第二章：词法分析

NFA与自动机的最小化

内容介绍

- 非确定有限自动机（NFA）的定义
- NFA到DFA的转换
- 自动机的最小化
- 自动机的化简

1.1 非确定有限自动机的定义

- 非确定有限自动机NFA为一个五元组 (Σ, S, S_0, f, Z)，其中：
  - Σ是一个有穷字母表，它的每个元素称为一个输入字符；
  - S是状态的集合，它的每个元素称为一个状态；
  - $S_0 \subseteq S$，是非确定有限自动机的初始状态集；
  - f是一个从$S \times (\Sigma \cup \{\varepsilon\})$到S 的子集的映射，即 $S \times (\Sigma \cup \{\varepsilon\}) \rightarrow 2^s$
  - $Z \subseteq S$，是一个终止状态集，又称为接受状态集

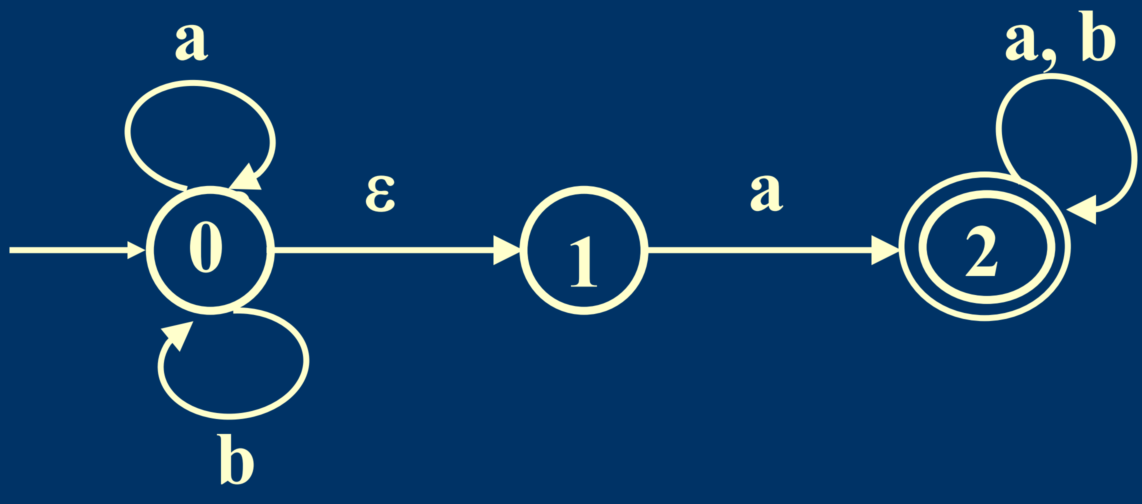
1.2 DFA和NFA的区别

- 总结来看有3点区别

1. 一个状态的不同输出边可以标有相同符号
2. 允许有多个开始状态
3. 允许有空边

1.3 NFA的一些问题

- NFA所能接受的串与DFA的定义是相同的



- 实现起来很困难

2.1 自动机等价

- 定义：设A_1和A_2是同一个字母表上的自动机，如果有$L(A_1)=L(A_2)$，则称A_1和A_2等价。

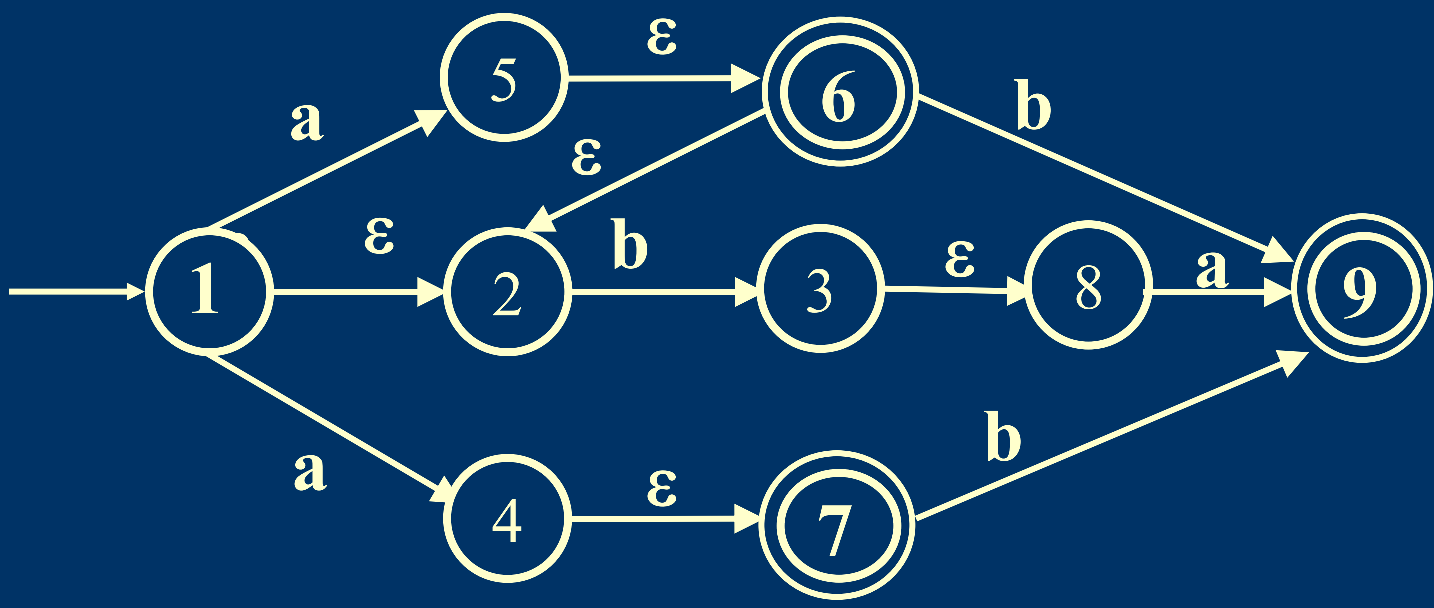
- 定理：对于每一个非确定自动机A，存在一个确定自动机A’，使得L(A)=L(A’).

2.2 NFA到DFA的转换

- 状态集I的ε闭包：设I是NFA M状态集的子集，定义I的ε闭包ε_CLOSURE(I)为：

1. 若q ∈I ，则q ∈ ε_CLOSURE(I).
2. 若q ∈I，那么从q出发经任意条ε弧而能到达的任何状态q’都属于 ε_CLOSURE(I).

2.2 NFA到DFA的转换



- 例: ε_CLOSURE({1})={1,2}

  ε_CLOSURE({1,5})={1,5,2,6}

2.2 NFA到DFA的转换

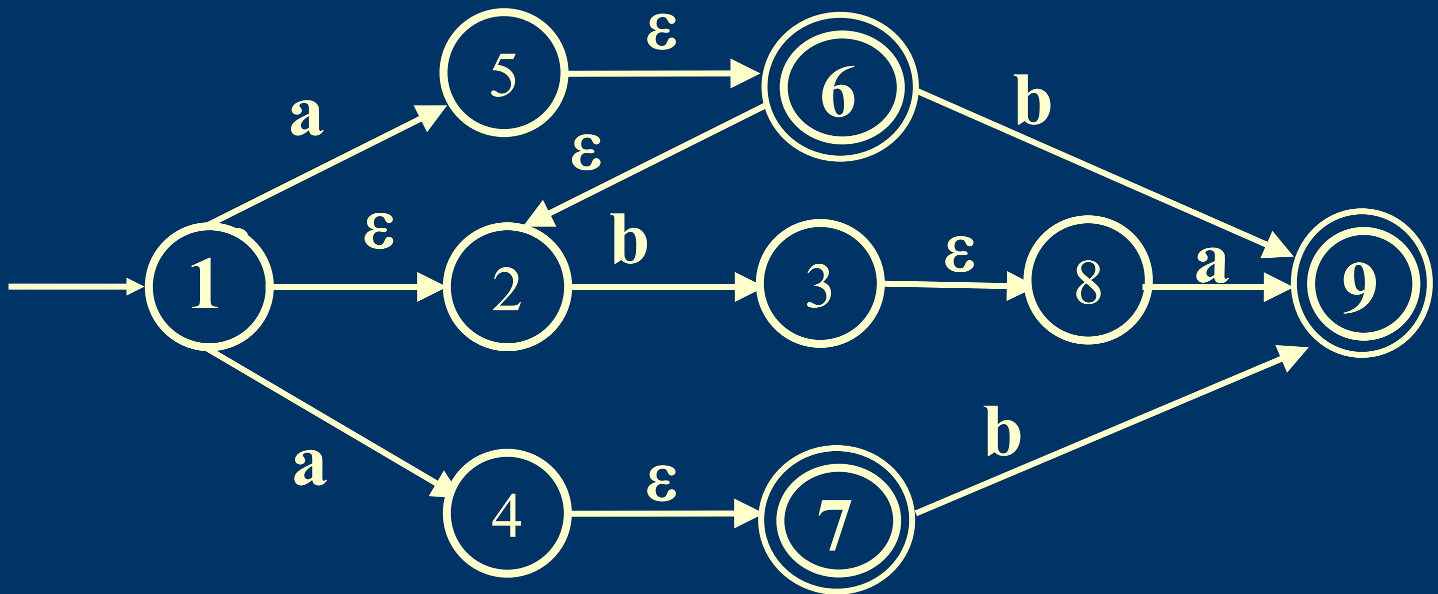
- 状态集I的a转换：若$I=\{S_1, \cdots, S_m\}$是NFA的状态集的一个子集，$a \in \Sigma$，则定义：

$$I_a = \varepsilon\_CLOSURE(J)$$

其中：

$$J = f(S_1, a) \cup f(S_2, a) \cdots \cup f(S_m, a)$$

2.2 NFA到DFA的转换



- 例: $\{1, 2\}_a = \varepsilon\_CLOSURE(J)$

  $J = f(1,a) \cup f(2,a) = \{4,5\}$

  $\{1, 2\}_a = \varepsilon\_CLOSURE(\{4,5\}) = \{4,5,7,6,2\}$

2.2 NFA到DFA的转换

- **NFA A'到DFA A的转换过程(确定化):**

1. 令I_0= ε_CLOSURE(S_0)作为DFA的初始状态，其中S_0为NFA初始状态集.

2. 若DFA中的每个状态都经过本步骤处理过.则转步骤3；否则任选一个未经本步骤处理的DFA状态S_i，对每一个$a\in\Sigma$,进行下述处理：

   ①计算 $S_j = S_{ia}$

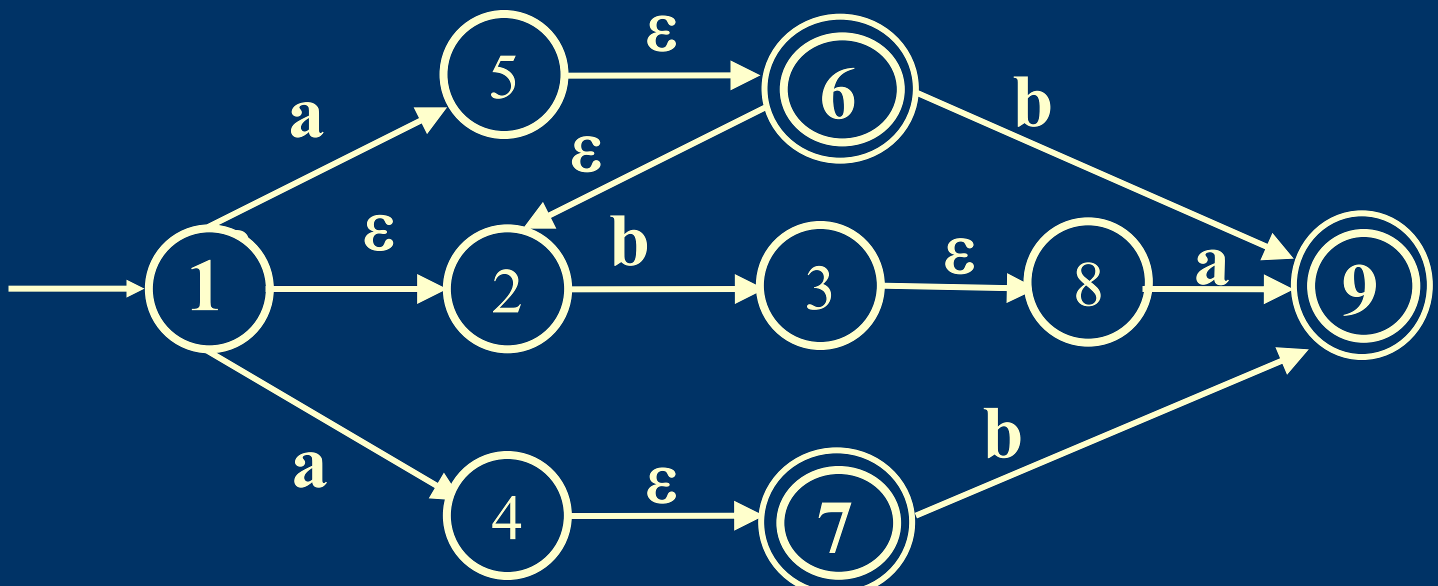
   ② 若$S_j \neq \Phi$ ，则令$f(S_i, a) = S_j$ ，

   若S_j不为当前DFA的状态，则将其作为DFA的一个状态.

   转步骤2.

3. 若S' =[S_1, ..., S_n]是A的一个状态,且存在一个S_i是A'的终止状态，则令S'为A的终止状态.

2.2 NFA到DFA的转换



2.2 NFA到DFA的转换

状态 \ 输入字	a	b
+{1, 2}	{2, 4, 5, 6, 7}	{3, 8}
−{2, 4, 5, 6, 7}	{}	{3, 8, 9}
{3, 8}	{9}	{}
−{3, 8, 9}	{9}	{}
−{9}	{}	{}

2.2 NFA到DFA的转换

- 转化后的结果

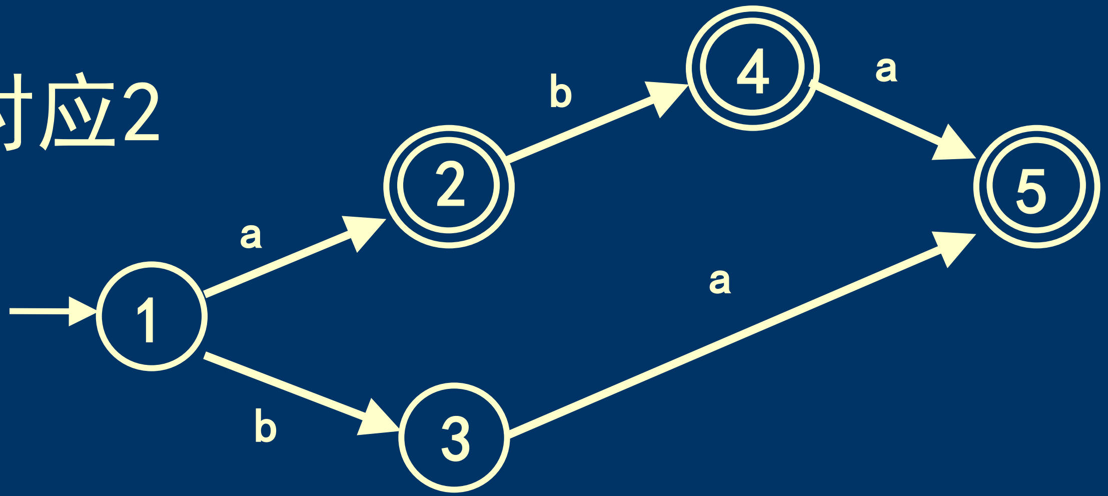
{1, 2}对应1

{2, 4, 5, 6, 7}对应2

{3, 8}对应3

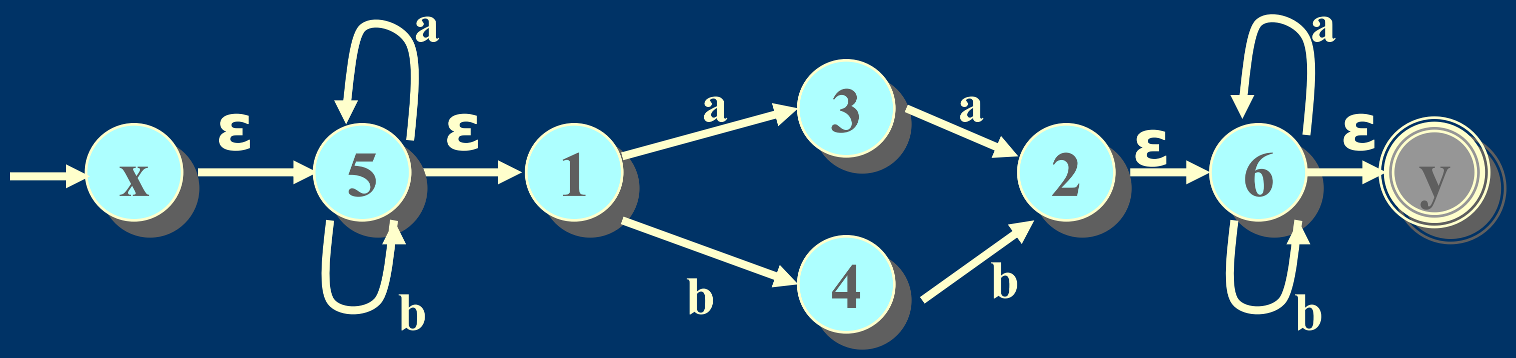
{3, 8, 9}对应4

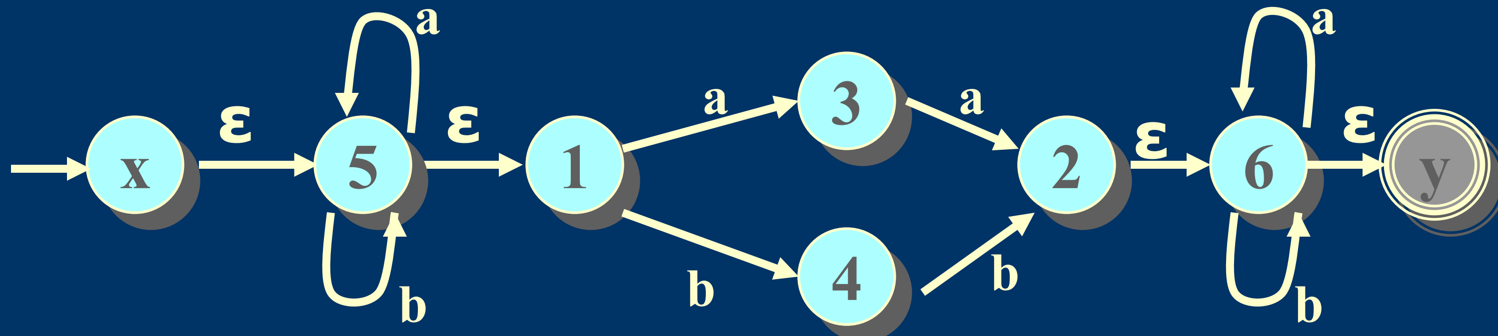
{9}对应5



2.2 NFA到DFA的转换

例:将如下的NFA转化为DFA





$\{5,1,4\}_a$

$\{5,1,4\}_b$
$=\{5,4,2,1,6,y\}$

$\{5,1,3,2,6,y\}_a$
$=\{5,3,2,6,1,y\}$

$\{5,1,3,2,6,y\}_b$
$=\{5,4,6,1,y\}$

$\{x,5,1\}_b$
$=\{5,4,1\}$

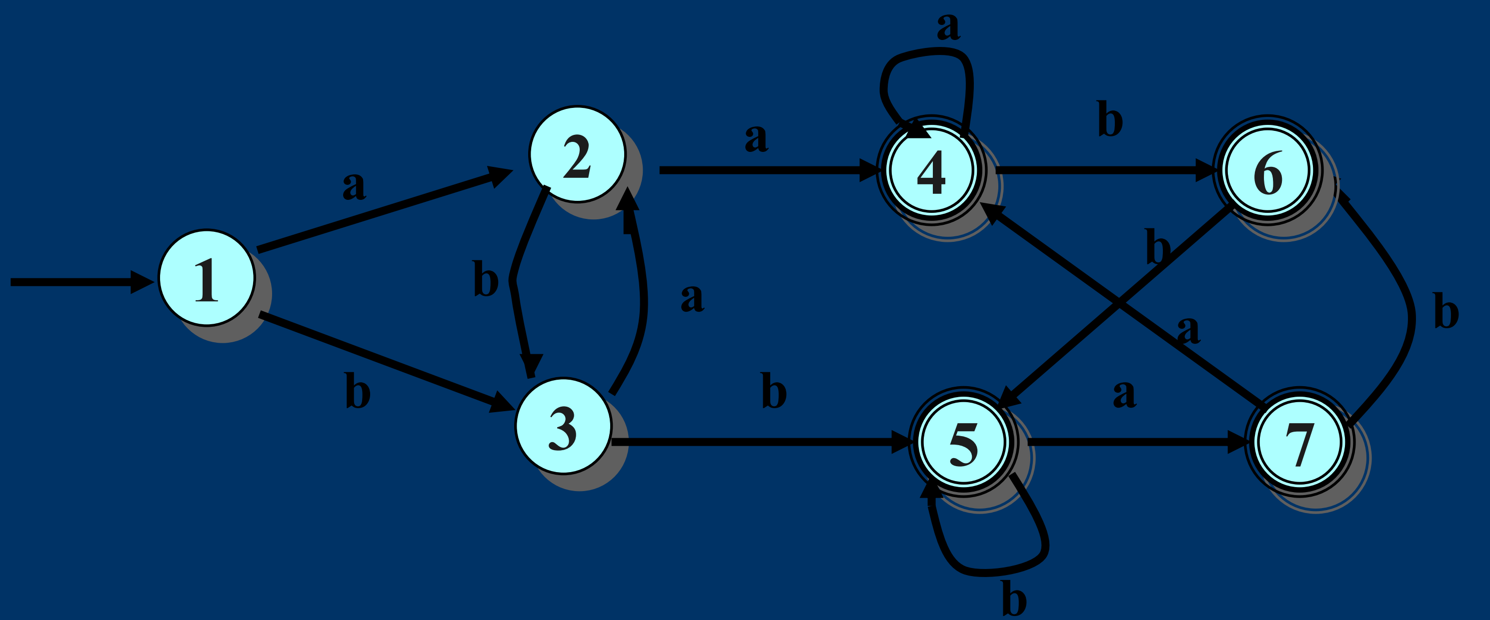
$\{5,1,3\}_a$

$\{5,1,3\}_b$
$=\{5,4,1\}$

I	a	b
{x,5,1}	{5,1,3}	{5,1,4}
{5,1,3}	{5,1,3,2,6,y}	{5,1,4}
{5,1,4}	{5,1,3}	{5,1,4,2,6,y}
{5,1,3,2,6,y}	{5,1,3,2,6,y}	{5,1,4, 6,y}
{5,1,4,2,6,y}	{5,1,3, 6,y}	{5,1,4,2,6,y}
{5,1,4, 6,y}	{5,1,3, 6,y}	{5,1,4,2,6,y}
{5,1,3, 6,y}	{5,1,3,2,6,y}	{5,1,4, 6,y}

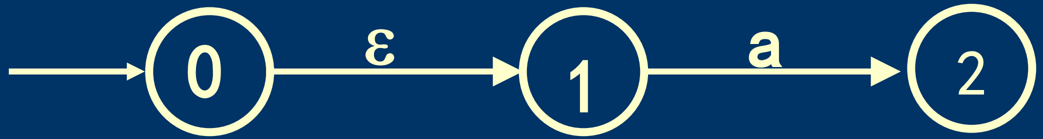
	a	b
{x,5,1} 1	{5,1,3} 2	{5,1,4} 3
{5,1,3} 2	{5,1,3,2,6,y} 4 *	{5,1,4} 3
{5,1,4} 3	{5,1,3} 2	{5,1,4,2,6,y} 5 *
{5,1,3,2,6,y} 4*	{5,1,3,2,6,y} 4	{5,1,4, 6,y} 6 *
{5,1,4,2,6,y} 5 *	{5,1,3, 6,y} 7 *	{5,1,4,2,6,y} 5 *
{5,1,4, 6,y} 6 *	{5,1,3, 6,y} 7 *	{5,1,4,2,6,y} 5 *
{5,1,3, 6,y} 7 *	{5,1,3,2,6,y} 4 *	{5,1,4, 6,y} 6 *

包含原终态的状态作为新的终态

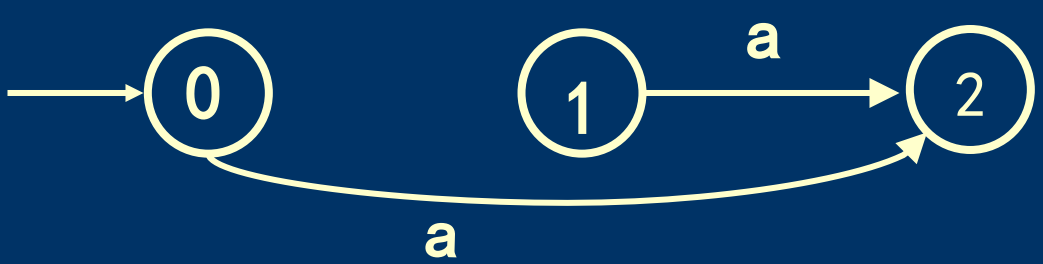


2.2 NFA到DFA的转换

- 另外一种消除空边的转换方式



- 删去空边，增加0到2的边

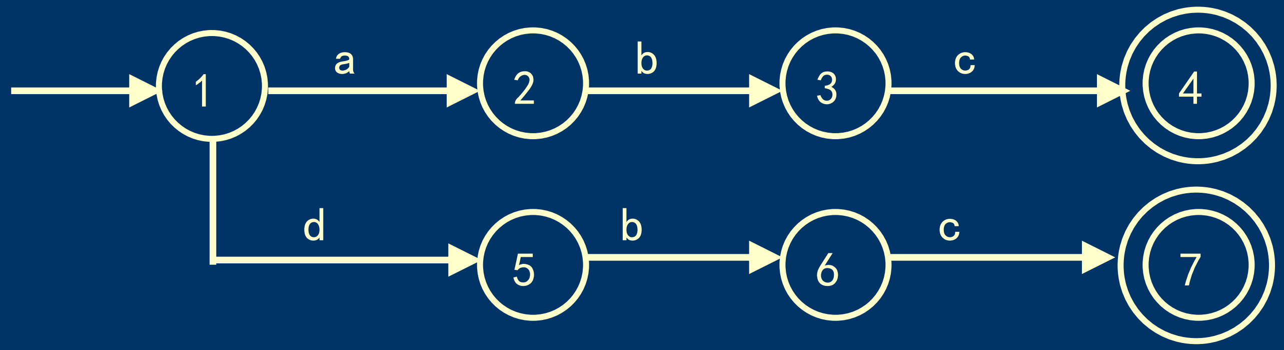


- ε 环路的时候，就把这几个状态合成一个

2.3 自动机的最小化

- 定义：　等价状态

　　设DFA M 的两个状态S_1和S_2，如果对任意输入的符号串x，从S_1和S_2出发，总是都到达接受状态或拒绝状态中，则称S_1和S_2是等价的。



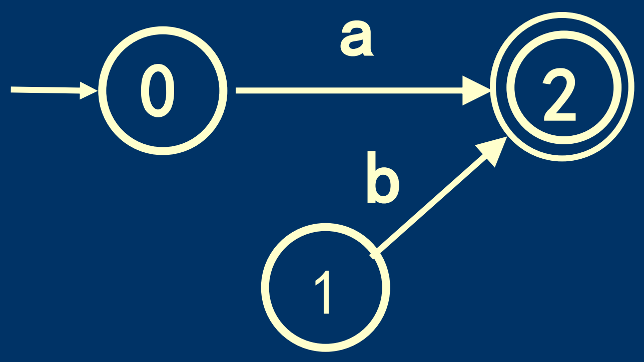
2.3 自动机的最小化

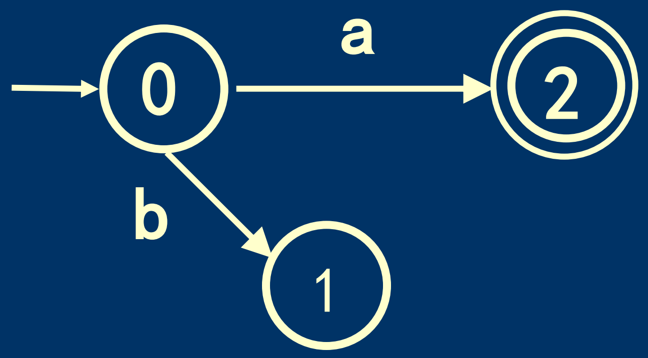
- 定义：无关状态

  设S是DFA M的一个状态，若：

1. 从开始状态无到S的通路，或
2. S到任意终止状态无通路

则称S为M的无关状态





2.3自动机的最小化

- 定义：最小(最简)自动机
  如果DFA M 没有无关状态，也没有等价状态，则称M 为最小自动机

- 结论：任一DFA都可以化为最简自动机，即任一DFA M都存在DFA M'，使得
  L(M)=L(M')，且M'是最简自动机

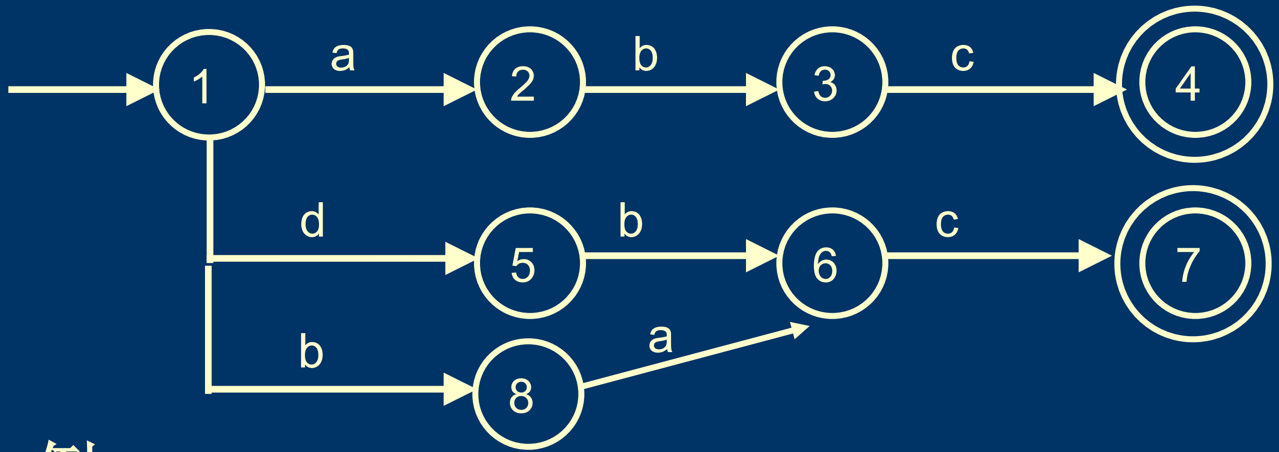
2.4自动机的化简

- **状态分离法**
- 状态分离法的目标:

$SS = SS_1 \cup SS_2 \cup \ldots\ldots \cup SS_n$ 其中: $SS_i \cap SS_j = \Phi$ ($i \neq j$时), 并且每个SS_i中的所有状态均等价.

- 状态集SS_i对a($a \in \Sigma$)是不可区分的:若SS_i中元素对输入符a都转到相同的状态集中.
- 状态集SS_i对a($a \in \Sigma$)是可区分的: 若SS_i中元素对输入符a转到不同的状态集中.
- 状态集SS_i对a($a \in \Sigma$)进行划分:若SS_i中元素对输入符a转到不同的状态集中,则分别将转到相同集合中的状态组成一个新的状态集.

2.4自动机的化简



例：

{1，8} U{ 2，5 } U{3，6} U{4，7}

{ 2，5 } 对b是不可区分的；

{ 3，6 } 对c是不可区分的；

{ 1，8 } 对**a**是可区分的；

{ 1，8 } 对**a**进行划分：**{ 1 } U{ 8 }**

2.4自动机的化简

状态分离法算法

$STEP_1$ 求初始划分:

SS=非终止状态集∪终止状态集.

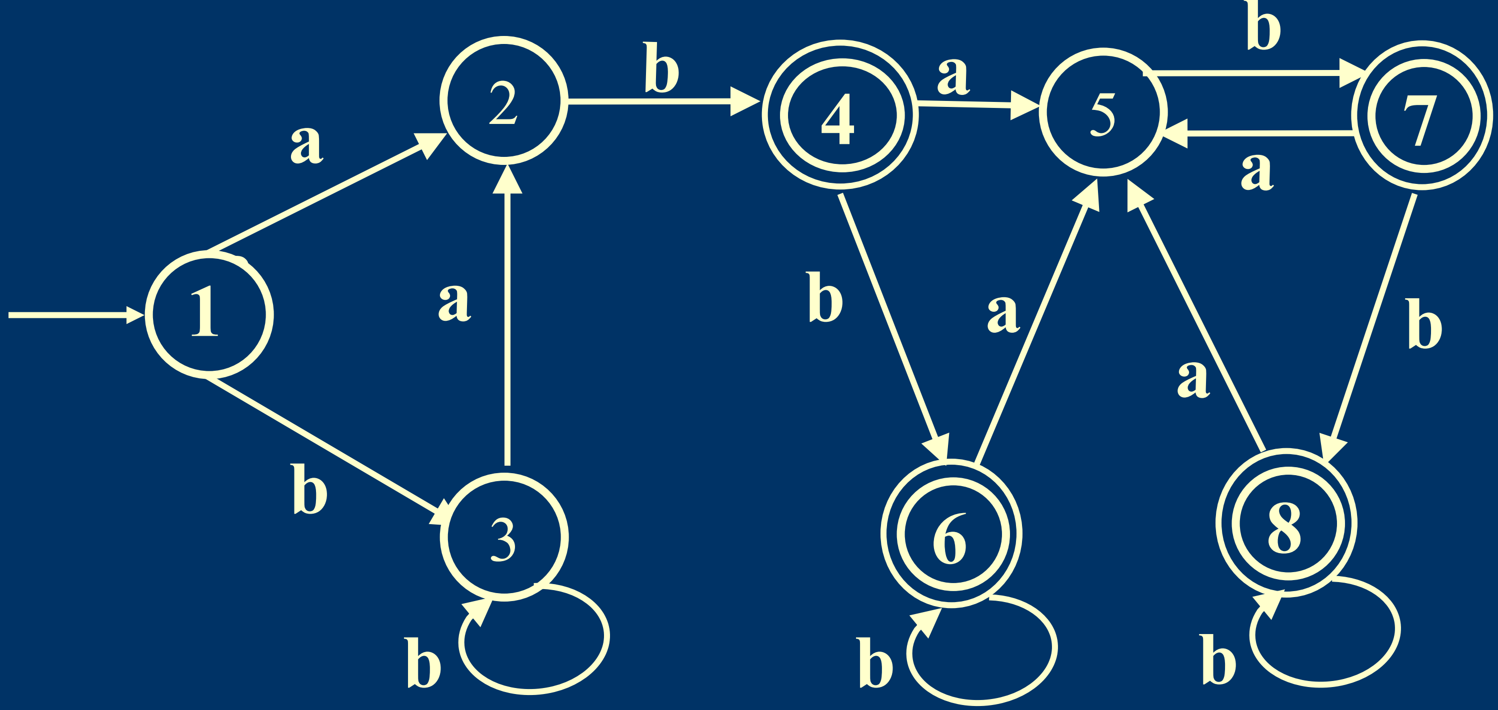
$STEP_2$ 若SS中的每个子集对每一个a(a∈Σ)都是不可区分的,则转 $STEP_3$;否则对可分的子集按相应的a(a∈Σ)进行划分.

转$STEP_2$.

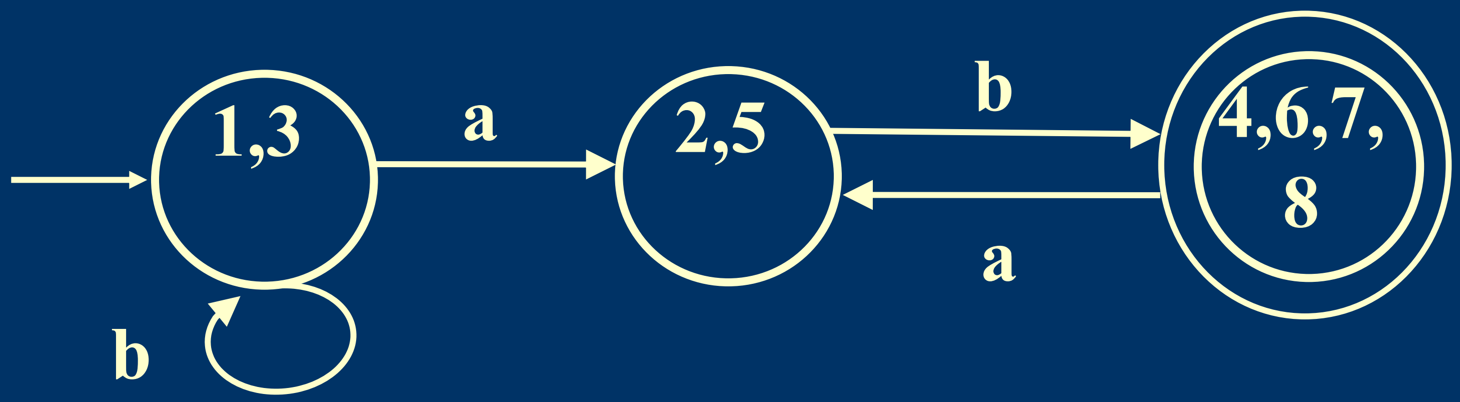
$STEP_3$ 每个子集中元素合并为一个状态，只含终态的集合为一个终态，对边作相应的调整.

2.4自动机的化简

- 例1

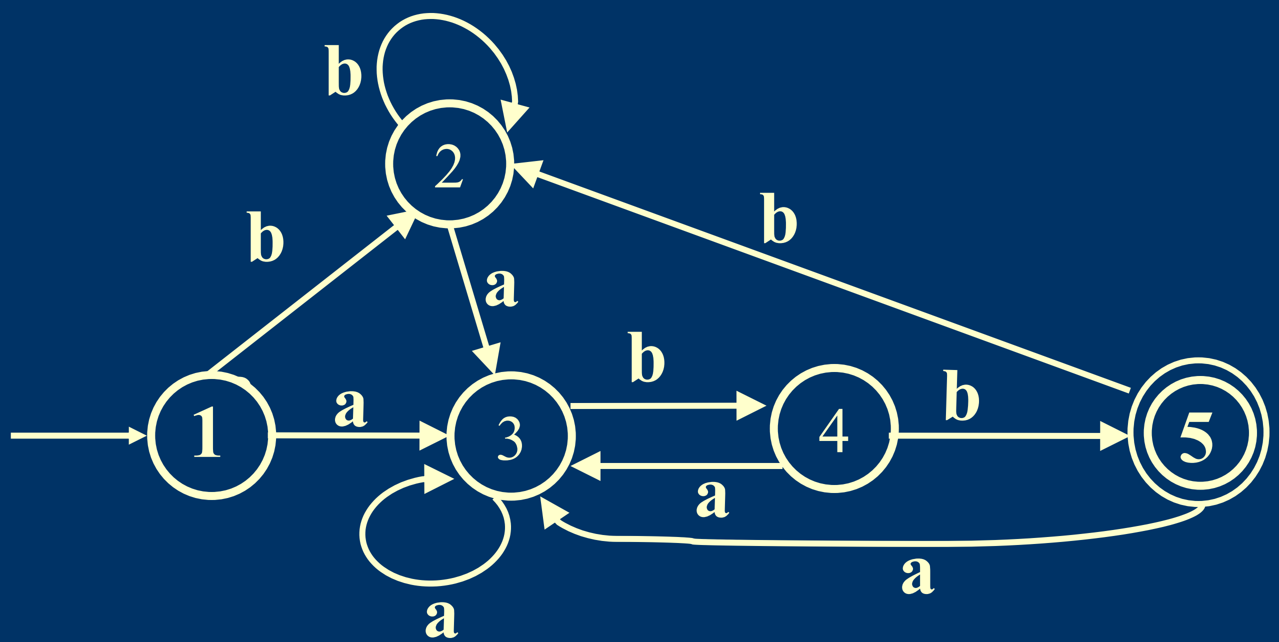


2.4 自动机的化简



2.4 自动机的化简

- 例2



2.4 自动机的化简