讲稿（教学内容、步骤）

第 3 章 词法分析（2）

1. NFA 到 DFA 的等价变换（子集法）

(1) 基本运算：

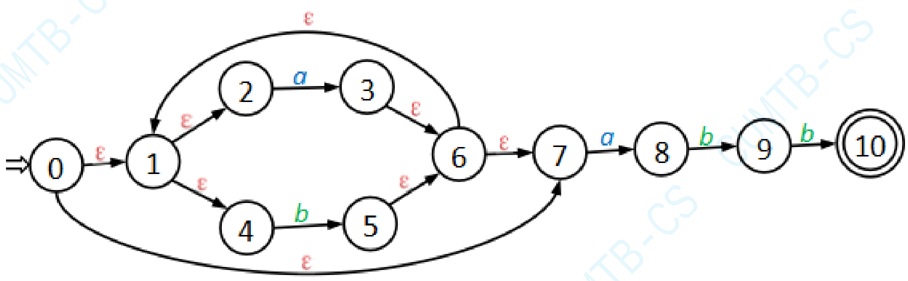
- 状态集合 I 的ε闭包：表示为ε-closure(I)

  状态集 I 中的任何状态 S 经任意条ε弧而能到达的状态的集合。

  注：状态集 I 的任何状态 S 都属于ε-closure(I)
- 状态集合 I 的 a 弧转换：表示为 move(I,a)

  定义为状态集合 J，其中 J 是所有那些可从 I 的某一状态经过一条 a 弧而到达的状态的全体。

  【举例】



则

ε -closure(0)={ 0, 1, 2, 4, 7 }	move(0,a)= Φ
ε -closure(1)={ 1, 2, 4 }	move(0,b)= Φ
ε -closure(3)={ 3, 6, 7, 1, 2, 4 }	move(1,a)= Φ
ε -closure({1,3})={ 1, 2, 3, 4, 6, 7 }	move(1,b)= Φ
	move({1, 2, 4},a)={ 3 }
	move({1, 2, 4},b)={ 5 }

(2) 转换的主要思想

DFA 的一个状态可能对应 NFA 的一个或一组状态

DFA 同样记录 在 NFA 上读入某个终结符后可能到达的所有状态

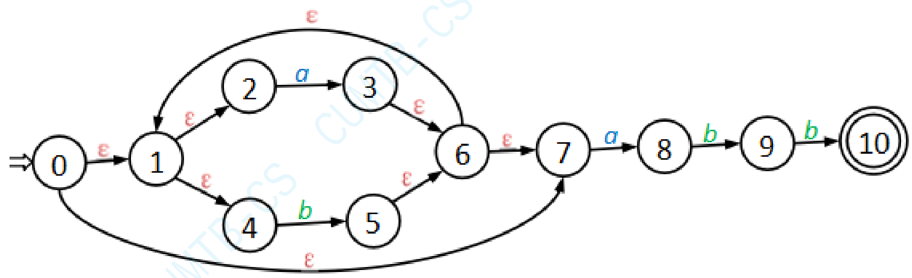
(3) 子集法构造方法

构造方法	对应算法

NFA M'=(K',Σ',f',S',Z')构造
DFA M =(K, Σ, f, S, Z)，使得 L(M')=L(M)：
1. M的状态集K由K'的一些子集组成
2. M的输入字母表Σ = Σ'
3. M的转换函数f定义为：
$\boxed{f([K_1,...,K_j],a) = \varepsilon\text{-closure}(move([K_1,...,K_j],a))}$
4. M的初态S = ε-closure(S')
5. M的终态集Z = {[K_i,K_k,...,K_e]}，
其中[K_i,K_k,...,K_e]∈K且{K_i,K_k,...K_e}∩Z'≠Φ}
	假定所构造的子集族 C=(T_0, T_1,... T_i)，其中T_0, T_1,... T_i为状态 K 的子集。
	
	令ε-closure(K_0)为C中唯一成员，并且它是未被标记的。
	
	While(C中存在尚未被标记的子集T) do
	{ 标记T;
	for(每个输入字母a) do
	{ U:= ε-closure(move(T,a));
	if (U不在C中) then
	{将U作为未标记的子集加入C }
	}
	}

【举例】

对如下 NFA 构造其等价的 DFA



解：子集法求等价 DFA

C(T0, T1, T2, T3, T4)

初态 T0 = ε-closure(0)={0, 1, 2, 4, 7}

T1 = ε-closure(move(T0, a))

= ε-closure({3, 8})={1, 2, 3, 4, 6, 7, 8}

T2 = ε-closure(move(T0, b))

= ε-closure({5})={1, 2, 4, 5, 6, 7}

ε-closure(move(T1, a))

= ε-closure({3, 8})= T1

T3 = ε-closure(move(T1, b))

= ε-closure({5, 9})={1, 2, 4, 5, 6, 7, 9}

ε-closure(move(T2, a))

= ε-closure({3, 8})= T1

ε-closure(move(T2, b))

= ε-closure({5})= T2

ε-closure(move(T3, a))

= ε-closure({3, 8})= T1

终态 T4 = ε-closure(move(T3, b))

= ε-closure({5, 10})={1, 2, 4, 5, 6, 7, 10}

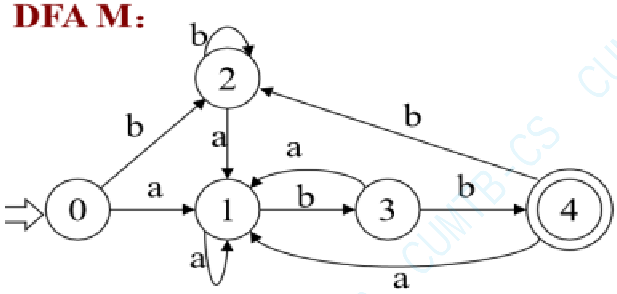
ε-closure(move(T4, a))

= ε-closure({3, 8})= T1

ε-closure(move(T4, b))

= ε-closure({5})= T2

DFA M:



【练习】课后习题 2，4(a) （其中最小化部分等最小化讲完再做）

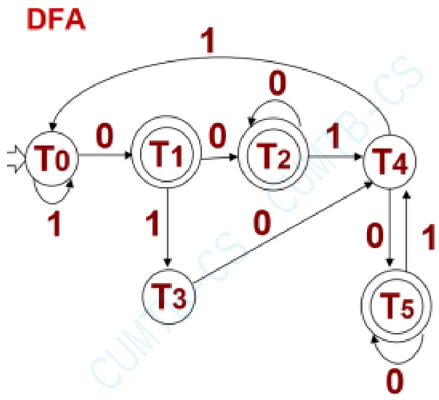
2. 已知 NFA=({x,y,z}, {0,1}, M, {x}, {z})，其中：

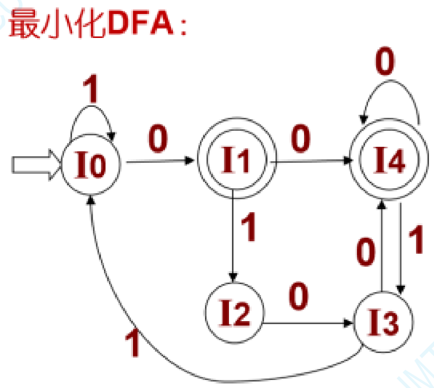
M(x, 0)={z} M(y,0)={x, y} M(z, 0)={x, z}

M(x, 1)={x} M(y,1)=Φ M(z,1)={y}

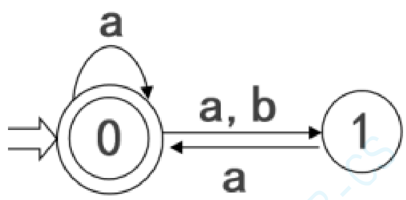
构造相应的 DFA。

解：



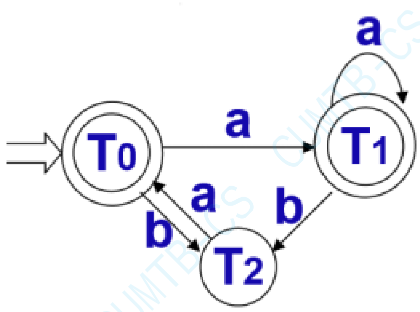


4. 图(a)确定化和最小化

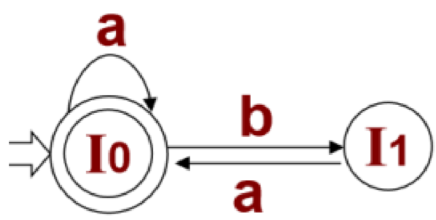


解：

DFA：



最小化 DFA：



2. DFA 的化简（最小化 DFA）

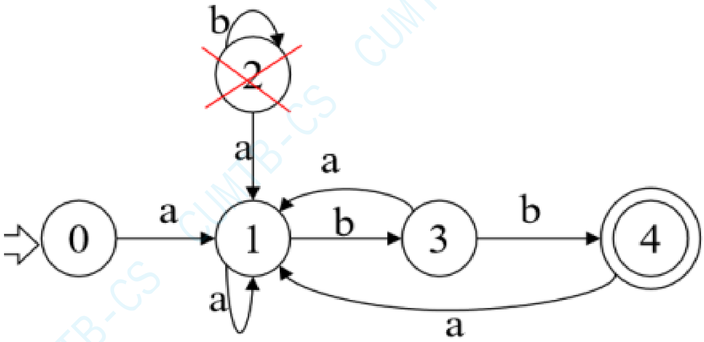
(1) 最小化 DFA

- 没有多余状态
- 没有等价状态

(2) 多余状态：从开始状态出发，任何输入串也不能到达的状态

处理：消除多余状态，并删除与多余状态相连的边

【举例】删除下图 DFA 中的多余状态

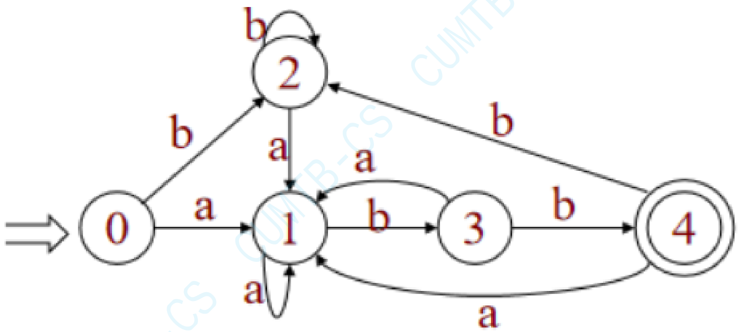


(3) 等价状态：若两个状态 s 和 t 等价，则同时满足

- 一致性——同是终态 或 同是非终态
- 蔓延性——从 s 出发读入某个 a a   和从 t 出发读入某个 a 到达的状态等价。

处理：合并等价状态（使用"分割法"）

【举例】将下图 DFA 最小化



DFA M

解：

步骤一：消除多余状态（无多余状态，不需消除）

步骤二：使用分割法，合并等价状态。

非终态集　　终态集

{ 0 , 1 , 2 , 3 }　　{ 4 }

Ia　1　1　1　1　过a弧，下一个状态在同一集合里，无需分割

Ib　2　3　2　4　过b弧，下一个状态在不同集合里，需分割

分割后：{ 0 , 1 , 2 }　{ 3 }　{ 4 }

Ia　1　1　1　过a弧，下一个状态在同一集合里，无需分割

Ib　2　3　2　过b弧，下一个状态在不同集合里，需分割

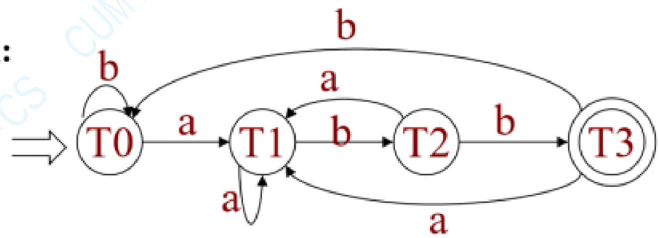
分割后：{ 0 , 2 }　{ 1 }　{ 3 }　{ 4 }

Ia　1　1　过a弧，下一个状态在同一集合里，无需分割

Ib　2　2　过b弧，下一个状态在同一集合里，无需分割

最终合并结果是	{0,2}	{1}	{3}	{4}
重命名为:	T0	T1	T2	T3
Ia:	T1	T1	T1	T1
Ib:	T0	T2	T3	T0

最小化DFA:



3. 正规式和有穷自动机的等价转换

- 对于∑上的 NFA M，可以构造一个∑上的正规式 R，使得 L(R)=L(M)。
- 对于∑上的正规式 R，可以构造一个∑上的 NFA M，使得 L(M)=L(R)。

4. 有穷自动机转换为等价的正规式

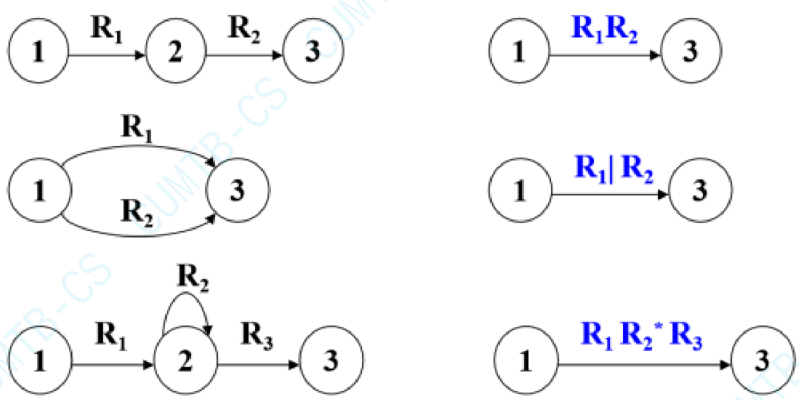
(1) 转换方法

① 在 FA M 的状态图上加两个状态结点 x 和 y。

- 从 x 结点出发，用ε弧连接 x 结点 到 所有初态结点
- 从 M 的 所有终态结点 用ε弧连接到 y 结点

此时， x 为初态和 y 为终态。

② 利用消结规则，逐步消去 M′中的所有结点，直至只剩下 x 和 y。



③ 最后 x 和 y 结点间的弧上的标记则为所求的正规式 R。

【举例】将图 1 的 FA 转换成等价的正规式

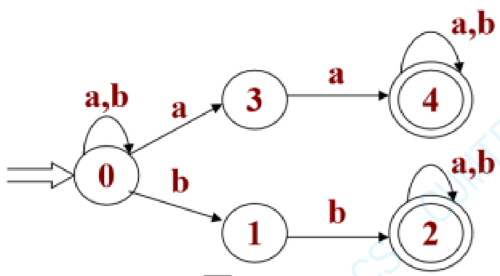


图1

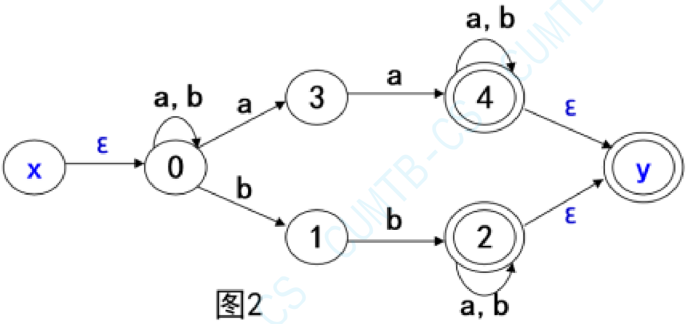

x
ε
0
a, b
a
b
3
1
a
b
4
2
a, b
a, b
ε
ε
y

图2

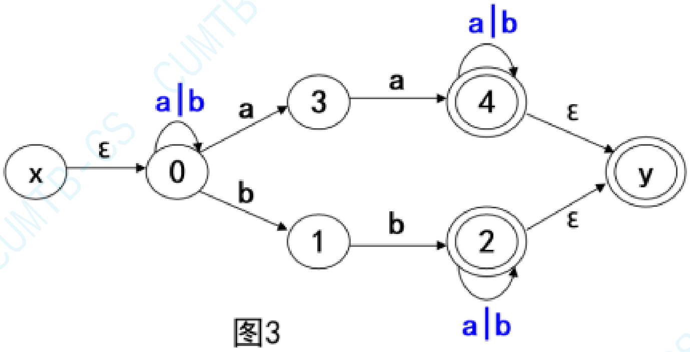

x
ε
0
a|b
a
b
3
1
a
b
4
2
a|b
a|b
ε
ε
y

图3

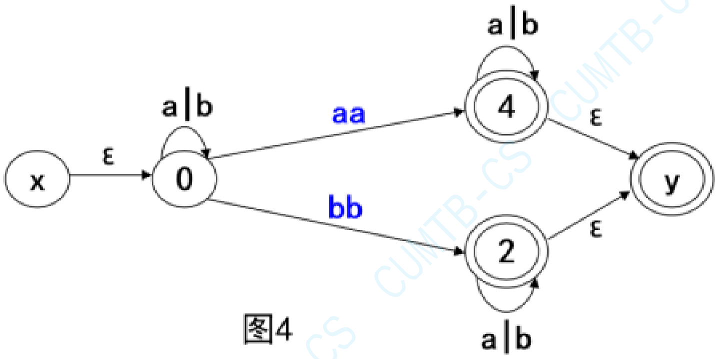

x
ε
0
a|b
aa
bb
4
2
a|b
a|b
ε
ε
y

图4

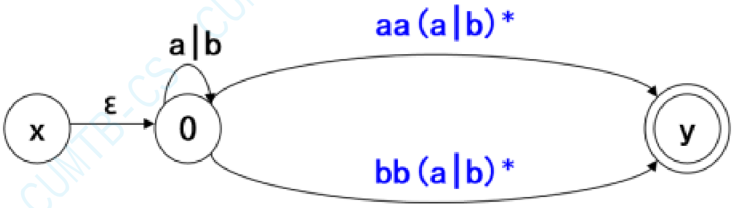

x
ε
0
a|b
aa(a|b)*
bb(a|b)*
y

图5

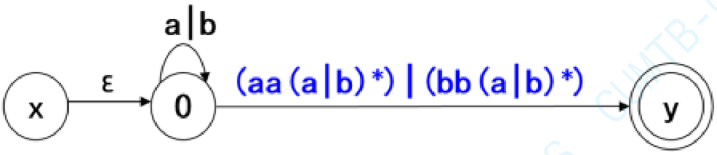

x
ε
0
a|b
(aa(a|b)*)|(bb(a|b)*)
y

图6

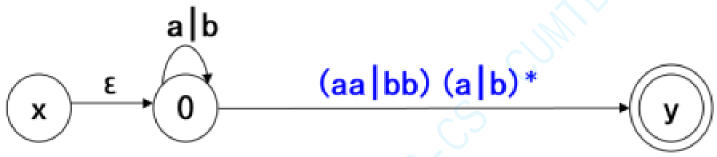


图7

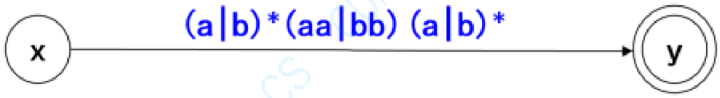
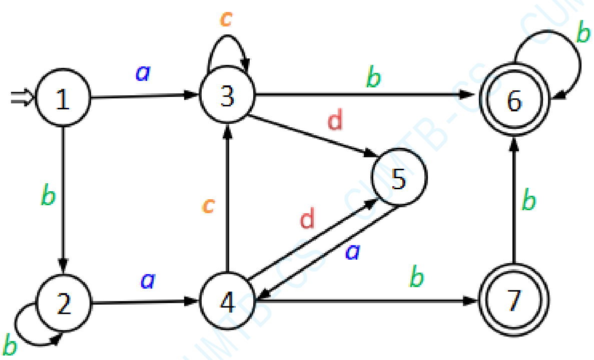


图8

所求正规式 R 为 (a|b)* (aa|bb)(a|b)*

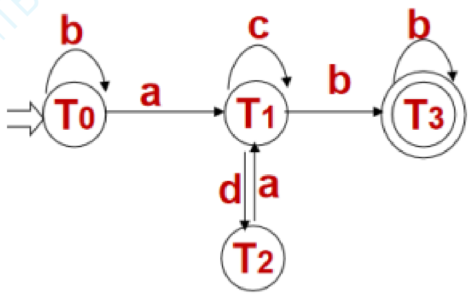
【练习】课后习题 9

9. 将 DFA 最小化，并用正规式描述它所识别的语言。



解：

最小化 DFA：



正规式：　　b*a(da|c)*bb*