第5章　正则语言的性质

主要内容：

5.1 FA与RG的等价性

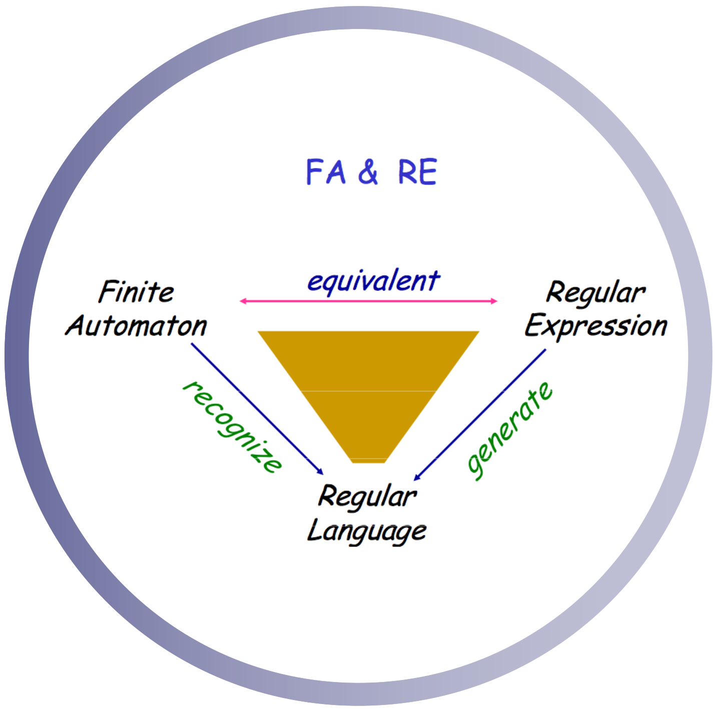
5.2 正则语言的泵引理

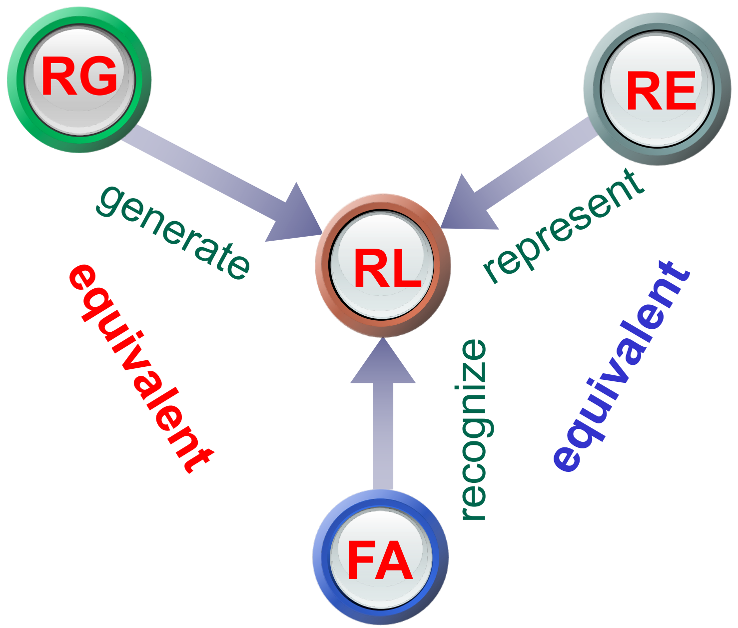
5.3 正则语言的封闭性

5.4 正则语言的判定算法

5.5 自动机的等价性与最小化

5.1 FA与RG的等价性





5.1 FA与RG的等价性

考察RG、FA的工作机制

$A_0 \Rightarrow a_1A_1$ 对应产生式$A_0 \rightarrow a_1A_1$

$\Rightarrow a_1a_2A_2$ 对应产生式$A_1 \rightarrow a_2A_2$

…

$\Rightarrow a_1a_2\ldots a_{n-1}A_{n-1}$ 对应产生式$A_{n-2} \rightarrow a_{n-1}A_{n-1}$

$\Rightarrow a_1a_2\ldots a_{n-1}a_n$ 对应产生式$A_{n-1} \rightarrow a_n$

$q_0\ a_1a_2\ldots a_{n-1}a_n$

$\vdash a_1q_1\ a_2\ldots a_{n-1}a_n$ 对应$\delta(q_0, a_1)=q_1$

$\vdash a_1a_2q_2\ldots a_{n-1}a_n$ 对应$\delta(q_1, a_2)=q_2$

……

$\vdash a_1a_2\ldots a_{n-1}q_{n-1}a_n$ 对应$\delta(q_{n-2}, a_{n-1})=q_{n-1}$

$\vdash a_1a_2\ldots a_{n-1}a_nq_n$ 对应$\delta(q_{n-1}, a_n)=q_n$

$\delta(A_0, a_1)=A_1$
$\delta(A_1, a_1)=A_2$
……
$\delta(A_{n-2}, a_1)=A_{n-1}$
$\delta(A_{n-1}, a_1)=A_n=f$，其中，$f \in F$。

可见：正则文法的推导与 DFA的状态转移可 **相互模拟.**

定理 5.1 FA接受的语言是正则语言。

证明：

构造出来的文法G与M等价吗？

(1) 根据FA M，构造 RG G。

基本思想是让RG的派生与DFA的状态转移相对应。

设DFA $M=(Q，\sum，\delta，q_0，F)$，

取右线性文法 $G=(Q，\sum，P，q_0)$，

$P=\{ q\rightarrow ap|\delta(q，a)=p \}\cup\{ q\rightarrow a|\delta(q，a)=p，p\in F \}$

（2）证明 $L(G)=L(M)-\{\varepsilon\}$。

对于$a_1a_2\ldots a_{n-1}a_n \in \Sigma^+$，且$a_1a_2\ldots a_{n-1}a_n \in L(G)$

$q_0 \Rightarrow^+ a_1a_2\ldots a_{n-1}a_n$

$\Leftrightarrow q_0 \to a_1q_1$， $q_1 \to a_2q_2$， …，

$q_{n-2} \to a_{n-1}q_{n-1}$， $q_{n-1} \to a_n \in P$

正则文法领域

$\Leftrightarrow \delta(q_0, a_1)=q_1$， $\delta(q_1, a_2)=q_2$， …

， $\delta(q_{n-2}, a_{n-1})=q_{n-1}$， $\delta(q_{n-1}, a_n)=q_n$， 且$q_n \in F$

$\Leftrightarrow \delta(q_0, a_1a_2\ldots a_{n-1}a_n)= q_n \in F$

$\Leftrightarrow a_1a_2\ldots a_{n-1}a_n \in L(M)$

自动机领域

（3）关于ε句子。

- 如果$q_0 \notin F$，则$\varepsilon \notin L(M)$，$L(G)=L(M)$。
- 如果$q_0 \in F$，则扩充G为G′，

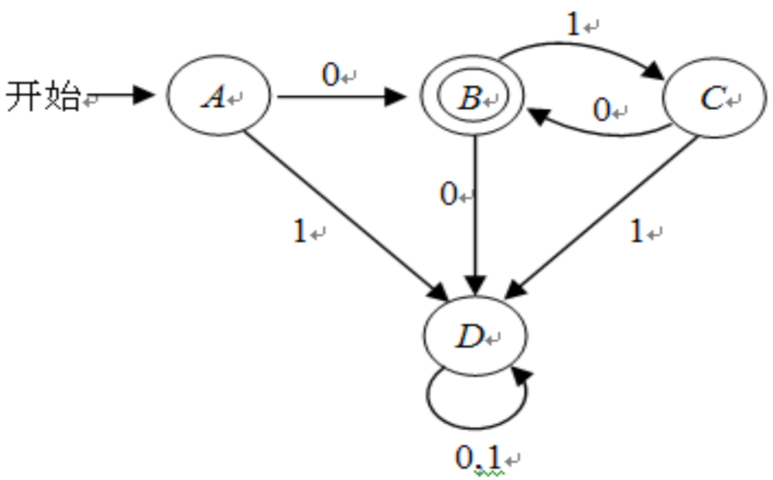
$L(G')=L(G) \cup \{\varepsilon\}=L(M)$。其中，G′比G增加一个开始符号S和两个产生式$S \rightarrow q_0 \mid \varepsilon$。

综上所述，对于任意DFA M，存在正则文法G，使得$L(G)=L(M)$。

定理得证。

5.1 FA与RG的等价性

EXP5.1 将下列DFA转化为等价的正则文法。



按照定理5 .1的构造方法，得出对应的正则文法是（A为开始符号）：

A→0B|0 , A→1D ,
B→0D ,　 B→1C ,
C→0B|0 , C→1D ,
D→0D ,　 D→1D 。

定理5.2 正则语言可以由FA接受。

证明：

（1）根据RG，构造FA。

基本思想：让FA模拟RG的派生过程。

设G=(V，T，P，S)，且$\varepsilon \notin L(G)$，

取FA M=($V \cup \{f\}$，T，δ，S，{f})，$f \notin V$。

5.1 FA与RG的等价性

定义产生式如下：

$$\delta(A, a)=\begin{cases}\{B|A\to aB\in P\}\cup\{f\} & \text{如果}A\to a\in P\\ \{B|A\to aB\in P\} & \text{如果}A\to a\notin P\end{cases}$$

也就是:

- 用$B\in\delta(A, a)$与产生式$A\to aB$对应；
- 用$f\in\delta(A, a)$与产生式$A\to a$对应。

思考：为什么要定义接受状态{ f }呢？

（2）证明L(M)=L(G)

对于$a_1a_2\ldots a_{n-1}a_n \in T^+$，

$a_1a_2\ldots a_{n-1}a_n \in L(G) \Leftrightarrow S \Rightarrow^+ a_1a_2\ldots a_{n-1}a_n$

$\Leftrightarrow S \Rightarrow a_1A_1 \Rightarrow a_1a_2A_2 \Rightarrow \ldots$

$\Rightarrow a_1a_2\ldots a_{n-1}A_{n-1} \Rightarrow a_1a_2\ldots a_{n-1}a_n$

$\Leftrightarrow S \to a_1A_1$，$A_1 \to a_2A_2$，…，

$A_{n-2} \to a_{n-1}A_{n-1}$，$A_{n-1} \to a_n \in P$

5.1 FA与RG的等价性

$\Leftrightarrow A_1 \in \delta(S，a_1)，A_2 \in \delta(A_1，a_2)，\ldots，$

$A_{n-1} \in \delta(A_{n-2}，a_{n-1})，f \in \delta(A_{n-1}，a_n)$

$\Leftrightarrow f \in \delta(S，a_1 a_2 \ldots a_{n-1} a_n)$

$\Leftrightarrow a_1 a_2 \ldots a_{n-1} a_n \in L(M)$

5.1 FA与RG的等价性

- 例5.2 给出正则文法G_1如下：

  $S\rightarrow 0B$， $B\rightarrow 0B$，

  $B\rightarrow 1S$， $B\rightarrow 0$ 。

- 根据定理5.2给出的方法，我们构造对应的有穷自动机 $M=(\{S,B,f\},\{0,1\},\delta,S,\{f\})$,其中：

  $\delta(S,0)=\{B\}$， $\delta(B,0)=\{B,f\}$，

  $\delta(B,1)=\{S\}$。

- 例5.3 给出正则文法G_2如下：

  S→0A， A→1A，

  A→B，B→0，B→ε。

- 我们构造对应的有穷自动机 M＝({S,A,B,f},{0,1},δ,S,{f}),其中：

  δ(S,0)={A}， δ(A,1)={A}，

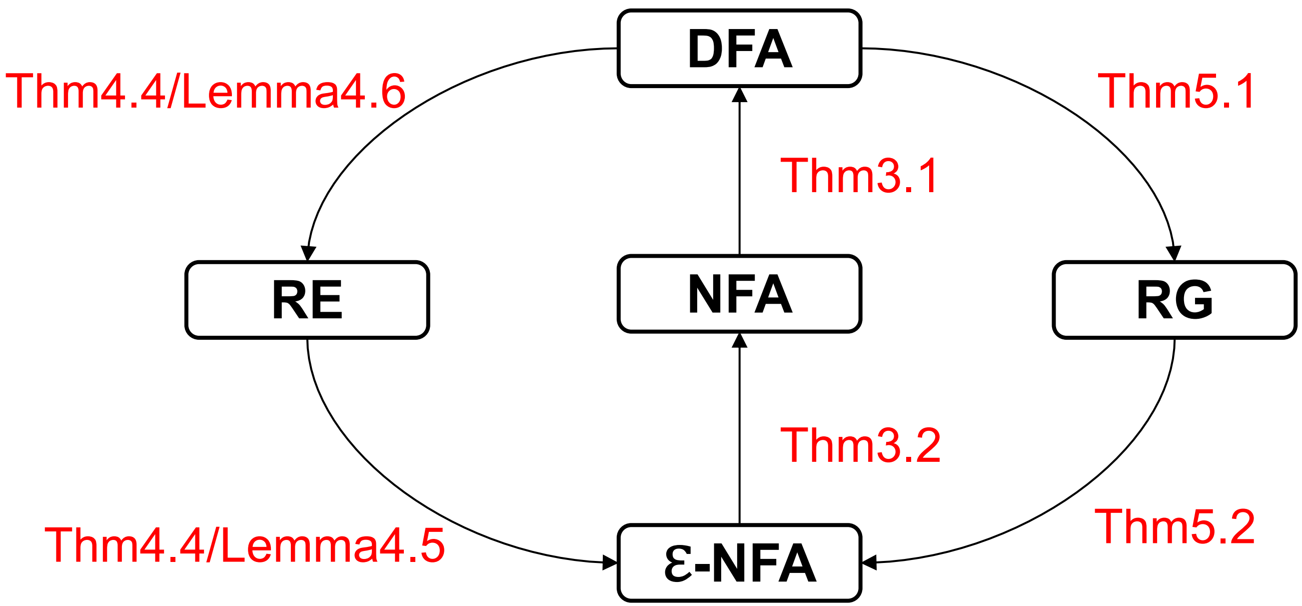
  δ(A,ε)={B}， δ(B,0)={f}，δ(B,ε)={f}。

- 注意：这个有穷自动机 M是具有ε-移动功能的。由于要考虑到一般情况，所以定理5.2 中必须要构造一个具有ε-移动的NFA M，才能接受一切由正则文法产生的语言。

5.1 FA与RG的等价性

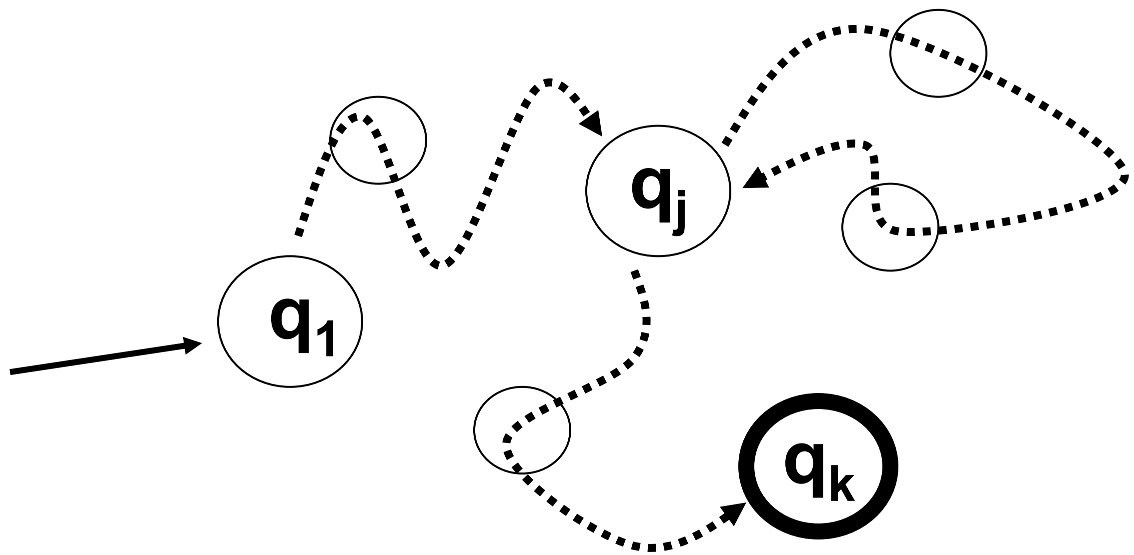
正则语言各种表达形式的关系

- 有向边代表两种表达式之间的**构造关系**



5.2 正则语言的泵引论

1. 有穷语言一定是正则的，无穷语言可能是正则的。
2. **如何判断一个无穷语言是否正则呢？**
3. 正则语言是靠***打圈***，来描述（有某种规律的）无限集合；



5.2 正则语言的泵引论

Pigeonhole principle(鸽巢原理): If p pigeons are placed into fewer than p holes, some hole has to have more than one pigeon in it.

m pigeons

n pigeonholes $\quad m > n$

There is a pigeonhole with at least 2 pigeons

5.2 正则语言的泵引论

The DFA Principle

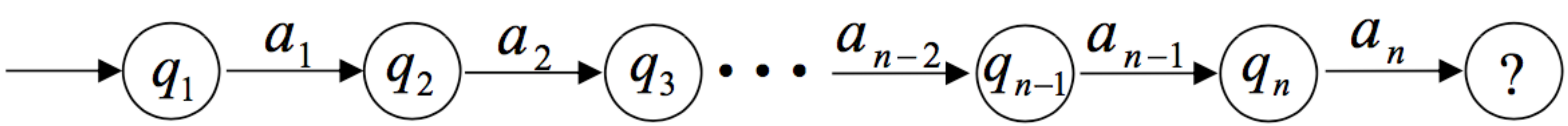
m symbols

$$w = a_1 a_2 \cdots\cdots a_m$$

n states

$$a_n \cdots\cdots a_m \ ?$$

$$m \geq n$$



5.2 正则语言的泵引论

Property of regular languages

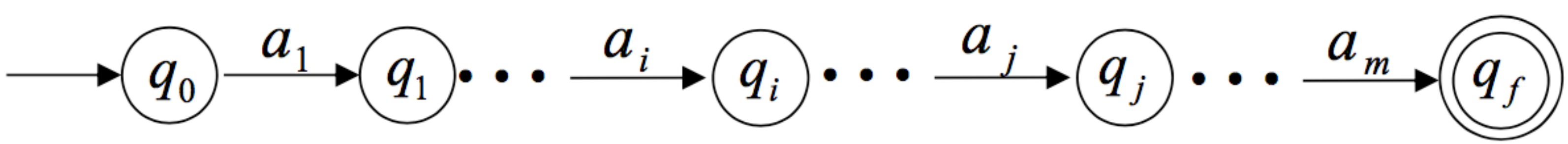
L is a regular language $\Rightarrow \exists$DFA $A : L(A) = L$

Let $A = (\mathrm{Q}, \Sigma, \delta, \mathrm{q}_0, F)$, and $n = |Q|$

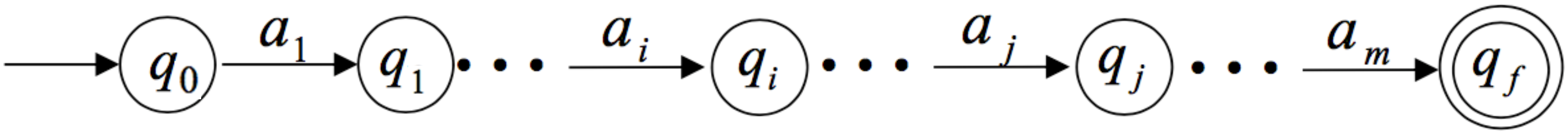
Get $w \in L$, and suppose $w = a_1 a_2 \cdots\cdots a_m, m \geq n$

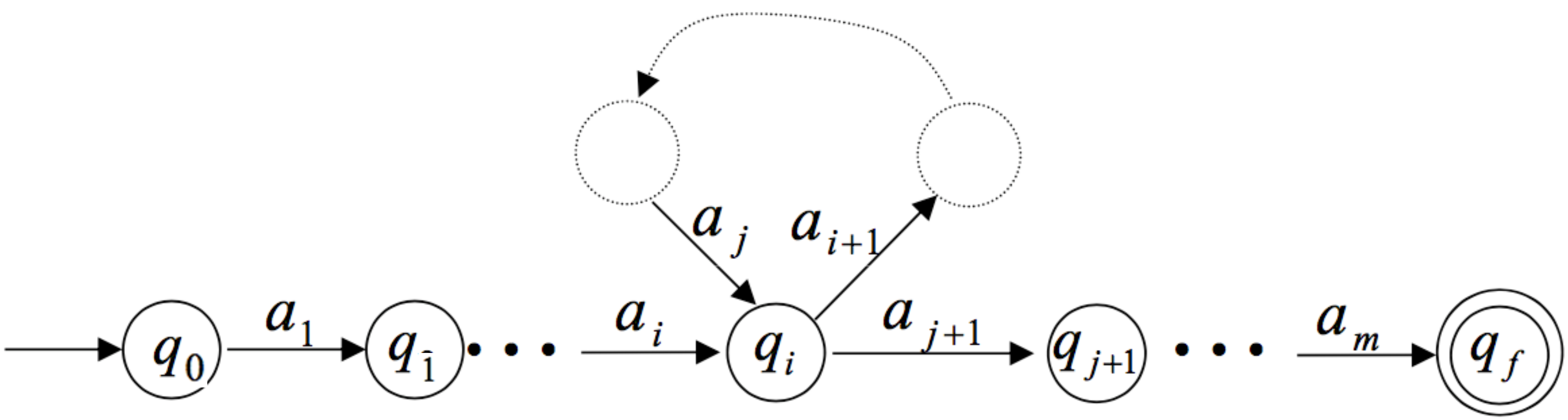
Let $q_{\mathrm{i}} = \bar{\delta}(q_0, a_1 a_2 \cdots a_i)$

$$\Rightarrow \exists 0 < i < j \leq n : q_i = q_j$$

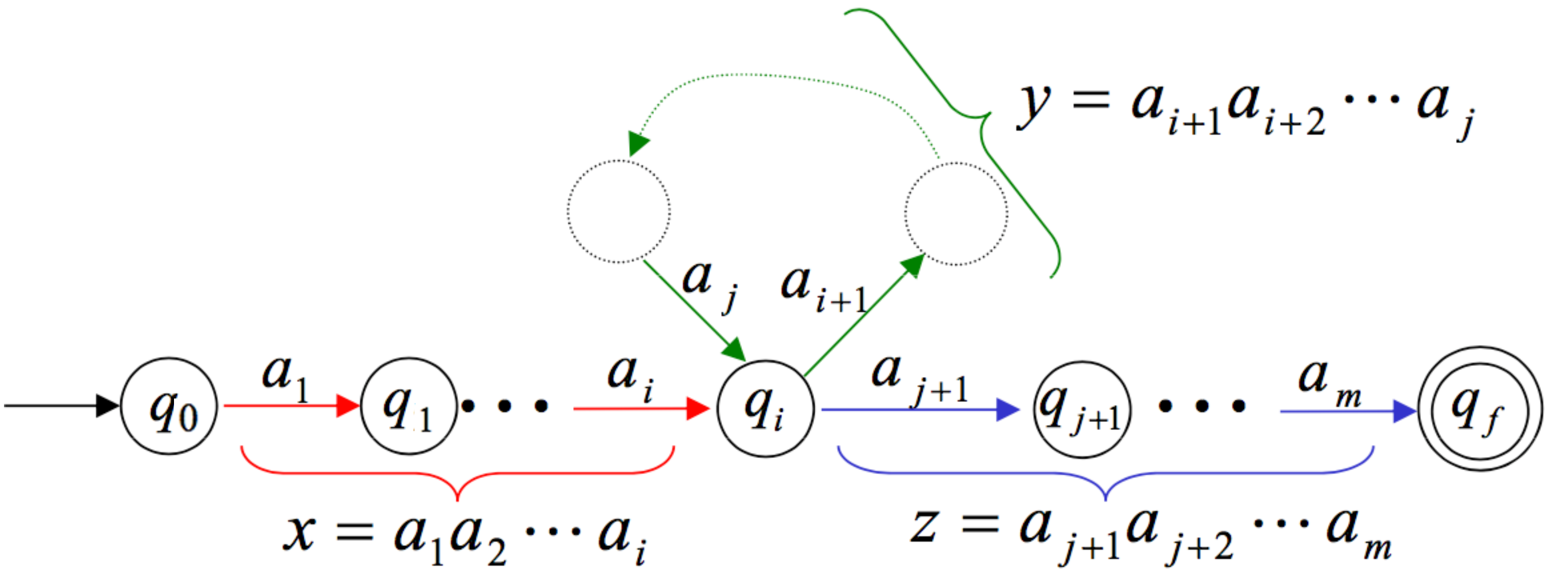


Property of regular languages





Property of regular languages



$$\Rightarrow w = xyz \begin{cases} |xy| \le n \\ |y| \ge 1 \text{ or } y \ne \varepsilon \\ xy^kz \in L, \text{ for any } k \ge 0 \end{cases}$$

5.2 正则语言的泵引论

Pumping lemma: For every regular language L, there is a *pumping length* **p**, such that for any string $s \in L$ and $|s| \geq p$, we can write $s=xyz$ with

1) $x\,y^i z \in L$ for every $i \in \{0,1,2,\ldots\}$ 为什么打圈？
2) $|y| > 0$
3) $|xy| \leq p$ 什么时候打圈？

Note that

1) implies that $xz \in L$
2) says that y cannot be the empty string ε
3) is not always used

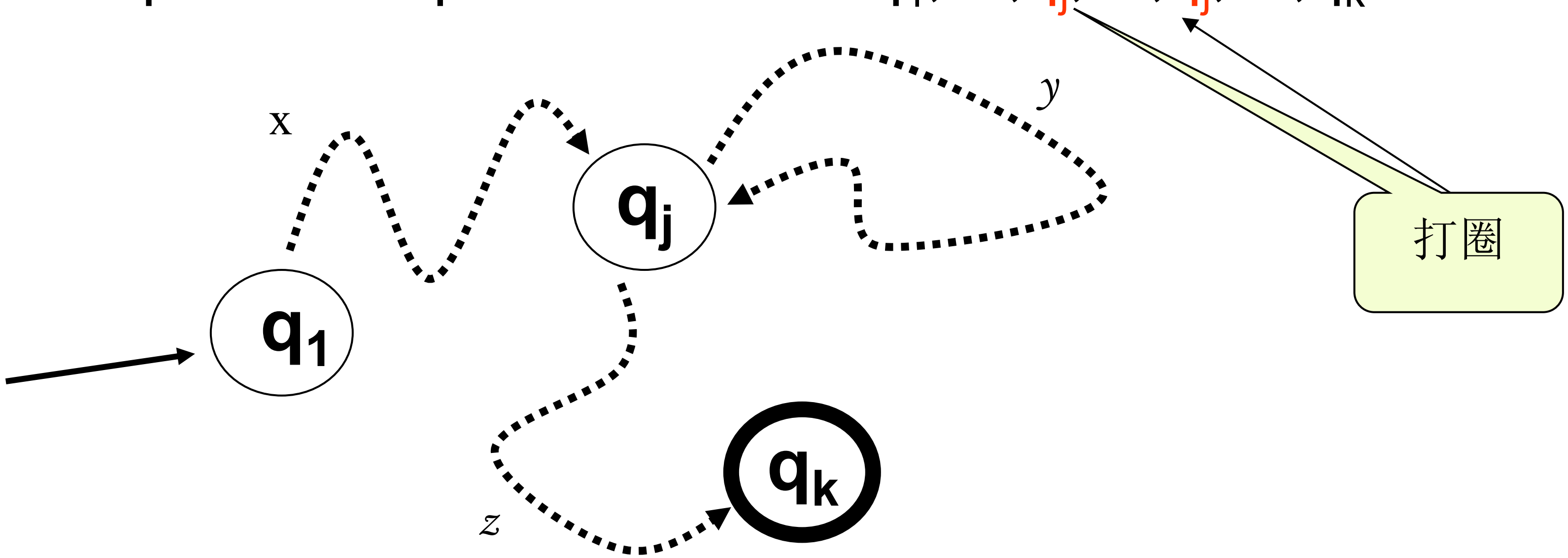
■ 经得起泵测试是**RL**的必要条件（不充分）。

Proof Idea:

① Consider an accepting DFA M with size |Q| 机器状态数

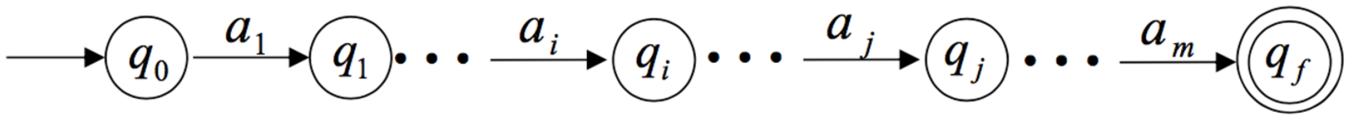
② On a string of length p, **p+1** states（识别某词的状态路径长度）

③ get visited for $p \geq |Q|$, there must be q_j ,such that the computational path looks like: $q_1, \ldots, q_j, \ldots, q_j, \ldots, q_k$

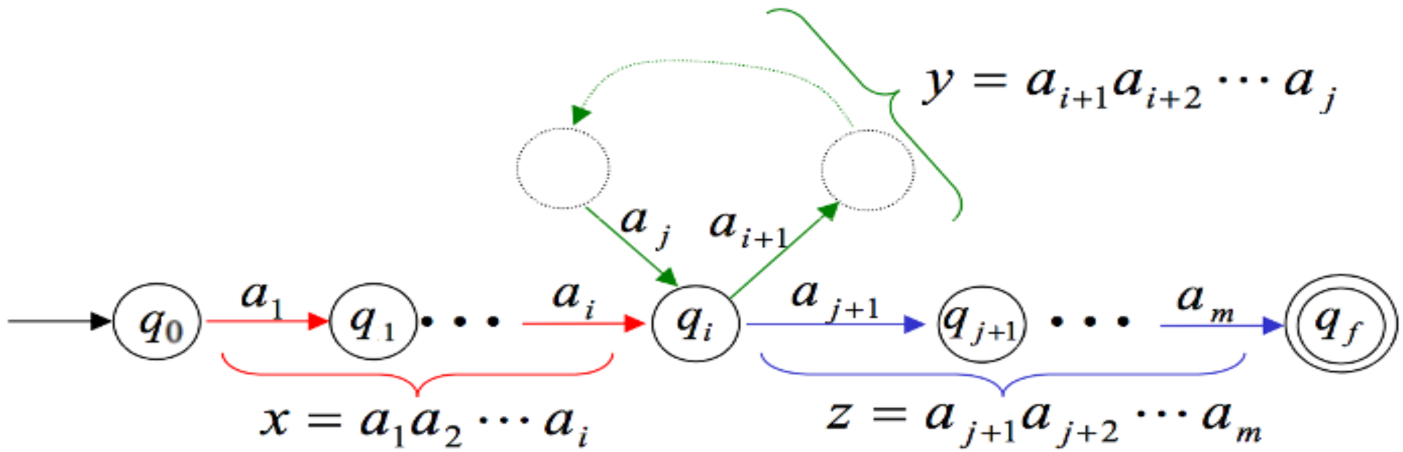


5.2 正则语言的泵引论

设p=｜Q｜=n；s=$a_1a_2\ldots a_m$，其中 m>n；q_f 是可接受状态。



根据鸽巢原理，上图中一定有两个状态相同：$q_i = q_j$，其中 $0 < i < j \le n$.



由上图可知：

1. $s = xy^kz,\ \forall k \ge 0 \Rightarrow s \in L$；
2. $\because i \ne j,\ \therefore |y| > 0$；
3. $\because j \le n,\ \therefore |xy| \le n$.

5.2 正则语言的泵引论

Proof Let $M = (Q, \Sigma, \delta, q_1, F)$ be a DFA recognizing Language A and p be the number of states of M. ($p = |Q|$)

Let $s=s_1 s_2 \cdots s_n$ be a string in A of length n, where $n \geq p$. Let $r_1,\ldots,r_{n+1}$ be the sequence of states that M enters while processing s, so $r_{i+1} = \delta(r_i,s_i)$ for $1 \leq i \leq n$. This sequence has length n+1, which is at least p+1. Among the first p + 1 elements in the sequence, two must be the same state, by the pigeonhole principle. We call the first of these r_j and the second r_k. Because r_k occurs among the first p+1 places in a sequence starting at r_1, we have $k \leq p+1$. Now let $x = s_1 \cdots s_{j-1}$, $y = s_j \cdots s_{k-1}$, and $z = s_k \cdots s_n$.

As x takes M from r_1 to r_j, y takes M from r_j to r_j, and z takes M from r_j to r_{n+1}, which is an accept state, M must accept xy^iz for $i \geq 0$. We know that $j \neq k$, so $|y| > 0$; and $k \leq p+1$, so $|xy| \leq p$. Thus we have satisfied all conditions of the pumping lemma.

EXP1:Prove B = $\{0^n1^n \mid n \geq 0\}$ is not regular.

1. Assume that B is regular 反证法
2. Let p be the pumping length, and s = $0^p1^p \in B$
   s = xyz = 0^p1^p, with $xy^iz \in B$ for all $i \geq 0$
   Three options for y:
      1) $y=0^k$, hence $xyyz = 0^{p+k}1^p \notin B$
      2) $y=1^k$, hence $xyyz = 0^p1^{k+p} \notin B$
      3) $y=0^k1^l$, hence $xyyz = 0^p1^l0^k1^p \notin B$
3. Conclusion: The pumping result does not hold, the language B is not regular.

EXP2, Show $F = \{ ww \mid w \in \{0,1\}^* \}$ is not RL.

反证法

Let p be the pumping length, and take word $s = 0^p10^p1$, $w=0^p1$

Let $s = xyz = 0^p10^p1$,

with condition 3) $|xy| \le p$

Only one option: $x=0^{p-k}$,$y=0^k$, $z=10^{p-k}1$, (保证xz in L)

with $xyyz = 0^{p+k}10^{p-k}1 \notin F$

Without 3) this would have been a pain.

思考题：

1. $L = \{ 0^n1^n \mid 0 \le n \le 100 \}$ 是正则语言吗？
2. 有限语言是否符合泵引论？

EXP3, Show $E=\{0^i1^j \mid i > j\}$ is not RL. *证明思路回顾*

Step 1: 选择反证法;

Step 2: 构造 string $s = 0^{p+1}1^p$; 利用泵长度p

Step 3: 发现矛盾

$s=xyz$，由引论3）$|xy|\le p$知：$y=0s$，令 $y=0^k$, $k>0$; y不可能包含1

$xy^iz = 0^{p-k}0^{k*i}1^p = 0^{p+k(i-1)}1^p$，当i=2时，显然 $xyyz \in E$; 可惜没矛盾，只好换一条路走

Pumping Down：The pumping lemma states that $xy^iz \in E$ even if when i=0, so lets consider the string $xy^0z=xz$.

结果怎么样呢？

$xz=0^p1^p \notin E$; 矛盾出现

Step 4: 得出结论。

再论泵引论

Pumping lemma: For every regular language L, there is a *pumping length* **p**, such that for any string $s \in L$ and $|s| \geq p$, we can write $s=xyz$ with

1) $x\,y^i z \in L$ for every $i \in \{0,1,2,\ldots\}$ 为什么打圈？鸽子比鸽笼多。
2) $|y| > 0$
3) $|xy| \leq p$ 什么时候打圈？字符数 ≥ 状态数。

Note that

1) implies that $xz \in L$，正则语言靠打圈

2) （1）$y \neq \varepsilon$，但x,z可以为空；（2）如果$y = \varepsilon$,引论也成立，只是毫无意义；（3）y不能为空，因为打的不是空圈（此q_i非彼q_j）。

3) $|xy| = p$ 时，至少打圈一次。

■ 泵引论描述了RL必须满足的条件（**必要条件，不充分**）。

再论泵引论

EXP4: 证明 $\{L = 1^p | p$ 是素数 $\}$ 不是正则语言。

证明：假设 L 是正则语言，则存在 n_0 满足泵引理的性质。那么由于串 $w = xyz = 1^{p_0}$ （其中 p_0 为大于 n_0 的素数）属于 L，串 $w' = xy^kz = 1^{p_0+(k-1)|y|}$ 也属于 L。取 $k = p_0 + 1$，$w' = 1^{p_0(1+|y|)}$ 不属 于 L，产生矛盾，因此 L 不是正则语言。

注意：这里使用了素数有无穷个的引论。

课堂作业：试证 $L = \{1^p | p$ 是合数 $\}$ 不是正则语言。

狄利克雷定理 对于任意互质的正整数 a, d，形式如 a + nd 的素数有无限多个，其中 n 为正整数。

再论泵引论

课堂作业：试证 $L = \{1^p | p$ 是合数 $\}$ 不是正则语言。

狄利克雷定理 对于任意互质的正整数 a, d, 形式如 a + nd 的素数有无限多个，其中 n 为正整数。

证明： 若 L 是正则语言，则存在 n_0 满足泵引理的性质。

那么由于串 $w = xyz = 1^{p.p}$ (其中 p 为大于 n_0 的素数)属于 L，串 $w' = xy^kz = 1^{p.p+(k-1)|y|}$ 也属于 L。

由于 $|y| \le n_0$，p^2 和 $|y|$ 互素，由狄利克雷定理，存在正整数 k 使得 $p^2 + (k-1)|y|$ 为素数，有 w' 不属于 L，产生矛盾，因此 L不是正则语言。

5.3 正则语言的封闭性

- 定义 5.1 如果属于某个语言类的任何语言在某个特定运算下所得的结果仍然属于该语言类,则称该语言类对这个运算是封闭的，并称该语言类对这个运算具有**封闭性**。
- 正则语言对于许多运算都是封闭的，下面先列出正则语言的一些主要的封闭性，然后逐个加以证明。

①两个正则语言的**并**是正则语言。

②两个正则语言的**连接**是正则语言。

③正则语言的**闭包**是正则语言。

④两个正则语言的**交**是正则语言。

⑤正则语言（对全集）的**补**是正则语言。

⑥两个正则语言的**差**是正则语言。

⑦正则语言的**逆转**是正则语言。

⑧正则语言的**同态**是正则语言。

⑨正则语言的**逆同态**是正则语言。

5.3 正则语言的封闭性

定理 5.3　正则语言在并、连接和闭包运算下是封闭的。

证明　已知M,N为两个正则语言，代表它们的正则表达式分别是R和S，即L(R)=M ,L(S)=N。因为正则表达式R+S代表语言L(R)∪L(S)= M∪N，所以M∪N仍是正则语言。至于正则语言在连接和闭包运算下的封闭性，只要考虑正则表达式和正则语言的关系，参照上述论述，不难证明它们的正确性。定理证完。

定理 5.4 正则语言在补运算下封闭，即若L是字母表$\sum$上的正则语言，则$\sum^*$ - L也是正则语言。

证明 设L被某个DFA M =(Q,$\sum$,δ,q_0 ,F)接受，为了接受语言$\sum^*$ - L，现在由M构造一个新的DFA M´，M´= (Q,$\sum$,δ,q_0，Q-F)。这里M´与M的终结状态对Q而言互为补集，也就是说，M的终结状态变为M´的非终结状态, M的非终结状态变为M´的终结状态。M´和M的其余的结构相同。显然，对于M接受的字符串，M'将不接受；对于M'接受的字符串，M将不接受。因此，L(M´)= $\sum^*$ - L。定理证完。

5.3 正则语言的封闭性

定理 5.5 正则语言在交运算下封闭。

证明 设L_1和L_2都是某个字母表Σ上的正则语言，根据集合运算的狄·摩尔根定律，有

$$\overline{L_1 \cap L_2} = \overline{L_1} \cup \overline{L_2}$$

这里求补运算是在全集Σ^*下进行的。根据正则语言对补运算和并运算的封闭性，则$L_1 \cap L_2$仍是正则语言。定理得证。

5.3 正则语言的封闭性

交运算下封闭的构造性证明

- 设有两个DFA M_1和M_2，其中：

  $M_1=(Q_1, \Sigma, \delta_1, q_1, F_1)$,使得$L(M_1)=L_1$。

  $M_2=(Q_2, \Sigma, \delta_2, q_2, F_2)$,使得$L(M_2)=L_2$。

- 现在构造新的DFA M:

  $M=(Q_1 \times Q_2, \Sigma, \delta, [q_1, q_2], F_1 \times F_2)$。

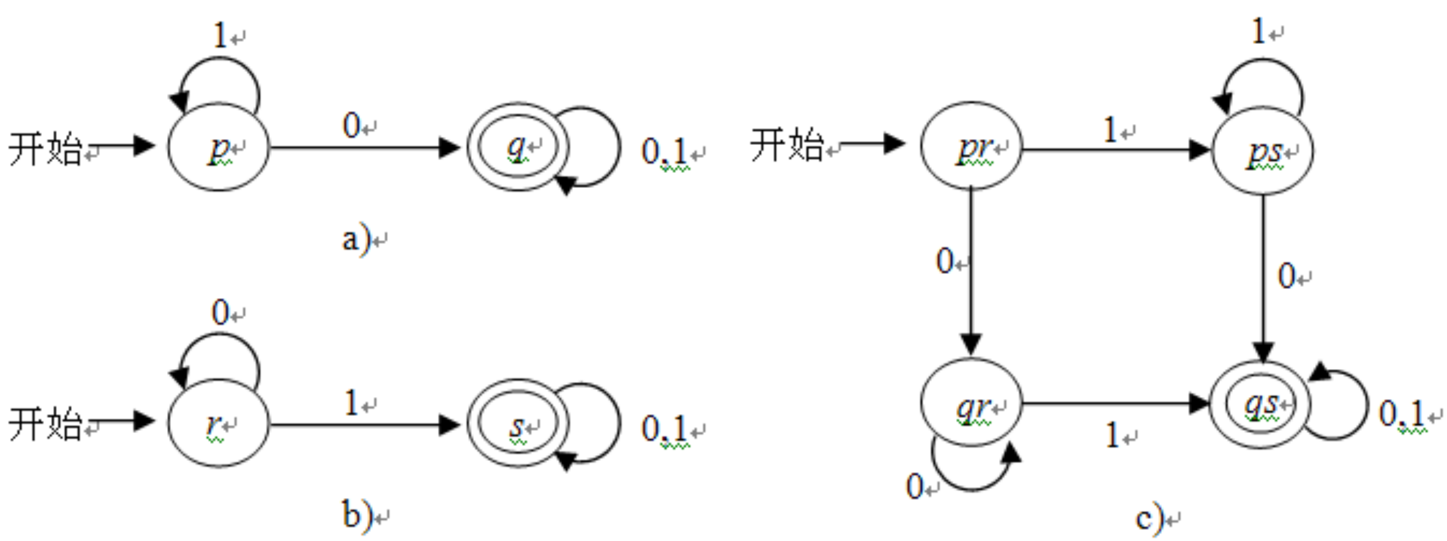
  这里，对一切$p_1 \in Q_1$, $p_2 \in Q_2$,和一切$a \in \Sigma$,

  $\delta([p_1, p_2], a)=[\delta_1(p_1, a), \delta_2(p_2, a)]$。

- 直观来看，这里的M是将M_1和M_2“捆绑”在一起运行，它从初始状态$[q_1, q_2]$开始，在输入串上让M_1和M_2并行工作，只有当读完输入串以后，M_1和M_2同时到达它们各自的终结状态，M才接受该输入串。显然，$L(M)= L(M_1) \cap L(M_2)$。

5.3 正则语言的封闭性

- 例 5.4 下图左边可以看到两个DFA a)，b)，将它们按定理 5. 5 的构造方法“捆绑”在一起，得到右边的DFA c)。可以验证，DFA c)接受的语言恰好是 a)，b)，两个DFA接受语言的交集。



5.3 正则语言的封闭性

定理 5.6　如果L和M是正则语言，则L- M也是正则语言。

证明 根据集合运算关系，$L-M = L \cap \overline{M}$。由于已证过的正则语言对补运算和交运算的封闭性，可知L-M也是正则语言。

5.3 正则语言的封闭性

定理 5.7　如果L是正则语言，则$L^R = \{x \mid x^R \in L\}$也是正则语言。

- 要证明L^R是正则语言，有两种方法：一种是基于有穷自动机，另一种是基于正则表达式。
- 先非形式地给出基于有穷自动机的证明。因为L是正则语言，则一定有一个DFA M来接受它。我们通过下述方法构造接受L^R的有穷自动机：

① 把M的状态转移图中的所有有向边的指向逆转。

② 令M的初始状态为新的有穷自动机的唯一的终结状态。

③ 增加一个状态p_0为新的有穷自动机的初始状态，同时从该状态出发到M的所有终结状态都建立一个ε转移。

- 以上的结果是构造一个“倒过来”模拟M的具有ε转移的有穷自动机，它接受串x当且仅当M接受x^R。下面给出基于正则表达式的严格证明。

5.3 正则语言的封闭性

- 设正则语言L由正则表达式E代表，现在要由E构造新的正则表达式E^R代表L^R。这个过程对E的构造次数进行归纳来完成。
- 归纳基础 E的构造次数为0。即E是ε、φ或∑中的某个符号a ,则E^R和E相同。也就是说，我们有$\{\varepsilon\}^R=\{\varepsilon\}$，$\varphi^R=\varphi$，$\{a\}^R=\{a\}$。
- 归纳步骤 设当E的构造次数小于k时，能由E构造新的正则表达式E^R代表L^R，现在考虑E的构造次数等于k。根据E的最后一次构造，共分三种情况：

1. $E = E_1 + E_2$。其中E_1和E_2都是由小于k次构造构成的正则表达式，由归纳法假设，可以构造相应的$E^R{}_1$和$E^R{}_2$，其中$L(E^R{}_1)=L(E_1)^R$，$L(E^R{}_2)=L(E_2)^R$。因为$L(E)=L(E_1)\cup L(E_2)$，所以$L(E)^R=L(E_1)^R\cup L(E_2)^R$。得出 $E^R = E^R{}_1 + E^R{}_2$ 就是代表$L(E)^R$的正则表达式。

5.3 正则语言的封闭性

2. $E = E_1 E_2$。由归纳法假设，可以构造$E^R{}_1$和$E^R{}_2$，其中$L(E^R{}_1)=L(E_1)^R$，$L(E^R{}_2)=L(E_2)^R$。这里要注意：因为$L(E)=L(E_1)L(E_2)$，而$L(E)^R =L(E_2)^R L(E_1)^R$，得出 $E^R = E^R{}_2 E^R{}_1$就是代表$L(E)^R$的正则表达式。
3. $E = E^*{}_1$。由归纳法假设，可以构造$E^R{}_1$，使得$L(E^R{}_1)=L(E_1)^R$。因为$L(E)=L(E_1)^*$，则任何$L(E)$中的串w都可以写成$w_1w_2\ldots w_n$，其中每个w_i都属于$L(E_1)$。而$w^R = w_n{}^R w_{n-1}{}^R \ldots w_1{}^R$，且其中每个$w_i{}^R$都属于$L(E_1)^R$，也就是属于$L(E_1{}^R)$，因此$w^R$属于$L((E_1{}^R)^*)$。反之，$L((E_1{}^R)^*)$中任何串w都有$w_1w_2\ldots w_n$的形式，其中每个$w_i$是$L(E_1)$中某个串的逆转。因此，$w^R = w_n{}^R w_{n-1}{}^R \ldots w_1{}^R$在$L(E_1{}^*)$中，也就是在$L(E)$中。所得结论就是："一个串在$L(E)$中当且仅当它的逆转在$L((E_1{}^R)^*)$中"。于是得出$(E_1{}^R)^*$就是代表$L(E)^R$的正则表达式。

5.3 正则语言的封闭性

定义 5.2 设$\sum$，Δ是两个字母表，h是从$\sum$到Δ^*的全映射。如果对于一切x，y$\in\sum^*$，都有h(xy)= h(x)h(y)，则h称为从$\sum$到Δ^*的**同态映射**。

对于所有L $\sum^*$，L的**同态像**是Δ^*的一个子集，定义为：

$$h(L)= \quad \{h(x)\}。$$

对于所有w$\in\Delta^*$，w的**同态原像**是$\sum^*$的一个子集，定义为：

$$h^{-1}(w)=\{x|h(x)=w\}。$$

对于所有L Δ^*，L的**同态原像**是$\sum^*$的一个子集，定义为：

$$h^{-1}(L)=\{x|h(x)\in L\}。$$

并且称$h^{-1}(L)$为L的**逆同态**。

5.3 正则语言的封闭性

- 例 5.5 设$\Sigma=\{0,1\}$，$\Delta=\{a,b\}$,同态映射h由下式给出：

  $h(0)=aa$,

  $h(1)=aba$ 。

- 在这个同态映射下，我们有

(1) $h(010)=aaabaaa$ 。

(2) $h((01)^*)=(aaaba)^*$ 。

(3) $h^{-1}(baa)=\varphi$。

(4) $h^{-1}(aa)=\{0\}$

(5) $h^{-1}(\{aaa,aba,aaaaaba,abaaa,bba\})=\{1,001,10\}$。

(6) $h(h^{-1}(\{aaa,aba,aaaaaba,abaaa,bba\}))=$

$$h(\{1,001,10\})=\{aba,aaaaaba,abaaa\}$$ 。

从（6）式可以看出，一般来说，对于任意语言L，$h(h^{-1}(L))\neq L$。但是，不难证明，对于任意语言L和同态映射h，总有 $h(h^{-1}(L))$ L。

5.3 正则语言的封闭性

定理 5.8　如果L是字母表∑上的正则语言，h是字母表∑到Δ上的一个同态映射，则h(L)也是Δ上的正则语言。

证明　设L=L(R)，其中R是∑上的正则表达式。一般地，对于∑上的正则表达式E，用h(E)表示把E中每个∑中的符号a用h(a)来代换后所得到的Δ上的正则表达式。我们将要证明h(R)定义的语言是h(L)。

对R的构造次数用归纳法证明下述命题，h(R)代表的语言就和对语言L(R)应用h后得到的语言相同。形式化描述就是L(h(R))=h(L(R))。

5.3 正则语言的封闭性

- 归纳基础 R的构造次数为0。即R是ε、φ或者∑中的某个符号a。对于前两种，显然h(R)和R相同，因为映射h对ε和φ不起作用，所以L(h(R))=L(R)。
  而当R是ε或φ时，L(R)中或者只含空串，或者没有串，在这两种情况下，h(L(R))= L(R)。因此得出L(h(R))=h(L(R))。
  当R是∑中的某个符号a时，此时L(R)={a}，所以h(L(R))={h(a)}。另一方面，h(R)也正是由符号串h(a)表示的Δ上的正则表达式，因此L(h(R))也就是{h(a)}，最后得出L(h(R))=h(L(R))。

5.3 正则语言的封闭性

- 归纳步骤 R的构造次数为k。根据最外层的构造，共分三种情况：

① R=F+G。这里F和G都是由小于k次构造出的正则表达式。先由正则表达式应用同态的规定，我们有h(R)=h(F+G)=h(F)+h(G)。从而有

$$L(h(R))=L(h(F)+h(G))= L(h(F))\cup L(h(G)) \qquad (5\text{-}3)$$

再者，由于当把映射h作用于一个语言上时，相当于把它单独作用到该语言的每一个串，于是我们得到

$$h(L(R))= h(L(F)\cup L(G))= h(L(F))\cup h(L(G)) \qquad (5\text{-}4)$$

最后根据归纳法假设，有L(h(F))=h(L(F))和L(h(G))=h(L(G))。得知(5-3)、(5-4)两式的右端是相等的，所以两式的左端也相等，即L(h(R))= h(L(R))。

5.3 正则语言的封闭性

②R=FG。与R=F+G情形完全类似，可以平行地证明。

③$E=F^*$ 。参照②也可类似证明。

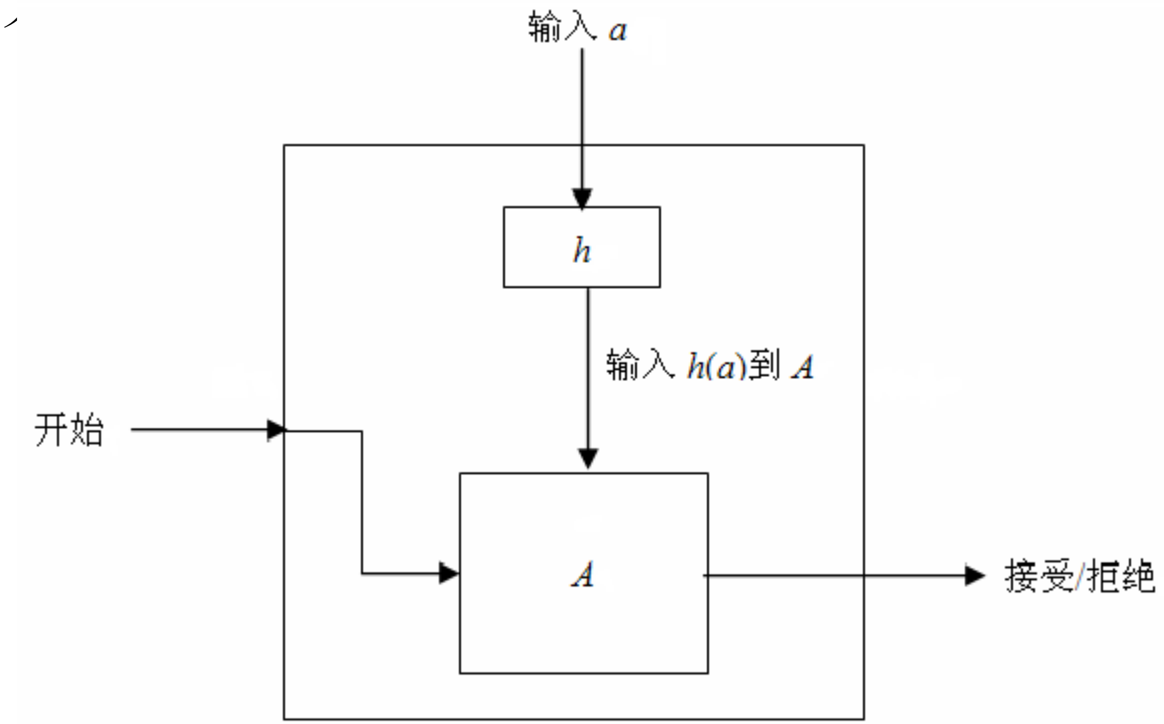
- 结论是对任何正则表达式R，L(h(R))= h(L(R))。也就是说，对代表一个语言的正则表达式应用同态h所得到的正则表达式恰好带表了语言h(L)。也就是说，h(L)是Δ上的正则语言。定理证完。

5.3 正则语言的封闭性

定理 5.9 如果h是字母表∑到字母表Δ的同态，且L是字母表Δ上的正则语言，则$h^{-1}(L)$也是字母表∑上的正则语言。

证明

- 首先从接受L的一个DFA A出发，我们从A和h构造一个接受$h^{-1}(L)$的DFA B（如图所示）。这个DFA B使用A的状态，但是在决定转移到哪一个



5.3 正则语言的封闭性

- 上述思想的形式化描述就是：设L是L(A)，其中DFA A=(Q,Δ,δ,q_0,F)。定义一个DFA

$$B=(Q,\sum,\gamma,q_0,F)$$

- 其中转移函数γ定义为：$\gamma(q,a)= \ \ (q, h(a))$。也就是说，DFA B对于输入a的转移结果就是DFA A对于符号序列h(a)的一系列转移的结果。虽然h(a)可能是ε、一个符号或多个符号，但对这些情况 $(q, h(a))$都有定义。
- 很容易对输入串w的长度|w|进行归纳来证明 $(q_0,w)= \ \ (q_0,h(w))$。因为A和B的终结状态相同，所以B接受w 当且仅当A 接受h(w)。换句话说，DFA B恰好接受那些$h^{-1}(L)$中的串w，即$h^{-1}(L)$是正则语言。定理证完。

5.4 正则语言的判定

- “给定的DFA接受的集合是空集吗？”
- “给定的DFA接受的集合是有穷集（或无穷集）吗？”
- “给定两个DFA是否接受同一个集合（是否等价）？”
- “给定一个DFA M和一个字符串x，M能接受x吗？”

对于这些问题，是否存在一个算法能回答“是”或者“否”，如存在，则称该算法为判定算法，相应的问题称为可判定的，否则称为不可判定的。

5.4 正则语言的判定

1. 判定（decide）和识别（recognize）的区别？

2. 可判定（decidable）与可计算是等价的；

3. 一个问题能否被一个算法解决，也就是这个问题是否可计算；

4. 通常，我们将计算问题转化成语言的归属问题；

EXP：问题1: 检测DFA B是否接受输入字符串ω？

等价于 问题2: 检测<B,ω>是否属于语言A_{DFA}？

$A_{DFA}=\{\langle B,\omega\rangle \mid B \text{ is a DFA that accepts input string } \omega\}$

5.4 正则语言的判定

定理 5.10 设DFA $M=(Q, \Sigma, \delta, q_0, F)$，$L=L(M)$非空的充分必要条件是：存在$x\in\Sigma^*$，$|x|<|Q|$，$\delta(q_0, x)\in F$。

定理5.11设DFA $M=(Q, \Sigma, \delta, q_0, F)$，$L=L(M)$为无穷的充分必要条件是：存在$x\in\Sigma^*$，$|Q|\leqslant|x|<2|Q|$，$\delta(q_0, x)\in F$。

定理 5.12 设DFA $M_1=(Q_1, \Sigma, \delta_1, q_{01}, F_1)$，DFA $M_2=(Q_2, \Sigma, \delta_2, q_{02}, F_2)$，则存在判定$M_1$与 M_2是否等价的算法。

定理 5.13 设L是字母表Σ上的 RL ，对任意$x\in\Sigma^*$，存在判定x是不是L的句子的算法。

关于判定的详细内容，参见《Introduction to the Theory of Computation 》Part Two: Computability Theory.