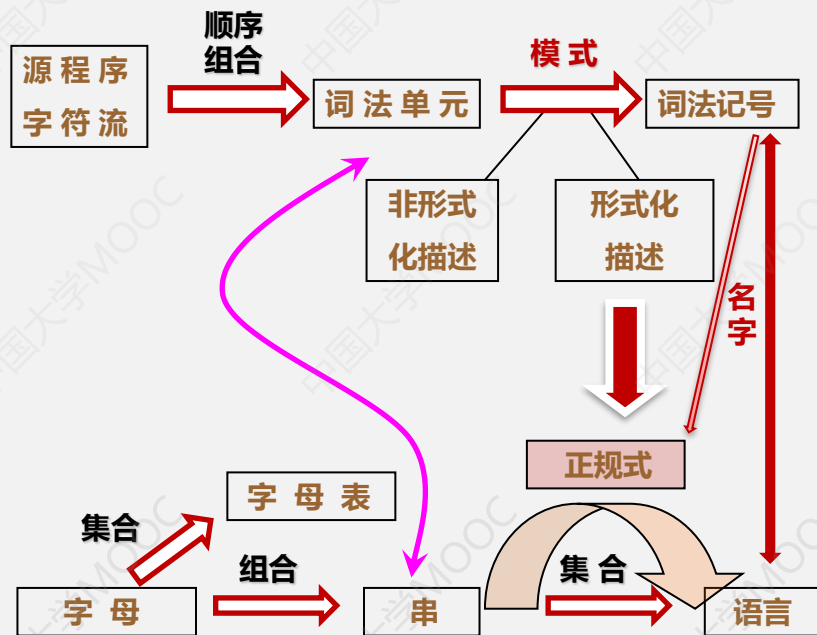




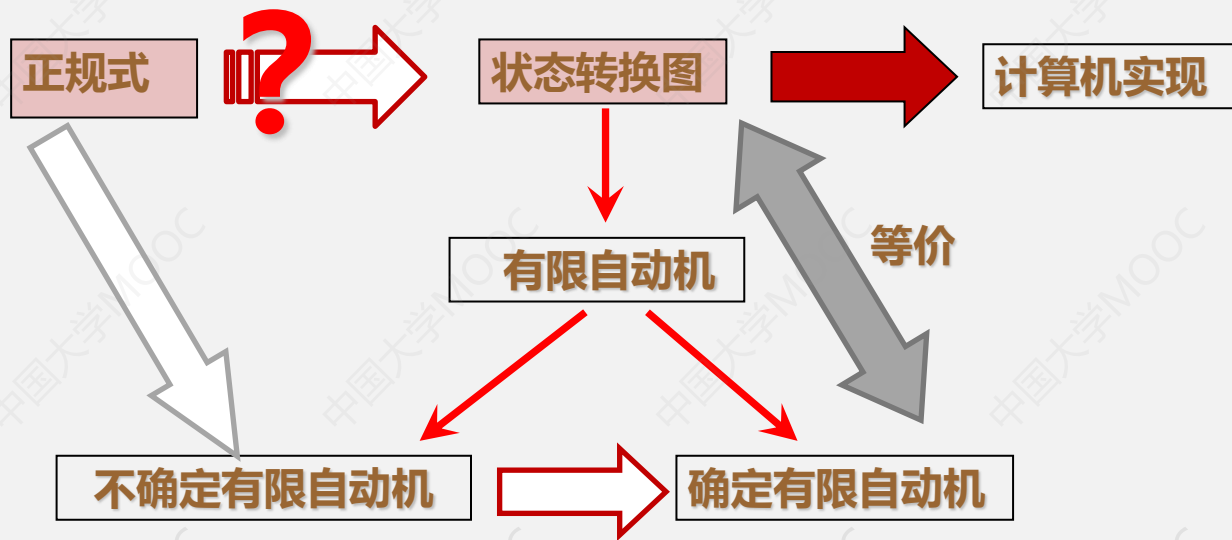
编译技术 词法分析

大连理工大学软件学院

词法分析器工作原理:



有限自动机



本讲 纲要

01

有限自动机定义

02

DFA构建方法

03

子集构造法

04

DFA化简



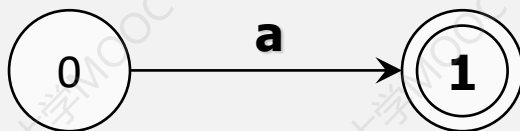
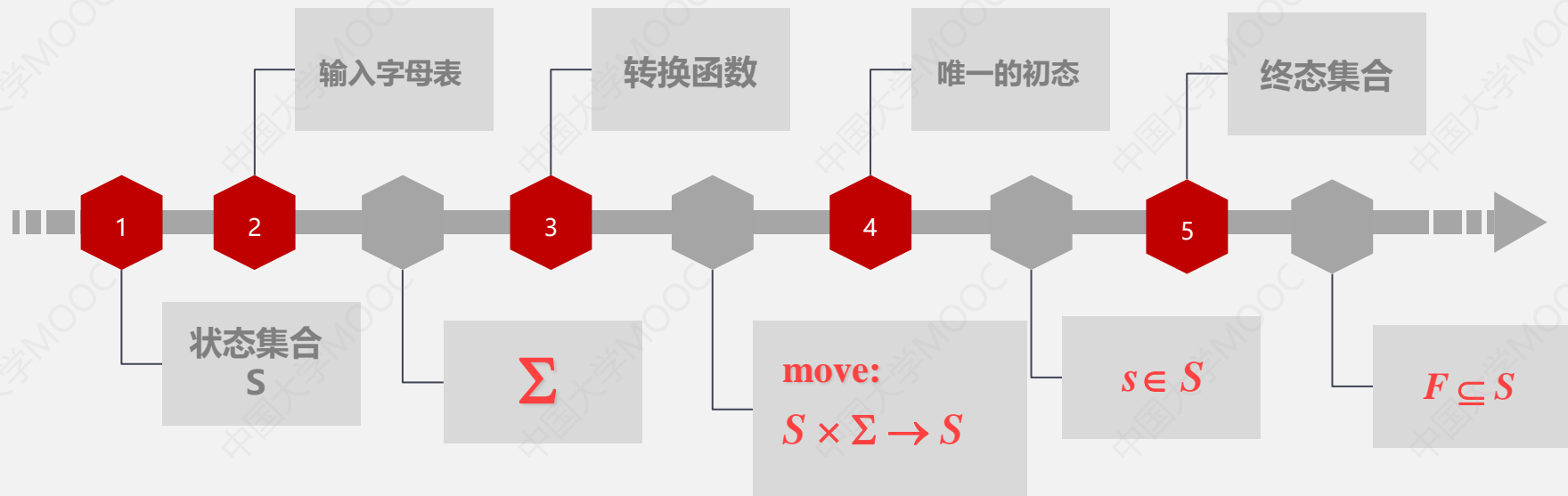
01

确定的有限自动机 (简称DFA)

02

数学形式定义

DFA是这样一个数学模型，包括





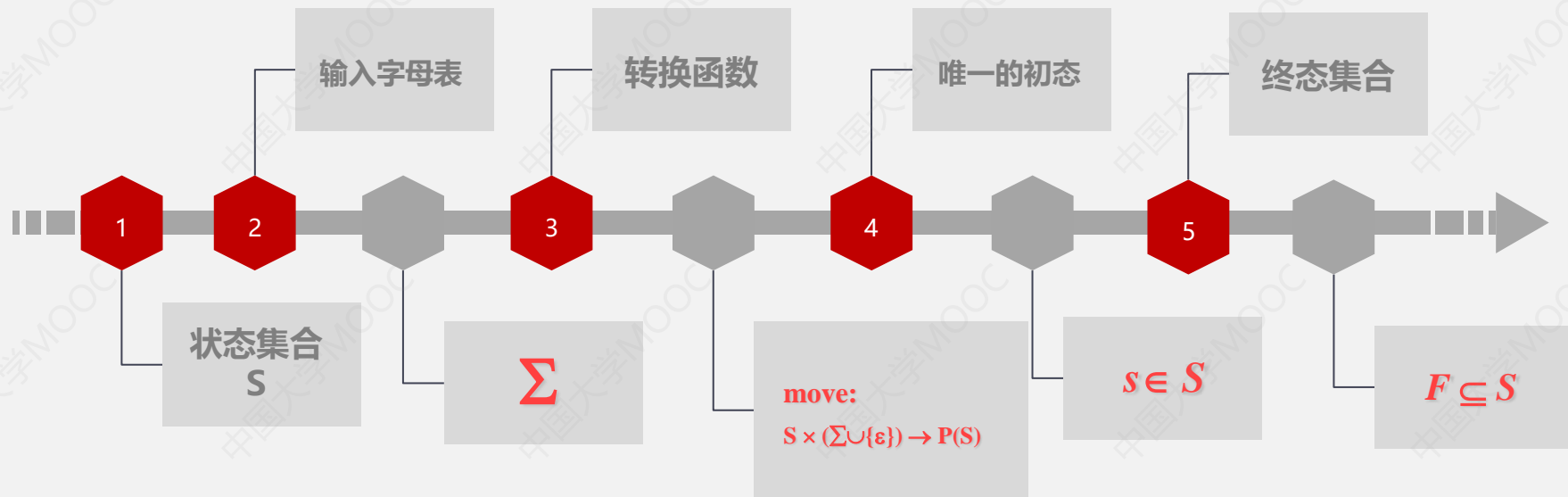
01

不确定的有限自动机 (简称NFA)

02

数学形式定义

NFA是这样—一个数学模型，包括



缺点

1. 输入字符包括 ϵ

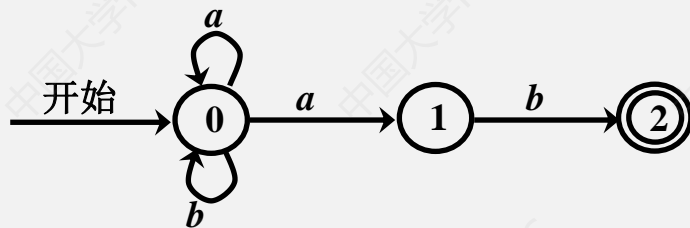
缺点

2. 一个状态对于某个字符，可能有多条输出边

NFA
与
DFA

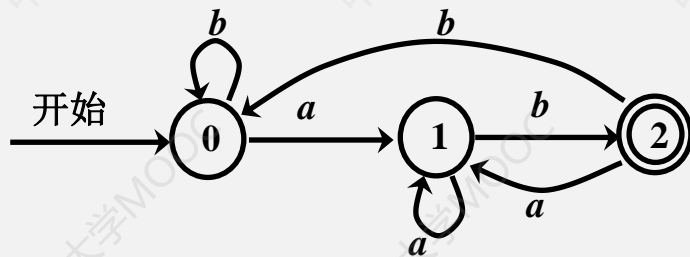
01

$(a|b)^*ab$ 的NFA



02

$(a|b)^*ab$ 的DFA



DFA与NFA的区别

01

NFA中允许 ϵ 转换边，而DFA中不允许

02

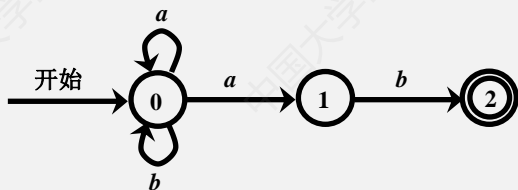
NFA中 $\text{move}(s,a)$ 可能是一个多元集合，
而DFA中 $\text{move}(s,a)$ 最多有一个元素

状态转移表 $(a|b)^*ab$

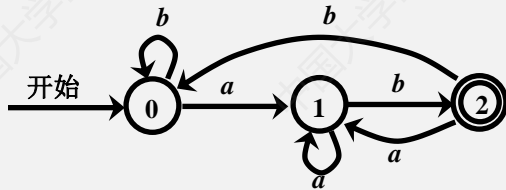


状态迁移动作，从开始状态到目标状态

	输入符号	
	a	b
0	$\{0, 1\}$	$\{0\}$
1	ϕ	$\{2\}$
2	ϕ	ϕ



	输入符号	
	a	b
0	$\{1\}$	$\{0\}$
1	$\{1\}$	$\{2\}$
2	$\{1\}$	$\{0\}$



优点：快速定位



字母表过大或大部分转换状态为空集时浪费空间

本讲 纲要

01

有限自动机定义

02

DFA构建方法

03

子集构造法

04

DFA化简



在机器上实现字符串识别过程

基于DFA?还是基于NFA?

NFA更贴近于人们对正规式的认识

DFA因为每次状态转换都是确定性的，即从当前状态 s 与当前字符 a ，可以转换到唯一的目标状态 s'

正规式 \Rightarrow NFA \Rightarrow DFA

途径3

先构造NFA，再将NFA转换为DFA

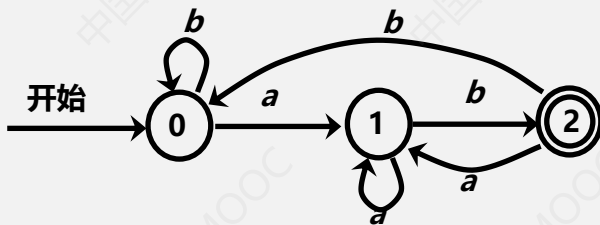
三大步骤：

- **NFA构建**
- NFA \rightarrow DFA的转化（子集构造法）
- DFA化简

正规式 \Rightarrow DFA?

从正规式 $(a|b)^*ab$ 的自动机构造讲起

状态0, 1, 2的含义并不太容易说明白



从正规式到NFA

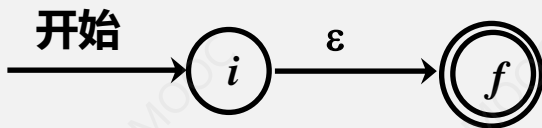
按照正规式的构建规则，逐步从简单到复杂地讨论从正规式构建NFA的过程



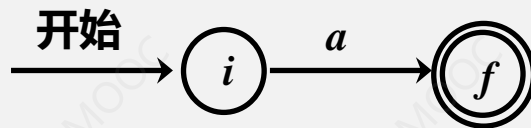
2.4 从正规式到有限自动机



首先构造识别 ε 和字母表中一个符号的NFA



识别正规式 ε 的NFA

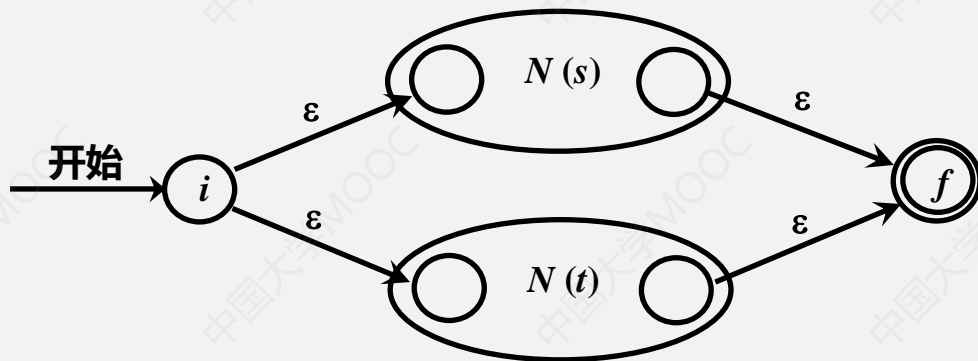


识别正规式 a 的NFA

2.4 从正规式到有限自动机



构造识别主算符为**选择**的正规式的NFA

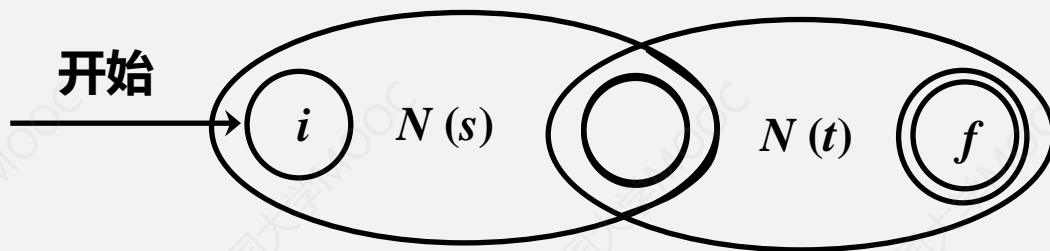


识别正规式 $s \mid t$ 的NFA

2.4 从正规式到有限自动机



构造识别主算符为**连接**的正规式的NFA

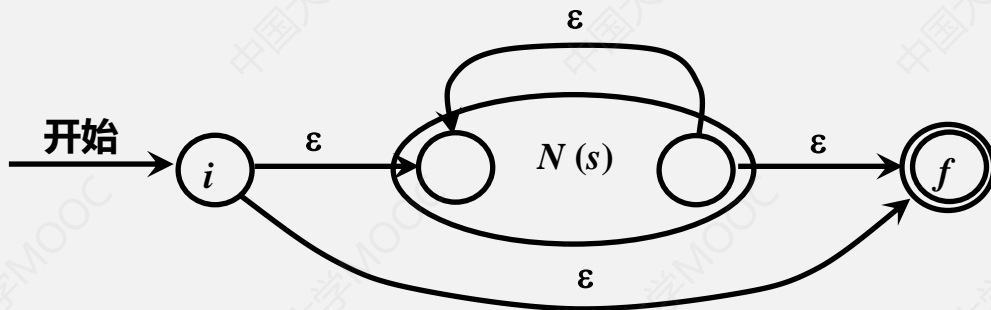


识别正规式 st 的NFA

2.4 从正规式到有限自动机



构造识别主算符为**闭包**的正规式的NFA



识别正规式 s^* 的NFA

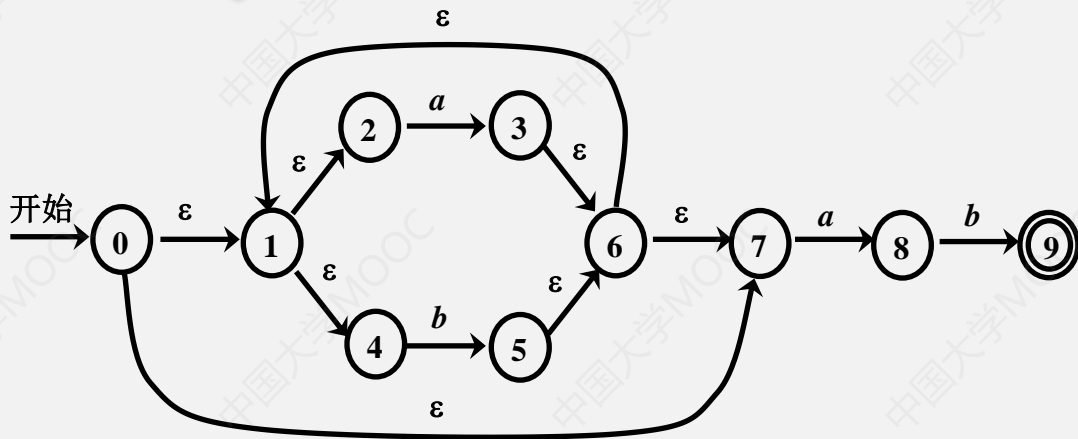


对于**加括号**的正规式(s), 使用 $N(s)$ 本身作为它的NFA。

NFA 构建实例

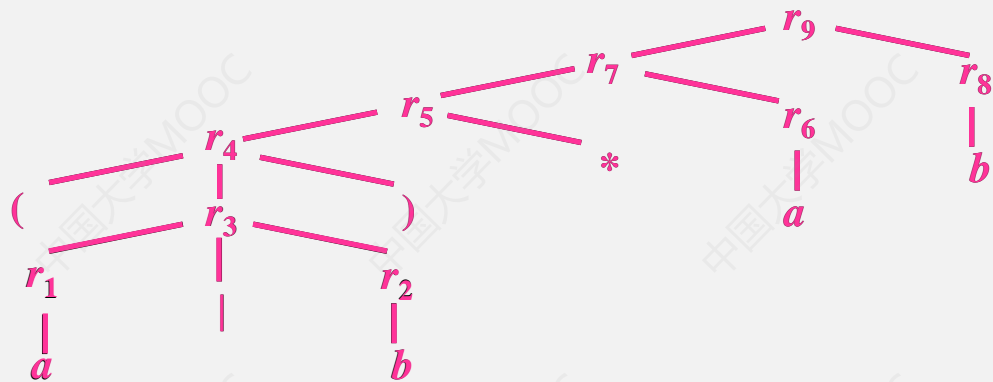


$(a|b)^*ab \Rightarrow \text{NFA}$

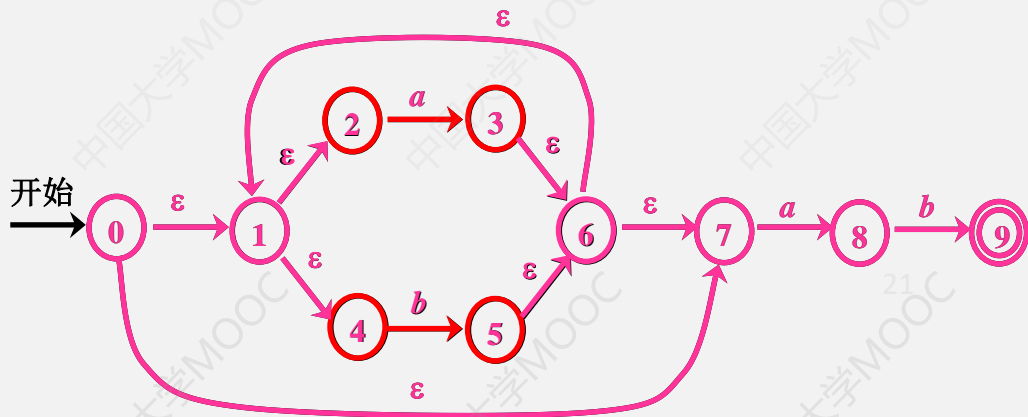


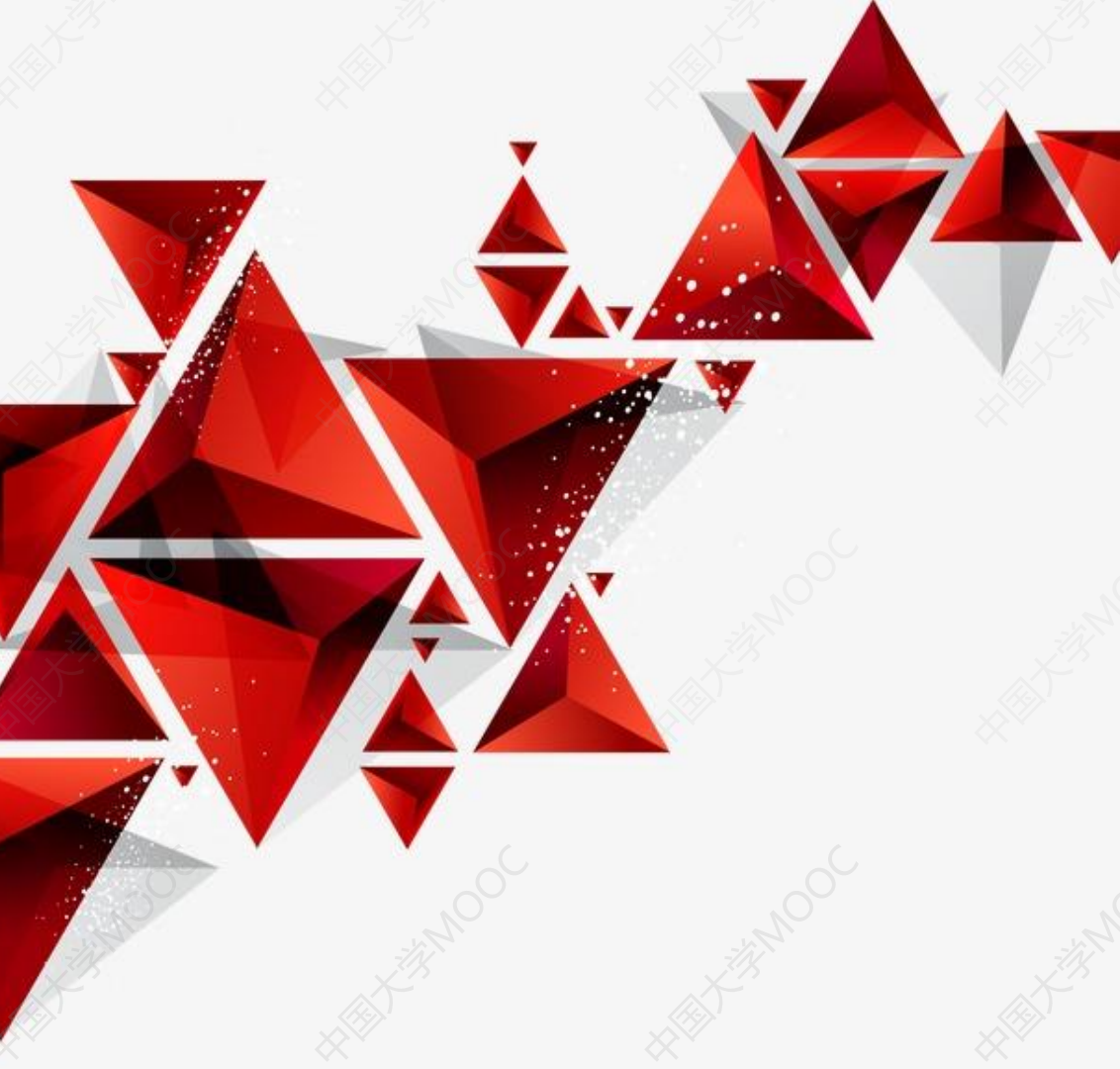
下面来看一看正规式 $(a|b)^*ab$ 的分解动作

2.4 从正规式到有限自动机



$(a|b)^*ab$ 的分解





编译技术 词法分析

大连理工大学软件学院