

编译原理实验一

设计说明

实验题目 正规式与状态图

学生姓名 陈南

学生学号 201806062101

学生班级 软工1801

提交日期 　 2020年10月29日

**计算机科学与技术**

# 一、实验内容

# 1、实验目的

编程实现正规式转换到非确定有穷自动机转换的一般算法

# 2、实验要求

**1. 用C++ 实现；**

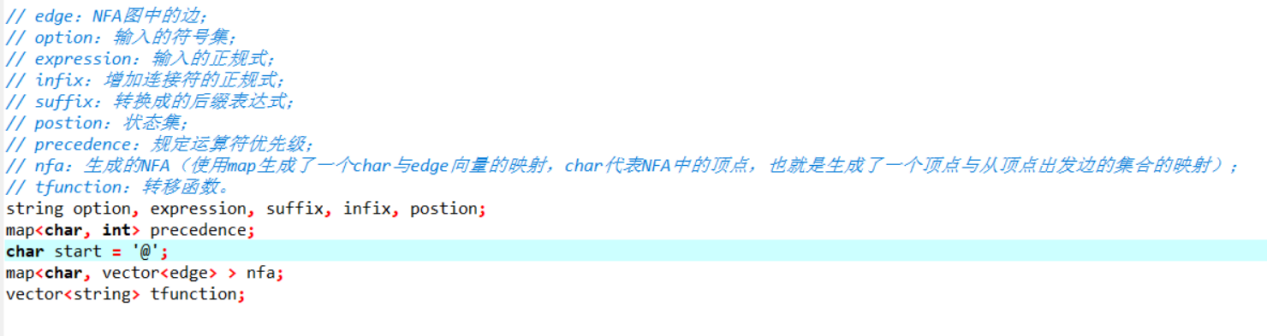
**2. input.txt描述正规式，需要编程加以解析**，该文件规定的格式为：第一行写处正规式的符号集，规定符号集为0-9共10个数字符，第二行写正规式，如： 0123456789 (0|1)\*0.10\* （注：\*表示闭包运算，.表示连接该符号可省略）

**3. output.txt为描述非确定有穷自动机的文件，有程序生成**，生成时需按如下格式： 第一行写出状态集，用大写字母A-Z表示，如：ABCD 第二行描述自动机的符号集为0-9共10个数字符，如：0123456789 第三行写初态集，如：A 第四行写终态集，如：CD 第五行写出转移函数,如(A,0,B)表示A状态的出边0到达B状态。上例正规式可表为(A,0,A)(A,1,A)(A,0,B)(B,1,C)(C,0,D)(D,0,D)。

**4.提供名为retonfa.exe的执行文件实现上述功能**，还需提供源文件以及名为设计说明.doc的说明文件。

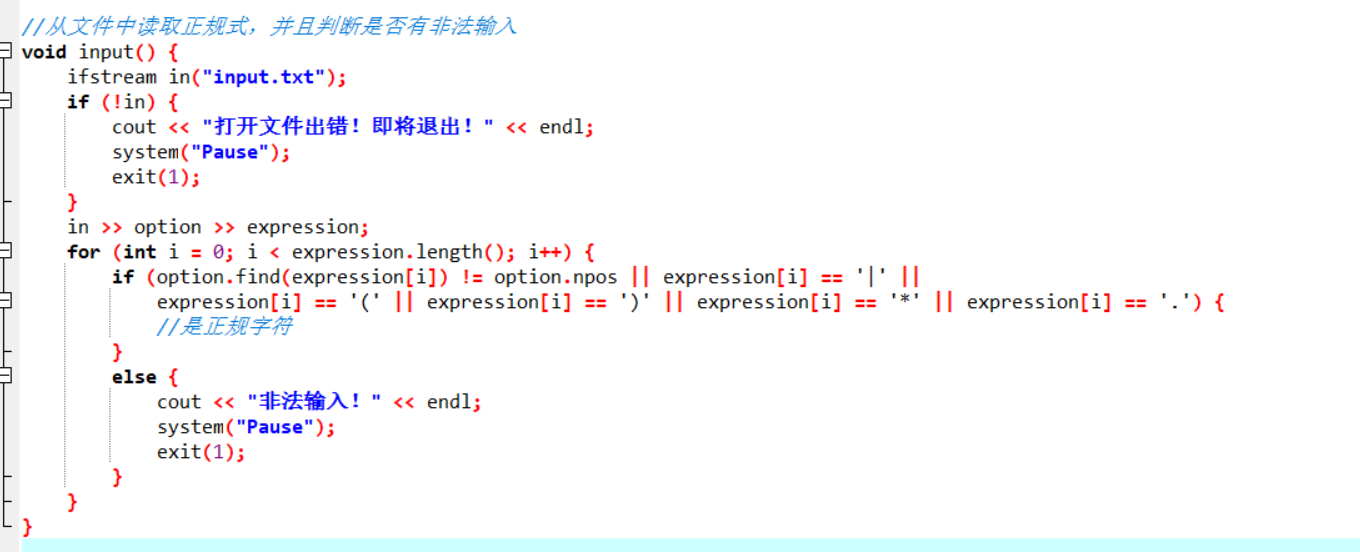
# 3、设计说明

1. 数据结构：



1. input函数：

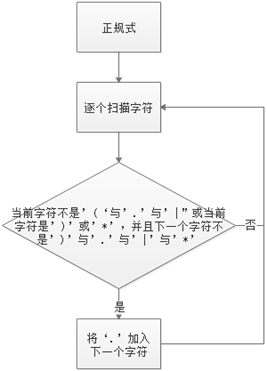
从文件中读取正规式，并且判断该正规式中是否有非法输入

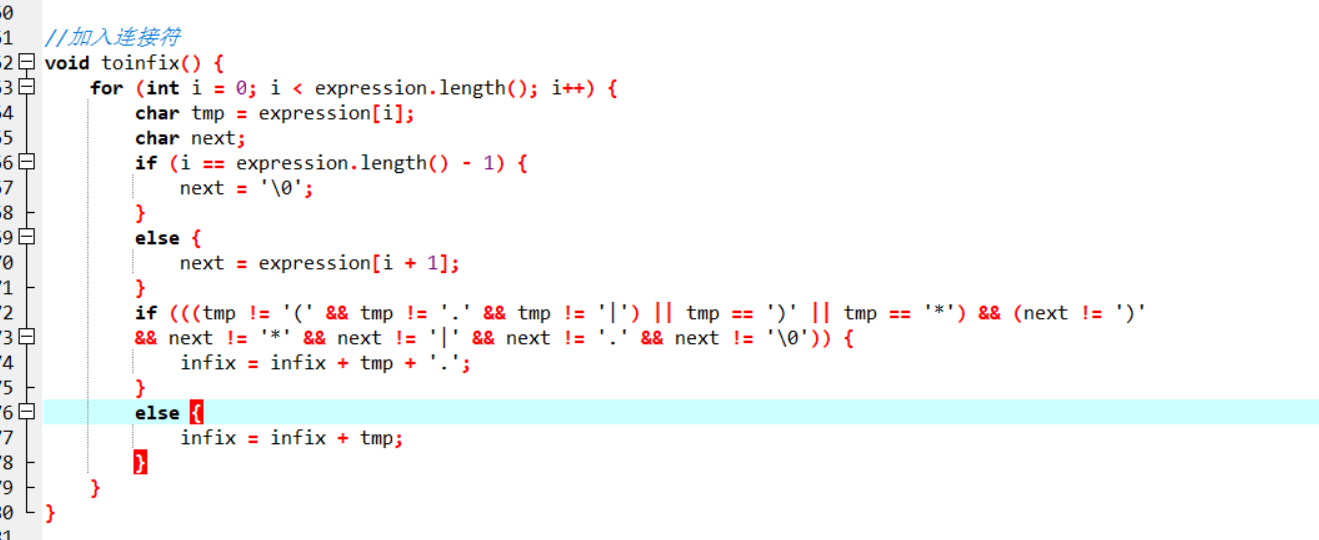


1. toinfix函数:

加入连接符，更加方便后面的中缀转后缀，将省略的连接符加入，可以一次遍历完成，需要添加的有ab、\*b、a(、)b等情况

具体流程如下：



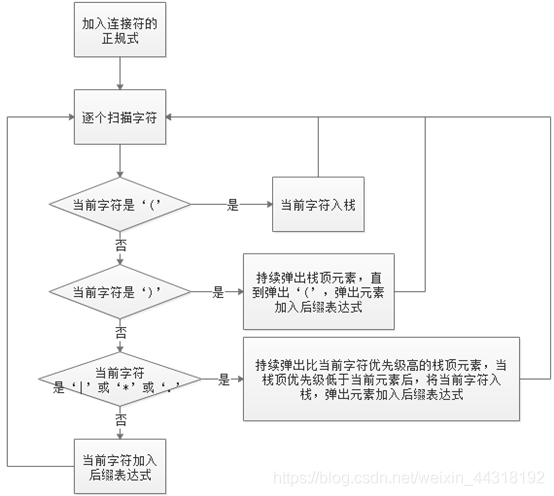


1. tosuffix函数

tosuffix函数：

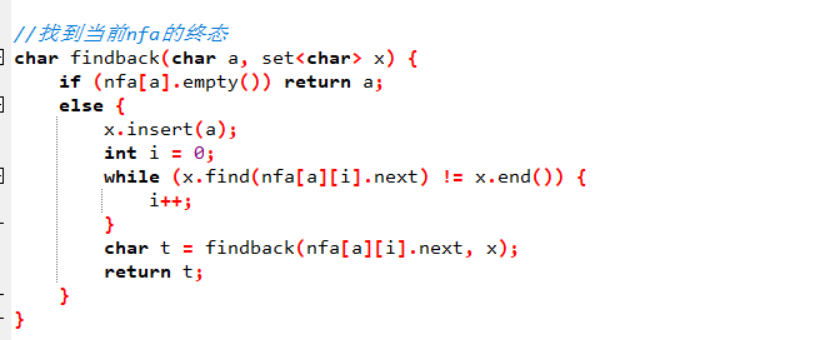
中缀转后缀

具体流程：



1. findback函数

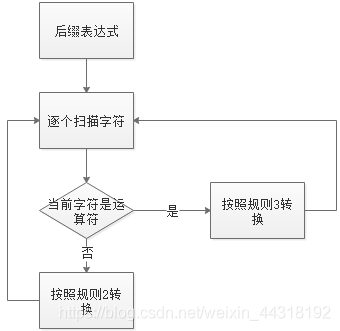
通过函数的递归调用寻找到nfa的最终状态。

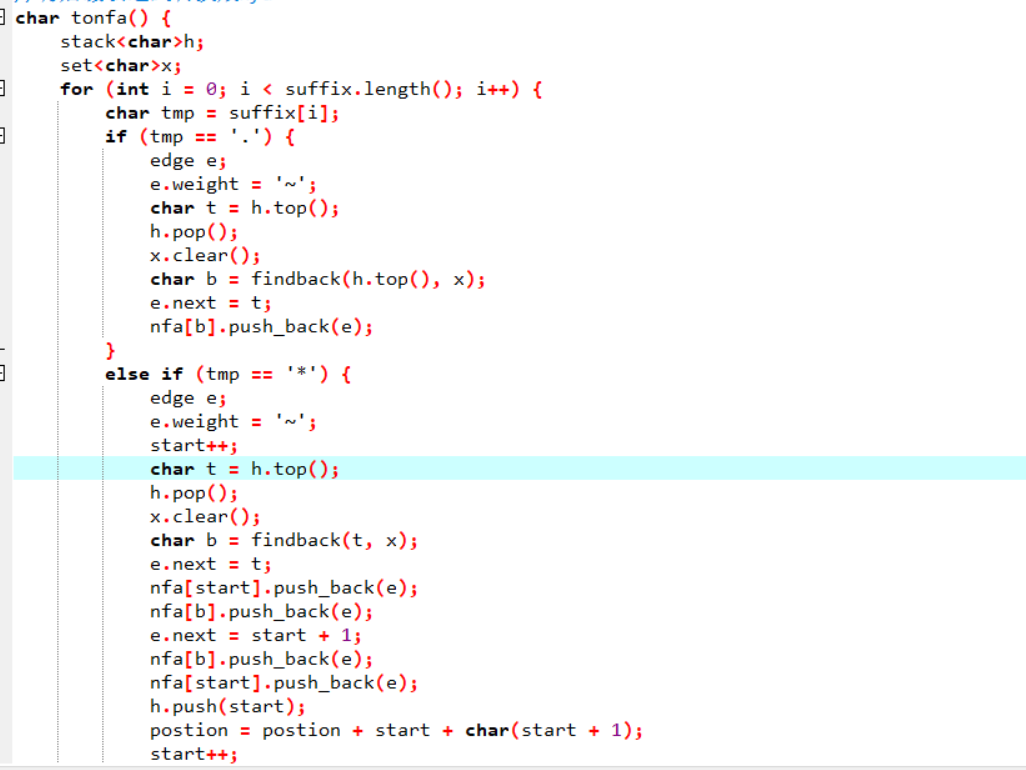


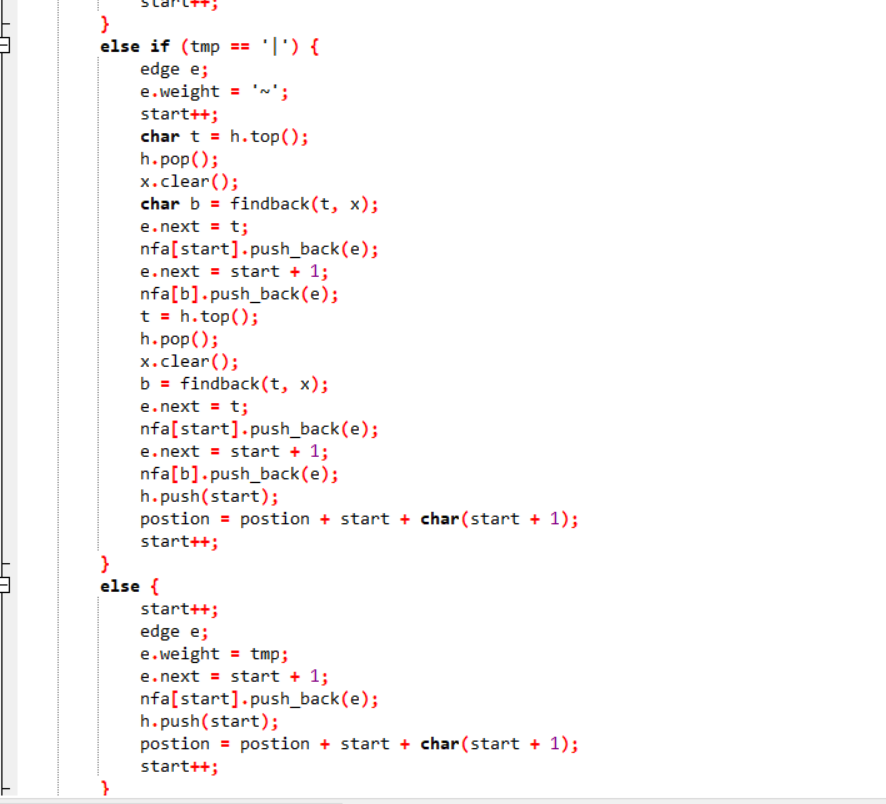
1. tonfa函数

将后缀转化成nfa

具体流程如下：







7.getfuncion函数

转化方程，把每一个转移函数放到push到string数组中，最后方便输出。

