《编译原理与设计》

语义分析程序的设计与实现

实验报告

班级	2014211304
学号	2014210336
姓名	杨炫越
日期	2016.11.14

1. 实验内容

编写语义分析和翻译程序，实现对算术表达式的类型检查和求值。要求所分析算术表达式由如下的文法产生：

$E \rightarrow E+T \mid E-T \mid T$

$T \rightarrow T*F \mid T/F \mid F$

$^{*}F \rightarrow$ int_num | real_num | (E)

*注：课本原题中该产生式为 $F \rightarrow$ num | num.num | (E)，本程序将 real 类型的数值 num.num **改为**在**词法分析**阶段进行处理，在语法制导翻译阶段可直接获得一个 real 类型的 token，更为合理。

2. 实验要求

用自底向上的语法制导翻译技术实现对表达式的分析和翻译。

(1) 写出满足要求的语法制导定义或翻译方案。

(2) 编写语义分析和翻译程序，实现对表达式的类型进行检查和求值，并输出：

a. 分析过程中所用产生式；

b. 识别出的子表达式的类型；

c. 识别出的子表达式的值。

3. 开发环境

操作系统：Microsoft Windows 10.0.14393 (x64)

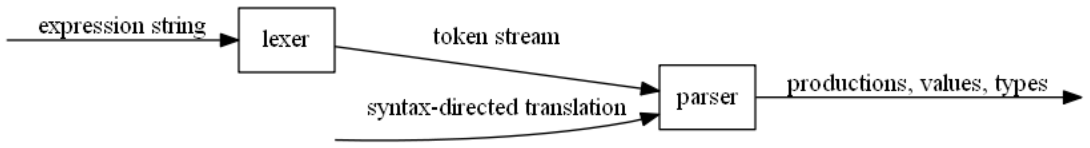
IDE：Microsoft Visual Studio Community 2015

编译器：MSVC++ 14.0

附加库：Boost 1.62.0

4. 设计思路

本语义分析程序在**上次实验实现的 LR 语法分析程序**的基础上进行语法制导翻译。流程如下：



对表达式的求值及类型检查可由 S 属性定义完成，对每个非终结符赋予综合属性 *val* 与 *type* 分别代表其值与类型，语法制导定义及翻译代码如下：

产生式	语义规则	翻译函数（代码段）

$E \rightarrow E_1+T$	$E.val = E_1.val + T.val$
	$E.type \rightarrow E_1.type \wedge T.type^*$
		`val[nt] = val[t - 2] + val[t]`
		`type[nt] = get_bin_op_res_type(`
		`    type[t - 2], type[t])`

$E \rightarrow E_1–T$	$E.val = E_1.val - T.val$
	$E.type \rightarrow E_1.type \wedge T.type$
		`val[nt] = val[t - 2] - val[t]`
		`type[nt] = get_bin_op_res_type(`
		`    type[t - 2], type[t])`

$E \rightarrow T$	$E.val = T.val$ $E.type \rightarrow T.type$	

$T \rightarrow T_1*F$	$T.val = T_1.val * F.val$
	$T.type \rightarrow T_1.type \wedge F.type$
		`val[nt] = val[t - 2] * val[t]`
		`type[nt] = get_bin_op_res_type(`
		`    type[t - 2], type[t])`

$T \rightarrow T_1/F$	$T.val = T_1.val / F.val$
	$T.type \rightarrow T_1.type \wedge F.type$
		`val[nt] = val[t - 2] / val[t]`
		`type[nt] = get_bin_op_res_type(`
		`    type[t - 2], type[t])`

$T \rightarrow F$	$T.val = F.val$ $E.type \rightarrow F.type$	
$F \rightarrow$ int_num	$F.val =$ int_num.*lexval()* $F.type \rightarrow$ int	
$F \rightarrow$ real_num	$F.val =$ real_num.*lexval()* $F.type \rightarrow$ real	

$F \rightarrow (E)$	$F.val = E.val$
	$F.type \rightarrow E.type$
		`val[nt] = val[t - 1]`
		`type[nt] = type[t - 1]`

*注：∧为自定义类型运算符，其两端类型有一为 real 时，结果为 real，否则为 integer。右边自定义函数 `get_bin_op_res_type()` 完成相同作用。

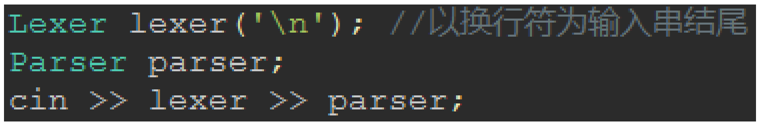
5. 程序实现

源码：

Local：/src_code

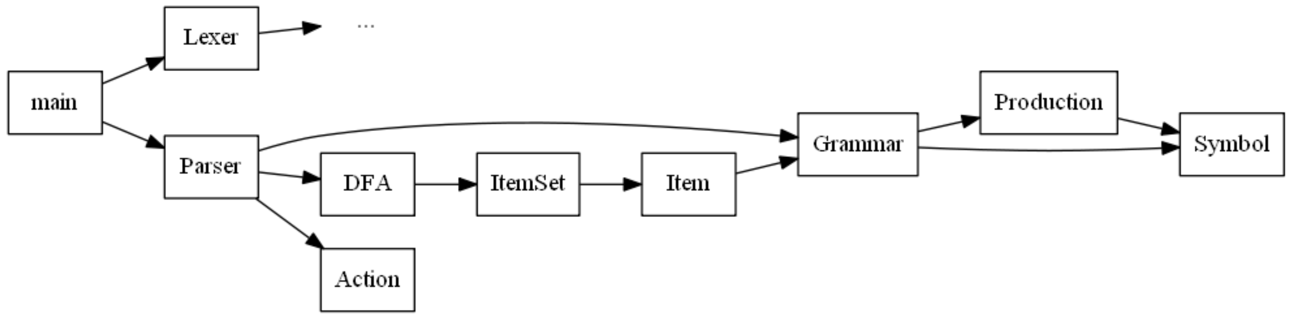
Online：https://github.com/YangXuanyue/Compiler

本语义分析程序采用了 C++来编写，由于基于语法制导翻译技术，主体为一个语法分析器。程序中实现了一个`Parser`类作为语法分析器的对外接口，并为其重载了输入流操作符，可与之前实验实现的词法分析类`Lexer`连接如下：



```
Lexer lexer('\n'); //以换行符为输入串结尾
Parser parser;
cin >> lexer >> parser;
```

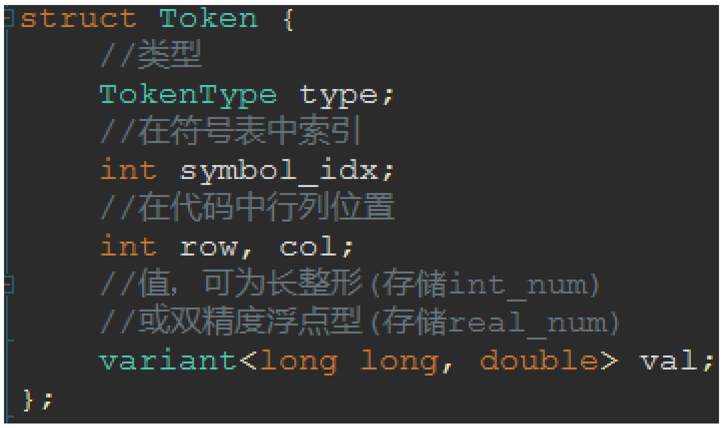
本程序的结构示意图如下所示：



其结构与上次实验实现的 LR 语法分析程序大致相同，在其基础上新增语法制导翻译的功能，有更新的模块如下：

5.1. Token

该模块定义了 Token 结构体如下：



```
struct Token {
    //类型
    TokenType type;
    //在符号表中索引
    int symbol_idx;
    //在代码中行列位置
    int row, col;
    //值，可为长整形(存储int_num)
    //或双精度浮点型(存储real_num)
    variant<long long, double> val;
};
```

在之前的基础上增加了一个`val`域记录一个 token 的值，缺省为 0.0。当该 token 类型为整型常数`INT_CONSTANT`，即 integer 时，`val`为一个`long long`型数据，其值等于该 token 对应符号表项的数值；当该 token 类型为实型常数`REAL_CONSTANT`，即 real 时，`val`为一个`double`型数据，其值等于该 token 对应符号表项的数值。

5.2. Production

该模块定义了产生式结构体如下：

```
enum { //LrStackItem中各项信息
    STATE_IDX, //状态编号
    SYMBOL, //文法符号
    VAL, //值
    TYPE //类型
};
//LR分析栈内容
typedef tuple<int, Symbol, variant<long long, double>, TokenType> LrStackItem;
//对LR分析栈进行操作的翻译函数
typedef function<void(deque<LrStackItem>&)> Translate;

struct Production {
    //左部文法符号
    Symbol left;
    //右部文法符号串
    deque<Symbol> right;
    //该产生式归约时的翻译函数
    Translate translate;
};
```

在之前基础上为每个产生式增加了一个归约时进行的翻译函数`translate`，用 C++的 STL 中的函数模板`std::function`实现，使得其可在外部进行初始化（见 5.3. Grammar）。因本程序中采用的语法制导定义对分析栈进行操作，该函数以一个 LR 语法分析栈的引用作为参数。

5.3. Grammar

该模块定义了文法类。在之前的基础上增加了对产生式对应的翻译函数的初始化，采用了 C++的 **lambda 表达式**。

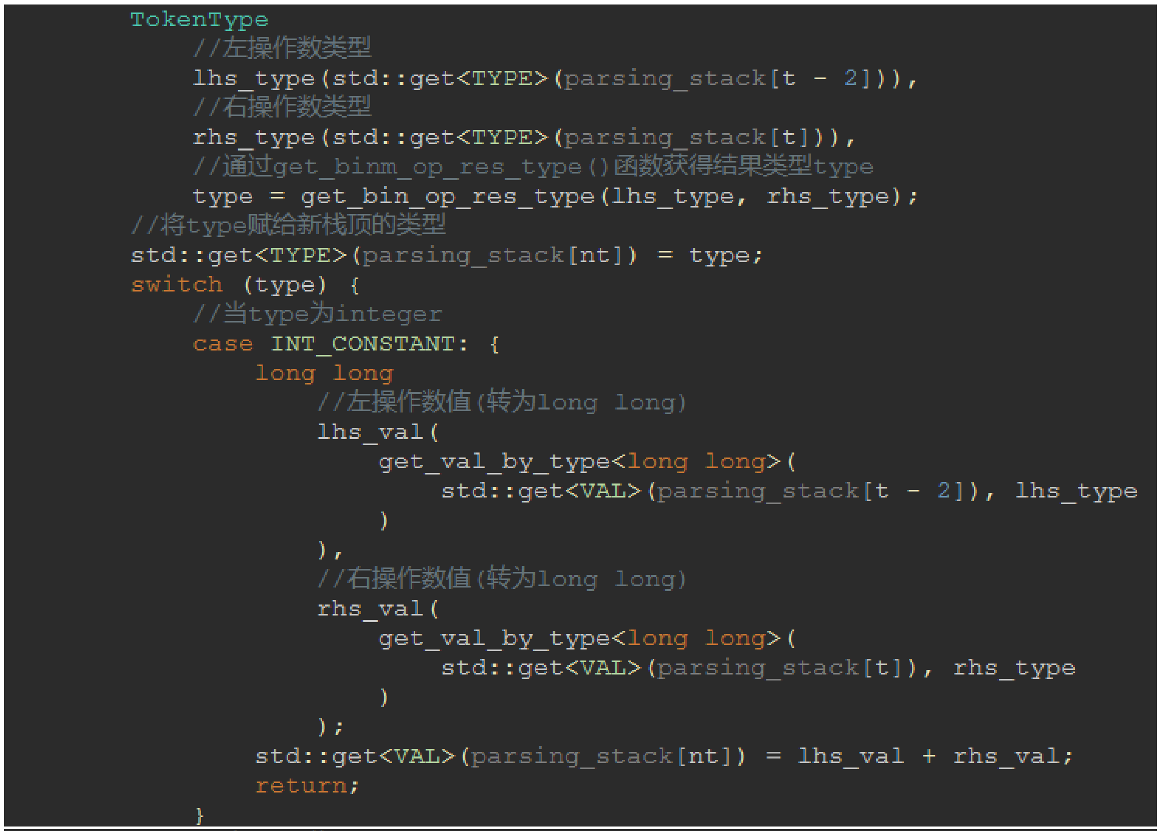
以第一个产生式及其语义规则为例：

产生式	语义规则	翻译函数（代码段）

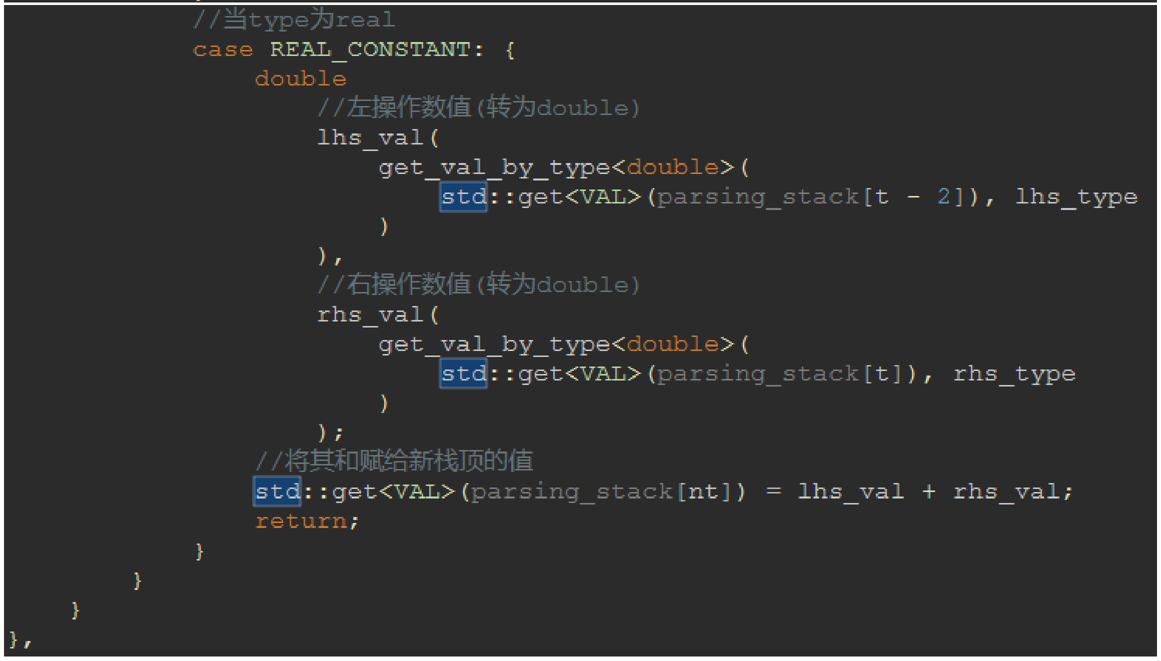
$E \rightarrow E_1+T$	$E.val = E_1.val + T.val$
	$E.type \rightarrow E_1.type \wedge T.type$
		`val[nt] = val[t - 2] + val[t]`
		`type[nt] = get_bin_op_res_type(`
		`    type[t - 2], type[t])`

初始化代码为：

```
{
    //产生式左部及右部
    "E", {"E", ADD, "T"},
    //使用lambda表达式初始化的translate函数
    [](deque<LrStackItem>& parsing_stack) {
        int
            //原栈顶位置
            t(parsing_stack.size() - 1),
            //新栈顶位置
            nt(parsing_stack.size() - 3);
```



```
        TokenType
            //左操作数类型
            lhs_type(std::get<TYPE>(parsing_stack[t - 2])),
            //右操作数类型
            rhs_type(std::get<TYPE>(parsing_stack[t])),
            //通过get_binm_op_res_type()函数获得结果类型type
            type = get_bin_op_res_type(lhs_type, rhs_type);
        //将type赋给新栈顶的类型
        std::get<TYPE>(parsing_stack[nt]) = type;
        switch (type) {
            //当type为integer
            case INT_CONSTANT: {
                long long
                    //左操作数值(转为long long)
                    lhs_val(
                        get_val_by_type<long long>(
                            std::get<VAL>(parsing_stack[t - 2]), lhs_type
                        )
                    ),
                    //右操作数值(转为long long)
                    rhs_val(
                        get_val_by_type<long long>(
                            std::get<VAL>(parsing_stack[t]), rhs_type
                        )
                    );
                std::get<VAL>(parsing_stack[nt]) = lhs_val + rhs_val;
                return;
            }
```

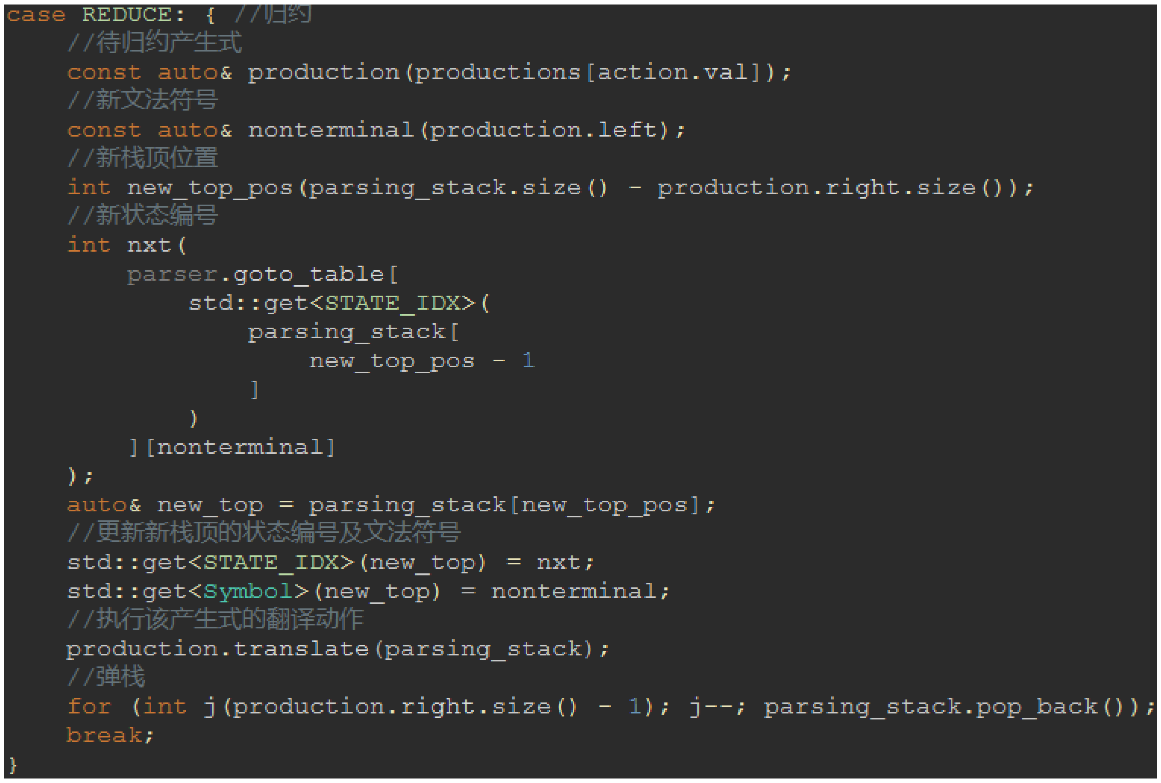


```
            //当type为real
            case REAL_CONSTANT: {
                double
                    //左操作数值(转为double)
                    lhs_val(
                        get_val_by_type<double>(
                            std::get<VAL>(parsing_stack[t - 2]), lhs_type
                        )
                    ),
                    //右操作数值(转为double)
                    rhs_val(
                        get_val_by_type<double>(
                            std::get<VAL>(parsing_stack[t]), rhs_type
                        )
                    );
                //将其和赋给新栈顶的值
                std::get<VAL>(parsing_stack[nt]) = lhs_val + rhs_val;
                return;
            }
        }
    }
},
```

5.4. Parser

该模块定义了 LR 语法分析器类。在之前的基础上，分析栈被扩充为一个四元组栈，带有状态编号、文法符号、值与类型四个信息，并在进行产生

式的归约时增加了求值与类型检查的翻译动作，由产生式的`translate`函数实现。实现如下：



```
case REDUCE: { //归约
    //待归约产生式
    const auto& production(productions[action.val]);
    //新文法符号
    const auto& nonterminal(production.left);
    //新栈顶位置
    int new_top_pos(parsing_stack.size() - production.right.size());
    //新状态编号
    int nxt(
        parser.goto_table[
            std::get<STATE_IDX>(
                parsing_stack[
                    new_top_pos - 1
                ]
            )
        ][nonterminal]
    );
    auto& new_top = parsing_stack[new_top_pos];
    //更新新栈顶的状态编号及文法符号
    std::get<STATE_IDX>(new_top) = nxt;
    std::get<Symbol>(new_top) = nonterminal;
    //执行该产生式的翻译动作
    production.translate(parsing_stack);
    //弹栈
    for (int j(production.right.size() - 1); j--; parsing_stack.pop_back());
    break;
}
```

6. 程序测试

测试样例格式如下：

```
（输入算术表达式）
--------------------------------------------------------------
current parsing stack:
                          （当前分析栈）
current token stream:
                          （当前待扫描 token 流） 注：已经词法分析将符号串转为 token 流
output:
                          （分析动作：移进或规约为某个产生式）
current subexpression: 注：当进行归约动作时
                          （当前子表达式）
                          type = （类型）
                          value = （值）
```

以下为测试样例：

```
1
--------------------------------------------------------------
current parsing stack:
```

```
                              [0 end]
current token stream:
                              int_num end
output:
                              shift 5

current parsing stack:
                              [0 end] [5 int_num]
current token stream:
                              end
output:
                              reduce F -> int_num
current subexpression:
                              1
                              type = INT_CONSTANT
                              value = 1

current parsing stack:
                              [0 end] [2 F]
current token stream:
                              end
output:
                              reduce T -> F
current subexpression:
                              1
                              type = INT_CONSTANT
                              value = 1

current parsing stack:
                              [0 end] [3 T]
current token stream:
                              end
output:
                              reduce E -> T
current subexpression:
                              1
                              type = INT_CONSTANT
                              value = 1

current parsing stack:
                              [0 end] [1 E]
current token stream:
                              end
output:
                              acc
```

```
1+1
current parsing stack:
                              [0 end]
current token stream:
                              int_num + int_num end
output:
                              shift 5

current parsing stack:
```

```
                                [0 end] [5 int_num]
current token stream:
                                + int_num end
output:
                                reduce F -> int_num
current subexpression:
                                1
                                type = INT_CONSTANT
                                value = 1

current parsing stack:
                                [0 end] [2 F]
current token stream:
                                + int_num end
output:
                                reduce T -> F
current subexpression:
                                1
                                type = INT_CONSTANT
                                value = 1

current parsing stack:
                                [0 end] [3 T]
current token stream:
                                + int_num end
output:
                                reduce E -> T
current subexpression:
                                1
                                type = INT_CONSTANT
                                value = 1

current parsing stack:
                                [0 end] [1 E]
current token stream:
                                + int_num end
output:
                                shift 7

current parsing stack:
                                [0 end] [1 E] [7 +]
current token stream:
                                int_num end
output:
                                shift 5

current parsing stack:
                                [0 end] [1 E] [7 +] [5 int_num]
current token stream:
                                end
output:
                                reduce F -> int_num
current subexpression:
                                1
                                type = INT_CONSTANT
                                value = 1
```

```
current parsing stack:
                                [0 end] [1 E] [7 +] [2 F]
current token stream:
                                end
output:
                                reduce T -> F
current subexpression:
                                1
                                type = INT_CONSTANT
                                value = 1

current parsing stack:
                                [0 end] [1 E] [7 +] [12 T]
current token stream:
                                end
output:
                                reduce E -> E + T
current subexpression:
                                1 + 1
                                type = INT_CONSTANT
                                value = 2

current parsing stack:
                                [0 end] [1 E]
current token stream:
                                end
output:
                                acc
```

```
2.3*4.5e6
current parsing stack:
                                [0 end]
current token stream:
                                real_num + real_num end
output:
                                shift 6

current parsing stack:
                                [0 end] [6 real_num]
current token stream:
                                + real_num end
output:
                                reduce F -> real_num
current subexpression:
                                2.3
                                type = REAL_CONSTANT
                                value = 2.3

current parsing stack:
                                [0 end] [2 F]
current token stream:
                                + real_num end
output:
                                reduce T -> F
current subexpression:
```

```
                                2.3
                                type = REAL_CONSTANT
                                value = 2.3

current parsing stack:
                                [0 end] [3 T]
current token stream:
                                + real_num end
output:
                                reduce E -> T
current subexpression:
                                2.3
                                type = REAL_CONSTANT
                                value = 2.3

current parsing stack:
                                [0 end] [1 E]
current token stream:
                                + real_num end
output:
                                shift 7

current parsing stack:
                                [0 end] [1 E] [7 +]
current token stream:
                                real_num end
output:
                                shift 6

current parsing stack:
                                [0 end] [1 E] [7 +] [6 real_num]
current token stream:
                                end
output:
                                reduce F -> real_num
current subexpression:
                                4.5e+06
                                type = REAL_CONSTANT
                                value = 4.5e+06

current parsing stack:
                                [0 end] [1 E] [7 +] [2 F]
current token stream:
                                end
output:
                                reduce T -> F
current subexpression:
                                4.5e+06
                                type = REAL_CONSTANT
                                value = 4.5e+06

current parsing stack:
                                [0 end] [1 E] [7 +] [12 T]
current token stream:
                                end
output:
                                reduce E -> E + T
```

```
current subexpression:
                                2.3 + 4.5e+06
                                type = REAL_CONSTANT
                                value = 4.5e+06

current parsing stack:
                                [0 end] [1 E]
current token stream:
                                end
output:
                                acc
input string:
                                2.3*4.5e6

current parsing stack:
                                [0 end]
current token stream:
                                real_num * real_num end
output:
                                shift 6

current parsing stack:
                                [0 end] [6 real_num]
current token stream:
                                * real_num end
output:
                                reduce F -> real_num
current subexpression:
                                2.3
                                type = REAL_CONSTANT
                                value = 2.3

current parsing stack:
                                [0 end] [2 F]
current token stream:
                                * real_num end
output:
                                reduce T -> F
current subexpression:
                                2.3
                                type = REAL_CONSTANT
                                value = 2.3

current parsing stack:
                                [0 end] [3 T]
current token stream:
                                * real_num end
output:
                                shift 10

current parsing stack:
                                [0 end] [3 T] [10 *]
current token stream:
                                real_num end
output:
                                shift 6
```

```
current parsing stack:
                              [0 end] [3 T] [10 *] [6 real_num]
current token stream:
                              end
output:
                              reduce F -> real_num
current subexpression:
                              4.5e+06
                              type = REAL_CONSTANT
                              value = 4.5e+06

current parsing stack:
                              [0 end] [3 T] [10 *] [15 F]
current token stream:
                              end
output:
                              reduce T -> T * F
current subexpression:
                              2.3 * 4.5e+06
                              type = REAL_CONSTANT
                              value = 1.035e+07

current parsing stack:
                              [0 end] [3 T]
current token stream:
                              end
output:
                              reduce E -> T
current subexpression:
                              1.035e+07
                              type = REAL_CONSTANT
                              value = 1.035e+07

current parsing stack:
                              [0 end] [1 E]
current token stream:
                              end
output:
                              acc
```

1/3注：整型除法，结果为0

```
current parsing stack:
                              [0 end]
current token stream:
                              int_num / int_num end
output:
                              shift 5

current parsing stack:
                              [0 end] [5 int_num]
current token stream:
                              / int_num end
output:
                              reduce F -> int_num
current subexpression:
```

```
                                1
                                type = INT_CONSTANT
                                value = 1

current parsing stack:
                                [0 end] [2 F]
current token stream:
                                / int_num end
output:
                                reduce T -> F
current subexpression:
                                1
                                type = INT_CONSTANT
                                value = 1

current parsing stack:
                                [0 end] [3 T]
current token stream:
                                / int_num end
output:
                                shift 9

current parsing stack:
                                [0 end] [3 T] [9 /]
current token stream:
                                int_num end
output:
                                shift 5

current parsing stack:
                                [0 end] [3 T] [9 /] [5 int_num]
current token stream:
                                end
output:
                                reduce F -> int_num
current subexpression:
                                3
                                type = INT_CONSTANT
                                value = 3

current parsing stack:
                                [0 end] [3 T] [9 /] [14 F]
current token stream:
                                end
output:
                                reduce T -> T / F
current subexpression:
                                1 / 3
                                type = INT_CONSTANT
                                value = 0

current parsing stack:
                                [0 end] [3 T]
current token stream:
                                end
output:
                                reduce E -> T
```

```
current subexpression:
                                0
                                type = INT_CONSTANT
                                value = 0

current parsing stack:
                                [0 end] [1 E]
current token stream:
                                end
output:
                                acc
```

1./3 注：实型除法，结果为0.333333

```
current parsing stack:
                                [0 end]
current token stream:
                                real_num / int_num end
output:
                                shift 6

current parsing stack:
                                [0 end] [6 real_num]
current token stream:
                                / int_num end
output:
                                reduce F -> real_num
current subexpression:
                                1
                                type = REAL_CONSTANT
                                value = 1

current parsing stack:
                                [0 end] [2 F]
current token stream:
                                / int_num end
output:
                                reduce T -> F
current subexpression:
                                1
                                type = REAL_CONSTANT
                                value = 1

current parsing stack:
                                [0 end] [3 T]
current token stream:
                                / int_num end
output:
                                shift 9

current parsing stack:
                                [0 end] [3 T] [9 /]
current token stream:
                                int_num end
output:
                                shift 5
```

```
current parsing stack:
                                [0 end] [3 T] [9 /] [5 int_num]
current token stream:
                                end
output:
                                reduce F -> int_num
current subexpression:
                                3
                                type = INT_CONSTANT
                                value = 3

current parsing stack:
                                [0 end] [3 T] [9 /] [14 F]
current token stream:
                                end
output:
                                reduce T -> T / F
current subexpression:
                                1 / 3
                                type = REAL_CONSTANT
                                value = 0.333333

current parsing stack:
                                [0 end] [3 T]
current token stream:
                                end
output:
                                reduce E -> T
current subexpression:
                                0.333333
                                type = REAL_CONSTANT
                                value = 0.333333

current parsing stack:
                                [0 end] [1 E]
current token stream:
                                end
output:
                                acc
```

```
(1 + 3.8) * (3 / 0.2 + 4)
current parsing stack:
                                [0 end]
current token stream:
                                ( int_num + real_num ) * ( int_num /
real_num + int_num ) end
output:
                                shift 4

current parsing stack:
                                [0 end] [4 (]
current token stream:
```

```
                                int_num + real_num ) * ( int_num /
real_num + int_num ) end
output:
                                shift 5

current parsing stack:
                                [0 end] [4 (] [5 int_num]
current token stream:
                                + real_num ) * ( int_num / real_num +
int_num ) end
output:
                                reduce F -> int_num
current subexpression:
                                1
                                type = INT_CONSTANT
                                value = 1

current parsing stack:
                                [0 end] [4 (] [2 F]
current token stream:
                                + real_num ) * ( int_num / real_num +
int_num ) end
output:
                                reduce T -> F
current subexpression:
                                1
                                type = INT_CONSTANT
                                value = 1

current parsing stack:
                                [0 end] [4 (] [3 T]
current token stream:
                                + real_num ) * ( int_num / real_num +
int_num ) end
output:
                                reduce E -> T
current subexpression:
                                1
                                type = INT_CONSTANT
                                value = 1

current parsing stack:
                                [0 end] [4 (] [11 E]
current token stream:
                                + real_num ) * ( int_num / real_num +
int_num ) end
output:
                                shift 7

current parsing stack:
                                [0 end] [4 (] [11 E] [7 +]
current token stream:
                                real_num ) * ( int_num / real_num +
int_num ) end
output:
                                shift 6
```

```
current parsing stack:
                                [0 end] [4 (] [11 E] [7 +] [6 real_num]
current token stream:
                                ) * ( int_num / real_num + int_num ) end
output:
                                reduce F -> real_num
current subexpression:
                                3.8
                                type = REAL_CONSTANT
                                value = 3.8

current parsing stack:
                                [0 end] [4 (] [11 E] [7 +] [2 F]
current token stream:
                                ) * ( int_num / real_num + int_num ) end
output:
                                reduce T -> F
current subexpression:
                                3.8
                                type = REAL_CONSTANT
                                value = 3.8

current parsing stack:
                                [0 end] [4 (] [11 E] [7 +] [12 T]
current token stream:
                                ) * ( int_num / real_num + int_num ) end
output:
                                reduce E -> E + T
current subexpression:
                                1 + 3.8
                                type = REAL_CONSTANT
                                value = 4.8

current parsing stack:
                                [0 end] [4 (] [11 E]
current token stream:
                                ) * ( int_num / real_num + int_num ) end
output:
                                shift 16

current parsing stack:
                                [0 end] [4 (] [11 E] [16 )]
current token stream:
                                * ( int_num / real_num + int_num ) end
output:
                                reduce F -> ( E )
current subexpression:
                                ( 4.8 )
                                type = REAL_CONSTANT
                                value = 4.8

current parsing stack:
                                [0 end] [2 F]
current token stream:
                                * ( int_num / real_num + int_num ) end
output:
                                reduce T -> F
```

```
current subexpression:
                              4.8
                              type = REAL_CONSTANT
                              value = 4.8

current parsing stack:
                              [0 end] [3 T]
current token stream:
                              * ( int_num / real_num + int_num ) end
output:
                              shift 10

current parsing stack:
                              [0 end] [3 T] [10 *]
current token stream:
                              ( int_num / real_num + int_num ) end
output:
                              shift 4

current parsing stack:
                              [0 end] [3 T] [10 *] [4 (]
current token stream:
                              int_num / real_num + int_num ) end
output:
                              shift 5

current parsing stack:
                              [0 end] [3 T] [10 *] [4 (] [5 int_num]
current token stream:
                              / real_num + int_num ) end
output:
                              reduce F -> int_num
current subexpression:
                              3
                              type = INT_CONSTANT
                              value = 3

current parsing stack:
                              [0 end] [3 T] [10 *] [4 (] [2 F]
current token stream:
                              / real_num + int_num ) end
output:
                              reduce T -> F
current subexpression:
                              3
                              type = INT_CONSTANT
                              value = 3

current parsing stack:
                              [0 end] [3 T] [10 *] [4 (] [3 T]
current token stream:
                              / real_num + int_num ) end
output:
                              shift 9

current parsing stack:
                              [0 end] [3 T] [10 *] [4 (] [3 T] [9 /]
current token stream:
                                real_num + int_num ) end
output:
                                shift 6

current parsing stack:
                                [0 end] [3 T] [10 *] [4 (] [3 T] [9 /]
[6 real_num]
current token stream:
                                + int_num ) end
output:
                                reduce F -> real_num
current subexpression:
                                0.2
                                type = REAL_CONSTANT
                                value = 0.2

current parsing stack:
                                [0 end] [3 T] [10 *] [4 (] [3 T] [9 /]
[14 F]
current token stream:
                                + int_num ) end
output:
                                reduce T -> T / F
current subexpression:
                                3 / 0.2
                                type = REAL_CONSTANT
                                value = 15

current parsing stack:
                                [0 end] [3 T] [10 *] [4 (] [3 T]
current token stream:
                                + int_num ) end
output:
                                reduce E -> T
current subexpression:
                                15
                                type = REAL_CONSTANT
                                value = 15

current parsing stack:
                                [0 end] [3 T] [10 *] [4 (] [11 E]
current token stream:
                                + int_num ) end
output:
                                shift 7

current parsing stack:
                                [0 end] [3 T] [10 *] [4 (] [11 E] [7 +]
current token stream:
                                int_num ) end
output:
                                shift 5

current parsing stack:
                                [0 end] [3 T] [10 *] [4 (] [11 E] [7 +]
[5 int_num]
```

```
current token stream:
                                ) end
output:
                                reduce F -> int_num
current subexpression:
                                4
                                type = INT_CONSTANT
                                value = 4

current parsing stack:
                                [0 end] [3 T] [10 *] [4 (] [11 E] [7 +]
[2 F]
current token stream:
                                ) end
output:
                                reduce T -> F
current subexpression:
                                4
                                type = INT_CONSTANT
                                value = 4

current parsing stack:
                                [0 end] [3 T] [10 *] [4 (] [11 E] [7 +]
[12 T]
current token stream:
                                ) end
output:
                                reduce E -> E + T
current subexpression:
                                15 + 4
                                type = REAL_CONSTANT
                                value = 19

current parsing stack:
                                [0 end] [3 T] [10 *] [4 (] [11 E]
current token stream:
                                ) end
output:
                                shift 16

current parsing stack:
                                [0 end] [3 T] [10 *] [4 (] [11 E] [16 )]
current token stream:
                                end
output:
                                reduce F -> ( E )
current subexpression:
                                ( 19 )
                                type = REAL_CONSTANT
                                value = 19

current parsing stack:
                                [0 end] [3 T] [10 *] [15 F]
current token stream:
                                end
output:
                                reduce T -> T * F
```

```
current subexpression:
                                 4.8 * 19
                                 type = REAL_CONSTANT
                                 value = 91.2

current parsing stack:
                                 [0 end] [3 T]
current token stream:
                                 end
output:
                                 reduce E -> T
current subexpression:
                                 91.2
                                 type = REAL_CONSTANT
                                 value = 91.2

current parsing stack:
                                 [0 end] [1 E]
current token stream:
                                 end
output:
                                 acc
```

```
(3.2 + 6.9)
current parsing stack:
                                 [0 end]
current token stream:
                                 ( real_num + real_num ) end
output:
                                 shift 4

current parsing stack:
                                 [0 end] [4 (]
current token stream:
                                 real_num + real_num ) end
output:
                                 shift 6

current parsing stack:
                                 [0 end] [4 (] [6 real_num]
current token stream:
                                 + real_num ) end
output:
                                 reduce F -> real_num
current subexpression:
                                 3.2
                                 type = REAL_CONSTANT
                                 value = 3.2

current parsing stack:
                                 [0 end] [4 (] [2 F]
current token stream:
                                 + real_num ) end
output:
                                 reduce T -> F
```

```
current subexpression:
                              3.2
                              type = REAL_CONSTANT
                              value = 3.2

current parsing stack:
                              [0 end] [4 (] [3 T]
current token stream:
                              + real_num ) end
output:
                              reduce E -> T
current subexpression:
                              3.2
                              type = REAL_CONSTANT
                              value = 3.2

current parsing stack:
                              [0 end] [4 (] [11 E]
current token stream:
                              + real_num ) end
output:
                              shift 7

current parsing stack:
                              [0 end] [4 (] [11 E] [7 +]
current token stream:
                              real_num ) end
output:
                              shift 6

current parsing stack:
                              [0 end] [4 (] [11 E] [7 +] [6 real_num]
current token stream:
                              ) end
output:
                              reduce F -> real_num
current subexpression:
                              6.9
                              type = REAL_CONSTANT
                              value = 6.9

current parsing stack:
                              [0 end] [4 (] [11 E] [7 +] [2 F]
current token stream:
                              ) end
output:
                              reduce T -> F
current subexpression:
                              6.9
                              type = REAL_CONSTANT
                              value = 6.9

current parsing stack:
                              [0 end] [4 (] [11 E] [7 +] [12 T]
current token stream:
                              ) end
output:
```

```
                                reduce E -> E + T
current subexpression:
                                3.2 + 6.9
                                type = REAL_CONSTANT
                                value = 10.1

current parsing stack:
                                [0 end] [4 (] [11 E]
current token stream:
                                ) end
output:
                                shift 16

current parsing stack:
                                [0 end] [4 (] [11 E] [16 )]
current token stream:
                                end
output:
                                reduce F -> ( E )
current subexpression:
                                ( 10.1 )
                                type = REAL_CONSTANT
                                value = 10.1

current parsing stack:
                                [0 end] [2 F]
current token stream:
                                end
output:
                                reduce T -> F
current subexpression:
                                10.1
                                type = REAL_CONSTANT
                                value = 10.1

current parsing stack:
                                [0 end] [3 T]
current token stream:
                                end
output:
                                reduce E -> T
current subexpression:
                                10.1
                                type = REAL_CONSTANT
                                value = 10.1

current parsing stack:
                                [0 end] [1 E]
current token stream:
                                end
output:
                                acc
```

```
((((((4))))))
```

```
current parsing stack:
```

```
                                [0 end]
current token stream:
                                ( ( ( ( ( ( int_num ) ) ) ) ) ) end
output:
                                shift 4

current parsing stack:
                                [0 end] [4 (]
current token stream:
                                ( ( ( ( ( int_num ) ) ) ) ) ) end
output:
                                shift 4

current parsing stack:
                                [0 end] [4 (] [4 (]
current token stream:
                                ( ( ( ( int_num ) ) ) ) ) ) end
output:
                                shift 4

current parsing stack:
                                [0 end] [4 (] [4 (] [4 (]
current token stream:
                                ( ( ( int_num ) ) ) ) ) ) end
output:
                                shift 4

current parsing stack:
                                [0 end] [4 (] [4 (] [4 (] [4 (]
current token stream:
                                ( ( int_num ) ) ) ) ) ) end
output:
                                shift 4

current parsing stack:
                                [0 end] [4 (] [4 (] [4 (] [4 (] [4 (]
current token stream:
                                ( int_num ) ) ) ) ) ) end
output:
                                shift 4

current parsing stack:
                                [0 end] [4 (] [4 (] [4 (] [4 (] [4 (] [4
(]
current token stream:
                                int_num ) ) ) ) ) ) end
output:
                                shift 5

current parsing stack:
                                [0 end] [4 (] [4 (] [4 (] [4 (] [4 (] [4
(] [5 int_num]
current token stream:
                                ) ) ) ) ) ) end
output:
                                reduce F -> int_num
current subexpression:
```

```
                                4
                                type = INT_CONSTANT
                                value = 4

current parsing stack:
                                [0 end] [4 (] [4 (] [4 (] [4 (] [4 (] [4
(] [2 F]
current token stream:
                                ) ) ) ) ) ) end
output:
                                reduce T -> F
current subexpression:
                                4
                                type = INT_CONSTANT
                                value = 4

current parsing stack:
                                [0 end] [4 (] [4 (] [4 (] [4 (] [4 (] [4
(] [3 T]
current token stream:
                                ) ) ) ) ) ) end
output:
                                reduce E -> T
current subexpression:
                                4
                                type = INT_CONSTANT
                                value = 4

current parsing stack:
                                [0 end] [4 (] [4 (] [4 (] [4 (] [4 (] [4
(] [11 E]
current token stream:
                                ) ) ) ) ) ) end
output:
                                shift 16

current parsing stack:
                                [0 end] [4 (] [4 (] [4 (] [4 (] [4 (] [4
(] [11 E] [16 )]
current token stream:
                                ) ) ) ) ) end
output:
                                reduce F -> ( E )
current subexpression:
                                ( 4 )
                                type = INT_CONSTANT
                                value = 4

current parsing stack:
                                [0 end] [4 (] [4 (] [4 (] [4 (] [4 (] [2
F]
current token stream:
                                ) ) ) ) ) end
output:
                                reduce T -> F
current subexpression:
                                4
                                type = INT_CONSTANT
                                value = 4

current parsing stack:
                                [0 end] [4 (] [4 (] [4 (] [4 (] [4 (] [3
T]
current token stream:
                                ) ) ) ) ) end
output:
                                reduce E -> T
current subexpression:
                                4
                                type = INT_CONSTANT
                                value = 4

current parsing stack:
                                [0 end] [4 (] [4 (] [4 (] [4 (] [4 (]
[11 E]
current token stream:
                                ) ) ) ) ) end
output:
                                shift 16

current parsing stack:
                                [0 end] [4 (] [4 (] [4 (] [4 (] [4 (]
[11 E] [16 )]
current token stream:
                                ) ) ) ) end
output:
                                reduce F -> ( E )
current subexpression:
                                ( 4 )
                                type = INT_CONSTANT
                                value = 4

current parsing stack:
                                [0 end] [4 (] [4 (] [4 (] [4 (] [2 F]
current token stream:
                                ) ) ) ) end
output:
                                reduce T -> F
current subexpression:
                                4
                                type = INT_CONSTANT
                                value = 4

current parsing stack:
                                [0 end] [4 (] [4 (] [4 (] [4 (] [3 T]
current token stream:
                                ) ) ) ) end
output:
                                reduce E -> T
current subexpression:
                                4
                                type = INT_CONSTANT
                                value = 4
```

```
current parsing stack:
                                [0 end] [4 (] [4 (] [4 (] [4 (] [11 E]
current token stream:
                                ) ) ) ) end
output:
                                shift 16

current parsing stack:
                                [0 end] [4 (] [4 (] [4 (] [4 (] [11 E]
[16 )]
current token stream:
                                ) ) ) end
output:
                                reduce F -> ( E )
current subexpression:
                                ( 4 )
                                type = INT_CONSTANT
                                value = 4

current parsing stack:
                                [0 end] [4 (] [4 (] [4 (] [2 F]
current token stream:
                                ) ) ) end
output:
                                reduce T -> F
current subexpression:
                                4
                                type = INT_CONSTANT
                                value = 4

current parsing stack:
                                [0 end] [4 (] [4 (] [4 (] [3 T]
current token stream:
                                ) ) ) end
output:
                                reduce E -> T
current subexpression:
                                4
                                type = INT_CONSTANT
                                value = 4

current parsing stack:
                                [0 end] [4 (] [4 (] [4 (] [11 E]
current token stream:
                                ) ) ) end
output:
                                shift 16

current parsing stack:
                                [0 end] [4 (] [4 (] [4 (] [11 E] [16 )]
current token stream:
                                ) ) end
output:
                                reduce F -> ( E )
current subexpression:
                                ( 4 )
                                type = INT_CONSTANT
                                value = 4

current parsing stack:
                                [0 end] [4 (] [4 (] [2 F]
current token stream:
                                ) ) end
output:
                                reduce T -> F
current subexpression:
                                4
                                type = INT_CONSTANT
                                value = 4

current parsing stack:
                                [0 end] [4 (] [4 (] [3 T]
current token stream:
                                ) ) end
output:
                                reduce E -> T
current subexpression:
                                4
                                type = INT_CONSTANT
                                value = 4

current parsing stack:
                                [0 end] [4 (] [4 (] [11 E]
current token stream:
                                ) ) end
output:
                                shift 16

current parsing stack:
                                [0 end] [4 (] [4 (] [11 E] [16 )]
current token stream:
                                ) end
output:
                                reduce F -> ( E )
current subexpression:
                                ( 4 )
                                type = INT_CONSTANT
                                value = 4

current parsing stack:
                                [0 end] [4 (] [2 F]
current token stream:
                                ) end
output:
                                reduce T -> F
current subexpression:
                                4
                                type = INT_CONSTANT
                                value = 4

current parsing stack:
                                [0 end] [4 (] [3 T]
current token stream:
                                ) end
```

```
output:
                                reduce E -> T
current subexpression:
                                4
                                type = INT_CONSTANT
                                value = 4

current parsing stack:
                                [0 end] [4 (] [11 E]
current token stream:
                                ) end
output:
                                shift 16

current parsing stack:
                                [0 end] [4 (] [11 E] [16 )]
current token stream:
                                end
output:
                                reduce F -> ( E )
current subexpression:
                                ( 4 )
                                type = INT_CONSTANT
                                value = 4

current parsing stack:
                                [0 end] [2 F]
current token stream:
                                end
output:
                                reduce T -> F
current subexpression:
                                4
                                type = INT_CONSTANT
                                value = 4

current parsing stack:
                                [0 end] [3 T]
current token stream:
                                end
output:
                                reduce E -> T
current subexpression:
                                4
                                type = INT_CONSTANT
                                value = 4

current parsing stack:
                                [0 end] [1 E]
current token stream:
                                end
output:
                                acc
```

```
((0.2-9) *(5*9/9+(10)))
```

```
current parsing stack:
                                [0 end]
current token stream:
                                ( ( real_num - int_num ) * ( int_num *
int_num / int_num + ( int_num ) ) ) end
output:
                                shift 4

current parsing stack:
                                [0 end] [4 (]
current token stream:
                                ( real_num - int_num ) * ( int_num *
int_num / int_num + ( int_num ) ) ) end
output:
                                shift 4

current parsing stack:
                                [0 end] [4 (] [4 (]
current token stream:
                                real_num - int_num ) * ( int_num *
int_num / int_num + ( int_num ) ) ) end
output:
                                shift 6

current parsing stack:
                                [0 end] [4 (] [4 (] [6 real_num]
current token stream:
                                - int_num ) * ( int_num * int_num /
int_num + ( int_num ) ) ) end
output:
                                reduce F -> real_num
current subexpression:
                                0.2
                                type = REAL_CONSTANT
                                value = 0.2

current parsing stack:
                                [0 end] [4 (] [4 (] [2 F]
current token stream:
                                - int_num ) * ( int_num * int_num /
int_num + ( int_num ) ) ) end
output:
                                reduce T -> F
current subexpression:
                                0.2
                                type = REAL_CONSTANT
                                value = 0.2

current parsing stack:
                                [0 end] [4 (] [4 (] [3 T]
current token stream:
                                - int_num ) * ( int_num * int_num /
int_num + ( int_num ) ) ) end
output:
                                reduce E -> T
current subexpression:
                                0.2
                                type = REAL_CONSTANT
                                value = 0.2

current parsing stack:
                                [0 end] [4 (] [4 (] [11 E]
current token stream:
                                - int_num ) * ( int_num * int_num /
int_num + ( int_num ) ) ) end
output:
                                shift 8

current parsing stack:
                                [0 end] [4 (] [4 (] [11 E] [8 -]
current token stream:
                                int_num ) * ( int_num * int_num /
int_num + ( int_num ) ) ) end
output:
                                shift 5

current parsing stack:
                                [0 end] [4 (] [4 (] [11 E] [8 -] [5
int_num]
current token stream:
                                ) * ( int_num * int_num / int_num +
( int_num ) ) ) end
output:
                                reduce F -> int_num
current subexpression:
                                9
                                type = INT_CONSTANT
                                value = 9

current parsing stack:
                                [0 end] [4 (] [4 (] [11 E] [8 -] [2 F]
current token stream:
                                ) * ( int_num * int_num / int_num +
( int_num ) ) ) end
output:
                                reduce T -> F
current subexpression:
                                9
                                type = INT_CONSTANT
                                value = 9

current parsing stack:
                                [0 end] [4 (] [4 (] [11 E] [8 -] [13 T]
current token stream:
                                ) * ( int_num * int_num / int_num +
( int_num ) ) ) end
output:
                                reduce E -> E - T
current subexpression:
                                0.2 - 9
                                type = REAL_CONSTANT
                                value = -8.8

current parsing stack:
```

```
                                [0 end] [4 (] [4 (] [11 E]
current token stream:
                                ) * ( int_num * int_num / int_num +
( int_num ) ) ) end
output:
                                shift 16

current parsing stack:
                                [0 end] [4 (] [4 (] [11 E] [16 )]
current token stream:
                                * ( int_num * int_num / int_num +
( int_num ) ) ) end
output:
                                reduce F -> ( E )
current subexpression:
                                ( -8.8 )
                                type = REAL_CONSTANT
                                value = -8.8

current parsing stack:
                                [0 end] [4 (] [2 F]
current token stream:
                                * ( int_num * int_num / int_num +
( int_num ) ) ) end
output:
                                reduce T -> F
current subexpression:
                                -8.8
                                type = REAL_CONSTANT
                                value = -8.8

current parsing stack:
                                [0 end] [4 (] [3 T]
current token stream:
                                * ( int_num * int_num / int_num +
( int_num ) ) ) end
output:
                                shift 10

current parsing stack:
                                [0 end] [4 (] [3 T] [10 *]
current token stream:
                                ( int_num * int_num / int_num +
( int_num ) ) ) end
output:
                                shift 4

current parsing stack:
                                [0 end] [4 (] [3 T] [10 *] [4 (]
current token stream:
                                int_num * int_num / int_num +
( int_num ) ) ) end
output:
                                shift 5

current parsing stack:
```

```
                                  [0 end] [4 (] [3 T] [10 *] [4 (] [5
int_num]
current token stream:
                                  * int_num / int_num + ( int_num ) ) )
end
output:
                                  reduce F -> int_num
current subexpression:
                                  5
                                  type = INT_CONSTANT
                                  value = 5

current parsing stack:
                                  [0 end] [4 (] [3 T] [10 *] [4 (] [2 F]
current token stream:
                                  * int_num / int_num + ( int_num ) ) )
end
output:
                                  reduce T -> F
current subexpression:
                                  5
                                  type = INT_CONSTANT
                                  value = 5

current parsing stack:
                                  [0 end] [4 (] [3 T] [10 *] [4 (] [3 T]
current token stream:
                                  * int_num / int_num + ( int_num ) ) )
end
output:
                                  shift 10

current parsing stack:
                                  [0 end] [4 (] [3 T] [10 *] [4 (] [3 T]
[10 *]
current token stream:
                                  int_num / int_num + ( int_num ) ) ) end
output:
                                  shift 5

current parsing stack:
                                  [0 end] [4 (] [3 T] [10 *] [4 (] [3 T]
[10 *] [5 int_num]
current token stream:
                                  / int_num + ( int_num ) ) ) end
output:
                                  reduce F -> int_num
current subexpression:
                                  9
                                  type = INT_CONSTANT
                                  value = 9

current parsing stack:
                                  [0 end] [4 (] [3 T] [10 *] [4 (] [3 T]
[10 *] [15 F]
current token stream:
                                  / int_num + ( int_num ) ) ) end
```

```
output:
                                reduce T -> T * F
current subexpression:
                                5 * 9
                                type = INT_CONSTANT
                                value = 45

current parsing stack:
                                [0 end] [4 (] [3 T] [10 *] [4 (] [3 T]
current token stream:
                                / int_num + ( int_num ) ) ) end
output:
                                shift 9

current parsing stack:
                                [0 end] [4 (] [3 T] [10 *] [4 (] [3 T]
[9 /]
current token stream:
                                int_num + ( int_num ) ) ) end
output:
                                shift 5

current parsing stack:
                                [0 end] [4 (] [3 T] [10 *] [4 (] [3 T]
[9 /] [5 int_num]
current token stream:
                                + ( int_num ) ) ) end
output:
                                reduce F -> int_num
current subexpression:
                                9
                                type = INT_CONSTANT
                                value = 9

current parsing stack:
                                [0 end] [4 (] [3 T] [10 *] [4 (] [3 T]
[9 /] [14 F]
current token stream:
                                + ( int_num ) ) ) end
output:
                                reduce T -> T / F
current subexpression:
                                45 / 9
                                type = INT_CONSTANT
                                value = 5

current parsing stack:
                                [0 end] [4 (] [3 T] [10 *] [4 (] [3 T]
current token stream:
                                + ( int_num ) ) ) end
output:
                                reduce E -> T
current subexpression:
                                5
                                type = INT_CONSTANT
                                value = 5
```

```
current parsing stack:
                                [0 end] [4 (] [3 T] [10 *] [4 (] [11 E]
current token stream:
                                + ( int_num ) ) ) end
output:
                                shift 7

current parsing stack:
                                [0 end] [4 (] [3 T] [10 *] [4 (] [11 E]
[7 +]
current token stream:
                                ( int_num ) ) ) end
output:
                                shift 4

current parsing stack:
                                [0 end] [4 (] [3 T] [10 *] [4 (] [11 E]
[7 +] [4 (]
current token stream:
                                int_num ) ) ) end
output:
                                shift 5

current parsing stack:
                                [0 end] [4 (] [3 T] [10 *] [4 (] [11 E]
[7 +] [4 (] [5 int_num]
current token stream:
                                ) ) ) end
output:
                                reduce F -> int_num
current subexpression:
                                10
                                type = INT_CONSTANT
                                value = 10

current parsing stack:
                                [0 end] [4 (] [3 T] [10 *] [4 (] [11 E]
[7 +] [4 (] [2 F]
current token stream:
                                ) ) ) end
output:
                                reduce T -> F
current subexpression:
                                10
                                type = INT_CONSTANT
                                value = 10

current parsing stack:
                                [0 end] [4 (] [3 T] [10 *] [4 (] [11 E]
[7 +] [4 (] [3 T]
current token stream:
                                ) ) ) end
output:
                                reduce E -> T
current subexpression:
                                10
                                type = INT_CONSTANT
```

```
                                  value = 10

current parsing stack:
                                  [0 end] [4 (] [3 T] [10 *] [4 (] [11 E]
[7 +] [4 (] [11 E]
current token stream:
                                  ) ) ) end
output:
                                  shift 16

current parsing stack:
                                  [0 end] [4 (] [3 T] [10 *] [4 (] [11 E]
[7 +] [4 (] [11 E] [16 )]
current token stream:
                                  ) ) end
output:
                                  reduce F -> ( E )
current subexpression:
                                  ( 10 )
                                  type = INT_CONSTANT
                                  value = 10

current parsing stack:
                                  [0 end] [4 (] [3 T] [10 *] [4 (] [11 E]
[7 +] [2 F]
current token stream:
                                  ) ) end
output:
                                  reduce T -> F
current subexpression:
                                  10
                                  type = INT_CONSTANT
                                  value = 10

current parsing stack:
                                  [0 end] [4 (] [3 T] [10 *] [4 (] [11 E]
[7 +] [12 T]
current token stream:
                                  ) ) end
output:
                                  reduce E -> E + T
current subexpression:
                                  5 + 10
                                  type = INT_CONSTANT
                                  value = 15

current parsing stack:
                                  [0 end] [4 (] [3 T] [10 *] [4 (] [11 E]
current token stream:
                                  ) ) end
output:
                                  shift 16

current parsing stack:
                                  [0 end] [4 (] [3 T] [10 *] [4 (] [11 E]
[16 )]
current token stream:
```

```
                                ) end
output:
                                reduce F -> ( E )
current subexpression:
                                ( 15 )
                                type = INT_CONSTANT
                                value = 15

current parsing stack:
                                [0 end] [4 (] [3 T] [10 *] [15 F]
current token stream:
                                ) end
output:
                                reduce T -> T * F
current subexpression:
                                -8.8 * 15
                                type = REAL_CONSTANT
                                value = -132

current parsing stack:
                                [0 end] [4 (] [3 T]
current token stream:
                                ) end
output:
                                reduce E -> T
current subexpression:
                                -132
                                type = REAL_CONSTANT
                                value = -132

current parsing stack:
                                [0 end] [4 (] [11 E]
current token stream:
                                ) end
output:
                                shift 16

current parsing stack:
                                [0 end] [4 (] [11 E] [16 )]
current token stream:
                                end
output:
                                reduce F -> ( E )
current subexpression:
                                ( -132 )
                                type = REAL_CONSTANT
                                value = -132

current parsing stack:
                                [0 end] [2 F]
current token stream:
                                end
output:
                                reduce T -> F
current subexpression:
                                -132
                                type = REAL_CONSTANT
                                value = -132

current parsing stack:
                                [0 end] [3 T]
current token stream:
                                end
output:
                                reduce E -> T
current subexpression:
                                -132
                                type = REAL_CONSTANT
                                value = -132

current parsing stack:
                                [0 end] [1 E]
current token stream:
                                end
output:
                                acc
```

```
((2*(5 - 9)/8)*(9/(1/(9-9.36e5))))
current parsing stack:
                                [0 end]
current token stream:
                                ( ( int_num * ( int_num - int_num )
/ int_num ) * ( int_num / ( int_num / ( int_num - real_num ) ) ) )
end
output:
                                shift 4

current parsing stack:
                                [0 end] [4 (]
current token stream:
                                ( int_num * ( int_num - int_num ) /
int_num ) * ( int_num / ( int_num / ( int_num - real_num ) ) ) )
end
output:
                                shift 4

current parsing stack:
                                [0 end] [4 (] [4 (]
current token stream:
                                int_num * ( int_num - int_num ) /
int_num ) * ( int_num / ( int_num / ( int_num - real_num ) ) ) )
end
output:
                                shift 5

current parsing stack:
                                [0 end] [4 (] [4 (] [5 int_num]
current token stream:
                                * ( int_num - int_num ) / int_num )
* ( int_num / ( int_num / ( int_num - real_num ) ) ) ) end
output:
                                reduce F -> int_num
current subexpression:
```

```
                                2
                                type = INT_CONSTANT
                                value = 2

current parsing stack:
                                [0 end] [4 (] [4 (] [2 F]
current token stream:
                                * ( int_num - int_num ) / int_num )
* ( int_num / ( int_num / ( int_num - real_num ) ) ) ) end
output:
                                reduce T -> F
current subexpression:
                                2
                                type = INT_CONSTANT
                                value = 2

current parsing stack:
                                [0 end] [4 (] [4 (] [3 T]
current token stream:
                                * ( int_num - int_num ) / int_num )
* ( int_num / ( int_num / ( int_num - real_num ) ) ) ) end
output:
                                shift 10

current parsing stack:
                                [0 end] [4 (] [4 (] [3 T] [10 *]
current token stream:
                                ( int_num - int_num ) / int_num ) *
( int_num / ( int_num / ( int_num - real_num ) ) ) ) end
output:
                                shift 4

current parsing stack:
                                [0 end] [4 (] [4 (] [3 T] [10 *] [4
(]
current token stream:
                                int_num - int_num ) / int_num ) *
( int_num / ( int_num / ( int_num - real_num ) ) ) ) end
output:
                                shift 5

current parsing stack:
                                [0 end] [4 (] [4 (] [3 T] [10 *] [4
(] [5 int_num]
current token stream:
                                - int_num ) / int_num ) * ( int_num
/ ( int_num / ( int_num - real_num ) ) ) ) end
output:
                                reduce F -> int_num
current subexpression:
                                5
                                type = INT_CONSTANT
                                value = 5

current parsing stack:
                                [0 end] [4 (] [4 (] [3 T] [10 *] [4
(] [2 F]
current token stream:
                                - int_num ) / int_num ) * ( int_num
/ ( int_num / ( int_num - real_num ) ) ) ) end
output:
                                reduce T -> F
current subexpression:
                                5
                                type = INT_CONSTANT
                                value = 5

current parsing stack:
                                [0 end] [4 (] [4 (] [3 T] [10 *] [4
(] [3 T]
current token stream:
                                - int_num ) / int_num ) * ( int_num
/ ( int_num / ( int_num - real_num ) ) ) ) end
output:
                                reduce E -> T
current subexpression:
                                5
                                type = INT_CONSTANT
                                value = 5

current parsing stack:
                                [0 end] [4 (] [4 (] [3 T] [10 *] [4
(] [11 E]
current token stream:
                                - int_num ) / int_num ) * ( int_num
/ ( int_num / ( int_num - real_num ) ) ) ) end
output:
                                shift 8

current parsing stack:
                                [0 end] [4 (] [4 (] [3 T] [10 *] [4
(] [11 E] [8 -]
current token stream:
                                int_num ) / int_num ) * ( int_num /
( int_num / ( int_num - real_num ) ) ) ) end
output:
                                shift 5

current parsing stack:
                                [0 end] [4 (] [4 (] [3 T] [10 *] [4
(] [11 E] [8 -] [5 int_num]
current token stream:
                                ) / int_num ) * ( int_num /
( int_num / ( int_num - real_num ) ) ) ) end
output:
                                reduce F -> int_num
current subexpression:
                                9
                                type = INT_CONSTANT
                                value = 9

current parsing stack:
                                [0 end] [4 (] [4 (] [3 T] [10 *] [4
(] [11 E] [8 -] [2 F]
current token stream:
                                ) / int_num ) * ( int_num /
( int_num / ( int_num - real_num ) ) ) ) end
output:
                                reduce T -> F
current subexpression:
                                9
                                type = INT_CONSTANT
                                value = 9

current parsing stack:
                                [0 end] [4 (] [4 (] [3 T] [10 *] [4
(] [11 E] [8 -] [13 T]
current token stream:
                                ) / int_num ) * ( int_num /
( int_num / ( int_num - real_num ) ) ) ) end
output:
                                reduce E -> E - T
current subexpression:
                                5 - 9
                                type = INT_CONSTANT
                                value = -4

current parsing stack:
                                [0 end] [4 (] [4 (] [3 T] [10 *] [4
(] [11 E]
current token stream:
                                ) / int_num ) * ( int_num /
( int_num / ( int_num - real_num ) ) ) ) end
output:
                                shift 16

current parsing stack:
                                [0 end] [4 (] [4 (] [3 T] [10 *] [4
(] [11 E] [16 )]
current token stream:
                                / int_num ) * ( int_num / ( int_num
/ ( int_num - real_num ) ) ) ) end
output:
                                reduce F -> ( E )
current subexpression:
                                ( -4 )
                                type = INT_CONSTANT
                                value = -4

current parsing stack:
                                [0 end] [4 (] [4 (] [3 T] [10 *]
[15 F]
current token stream:
                                / int_num ) * ( int_num / ( int_num
/ ( int_num - real_num ) ) ) ) end
output:
                                reduce T -> T * F
current subexpression:
                                2 * -4
                                type = INT_CONSTANT
                                value = -8
```

```
current parsing stack:
                                [0 end] [4 (] [4 (] [3 T]
current token stream:
                                / int_num ) * ( int_num / ( int_num
/ ( int_num - real_num ) ) ) ) end
output:
                                shift 9

current parsing stack:
                                [0 end] [4 (] [4 (] [3 T] [9 /]
current token stream:
                                int_num ) * ( int_num / ( int_num /
( int_num - real_num ) ) ) ) end
output:
                                shift 5

current parsing stack:
                                [0 end] [4 (] [4 (] [3 T] [9 /] [5
int_num]
current token stream:
                                ) * ( int_num / ( int_num /
( int_num - real_num ) ) ) ) end
output:
                                reduce F -> int_num
current subexpression:
                                8
                                type = INT_CONSTANT
                                value = 8

current parsing stack:
                                [0 end] [4 (] [4 (] [3 T] [9 /] [14
F]
current token stream:
                                ) * ( int_num / ( int_num /
( int_num - real_num ) ) ) ) end
output:
                                reduce T -> T / F
current subexpression:
                                -8 / 8
                                type = INT_CONSTANT
                                value = -1

current parsing stack:
                                [0 end] [4 (] [4 (] [3 T]
current token stream:
                                ) * ( int_num / ( int_num /
( int_num - real_num ) ) ) ) end
output:
                                reduce E -> T
current subexpression:
                                -1
                                type = INT_CONSTANT
                                value = -1

current parsing stack:
                                [0 end] [4 (] [4 (] [11 E]
current token stream:
                                ) * ( int_num / ( int_num /
( int_num - real_num ) ) ) ) end
output:
                                shift 16

current parsing stack:
                                [0 end] [4 (] [4 (] [11 E] [16 )]
current token stream:
                                * ( int_num / ( int_num / ( int_num
- real_num ) ) ) ) end
output:
                                reduce F -> ( E )
current subexpression:
                                ( -1 )
                                type = INT_CONSTANT
                                value = -1

current parsing stack:
                                [0 end] [4 (] [2 F]
current token stream:
                                * ( int_num / ( int_num / ( int_num
- real_num ) ) ) ) end
output:
                                reduce T -> F
current subexpression:
                                -1
                                type = INT_CONSTANT
                                value = -1

current parsing stack:
                                [0 end] [4 (] [3 T]
current token stream:
                                * ( int_num / ( int_num / ( int_num
- real_num ) ) ) ) end
output:
                                shift 10

current parsing stack:
                                [0 end] [4 (] [3 T] [10 *]
current token stream:
                                ( int_num / ( int_num / ( int_num -
real_num ) ) ) ) end
output:
                                shift 4

current parsing stack:
                                [0 end] [4 (] [3 T] [10 *] [4 (]
current token stream:
                                int_num / ( int_num / ( int_num -
real_num ) ) ) ) end
output:
                                shift 5

current parsing stack:
                                [0 end] [4 (] [3 T] [10 *] [4 (] [5
int_num]
```

```
current token stream:
                                / ( int_num / ( int_num -
real_num ) ) ) ) end
output:
                                reduce F -> int_num
current subexpression:
                                9
                                type = INT_CONSTANT
                                value = 9

current parsing stack:
                                [0 end] [4 (] [3 T] [10 *] [4 (] [2
F]
current token stream:
                                / ( int_num / ( int_num -
real_num ) ) ) ) end
output:
                                reduce T -> F
current subexpression:
                                9
                                type = INT_CONSTANT
                                value = 9

current parsing stack:
                                [0 end] [4 (] [3 T] [10 *] [4 (] [3
T]
current token stream:
                                / ( int_num / ( int_num -
real_num ) ) ) ) end
output:
                                shift 9

current parsing stack:
                                [0 end] [4 (] [3 T] [10 *] [4 (] [3
T] [9 /]
current token stream:
                                ( int_num / ( int_num -
real_num ) ) ) ) end
output:
                                shift 4

current parsing stack:
                                [0 end] [4 (] [3 T] [10 *] [4 (] [3
T] [9 /] [4 (]
current token stream:
                                int_num / ( int_num -
real_num ) ) ) ) end
output:
                                shift 5

current parsing stack:
                                [0 end] [4 (] [3 T] [10 *] [4 (] [3
T] [9 /] [4 (] [5 int_num]
current token stream:
                                / ( int_num - real_num ) ) ) ) end
output:
                                reduce F -> int_num
```

```
current subexpression:
                                1
                                type = INT_CONSTANT
                                value = 1

current parsing stack:
                                [0 end] [4 (] [3 T] [10 *] [4 (] [3
T] [9 /] [4 (] [2 F]
current token stream:
                                / ( int_num - real_num ) ) ) ) end
output:
                                reduce T -> F
current subexpression:
                                1
                                type = INT_CONSTANT
                                value = 1

current parsing stack:
                                [0 end] [4 (] [3 T] [10 *] [4 (] [3
T] [9 /] [4 (] [3 T]
current token stream:
                                / ( int_num - real_num ) ) ) ) end
output:
                                shift 9

current parsing stack:
                                [0 end] [4 (] [3 T] [10 *] [4 (] [3
T] [9 /] [4 (] [3 T] [9 /]
current token stream:
                                ( int_num - real_num ) ) ) ) end
output:
                                shift 4

current parsing stack:
                                [0 end] [4 (] [3 T] [10 *] [4 (] [3
T] [9 /] [4 (] [3 T] [9 /] [4 (]
current token stream:
                                int_num - real_num ) ) ) ) end
output:
                                shift 5

current parsing stack:
                                [0 end] [4 (] [3 T] [10 *] [4 (] [3
T] [9 /] [4 (] [3 T] [9 /] [4 (] [5 int_num]
current token stream:
                                - real_num ) ) ) ) end
output:
                                reduce F -> int_num
current subexpression:
                                9
                                type = INT_CONSTANT
                                value = 9

current parsing stack:
                                [0 end] [4 (] [3 T] [10 *] [4 (] [3
T] [9 /] [4 (] [3 T] [9 /] [4 (] [2 F]
current token stream:
```

```
                                - real_num ) ) ) ) end
output:
                                reduce T -> F
current subexpression:
                                9
                                type = INT_CONSTANT
                                value = 9

current parsing stack:
                                [0 end] [4 (] [3 T] [10 *] [4 (] [3
T] [9 /] [4 (] [3 T] [9 /] [4 (] [3 T]
current token stream:
                                - real_num ) ) ) ) end
output:
                                reduce E -> T
current subexpression:
                                9
                                type = INT_CONSTANT
                                value = 9

current parsing stack:
                                [0 end] [4 (] [3 T] [10 *] [4 (] [3
T] [9 /] [4 (] [3 T] [9 /] [4 (] [11 E]
current token stream:
                                - real_num ) ) ) ) end
output:
                                shift 8

current parsing stack:
                                [0 end] [4 (] [3 T] [10 *] [4 (] [3
T] [9 /] [4 (] [3 T] [9 /] [4 (] [11 E] [8 -]
current token stream:
                                real_num ) ) ) ) end
output:
                                shift 6

current parsing stack:
                                [0 end] [4 (] [3 T] [10 *] [4 (] [3
T] [9 /] [4 (] [3 T] [9 /] [4 (] [11 E] [8 -] [6 real_num]
current token stream:
                                ) ) ) ) end
output:
                                reduce F -> real_num
current subexpression:
                                936000
                                type = REAL_CONSTANT
                                value = 936000

current parsing stack:
                                [0 end] [4 (] [3 T] [10 *] [4 (] [3
T] [9 /] [4 (] [3 T] [9 /] [4 (] [11 E] [8 -] [2 F]
current token stream:
                                ) ) ) ) end
output:
                                reduce T -> F
current subexpression:
                                936000
                                type = REAL_CONSTANT
                                value = 936000

current parsing stack:
                                [0 end] [4 (] [3 T] [10 *] [4 (] [3
T] [9 /] [4 (] [3 T] [9 /] [4 (] [11 E] [8 -] [13 T]
current token stream:
                                ) ) ) ) end
output:
                                reduce E -> E - T
current subexpression:
                                9 - 936000
                                type = REAL_CONSTANT
                                value = -935991

current parsing stack:
                                [0 end] [4 (] [3 T] [10 *] [4 (] [3
T] [9 /] [4 (] [3 T] [9 /] [4 (] [11 E]
current token stream:
                                ) ) ) ) end
output:
                                shift 16

current parsing stack:
                                [0 end] [4 (] [3 T] [10 *] [4 (] [3
T] [9 /] [4 (] [3 T] [9 /] [4 (] [11 E] [16 )]
current token stream:
                                ) ) ) end
output:
                                reduce F -> ( E )
current subexpression:
                                ( -935991 )
                                type = REAL_CONSTANT
                                value = -935991

current parsing stack:
                                [0 end] [4 (] [3 T] [10 *] [4 (] [3
T] [9 /] [4 (] [3 T] [9 /] [14 F]
current token stream:
                                ) ) ) end
output:
                                reduce T -> T / F
current subexpression:
                                1 / -935991
                                type = REAL_CONSTANT
                                value = -1.06839e-06

current parsing stack:
                                [0 end] [4 (] [3 T] [10 *] [4 (] [3
T] [9 /] [4 (] [3 T]
current token stream:
                                ) ) ) end
output:
                                reduce E -> T
current subexpression:
                                -1.06839e-06
                                type = REAL_CONSTANT
                                value = -1.06839e-06

current parsing stack:
                                [0 end] [4 (] [3 T] [10 *] [4 (] [3
T] [9 /] [4 (] [11 E]
current token stream:
                                ) ) ) end
output:
                                shift 16

current parsing stack:
                                [0 end] [4 (] [3 T] [10 *] [4 (] [3
T] [9 /] [4 (] [11 E] [16 )]
current token stream:
                                ) ) end
output:
                                reduce F -> ( E )
current subexpression:
                                ( -1.06839e-06 )
                                type = REAL_CONSTANT
                                value = -1.06839e-06

current parsing stack:
                                [0 end] [4 (] [3 T] [10 *] [4 (] [3
T] [9 /] [14 F]
current token stream:
                                ) ) end
output:
                                reduce T -> T / F
current subexpression:
                                9 / -1.06839e-06
                                type = REAL_CONSTANT
                                value = -8.42392e+06

current parsing stack:
                                [0 end] [4 (] [3 T] [10 *] [4 (] [3
T]
current token stream:
                                ) ) end
output:
                                reduce E -> T
current subexpression:
                                -8.42392e+06
                                type = REAL_CONSTANT
                                value = -8.42392e+06

current parsing stack:
                                [0 end] [4 (] [3 T] [10 *] [4 (]
[11 E]
current token stream:
                                ) ) end
output:
                                shift 16

current parsing stack:
                                [0 end] [4 (] [3 T] [10 *] [4 (]
[11 E] [16 )]
current token stream:
                                ) end
output:
                                reduce F -> ( E )
current subexpression:
                                ( -8.42392e+06 )
                                type = REAL_CONSTANT
                                value = -8.42392e+06

current parsing stack:
                                [0 end] [4 (] [3 T] [10 *] [15 F]
current token stream:
                                ) end
output:
                                reduce T -> T * F
current subexpression:
                                -1 * -8.42392e+06
                                type = REAL_CONSTANT
                                value = 8.42392e+06

current parsing stack:
                                [0 end] [4 (] [3 T]
current token stream:
                                ) end
output:
                                reduce E -> T
current subexpression:
                                8.42392e+06
                                type = REAL_CONSTANT
                                value = 8.42392e+06

current parsing stack:
                                [0 end] [4 (] [11 E]
current token stream:
                                ) end
output:
                                shift 16

current parsing stack:
                                [0 end] [4 (] [11 E] [16 )]
current token stream:
                                end
output:
                                reduce F -> ( E )
current subexpression:
                                ( 8.42392e+06 )
                                type = REAL_CONSTANT
                                value = 8.42392e+06

current parsing stack:
                                [0 end] [2 F]
current token stream:
                                end
output:
                                reduce T -> F
current subexpression:
                                8.42392e+06
```

```
                              type = REAL_CONSTANT
                              value = 8.42392e+06

current parsing stack:
                              [0 end] [3 T]
current token stream:
                              end
output:
                              reduce E -> T
current subexpression:
                              8.42392e+06
                              type = REAL_CONSTANT
                              value = 8.42392e+06

current parsing stack:
                              [0 end] [1 E]
current token stream:
                              end
output:
                              acc
```

7. 分析总结

经分析，测试结果正确。

由于采用语法指导翻译，本次实验在上次的 LR 语法分析基础上加一些翻译动作即可完成，较为轻松。

至此，本学期编译原理课程的实验全部完成，个人在代码架构上坚持按照编译器的工作流程来设计，注重整体性与兼容性，如**词法分析实验完成的词法分析器可为语法分析所用，语法分析实验完成的语法分析器可为语义分析所用**，从而将词法分析、语法分析、语义分析三个实验融为一个大工程，完成了一小部分编译器前端的工作。