Report on Lab1

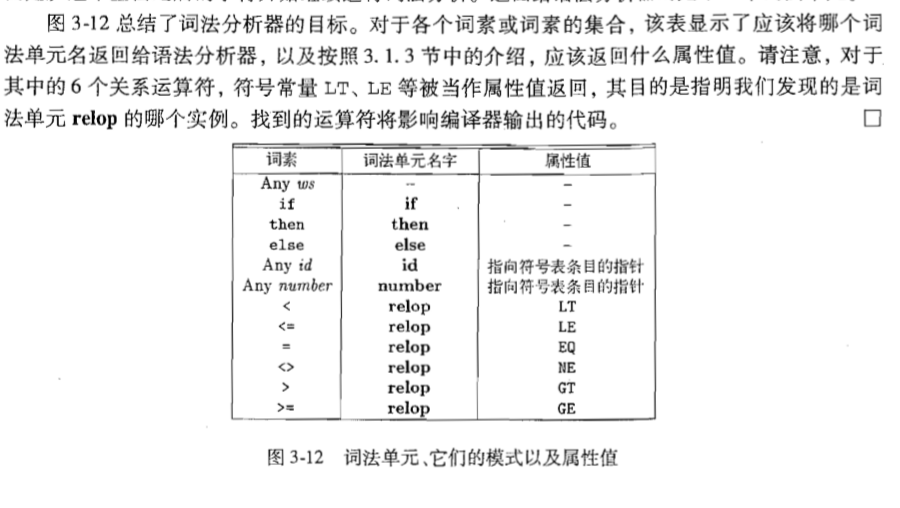
71Y17121 杜昕昱

1. 实验目的
2. 加深对RE到NFA到DFA到DFA（o）的转换的理解，掌握词法分析流程。
3. 掌握词法分析实现方法，锻炼编程能力。
4. 实验内容

设计一个可以对C++代码文件进行词法分析的程序，通过自己设计的Res来对输入文件进行分析，输出token序列，并有一定的报错机制。

1. 实验思路与设想

词法分析器基于课堂学习的词法分析流程来进行，即RE->NFA->DFA->DFA（o），最终词法分析器通过DFA（o）来对输入的字符流进行词法分析。对于一个C++代码文件，其中包含的词法单元主要包含有数字、变量名、关键字（int、const）、运算符、、、、对于其中的每一种此法单元，我们都可以构造对应的RE表达式，在经过RE->NFA->DFA->DFA（o）构造出相应的DFA(o)来对词法单元进行分析。

龙书上说明的词法分析器的目标如下，即输入一个字符串，将其中不同的语法单元以及属性值返回给语法分析器作进一步分析，其形式如下：

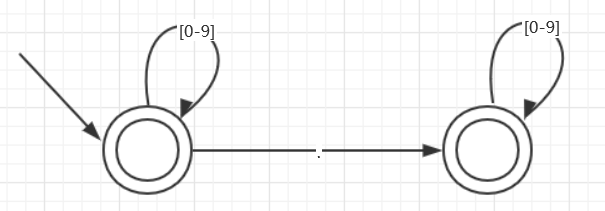
所以我们对于每一个词素都要对应到他相应的词法单元以及属性值，对于操作符等符号，其属性值我们先自行以数字编号形式设定。

1. FA描述

对于各种词法单元，我们可以构造相应的FA来对其进行表达

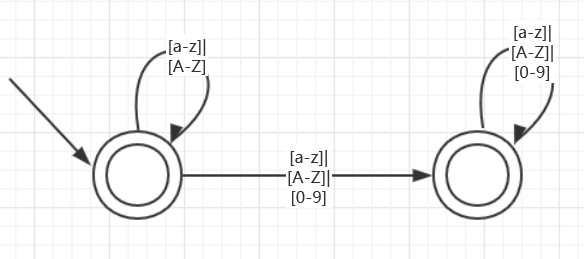
1. 数字类型：

该类型由数字组成，包括整数或浮点数等，可以绘制其FA为：



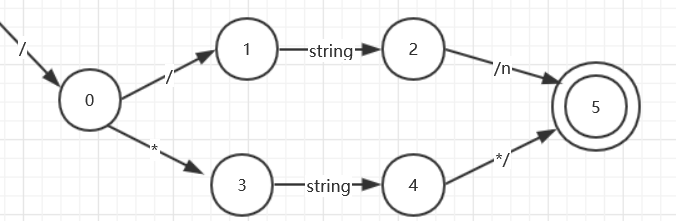
1. 字母+数字类型：

该类型由字母和数字组成，但必须由字母开头，包括自定义变量名、关键字等，其FA为：



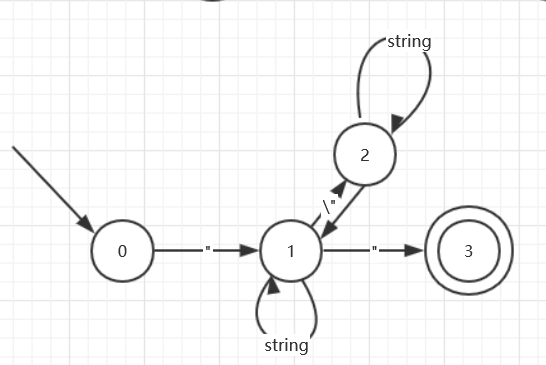
1. 注释类型

该类型以字符‘/’开头，有两种注释方式，FA如下：



1. 字符串类型

该类型以双引号“开头，以双引号”结尾，中间内容除非以\”的形式出现，否则不允许出现双引号”,FA如下：



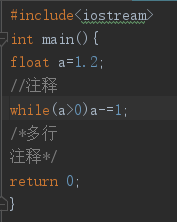
1. 数据结构及核心算法

词法分析器设计为一个扫描器加若干词法单元分析器组合而成，扫描器（Scanner（））中含有待分析文件以及读头下标，读头下标为当前分析器分析位置。词法单元分析器（analyzeDigit（）等）是当扫描到可能为该词法单元的字符是调用相应的词法单元分析器，由其划分出相应的token以及进行错误处理，将结果返回至扫描器中继续扫描，扫描器每次进行一个词法单元的划分直到走到尽头。

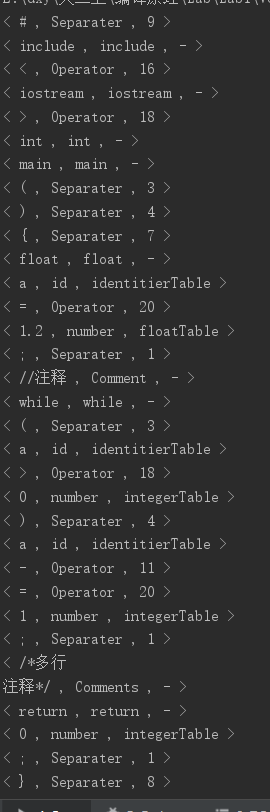
扫描器通过while循环使读头不断前进，通过对每个token的首字符进行分析来进入相应的词法单元分析器，在词法单元分析器中，模拟DFA中的路线，通过while循环不断向前移动读头，直到不能继续前进，将识别出的token在词表中进行查找并返回到Scanner放入token序列 ，Scanner移动读头进行下一个token的识别。

1. 测试用例

Test1：

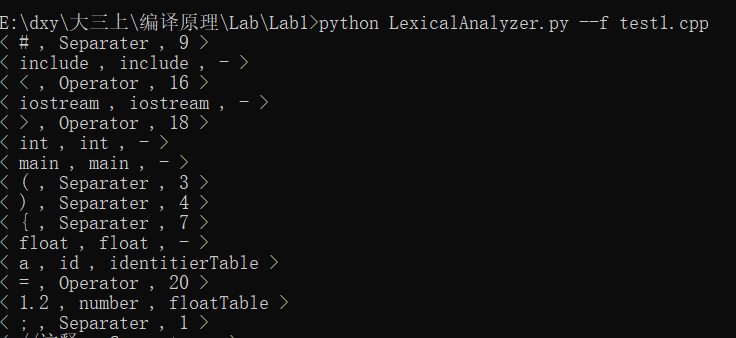


Output：



上面是test1的解析结果，其他测试文件Test2,3是测试错误的，test4是找了一个完整的数据结构的C++代码进行解析。

还可以使用命令行启动：



1. 问题及解决方法

在词法分析器扫描的过程中，按照最开始的判断顺序（先判断操作符后判断注释）会造成注释判断错误，原因在于注释串的第一个符号为‘/’，也是一个操作符，当先进行操作符的判断时会造成注释的符号误判为操作符从而使得注释不被识别导致分析错误，解决方法为将注释的判定放在操作符判定之前进行，从而确保注释被识别成功。

1. 个人感受

通过这次设计词法分析器的实验，我对于词法分析的流程有了更加深入的认识，经历了从分析问题到设计数据结构与算法，我对于词法分析有了一个清晰深入的理解，同时较大的代码量也锻炼了我的编程能力与细心程度。