**《编译原理》课程**

**实验报告**

**题目 语法分析**

**专业 软件工程**

**班级 10065502**

**学号 1006840639**

**姓名 杨达一**

**指导教师 朱保平**

**南京理工大学计算机学院**

**2012年4月**

**实验2：语法分析**

|  |
| --- |
| **一、实验目的** |
| 1. 巩固对语法分析的基本功能和原理的认识。 2. 通过对语法分析表的自动生成加深语法分析表的认识。 3. 理解并处理语法分析中的异常和错误。 |
| **二、实验内容** |
| 要求：对如下工作进行展开描述  (1) 掌握语法分析程序的总体框架，并将其实现。  (2) 在手工构造的文法的基础上实现LR(1)分析，给出其语法分析表的生成程序(对应不同的语法分析方法产生不同的分析表) 及其数据结构和查找算法。  (3) 给出错误处理方法，错误的位置及类型等。  (4) 对一行字符串代码进行语法分析，并输出其语法分析结果。 |
| **三、实验结果** |
| 要求：将实验获得的结果进行描述，基本内容包括：   1. 针对某测试程序输出其语法法分析结果； 2. 输入文法产生式   允许从文件读入，并能对输入作简单的处理，比如混乱的空格，换行等     1. 计算文法基本信息      1. 计算项目集和项目集之间的转换关系      1. 构造分析表      1. 输入句子并分析      1. 输出针对此测试程序对应的语法错误报告；   用户在使用的时候主要会出现如下错误：   1. 输入产生式和待分析串格式不正确，对于这种错误，程序将进行简单的容错处理，如自动整理格式等。处理不了的，读入模块会抛出异常。 |
| **四、实验中遇到的问题总结** |
| 主要阐述两方面的问题   1. 实验过程中遇到的问题如何解决的？   关于First集的计算，在遇到A→B时，会把B的First集不包括空的元素添加到A的First中。  第一想法是用一个bool数组记录是否含有空，剩下的用set存着。函数First（x）用来计算x的First集，当遇到A→B时，在First（A）中调用First（B）。但是很快就发现了其中的问题，当出现A→B，B→A或直接出现A→A时显然会无穷递归下去。  之后想到了第二种方法，还是将空串单独考虑，将非终结符以及它们之间的关系抽象成图模型，非终结符看作图中的顶点，当遇到A→B这样的产生式时，添加有向边（B→A），表示他们之间的依赖关系，即要先求出B的First集才能求A的，当图中有A→B，B→A或直接出现A→A等产生式时，图中会出现环，成环的点会形成一个强连通分量，运用求强连通分量的算法将强连通分量缩成一个点，然后对缩点后的图进行拓扑排序，按照拓扑序计算First集，可以在线性时间类求得所有元素的First集。这个看似较好的方法还是有问题的。当有产生式A→BC，且B中不含空时，A只依赖于B，但是在计算之后，B可能有含有空了，此时A又会依赖于C，于是又要重新构图计算。  最后采用了一种较为朴素的算法，对所有产生式独立分析，若有A→B，则将First(B)添加到First(A)中，若B没有计算，则什么都不做，如果对所有产生式分析一轮之后有某个元素的First集发生了变化，则重复，知道所有元素的First集收敛。  在计算项目集的闭包时也采用了类似的算法。 |
| **五、实验体会** |
| 加深了对LR（1）分析过程的理解，提高了代码能力 |