

程序设计语言的循环结构跟踪分析法

周 健,夏 日,孙丽艳

(安徽财经大学 管理科学与工程学院,安徽 蚌埠 233041)

摘 要:循环结构是高级计算机语言中的重要结构,由于语句跳转和变量频繁变化使得循环结构成为程序设计语言教学的难点,学生难于从循环结构中分析出正确的结果。针对该问题,提出一种循环结构的跟踪分析方法,以图表的方式跟踪循环结构和变量,顺序的方式描述变量变化规律,掌握变量变化之间的关系,使得教师教学和学生同步,提高学生分析程序的正确率和效率。通过教学和实验分析证明了该方法的有效性,取得了良好的教学效果,能够培养学生独立分析程序的能力。

关键词:程序设计;循环结构;效率;语意分析;教学效果

中图分类号:TP39

文献标识码:A

文章编号:1673-629X(2013)03-0191-04

doi:10.3969/j.issn.1673-629X.2013.03.048

Trace Analysis Method on Iteration Construction for Programming Language

ZHOU Jian, XIA Ri, SUN Li-yan

(School of Management Science and Eng., Anhui Univ. of Finance and Economics, Bengbu 233041, China)

Abstract: Cycle structure is important in advanced programming language. However, it is difficult to teach and comprehend for students and teachers due to jump construct and variable. So cycle structure becomes a teaching difficulty, most of students difficultly consult the right result of cycle structure. In order to deal with the question, put forward a kind of circular structure tracking analysis method, which uses chart to track loop structure and variable, sequence to describe the change rule of variable, grasping the relationship between the variables, and makes the teachers teaching and students learning synchronization. The teaching method enhances the process accuracy and efficiency for students. Through the teaching application and experimental analysis shows that this method is effective, and has achieved good teaching effect.

Key words: program design; iteration construction; efficiency; semantic analysis; teaching effect

0 引言

程序设计语言的学习不仅是信息类学生必须学习的内容,而且也是大学新生基础课程的必修内容^[1]。一些语言已经称为大学生的学习内容,如VB、C、Java等,这些语言大都为高级语言^[2,3],不仅有利于培养学生的计算思维,而且为日后走向工作岗位,打下良好的计算机基础。因此培养学生学习程序语言的学习兴趣,提高程序设计语言的教学质量,改善学生学习方法是程序设计语言任课教师义不容辞的责任和义务^[4]。

然而程序设计语言的学习与其它大学课程比较,具有它显著的特点。这些特点也使得这门课程在学生

学习中遇到不少的问题^[5,6]。首先,程序设计语言是大学生学习的第一门工科性质的课程,大学入学新生往往过于着重理论的学习,较少的动手实践,因此在学习工科课程内容时存在思维模式和学习方法的转变。其次,过多的先入为主的概念,使得学生难于理解,基本数据结构和指针中过多地涉及硬件知识,使得学生无法从本质上了解基础概念和高级内容。最后,课本知识无法满足程序实践的要求,程序的调试和纠错更多地依靠教师的亲手指导。这些都需要教师设计良好的教学方法,为学生和课程学习间建立便捷的桥梁。

基本数据结构和程序语句结构是程序设计语言的两个基础内容,目前大部分程序设计内容都包含这两部分内容,内容大同小异,也是学生最早必须掌握的内容。学生经过这两部分的学习,可以独立设计一些较为完整的程序,因此这两部的学习是决定学生后续内容能否学好的前提,然而该部分的内容学习情况也不是十分乐观。其中的主要问题集中于循环结构^[7]。因

收稿日期:2012-06-21;修回日期:2012-09-26

基金项目:信息管理与信息系统省级特色专业项目;安徽财经大学教研项目(ACJYYB2011104)

作者简介:周 健(1979-),男,安徽凤阳人,讲师,博士研究生,CCF会员,从事高级程序语言设计、数据结构和数据挖掘等方面研究。

此文中提供一种教学方法,增强学生独立准确地分析循环结构的能力。

1 循环结构的分析难点

循环结构是程序设计课程的一个重要内容^[8,9]。从课程设计上说,程序设计语言的循环结构介于基础内容最后和高级内容之前,因此该节内容是一个承上启下的部分。一方面该部分内容学习完成后,学生还不能实现全部语言程序,但是可以独立完成一个比较完整的程序设计。因此,该内容掌握的好坏直接关系到后续内容的进展,也是程序设计语句结构的核心内容。从后续课程上,数据结构,数据库语言设计,网页设计,都会大量地涉及该部分内容。从实践的角度说,该内容后的几乎所有程序都包括循环结构,几乎每个例子中都包括有三种循环结构中的一种,而且多有混合使用,学生如果正确分析程序,并能够独立地完整写出,必须要理解掌握这些循环结构。由此,循环结构内容的理解和掌握是学生能否学好程序设计课程的关键内容^[10,11]。

循环结构的学习有别于其他语句结构学习,在教师学习和学生学习中它有它自己的特点。

(1)结构化的循环结构只包括三种语句(For 子句、While 子句、Do-While 子句),有别于 GOTO 语句,跳转具有一定的规律性;

(2)循环结构往往处于算法的核心部分,一般该结构如果能够分析正确,整体程序分析正确率将会大大提高;

(3)循环结构往往具有较多的变量,这些变量会随着循环结构而变化;

(4)循环结构的内嵌加剧了分析的困难性,一般内嵌的层次为 2 到 3 层,因此一般程序的复杂度为 $\leq O(N^3)$,内嵌结构的多样性,以两层嵌套为例,三种循环结构可以有 9 种不同嵌套方法;

(5)循环结构的终止条件,准确地分析循环条件的成立与否,关系程序分析的正确性;

(6)循环结构的教学过于重视结果,而较少重视内部结构的运行规律分析;

(7)在教学中为了简化循环的步骤,节约教学实践,往往会采取跳跃的步骤。

上述的特点造成学生学习的障碍性。

(1)循环程序分析的复杂性,一方面由于跳转结构,使得循环语句中的语句执行次数较多,容易导致分析中语句的遗漏和过多的执行,从而导致分析失败;另一方面,内嵌情况的出现,更加重了这一情况;

(2)环境变量的多变性,一方面,循环结构中的变量在每次循环中,可能发生变化,也可能不发生变化,

当变量的数量增多时,只要有一个变量分析错误,结果就不正确;另一方面,循环退出条件的准确性也直接关系到结果的正确性;

(3)教师教学和学生之间的非同步性,简化循环的步骤以及过于重视程序分析结果,使得学生不能深刻理解循环结构内容;

(4)缺乏独立分析循环结构的能力,循环结构分析更多的关注结果,使得学生不能找出分析循环结构的共性,从而产生独立分析程序的能力,学生学习陷入死记硬背;

(5)由于作为初学者,程序分析中如果变量和条件的增多,往往使得学生不能准确地跟踪变量和条件的变化过程,进而导致程序分析失误。

由于上述的特点,学生在学习和考试中,往往陷入困难。笔者在教学中发现,学生在面临此类问题时,往往陷入苦思,不仅浪费大量时间,而且往往由于某个环节的错误导致结果的错误。学生为了解决此类问题,往往采用死背和猜测的方法,效果并不好,也违背了程序设计教学的本意,独立分析和设计程序的能力^[12]。

2 跟踪分析法

使用图表的方法比单纯的从语言出发,更容易让学生理解程序的执行过程,这一点在数据结构课程中突出的较为明显,例如用伪代码简化语言环境,以及动态图的方式演示重要环境变量的变化和步骤,可以使得学生尽快地理解算法的核心思想。同时从编程语言集成开发环境中的变量跟踪中,也可以得到启发,利用环境变量的跟踪分析错误环境,是否可以用环境变量的跟踪分析来分析循环变量。

多种嵌套结构的分析方法:

课程教学设计前,强调三种循环结构的跳转规律及它们的区别,明细循环初始条件、循环退出条件和循环增量间的逻辑关系。分析结果,找出循环结构中变量和条件的变化规律,进而总结出循环结构的结果。

步骤 1: 选择恰当的例子,本着由简到难,由复杂度较低到复杂度较高的次序选择程序进行讲解,文中以冒泡排序法程序为样例,程序如下所示(教学中将程序分为三个部分,其中该程序核心分析部分为语句 5 到语句 8 这一部分。因此循环程序跟踪分析法针对这一部分内容)。

```
1: Void main() {  
2:   Int i,j,temp;  
3:   Int a[10];  
4:   For(i=0;i<=10;i++) { Scanf(“%d”,&a[i]);}  
5:   For(j=0;j<=4;j++)  
6:       For(i=0;i<5-j;i++) {
```

表1 跟踪变量表

循环跟踪变量						循环判断条件			
	j	i	temp	$a[i]$	$a[i+1]$	$j \leq 4$	$i < 5 - j$	$a[i] > a[i+1]$	结果集合
1	1	1	34	34	23	T	T	T	{34,23,56,100,5}
2	1	2	34	34	23	T	T	T	{23,34,56,100,5}
3	1	3	×	×	×	T	T	F	{23,34,56,100,5}
4	1	4	100	100	5	T	T	F	{23,34,56,5,100}
5	1	5	×	×	×	T	F	×	{23,34,56,5,100}
6	2	1	×	×	×	T	T	F	{23,34,56,5,100}
7	2	2	×	×	×	T	T	T	{23,34,56,5,100}
8	2	3	56	56	5	T	T	T	{23,34,56,5,100}
9	2	4	×	×	×	T	F	×	{23,34,5,56,100}
10	3	1	×	×	×	T	F	×	{23,34,5,56,100}
11	3	2	34	34	5	T	T	F	{23,34,5,56,100}
12	3	3	×	×	×	T	F	F	{23,5,34,56,100}
13	4	1	23	23	5	T	T	T	{23,5,34,56,100}
14	4	2	×	×	×	T	T	F	{5,23,34,56,100}
15	5	1	×	×	×	F	×	×	{5,23,34,56,100}

7: If (a[i]>a[i+1]){temp=a[i],a[i]
=a[i+1];a[i+1]=temp}
8: }
9: For(int i=0;i<5;i++){printf(“%5d”,a[i])}
10: }

步骤2:分析循环结构的重要变量,以表的形式列出重要的变量,以在语句5和语句8中出现的次序列出,包括 j , i ,temp, $a[i]$, $a[i+1]$,注意初始值,在教学中强调变量之间的变化关系。

步骤3:分析循环结构的重要条件,语句5和语句8之间有两个FOR循环,它们之间的关系为嵌套关系,两个循环结构的判断语句分别为 $j \leq 4$ 和 $i < 5 - j$;另外增加一个IF语句的条件语句 $a[i] > a[i+1]$,在教学中强调条件之间的逻辑因果关系。

步骤4:画出图表,根据步骤2和步骤3得到所有需要列出的跟踪变量和跟踪条件,以表的形式给出。

步骤5:环境变量和条件跟踪,程序的输入为{34,23,56,100,5},如表1所示,在表1的基础上增加循环次数和循环各个阶段的结果。

步骤6:变量跟踪规律总结,列出循环的前几次循环,具体次数以达到分析清楚为目的以初始变量和条件尽快总结出各个变量之间的关系,从而摆脱程序,直接从表得到各个变量间的变化关系。教学分析中强调外层循环规律总结和内存循环规律总结,经过表1跟踪图示:外层循环受到 j 值控制,该值小于5就可以不断地执行内层循环;内存循环受到 j 值和 i 值的联合控制,如果 $i+j < 5$ 执行内层循环,由于外层循环 j 值的

不断增加,内层循环的循环次数不断减少。

步骤7:得到分析结果,通过表1很容易找出各个变量间的关系,如表2所示,以及各变量值和条件的值变化规律:

$$j \rightarrow (j \leq 5) \xrightarrow{T} i \rightarrow (i < 5 - j) \xrightarrow{T} (a[i] > a[i + 1]) \xrightarrow{T} (a[i] \overset{\text{exchange}}{\leq} a[i + 1])$$

表2 各循环变量的作用

j	初始值为1,每次增加1
$j \leq 4$	终止条件为5,外层最大循环5次,条件F是,终止所有变量和条件判断
i	初始值为1,每次增加1
$i < 5 - j$	j 值每次增加1,则内层循环每次减1条件为F时,终止内层循环变量和条件判断
$a[i] < a[i + 1]$	如果条件为真,则 $a[i]$ 和 $a[i + 1]$ 值置换条件F时,保持上次集合结果
集合	比较位置 i 和位置 $i + 1$ 的数组分量,大者置后

该教学方法中,可以适当修剪程序的复杂规模。在学生练习时,首先以复杂度规模较小的程序给予练习,随着分析程度的数量增加,增加程序复杂度。教学中,开始以复杂度较小的程序完全跟踪所有的循环,增加程序复杂度后,着重分析循环的开始的10次以内循环,并配以循环变量变化规律的分析。

3 实践检验

在教学中分别对两个班进行试验,一个班使用该

方法分析和讲解程序,并要求学生使用该方法分析程序,另一个班不使用该方法讲解程序,学生自主分析和理解程序。从作业,考试和课堂分析三个方面考察学生分析理解程序的能力。并统计结果,从理解率,分析程序正确率,分析程序耗费时间,比较验证效果。测试的程序为使用三种不同的循环结构程序,复杂度分别为 $O(N)$, $O(N^2)$, $O(N^3)$ 。

讲解的次数为1到10次,根据问卷形式调查学生的理解程度,如图1所示。

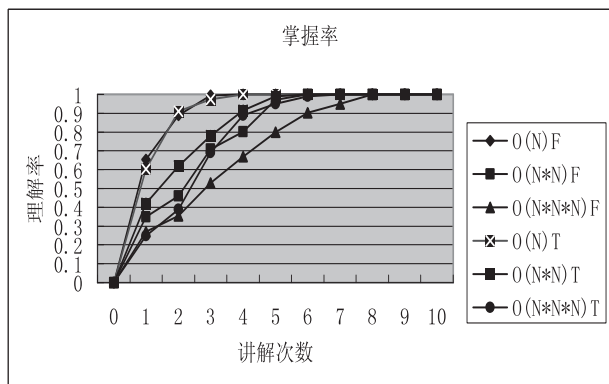


图1 理解程度

让学生分析程序的语意,考察学生准确分析程序的效率,分析前2~10次的准确率,如图2所示。

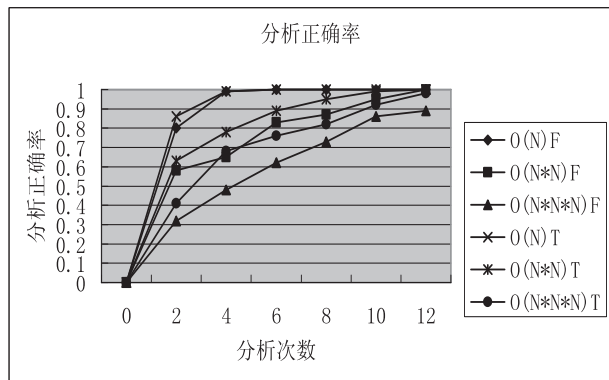


图2 分析正确率

让学生分析程序的语意,考察学生准确分析程序平均花费时间,如图3所示。

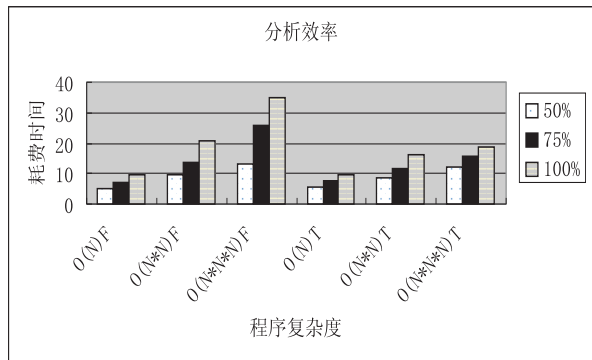


图3 程序复杂度

从上述测试的结果数据图可以看出,该方法可以有效提高学生分析程序的正确率,提高授课的效率,尤其在分析复杂度较高的程序时,该方法具有明显的优越性。同时根据学生的反馈情况,循环程序的规律总结在分析5~6次之前就可以得到,学生可以摆脱程序语句,而根据循环变量跟踪表来推导出程序的运行规律,大大缩短了学生分析程序的时间,并提高他们分析程序的准确率。

4 结束语

文中提出一种循环程序跟踪分析教学方法,利用图表的方式渐进式分析循环程序,使用尽量少的循环步骤总结出各个变量和条件之间的变化关系,提高学生分析程序的能力,使得教师教学和学生理解能够达到同步,尤其适合复杂度较高的程序分析。通过作业和考试形式证明该方案的有效性。在下一步的工作中,将以该方法细化一些经典的循环程序,并提出这些程序的共性,从而提高教学质量,进一步优化循环跟踪表中变量和条件的结构。

参考文献:

- [1] 孙英,徐顺琼,李兴姜. C语言中循环结构程序课的教学设计与探讨[J]. 计算机教育, 2009(12): 186-187.
- [2] 谭浩强. C语言程序设计[M]. 第3版. 北京:清华大学出版社, 2005.
- [3] 谭浩强. C语言程序设计[M]. 第4版. 北京:清华大学出版社, 2010.
- [4] 刘卫国,施荣华. 计算机程序设计教学中的抽象思维能力培养[J]. 计算机教育, 2012(3): 27-30.
- [5] 孙垂柳,田地. 基于程序设计思想的C语言教学方案[J]. 吉林大学学报, 2005, 23(8): 5-6.
- [6] 吴文虎. 程序设计中的切入点很重要[J]. 计算机教育, 2005(2): 12-15.
- [7] 陈元琰. 程序设计课程中/循环结构一节的教学方法[J]. 高教论坛, 2004(1): 76-80.
- [8] 范华,秦茂玲,张俊. 透视C程序设计语言[J]. 计算机技术与发展, 2008, 18(6): 64-66.
- [9] 汤亚玲. 递归算法设计及其非递归化研究[J]. 计算机技术与发展, 2009, 19(11): 85-89.
- [10] 倪瑞晓. C语言编程技术的分析研究[J]. 计算机技术与发展, 2009, 19(12): 251-254.
- [11] 陈莲君,朱晴婷. 培养能力为主线的C语言程序设计[J]. 计算机教育, 2011(14): 102-106.
- [12] Song J S, Hahn S H. An intelligent tutoring system for introductory C language course[J]. Computers & Education, 1997, 28(2): 93-102.

程序设计语言的循环结构跟踪分析法

作者: [周健](#), [夏日](#), [孙丽艳](#)
作者单位: [安徽财经大学 管理科学与工程学院, 安徽 蚌埠 233041](#)
刊名: [计算机技术与发展](#)
英文刊名: [Computer Technology and Development](#)
年, 卷(期): 2013(3)

本文链接: http://d.g.wanfangdata.com.cn/Periodical_wjfz201303050.aspx