

文章编号: 2096-1472(2016)-02-16-03

数学解题方法与程序设计关系的研究

迟呈英¹, 王玄同², 王子涵³, 战学刚¹

(1. 辽宁科技大学软件学院, 辽宁 鞍山 114051;

2. 中国海洋大学工程学院, 山东 青岛 266100;

3. 中国海洋大学外国语学院, 山东 青岛 266100)

摘要: 在实践过程中, 人们往往忽略了数学的重要性。我们通过理论分析和编程实例, 对数学解题方法与程序设计过程进行对比, 讨论了数学解题方法与程序设计关系, 阐述了数学思维在程序设计中的重要性。

关键词: 数学解题方法; 程序设计; 递归函数

中图分类号: TP311.1

文献标识码: A

Studies on Relationship between Mathematical Methods and Programming

CHI Chengying¹, WANG Xuanton², WANG Zihan³, ZHAN Xuegang¹

(1. School of Software Engineering, University of Science and Technology Liaoning, Anshan 114051, China;

2. College of Engineering, Ocean University of China, Qingdao 266100, China;

3. College of Foreign Languages, Ocean University of China, Qingdao 266100, China)

Abstract: In practice, the importance of mathematics is often ignored. In this paper, through theoretical analysis and practical programming examples, we compare the mathematical methods and programming process to reveal the relationship between them and the importance of mathematical thinking in programming.

Keywords: mathematical methods; programming; recursive function

1 引言(Introduction)

计算机程序, 简称程序, 是指一组指示计算机每一步动作的指令, 用某种程序设计语言编写, 运行于某种计算机之上。与程序相关的另一个概念是软件。软件包括程序、支撑程序运行所需的数据资源、以及与这些程序相关的文档。程序设计是给出解决特定问题程序的过程, 程序设计过程通常包括分析、设计、编码、测试、排错等不同阶段。程序设计语言是人与计算机交流的工具。当然, 从工程角度来看, 它也是软件人员之间的交流工具。

澄清了程序、程序设计、程序设计语言等概念后, 我们换一个角度考察机器, 即计算机。从机器功能方面, 我们可以将计算机看作是具有很大大存储容量的计算机, 其计算功能是通过自然数系统、布尔代数、整数系统和有理数系统完成的。

自然数系统和布尔代数是基本的, 整数系统和有理数系统可以由前两项通过编码完成(事实上, 早期的计算机就是这样做的), 也可以由硬件直接实现。这样, 我们可以把程序看作是上述几个代数系统对计算过程的描述。当然, 我们所使用的计算机作为具体计算过程的执行者, 其本身的特性是程序设计语言和程序设计者必须考虑的。正因为如此, 我们

可以将程序设计语言看作是受计算机特性约束的数学语言。

2 数学解题方法和程序设计过程对照(Comparison between mathematical methods and programming process)

澄清了上述的概念, 我们就很容易接受这样的观点: 程序设计实质上是数学解题过程^[1]。程序设计语言只是为我们提供了描述工具。下面我们几个例子来对照一下。

[例1]求一元二次方程的实根。

我们在中学时代就熟知求解一元二次方程的方法, 对于方程:

$$ax^2+bx+c=0$$

先计算其判别式 $d=b^2-4ac$ 。

如果 $d<0$, 则方程无实数解; 否则,

$$x=\frac{-b\pm\sqrt{d}}{2a}$$

将相应的计算过程翻译成C++语言^[2], 便得到了如下代码。

```
bool quadratic(double a, double b, double c, double&
x1, double& x2)
{
```

```
double d=b*b-4*a*c; //判别式
if(d<0)
return false;          //判别式小于零无解
double sqrt d=sqrt(d);
x1=(-b-sqrt d)/(2*a); //判别式大于等于零有解
x2=(-b+sqrt d)/(2*a);
return true;
}
```

[例2]打印菱形，要求用星号打印如下的菱形。

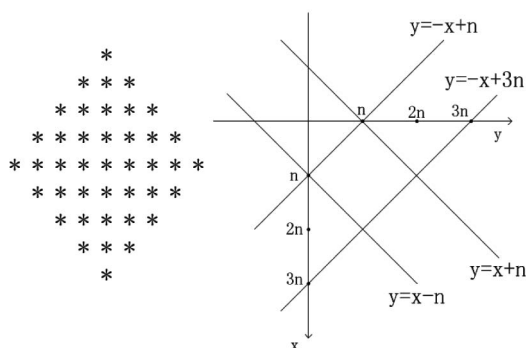


图1 打印星号与对应的几何图形

Fig.1 Star printing and corresponding geometric figure

几乎所有的C/C++语言教材中都有此内容，但是没有一本书籍详细介绍其设计思路，学生也是死记硬背书中的循环控制方法，至于为什么采用此方法控制循环，教师在授课过程中也没有讲述清晰。实际上该问题就是中学所学的解析几何的应用。

问题分析：我们将控制台屏幕看作是以左上角为原点，向下方向为x轴正向，向右方向为y轴正向的直角坐标系，则上面菱形星阵的四个边所在的直线方程分别为： $y=x+n$ ； $y=-x+n$ ； $y=x-n$ ； $y=-x-3n$ 。

这四条直线圈定了我们所要打印的空格和星号的范围，如图1所示。

```
void diamond(int n) // *号的上半部分
{
for (int x=0;x<=n;x++)
{
for (int y=0;y<=n-x;y++)
cout<<" ";
for (int y=n-x;y<=n+x;y++)
cout<<"*";
}
```

```
cout<<endl;
}
for (int x=1;x<=n;x++) // *号的下半部分
{
for (int y=0;y<=x;y++)
cout<<" ";
for (int y=x;y<=2*n-x;y++)
cout<<"*";
cout<<endl;
}
}
```

类似地，我们可以处理许多这样的例子，其中包括打印杨辉三角形等。

3 递归函数(Recursive functions)

递归的概念与自然数的归纳性质有着紧密的联系，因此我们先介绍自然数的一种定义形式^[3]。

自然数集合N一种定义形式(Peano公理)，就是递归定义的^[4]。定义中假定后继函数S存在。

- (1) $0 \in N$;
- (2) 如果 $x \in N$ ，则 $S(x) \in N$;
- (3) 如果 $x \in N$ ，则 $S(x) \neq 0$;
- (4) 如果 $S(x)=S(y)$ ，则 $x=y$;
- (5) 如果M是N的子集，并且 $0 \in M$ ，如果 $x \in M$ ，则 $S(x) \in M$ ；则 $M=N$ 。

其中的最后一条称为归纳公理，它是数学归纳法的理论基础。

这种定义形式的重要作用是，它使我们能够用有限的语句描述无穷集合。计算机系统完全实现了自然数系统的基本功能(当然存在最大数的限制)。因此，我们可以在计算机程序中使用自然数系统的任何性质。

如果我们把S定义如下：

$$S(0)=1, S(1)=10, S(10)=11, S(11)=100 \dots$$

则给出了自然数的二进制表示。类似地，我们也可以给出其他进制的表示形式。

自然数加法的定义：

$$x+0=x;$$

$$x+S(y)=S(x+y).$$

当我们用二进制表示时，

$$1+1=1+S(0)=S(1+0)=S(1)=10。$$

自然数减法的定义：

如果 $z=x+y$ ，则 $y=z-x$ 。

小于关系的定义：

如果 $z=x+y$ ，且 $y \neq 0$ ，则 $x < z$ 。等价地， $z > x$ 。

自然数乘法的定义：

$$x*0=0；$$

$$x*S(y)=x*y+x。$$

与它等价的另一种形式是：

$$x*y = \begin{cases} 0 & \text{如果 } y=0 \\ x*(y-1)+x & \text{如果 } y>0 \end{cases}$$

将这种递归函数(运算是函数的另一种说法)用程序设计语言描述。

根据C/C++的语法规则，上述数学形式可以写为如下形式。

[例3]自然数乘法。

```
int Mult(int x, int y)
{
    if (y==0) return 0;
    return Mult(x,y-1)+x;
}
```

相应的非递归形式：

```
int mult(int x, int y)
{
    int p=0;
    while (y>0)
    {
        p+=x;
        y--;
    }
    return p;
}
```

递归函数虽然有直观简洁的优点，但在运行时耗费系统资源。好在任何递归函数都可以转换成非递归(循环)形式。Bohm和Jacopini于1966年证明了控制结构组(顺序、分支、循环)与(顺序、分支、递归)是等价的^[5]。

[例4]Hanoi塔问题。

问题描述：在Hanoi这个地方有一个寺庙，这里有3根柱子和64个大小不同的金碟子。每个碟子有一个孔可以穿过。所有的碟子都放在第一个柱子上，而且按照从上到下碟子的大小依次增大的顺序摆设，如图2所示。

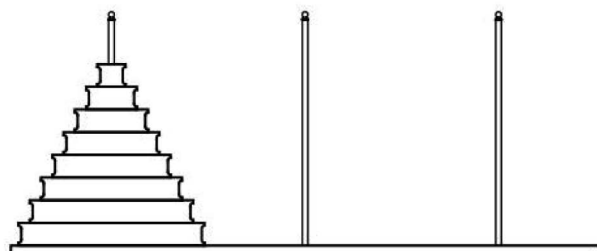


图2 Hanoi塔

Fig.2 Tower of Hanoi

现在，假定寺庙里的僧侣要移动这些碟子，将它们从最左边移动到最右边的柱子上。其移动的规则如下：

(1)每次只能从一个柱子的最上面移动一个碟子到另外一个柱子上。

(2)不能将大碟子放到小碟子的上面。

下面的描述本身就是对问题可解的数学归纳法证明。

$n=0$ 时，无须任何操作，任务完成。

$n>0$ 时，我们假设对于 $n-1$ 个盘子问题可解，那么，对于 n 个盘子，我们可以将 $n-1$ 个盘子由1号柱移动到2号柱；将第 n 个盘子由1号柱移动到3号柱；将 $n-1$ 个盘子由2号柱移动到3号柱。

将上述过程用C++描述，便得到如下的程序。

```
void Hanoi(int n,int p1,int p2,int p3)
{
    if(n>0)
    {
        Hanoi(n-1,p1,p3,p2);
        cout<<p1<<"=="<<p3<<endl;
        Hanoi(n-1,p2,p1,p3);
    }
}
```

4 结论(Conclusion)

数学是各门科学的基础，起初计算机科学只是数学的一个分支，随着时代的发展，人们越来越理解计算机与数学是

(下转第6页)