**北京邮电大学《编译原理与技术》课程实验报告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **实验**  **名称** | 语法分析程序的设计与实现 | | **学 院** | 计算机学院 | **指导教师** |  |
| **班 级** | **班内序号** | **学 号** | | **学生姓名** | **成绩** | |
| 2018211305 | 27 | 2018211279 | | 索政铎 |  | |
|  |  |  | |  |  | |
|  |  |  | |  |  | |
| **实**  **验**  **内**  **容** | 利用所学词法分析原理，自行设计实现一个简易的语法分析程序，实现对算术表达式的语法分析。 | | | | | |
| **学生实验报告** | （详见“实验报告和源程序”册） | | | | | |
| **课**  **程**  **设**  **计**  **成**  **绩**  **评**  **定** | **评语**:  **成绩**:  指导教师签名：  年 月 日 | | | | | |

注：评语要体现每个学生的工作情况，可以加页。

1. **问题描述**

编写语法分析程序，实现对算术表达式的语法分析。要求所分析算术表达式由如下的文法产生：

要求在对输入的算术表达式进行分析的过程中，依次输出所采用的的产生式。

1. **LL(1)文法**

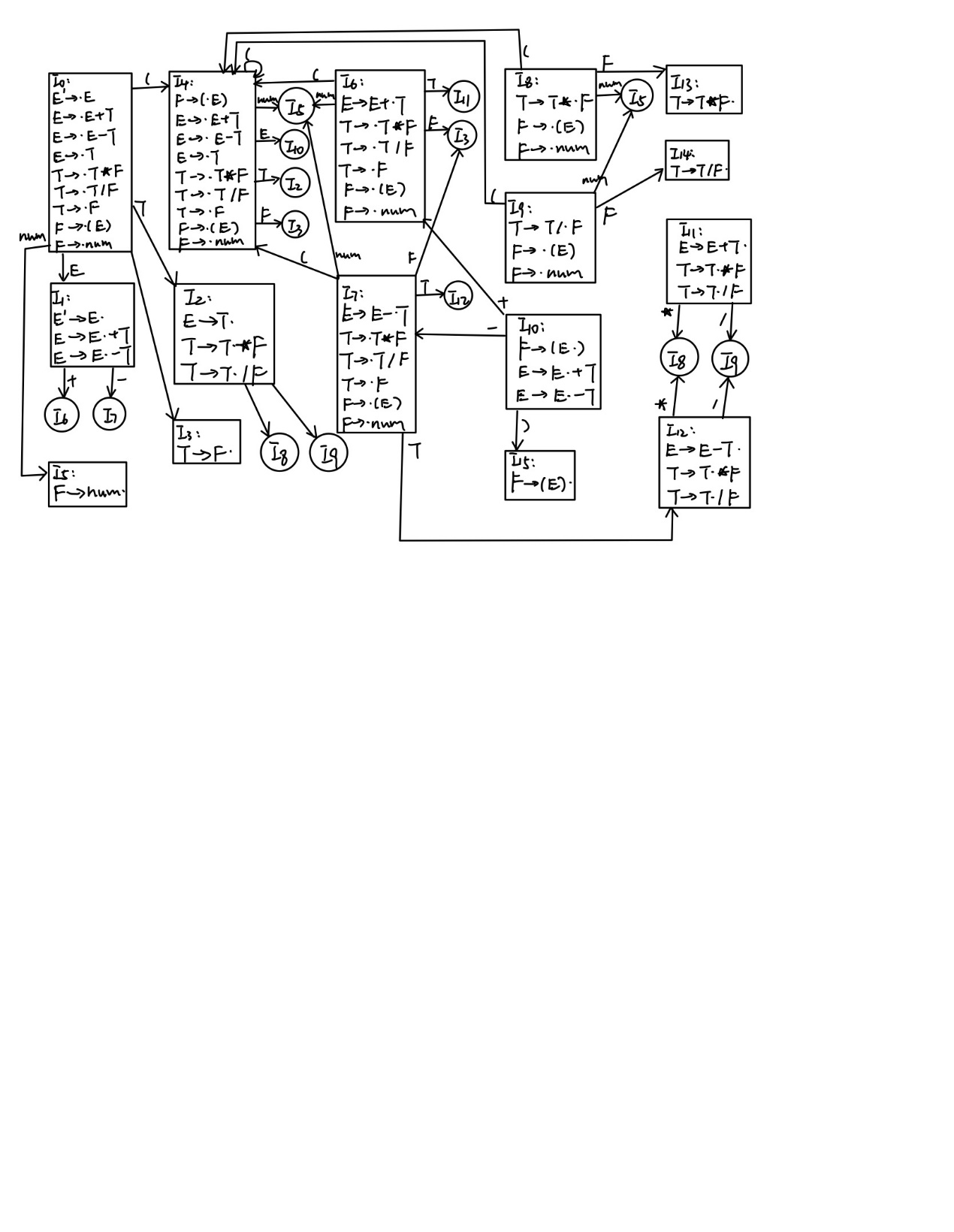
对原文法消除左递归，结果如下：

实现思路：实现求解FIRST集、FOLLOW集的函数getFirst()、getFollow()，读取文法产生式，根据构造的FIRST集、FOLLOW集实现构造分析表的函数generateTable()。最后，读入待分析的字符串，实现splitTerminal()函数将其分解为终结符的形式（主要将数字转换为num），通过analyse()函数，借助栈对终结符序列进行分析。

1. **LR文法**

对原文法进行拓展文法，结果如下：

根据文法画出LR(1)，并合并得到LALR(1)，且无冲突，项目集DFA如下：



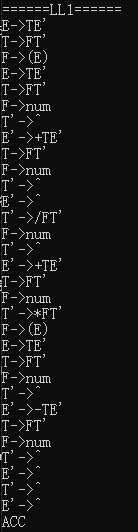
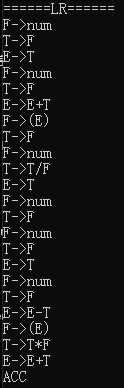
由图可得分析表如下：

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **ACTION** | | | | | | | | **GOTO** | | | |
|  | **$** | **(** | **)** | **+** | **-** | **\*** | **/** | **num** | **E’** | **E** | **T** | **F** |
| **0** |  | S4 |  |  |  |  |  | S5 |  | 1 | 2 | 3 |
| **1** | ACC |  |  | S6 | S7 |  |  |  |  |  |  |  |
| **2** | R3 |  | R3 | R3 | R3 | S8 | S9 |  |  |  |  |  |
| **3** | R6 |  | R6 | R6 | R6 | R6 | R6 |  |  |  |  |  |
| **4** |  | S4 |  |  |  |  |  | S5 |  | 10 | 2 | 3 |
| **5** | R8 |  | R8 | R8 | R8 | R8 | R8 |  |  |  |  |  |
| **6** |  | S4 |  |  |  |  |  | S5 |  |  | 11 | 3 |
| **7** |  | S4 |  |  |  |  |  | S5 |  |  | 12 | 3 |
| **8** |  | S4 |  |  |  |  |  | S5 |  |  |  | 13 |
| **9** |  | S4 |  |  |  |  |  | S5 |  |  |  | 14 |
| **10** |  |  | S15 | S6 | S7 |  |  |  |  |  |  |  |
| **11** | R1 |  | R1 | R1 | R1 | S8 | S9 |  |  |  |  |  |
| **12** | R2 |  | R2 | R2 | R2 | S8 | S9 |  |  |  |  |  |
| **13** | R4 |  | R4 | R4 | R4 | R4 | R4 |  |  |  |  |  |
| **14** | R5 |  | R5 | R5 | R5 | R5 | R5 |  |  |  |  |  |
| **15** | R7 |  | R7 | R7 | R7 | R7 | R7 |  |  |  |  |  |

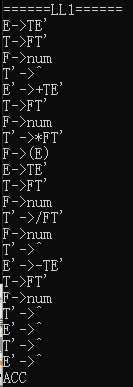
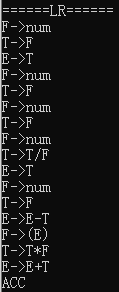
实现思路：将分析表内容通过generateTable()函数导入。读入待分析的字符串，实现splitTerminal()函数将其分解为终结符的形式（主要将数字转换为num），通过analyse()函数，借助栈（状态栈、符号栈）对终结符序列进行分析。

1. **测试情况**

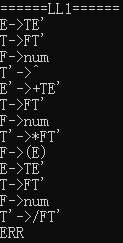
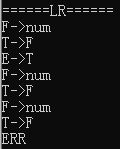
分析字符串：(1+1)/2+3\*(4-4)

分析字符串：1.2E3+4.56\*(77.889/0.001-2)

分析错误串：1-2\*(3+)

1. **总结与感悟**

通过本次语法分析的程序实现，巩固了关于LL(1)与LR分析流程的内容，熟悉了分析过程中各种需要注意的细节，例如在寻找FIRST集时，首字符为空时需要继续寻找余下式子的FIRST集等等。使得我对自下而上、自上而下的两种典型语法分析模式有了更深入的理解。

另一方面，两种语法分析都采用面向对象的方式进行实现，模块化明显，较为容易进行调整和改进。

非常感谢老师的实验设计！

1. **源代码**

另附