第 三 章

词 法 分 析

广东工业大学计算机学院

主观题 10分

设置

给出接受下列在字母表{0,1}上的语言的正规式

(a) 所有以00结束的串的集合;

(b) 所有具有三个0的串的集合。

正常使用主观题需2.0以上版本雨课堂

作答

主观题 10分

设置

给定如下正规式

0(0|1)*1

写出相应的正规文法。

正常使用主观题需2.0以上版本雨课堂

作答

DFA：练习1

- 设有 DFA　M=({0,1,2},{a,b},δ,0,{1,2})

  其中：　δ (0,a)=2;　　　δ (0,b)=1

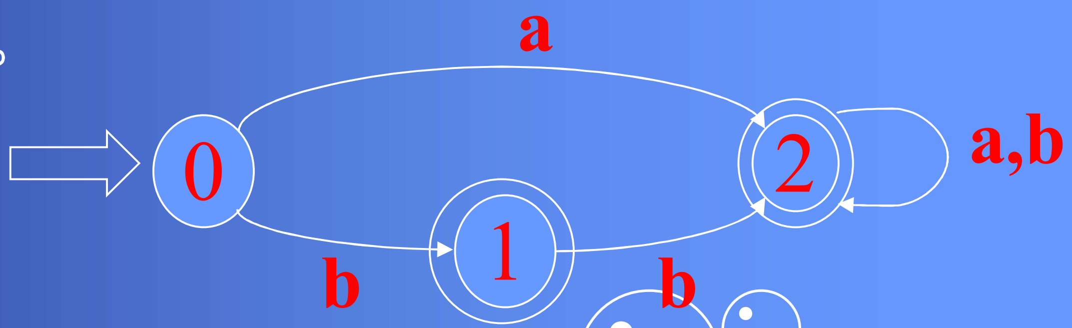
  　　　　δ (1,a)=φ ;　　 δ (1,b)=2

  　　　　δ (2,a)=2;　　　δ (2,b)=2

- 问：该DFA有几个状态？几个输入字符？初态？终态？画出其转换图。

■ 解：有0，1，2共三个状态。0为初态，1和2为终态。输入字符为a,b两个。

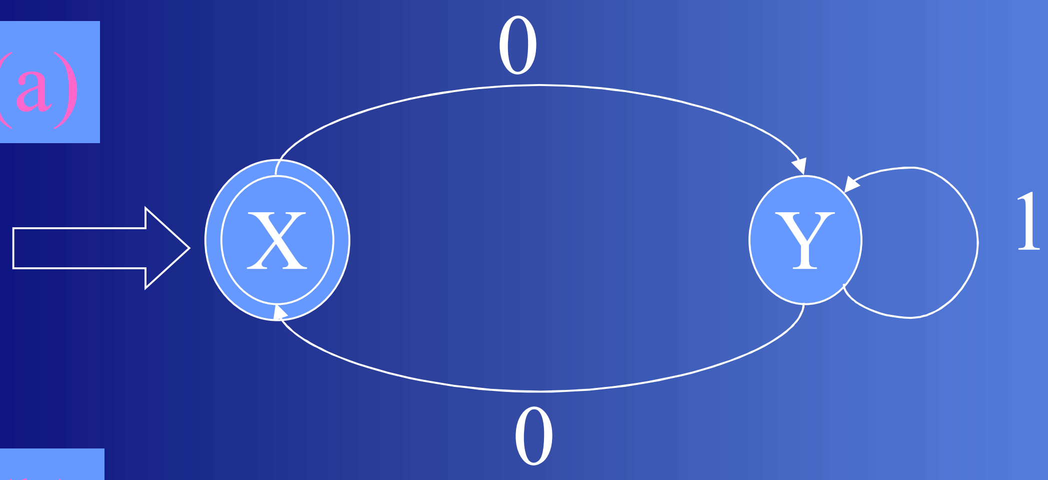
其状态转换图如：



DFA：练习2

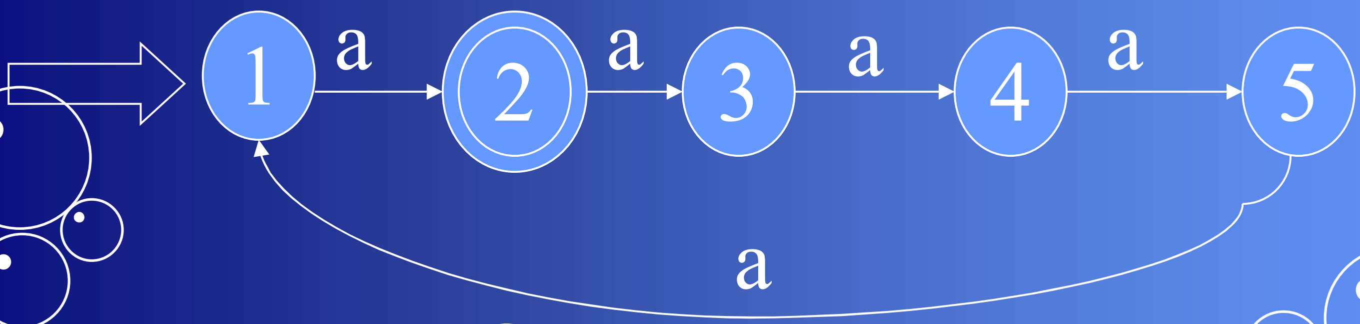
解释下面每个有限自动机识别的语言是什么？

(a)



含偶数个 0的二进制数串的集合

(b)



含(1+5*n)个a的符号串组成的集合, n≥0

本课内容

- 3.4 有穷自动机(之NFA)
- 3.5 正规式和有穷自动机的等价性
- 3.6 正规文法和有穷自动机的等价性
- 3.7 词法分析程序的自动构造工具

2. 不确定的有穷自动机NFA

- 不确定的有穷自动机NFA是一个五元组，N＝(K，∑，f，S，Z)，其中：
- (1) K为状态的有穷非空集
- (2) ∑为有穷输入字母表
- (3) f为K × ∑*到K的子集的映射，即$K \times \Sigma^* \rightarrow 2^K$，$2^K$表示K的幂集
- (4) $S \subset K$是初始状态集
- (5) $Z \subset K$为终止状态集

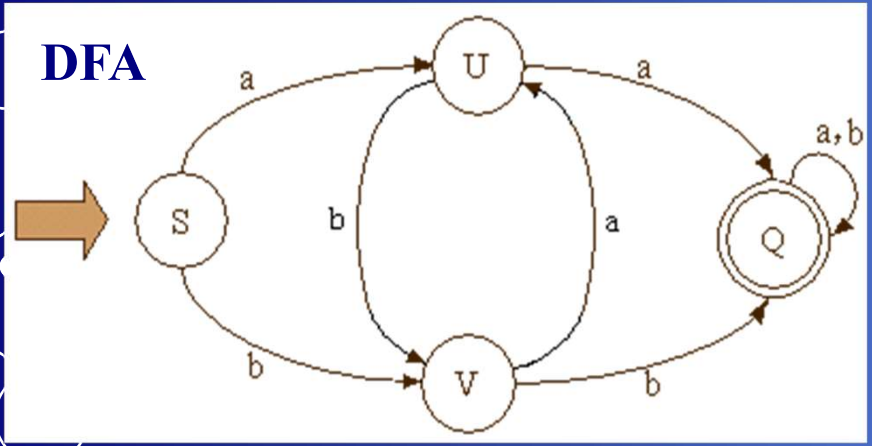
幂集就是所有A的子集所组成的集合。比如集合{1,2,3},它的幂集B就是{∅，{1}，{2}，{3}，{1,2}，{2,3}，{1,3}，{1,2,3}}

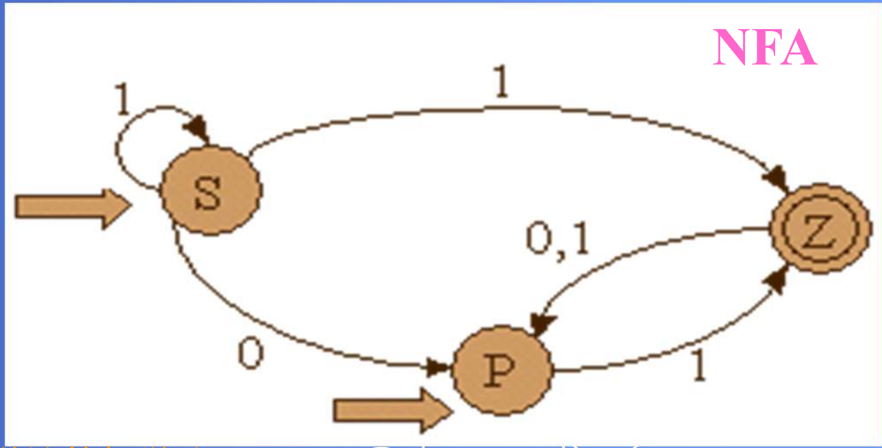
- NFA与DFA的区别：
- (1) DFA中的f：K×Σ→K是单值函数，但在NFA中没有此限制。即在DFA中，知道当前状态和当前输入后，能确定其唯一的后继状态，但在NFA中，后继状态不一定唯一。
- (2) DFA的初态唯一，但NFA中可能有多个初态。

NFA的状态图表示

- 设有NFA N = ({S，P，Z}, {0，1}, f, {S，P}, {Z})，其中
  f(S，0) = {P}　f(Z，0) = {P}　f(P，1) = {Z}
  f(Z，1) = {P}　f(S，1) = {S，Z}

- NFA的状态图：设含有m个状态结点，
- (1) 每个结点可射出若干条弧与别的结点相链接；
- (2) 每条弧用∑*中的一个串作标记（不一定要不同的字而且可以是空字 ε）；
- (3) 整个状态图至少含有一个初态结点和若干个终态结点。





跳转至NFA⇒DFA (P10)

NFA的矩阵表示

- 设有NFA N = ({S，P，Z}, {0，1}, f, {S，P}, {Z})，其中
  f(S，0) = {P}　f(Z，0) = {P}　f(P，1) = {Z}
  f(Z，1) = {P}　f(S，1) = {S，Z}
- NFA的矩阵表示规则与DFA相似。NFA的矩阵表示：
- 不同之处：NFA的矩阵元素有可能是一个状态集合，而DFA的矩阵元素只可能是一个状态。

DFA

状态　符号	a	b	状态标志
S	U	V	0
U	Q	V	0
V	U	Q	0
Q	Q	Q	1

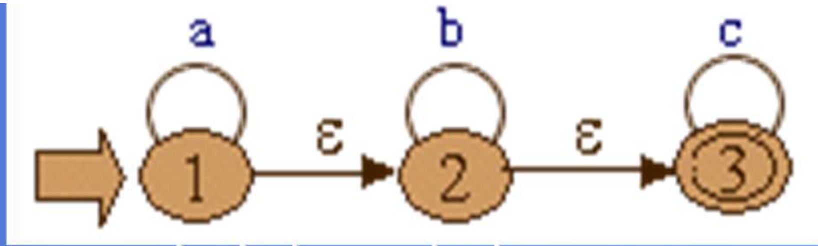
	0	1	
S	{P}	{S,Z}	0
P	{}	{Z}	0
Z	{P}	{P}	1

简化为

	0	1	
S	P	S,Z	0
P	.	Z	0
Z	P	P	1

NFA

在NFA上接受符号串



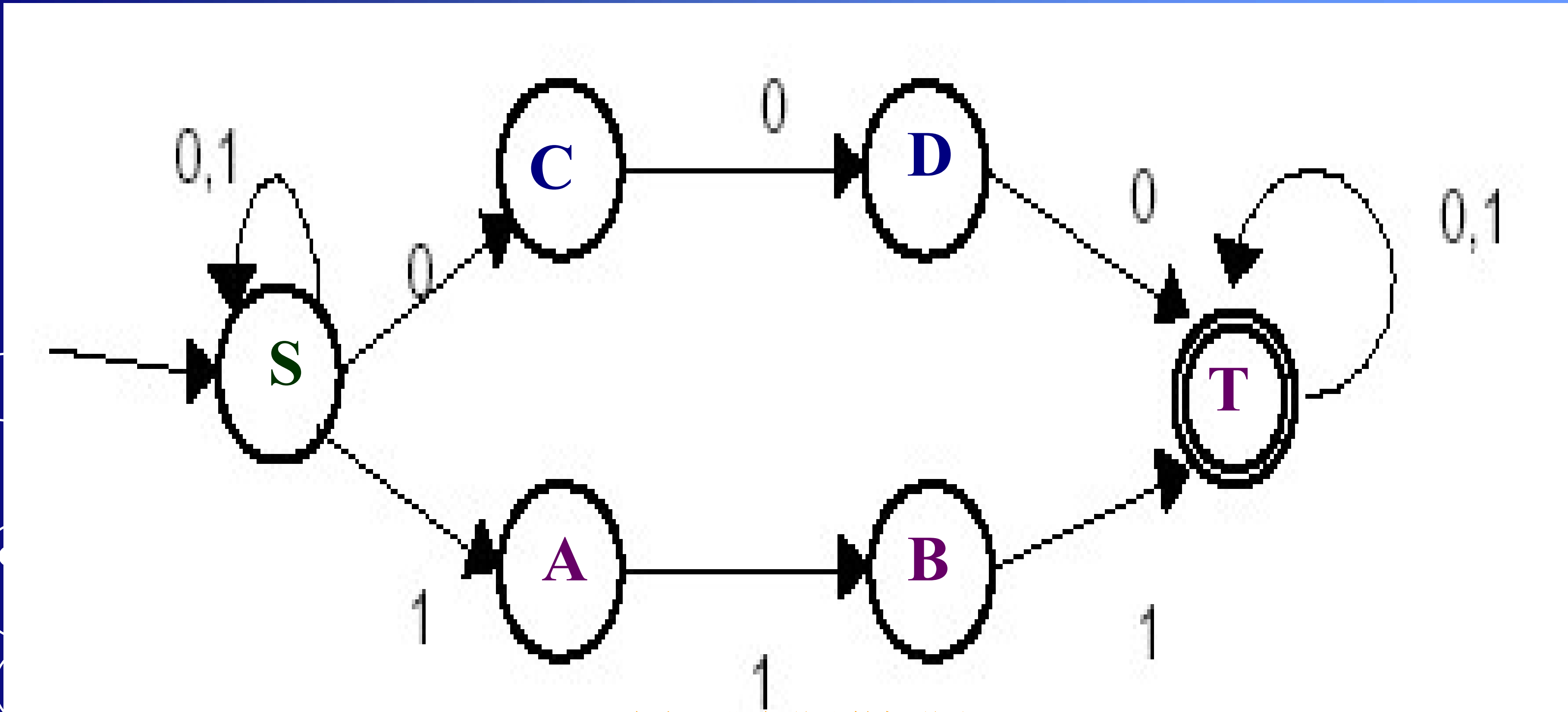
- 类似DFA, 对NFA N = (K，∑，f，S，Z)也有如下定义：
- **1. ∑*上的符号串t在NFA N上运行**
- 一个输入符号串t∈∑*，将t表示成t_1t_x，其中t_1∈∑，t_x∈∑*，如果

$$f(Q, t_1t_x) = f(f(Q, t_1), t_x)，\text{其中}Q \in K$$

- 则称t在NFA N上运行。
- **2. ∑*上的符号串t被NFA N接受**
- 在1成立的基础上，若t∈∑*，f(S，t) = P，其中S为N的开始状态，P∈Z，Z为终态集，则称t为NFA N所接受(识别)。
- 特别地，如果存在下列情况之一：
- (1) 存在N的某些节点既是初态结点又是终态结点
- (2) 存在一条从某个初态结点到某个终态结点的的ε道路
- 则称空符号串(也称作空字)也可为NFA N所接受。

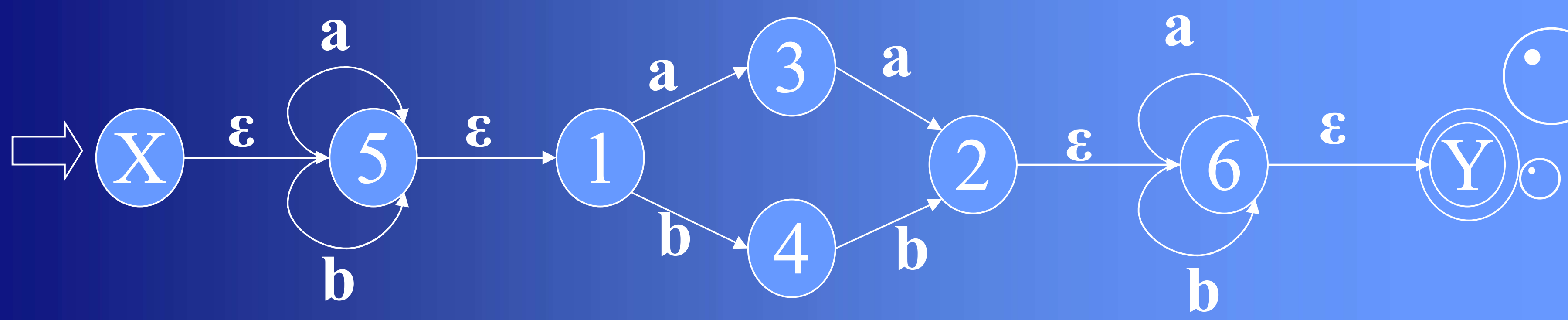
在NFA上接受符号串举例

- 对于串(0|1)*(000|111)(0|1)*，可以被下面的自动机接受
- 问题：下图表示是DFA还是NFA啊？！



NFA：例2

- 例2　NFA M：



- 识别字 abbab，路径是X55142666Y
- 不接受字 ababa，不接受 ε
- L(M)={∑上所有含有相继两个a或相继两个b的字}

NFA有关的结论

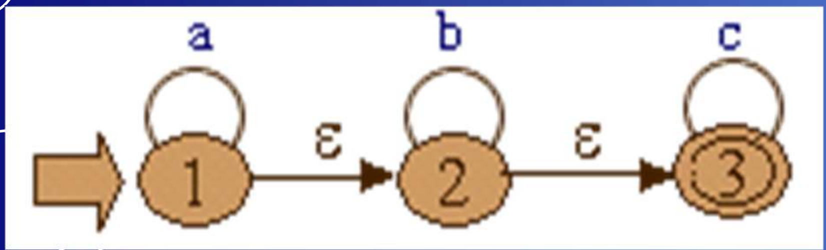
- NFA N所能接受的符号串的全体记为L(N) 。
- NFA有关的结论：∑上一个符号串集V ⊂ ∑*是正规的，当且仅当存在一个∑上的不确定的有穷自动机N，使得V = L(N)。

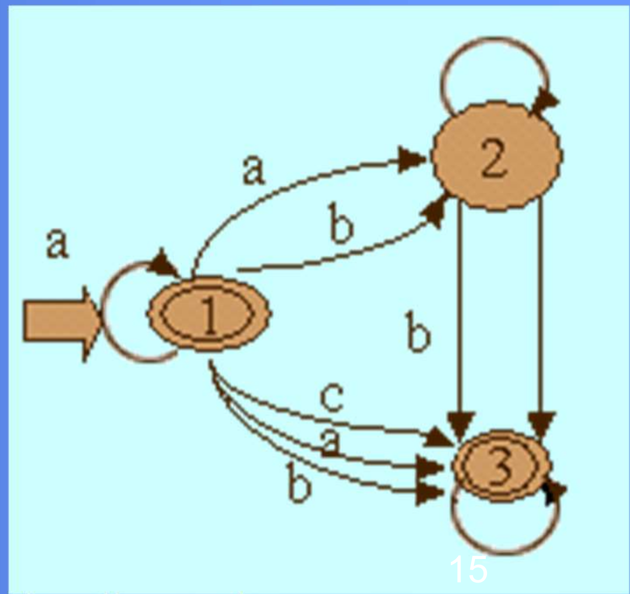
DFA与NFA的区别

- 1. 显然DFA是NFA的特例。
- 2. DFA与NFA的区别:
  - 初态: 数目, 集合与否
  - 状态转换函数δ: $S \times \Sigma \rightarrow S$, $S \times \Sigma^* \rightarrow 2^S$
  - 状态图上: 射出的箭弧数, 弧上的标记形式

DFA与NFA的联系

- FA的等价性 对于每个NFA M, 都存在一个DFA M’，使得L(M’)=L(M)。
  - 后面有证明, 这个证明需要掌握, 它给出了NFA确定化为DFA的方法。
- 对于每一个具有ε转移的NFA，必定存在一个不具有ε转移的NFA，使得L(M) = L(N)：





3. NFA转换为等价的DFA

- 对于两个有穷自动机M和M’，如果L(M) = L(M’)，则称M和M’是等价的。
- 对每个NFA N，一定存在一个DFA M，使得L(M) = L(N)，即：
  - 对每个NFA N存在着与之等价的DFA M；
  - 与某一NFA等价的DFA不一定唯一。
- 存在这样的有穷自动机理论：设L为一个由NFA接受的字符串集合，则存在一个接受L的确定的DFA。
- 疑问：为什么要将NFA转换成DFA？
- Anwser：DFA的行为很容易用程序来模拟。

NFA ⇒ DFA子集法：基本思想

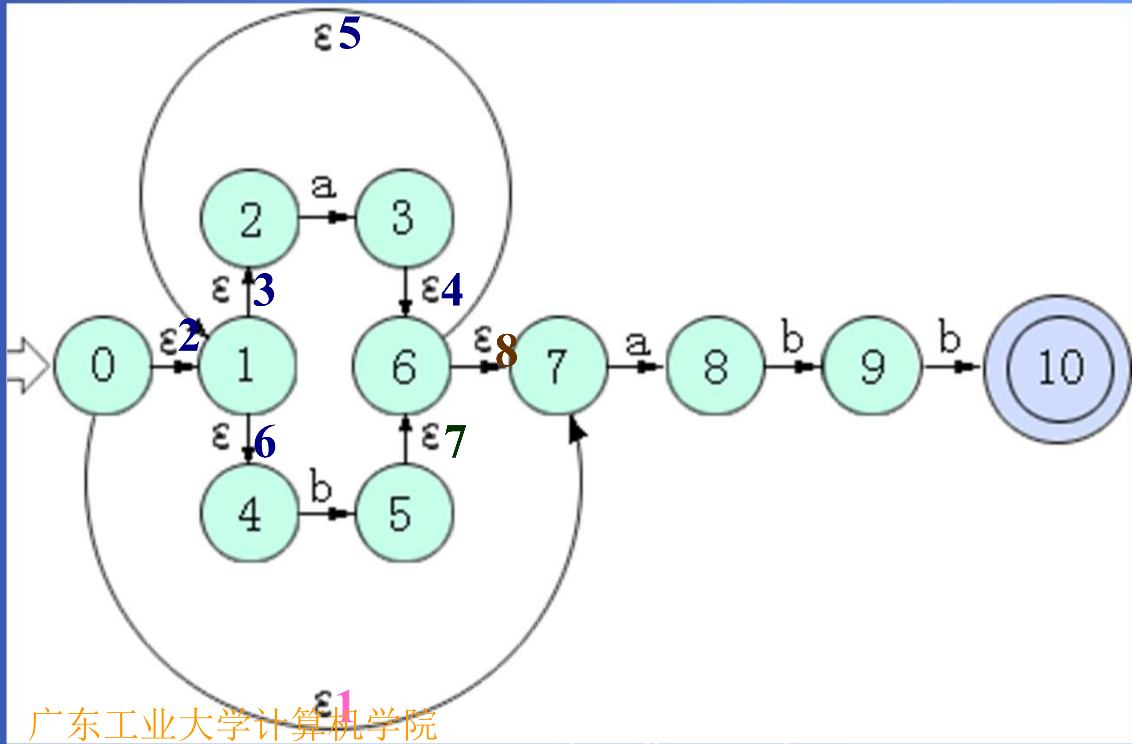
- 从NFA的矩阵表示可以看出，表项可能是一个状态集合，而在DFA的矩阵表示中，表项必是一个状态。
- NFA到相应的DFA的构造的基本思想是：
- 令该DFA的每一个状态对应NFA的一组状态。
- 也就是说，该DFA使用一个状态，去记录在NFA读入一个输入符号后可能达到的所有状态。
- 再换一种说法：在读入输入符号串$a_1a_2...a_n$之后，该DFA处于这样一个状态P：
- 该状态P表示这个NFA的状态的一个子集T，T是从NFA的开始状态沿着某个标记为$a_1a_2...a_n$的路径，最终可以到达的所有那些状态的集合。

跳转至NFA矩阵表示(P5)

子集法：基本思想例解

- 对于如图所示的NFA，从状态0开始，经过字符串a，可以到达哪些状态：
- (1) 实际经过路径ε_1a：状态8
- (2) 实际经过路径$\varepsilon_2\varepsilon_3$a：状态3

- 此NFA中的状态集{8, 3}，在对应的DFA中就可以用一个新的状态来代替

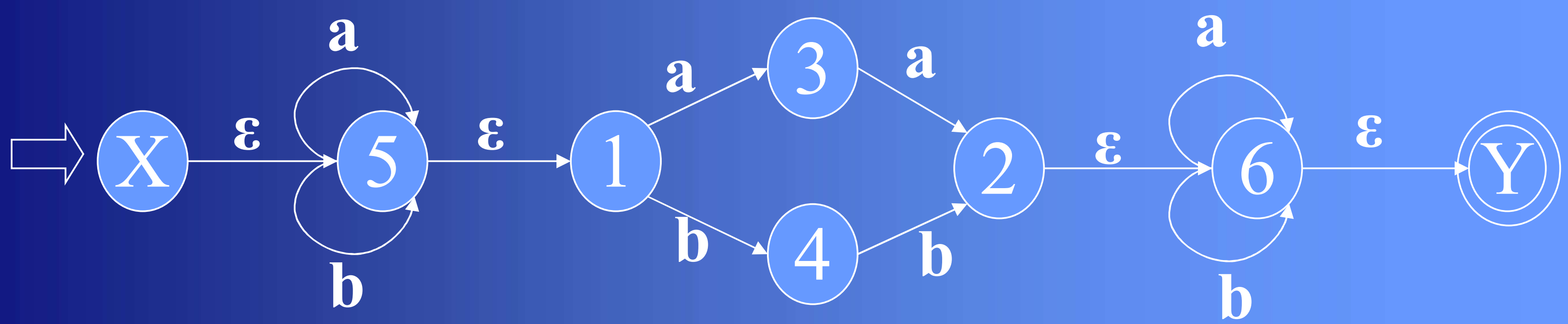


NFA ⇒ DFA子集法：基本运算1

- 在介绍子集法之前，首先介绍几个与状态集合I有关的运算：
- 1.状态集合I的ε- 闭包
- 表示为ε_closure(I)，定义为一状态集，是状态集I中的任意一个状态S经任意条ε弧而能到达的状态的集合。
- 也就是说
  - ① 若q∈I，则q∈ ε_closure(I);
  - ② 若q∈I，则从q出发经任意条ε弧而能到达的任何状态q'∈ε_closure(I)。

状态子集 I 的ε闭包：实例

- 例若 I={X}　　　　则ε_closure(I)={X, 5, 1}
  - I={5, 1}　　　　ε_closure(I)={5, 1}
  - I={2}　　　　ε_closure(I)={2, 6, Y}



NFA ⇒ DFA子集法：基本运算2

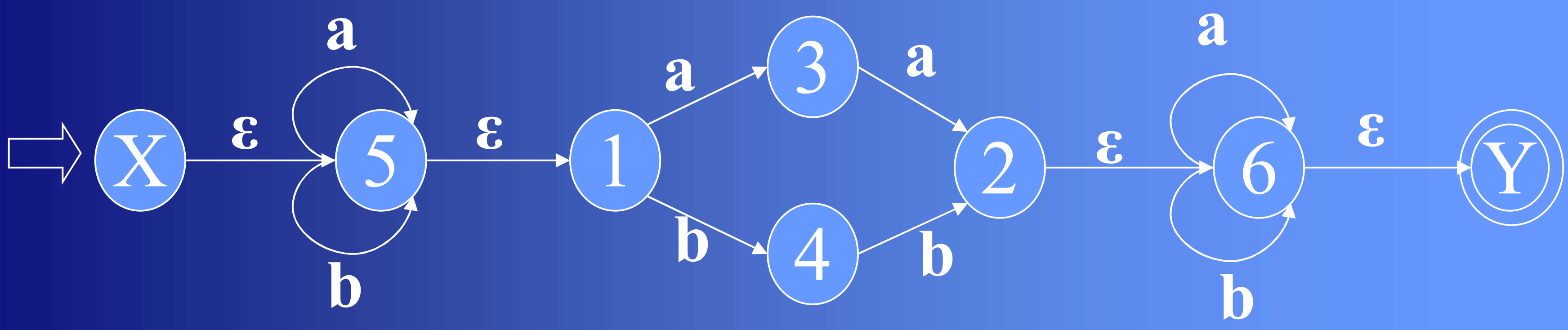
- 在介绍子集法之前，首先介绍几个与状态集合I有关的运算：
- 2.状态集合I的a弧转换Ia
  - (1) 首先求J=move(I, a)

    $J = \{ q' \mid \delta(q, a) = q'$ 且 $q \in I \}$; $a \in \Sigma$

    表示: J是从I中的状态结点出发经过一条a弧而到达的状态结点的集合。(J是所有那些可从I中的某一状态经过一条a弧而到达的所有状态的集合。不能进行ε的扩展！！！)
  - （2）Ia= ε_closure(J)

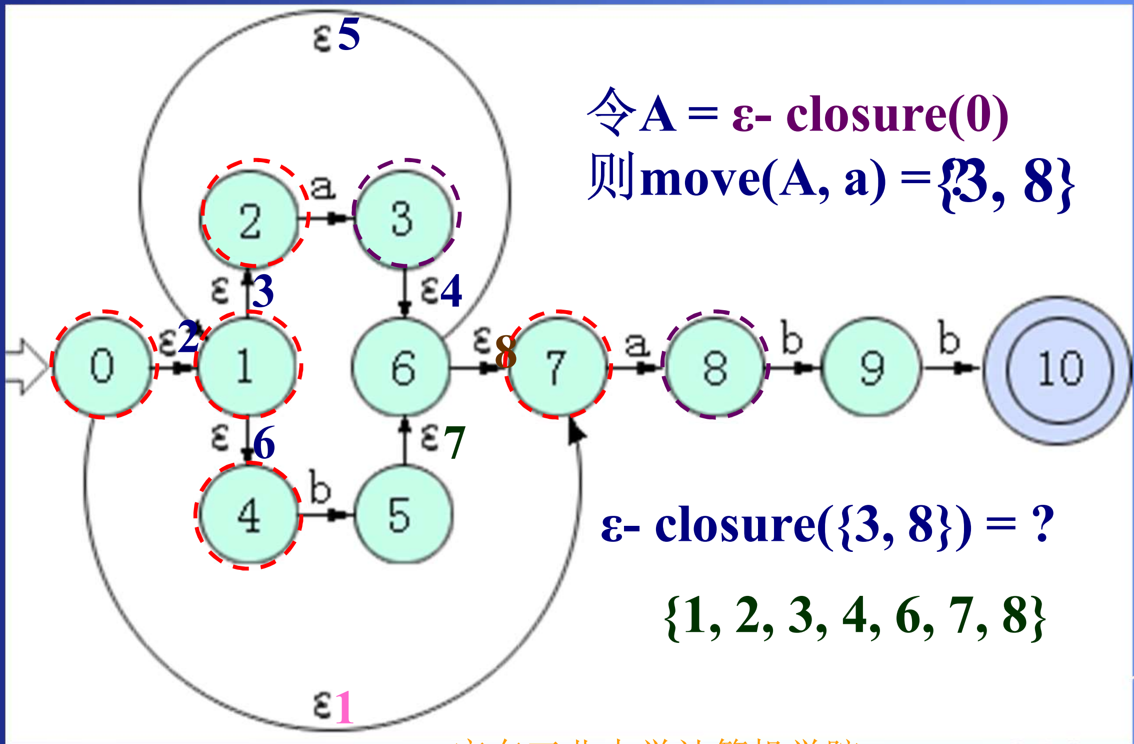
状态集合I的a弧转换:实例

- 若 I={5}　　则 move(I,a)=J = {5}　　Ia={ 5, 1}
  - I={X, 5, 1}　　move(I,a)=J = {5, 3}　Ia= {5, 3, 1}



练习1

- 在右图令A=ε- closure(0) = ?{0, 1, 2, 4, 7}
- Aa= { 1, 2, 3, 4, 6, 7, 8}



令A = ε- closure(0)

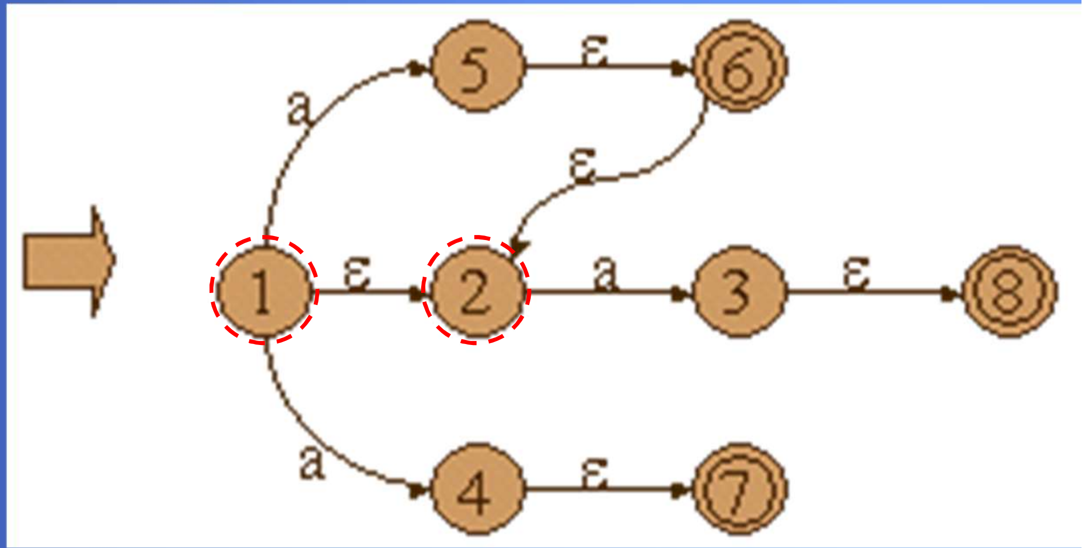
则move(A, a) ={3, 8}

ε- closure({3, 8}) = ?

{1, 2, 3, 4, 6, 7, 8}

基本运算练习2

- 设有状态图：



- (1) 令I = {1},
- ε_closure(I) = ? {1, 2}
- (2)令I = {5},
- ε-closure(I) = ?{5, 6, 2}

(3) 令I={1,2},求Ia

- J=move({1,2}, a) = ?{5, 3, 4}

注意此时不必作ε扩展！

- Ia=ε_closure(J) = ?

{2, 3, 4, 5, 6, 7, 8}

NFA确定化算法：子集法

- 假设NFA N = (K, ∑, f, K_0, K_t)按如下办法构造一个DFA M = (S, ∑, D, S_0, S_t)，使得L(M) = L(N)：
- ① 状态集S由K的一些子集组成(子集构造算法见下页)。用[S_1 S_2 ... S_j]表示S中的一个元素，S_1, S_2, ... S_j都是K的状态。
- 并且约定，状态S_1, S_2, ... S_j按某种规则排列，即对于子集{S_1, S_2} = {S_2, S_1}来说，S的状态就是[S_1 S_2]；
- ② M和N的输入字母表是相同的，都是∑；
- ③ DFA中的状态转换函数定义如下：

D([S_1 S_2, ..., S_j], a) = ε_closure(move([S_1 S_2, ..., S_j], a)) （其实就是某个Si,只不过对于NFA来说是有可能是一个状态集合）

- ④ S_0 = ε_closure(K_0)为M的开始状态；K_0是N开始状态集。
- ⑤ S_t = {[S_i, S_k, ..., S_e]，其中[S_i, S_k, ... S_e]∈S且{S_i, S_k, ..., S_e}∩K_t ≠ ∅}。即S_t是DFA M的终结状态，并且S_t中必须至少包含一个NFA N中的终结状态。（DFA M的终态是含有原来的终态Kt的状态子集）

构造NFA N的状态K的子集的算法之造表法

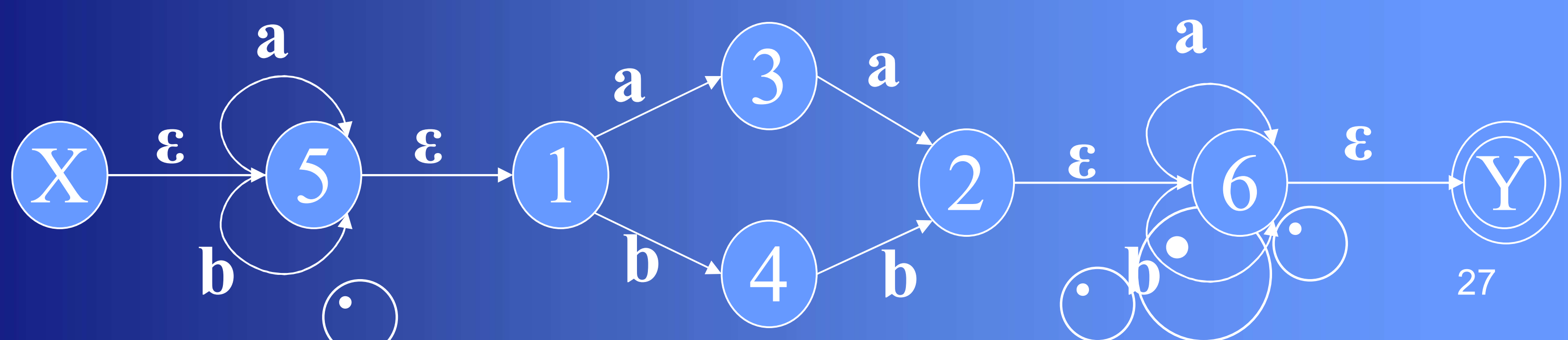
- 构造一张状态转换子集表。设$\sum=\{a_1,a_2,\ldots,a_k\}$
  - ① 第一行第一列为 I =ε_closure(X)，X是唯一的初态；以此 I 求 I_{a1}，I_{a2}，…，I_{ak}。
  - ② 把没有在第一列出现过的I_{ai}填入空行第一列，以此 I_{ai}为新的 I，再求 I_{a1}，I_{a2}，…，I_{ak}。
  - ③ 重复②的过程，直到所有求出的 I_{ai} 都在第一列出现为止。

I	I_{a1}	I_{a2}	…………	I_{ak}
ε_CLOSUR(X)				
……				

子集法例

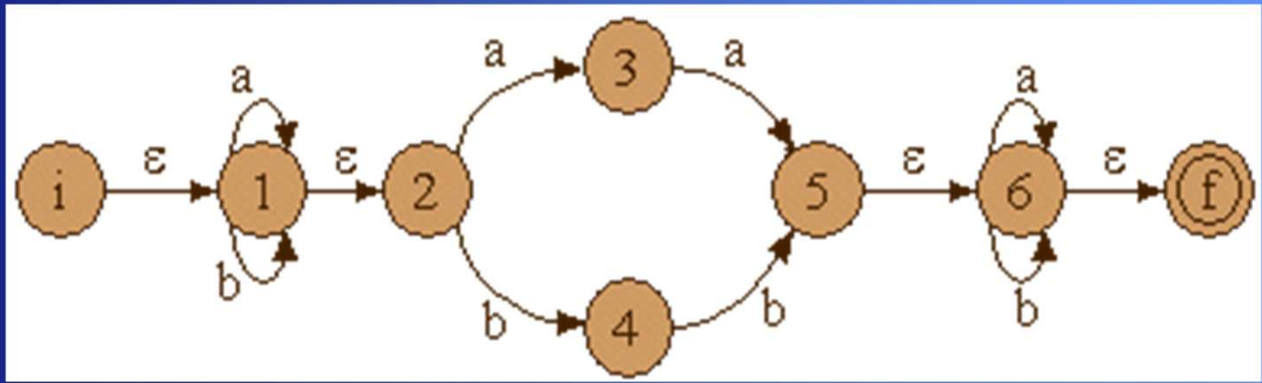
- 对图3.6的NFA构造一张状态子集表。∑={ a, b }

I		I_a		I_b	
{ X, 5, 1}	初 0	{ 5, 3, 1}	1	{ 5, 4, 1}	2
{ 5, 3, 1}	1	{ 5, 3, 1, 2, 6, Y}	3	{ 5, 4, 1 }	2
{ 5, 4, 1}	2	{ 5, 3, 1}	1	{ 5, 4, 1, 2, 6, Y}	4
{ 5, 3, 1, 2, 6, Y}	终3	{ 5, 3, 1, 2, 6, Y}	3	{ 5, 4, 1, 6, Y}	5
{ 5, 4, 1, 2, 6, Y}	终4	{ 5, 3, 1, 6, Y}	6	{ 5, 4, 1, 2, 6, Y}	4
{ 5, 4, 1, 6,Y}	终 5	{ 5, 3, 1, 6, Y}	6	{ 5, 4, 1, 2, 6, Y}	4
{ 5, 3, 1, 6, Y}	终6	{ 5, 3, 1, 2, 6, Y}	3	{ 5, 4, 1, 6, Y}	5



NFA⇒DFA练习

- 将下图的NFA确定化

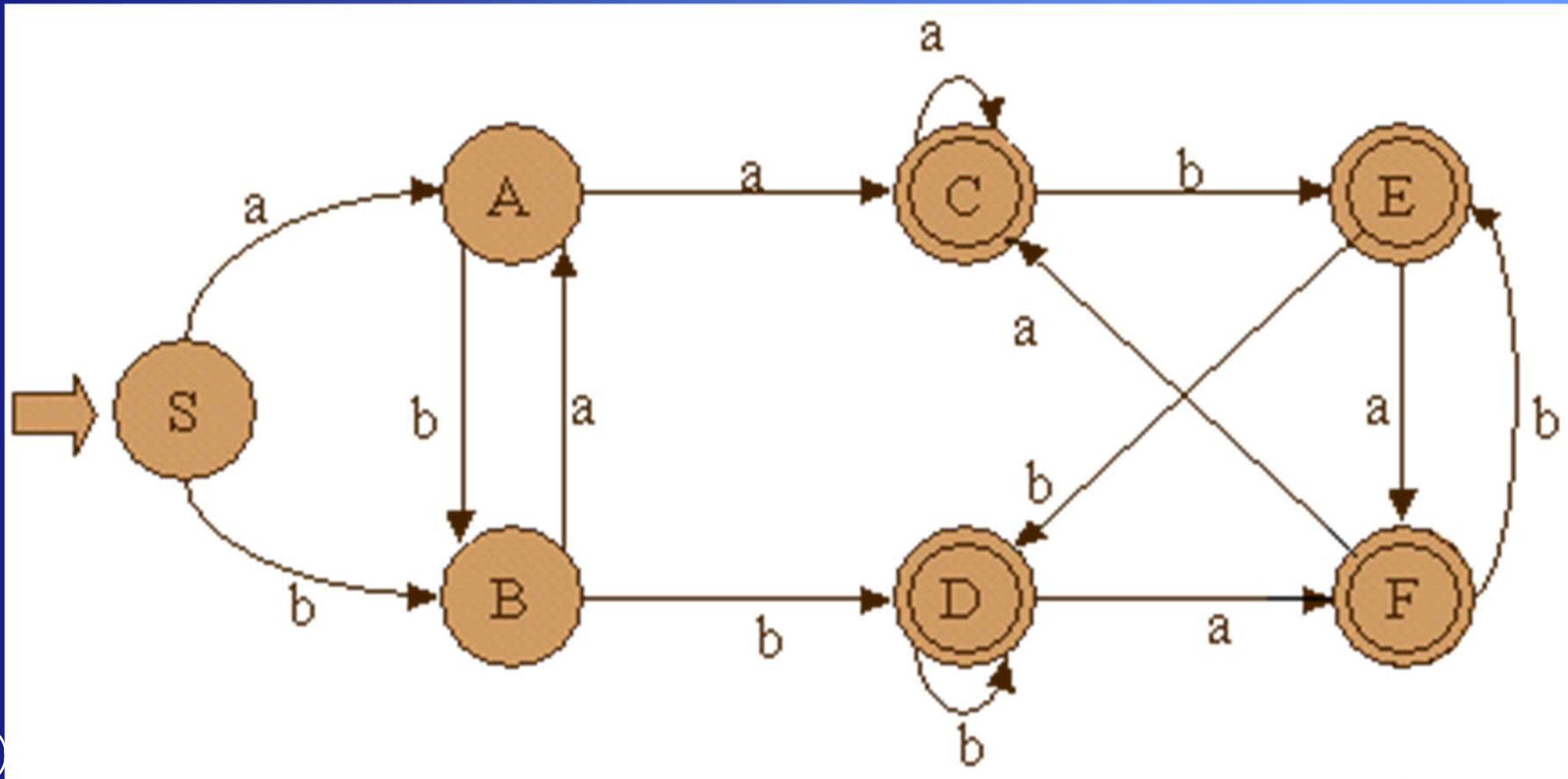


- 划分子集及重新命名

	Ia	Ib
{i,1,2} S	{1,2,3} A	{1,2,4} B
{1,2,3} A	{1,2,3,5,6,f} C	{1,2,4} B
{1,2,4} B	{1,2,3} A	{1,2,4,5,6,f} D
{1,2,3,5,6,f} C	{1,2,3,5,6,f} C	{1,2,4,6,f} E
{1,2,4,5,6,f} D	{1,2,3,6,f} F	{1,2,4,5,6,f} D
{1,2,4,6,f} E	{1,2,3,6,f} F	{1,2,4,5,6,f} D
{1,2,3,6,f} F	{1,2,3,5,6,f} C	{1,2,4,6,f} E

NFA⇒DFA练习

- 确定化后的自动机：



本课内容

- 3.4 有穷自动机(之DFA的化简)
- 3.5 正规式和有穷自动机的等价性
- 3.6 正规文法和有穷自动机的等价性
- 3.7 词法分析程序的自动构造工具

DFA的化简

- 我们说一个有穷自动机是化简了的，是该自动机指没有多余状态，且不存在两个互相等价的状态。
- 对于一个有穷自动机，可以通过消除多余状态和合并等价状态而转换成一个最小的与之等价的有穷自动机。
- 多余状态：从该自动机的开始状态出发，任何输入串也不能到达的状态。
- 在右图(a)到图(b)，哪个多余状态被消去了？
- **Answer**：S_4
- 在右图(b)中，那个状态是多余状态？
- **Answer**：S_6　S_8

	0	1	
S_0	S_1	S_5	0
S_1	S_2	S_7	1
S_2	S_2	S_5	1
S_3	S_5	S_7	0
S_4	S_5	S_6	0
S_5	S_3	S_1	0
S_6	S_8	S_0	1
S_7	S_0	S_1	1
S_8	S_3	S_6	0

(a)

	0	1	
S_0	S_1	S_5	0
S_1	S_2	S_7	1
S_2	S_2	S_5	1
S_3	S_5	S_7	0
S_5	S_3	S_1	0
S_6	S_8	S_0	1
S_7	S_0	S_1	1
S_8	S_3	S_6	0

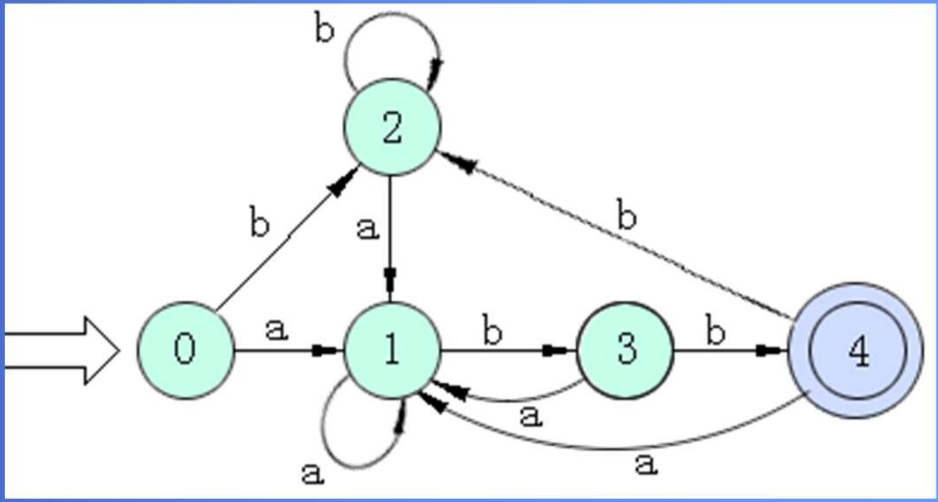
(b)

	0	1	
S_0	S_1	S_5	0
S_1	S_2	S_7	1
S_2	S_2	S_5	1
S_3	S_5	S_7	0
S_5	S_3	S_1	0
S_7	S_0	S_1	1

(c)

状态的等价条件

- 在有穷自动机中，两个状态s和t等价的条件是：
- ① 一致性条件：状态s和t必须同时为可接受状态(终态)或不可接受状态(初态)。
- ② 蔓延性条件：对于所有输入符号，状态s和状态t必须转换到等价的状态里。
- 如果有穷自动机的状态s和t不等价，则称这两个状态是可区别的。
- 如右图所示的有穷自动机
- (1) 状态0和状态4是可区别的：状态0是初态，状态4是终态。
- (2) 状态2和状态3是可区别的：状态2读出b后得到到达状态2；状态3读出b后达到状态4



DFA化简方法：分割法

- "分割法"的基本思想：把一个**DFA M**(不含多余状态)的状态分成一些不相交的子集，使得任意不同子集的状态都是可区别的，而同一子集内的任何两个状态都是等价的。

- 构造过程：
- **(1)** 将**M**的状态集为∏，分为两个子集：K_t由终态(可接受态)组成，另一个$K - K_t$由非终态组成。

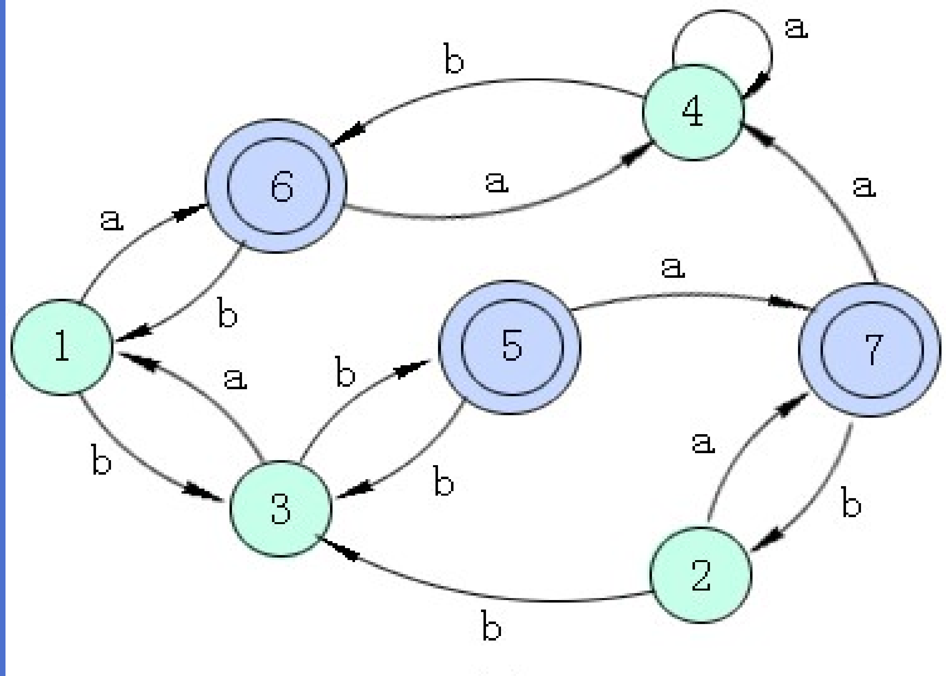
- **(2)** 对每一个子集G_i再次拆分为更小的子集。两个状态**s**和**t**分在同一子组的充要条件是：对某个输入符号**a**，状态**s**和**t**的**a**转换状态在都同一个子集中。

- **(3)** 重复**(2)**，直到所有的子集都不能再划分为止。

- **(4)** 对一个子集，抽取其中一个状态代替该子集。

分割法举例1

- 分割步骤：
- (1) 将M的状态分成两个子集：一个由终态组成，一个由非终态组成，初始划分P_0 = ({1, 2, 3, 4}, {5, 6, 7})。
- (2) 将{1, 2, 3, 4}读入输入符号a后，划分成{1, 2}和{3, 4}。此时有P_1 = ({1, 2}, {3, 4}, {5, 6, 7})



(a)

问题：可否首先试图细分{5, 6, 7}？

可以，读入a时划分划分成{5} {6, 7}

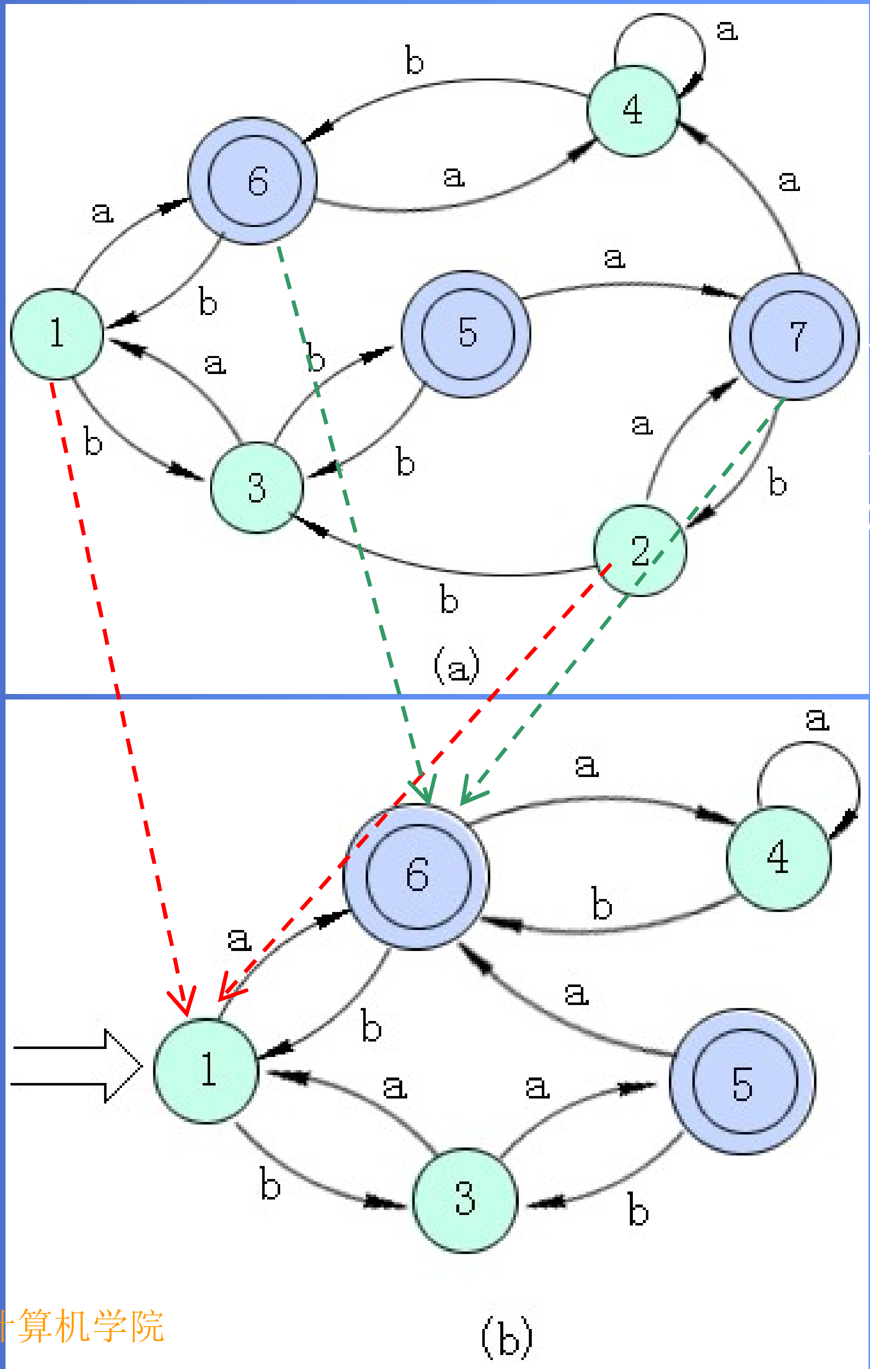
- (3) P_1中的子集{3, 4}读入输入符号a后，得到划分P_2 = ({1, 2}, {3}, {4}, {5, 6, 7})。

实际上，同样需要尝试划分{1, 2}

- (4) P_2中的{5, 6, 7}读入输入符号a或b后，得到划分P_3 = ({1, 2}, {3}, {4}, {5}, {6, 7})。

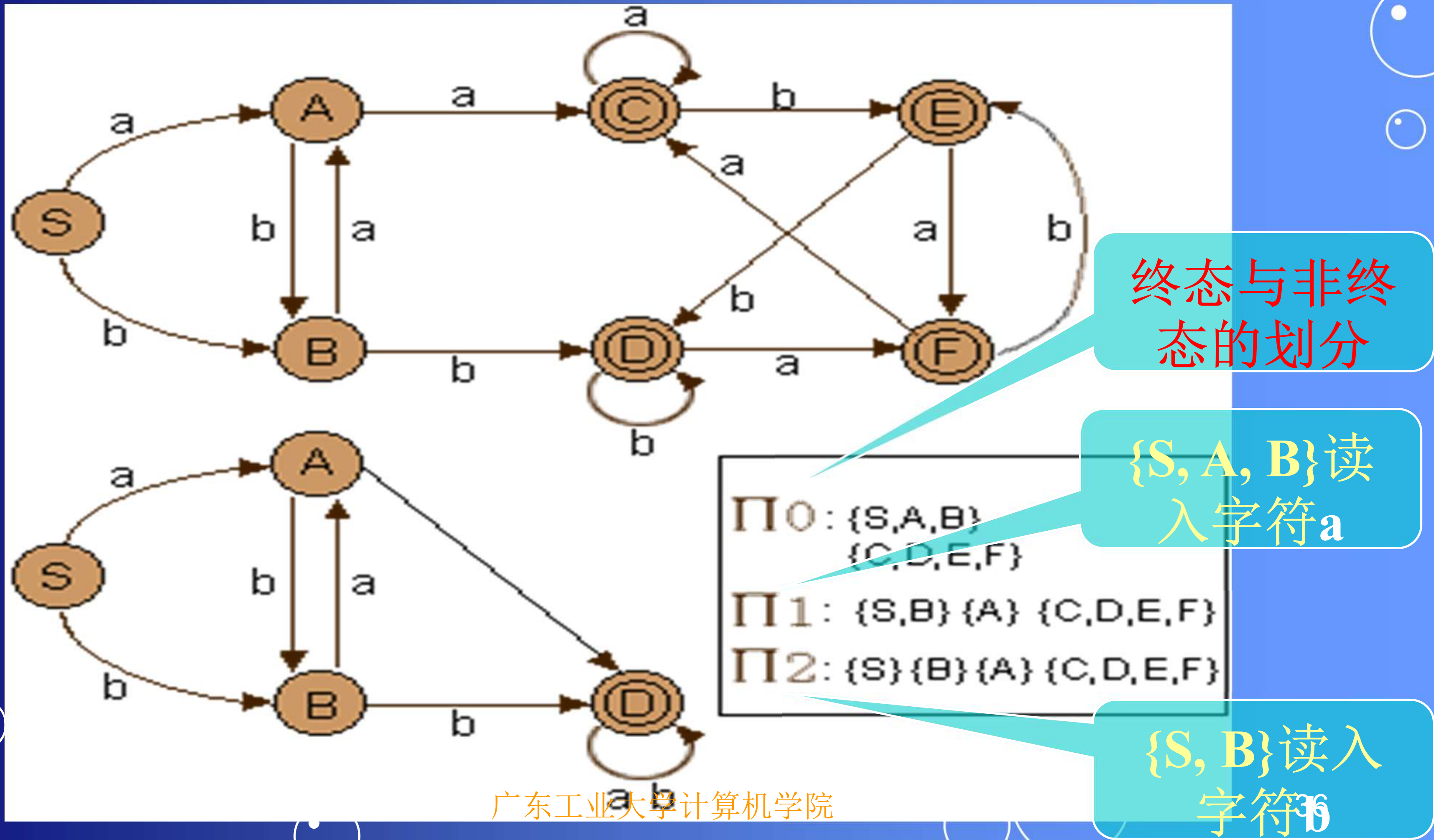
分割法举例1(续)

- 经考察，P_3 = ({1, 2}, {3}, {4}, {5}, {6, 7})。已不可再划分。
- 令状态1代表{1, 2}消去2，令状态6代表{6, 7}消去7，可得最小化的DFA M’
- 比起原来的有穷自动机，化简了的有穷自动机具有较少的状态，因而在计算机上实现起来将简洁些。



分割法举例2

- 这是另一个应用分割法化简DFA的例子：



作业

- 作业格式：
- (1) 在每一次的作业开头，需要写上日期：
- (2) 每道题目的题号要写清楚

主观题 10分

设置

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。

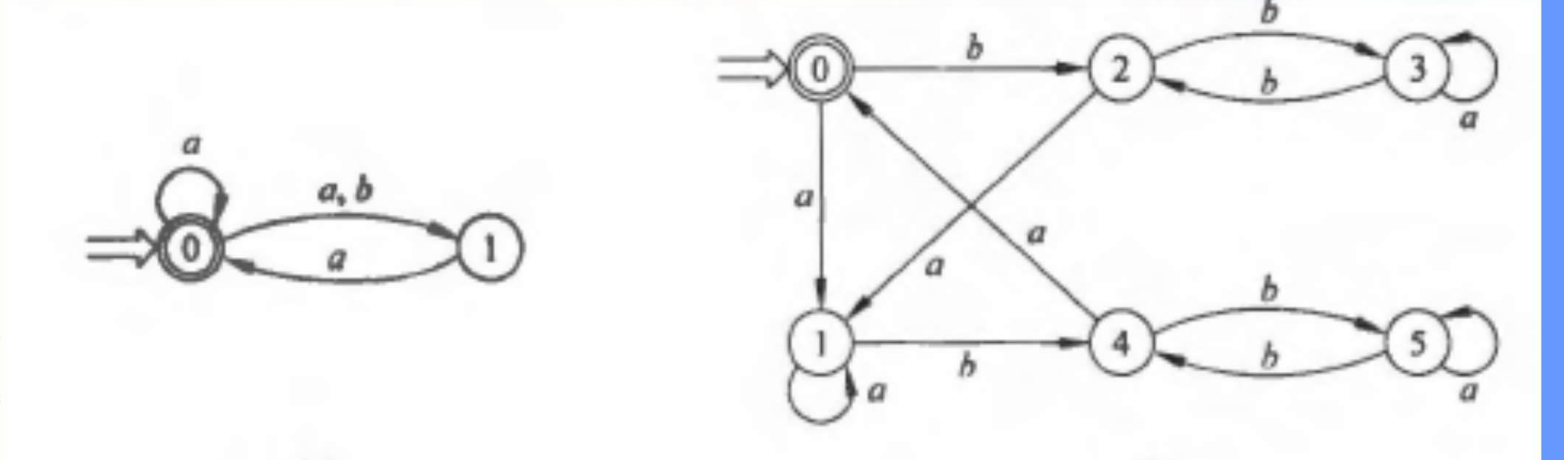
正常使用主观题需2.0以上版本雨课堂

作答

主观题 10分

设置

把下图（a)的NFA确定化，下图（b)的DFA最小化



(a)

(b)

正常使用主观题需2.0以上版本雨课堂

作答