第二章：词法分析

DFA——确定有限自动机

内容介绍

- 确定有限自动机（DFA）的定义
- DFA的表示
- DFA接受的集合
- 应用实例
- 用DFA描述单词
- 自动机的实现
- 注意的问题
- 自动机等价

1. 确定有限自动机的定义

确定有限自动机M为一个五元组:

$$M = (S, \Sigma, S_0, f, Z)$$，其中:

1. S是一个有穷状态集，它的每个元素称为一个状态;
2. Σ是一个有穷字母表，它的每个元素称为一个输入字符;
3. $S_0 \in S$，是唯一的一个初始状态(开始状态);
4. f是状态转换函数:$S \times \Sigma \rightarrow S$，且单值函数.$f(S_i, a)=S_k$ 表示：当前状态为S_i，遇输入字符a 时，自动机将唯一地转换到状态 S_k，称S_k为 S_i的一个后继状态;
5. $Z \subseteq S$，是终止状态集（可接受状态集、结束状态集），其中的每个元素称为终止状态（可接受状态、结束状态），Z可空.

1.确定有限自动机的定义

- DFA的确定性：

1. 初始状态唯一
2. 状态转换函数f: $S \times \Sigma \rightarrow S$是一个单值函数，也就是说，对任何状态$s \in S$，和输入符号 $a \in \Sigma$，$f(S,a)$唯一地确定了下一个状态，即至多确定一个状态

1. 确定有限自动机的定义

- 例子

$S\{0, 1, 2\}$, $\sum=\{a, b\}$,

$f(0, a)=1$, $f(0, b)=2$, $f(1, a)=2$

$S_0=0$, $Z=\{2\}$

2.DFA的表示

状态转换矩阵

- 用自动机的输入字符表示列标
- 用状态来表示行标
- 矩阵元素表示自动机的状态转换函数.
- 标识初始状态和终止状态：一般约定，第一行表示开始状态S_0 ，或在右上角标注“+”；右上角标有“*”或“-”的状态为终止状态 ；

2. DFA的表示

- DFA M=({S,U,V,Q}, {a,b}, f, S, {Q}),其中

f 定义为：

f (S, a)=U
f (V, a)=U
f (S, b)=V
f (V, b)=Q
f (U, a)=Q
f (Q, a)=Q
f (U, b)=V
f (Q, b)=Q

输入字符 / 状态	a	b
S^+	U	V
U	Q	V
V	U	Q
Q^*	Q	Q

2. DFA的表示

状态转换图 ：用有向图表示自动机

1. 结点表示状态：

   1.1 非终止状态由单圆圈围住的状态标识来表示;

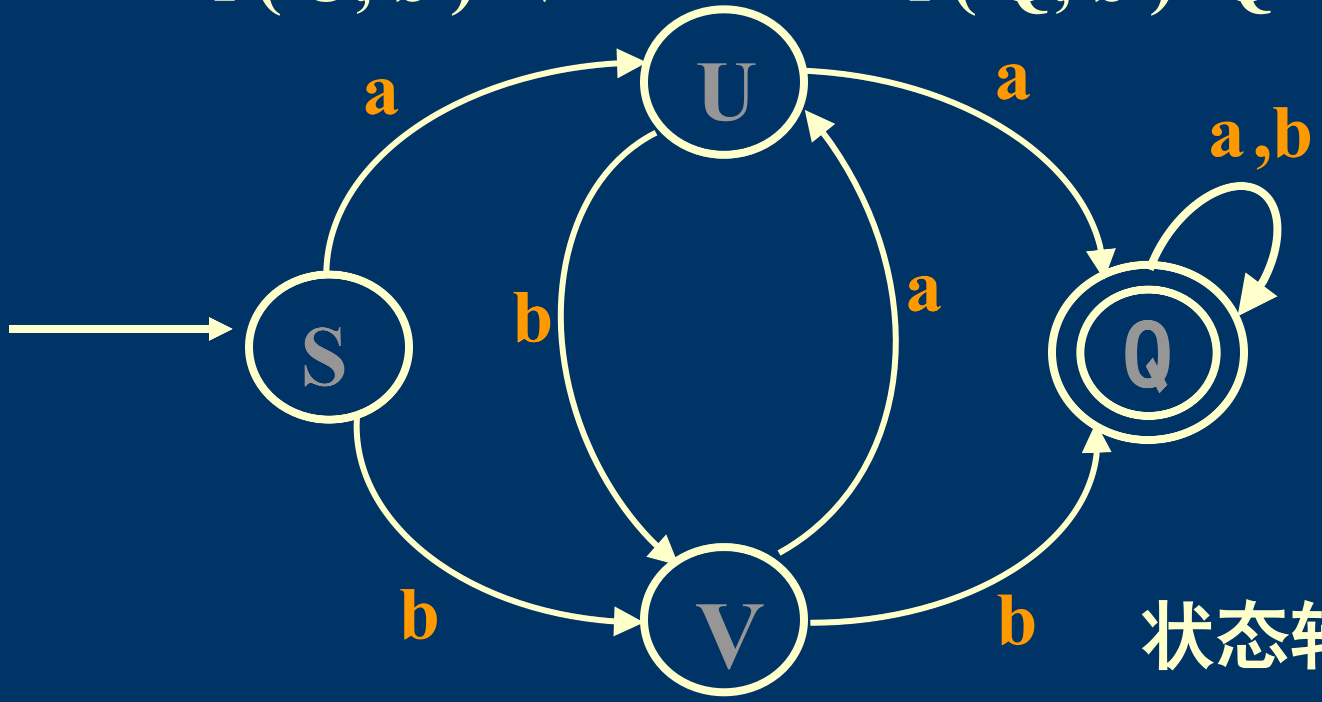
   1.2 终止状态由双圆圈围住的状态标识来表示（或由单圆圈围住的状态标识并标识以“-” 来表示）；

   1.3 开始状态由一个箭头指向的状态结点来表示(或标识以“+”来表示).

2. 状态转换函数用有向边来表示，若 $f(S_i,a)=S_k$，则由表示Si的状态节点到表示S_k的状态节点发出一条标识为a的有向边.

- DFA M=({S,U,V,Q}, {a,b}, f, S, {Q}), 其中 f 定义为：

f (S, a)=U	f (V, a)=U
f (S, b)=V	f (V, b)=Q
f (U, a)=Q	f (Q, a)=Q
f (U, b)=V	f (Q, b)=Q

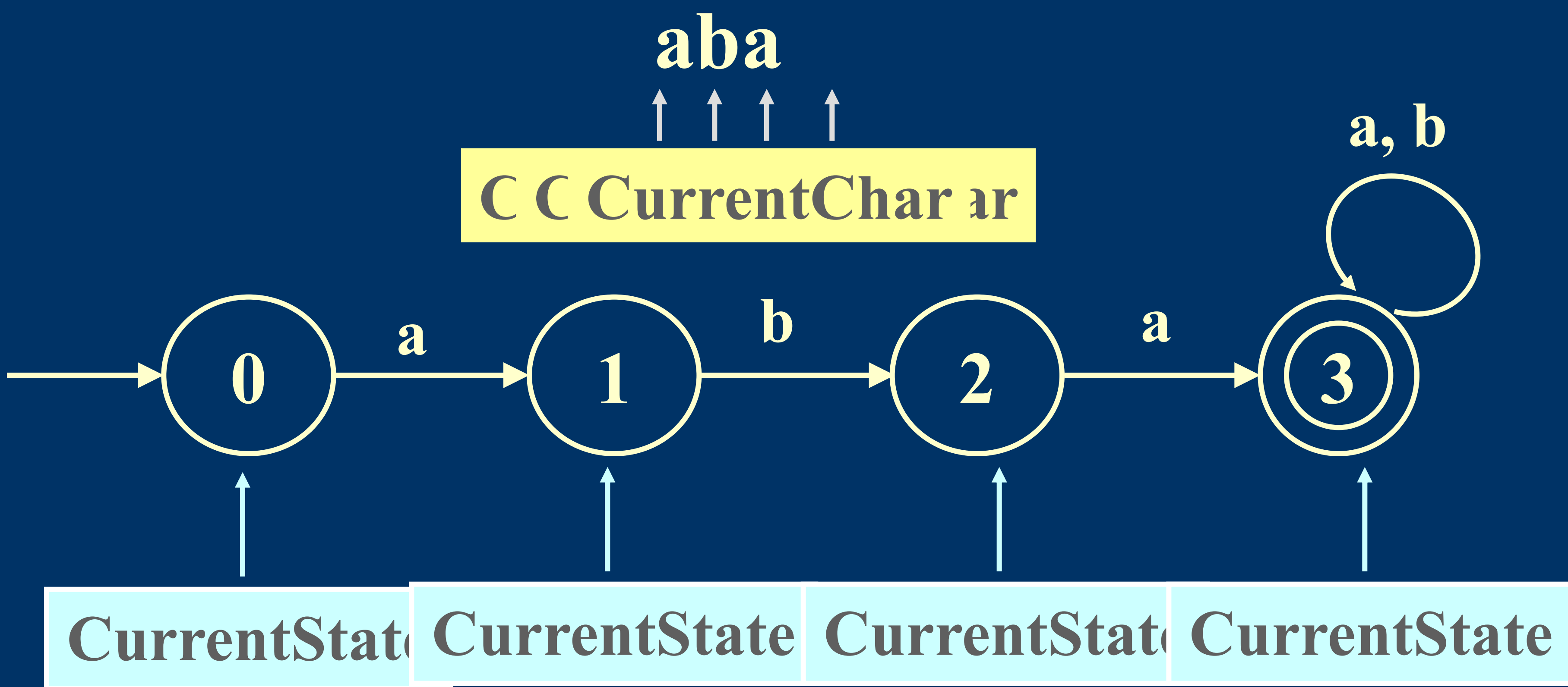


状态转换图

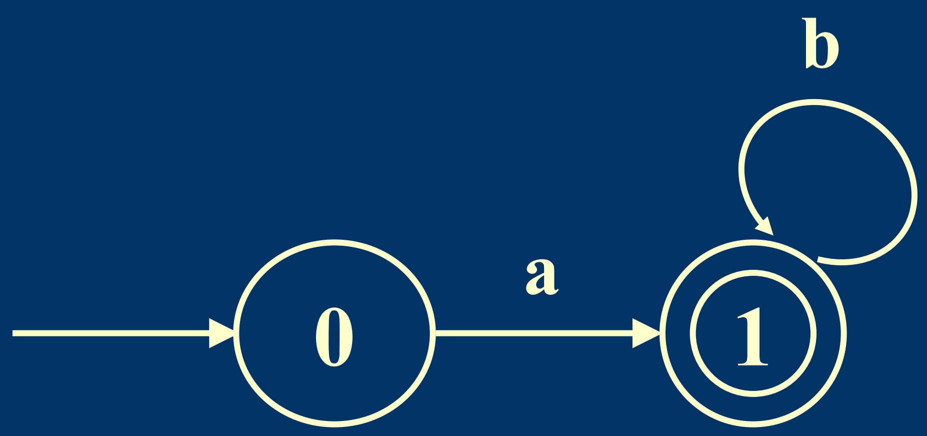
3. DFA接受的串

- 对于Σ^*中的任何字符串t，若存在一条从初始结点到某一终止结点的路径，且这条路上所有弧的标记符连接成的字符串等于t，则称t可为DFA M所接受（识别）。
- 若DFA M的初始状态同时又是终止状态，则空字符串可为DFA M所接受（识别）
- DFA M 所能接受的字符串的全体记为L(M).

例： $L(M_1)=\{aba, abaa, abab, abaab, \ldots\}$

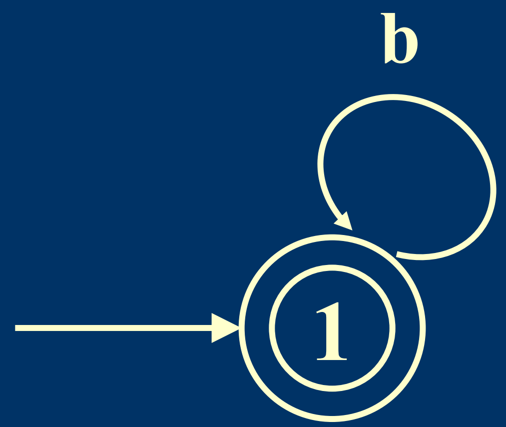


例：$L(M_2)=\{a, ab, abb, abbb, \ldots\}$



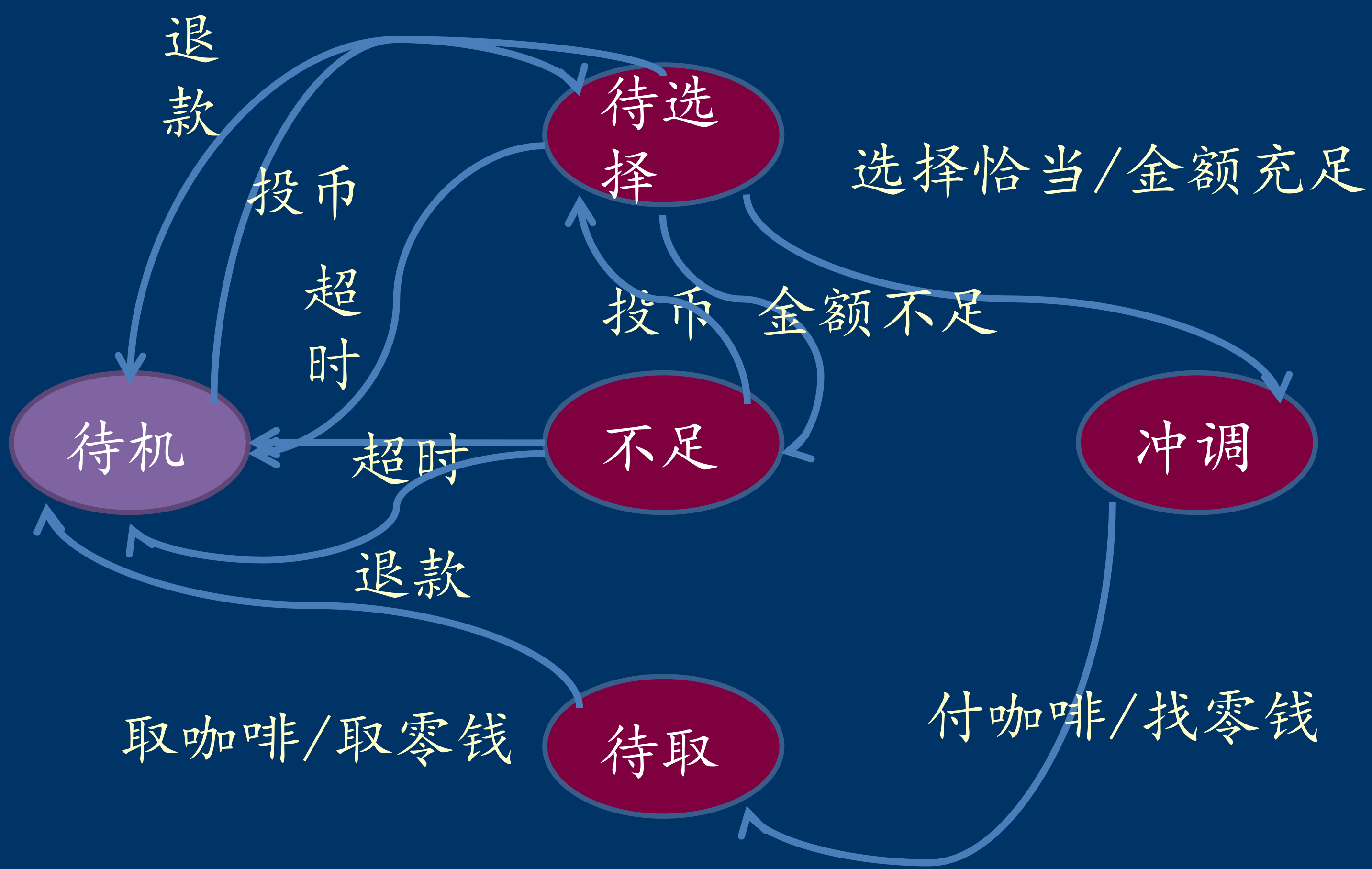
DFA M_2

例： $L(M_3)=\{\varepsilon, b, bb, bbb, \ldots\}$



DFA M_3

4. 应用实例



4. 应用实例

- 一道考研题：给定一个字符串，判断其是否合法

  1）这个字符串由0,1组成，由2作为结尾

  2）合法的要求是不能出现两个1串。

  如01012就是非法的，0112是合法的。

4. 应用实例

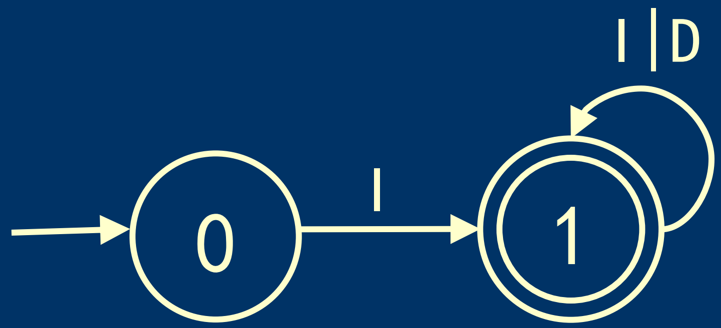
- 一个稍微需要技巧性的例子：

  用自动机描述被3整除的数。

5.DFA描述单词

- 标识符的描述

我们用l表示所有字母，D表示数字，则有：

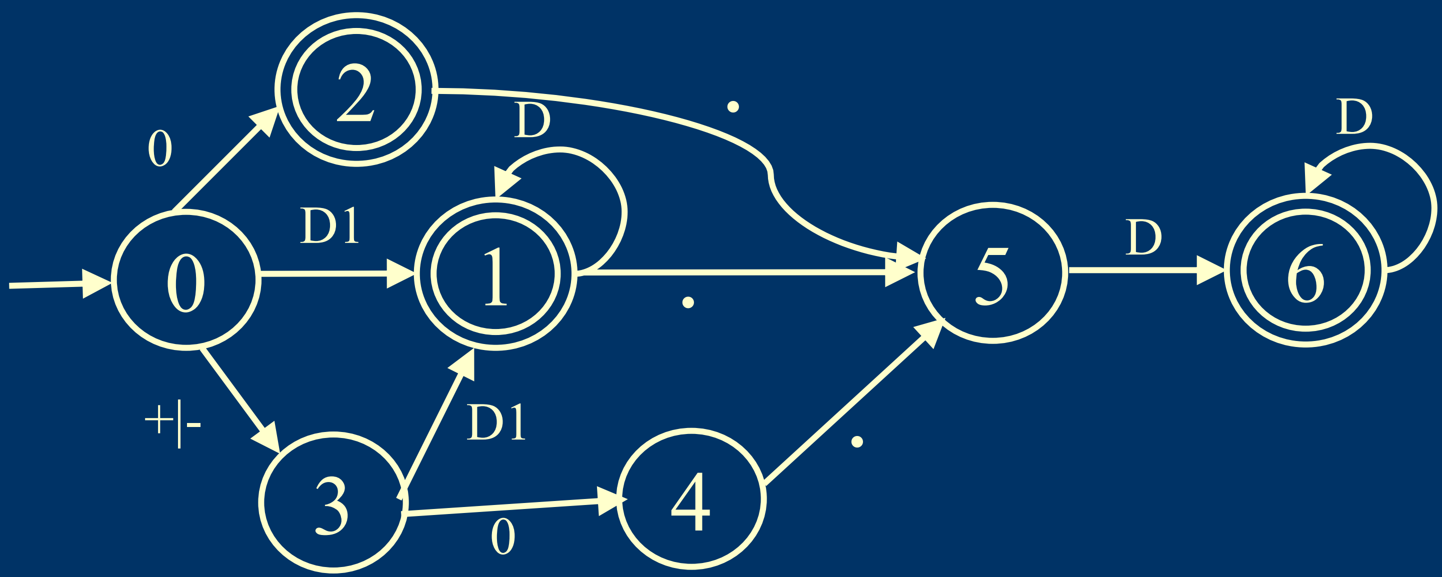


5.DFA描述单词

- 常数的描述

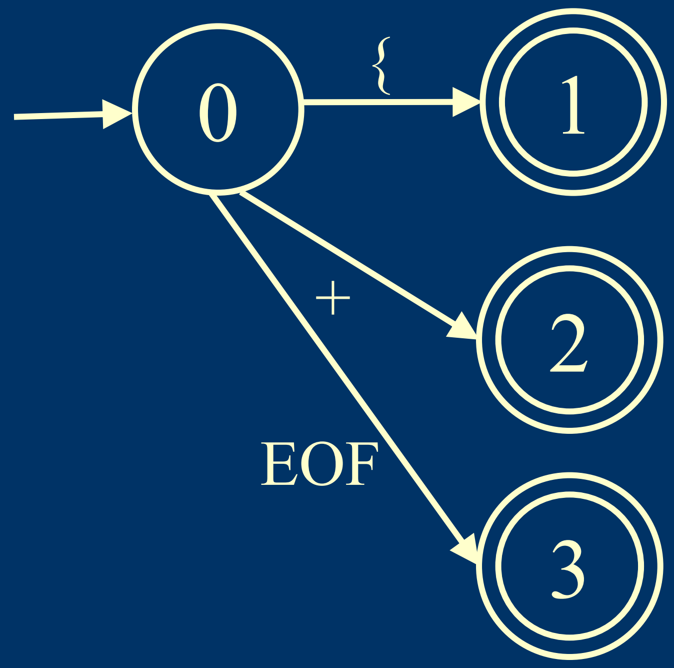
D1=1|2|3|...|9

无符号整数　　　带符号整数　　　实数



5.DFA描述单词

- 特殊符号



6. 自动机的实现

- 直接转换法

  每个状态对应一个带标号的switch语句

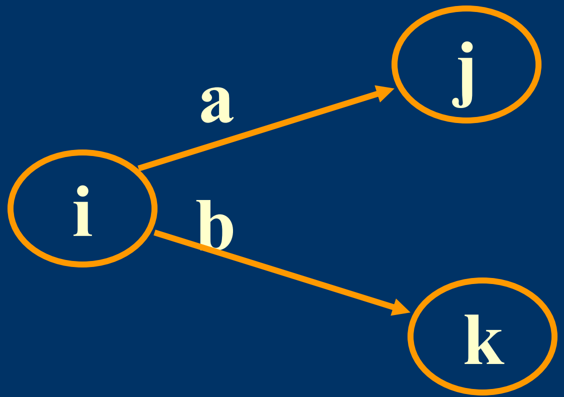
  转向边对应goto语句

- 状态转换矩阵

  自动机存储在矩阵中，状态比较多，每次都去查表，跳转。

直接转换法的实现

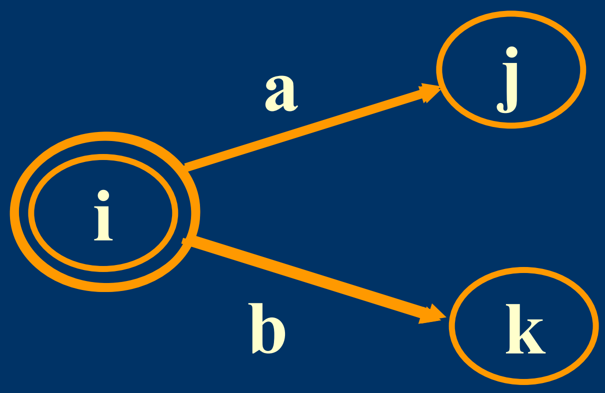
1.非终止状态对应的switch语句



L_i: switch (CurrentChar)
{ case ‘ a’ : goto L_j;
case ‘b’ : goto L_k;
default : Error( );
}

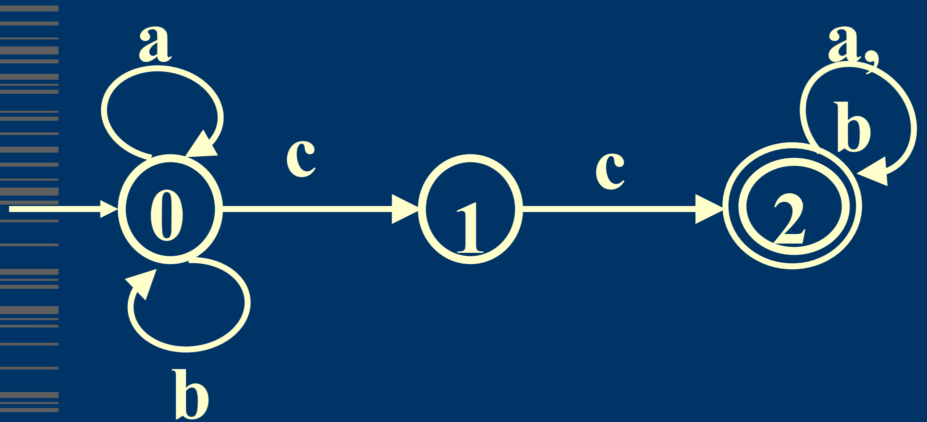
直接转换法的实现

2. 终止状态对应的switch语句




```
Li: switch ( CurrentChar )
 { case ‘ a’      :   goto Lj;
   case  ‘b’      :   goto Lk;
   case ‘Eof’     :   Accept;
    default       :  Error( );
}
```

直接转换法实现的例子



L_0: switch (CurrentChar)

```
{ case ' c'    :   goto L1;
  case ' a'    :   goto L0;
  case 'b'     :   goto L0;
  default      :   Error( );
}
```

状态转换矩阵的实现

1. 当前状态State置为初始状态;

2. 读一个字符 → CurrentChar;

3. 如果CurrentChar≠Eof并且

   T(State,CurrentChar)≠error,

   则当前状态转为新的状态:

   State = T(State,Current),

   读下一字符 → CurrentChar ;

   重复第3步工作.

4. 如果当前字符为Eof并且当前状态属于终止状态，则接受当前字符串，程序结束；否则报错.

7. 注意的问题

- 特殊的状态“死状态”
- 缺省状态
- 缺省转换函数

8. 自动机等价

- 对于两个DFA M1和M2，若有L(M1)=L(M2)则称M1和M2等价

习题

- 设计一个DFA，使其能接受被3整除的二进制数