**软件学院实践报告（编译原理）**

**课程编号： 实践课程名称：编译原理 学年：2019-2020 学期：秋**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **学生姓名** | | 吴志镛，陈凌云 | **学号** | 2017211869，2017211868 |
| **指导教师姓名** | | 修佳鹏 | 完成时间 | 2019年12月15日 |
| **实验名称** | | 语法分析器 | | |
| **实验目的** | 掌握自上而下语法分析方法、自下而上语法分析方法  构造LL和LR 分析程序，利用它进行语法分析，判断给出的符号串是否为该文法识别的句子； | | | |
| **实验要求** | * 1. 设计及实现能够识别表达式的LL预测分析程序。   文法如下：  G[E]：E->E+T|E-T|T  T->T\*F|T/F|F  F->(E)|i   * 1. 设计及实现能够识别表达式的LR分析程序。   文法如下：  G[E]：E->E+T|E-T|T  T->T\*F|T/F|F  F->(E)|i | | | |
| **实验环境** | Windows 10系统  IDEAL编译器  JAVA语言 JDK1.8 | | | |
| **人员分工** | 在我们两人共同商讨之后，我们决定要实验LL（1）和LR的语法分析器不单单针对于题目所给的文法，我们希望实现一个自推的分析预测程序，可以根据输入的不同文法然后自动推导出LL和LR分析表。于是我们分工如下：  吴志镛负责完成LL（1）分析表构造，first集和follow集自动推演程序的实验。  陈凌云负责完成SLR分析表的自动推演制作，以及LR和LL分析过程的自动程序是实现 | | | |
| **设计方案** | 基本实现思想：接收用户输入的字符串（字符串以“#”表示结束）后，对用做分析栈的一维数组和存放分析表的二维数组进行初始化。然后取出分析栈的栈顶字符，判断是否为终结符，若为终结符则判断是否为“#”且与当前输入符号一样，若是则语法分析结束，输入的字符串为文法的一个句子，否则出错若不为“#”且与当前输入符号一样则将栈顶符号出栈，当前输入符号从输入字符串中除去，进入下一个字符的分析。若不为“#”且不与当前输入符号一样，则出错。  LL(1)分析流程图    LR分析程序流程图 | | | |
| **成源程序** | 见附录 | | | |
| **实验结果** | 输入 i\*(i+i)表达式 | | | |
| **实验分析** | 自底向下的语法分析法。这种方法来分析的固定的文法，是可以凸现其通用性，虽然它没有算符优先分析方法效率那么高。但是由于算符优先分析法是将当前句型中的最左素短语而不是句柄（最左直接短语）作为规约的子串，省略了所有单非终极符产生式对应的归约步骤，因此在分析过程中，虽然采取了相应的一些检查措施，但仍然有可能将错误的输入符号串归约为正确的句子。而且很多的程序设计语言的文法也很难满足算符优先文法的条件，所以它只能局限在表达式的语法分析中。但是对于LR分析法，其分析的过程更加符合程序设计语言文法的要求，这样使之通用性加强 | | | |
| **心得体会** | 这次实验编写了一个语法分析方法的程序，是在LL（1）分析器和SLR分析器中的实现了可以根据文法自动生成是的select集，first集，follow集然后通过程序将预测分析表构造出来，然后通过总控程序根据分析表进行分析。  通过本次试验，我们设计两种比较完整的语法分析程序，实现对词法分析程序所提供的单词序列进行语法检查和结构分析，进一步掌握常用的语法分析方法。还能选择最有代表性的语法分析方法，如LL(1) 语法分析程序、算符优先分析程序和LR分析分析程序。 | | | |

程序附录：