基于最简DFA的词法分析程序生成器设计说明书

组长： 1952726 李梦祥

组员： 1951220 伊 啸

1953103 刘思睿

1953704 吕虹霖

# 1 程序功能描述

描述程序是用来干什么的、主要流程、输入输出（简要介绍）

程序功能简介：

将程序设计语言的每一种单词类别整理成若干个正规表达式，通过Thompson算法分别转换成NFA。通过子集法把NFA转换成DFA，进而利用状态等价法进行DFA的化简，再通过基于DFA的词法分析驱动程序即可进行词法分析。

程序运行流程简介：

输入输出：

输入：PL/0语言

输出：对于代码的识别结果，包括关键字和属性值两部分。

# 2 程序实现过程

大体也是这四个部分，讲每个部分的算法原理（中间要贴一些代码，写一点分析）

2.1 正规式转化为NFA （Thompson 算法）：

输入正规表达式，通过状态转移矩阵和图的形式输出NFA。

算法原理：

1完成对于基本元素模块的算法构造

2进而完成对于R｜S，RS，R\*等基本运算的归纳构造

3 根据输入的正规表达式转化生成为非确定有限自动机NFA，并一状态转移矩阵的形式输出。

相应代码展示：

2.2 NFA转化为DFA（子集法）

算法原理：

1首先引入ε-CLOSURE(I)运算和Ia运算，分别用于定义从I中某一个结点出发经过一条ε弧和a弧能够到达的状态节点的全体集合

2构造状态转移矩阵

3将表中的每一列的每个子集视作一个新的状态并重新命名

相应代码展示：

2.3 DFA 化简（状态等价法）

算法原理：

1 将状态集S划分为终态集和非终态集两个集合作为初始划分

2 检查状态子集中的每个集合是否可以再分，依据为如果存在一个输入字符a，使得I不全包含在现行状态集合的子集中，就将I进行划分

相应代码展示：

2.4 DFA 代码化

算法原理：

1 初始化指针指向输入符号串的首字符，对字符串进行一次扫描，将扫描到的字符根据2 DFA状态迁移矩阵进行状态迁移。若迁移成功则进行后续扫描迁移，无法迁移则根据是否为终态判断成功或者报错。

相应代码展示：

# 3 运行界面和运行结果

运行几个测试用例，贴图，描述一下结果，写一点分析

# 4 设计体会（难点及解决方法）

难点及解决方法：

心得体会（供参考和挑选的几点）：

通过对于词法分析器项目的设计和实现，项目本身的编程过程也让我对于C++编程语言的编程方法和编程思想有了更加熟练的掌握，更重要的是加深了我对于课程上所学习的编译原理的理论知识的理解和掌握，同时也深刻理解了词法分析器的运作过程和实现原理。认识到“从正规式转化生成NFA”，“NFA到DFA”，“DFA的化简”和“DFA的代码化”整个看似相对独立实则紧密联系的环节，各个环节仅仅相扣在一起，对于相关算法的实现有了更加整体更加系统的认识，也有了更加透彻的理解。

另一方面，编译原理是一门概念性和理论性很强，内容丰富且抽象、且具有严密的逻辑性的课程，但是课程组队项目的完成过程让我从代码化的实践角度对于课堂上的理论知识又了新的理解和体会，实践的过程拉近了我和理论知识的距离。

一、对实验原理有更深的理解

通过该课程设计，掌握了什么是编译程序，编译程序工作的基本过程及其各阶段的基本任务，熟悉了编译程序总流程框图，了解了编译程序的生成过程、构造工具及其相关的技术对课本上的知识有了更深的理解，课本上的知识只是机械且表面的。通过把该算法的内容，算法的执行顺序在计算机上实现，将看似深奥的书本知识变的更为简单，同时对实验原理有更深的理解。

二、对该理论在实践中的应用有深刻的理解

通过把该算法的内容，算法的执行顺序在计算机上实现，知道和理解了该理论在计算机中是怎样执行的，对该理论在实践中的应用有深刻的理解。

三、激发了学习的积极性

通过该课程设计，全面系统的理解了编译原理程序构造的一般原理和基本实现方法。把死板的课本知识变得生动有趣，激发了学习的积极性。把学过的计算机编译原理的知识强化，能够把课堂上学的知识通过自己设计的程序表示出来，对计算机编译原理的认识更加深刻。

在这次课程设计中，我就是按照实验指导的思想来完成。加深了理解文件系统的内部功能及内部实现，培养实践动手能力和程序开发能力的目的。

四、理解了该知识点以及学科之间的融合渗透

本次课程设计程序部分是用c语言编写的，把《编译原理》，《算法分析与设计》，《高级语言程序设计》三门学科联系起来，把各个学科之间的知识融合起来，把各门课程的知识联系起来，对计算机整体的认识更加深刻。使我加深了对《计算机操作系统》，《编译原理》，《算法分析与设计》，《高级语言程序设计》三门课程的认识。