编译原理实验报告

--语法分析器的设计与实现

班级：07111405

姓名：刘嘉豪

学号：1120141883

一、实验目标

构造语法分析器实现对经过词法分析生成的.xml文件的分析，支持对于下述语句的处理：赋值语句、返回语句、一种分支语句和一种循环语句；同时支持对于运算符：+、-、\*、/的识别和处理。语法分析器的输入、输出结果按照框架进行。

二、实验过程

该语法分析器的实现基于LL（1）文法。首先根据规定的文法进行对非终结符的FIRST、FOLLOW集的求解，然后根据FIRST、FOLLOW集求解结果构造分析栈，最后根据该分析栈生成一棵前序遍历N叉树，最后根据这棵树的节点关系生成相应的.xml文件。

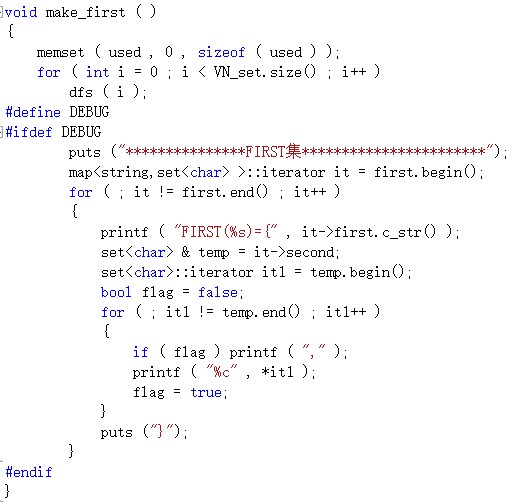
三、实验方法

1、所用文法

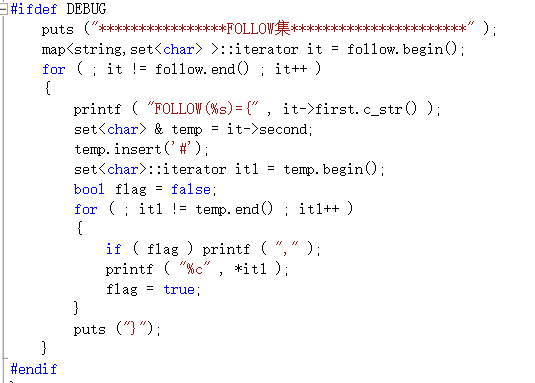


2、FIRST、FOLLOW集合的构造

根据构造的文法设计相应的FIRST集函数，需要对每个读入的非终结符进行处理，产生对应于每个非终结符的子集，相关代码如下：



在构造FIRST集遇到产生式为空的情况需要产生FOLLOW集，主要使用make\_follow ( )函数对遇到空的形式进行推导，具体过程见源码文件。产生FOLLOW集的相应代码如下：



3、分析栈以及生成树的构造

在构造生成树时，采用构造前序遍历的N叉树的形式产生，通过设计Tree类和结点CTreeNode、CLeafNode结构构造树，相关调用函数为Tree：：系列函数，具体过程见源代码。

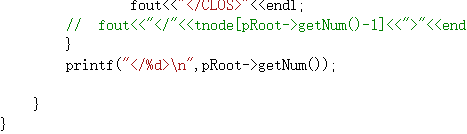
4、xml文件的生成

根据构造的N叉树，我们利用每个子节点与父节点之间的关系，采用前序遍历优先搜索，根据每个结点的属性构造相应的xml文件层次，相关代码如下：



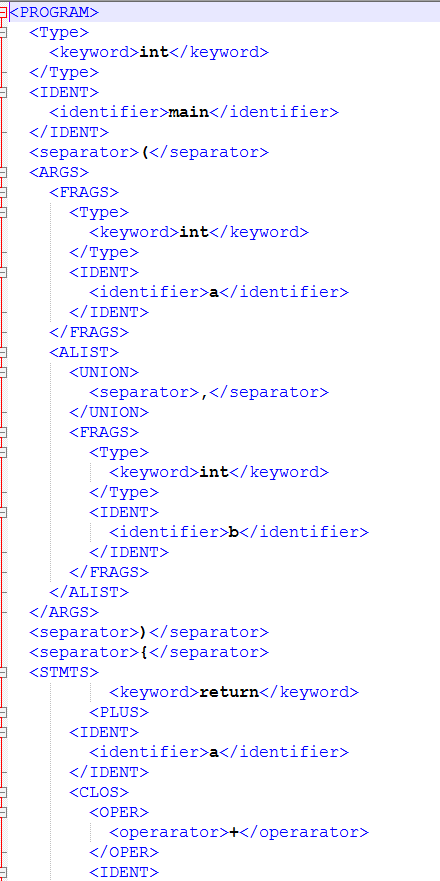






四、实验结果

通过对程序字符的输入，经过LL（1）文法处理可以得到相关的xml文件：test.tree.xml。部分截图如下：



五、实验心得

通过对语法分析器的编程设计，我可以明显感觉到语法分析器的设计实现明显难于词法分析器。在实验过程中遇到的主要问题是分析栈的构建以及如何构造一棵按照分析栈对字符的处理相对应的N叉树，最后还需要将N叉树的结点通过相应的结点关系与输出xml文件层次关系相对应。

完成了对语法分析器的设计，使我更加清楚地认识到编译程序如何按照文法规则识别并处理字符间的顺序关系，也让我认识到了一套好的文法定义将会为后续的语法语义处理带来很大的便利。这提醒我：一套好的理论可以为最后的设计提出最简单快捷的道路，同时，在设计过程中，我们也要不断找到方法去完善、实现这套理论。编写语法分析器让我意识到了编译程序的重要性与编写难度，也让我更加熟悉了语义处理文法的理论。