第五讲

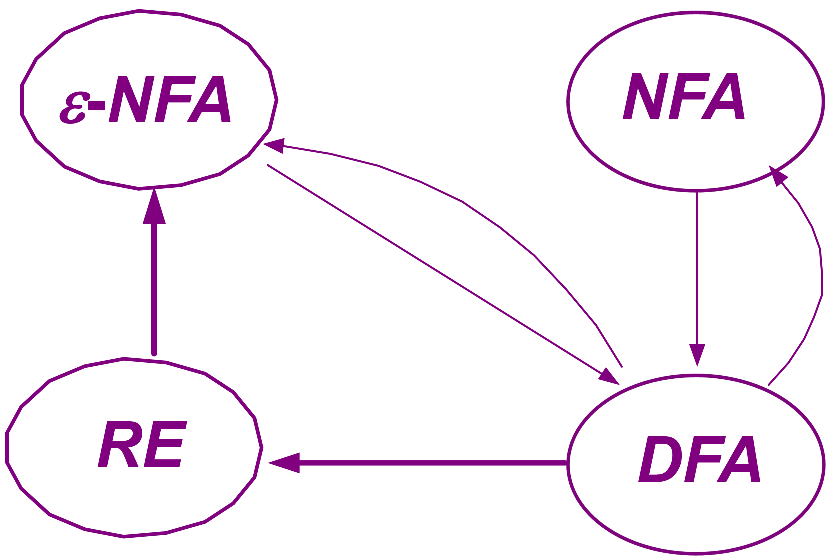
- ✧ 有限状态自动机 ⇔ 正规表达式

有限状态自动机 ⇔ 正规表达式

- ✧ 有限自动机与正规表达式的关系
- ✧ 几个转换算法的复杂度（选讲）

◇ 结论:有限自动机所表示的语言是正规语言

– 证明策略



✧ 从正规表达式构造等价的ε - NFA

– 定理: *L* 是正规表达式 *R* 表示的语言, 则存在一个 ε - NFA *E* ,满足 *L(E)* = *L(R)* = *L*.

证明: 构造性证明. 可以通过结构归纳法证明从 *R* 可以构造出与其等价的，满足如下条件的ε - NFA :

(1) 恰好一个终态;

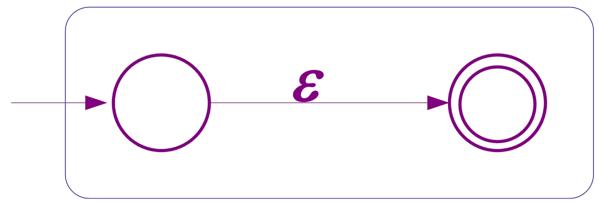
(2) 没有弧进入初态;

(3) 没有弧离开终态;

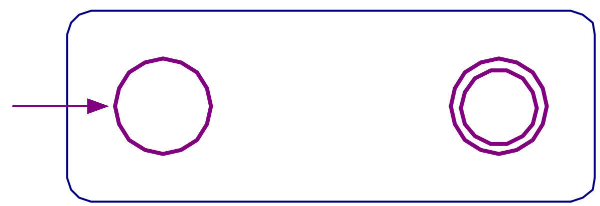
✧ 归纳构造过程 (从正规表达式构造等价的ε - *NFA*)（*Thompson* 构造法）

– 基础:

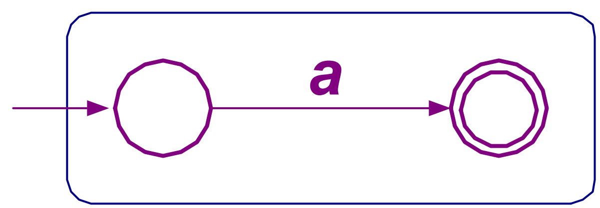
1 对于 ε，构造为



2 对于 ϕ，构造为



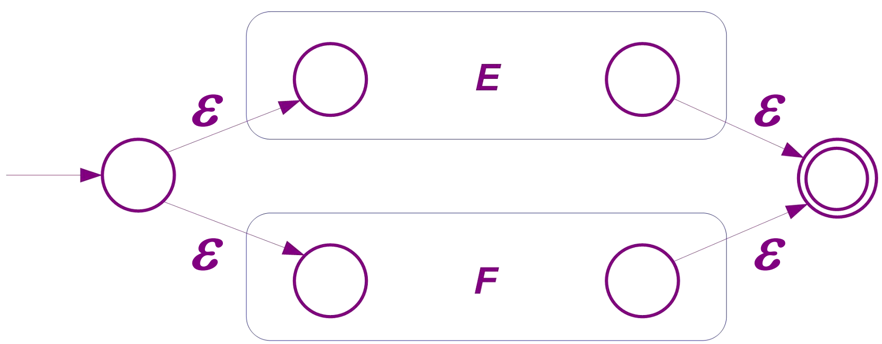
3 对于 ***a***，构造为



✧ 归纳构造过程 (从正规表达式构造等价的ε - *NFA*)（*Thompson* 构造法）

– 归纳:

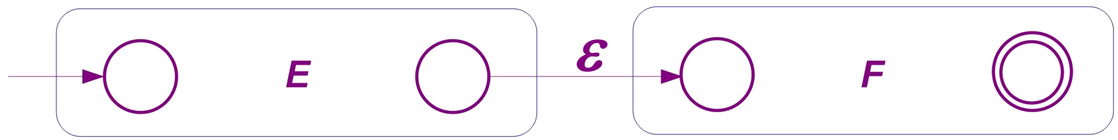
1 对于 ***E+F*****，**构造为



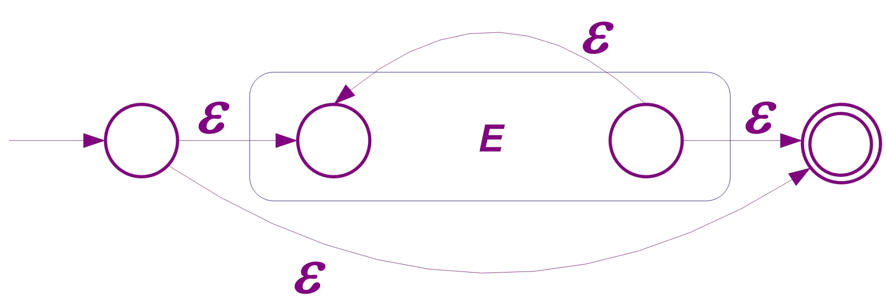
✧ 归纳构造过程 (从正规表达式构造等价的ε - ***NFA***)
（*Thompson* 构造法）

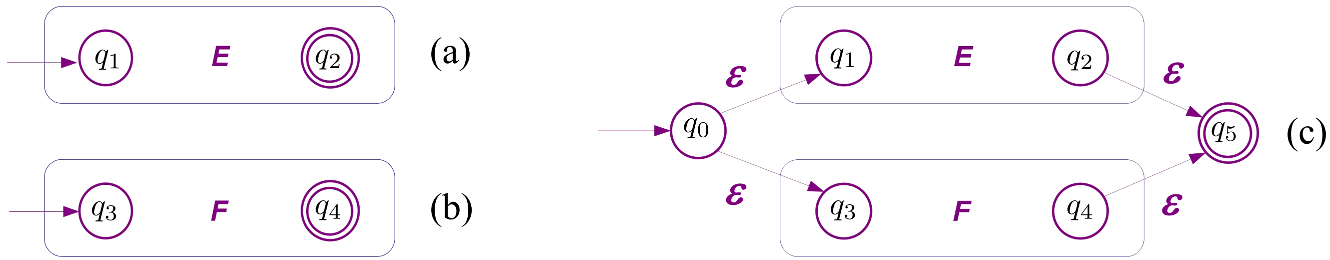
– 归纳:

2 对于 ***EF***，构造为



3 对于 ***E****，构造为





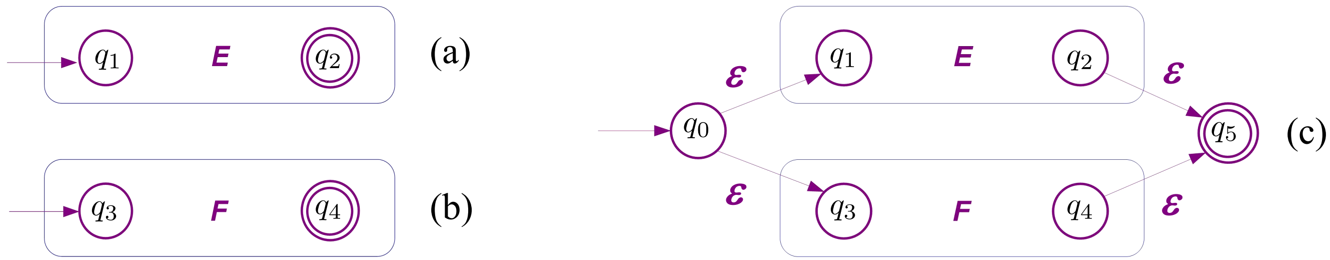
例1：令正规表达式E所对应的ϵ-NFA如图(a)所示，正规表达式F对应的ϵ-NFA如图(b)所示，证明正规表达式$E+F$的表达式如图(c)所示。

证明：考察正规表达式$G=E+F$。令n表示正规表示式中所包含的运算符的个数。我们略过$n=0$的基础情况，重点看归纳情况。假设结论对$n\geq k(k\geq 0)$成立，此时E和F中运算符的个数不大于k。根据归纳假设，存在符合要求的ϵ-NFA：

$$M_E=(Q_1,\Sigma,\delta_1,q_1,\{q_2\})$$
$$M_F=(Q_2,\Sigma,\delta_2,q_3,\{q_4\})$$

使得$L(M_E)=L(E)$，$L(M_F)=L(F)$，并且$Q_1\cap Q_2=\emptyset$。

有限自动机与正规表达式的关系



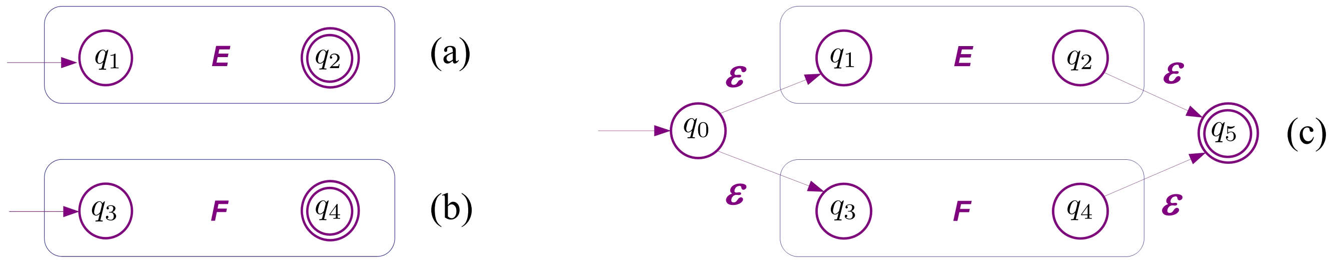
证明：取$q_0, q_5 \notin Q_1 \cup Q_2$，构造一个新的$\epsilon$-NFA：

$$M = (Q_1 \cup Q_2 \cup \{q_0, q_5\}, \Sigma, \delta, q_0, \{q_5\})$$

其中，转移函数的定义如下

1. $\delta(q_0, \epsilon) = \{q_1, q_3\}$
2. $\forall q \in Q_1 - \{q_2\}, \forall a \in \Sigma \cup \{\epsilon\} : \delta(q, a) = \delta_1(q, a)$
3. $\forall q \in Q_2 - \{q_4\}, \forall a \in \Sigma \cup \{\epsilon\} : \delta(q, a) = \delta_2(q, a)$
4. $\delta(q_2, \epsilon) = q_5$
5. $\delta(q_4, \epsilon) = q_5$

有限自动机与正规表达式的关系



证明：需要证明$L(E+F)=L(M)$。根据归纳假设，$L(E+F)=L(E)\cup L(F)=L(M_E)\cup L(M_F)$，因此问题转化为证明：

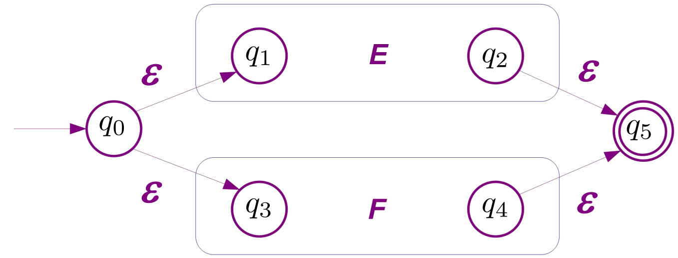
$$L(M)=L(M_E)\cup L(M_F)$$

首先证明$L(M_E)\cup L(M_F)\subseteq L(M)$。设$x\in L(M_E)\cup L(M_F)$，则$x\in L(M_E)$或$x\in L(M_F)$。当$x\in L(M_E)$时，有$\delta_1'(q_1,x)=\{q_2\}$。根据$M$的定义：

$$\begin{aligned}
\delta'(q_0,x) &= \delta'(q_0,\epsilon x\epsilon)\\
&= \delta'(\delta(q_0,\epsilon),x\epsilon)\\
&= \delta'(\{q_1,q_3\},x\epsilon)\\
&= \delta'(q_1,x\epsilon)\cup\delta'(q_3,x\epsilon)
\end{aligned}$$

(a)

(b)



(c)

证明：

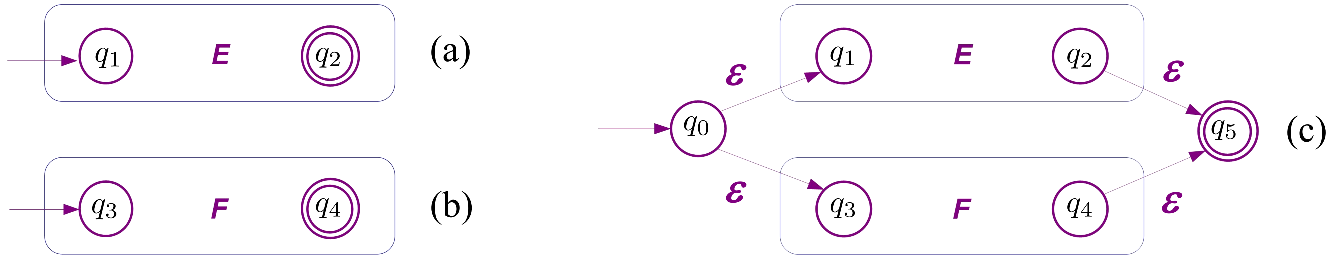
$$
\begin{aligned}
&= \delta(\delta'(q_1,x),\epsilon)\cup\delta(\delta'(q_3,x),\epsilon)\\
&= \delta(\delta_1'(q_1,x),\epsilon)\cup\delta(\delta_2'(q_3,x),\epsilon)\\
&= \delta(q_2,\epsilon)\cup\delta(\delta'(q_3,x),\epsilon)\\
&= \{q_5\}\cup\delta(\delta'(q_3,x),\epsilon)
\end{aligned}
$$

即$x\in L(M)$。

同理可证，当$x\in L(M_F)$时，$x\in L(M)$。因此，可以得到

$$L(M_E)\cup L(M_F)\subseteq L(M)$$

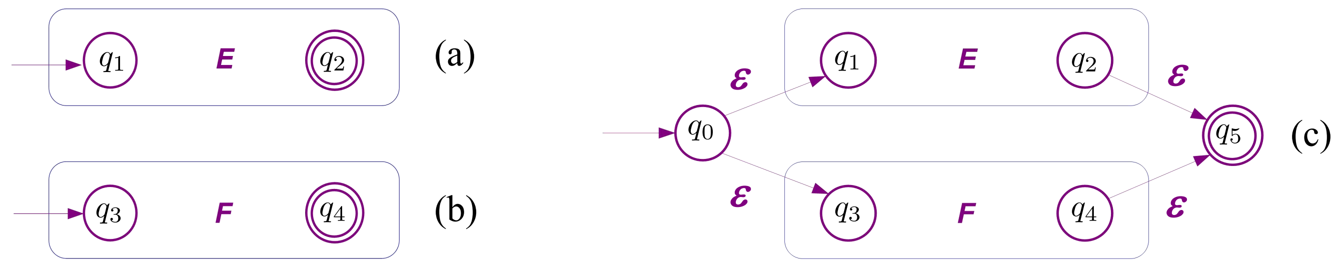
有限自动机与正规表达式的关系



证明：再证$L(M) \subseteq L(M_E) \cup L(M_F)$。

设$x \in L(M)$，也就是说$q_5 \in \delta'(q_0, x)$。按照M的定义

$$
\begin{aligned}
\delta'(q_0, x) &= \delta'(q_0, \epsilon x \epsilon) \\
&= \delta'(\delta(q_0, \epsilon), x\epsilon) \\
&= \delta'(\{q_1, q_3\}, x\epsilon) \\
&= \delta'(\{q_1\}, x\epsilon) \cup \delta'(\{q_3\}, x\epsilon) \\
&= \delta(\delta'(q_1, x), \epsilon) \cup \delta(\delta'(q_3, x), \epsilon) \\
&= \delta(\delta_1'(q_1, x), \epsilon) \cup \delta(\delta_2'(q_3, x), \epsilon)
\end{aligned}
$$



证明：由于$q_5 \in \delta'(q_0, x)$，则必有$\delta'_1(\{q_1\}, x) = \{q_2\}$或者$\delta'_2(\{q_3\}, x) = \{q_4\}$。当$\delta'_1(\{q_1\}, x) = \{q_2\}$时，$x \in L(M_E)$。当$\delta'_2(\{q_3\}, x) = \{q_4\}$时，$x \in L(M_F)$。由此可以得出结论：

$$L(M) \subseteq L(M_E) \cup L(M_F)$$

综合两个方向的结论，可以得到：

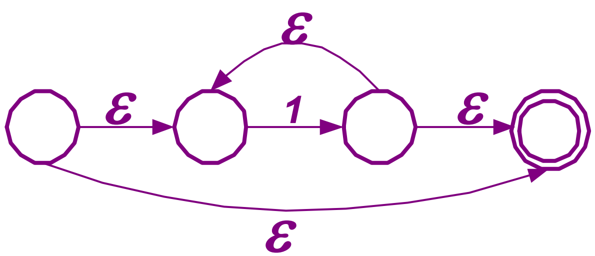
$$L(M) = L(M_E) \cup L(M_F)$$

在这个例子里面，我们主要利用了Thompson构造法的转移函数的定义和转移函数本身的性质。

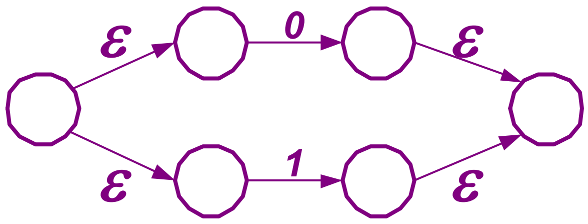
✧ 举例 (从正规表达式构造等价的ε - NFA)

设正规表达式 *1*0(0+1)**, 构造等价的ε - NFA.

*1**

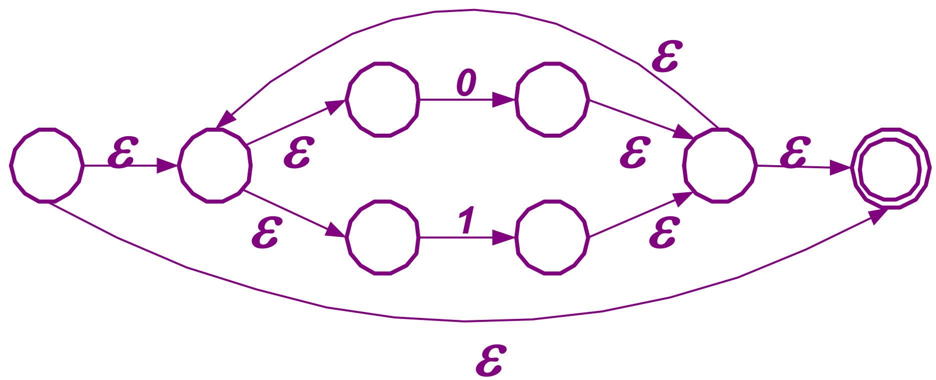


0+1

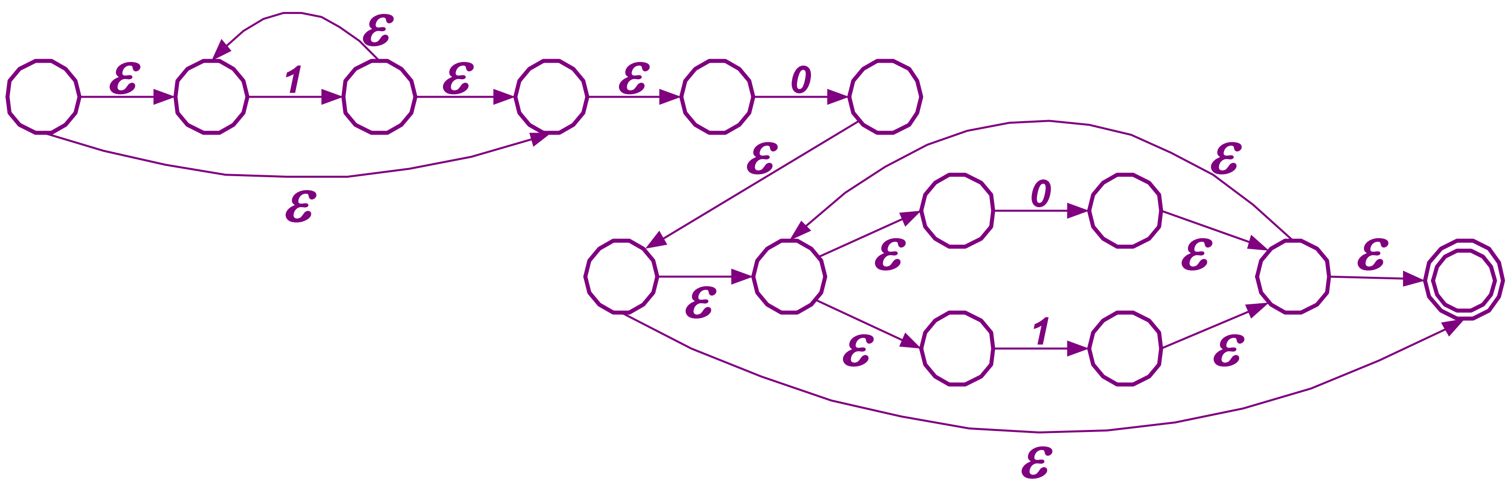


✧ 举例 (从正规表达式构造等价的ε - NFA)

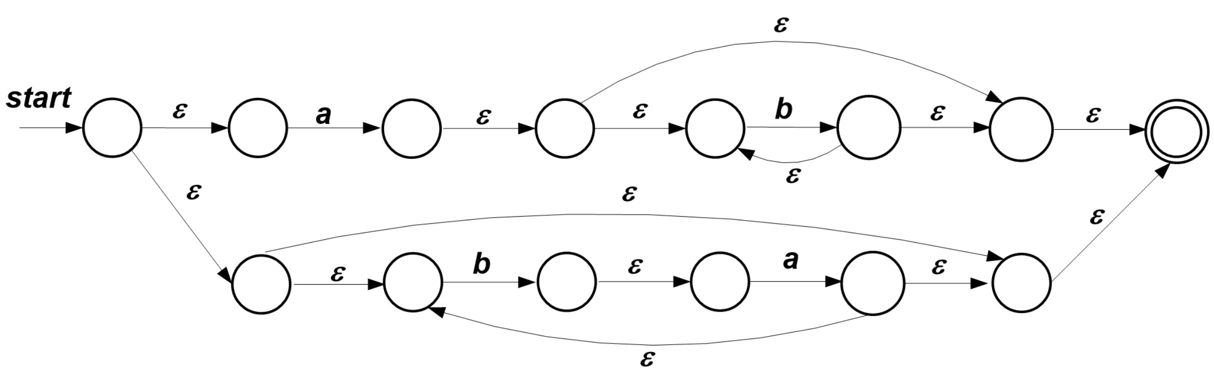
*(0+1)**



*1*0(0+1)**



例2：若严格依课程所介绍的算法（Thompson 构造法）将某个正规表达式转换为等价的 ε–NFA，下图所示为该ε–NFA的转移图表示。试给出这个正规表达式。



解：相应的正规表达式为

$$\mathbf{ab}^* + (\mathbf{ba})^*$$

✧ 从 *DFA* 构造等价的正规表达式

– 定理: *L* 是某个 *DFA D* 的语言, 则存在一个正规表达式 *R* ,满足 *L(R) = L(D) = L.*

证明: 构造性证明. 以下是两种构造方法

(1) 路径迭代法（*Kleene* 构造法）;

(2) 状态消去法

✧ 路径迭代法（从 *DFA* 构造等价的正规表达式）

– 步骤:

(1) 将 *DFA D* 的状态集用{1, 2, … , n}表达，且初态为1

(2) 对所有$1 \le i, j \le n$, $0 \le k \le n$ ，迭代计算$R_{ij}^{(k)}$；这里, $R_{ij}^{(k)}$为表示如下语言的正规表达式：
$w \in L(R_{ij}^{(k)})$ *iff* 从 i 到 j 有一条标记为 w 的路径, 且这条路径上除 i 和 j 之外的所有状态的编号均不大于 k

(3) 通过*(2)*的迭代过程，最终可计算出 $R_{ij}^{(n)}$（$i, j = 1, 2, \ldots, n$）

(4) 将所有 $R_{1j}^{(n)}$（j 为任一终态）相 “+”

✧ 计算 $R_{ij}^{(k)}$ 的迭代过程

– 基础: $k = 0$

Case 1 $i \neq j$

若不存在从 i 到 j 的弧，则$R_{ij}^{(0)} = \phi$;
若仅存在一条从 i 到 j 的弧，且标记为a，则$R_{ij}^{(0)} = a$;
若存在多条从 i 到 j 的弧，且标记为$a_1, a_2, \ldots, a_m$,
则 $R_{ij}^{(0)} = a_1 + a_2 + \ldots + a_m$;

Case 2 $i = j$

若不存在从 i 到自身的圈，则$R_{ij}^{(0)} = \varepsilon$;
若存在一个从 i 到自身的圈且标记为a，则$R_{ij}^{(0)} = \varepsilon + a$;
若存在多个从 i到自身的圈，且标记为$a_1, a_2, \ldots, a_m$,
则 $R_{ij}^{(0)} = \varepsilon + a_1 + a_2 + \ldots + a_m$;

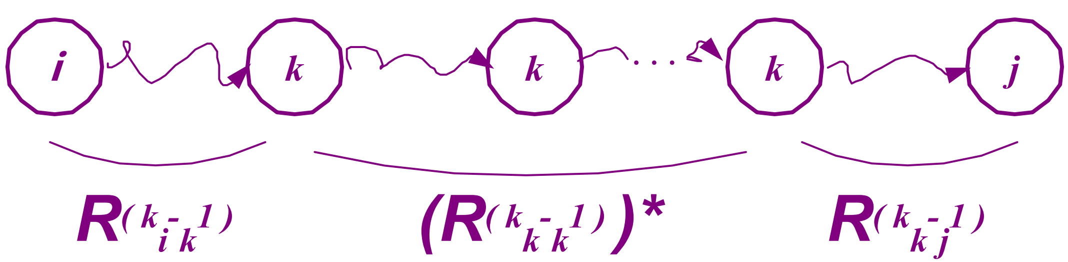
✧ 计算 $R^{(k)}_{ij}$ 的迭代过程

– 归纳: 假设 $R^{(k-1)}_{ij}$（$i, j = 1, 2, \ldots, n$）已经求出. 则迭代公式为 $R^{(k)}_{ij} = R^{(k-1)}_{ij} + R^{(k-1)}_{ik}\,(R^{(k-1)}_{kk})^*\,R^{(k-1)}_{kj}$

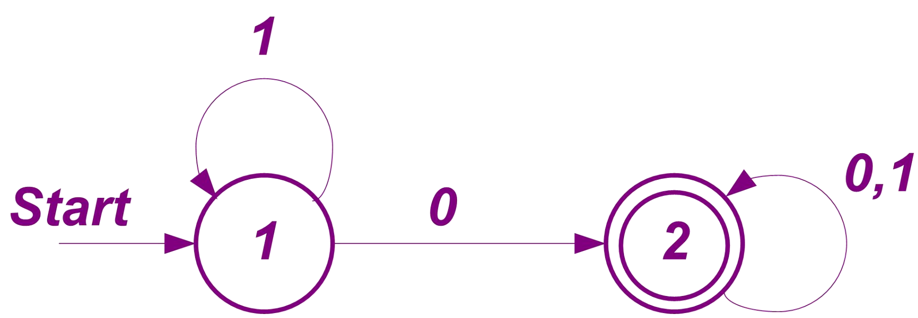
分析: 考虑从 i 到 j 的路径（除 i 和 j 之外的所有状态的编号不大于 k）

Case 1 路径不经过 k. 此时，标记该路径的字符串属于 $L(R^{(k-1)}_{ij})$;

Case 2 路径经过 k至少一次. 此时，标记该路径的字符串属于 $L(R^{(k-1)}_{ik}\,(R^{(k-1)}_{kk})^*\,R^{(k-1)}_{kj})$. 如下图所示:

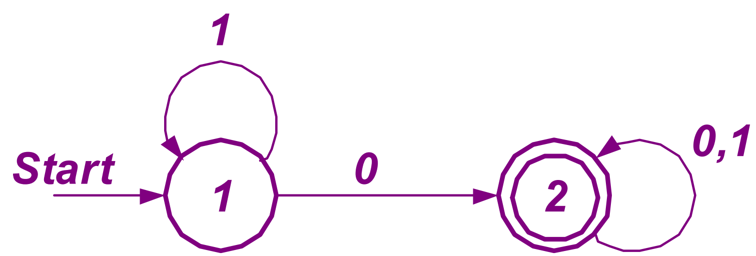


✧ 路径迭代法举例



$R_{11}^{(0)}$	$\varepsilon + 1$
$R_{12}^{(0)}$	0
$R_{21}^{(0)}$	ϕ
$R_{22}^{(0)}$	$\varepsilon + 0 + 1$

✧ 路径迭代法举例

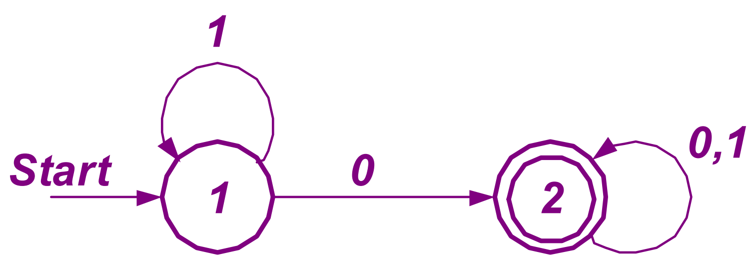


$R_{11}^{(0)}$	$\varepsilon + 1$
$R_{12}^{(0)}$	0
$R_{21}^{(0)}$	ϕ
$R_{22}^{(0)}$	$\varepsilon + 0 + 1$

	直接替换	化简
$R_{11}^{(1)}$	$\varepsilon+1+(\varepsilon+1)(\varepsilon+1)^*(\varepsilon+1)$	1^*
$R_{12}^{(1)}$	$0+(\varepsilon+1)(\varepsilon+1)^*0$	1^*0
$R_{21}^{(1)}$	$\phi+\phi(\varepsilon+1)^*(\varepsilon+1)$	ϕ
$R_{22}^{(1)}$	$\varepsilon+0+1+\phi(\varepsilon+1)^*0$	$\varepsilon+0+1$

$$R_{ij}^{(1)} = R_{ij}^{(0)} + R_{i1}^{(0)}(R_{11}^{(0)})^* R_{1j}^{(0)}$$

✧ 路径迭代法举例

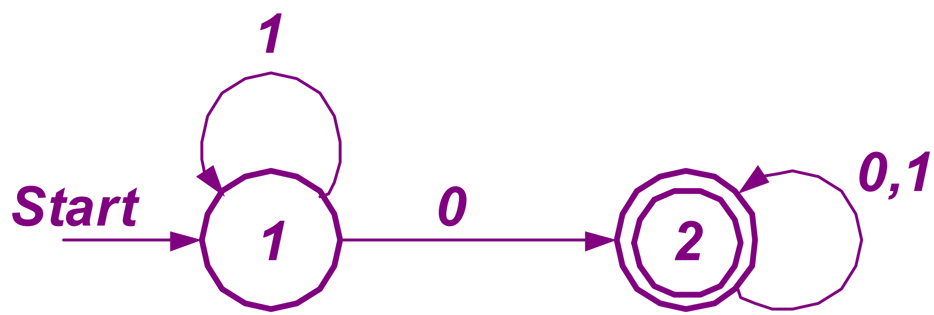


$R_{11}^{(1)}$	1^*
$R_{12}^{(1)}$	1^*0
$R_{21}^{(1)}$	ϕ
$R_{22}^{(1)}$	$\varepsilon + 0 + 1$

	直接替换	化简
$R_{11}^{(2)}$	$1^*+1^*0(\varepsilon+0+1)^*\phi$	1^*
$R_{12}^{(2)}$	$1^*0+1^*0(\varepsilon+0+1)^*(\varepsilon+0+1)$	$1^*0(0+1)^*$
$R_{21}^{(2)}$	$\phi+(\varepsilon+0+1)(\varepsilon+0+1)^*\phi$	ϕ
$R_{22}^{(2)}$	$\varepsilon+0+1+(\varepsilon+0+1)(\varepsilon+0+1)^*(\varepsilon+0+1)$	$(0+1)^*$

$$R_{ij}^{(2)} = R_{ij}^{(1)} + R_{i2}^{(1)}\left(R_{22}^{(1)}\right)^* R_{2j}^{(1)}$$

✧ 路径迭代法举例



结果: 初态为1, 终态只有一个 2，所以， 一个与上图的 *DFA* 等价的正规表达式为

$$R_{12}^{(2)} = 1^*0(0+1)^*$$

✧ 状态消去法（从 *DFA* 构造等价的正规表达式）

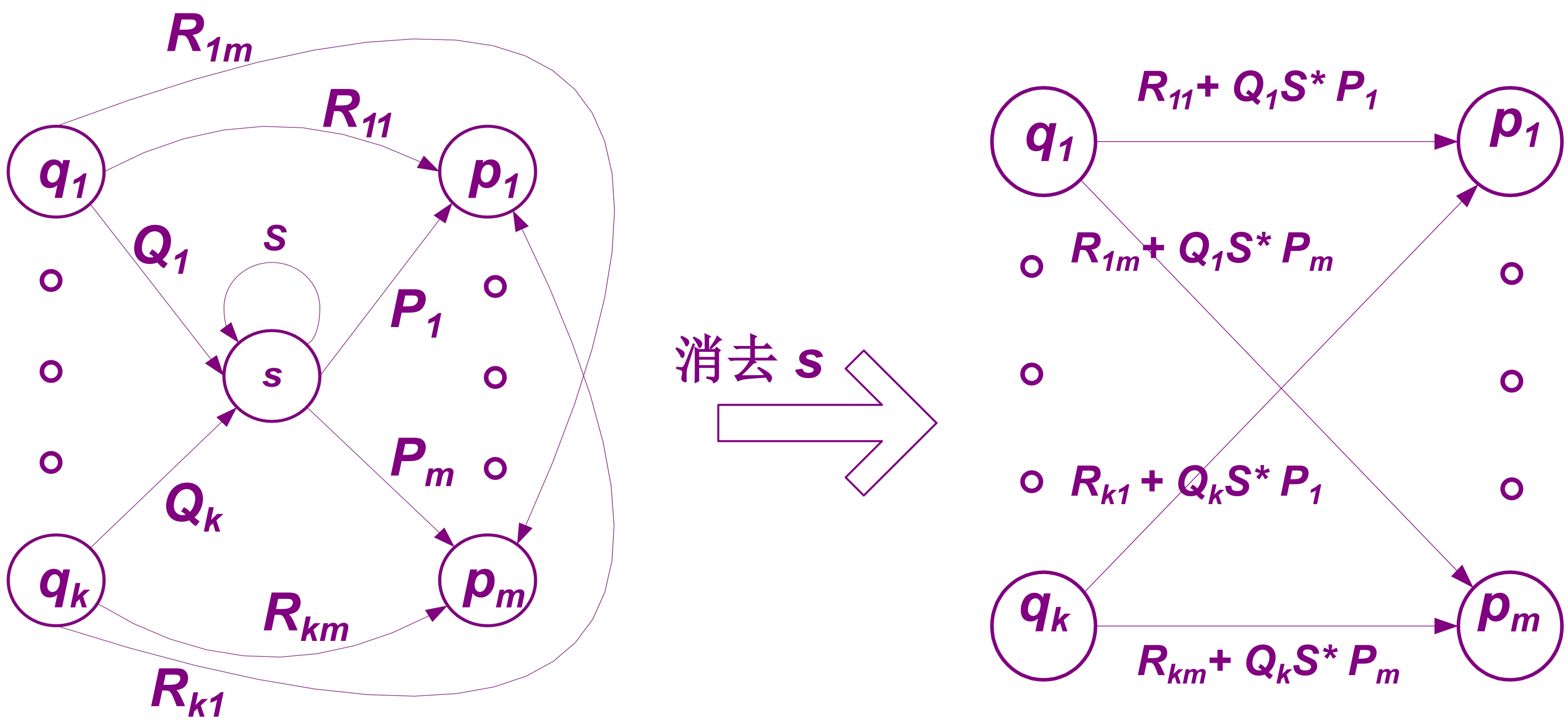
– 思路:

(1) 扩展自动机的概念，允许正规表达式作为转移弧的标记. 这样，就有可能在消去某一中间状态时，保证自动机能够接受的字符串集合保持不变.

(2) 在消去某一中间状态时，与其相关的转移弧也将同时消去，所造成的影响将通过修改从每一个前趋状态到每一个后继状态的转移弧标记来弥补.

以下分别介绍中间状态的消去与正规表达式构造过程.

✧ 中间状态的消去


R1m
R11
q1
p1
Q1
s
s
P1
Pm
Qk
qk
pm
Rkm
Rk1
消去 s
q1
p1
R11+ Q1S* P1
R1m+ Q1S* Pm
Rk1 + QkS* P1
qk
pm
Rkm+ QkS* Pm

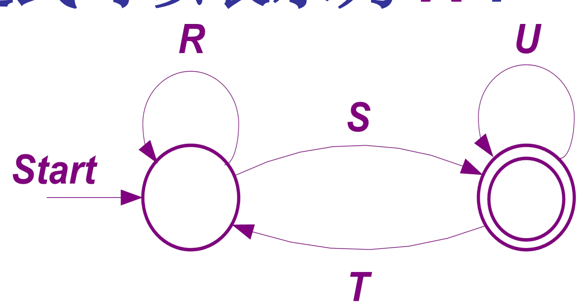
✧ 状态消去法（从 *DFA* 构造等价的正规表达式）

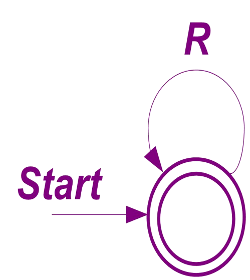
– 步骤：（假设自动机已转化为扩展的形式）

(1) 对每一终态 q，依次消去除 q 和初态 q_0 之外的其它状态;

(2) 若 $q \neq q_0$，最终可得到一般形式如下左图两状态自动机，该自动机对应的正规表达式可表示为 $(R+SU^*T)^*SU^*$.

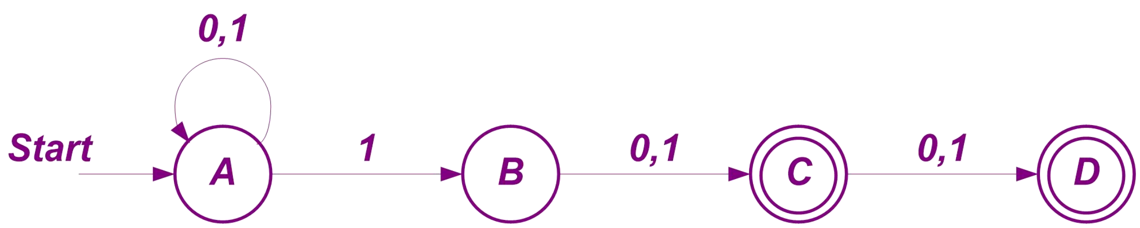
(3) 若 $q = q_0$，最终可得到如下右图的自动机，它对应的正规表达式可以表示为 R^*.



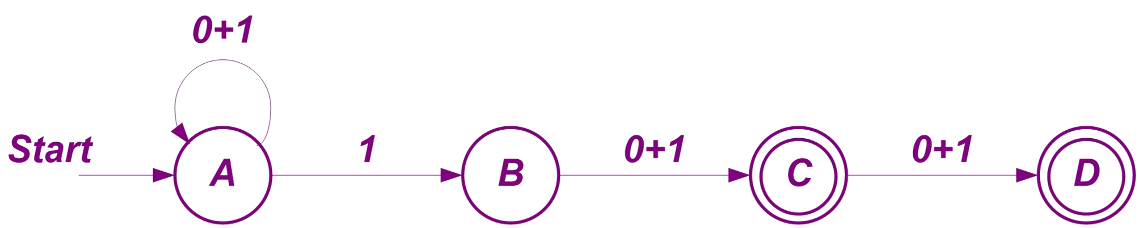


(4) 最终的正规表达式为每一终态对应的正规表达式之和（并）.

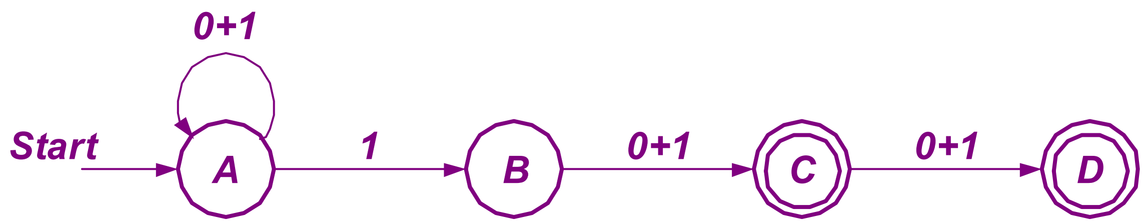
✧ 状态消去法举例（推广至非*DFA*的情形）



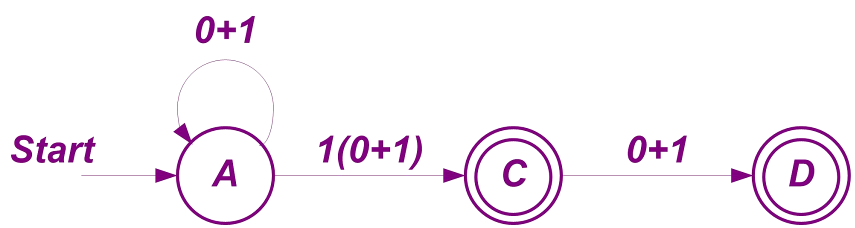
⇨

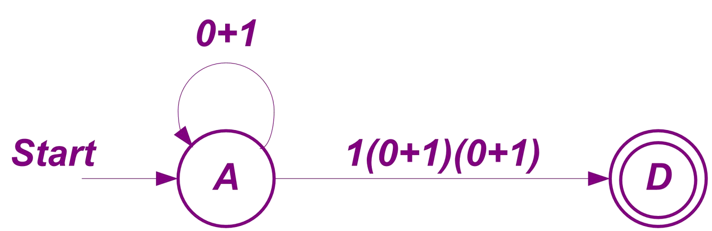


✧ 状态消去法举例

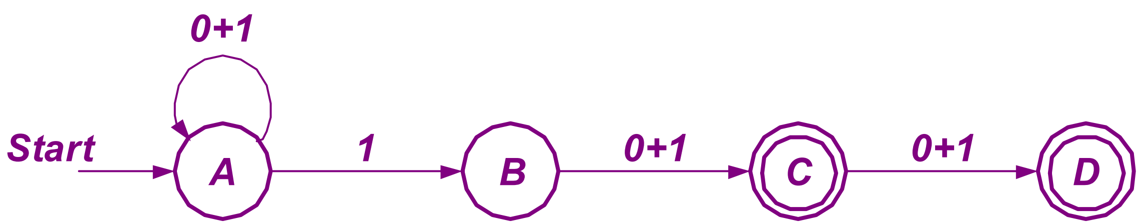


对于终态*D*

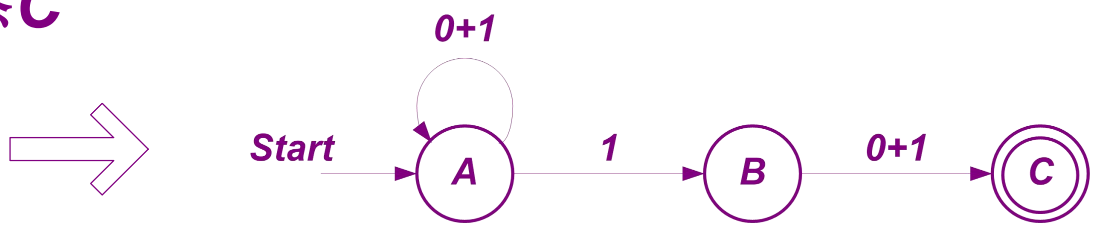


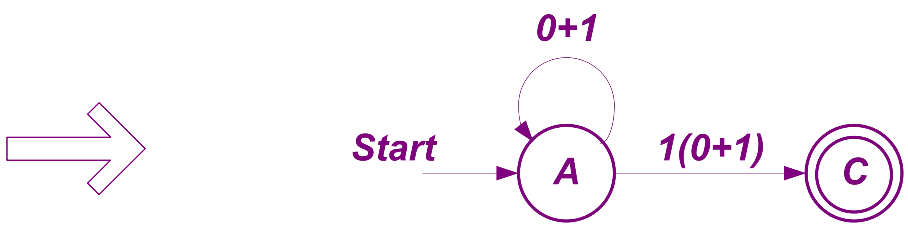


✧ 状态消去法举例

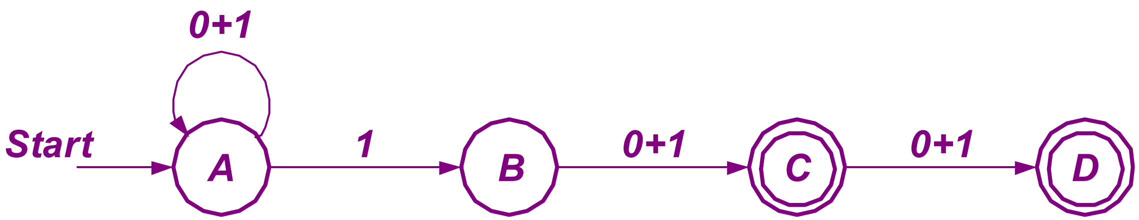


对于终态C

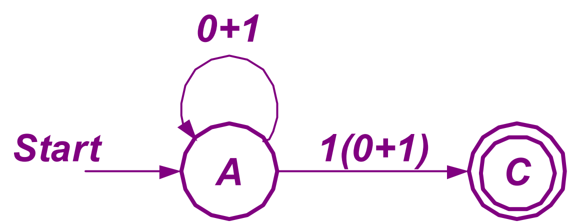




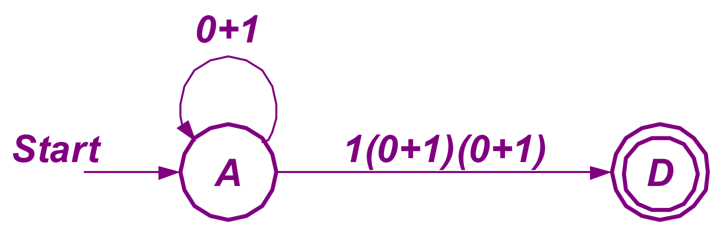
✧ 状态消去法举例



对于终态*C*



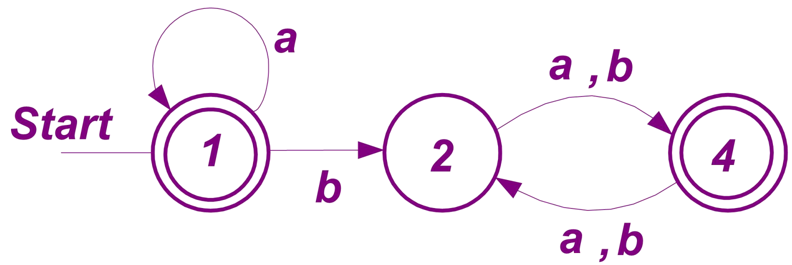
对于终态*D*



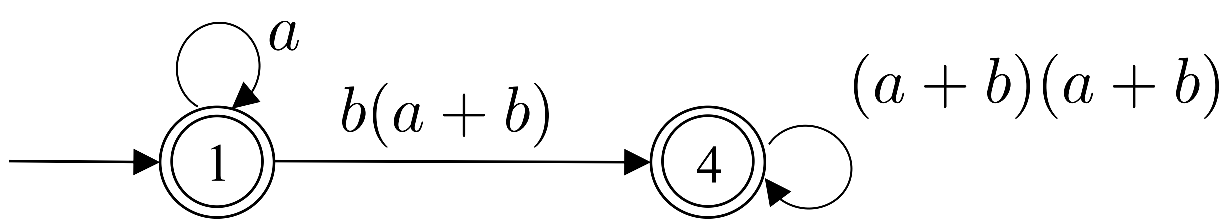
等价的正规表达式

$$(0+1)^*1(0+1)+(0+1)^*1(0+1)(0+1)$$

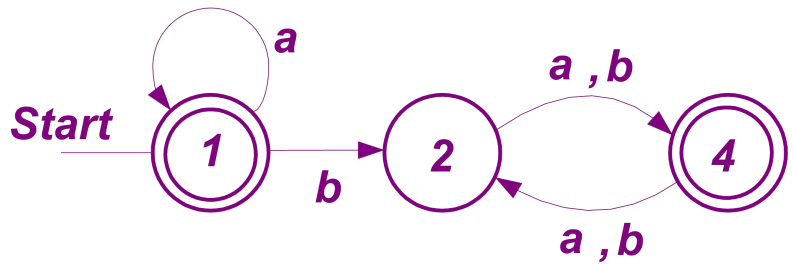
例3：用状态消去法将以下DFA转为正规表达式。



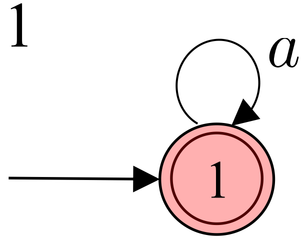
解：首先要消除中间节点2。根据状态消去法，需要找出节点2的所有前继和后继节点。观察DFA，可以发现节点2有两个前驱，一个是节点1，另一个是节点4（这种情况特别容易被漏掉）。在这两种情况下，后继都是节点4。因此，



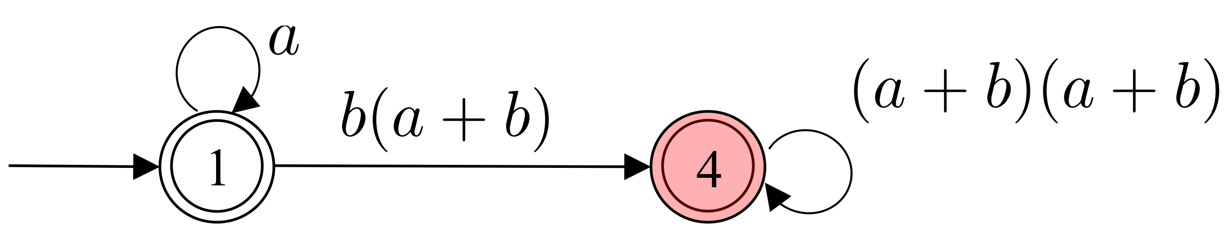
例3：用状态消去法将以下DFA转为正规表达式。



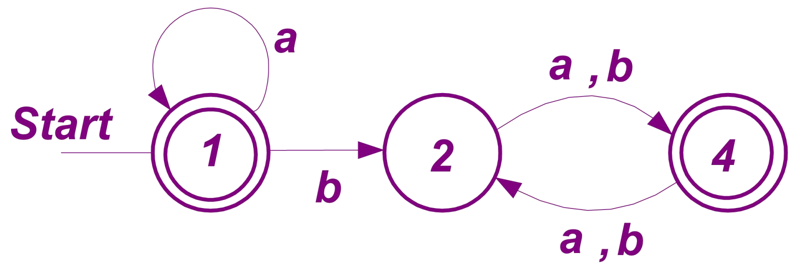
解：考虑终态1



考虑终态4



例3：用状态消去法将以下DFA转为正规表达式。



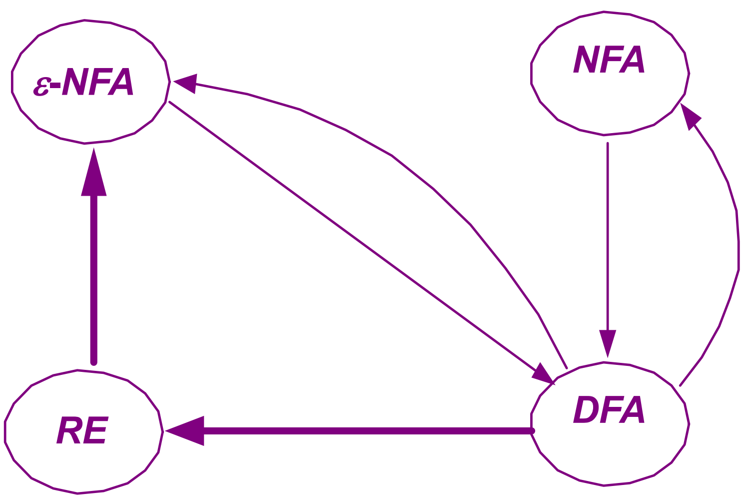
解：因此最终的正规表达式为

$$\mathbf{a}^* + \mathbf{a}^*\mathbf{b}(\mathbf{a}+\mathbf{b})((\mathbf{a}+\mathbf{b})(\mathbf{a}+\mathbf{b}))^*$$

几个转换算法的复杂度（选讲）

✧ 几个转换算法

- 从 *DFA* 构造 *NFA*
- 从 *NFA* 构造 *DFA*
- 从 *DFA* 构造 ε - *NFA*
- 从 ε - *NFA* 构造 *DFA*
- 从 *DFA* 构造正规表达式
- 从正规表达式构造 ε - *NFA*



几个转换算法的复杂度（选讲）

✧ 从 DFA 构造 NFA

– 回顾：设 DFA $D = (Q, \Sigma, \delta_D, q_0, F)$, 构造 NFA $N = (Q, \Sigma, \delta_N, q_0, F_N)$，其中 δ_N 定义为

  • 对$q \in Q$ 和$a \in \Sigma$,
  若 $\delta_D(q,a) = p$, 则 $\delta_N(q,a) = \{p\}$.

– 设 $|Q|=n$, 该构造过程复杂度为$O(n)$, 即线性时间.

几个转换算法的复杂度（选讲）

✧ 从 NFA 构造 DFA

– 回顾：设 NFA $N = (Q, \Sigma, \delta_N, q_0, F)$，构造 $D = (Q_D, \Sigma, \delta_D, \{q_0\}, F_D)$，其中

- $Q_D = \{ S \mid S \subseteq Q \}$
- 对 $S \in Q_D$ 和 $a \in \Sigma$，$\delta_D(S, a) = \bigcup_{q \in S} \delta_N(q,a)$.
- $F_D = \{S \mid S \subseteq Q \wedge S \cap F \neq \phi \}$

– 设 $|Q|=n$, 该构造过程复杂度为$O(n^2 2^n)$. 但实际运行时间的上界可以是$O(n^2 s)$，其中 s 为 DFA 实际状态数。

✧ 从 DFA 构造 ε - NFA

– 回顾：设 DFA $D = (Q, \Sigma, \delta_D, q_0, F)$，构造 $E = (Q, \Sigma, \delta_E, q_0, F_E)$，其中 δ_E 定义为

  • 对任何 $q \in Q$, $\delta_E(q, \varepsilon) = \phi$
  • 对任何 $q \in Q$ 和 $a \in \Sigma$,
  若 $\delta_D(q,a) = p$, 则 $\delta_N(q,a) = \{p\}$

– 设 $|Q|=n$, 该构造过程复杂度为 $O(n)$.

几个转换算法的复杂度（选讲）

✧ 从 ε - NFA 构造 DFA

– 回顾：设 ε - NFA $E = (Q_E, \Sigma, \delta_E, q_0, F_E)$，构造 $D = (Q_D, \Sigma, \delta_D, q_D, F_D)$，其中

- $Q_D = \{ S \mid S \subseteq Q_E \wedge S = ECLOSE(S) \}$
- $q_D = ECLOSE(q_0)$
- $F_D = \{ S \mid S \in Q_D \wedge S \cap F_E \neq \phi \}$
- 对 $S \in Q_D$ 和 $a \in \Sigma$，令 $S = \{ p_1, p_2, \ldots, p_k \}$，并设 $\bigcup_{i=1}^{k} \delta_E(p_i, a) = \{ r_1, r_2, \ldots, r_m \}$，则

$$\delta_D(S, a) = \bigcup_{j=1}^{m} ECLOSE(r_j).$$

– 设 $|Q_E| = n$，该构造过程复杂度为 $O(n^3 2^n)$。但实际运行时间的上界可以是 $O(n^3 s)$，其中 s 为 DFA 实际状态数。

几个转换算法的复杂度（选讲）

✧ 从 *DFA* 构造正规表达式

- 回顾： *(路径迭代法)*

  (1) 将 *DFA D* 的状态集用*{1, 2, … , n}*表达，且初态为*1*;

  (2) 对所有*i, j, k = 1, 2, … , n*， 迭代计算$R_{ij}^{(k)}$;

  (3) 将所有 $R_{1j}^{(n)}$（*j* 为任一终态）相 “+”

- 该构造过程复杂度为*O(n³4ⁿ)*（考虑表达式的大小）
- 采用状态消去法具有同样的复杂度

几个转换算法的复杂度（选讲）

✧ 从正规表达式构造ε - NFA

– 回顾：

归纳于正规表达式的结构，或通过构造一棵表达式树，然后根据归纳构造规则得到ε - ***NFA***；每一结点上的工作只是增加不超过两个新的状态，以及不超过四条新的弧.

– 该构造过程复杂度为***O(n)***, 这里 ***n*** 为正规表达式的大小.

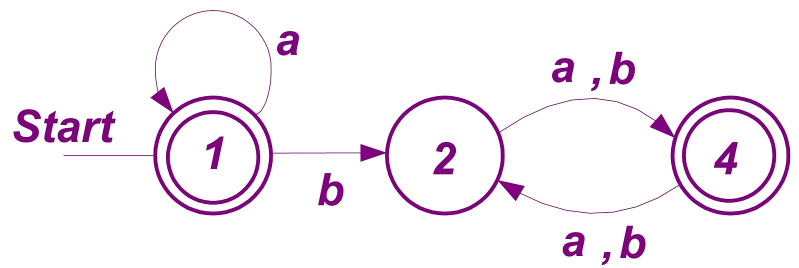
课后练习

✧ 必做题:

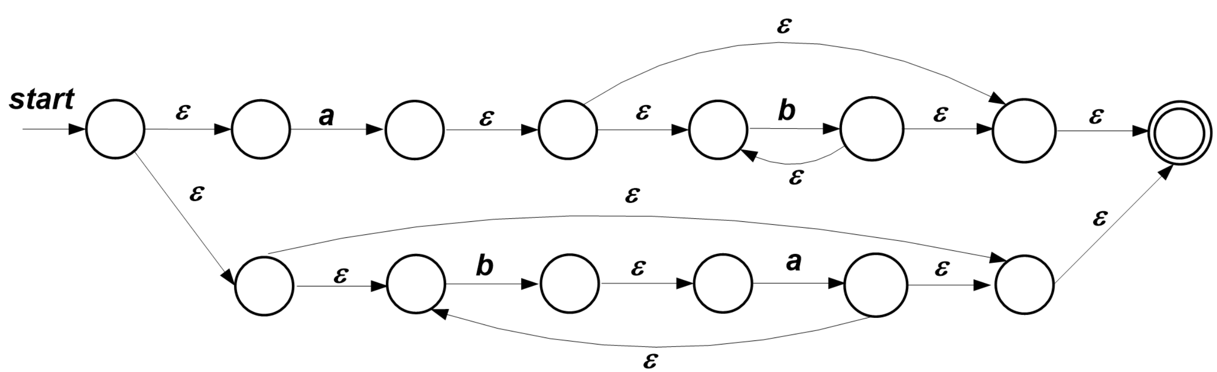
- *Ex.3.2.1 (c),(d)*
- *Ex. 3.2.3*
- *Ex.3.2.4 (b),(c)*
- *! Ex.3.2.6*

课后练习

- ✧ 自测题:
  - 下图表示一个 DFA . 使用状态消去技术，求出与此*DFA*等价的一个正规表达式. (分主要步骤或直接写出结果均可)）



  - 若严格依课程所介绍的算法（Thompson 构造法）将某个正规表达式转换为等价的 ε–NFA，下图所示为该ε–NFA的转移图表示。试给出这个正规表达式。



That’s all for today.

Thank You