

Data Structures and Algorithms

Laboratory Projects

Minimum Requirements on Writing a  
Project Report

# Linear List

詹江岳

Date: 2017-11-12

## Chapter 1: Introduction

问题描述：输入一个算数表达式（中序），包含数字和英文圆括号(), 运算符+、-、\*、/、#（单目负）。输出其对应的后序形式和计算结果。

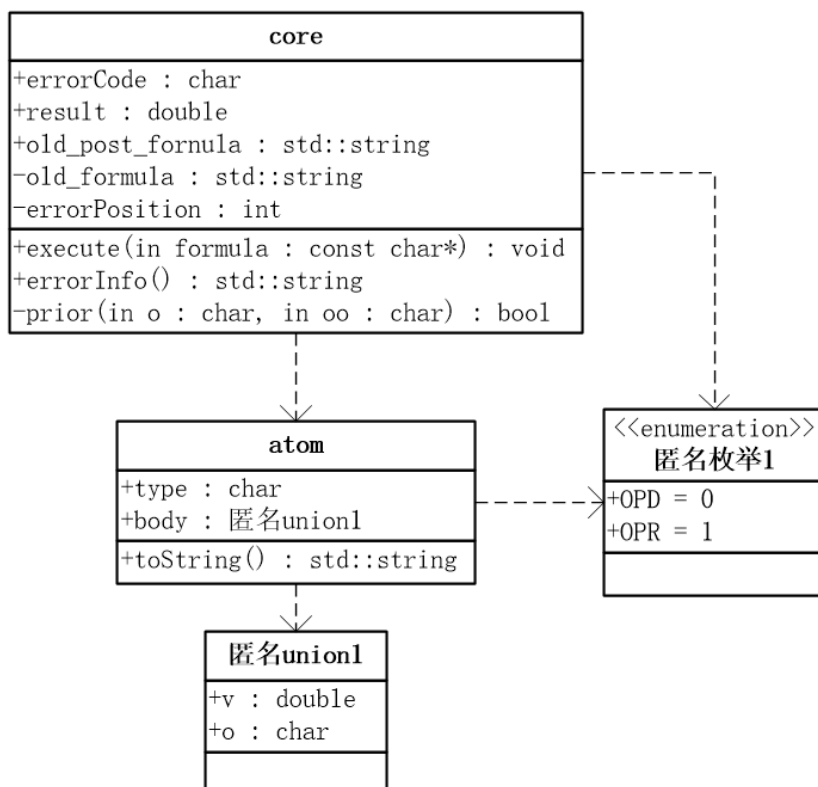
算法背景：后序表达式可以利用栈进行计算，比中序表达式更容易执行。因此，我们先将中序表达式解析成相应的后序表达式，然后再执行计算。

## Chapter 2: Algorithm Specification

- 主要数据结构设计说明

| 数据结构              | 方法                             | 方法说明     |
|-------------------|--------------------------------|----------|
| std::queue STL 队列 | void push(const value_type& x) | 将 x 入队   |
|                   | value_type& front()            | 取出队首的元素  |
|                   | void pop()                     | 删除队首的元素  |
|                   | size_type size() const         | 获取队列长度   |
| std::stack STL 栈  | void push(const value_type& x) | 将 x 入栈   |
|                   | value_type& top()              | 获取栈顶元素   |
|                   | void pop()                     | 删除栈顶元素   |
|                   | size_type size() const         | 获取栈中元素个数 |

- 系统设计思想



- 程序流程图

（见文件夹下文件“算法.jpg”）

## Chapter 3: Testing Results

| 序号 | 模块         | 子模块    | 用例  | 用例描述            | 预期结果  | 实际结果   | 错误估计  | 现状   |
|----|------------|--------|---|-----------------|-------|--------|---|------|
| 1  | 编写代码时使用的用例 |        | 1+#2  | 不含括号的合法表达式      | -1    | 崩溃     | 优先级判定没有判定'#'的优先级, 导致错误的后序表达式, 引起程序崩溃  |      |
| 2  |            |        | 2.23+66                                       | 不含括号的合法表达式      | 68.23 | 语法错误   | 没有处理小数点   |      |
| 3  |            |        | (2)   | 括号包含单个数         | 2     | 语法错误   | 在处理符号时, '('和')'直接相遇, 判断if (lstk.size()    prior(c, stk.top().body.o))导致将')'入栈, 引发错误 |      |
| 4  |            |        | 1+  | 以双目运算符结尾        | 语法错误  | 崩溃     | 对于表达式合法性只是检查前驱, 没有检查后继是否合法。当算符没有后继时解析停止, 没有检测                                       |      |
| 5  | 合法表达式      | 不含括号   | 1/0   | 除以零             | 运行时错误 | 1.#INF | 未检查除以零  | pass |
| 6  |            |        | 1+2.0-02.6*5+#2.5/#10                         | 不含括号的合法表达式      | -9.75 | -9.75  |   |      |
| 7  |            |        | 1+2.0*3-2                                     | 不含括号的合法表达式      | 5     | 5      |   |      |
| 8  |            |        | 1+(2.0-02.6)*(5+#2.5)/#10                     | 含括号的合法表达式       | 1.15  | 1.15   |   |      |
| 9  |            | 括号     | (1+(2.0-02.6)*(5+#2.5)/#10)                   | 括号包含整个表达式       | 1.15  | 1.15   |   |      |
| 10 |            |        | (2)   | 括号包含单个数         | 2     | 2      |   |      |
| 11 |            |        | (2+3)-6                                       | 括号开头            | -1    | -1     |   |      |
| 12 |            |        | 2+(3-6)                                       | 括号结尾            | -1    | -1     |   |      |
| 13 |            | 空格     | ( 1 + ( 2.0 - 02.6 ) * ( 5 + # 2.5 ) / # 10 ) | 含空格的表达式         | 1.15  | 1.15   |   |      |
| 14 |            | 单目负    | 1+#####2                                      | 多重单目负           | -1    | -1     |   |      |
| 15 |            |        | #1+#####2                                     | 单目负开头           | 1     | 1      |   |      |
| 16 |            | 空      | #(2)  | 单目负后接括号         | -2    | -2     |   |      |
| 17 |            |        |   | 空表达式            |       |        |   |      |
| 18 |            | 双目运算符  | +1+2.0*3-2                                    | 以双目运算符开头        |       |        |   |      |
| 19 |            |        | 1+2.0*3-2*                                    | 以双目运算符结尾        |       |        |   |      |
| 20 |            |        | 1+2.0**3-2*                                   | 邻接的双目运算符        |       |        |   |      |
| 21 |            | 单目负    | 1+2.0#3-2*                                    | 单目负连接两个操作数      |       |        |   |      |
| 22 |            |        | 1+2.0*3-2#                                    | 以单目运算符结尾        |       |        |   |      |
| 23 |            |        | 1+2.0#*3-2*                                   | 单目负放在数后面        |       |        |   |      |
| 24 | 非法表达式      | 括号     | ()+2.0*3-2                                    | 空括号开头           |       |        |   |      |
| 25 |            |        | 1+2.0*3-( )                                   | 空括号结尾           |       |        |   |      |
| 26 |            |        | 1+2.0*( )-2                                   | 空括号在中间          |       |        |   |      |
| 27 |            |        | 1+2.0(3)-2                                    | 数后接括号           |       |        |   |      |
| 28 |            |        | 1+(2.0)3-2                                    | 括号后接数           |       |        |   |      |
| 29 |            |        | 1+(2.0)#3-2                                   | 括号后接单目负后接数      |       |        |   |      |
| 30 |            |        | 1+2.0#(3)-2                                   | 数后接单目负后接括号      |       |        |   |      |
| 31 |            |        | ((1+2.0*3-2)                                  | 开头多左括号          |       |        |   |      |
| 32 |            |        | 1+(2.0*(3-2)                                  | 中间多左括号          |       |        |   |      |
| 33 |            | 非合法表达式 | (1+2.0*3-2))                                  | 结尾多右括号          | 语法错误  | 崩溃     | '')入栈, 是符号入栈的检测逻辑有误   |      |
| 34 |            |        | (1+2.0)*3)-2                                  | 中间多右括号          |       |        |   |      |
| 35 |            |        | )(1+2.0*3-2                                   | 数量对应但顺序错误的括号在开头 |       |        |   |      |
| 36 |            |        | 1+2.0*3-2)(                                   | 数量对应但顺序错误的括号在结尾 |       |        |   |      |
| 37 |            |        | 1+2.0)*(3-2                                   | 数量对应但顺序错误的括号在中间 |       |        |   |      |
| 38 |            |        | (1+2.0)+(*3-2)                                | 括号内以双目运算符开头     |       |        |   |      |
| 39 |            |        | (1+2.0)+*(3-2)                                | 括号内以双目运算符结尾     |       |        |   |      |
| 40 |            |        | (1+2.0#)+(3-2)                                | 括号内以单目运算符结尾     |       |        |   |      |
| 41 |            | 数的格式   | .1+2.0*3-2                                    | 开头以小数点开头        |       |        |   |      |
| 42 |            |        | 1+2.0*3-2.                                    | 结尾以小数点结尾        |       |        |   |      |
| 43 |            |        | 1+2.0.*3-2                                    | 中间以小数点开头        |       |        |   |      |
| 44 |            |        | 1+2.0*3.-2                                    | 中间以小数点结尾        |       |        |   |      |
| 45 |            |        | 1+2.0*3.2.1-2                                 | 一个数包含多个小数点      |       |        |   |      |
| 46 |            |        | 1+2.0*3-2                                     | 小数点前空格          |       |        |   |      |
| 47 |            |        | 1+2.0*3-2                                     | 小数点后空格          |       |        |   |      |
| 48 |            |        | 1+2.0 3*32-2                                  | 小数位含空格          |       |        |   |      |
| 49 |            |        | 1+2.0*3 2-2                                   | 整数位含空格          |       |        |   |      |
| 50 | 除以零        |        | 1+2.0*3/0-2                                   | 除以零             | 运行时错误 | 运行时错误  |   |      |
| 51 |            |        | (1+2.0*3-2)/0                                 | 含括号的除以零         |       |        |   |      |

## Chapter 4: Analysis and Comments

### ● 算法分析

时间复杂度: 算法分为将中序表达式转换成后缀表达式和计算后序表达式两部分。对于一个长度为  $n$  的字符串输入, 中序表达式转后序表达式只要遍历一遍字符串, 而对于每个字符的处理用时可以看作是常数。因此, 转换的时间复杂度为  $\Theta(n)$ 。对于计算后序表达式, 它要遍历一遍后序表达式队列, 队列长度等于所有操作数和运算符的总数, 小于  $n$ ; 解析一个操作数或运算符的时间可以看作常数。因此, 整个算法的时间复杂度为  $\Theta(n)$ 。

空间复杂度: 对于一个长度为  $n$  的字符串输入, 只会将完整的操作数和运算符存储在队列或栈中, 且同一个操作数或运算符一个时刻只能存在于一个结构中, 而一个操作数或运算符的空间大小是个

常数  $C$ 。设所有操作数和运算符的总数为  $N$ ，则空间开销  $S = N \times C$ 。又因为  $N \leq n$ ，所以空间复杂度为  $O(n)$ 。

- 不足

可以考虑实现表达式内部变量、全局变量和函数，以及更灵活的语法解析和后序表达式计算过程。

### **Declaration**

*I hereby declare that all the work done in this project titled "Linear List" is of my independent effort as an individual.*

### **Duty Assignments:**

**Programmer:** 詹江岳

**Tester:** 詹江岳

**Report Writer:** 詹江岳