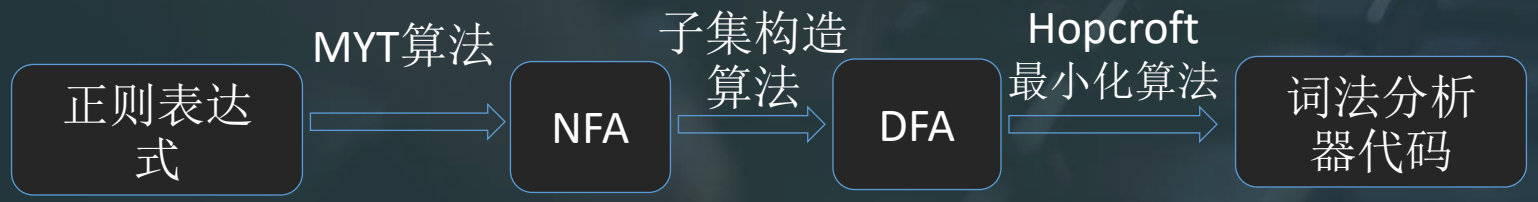
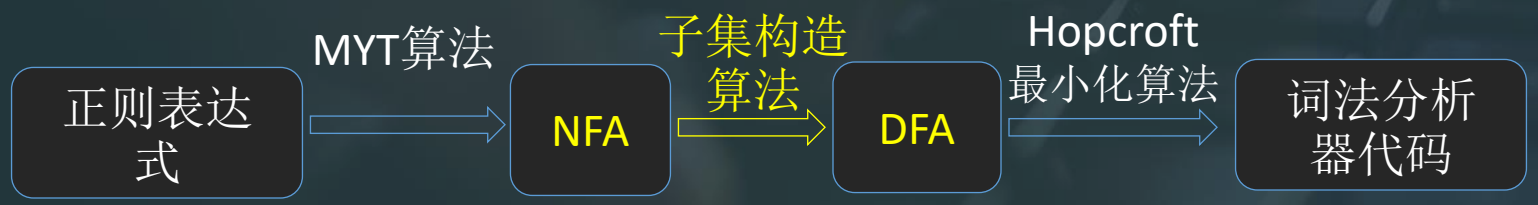
从正则表达式到DFA

2.2.2 从NFA到DFA的转换



2.2.2 从NFA到DFA的转换



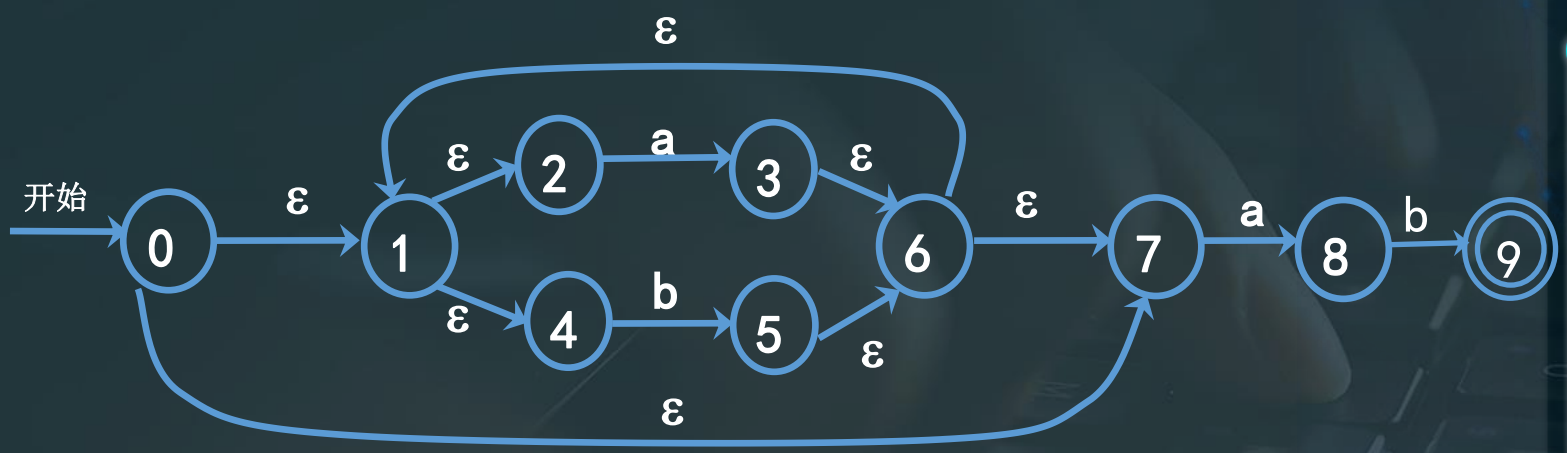
2.2.2 从NFA到DFA的转换

确定的有限自动机 (Deterministic Finite Automata, DFA)

- 由以下几个部分组成
  - 有限的状态集合S
  - 输入符号集合
  - 转换函数$move : S \quad (\quad \{ \}) \quad P(S)$
  - 状态 S_0 是唯一的开始状态
  - $F \quad S$ 是接受状态集合
  - 任何状态下都没有ε转换
  - 一个符号标记离开同一状态只有一条边

2.2.2 从NFA到DFA的转换

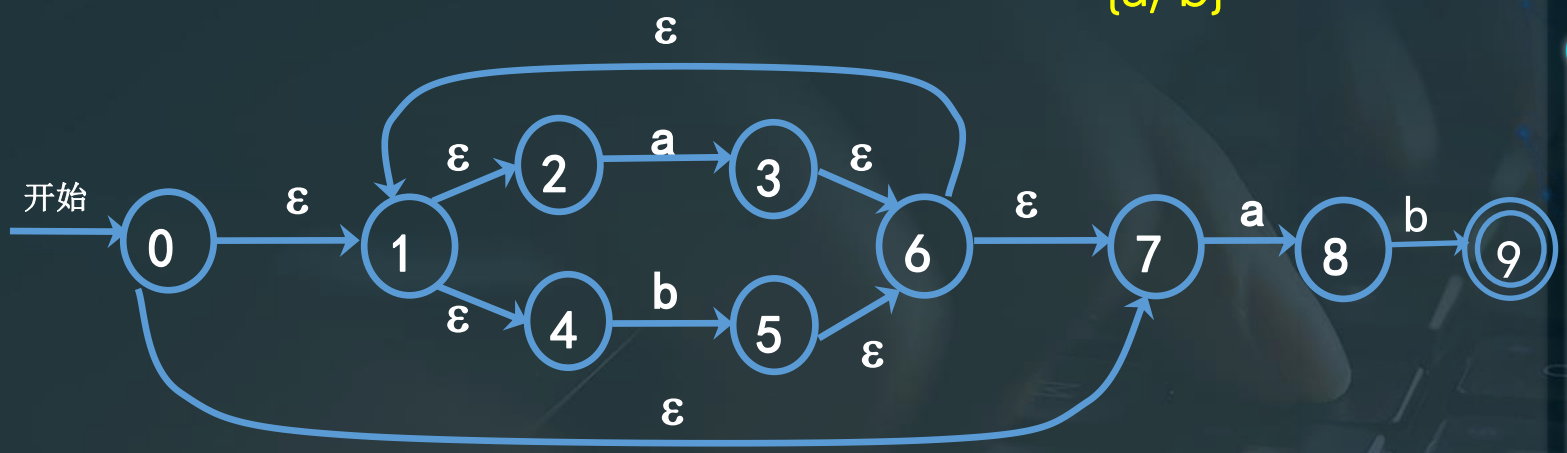
将(a|b)*ab的NFA转换为DFA



2.2.2 从NFA到DFA的转换

将(a|b)*ab的NFA转换为DFA

= {a, b}

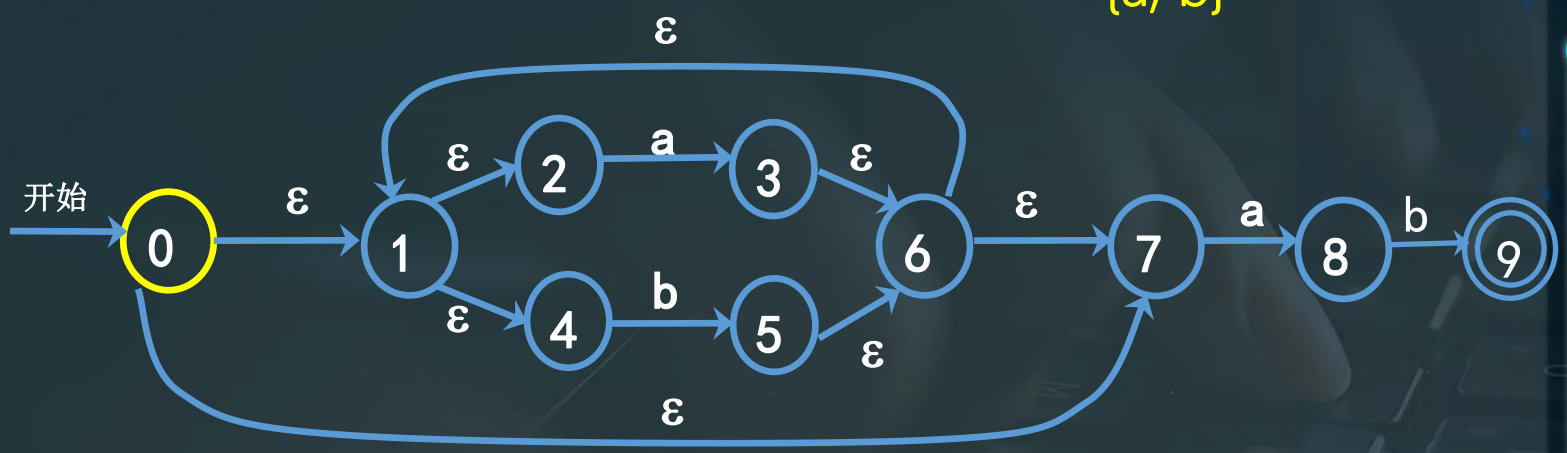


{0} —ε→

2.2.2 从NFA到DFA的转换

将(a|b)*ab的NFA转换为DFA

= {a, b}

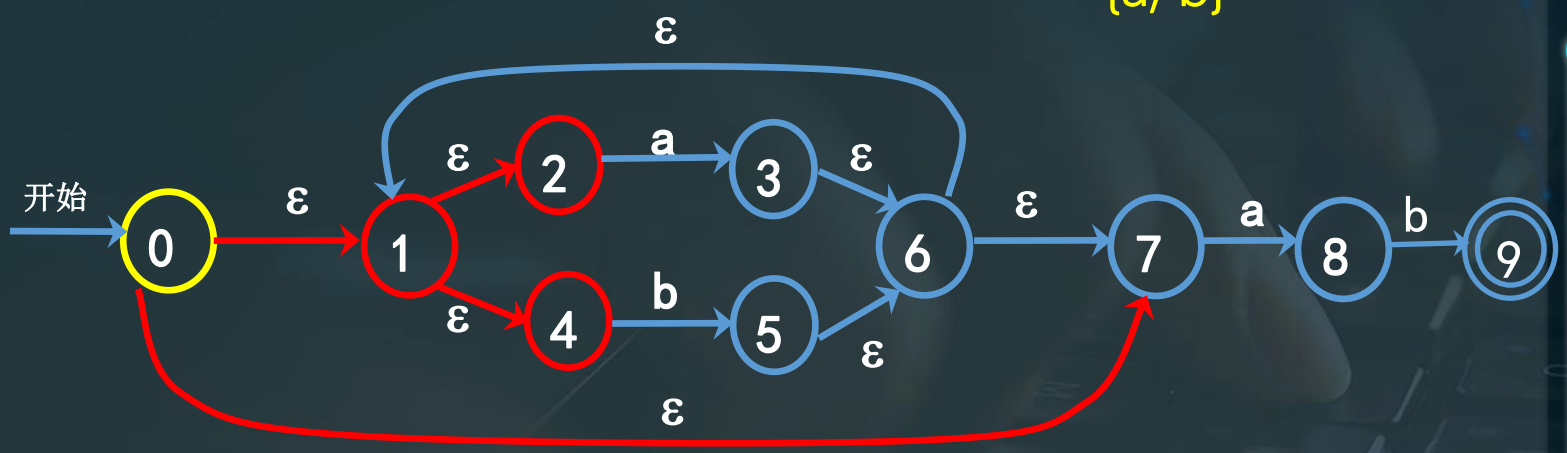


{0} —ε→

2.2.2 从NFA到DFA的转换

将(a|b)*ab的NFA转换为DFA

= {a, b}

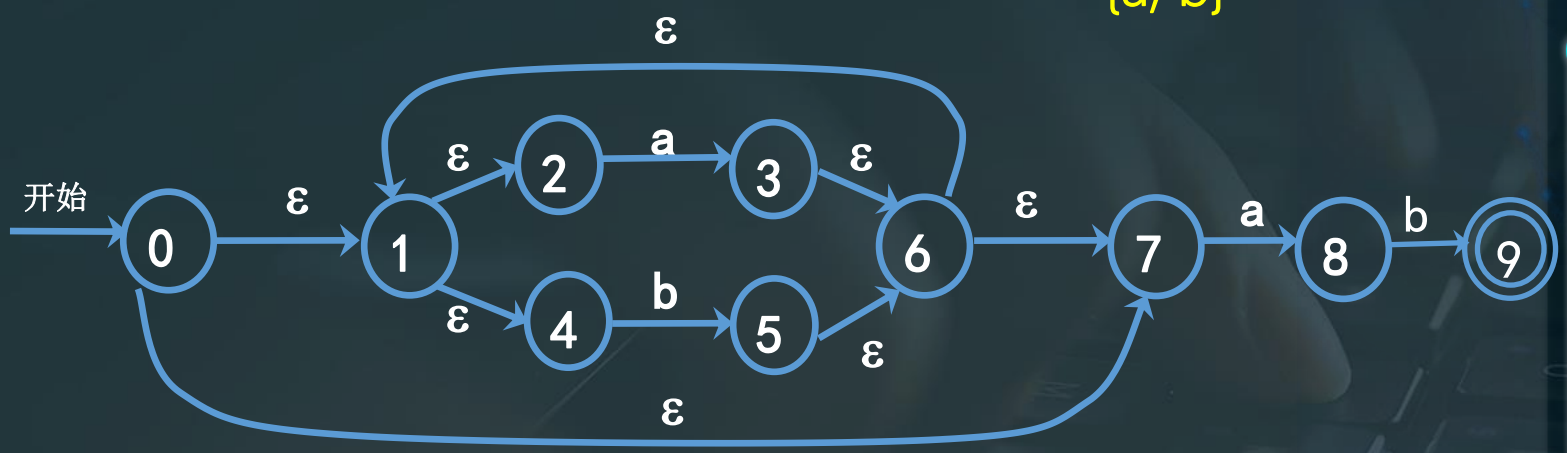


{0} $\xrightarrow{\varepsilon}$ {0, 1, 2, 4, 7}　只用 ε 转换能到达的NFA状态集合

2.2.2 从NFA到DFA的转换

将(a|b)*ab的NFA转换为DFA

= {a, b}



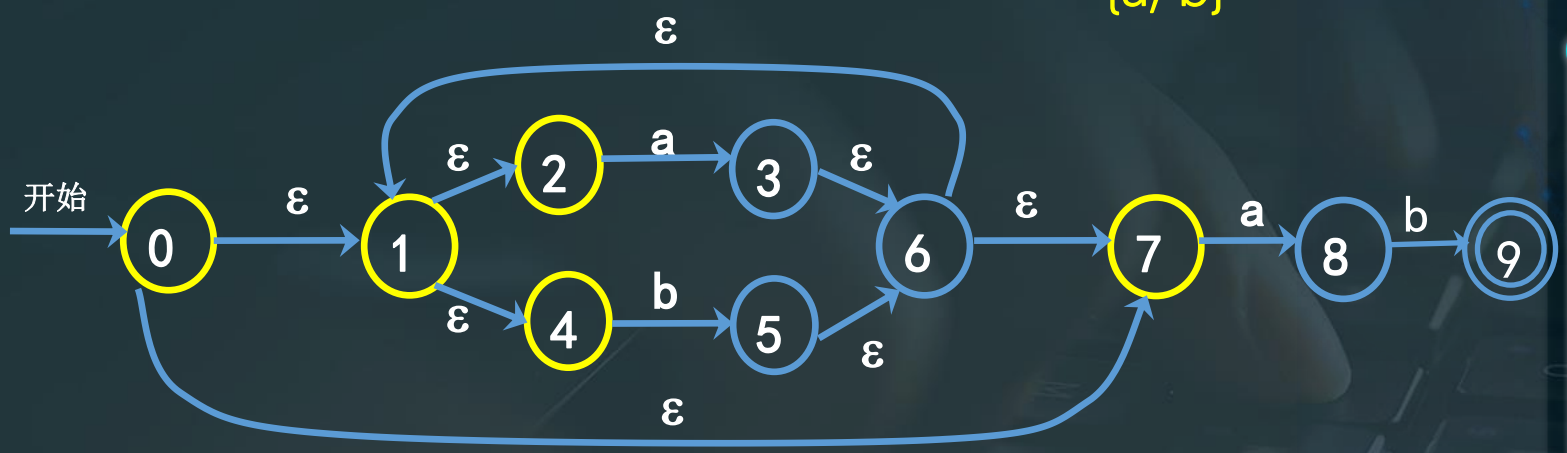
{0} $\xrightarrow{\varepsilon}$ {0, 1, 2, 4, 7}　只用ε转换能到达的NFA状态集合

{0, 1, 2, 4, 7} $\xrightarrow{a}$

2.2.2 从NFA到DFA的转换

将(a|b)*ab的NFA转换为DFA

= {a, b}



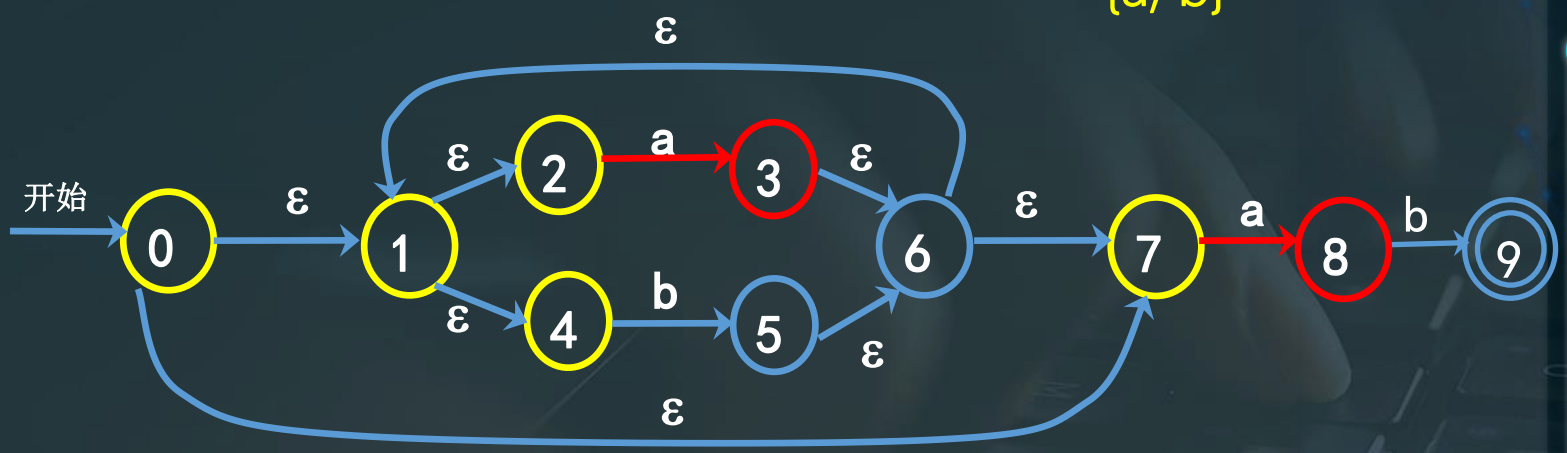
{0} $\xrightarrow{\varepsilon}$ {0, 1, 2, 4, 7}　只用ε转换能到达的NFA状态集合

{0, 1, 2, 4, 7} $\xrightarrow{a}$

2.2.2 从NFA到DFA的转换

将(a|b)*ab的NFA转换为DFA

= {a, b}



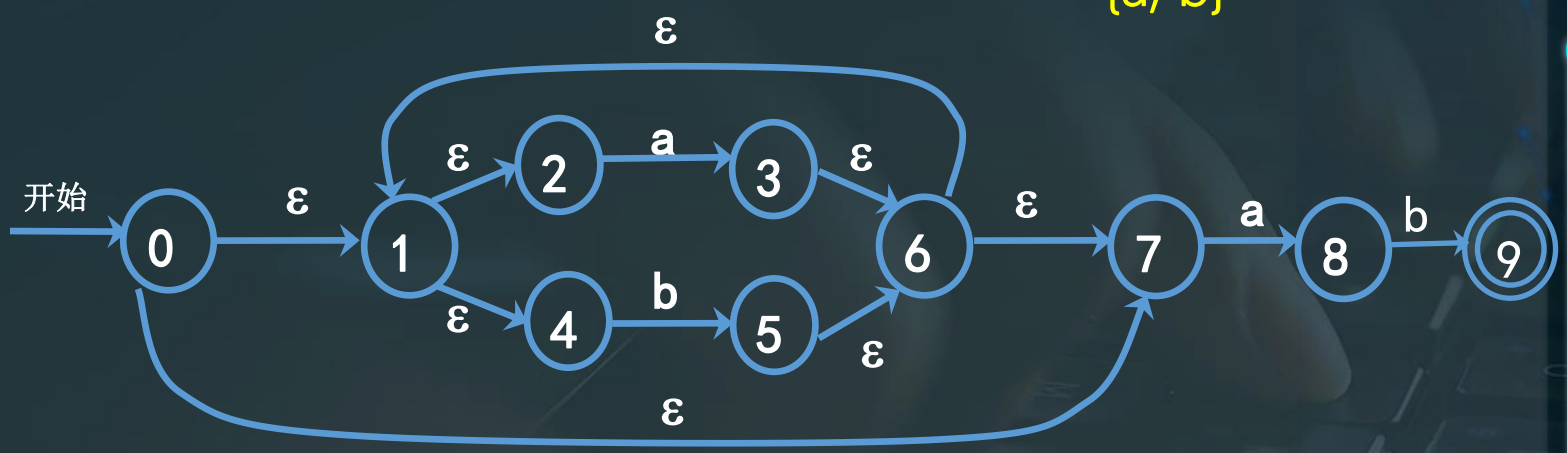
{0} $\xrightarrow{\varepsilon}$ {0, 1, 2, 4, 7}　只用ε转换能到达的NFA状态集合

{0, 1, 2, 4, 7} $\xrightarrow{a}$

2.2.2 从NFA到DFA的转换

将(a|b)*ab的NFA转换为DFA

= {a, b}



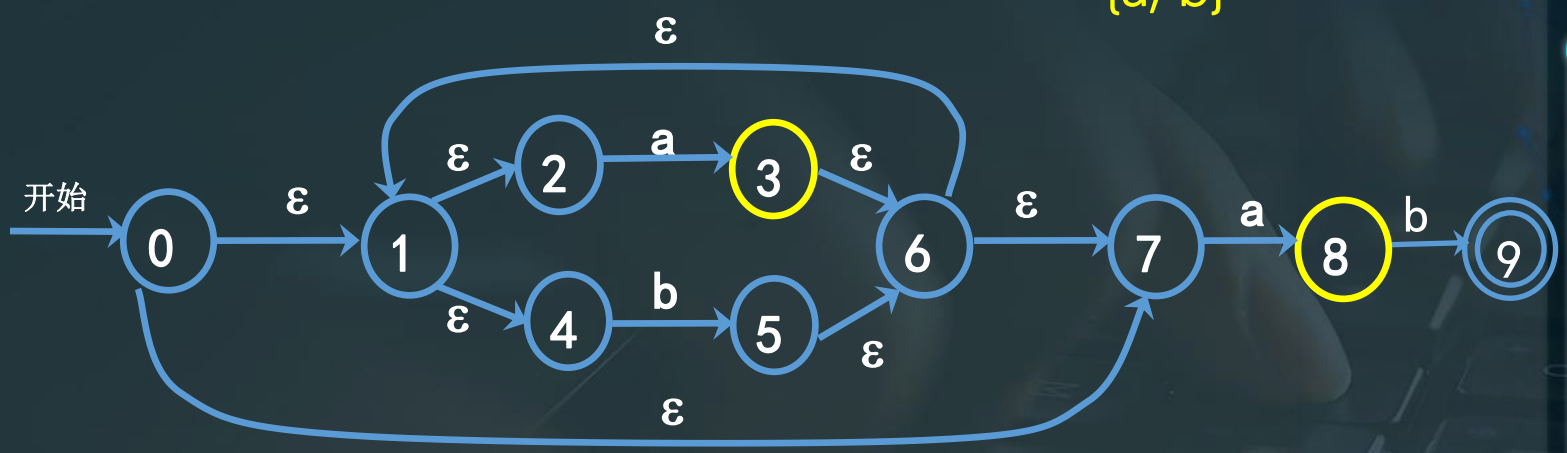
{0} $\xrightarrow{\varepsilon}$ {0, 1, 2, 4, 7}　只用ε转换能到达的NFA状态集合

{0, 1, 2, 4, 7} $\xrightarrow{a}$ {3, 8}

2.2.2 从NFA到DFA的转换

将(a|b)*ab的NFA转换为DFA

= {a, b}



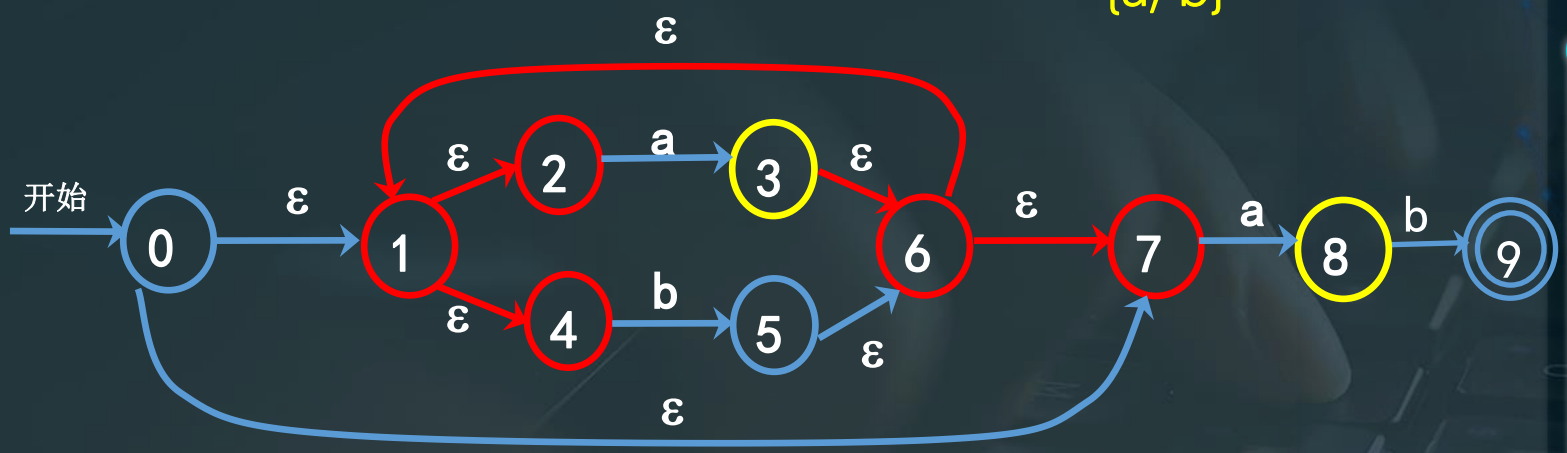
{0} $\xrightarrow{\varepsilon}$ {0, 1, 2, 4, 7}　只用 ε 转换能到达的NFA状态集合

{0, 1, 2, 4, 7} $\xrightarrow{a}$ {3, 8} $\xrightarrow{\varepsilon}$

2.2.2 从NFA到DFA的转换

将(a|b)*ab的NFA转换为DFA

= {a, b}



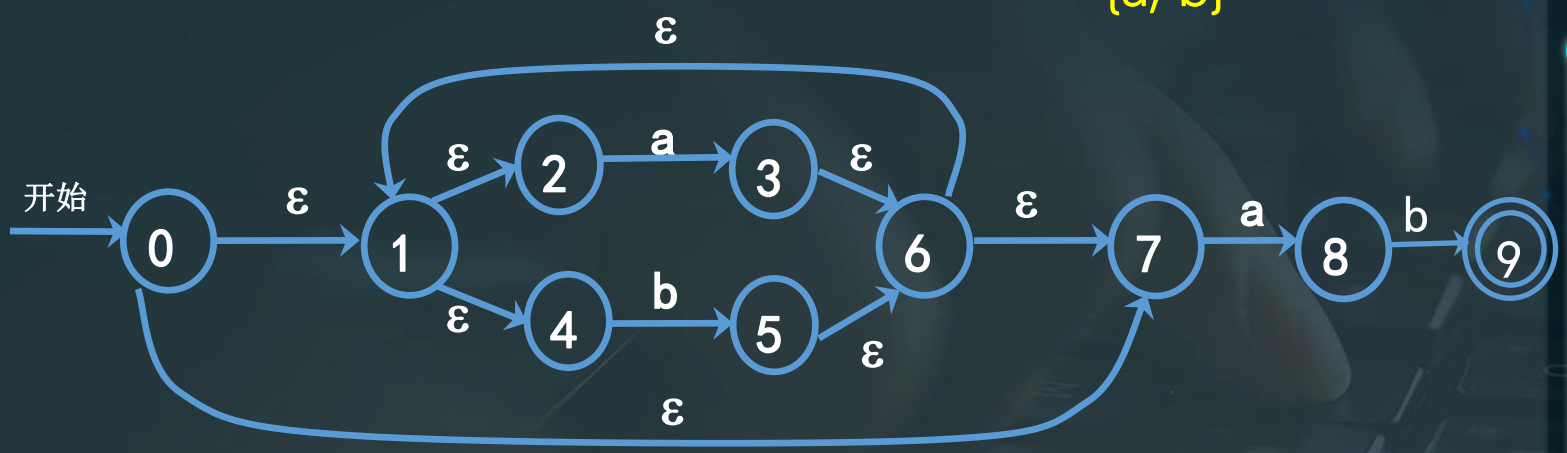
$\{0\} \xrightarrow{\varepsilon} \{0, 1, 2, 4, 7\}$ 只用ε转换能到达的NFA状态集合

$\{0, 1, 2, 4, 7\} \xrightarrow{a} \{3, 8\} \xrightarrow{\varepsilon}$

2.2.2 从NFA到DFA的转换

将(a|b)*ab的NFA转换为DFA

= {a, b}



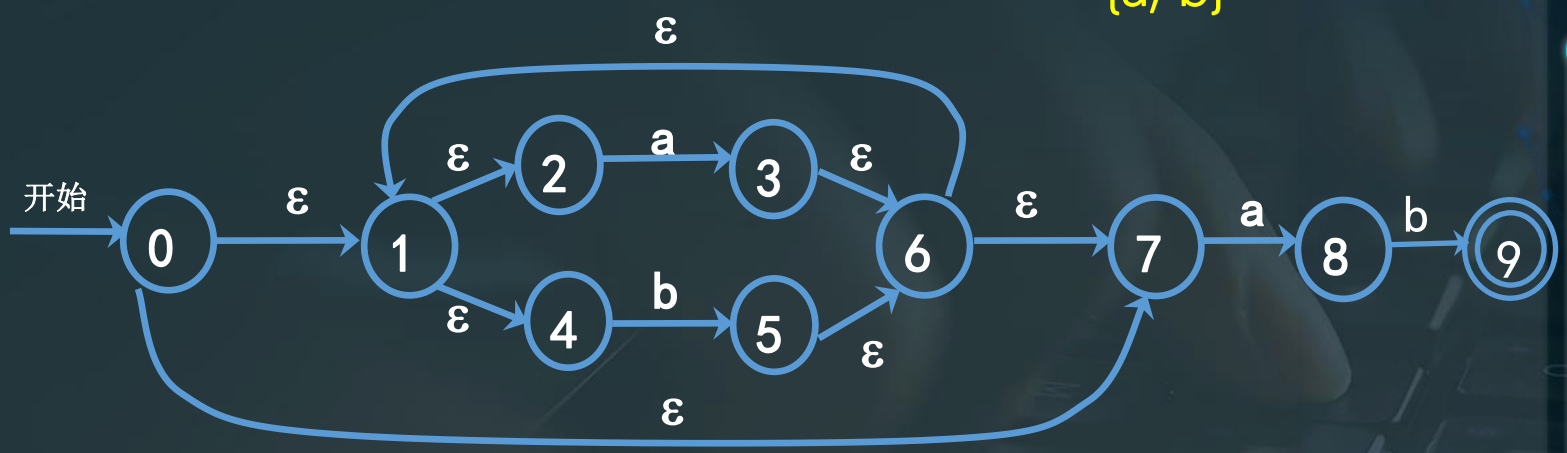
{0} $\xrightarrow{\varepsilon}$ {0, 1, 2, 4, 7}　只用ε转换能到达的NFA状态集合

{0, 1, 2, 4, 7} $\xrightarrow{a}$ {3, 8} $\xrightarrow{\varepsilon}$ {3, 8, 6, 1, 7, 2, 4}

2.2.2 从NFA到DFA的转换

将(a|b)*ab的NFA转换为DFA

= {a, b}



{0} $\xrightarrow{\varepsilon}$ {0, 1, 2, 4, 7}　只用ε转换能到达的NFA状态集合

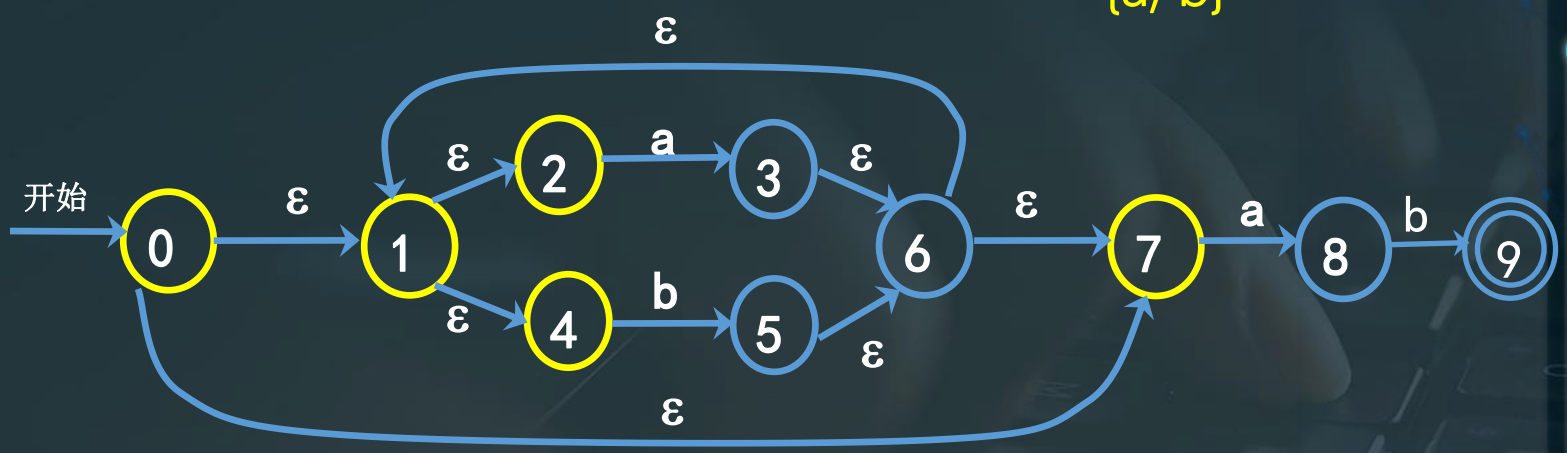
{0, 1, 2, 4, 7} $\xrightarrow{a}$ {3, 8} $\xrightarrow{\varepsilon}$ {3, 8, 6, 1, 7, 2, 4}

{0, 1, 2, 4, 7} $\xrightarrow{b}$

2.2.2 从NFA到DFA的转换

将(a|b)*ab的NFA转换为DFA

= {a, b}



$\{0\} \xrightarrow{\varepsilon} \{0, 1, 2, 4, 7\}$ 只用ε转换能到达的NFA状态集合

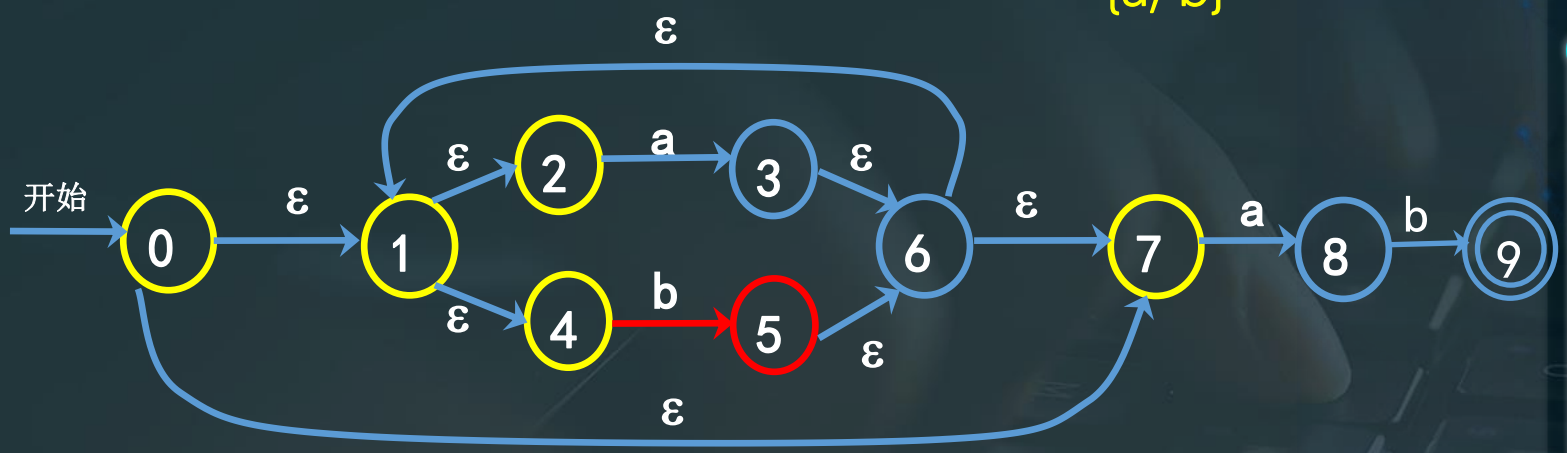
$\{0, 1, 2, 4, 7\} \xrightarrow{a} \{3, 8\} \xrightarrow{\varepsilon} \{3, 8, 6, 1, 7, 2, 4\}$

$\{0, 1, 2, 4, 7\} \xrightarrow{b}$

2.2.2 从NFA到DFA的转换

将(a|b)*ab的NFA转换为DFA

= {a, b}



$\{0\} \xrightarrow{\varepsilon} \{0, 1, 2, 4, 7\}$ 只用ε转换能到达的NFA状态集合

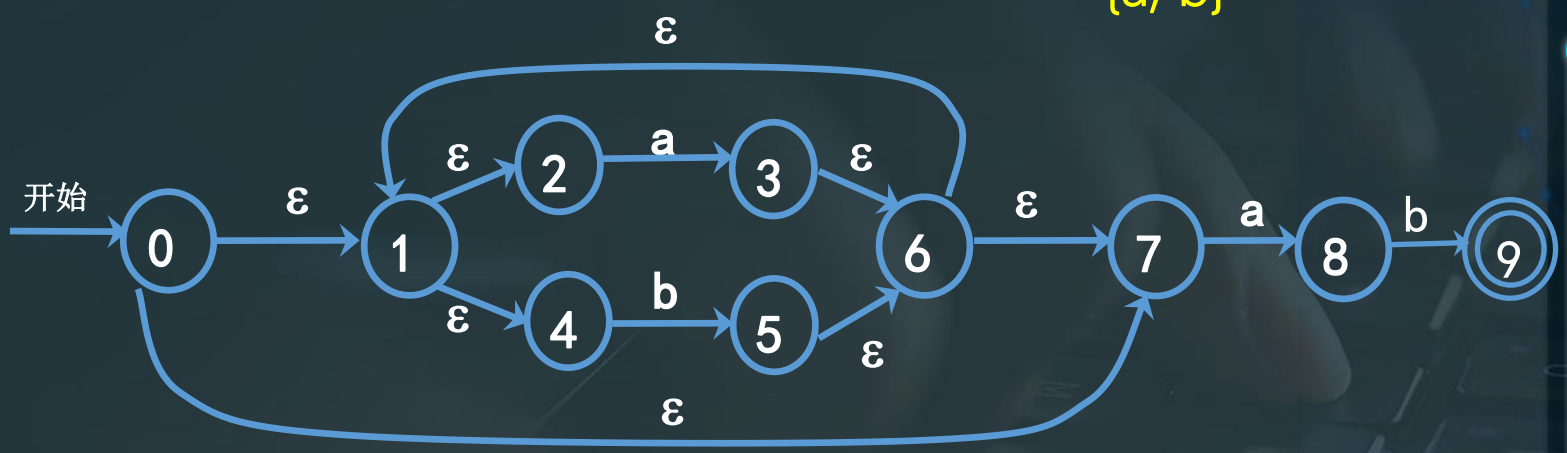
$\{0, 1, 2, 4, 7\} \xrightarrow{a} \{3, 8\} \xrightarrow{\varepsilon} \{3, 8, 6, 1, 7, 2, 4\}$

$\{0, 1, 2, 4, 7\} \xrightarrow{b}$

2.2.2 从NFA到DFA的转换

将(a|b)*ab的NFA转换为DFA

= {a, b}



{0} $\xrightarrow{\varepsilon}$ {0, 1, 2, 4, 7}　只用ε转换能到达的NFA状态集合

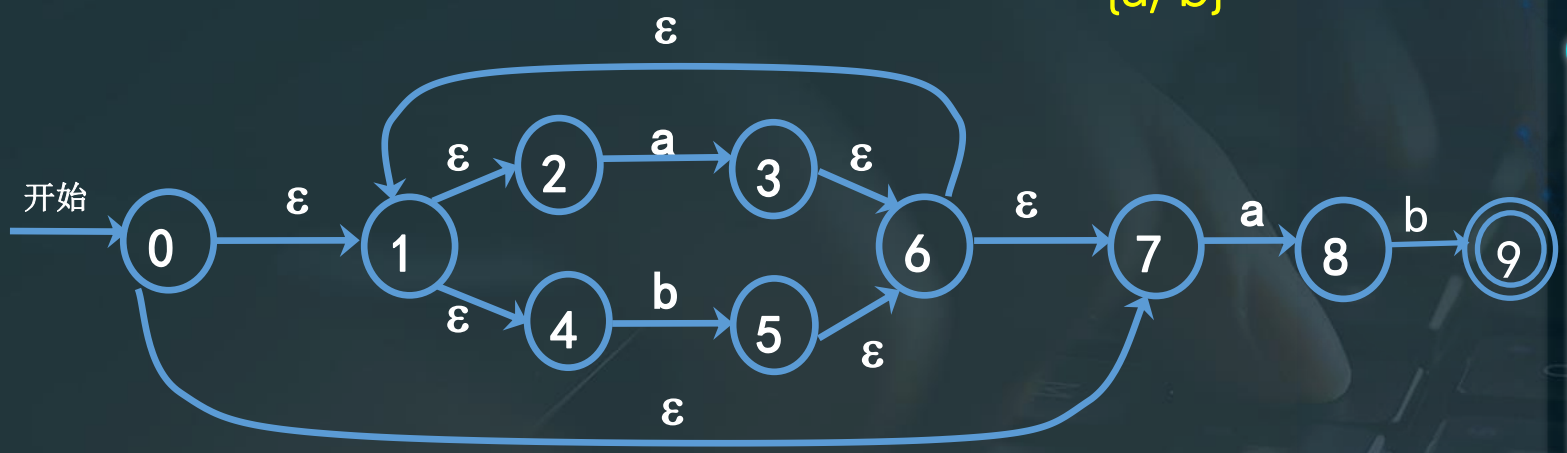
{0, 1, 2, 4, 7} $\xrightarrow{a}$ {3, 8} $\xrightarrow{\varepsilon}$ {3, 8, 6, 1, 7, 2, 4}

{0, 1, 2, 4, 7} $\xrightarrow{b}$ {5}

2.2.2 从NFA到DFA的转换

将(a|b)*ab的NFA转换为DFA

= {a, b}



{0} $\xrightarrow{\varepsilon}$ {0, 1, 2, 4, 7}　只用ε转换能到达的NFA状态集合

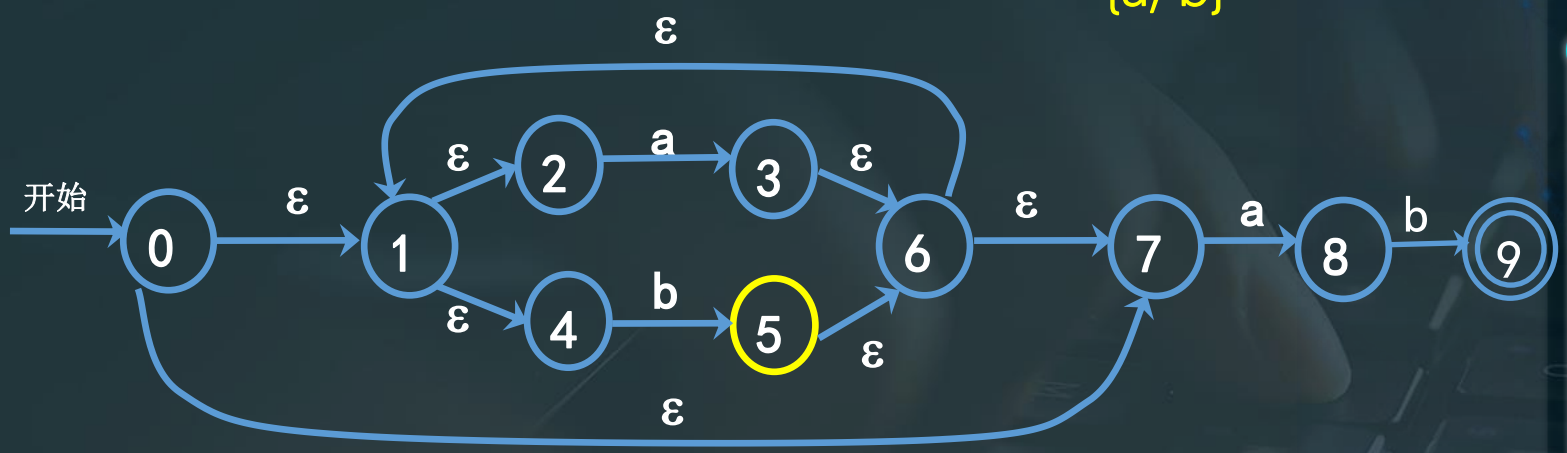
{0, 1, 2, 4, 7} $\xrightarrow{a}$ {3, 8} $\xrightarrow{\varepsilon}$ {3, 8, 6, 1, 7, 2, 4}

{0, 1, 2, 4, 7} $\xrightarrow{b}$ {5} $\xrightarrow{\varepsilon}$

2.2.2 从NFA到DFA的转换

将(a|b)*ab的NFA转换为DFA

= {a, b}



{0} $\xrightarrow{\varepsilon}$ {0, 1, 2, 4, 7}　只用 ε 转换能到达的NFA状态集合

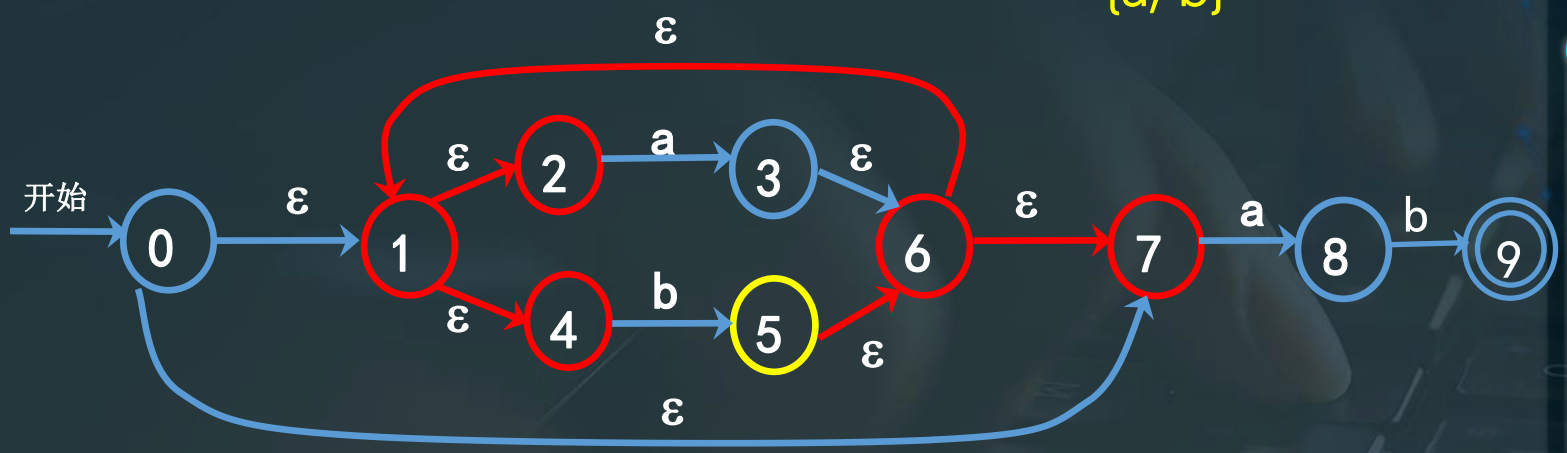
{0, 1, 2, 4, 7} $\xrightarrow{a}$ {3, 8} $\xrightarrow{\varepsilon}$ {3, 8, 6, 1, 7, 2, 4}

{0, 1, 2, 4, 7} $\xrightarrow{b}$ {5} $\xrightarrow{\varepsilon}$

2.2.2 从NFA到DFA的转换

将(a|b)*ab的NFA转换为DFA

= {a, b}



{0} $\xrightarrow{\varepsilon}$ {0, 1, 2, 4, 7}　只用ε转换能到达的NFA状态集合

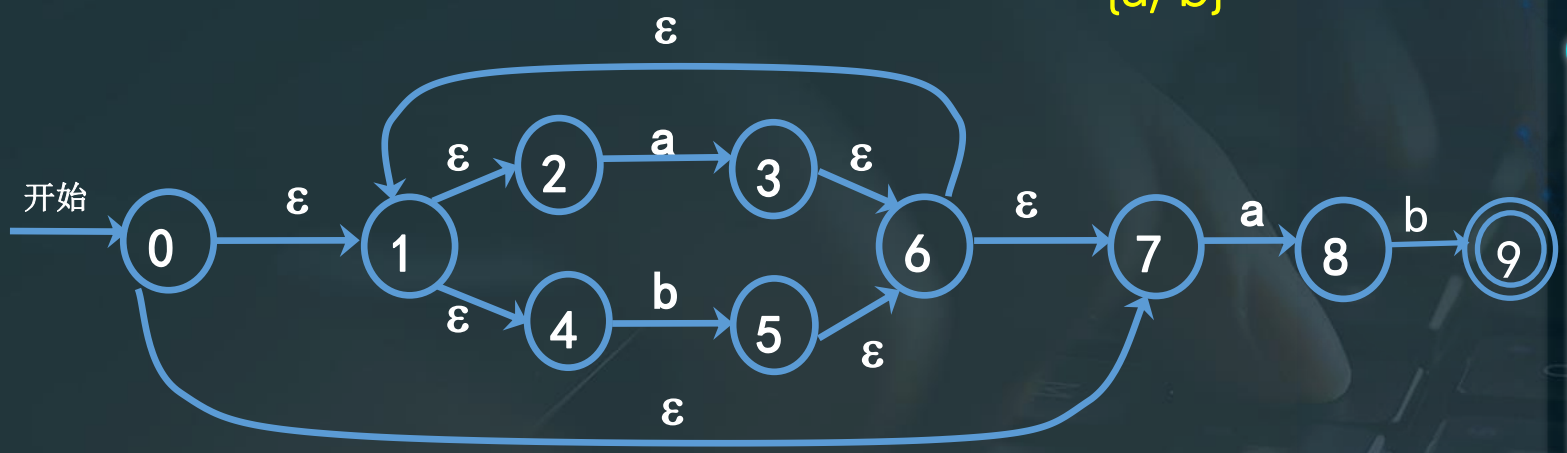
{0, 1, 2, 4, 7} $\xrightarrow{a}$ {3, 8} $\xrightarrow{\varepsilon}$ {3, 8, 6, 1, 7, 2, 4}

{0, 1, 2, 4, 7} $\xrightarrow{b}$ {5} $\xrightarrow{\varepsilon}$

2.2.2 从NFA到DFA的转换

将(a|b)*ab的NFA转换为DFA

= {a, b}



{0} $\xrightarrow{\varepsilon}$ {0, 1, 2, 4, 7}　只用ε转换能到达的NFA状态集合

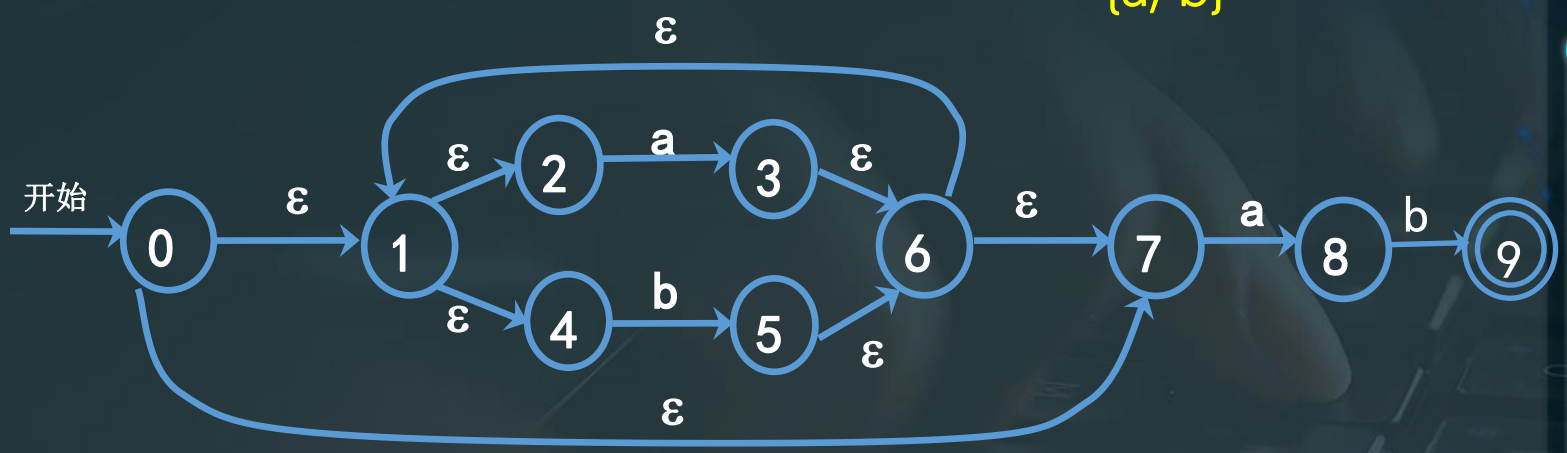
{0, 1, 2, 4, 7} $\xrightarrow{a}$ {3, 8} $\xrightarrow{\varepsilon}$ {3, 8, 6, 1, 7, 2, 4}

{0, 1, 2, 4, 7} $\xrightarrow{b}$ {5} $\xrightarrow{\varepsilon}$ {5, 6, 1, 7, 2, 4}

2.2.2 从NFA到DFA的转换

将(a|b)*ab的NFA转换为DFA

= {a, b}



A

$\{0\} \xrightarrow{\varepsilon} \{0, 1, 2, 4, 7\}$ 只用ε转换能到达的NFA状态集合

A B

$\{0, 1, 2, 4, 7\} \xrightarrow{a} \{3, 8\} \xrightarrow{\varepsilon} \{3, 8, 6, 1, 7, 2, 4\}$

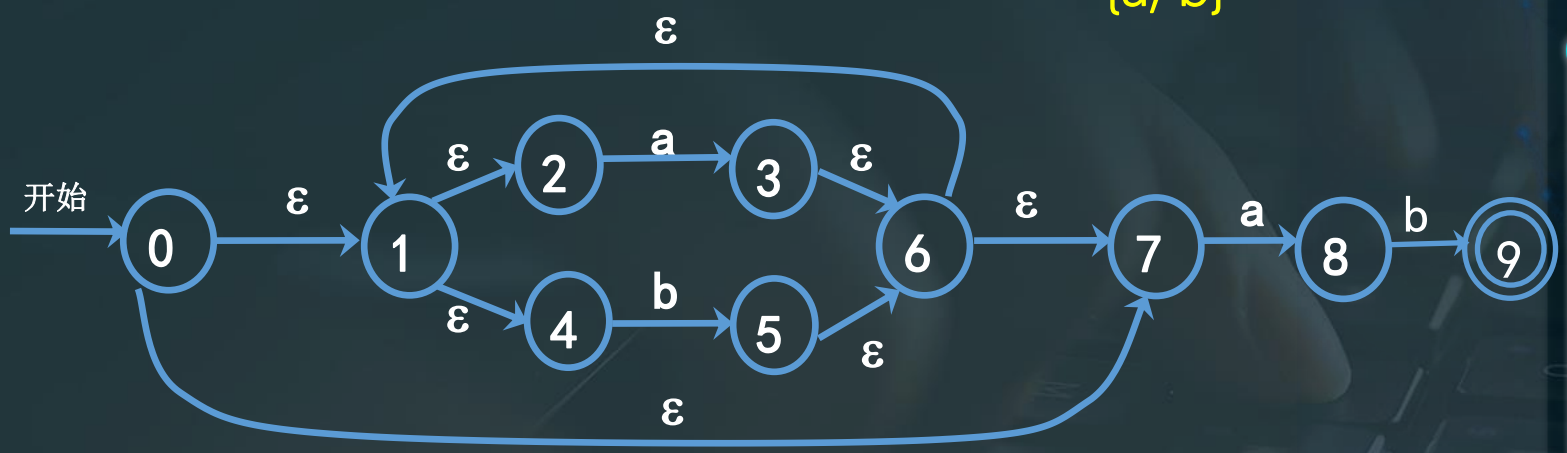
A C

$\{0, 1, 2, 4, 7\} \xrightarrow{b} \{5\} \xrightarrow{\varepsilon} \{5, 6, 1, 7, 2, 4\}$

2.2.2 从NFA到DFA的转换

将(a|b)*ab的NFA转换为DFA

= {a, b}



A = {0, 1, 2, 4, 7}

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

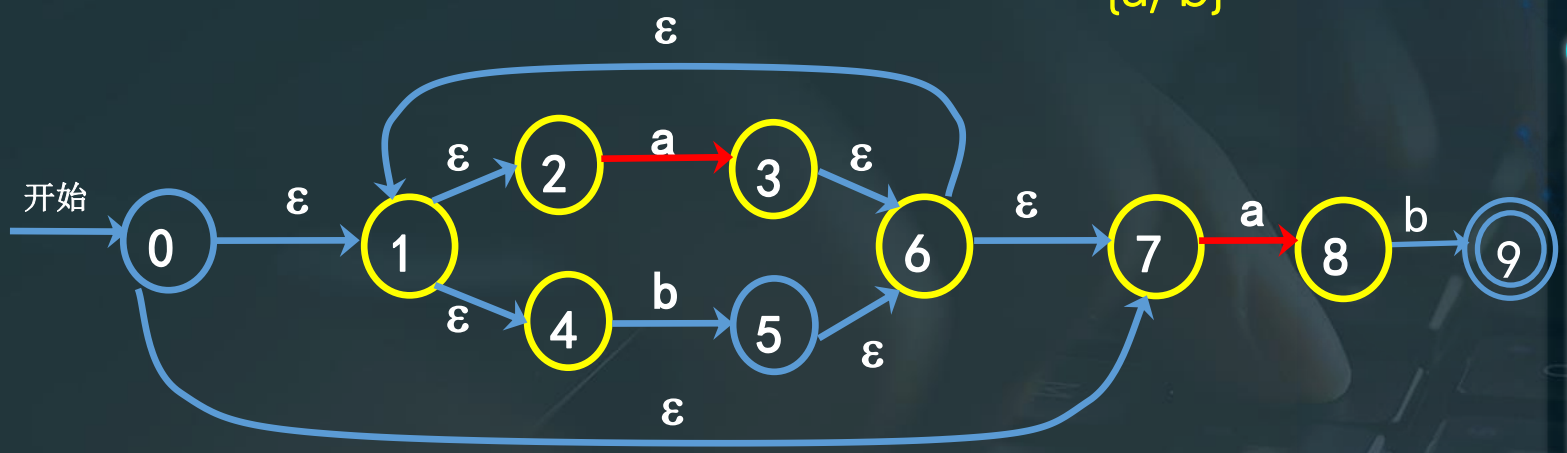
C = {1, 2, 4, 5, 6, 7}

状态	a	b
A	B	C

2.2.2 从NFA到DFA的转换

将(a|b)*ab的NFA转换为DFA

= {a, b}



A = {0, 1, 2, 4, 7}

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

C = {1, 2, 4, 5, 6, 7}

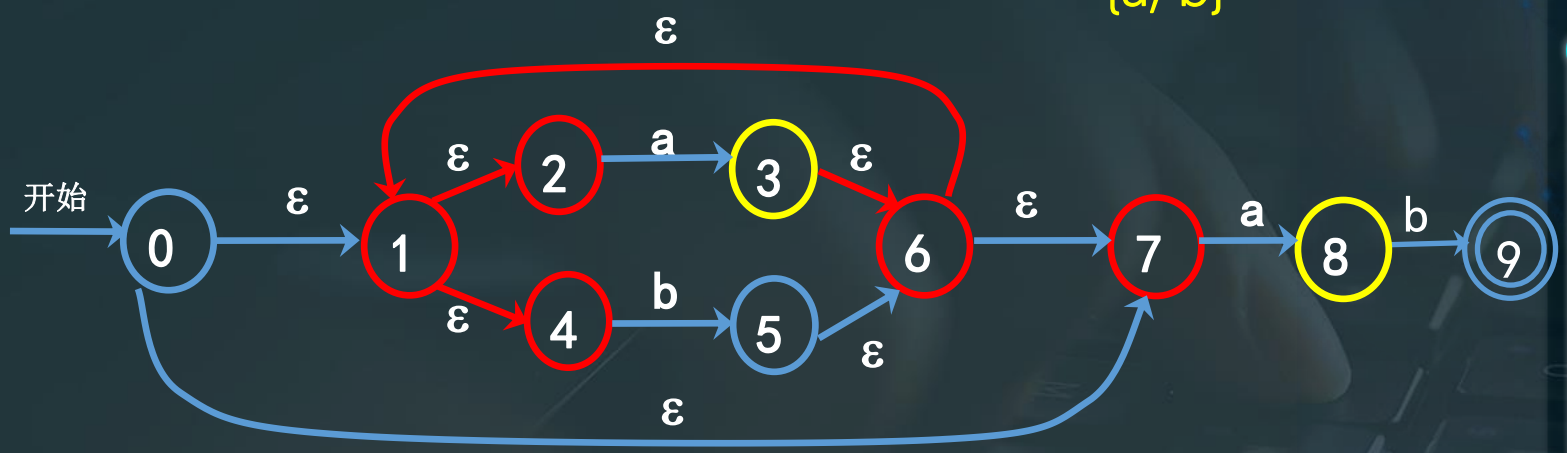
B $\xrightarrow{a}$ {3, 8}

状态	a	b
A	B	C
B		

2.2.2 从NFA到DFA的转换

将(a|b)*ab的NFA转换为DFA

= {a, b}



A = {0, 1, 2, 4, 7}

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

C = {1, 2, 4, 5, 6, 7}

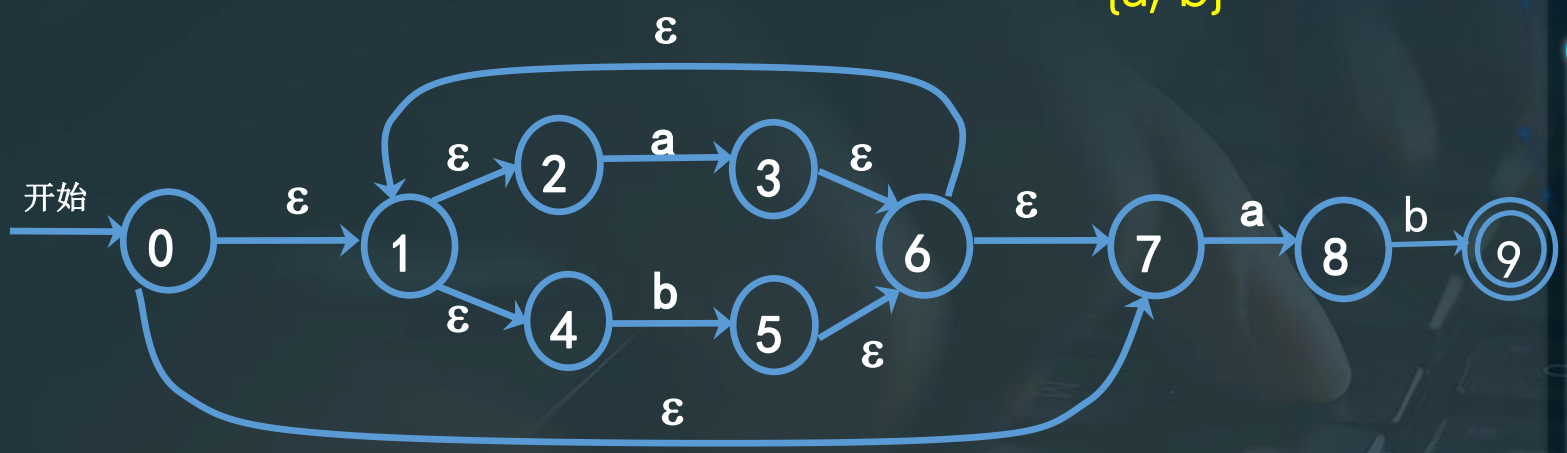
$B \xrightarrow{a} \{3, 8\} \xrightarrow{\varepsilon} \{3, 8, 6, 1, 7, 2, 4\}$

状态	a	b
A	B	C
B		

2.2.2 从NFA到DFA的转换

将(a|b)*ab的NFA转换为DFA

= {a, b}



A = {0, 1, 2, 4, 7}

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

C = {1, 2, 4, 5, 6, 7}

B

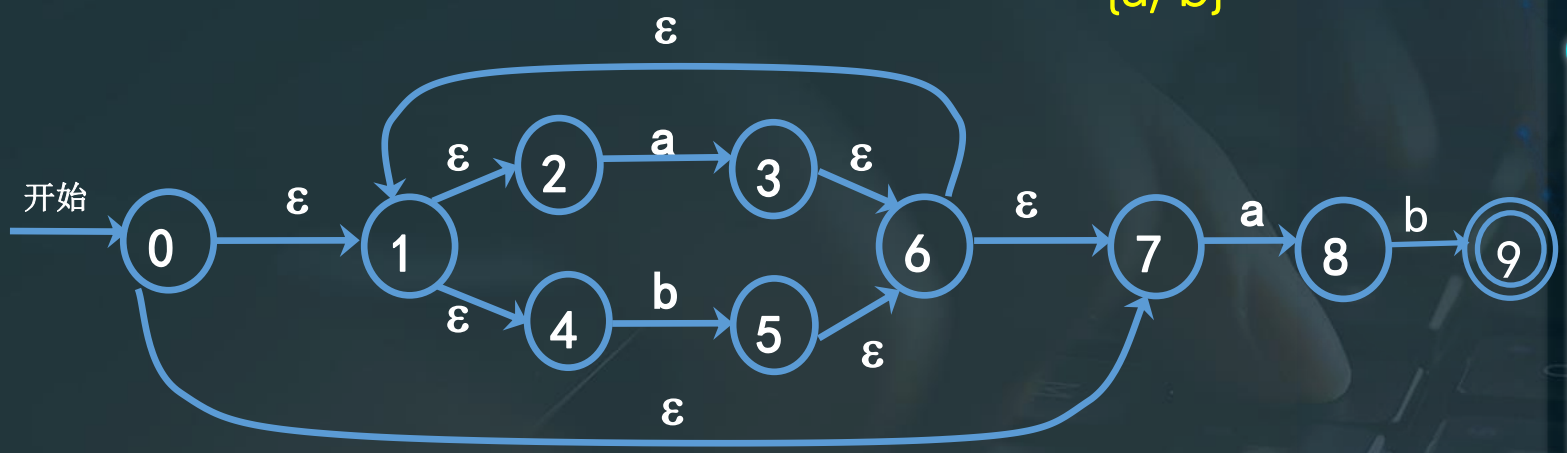
$B \xrightarrow{a} \{3, 8\} \xrightarrow{\varepsilon} \{3, 8, 6, 1, 7, 2, 4\}$

状态	a	b
A	B	C
B	B	

2.2.2 从NFA到DFA的转换

将(a|b)*ab的NFA转换为DFA

= {a, b}



A = {0, 1, 2, 4, 7}

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

C = {1, 2, 4, 5, 6, 7}

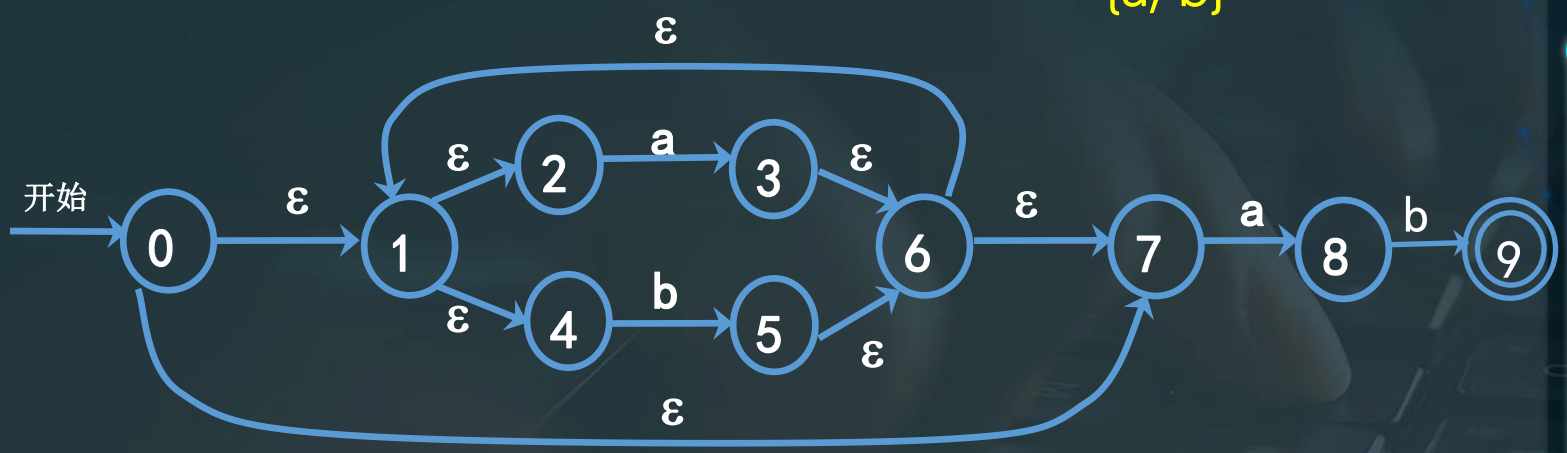
$B \xrightarrow{b} \{5, 9\} \xrightarrow{\varepsilon} \{5, 9, 6, 1, 7, 2, 4\}$

状态	a	b
A	B	C
B	B	

2.2.2 从NFA到DFA的转换

将(a|b)*ab的NFA转换为DFA

= {a, b}



A = {0, 1, 2, 4, 7}

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

C = {1, 2, 4, 5, 6, 7}

D

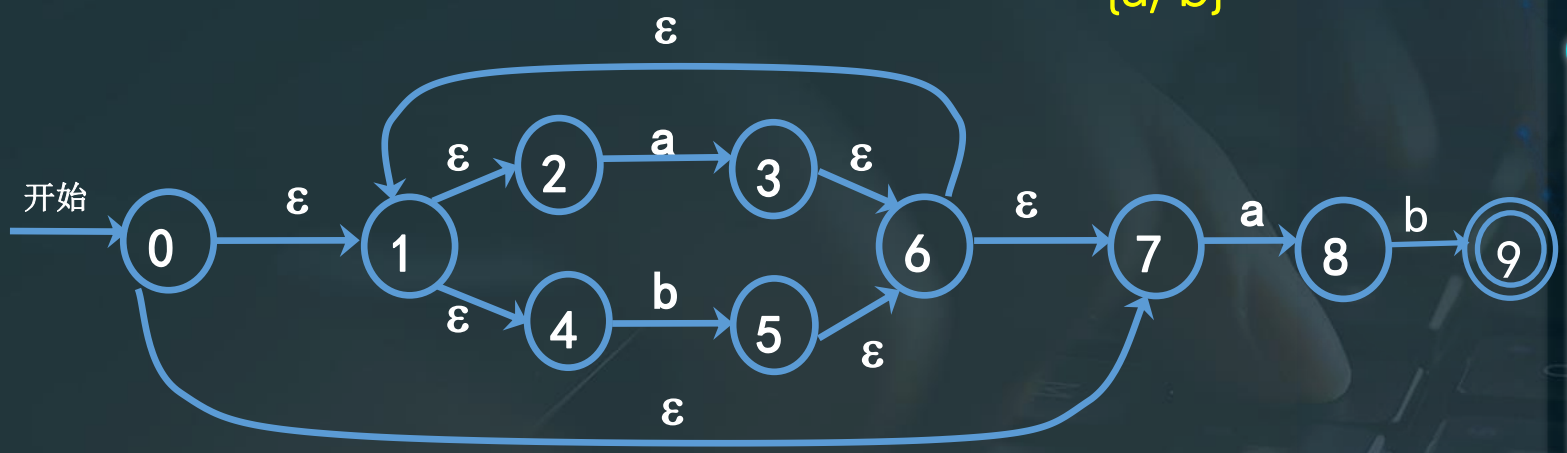
$B \xrightarrow{b} \{5, 9\} \xrightarrow{\varepsilon} \{5, 9, 6, 1, 7, 2, 4\}$

状态	a	b
A	B	C
B	B	D

2.2.2 从NFA到DFA的转换

将(a|b)*ab的NFA转换为DFA

= {a, b}



A = {0, 1, 2, 4, 7}

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

C = {1, 2, 4, 5, 6, 7}

D = {1, 2, 4, 5, 6, 7, 9}

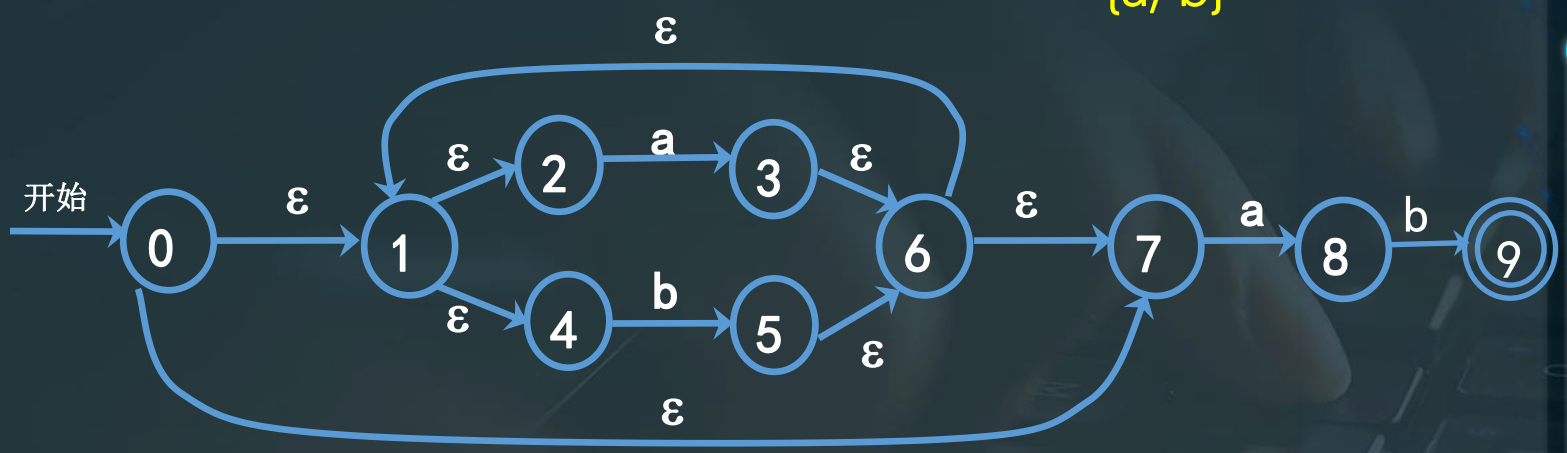
$C \xrightarrow{a} \{3, 8\} \xrightarrow{\varepsilon} \{3, 8, 6, 1, 7, 2, 4\}$

状态	a	b
A	B	C
B	B	D
C		

2.2.2 从NFA到DFA的转换

将(a|b)*ab的NFA转换为DFA

= {a, b}



A = {0, 1, 2, 4, 7}

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

C = {1, 2, 4, 5, 6, 7}

D = {1, 2, 4, 5, 6, 7, 9}

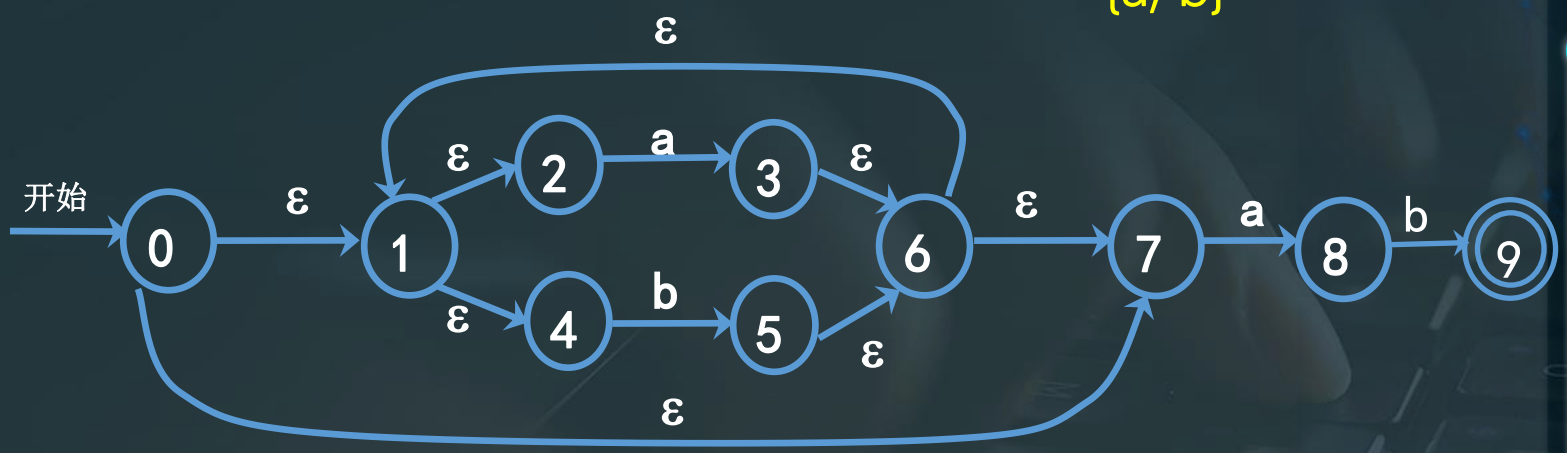
$C \xrightarrow{a} \{3, 8\} \xrightarrow{\varepsilon} \{3, 8, 6, 1, 7, 2, 4\}$ B

状态	a	b
A	B	C
B	B	D
C	B	

2.2.2 从NFA到DFA的转换

将(a|b)*ab的NFA转换为DFA

= {a, b}



A = {0, 1, 2, 4, 7}

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

C = {1, 2, 4, 5, 6, 7}

D = {1, 2, 4, 5, 6, 7, 9}

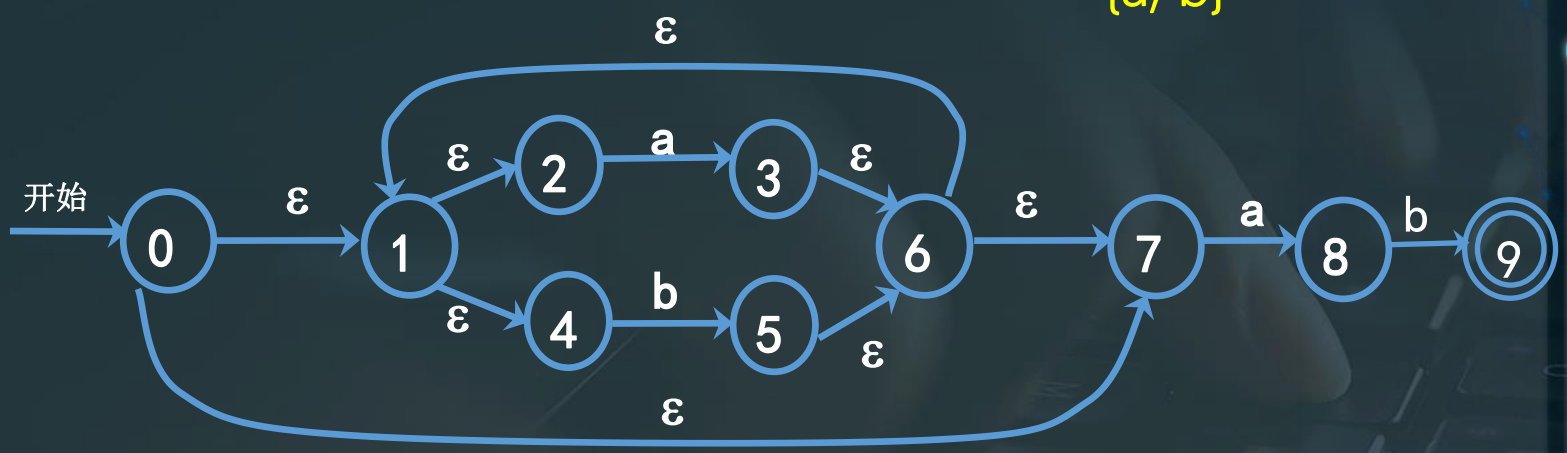
$C \xrightarrow{b} \{5\} \xrightarrow{\varepsilon} \{5, 6, 1, 7, 2, 4\}$

状态	a	b
A	B	C
B	B	D
C	B	

2.2.2 从NFA到DFA的转换

将(a|b)*ab的NFA转换为DFA

= {a, b}



A = {0, 1, 2, 4, 7}

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

C = {1, 2, 4, 5, 6, 7}

D = {1, 2, 4, 5, 6, 7, 9}

C $\xrightarrow{b}$ {5} $\xrightarrow{\varepsilon}$ {5, 6, 1, 7, 2, 4}

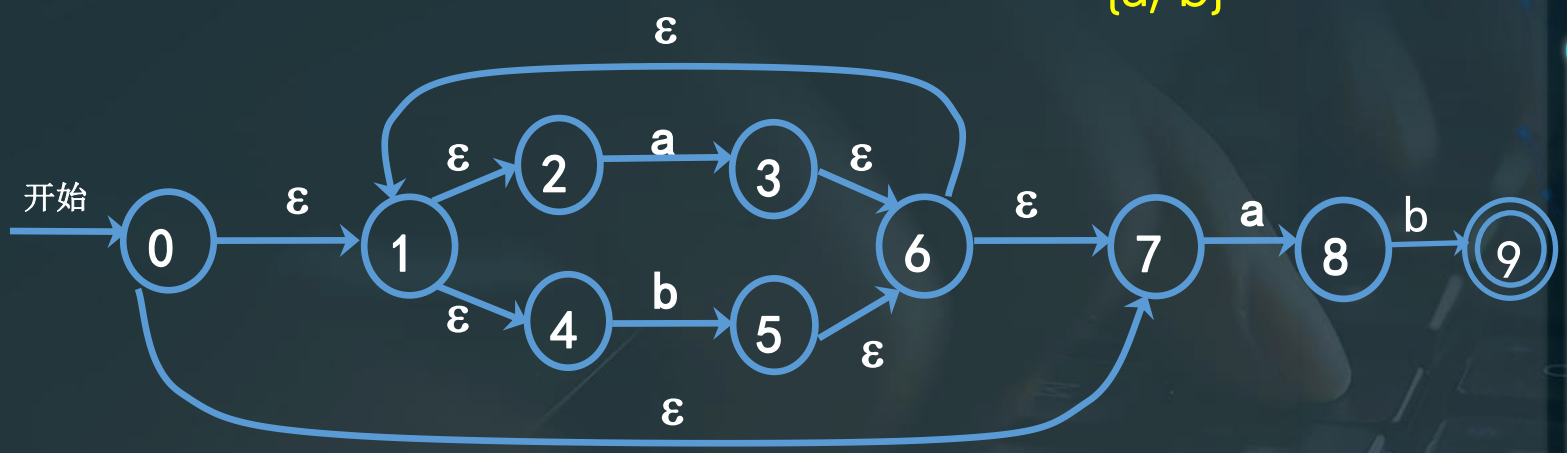
C

状态	a	b
A	B	C
B	B	D
C	B	C

2.2.2 从NFA到DFA的转换

将(a|b)*ab的NFA转换为DFA

= {a, b}



A = {0, 1, 2, 4, 7}

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

C = {1, 2, 4, 5, 6, 7}

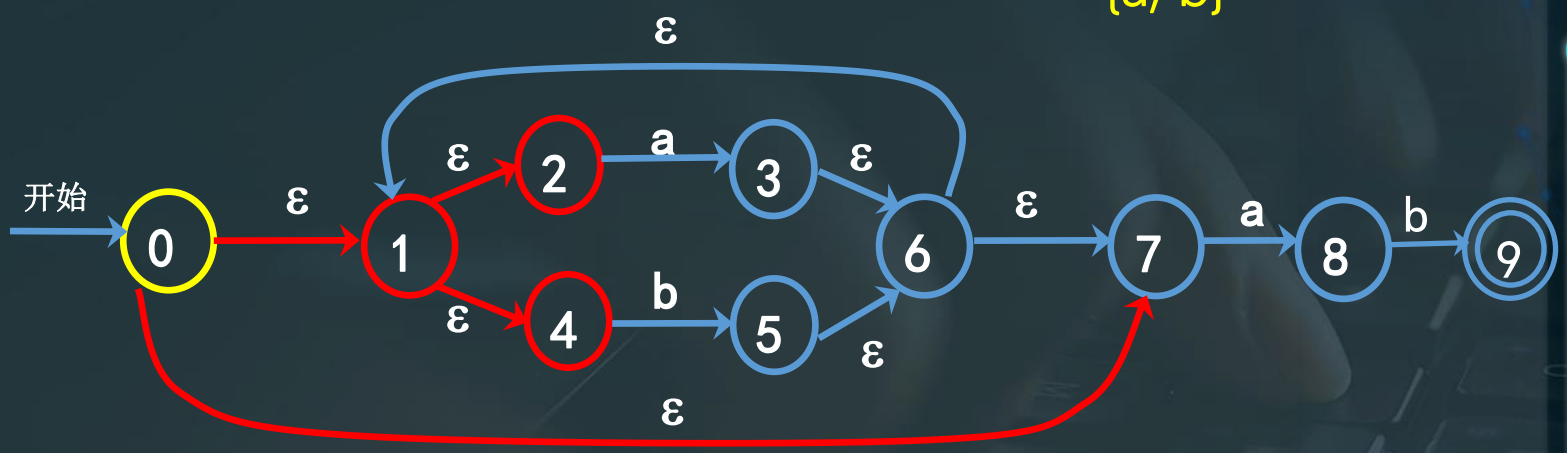
D = {1, 2, 4, 5, 6, 7, 9}

状态	a	b
A	B	C
B	B	D
C	B	C
D	B	C

2.2.2 从NFA到DFA的转换

将(a|b)*ab的NFA转换为DFA

= {a, b}



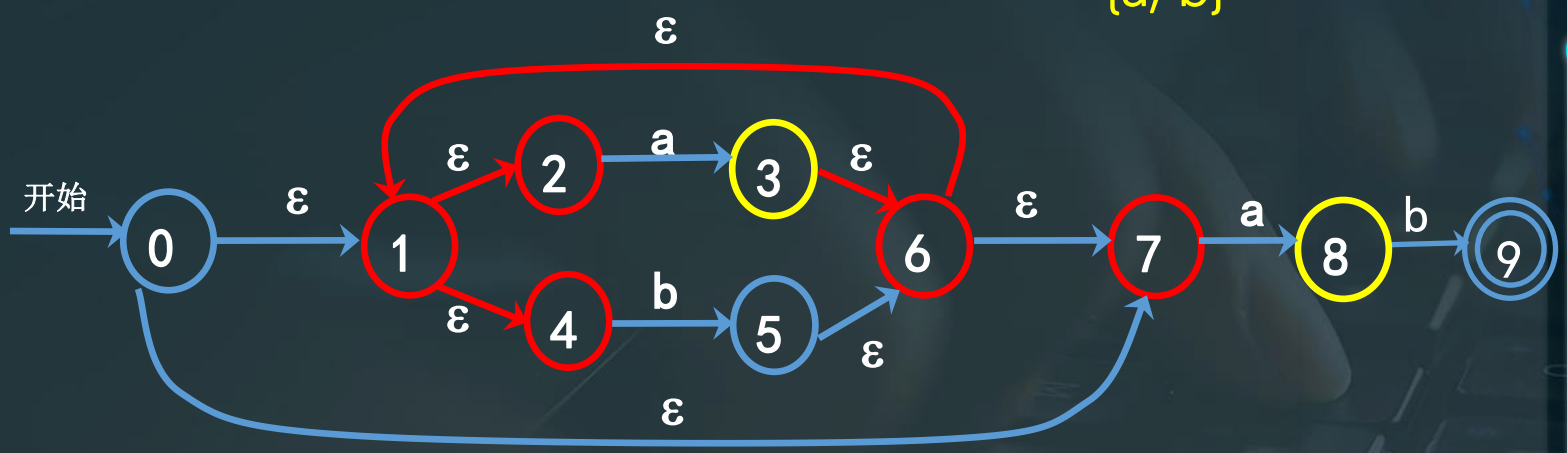
{0} $\xrightarrow{\varepsilon}$ {0, 1, 2, 4, 7}　只用 ε 转换能到达的NFA状态集合

ε-closure(T)：从NFA的状态集合 *T* 中每个状态出发，只用ε转换就能到达的状态的集合

2.2.2 从NFA到DFA的转换

将(a|b)*ab的NFA转换为DFA

= {a, b}



A = {0, 1, 2, 4, 7}

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

C = {1, 2, 4, 5, 6, 7}

$B \xrightarrow{a} \{3, 8\} \xrightarrow{\varepsilon} \{3, 8, 6, 1, 7, 2, 4\}$

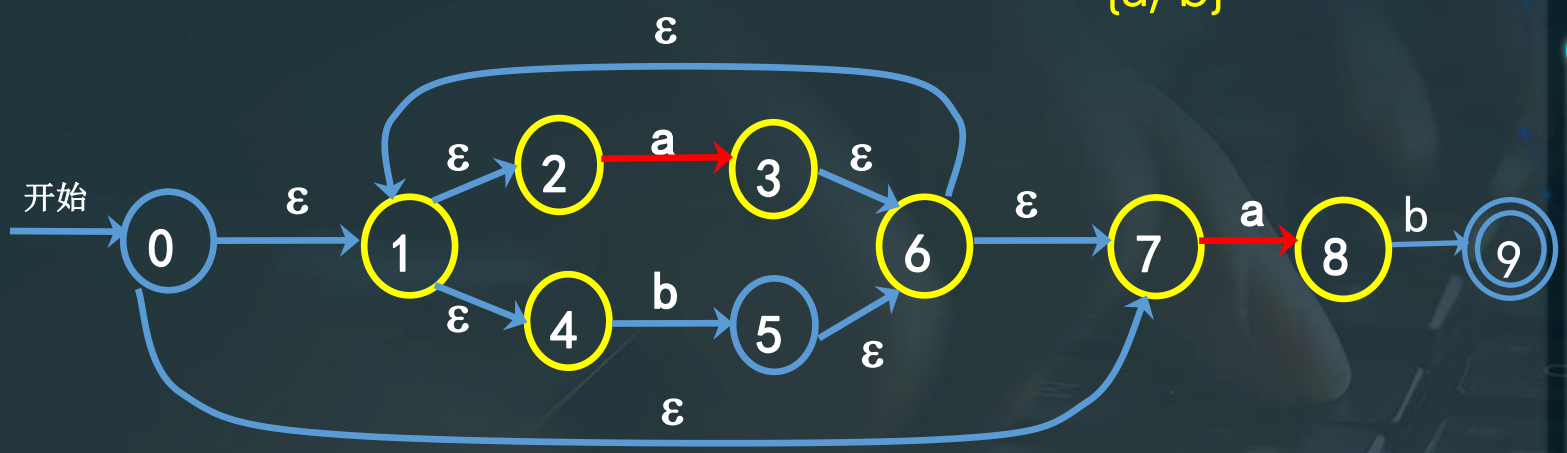
ε-closure({3, 8})

状态	a	b
A	B	C
B		

2.2.2 从NFA到DFA的转换

将(a|b)*ab的NFA转换为DFA

= {a, b}



A = {0, 1, 2, 4, 7}

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

C = {1, 2, 4, 5, 6, 7}

状态	a	b
A	B	C
B		

B $\xrightarrow{a}$ {3, 8}

move (T, c)：从NFA的状态集合 *T* 中每个状态出发，通过符号 *c* 能到达的所有状态集合

2.2.2 从NFA到DFA的转换

子集构造算法

- *ε-closure(T)*：从NFA的状态集合 *T* 中每个状态出发，只用ε 转换就能到达的状态的集合
- *move (T, c)*：从NFA的状态集合 *T* 中每个状态出发，通过符号 *c* 能到达的所有状态集合
- 输入: 一个NFA *N*
- 输出: 一个DFA *D*　　*D* 的转换表: *Dtran*
  状态集: *Dstates*

如果*D* 的某个状态 *V* 至少包含一个 *N* 的接收状态，那么 *V* 是 *D* 的一个接收状态

2.2.2 从NFA到DFA的转换

子集构造算法

```
T = ε-closure({s0}) //（s0是N的开始状态）
T.tag = 0 // (tag = 0 表示T未处理)
Dstates  = {T}
while (Dstates 中有一个状态 T 并且T.tag = 0) {
    for （每个字母表中的符号 a ）{
        V = ε-closure(move(T，a))
        if V 不在 Dstates 中 {
            V.tag = 0
            Dstates = Dstates U {V}
            Dran[T,a] = V
        }
    }
    T.tag = 1 //已处理 T 完毕
 }
```

2.2.2 从NFA到DFA的转换

子集构造算法

```
T = ε-closure({s0}) //（s0是N的开始状态）
T.tag = 0 // (tag = 0 表示T未处理)
Dstates  = {T}
while (Dstates 中有一个状态 T 并且T.tag = 0) {
    for （每个字母表中的符号 a ）{
        V = ε-closure(move(T，a))
        if V 不在 Dstates 中 {
            V.tag = 0
            Dstates = Dstates U {V}
            Dran[T,a] = V
        }
    }
    T.tag = 1 //已处理 T 完毕
}
```

2.2.2 从NFA到DFA的转换

子集构造算法

```
T = ε-closure({s0}) //（s0是N的开始状态）
T.tag = 0 // (tag = 0 表示T未处理)
Dstates  = {T}
while (Dstates 中有一个状态 T 并且T.tag = 0) {
    for （每个字母表中的符号 a ）{
        V = ε-closure(move(T，a))
        if V 不在 Dstates 中 {
            V.tag = 0
            Dstates = Dstates U {V}
            Dran[T,a] = V
        }
    }
    T.tag = 1 //已处理 T 完毕
}
```

子集构造算法

```
T = ε-closure({s0}) //（s0是N的开始状态）
T.tag = 0 // (tag = 0 表示T未处理)
Dstates  = {T}
while (Dstates 中有一个状态 T 并且T.tag = 0) {
    for （每个字母表中的符号 a ）{
        V = ε-closure(move(T，a))
        if V 不在 Dstates 中 {
            V.tag = 0
            Dstates = Dstates U {V}
            Dran[T,a] = V
        }
    }
    T.tag = 1 //已处理 T 完毕
 }
```

2.2.2 从NFA到DFA的转换

子集构造算法

```
T = ε-closure({s0}) //（s0是N的开始状态）
T.tag = 0 // (tag = 0 表示T未处理)
Dstates  = {T}
while (Dstates 中有一个状态 T 并且T.tag = 0) {
    for （每个字母表中的符号 a ）{
        V = ε-closure(move(T，a))
        if V 不在 Dstates 中 {
            V.tag = 0
            Dstates = Dstates U {V}
            Dran[T,a] = V
        }
    }
    T.tag = 1 //已处理 T 完毕
}
```

子集构造算法

```
T = ε-closure({s0}) //（s0是N的开始状态）
T.tag = 0 // (tag = 0 表示T未处理)
Dstates  = {T}
while (Dstates 中有一个状态 T 并且T.tag = 0) {
    for （每个字母表中的符号 a ）{
        V = ε-closure(move(T，a))
        if V 不在 Dstates 中 {
            V.tag = 0
            Dstates = Dstates U {V}
            Dran[T,a] = V
        }
    }
    T.tag = 1 //已处理 T 完毕
 }
```

2.2.2 从NFA到DFA的转换

子集构造算法

```
T = ε-closure({s0}) //（s0是N的开始状态）
T.tag = 0 // (tag = 0 表示T未处理)
Dstates  = {T}
while (Dstates 中有一个状态 T 并且T.tag = 0) {
    for （每个字母表中的符号 a ）{
        V = ε-closure(move(T，a))
        if V 不在 Dstates 中 {
            V.tag = 0
            Dstates = Dstates U {V}
            Dran[T,a] = V
        }
    }
    T.tag = 1 //已处理 T 完毕
}
```

编译原理

苏州大学　李军辉