



山东大学
SHANDONG UNIVERSITY

编译原理

第三章 词法分析

授 课 教 师 : 郑艳伟
手 机 : 18614002860 (微信同号)
邮 箱 : zhengyw@sdu.edu.cn

□ 3.1 对词法分析器的设计

- 3.1.1 词法分析器的任务
- 3.1.2 词法分析器设计需要考虑的问题
- 3.1.3 状态转换图

□ 3.2 有限自动机

- 3.2.1 确定有限自动机
- 3.2.2 非确定有限自动机
- 3.2.3 非确定有限自动机确定化
- 3.2.4 确定有限自动机化简
- 3.2.5 正规式与有限自动机的等价性

□ 3.3 正规文法

- 3.3.1 右线性文法转有限自动机
- 3.3.2 左线性文法转有限自动机

- 3.3.3 有限自动机转右线性文法
- 3.3.4 有限自动机转左线性文法
- 3.3.5 正规式转右线性文法
- 3.3.6 正规式转左线性文法
- 3.3.7 正规文法转正规式
- 3.3.8 三种工具的转换
- 3.3.9 有限自动机转正规式

□ 3.4 词法分析器的实现

- 3.4.1 词法分析器边界
- 3.4.2 单词正规式
- 3.4.3 识别单词的DFA
- 3.4.4 单词识别算法

□ 3.1 对词法分析器的设计

➤ 3.1.1 词法分析器的任务

- 3.1.2 词法分析器设计需要考虑的问题
- 3.1.3 状态转换图

□ 3.2 有限自动机

- 3.2.1 确定有限自动机
- 3.2.2 非确定有限自动机
- 3.2.3 非确定有限自动机确定化
- 3.2.4 确定有限自动机化简
- 3.2.5 正规式与有限自动机的等价性

□ 3.3 正规文法

- 3.3.1 右线性文法转有限自动机
- 3.3.2 左线性文法转有限自动机

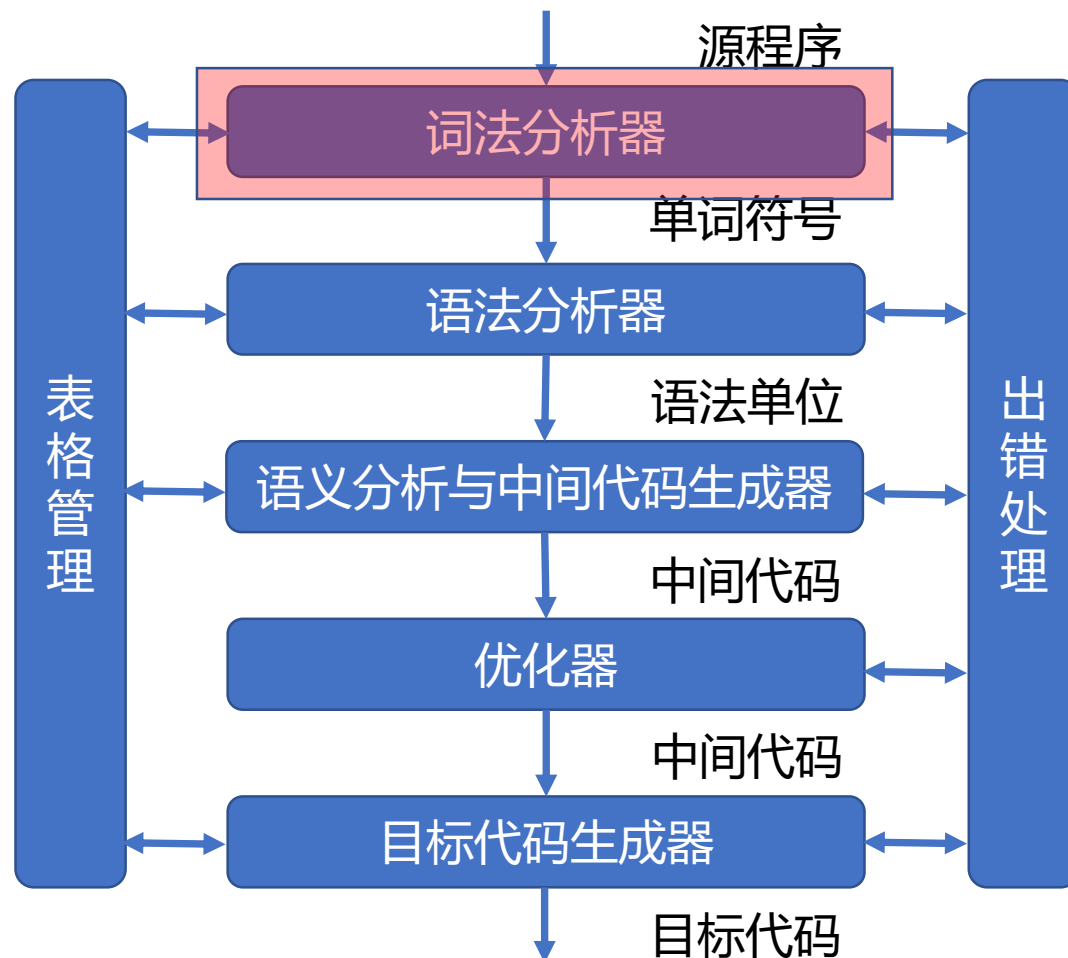
- 3.3.3 有限自动机转右线性文法
- 3.3.4 有限自动机转左线性文法
- 3.3.5 正规式转右线性文法
- 3.3.6 正规式转左线性文法
- 3.3.7 正规文法转正规式
- 3.3.8 三种工具的转换
- 3.3.9 有限自动机转正规式

□ 3.4 词法分析器的实现

- 3.4.1 词法分析器边界
- 3.4.2 单词正规式
- 3.4.3 识别单词的DFA
- 3.4.4 单词识别算法

3.1.1 词法分析器的任务

□ **词法分析器**，又称**扫描器**，其任务是从左到右逐个字符地对源程序进行扫描，产生一个个的单词符号，把作为字符串的源程序改造为单词符号串的中程序。



3.1.1 词法分析器的任务

```
1  int Fun(int x, int y) {  
2      int a, b, var;  
3      a = x + y;  
4      b = x + y;  
5      var = a * b;  
6      return var;  
7  }
```

(1) (int, 关键字)

(2) (Fun, 标识符)

(3) ((, 界符)

(4) (int, 关键字)

(5) (x, 标识符)

(6) (,, 界符)

(7) (int, 关键字)

(8) (y, 标识符)

(9) (,), 界符)

(10) ({, 界符)

(11) (int, 关键字)

(12) (a, 标识符)

(13) (,, 界符)

(14) (b, 标识符)

(15) (,, 界符)

(16) (var, 标识符)

(17) (;, 界符)

(18) (a, 标识符)

(19) (=, 运算符)

(20) (x, 标识符)

(21) (+, 运算符)

(22) (y, 标识符)

(23) (;, 界符)

(24) (b, 标识符)

(25) (=, 运算符)

(26) (x, 标识符)

(27) (+, 运算符)

(28) (y, 标识符)

(29) (;, 界符)

(30) (var, 标识符)

(31) (=, 运算符)

(32) (a, 标识符)

(33) (*, 运算符)

(34) (b, 标识符)

(35) (;, 界符)

(36) (return, 关键字)

(37) (var, 标识符)

(38) (;, 界符)

(39) (}, 界符)

3.1.1 词法分析器的任务

□ 程序语言的单词符号一般分为以下5种：

- **关键字**：由程序语言定义的有固定意义的标识符，有时称为**保留字**或**基本字**，如C语言中的int、while、if等。
- **标识符**：用来表示各种名字，如变量名、数组名、过程名等。
- **常数**：一般有整型、实型、布尔型、文字型等等，如100, 3.1415926, true, “sample”。
- **运算符**：如+、-、*、/等。
- **界符**：如逗号、分号、括号、//、/*、*/等等。

□ **说明**

- **基本字**、**运算符**和**界符**都是确定的，一般只有几十个或上百个。
- **标识符**和**常数**的数量一般不加限制，**标识符**一般有长度限制。

3.1.1 词法分析器的任务

- **单词种别编码**：一个语言的单词符号如何分种、分几种、怎样编码，是一个技术性问题，主要取决于处理上的方便
 - **标识符**一般统归为一种。
 - **常数**宜按类型（整型、实型、布尔型等）分种。
 - **关键字**可以将全体视为一种，也可以一字一种。
 - **运算符**可以一符一种，也可以把具有一定共性的运算符视为一种。
 - **界符**一般一符一种。

□ 3.1 对词法分析器的设计

- 3.1.1 词法分析器的任务
- 3.1.2 词法分析器设计需要考虑的问题
- 3.1.3 状态转换图

□ 3.2 有限自动机

- 3.2.1 确定有限自动机
- 3.2.2 非确定有限自动机
- 3.2.3 非确定有限自动机确定化
- 3.2.4 确定有限自动机化简
- 3.2.5 正规式与有限自动机的等价性

□ 3.3 正规文法

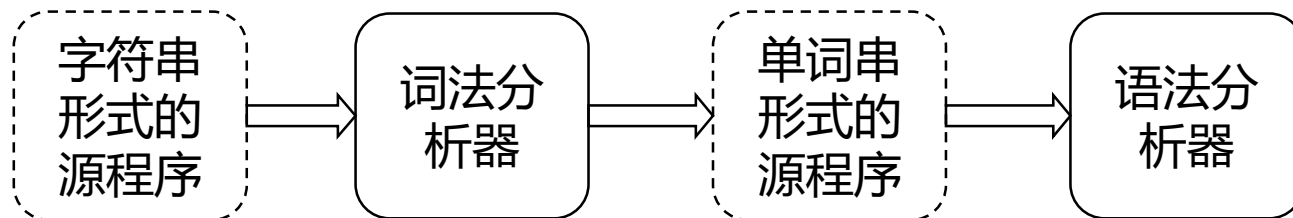
- 3.3.1 右线性文法转有限自动机
- 3.3.2 左线性文法转有限自动机

- 3.3.3 有限自动机转右线性文法
- 3.3.4 有限自动机转左线性文法
- 3.3.5 正规式转右线性文法
- 3.3.6 正规式转左线性文法
- 3.3.7 正规文法转正规式
- 3.3.8 三种工具的转换
- 3.3.9 有限自动机转正规式

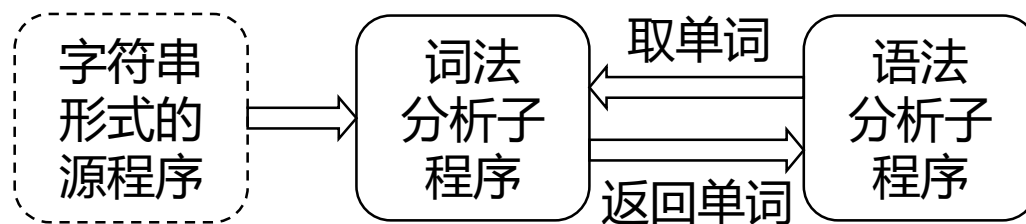
□ 3.4 词法分析器的实现

- 3.4.1 词法分析器边界
- 3.4.2 单词正规式
- 3.4.3 识别单词的DFA
- 3.4.4 单词识别算法

词法分析器独立一遍还是作为一个独立子程序



词法分析器作为独立的一遍



词法分析器作为一个独立子程序

注释和空白的预处理

□ **预处理**主要是为方便单词的识别工作，**也可以与词法分析一起处理**

- 将跳格符、回车符、换行符等替换为空白符，并将连续空白合并为1个：‘ ‘, ‘\t’, ‘\r’, ‘\n’
- 剔除注释： /*.....*/, //

【例】`int max(int x, int y)// 求x,y的最大值`
`{`
 `int z;`
 `z = (x > y ? x : y);`
 `return z;`
`}`

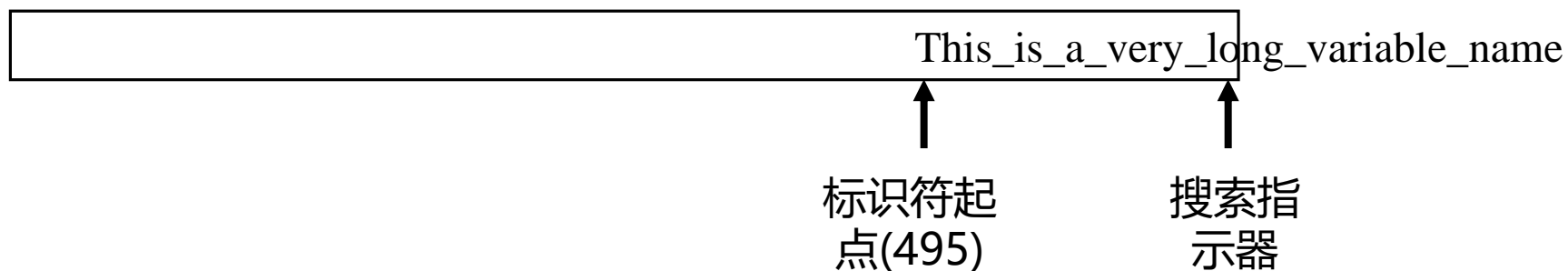
预处理后：

`int max(int x,int y) {int z; z=(x>y ? x : y); return z;}`

输入的双缓冲设计

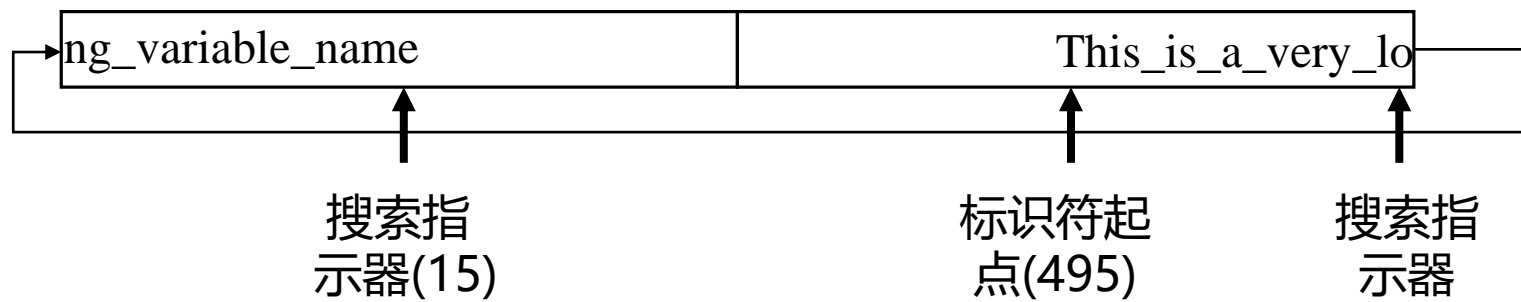
□ 假设标识符和常数的最大长度为256

缓冲区长度: 512



缓冲区长度: 256

缓冲区长度: 256



□ 3.1 对词法分析器的设计

- 3.1.1 词法分析器的任务
- 3.1.2 词法分析器设计需要考虑的问题
- 3.1.3 状态转换图

□ 3.2 有限自动机

- 3.2.1 确定有限自动机
- 3.2.2 非确定有限自动机
- 3.2.3 非确定有限自动机确定化
- 3.2.4 确定有限自动机化简
- 3.2.5 正规式与有限自动机的等价性

□ 3.3 正规文法

- 3.3.1 右线性文法转有限自动机
- 3.3.2 左线性文法转有限自动机

- 3.3.3 有限自动机转右线性文法
- 3.3.4 有限自动机转左线性文法
- 3.3.5 正规式转右线性文法
- 3.3.6 正规式转左线性文法
- 3.3.7 正规文法转正规式
- 3.3.8 三种工具的转换
- 3.3.9 有限自动机转正规式

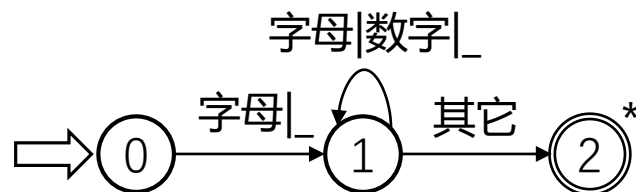
□ 3.4 词法分析器的实现

- 3.4.1 词法分析器边界
- 3.4.2 单词正规式
- 3.4.3 识别单词的DFA
- 3.4.4 单词识别算法

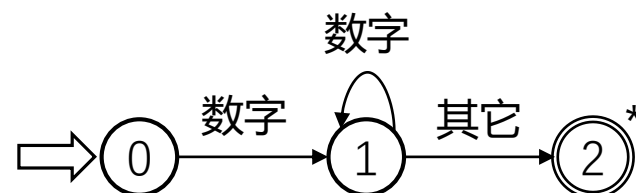
状态转换图

□ 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，至少要有有一个终态。
- 初态加箭头指向表示，终态用双圆圈表示。
- 终态结点上打星号表示多读进一个不属于标识符的字符，应退回给输入串。



识别标识符的状态转换图



识别整数的状态转换图

超前搜索

□ **关键字**的识别: Fortran确定如下DO和IF是否为关键字, 需要超前扫描

- ① DO99K=1,10 ! 等价于for(K=1;i<=10;i++), 99是循环体最后一行
- ② DO99K=1.10 ! 变量赋值一个实型常数1.10
- ③ IF(5.EQ.M)I=10 ! F77中.EQ.表示等于, F90改为关系运算符=
- ④ IF(5)=77 ! 数组IF的第5个元素赋值为整型常数77

□ **标识符**的识别: 一般是字母开头的“字母/数字”串。

□ **常数**的识别: Fortran中有5.E08这种常数, 因此③需要扫描到Q才能确定5是常数。

□ **算符和界符**的识别: 如果C++和Java中的++、--、>=等算符, 需要超前搜索

超前搜索

□ 避免超前搜索的几点重要限制

- 所有基本字都是保留字，用户不能用它们作自己的标识符。
- 基本字作为特殊标识符来处理，不用特殊的状态图来识别，只要查保留字表。
(适用Fortran)
- 如果基本字、标识符和常数(或标号)之间没有确定的运算符或界符作间隔，则必须使用一个空白符作间隔。如DO99K=1,10要写成：DO 99 K=1, 10

3.1.3 状态转换图

□ 思考

- 实型常数怎么设计? 12.5, +12.5, -12.5, 1e-1, 1.e-1, 1.12e-1, .123e-1, 1e1, 1e+1, ...
- 如何把多个单词种类的自动机合并成一个?

□ 这些问题请保持好奇心, 等到学完有限状态自动机再考虑, 会发现非常简单。

□ 3.1 对词法分析器的设计

- 3.1.1 词法分析器的任务
- 3.1.2 词法分析器设计需要考虑的问题
- 3.1.3 状态转换图

□ 3.2 有限自动机

- 3.2.1 确定有限自动机
- 3.2.2 非确定有限自动机
- 3.2.3 非确定有限自动机确定化
- 3.2.4 确定有限自动机化简
- 3.2.5 正规式与有限自动机的等价性

□ 3.3 正规文法

- 3.3.1 右线性文法转有限自动机
- 3.3.2 左线性文法转有限自动机

- 3.3.3 有限自动机转右线性文法
- 3.3.4 有限自动机转左线性文法
- 3.3.5 正规式转右线性文法
- 3.3.6 正规式转左线性文法
- 3.3.7 正规文法转正规式
- 3.3.8 三种工具的转换
- 3.3.9 有限自动机转正规式

□ 3.4 词法分析器的实现

- 3.4.1 词法分析器边界
- 3.4.2 单词正规式
- 3.4.3 识别单词的DFA
- 3.4.4 单词识别算法

3.2 有限自动机

□ 当一个问题比较复杂时，通常从两个角度切入：

- 从人的角度，设计一个人类比较容易理解、且容易构造的规则或模型。对单词描述来说，第2.3.1节介绍的正规式正是这样的一个工具。
- 从计算机的角度，设计一个计算机算法比较容易实现的模型。前面所述的状态转换图是一个好的工具，但是需要对其施加一定的限制，后面定义为确定有限自动机。
- 然后，我们需要寻找一个算法，实现两种模型的转换。

□ 3.1 对词法分析器的设计

- 3.1.1 词法分析器的任务
- 3.1.2 词法分析器设计需要考虑的问题
- 3.1.3 状态转换图

□ 3.2 有限自动机

- 3.2.1 确定有限自动机
- 3.2.2 非确定有限自动机
- 3.2.3 非确定有限自动机确定化
- 3.2.4 确定有限自动机化简
- 3.2.5 正规式与有限自动机的等价性

□ 3.3 正规文法

- 3.3.1 右线性文法转有限自动机
- 3.3.2 左线性文法转有限自动机

- 3.3.3 有限自动机转右线性文法
- 3.3.4 有限自动机转左线性文法
- 3.3.5 正规式转右线性文法
- 3.3.6 正规式转左线性文法
- 3.3.7 正规文法转正规式
- 3.3.8 三种工具的转换
- 3.3.9 有限自动机转正规式

□ 3.4 词法分析器的实现

- 3.4.1 词法分析器边界
- 3.4.2 单词正规式
- 3.4.3 识别单词的DFA
- 3.4.4 单词识别算法

确定有限自动机

□ 一个**确定有限自动机(DFA)** M 是一个五元式: $M = (S, \Sigma, \delta, s_0, F)$, 其中:

- ① S 是一个**有限状态集**, 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 δ 是一个从 $S \times \Sigma$ 至 S 的**单值映射**, $\delta(s, a) = s'$ 意味着, 当当前状态为 s 、输入字符为 a 时, 将转换到下一个状态 s' , 称 s' 为 s 的一个**后继状态**;
- ④ $s_0 \in S$, 是**唯一的初态**;
- ⑤ $F \subseteq S$, 是一个**终态集** (可空)。

状态转换矩阵

□ 一个DFA可以用一个矩阵表示, 该矩阵的行表示状态, 列表示输入字符, 矩阵元素表示 $\delta(s, a)$ 的值, 这个矩阵称为**状态转换矩阵**。


【例】有DFA $M = (\{0,1,2,3\}, \{a, b\}, \delta, 0, \{3\})$, 其中 δ 为:

$$\delta(0, a) = 1 \quad \delta(0, b) = 2 \quad \delta(1, a) = 3 \quad \delta(1, b) = 2$$

$$\delta(2, a) = 1 \quad \delta(2, b) = 3 \quad \delta(3, a) = 3 \quad \delta(3, b) = 3$$

对应的状态转换矩阵为:

状态	a	b
0	1	2
1	3	2
2	1	3
3	3	3

 $\delta = \begin{pmatrix} 1 & 2 \\ 3 & 2 \\ 1 & 3 \\ 3 & 3 \end{pmatrix}$

状态转换图

□ 一个DFA可以表示为一张状态转换图。

- 对 Σ^* 上的任何字 α , 若存在一条从初态结点到某一终态结点的通路, 且这条通路上所有弧的标记符连接成的字等于 α , 则称 α 可为DFA M 所识别 (读出或接受)
- 若 M 的初态结点同时又是终态结点, 则空字 ε 可为 M 所识别 (或接受)。
- DFA M 所能识别的字的全体记为 $L(M)$ 。
- Σ 上的字集 $V \subseteq \Sigma^*$ 是正规的, 等价于存在 Σ 上的DFA M , 使得 $V = L(M)$ 。

【例】有DFA $M = (\{0,1,2,3\}, \{a, b\}, \delta, 0, \{3\})$, 其中 δ 为:

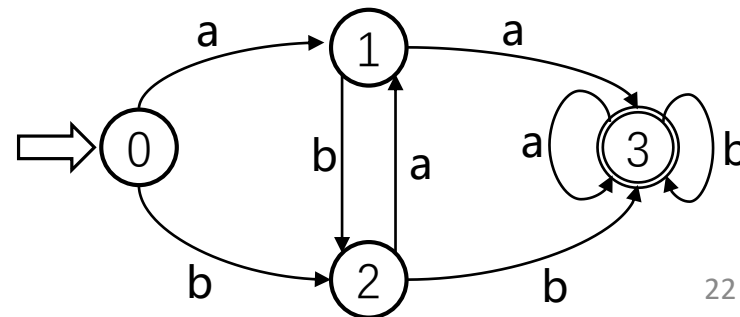
$$\delta(0, a) = 1 \quad \delta(0, b) = 2 \quad \delta(1, a) = 3 \quad \delta(1, b) = 2$$

$$\delta(2, a) = 1 \quad \delta(2, b) = 3 \quad \delta(3, a) = 3 \quad \delta(3, b) = 3$$

对应的状态转换图如右图。

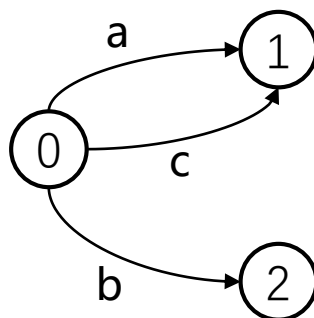
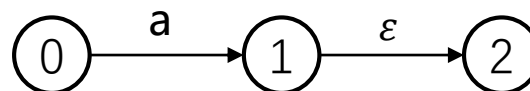
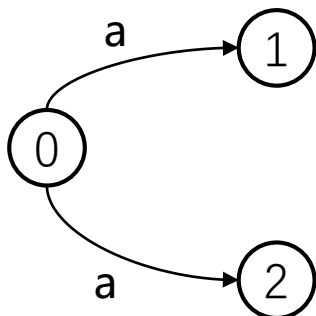
识别的字:

- 包含两个连续 a 或者两个连续 b 的字。



DFA的确定性

□ DFA的**确定性**表现在存在映射 $\delta: S \times \Sigma \rightarrow S$ 。



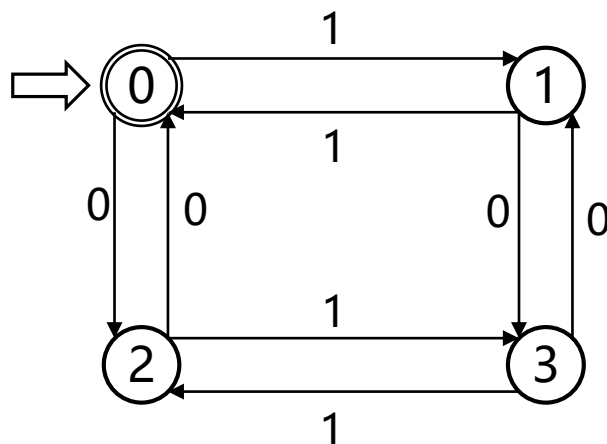
DFA例题

【例】设计DFA M , 使其识别偶数个0偶数个1的字 (含空字)。

【分析】问题的状态空间

- 0: 偶数个0偶数个1
- 1: 偶数个0奇数个1
- 2: 奇数个0偶数个1
- 3: 奇数个0奇数个1

输入一个字符后从一个状态转换到另外一个状态。

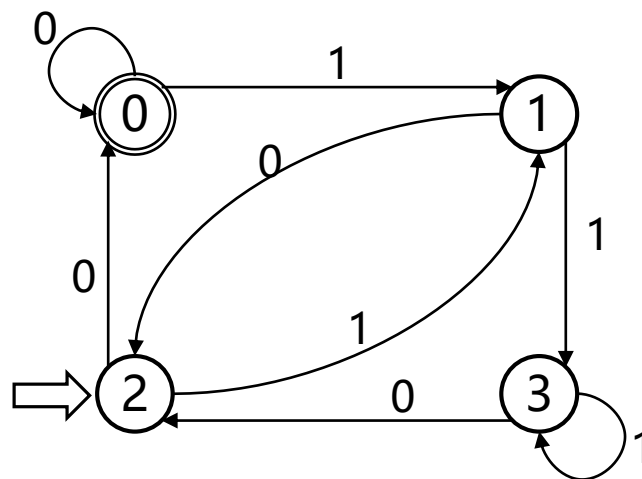


DFA例题

【例】设计DFA M , 使其接受 $\Sigma = \{0,1\}$ 上能被4整除的二进制数。

【分析】问题的状态空间

- 任意二进制数除以4, 只有余数为0、1、2、3四种情况。
- 一个二进制数后面加0, 变为原来的2倍; 后面加1, 变为原来的2倍加1。



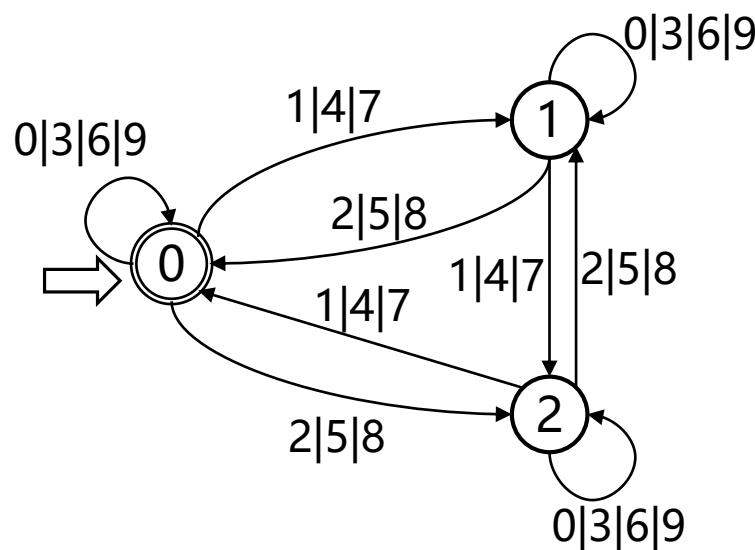
初态确定: 接受字
“0” 的只有状态
0,2; 去掉空字。

DFA例题

【例3】设计DFA M , 使其接受 $\Sigma = \{0, 1, \dots, 9\}$ 上能被3整除的十进制数。

【分析】问题的状态空间

- 任意十进制数除以3, 只有余数为0、1、2三种情况。
- 一个十进制 n 数后面加 i , 变为 $10n + i$ 。



初态只能是0, 但会接受空字。

DFA例题

【例】 一个人带着狼、山羊和白菜要从一条河左岸渡到右岸。有一条船，恰好能装下人和其它三件东西中的一件。用确定有限自动机找出渡河方案。

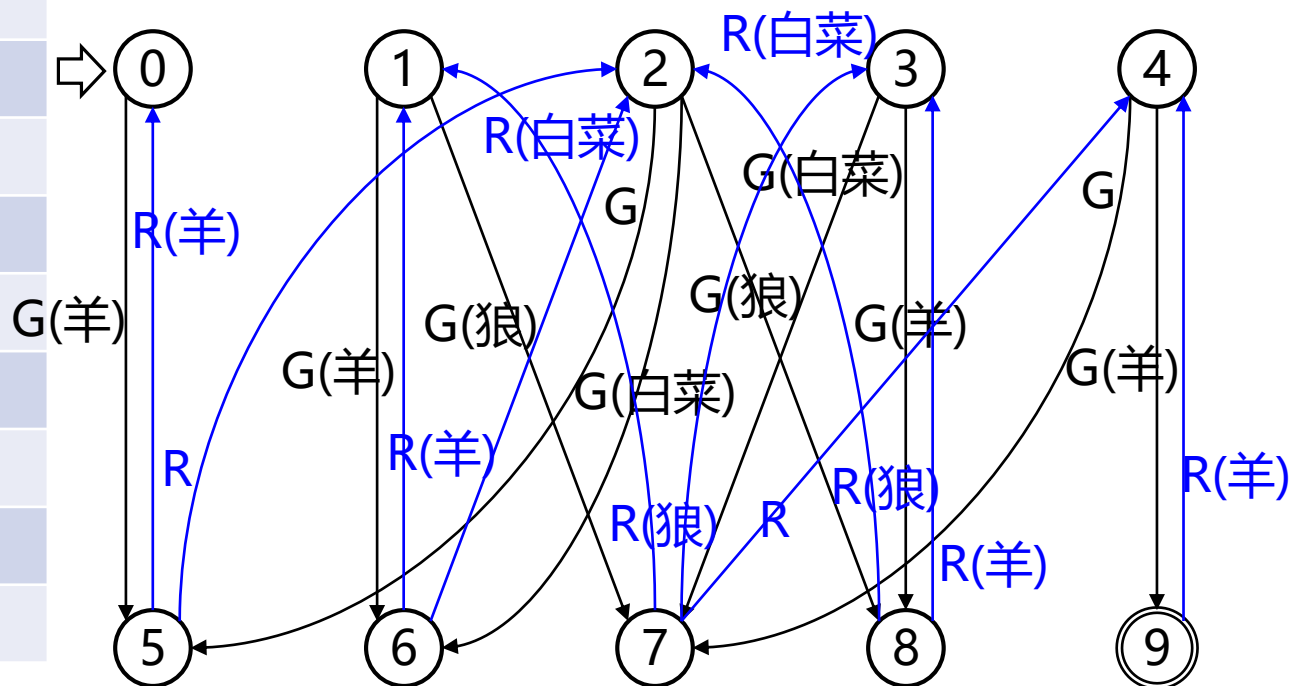
【分析】 问题的状态空间

➤ 左岸存在不同东西组合作为状态；初态为{人, 狼, 羊, 白菜}, 终态为 Φ 。

{人, 狼, 羊, 白菜}	➡ 0	{狼, 羊}	✗
{人, 狼, 羊}	➡ 1	{狼, 白菜}	➡ 5
{人, 狼, 白菜}	➡ 2	{羊, 白菜}	✗
{人, 羊, 白菜}	➡ 3	{人}	✗
{狼, 羊, 白菜}	✗	{狼}	➡ 6
{人, 狼}	✗	{羊}	➡ 7
{人, 羊}	➡ 4	{白菜}	➡ 8
{人, 白菜}	✗	Φ	➡ 9

编号	状态
0	{人, 狼, 羊, 白菜}
1	{人, 狼, 羊}
2	{人, 狼, 白菜}
3	{人, 羊, 白菜}
4	{人, 羊}
5	{狼, 白菜}
6	{狼}
7	{羊}
8	{白菜}
9	Φ

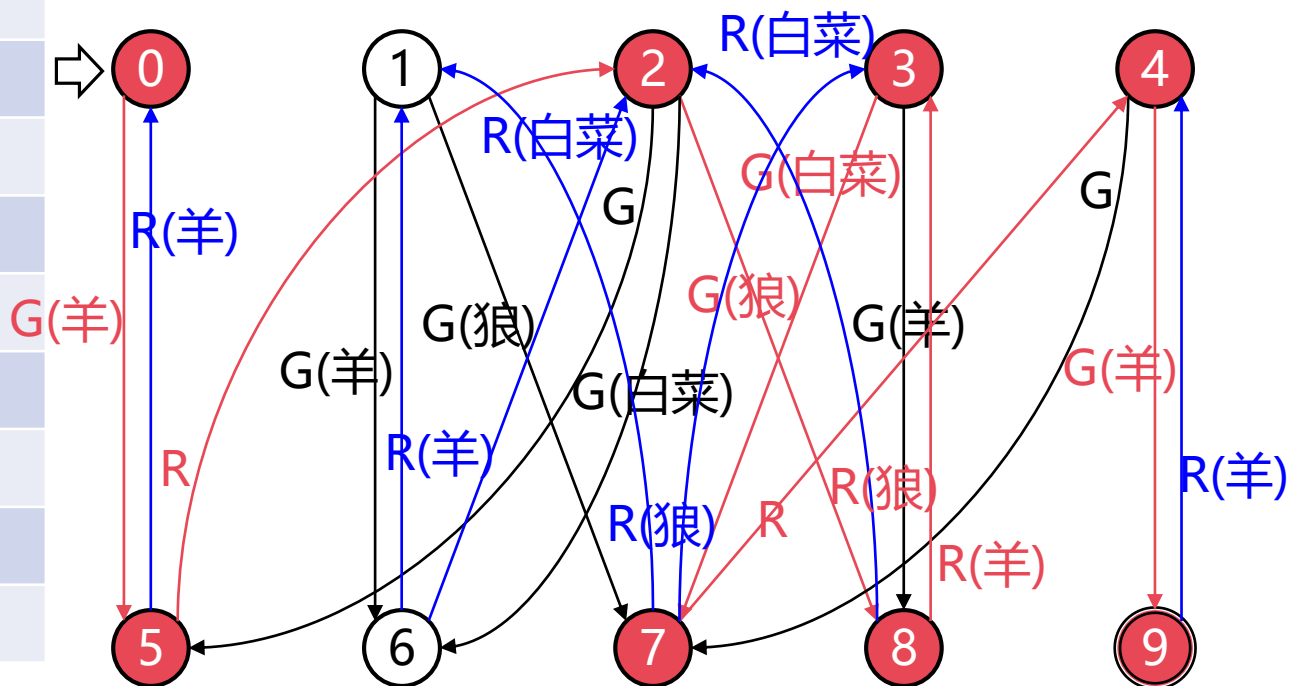
DFA例题



- G(x): 人带x从左岸渡河
- R(x): 人带x从右岸返回
- G: 人自己从左岸渡河
- R: 人自己从右岸返回

DFA例题

编号	状态
0	{人, 狼, 羊, 白菜}
1	{人, 狼, 羊}
2	{人, 狼, 白菜}
3	{人, 羊, 白菜}
4	{人, 羊}
5	{狼, 白菜}
6	{狼}
7	{羊}
8	{白菜}
9	Φ



▣ $0 \rightarrow 5 \rightarrow 2 \rightarrow 8 \rightarrow 3 \rightarrow 7 \rightarrow 4 \rightarrow 9$

➤ $0 \rightarrow 5$ G(羊): 5{狼, 白菜}

➤ $5 \rightarrow 2$ R: 2{人, 狼, 白菜}

➤ $2 \rightarrow 8$ G(狼): 8{白菜}

➤ $8 \rightarrow 3$ R(羊): 3{人, 羊, 白菜}

➤ $3 \rightarrow 7$ G(白菜): 7{羊}

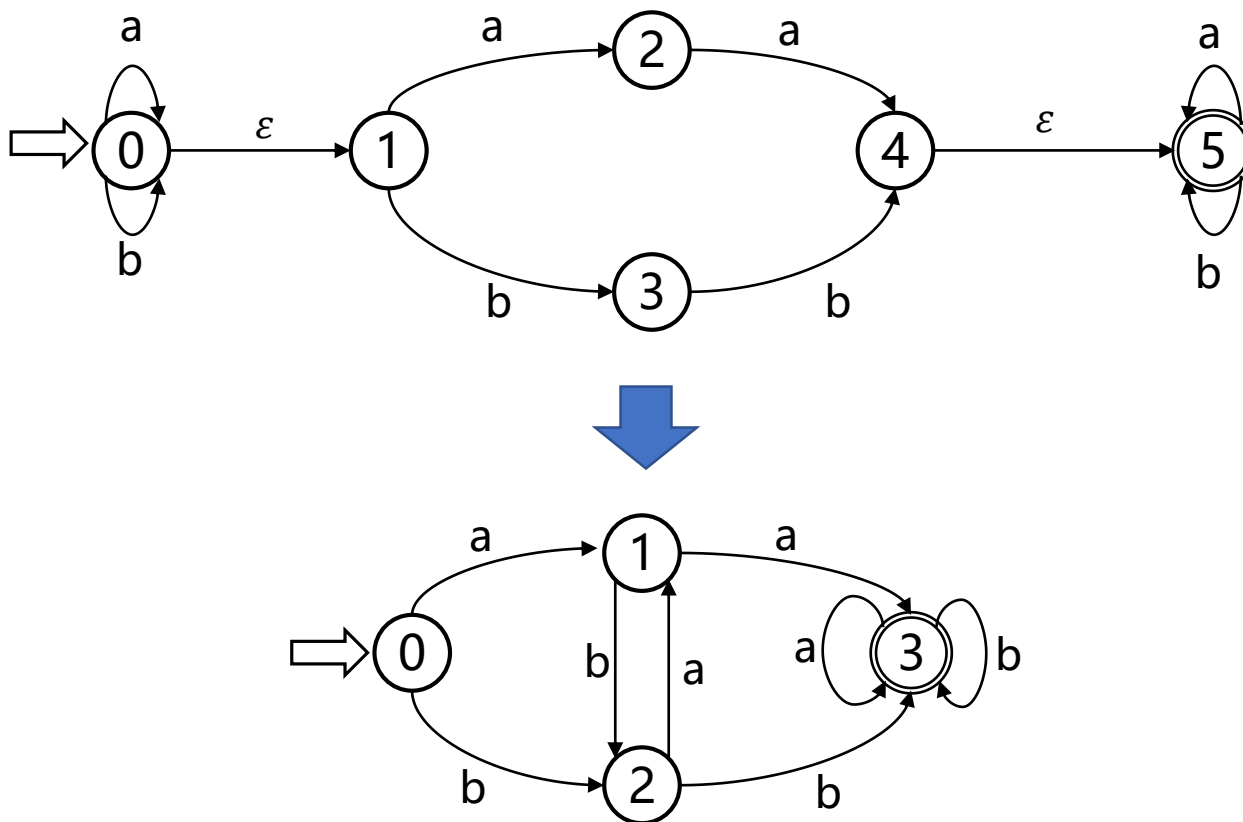
➤ $7 \rightarrow 4$ R: 4{人, 羊}

➤ $4 \rightarrow 9$ G(羊): Φ

DFA问题



□ 任给一个正规式, 如 $(a|b)^*(aa|bb)(a|b)^*$, 如何构造DFA?



□ 3.1 对词法分析器的设计

- 3.1.1 词法分析器的任务
- 3.1.2 词法分析器设计需要考虑的问题
- 3.1.3 状态转换图

□ 3.2 有限自动机

- 3.2.1 确定有限自动机
- 3.2.2 非确定有限自动机
- 3.2.3 非确定有限自动机确定化
- 3.2.4 确定有限自动机化简
- 3.2.5 正规式与有限自动机的等价性

□ 3.3 正规文法

- 3.3.1 右线性文法转有限自动机
- 3.3.2 左线性文法转有限自动机

- 3.3.3 有限自动机转右线性文法
- 3.3.4 有限自动机转左线性文法
- 3.3.5 正规式转右线性文法
- 3.3.6 正规式转左线性文法
- 3.3.7 正规文法转正规式
- 3.3.8 三种工具的转换
- 3.3.9 有限自动机转正规式

□ 3.4 词法分析器的实现

- 3.4.1 词法分析器边界
- 3.4.2 单词正规式
- 3.4.3 识别单词的DFA
- 3.4.4 单词识别算法

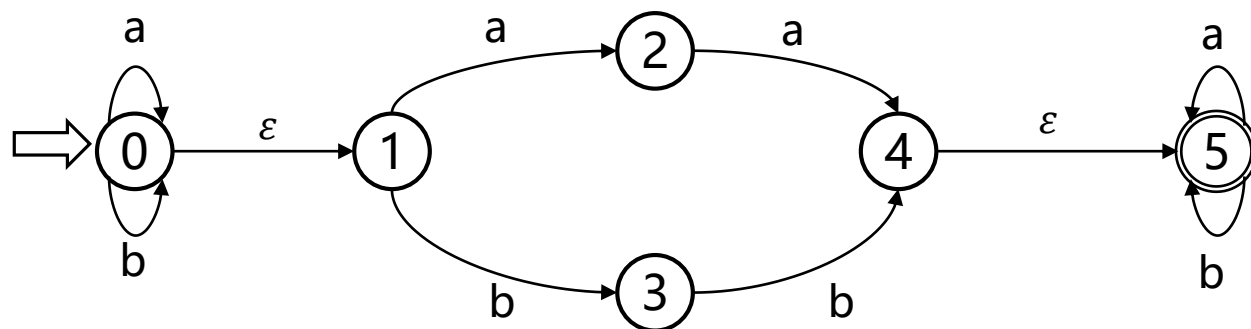
非确定有限自动机

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

- ① S 是一个有限集, 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 δ 是一个从 $S \times \Sigma^*$ 至 S 的子集的映射, 即: $\delta: S \times \Sigma^* \rightarrow 2^S$;
- ④ $S_0 \subseteq S$, 是非空初态集;
- ⑤ $F \subseteq S$, 是一个终态集 (可空)。

非确定有限自动机

【例】



NFA $M = (\{0,1,2,3,4,5\}, \{a,b\}, \delta, \{0\}, \{5\})$

δ : $\delta(0,a) = \{0,1\}$, $\delta(0,b) = \{0,1\}$,

$\delta(1,a) = \{2\}$, $\delta(1,b) = \{3\}$,

$\delta(2,a) = \{4,5\}$, $\delta(2,b) = \Phi$,

$\delta(3,a) = \Phi$, $\delta(3,b) = \{4,5\}$,

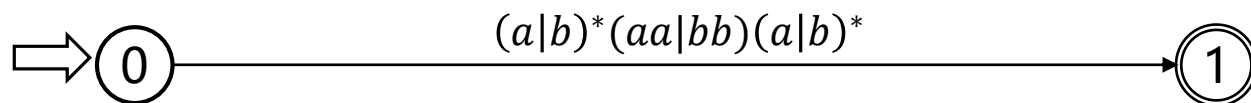
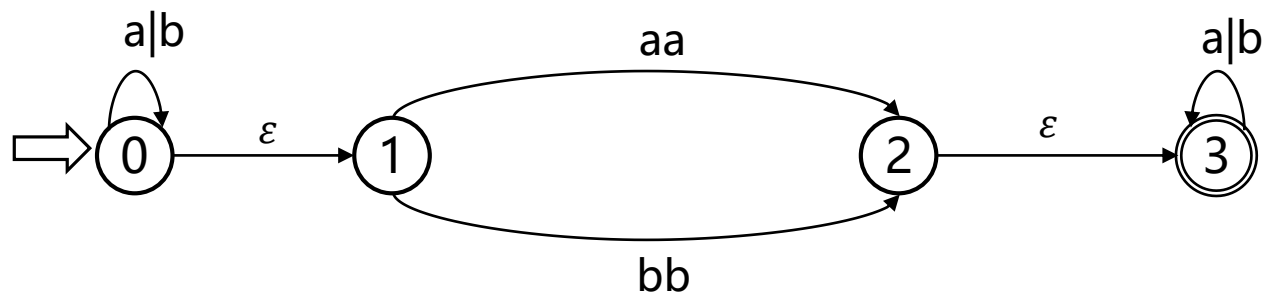
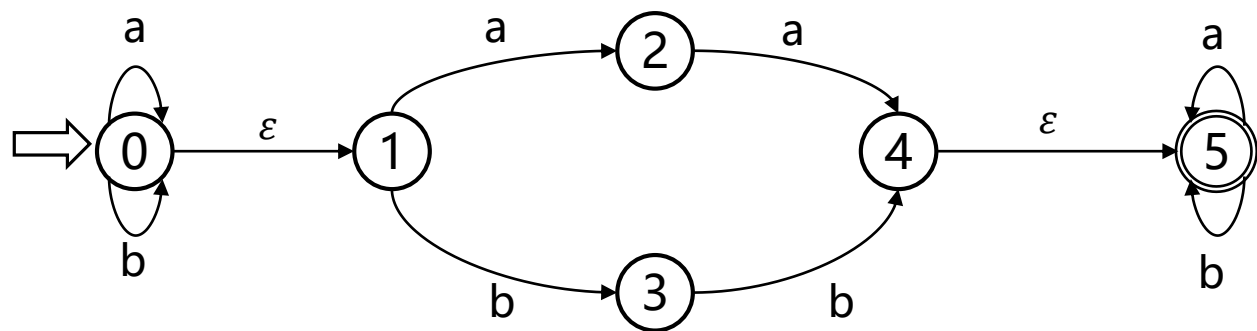
$\delta(4,a) = \{5\}$, $\delta(4,b) = \{5\}$,

$\delta(5,a) = \{5\}$, $\delta(5,b) = \{5\}$.

状态	a	b
0	{0,1}	{0,1}
1	{2}	{3}
2	{4,5}	Φ
3	Φ	{4,5}
4	{5}	{5}
5	{5}	{5}

非确定有限自动机

【例】构造NFA: $(a|b)^*(aa|bb)(a|b)^*$



非确定有限自动机

□ NFA与DFA的优缺点比较

- DFA编程实现容易, 效率高, 但构造困难
- NFA构造容易, 但编程实现有回溯

□ NFA是否能转换为DFA?

- $\text{DFA} \subseteq \text{NFA}$
- 对每一个NFA M , 都存在一个DFA M' , 使得 $L(M) = L(M')$
- 即: $\text{DFA} \Leftrightarrow \text{NFA}$

□ 3.1 对词法分析器的设计

- 3.1.1 词法分析器的任务
- 3.1.2 词法分析器设计需要考虑的问题
- 3.1.3 状态转换图

□ 3.2 有限自动机

- 3.2.1 确定有限自动机
- 3.2.2 非确定有限自动机
- 3.2.3 非确定有限自动机确定化
- 3.2.4 确定有限自动机化简
- 3.2.5 正规式与有限自动机的等价性

□ 3.3 正规文法

- 3.3.1 右线性文法转有限自动机
- 3.3.2 左线性文法转有限自动机

- 3.3.3 有限自动机转右线性文法
- 3.3.4 有限自动机转左线性文法
- 3.3.5 正规式转右线性文法
- 3.3.6 正规式转左线性文法
- 3.3.7 正规文法转正规式
- 3.3.8 三种工具的转换
- 3.3.9 有限自动机转正规式

□ 3.4 词法分析器的实现

- 3.4.1 词法分析器边界
- 3.4.2 单词正规式
- 3.4.3 识别单词的DFA
- 3.4.4 单词识别算法

NFA确定化

□ NFA $M = (S, \Sigma, \delta, S_0, F)$ 确定化为 DFA $M_1 = (S_1, \Sigma_1, \delta_1, s_{10}, F_1)$

- 状态集、字母表、终态集不需要转换;
- 初态, NFA 有多个, DFA 有一个, 需要使初态唯一; 为统一使终态也唯一;
- 转换函数: DFA $\delta: S \times \Sigma \rightarrow S$ vs. NFA $\delta: S \times \Sigma^* \rightarrow 2^S$ 。

□ $S \times \Sigma^* \rightarrow 2^S \Rightarrow S \times \Sigma \rightarrow S$

- $S \times \Sigma^* \rightarrow 2^S \Rightarrow S \times (\Sigma \cup \{\varepsilon\}) \rightarrow 2^S$
 $\Rightarrow S \times \Sigma \rightarrow S$

□ NFA确定化步骤

- 初态终态唯一化
- 箭弧单符化
- NFA确定化

初态终态唯一化

(1) 使初态、终态唯一

算法 3.2 使 NFA 初态和终态唯一

输入: NFA $M = (S, \Sigma, \delta, S_0, F)$

输出: NFA $M' = (S', \Sigma', \delta', S'_0, F')$

1 NFA makeSingleStartAndEndState(M):

2 $S' = S \cup \{X, Y\}, \Sigma' = \Sigma, \delta' = \delta, S'_0 = \{X\}, F' = \{Y\};$

3 foreach $s \in S_0$ do

4 $\delta' \cup = \{\delta(X, \varepsilon) = s\};$

5 end

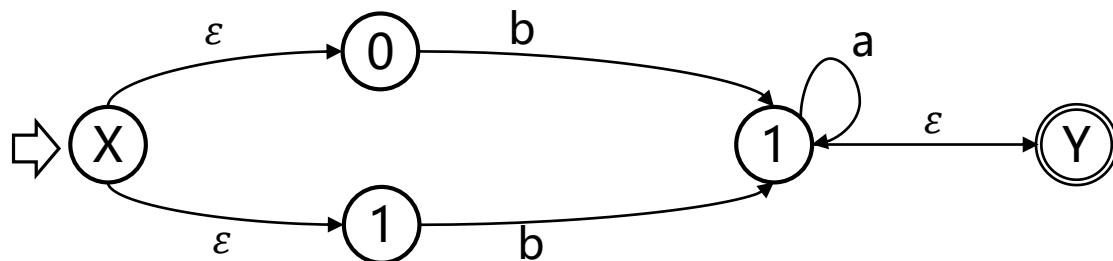
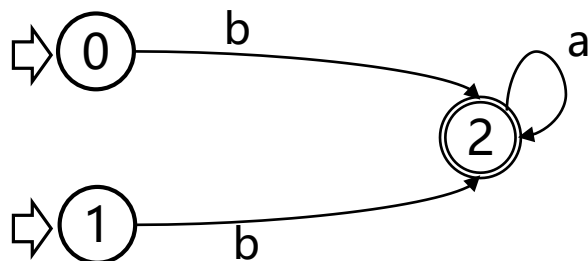
6 foreach $s \in F_0$ do

7 $\delta' \cup = \{\delta(s, \varepsilon) = Y\};$

8 end

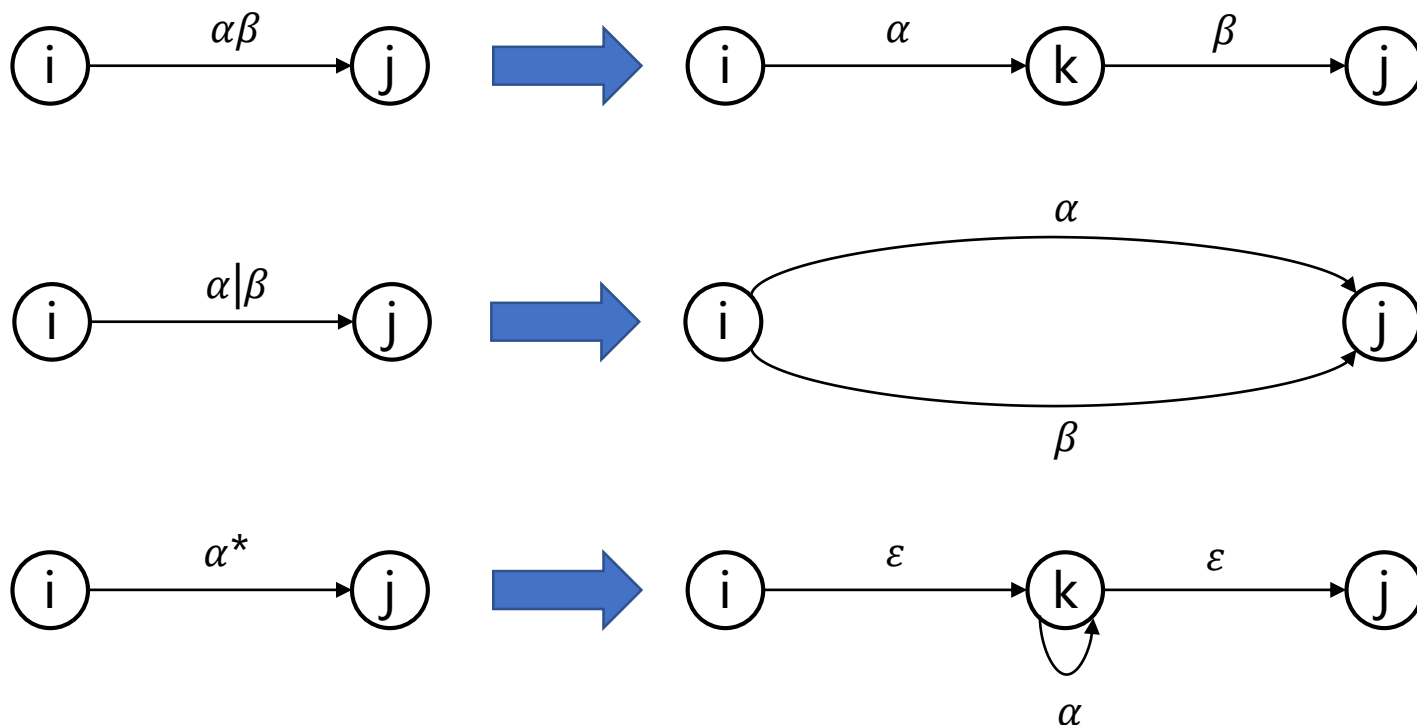
9 return M' ;

10 end makeSingleStartAndEndState



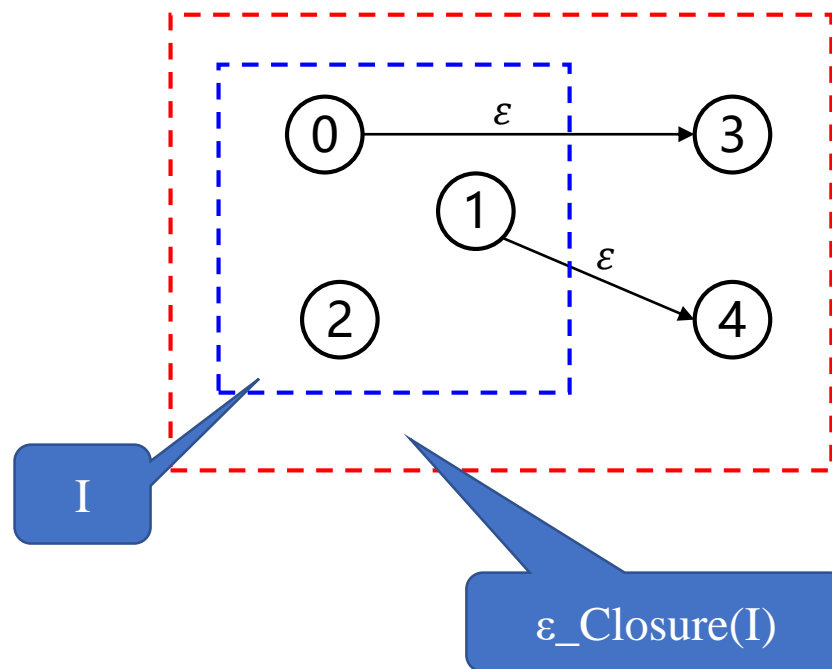
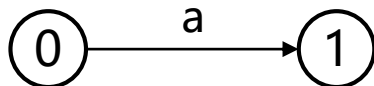
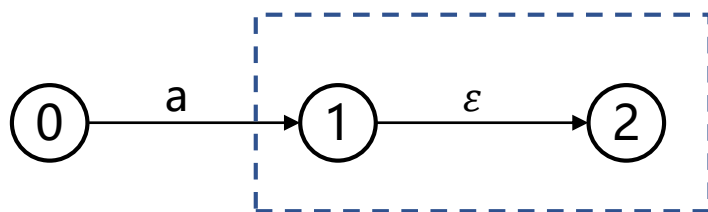
箭弧单符化

(2) 分裂, 直至每条箭弧上或为 ε , 或为 Σ 中的单个字符



NFA确定化

(3) 寻找可合并状态

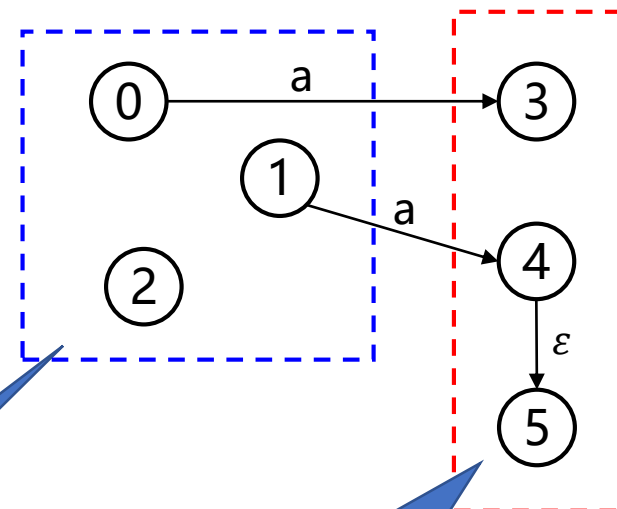
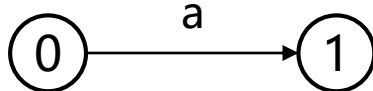
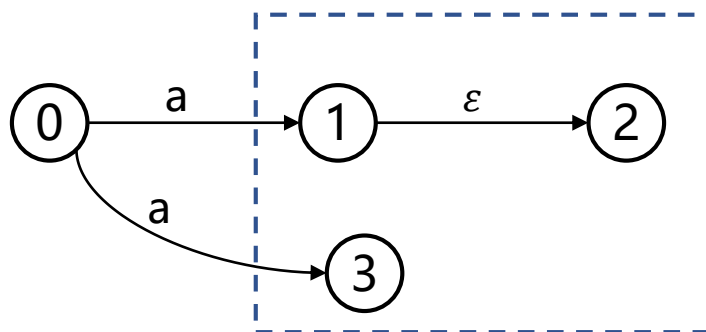


□ 定义 I 的 ϵ 闭包 $\epsilon_Closure(I)$ 为:

- 若 $q \in I$, 则 $q \in \epsilon_Closure(I)$;
- 若 $q \in I, \delta(q, \epsilon) = q'$, 则 $q' \in \epsilon_Closure(I)$ 。

NFA确定化

(3) 寻找可合并状态



I

$I_a = \varepsilon_Closure(J)$
 其中 $J = \{s' | s \in I, \delta(s, a) = s'\}$

NFA确定化

算法 3.3 NFA 确定化

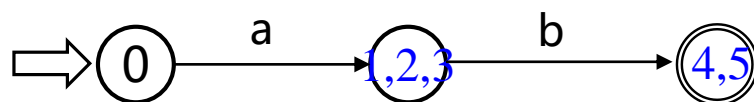
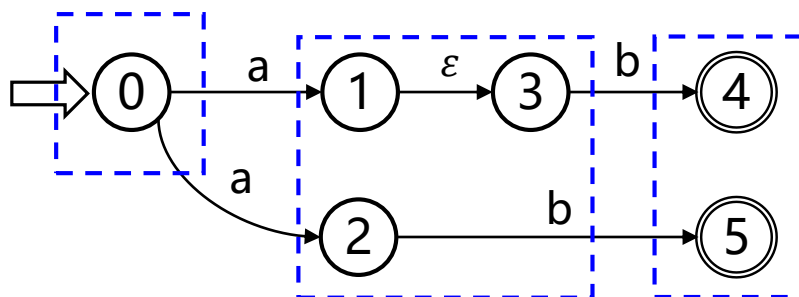
输入: NFA $M = (S, \{a_1, a_2, \dots, a_k\}, \delta, \{X\}, \{Y\})$, 其中映射 $\delta : S \times (\Sigma \cup \{\varepsilon\}) \rightarrow 2^S$

输出: 等价的 DFA M' , 其中映射 $\delta' : S \times \Sigma \rightarrow S$

```
1 DFA determineNFA( $M$ ):  
2   构造具有  $k + 1$  列的表, 记作第  $0, 1, 2, \dots, k$  列;  
3   首行首列 (第 0 列) 置为  $\varepsilon - \text{Closure}(\{X\})$ ;  
4   do  
5       如果某一行的第 0 列已确定, 记为  $I$ , 则该行第  $i$  列填入  $I_{a_i}$ ;  
6       检查该行上的所有状态子集  $I_{a_i}$ , 如果未出现在第 0 列, 则填充到后面空行的第 0 列;  
7   while 所有行都计算完毕;  
8   每个状态子集视为新的状态, 首行首列为初态, 包含原终态  $Y$  的状态子集为新终态, 得到  
   DFA  $M'$ ;  
9   return  $M'$ ;  
10 end determineNFA
```

NFA确定化

【例】



I	I_a	I_b
{0}	{1,2,3}	Φ
{1,2,3}	Φ	{4,5}
{4,5}	Φ	Φ

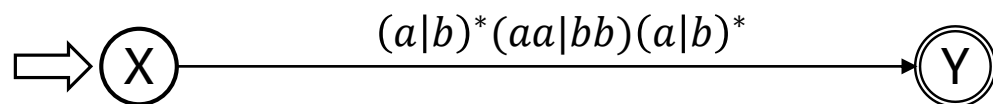
设 $\Sigma = \{a_1, a_2, \dots, a_k\}$, 初态为 X

- ① 构造具有 $k + 1$ 列的表;
- ② 首行首列设置为 $\varepsilon_Closure(X)$
- ③ 如果某一行的第0列已确定, 记为 I , 则该行第 i 列填入 I_{a_i} ;
- ④ 检查该行上的所有状态子集, 看是否出现在第0列, 如果没有出现, 则填充到后面空行的第0列;
- ⑤ 重复上述过程, 直到所有出现在第 i ($i = 1, 2, \dots, k$) 列上的状态子集都在第0列出现。

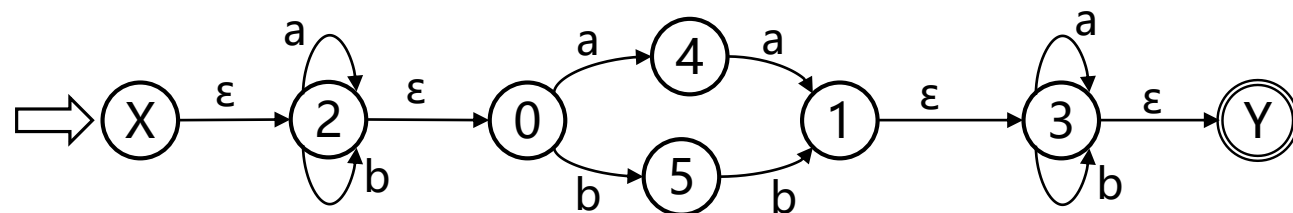
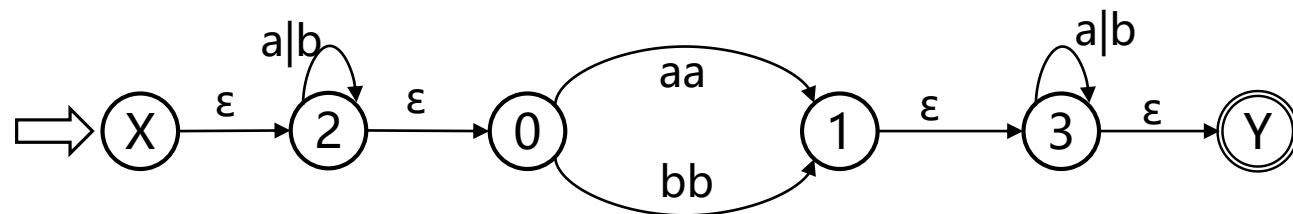
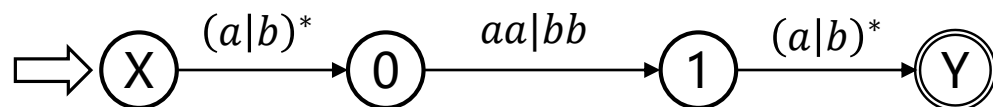
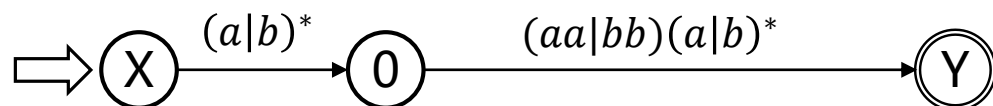
NFA确定化例题1

【例】构造正规式 $(a|b)^*(aa|bb)(a|b)^*$ 的DFA

(1) 初态终态唯一化

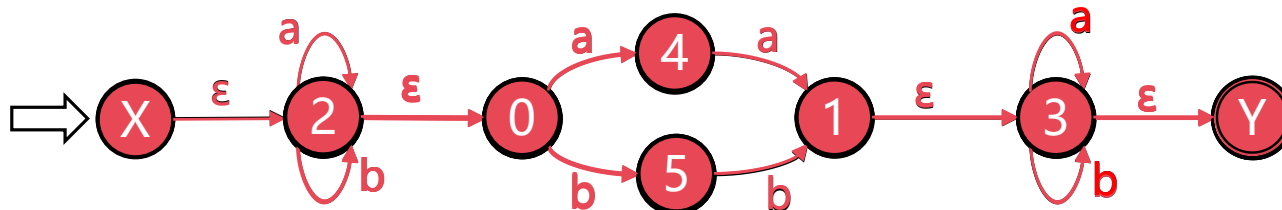


(2) 箭弧上单符化



NFA确定化例题1

(3) 寻找可合并状态



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

NFA确定化例题1

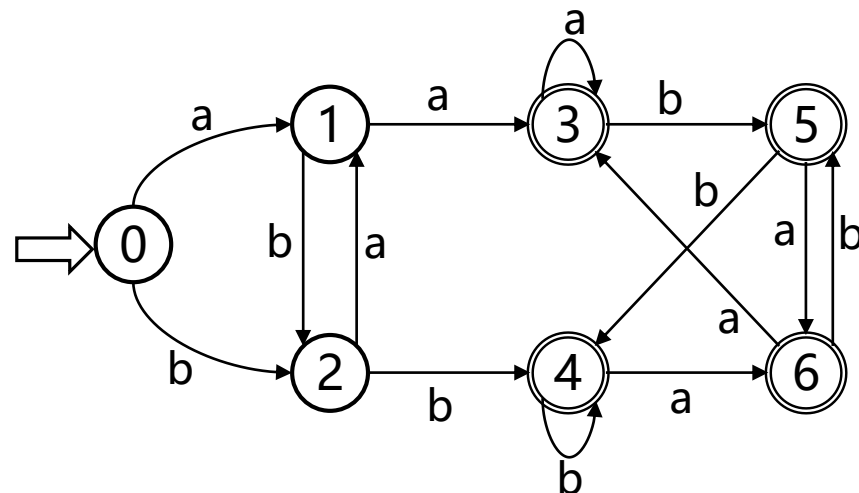
(4) 状态合并

		I	I _a	I _b
0	←	0	1	2
1	←	1	3	2
2	←	2	1	4
3	←	3	3	5
4	←	4	6	4
5	←	5	6	4
6	←	6	3	5

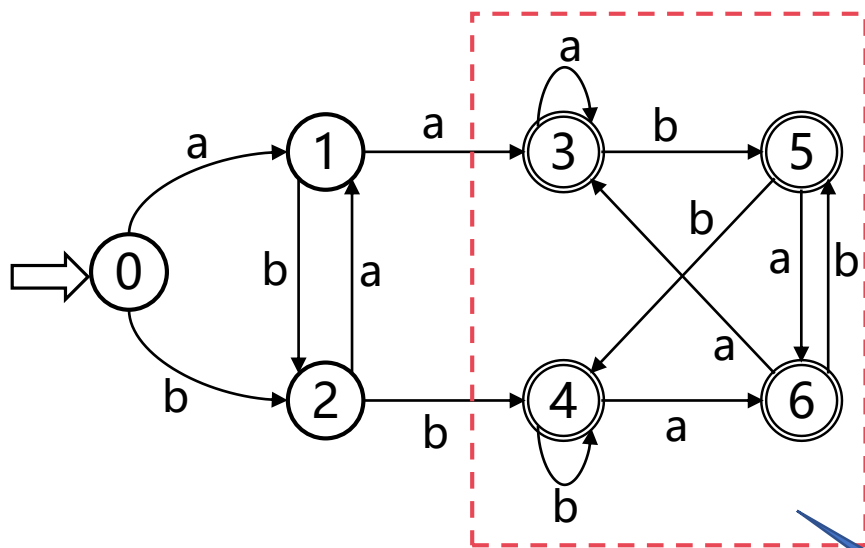
NFA确定化例题1

(4) 状态合并

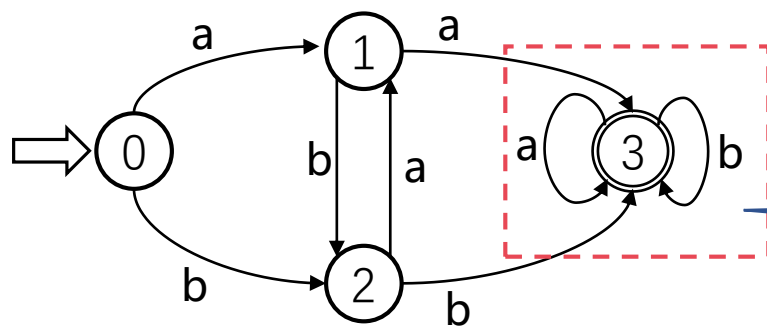
	I	I_a	I_b
$\{X, 0, 2\}$	0	1	2
$\{0, 2, 4\}$	1	3	2
$\{0, 2, 5\}$	2	1	4
$\{0, 1, 2, 3, 4, Y\}$	3	3	5
$\{0, 1, 2, 3, 5, Y\}$	4	6	4
$\{0, 2, 3, 5, Y\}$	5	6	4
$\{0, 2, 3, 4, Y\}$	6	3	5



与原例题比较



- DFA是否可化简?
- 如何化简?
- 怎样才是最简DFA?



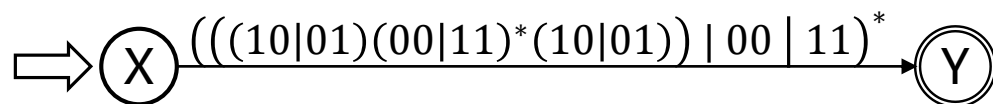
4个终态

1个终态

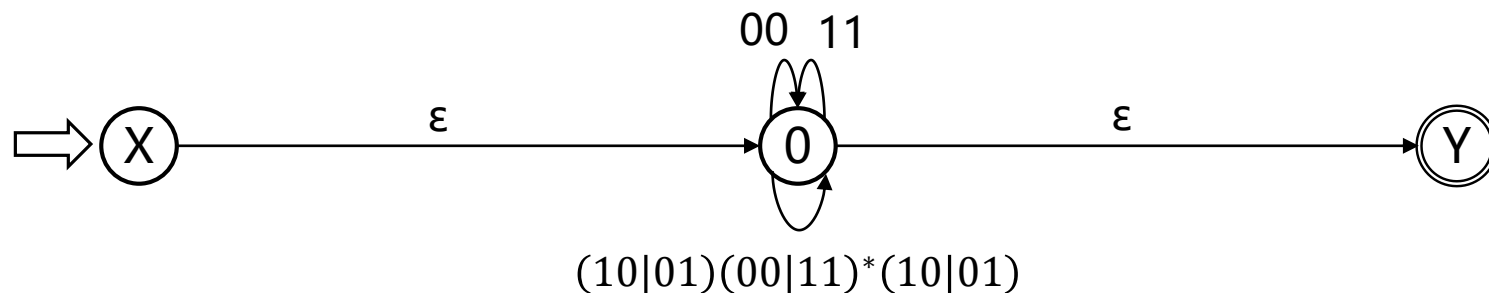
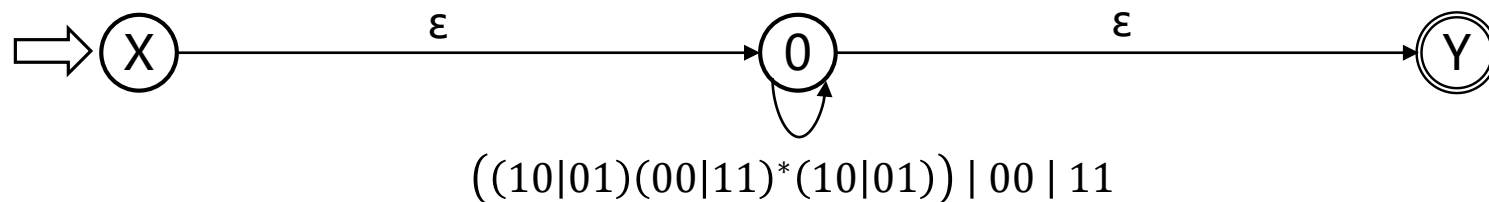
NFA确定化例题2

【例】构造DFA，使其能接受所有由偶数个0和偶数个1所组成串。

(1) 初态终态唯一化

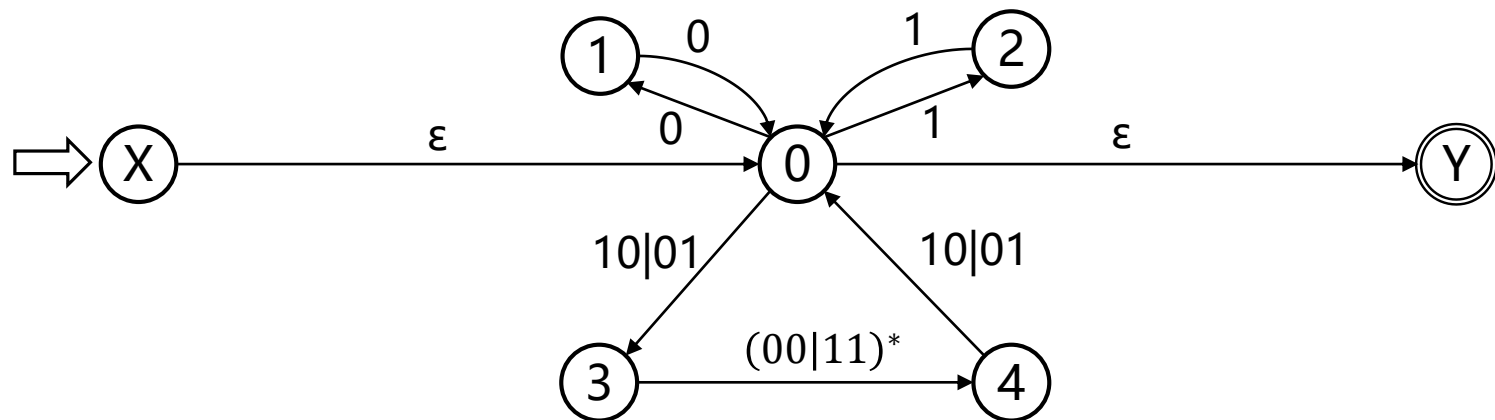
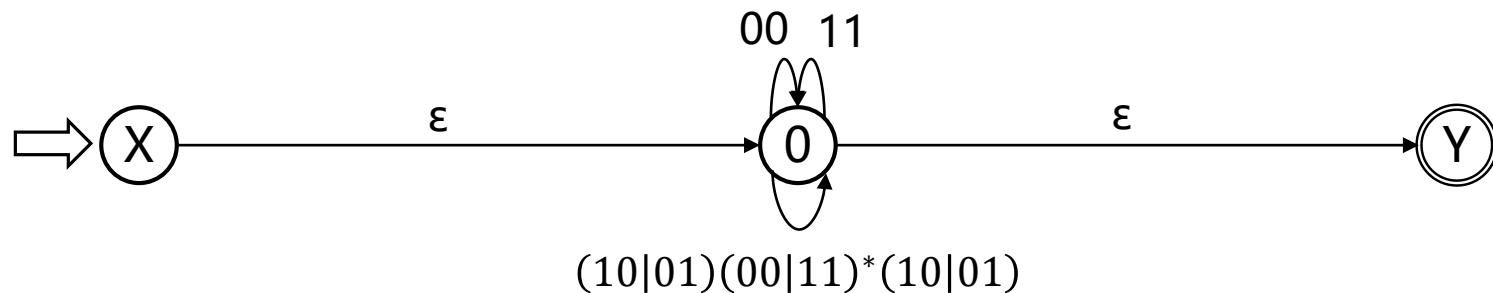


(2) 箭弧单符化



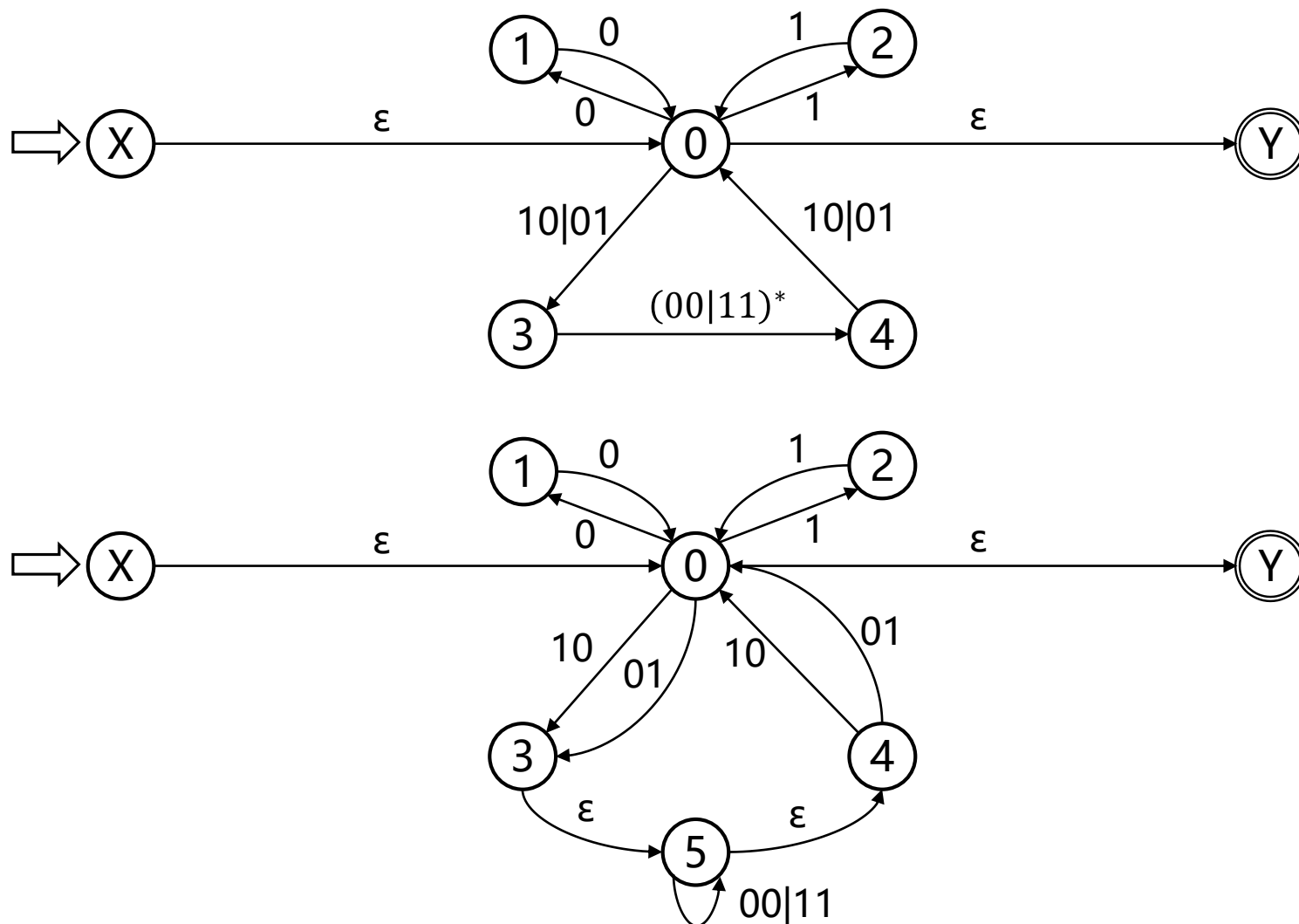
NFA确定化例题2

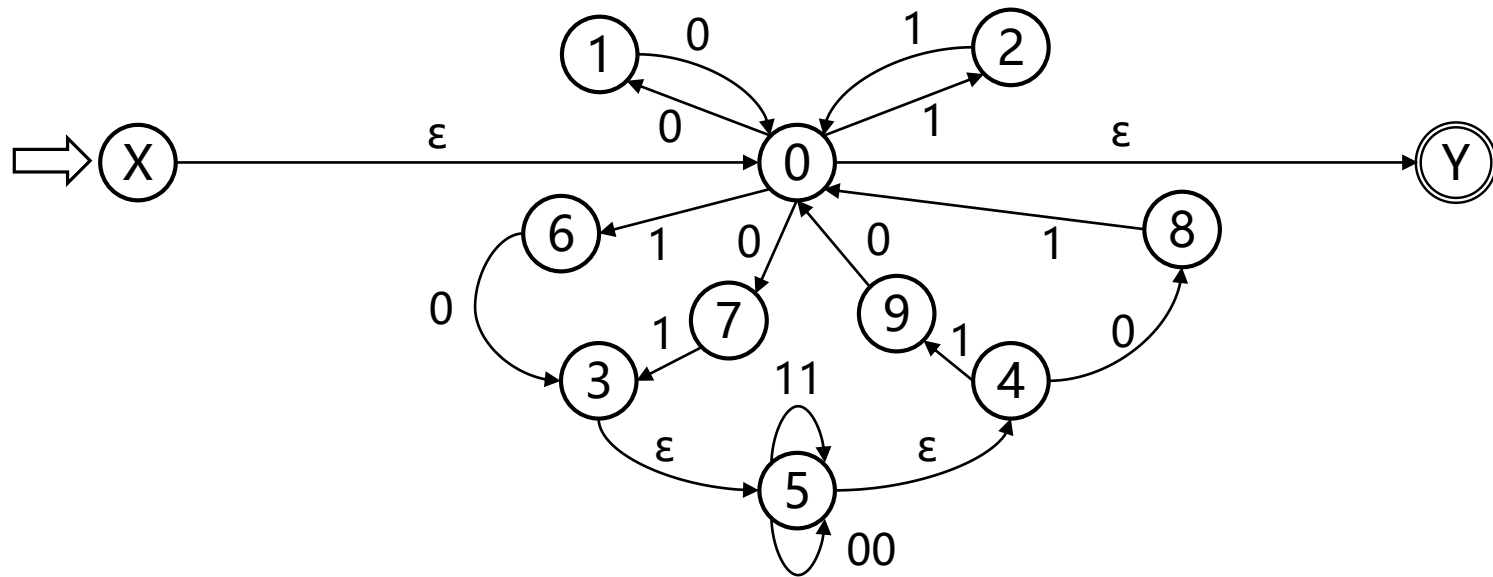
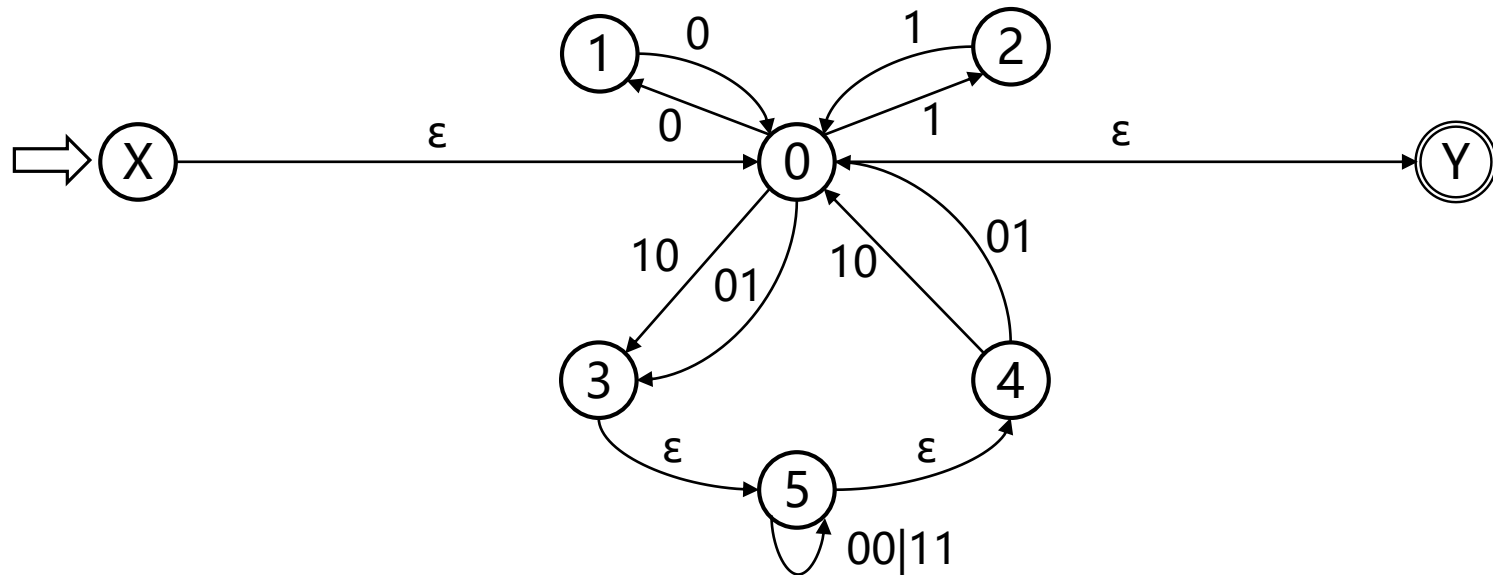
(2) 箭弧单符化

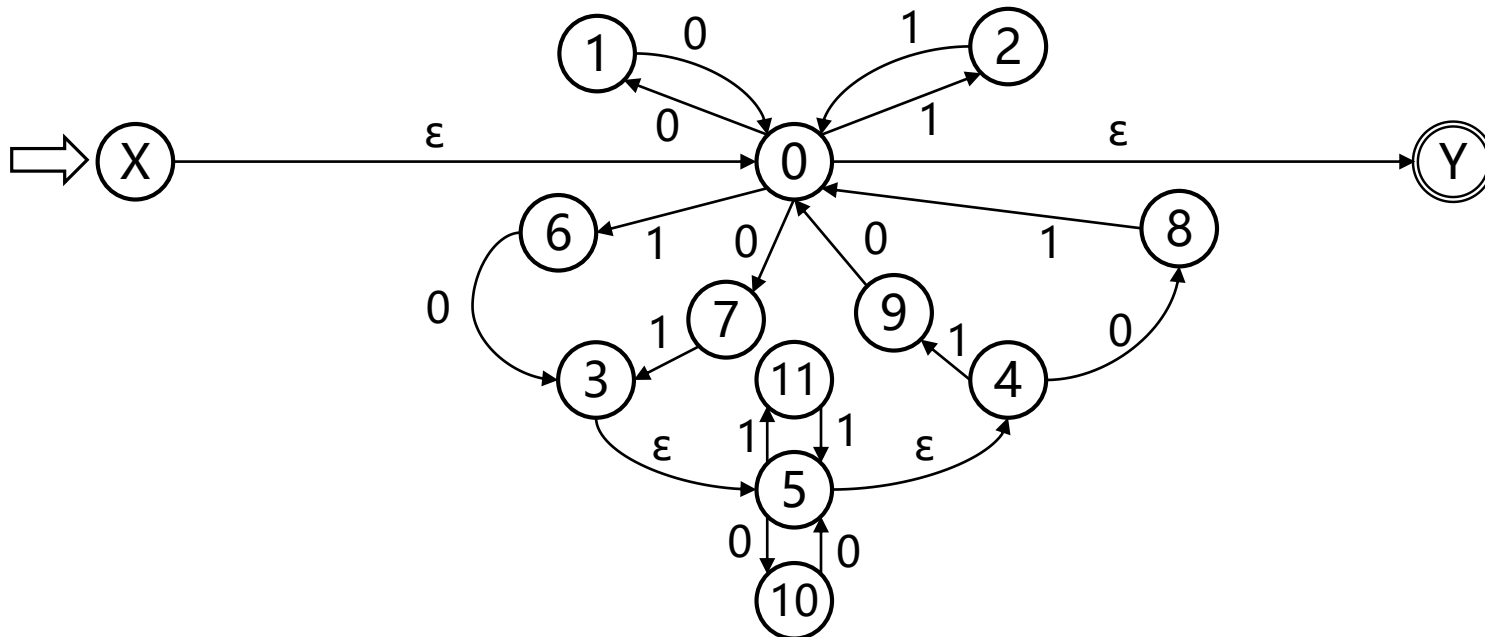
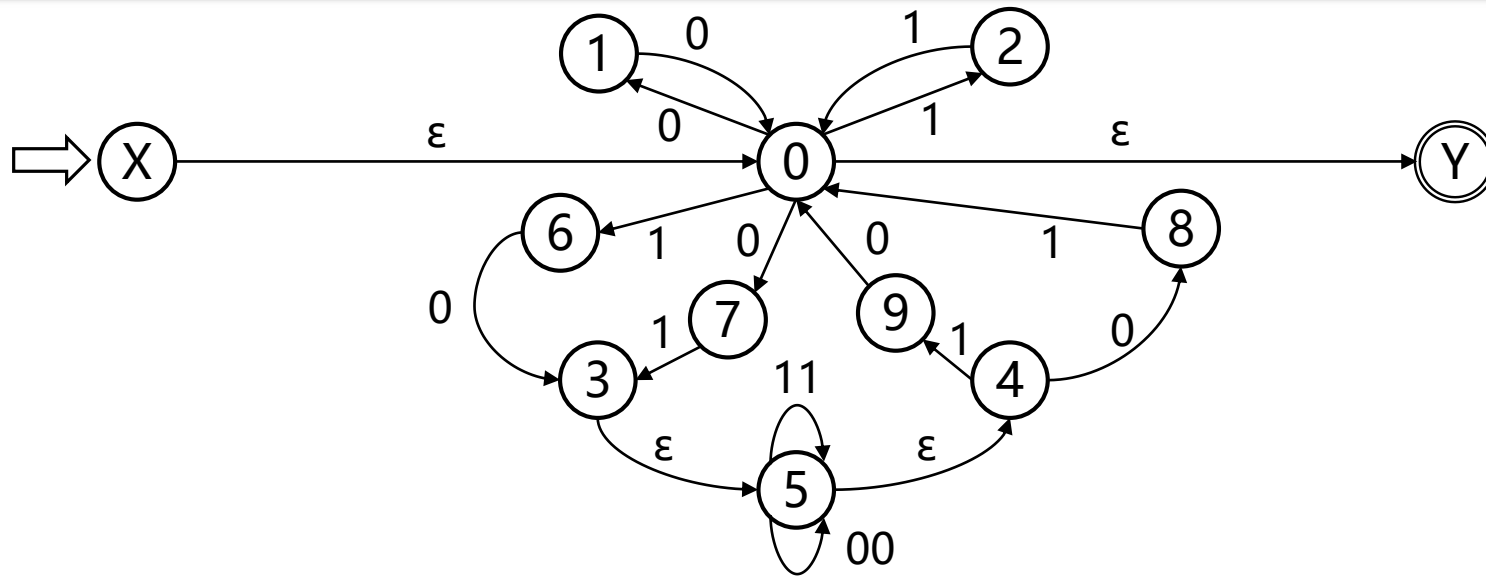


NFA确定化例题2

(2) 箭弧单符化

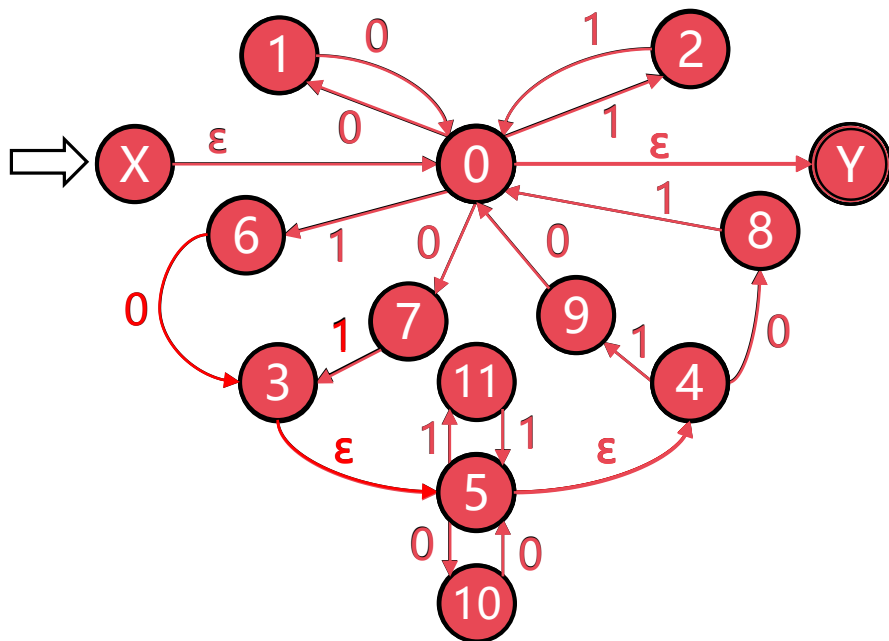






NFA确定化例题2

(3) 寻找可合并状态



I	I_0	I_1
{X,0,Y}	{1,7}	{2,6}
{1,7}	{0,Y}	{3,4,5}
{2,6}	{3,4,5}	{0,Y}
{0,Y}	{1,7}	{2,6}
{3,4,5}	{8,10}	{9,11}
{8,10}	{4,5}	{0,Y}
{9,11}	{0,Y}	{4,5}
{4,5}	{8,10}	{9,11}

NFA确定化例题2

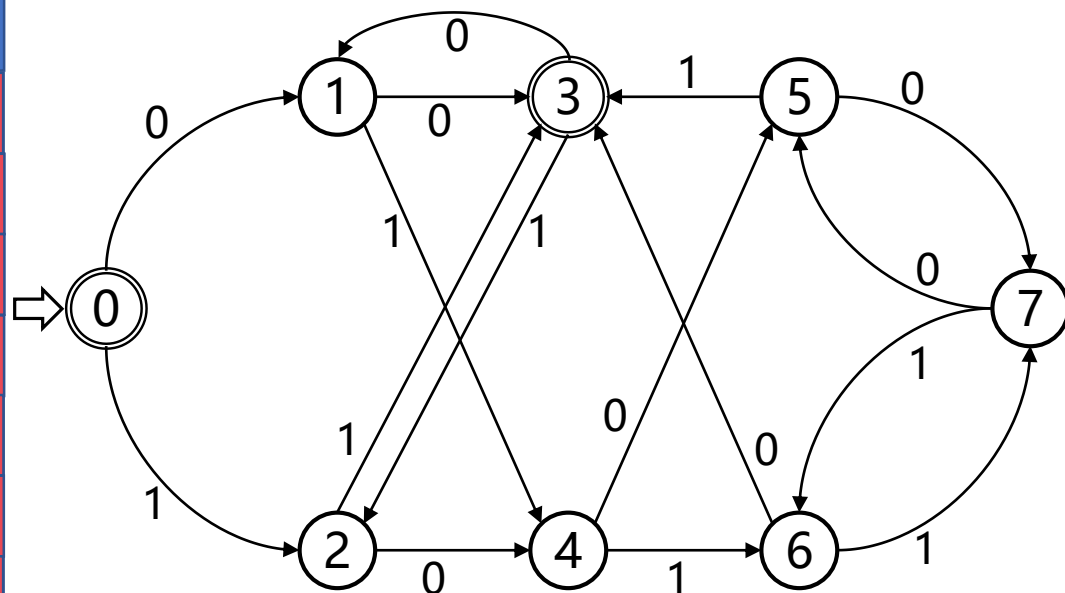
(4) 状态合并

		I	I_0	I_1
0	←	0	1	2
1	←	1	3	4
2	←	2	4	3
3	←	3	1	2
4	←	4	5	6
5	←	5	7	3
6	←	6	3	7
7	←	7	5	6

NFA确定化例题2

(4) 状态合并

	I	I_0	I_1
$\{X,0,Y\}$	0	1	2
$\{1,7\}$	1	3	4
$\{2,6\}$	2	4	3
$\{0,Y\}$	3	1	2
$\{3,4,5\}$	4	5	6
$\{8,10\}$	5	7	3
$\{9,11\}$	6	3	7
$\{4,5\}$	7	5	6



□ 3.1 对词法分析器的设计

- 3.1.1 词法分析器的任务
- 3.1.2 词法分析器设计需要考虑的问题
- 3.1.3 状态转换图

□ 3.2 有限自动机

- 3.2.1 确定有限自动机
- 3.2.2 非确定有限自动机
- 3.2.3 非确定有限自动机确定化
- 3.2.4 确定有限自动机化简
- 3.2.5 正规式与有限自动机的等价性

□ 3.3 正规文法

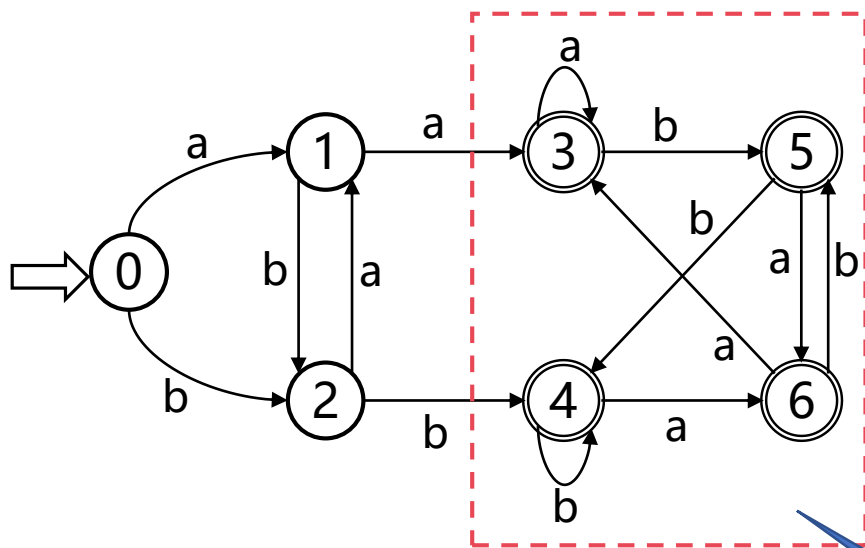
- 3.3.1 右线性文法转有限自动机
- 3.3.2 左线性文法转有限自动机

- 3.3.3 有限自动机转右线性文法
- 3.3.4 有限自动机转左线性文法
- 3.3.5 正规式转右线性文法
- 3.3.6 正规式转左线性文法
- 3.3.7 正规文法转正规式
- 3.3.8 三种工具的转换
- 3.3.9 有限自动机转正规式

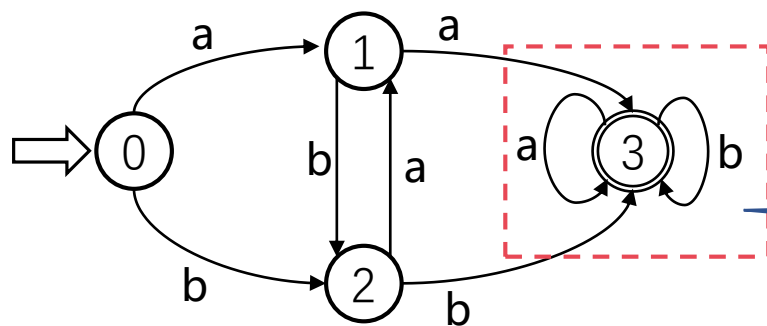
□ 3.4 词法分析器的实现

- 3.4.1 词法分析器边界
- 3.4.2 单词正规式
- 3.4.3 识别单词的DFA
- 3.4.4 单词识别算法

遗留问题



- DFA是否可化简?
- 如何化简?
- 怎样才是最简DFA?



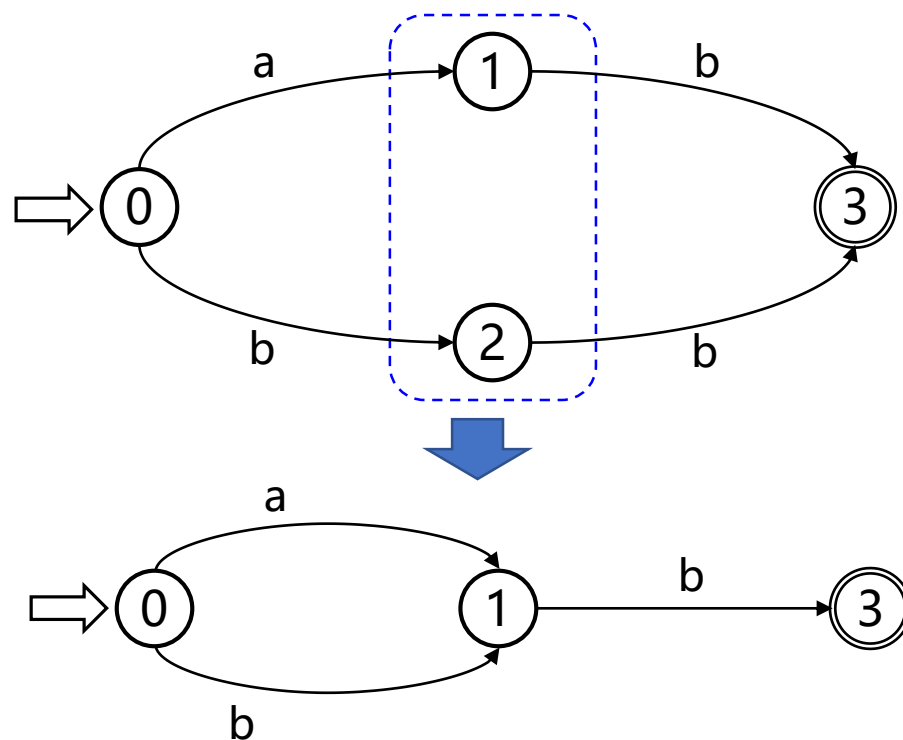
4个终态

1个终态

DFA M 化简的目标

- 寻找一个状态数比 M 少的DFA M' , 使得 $L(M) = L(M')$ 。
- 最终目标是找到状态最少的那个 M' 。

等价状态和可区别状态



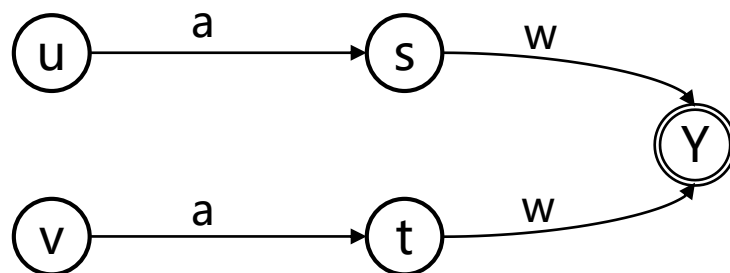
□ **等价状态**: 如果状态 s 和 t 是**等价**的, 则以下两个条件要同时满足

- 如果从 s 出发能读出某个字 w 而停在终态, 那么从 t 出发也能读出字 w 停在终态;
- 如果从 t 出发能读出某个字 w 而停在终态, 那么从 s 出发也能读出字 w 停在终态。

□ **可区别状态**: 如果状态 s 和 t **不等价**, 则称它们是**可区别的**。

等价状态的情况

□ 若 u 、 v 通过所有 a 弧可以到达的状态 s 、 t 等价, 那么 u 、 v 等价。

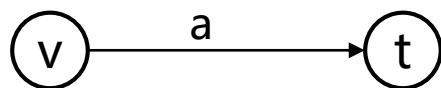
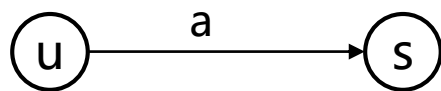
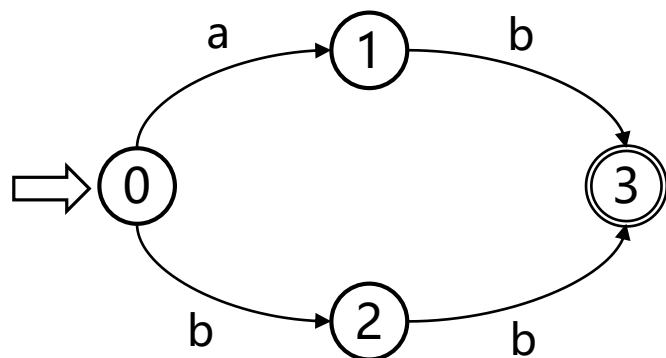


若 u 、 v 通过 a 分别到达 s 、 t :

若 u 能接受 aw , 则 s 、 t 必能接受 w , v 必可接受 aw ;

若 v 能接受 aw , 则 t 、 s 必能接受 w , u 必可接受 aw 。

可区别状态的情况

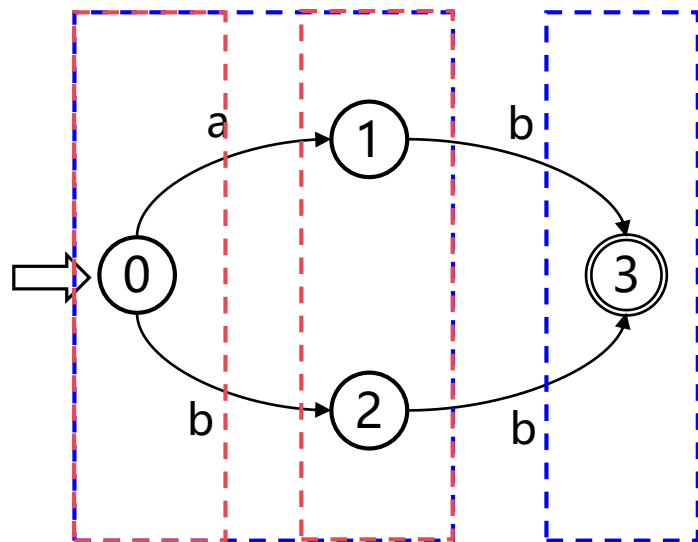


- 终态可非终态是可区别的;
- 有射出弧的状态和没有射出弧的状态可区别的;
- $\delta(u, a) = s, \delta(v, a) = t$, 若 s, t 可区别, 则 u, v 可区别;
- $\delta(u, a) = s$, 对 s 的任意等价状态 t , v 均没有 a 弧到达, 则 u, v 可区别。

□ 假设 u, v 等价

- 若 s 可接受字 w , 则 u, v 可接受字 aw , v 通过 a 弧到达的状态可接受 w ;
- 若 v 通过 a 弧到达的状态可接受 w , 则 u, v 可接受字 aw , s 可接受 w , 矛盾。

可区别状态的情况



- 终态可非终态是可区别的;
- 有射出 a 弧的状态和没有射出 a 弧的状态可区别的;
- $\delta(u, a) = s, \delta(v, a) = t$, 若 s, t 可区别, 则 u, v 可区别;
- $\delta(u, a) = s$, 对 s 的任意等价状态 t , v 均没有 a 弧到达, 则 u, v 可区别。

- 终态和非终态可区别;
- 1、2有 b 弧到达3, 0没有。

DFA $M = (S, \Sigma, \delta, s_0, F)$ 的化简

算法 3.4 DFA 化简的状态子集划分算法

输入: DFA $M = (S, \Sigma, \delta, s_0, F)$

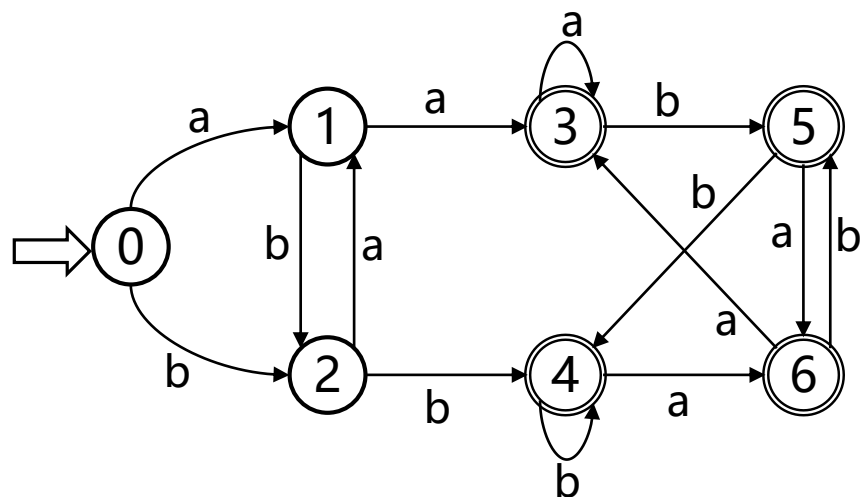
输出: DFA M' , M' 是使得 $L(M) = L(M')$ 成立的状态数最少的 DFA

```

1  $\Pi = \{S - F, F\};$ 
2 do
3   foreach  $I^{(i)} \in (\Pi = \{I^{(1)}, I^{(2)}, \dots, I^{(m)}\})$  do
4     if  $(\exists a \in \Sigma \wedge \exists s_1 \in I^{(i)} \wedge \exists s_2 \in I^{(i)})$  使得  $(\delta(s_1, a) \in I^{(j)} \wedge \delta(s_2, a) \notin I^{(j)})$  then
5        $I^{(i1)} = \{s | s \in I^{(i)} \wedge \delta(s, a) \in I^{(j)}\};$ 
6        $I^{(i2)} = I^{(i)} - I^{(i1)};$ 
7        $\Pi - = I^{(i)};$ 
8        $\Pi \cup = (I^{(i1)} \cup I^{(i2)});$ 
9     end
10  end
11 while  $\Pi$  不能再划分;
12 合并  $\Pi$  中的状态子集  $I^{(i)};$ 
13 含有原初态的状态子集为新初态, 含有原终态的状态子集为新终态;
```

DFA化简例题1

【例】化简如下DFA M



□ 初次划分: $\Pi_0 = \{\{0,1,2\}, \{3,4,5,6\}\}$

□ 考察子集 $\{0,1,2\}$

➤ $\delta(0, a) = 1 \in \{0,1,2\}$

➤ $\delta(1, a) = 3 \in \{3,4,5,6\}$

➤ $\delta(2, a) = 1 \in \{0,1,2\}$

□ $\Pi_1 = \{\{0,2\}, \{1\}, \{3,4,5,6\}\}$

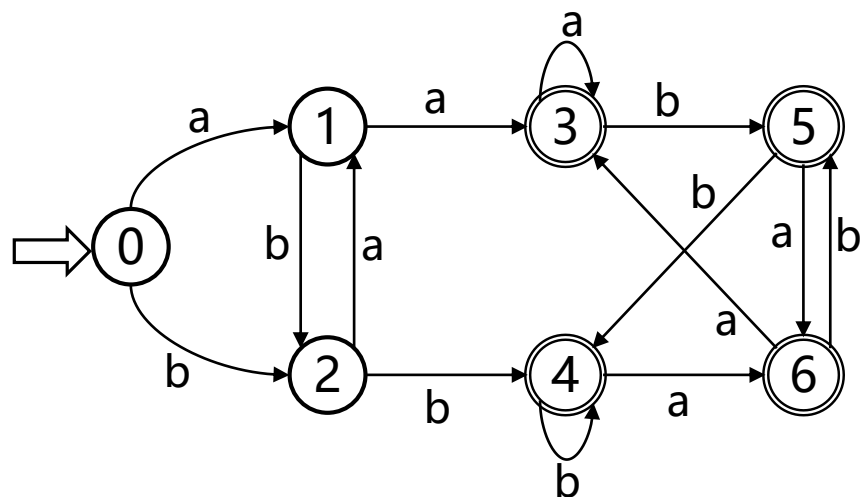
□ 考察子集 $\{0,2\}$

➤ $\delta(0, b) = 2 \in \{0,2\}$, $\delta(2, b) = 4 \in \{3,4,5,6\}$

□ $\Pi_2 = \{\{0\}, \{2\}, \{1\}, \{3,4,5,6\}\}$

DFA化简例题1

【例】化简如下DFA M



□ 考察子集 $\{3,4,5,6\}$

- $\delta(3, a) = 3 \in \{3,4,5,6\}$
- $\delta(4, a) = 6 \in \{3,4,5,6\}$
- $\delta(5, a) = 6 \in \{3,4,5,6\}$
- $\delta(6, a) = 3 \in \{3,4,5,6\}$
- $\delta(3, b) = 5 \in \{3,4,5,6\}$
- $\delta(4, b) = 4 \in \{3,4,5,6\}$
- $\delta(5, b) = 4 \in \{3,4,5,6\}$
- $\delta(6, b) = 5 \in \{3,4,5,6\}$

□ $\Pi_2 = \{\{0\}, \{2\}, \{1\}, \{3,4,5,6\}\}$

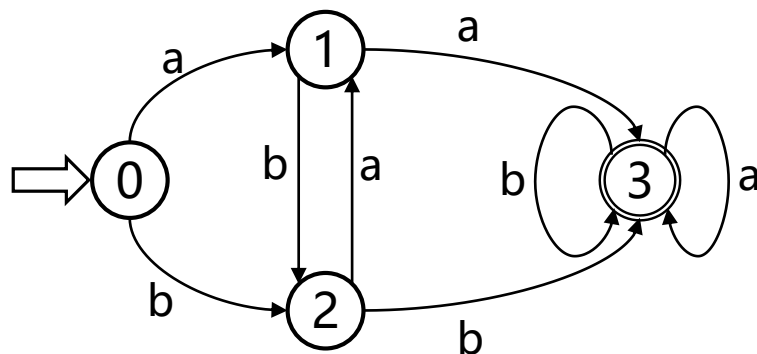
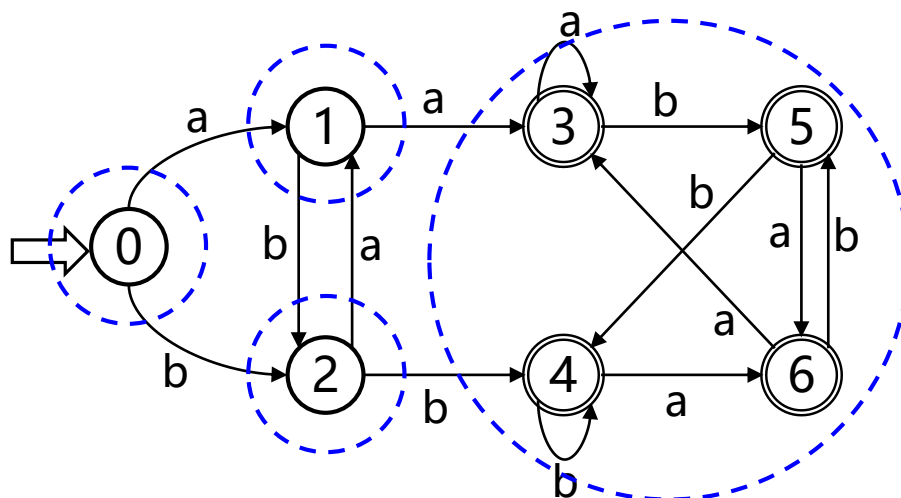
□ 初态 $\{0\}$, 终态 $\{3,4,5,6\}$

DFA化简例题1

【例】化简如下DFA M

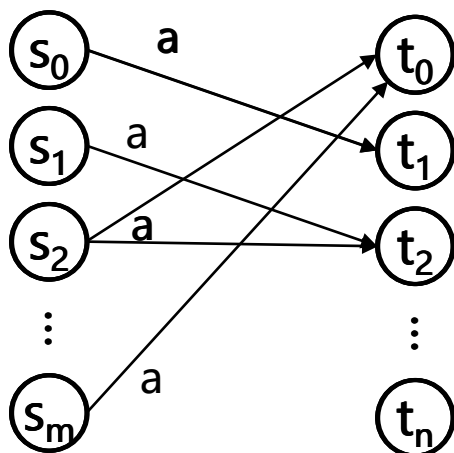
$$\square \Pi_2 = \{\{0\}, \{2\}, \{1\}, \{3,4,5,6\}\}$$

□ 初态 $\{0\}$, 终态 $\{3,4,5,6\}$



合并状态后创建弧的原则

□ 两个合并的状态集: $S = \{s_0, s_1, \dots, s_m\}$, $T = \{t_0, t_1, \dots, t_n\}$



□ 全部 s_i 到部分或全部 t_j 有 a 弧: $\delta(S, a) = T$

□ 全部 s_i 到所有 t_j 没有 a 弧: $\delta(S, a) \neq T$

□ 部分 s_i 到 t_j 有 a 弧, 部分没有: 说明 S 可划分

合并状态后创建弧的原则

□ 两个合并的状态集: $S = \{s_0, s_1, \dots, s_m\}$, $T = \{t_0, t_1, \dots, t_n\}$

□ 全部 s_i 到部分或全部 t_j 有 a 弧: $\delta(S, a) = T$

□ 全部 s_i 到所有 t_j 没有 a 弧: $\delta(S, a) \neq T$

□ 部分 s_i 到 t_j 有 a 弧, 部分没有: 说明 S 可划分

□ $S=T$ 的情况

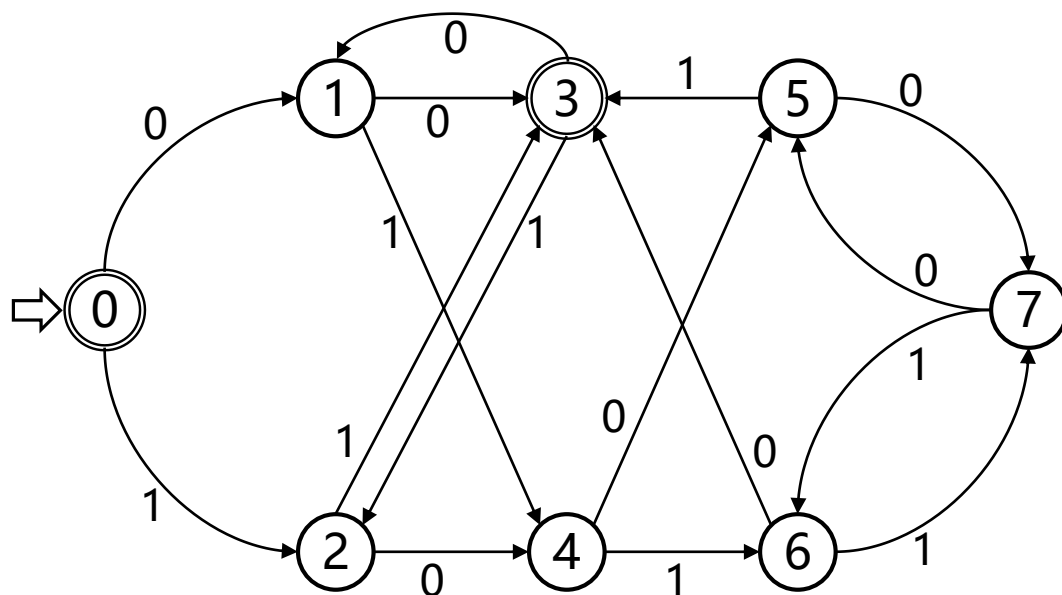
□ 全部 s_i 到部分或全部 s_j 有 a 弧: $\delta(S, a) = S$

□ 全部 s_i 到所有 s_j 没有 a 弧: $\delta(S, a) \neq S$

□ 部分 s_i 到 s_j 有 a 弧, 部分没有: 说明 S 可划分

DFA化简例题2

【例】化简如下DFA M



□ 初次划分:

□ $\Pi_0 = \{\{1,2,4,5,6,7\}, \{0,3\}\}$

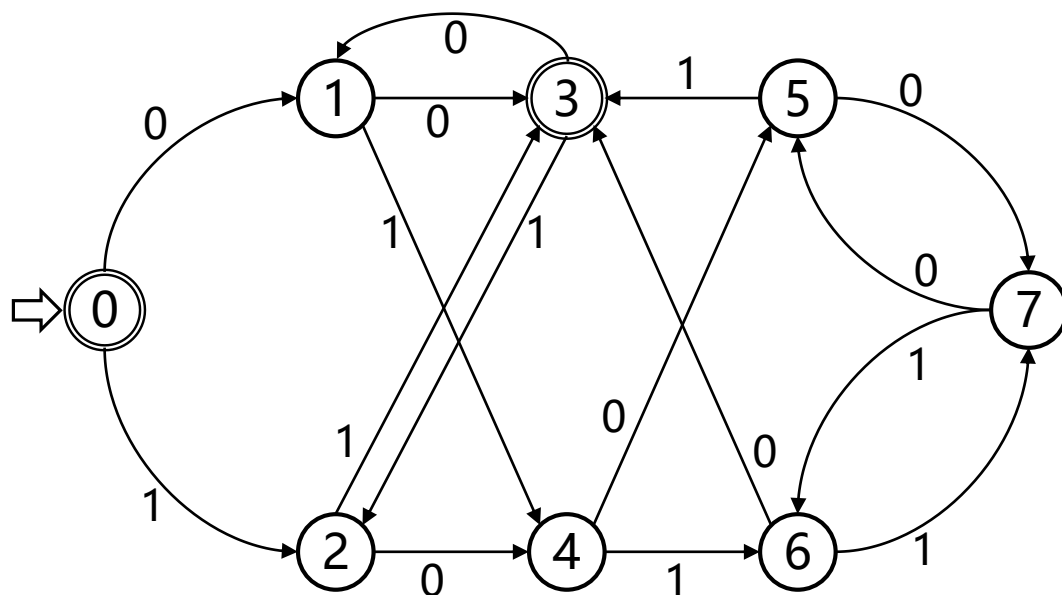
□ 考察子集 $\{1,2,4,5,6,7\}$

- $\delta(1,0) = 3 \in \{0,3\}$
- $\delta(2,0) = 4 \in \{1,2,4,5,6,7\}$
- $\delta(4,0) = 5 \in \{1,2,4,5,6,7\}$
- $\delta(5,0) = 7 \in \{1,2,4,5,6,7\}$
- $\delta(6,0) = 3 \in \{0,3\}$
- $\delta(7,0) = 5 \in \{1,2,4,5,6,7\}$

□ $\Pi_1 = \{\{2,4,5,7\}, \{1,6\}, \{0,3\}\}$

DFA化简例题2

【例】化简如下DFA M



$$\square \Pi_1 = \{\{2,4,5,7\}, \{1,6\}, \{0,3\}\}$$

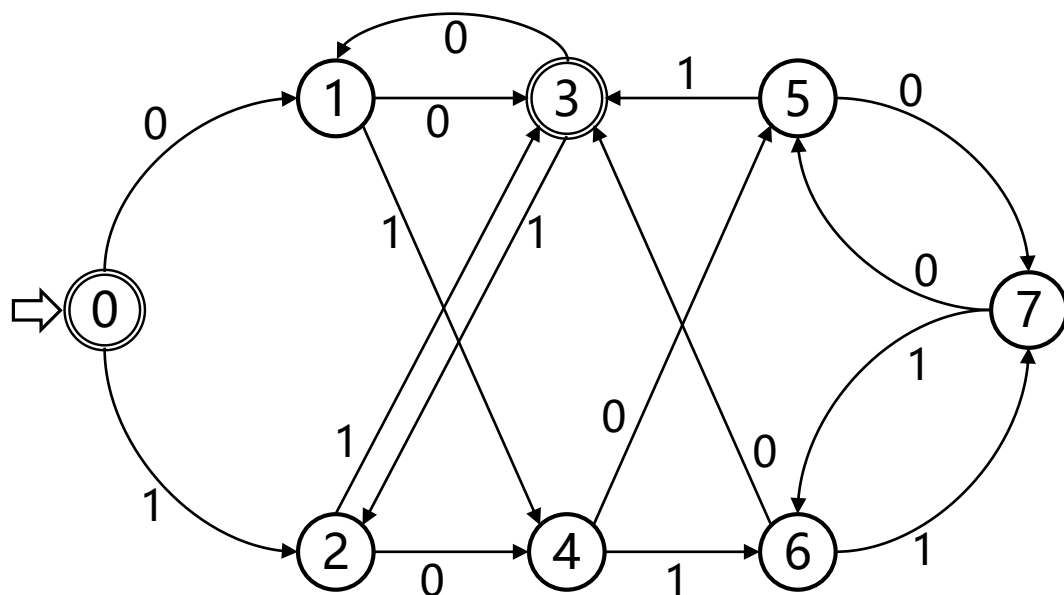
考察子集 $\{2,4,5,7\}$

- $\delta(2,0) = 4 \in \{2,4,5,7\}$
- $\delta(4,0) = 5 \in \{2,4,5,7\}$
- $\delta(5,0) = 7 \in \{2,4,5,7\}$
- $\delta(7,0) = 5 \in \{2,4,5,7\}$
- $\delta(2,1) = 3 \in \{0,3\}$
- $\delta(4,1) = 6 \in \{1,6\}$
- $\delta(5,1) = 3 \in \{0,3\}$
- $\delta(7,1) = 6 \in \{1,6\}$

$$\square \Pi_2 = \{\{2,5\}, \{4,7\}, \{1,6\}, \{0,3\}\}$$

DFA化简例题2

【例】化简如下DFA M



$$\square \Pi_2 = \{\{2,5\}, \{4,7\}, \{1,6\}, \{0,3\}\}$$

考察子集 $\{2,5\}$

$$\triangleright \delta(2,0) = 4 \in \{4,7\}$$

$$\triangleright \delta(5,0) = 7 \in \{4,7\}$$

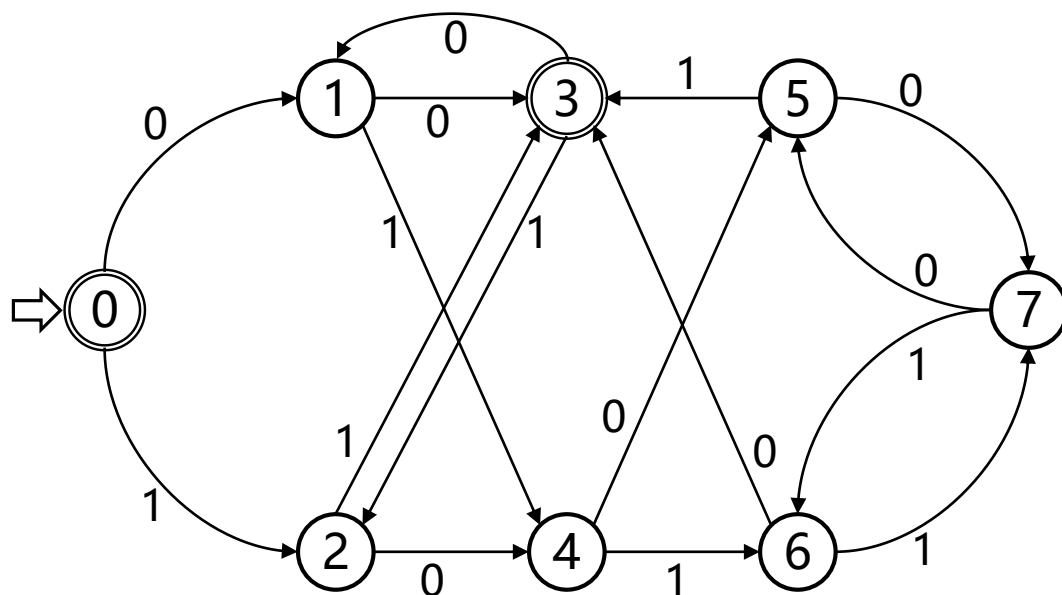
$$\triangleright \delta(2,1) = 3 \in \{0,3\}$$

$$\triangleright \delta(5,1) = 3 \in \{0,3\}$$

$$\square \Pi_2 = \{\{2,5\}, \{4,7\}, \{1,6\}, \{0,3\}\}$$

DFA化简例题2

【例】化简如下DFA M



$$\Pi_2 = \{\{2,5\}, \{4,7\}, \{1,6\}, \{0,3\}\}$$

考察子集 $\{4,7\}$

$$\triangleright \delta(4,0) = 5 \in \{2,5\}$$

$$\triangleright \delta(7,0) = 5 \in \{2,5\}$$

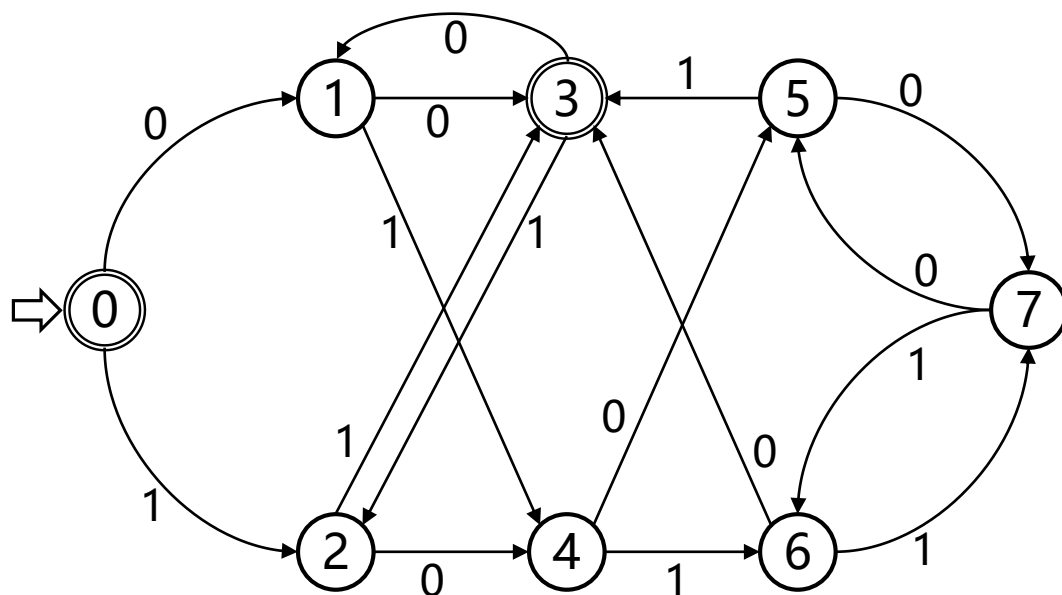
$$\triangleright \delta(4,1) = 6 \in \{1,6\}$$

$$\triangleright \delta(7,1) = 6 \in \{1,6\}$$

$$\Pi_2 = \{\{2,5\}, \{4,7\}, \{1,6\}, \{0,3\}\}$$

DFA化简例题2

【例】化简如下DFA M



$$\Pi_2 = \{\{2,5\}, \{4,7\}, \{1,6\}, \{0,3\}\}$$

考察子集 $\{1,6\}$

$$\triangleright \delta(1,0) = 3 \in \{0,3\}$$

$$\triangleright \delta(6,0) = 3 \in \{0,3\}$$

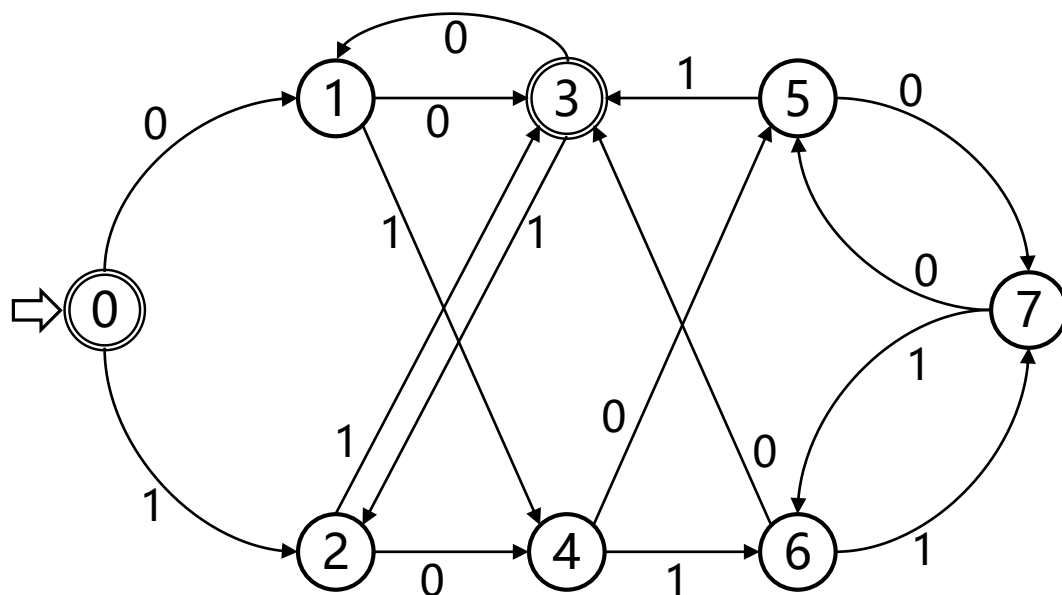
$$\triangleright \delta(1,1) = 4 \in \{4,7\}$$

$$\triangleright \delta(6,1) = 7 \in \{4,7\}$$

$$\Pi_2 = \{\{2,5\}, \{4,7\}, \{1,6\}, \{0,3\}\}$$

DFA化简例题2

【例】化简如下DFA M



$$\Pi_2 = \{\{2,5\}, \{4,7\}, \{1,6\}, \{0,3\}\}$$

考察子集 $\{0,3\}$

$$\triangleright \delta(0,0) = 1 \in \{1,6\}$$

$$\triangleright \delta(3,0) = 1 \in \{1,6\}$$

$$\triangleright \delta(0,1) = 2 \in \{2,5\}$$

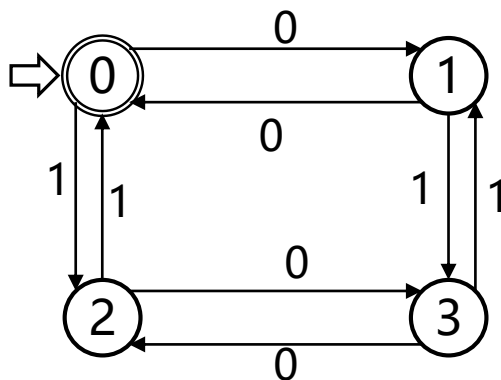
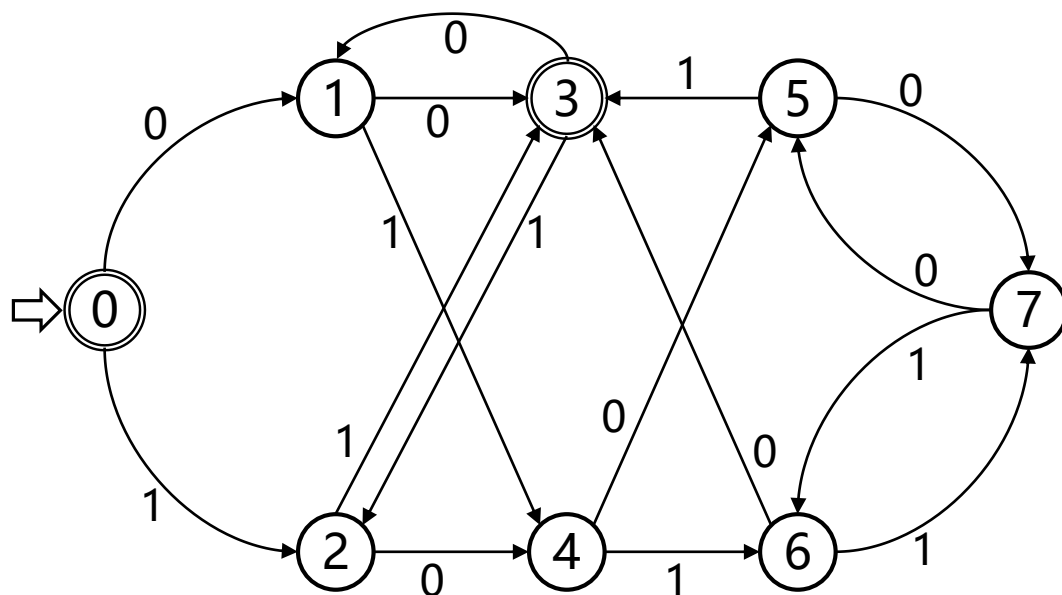
$$\triangleright \delta(3,1) = 2 \in \{2,5\}$$

$$\Pi_2 = \{\{2,5\}, \{4,7\}, \{1,6\}, \{0,3\}\}$$

初态 $\{0,3\}$, 终态 $\{0,3\}$

DFA化简例题2

【例】化简如下DFA M



$$\Pi_2 = \{\{2,5\}, \{4,7\}, \{1,6\}, \{0,3\}\}$$

初态 $\{0,3\}$, 终态 $\{0,3\}$

$I^{(i)}$	I	I_0	I_1
$\{0,3\}$	0	$\{1,6\}$	$\{2,5\}$
$\{1,6\}$	1	$\{0,3\}$	$\{4,7\}$
$\{2,5\}$	2	$\{4,7\}$	$\{0,3\}$
$\{4,7\}$	3	$\{2,5\}$	$\{1,6\}$

$I^{(i)}$	I	I_0	I_1
$\{0,3\}$	0	1	2
$\{1,6\}$	1	0	3
$\{2,5\}$	2	3	0
$\{4,7\}$	3	2	1

第三章作业

【作业3-1】正规式: $1(0|1)^*101$

- (1) 构造NFA, 要求每条弧上或为单个字符, 或为 ε 。
- (2) 确定化。
- (3) 最小化

【作业3-2】正规式: $(01|10)^*$

- (1) 构造NFA, 要求每条弧上或为单个字符, 或为 ε 。
- (2) 确定化。
- (3) 最小化

□ 3.1 对词法分析器的设计

- 3.1.1 词法分析器的任务
- 3.1.2 词法分析器设计需要考虑的问题
- 3.1.3 状态转换图

□ 3.2 有限自动机

- 3.2.1 确定有限自动机
- 3.2.2 非确定有限自动机
- 3.2.3 非确定有限自动机确定化
- 3.2.4 确定有限自动机化简
- 3.2.5 正规式与有限自动机的等价性

□ 3.3 正规文法

- 3.3.1 右线性文法转有限自动机
- 3.3.2 左线性文法转有限自动机

- 3.3.3 有限自动机转右线性文法
- 3.3.4 有限自动机转左线性文法
- 3.3.5 正规式转右线性文法
- 3.3.6 正规式转左线性文法
- 3.3.7 正规文法转正规式
- 3.3.8 三种工具的转换
- 3.3.9 有限自动机转正规式

□ 3.4 词法分析器的实现

- 3.4.1 词法分析器边界
- 3.4.2 单词正规式
- 3.4.3 识别单词的DFA
- 3.4.4 单词识别算法

正规式与有限自动机的等价性

□ 正规式与有限自动机的等价性

- 对任何FA M , 都存在一个正规式 r , 使得 $L(r) = L(M)$ 。
- 对任何正规式 r , 都存在一个FA M , 使得 $L(M) = L(r)$ 。

FA $M \Rightarrow$ 正规式 r

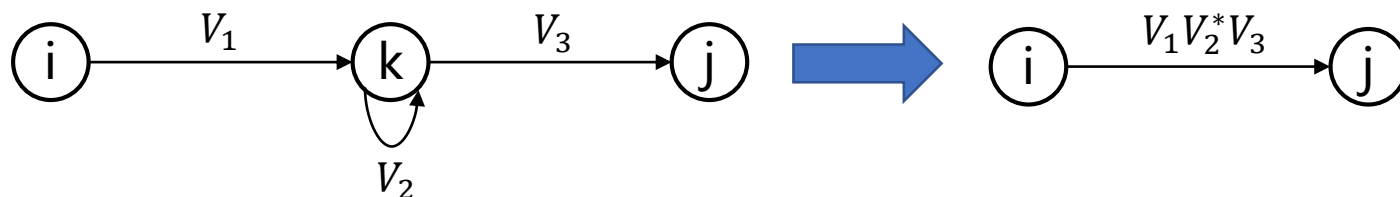
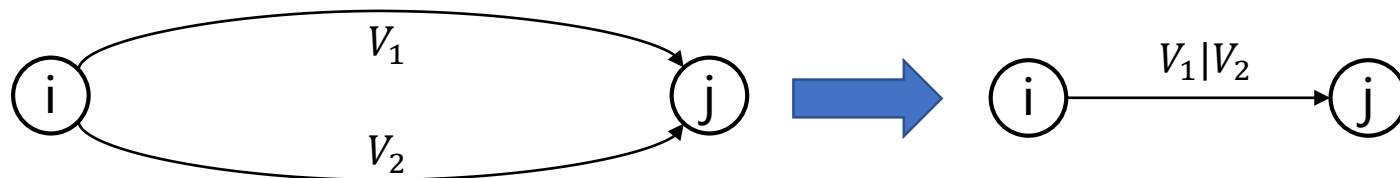
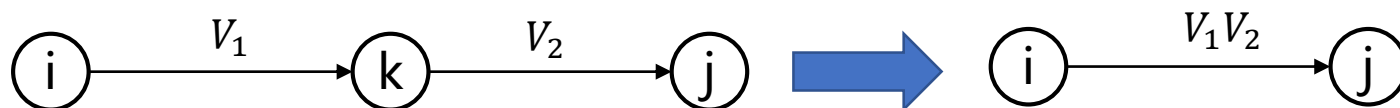
(1) 初态终态唯一化

- 在 M 的状态转换图上增加两个结点 X 和 Y ;
- 从 X 用 ε 弧连接到 M 的所有初态结点;
- 从 M 的所有终态结点用 ε 弧连接到 Y ;
- 形成的新的NFA记为 M' , 它只有一个初态 X 和一个终态 Y , 显然 $L(M') = L(M)$ 。

FA $M \Rightarrow$ 正规式 r

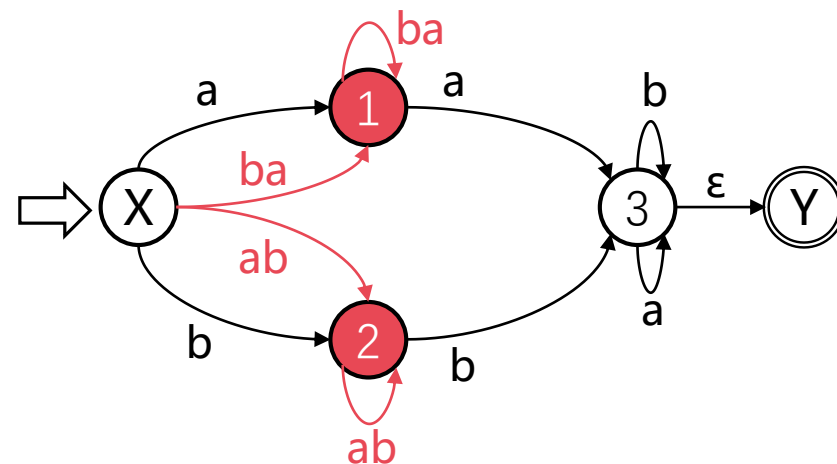
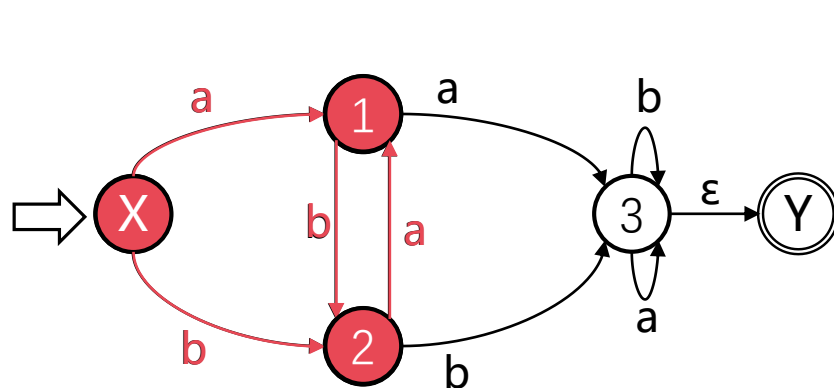
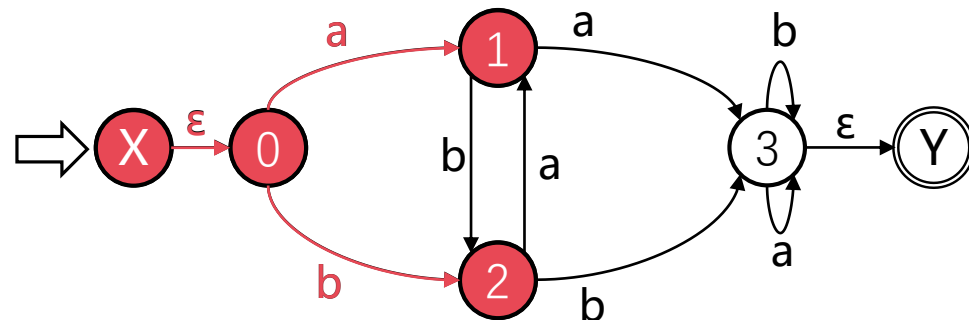
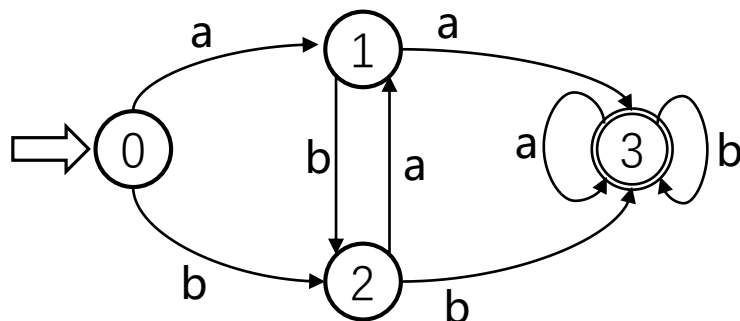
(2) 合并状态, 使 M' 只剩初态 x 、终态 y 这两个状态

➤ 反复利用以下规则替换。



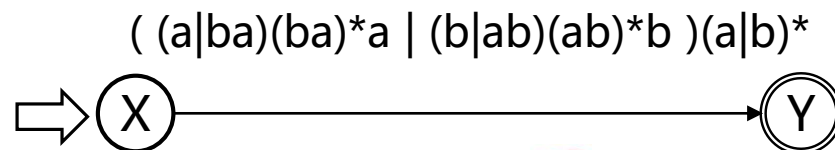
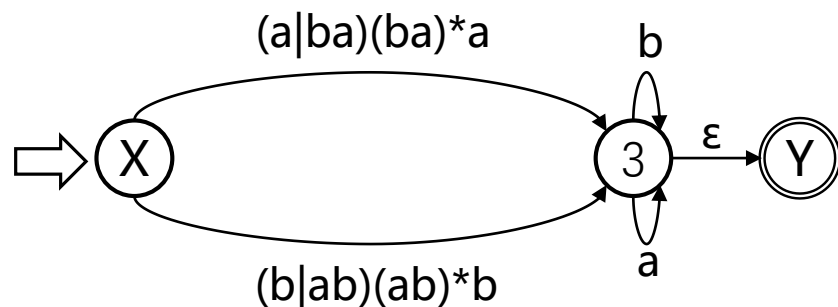
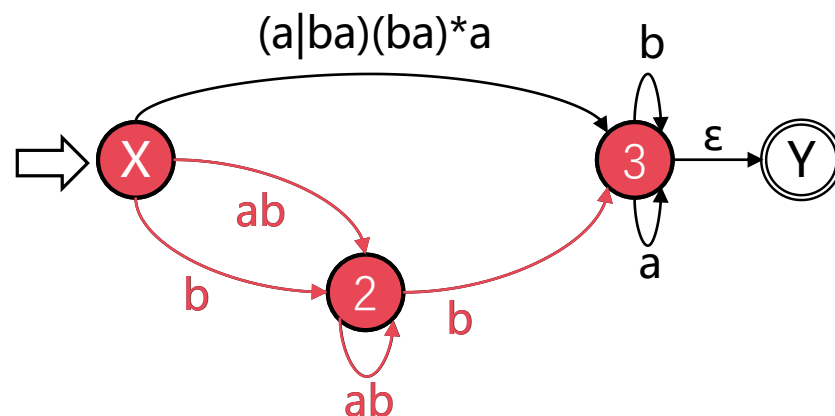
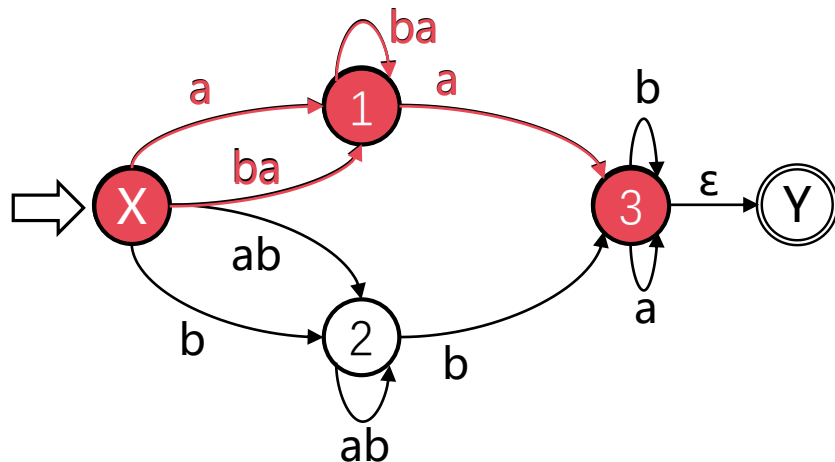
FA直接转换为正规式是非常困难的

【例】写出如下FA M 的正规式



FA直接转换为正规式是非常困难的

【例】写出如下FA M 的正规式



FA直接转换为正规式是非常困难的

【例】写出如下FA M 的正规式

$$((a|ba)(ba)^*a \mid (b|ab)(ab)^*b) (a|b)^*$$

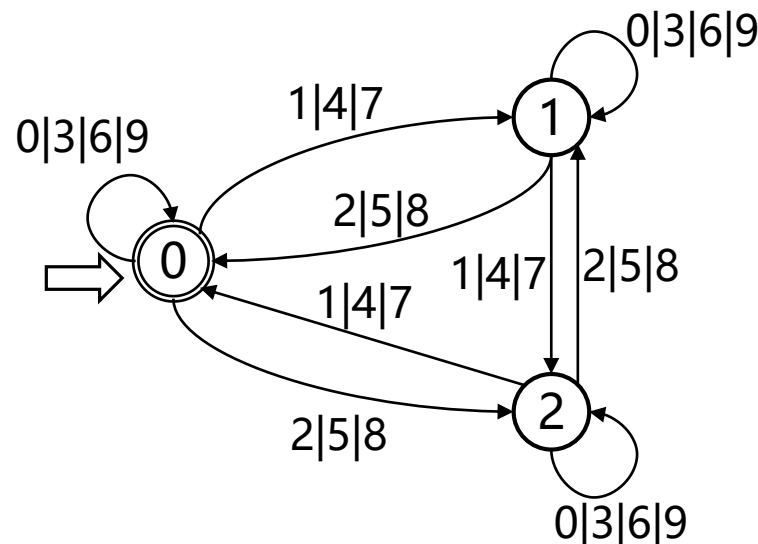
$$\Leftrightarrow (((b \mid \varepsilon) (ab)^*aa) \mid ((a \mid \varepsilon) (ba)^*bb)) (a|b)^*$$

□ 最终得到正规式:

$$(((b \mid \varepsilon) (ab)^*aa) \mid ((a \mid \varepsilon) (ba)^*bb)) (a|b)^*$$

?? 看上去与 $(a|b)^*(aa|bb)(a|b)^*$ 并不等价

?? 再复杂一点怎么转, 如被3整除的整数?

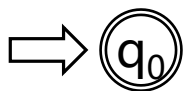


正规式 $r \Rightarrow$ FA M

【思路】使用关于 r 中运算符数目（或、连接、闭包）的数学归纳法证明。

(1) 若 r 具有0个运算符，则 $r = \varepsilon$ 或 $r = \phi$ 或 $r = a$ ，其中 $a \in \Sigma$

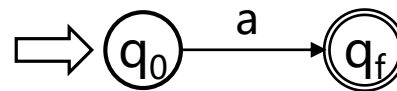
➤ 下图3个FA显然符合要求



$r = \varepsilon$



$r = \phi$



$r = a$

正规式 $r \Rightarrow \text{FA } M$

(2) 假设结论对少于 $k (k \geq 1)$ 个运算符的正规式成立, 当 r 中含有 k 个运算符时, 有3种情形 (或、连接、闭包)

【情形1】 $r = r_1 | r_2$, 其中 r_1, r_2 中运算符个数少于 k

- 由归纳假设, 对 r_i , $\exists M_i = (S_i, \Sigma_i, \delta_i, \{q_i\}, \{f_i\}) \Rightarrow L(M_i) = L(r_i)$, 并且 M_i 没有从终态发出的箭弧 ($i = 1, 2$)
- 不妨设 $S_1 \cap S_2 = \phi$, 在 $S_1 \cup S_2$ 中加入两个新状态 q_0, f_0
- 令 $M = (S_1 \cup S_2 \cup \{q_0, f_0\}, \Sigma_1 \cup \Sigma_2, \delta, \{q_0\}, \{f_0\})$, 其中 δ 定义如下:

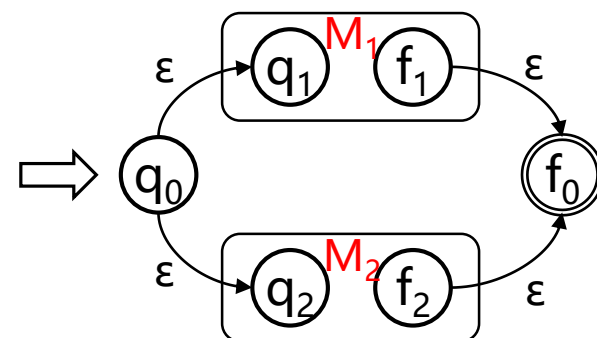
① $\delta(q_0, \varepsilon) = \{q_1, q_2\}$

② $\delta(q, a) = \delta_1(q, a)$, 当 $q \in S_1 - \{f_1\}, a \in \Sigma_1 \cup \{\varepsilon\}$

③ $\delta(q, a) = \delta_2(q, a)$, 当 $q \in S_2 - \{f_2\}, a \in \Sigma_2 \cup \{\varepsilon\}$

④ $\delta(f_1, \varepsilon) = \delta(f_2, \varepsilon) = \{f_0\}$

- 显然: $L(M) = L(M_1) \cup L(M_2) = L(r_1) \cup L(r_2) = L(r)$



正规式 $r \Rightarrow$ FA M 【情形2】 $r = r_1 r_2$

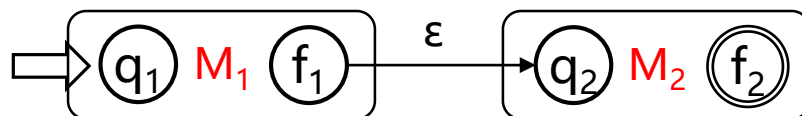
➤ 令 $M = (S_1 \cup S_2, \Sigma_1 \cup \Sigma_2, \delta, \{q_1\}, \{f_2\})$, 其中 δ 定义如下:

① $\delta(q, a) = \delta_1(q, a)$, 当 $q \in S_1 - \{f_1\}, a \in \Sigma_1 \cup \{\varepsilon\}$

② $\delta(q, a) = \delta_2(q, a)$, 当 $q \in S_2 - \{f_2\}, a \in \Sigma_2 \cup \{\varepsilon\}$

③ $\delta(f_1, \varepsilon) = \{q_2\}$

➤ 显然: $L(M) = L(M_1)L(M_2) = L(r_1)L(r_2) = L(r)$



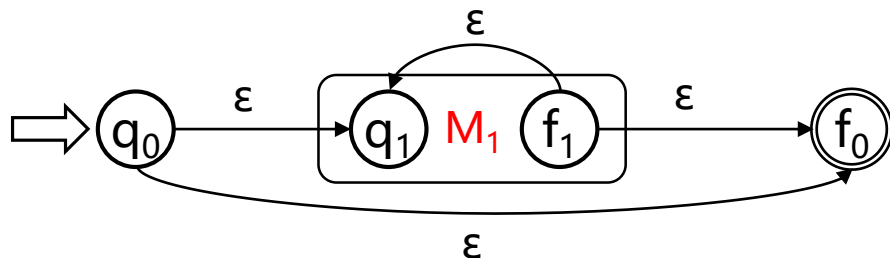
正规式 $r \Rightarrow$ FA M 【情形3】 $r = r_1^*$

➤ 令 $M = (S_1 \cup S_2 \cup \{q_0, f_0\}, \Sigma_1, \delta, \{q_0\}, \{f_0\})$, 其中 δ 定义如下:

① $\delta(q_0, \varepsilon) = \delta(f_1, \varepsilon) = \{q_1, f_0\}$

② $\delta(q, a) = \delta_1(q, a)$, 当 $q \in S_1 - \{f_1\}, a \in \Sigma_1 \cup \{\varepsilon\}$

➤ 显然: $L(M) = L(M_1)^* = L(r_1)^* = L(r)$



□ 3.1 对词法分析器的设计

- 3.1.1 词法分析器的任务
- 3.1.2 词法分析器设计需要考虑的问题
- 3.1.3 状态转换图

□ 3.2 有限自动机

- 3.2.1 确定有限自动机
- 3.2.2 非确定有限自动机
- 3.2.3 非确定有限自动机确定化
- 3.2.4 确定有限自动机化简
- 3.2.5 正规式与有限自动机的等价性

□ 3.3 正规文法

- 3.3.1 右线性文法转有限自动机
- 3.3.2 左线性文法转有限自动机

- 3.3.3 有限自动机转右线性文法
- 3.3.4 有限自动机转左线性文法
- 3.3.5 正规式转右线性文法
- 3.3.6 正规式转左线性文法
- 3.3.7 正规文法转正规式
- 3.3.8 三种工具的转换
- 3.3.9 有限自动机转正规式

□ 3.4 词法分析器的实现

- 3.4.1 词法分析器边界
- 3.4.2 单词正规式
- 3.4.3 识别单词的DFA
- 3.4.4 单词识别算法

正规文法

□ 正规文法:

- **右线性正规文法**: 产生式形式为 $A \rightarrow \alpha B$ 或 $A \rightarrow \beta$, 其中 $\alpha \in V^T, \beta \in V^T \cup \{\varepsilon\}, A \in V_N, B \in V_N$ 。
- **左线性正规文法**: 产生式形式为 $A \rightarrow B\alpha$ 或 $A \rightarrow \beta$, 其中 $\alpha \in V^T, \beta \in V^T \cup \{\varepsilon\}, A \in V_N, B \in V_N$ 。

□ 正规文法与有限自动机的等价性

- 对每一个右线性正规文法 G 或左线性正规文法 G , 都存着一个FA M , 使得 $L(M) = L(G)$ 。
- 对每一个FA M , 都存在一个右线性正规文法 G_R 及一个左线性正规文法 G_L , 使得 $L(M) = L(G_R) = L(G_L)$ 。

□ 3.1 对词法分析器的设计

- 3.1.1 词法分析器的任务
- 3.1.2 词法分析器设计需要考虑的问题
- 3.1.3 状态转换图

□ 3.2 有限自动机

- 3.2.1 确定有限自动机
- 3.2.2 非确定有限自动机
- 3.2.3 非确定有限自动机确定化
- 3.2.4 确定有限自动机化简
- 3.2.5 正规式与有限自动机的等价性

□ 3.3 正规文法

- 3.3.1 右线性文法转有限自动机
- 3.3.2 左线性文法转有限自动机

- 3.3.3 有限自动机转右线性文法
- 3.3.4 有限自动机转左线性文法
- 3.3.5 正规式转右线性文法
- 3.3.6 正规式转左线性文法
- 3.3.7 正规文法转正规式
- 3.3.8 三种工具的转换
- 3.3.9 有限自动机转正规式

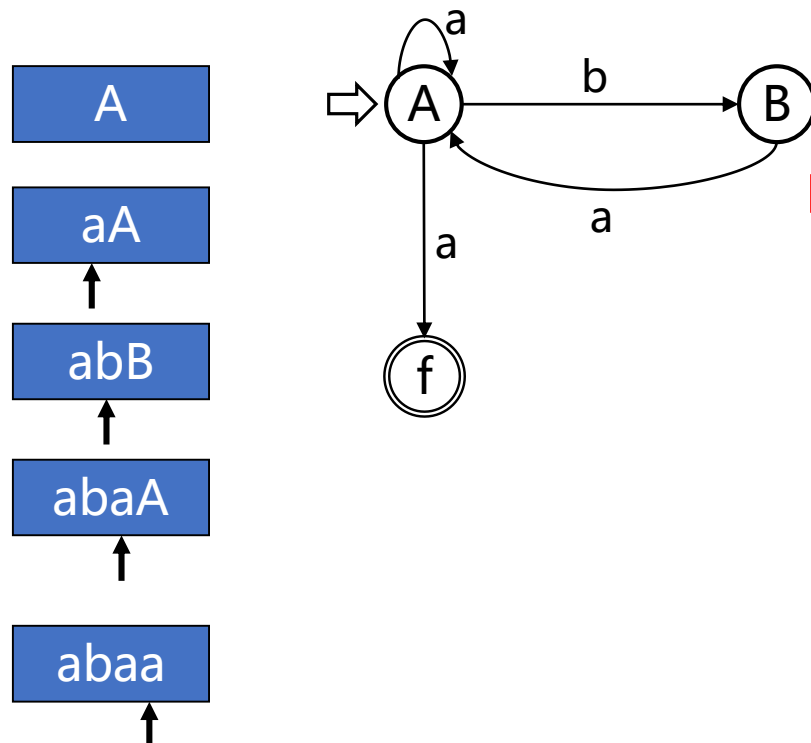
□ 3.4 词法分析器的实现

- 3.4.1 词法分析器边界
- 3.4.2 单词正规式
- 3.4.3 识别单词的DFA
- 3.4.4 单词识别算法

右线性正规文法 $G \rightarrow \text{FA } M$

□ 文法 $G[A]: A \rightarrow aA, A \rightarrow a, A \rightarrow bB, B \rightarrow aA$

abaa 的识别过程: $A \Rightarrow aA \Rightarrow abB \Rightarrow abaA \Rightarrow abaa$



□ 右线性正规文法 $G = (V_N, V_T, P, S)$ 构造 FA

- V_N 中的每个非终结符号视为状态符号;
 - 增加一个终态符号 f , 且 $f \notin V_N$;
 - 令 $M = (V_N \cup \{f\}, V_T, \delta, \{S\}, \{f\})$:
- ① 对产生式 $A \rightarrow a$, 令 $\delta(A, a) = f$
 - ② 对产生式 $A \rightarrow aB$, 令 $\delta(A, a) = B$

右线性正规文法 $G \rightarrow \text{FA } M$

□ 右线性正规文法 $G = (V_N, V_T, P, S)$ 构造 FA

- ① 对产生式 $A \rightarrow a$, 令 $\delta(A, a) = f$
- ② 对产生式 $A \rightarrow aB$, 令 $\delta(A, a) = B$

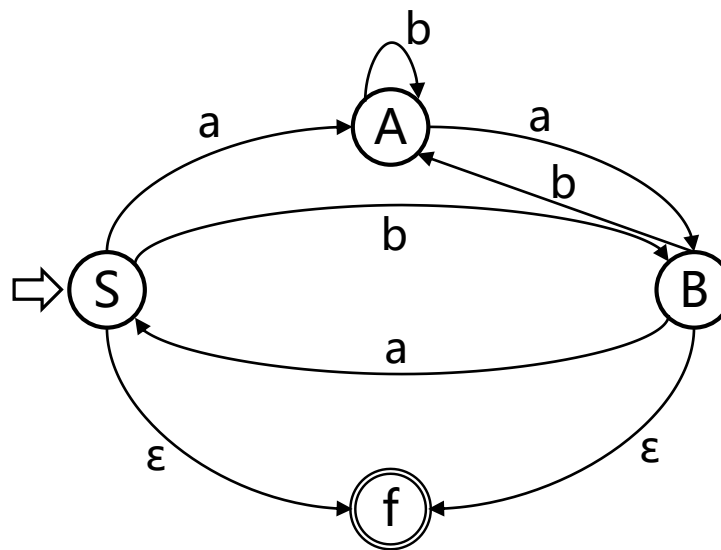
□ 【证明】 $L(G) = L(M)$

- G 在 $S \xRightarrow{+} w$ 的推导过程中, 利用 $A \rightarrow aB$ 一次, 就相当于在 M 中从状态 A 经过标记为 a 的箭弧到达状态 B (包括 $a = \varepsilon$) ;
- 推导的最后, 利用 $A \rightarrow a$ 一次, 相当于在 M 中从状态 A 经过标记为 a 的箭弧到达终结状态 f ;
- 综上, G 中在 $S \xRightarrow{+} w$ 的充要条件是: 在 M 中从状态 S 到终结状态 f 存在一条通路, 其上所有箭弧的标记符号依次连接起来恰好等于 w ;
- 即: $w \in L(G)$ 当且仅当 $w \in L(M)$, 因此 $L(G) = L(M)$ 。

右线性正规文法 $G \rightarrow$ FA M

□ 【例】构造如下文法 $G[S]$ 的FA

$S \rightarrow aA$
$S \rightarrow bB$
$S \rightarrow \varepsilon$
$A \rightarrow aB$
$A \rightarrow bA$
$B \rightarrow aS$
$B \rightarrow bA$
$B \rightarrow \varepsilon$



□ 3.1 对词法分析器的设计

- 3.1.1 词法分析器的任务
- 3.1.2 词法分析器设计需要考虑的问题
- 3.1.3 状态转换图

□ 3.2 有限自动机

- 3.2.1 确定有限自动机
- 3.2.2 非确定有限自动机
- 3.2.3 非确定有限自动机确定化
- 3.2.4 确定有限自动机化简
- 3.2.5 正规式与有限自动机的等价性

□ 3.3 正规文法

- 3.3.1 右线性文法转有限自动机
- 3.3.2 左线性文法转有限自动机

- 3.3.3 有限自动机转右线性文法
- 3.3.4 有限自动机转左线性文法
- 3.3.5 正规式转右线性文法
- 3.3.6 正规式转左线性文法
- 3.3.7 正规文法转正规式
- 3.3.8 三种工具的转换
- 3.3.9 有限自动机转正规式

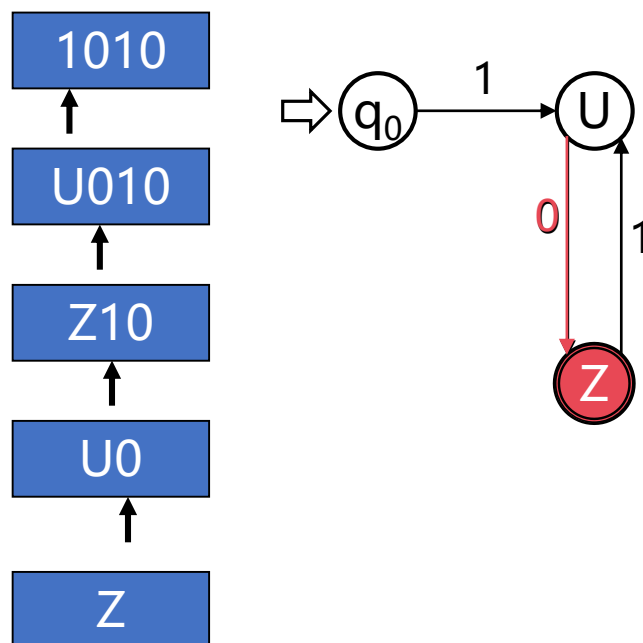
□ 3.4 词法分析器的实现

- 3.4.1 词法分析器边界
- 3.4.2 单词正规式
- 3.4.3 识别单词的DFA
- 3.4.4 单词识别算法

左线性正规文法 $G \rightarrow \text{FA } M$

□ 文法 $G[Z]: Z \rightarrow U0, Z \rightarrow V1, U \rightarrow Z1, U \rightarrow 1, V \rightarrow Z0, V \rightarrow 0$

1010的识别过程: $Z \Rightarrow U0 \Rightarrow Z10 \Rightarrow U010 \Rightarrow 1010$



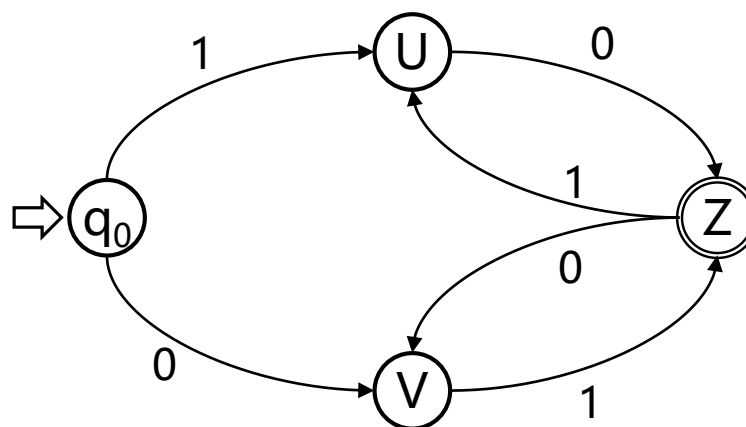
□ 左线性正规文法 $G = (V_N, V_T, P, S)$ 构造 FA

- V_N 中的每个非终结符号视为状态符号;
- 增加一个初态符号 q_0 , 且 $q_0 \notin V_N$;
- 令 $M = (V_N \cup \{q_0\}, V_T, \delta, \{q_0\}, \{S\})$:
- ① 对产生式 $A \rightarrow a$, 令 $\delta(q_0, a) = A$
- ② 对产生式 $A \rightarrow Ba$, 令 $\delta(B, a) = A$
- 与前述类似可证明: $L(G) = L(M)$

左线性正规文法 $G \rightarrow$ FA M

□ 【例】构造如下文法 $G[Z]$ 的FA

$Z \rightarrow U0$
$Z \rightarrow V1$
$U \rightarrow Z1$
$U \rightarrow 1$
$V \rightarrow Z0$
$V \rightarrow 0$



□ 3.1 对词法分析器的设计

- 3.1.1 词法分析器的任务
- 3.1.2 词法分析器设计需要考虑的问题
- 3.1.3 状态转换图

□ 3.2 有限自动机

- 3.2.1 确定有限自动机
- 3.2.2 非确定有限自动机
- 3.2.3 非确定有限自动机确定化
- 3.2.4 确定有限自动机化简
- 3.2.5 正规式与有限自动机的等价性

□ 3.3 正规文法

- 3.3.1 右线性文法转有限自动机
- 3.3.2 左线性文法转有限自动机

➤ 3.3.3 有限自动机转右线性文法

- 3.3.4 有限自动机转左线性文法
- 3.3.5 正规式转右线性文法
- 3.3.6 正规式转左线性文法
- 3.3.7 正规文法转正规式
- 3.3.8 三种工具的转换
- 3.3.9 有限自动机转正规式

□ 3.4 词法分析器的实现

- 3.4.1 词法分析器边界
- 3.4.2 单词正规式
- 3.4.3 识别单词的DFA
- 3.4.4 单词识别算法

FA $M \rightarrow$ 右线性正规文法 G_R

□ 右线性正规文法 $G = (V_N, V_T, P, S)$ 构造 FA

- ① 对产生式 $A \rightarrow a$, 令 $\delta(A, a) = f$
- ② 对产生式 $A \rightarrow aB$, 令 $\delta(A, a) = B$

□ FA $M = (S, \Sigma, \delta, \{s_0\}, F)$, 构造右线性文法 G_R

(1) 若 $s_0 \notin F$, 令 $G_R = (S, \Sigma, P, s_0)$, 其中 P

➤ 对 $\forall a \in \Sigma, A \in S, B \in S$, 若有 $\delta(A, a) = B$, 则:

- ① 当 $B \notin F$ 时, 令 $A \rightarrow aB$;
- ② 当 $B \in F$ 时, 令 $A \rightarrow a|aB$ 。

FA $M \rightarrow$ 右线性正规文法 G_R

□ FA $M = (S, \Sigma, \delta, \{s_0\}, F)$, 构造右线性文法 G_R

(1) 若 $s_0 \notin F$, 令 $G_R = (S, \Sigma, P, s_0)$, 其中 P

➤ 对 $\forall a \in \Sigma, A \in S, B \in S$, 若有 $\delta(A, a) = B$, 则:

① 当 $B \notin F$ 时, 令 $A \rightarrow aB$;

② 当 $B \in F$ 时, 令 $A \rightarrow a|aB$ 。

□ 【证明 (1)】

➤ 对 $\forall w \in \Sigma^*$, 不妨设 $w = a_1 a_2 \dots a_k$, 其中 $a_i \in \Sigma$

➤ 若 $S \xRightarrow{+} w$, 则存在一个最右推导: $s_0 \Rightarrow a_1 A_1 \Rightarrow a_1 a_2 A_2 \Rightarrow \dots \Rightarrow a_1 \dots a_k$

➤ 因此 M 中有一条从 s_0 出发, 依次经过 A_1, \dots, A_{k-1} , 最后到达终态的通路, 该通路上所有箭弧的标记依次为 a_1, \dots, a_k 。

➤ 反之, 若 M 中 s_0 到终态有通路, 则 M 中必存在该字串的推导。

➤ 因此: $w \in L(M) \Leftrightarrow w \in L(G_R)$, 故有 $L(M) = L(G_R)$

FA $M \rightarrow$ 右线性正规文法 G_R

□ FA $M = (S, \Sigma, \delta, \{s_0\}, F)$, 构造右线性文法 G_R

(1) 若 $s_0 \notin F$, 令 $G_R = (S, \Sigma, P, s_0)$, 其中 P

➤ 对 $\forall a \in \Sigma, A \in S, B \in S$, 若有 $\delta(A, a) = B$, 则:

① 当 $B \notin F$ 时, 令 $A \rightarrow aB$;

② 当 $B \in F$ 时, 令 $A \rightarrow a|aB$ 。

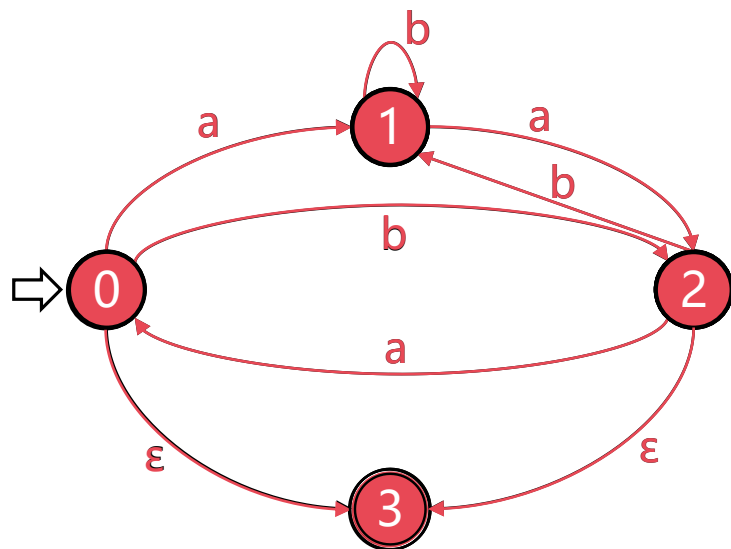
(2) 若 $s_0 \in F$, 则 $\delta(s_0, \varepsilon) = s_0 \Rightarrow L(G_R) = L(M) - \{\varepsilon\}$, 改造 G_R 如下:

➤ $P' = P \cup \{s'_0 \rightarrow s_0|\varepsilon\}$;

➤ $G'_R = (S \cup \{s'_0\}, \Sigma, P', s'_0)$ 。

FA $M \rightarrow$ 右线性正规文法 G_R

□ 【例】构造右线性文法 G_R



□ $G_R = (\{S_0, S_1, S_2, S_3\}, \{a, b\}, P, S_0)$,
其中 P

$S_0 \rightarrow aS_1$
$S_0 \rightarrow bS_2$
$S_0 \rightarrow \varepsilon S_3$
$S_1 \rightarrow aS_2$
$S_1 \rightarrow bS_1$
$S_2 \rightarrow aS_0$
$S_2 \rightarrow bS_1$
$S_2 \rightarrow \varepsilon S_3$


 $S_0 \rightarrow \varepsilon$

 $S_2 \rightarrow \varepsilon$

可以消除的产生式:

➤ 无用符号、无用产生式、单非产生式

整理得: $S_0 \rightarrow aS_1 | bS_2 | \varepsilon$

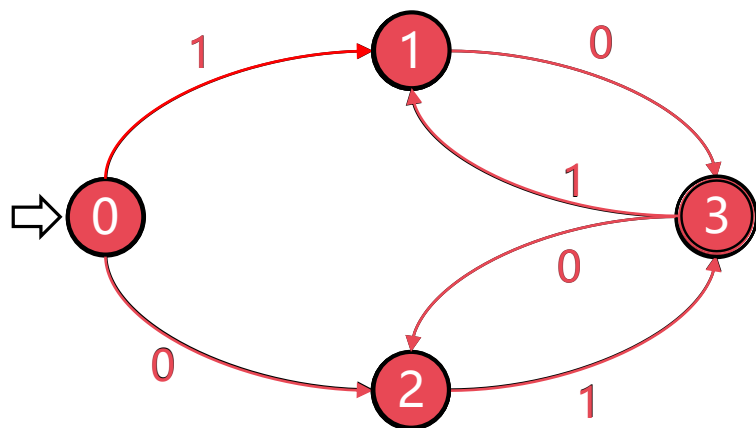
➤ $S_1 \rightarrow aS_2 | bS_1$

➤ $S_2 \rightarrow aS_0 | bS_1 | \varepsilon$

□ $S_0 \notin F$, 结束。

FA $M \rightarrow$ 右线性正规文法 G_R

□ 【例】构造右线性文法 G_R



□ $G_R = (\{S_0, S_1, S_2, S_3\}, \{0, 1\}, P, S_0)$,

其中 P

$S_0 \rightarrow 0S_2$

$S_0 \rightarrow 1S_1$

$S_1 \rightarrow 0 0S_3$

$S_2 \rightarrow 1 1S_3$

$S_3 \rightarrow 0S_2$

$S_3 \rightarrow 1S_1$

整理得:

➤ $S_0 \rightarrow 0S_2|1S_1$

➤ $S_1 \rightarrow 0|0S_3$

➤ $S_2 \rightarrow 1|1S_3$

➤ $S_3 \rightarrow 0S_2|1S_1$

□ $S_0 \notin F$, 结束。

□ 3.1 对词法分析器的设计

- 3.1.1 词法分析器的任务
- 3.1.2 词法分析器设计需要考虑的问题
- 3.1.3 状态转换图

□ 3.2 有限自动机

- 3.2.1 确定有限自动机
- 3.2.2 非确定有限自动机
- 3.2.3 非确定有限自动机确定化
- 3.2.4 确定有限自动机化简
- 3.2.5 正规式与有限自动机的等价性

□ 3.3 正规文法

- 3.3.1 右线性文法转有限自动机
- 3.3.2 左线性文法转有限自动机

➤ 3.3.3 有限自动机转右线性文法

➤ 3.3.4 有限自动机转左线性文法

➤ 3.3.5 正规式转右线性文法

➤ 3.3.6 正规式转左线性文法

➤ 3.3.7 正规文法转正规式

➤ 3.3.8 三种工具的转换

➤ 3.3.9 有限自动机转正规式

□ 3.4 词法分析器的实现

➤ 3.4.1 词法分析器边界

➤ 3.4.2 单词正规式

➤ 3.4.3 识别单词的DFA

➤ 3.4.4 单词识别算法

FA $M \rightarrow$ 左线性正规文法 R_L

□ 左线性正规文法 $G = (V_N, V_T, P, S)$ 构造 FA

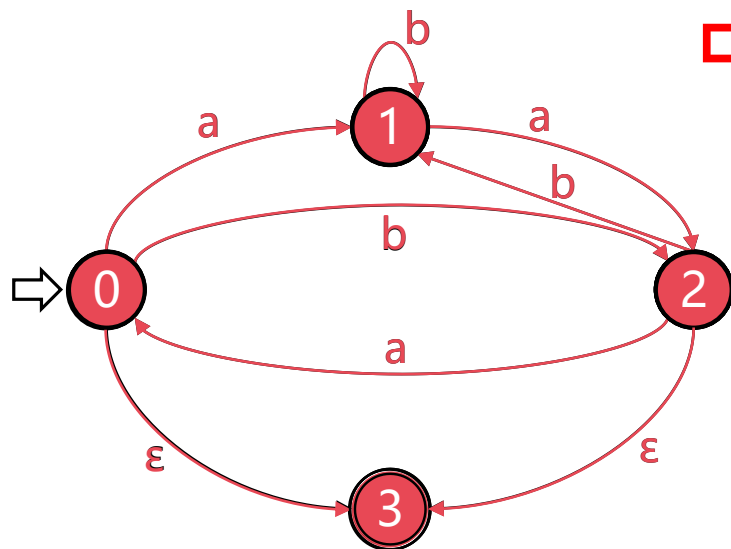
- ① 对产生式 $A \rightarrow a$, 令 $\delta(q_0, a) = A$
- ② 对产生式 $A \rightarrow Ba$, 令 $\delta(B, a) = A$

□ FA $M = (S, \Sigma, \delta, \{s_0\}, \{f\})$, 构造左线性文法 G_L

- 对于多终态的, 增加新的终态, 从原终态向新终态引 ε 弧
- 若初态 s_0 有射入弧, 则令 $G_L = (S, \Sigma, P, f)$, 否则令 $G_L = (S - \{s_0\}, \Sigma, P, f)$
- P : 对 $\forall a \in \Sigma, A \in S, B \in S$, 若有 $\delta(A, a) = B$, 则:
 - ① 当 $A \neq s_0$ 时, 令 $B \rightarrow Aa$;
 - ② 当 $A = s_0$ 时, 令 $B \rightarrow Aa|a$ 。

FA $M \rightarrow$ 左线性正规文法 G_L

□ 【例】构造左线性文法 G_L



□ $G_L = (\{S_0, S_1, S_2, S_3\}, \{a, b\}, P, S_3)$, 其中 P

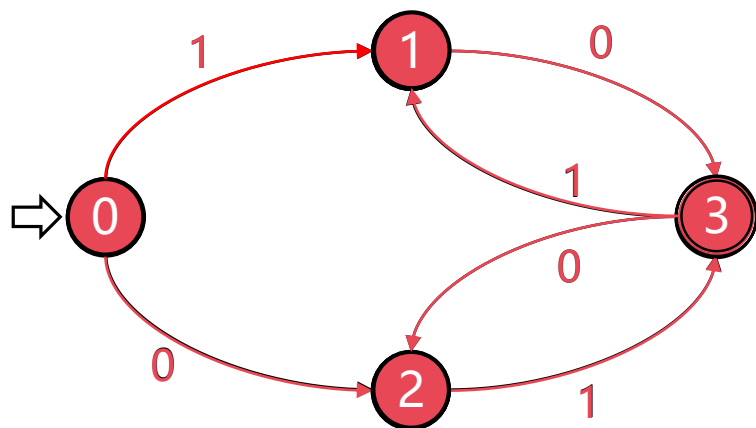
$S_1 \rightarrow a S_0a$
$S_2 \rightarrow b S_0b$
$S_3 \rightarrow \varepsilon S_0$
$S_2 \rightarrow S_1a$
$S_1 \rightarrow S_1b$
$S_0 \rightarrow S_2a$
$S_1 \rightarrow S_2b$
$S_3 \rightarrow S_2$

整理得:

- $S_0 \rightarrow S_2a$
- $S_1 \rightarrow a|S_0a|S_1b|S_2b$
- $S_2 \rightarrow b|S_0b|S_1a$
- $S_3 \rightarrow \varepsilon|b|S_0b|S_1a|S_2a$

FA $M \rightarrow$ 左线性正规文法 G_L

□ 【例】构造左线性文法 G_L



□ $G_L = (\{S_1, S_2, S_3\}, \{0, 1\}, P, S_3)$, 其中 P

整理得:

- $S_1 \rightarrow 1|S_0 1|S_3 1$
- $S_2 \rightarrow 0|S_0 0|S_3 0$
- $S_3 \rightarrow S_1 0|S_2 1$

$S_2 \rightarrow 0 S_0 0$
$S_1 \rightarrow 1 S_0 1$
$S_3 \rightarrow S_1 0$
$S_3 \rightarrow S_2 1$
$S_2 \rightarrow S_3 0$
$S_1 \rightarrow S_3 1$

□ 3.1 对词法分析器的设计

- 3.1.1 词法分析器的任务
- 3.1.2 词法分析器设计需要考虑的问题
- 3.1.3 状态转换图

□ 3.2 有限自动机

- 3.2.1 确定有限自动机
- 3.2.2 非确定有限自动机
- 3.2.3 非确定有限自动机确定化
- 3.2.4 确定有限自动机化简
- 3.2.5 正规式与有限自动机的等价性

□ 3.3 正规文法

- 3.3.1 右线性文法转有限自动机
- 3.3.2 左线性文法转有限自动机

➤ 3.3.3 有限自动机转右线性文法

➤ 3.3.4 有限自动机转左线性文法

➤ 3.3.5 正规式转右线性文法

➤ 3.3.6 正规式转左线性文法

➤ 3.3.7 正规文法转正规式

➤ 3.3.8 三种工具的转换

➤ 3.3.9 有限自动机转正规式

□ 3.4 词法分析器的实现

➤ 3.4.1 词法分析器边界

➤ 3.4.2 单词正规式

➤ 3.4.3 识别单词的DFA

➤ 3.4.4 单词识别算法

正规式 $r \rightarrow$ 右线性正规文法 G_R

□ $G_R = (V_N, V_T, P, S)$

- (1) 令 $V_T = \Sigma$
- (2) 增加产生式 $S \rightarrow r$, 其中 S 为开始符号
- (3) 对已有的产生式, 按以下原则进行变换, 直到每个产生式只有一个终结符号为止:

$A \rightarrow xy \rightarrow A \rightarrow xB, B \rightarrow y$

$A \rightarrow x|y \rightarrow A \rightarrow x, A \rightarrow y$

$A \rightarrow x^* \rightarrow A \rightarrow xA, A \rightarrow \varepsilon$

$A \rightarrow x^*y \rightarrow A \rightarrow xA, A \rightarrow y$

- (4) 第(3)步所确定的产生式组为 P , 而非终结符号组成 V_N 。

正规式 $r \rightarrow$ 右线性正规文法 G_R

【例】将正规式 $r = (a|b)^*(aa|bb)(a|b)^*$ 转为右线性文法 G_R , 使 $L(G_R) = L(r)$

【解】 $G_R = (V_N, \{a, b\}, P, S)$ 。

$S \rightarrow (a|b)^*(aa|bb)(a|b)^*$

$S \rightarrow (a|b)S$

$S \rightarrow (aa|bb)(a|b)^*$

$S \rightarrow aS$

$S \rightarrow bS$

$S \rightarrow (aa|bb)(a|b)^*$

$S \rightarrow aS$

$S \rightarrow bS$

$S \rightarrow (aa|bb)A$

$A \rightarrow (a|b)^*$

$S \rightarrow aS$

$S \rightarrow bS$

$S \rightarrow aaA$

$S \rightarrow bbA$

$A \rightarrow (a|b)^*$

$A \rightarrow xy \Rightarrow A \rightarrow xB, B \rightarrow y$

$A \rightarrow x|y \Rightarrow A \rightarrow x, A \rightarrow y$

$A \rightarrow x^* \Rightarrow A \rightarrow xA, A \rightarrow \varepsilon$

$A \rightarrow x^*y \Rightarrow A \rightarrow xA, A \rightarrow y$

$S \rightarrow aS$

$S \rightarrow bS$

$S \rightarrow aB$

$B \rightarrow aA$

$S \rightarrow bbA$

$A \rightarrow (a|b)^*$

$S \rightarrow aS$

$S \rightarrow bS$

$S \rightarrow aB$

$B \rightarrow aA$

$S \rightarrow bC$

$C \rightarrow bA$

$A \rightarrow (a|b)^*$

正规式 $r \rightarrow$ 右线性正规文法 G_R

【例】将正规式 $r = (a|b)^*(aa|bb)(a|b)^*$ 转为右线性文法 G_R , 使 $L(G_R) = L(r)$

【解】 $G_R = (V_N, \{a, b\}, P, S)$ 。

$S \rightarrow aS$
$S \rightarrow bS$
$S \rightarrow aB$
$B \rightarrow aA$
$S \rightarrow bC$
$C \rightarrow bA$
$A \rightarrow (a b)^*$

$S \rightarrow aS$
$S \rightarrow bS$
$S \rightarrow aB$
$B \rightarrow aA$
$S \rightarrow bC$
$C \rightarrow bA$
$A \rightarrow (a b)A$
$A \rightarrow \varepsilon$

$S \rightarrow aS$
$S \rightarrow bS$
$S \rightarrow aB$
$B \rightarrow aA$
$S \rightarrow bC$
$C \rightarrow bA$
$A \rightarrow aA$
$A \rightarrow bA$
$A \rightarrow \varepsilon$

$$A \rightarrow xy \Rightarrow A \rightarrow xB, B \rightarrow y$$

$$A \rightarrow x|y \Rightarrow A \rightarrow x, A \rightarrow y$$

$$A \rightarrow x^* \Rightarrow A \rightarrow xA, A \rightarrow \varepsilon$$

$$A \rightarrow x^*y \Rightarrow A \rightarrow xA, A \rightarrow y$$

【整理】 $G_R = (\{S, A, B, C\}, \{a, b\}, P, S)$,

P :

$$S \rightarrow aS|bS|aB|bC$$

$$A \rightarrow aA|bA|\varepsilon$$

$$B \rightarrow aA$$

$$C \rightarrow bA$$

□ 3.1 对词法分析器的设计

- 3.1.1 词法分析器的任务
- 3.1.2 词法分析器设计需要考虑的问题
- 3.1.3 状态转换图

□ 3.2 有限自动机

- 3.2.1 确定有限自动机
- 3.2.2 非确定有限自动机
- 3.2.3 非确定有限自动机确定化
- 3.2.4 确定有限自动机化简
- 3.2.5 正规式与有限自动机的等价性

□ 3.3 正规文法

- 3.3.1 右线性文法转有限自动机
- 3.3.2 左线性文法转有限自动机

- 3.3.3 有限自动机转右线性文法
- 3.3.4 有限自动机转左线性文法
- 3.3.5 正规式转右线性文法
- **3.3.6 正规式转左线性文法**
- 3.3.7 正规文法转正规式
- 3.3.8 三种工具的转换
- 3.3.9 有限自动机转正规式

□ 3.4 词法分析器的实现

- 3.4.1 词法分析器边界
- 3.4.2 单词正规式
- 3.4.3 识别单词的DFA
- 3.4.4 单词识别算法

正规式 $r \rightarrow$ 左线性正规文法 G_L

□ $G_L = (V_N, V_T, P, S)$

- (1) 令 $V_T = \Sigma$
- (2) 增加产生式 $S \rightarrow r$, 其中 S 为开始符号
- (3) 对已有的产生式, 按以下原则进行变换, 直到每个产生式只有一个终结符号为止:

$A \rightarrow xy \rightarrow A \rightarrow By, B \rightarrow x$

$A \rightarrow x|y \rightarrow A \rightarrow x, A \rightarrow y$

$A \rightarrow x^* \rightarrow A \rightarrow Ax, A \rightarrow \varepsilon$

$A \rightarrow xy^* \rightarrow A \rightarrow x, A \rightarrow Ay$

- (4) 第(3)步所确定的产生式组为 P , 而非终结符号组成 V_N 。

正规式 $r \rightarrow$ 左线性正规文法 G_L

【例】将正规式 $r = (a|b)^*(aa|bb)(a|b)^*$ 转为左线性文法 G_L , 使 $L(G_L) = L(r)$

【解】 $G_L = (V_N, \{a, b\}, P, S)$ 。

$S \rightarrow (a b)^*(aa bb)(a b)^*$

$S \rightarrow (a b)^*(aa bb)$

$S \rightarrow S(a b)$

$S \rightarrow A(aa bb)$

$A \rightarrow (a b)^*$

$S \rightarrow S(a b)$

$S \rightarrow Aaa$

$S \rightarrow Abb$

$A \rightarrow (a b)^*$

$S \rightarrow S(a b)$

$S \rightarrow Ba$

$B \rightarrow Aa$

$S \rightarrow Abb$

$A \rightarrow (a b)^*$

$S \rightarrow S(a b)$

$A \rightarrow xy \Rightarrow A \rightarrow By, B \rightarrow x$
--

$A \rightarrow x y \Rightarrow A \rightarrow x, A \rightarrow y$
--

$A \rightarrow x^* \Rightarrow A \rightarrow Ax, A \rightarrow \varepsilon$

$A \rightarrow xy^* \Rightarrow A \rightarrow x, A \rightarrow Ay$
--

$S \rightarrow Ba$

$B \rightarrow Aa$

$S \rightarrow Cb$

$C \rightarrow Ab$

$A \rightarrow (a b)^*$

$S \rightarrow S(a b)$

$S \rightarrow Ba$

$B \rightarrow Aa$

$S \rightarrow Cb$

$C \rightarrow Ab$

$A \rightarrow A(a b)$

$A \rightarrow \varepsilon$

$S \rightarrow S(a b)$

正规式 $r \rightarrow$ 左线性正规文法 G_L

【例】将正规式 $r = (a|b)^*(aa|bb)(a|b)^*$ 转为左线性文法 G_L , 使 $L(G_L) = L(r)$

【解】 $G_L = (V_N, \{a, b\}, P, S)$ 。

$S \rightarrow Ba$	$S \rightarrow Ba$
$B \rightarrow Aa$	$B \rightarrow Aa$
$S \rightarrow Cb$	$S \rightarrow Cb$
$C \rightarrow Ab$	$C \rightarrow Ab$
$A \rightarrow A(a b)$	$A \rightarrow Aa$
$A \rightarrow \varepsilon$	$A \rightarrow Ab$
$S \rightarrow S(a b)$	$A \rightarrow \varepsilon$
	$S \rightarrow Sa$
	$S \rightarrow Sb$

$$A \rightarrow xy \Rightarrow A \rightarrow By, B \rightarrow x$$

$$A \rightarrow x|y \Rightarrow A \rightarrow x, A \rightarrow y$$

$$A \rightarrow x^* \Rightarrow A \rightarrow Ax, A \rightarrow \varepsilon$$

$$A \rightarrow xy^* \Rightarrow A \rightarrow x, A \rightarrow Ay$$

【整理】 $G_L = (\{a, b\}, \{S, A, B, C\}, S, P)$,

P :

$$S \rightarrow Ba|Cb|Sa|Sb$$

$$A \rightarrow Aa|Ab|\varepsilon$$

$$B \rightarrow Aa$$

$$C \rightarrow Ab$$

□ 3.1 对词法分析器的设计

- 3.1.1 词法分析器的任务
- 3.1.2 词法分析器设计需要考虑的问题
- 3.1.3 状态转换图

□ 3.2 有限自动机

- 3.2.1 确定有限自动机
- 3.2.2 非确定有限自动机
- 3.2.3 非确定有限自动机确定化
- 3.2.4 确定有限自动机化简
- 3.2.5 正规式与有限自动机的等价性

□ 3.3 正规文法

- 3.3.1 右线性文法转有限自动机
- 3.3.2 左线性文法转有限自动机

- 3.3.3 有限自动机转右线性文法
- 3.3.4 有限自动机转左线性文法
- 3.3.5 正规式转右线性文法
- 3.3.6 正规式转左线性文法
- 3.3.7 正规文法转正规式
- 3.3.8 三种工具的转换
- 3.3.9 有限自动机转正规式

□ 3.4 词法分析器的实现

- 3.4.1 词法分析器边界
- 3.4.2 单词正规式
- 3.4.3 识别单词的DFA
- 3.4.4 单词识别算法

正规文法 \rightarrow 正规式 r

- 反复利用以下规则合并文法的产生式，最后只剩下一个开始符号定义的产生式，并且产生式右部不含非终结符合。

- 右线性文法：

$$A \rightarrow \alpha B, B \rightarrow \beta \quad \longrightarrow \quad A \rightarrow \alpha\beta$$

$$A \rightarrow \alpha_1, A \rightarrow \alpha_2 \quad \longrightarrow \quad A \rightarrow \alpha_1 | \alpha_2$$

$$A \rightarrow \alpha A, A \rightarrow \beta \quad \longrightarrow \quad A \rightarrow \alpha^* \beta$$

- 左线性文法：

$$A \rightarrow B\alpha, B \rightarrow \beta \quad \longrightarrow \quad A \rightarrow \beta\alpha$$

$$A \rightarrow \alpha_1, A \rightarrow \alpha_2 \quad \longrightarrow \quad A \rightarrow \alpha_1 | \alpha_2$$

$$A \rightarrow A\alpha, A \rightarrow \beta \quad \longrightarrow \quad A \rightarrow \beta\alpha^*$$

正规文法 \rightarrow 正规式 r

【例】将以下正规文法 $G[S]$ 转换为正规式 r , 使 $L(r) = L(G)$

$$S \rightarrow aS | bS | aB | bC$$

$$A \rightarrow aA | bA | \varepsilon$$

$$B \rightarrow aA$$

$$C \rightarrow bA$$

$$\textcircled{1} S \Rightarrow (a|b)S \mid (aB|bC) \Rightarrow (a|b)^*(aB|bC)$$

$$\textcircled{2} S \overset{+}{\Rightarrow} (a|b)^*(aaA|bbA) \Rightarrow (a|b)^*(aa|bb)A$$

$$\textcircled{3} A \Rightarrow (a|b)A \mid \varepsilon \Rightarrow (a|b)^*$$

$$\textcircled{4} \textcircled{3} \text{代入} \textcircled{2}, \text{得: } S \overset{+}{\Rightarrow} (a|b)^*(aa|bb)(a|b)^*$$

正规文法 \rightarrow 正规式 r

【例】将以下正规文法 $G[S]$ 转换为正规式 r , 使 $L(r) = L(G)$

$$S \rightarrow Ba|Cb|Sa|Sb$$

$$A \rightarrow Aa|Ab|\varepsilon$$

$$B \rightarrow Aa$$

$$C \rightarrow Ab$$

$$\textcircled{1} S \Rightarrow (Ba|Cb) \mid S(a|b) \Rightarrow (Ba|Cb)(a|b)^*$$

$$\textcircled{2} S \overset{+}{\Rightarrow} (Aaa|Abb)(a|b)^* \Rightarrow A(aa|bb)(a|b)^*$$

$$\textcircled{3} A \Rightarrow A(a|b) \mid \varepsilon \Rightarrow (a|b)^*$$

$$\textcircled{4} \textcircled{3} \text{代入} \textcircled{2}, \text{得: } S \overset{+}{\Rightarrow} (a|b)^*(aa|bb)(a|b)^*$$

□ 3.1 对词法分析器的设计

- 3.1.1 词法分析器的任务
- 3.1.2 词法分析器设计需要考虑的问题
- 3.1.3 状态转换图

□ 3.2 有限自动机

- 3.2.1 确定有限自动机
- 3.2.2 非确定有限自动机
- 3.2.3 非确定有限自动机确定化
- 3.2.4 确定有限自动机化简
- 3.2.5 正规式与有限自动机的等价性

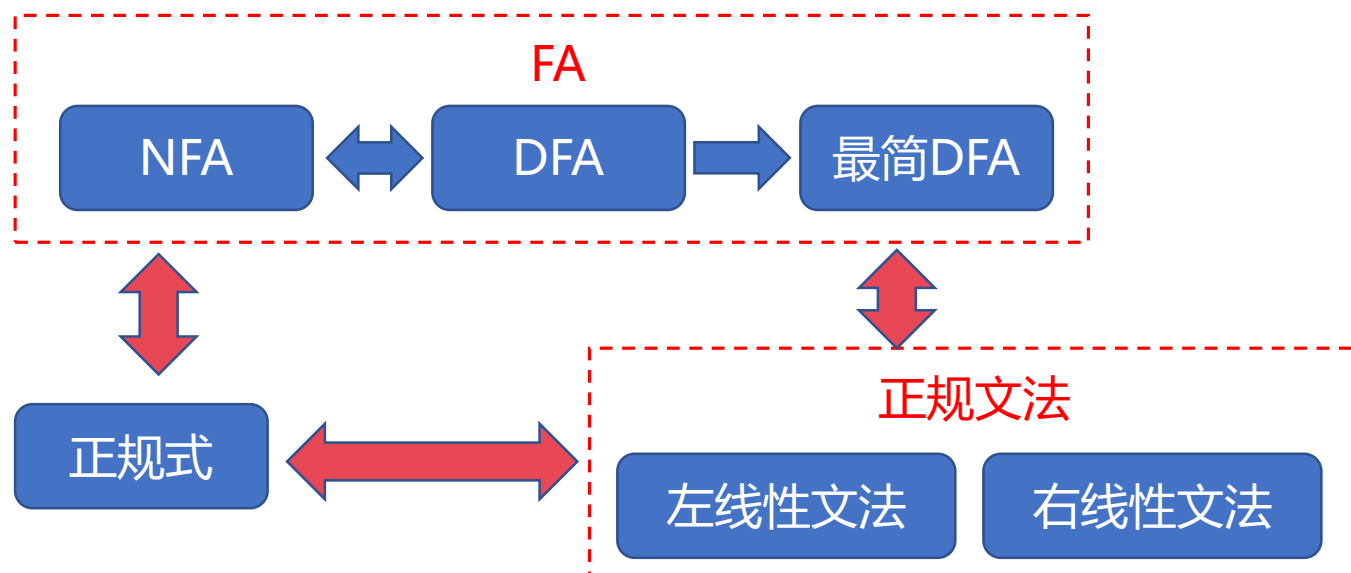
□ 3.3 正规文法

- 3.3.1 右线性文法转有限自动机
- 3.3.2 左线性文法转有限自动机

- 3.3.3 有限自动机转右线性文法
- 3.3.4 有限自动机转左线性文法
- 3.3.5 正规式转右线性文法
- 3.3.6 正规式转左线性文法
- 3.3.7 正规文法转正规式
- 3.3.8 三种工具的转换
- 3.3.9 有限自动机转正规式

□ 3.4 词法分析器的实现

- 3.4.1 词法分析器边界
- 3.4.2 单词正规式
- 3.4.3 识别单词的DFA
- 3.4.4 单词识别算法



□ 转换总结

- **FA内部**, NFA和DFA可以互转, DFA可以进一步化简为最简DFA;
- **正规文法内部**, 左、右线性文法互转并不容易, 一般**先转换为FA**, 再进一步转换为另一种正规文法;
- **FA与正规文法之间**、**正规文法与正规式**之间容易转换;
- **正规式容易转换为FA**, 但是**FA转换为正规式并不容易**, 一般需要**借助正规文法**完成转换。

□ 3.1 对词法分析器的设计

- 3.1.1 词法分析器的任务
- 3.1.2 词法分析器设计需要考虑的问题
- 3.1.3 状态转换图

□ 3.2 有限自动机

- 3.2.1 确定有限自动机
- 3.2.2 非确定有限自动机
- 3.2.3 非确定有限自动机确定化
- 3.2.4 确定有限自动机化简
- 3.2.5 正规式与有限自动机的等价性

□ 3.3 正规文法

- 3.3.1 右线性文法转有限自动机
- 3.3.2 左线性文法转有限自动机

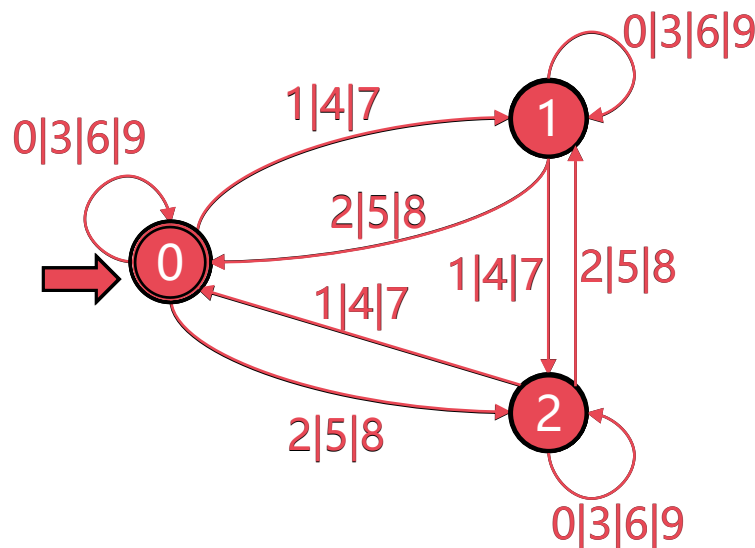
- 3.3.3 有限自动机转右线性文法
- 3.3.4 有限自动机转左线性文法
- 3.3.5 正规式转右线性文法
- 3.3.6 正规式转左线性文法
- 3.3.7 正规文法转正规式
- 3.3.8 三种工具的转换
- 3.3.9 有限自动机转正规式

□ 3.4 词法分析器的实现

- 3.4.1 词法分析器边界
- 3.4.2 单词正规式
- 3.4.3 识别单词的DFA
- 3.4.4 单词识别算法

FA $M \Rightarrow$ 正规式 r

【例】写出如下能被3整除的正规式



□ 先转右线性文法

$$S_0 \rightarrow (0|3|6|9) \mid (0|3|6|9)S_0$$

$$S_0 \rightarrow (1|4|7)S_1$$

$$S_0 \rightarrow (2|5|8)S_2$$

$$S_1 \rightarrow (0|3|6|9)S_1$$

$$S_1 \rightarrow (1|4|7)S_2$$

$$S_1 \rightarrow (2|5|8) \mid (2|5|8)S_0$$

$$S_2 \rightarrow (0|3|6|9)S_2$$

$$S_2 \rightarrow (1|4|7) \mid (1|4|7)S_0$$

$$S_2 \rightarrow (2|5|8)S_1$$

$$S'_0 \rightarrow S_0 \mid \varepsilon$$

➤ 若有 $\delta(A, a) = B$, 则:

① 当 $B \notin F$ 时, 令 $A \rightarrow aB$;

② 当 $B \in F$ 时, 令 $A \rightarrow a \mid aB$ 。

➤ 若 $s_0 \in F$, $P' = P \cup \{s'_0 \rightarrow s_0 \mid \varepsilon\}$

FA $M \Rightarrow$ 正规式 r

【例】写出如下能被3整除的正规式

$$S_0 \rightarrow (0|3|6|9) | (0|3|6|9)S_0$$

$$S_0 \rightarrow (1|4|7)S_1$$

$$S_0 \rightarrow (2|5|8)S_2$$

$$S_1 \rightarrow (0|3|6|9)S_1$$

$$S_1 \rightarrow (1|4|7)S_2$$

$$S_1 \rightarrow (2|5|8) | (2|5|8)S_0$$

$$S_2 \rightarrow (0|3|6|9)S_2$$

$$S_2 \rightarrow (1|4|7) | (1|4|7)S_0$$

$$S_2 \rightarrow (2|5|8)S_1$$

$$S'_0 \rightarrow S_0 | \varepsilon$$

再转正规式

$$\textcircled{1} S_0 \Rightarrow (0|3|6|9) | (1|4|7)S_1 | (2|5|8)S_2 | (0|3|6|9)S_0$$

$$\Rightarrow (0|3|6|9)^*((0|3|6|9) | (1|4|7)S_1 | (2|5|8)S_2)$$

$$\textcircled{2} S_1 \Rightarrow (2|5|8) | (2|5|8)S_0 | (1|4|7)S_2 | (0|3|6|9)S_1$$

$$\Rightarrow (0|3|6|9)^*((2|5|8) | (2|5|8)S_0 | (1|4|7)S_2)$$

$$\textcircled{3} S_2 \Rightarrow (1|4|7) | (1|4|7)S_0 | (2|5|8)S_1 | (0|3|6|9)S_2$$

$$\Rightarrow (0|3|6|9)^*((1|4|7) | (1|4|7)S_0 | (2|5|8)S_1)$$

$$\textcircled{4} S'_0 \rightarrow S_0 | \varepsilon$$

FA $M \Rightarrow$ 正规式 r

【例】写出如下能被3整除的正规式

$$\textcircled{1} S_0 \Rightarrow (0|3|6|9)^*((0|3|6|9) | (1|4|7)S_1 | (2|5|8)S_2)$$

$$\textcircled{2} S_1 \Rightarrow (0|3|6|9)^*((2|5|8) | (2|5|8)S_0 | (1|4|7)S_2)$$

$$\textcircled{3} S_2 \Rightarrow (0|3|6|9)^*((1|4|7) | (1|4|7)S_0 | (2|5|8)S_1)$$

$$\textcircled{4} S'_0 \rightarrow S_0 | \varepsilon$$

为方便书写, 记: $\alpha = 0|3|6|9, \beta = 1|4|7, \gamma = 2|5|8$, 则:

$$\textcircled{1} S_0 \Rightarrow \alpha^*(\alpha | \beta S_1 | \gamma S_2)$$

$$\textcircled{2} S_1 \Rightarrow \alpha^*(\gamma | \gamma S_0 | \beta S_2)$$

$$\textcircled{3} S_2 \Rightarrow \alpha^*(\beta | \beta S_0 | \gamma S_1)$$

$$\textcircled{4} S'_0 \rightarrow S_0 | \varepsilon$$

FA $M \Rightarrow$ 正规式 r

【例】写出如下能被3整除的正规式

$$\textcircled{1} S_0 \Rightarrow \alpha^*(\alpha | \beta S_1 | \gamma S_2)$$

$$\textcircled{2} S_1 \Rightarrow \alpha^*(\gamma | \gamma S_0 | \beta S_2)$$

$$\textcircled{3} S_2 \Rightarrow \alpha^*(\beta | \beta S_0 | \gamma S_1)$$

$$\textcircled{4} S'_0 \rightarrow S_0 | \varepsilon$$

③代入②:

$$\begin{aligned} \textcircled{5} S_1 &\Rightarrow \alpha^*(\gamma | \gamma S_0 | \beta(\alpha^*(\beta | \beta S_0 | \gamma S_1))) \\ &\Rightarrow \alpha^*(\gamma | \gamma S_0 | \beta\alpha^*\beta | \beta\alpha^*\beta S_0 | \beta\alpha^*\gamma S_1) \\ &\Rightarrow (\alpha^*\beta\alpha^*\gamma)^*\alpha^*(\gamma | \gamma S_0 | \beta\alpha^*\beta | \beta\alpha^*\beta S_0) \end{aligned}$$

FA $M \Rightarrow$ 正规式 r

【例】写出如下能被3整除的正规式

$$\textcircled{1} S_0 \Rightarrow \alpha^*(\alpha | \beta S_1 | \gamma S_2)$$

$$\textcircled{3} S_2 \Rightarrow \alpha^*(\beta | \beta S_0 | \gamma S_1)$$

$$\textcircled{4} S'_0 \rightarrow S_0 | \varepsilon$$

$$\textcircled{5} S_1 \Rightarrow (\alpha^* \beta \alpha^* \gamma)^* \alpha^* (\gamma | \gamma S_0 | \beta \alpha^* \beta | \beta \alpha^* \beta S_0)$$

③代入①:

$$\textcircled{6} S_0 \Rightarrow \alpha^*(\alpha | \beta S_1 | \gamma \alpha^*(\beta | \beta S_0 | \gamma S_1))$$

$$\Rightarrow \alpha^*(\alpha | \beta S_1 | \gamma \alpha^* \beta | \gamma \alpha^* \beta S_0 | \gamma \alpha^* \gamma S_1)$$

$$\Rightarrow (\alpha^* \gamma \alpha^* \beta)^* \alpha^*(\alpha | \beta S_1 | \gamma \alpha^* \beta | \gamma \alpha^* \gamma S_1)$$

FA $M \Rightarrow$ 正规式 r

【例】写出如下能被3整除的正规式

$$\textcircled{4} S'_0 \rightarrow S_0 | \varepsilon$$

$$\textcircled{5} S_1 \Rightarrow (\alpha^* \beta \alpha^* \gamma)^* \alpha^* (\gamma | \gamma S_0 | \beta \alpha^* \beta | \beta \alpha^* \beta S_0)$$

$$\textcircled{6} S_0 \Rightarrow (\alpha^* \gamma \alpha^* \beta)^* \alpha^* (\alpha | \beta S_1 | \gamma \alpha^* \beta | \gamma \alpha^* \gamma S_1)$$

⑤代入⑥:

$$\textcircled{7} S_0 \Rightarrow (\alpha^* \gamma \alpha^* \beta)^* (\alpha^* \alpha | \alpha^* \gamma \alpha^* \beta) | (\alpha^* \gamma \alpha^* \beta)^* (\alpha^* \beta | \alpha^* \gamma \alpha^* \gamma) S_1$$

$$\Rightarrow (\alpha^* \gamma \alpha^* \beta)^* (\alpha^* \alpha | \alpha^* \gamma \alpha^* \beta)$$

$$| (\alpha^* \gamma \alpha^* \beta)^* (\alpha^* \beta | \alpha^* \gamma \alpha^* \gamma) (\alpha^* \beta \alpha^* \gamma)^* \alpha^* (\gamma | \gamma S_0 | \beta \alpha^* \beta | \beta \alpha^* \beta S_0)$$

$$\Rightarrow (\alpha^* \gamma \alpha^* \beta)^* (\alpha^* \alpha | \alpha^* \gamma \alpha^* \beta) | (\alpha^* \gamma \alpha^* \beta)^* (\alpha^* \beta | \alpha^* \gamma \alpha^* \gamma) (\alpha^* \beta \alpha^* \gamma)^* \alpha^* (\gamma | \gamma S_0 | \beta \alpha^* \beta)$$

$$| (\alpha^* \gamma \alpha^* \beta)^* (\alpha^* \beta | \alpha^* \gamma \alpha^* \gamma) (\alpha^* \beta \alpha^* \gamma)^* \alpha^* \beta \alpha^* \beta S_0$$

$$\Rightarrow ((\alpha^* \gamma \alpha^* \beta)^* (\alpha^* \beta | \alpha^* \gamma \alpha^* \gamma) (\alpha^* \beta \alpha^* \gamma)^* \alpha^* \beta \alpha^* \beta)^*$$

$$(\alpha^* \gamma \alpha^* \beta)^* (\alpha^* \alpha | \alpha^* \gamma \alpha^* \beta) | (\alpha^* \gamma \alpha^* \beta)^* (\alpha^* \beta | \alpha^* \gamma \alpha^* \gamma) (\alpha^* \beta \alpha^* \gamma)^* \alpha^* (\gamma | \gamma S_0 | \beta \alpha^* \beta)$$

第三章作业

【作业3-3】将右线性文法 $G[S]: S \rightarrow xA \mid yB \mid \varepsilon, A \rightarrow yA \mid y, B \rightarrow xB \mid x$, 转换为:

(1) 有限自动机。

(2) 正规式。

【作业3-4】给定右线性文法 $G[S]$, 求其等价的左线性文法:

$$S \rightarrow 0S \mid 1S \mid 1A \mid 0B$$

$$A \rightarrow 1C \mid 1$$

$$B \rightarrow 0C \mid 0$$

$$C \rightarrow 0C \mid 1C \mid 0 \mid 1$$

□ 3.1 对词法分析器的设计

- 3.1.1 词法分析器的任务
- 3.1.2 词法分析器设计需要考虑的问题
- 3.1.3 状态转换图

□ 3.2 有限自动机

- 3.2.1 确定有限自动机
- 3.2.2 非确定有限自动机
- 3.2.3 非确定有限自动机确定化
- 3.2.4 确定有限自动机化简
- 3.2.5 正规式与有限自动机的等价性

□ 3.3 正规文法

- 3.3.1 右线性文法转有限自动机
- 3.3.2 左线性文法转有限自动机

- 3.3.3 有限自动机转右线性文法
- 3.3.4 有限自动机转左线性文法
- 3.3.5 正规式转右线性文法
- 3.3.6 正规式转左线性文法
- 3.3.7 正规文法转正规式
- 3.3.8 三种工具的转换
- 3.3.9 有限自动机转正规式

□ 3.4 词法分析器的实现

- 3.4.1 词法分析器边界
- 3.4.2 单词正规式
- 3.4.3 识别单词的DFA
- 3.4.4 单词识别算法

3.4.1 词法分析器边界

- ❑ **两个连续标识符**，如“x y”，其中x和y之间有空格，应识别为两个标识符，两个标识符连接在一起的错误**属于语法错误**，需要语法分析时才能发现。
- ❑ **标识符和数字连在一起**，如“x 100”，其中x和100之间有空格，应识别为一个标识符和一个数字，它们连接在一起的错误**属于语法错误**，需要语法分析时才能发现。
- ❑ **数字和标识符连在一起**，如“2x”
 - 如果数字和标识符之间**没有空格**，词法分析可以发现这种错误；
 - 如果数字和标识符之间**有空格**，词法分析可能无法发现这种错误；
 - 这种错误**放到语法分析时处理**更加容易。

3.4.1 词法分析器边界

- 数字常数的符号, 如“x---5”, 不再词法分析时确定, 到语法和语义分析再区分符号和加减号。
- 两个符号组合成的新符号, 如“**”、“++”、“--”、“<<”、“==”、“<=”等
 - 可以在词法分析中识别;
 - 也可以在语法和语义分析时识别。
- 三目运算符“?:”, 看做两个单词, 到语法和语义分析再组合。

□ 3.1 对词法分析器的设计

- 3.1.1 词法分析器的任务
- 3.1.2 词法分析器设计需要考虑的问题
- 3.1.3 状态转换图

□ 3.2 有限自动机

- 3.2.1 确定有限自动机
- 3.2.2 非确定有限自动机
- 3.2.3 非确定有限自动机确定化
- 3.2.4 确定有限自动机化简
- 3.2.5 正规式与有限自动机的等价性

□ 3.3 正规文法

- 3.3.1 右线性文法转有限自动机
- 3.3.2 左线性文法转有限自动机

- 3.3.3 有限自动机转右线性文法
- 3.3.4 有限自动机转左线性文法
- 3.3.5 正规式转右线性文法
- 3.3.6 正规式转左线性文法
- 3.3.7 正规文法转正规式
- 3.3.8 三种工具的转换
- 3.3.9 有限自动机转正规式

□ 3.4 词法分析器的实现

- 3.4.1 词法分析器边界
- 3.4.2 单词正规式
- 3.4.3 识别单词的DFA
- 3.4.4 单词识别算法

字符集

类别	正规式	说明
< 字母 >	$a b \dots z A B \dots Z$	大小写字母
< 数字 >	$0 1 2 \dots 9$	一位数字
< 任意 >	Σ	任意字符
< 其他 >		射出弧标记之外的任意字符
<H 数字 >	$0 1 2 \dots 9 A B C D E F a b c d e f$	16 进制数字
< 空白 >	$\backslash t \backslash n \backslash r _$	$_$ 表示空格
< 标首 >	$\langle \text{字母} \rangle _$	标识符首字母
< 标中 >	$\langle \text{字母} \rangle \langle \text{数字} \rangle _$	标识符非首字母

常用单词的正规式

类别	正规式
< 标识符 >	< 标首 >< 标中 >*
< 整数 >	< 数字 >+
< H 整数 >	0x< H 数字 >+
< 实数 >	< 数字 >+ . < 数字 >+
< 字符 >	'< 任意 >'
< 字符串 >	"< 任意 >*" <div data-bbox="1062 471 1371 542"> <p>□ 转义字符</p> <ul style="list-style-type: none"> ➤ \r: 回车符 ➤ \n: 换行符 ➤ \t: 跳格符 ➤ \': 单引号 ➤ \": 双引号 ➤ \\: 反斜杠 ➤ \<整数>: 10进制ASCII码 ➤ \<H整数>+: 16进制ASCII码 </div>
< 单行注释 >	//< 任意 >*(\r \n)
< 多行注释 >	/*< 任意 >**/

其他单词

□ 其他约定

- 界符包括：逗号、分号、小括号、花括号。
- 关键字包括：byte、ubyte、char、bool、short、ushort、int、uint、float、double、var、true、false、if、else、while、for、switch、case、default、goto、foreach、void、main。
- 运算符包括：=、+、-、*、/、\%、**、<、<=、>、>=、==、!=、!、&&、||、&、|、~、^、<<、>>、?、:。
- 界符、关键字和运算符都采用一字（符）一类。关键字可以先按标识符识别，再去查表得到；也可以直接用DFA识别，我们选择后者。

□ 3.1 对词法分析器的设计

- 3.1.1 词法分析器的任务
- 3.1.2 词法分析器设计需要考虑的问题
- 3.1.3 状态转换图

□ 3.2 有限自动机

- 3.2.1 确定有限自动机
- 3.2.2 非确定有限自动机
- 3.2.3 非确定有限自动机确定化
- 3.2.4 确定有限自动机化简
- 3.2.5 正规式与有限自动机的等价性

□ 3.3 正规文法

- 3.3.1 右线性文法转有限自动机
- 3.3.2 左线性文法转有限自动机

- 3.3.3 有限自动机转右线性文法
- 3.3.4 有限自动机转左线性文法
- 3.3.5 正规式转右线性文法
- 3.3.6 正规式转左线性文法
- 3.3.7 正规文法转正规式
- 3.3.8 三种工具的转换
- 3.3.9 有限自动机转正规式

□ 3.4 词法分析器的实现

- 3.4.1 词法分析器边界
- 3.4.2 单词正规式
- 3.4.3 识别单词的DFA
- 3.4.4 单词识别算法

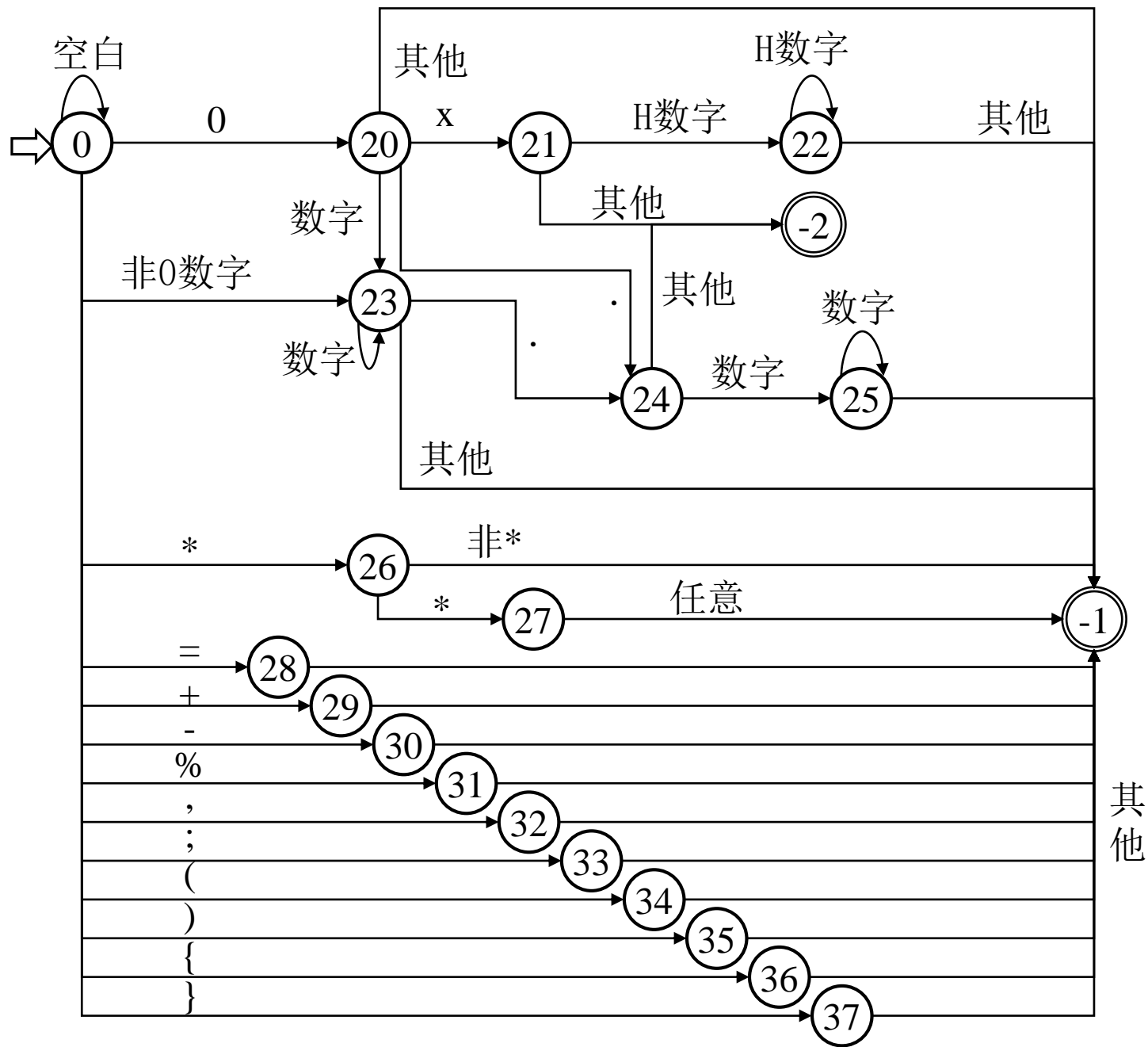
3.4.3 识别单词的DFA

□ 只展示一个单词子集的设计

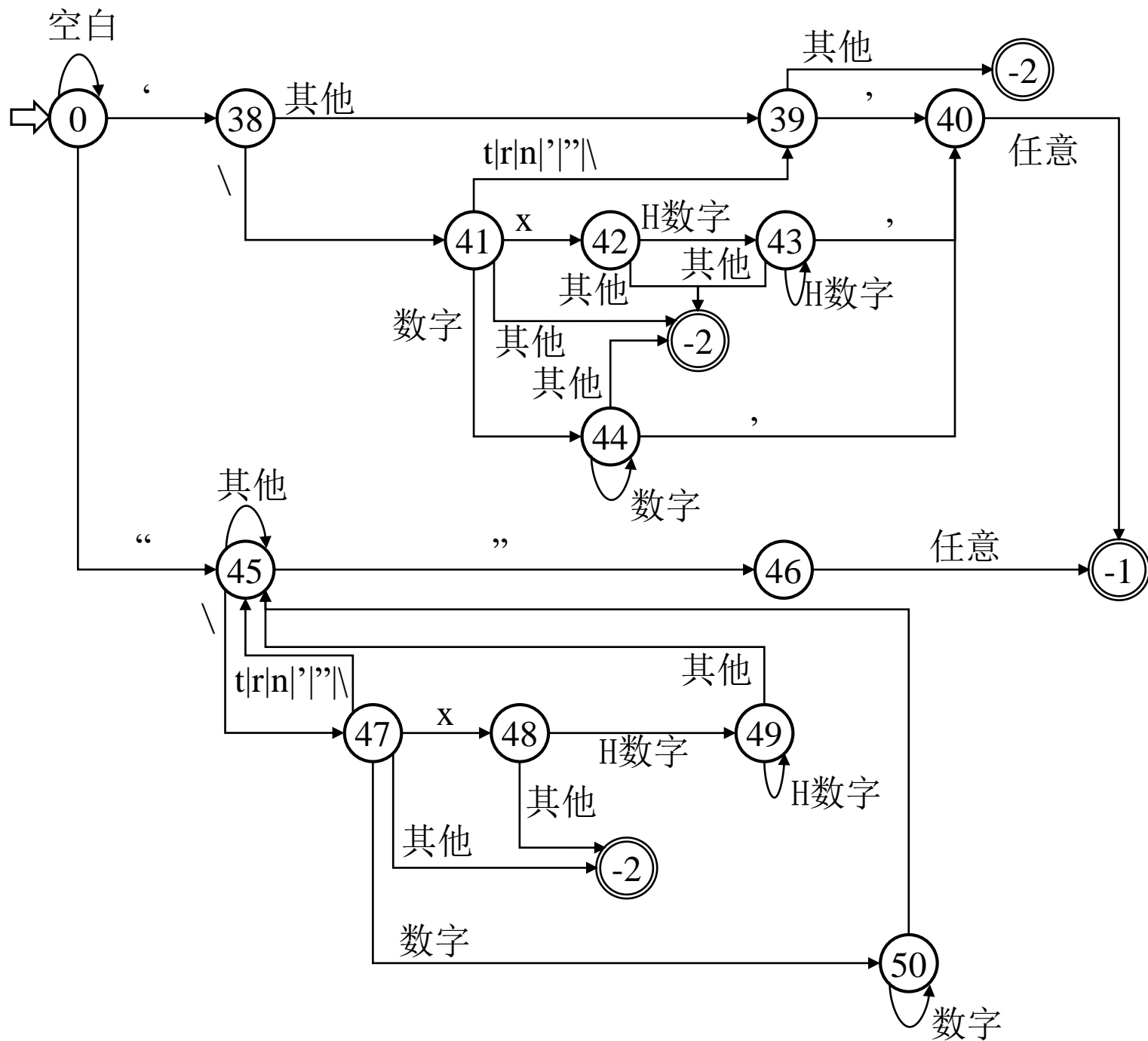
- 包括了所有标识符、常量、注释、界符。
- 关键字选择int、float、if、else、while等少数几个，其他关键字类似处理。
- 运算符包括=、+、-、*、/、%、**，其中展示了*和**的区分方法。

□ DFA状态

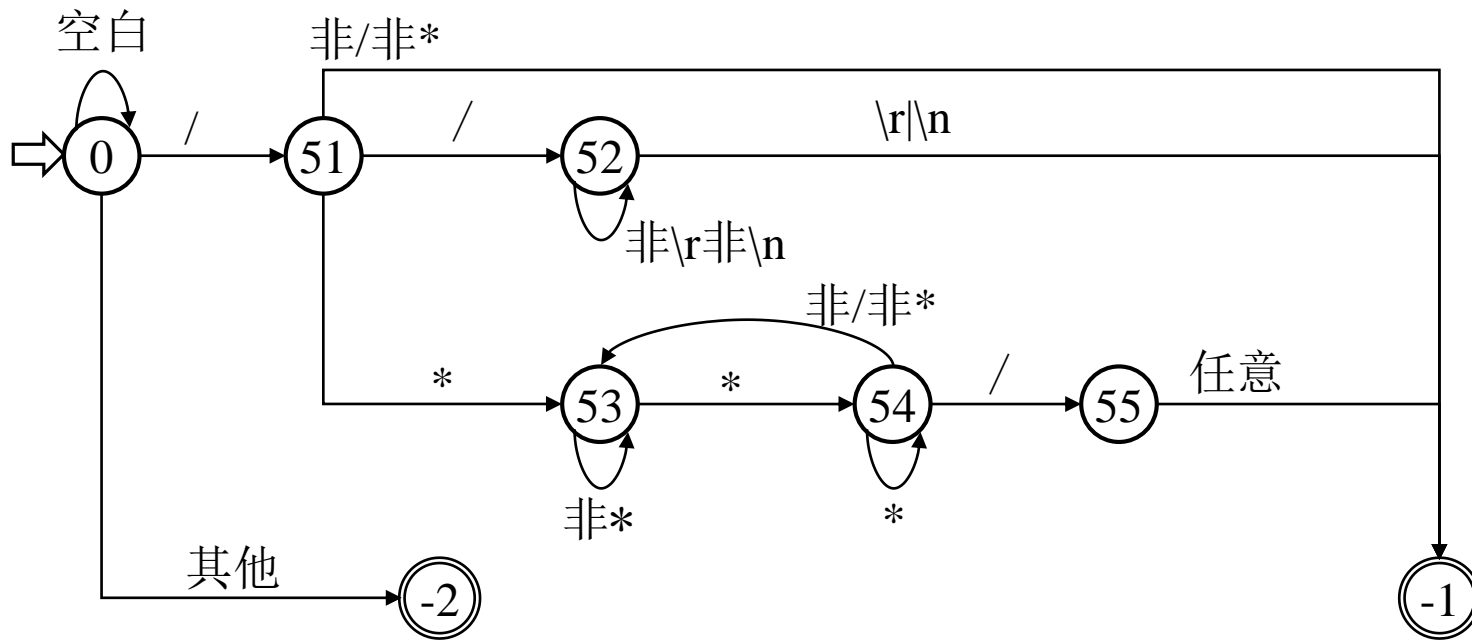
- 状态0：是唯一的初态；
- 状态-1：是两个终态之一，为识别一个单词成功的状态；
- 状态-2：是两个终态之一，为出错状态；
- 状态-1的前驱状态：表示单词类别，可以用作类别编码；
- 其他状态：正常的内部状态。



识别单词的
DFA之2/4



识别单词的
DFA之3/4



识别单词的
DFA之4/4

□ 3.1 对词法分析器的设计

- 3.1.1 词法分析器的任务
- 3.1.2 词法分析器设计需要考虑的问题
- 3.1.3 状态转换图

□ 3.2 有限自动机

- 3.2.1 确定有限自动机
- 3.2.2 非确定有限自动机
- 3.2.3 非确定有限自动机确定化
- 3.2.4 确定有限自动机化简
- 3.2.5 正规式与有限自动机的等价性

□ 3.3 正规文法

- 3.3.1 右线性文法转有限自动机
- 3.3.2 左线性文法转有限自动机

- 3.3.3 有限自动机转右线性文法
- 3.3.4 有限自动机转左线性文法
- 3.3.5 正规式转右线性文法
- 3.3.6 正规式转左线性文法
- 3.3.7 正规文法转正规式
- 3.3.8 三种工具的转换
- 3.3.9 有限自动机转正规式

□ 3.4 词法分析器的实现

- 3.4.1 词法分析器边界
- 3.4.2 单词正规式
- 3.4.3 识别单词的DFA
- 3.4.4 单词识别算法

3.4.4 单词识别算法

算法 3.11 单词识别

输入: 文件名称

输出: 单词序列 *tokens*, 其每个结点为单词类别和单词值的二元组

```
1 根据文件名称将源程序读入字符数组 buffer, 并在最后加一个空白符号;  
2 当前状态 state = 0, 前一个状态 preState = -1;  
3 当前符号指针 pCur = 0, 开始位置指针 pStart = -1;  
4 while pCur < buffer.length do  
5     pStart = pCur;  
6     while state ≠ -1 do  
7         preState = state;  
8         state = getNextState(preState, buffer[pCur]);  
9         if state = -2 then 报错退出;  
10        判断该状态是否需要特殊处理;  
11        pCur++;  
12    end  
13    // 目前识别出了一个单词  
14    if PreState ≠ 52 ∧ PreState ≠ 55 then    // 不是注释  
15        | tokens.add(preState, buffer[pStart : pCur - 1]);  
16    end  
17    pCur - -;  
18    state = 0;  
19 end
```

第 3 章 词法分析 内容小结

- ❑ 词法分析器可以作为独立的一遍，也可以作为一个独立子程序由语法分析器调用。
- ❑ DFA 由当前状态和当前符号唯一确定下一个状态，容易编程实现，用于单词识别。
- ❑ NFA 容易设计，但不适合编程实现，可以通过确定化算法转换为 DFA。
- ❑ 正规文法、正规式和 FA 是等价的，可以互相转换。



山东大学
SHANDONG UNIVERSITY

第三章 词法分析

The End

谢谢

授 课 教 师 : 郑艳伟
手 机 : 18614002860 (微信同号)
邮 箱 : zhengyw@sdu.edu.cn