**编译原理实践第3次课**

**（从NFA到DFA）**

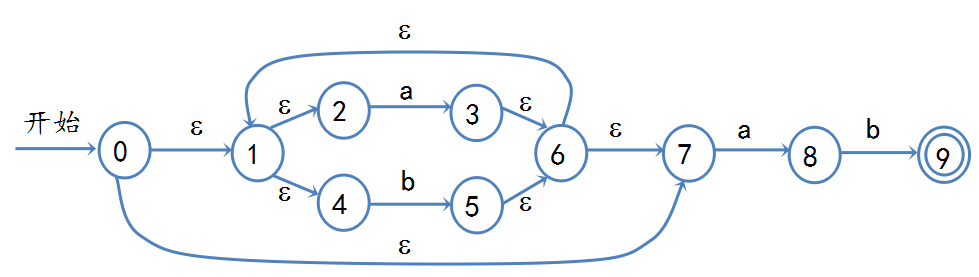
1. 基于子集构造法，从NFA构建DFA

1）输入为基于NFA的状态转换表，如下图所示（和上次课得到的NFA状态转换表相同）

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

2）输出为基于DFA的状态转换表

1. 核心是要实现eclosure（S）函数，即从状态集合S出发经过ε可以达到的所有状态集合。
2. 完成(a|b)\*ab的NFA到DFA的转换，NFA如下图所示：



1. 给定一个字符串，通过遍历生成的DFA，判断是否可到到达结束状态，如果可以到达，则接受，否则不接受。

DFA为上一题生成得到，字符串序列如下：

1）aab

2）ababab

3）aabbab

（*通过本次课和上次课的实验，我们能够构建一个完成的正则表达式解析器*）

**提交代码与实验报告（实验报告里需要有对于实验过程的文字描述与运行结果截图）**