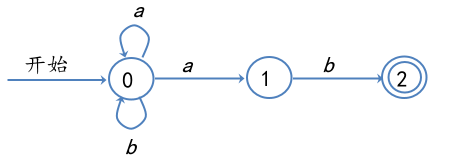
第三章 词法分析 →∪Σε

例(1) 为(a|b)\*ab构造NFA (状态转换图)



3.7 从正则表达式到自动机

从正则表达式到NFA （不确定）

从NFA到DFA

极小化DFA

从正则表达式到DFA

3.7.4 从正则表达式构造NFA

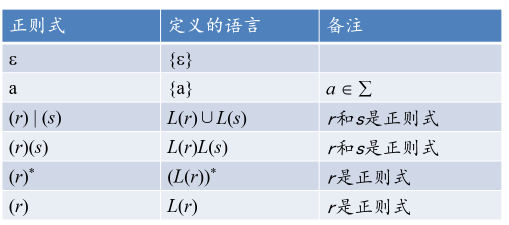
从正则表达式到NFA：

构造识别ε和字母表中一个符号的NFA

构造识别主算符为选择正则式的NFA

构造识别主算符为连接正则式的NFA

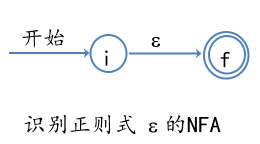
构造识别主算符为闭包正则式的NFA

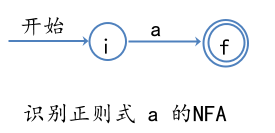


MYT算法(Thompson算法, McMaughton-Yamada-Thompson算法)

1. 构造识别ε和字母表中一个符号的NFA

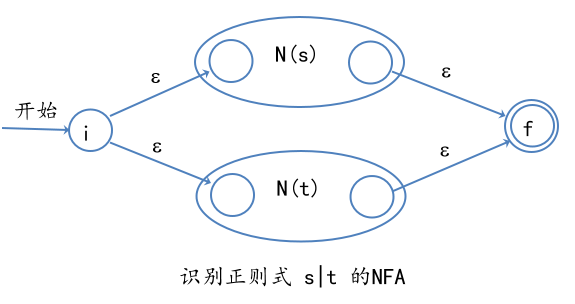
特点：仅一个接受状态，它没有向外的转换





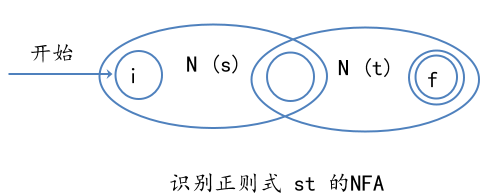
2. 构造识别主算符为选择正则式(s|t)的NFA

特点：仅一个接受状态，它没有向外的转换



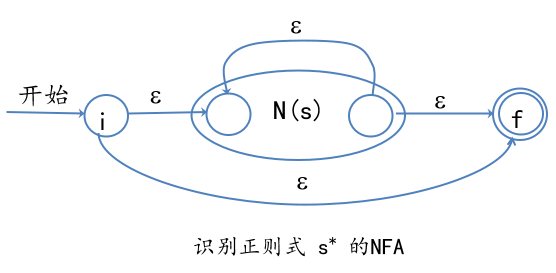
3. 构造识别主算符为连接正则式 st的NFA

特点：仅一个接受状态 ，它没有向外的转换



4. 构造识别主算符为闭包 s\*正则式的NFA

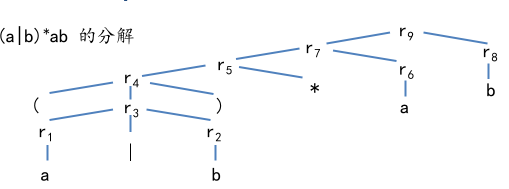
特点：仅一个接受状态，它没有向外的转换

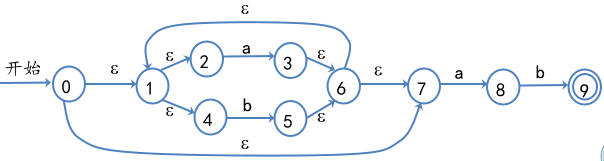


注意: 对于加括号的正则式(s)，使用N(s)本身作为它的NFA

* 本方法产生的NFA有下列性质：
  1. N(r)的状态数最多是r中符号和运算符总数的两倍
  2. N(r)只有一个接受状态，接受状态没有向外的转换
  3. N(r)的每个状态有一个用 Σ的符号标记的指向其它结点的转换，
  4. 或者最多两个指向其它结点的 ε转换

例(2) 构造 (a|b)\*ab 的NFA





练习:

1. 利用MYT算法构建a\*的NFA

2. 利用MYT算法构建a|b的NFA

3. 利用MYT算法构建(a|b)\*的NFA

11条边(8条空串边), 8上结点

4. 利用MYT算法构建(a|b)c\*的NFA

11条边(8条空串边) 9个结点