



廣東工業大學

数据结构课程设计报告

题目：表达式类型的实现（难度系数：1.2）

学 院 计算机
专 业 计算机科学与技术
年级班别 2015 级 8 班
学 号 3115005210
学生姓名 杨嘉慧
指导教师 李杨
编 号 _____
成 绩 _____

2017 年 1 月

报告:

- 报告内容: 详细 完整 基本完整 不完整
- 设计方案: 非常合理 合理 基本合理 较差
- 算法实现: 全部实现 基本实现 部分实现 实现较差
- 测试样例: 完备 比较完备 基本完备 不完备
- 文档格式: 规范 比较规范 基本规范 不规范

答辩:

- 理解题目透彻，问题回答流利
- 理解题目较透彻，回答问题基本正确
- 部分理解题目，部分问题回答正确
- 未能完全理解题目，答辩情况较差

总评成绩:

- 优 良 中 及格 不及格

运行环境: CodeBlocks

完成的题目: 表达式类型的实现 (难度系数: 1.2)

选做的内容: (4) 在表达式内增加对三角函数等初等函数的操作。

一、需求分析【课程设计要求】

【问题的描述】

一个表达式和一棵二叉树之间, 存在着自然的对应关系。写一个程序, 实现基于二叉树表示的算术表达式 Expression 的操作。

【基本要求】

【一】【必做部分】

假设算术表达式 Expression 内可以含有变量 (a-z), 常量 (0-9) 和二元运算符 (+, -, *, /, ^ (乘幂))。实现以下操作:

(1) ReadExpr(E) —— 以字符序列的形式输入语法正确的前缀表达式并构造表达式 E。

(2) WriteExpr(E) —— 用带括号的中缀表达式输出表达式 E。

(3) Assign(V, c) —— 实现对变量 V 的赋值 (V=c), 变量的初值为 0。 (4) Value(E) —— 对算术表达式 E 求值。

(5) CompoundExpr(p, E1, E2) —— 构造一个新的复合表达式 (E1) p (E2)。

【二】【选做部分】

(1) 以表达式的原书写形式输入, 支持大于 0 的正整数常量;

(2) 增加常数合并操作 MergeConst(E) —— 合并表达式 E 中所有常数运算。例如, 对表达式 $E=(2+3-a)*(b+3*4)$ 进行合并常数的操作后, 求得 $E=(5-a)*(b+12)$

(3) 增加对求偏导数的运算 Diff(E, V) —— 求表达式 E 对 V 的导数

(4) 在表达式内增加对三角函数等初等函数的操作。

【测试数据】

(1) 分别输入 0; a;-91;+a*bc;+*5x2*8x;+++3^*2^x2x6 并输出。

(2) 每当输入一个表达式后, 对其中的变量赋值, 然后对表达式求值。

二、【概要设计】

1、数据类型的声明:

在这个课程设计中, 采用了链表二叉树的存储结构, 以及两个顺序栈的辅助存储结构

```
/*头文件以及存储结构*/
```

```
#include<stdio.h>
```

```
#include<conio.h>
```

```
#include<stdlib.h>
```

```
#include<string.h>
```

```
#define TRUE 1
```

```
#define FALSE 0
```

```
#define OK 1
```

```
#define ERROR 0
```

```
typedef int Status;
```

2、表达式的抽象数据类型定义

基本操作:

```
void judge_str(&E, &string1)
```

初始条件: 树 E 存在, 表达式的前缀字符串 string 存在;

操作结果: 判断字符 string[i], 如果是'0'-'9' 常量之间, 二叉树结点 E 存为整型; 否则, 存为字符型。

```
Status ReadExpr(&E, &string1)
```

初始条件: 表达式的前缀形式字符串 exprstring 存在;

操作结果: 以正确的前缀表示式 exprstring 并构造表达式 E, 构造成功, 返回 OK, 否则返回 ERROR。 Status Pri_Compare(c1, c2)

初始条件: c1 和 c2 是字符;

操作结果: 如果两个字符是运算符, 比较两个运算符的优先级, c1 比 c2 优先, 返回 OK, 否则返回 ERROR。

```
void WriteExpr(&E)
```

初始条件: 表达式 E 存在;

操作条件: 用带括弧的中缀表达式输入表达式 E。

```
void Assign(&E, V, c)
```

初始条件: 表达式 E 存在, flag 为标志是否有赋值过;

操作结果：实现对表达式 E 中的所有变量 V 的赋值 (V=c)。

long Operate(opr1, opr, opr2)

初始条件：操作数 opr1 和操作数 opr2 以及操作运算符 opr；

操作结果：运算符运算求值，参数 opr1, opr2 为常量，opr 为运算符，根据不同的运算符，实现不同的运算，返回运算结果。

Status Check(E)

初始条件：表达式 E 存在；

操作结果：检查表达式 E 是否还存在没有赋值的变量，以便求算数表达式 E 的值。

long Value(E)

初始条件：表达式 E 存在；

操作结果：对算术表达式求值，返回求到的结果。

void CompoundExpr(P, &E1, E2)

初始条件：表达式 E1 和 E2 存在；

操作条件：构造一个新的复合表达式 (E1)P(E2)。

3、整体设计

在这个课程设计中，有一个源代码文件：expression.c。

在 expression.c 文件中，是实现课程设计要求的各个函数。

主程序的流程以及各程序模块之间的调用关系：

- 1、各个存储结构的声明；
- 2、顺序栈的基本操作。其基本操作如下：

对于栈 SqStack：

```
Status InitStack(SqStack *S) /* 构造一个空栈 S */
```

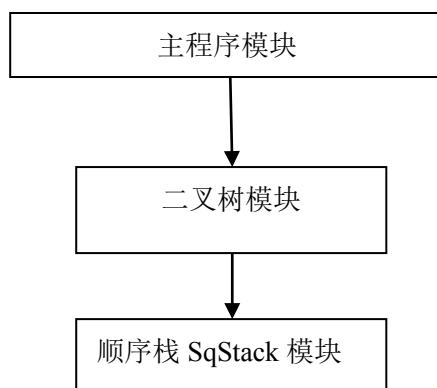
```
Status StackEmpty(SqStack S) /* 若栈 S 为空栈，则返回 TRUE，否则返回 FALSE */
```

```
Status Push(SqStack *S, SElemType e) /* 插入元素 e 为新的栈顶元素 */
```

```
Status Pop(SqStack *S, SElemType *e) /* 若栈不空，则删除 S 的栈顶元素，用 e 返回其值，并返回 OK；否则返回 ERROR */
```

```
Status GetTop(SqStack S, SElemType *e) /* 若栈不空, 则用 e 返回 S 的栈顶元素, 并返回 OK; 否则返回 ERROR */
```

3、本程序有三个模块，主程序模块，二叉树模块，一个顺序栈模块。三者者的调用关系如下：



三、【详细设计】

1、二叉树的存储类型 /*二叉树结点类型*/

```
typedef enum{INT, CHAR} ElemTag; /*INT 为整型数据 num, CHAR 为字符型数据 c*/
```

```
typedef struct TElemType
```

```
{ ElemTag tag; /*{INT, CHAR} 指示是整型还是字符型*/
```

```
union {
```

```
int num; /*tag=INT 时, 为整型*/
```

```
char c; /*tag=CHAR 时, 为字符型*/
```

```
};
```

```
} TElemType; /*二叉树的二叉链表存储表示 */
```

```
typedef struct BiTNode {
```

```
TElemType data;
```

```
struct BiTNode *lchild, *rchild; /* 左右孩子指针 */ }BiTNode, *BiTree;
```

二叉树的基本操作已经在构造表达式和表达式中的基本操作中根据不同的功能和实际 情况修改了，详细见各个函数操作的算法设计。

2、顺序栈的存储类型 /*栈的顺序存储表示 */

```
#define STACK_INIT_SIZE 10 /* 存储空间初始分配量 */
```

```

#define STACKINCREMENT 2 /* 存储空间分配增量 */

/*顺序栈*/

typedef struct SqStack {

SElemType *base; /* 在栈构造之前和销毁之后, base 的值为 NULL */
SElemType *top; /* 栈顶指针 */

int stacksize; /* 当前已分配的存储空间, 以元素为单位*/

}SqStack; /* 顺序栈 SqStack */

```

3、表达式的基本操作

```

Status Input_Expr(char *string, int flag);

```

```

/*以字符序列的形式输入语法正确的前缀表达式, 保存到字符串 string*/ /*
参数 flag=0 表示输出的提示信息是"请输入正确的前缀表示式: "*/
/*flag=1 表示输出的提示信息为"请以表达式的原书写形式输入正确表示式:
"*/

```

```

void judge_str(BiTree *E, char *string, int i);

```

```

/*判断字符 string[i], 如果是'0'-'9' 常量之间, 二叉树结点存为整型; 否
则, 存为字符型*/

```

```

Status Pri_Compare(char c1, char c2);

```

```

/*如果两个字符是运算符, 比较两个运算符的优先级, c1 比 c2 优先, 返回
OK, 否则返回 ERROR*/

```

```

void WriteExpr(BiTree E); /*用带括弧的中缀表达式输入表达式*/ void
Assign(BiTree *E, char V, int c, int *flag);

```

```

/*实现对表达式中的所有变量 V 的赋值(V=c), 参数 flag 为表示是否赋值过
的标志*/

```

```

long Operate(int opr1, char opr, int opr2);

```

```

/*运算符运算求值, 参数 opr1, opr2 为常量, opr 为运算符, 根据不同的运算
符, 实现不同的运算, 返回运算结果*/ Status Check(BiTree E); /*检查表
达式是否还存在没有赋值的变量, 以便求算数表达式的值*/

```

```

long Value(BiTree E); /*对算术表达式求值*/

```

```

void CompoundExpr(char P, BiTree *E1, BiTree E2);

```

```

/*构造一个新的复合表达式*/

```

4、主程序和其他伪码算法

```
void main() {  
  
    BiTree E1, E2;  
  
    char V, P;  
  
    int c;  
  
    ReadExpr(&E1);  
  
    printf("\nE1 带括弧的中缀表示式为: ");  
  
    WriteExpr(E1);  
  
    while(Check(E1)==TRUE) {  
  
        printf("\n 请输入要赋值的字符: ");  
  
        V=getchar();  
  
        printf("请输入要将赋值为: ");  
  
        scanf("%d",&c);  
  
        Assign(&E1, V, c);  
  
        getchar();  
  
        WriteExpr(E1);  
  
        printf("\n 输入未知数后 E1 表达式为: ");  
  
        WriteExpr(E1);  
  
    }  
  
    printf("\nE1 表达式的值为: %d", Value(E1));  
  
    ReadExpr(&E2);  
  
    printf("\nE2 带括弧的中缀表示式为: ");  
  
    WriteExpr(E2);  
  
    Assign(&E2, V, c);  
  
    CompoundExpr(P, &E1, E2);  
  
}
```


5、函数的调用关系

除了主函数 `main()` 外，其他各个函数相对于其它函数来说是独立的，函数的使用都由主函数 `main()` 调用使用的，可以简单的说，各个函数都是主函数下的从函数。

四、【调试分析】

1. 开始设计时我设想建树时可以设定五个域，左右孩子，标志 `tag`，`int` 型值域，`char` 型值域。但是在存储时发现每个字符只需占一个域就可以，所以我又采用共同体这样节约了内存。

2. 在算法设计中，构造表达式树的时候，本来以为使用递归构造表达式会很难做到出错处理的，所以采用了顺序栈辅助构造方法，并且尽可能地对程序进行完善，出错处理。但是经过与同学的相互讨论和研究，发现自己的想法犯了很大的错误，递归构造表达式对于出错处理很简单也很完善，这一点让我加深了递归的使用和理解。

3. 也就是三角函数问题，我最头疼的地方。首先开始运行时会出现错误，无法输出正确结果。通过网上搜索，我发现对于三角函数的定义类型必须是 `double`，这样的话，如果要改的话，差不多改大半程序，所以我就让此功能单独出来，由提示让用户手动完成。

4. 在调试的过程中，花费时间最为多的是对输入错误表达式的出错处理，更改增加的代码几乎都是为了出错处理，但是，觉得这样的调试才更能锻炼一个人的编程能力。

五、【用户使用说明】

打开程序，按屏幕上的提示输入数据，随后就可以看到结果了。

六、【测试结果】

1. 输入 0



2. 输入 a

```
D:\c文件\keshe.exe
请输入语法正确的前缀表示式: a
E1带括弧的中缀表示式为: a
表达式中仍存在变量没有赋值! 没法求出表达式的值!
请输入要赋值的字符: a
请输入要将赋值为: 2
2
输入未知数后E1表达式为: 2
E1表达式的值为: 2
```

3. 输入-91

```
D:\c文件\keshe.exe
请输入语法正确的前缀表示式: -91
E1带括弧的中缀表示式为: 9-1
E1表达式的值为: 8
```

4. 输入+a*bc

```
D:\c文件\keshe.exe
请输入语法正确的前缀表示式: +a*bc
E1带括弧的中缀表示式为: a+b*c
表达式中仍存在变量没有赋值! 没法求出表达式的值!
请输入要赋值的字符: a
请输入要将赋值为: 1
1+b*c
输入未知数后E1表达式为: 1+b*c
表达式中仍存在变量没有赋值! 没法求出表达式的值!
请输入要赋值的字符: b
请输入要将赋值为: 1
1+1*c
输入未知数后E1表达式为: 1+1*c
表达式中仍存在变量没有赋值! 没法求出表达式的值!
请输入要赋值的字符: c
请输入要将赋值为: 1
1+1*1
输入未知数后E1表达式为: 1+1*1
E1表达式的值为: 2
```

5. 输入+*5x2*8x

```
D:\c文件\keshe.exe
请输入语法正确的前缀表示式: +*5x2*8x
输入的表达式有误!
E1带括弧的中缀表示式为: 5*x+2
表达式中仍存在变量没有赋值! 没法求出表达式的值!
```

6. 输入+++3^x3*2^x2x6

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。

如要下载或阅读全文，请访问：

<https://d.book118.com/886024232102010213>