

# 浅谈递归程序的分析方法<sup>\*</sup>

朱恩亮 陈瑶琪

(盐城工学院广电处,盐城,224003)

**摘要** 通过实例介绍嵌套分析法、“回推”和“递推”分析法两种分析递归程序的方法,旨在对递归程序的执行过程有较深刻的理解。

**关键词** 递归程序 嵌套 回推 递推 分析

**分类号** TP3

一个函数或过程,如果在它们定义的内部又出现定义本身的应用,则称它们是递归的。递归程序在完成阶乘运算、级数运算、幂指数函数运算以及一些特殊的数学问题求解等方面特别有效。PASCAL 和 C 等计算机高级语言都允许采用递归算法设计程序。对于递归程序如何进行分析呢?本文拟用“嵌套分析”、“回推”和“递回”两种方法进行分析。

## 1 递归程序的嵌套分析法

嵌套是一层套一层的程序结构。比如,过程 A 套住过程 B,过程 B 套住过程 C,过程 C 套住过程 D……。嵌套中要强调的是必须完全套住,局部套住是不行的,即不允许有两个或两个以上的过程彼此交叉,或称相互“跨骑”的现象。递归程序在执行过程中,遇到函数或过程调用语句时,就转去执行相应的函数或过程,同时进行实参到形参的传递,在执行完被调函数或过程后返回原来的程序。从分析程序的角度来看,调用函数或过程相当于将被调函数或过程包含(或形象地设想为搬迁)到主调函数或过程的相应位置上来。下面通过一个 PASCAL 语言的过程递归调用程序,用嵌套的分析法来说明程序的执行过程。

### 1.1 PASCAL 源程序

```
PROGRAM example;  
VAR m:integer;  
PROCEDURE print(w:integer);  
VAR i:integer;
```

```
BEGIN  
IF w>0 THEN  
BEGIN  
    print(w-1); (* 递归调用 *)  
    FOR i:=1 to w Do write(w,3);  
    writeln;  
    print(w-1) (* 递归调用 *)  
END  
END;  
BEGIN  
    m:=2;  
    print(m) (* 过程调用 *)  
END.
```

### 1.2 嵌套分析

(见图 1)

### 1.3 程序运行结果

```
1  
2 2  
1
```

## 2 递归程序的“回推”和“递推”分析法

对于递归问题的求解,可以分成两个阶段,第一阶段是“回推”,即要知道问题的现在结论必须先知道其前期结论,而要知道其前期结论又要知道其前期结论的前期结论,如此下去,直到问题的原始已知条件。第二阶段是“递推”,即由一个结论可以推出其下一个结论,有了下一个结论又可以推出其下一个结论,直到求解出问题。在进行某一

\* 收稿日期:1998-10-21

```

BEGIN
m:=2;
print(2); 将实参2传给形参w, 调用print过程。
print(1):
  print(0):
    w不大于0. 返回
  FOR i:=1 To 1 DO write(1:3):writeln;
  print(0);
  w不大于0. 返回
FOR i:=1 To 2 DO write(2:3):writeln;
print(1):
  print(0);
  w不大于0. 返回
  FOR i:=1 To 1 DO write(1:3):writeln;
  print(0);
  w不大于0. 返回
END.

```

图1 嵌套分析图

次函数或过程调用时,并不是立即得到结果,而是一次又一次地进行递归调用,直到问题的原始已知条件,然后再“递推”到问题的本身,得出结论。

19 世纪欧洲人提出的汉诺塔(Tower of Hanoi)问题就是一个典型的只有用递归方法(而不能用其它方法)解决的问题。问题是这样的:有三根针 A、B、C。A 针上有 64 个盘子,盘子大小不等,大的在下,小的在上。要求把这 64 个盘子从 A 针移到 C 针,在移动过程中可以借助 B 针,每次只允许移动一个盘,且在移动过程中在三根针上都保持大盘在下,小盘在上。关于此问题的求解思路和编程方法见参考文献 2,下面给出此问题的 C 语言源程序及其分析方法。

### 2.1 汉诺塔问题的 C 语言源程序

```

Void move(getone,putone)
Char getone,putone;
{
printf("%c-->%c\n",getone,putone);
}
Void hanoi(n,one,two,three)
/* 将 n 个盘子从 one 借助 two 移到 three

```

```

*/
Char one,two,three;
int n;
{
if (n>0)
{ hanoi(n-1,one,three,two);
move(one,three);
hanoi(n-1,two,one,three);}
}
main()
{ int m;
printf("input the number of diskess:");
Scanf("%d",&m);
printf("The step to moving %3d diskess:\n",m);
hanoi(m,'A','B','C');
}

```

### 2.2 运行过程分析

设 A 针上放 3 个盘子,采用递归程序的“回推”和“递推”分析法分析如图 2。

### 2.3 程序运行情况

```

input the number of diskess:3<Enter>
The step to moving 3 diskess:
A-->C
A-->B
C-->B
A-->C
B-->A
B-->C
A-->C

```

### 3 结束语

上述递归程序的两种分析方法,其本质上是一致的。另外,对于 PASCAL 程序来说,过程调用能作为单独的语句使用,而函数调用只能出现在表达式中,所以过程调用的返回是返回到主调语句的下一语句继续程序执行,而函数调用的返回是返回到原表达式处。对 C 程序来说,不存在过程调用,函数调用既可以单独作为一条语句,也可以作为表达式的组成部分。

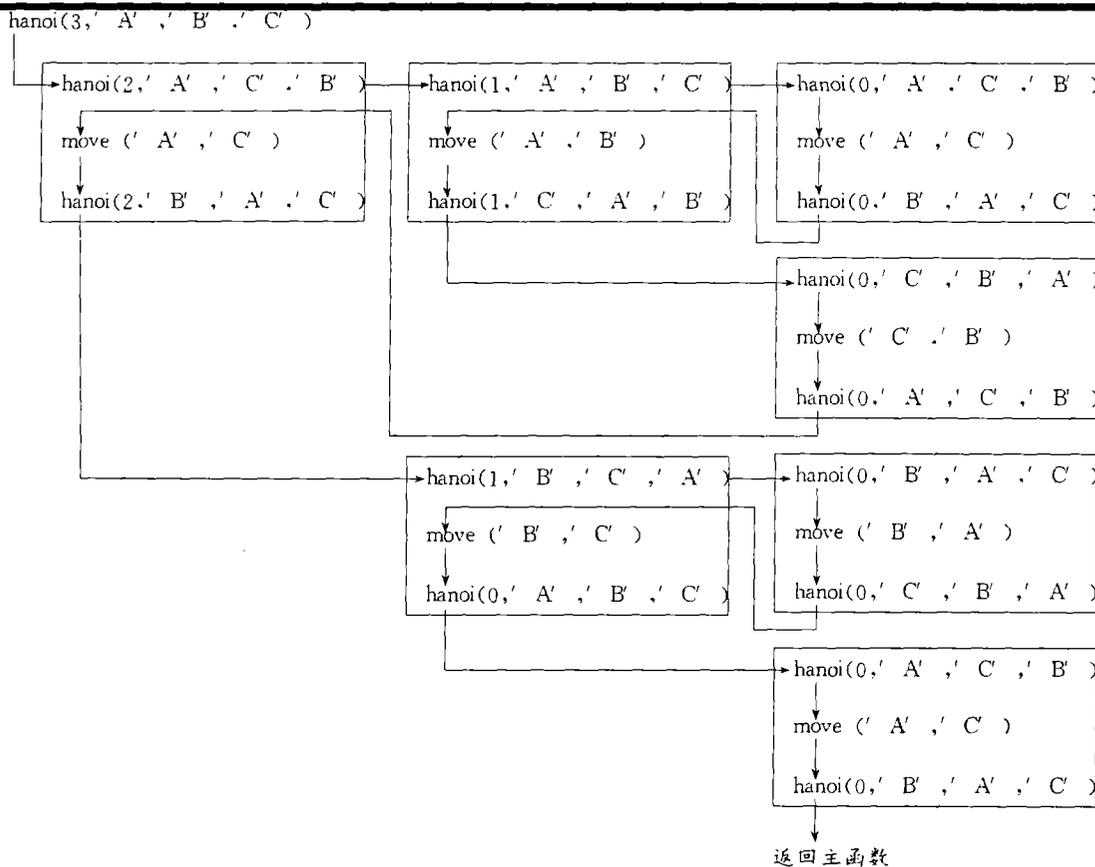


图2 “回推”和“递推”法分析图

参 考 文 献

- 1 朱敏. PASCAL 程序设计教程. 南京: 东南大学出版社, 1995. 120
- 2 谭浩强. C 程序设计. 北京: 清华大学出版社, 1991. 115~120
- 3 王诚. PASCAL 程序设计及其应用. 北京: 清华大学出版社, 1983. 130

# A First Study of the Analysing Methods of Recursive Program

Zhu Enliang    Chen Yaoying

(Dept. of broadcast TV education of Yancheng Institute of Technology, Yancheng, 224003, PRC)

**Abstract** This essay demonstrates with examples two methods of analysing recursive program: First, analysing through the nest, second, analysing through backward inference and progressive inference. The aim of this essay is to deepen the understand of the executing process of recursive inference.

**Keywords** recursive program; nest; backward inference; progressive inference; analyse