说明文档

1900094619 元培 金镇雄

1. 使用语言：python
2. EBNF描述

   <赋值> → <标识符>（ = | += | -= | *= | /= | %= ）<表达式>

   <表达式> → <项> {（ + | - ）<项> }

   <项> → <因子> {（ * | / | % ）<因子> }

   <因子> → 标识符 | 整数常量 |（ <表达式> ）
3. 词法分析

   可分析的标记有：标识符、整数字面值、赋值运算符（ = | += | -= | *= | /= | %= ）、加减乘除运算符以及余数运算符。

   对应的token如下：

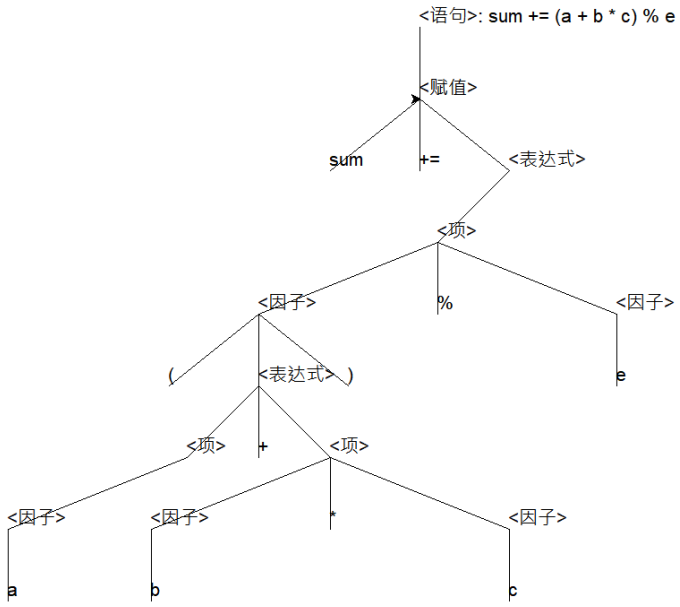
```
INT_LIT = 10
IDENT = 11

ASSIGN_OP = 20
ADD_OP = 21
SUB_OP = 22
MUL_OP = 23
DIV_OP = 24
REM_OP = 25

LEFT_PAREN = 25
RIGHT_PAREN = 26

ADD_ASSIGN = 30
SUB_ASSIGN = 31
MUL_ASSIGN = 32
DIV_ASSIGN = 33
REM_ASSIGN = 34
```

4. 输入：共有两行。第一行为语句（字符串），第二行为输入语句的类型

```
输入语句：sum += (a + b * c) % e
输入语句类型编号：1 - factor / 2 - term / 3 - expr / 4 - assign
4
```

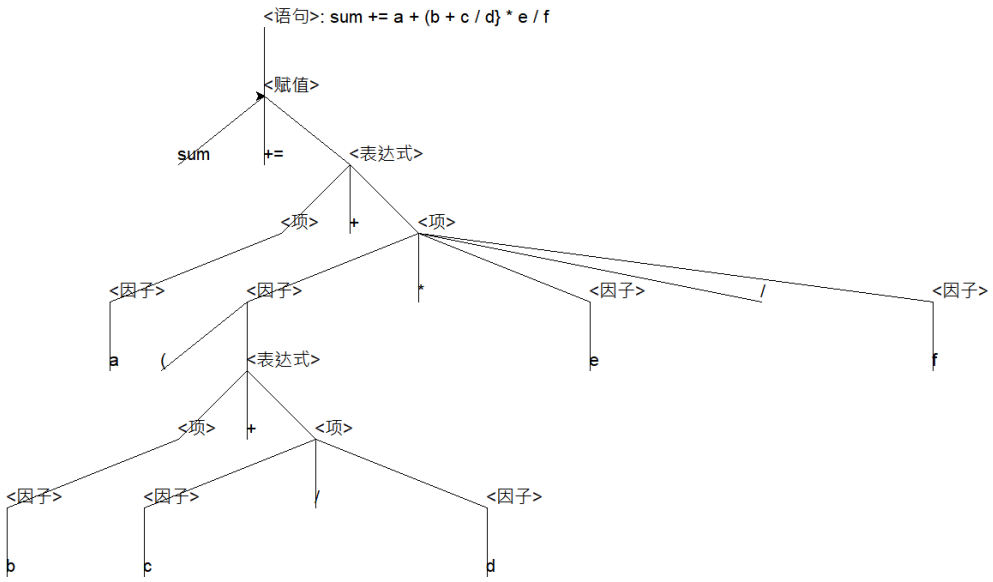
5. 输出：语法分析树（语法分析器）和获取输入中的下一个词素及标记（词法分析器）

   下面是输入语句为“sum += (a + b * c) % e“ 时对应的语法分析树和词法分析器的输出




```
next Token is 11, next Lexeme is sum
进入<赋值>
next Token is 30, next Lexeme is +=
next Token is 25, next Lexeme is (
进入<表达式>
进入<项>
进入<因子>
next Token is 11, next Lexeme is a
进入<表达式>
进入<项>
进入<因子>
next Token is 21, next Lexeme is +
退出<因子>
退出<项>
next Token is 11, next Lexeme is b
进入<项>
进入<因子>
next Token is 23, next Lexeme is *
退出<因子>
next Token is 11, next Lexeme is c
进入<因子>
next Token is 26, next Lexeme is )
退出<因子>
退出<项>
退出<表达式>
next Token is 25, next Lexeme is %
退出<因子>
next Token is 11, next Lexeme is e
进入<因子>
next Token is -1, next Lexeme is EOF
退出<因子>
退出<项>
退出<表达式>
退出<赋值>
```

6. 输入语句有语法错误时：sum += a + (b + c / d} * e / f



语法分析器发现没有右括号")"时，会输出“error: Token Isn't RIGHT_PAREN“

```
next Token is 11, next Lexeme is sum
进入<赋值>
next Token is 30, next Lexeme is +=
next Token is 11, next Lexeme is a
进入<表达式>
进入<项>
退出<因子>
退出<项>
next Token is 11, next Lexeme is c
进入<项>
进入<因子>
next Token is 24, next Lexeme is /
退出<因子>
next Token is 11, next Lexeme is d
进入<因子>
next Token is -1, next Lexeme is }
退出<因子>
退出<项>
退出<表达式>
error: Token Isn't RIGHT_PAREN
next Token is 23, next Lexeme is *
退出<因子>
next Token is 11, next Lexeme is e
进入<因子>
next Token is 24, next Lexeme is /
退出<因子>
next Token is 11, next Lexeme is f
进入<因子>
next Token is -1, next Lexeme is EOF
退出<因子>
退出<项>
退出<表达式>
退出<赋值>
```