**实验报告3**

2022 年 6 月4 日 成绩：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 姓名 | 孔佳玉 | 学号 | 20068704 | 班级 | 20052320 |
| 专业 | 计算机科学与技术 | | 课程名称 | 编译原理 | |
| 任课老师 | 王荣波 | 指导老师 | 王荣波 | 机位号 |  |
| 实验序号 | 4 / 6 | 实验名称 | LL(1)语法分析实验 / LL(1)文法判别方法 | | |
| 实验时间 | 8:30-11:50 | 实验地点 | 1教115 | 实验设备号 |  |
| **一、实验目的** | | | | | |
| 1. 理解LL(1)文法的判定方法，进一步掌握文法的改造原理（左递归消去、左公共因子提取），FIRST集、FOLLOW集的构造方法，预测分析表的的构造算法。 2. 了解 LL(1)语法分析是如何根据语法规则逐一分析词法分析所得到的单词，检查语法错误，即掌握语法分析过程。 3. 掌握LL(1)语法分析器的设计与调试。 | | | | | |
| **二、实验内容** | | | | | |
| 一、实验4要求  针对SysY语言中简单算术表达式文法G[E]:  E→TE’  E’→ATE’|ε  T→FT’  T’→MFT’ |ε  F→(E) | i  A → + | -  M → \* | /  求解相应的FIRST、FOLLOW集，构造预测分析表，并编写LL(1)语法分析程序，并给出测试句子的分析过程。（注：如果有选做专题6关于LL(1)文法判断的同学，可以将专题6的部分整合到这个实验的前面，自动产生预测分析表，相当于把这个程序做成一个通用的LL(1)分析器）  1. 输入：是词法分析输出的二元组序列，即任意简单算术表达式经过专题1程序输出后得到的结果。【上述文法中i即对应词法分析的标识符， +-\*/分别对应词法分析得到的运算符】  2. 处理：基于分析表进行 LL(1)语法分析，判断其是否符合文法。  3. 输出：串是否合法。  二、实验6要求  针对任意的文法，编写相应的左递归消除、左公共因子提取程序，求解相应的FIRST、FOLLOW集，构造预测分析表，并结合专题4的LL(1)语法分析程序，并给出测试句子的分析过程。  判断LL(1)文法部分：   1. 输入：文法 2. 处理：左递归消除、左公共因子提取，FIRST、FOLLOW等集合构造，判断LL(1) 3. 输出：是LL(1)的情况输出预测分析表，否则判断不是LL(1) | | | | | |
| **三、实验步骤** | | | | | |
| 1. 按照实验要求明确实验思路； 2. 作出程序思维流程图（如上）； 3. 创建各程序部分主函数、常量、各类辅助函数； 4. 完成各函数体内部逻辑代码的编写； 5. 修改并测试。 | | | | | |
| **四、碰到的问题及解决办法** | | | | | |
| 实验四：  在分析文法正确性时，不知道如何动态地显示分析过程  在编写程序时，定义了一个栈来存储待分析的标识符，每次弹出栈顶标识符，利用产生式进行推导，将产生式右边的标识符存入栈中，直到找到匹配当前待匹配串的第一个字符或空串。如此将当前串所有字符逐个匹配完成后结束分析。若期间发现推导到最后没有找到对应的标识符，则输出文法错误的提示并结束程序。  实验六：  由于文法并未被限定，而是由用户给出，所以不能像之前的递归程序一样按照文法特点编写程序，而要抽象出一种求FRIST和FOLLOW集的通用方法  首先遍历输入程序，抽取出文法中所有的标记符作为集合。再根据FIRST和FOLLOW集的定义，利用递归求文法中所有非终结符可能会推导出的第一个字符构成该文法的FRIST集；利用FOLLOW集的定义，分三种情况：找出文法中产生式右边有某字符出现的地方，看字符右边有无其他字符。若存在其他字符且该字符的FRIST集不存在ε，则将该字符右边的字符的FRIST集加入该FOLLOW集；否则将产生式左边的字符的FOLLOW集加入该字符的FOLLOW集，由此生成FRIST集和FOLLOW集。预测分析表的生成中，首先分别将非终结符和终结符作为表的行和列，并将内容初始化为空，再利用FRIST集和FOLLOW集求得结果向分析表插入生成式，即可输出预测分析表 | | | | | |
| **五、测试分析** | | | | | |
| 输入字符串基于文法：{ "E->TX", "X->ATX|ε", "T->FY", "Y->MFY|ε", "F->(E)|i", "A->+|-", "M->\*|/" } 进行翻译   1. 正常输入 2. "i+i\*i#"   Snipaste_2022-06-03_15-29-17  Snipaste_2022-06-03_15-35-18   1. "(i+i)/i#"   Snipaste_2022-06-03_15-29-46   1. "i+(i\*i)#"   Snipaste_2022-06-03_15-36-59   1. 错误输入 2. "i%+i\*i#"   Snipaste_2022-06-03_15-30-27  Snipaste_2022-06-03_15-31-19   1. ""   Snipaste_2022-06-03_15-33-35  由图中的测试结果可以看到，本LL1分析可以根据文法生成其FIRST集和FOLLOW集与LL1预测分析表，并且能动态展示其分析过程中如何根据文法对串中字符逐个进行匹配，并且测试用例涵盖了空串和不符合LL1文法的字符串，均能被程序捕获到问题 | | | | | |
| **六、总结体会** | | | | | |
| 通过LL1文法的语法分析和文法判定实验，我深化了对求解FIRST集和FOLLOW集、以及绘制预测分析表的流程，并且能够构造一个算法来普适化求解某一文法的FIRST集和FOLLOW集，理解了LL1语法是如何按照文法来分析词法分析得到的单词之间的联系是否符合文法，更大程度上掌握了语法分析的过程 | | | | | |