编译原理与技术

--词法分析III

刘爽

天津大学智算学部

内容回顾

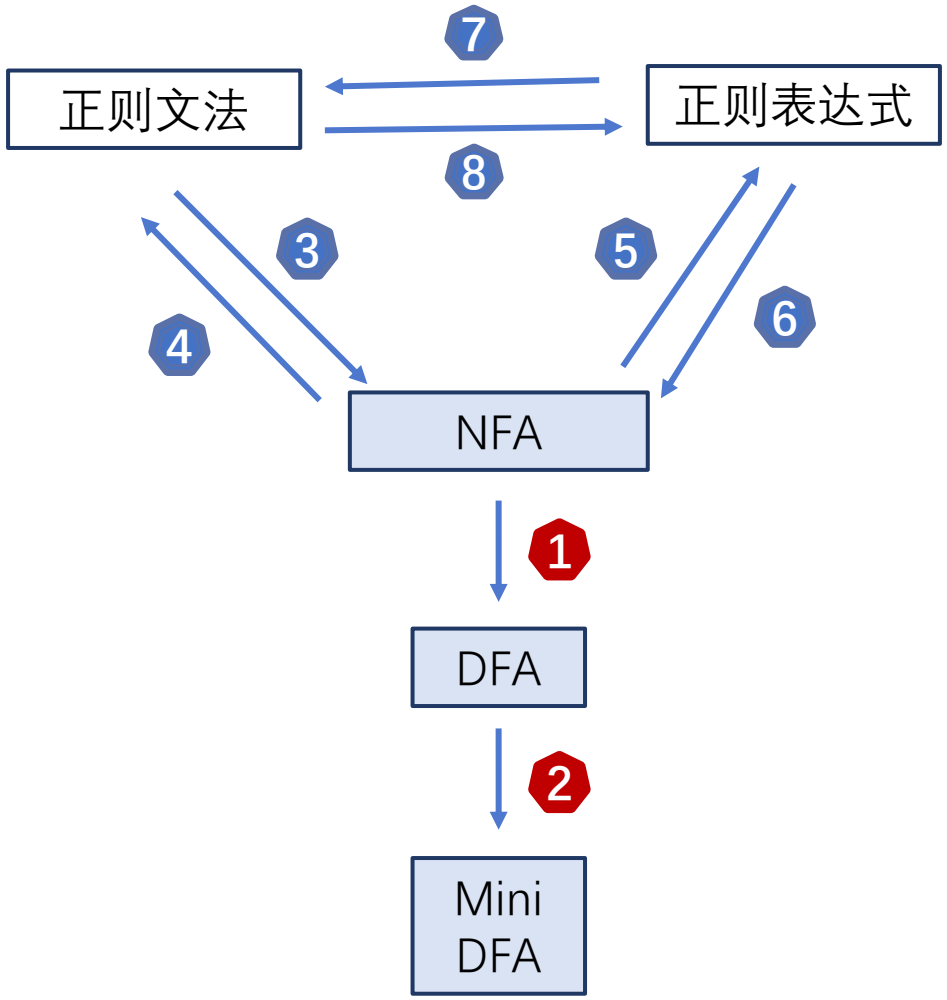
- 词法分析器的作用
- 词法分析程序的设计
  - 状态转换图
- 正规表达式和有限自动机
  - 有限自动机 （Finite Automata）
  - 正则表达式（Regular Expression）
  - 正则文法（Regular Gramma）
- 词法分析程序的自动生成

确定有限自动机 （DFA）
非确定有限自动机 （NFA）
NFA确定化
DFA最小化

内容提要

- 词法分析器的作用
- 词法分析程序的设计
  - 状态转换图
- 正规表达式和有限自动机
  - 有限自动机 （Finite Automata）
  - 正则表达式（Regular Expression）
  - 正则文法（Regular Gramma）
- 词法分析程序的自动生成

Overview

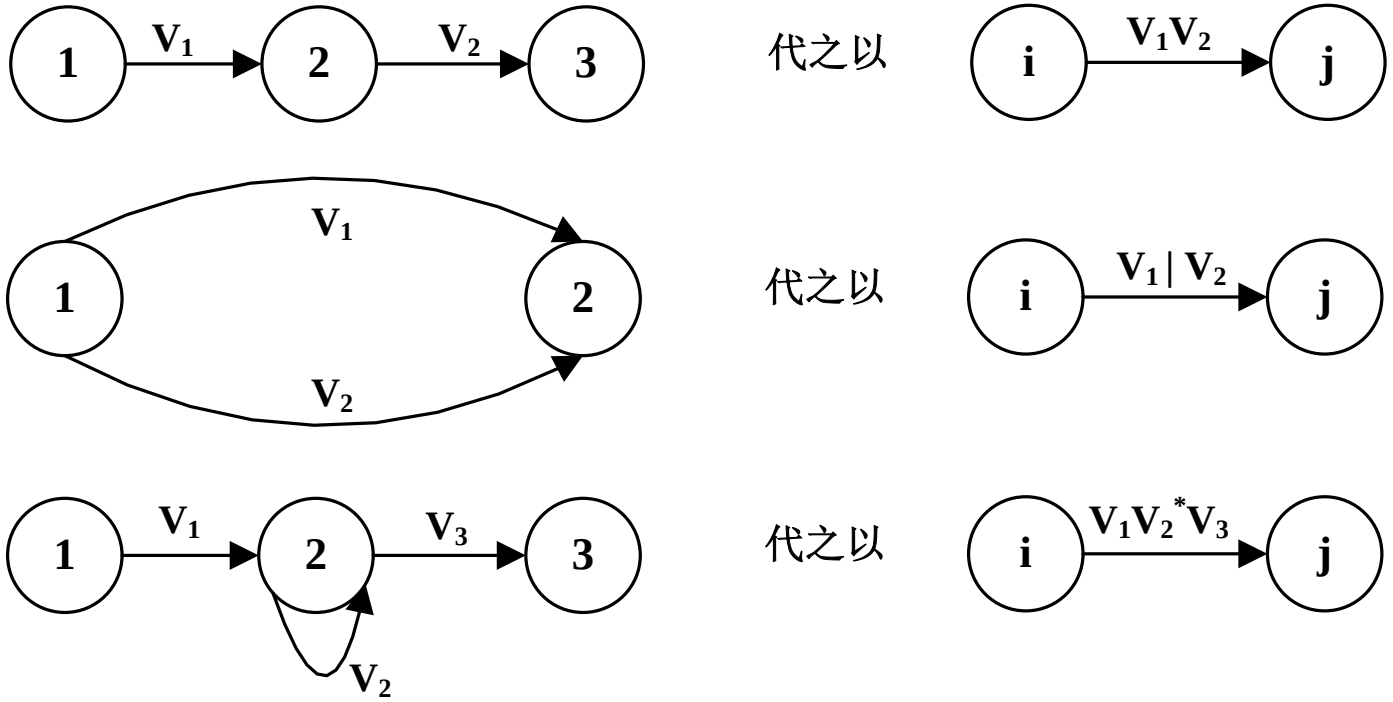


1. NFA→ DFA (NFA 确定化)
2. DFA→mini DFA (DFA最小化)
3. 正则文法→ NFA （自学）
4. DFA→正则文法（自学）
5. NFA→正则表达式
6. 正则表达式→NFA
7. 正则表达式→正则文法（自学）
8. 正则文法→正则表达式 (自学)

正则表达式和有限自动机

正则表达式和有限自动机的等价性
--有限自动机→正则表达式

- 关于正则表达式与自动机的等价性，有如下性质：
  - 对任何FA M，都存在一个正则表达式r，使得L(r)=L(M)。
  - 对任何正则表达式r，都存在一个FA M，使得L(M)=L(r)。



步骤：

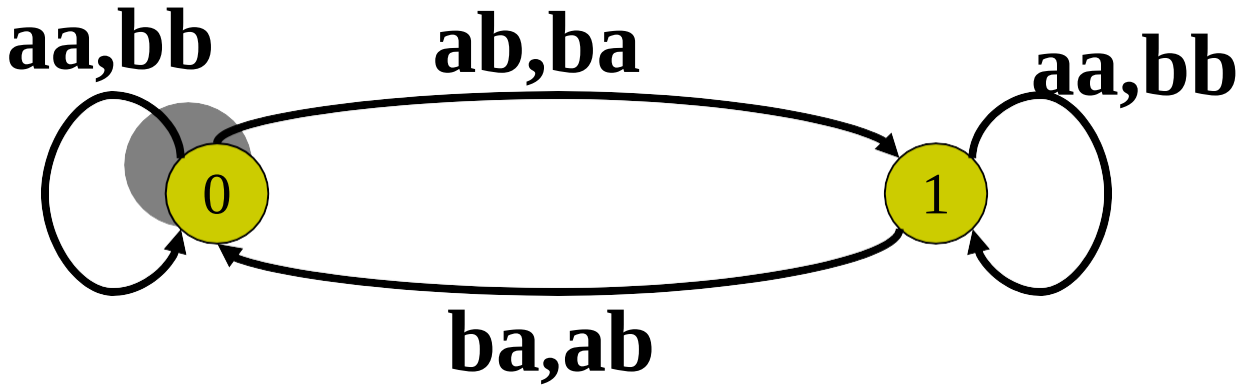
把转换图的意义拓宽,令每条弧上可以标记正则式.

1. 在给定的NFA M上加两个新状态, 一个为初态X,从X用ε弧连接M的所有初态,另一为Y,从M的所有终态用ε弧连到Y, 新的NFA M'与M等价.

2. 对M'用左侧等价规则,消去除X,Y以外的其它结.

例子

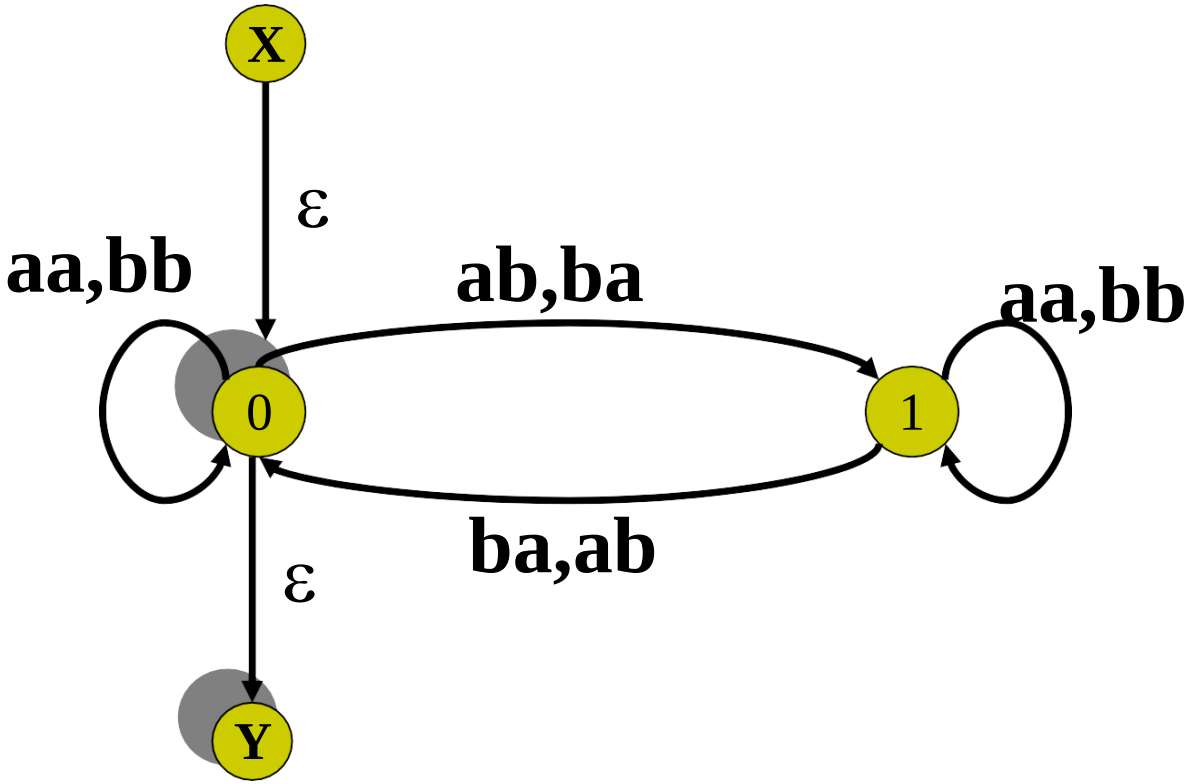
- 把下面的NFA M 转化成等价的正则表达式



NFA M 是识别具有偶数a或偶数个b的非确定有限自动机，则其初始状态是？

解

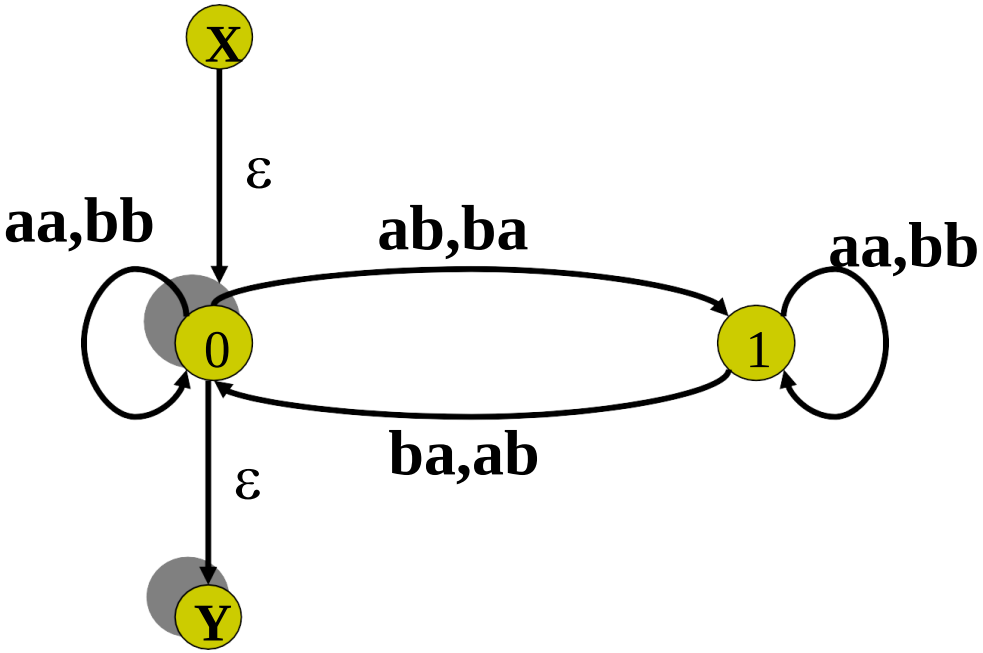
1. 在给定的NFA M上加两个新状态, 一个为初态X,从X用ε弧连接M的所有初态,另一为Y,从M的所有终态用ε弧连到Y, 新的NFA M’与M等价.

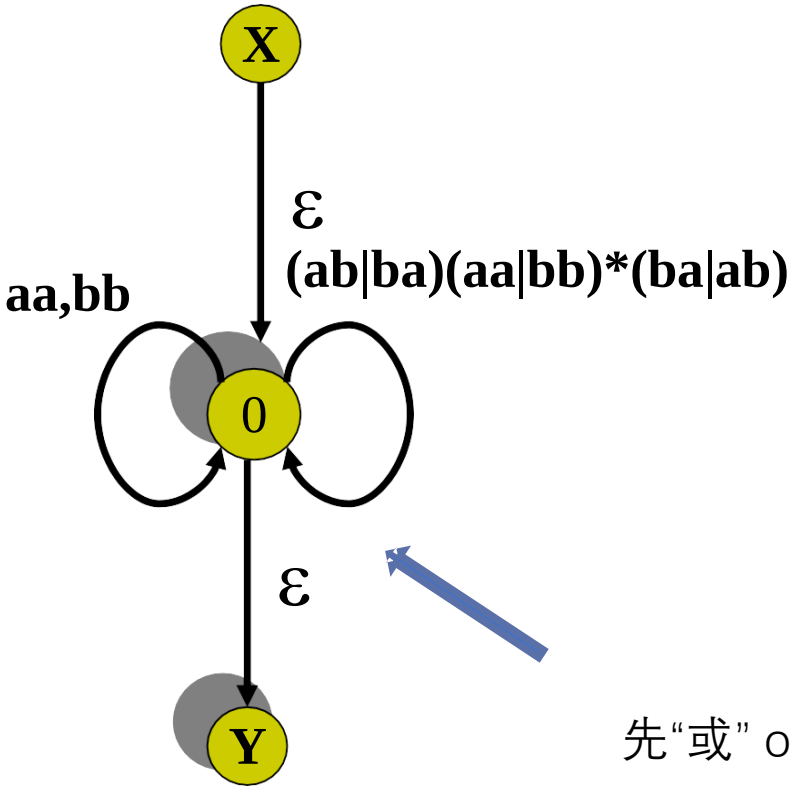


解

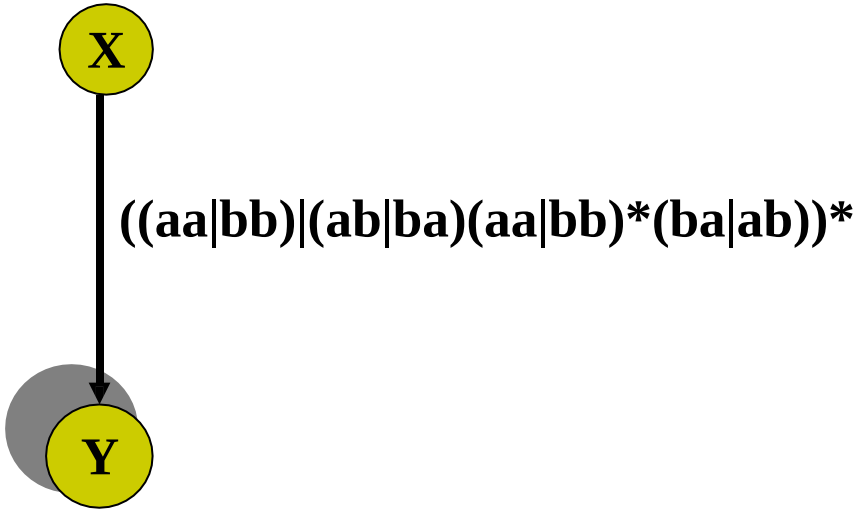
2. 对M'用等价规则,

消去除X,Y以外的其它结点.





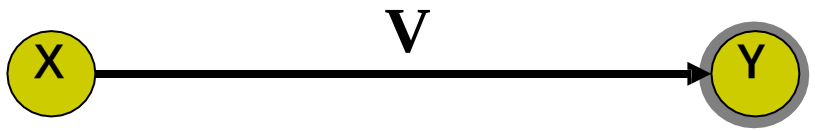
先"或" or先 "*"？



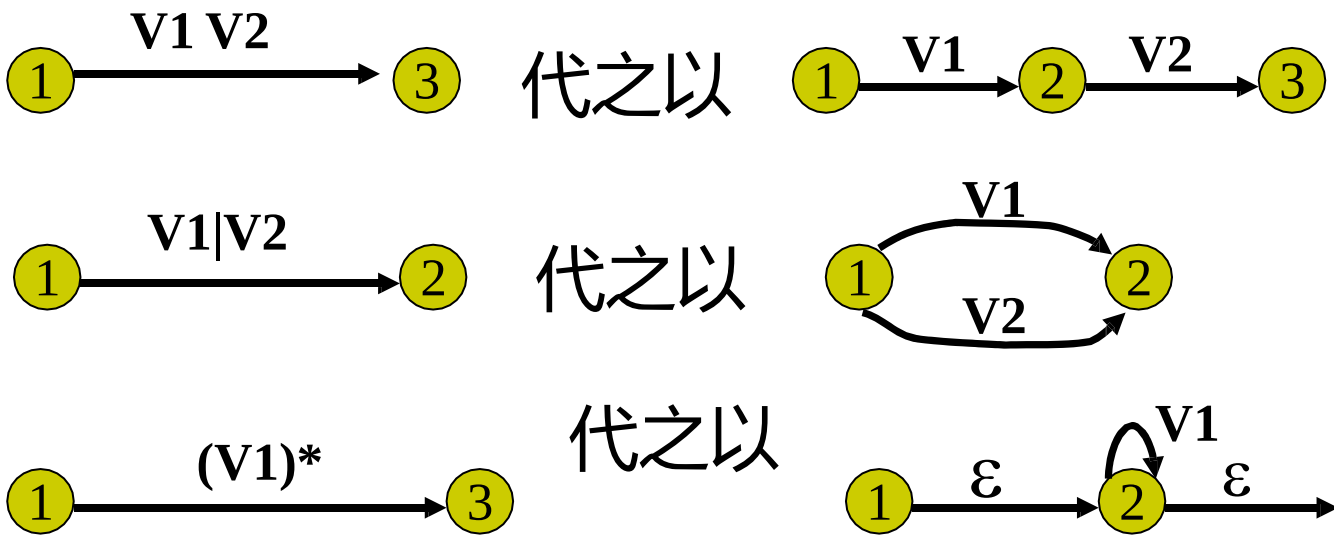
正则表达式和有限自动机的等价性

--正则表达式→有限自动机(I)

1. 对给定的正则表达式构成一个NFA M。 先写出:



2. 用以下规则对V进行分解并加进新节点



在分解过程中，要求:

(1) X,Y始终为唯一的初态和终态。

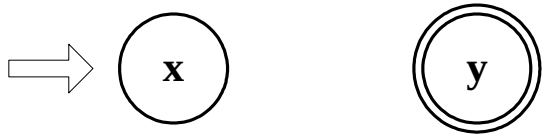
(2) 所加新结点其名字彼此不同。

(3) 弧上的标记必须是字符或空字。

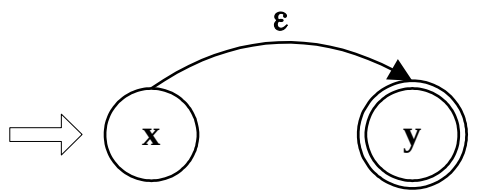
正则表达式和有限自动机的等价性

--正则表达式→有限自动机(II)

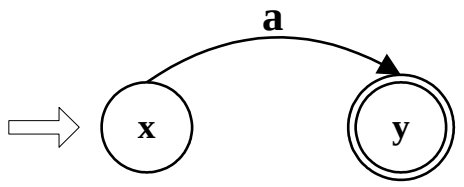
- 对于简单的正则表达式：
  - 对于正则表达式Φ，所构造的NFA为：



  - 对于正则表达式ε，构造的NFA为：



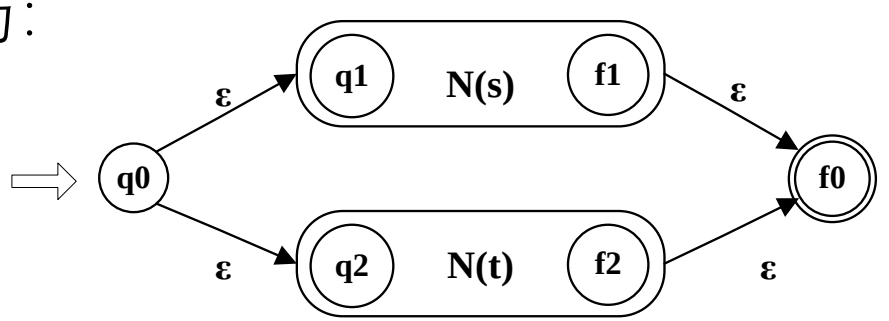
  - 对于正则表达式a，a∈∑，构造的NFA为：



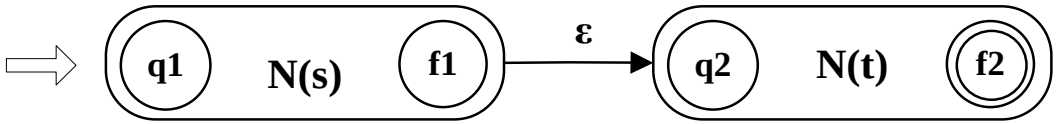
正则表达式和有限自动机的等价性

--正则表达式→有限自动机

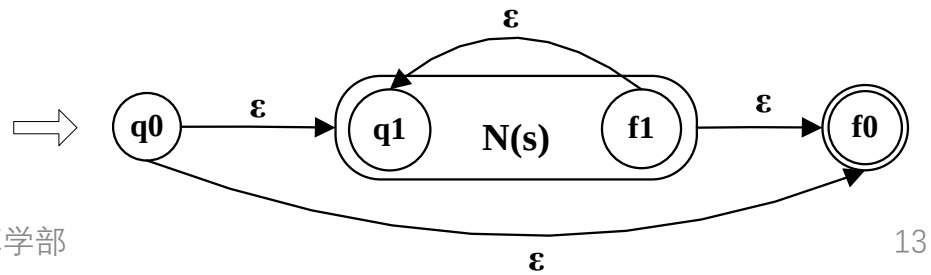
- 若s,t为∑上的正则表达式，相应的NFA分别为N(s)和N(t)，则
  - 对于正则表达式R=s | t，所构造的NFA(R)为：



  - 对于正则表达式R= st，构造的NFA(R)为：



  - 对于正则表达式R=s*，构造的NFA(R)为：

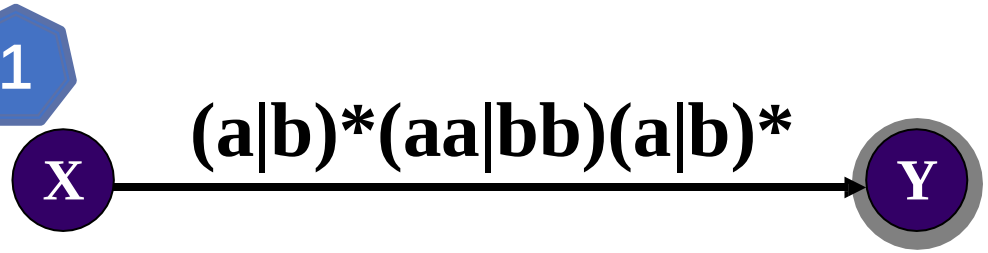


练习

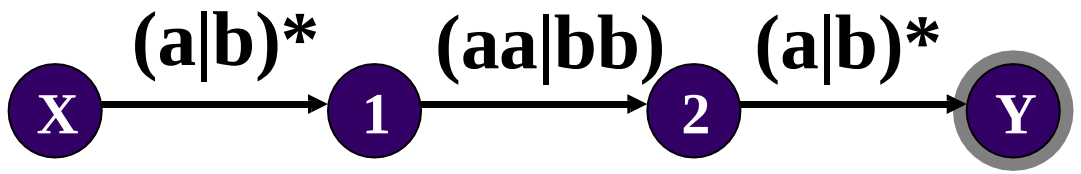
- 分别构造r1=1*, r2=01*, r3=01*|1 等价的有限自动机。

例子 （设V=(a|b)*(aa|bb)(a|b)*）

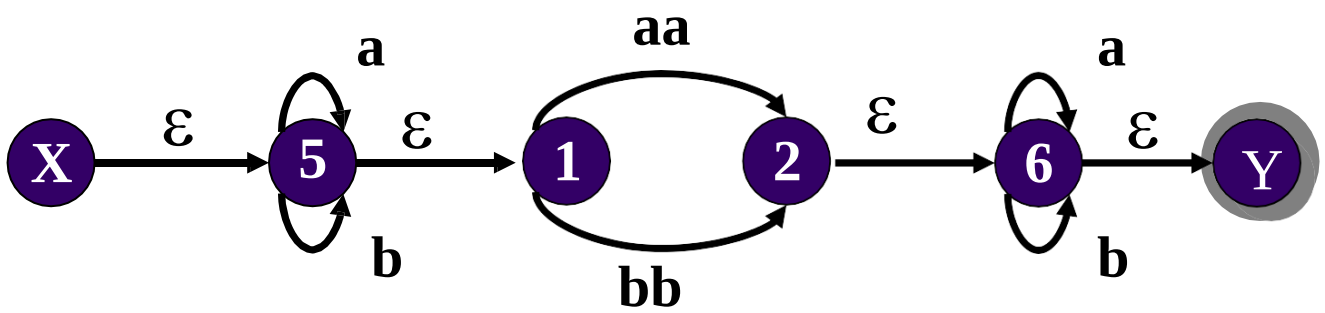
1



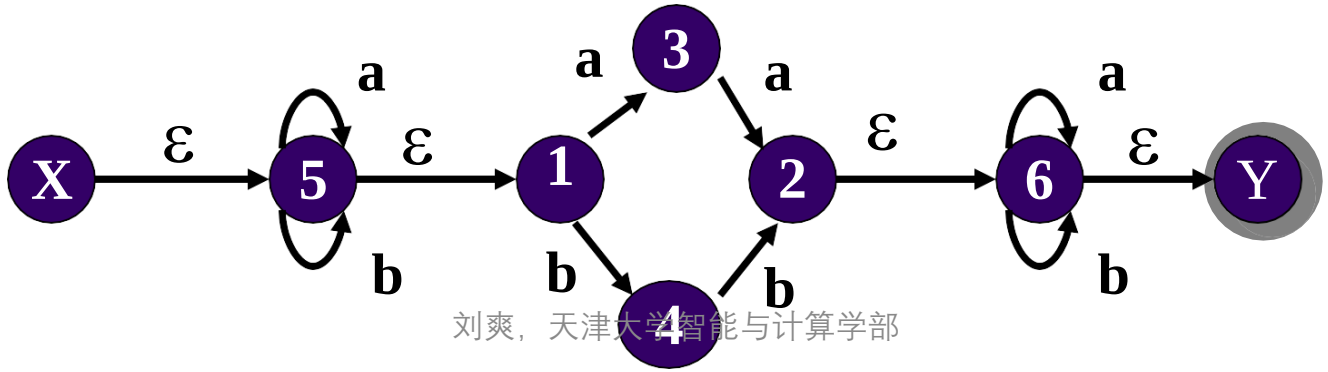
2



3



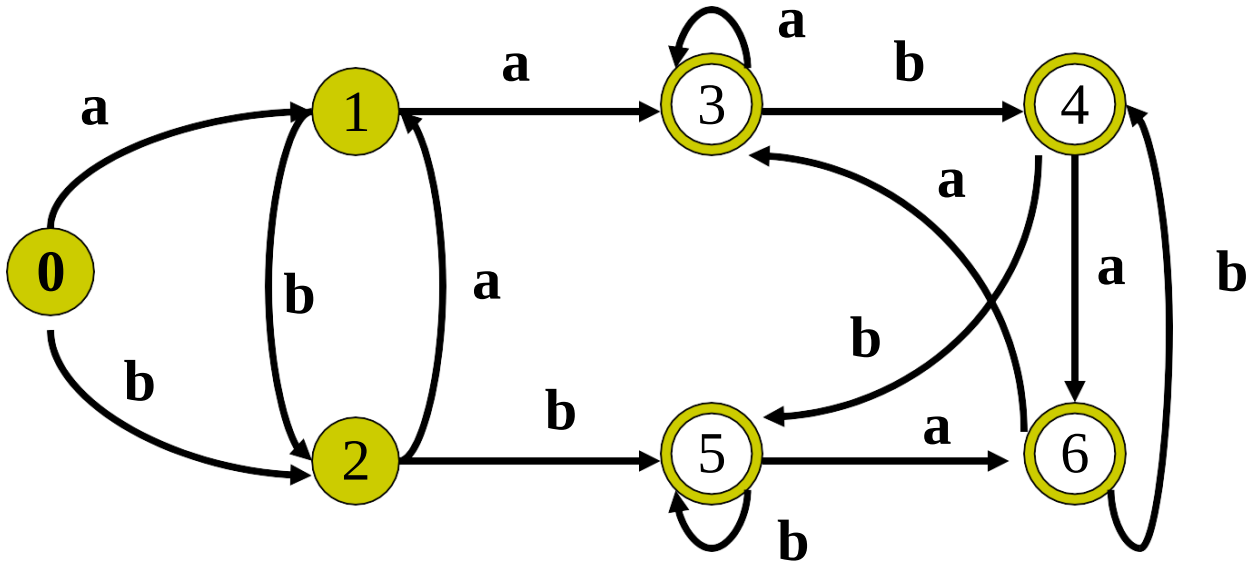
4



解

	I	Ia	Ib
0	{X,5,1}	{5,3,1}	{5,4,1}
1	{5,3,1}	{5,3,1,2,6,Y}	{5,4,1}
2	{5,4,1}	{5,3,1}	{5,3,1,2,6,Y}
3	{5,3,1,2,6,Y}	{5,3,1,2,6,Y}	{5,4,1,6,Y}
4	{5,4,1,6,Y}	{5,3,1,6,Y}	{5,3,1,2,6,Y}
5	{5,4,1,2,6,Y}	{5,3,1,6,Y}	{5,4,1,2,6,Y}
6	{5,3,1,6,Y}	{5,3,1,2,6,Y}	{5,4,1,6,Y}

解

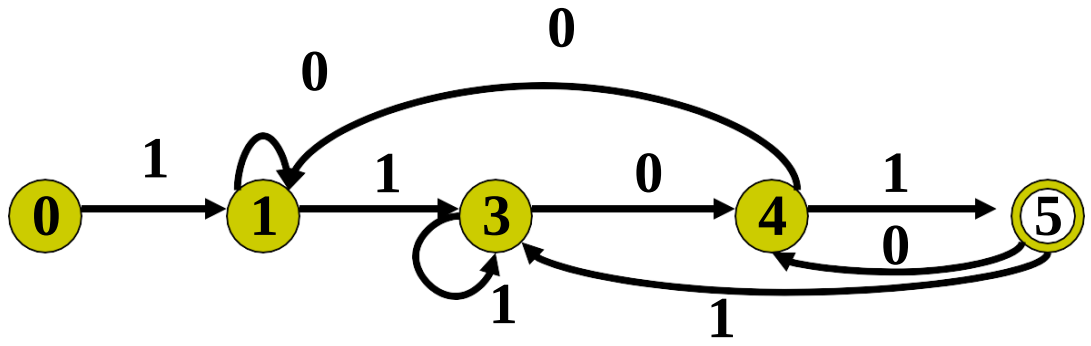


练习

构造下列正则表达式相应的DFA

1(0|1)*101

练习



构造下列正规式相应的DFA

1(0|1)*101

自学

正则文法和有限自动机

自学

正则文法

文法 $G=(V_T, V_N, P, S)$ V_T 是非空有限集, 每个元素是一个终结符。V_N 是非空有限集, 每个元素是一个非终结符. S是一个非终结符, 是开始符号 (S在产生式的左端必须至少出现一次), P是产生式的集合。

- 如果P中 的每一个产生式的形式为 (其中, A, $B \in V_N$, $a \in V_T \cup \{\varepsilon\}$)

  $A \rightarrow aB$ 或 $A \rightarrow a$, 则称 G 是右线性文法。

- 若文法G中的每一个产生式的形式为

  $A \rightarrow Ba$ 或 $A \rightarrow a$, 则称 G是左线性文法。

右线性文法和左线性文法都称为正则文法（3型文法）, 其所产生的语言都称为正则语言 或3型语言。

自学

正则文法与有限自动机的等价性

- 对于正则文法G和有限自动机M，如果L(G)=L(M)，则称G和M是等价的。关于正则文法和有限自动机的等价问题，有以下结论：
  - 对每一个右线性正则文法G_R或左线性正则文法G_L，都存在一个有限自动机(FA) M，使得L(M) =L(G_R)=L(G_L)。
  - 对每一个有限自动机(FA) M，都存在一个右线性正则文法G_R和左线性正则文法G_L，使得L(M)=L(G_R)=(G_L)。

自学

正则文法和有限自动机的等价性

--右线性文法→有限自动机

- 右线性文法$G = (V_N, V_T, S_G, P)$, 构造 $M = (S, \Sigma, \delta, S_0, F)$:
  - 令$S=V_N \cup \{f\}$, f为一新增非终结符号且f不属于V_N;
  - 令$\Sigma = V_T$;
  - 令$S_0 = \{S_G\}$;
  - 令$F= \{f\}$;
  - 令δ如下:
    - 对于G中每一形如A→ aB的产生式, 其中$A,B \in V_N$ 且$a \in V_T \cup \{\varepsilon\}$, 从结点A引一条箭弧到结点B, 并用符号a标记这条箭弧，即$\delta(A, a)=B$.
    - 对于G中每一形如A→a的产生式, 其中$A \in V_N$ 且$a \in V_T \cup \{\varepsilon\}$, 从结点A引一条箭弧到终态结点f,并用符号a标记这条箭弧，即$\delta(A, a)=f$.

在G中，$S_G \overset{+}{\Rightarrow} w$的充要条件是：在M中，从状态$S_0$出发到f有一条通路，其上所有箭弧的标记符号依次连接起来恰好等于w, 这就是说$w \in L(G)$当且仅当$w \in L(M)$, 故$L(G)=L(M)$

自学

例题

设给定右线性文法G: $S \longrightarrow aS|bA|b$

$A \longrightarrow aS$

构造与其等价的有限状态自动机

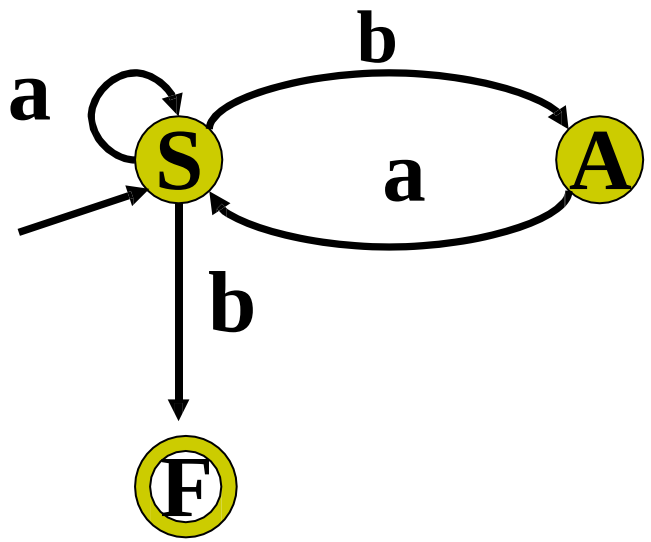
M=({S, A, f}, {a, b}, δ, {S}, {f})

δ(S, a)=S,

δ(S, b)=A,

δ(S, b)=f,

δ(A, a)=S



自学

正则文法和有限自动机的等价性

--左线性文法→有限自动机

- $G=\{V_N, V_T, P, S_G\}$是一个左线性文法,构造的状态转换图$M=(S, \Sigma, \delta, S_0, F)$ 如下:
  - 令$S=V_N \cup \{q0\}$, q0为一新增非终结符号且q0不属于V_N;
  - 令$\Sigma = V_T$;
  - 令$S_0 = \{q0\}$;
  - 令$F= \{S_G\}$;
  - 令δ如下:
    - 对于G中每一形如A→Ba的产生式,其中$A,B \in V_N$ 且$a \in V_T \cup \{\varepsilon\}$, 从结点B引一条箭弧到结点A,并用符号a标记这条箭弧, 即$\delta(B, a)=A$
    - 对于G中每一形如A → a的产生式,其中$A,B \in V_N$ 且$a \in V_T \cup \{\varepsilon\}$, 从初态q0引一条箭弧到结点A,并用符号a标记这条箭弧, $\delta(q0, a)=A$

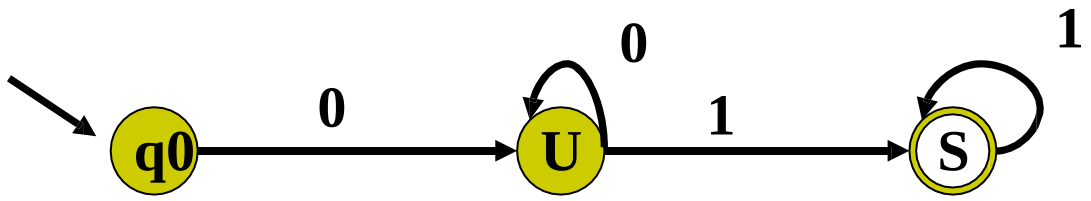
自学

例题

- 对于左线性文法G=（{S，U}，{0，1}，P，S）其中P={S→S1，S→U1，U→U0，U→0}

构造与其等价的有限自动机

M=({S, U, q0}, {0, 1}, δ, {q0}, {S})
δ(S, 1)=S,
δ(U, 1)=S,
δ(U, 0)=U,
δ(q0, 0)=U



自学

正则文法和有限自动机的等价性

--有限自动机→右线性文法

- 由DFA M = (S, Σ , δ , s_0, F)定义右线性文法G = (V_N , V_T , S_G , P):
  - 如果s_0不属于F, 令 V_N=S, V_T= Σ, S_G = s_0 , P是按下面规则定义: 对任何a ∈ Σ且A,B∈ S, 若有δ(A, a)=B
    - 当B不属于F, 令A →aB;
    - 当B ∈ F, 令A→a|aB

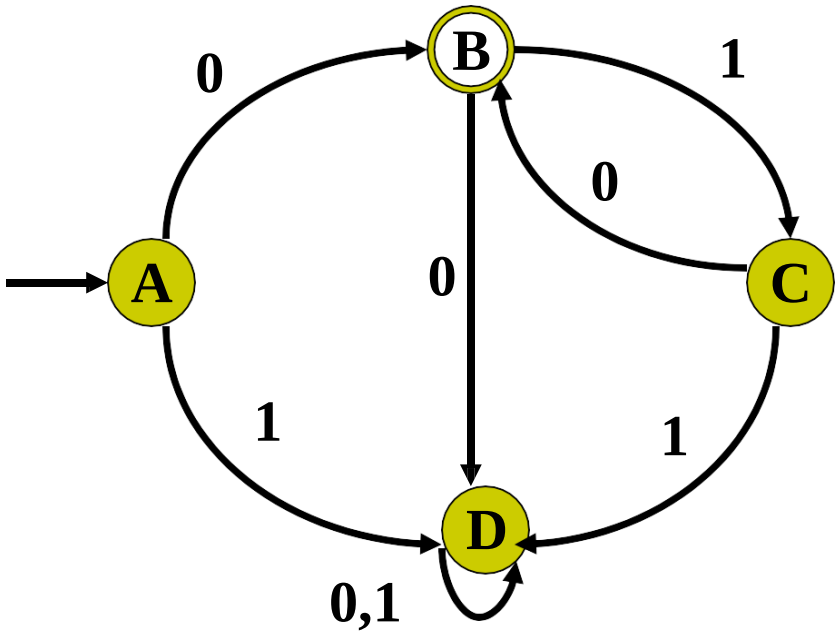
类似的从M出发可构造左线性文法。

  - 如果s_0∈F, 因为δ(s_0, ε)= s_0,所以, ε ∈ L(M), 但ε不属于上面构造的G所产生的语言L(G)。实际上L(G)=L(M)-{ε}， 因而对上面由M出发所构造的右线性正规文法G中添加一个非终结符号S'_0不属于 V_N (即$\mathbf{V_N}$ =S US'$_0$)和产生式S'_0→s_0| ε (即**P**=PU {S'_0→s_0| ε })并用S_G = S'_0代替s_0作开始符号。这样经过 修正后的G仍是右线性正规文法且L(G)=L(M).

自学

例题:

设DFA M=({A,B,C,D}, {0,1} ,f, A, {B})



L(M)=0(10)*.

构造与其等价的右线性正规文法:

自学

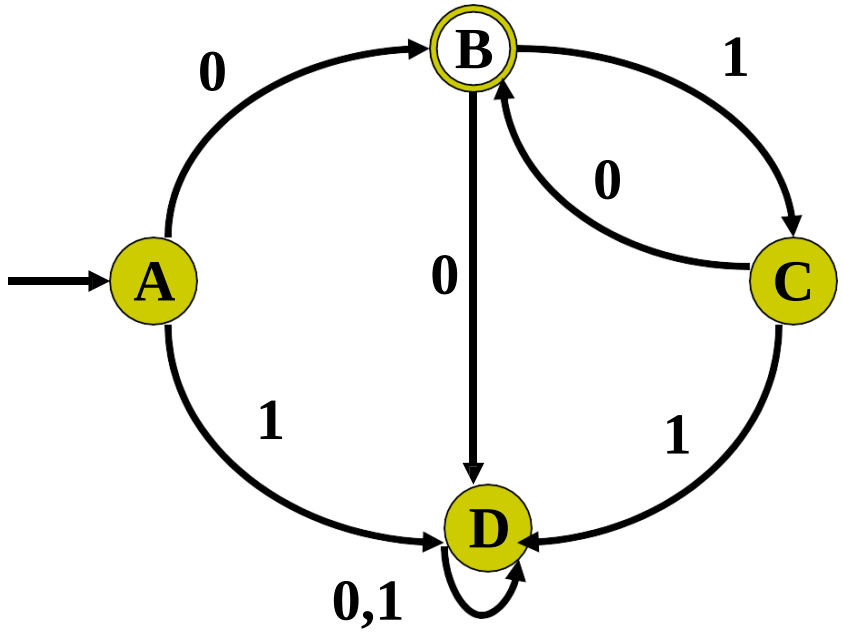
解

G= ({A，B，C，D}, {0，1}，A，P)

其中P为产生式的集合：

A→0|0B|1D　　B→0D|1C

C→0|0B|1D　　D→0D|1D



L(G)=L(M)=0(10)*

自学

例题：由右线性正规文法G出发构造NFA

A→0|0B|1D　　B→0D|1C

C→0|0B|1D　　D→0D|1D

M’=({A,B,C,D,f}, {0,1}, δ,{A},{f})

其中

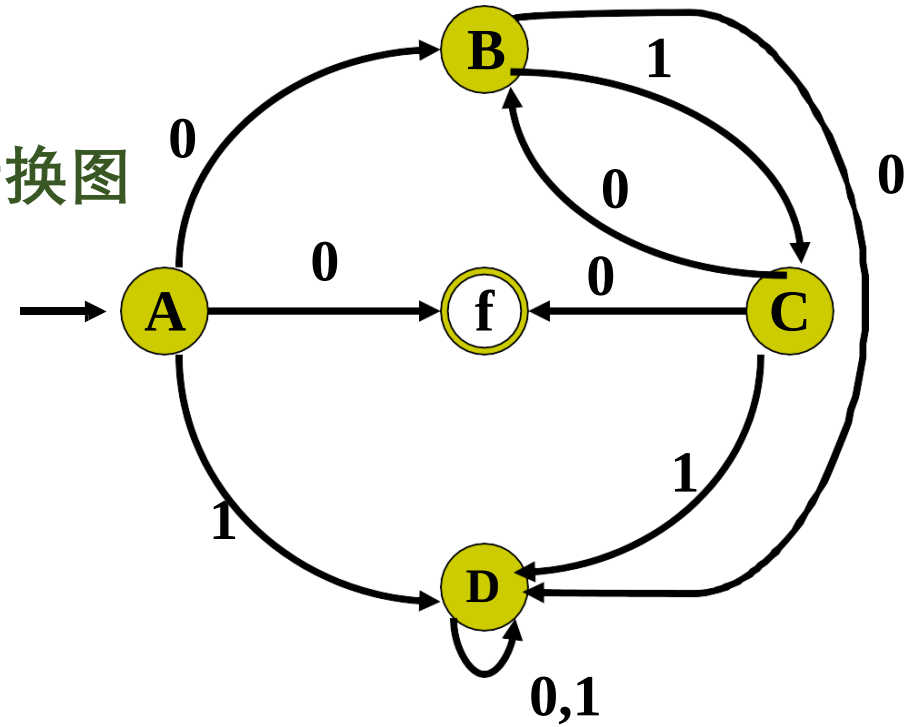
δ(A,0)={f,B}　　δ(A,1)={D}

δ(B,0)={D}　　δ(B,1)={C}

δ(C,0)={f,B}　　δ(C,1)={D}

δ(D,0)={D}　　δ(D,1)={D}

状态转换图



$$L(M') = L(M)$$

自学

例题：由NFA 构造左线性正规文法

- 从M’构造左线性文法G’的产生式P 的方法:
  - 若对任何S, S’ ∈{B, C, D, f}且a∈{0,1}有δ(S, a)=S’, 则令S’→Sa,
  - 若有δ(A, a)=S (A是M’的初态), 则令S→a,

- G’=（{B，C，D，S}，{0，1}，S，P）
- 其中S=f 为下面产生式的集合:

  S→0| C0　　C→B1　　B→0|C0　　D→1|C1|D0|D1|B0

- 　　有L(G’)=L(M’)=L(G)=L(M)=0(10)*

自学

练习

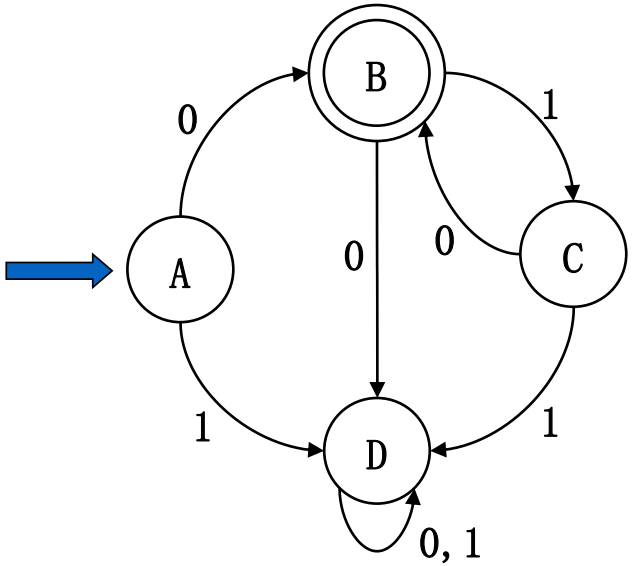


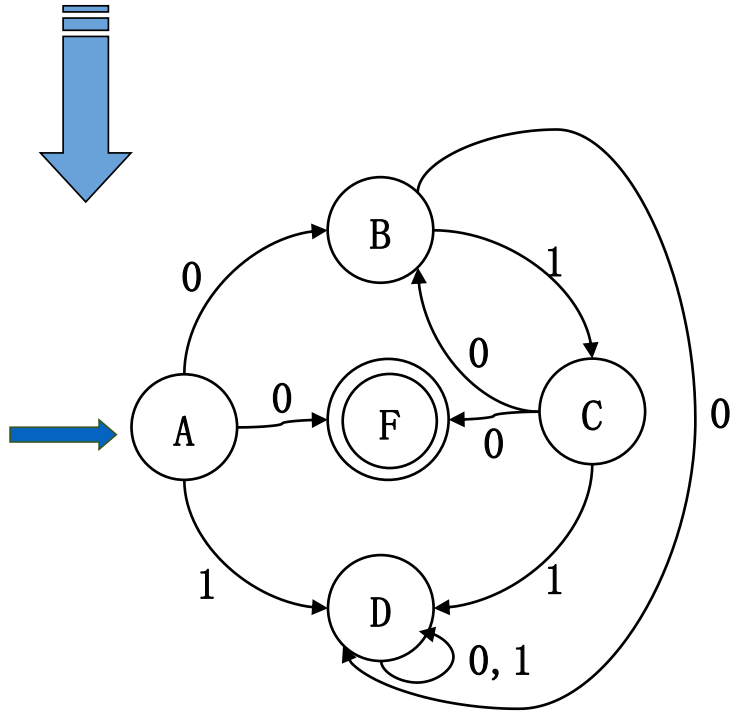
图1 DFA

正则文法

$G_R=<\{0,1\},\{A,B,C,D\},A,P>$其中P由以下产生式构成

A→0|0B|1D　　B→0D|1C

C→0|0B|1D　　D→0D|1D



（1）由图1的DFA构造与其等价的右线性文法，

（2）再从该文法出发构造等价的NFA

自学

正则表达式和正则文法

自学

正则表达式转正则文法（步骤）

- 对∑上的正则表达式U ,存在一个正则文法$G=(V_N,V_T,P,S)$: L(G)=L(U)

步骤：

初始： $V_T=\Sigma$, $S \in V_N$， 生成正则产生式$S \to U$

(1) 对形如 $A \to r_1 r_2$的正则产生式： $A \to r_1 B$

$B \to r_2$ $B \in V_N$

(2)对形如$A \to r^* r_1$的正则产生式： $A \to rB$

$A \to r_1$

$B \to rB$

$B \to r_1$ $B \in V_N$

(3)对形如$A \to r_1 \mid r_2$的正则产生式: $A \to r_1$

$A \to r_2$

不断应用上述规则做变换，直到每个产生式右端只含一个V_N

自学

正则表达式转正则文法举例

- 例1：r=a(a|d)*
  - V_T={a,d} S→a(a|d)*

(1) S→aA
A→(a|d)*

(2) A→(a|d)A
A→ε

(3) G[s]: S→aA
A→ε
A→aA
A→dA

P

V_T={a, d}
V_N={S,A}
初始符号为S

初始：$V_T=\Sigma$, $S \in V_N$，生成正则产生式S→U

(1) 对形如 $A\rightarrow r_1r_2$的正则产生式：
$A\rightarrow r_1B$
$B\rightarrow r_2$ $B\in V_N$

(2)对形如$A\rightarrow r^*r_1$的正则产生式：
$A\rightarrow rA$ $A\rightarrow r_1$

(3)对形如$A\rightarrow r_1|r_2$的正则产生式：
$A\rightarrow r_1$ $A\rightarrow r_2$

不断应用上述规则做变换，直到每个产生式右端只含一个V_N

自学

正则文法转正则表达式*

- 对G=(V_N,V_T,P,S),存在一个$\Sigma = V_T$上的正则表达式r : L(r)=L(G)

	正则文法	正则表达式
规则1	$A \rightarrow xB, B \rightarrow y$	$A=xy$
规则2	$A \rightarrow xA \mid y$	$A=x^*y$
规则3	$A \rightarrow x, A \rightarrow y$	$A=x \mid y$

自学

正则文法转正则表达式举例

- G[s]: S→aA
  A→aA
  A→dA
  S→a
  A→a
  A→d

① S→aA|a

② A→aA|a|dA|d

③ A→(a|d)A|(a|d)

④ A→(a|d)*(a|d)

S=a(a|d)*(a|d)|a =a((a|d)*(a|d)|ε) =a((a|d)$^{+}$|ε)

规则1 A→xB, B→y A=xy

规则2 A→xA|y A=x*y

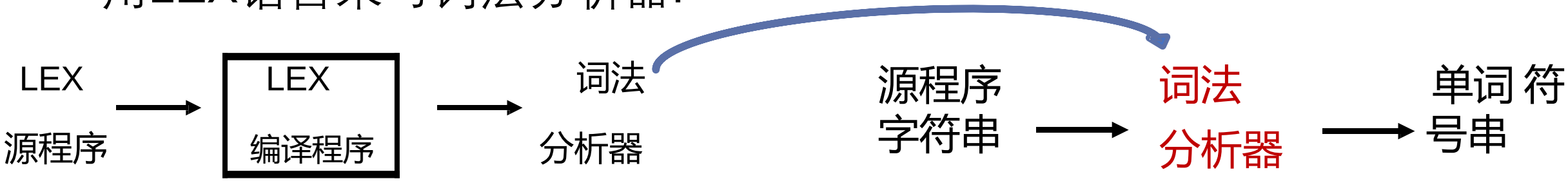
规则3 A→x, A→y A=x|y

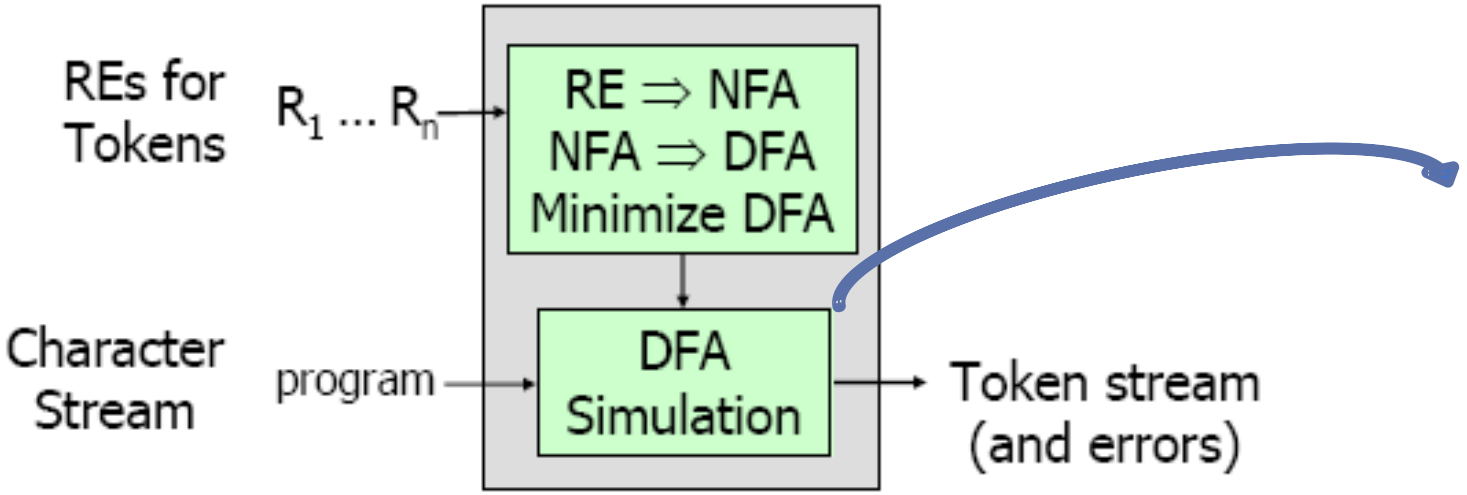
内容提要

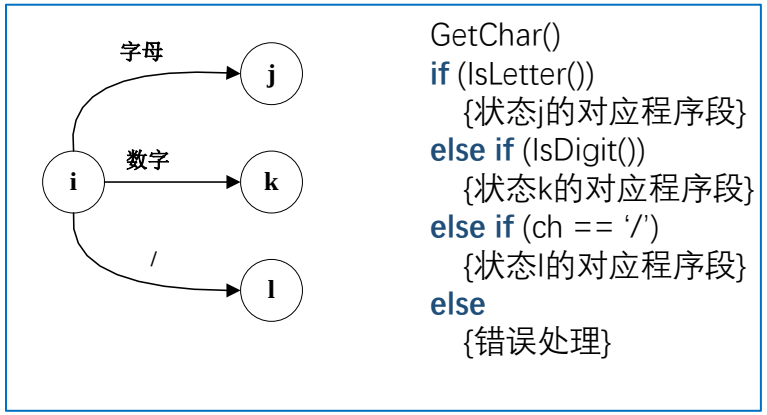
- 词法分析器的作用
- 词法分析程序的设计
  - 状态转换图
- 正规表达式和有限自动机
  - 正规表达式（Regular Expression）
  - 有限自动机 （Finite Automata）
- 词法分析程序的自动生成

词法分析器的自动生成

- 用LEX语言来写词法分析器:

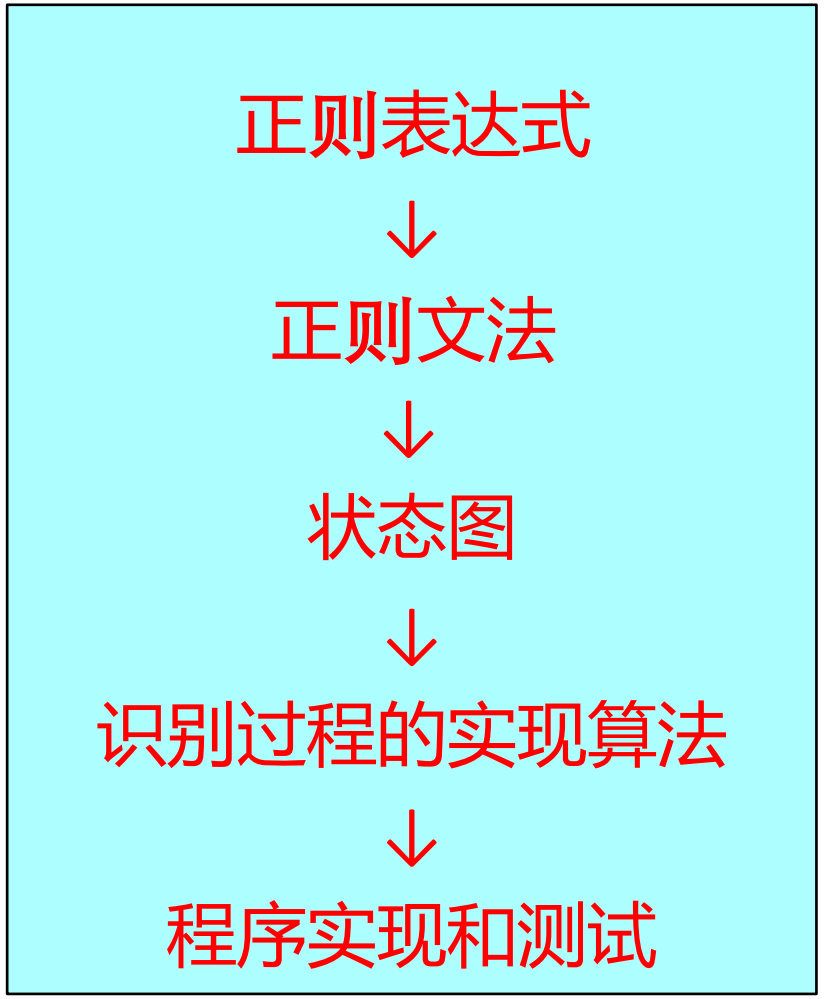






词法分析器的实现步骤

- 用正则表达式描述词法规则，设置单词种别和属性；
- 按照右图所示步骤，逐步实现词法分析器；

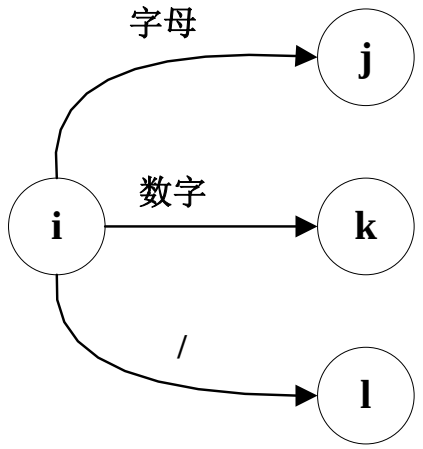


状态转换图的实现

- 程序实现:每个状态结点对应一段程序.

1) 不含回路的分叉结点:

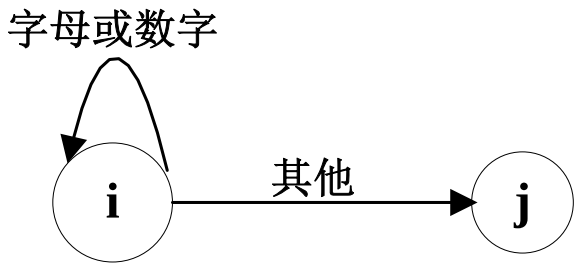
- CASE 或者 IF--THEN—ELSE




```
GetChar()
if (IsLetter())
    {状态j的对应程序段}
else if (IsDigit())
    {状态k的对应程序段}
else if (ch == '/')
    {状态l的对应程序段}
else
    {错误处理}
```

2) 含回路的分叉结:

- WHILE




```
GetChar();
while (IsLetter() or IsDigit())
      GetChar();
状态j的对应程序段
```

3) 终点结:

- RETURN(Code, Value)：返回调用者

种别编码，属性值

LEX语言的一般介绍:

LEX程序由两部分生成:

（1）正则式（辅助）定义式

由一些LEX语句组成,形式为:

$d_1 \rightarrow r_1$
$d_2 \rightarrow r_2$
…
$d_m \rightarrow r_m$

其中: r_i为正则式,它定义在$\Sigma \cup \{d_1,d_2,\ldots,d_{i-1}\}$. d_i 为其简名

（2）识别规则

LEX语句组成:

P1: {A1}
P2: {A2}
…
Pn: {An}

Pi是词形,由定义在 $\Sigma \cup \{d_1,d_2,\ldots,d_n\}$ 上的正则表达式表示.
Ai是动作,当识别出Pi后应做的工作.

例子

letter ⟶ A|B|C|…|Z
digit ⟶ 0|1|2|…|9
iden ⟶ (letter|digit)*

IF： {return(1, -)}
DO： {return(2, -)}
Iden： {return(3, getEntry())}
=： {retrun(4, -)}
…

LEX工作过程

- 首先，使用LEX语言写一个定义词法分析器的源程序lex.l。
- 然后利用LEX编译器将lex.l转换成C语言程序lex.yy.c。它包括从lex.l的正则表达式构造的状态转换表以及使用该表格识别词素的标准子程序。
- 与lex.l中正则表达式相关联的动作是C代码段，这些动作可以直接加入到lex.yy.c中。
- 最后，lex.yy.c通过C编译器生成目标程序，这个目标程序就是把输入流转换成记号序列的词法分析器。

Lex词法分析器如何工作:

P1: {A1}
P2: {A2}
…
Pn: {An}

1) 最长匹配原则:

L扫描输入串,寻找最长的子串匹配某一个Pi; 并把该子串截下放入TOKEN缓冲区; 然后调用动作Ai,把表示Pi对应的二元式送给语法分析器.

2)优先匹配原则:

- 在服从最长匹配的前提下,处于前面的Pi,匹配优先权就越高.
- 解决二义性问题

3)出错处理:

在输入串中找不到与某一个Pi匹配的子串,则要报告出错.

4) Ai返回单词的种别和内部值。在LEX程序中用 RETURN(C, LEXVAL)

LEX举例

例子：

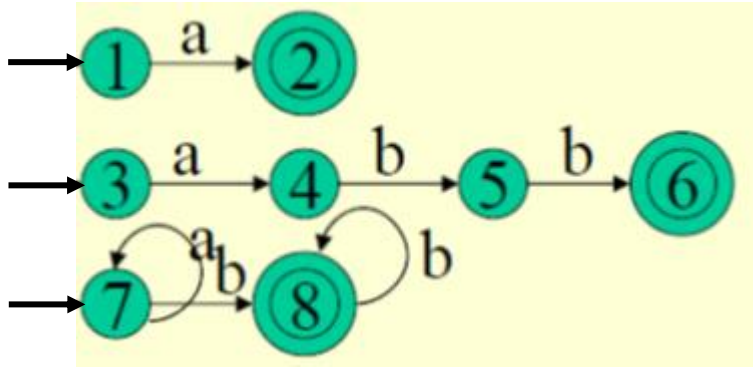
Lex 源程序

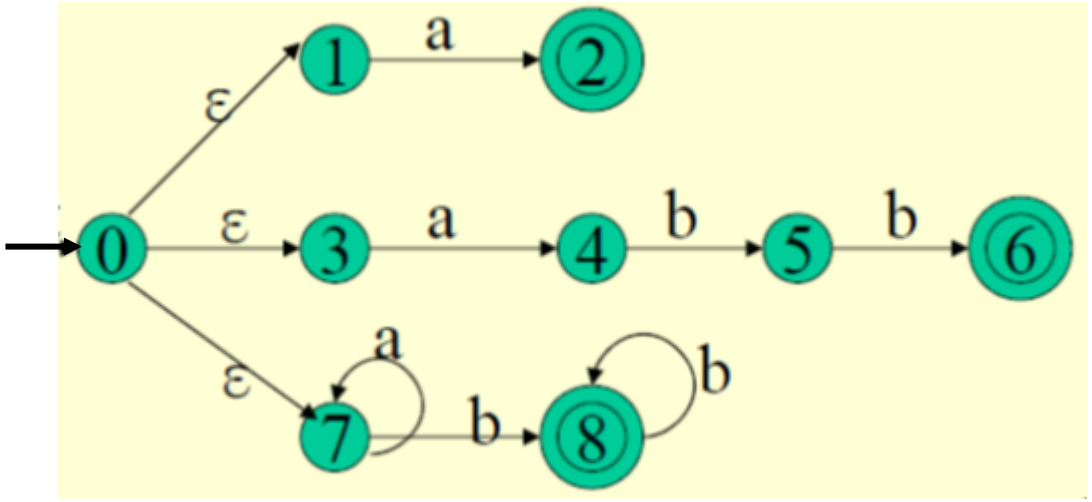
a　　{　}

abb　　{　}

a*bb*　　{　}

读入Lex源程序，生成NFA，合并成一个NFA，确定化最小化





小结

- 词法分析器的作用
- 词法分析程序的设计
  - 状态转换图
- 正规表达式和有限自动机
  - 有限自动机 （Finite Automata）
  - 正则表达式（Regular Expression）
  - 正则文法（Regular Gramma）
- 词法分析程序的自动生成

阅读材料：《程序设计语言编译原理（第3版）》，陈火旺等编著，国防工业出版社， 2004年----第三章