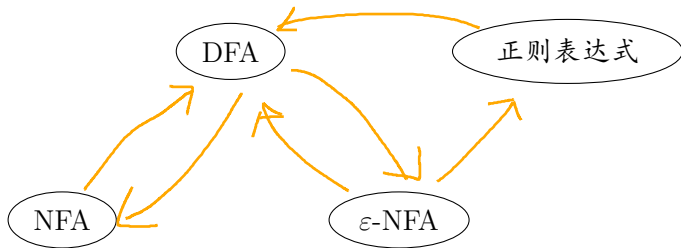


# 正则表达式

- 正则表达式
- 自动机和正则表达式
  - 由自动机到正则表达式
  - 由正则表达式到自动机
- 正则表达式的代数定律

DFA, NFA,  $\epsilon$ -NFA 和正则表达式在表示语言的能力上等价.



## 由 DFA 到正则表达式, 递归表达式法

### 定理 3

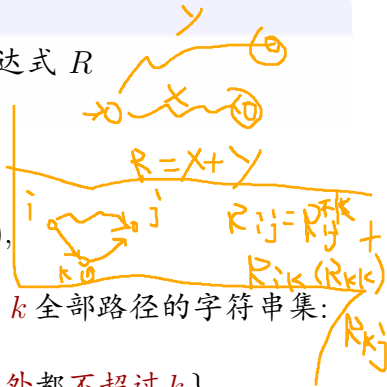
若  $L = L(A)$  是某 DFA  $A$  的语言, 那么存在正则表达式  $R$  满足  $L = L(R)$ .

证明: 对 DFA  $A$  的状态编号, 令 1 为开始状态, 即

$$A = (\{1, 2, \dots, n\}, \Sigma, \delta, 1, F),$$

设正则表达式  $R_{ij}^{(k)}$  表示从  $i$  到  $j$  但中间节点不超过  $k$  全部路径的字符串集:

$$R_{ij}^{(k)} = \{x \mid \hat{\delta}(i, x) = j, x \text{ 经过的状态除两端外都不超过 } k\}.$$



那么与  $A = (\{1, 2, \dots, n\}, \Sigma, \delta, 1, F)$  等价的正则表达式为

$$\bigcup_{j \in F} R_{1j}^{(n)}$$

且递归式为

$$R_{ij}^{(k)} = R_{ij}^{(k-1)} + R_{ik}^{(k-1)} (R_{kk}^{(k-1)})^* R_{kj}^{(k-1)}$$

$$R_{ij}^{(0)} = \begin{cases} \{a \mid \delta(q_i, a) = q_j\} & i \neq j \\ \{a \mid \delta(q_i, a) = q_j\} \cup \{\varepsilon\} & i = j \end{cases}$$

下面对  $k$  归纳, 证明可用以上递归式求得  $R_{ij}^{(k)}$ .

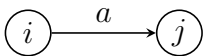
归纳基础: 当  $i \neq j$ ,  $k = 0$  时, 即  $i$  到  $j$  没经过任何中间节点

- 没有  $i$  到  $j$  的状态转移



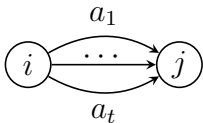
$$R_{ij}^{(0)} = \emptyset$$

- 有一个  $i$  到  $j$  的状态转移



$$R_{ij}^{(0)} = \mathbf{a}$$

- 有多个  $i$  到  $j$  的状态转移



$$R_{ij}^{(0)} = \mathbf{a}_1 + \mathbf{a}_2 + \cdots + \mathbf{a}_t$$

$$R_{ij}^{(k)} = R_{ij}^{(k-1)} + R_{ik}^{(k-1)}(R_{kk}^{(k-1)})^* R_{kj}^{(k-1)}$$

$$R_{ij}^{(0)} = \begin{cases} \{a \mid \delta(q_i, a) = q_j\} & i \neq j \\ \{a \mid \delta(q_i, a) = q_j\} \cup \{\varepsilon\} & i = j \end{cases}$$

归纳基础 (续): 当  $i = j$ ,  $k = 0$  时, 即从  $i$  到自身没经过任何中间节点

- 状态  $i$  没有到自己的转移



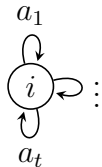
$$R_{ii}^{(0)} = \epsilon$$

- 状态  $i$  有一个到自己的转移



$$R_{ii}^{(0)} = \mathbf{a} + \epsilon$$

- 状态  $i$  有多个到自己的转移



$$R_{ii}^{(0)} = \mathbf{a}_1 + \mathbf{a}_2 + \dots + \mathbf{a}_t + \epsilon$$

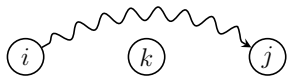
$$R_{ij}^{(k)} = R_{ij}^{(k-1)} + R_{ik}^{(k-1)}(R_{kk}^{(k-1)})^* R_{kj}^{(k-1)}$$

$$R_{ij}^{(0)} = \begin{cases} \{a \mid \delta(q_i, a) = q_j\} & i \neq j \\ \{\mathbf{a} \mid \delta(q_i, \mathbf{a}) = q_j\} \cup \{\epsilon\} & i = j \end{cases}$$

归纳假设: 已知  $R_{ij}^{(k-1)}$  是从  $i$  到  $j$  但中间节点不超过  $k-1$  的全部路径,  
同理已知  $R_{ik}^{(k-1)}$ ,  $R_{kk}^{(k-1)}$  和  $R_{kj}^{(k-1)}$ .

归纳递推: 那么  $R_{ij}^{(k)}$  中全部路径, 可用节点  $k$  分为两部分

- 从  $i$  到  $j$  不经过  $k$  的



$$R_{ij}^{(k)} = R_{ij}^{(k-1)}$$

- 从  $i$  到  $j$  经过  $k$  的



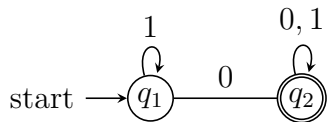
$$R_{ij}^{(k)} = R_{ik}^{(k-1)} (R_{kk}^{(k-1)})^* R_{kj}^{(k-1)}$$

$$R_{ij}^{(k)} = R_{ij}^{(k-1)} + R_{ik}^{(k-1)} (R_{kk}^{(k-1)})^* R_{kj}^{(k-1)}$$

$$R_{ij}^{(0)} = \begin{cases} \{a \mid \delta(q_i, a) = q_j\} & i \neq j \\ \{a \mid \delta(q_i, a) = q_j\} \cup \{\varepsilon\} & i = j \end{cases}$$

$$\text{因此 } R_{ij}^{(k)} = R_{ij}^{(k-1)} + R_{ik}^{(k-1)} (R_{kk}^{(k-1)})^* R_{kj}^{(k-1)}.$$

例8. 将如图 DFA 转换为正则表达式.



- 计算  $R_{ij}^{(0)}$

$R_{ij}^{(k)}$	$k = 0$
$R_{11}^{(0)}$	$\epsilon + 1$
$R_{12}^{(0)}$	$0$
$R_{21}^{(0)}$	$\emptyset$
$R_{22}^{(0)}$	$\epsilon + 0 + 1$



续例 8.

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

$R_{ij}^{(k)}$	$k = 0$	$R_{ij}^{(k)}$	$k = 1$
$R_{11}^{(0)}$	$\varepsilon + \mathbf{1}$	$R_{11}^{(1)}$	$(\varepsilon + \mathbf{1}) + (\varepsilon + \mathbf{1})(\varepsilon + \mathbf{1})^*(\varepsilon + \mathbf{1})$
$R_{12}^{(0)}$	$\mathbf{0}$	$R_{12}^{(1)}$	$\mathbf{0} + (\varepsilon + \mathbf{1})(\varepsilon + \mathbf{1})^*\mathbf{0}$
$R_{21}^{(0)}$	$\emptyset$	$R_{21}^{(1)}$	$\emptyset + \emptyset(\varepsilon + \mathbf{1})^*(\varepsilon + \mathbf{1})$
$R_{22}^{(0)}$	$\varepsilon + \mathbf{0} + \mathbf{1}$	$R_{22}^{(1)}$	$\varepsilon + \mathbf{0} + \mathbf{1} + \emptyset(\varepsilon + \mathbf{1})^*\mathbf{0}$

续例 8.

- 几个基本的化简规则

如果  $r$  和  $s$  是两个正则表达式

$$(\epsilon + r)^* = r^* = \{\epsilon, r, r^2, r^3, \dots\}$$
$$(\epsilon + r)r^* = r^* = \{\epsilon, \dots\}$$

$$r + rs^* = rs^*$$

$$\emptyset r = r\emptyset = \emptyset$$

(零元)

$$\emptyset + r = r + \emptyset = r$$

(单位元)

续例 8.

- 化简  $R_{ij}^{(1)}$

$R_{ij}^{(k)}$	$k = 1$	化简
$R_{11}^{(1)}$	$(\varepsilon + 1) + (\varepsilon + 1)(\varepsilon + 1)^*(\varepsilon + 1)$	$1^*$
$R_{12}^{(1)}$	$0 + (\varepsilon + 1)(\varepsilon + 1)^*0$	$1^*0$
$R_{21}^{(1)}$	$\emptyset + \emptyset(\varepsilon + 1)^*(\varepsilon + 1)$	$\emptyset$
$R_{22}^{(1)}$	$\varepsilon + 0 + 1 + \emptyset(\varepsilon + 1)^*0$	$\varepsilon + 0 + 1$

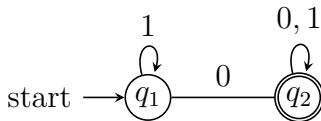
$\{\varepsilon, 0, 1\}$

续例 8.

- 计算  $R_{ij}^{(2)} = R_{ij}^{(1)} + R_{i2}^{(1)}(R_{22}^{(1)})^*R_{2j}^{(1)}$

$R_{ij}^{(k)}$	$k = 1$	$R_{ij}^{(k)}$	$k = 2$
$R_{11}^{(1)}$	$\mathbf{1}^*$	$R_{11}^{(2)}$	$\mathbf{1}^* + \mathbf{1}^*\mathbf{0}(\boldsymbol{\varepsilon} + \mathbf{0} + \mathbf{1})^*\emptyset$
$R_{12}^{(1)}$	$\mathbf{1}^*\mathbf{0}$	$R_{12}^{(2)}$	$\mathbf{1}^*\mathbf{0} + \mathbf{1}^*\mathbf{0}(\boldsymbol{\varepsilon} + \mathbf{0} + \mathbf{1})^*(\boldsymbol{\varepsilon} + \mathbf{0} + \mathbf{1})$
$R_{21}^{(1)}$	$\emptyset$	$R_{21}^{(2)}$	$\emptyset + (\boldsymbol{\varepsilon} + \mathbf{0} + \mathbf{1})(\boldsymbol{\varepsilon} + \mathbf{0} + \mathbf{1})^*\emptyset$
$R_{22}^{(1)}$	$\boldsymbol{\varepsilon} + \mathbf{0} + \mathbf{1}$	$R_{22}^{(2)}$	$\boldsymbol{\varepsilon} + \mathbf{0} + \mathbf{1} + (\boldsymbol{\varepsilon} + \mathbf{0} + \mathbf{1})(\boldsymbol{\varepsilon} + \mathbf{0} + \mathbf{1})^*(\boldsymbol{\varepsilon} + \mathbf{0} + \mathbf{1})$

续例 8.



- 化简  $R_{ij}^{(2)}$

$R_{ij}^{(k)}$	$k = 2$	化简
$R_{11}^{(2)}$	$1^* + 1^*0(\epsilon + 0 + 1)^*\emptyset$	$1^*$ ✓
$R_{12}^{(2)}$	$1^*0 + 1^*0(\epsilon + 0 + 1)^*(\epsilon + 0 + 1)$	$1^*0(0 + 1)^*$ ✓
$R_{21}^{(2)}$	$\emptyset + (\epsilon + 0 + 1)(\epsilon + 0 + 1)^*\emptyset$	$\emptyset$ ✓
$R_{22}^{(2)}$	$\epsilon + 0 + 1 + (\epsilon + 0 + 1)(\epsilon + 0 + 1)^*(\epsilon + 0 + 1)$	$(0 + 1)^*$ ✓

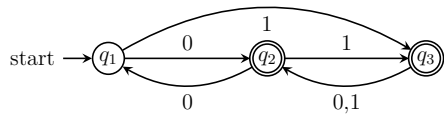
- 因只有  $q_2$  是接受状态, 所以该 DFA 正则表达式为

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

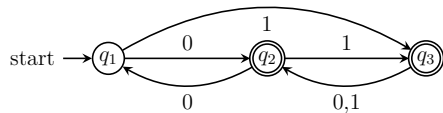
$\{0, 1\}^*$

$1 \rightarrow 2$

例 9. 将如图 DFA 转换为正则表达式.



续例 9. 将如图 DFA 转换为正则表达式.

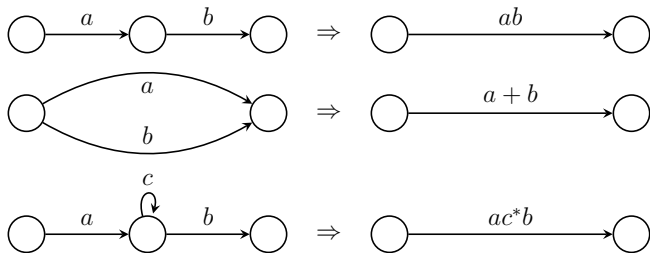


	$k = 0$	$k = 1$	$k = 2$	仅状态 2 和 3 是接受状态:
$R_{11}^{(k)}$	$\epsilon$	$\epsilon$	$(00)^*$	
$R_{12}^{(k)}$	$0$	$0$	$0(00)^*$	$R_{12}^{(3)} = R_{12}^{(2)} + R_{13}^{(2)}(R_{33}^{(2)})^*R_{32}^{(2)}$
$R_{13}^{(k)}$	$1$	$1$	$0^*1$	$= 0(00)^* + 0^*1(\epsilon + (0+1)0^*1)^*(0+1)(00)^*$
$R_{21}^{(k)}$	$0$	$0$	$0(00)^*$	$= 0(00)^* + 0^*1((0+1)0^*1)^*(0+1)(00)^*$
$R_{22}^{(k)}$	$\epsilon$	$\epsilon + 00$	$(00)^*$	
$R_{23}^{(k)}$	$1$	$1 + 01$	$0^*1$	$R_{13}^{(3)} = R_{13}^{(2)} + R_{13}^{(2)}(R_{33}^{(2)})^*R_{33}^{(2)}$
$R_{31}^{(k)}$	$\emptyset$	$\emptyset$	$(0+1)(00)^*0$	$= 0^*1 + 0^*1(\epsilon + (0+1)0^*1)^*(\epsilon + (0+1)0^*1)$
$R_{32}^{(k)}$	$0+1$	$0+1$	$(0+1)(00)^*$	$= 0^*1((0+1)0^*1)^*$
$R_{33}^{(k)}$	$\epsilon$	$\epsilon$	$\epsilon + (0+1)0^*1$	

$$R_{12}^{(3)} + R_{13}^{(3)} = 0^*1((0+1)0^*1)^*(\epsilon + (0+1)(00)^*) + 0(00)^*.$$

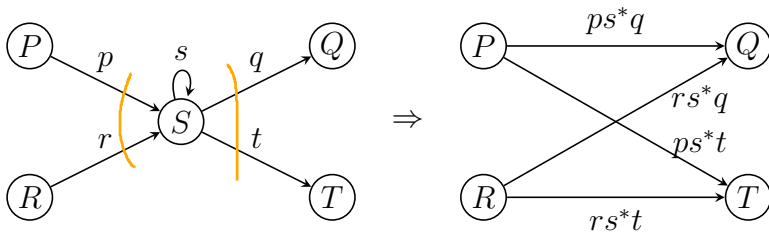
## 由 DFA 到正则表达式, 状态消除法

- 从 DFA 中逐个删除状态
- 用标记了正则表达式的新路径替换被删掉的路径
- 保持“自动机”等价.

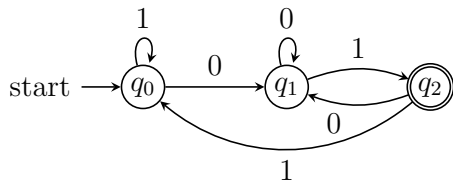




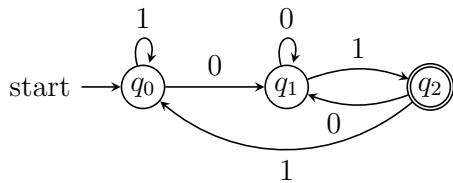
- 更一般的情况如图
- 若要删除状态  $S$ , 需添加相应路径



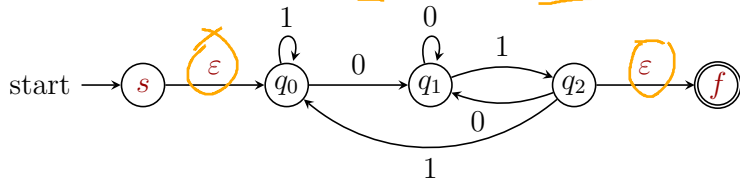
例 9. 利用状态消除法, 构造下图自动机的正则表达式.



续例9. 利用状态消除法, 构造下图自动机的正则表达式.

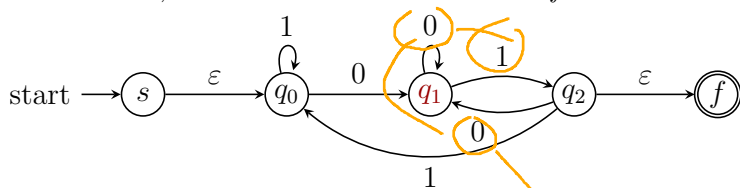


① 利用空转移, 添加新的开始  $s$  和结束状态  $f$ :

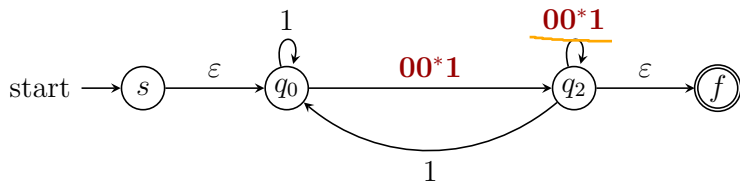


续例9. 利用状态消除法, 构造下图自动机的正则表达式.

- ① 利用空转移, 添加新的开始 $s$ 和结束状态 $f$ :

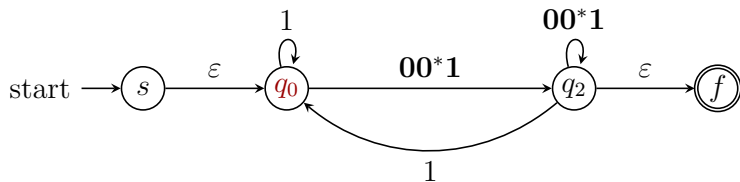


- ② 消除状态  $q_1$ , 添加路径  $q_0 \rightarrow q_2$  和  $q_2 \rightarrow q_0$ :

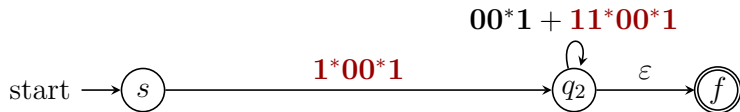


续例 9. 利用状态消除法, 构造下图自动机的正则表达式.

- ② 消除状态  $q_1$ , 添加路径  $q_0 \rightarrow q_2$  和  $q_2 \rightarrow q_2$ :

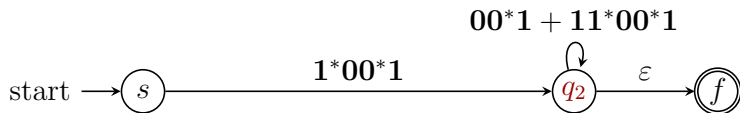


- ③ 消除状态  $q_0$ , 添加路径  $s \rightarrow q_2$  和  $q_2 \rightarrow q_2$ :



续例 9. 利用状态消除法, 构造下图自动机的正则表达式.

- ③ 消除状态  $q_0$ , 添加路径  $s \rightarrow q_2$  和  $q_2 \rightarrow q_2$ :

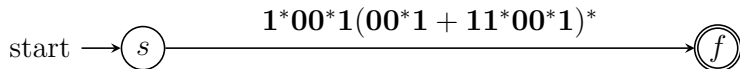


- ④ 消除状态  $q_2$ , 添加路径  $s \rightarrow f$ :



续例 9. 利用状态消除法, 构造下图自动机的正则表达式.

④ 消除状态  $q_2$ , 添加路径  $s \rightarrow f$ :



⑤ 因此该自动机的正则表达式为

$$\underline{1^*00^*1(00^*1 + 11^*00^*1)^*}.$$

# 由正则表达式到自动机

## 定理 4

正则表达式定义的语言, 都可被有穷自动机识别.

## 由正则表达式构造 $\epsilon$ -NFA

任何正则表达式  $e$ , 都存在与其等价的  $\epsilon$ -NFA  $A$ ,  
即  $L(A) = L(e)$ , 并且  $A$  满足:

- ① 仅有一个接收状态;
- ② 没有进入开始状态的边;
- ③ 没有离开接受状态的边.

$\rightarrow \bigcirc$

$\rightarrow \bigcirc^*$

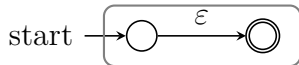


归纳基础:

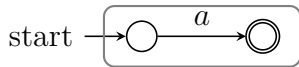
① 对于  $\emptyset$ , 有  $\varepsilon$ -NFA:



② 对于  $\varepsilon$ , 有  $\varepsilon$ -NFA:



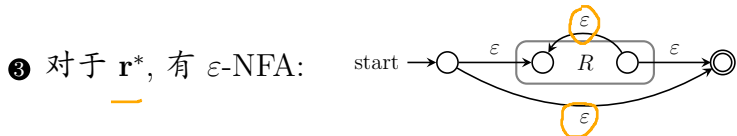
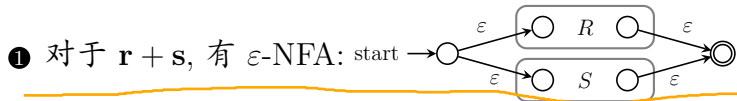
③  $\forall a \in \Sigma$ , 对于  $a$ , 有  $\varepsilon$ -NFA:



归纳递推: 若  $r$  和  $s$  为正则表达式, 则它们对应的  $\varepsilon$ -NFA 分别为  $R$  和  $S$



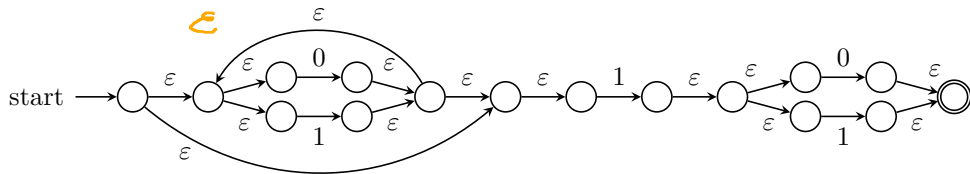
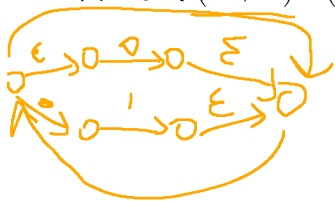
则正则表达式  $r + s$ ,  $rs$  和  $r^*$ , 可由  $R$  和  $S$  分别构造如下:



因此任何结构的正则表达式, 都可递归构造出等价的  $\varepsilon$ -NFA.

---

例9. 正则表达式  $(0+1)^*1(0+1)$  构造为  $\varepsilon$ -NFA.



### 思考题

正则表达式到  $\varepsilon$ -NFA 构造方法中的 3 个限制条件, 都有必要吗?