



中国科学技术大学
University of Science and Technology of China



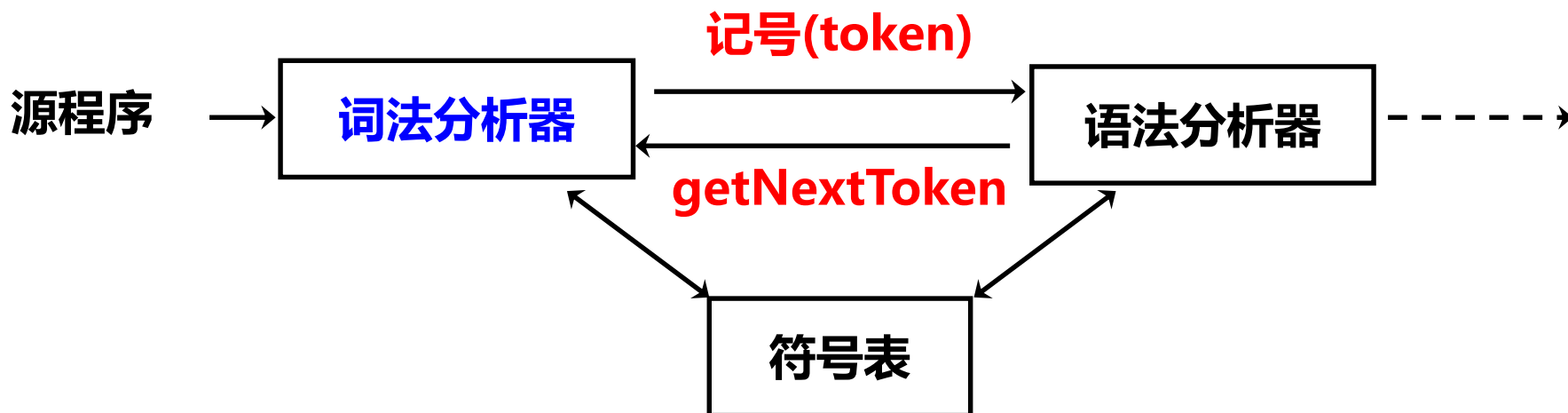
《编译原理与技术》

词法分析 II

计算机科学与技术学院

李诚

13/09/2018



□ 词法分析器的自动生成

❖ 正则表达式 → NFA → DFA → 化简的DFA

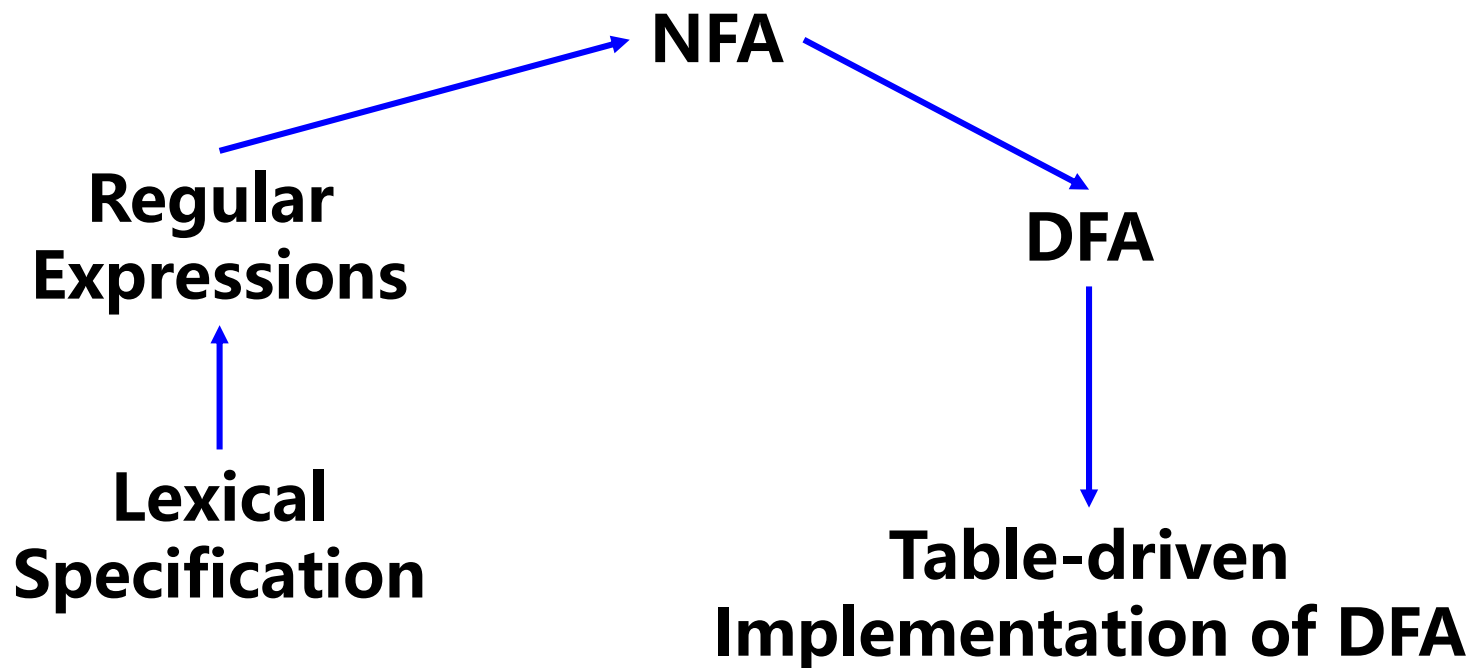
□ 词法分析器的生成器

❖ Lex: flex、jflex

Fast lexical analyzer
generator



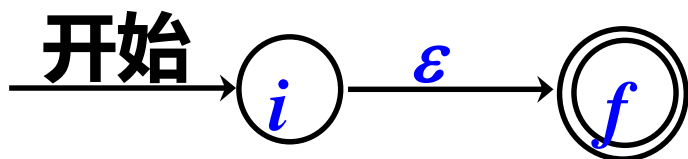
- ❑ 正则表达式 = Specification
- ❑ 有限自动机 = Implementation



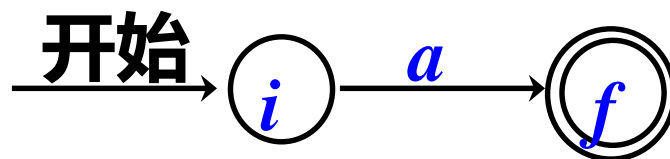
- ❑ 正则表达式 = Specification
- ❑ 有限自动机 = Implementation

- ❑ 二者之间的转换：
 - ❖ 用语法制导的算法，它用正则表达式的语法结构来指导有限自动机的构造过程

- 首先构造识别 ϵ 和字母表中一个符号 a 的 NFA
 - ❖ 重要特点：仅一个接受状态，它没有向外的转换



识别正则表达式 ϵ 的
NFA



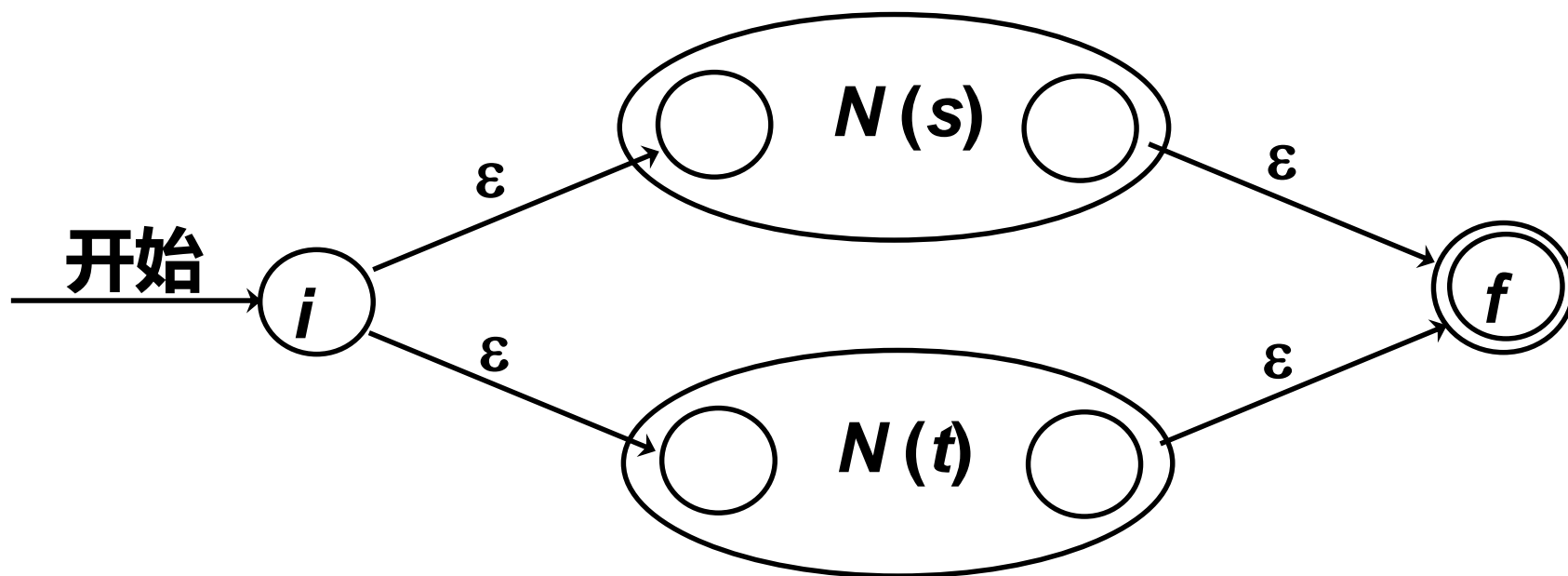
识别正则表达式 a 的
NFA

- 对于加括号的正则表达式 (s) ，使用 $N(s)$ 本身作为它的 NFA



□ 构造识别主算符为选择的正则表达式的NFA

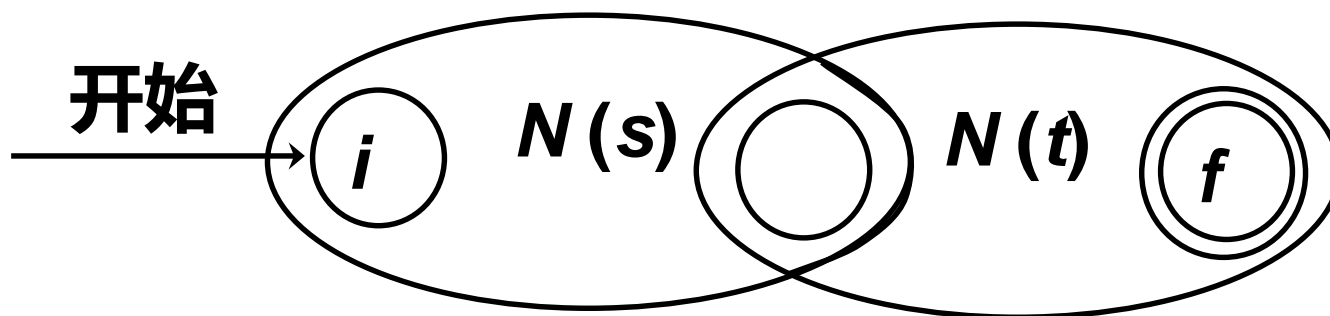
❖ 重要特点：仅一个接受状态，它没有向外的转换



识别正则表达式 $s | t$ 的NFA

□ 构造识别主算符为连接的正则表达式的NFA

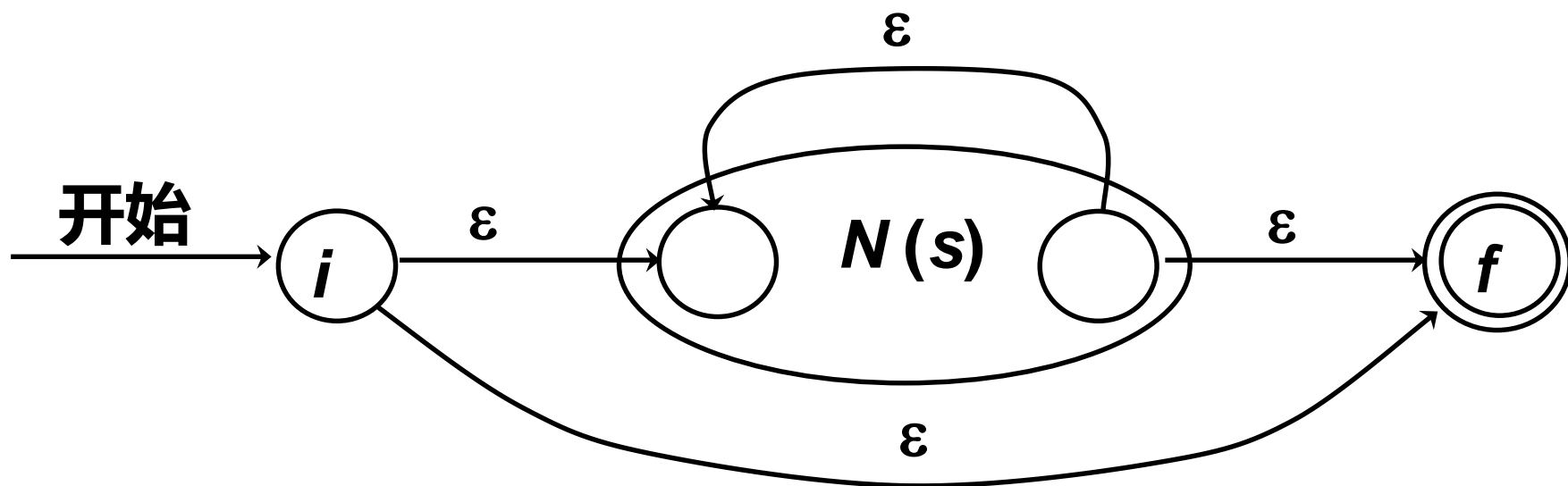
❖ 重要特点：仅一个接受状态，它没有向外的转换



识别正则表达式 st 的NFA

□ 构造识别主算符为闭包的正则表达式的NFA

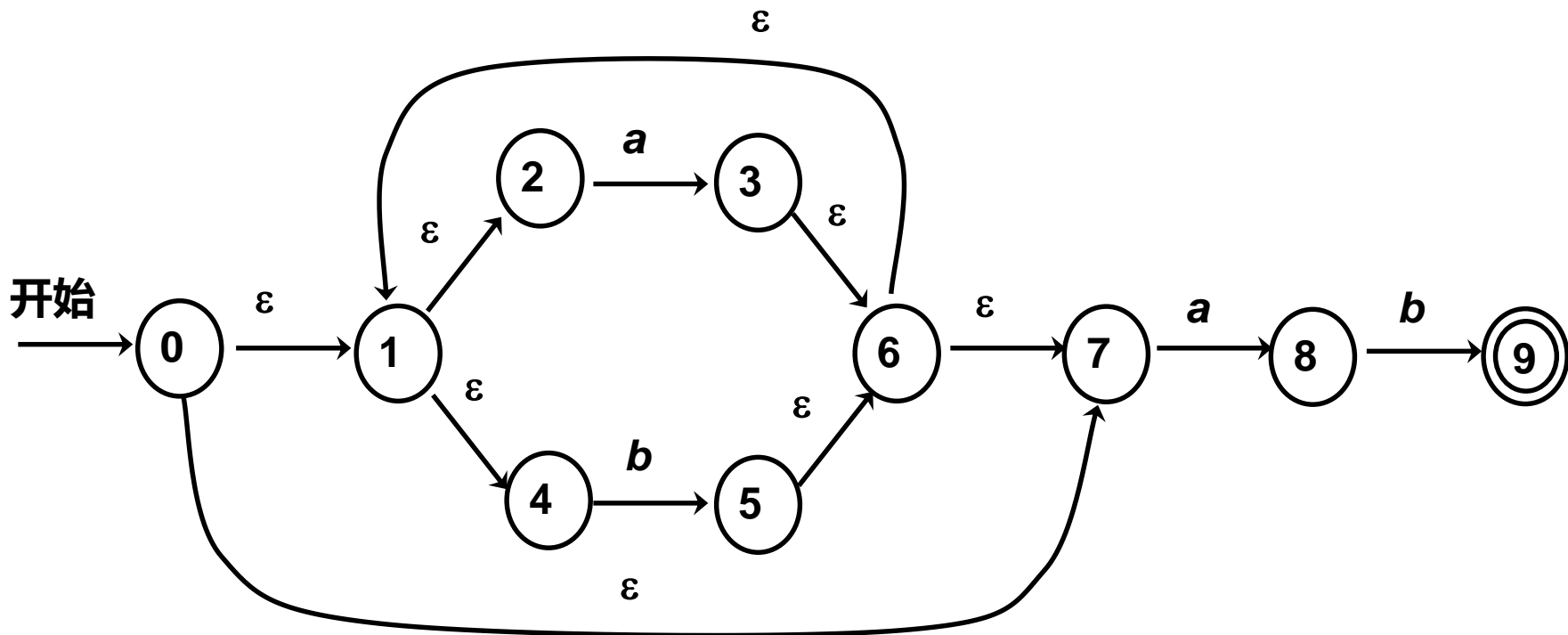
❖ 重要特点：仅一个接受状态，它没有向外的转换



识别正则表达式 s^* 的NFA

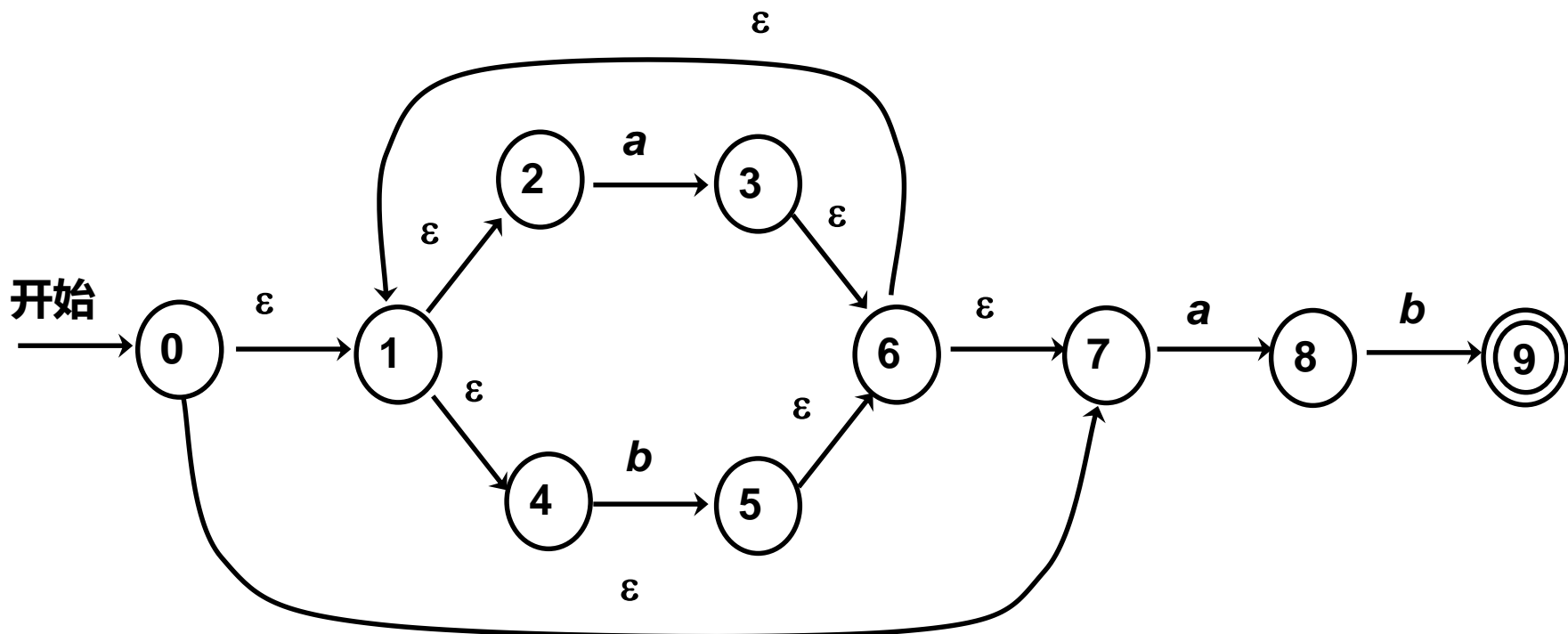
□ 由本方法产生的NFA具有下列性质：

- ❖ $N(r)$ 的状态数最多是 r 中符号和算符总数的两倍
- ❖ $N(r)$ 只有一个接受状态，接受状态没有向外的转换

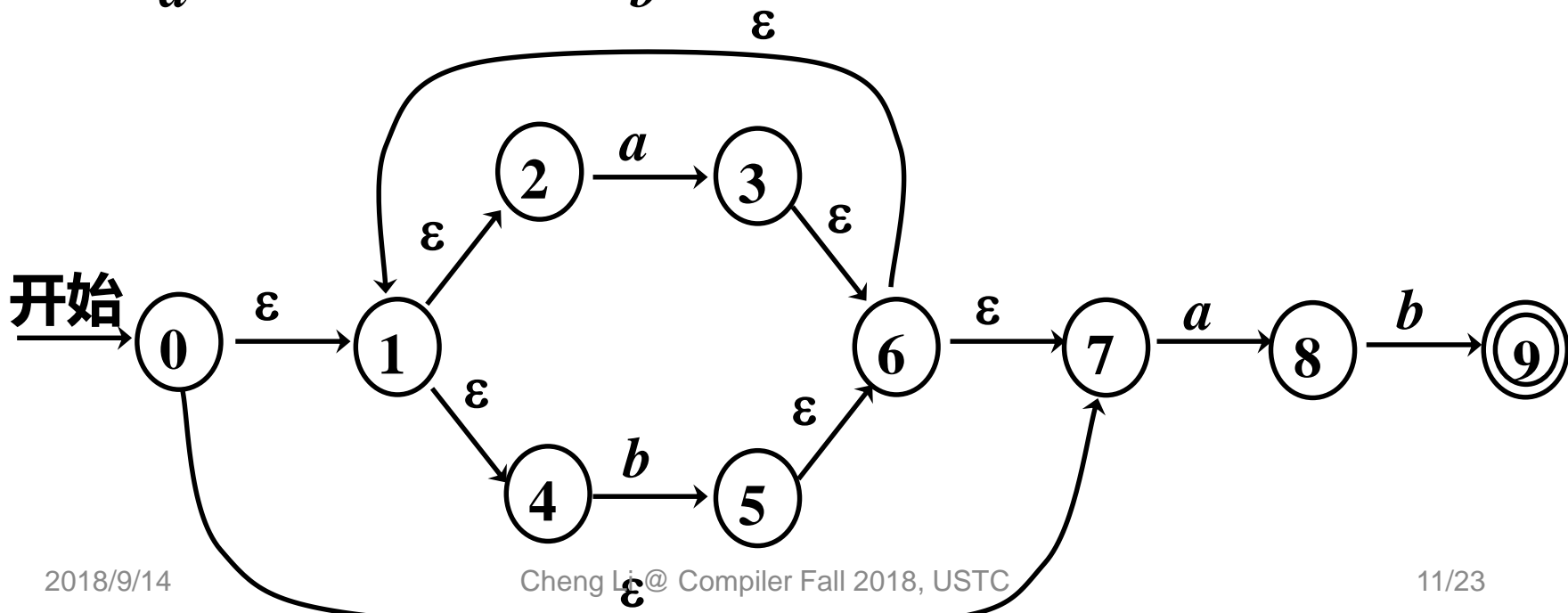
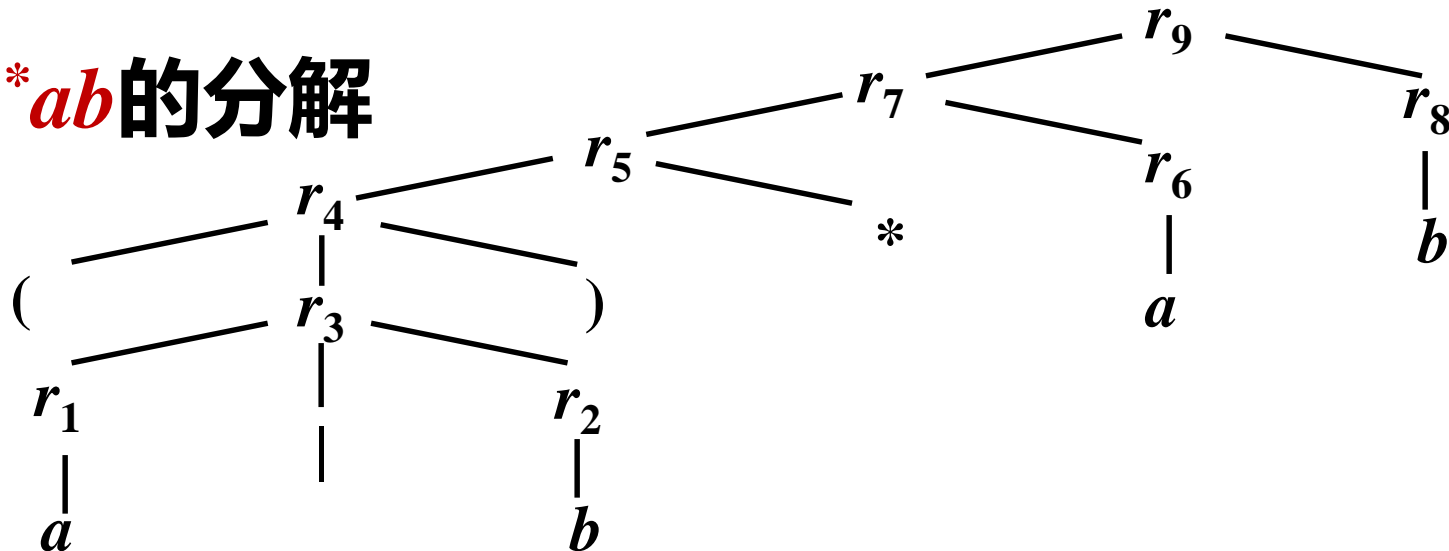


□ 由本方法产生的NFA具有下列性质：

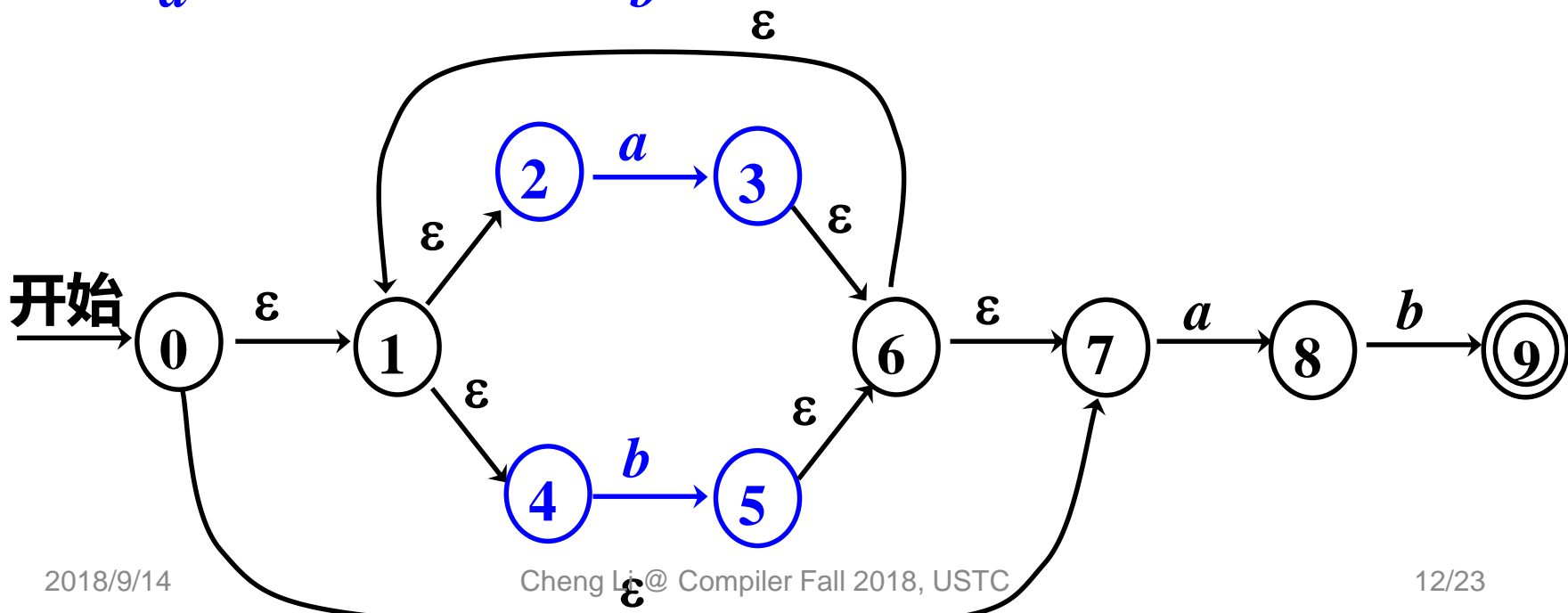
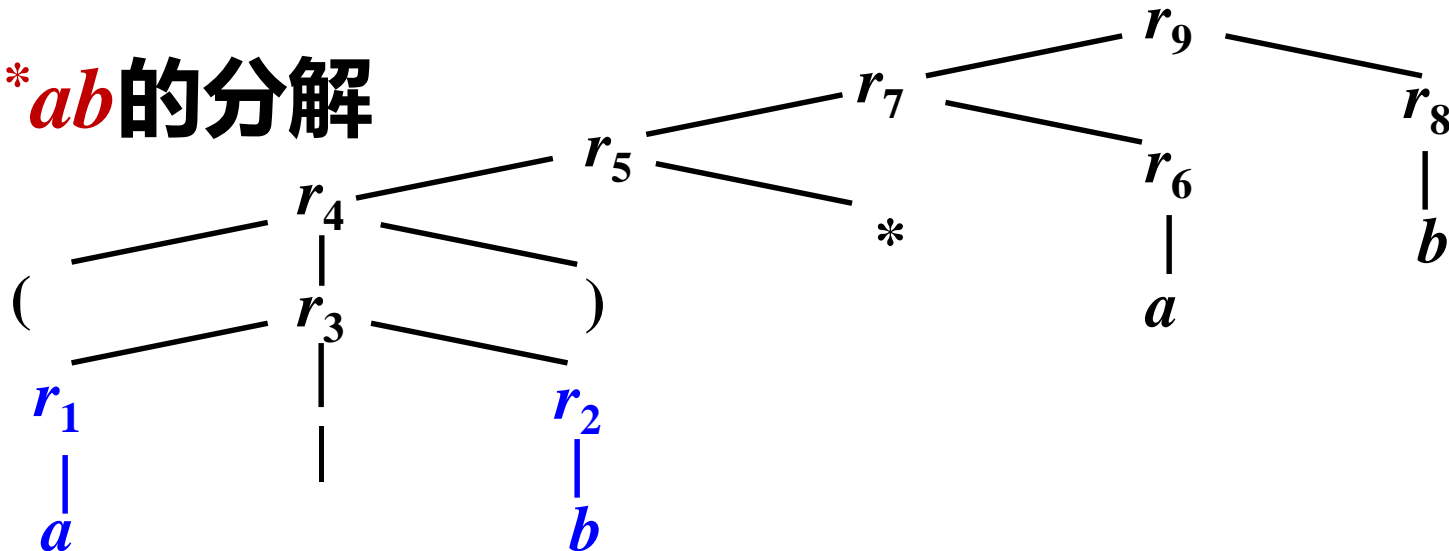
- ❖ $N(r)$ 的每个状态有一个用 Σ 的符号标记指向其它结点的转换，或者最多两个指向其它结点的 ϵ 转换



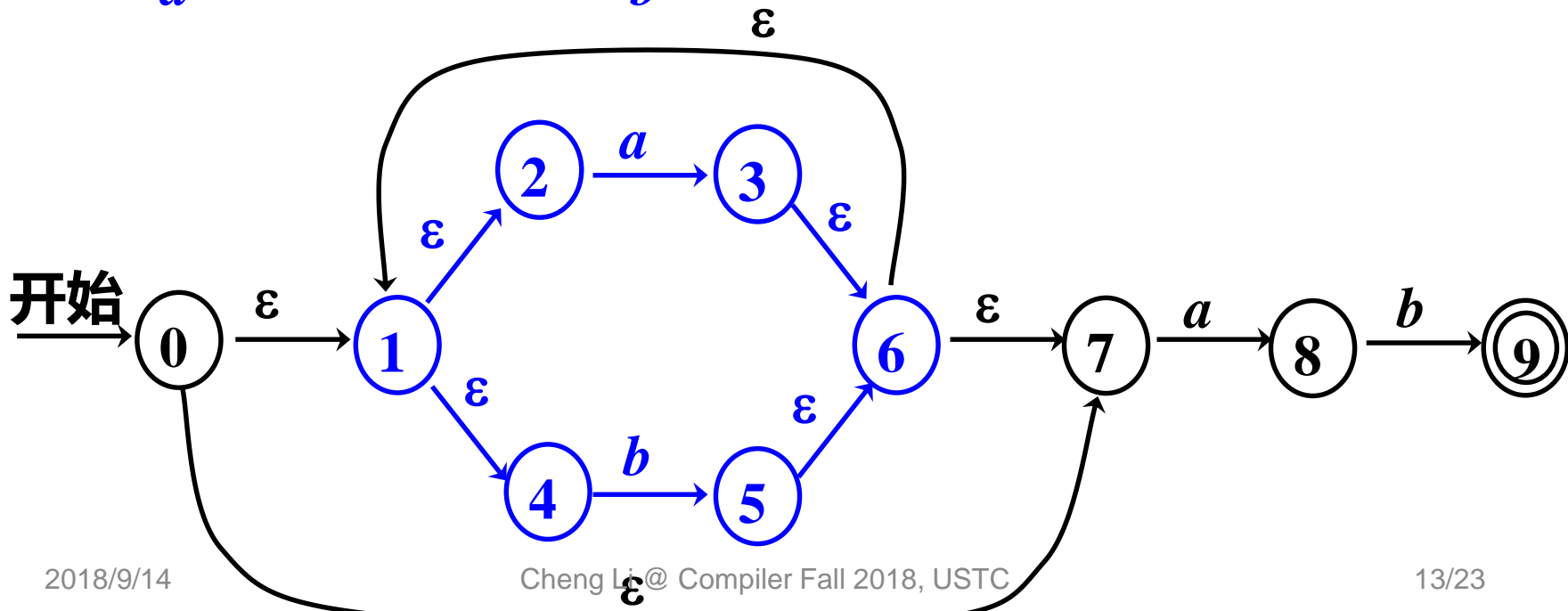
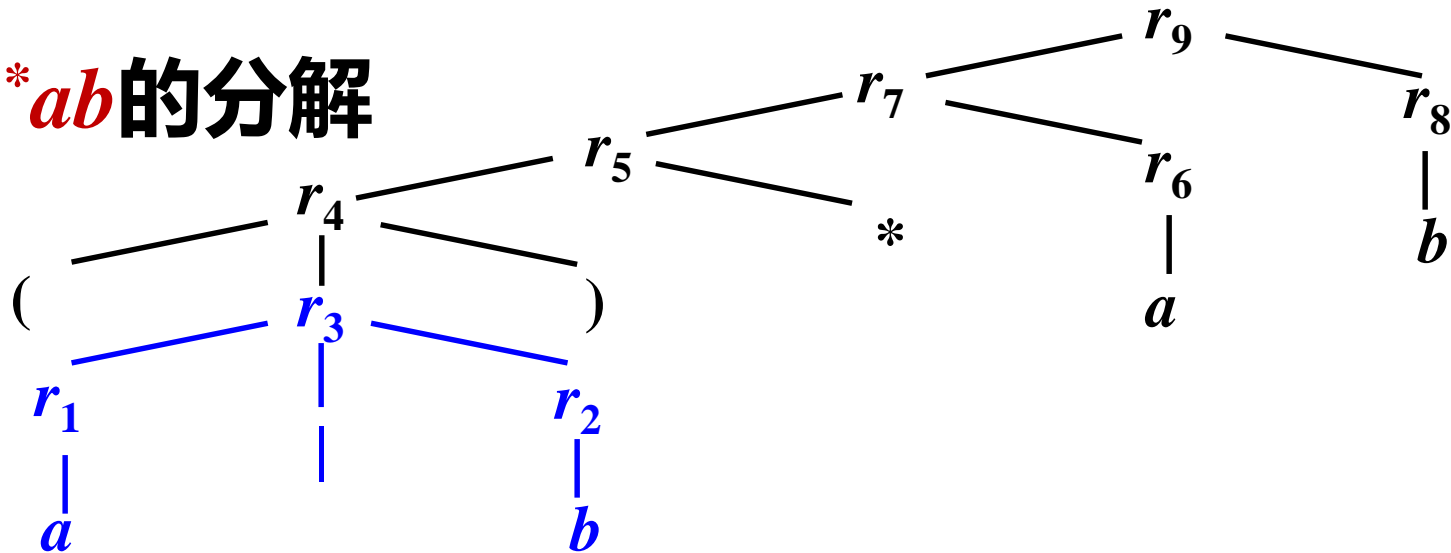
$(a|b)^*ab$ 的分解



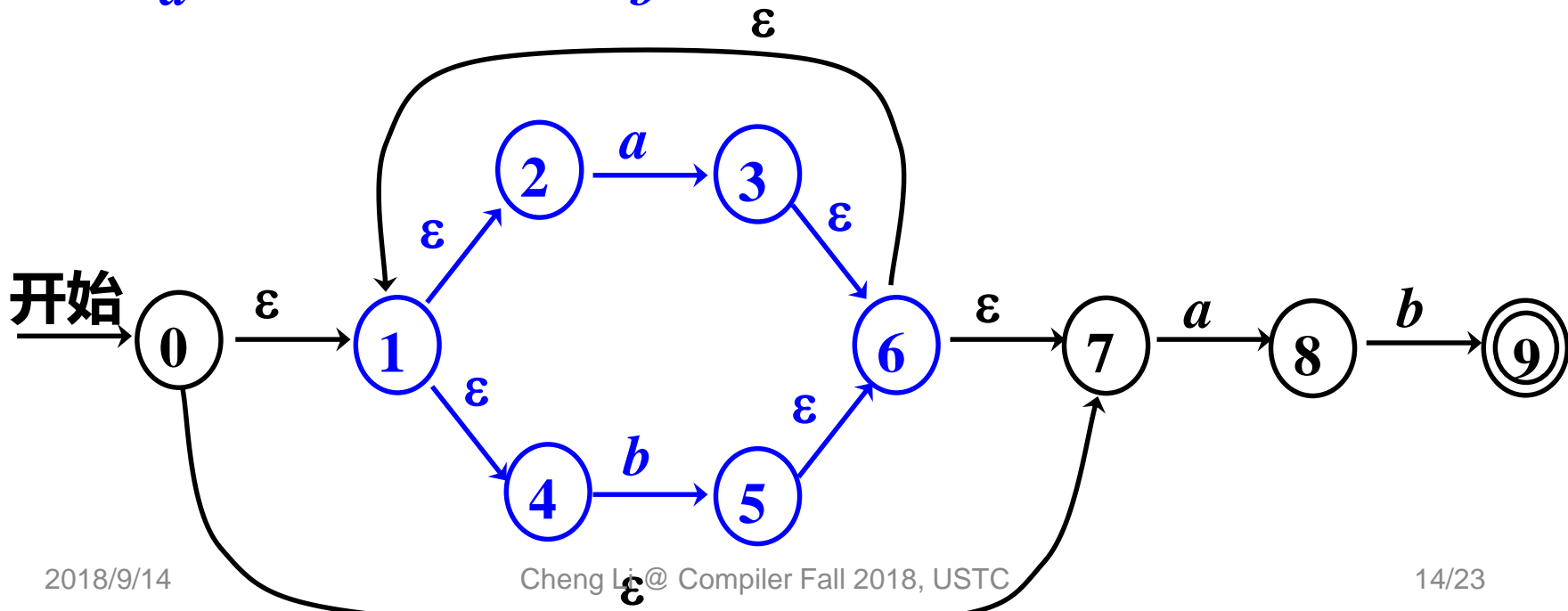
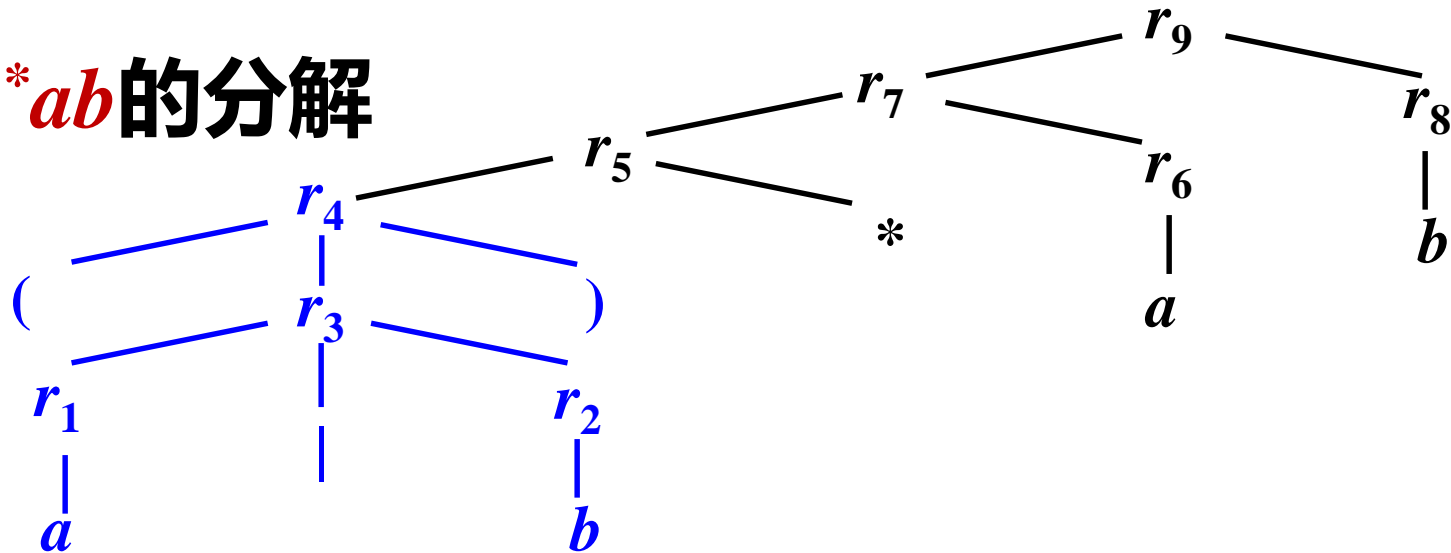
$(a|b)^*ab$ 的分解



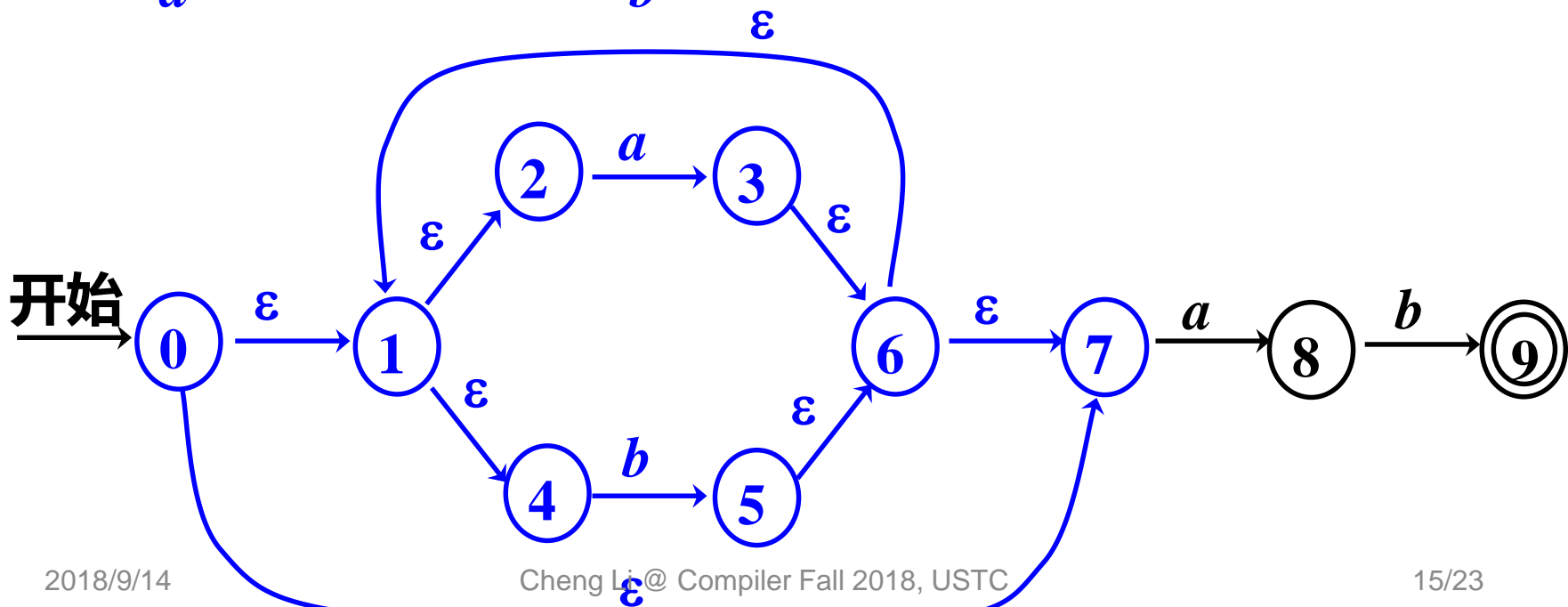
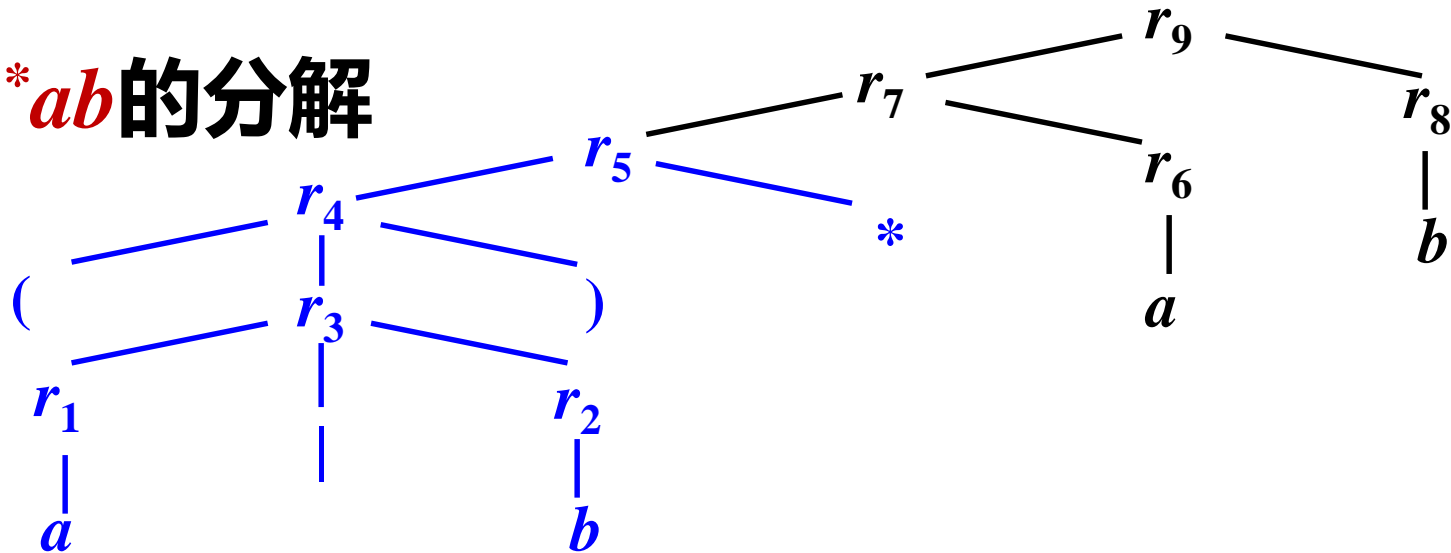
$(a|b)^*ab$ 的分解



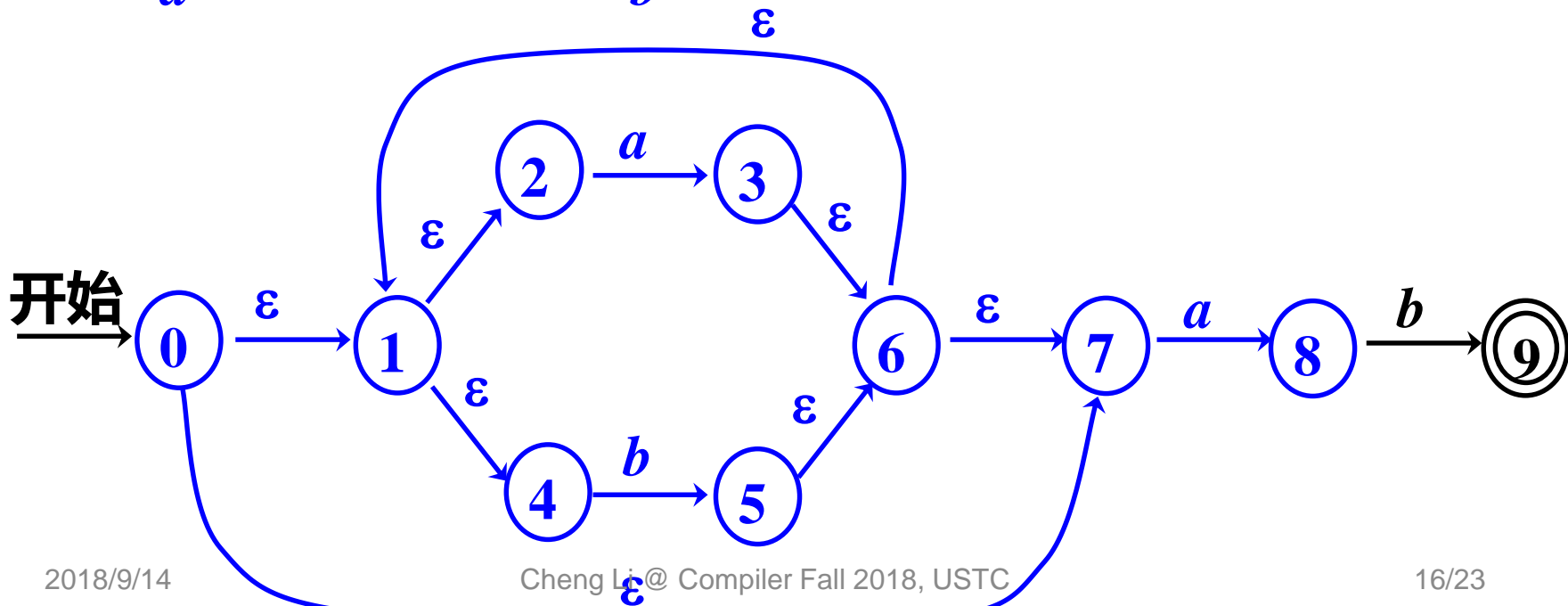
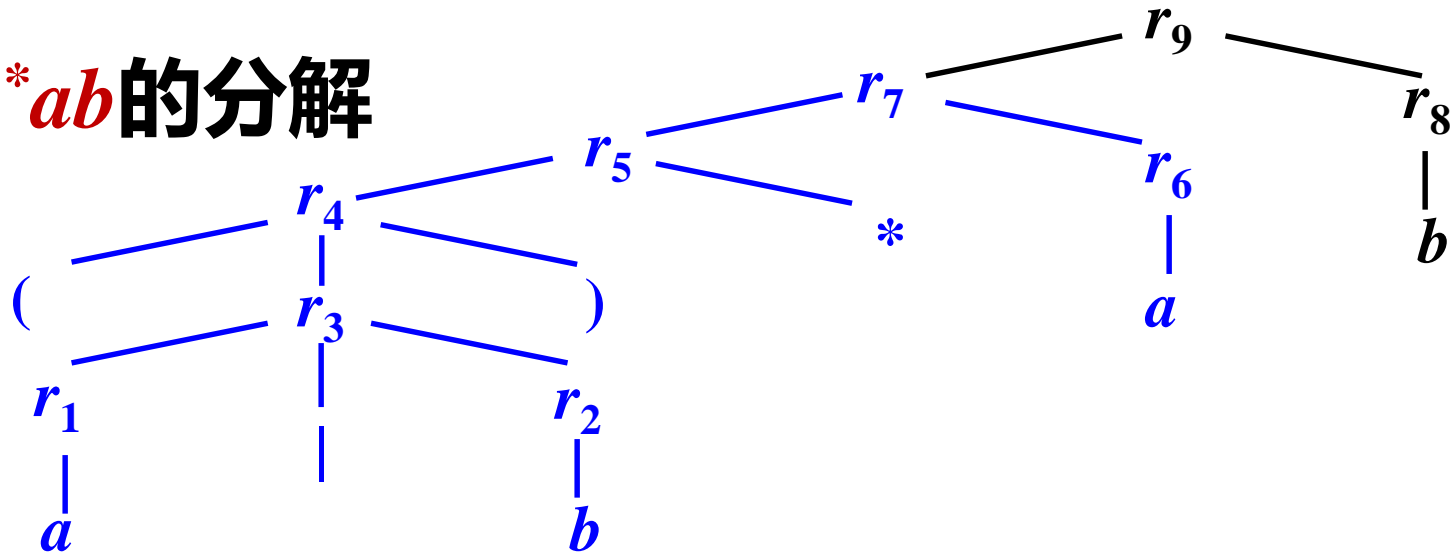
$(a|b)^*ab$ 的分解



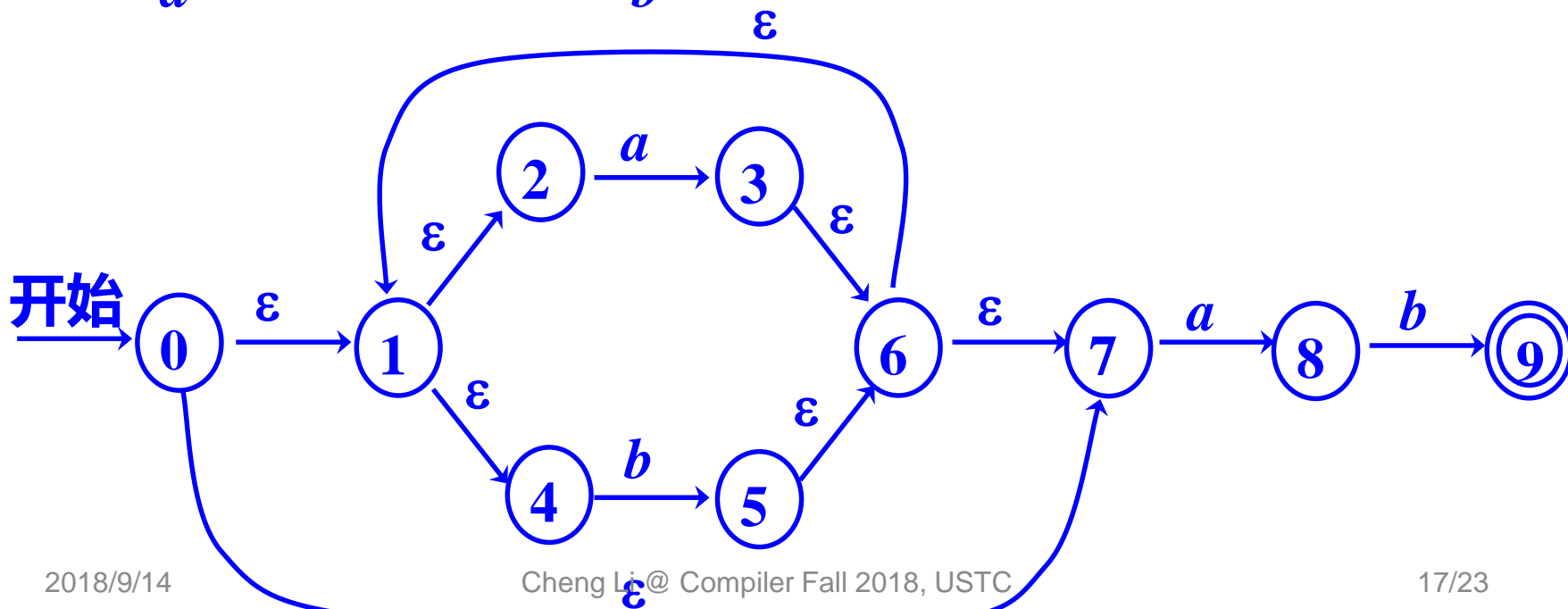
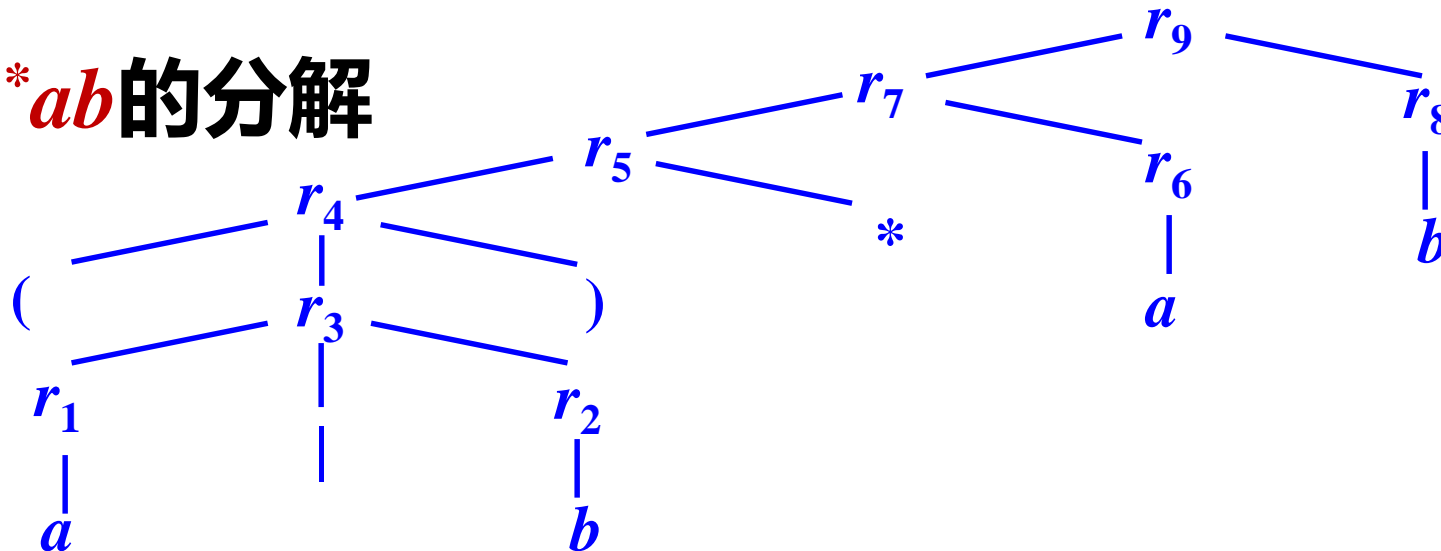
$(a|b)^*ab$ 的分解



$(a|b)^*ab$ 的分解

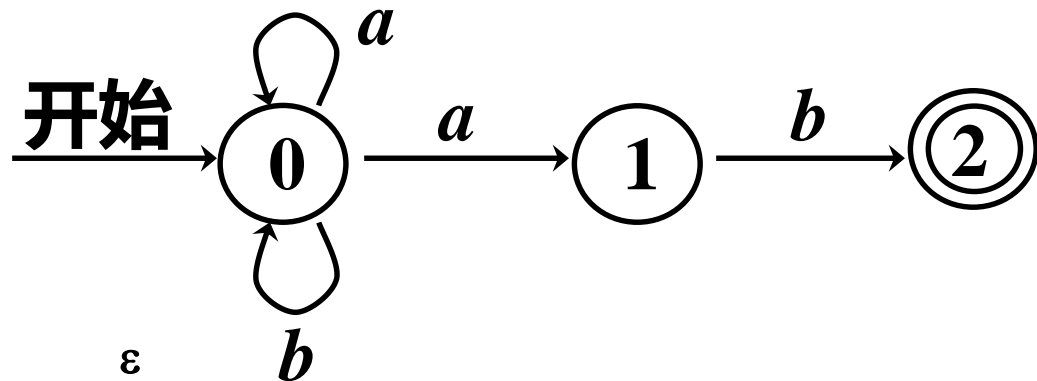


$(a|b)^*ab$ 的分解

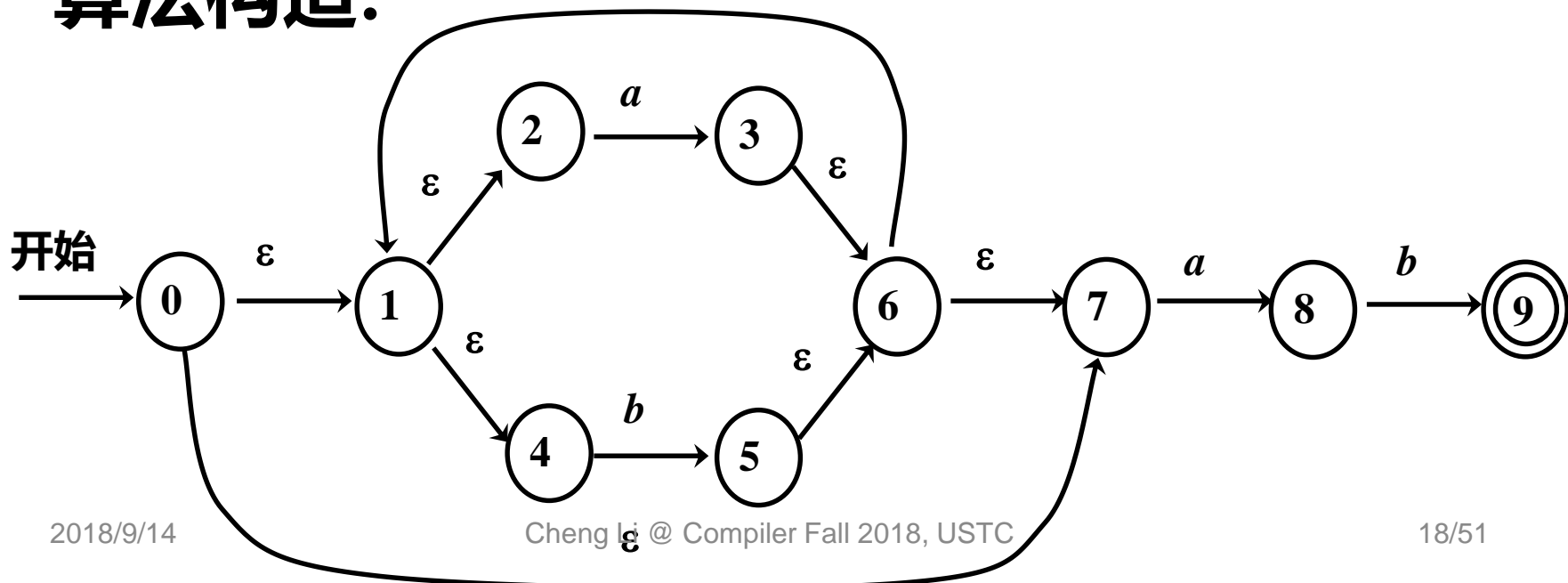


□ $(a|b)^*ab$ 的两个NFA的比较

手工构造:



算法构造:



□ 从正则表达式建立识别器的步骤

❖ 从正则表达式构造NFA（已介绍）

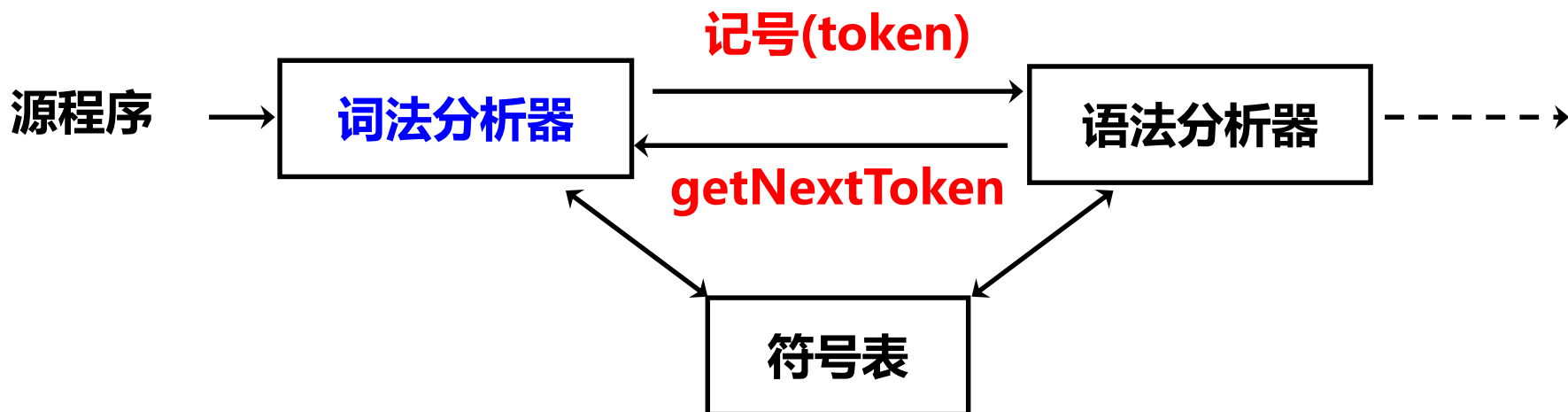
- 用语法制导的算法，它用正规式语法结构来指导构造过程

❖ 把NFA变成DFA（子集构造法，已介绍）

❖ 将DFA化简（合并不可区别状态，也已介绍）

□ 正则表达式 $(a|b)^*$ 与 $(a^*|b^*)^*$ 是否等价?

❖ 提示: 可利用其最简化DFA的



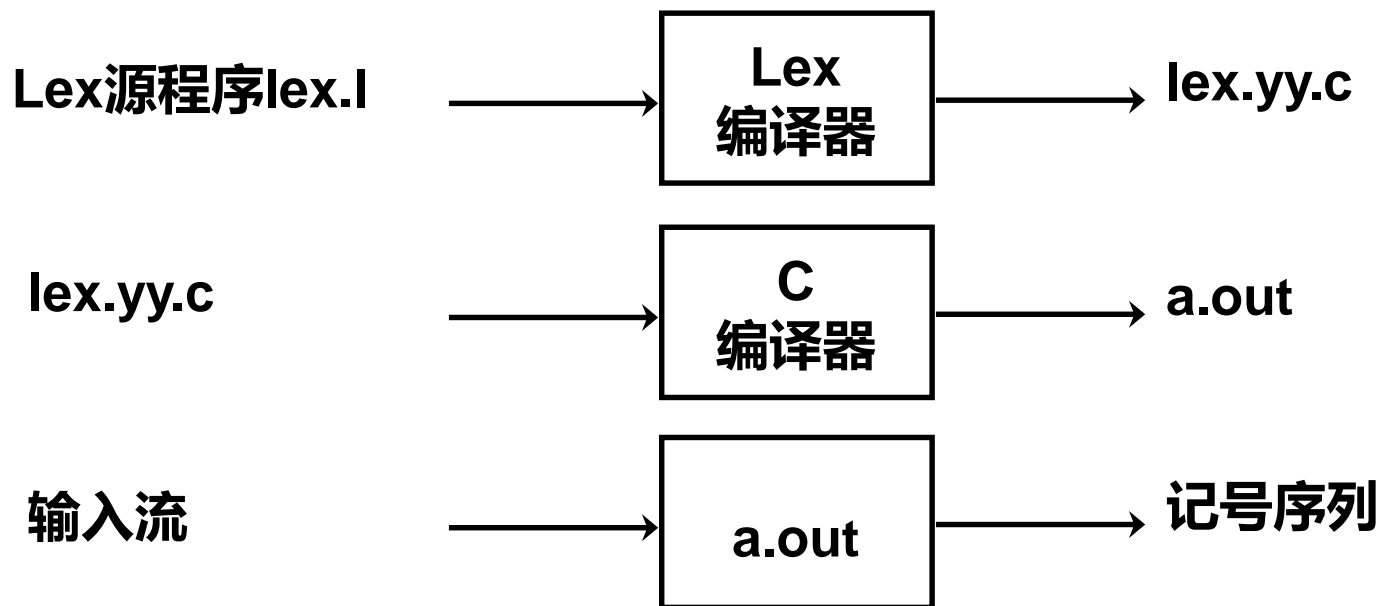
□ 词法分析器的自动生成

❖ 正则表达式 → NFA → DFA → 化简的DFA

□ 词法分析器的生成器

❖ Lex: flex、jflex

□ 用Lex建立词法分析器的步骤



□ 包括三个部分

声明

%%

翻译规则

%%

辅助过程

□ Lex程序的翻译规则

p_1 {动作1}

p_2 {动作2}

...

p_n {动作n}

```
%{  
/* 常量LT, LE, EQ, NE, GT, GE,  
   WHILE, DO, ID, NUMBER, RELOP的定义*/  
%}
```

```
/* 正则定义 */
```

```
delim    [ \t \n ]  
ws       {delim}+  
letter   [A-Za-z]  
digit    [0-9]  
id       {letter}({letter}|{digit})*  
number   {digit}+(\.{digit}+)?(E[+\-]?{digit}+)?
```




```
{ws}          { /* 没有动作，也不返回 */ }
while        { return (WHILE); }
do           { return (DO); }
{id}         { yylval = install_id ( ); return (ID); }
{number}     { yylval = install_num( );
              return (NUMBER); }
“ < ”        { yylval = LT; return (RELOP); }
“ <= ”       { yylval = LE; return (RELOP); }
“ = ”        { yylval = EQ; return (RELOP); }
“ <> ”       { yylval = NE; return (RELOP); }
“ > ”        { yylval = GT; return (RELOP); }
“ >= ”      { yylval = GE; return (RELOP); }
```



```
installId ( ) {
```

```
    /* 把词法单元装入符号表并返回指针。  
    yytext指向该词法单元的第一个字符，  
    yyleng给出的它的长度          */
```

```
}
```

```
installNum ( ) {
```

```
    /* 类似上面的过程，但词法单元不是标识符而  
    是数 */
```

```
}
```

□ **词法分析器的作用和接口，用高级语言编写词法分析器等内容**

□ **掌握下面涉及的一些概念，它们之间转换的技巧、方法或算法**

- ❖ 非形式描述的语言 \leftrightarrow 正规式
- ❖ 正规式 \rightarrow NFA
- ❖ 非形式描述的语言 \leftrightarrow NFA
- ❖ NFA \rightarrow DFA
- ❖ DFA \rightarrow 最简DFA
- ❖ 非形式描述的语言 \leftrightarrow DFA (或最简DFA)



《编译原理与技术》

词法分析 II

KISS (Keep It Simple, Stupid!)

It is easier to modify a working system than to get a system working.

——Wisdom