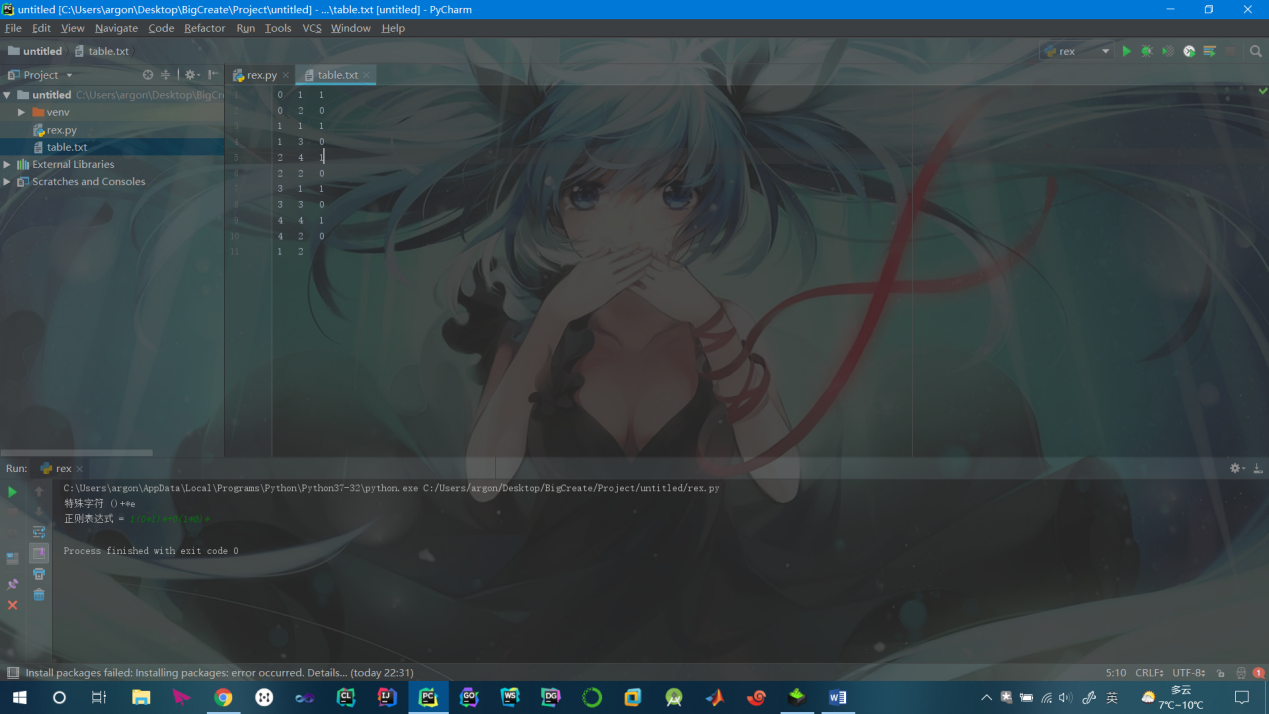
1. 使用说明

使用Python 3.x 运行代码 rex.py，输入正则表达式，DFA最小化表格在同一目录下的table.txt文件中，table.txt文件中，除了最后一行表示终态外，其他行的第一列、第二列、第三列分别表示和起始状态、目标状态和跳转的字符。



输入正则表达式1(0\*1)\*+0(1\*0)\*，输出

0 1 1 状态0经过字符1到达状态1

0 2 0 状态0经过字符0到达状态2

1 1 1 状态1经过字符1到达状态1

1 3 0 状态1经过字符0到达状态3

2 4 1 状态2经过字符1到达状态4

2 2 0 状态2经过字符0到达状态2

3 1 1 状态3经过字符1到达状态1

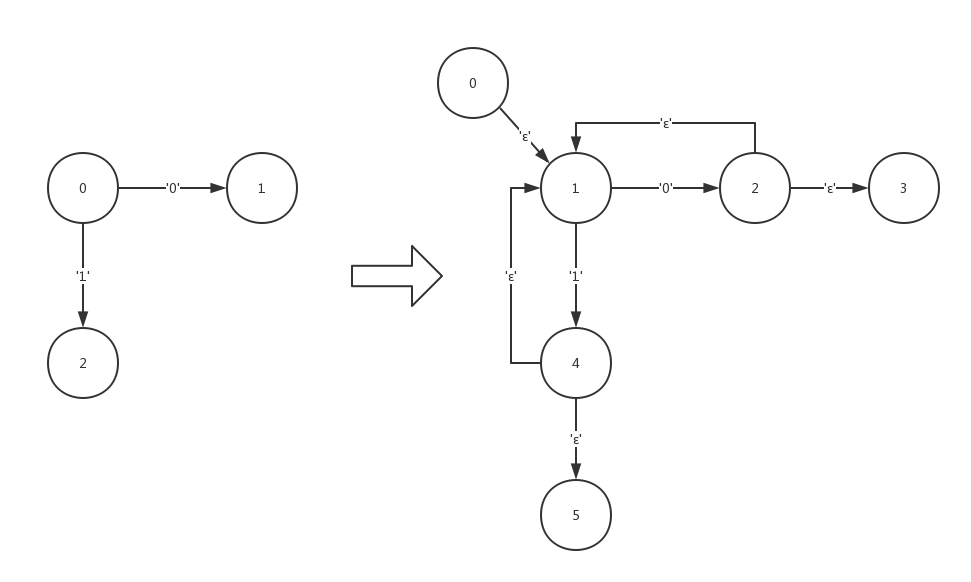
3 3 0 状态3经过字符0到达状态3

4 4 1 状态4经过字符1到达状态4

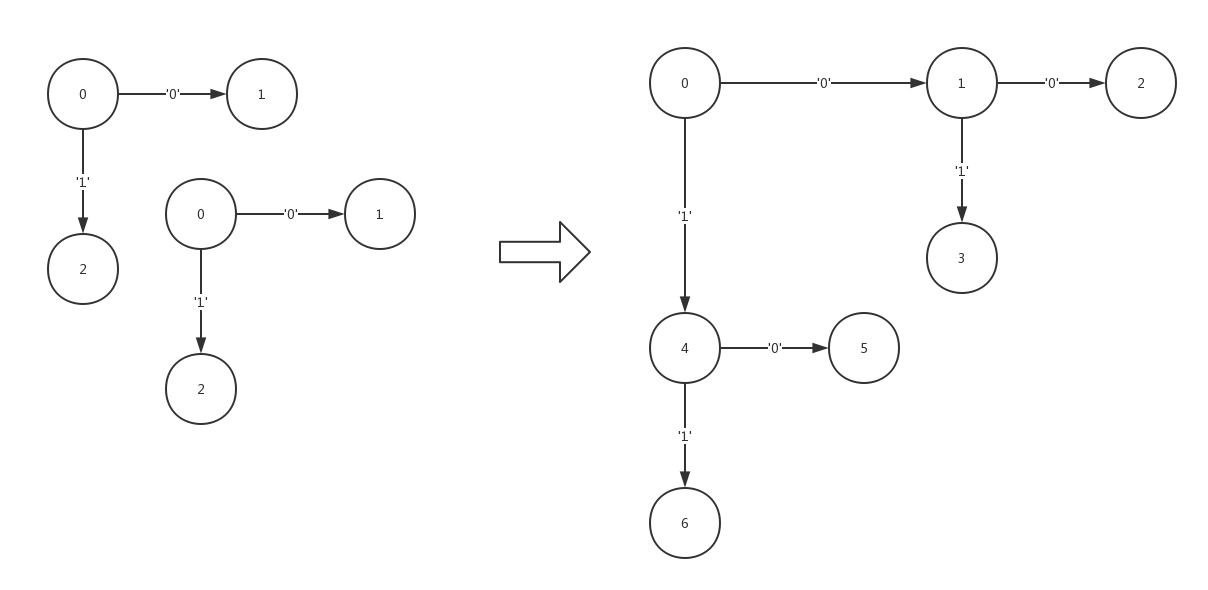
4 2 0 状态4经过字符0到达状态2

1 2 1和2是终态

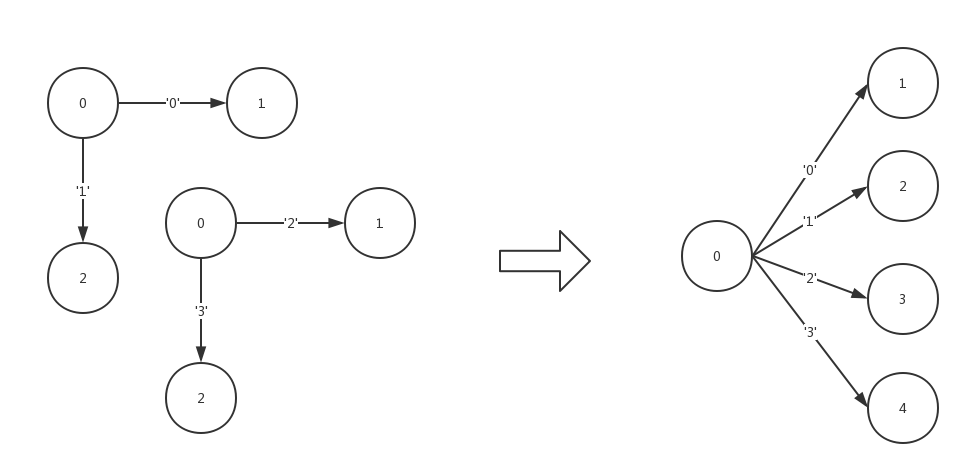
1. 生成流程
2. 用占位符（这里用e表示）替代正则表达式中的非特殊字符，每次替换在对应位置生成原子状态机，
3. 重新遍历字符串，将e\*用e代替，修改e中状态机，如下图所示，添加一个新的开始状态，对每一个终态，添加两条空的转换，对状态重新编号，更新终态



1. 重新遍历字符串，将ee用e代替，合并对应的状态机，第一个状态机保留开始节点，将第二个状态机合并到第一个状态机的每个终态，如下图所示，对状态重新编号，更新终态



1. 重新遍历字符串，将(e)用e代替，不修改状态
2. 重新遍历字符串，将e+e用e代替，合并两个状态机，两个状态机共用起始状态，对状态重新编号，更新终态



1. 当字符串为’e’时，NFA生成完成，否则，回到步骤2)
2. 将原始字符串中的所有非特殊字符添加进字符集
3. 将起始状态添加进二维表，添加从起始状态对于特定字符可达的状态进本行的对应于特定字符的列，同二维表中没有该状态作为起始状态的行，添加该状态作为起始状态的行
4. 对于二维表中的每一行的起始状态，添加从起始状态对于特定字符可达的状态进本行的对应于特定字符的列，同二维表中没有该状态作为起始状态的行，添加该状态作为起始状态的行
5. 对每一行编号，添加行与行之间的状态转移映射
6. 将状态按包含终态和不包含终态分为两类，合并两类中相同的状态
7. 用同一类状态中编号最小的状态编号更新行与行之间的状态转移映射
8. 合并相同行
9. 对状态重新编号（因为此时状态编号可能不连续）