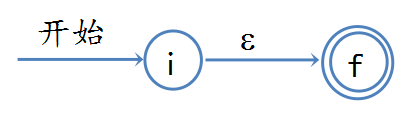
**实验3：从正则表达式到NFA**

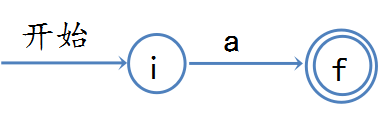
1. **预备知识**
2. **MYT算法**

McMaughton-Yamada-Thompson（MYT）算法可以将任何正则表达式转变为接受相同语言的NFA。该算法是语法制导的，即它沿着正则表达式的语法分析树自底向上递归的进行处理。对于每个子表达式，该算法构造一个只有一个接受状态的NFA。以下为基于MYT算法构建基本NFA单元的示例：

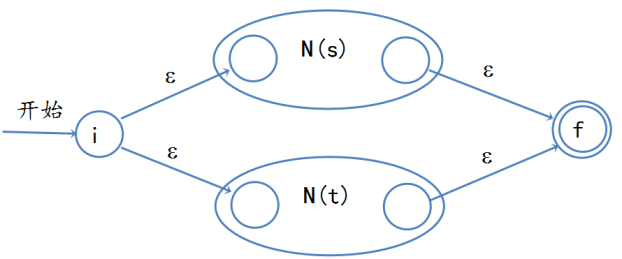
1）识别ε的NFA



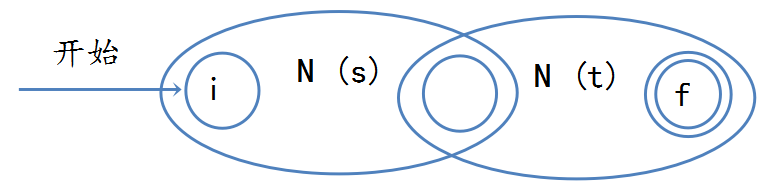
2）识别a的NFA



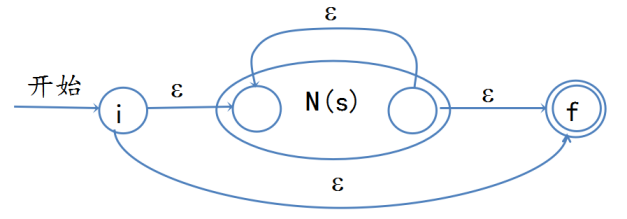
3）识别S|T的NFA



1. 识别ST的NFA



1. 识别S\*的NFA



1. **自动机的数据结构**

自动机（DFA和NFA）在计算机中可以存储为一组三元组，每个三元组为：（起始状态，符号，结束状态）。举例如下：

1）ab

|  |  |  |
| --- | --- | --- |
| 起始状态 | 符号 | 结束状态 |
| 0 | a | 1 |
| 1 | b | 2 |

2）ab\*

|  |  |  |
| --- | --- | --- |
| 起始状态 | 符号 | 结束状态 |
| 0 | a | 1 |
| 1 | b | 1 |

1. **实验要求**

本实验的难点为基于MYT算法为每种正则表达式的元字符设计一种存储三元组的方式。需完成以下正则表达式的解析，并构建相应的NFA。

1）ab

2）a\*b

3）a（a|b）c

4）a（ab）\*d

示例程序位于example/ 文件夹。

**建议用时：**2~4课时