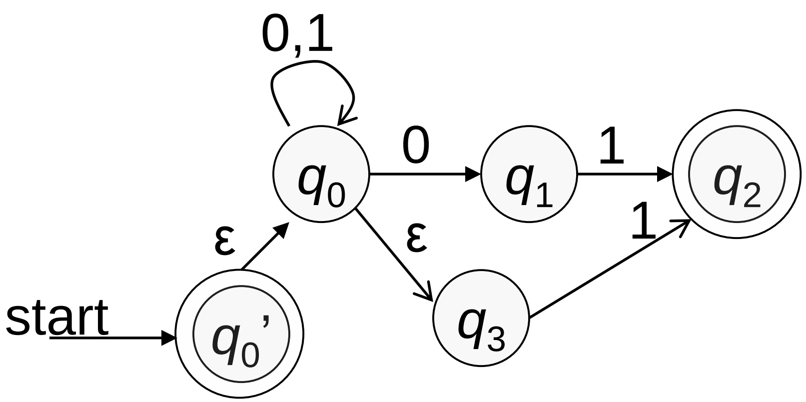
- **基础：** $\tilde{\upsilon}(q, \varepsilon) = \omega(q)$
- **归纳：** $\tilde{\upsilon}(q, xa) = ?$
  - 令： $\tilde{\upsilon}(q, x) = S$
  - 令： $T = \bigcup_{p \in S} \cdot \upsilon(p, a)$
  - 则： $\tilde{\upsilon}(q, xa) = \bigcup_{p \in T} \cdot \omega(p)$， 或者，

$$\tilde{\upsilon}(q, xa) = \omega(T)$$

- $\tilde{\upsilon}(q, w)$ 是始端为q，标记为w 的路径的末端之集合。
  - 注意，路径标记w和弧的标记a、ε的关系
- ε-NFA语言：
  - 对于ε-NFA $B = (Q, \Sigma, \upsilon, q_0, F)$，
  - 语言为$L(B) = \{w \in \Sigma^* \mid \tilde{\upsilon}(q_0, w) \cap F \neq \emptyset\}$

扩展转移函数的例子

$\tilde{v}(q,\varepsilon) = \omega(q)$；
$\tilde{v}(q,xa) = \omega(\bigcup_{r\in\tilde{v}(q,x)} v(r,a))$



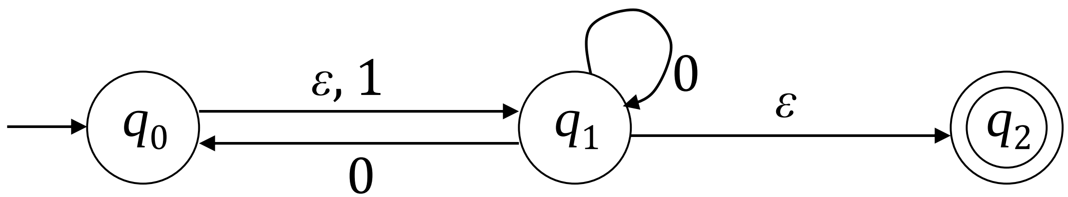
	0	1	ε
$\rightarrow{}^*q_0'$	$\emptyset$	$\emptyset$	$\{q_0\}$
q_0	$\{q_0,q_1\}$	$\{q_0\}$	$\{q_3\}$
q_1	$\emptyset$	$\{q_2\}$	$\emptyset$
$^*q_2$	$\emptyset$	$\emptyset$	$\emptyset$
q_3	$\emptyset$	$\{q_2\}$	$\emptyset$

- 模拟w =101?

$$\tilde{v}(q_0', 101) = \bigcup_{x\in\tilde{v}(q_0', 10)} \omega v(x, 1)$$
$$= \bigcup_{x\in(\bigcup_{y\in\tilde{v}(q_0', 1)} \omega v(y, 0))} \omega v(x, 1)$$
$$= \bigcup_{x\in(\bigcup_{y\in(\bigcup_{z\in\tilde{v}(q_0', \varepsilon)} \omega v(z, 1)))} \omega v(y, 0))} \omega v(x, 1)$$
$$= \bigcup_{x\in(\bigcup_{y\in(\bigcup_{z\in\{q_0', q_0, q_3\}} \omega v(z, 1))} \omega v(y, 0))} \omega v(x, 1)$$
$$= \bigcup_{x\in(\bigcup_{y\in\{q_0, q_2, q_3\}} \omega v(y, 0))} \omega v(x, 1)$$
$$= \bigcup_{x\in\{q_0, q_1, q_3\}} \omega v(x, 1) = \{q_0, q_2, q_3\}$$

消除ε转移（转为NFA）

- ε-NFA $(Q, \Sigma, \upsilon, q_0, F)$转为NFA $(Q, \Sigma, \text{move}[], q_0, F)$
- $\text{move}[q, a]=\omega(\cup p\in\omega(q)\cdot\upsilon(p, a)), q\in Q, a\in\Sigma$
- 然后消除NFA中不可达情形①和②即得结果。

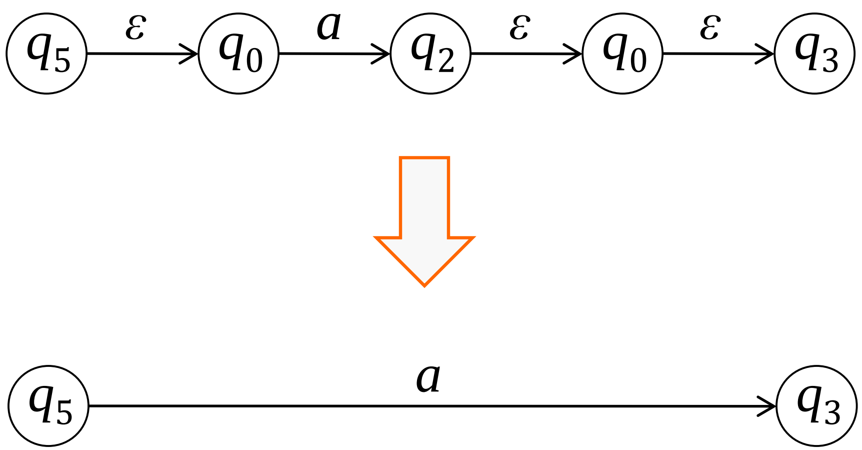


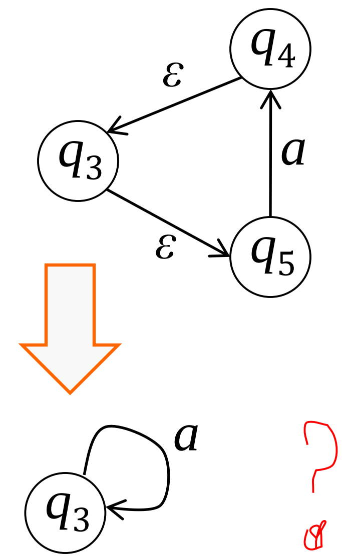
	0	1	ε
$\rightarrow q_0$	$\emptyset$	$\{q_1\}$	$\{q_1\}$
q_1	$\{q_0,q_1\}$	$\emptyset$	$\{q_2\}$
$*q_2$	$\emptyset$	$\emptyset$	$\emptyset$

	0	1
$\rightarrow q_0$	$\{q_0,q_1,q_2\}$	$\{q_1,q_2\}$
q_1	$\{q_0,q_1,q_2\}$	$\emptyset$
$*q_2$	$\emptyset$	$\emptyset$

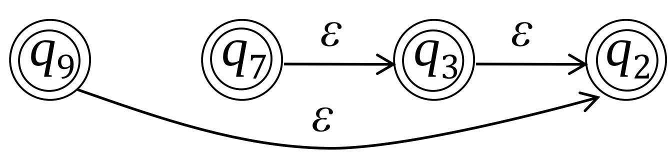
减少ε转移

- 带有ε弧的路径用单一的转移替代





- 经ε弧到达原终结状态的所有状态都是结束状态



ε-NFA 到 DFA 的子集构造法

输入：ε-NFA $(Q, \Sigma, \upsilon, q_0, F)$。

输出：DFA $(\mathbb{Q}, \Sigma, \text{move}[], \omega(q_0), \{S\in\mathbb{Q} \mid S\cap F\neq\emptyset\})$ 。

$\mathbb{Q} = \emptyset$; move[] = NIL；

$\omega(\{q_0\})$加入$\mathbb{Q}$ 且未标记；

while $\mathbb{Q}$ 中存在一个未标记元素S {

　　标记S；

　　for ($a\in\Sigma$) {

　　　　$T = \omega(\bigcup q\in S \cdot \upsilon(q, a))$;

　　　　if ($T\notin\mathbb{Q}$) T加入$\mathbb{Q}$ 中且未标记；

　　　　move[S, a] = T

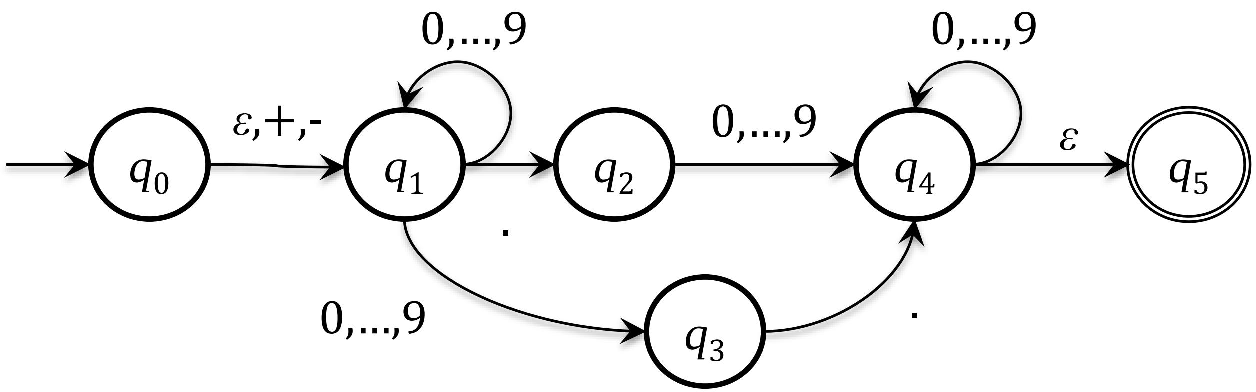
　　}

}。

	0	1	ε
$\to q_0$	$\emptyset$	$\{q_1\}$	$\{q_1\}$
q_1	$\{q_0,q_1\}$	$\emptyset$	$\{q_2\}$
$*q_2$	$\emptyset$	$\emptyset$	$\emptyset$

	0	1
$\to*\{q_0,q_1,q_2\}$	$\{q_0,q_1,q_2\}$	$\{q_1,q_2\}$
$*\{q_1,q_2\}$	$\{q_0,q_1,q_2\}$	$\emptyset$

子集法举例



	0-9	+,-	.
→$\{q_0,q_1\}$	$\{q_1,q_3\}$	$\{q_1\}$	$\{q_2\}$
$\{q_1,q_3\}$	$\{q_1,q_3\}$	$\emptyset$	$\{q_2,q_4,q_5\}$
$\{q_1\}$	$\{q_1,q_3\}$	$\emptyset$	$\{q_2\}$
$\{q_2\}$	$\{q_4,q_5\}$	$\emptyset$	$\emptyset$
*$\{q_2,q_4,q_5\}$	$\{q_4,q_5\}$	$\emptyset$	$\emptyset$
*$\{q_4,q_5\}$	$\{q_4,q_5\}$	$\emptyset$	$\emptyset$

子集构造法的正确性

- 定理2.3: 语言L 被某个ε-NFA接受当且仅当L 为某个DFA接受。
- 依照w 长度归纳 $\tilde{\upsilon}_N(q_0, w)=\tilde{\upsilon}_D(\omega(\{q_0\}), w)$。
  - 基础：对于$w=\varepsilon$，有 $\tilde{\upsilon}_N(q_0, \varepsilon)=\tilde{\upsilon}_D(\omega(\{q_0\}), \varepsilon)=\omega(\{q_0\})$。
  - 归纳：归纳假设IH对所有短于w 的串成立。
    令$w=xa$，由IH知对于x 成立，令其为
    $$\tilde{\upsilon}_N(q_0, x)=\tilde{\upsilon}_D(\omega(\{q_0\}), x)=S。$$
  - 那么有如下推导：
    ① 根据定义$\tilde{\upsilon}_N(q_0, xa)=\bigcup_{p\in S}\cdot\omega(\upsilon_N(p, a))=\omega(\bigcup_{p\in S}\cdot\upsilon_N(p, a))$，令为$T$;
    ② 则根据子集构造法有 $\upsilon_D(S, a)=\bigcup_{p\in S}\cdot\omega(\upsilon_N(p, a))=T$;
    ③ 那么$\tilde{\upsilon}_D(\omega(\{q_0\}), xa)=\upsilon_D(\tilde{\upsilon}_D(\omega(\{q_0\}), x), a)=\upsilon_D(S, a)=T=\tilde{\upsilon}_N(q_0, w)$。 得证。
  - 结论：DFA ≡ NFA ≡ ε-NFA，都接受正则语言

小结

- 知识点：NFA（包括ε-NFA）转DFA的子集构造法、状态的ε闭包、状态集合的ε闭包、ε闭集、扩展转移函数、NFA语言、NFA三种表示。
- 形式化记号：转移函数$\upsilon()$、扩展转移函数$\tilde{\upsilon}()$、ε闭包$\omega()$、集合运算$\bigcup_{q\in S} \cdot \upsilon(q, a)$。
- 作业：2.5节习题2.6~2.7