语法分析

班级：2014211306

姓名：徐翥

学号：2014211285

1. 题目

题目：语法分析程序的设计和实现

实验内容：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。

E->E+T|E-T|T

T->T\*F|T/F|F

F->(E)|num

1. LL(1)语法分析程序
2. 改写文法

E->TE’

E’->+TE’|-TE’|epsilon

T->FT’

T’->\*FT’|/FT’|epsilon

F->(E)|num

1. 预测分析表

算法：

输入：改写的文法

输出：预测分析表

方法：

For（文法A的每个产生式A->alpha） ｛

For（每个终结符号aFIRST(alpha)）

把A->alpha放入M[A,a]中；

If（epsilonFIRST(alpha)）

For（任何bFOLLOW(A)）

把A->放入M[A,b]中

｝

1. 非递归预测分析方法

算法：

输入：输入符号串w，预测分析表M

输出：若wL(G)，则输出w的最左推导，否则报告错误

方法：

Do｛

令X是栈顶文法符号，a是ip所指向的输入符号

If（X是终结符号或$）

If（X == a）

从栈顶弹出X，ip前移一个位置

Else error();

Else

If（M[X,a]=X->Y1Y2Y3…YK）｛

从栈顶弹出X

把YK…Y2Y1压人栈顶，Y1在栈顶

输出产生式X->Y1Y2…YK

｝

Else error();

｝while(X != $)

1. LL(1)分析控制程序

|  |  |
| --- | --- |
| ParserTable.py | 构造预测分析表 |
| Controller.py | 预测分析控制程序 |
| ParserMachine.py | LL(1)分析器 |

使用说明：

在Linux系统下运行：

python ParserMachine.py input\_string

1. ParserTable.py

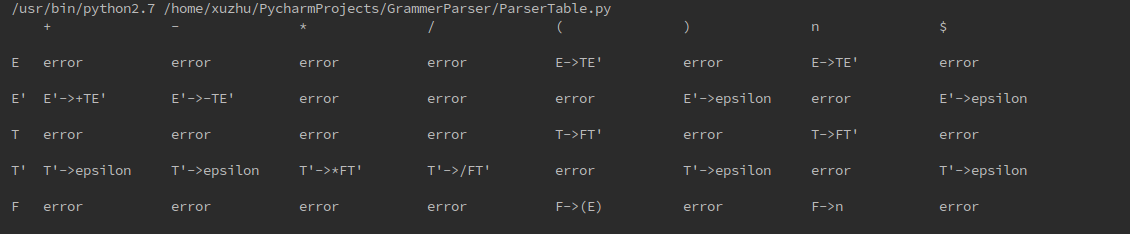
|  |  |
| --- | --- |
| self.first\_set | FIRST集 |
| self.follow\_set | FOLLOW集 |
| self.rule | 改写的文法，n表示num |
| self.table | 初始化一张空表 |

用n表示num

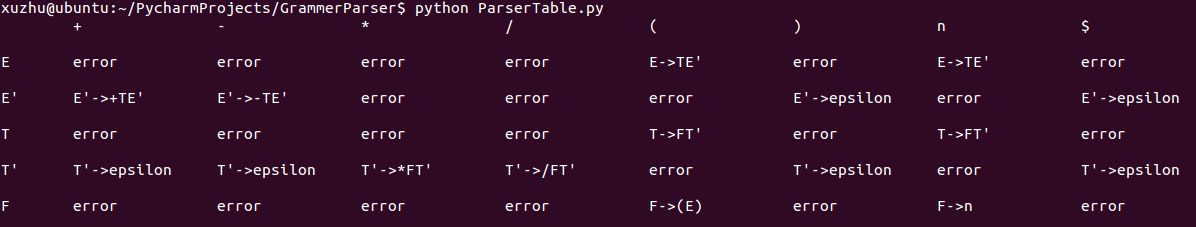
采用算法4.2，在课本92页

使用python ParserTable.py 命令可以构造预测分析表

在pycharm中的运行情况：



在终端中的运行情况：



1. Controller.py

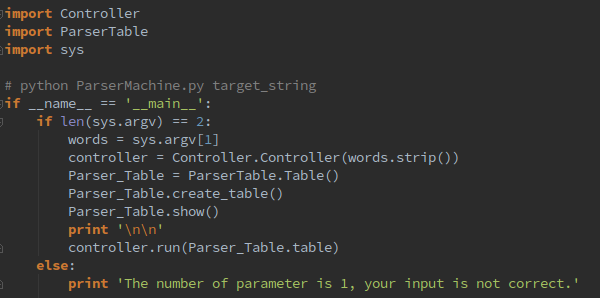
|  |  |
| --- | --- |
| self.stack | 栈 |
| self.input\_stack | 输入缓冲区 |
| self.words | 保存输出串的反向切片 |

在将串提交给run函数前，完成预测分析控制器的初始化，包括栈和输入缓冲区的初始化，使用正则表达式将串中的数全部替换为n。

采用算法4.1，课本89页。用run函数实现。

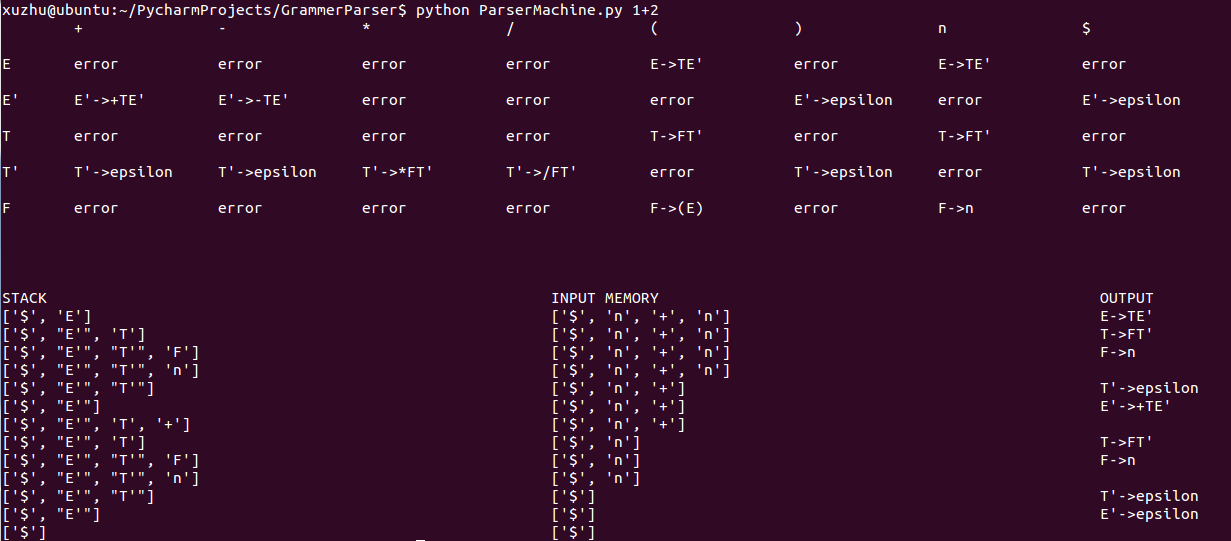
1. ParserMachine.py

封装预测分析表和分析控制程序

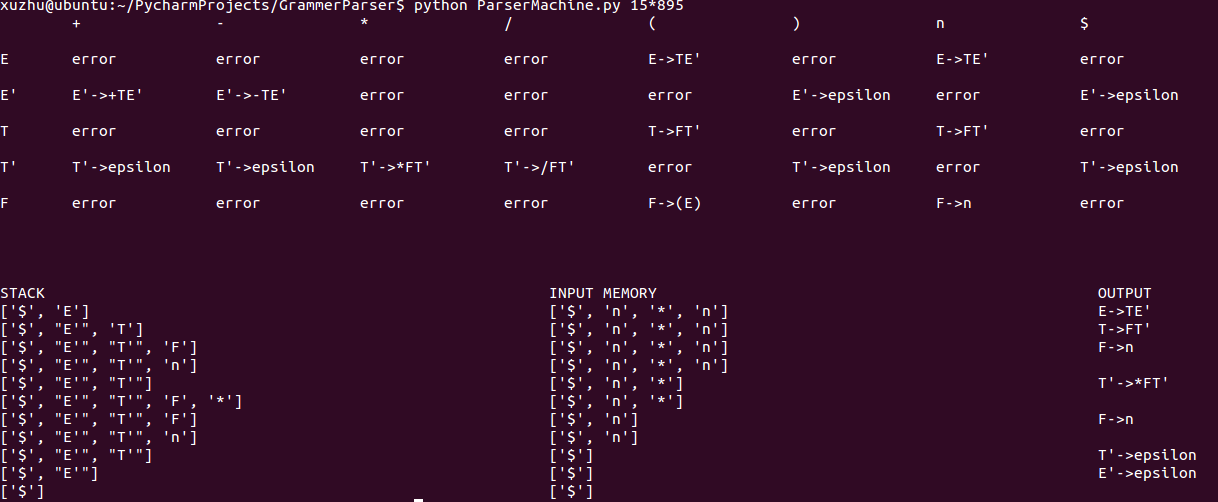


1. 运行实例

对1+2分析：



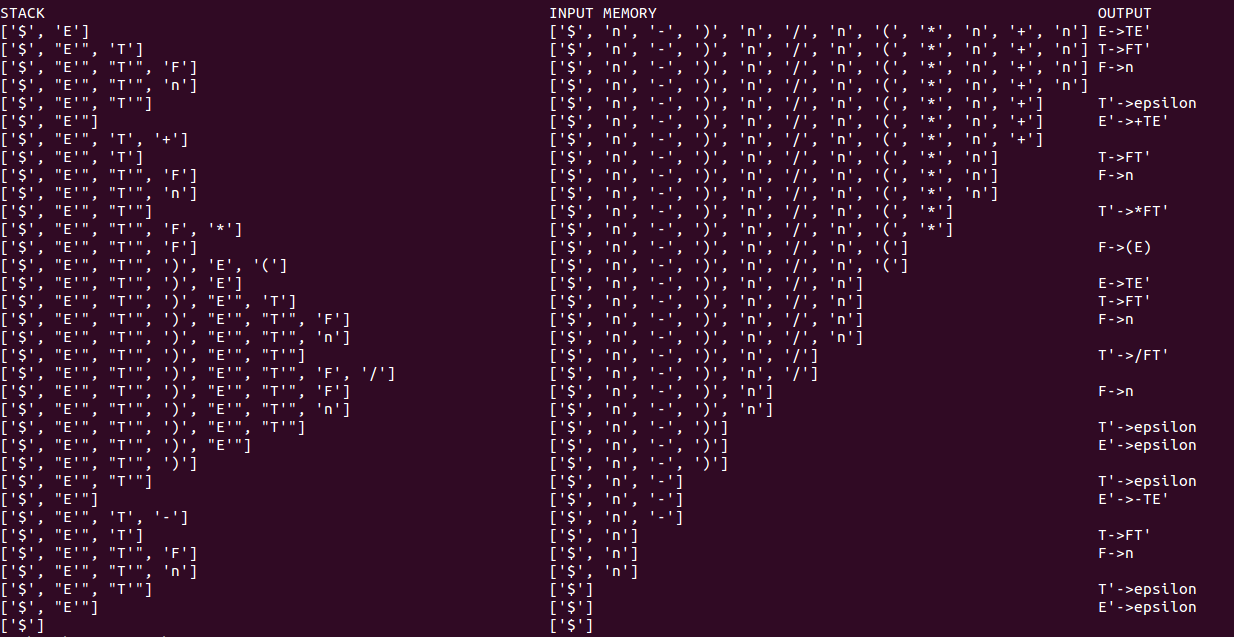
对15\*895分析：



对复杂串的分析：2013+2014\*(2016/2015)-2017

**注意：在linux shell中输入参数时，左括号和右括号的输入必须使用转义。比如 \(, \)**

由于预测分析表都一样，所以本次分析没有对预测分析表进行截图



对1937-2016\*(2008/(2013+2014))的分析：

**注意：在linux shell中输入参数时，左括号和右括号的输入必须使用转义。比如 \(, \)**



对小数也适用：1.2\*(1.33+2.16)

**注意：在linux shell中输入参数时，左括号和右括号的输入必须使用转义。比如 \(, \)**

