考点	重要程度	占分	题型
1.词法分析相关概念	★★★	3~5	简答题
2.NFA&DFA的概念及区别	★★★★	3~5	简答题
3.正规文法与有限自动机的转化及NFA转化为等价DFA相关问题	★★★★★	10~20	问答题

复习：上下文无关文法

上下文无关文法的定义：

一个上下文无关文法G是一个四元式

$G=(V_T, V_N, S, P)$，其中

V_T：终结符集合(非空)

V_N：非终结符集合(非空)，且$V_T \cap V_N=\varnothing$

S：文法的开始符号，$S \in V_N$

P：产生式集合(有限)，每个产生式形式为

$P \to \alpha$，$P \in V_N$，$\alpha \in (V_T \cup V_N)^*$

开始符号S至少必须在某个产生式的左部出现一次。

3.1 词法分析相关概念

一、词法分析阶段的主要任务

词法分析是编译的第一个阶段，在单词的级别上分析和翻译源程序。

词法分析的任务：从左至右逐个字符地对源程序进行扫描，产生一个个单词符号。

词法分析器又称扫描器：执行词法分析的程序。

视频讲解更清晰

二、对于词法分析器的要求

1.词法分析器的功能和输出形式

功能:输入源程序、输出单词符号

单词符号的种类:

基本字: 如 begin，repeat，…

标识符——表示各种名字：如变量名、数组名和过程名

常数: 各种类型的常数

运算符: +，-，*，/，…

界符: 逗号、分号、括号和空白

2.输出的单词符号的表示形式: **(单词种别，单词自身的值)**

例 C程序: while (i>=j) i--;

输出单词符号：

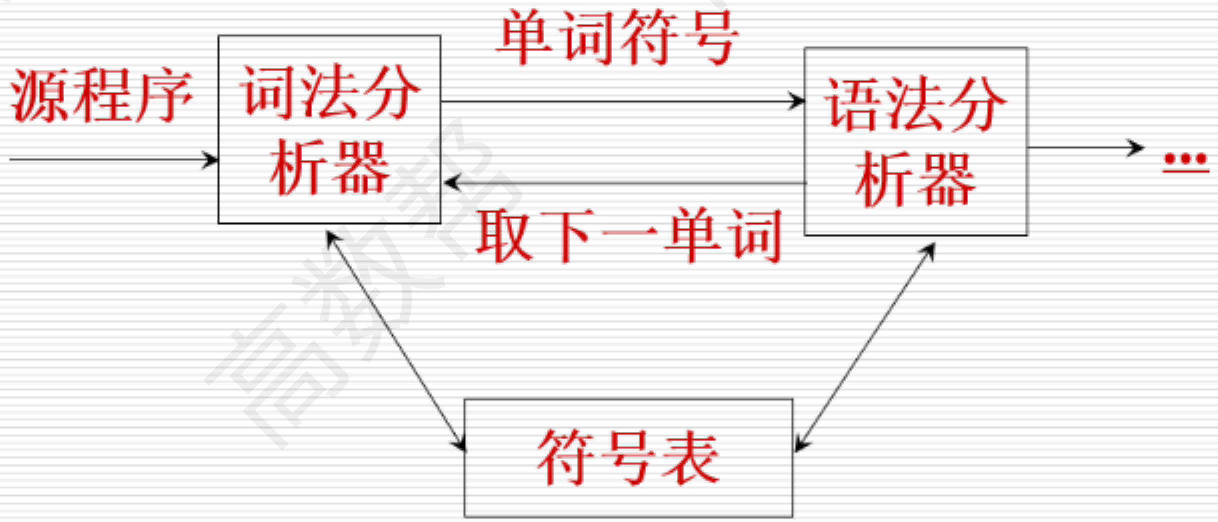
< while, - >
< (, - >
< id , 指向i的符号表项的指针 >
< >= , - >
< id , 指向j的符号表项的指针 >
<) , - >
< id , 指向i的符号表项的指针 >
< -- , - >
< ; , - >

三、词法分析器作为一个独立的子程序

【题1】词法分析是作为一个独立的阶段，是否应当将其处理为一遍呢？

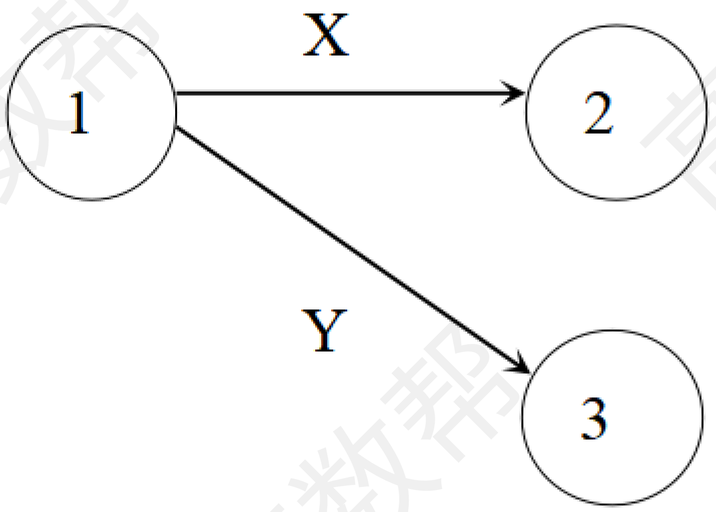
解： 作为独立阶段的优点：结构简洁、清晰和条理化，有利于集中考虑词法分析一些枝节问题。

不作为一遍：将其处理为一个子程序。



四、状态转换图

1.概念



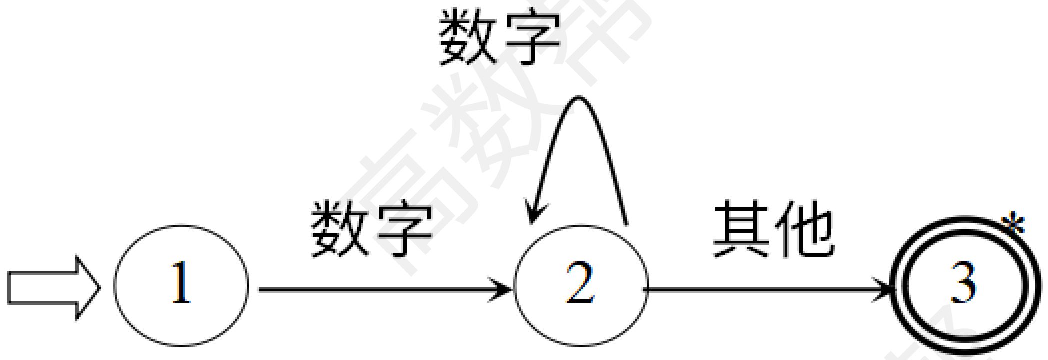
状态转换图是一张有限方向图。

结点代表状态，用圆圈表示。

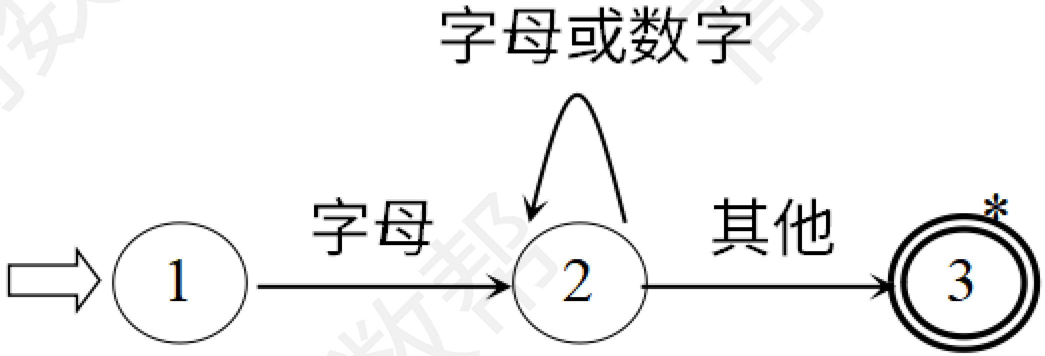
状态之间用箭弧连结，箭弧上的标记(字符)代表射出结点状态下可能出现的输入字符或字符类。

一张转换图只包含有限个状态，其中有一个为初态，至少要有一个终态。

一个状态转换图可用于识别(或接受)一定的字符串。



识别整常数的状态转换图



识别标识符的状态转换图

3.2 NFA&DFA的概念及区别

一、有限自动机(FA)

1.确定状态的有穷自动机（DFA）

一个DFA是一个五元式$M=(S, \Sigma, f, S_0, F)$,其中：

S：有穷状态集，

Σ：输入字母表(有穷)，

f：状态转换函数，为$S\times\Sigma\rightarrow S$的单值部分映射，$f(s，a)=s'$表示：当现行状态为s，输入字符为a时，将状态转换到下一状态s′。我们把s′称为s的一个后继状态。

$S_0\in S$是唯一的一个初态；

$F\subseteq S$：终态集(可空)。

【题2】DFA M=({0，1，2，3}，{a，b}，f，0，{3})， 其中：f 定义如下：

f(0，a)=1　　　f(0，b)=2

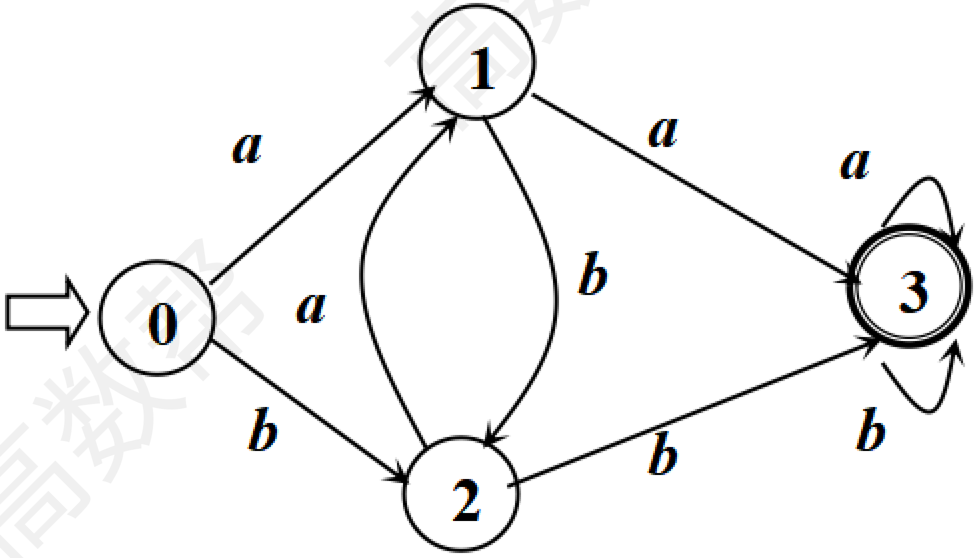
f(1，a)=3　　　f(1，b)=2

f(2，a)=1　　　f(2，b)=3

f(3，a)=3　　　f(3，b)=3

	a	b
0	1	2
1	3	2
2	1	3
3	3	3

状态转换矩阵



状态转换图

2.非确定有限自动机(NFA)

定义： 一个非确定有限自动机(NFA) M是一个五元式$M=(S, \Sigma, f, S_0, F)$，其中：

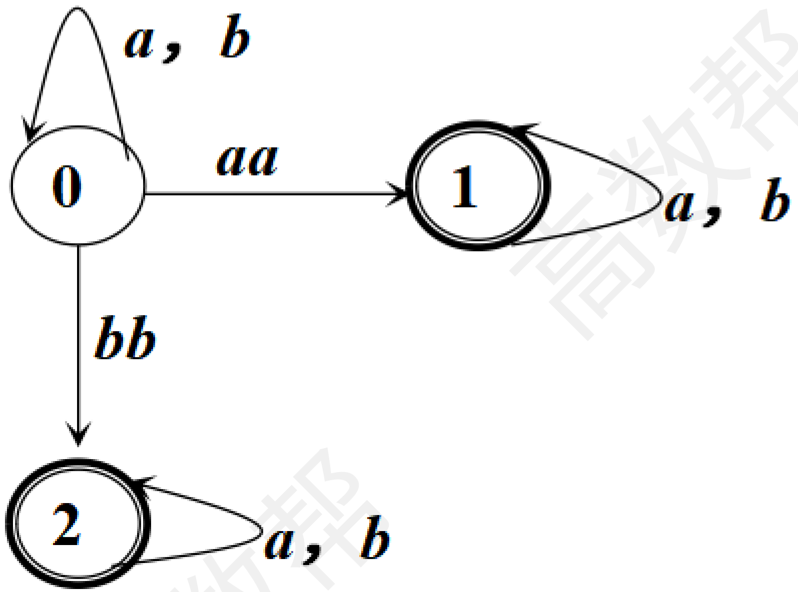
S: 有穷状态集；

Σ： 输入字母表(有穷)；

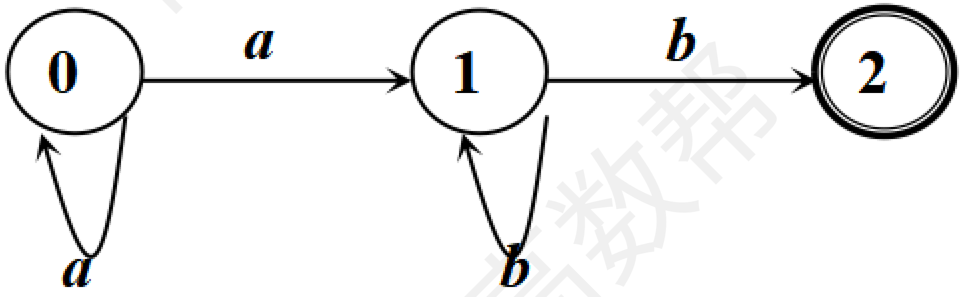
f: 状态转换函数，为$S\times\Sigma^*\to 2^S$的部分映射(非单值)；

$S_0 \subseteq S$是非空的初态集；

$F \subseteq S$： 终态集(可空)。



识别所有含相继两个a或相继两个b的字



$\{a^m b^n \mid m, n \geq 1\}$

二、DFA与NFA的区别

1.开始状态不同：唯一初态；初态集。

2.映射函数不同： $S\times\Sigma\rightarrow S$的单值部分映射； $S\times\Sigma^{*}\rightarrow 2^{S}$的部分映射(非单值)。

从状态图中看NFA和DFA的区别：

1.弧上的标记可以是Σ^{*}中的一个字，而不一定是单个字符；

2.同一个字可能出现在同状态射出的多条弧上。

DFA是NFA的特例

3.3 正规文法与有限自动机的转化及NFA转化为等价DFA相关问题

一、正规表达式

1.正规集可以用正规表达式（简称正规式）表示。

2.正规表达式是表示正规集一种方法。一个字集合是正规集当且仅当它能用正规式表示。

3.正规式和正规集的递归定义：对给定的字母表Σ

(1)ε 和∅都是Σ上的正规式，它们所表示的正规集为{ε}和∅;

(2)任何$a \in \Sigma$，a是Σ上的正规式，它所表示的正规集为$\{a\}$；

3.正规式和正规集的递归定义：对给定的字母表Σ

(3)假定e_1和e_2都是Σ上的正规式，它们所表示的正规集为$L(e_1)$和$L(e_2)$，则

1) $(e_1|e_2)$为正规式，它所表示的正规集为$L(e_1) \cup L(e_2)$，

2) $(e_1.e_2)$为正规式，它所表示的正规集为$L(e_1)L(e_2)$，

3) $(e_1)^*$为正规式，它所表示的正规集为$(L(e_1))^*$，

仅由有限次使用上述三步骤而定义的表达式才是Σ上的正规式，仅由这些正规式表示的字集才是Σ上的正规集。

所有词法结构一般都可以用正规式描述。

若两个正规式所表示的正规集相同，则称这两个正规式等价。如

$$b(ab)^*=(ba)^*b \qquad (a^*b^*)^*=(a|b)^*$$

$L(b(ab)^*)$
$= L(b)L((ab)^*)$
$= L(b)\,(L(ab))^*$
$= L(b)\,(L(a)L(b))^*$
$=\{b\}\ \{ab\}^*$
$= \{b\}\ \{\varepsilon, ab, abab, ababab, \ldots\}$
$= \{b, bab, babab, bababab, \ldots\}$

$L((ba)^*b)$
$= L((ba)^*)\ L(b)$
$= (L(ba))^*L(b)$
$= (L(b)L(a))^*\ L(b)$
$= \{ba\}^*\ \{b\}$
$= \{\varepsilon, ba, baba, bababa, \ldots\}\ \{b\}$
$= \{b, bab, babab, bababab, \ldots\}$

二、NFA转化为DFA

1.定义： 对于任何两个有限自动机M和M'，如果$L(M)=L(M')$，则称M与M'等价。

2.自动机理论中一个重要的结论： 判定两个自动机等价性的算法是存在的。

对于每个NFA M存在一个DFA M'，使得 L(M)=L(M')。

亦即DFA与NFA描述能力相同。

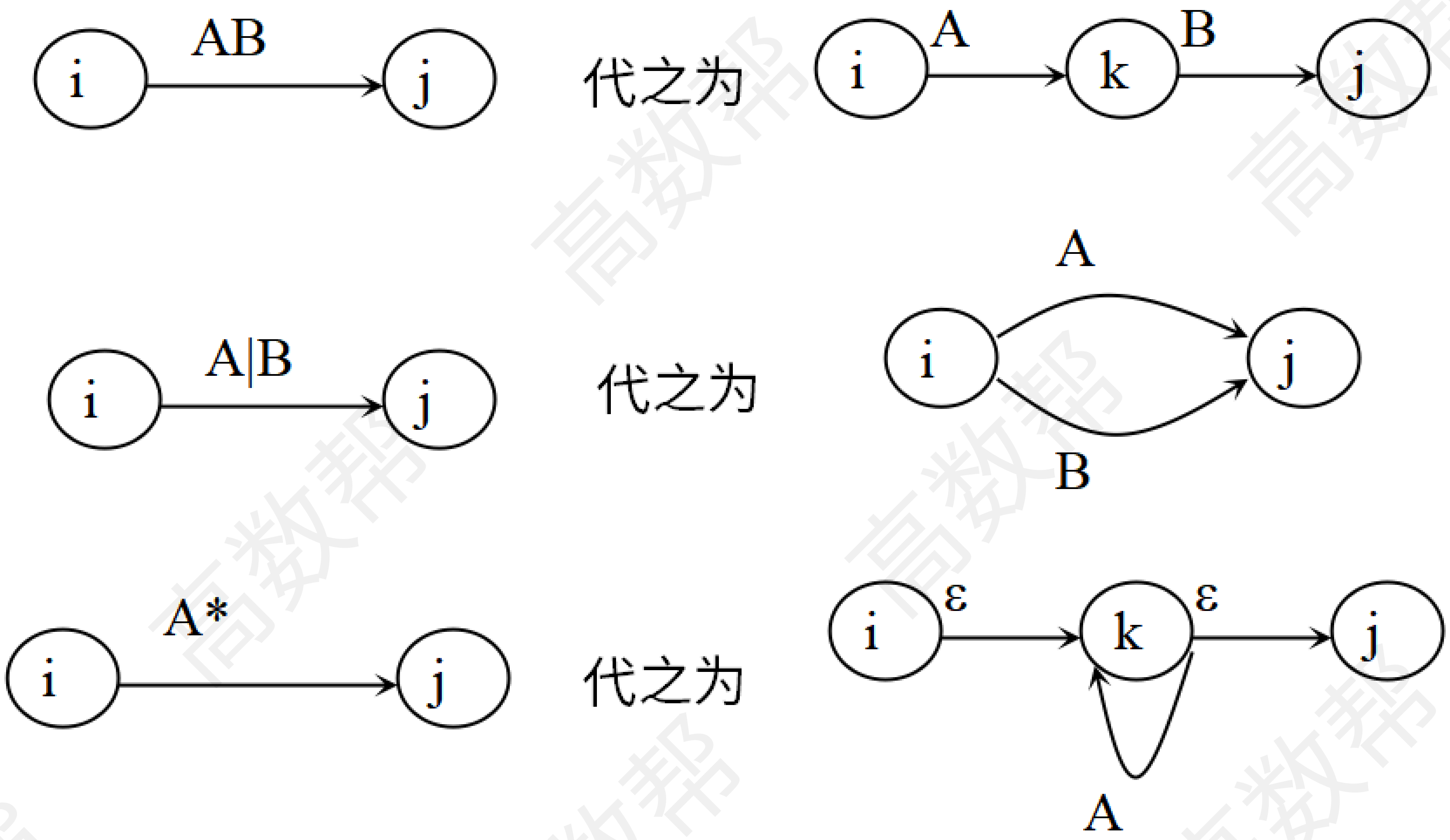
（1）假定NFA $M=<S, \Sigma, \delta, S_0, F>$，我们对M的状态转换图进行以下改造：

1) 引进新的初态结点X和终态结点Y，$X,Y \notin S$，

从X到S_0中任意状态结点连一条ε箭弧， 从F中任意状态结点连一条ε箭弧到Y。

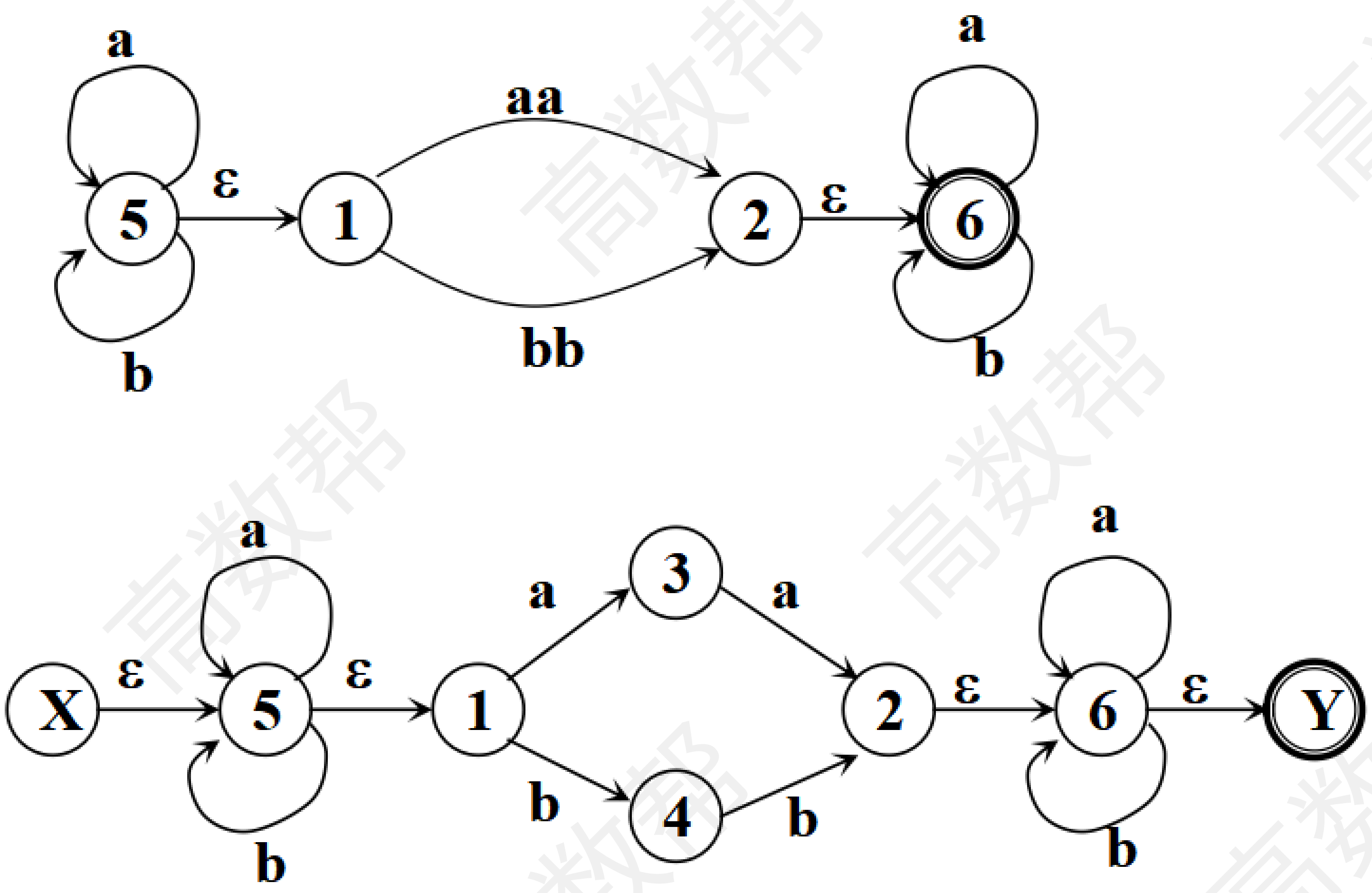
2) 对M的状态转换图进一步实行替换，其中k是新引入的状态。

按下面的三条规则对图进行分裂:



逐步把这个图转变为每条弧只标记为Σ上的一个字符或ε，最后得到一个NFA M’，显然L(M’)=L(M)

【题3】识别所有含相继两个a或相继两个b的字



（2）NFA确定化（NFA→DFA）:子集法

1.设I是状态集的一个子集，定义I的ε-闭包ε-closure(I)为:

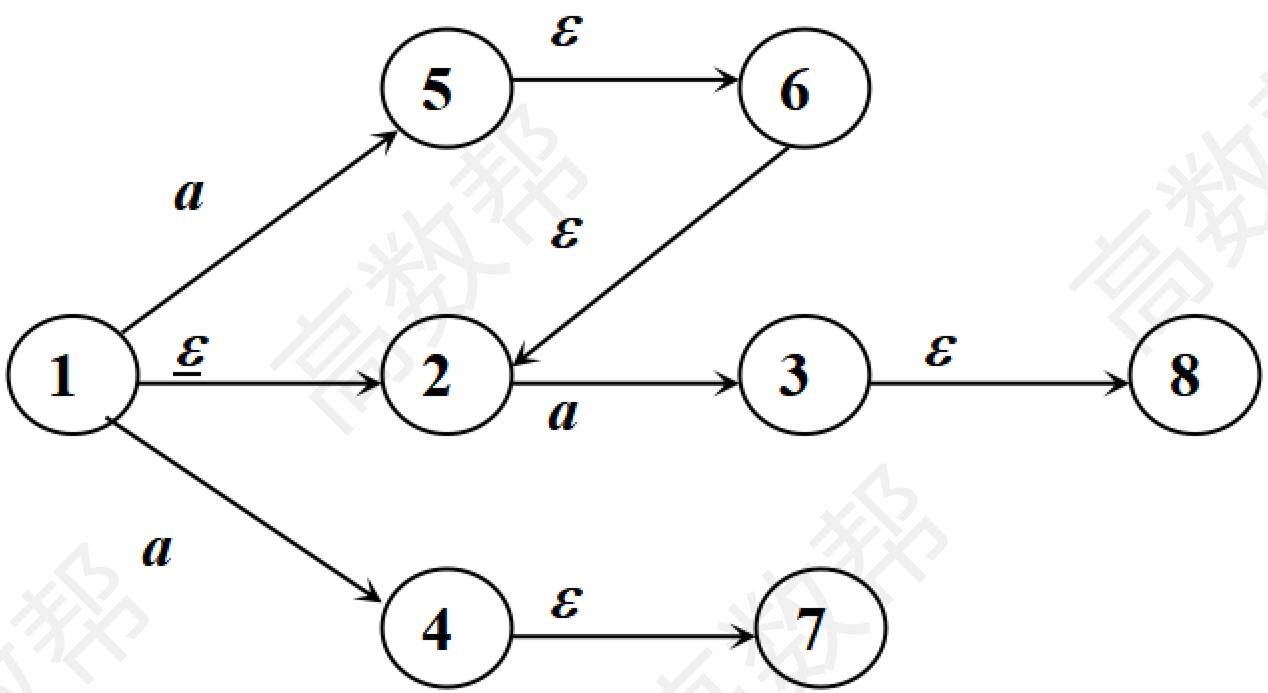
1）若s∈I，则s∈ε-closure(I)；

2）若s∈I，则从s出发经过任意条ε弧而能到达的任何状态s’都属于ε-closure(I)

即ε-closure(I)=I∪{s’|从某个s∈I出发经过任意条ε弧能到达s’}

2.设a是Σ中的一个字符，定义I_a= ε-closure(J)

其中，J为I中的某个状态出发经过一条a弧而到达的状态集合。



ε-closure({1})={1，2}=I

J={5，4，3}

I_a= ε-closure(J)= ε-closure({5，4，3})

={5，4，3，6，2，7，8}

确定化的过程:

不失一般性，设字母表只包含两个a和b，我们构造一张表:

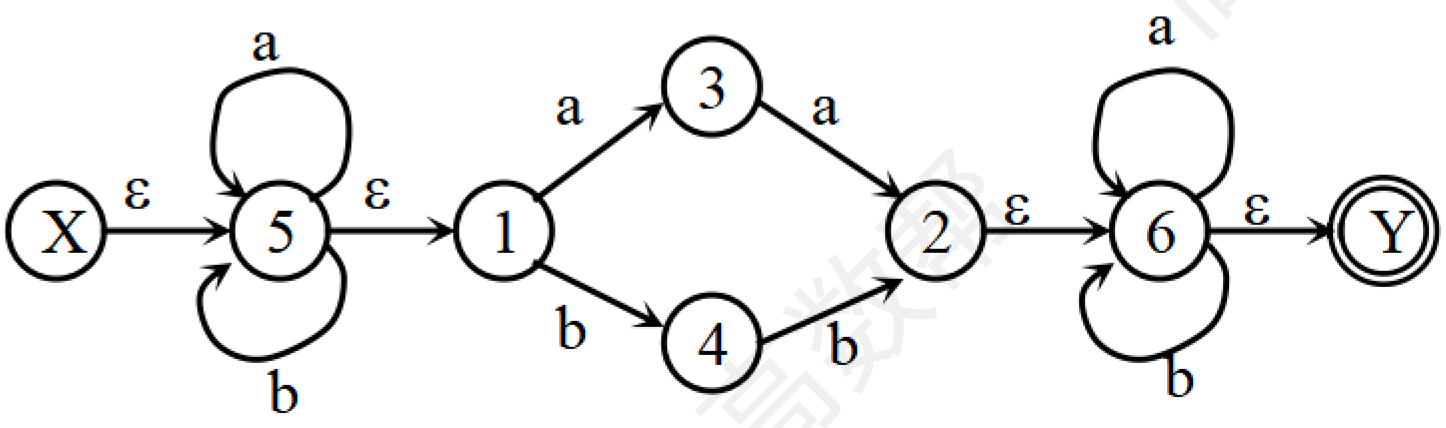
I	I_a	I_b
ε-$Closure(\{X\})$		

状态转换矩阵

首先，置第1行第1列为ε-closure({X})求出这一行的I_a，I_b；

然后，检查这两个I_a，I_b，看它们是否已在表中的第一列中出现，把未曾出现的填入后面的空行的第1列上，求出每行第2，3列上的集合...

重复上述过程，直到所有第2，3列子集全部出现在第一列为止。



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

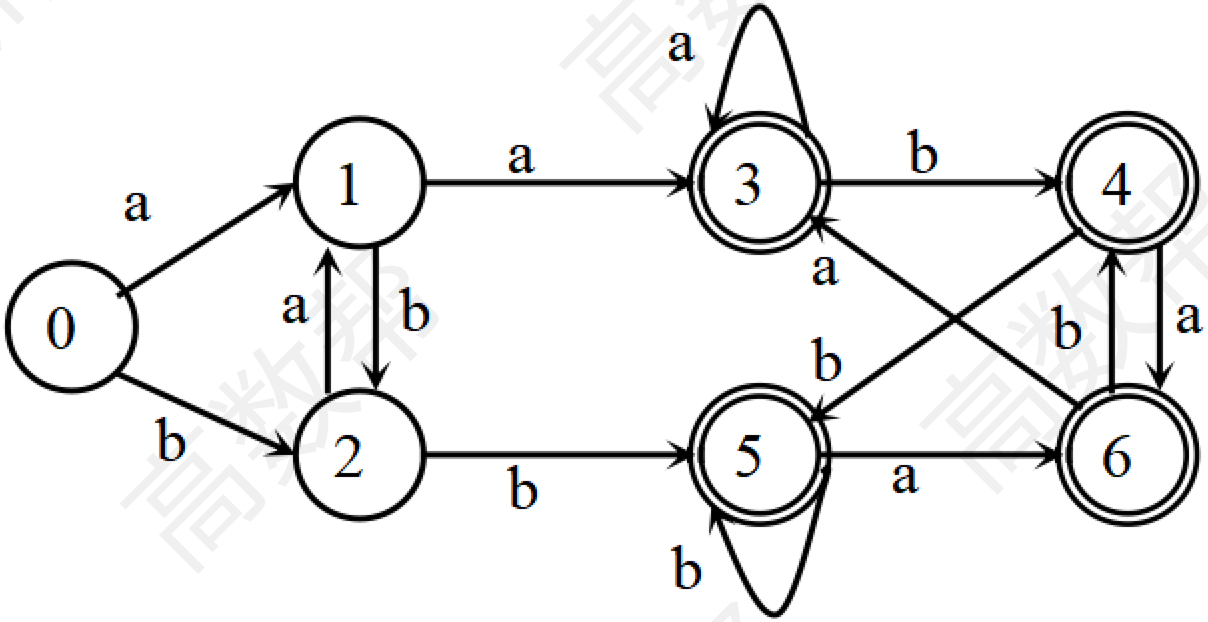
现在把这张表看成一个状态转换矩阵，把其中的每个子集看成一个状态。

这张表唯一刻划了一个确定的有限自动机M，它的初态是ε-closure({X})，它的终态是含有原终态Y的子集。

显然，这个DFA M与NFA M’等价。

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

I	a	b
0	1	2
1	3	2
2	1	5
3	3	4
4	6	5
5	6	5
6	3	4



DFA

(3）确定有限自动机的化简

对DFA M的化简:寻找一个状态数比M少的DFA M’，使得L(M)=L(M’)

两个状态s和t等价的条件为：

一致性条件—状态s和t必须同时为终态或非终态；

蔓延性条件—对于所有输入符号，状态s和t必须转换到等价的状态里。

两个状态不等价，则称它们是可区别的。

对一个DFA M最少化的基本思想（划分法）：

1.把M的状态集划分为一些不相交的子集，使得任何两个不同子集的状态是可区别的，而同一子集的任何两个状态是等价的。

2.让每个子集选出一个代表，同时消去其他状态。

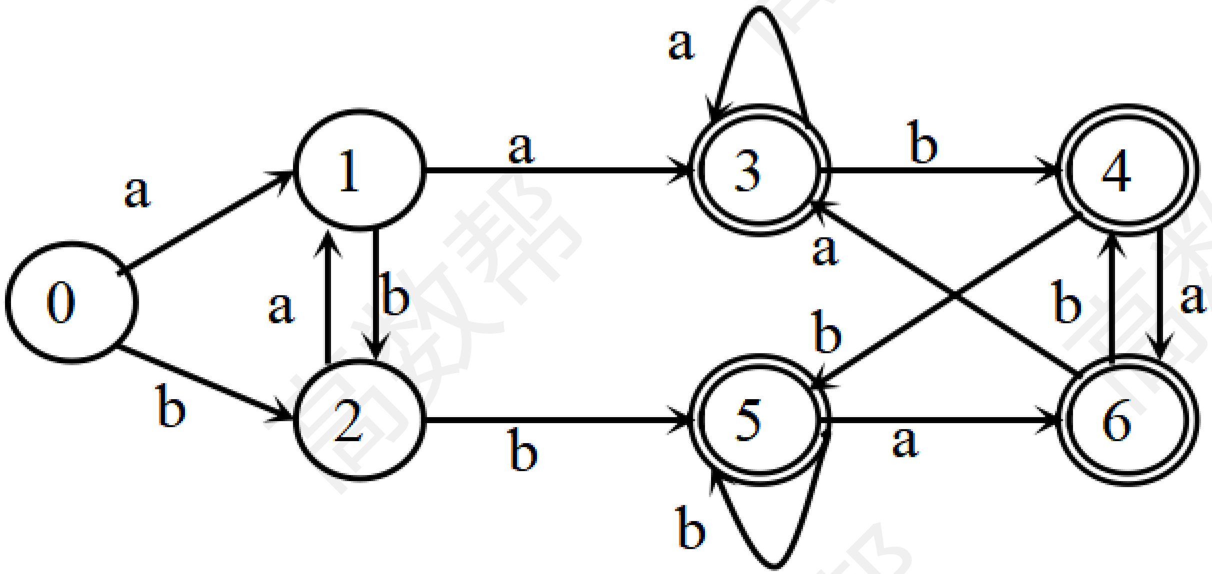
3.把那些原来射入其他等价状态的弧改为射入相应的代表状态。

具体做法: 对M的状态集进行划分

首先，把S划分为终态和非终态两个子集，形成基本划分Π。

假定到某个时候，Π已含m个子集，记为$\Pi=\{I^{(1)}, I^{(2)}, \ldots, I^{(m)}\}$，检查Π中的每个子集看是否能进一步划分:

- 对某个$I^{(i)}$，令$I^{(i)}=\{s_1,s_2, \ldots,s_k\}$，若存在一个输入字符a使得$I_a^{(i)}$不会包含在现行Π的某个子集$I^{(j)}$中，则至少应把$I^{(i)}$分为两个部分。
- 一般地，对某个a和$I^{(i)}$，若$I_a^{(i)}$落入现行Π中 N个不同子集，则应把$I^{(i)}$划分成N个不相交的组，使得每个组J的J_a都落入的Π同一子集。这样构成新的划分。
- 重复上述过程，直到Π所含子集数不再增长。
- 对于上述最后划分Π中的每个子集，我们选取每个子集I中的一个状态代表其他状态，则可得到化简后的DFA M’。
- 若I含有原来的初态，则其代表为新的初态，若I含有原来的终态，则其代表为新的终态。



$I^{(1)}=\{0, 1, 2\}$ $I^{(2)}=\{3, 4, 5, 6\}$

$I_a^{(1)}=\{1, 3\}$

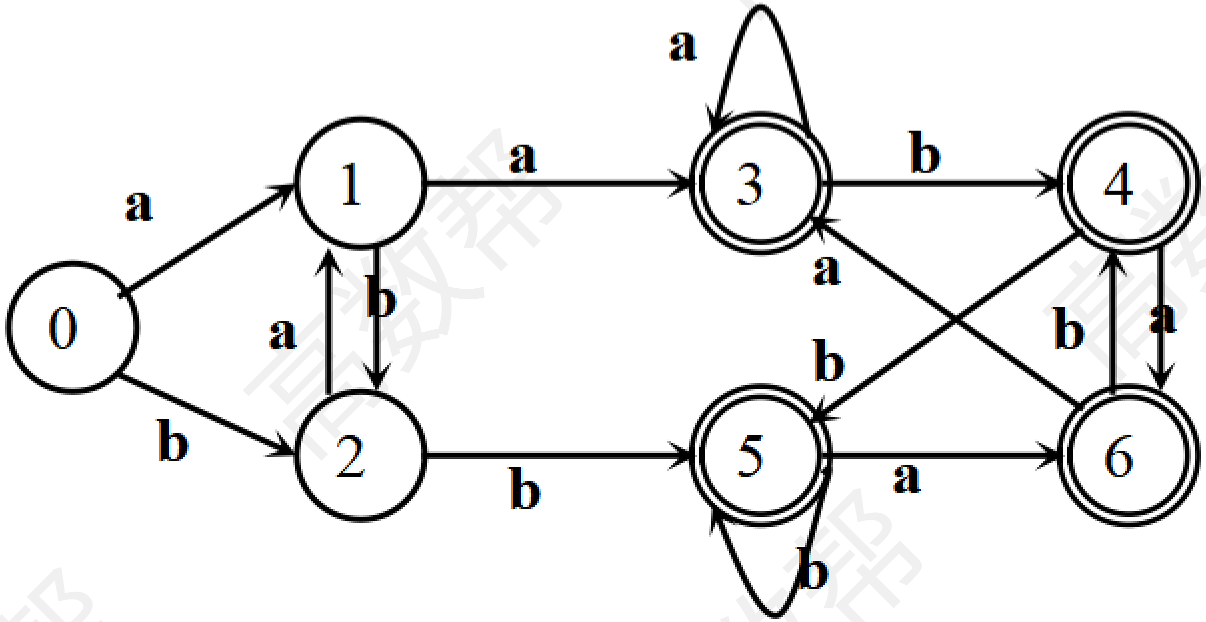
$I^{(11)}=\{0, 2\}$ $I^{(12)}=\{1\}$ $I^{(2)}=\{3, 4, 5, 6\}$

$I^{(11)}=\{0, 2\}$

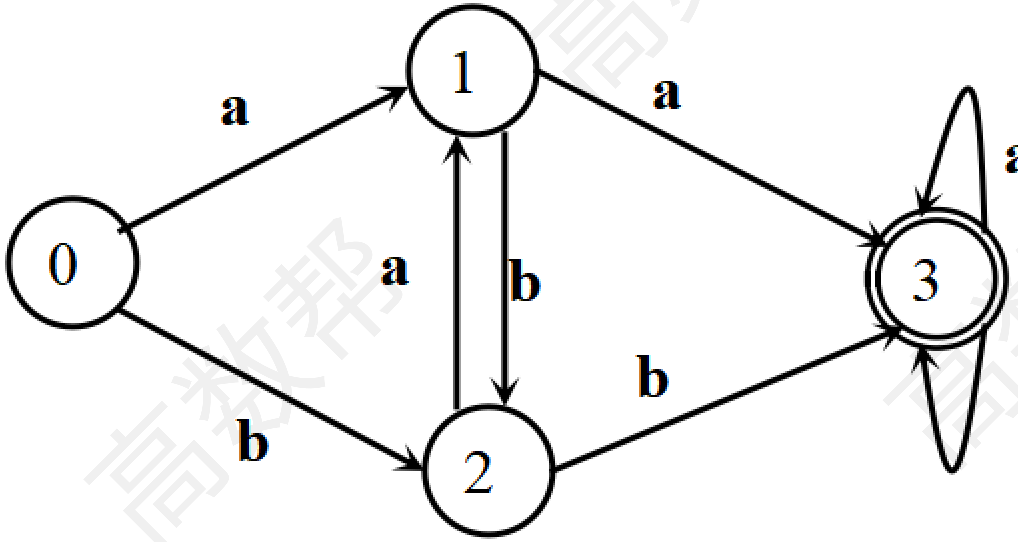
$I_a^{(11)}=\{1\}$ $I_b^{(11)}=\{2, 5\}$

$I^{(111)}=\{0\}$ $I^{(112)}=\{2\}$ $I^{(12)}=\{1\}$ $I^{(2)}=\{3, 4, 5, 6\}$

$I_a^{(2)}=\{3, 6\}$ $I_a^{(2)}=\{4, 5\}$



DFA



最小化DFA

三、正规文法与有限自动机的等价性

定理：

1.对每一个右线性正规文法G或左线性正规文法G，都存在一个有限自动机FA M，

使得L(G) ＝L(M) 。

2.对每一个FA M，都存在一个右线性正规文法G_R和左线性正规文法G_L，

使得L(M)＝L(G_R)＝L(G_L)

视频讲解更清晰