第 四 讲

Regular Languages and Regular Expressions

正 则 表 示

- 单一终结状态的 NFA
- 正则语言的运算性质
- 正则表示和语言
- 正则表示和正则语言
- 正则语言的同态
- 正则表示的代数定律

正 则 表 示

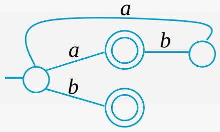
- 单一终结状态的 NFA
- 正则语言的运算性质
- 正则表示和语言
- 正则表示和正则语言
- 正则语言的同态
- 正则表示的代数定律

单一终态的NFA

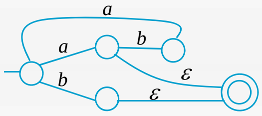
任何一个NFA都可以等价于只有一个终态的NFA

例

NFA

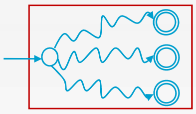


等价NFA

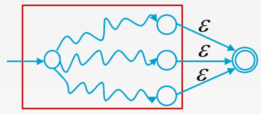


一般情形

NFA



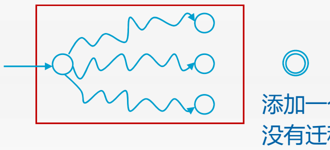
等价NFA



单一终结状态

一般情形

没有终结状态的NFA



添加一个
没有迁移的终态

正 则 表 示

- 单一终结状态的 NFA
- 正则语言的运算性质
- 正则表示和语言
- 正则表示和正则语言
- 正则语言的同态
- 正则表示的代数定律

正则语言的运算性质

并: $L_1 \cup L_2$

连接: $L_1 L_2$

星运算: $L_1{}^*$

反转: $L_1{}^R$

补: $\overline{L_1}$

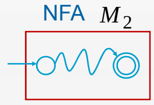
是否还是正则语言？

正则语言对上述运算封闭

正则语言的运算性质

正则语言 L_1 　　正则语言 L_2

NFA M_1 　　NFA M_2

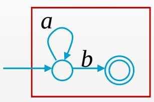


单一终结状态 　　单一终结状态

$L(M_1) = L_1$ 　　$L(M_2) = L_2$

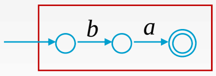
例

$L_1 = \{a^n b \mid n \geq 0\}$



M_1

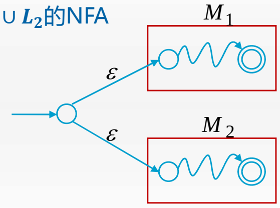
$L_2 = \{ba\}$

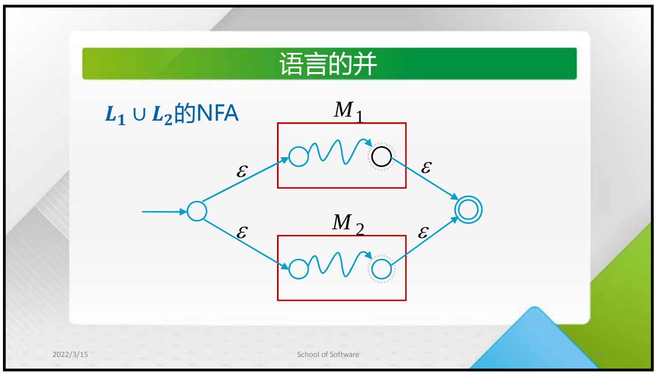


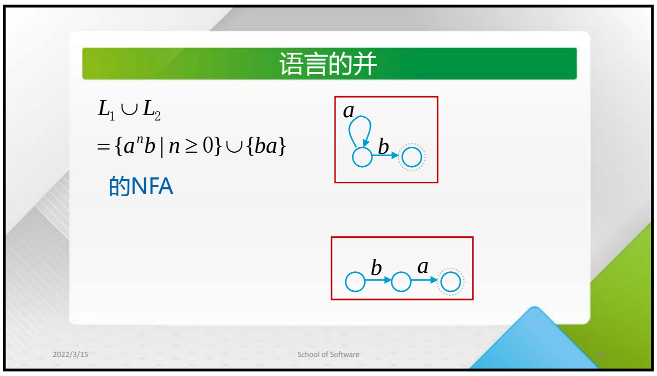
M_2

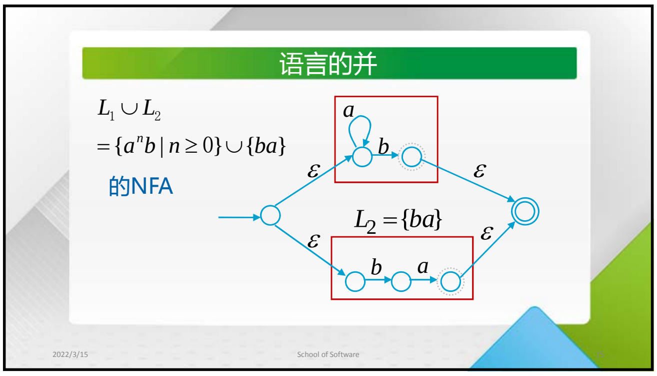
语言的并

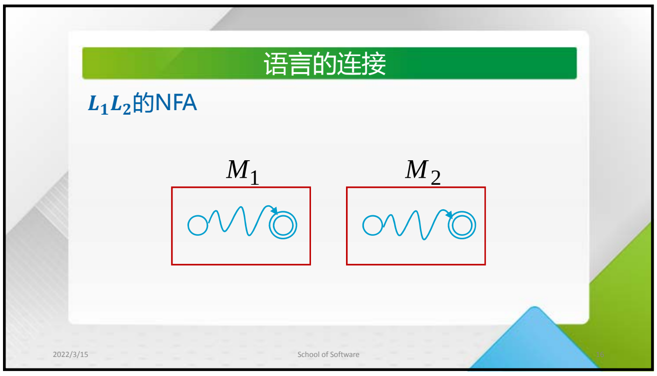
$L_1 \cup L_2$的NFA

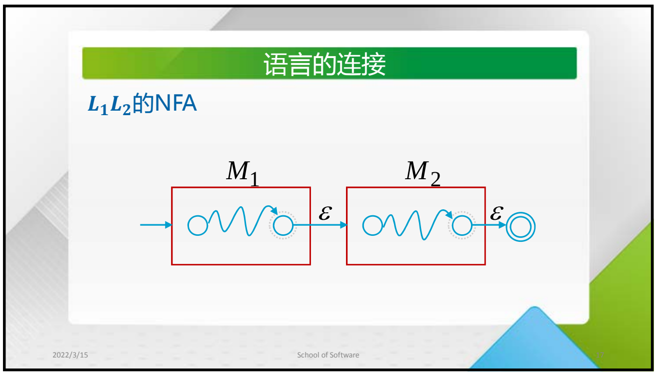


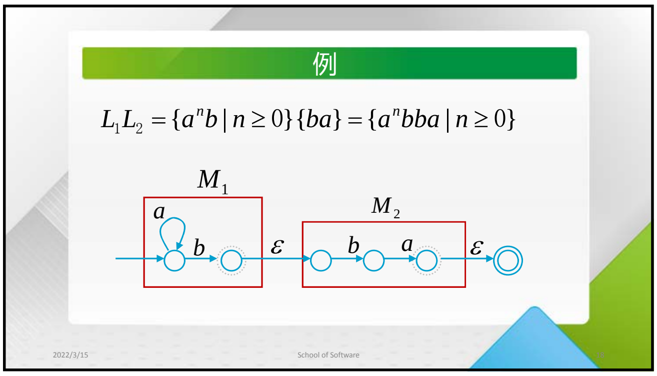

语言的并
$L_1 \cup L_2$的NFA
M_1
M_2
ε
ε
ε
ε


语言的并
$L_1 \cup L_2$
$=\{a^n b \mid n \geq 0\} \cup \{ba\}$
的NFA
a
b
b
a


语言的并
$L_1 \cup L_2$
$=\{a^n b \mid n \geq 0\} \cup \{ba\}$
的NFA
a
b
$L_2 = \{ba\}$
b
a
ε
ε
ε
ε

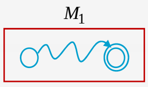

语言的连接
$L_1 L_2$的NFA
M_1
M_2


语言的连接
$L_1 L_2$的NFA
M_1
M_2
ε
ε


例
$L_1 L_2 = \{a^n b \mid n \geq 0\}\{ba\} = \{a^n bba \mid n \geq 0\}$
M_1
M_2
a
b
ε
b
a
ε

星闭包

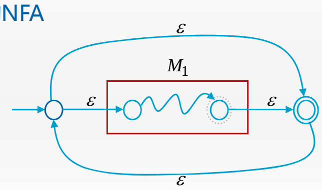
L_1^*的NFA



星闭包

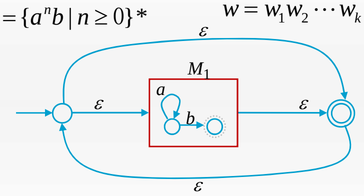
L_1^*的NFA

$\varepsilon \in L_1^*$

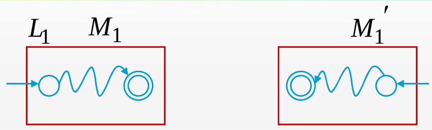


例

$L_1^* = \{a^n b \mid n \geq 0\}^*$ $w = w_1 w_2 \cdots w_k, w_i \in L_1$



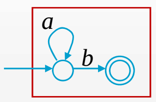
语言的反转



1. 反转所有的迁移
2. 将初态变为终态，反之亦然

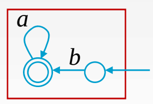
例

$L_1 = \{a^n b \mid n \geq 0\}$



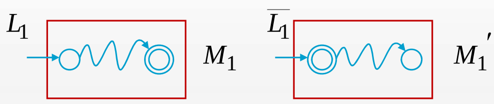
M_1

$L_1^R = \{ba^n \mid n \geq 0\}$



M_1'

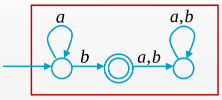
语言的补



1 .设计接受语言L_1的DFA
2 .将所有的终态改为非终态，反之亦然

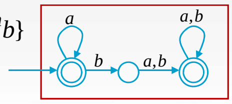
例

$L_1 = \{a^n b \mid n \geq 0\}$



M_1

$\overline{L_1} = \{a,b\}^* - \{a^n b\}$



$M_1{}'$

语言的交

正则语言 L_1, L_2

德摩根律：

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

L_1，L_2 正则

➡ $\overline{L_1}$，$\overline{L_2}$ 正则

➡ $\overline{L_1} \cup \overline{L_2}$ 正则

➡ $\overline{\overline{L_1} \cup \overline{L_2}}$ 正则

➡ $L_1 \cap L_2$ 正则

例

$L_1 = \{a^n b \mid n \geq 0\}$ 正则

$L_2 = \{ab, ba\}$ 正则

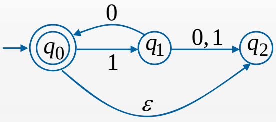
$\Rightarrow L_1 \cap L_2 = \{ab\}$

正则

正 则 表 示

- 单一终结状态的 NFA
- 正则语言的运算性质
- 正则表示和语言
- 正则表示和正则语言
- 正则语言的同态
- 正则表示的代数定律

例



$L(M) = \{\varepsilon,\ 10,\ 1010,\ 101010,\ \ldots\}$
$= \{10\}^*$

正则表示

正则表示也可以表示语言。如:

$(a + b \cdot c)^*$

描述了如下的语言

$\{a, bc\}^* = \{\varepsilon, a, bc, aa, abc, bca, \ldots\}$

正则表示

定义（归纳）：

基础： $\varnothing, \varepsilon, a$

给定两个正则表示 r_1, r_2

$$\left.\begin{array}{l} r_1 + r_2 \\ r_1 \cdot r_2 \\ r_1 * \\ (r_1) \end{array}\right\} \text{都是正则表示}$$

例

$$(a+b\cdot c)^* \cdot (c+\varnothing)$$

是正则表示

$$(a+b+)$$

不是正则表示

正则表示

正则表示算符优先级依次为

- * (闭包)
- • (连接)
- \+ (并)

正则表示的语言

正则表示可以描述语言

正则表示 r 的语言记为 $L(r)$

例：

$$L\left((a+b\cdot c)^*\right) = \{\varepsilon, a, bc, aa, abc, bca, \ldots\}$$

正则表示的语言

定义：对基本的正则表示

$$L(\varnothing) = \varnothing$$

$$L(\varepsilon) = \{\varepsilon\}$$

$$L(a) = \{a\}$$

正则表示的语言

对正则表示 r_1和 r_2

$$L(r_1 + r_2) = L(r_1) \cup L(r_2)$$

$$L(r_1 \cdot r_2) = L(r_1)\, L(r_2)$$

$$L((r_1)) = L(r_1)$$

$$L(r_1^*) = (L(r_1))^*$$

例

正则表示

$(a+b)\cdot a^*$

$$
\begin{aligned}
L\left((a+b)\cdot a^*\right) &= L\left((a+b)\right) L\left(a^*\right) \\
&= L(a+b)\, L\left(a^*\right) \\
&= \left(L(a)\cup L(b)\right)\left(L(a)\right)^* \\
&= \left(\{a\}\cup\{b\}\right)\left(\{a\}\right)^* \\
&= \{a,b\}\{\varepsilon,a,aa,aaa,\ldots\} \\
&= \{a,aa,aaa,\ldots,b,ba,baa,\ldots\}
\end{aligned}
$$

例

正则表示

$$r=(a+b)^*(a+bb)$$

$$L(r)=\{a,bb,aa,abb,ba,bbb,\ldots\}$$

例

正则表示

$$r=(aa)^*(bb)^*b$$

$$L(r)=\{a^{2n}b^{2m}b \mid n,m\geq 0\}$$

例

正则表示

$$r=(0+1)^*00\,(0+1)^*$$

$L(r)$ = {至少有2个连续0的字符串}

例

正则表示

$$r=(1+01)^*(0+\varepsilon)$$

$L(r)$ = { 没有两个连续0的字符串}

等价正则表示

定义:

对于正则表示 r_1 和 r_2，如果有

$$L(r_1)=L(r_2)$$

则称 r_1 和 r_2 是等价的，记为：

$$r_1=r_2$$

例

L = { 所有没有两个连续0的字符串}

$$r_1 = (1+01)^*(0+\varepsilon)$$

$$r_2 = (1^*011^*)^*(0+\varepsilon)+1^*(0+\varepsilon)$$

$L(r_1) = L(r_2) = L$ ⇒ $r_1 = r_2$

即r_1和 r_2是等价的正则表示

例

设计语言：

L={由 0，1交替构成的字符串}

的正则表示

- (01)* + (10)* + 0(10)* + 1(01)*
- (ε + 1) (01)* (ε + 0)
- (ε + 0) (10)* (ε + 1)

正 则 表 示

- 单一终结状态的 NFA
- 正则语言的运算性质
- 正则表示和语言
- 正则表示和正则语言
- 正则语言的同态
- 正则表示的代数定律

正则表示的语言

定理：正则表示的语言是正则语言.

{正则表示的语言} = { 正则语言 }

正则表示的语言

定理的第一部分：

{正则表示的语言} ⊆ { 正则语言 }

任何正则表示 r，它描述的语言$L(r)$是正则的

正则表示的语言

定理的第二部分：

{正则表示的语言} ⊇ { 正则语言 }

任何正则语言L，都存在正则表示 r，$L(r) = L$

定理证明

1. 任何正则表示 r，它的语言 $L(r)$ 是正则的

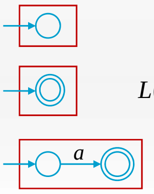
证明：通过对r的长度作归纳

基本正则表示

$$\varnothing, \quad \varepsilon, \quad a$$

定理证明

$\varnothing$, ε, a NFAs



$$\left.\begin{array}{l} L(M_1)=\varnothing=L(\varnothing) \\ L(M_2)=\{\varepsilon\}=L(\varepsilon) \\ L(M_3)=\{a\}=L(a) \end{array}\right\} \text{正则表示}$$

定理证明

假设正则表示 r_1 和 r_2 的语言 $L(r_1)$ 和 $L(r_2)$ 是正则语言

将证明

$$\left.\begin{array}{l} L(r_1+r_2) \\ L(r_1\cdot r_2) \\ L(r_1^*) \\ L((r_1)) \end{array}\right\} \text{是正则表示}$$

定理证明

由正则表示的定义知：

$$L(r_1+r_2)=L(r_1)\cup L(r_2)$$
$$L(r_1\cdot r_2)=L(r_1)L(r_2)$$
$$L(r_1^*)=(L(r_1))^*$$
$$L((r_1))=L(r_1)$$

定理证明

根据假定，$L(r_1)$ 和 $L(r_2)$ 是正则语言。因为：正则语言对如下运算是封闭的

$$L(r_1)\cup L(r_2)$$
$$L(r_1)L(r_2)$$
$$(L(r_1))^*$$

定理证明

因此:

$$\left.\begin{array}{l} L(r_1+r_2)=L(r_1)\cup L(r_2) \\ L(r_1\cdot r_2)=L(r_1)L(r_2) \\ L(r_1{}^*)=(L(r_1))^* \end{array}\right\} \text{是正则语言}$$

显然 $L((r_1))$ 是正则语言

由此得到正则表示的语言是正则语言

定理证明

2. 任何正则语言L，都存在正则表示r，满足

$$L(r) = L$$

证明：通过自动机构造正则表示来证明。
有两种构造方法

(1) 路径叠代法
(2) 状态消去法

路径迭代法

思想: 设正则语言对应的DFA A

(1) 将 A 的状态集用{1, 2, … , n}表示，且初态为1;

(2) $R_{ij}^{(k)}$ 为表示如下语言的正则表示：

$w \in L(R_{ij}^{(k)})$ 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，所有其它状态的编号不大于 k.

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

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

(3) 通过(2)的迭代过程，最终可计算出

$R_{ij}^{(n)}$, (i, j = 1, 2, … , n);

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

计算 $R_{ij}^{(k)}$ 的叠代过程

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 $a_1, a_2, \ldots, a_k$，则 $R_{ij}^{(0)} = a_1 + a_2 + \ldots + a_k$；

计算 $R_{ij}^{(k)}$ 的叠代过程

基础: $k = 0$

Case 2 $i=j$

若不存在 i 的自环，则 $R_{ij}^{(0)} = \varepsilon$；

若仅存在一个 i 的自环，且标记为 a，则 $R_{ij}^{(0)} = \varepsilon + a$;

若存在多个 i 的自环，且标记为 $a_1, a_2, \ldots, a_k$，则

$$R_{ij}^{(0)} = \varepsilon + a_1 + a_2 + \ldots + a_k$$

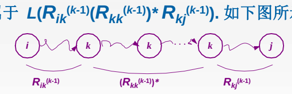
计算 $R_{ij}^{(k)}$ 的叠代过程

归纳：假设 $R_{ij}^{(k-1)}$ ($i, j = 1, 2, \ldots, n$) 为得到的正则表示.

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)

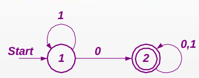
Case 1 路径不经过 k. 则标记该路径的字符串属于 $L(R_{ij}^{(k-1)})$;

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



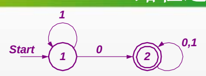
则叠代公式为 $R_{ij}^{(k)} = R_{ij}^{(k-1)} + R_{ik}^{(k-1)} (R_{kk}^{(k-1)})^* R_{kj}^{(k-1)}$

路径迭代法举例



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

路径迭代法举例

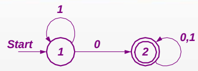


$R_{11}^{(0)}$	$\varepsilon + 1$
$R_{12}^{(0)}$	0
$R_{21}^{(0)}$	$\varnothing$
$R_{22}^{(0)}$	$\varepsilon + 0 + 1$

	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
$R_{21}^{(1)}$	$\varnothing+\varnothing(\varepsilon+1)^*(\varepsilon+1)$	$\varnothing$
$R_{22}^{(1)}$	$\varepsilon+0+1+\varnothing(\varepsilon+1)^*0$	$\varepsilon+0+1$

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

路径迭代法举例

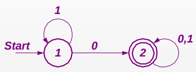


$R_{11}^{(1)}$	1^*
$R_{12}^{(1)}$	1^*0
$R_{21}^{(1)}$	$\varnothing$
$R_{22}^{(1)}$	$\varepsilon + 0 + 1$

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

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

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*** 等价的正则表示为

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

状态消去法

思想: 设正则语言对应NFA *M*

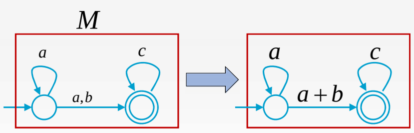
(1) 扩展自动机*M*，将正则表示作为转移的输入.

(2) 消去某一中间状态:

- 与其相关的转移弧同时消去;
- 通过修改每一个前趋状态及后继状态的转移弧标记，弥补消去的转移弧.

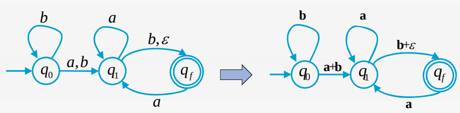
状态消去法

扩展的自动机 *M*, 其迁移标记都是正则表示。例：

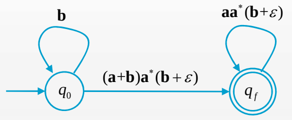


状态消去法

例：



状态消去法

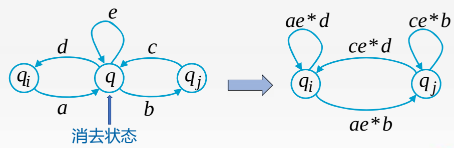


$$\mathbf{b}^*(\mathbf{a}+\mathbf{b})\mathbf{a}^*(\mathbf{b}+\varepsilon)\left(\mathbf{a}\mathbf{a}^*(\mathbf{b}+\varepsilon)\right)^*$$

$$L(r)=L(M)=L$$

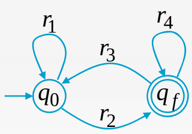
状态消去法

一般



状态消去法

最终的自动机



正则表示为：

$$r=r_1^*r_2\left(r_4+r_3r_1^*r_2\right)^*$$

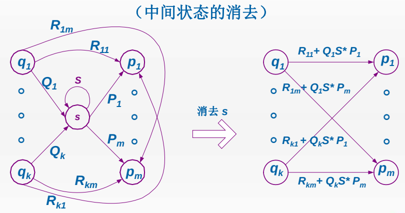
或者

$$r=\left(r_1+r_2r_4^*r_3\right)^*r_2r_4^*$$

$$L(r)=L(M)=L$$

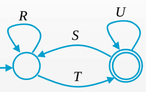
状态消去法

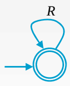
（中间状态的消去）



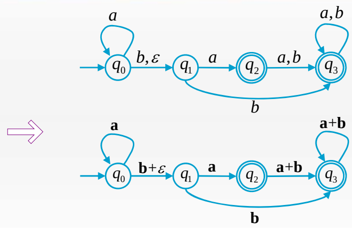
状态消去法步骤

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$q\neq q_0$, 得到如右上图自动机，对应的正则表示为 $R^*S(U+TR^*S)^*$ 或 $(R+SU^*T)^*SU^*$.
3. 若$q=q_0$, 得到如右下图的自动机，对应的正则表示为 R^*.
4. 最终的正则表示为每一终态对应的正则表示之和（并）.

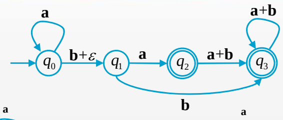


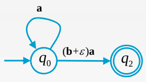


状态消去法举例

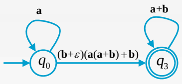


状态消去法举例



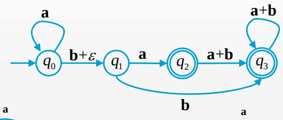


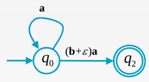
对于终态q_2



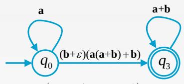
对于终态q_3

状态消去法举例





$r_1 = \mathbf{a}^*(\mathbf{b}+\varepsilon)\mathbf{a}$



$r_2 = \mathbf{a}^*\left((\mathbf{b}+\varepsilon)\left(\mathbf{a}(\mathbf{a}+\mathbf{b})+\mathbf{b}\right)\right)(\mathbf{a}+\mathbf{b})^*$

$$r = r_1 + r_2 = \mathbf{a}^*(\mathbf{b}+\varepsilon)\mathbf{a} + \mathbf{a}^*\left((\mathbf{b}+\varepsilon)\left(\mathbf{a}(\mathbf{a}+\mathbf{b})+\mathbf{b}\right)\right)(\mathbf{a}+\mathbf{b})^*$$

正 则 表 示

- 单一终结状态的 NFA
- 正则语言的运算性质
- 正则表示和语言
- 正则表示和正则语言
- 正则语言的同态
- 正则表示的代数定律

正则语言的同态

- 设映射 $h:\Sigma\to T^*$，对 $w=a_1a_2\ldots a_n\in\Sigma^*$，定义
  $h(w) = h(a_1)\, h(a_2) \ldots h(a_n)$
  则映射h为Σ一个同态(*homomorphism*)
- 对语言 $L\subseteq\Sigma^*$，定义 L 的同态
  $h(L) = \{\, h(w) \mid w\in L \,\}$

正则语言的同态

例:

设 $h(0)=ab$, $h(1)=\varepsilon$，则

$h(0101) = h(0)\, h(1)\, h(0)\, h(1) = abab$

设 $L = \{\, 0^k1^k \mid k \geq 0 \,\}$，则

$h(L) = \{h(0^k1^k) \mid k\geq 0\} = \{\, (ab)^k \mid k \geq 0 \,\}$
$= L((ab)^*)$

正则语言的同态

- 定理：若 L 为正则语言，$h:\Sigma\to T^*$, 则 $h(L)$ 也是正则语言。
- 证明：设 L 对应的正则表示为E，使得 $L(E)=L$.
  (1) $h(E)$为正则表示；
  (1) $L(h(E)) = h(L(E)) = h(L)$.

$h(E)$表示将Σ中符号a用$h(a)$代替后所得到的正则表示

正则语言的同态

(1) h(E)为正则表示；
(2) L(h(E)) = h(L(E)) = h(L).

对 E 的结构作归纳。

- 基础：若 E 为ε, $\varnothing$，有 $h(E) = E$, 则为正则表示，且显然 $L(h(E)) = h(L(E))$;

  若 E 为a，有 $h(E) = h(a)$, 则为正则表示， 且 $L(h(E)) = h(L(E)) = \{h(a)\}$；

正则语言的同态

(1) h(E)为正则表示；
(2) L(h(E)) = h(L(E)) = h(L).

- 归纳：

  若 $E=E_1E_2$，$h(E_1)$, $h(E_2)$ 为正则表示，且

  $$L(h(E_1)) = h(L(E_1))$$
  $$L(h(E_2)) = h(L(E_2))$$

  由定义

  $$h(E) = h(E_1)\ h(E_2)$$

正则语言的同态

(1) h(E)为正则表示；
(2) L(h(E)) = h(L(E)) = h(L).

$h(E)$为正则表示，且

$$L(h(E)) = L(h(E_1)\ h(E_2))$$
$$= L(h(E_1))\ L(h(E_2)) = h(L(E_1))\ h(L(E_2))$$
$$= h(L(E_1)L(E_2)) = h(L(E_1E_2)) = h(L(E))$$

- 类似证明$E=E_1+E_2$和 $E=E_1^*$ 的情形.

正则语言的逆同态

- 设同态映射 $h: \Sigma\to T^*$, 对语言 $L\subseteq T^*$, 定义 L 的逆同态(*inverse homomorphism*)

  $$h^{-1}(L) = \{ w \mid w\in \Sigma^* \wedge h(w) \in L \}$$

- 定理：
  若 $L\subseteq T^*$ 为正则语言，$h: \Sigma\to T^*$为同态映射, 则 $h^{-1}(L)$ 也是正则语言。

正则语言的逆同态

证明 设 $L = L(A)$，其中 DFA $A = (Q, T, \delta, q_0, F)$.

构造 DFA $B = (Q, \Sigma, \gamma, q_0, F)$, 其中

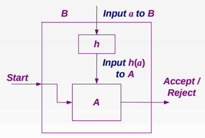
$\gamma(q, a) = \delta^*(q, h(a))$.

可证对任何w, 有

$\gamma(q_0, w) = \delta^*(q_0, h(w))$.

即

$h^{-1}(L) = L(B)$. (对 $|w|$ 归纳)



正 则 表 示

- 单一终结状态的 NFA
- 正则语言的运算性质
- 正则表示和语言
- 正则表示和正则语言
- 正则语言的同态
- 正则表示的代数定律

正则表示的代数律

- 交换律和结合律
  - $L+M = M+L$
  - $(L+M)+N = L+(M+N)$
  - $(LM)N = L(MN)$
- 单位元和零元
  - $\emptyset + L = L + \emptyset = L$
  - $\varepsilon L = L\varepsilon = L$
  - $\emptyset L = L\emptyset = \emptyset$

正则表示的代数律

- 分配律
  - $L(M+N) = LM+LN$
  - $(M+N)L = ML+NL$
- 等幂律
  - $L + L = L$

正则表示的代数律

- 闭包相关的定律
  - $(L^*)^* = L^*$
  - $\emptyset^* = \varepsilon$
  - $\varepsilon^* = \varepsilon$
  - $L^+ = LL^* = L^*L$　(L^+ 的定义)
  - $L^* = L^+ + \varepsilon$
  - $L? = \varepsilon + L$　($L?$的定义)

课后练习

✧ 必做题:
- P-89 *Ex.3.1.1* (*b*),(*c*)
- P-89 *Ex.3.1.2* (*b*)
- P-90 *Ex.3.1.3* (*a*)
- P-106 *Ex. 3.2.1* (*c*),(*d*)
- P-106 *Ex.3.2.3*
- P-121 *Ex.3.4.2* (*b*), (*d*)

✧ 思考题:
- P-90 *Ex.3.1.3* (*b*),(*c*)
- P-90 *Ex.3.1.5*
- P-107 *Ex.3.2.6*

That's all for today.

Thank You