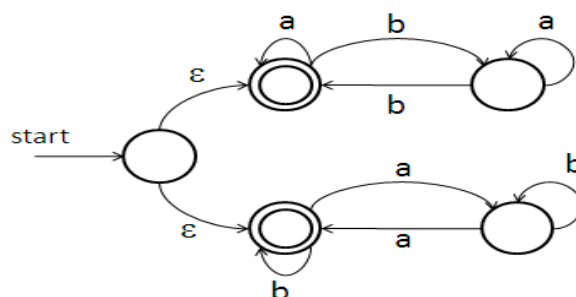


中山大学 软件学院 作业安排	
实验课程名称	编译原理（13 级软件工程 电政、嵌软、通软）
授课教师	万海
作业安排时间	2015 年 10 月 21 日 第五周
作业提交时间	作业一：纸质版在 10 月 28 日上课时，由学委收集并提交给任课老师。
	作业二：2015 年 10 月 28 日 星期三 18:00 前提交到 FTP 服务器

作业一

作业要求

一、考虑以下 NFA：



1. 这一 NFA 接受什么语言（用自然语言描述）？
2. 构造接受同一语言的 DFA.

二、正则语言补运算

1. 画出一个 DFA，该 DFA 恰好识别所有不含 011 子串的所有二进制串。
2. 假如要求你再画一个 DFA，该 DFA 恰好识别所有不含 101 子串的所有二进制串。你从中总结出解决这一类问题的普遍规律是什么？
3. 再证明：对任一正则表达式 R，一定存在另一正则表达式 R'，使得 L(R') 是 L(R) 的补集。

三、设有一门小小语言仅含 z、o、/（斜杠）3 个符号，该语言中的一个注释由 o 开始、以 o/ 结束，并且注释禁止嵌套。

1. 请给出单个正则表达式，它仅与一个完整的注释匹配，除此之外不匹配任何其他串。书写正则表达式时，要求仅使用最基本的正则表达式算子（ ϵ , |, *, +, ?）。
2. 给出识别上述正则表达式所定义语言的确有限自动机（DFA）。你可根据问题直接构造 DFA，不必运用机械的算法从上一小题的正则表达式转换得到 DFA。

提交说明

作业提交纸质版。

作业二

作业要求

Java 程序设计：

一. DFA.java

为 DFA.java 中的 DFA 类实现成员函数 `boolean recongnizeString(int move[], int accept_state[], String word)`，函数功能和参数说明如下。

函数功能：如果 word 被 move 和 accept_state 所代表的 DFA 识别，则返回 true，否则返回 false.

参数说明：

1) `move[][]` 是状态迁移函数, `move` 的每一行代表一个状态, 每一列代表一个输入符号, 第 0 列代表 'a', 第 1 列代表 'b', …… , 依此类推. 例如: `move = {{1,0}, {1,2}, …}`, 则表示此 DFA 在状态 0 时, 当输入为 'a' 时, 迁移到状态 1, 当输入为 'b' 时迁移到状态 0; 而 DFA 在状态 1 时, 当输入为 'a' 时, 迁移到状态 1, 当输入为 'b' 时迁移到状态 2. 注意: 默认状态 0 是 DFA 的初始状态.

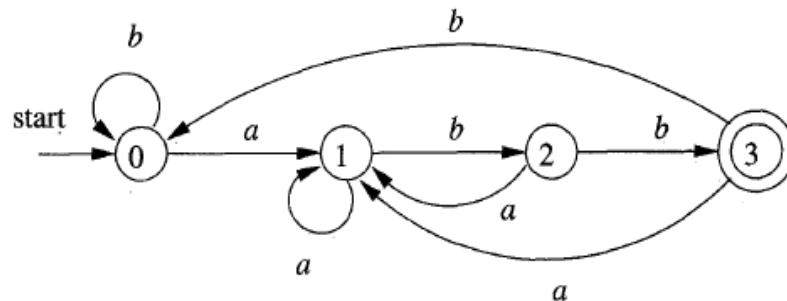
2) `accept_state[]` 是接受状态的集合, 如 `accept_state[] = {2, 3}`, 则表示状态 2 和状态 3 是接受状态.

3) `word` 是待识别的字符串.

如下例:

move	accept_state	word	recognizeString() 返回值
<code>{{1,0}, {1,2}, {1,3}, {1,0}}</code>	3	aaabb	true
同上	同上	abbab	false

上表中 `move` 和 `accept_state` 表示的自动机如下图所示



注意事项:

- 1) 字符串 `word` 的长度在 1 到 50 之间.
- 2) `move` 的状态数在 1 到 50 之间, 输入符号数在 1 到 26 之间.
- 3) 只须实现 `recognizeString` 函数, 注意不要修改 DFA.java 文件中其它部分的内容 (如果修改了, 请在提交之前改回来), 更不要修改 `recognizeString` 的函数名, 参数和返回类型. **但可以为 DFA 类添加新的成员变量和成员函数.**

- 4) DFA.in 文件是整个程序的输入文件, 在本地机器上调试或测试程序时, 可以修改 DFA.in 中的内容 (用 Editplus 或记事本都可以打开修改), 但遵照 DFA.in 的文件格式. DFA.in 的格式示例如下:

```

4 2      ----DFA 有 4 个状态, 2 个输入符号
1 0      ----接下来的 4 行 2 列代表状态迁移函数
1 2
1 3
1 0
3        ----这一行代表接收状态, 若有多个用空格隔开
aaabb    ----接下来的每行代表一个待识别的字符串
abbab
abbbaabb
abbb
#        ---- '#' 号代表待识别的字符串到此结束
1 3      ----新的 DFA 开始, 格式同上一个 DFA
0 0 0
    
```

```
0
cacba
#
0 0      ----两个 0 代表所有输入的结束
```

5) 当 DFA.in 内容如 4) 时, 屏幕应输出

```
YES
NO
YES
NO
YES
```

二. NFA.java

为 NFA.java 文件中的 NFA 类实现成员函数 `boolean recognizeString(int move[][][], int accept_state[], String word)`. 函数的参数和功能如下.

函数功能: 如果 word 被 move 和 accept_state 所代表的 NFA 识别, 则返回 true, 否则返回 false.

参数说明:

1) `move[][][]` 是状态迁移函数, `move` 的第一维代表状态; 第二维代表输入符号, 第 0 列代表 ϵ , 第 1 列代表 'a', 第 2 列代表 'b', …… , 依此类推; 第三维代表迁移到的状态的集合. 例如: `move[0][1] = {0, 1}`, 则表示此 NFA 在状态 0 时, 当输入为 'a' 时, 迁移到状态 0 或 1. 注意: 默认状态 0 是 NFA 的初始状态.

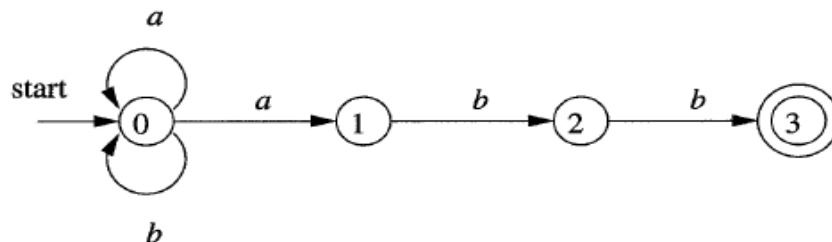
2) `accept_state[]` 是接受状态的集合, 如 `accept_state[] = {2, 3}`, 则表示状态 2 和状态 3 是接受状态.

3) `word` 是待识别的字符串.

如下例:

move	accept_state	word	recognizeString
<code>{{ {}, {0, 1}, {0} }, { {}, {}, {2} }, { {}, {}, {3} }, { {}, {}, {} }}</code>	3	aaabb	true
同上	同上	abbab	false

上表中 `move` 和 `accept_state` 表示的自动机如下图所示



注意事项:

- 1) 字符串 `word` 的长度在 1 到 50 之间.
- 2) `move` 的状态数在 1 到 50 之间, 输入符号数在 1 到 27 之间 (包括 ϵ 在内).
- 3) 只须实现 `recognizeString` 函数, 注意不要修改 NFA.java 文件中其它部分的内容 (如果修改了, 请在提交之前改回来), 更不要修改 `recognizeString` 的函数名, 参数和返回类型. 但可以为 NFA 类添加新的成员变量和成员函数.

	<p>4) NFA.in 文件是整个程序的输入文件，在本地机器上调试或测试程序时，可以修改 NFA.in 中的内容（用 Editplus 或记事本都可以打开修改），但遵照 NFA.in 的文件格式。NFA.in 的格式示例如下：</p> <pre> 4 3 ----NFA 有 4 个状态，包括ε共有 3 个输入符号 {} {0, 1} {0} ----接下来的 4 行 3 列代表状态迁移函数 {} {} {2} {} {} {3} {} {} {} 3 ----这一行代表接收状态，若有多个用空格隔开 aaabb ----接下来的每行代表一个待识别的字符串 abbab abbaaabb abbb # ---- ‘#’ 号代表待识别的字符串到此结束 0 0 ----两个 0 代表所有输入的结束 </pre> <p>5) 当 NFA.in 的内容如 4) 时，屏幕应输出：</p> <pre> YES NO YES NO </pre>
提交说明	<ol style="list-style-type: none"> 1. 实现功能的 DFA.java 和 NFA.java 源文件和 DFA.in/NFA.in 测试文件 2. 在 windows 或 linux 下的编译执行的脚本文件 3. 对 DFA.java 和 NFA.java 的程序设计说明文件 DFA.txt 和 NFA.txt <p>以上内容按“学号_姓名_实验 1.rar”打包后，提交到 FTP。</p> <p>13 级嵌软、通软、电政分别提交到各自方向所在目录：</p> <p>ftp://my.ss.sysu.edu.cn/~wh</p> <p>再访问以下目录：</p> <p>/homework_upload/13 级 编译原理/13 级嵌软（通软、电政）/Week5（2015 年 10 月 28 日 星期三 18:00 截止）/</p>