DOI:10.13657/j.cnki.gxkxyxb.1987.02.004

PASCAL语言中的GOTO语句

罗海鹏 匡小苏

（广西计算中心）

摘 要

本文介绍了关于GOTO语句争论的历史背景，给出了在PASCAL中取消GOTO语句的非形式化的证明。最后用一个取消GOTO语句的具体例子进行说明。

引 言

PASCAL程序设计语言是70年代初产生的，它的规则简单明了，适用于各种领域，特别适用于教学和编写系统程序，它首次体现了结构程序设计的思想，是程序语言发展史上的一个里程碑。我国计算机软件专家仲萃豪说，公认PASCAL是当前最好的语言，目前还没有一种语言能达到PASCAL语言的声望。欧美国家越来越多的大学选用PASCAL语言作为教学的第一计算机语言，教育部在1983年也规定了把PASCAL作为我国软件专业本科生学习的第一程序设计语言。

最近，我们参加了电视大学（南宁市）计算机专业PASCAL课程的教学活动，参考了几种PASCAL语言程序设计的教材。在教学实践中，我们感到在这些教材里，GOTO语句应如何介绍还值得商榷。下面就谈谈我们对GOTO语句的看法并举出一个在PASCAL中取消GOTO的实例。

§1.关于GOTO语句

大多数教材在GOGO语句部分未介绍有关的背景材料，不能使学生对GOTO语句有全面的正确的认识。而且，该语句在许多教材中的出现也太早，这样，很多学生在学过GOTO语句后，会不分场合地滥用，影响良好的程序设计风格的培养。我们认为，介绍GOTO语句的有关背景材料不仅是有益的，而且是必要的，因为它本身就是一个很好的程序设计的教学内容。

1968年ACM通讯发表了Dijkstra的著名短文“GOTO statement considered harmful”，指出GOTO语句太原始，是造成程序混乱不堪的祸根。他主张从所有高级程序设计语言中取消GOTO语句。1972年ACM的一次国际会议还专门讨论了GOTO语句问题。这样，围绕着GOTO语句展开了一场大辩论，要求消灭GOTO语句的呼声越来越高。

希望消除GOTO语句的主要理由如下：

本文于1987年9月3日收到

（1）程序的执行是动态的过程，若程序中不加限制地使用GOTO语句，会造成程序的静态结构和动态执行情况差异甚大。

（2）不加限制地使用GOTO语句，破坏了程序基本结构单入口单出口的原则，这将使程序的正确性证明复杂化。

（3）无GOTO语句的程序容易形成模块化结构，可读性好，容易理解。

（4）无GOTO语句的程序查错容易，进行修改时所引起的副作用也比较小。

（5）无GOTO语句的程序，可使编译程序的全局优化算法简化。

主张保留GOTO语句的理由主要是，使用GOTO语句将使某些程序执行时有更高的效率等。

还有一些人提出用若干类似功能的其他语句来代替GOTO，结果出现了一些令人难以掌握的不自然的控制语句，或者是变相的GOTO语句。

1976年美国计算机权威D.E.Knuth发表文章“Structured Programming With GOTO Statement”，这篇文章平息了关于GOTO语句的争论。在这篇文章里，Knuth主张在语言的控制成份中仍保留GOTO，在功能方面不加限制，但（1）限制其使用的范围，例如，不能在某一模块中使用GOTO语句转到另一模块；（2）对一般的用户不开放，即只有编写系统软件的人为了提高运行效率时才可以使用。

从理论上讲，GOTO语句是没有存在的必要的。数理逻辑中的能行性理论，递归函数论等都没有引入类似GOTO的成份，因此，GOTO不应是算法语言中的基本成份。一种关于GOTO可以从PASCAL语言中去掉的非形式化的证明如下：

分三种情形来讨论：

一、当程序中有一个向下的GOTO时

如图1，虚线所示为一种可能的程序流向，这时整个程序P从结构上可表示为

P = P1 + GOTO S + P2 + P3。

设N1为P1中最后一个必经结点，即对任何的必经点N≠N1，N∈P1，都有N1在N的后面。也就是说，总是先执行N然后才到N1。

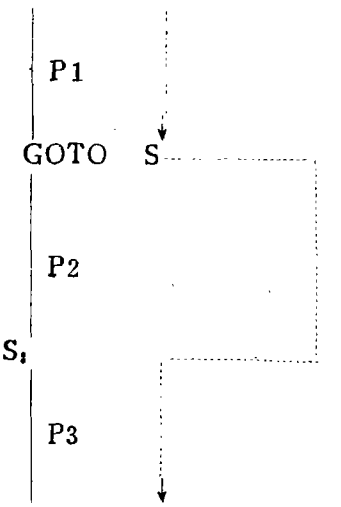

图1

情况1：当N1为非分枝语句时，GOTO S为必经结点，因此，P2永不可能执行。原程序等价于

P = P1 + P3，

即可取消GOTO语句。

情况2：当N1为分枝语句前面的一部分时，不妨设N1 = IF e THEN P1'，其中e为分枝条件，P1'和GOTO S组成复合语句，即当e为真时，执行P1' + GOTO S。我们设N1之前的程序段为P11，即P11 = P1 − N1。这时原程序可改写为

P = P11

+ IF e THEN P1' ELSE P2 + P3，

这样，也取消了GOTO语句。

二、当程序中有一个向上的GOTO时

如图2，虚线所示为一种可能的程序流向。同样假设N1为GOTOS之前的最后一个必经结点。

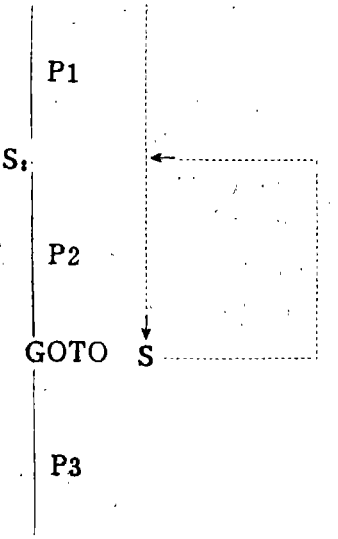

图2

情况1：N1为非分枝语句，这时程序无限循环，不可能正常结束。

情况2：N1为分枝语句。设当条件e为真时执行P2′+GOTO S，设P21=P2-N1，即

P2=P21+IF e THEN P2′+GOTOS。

这时整个程序可改写为

P=P1+P21
+WHILE e DO P2′+P21
+P3,

这样，也取消了GOTO语句。

三、当程序中有若干个GOTO语句时

情况1：若它们相互不交叉，则每一个GOTO语句都按照上述方式改写；

情况2：若它们有交叉，则我们可以把它们分层次分别处理，还是归结到上述的几种情况。

综上所述，在PASCAL程序设计中GOTO语句是可以被取消的。

§2.一个取消GOTO的例子

求2^{100}的精确值。

因为ln2=0.3010，因此2^{100}是一个31位数，在微型计算机上不采用特殊的方法是不能求出精确值的。可使用如下的方法：用数组元素来分段装乘幂的中间结果，每一个数组元素装三个数字，当这整个数乘以2时，我们让数组的每一个元素均乘以2，即得到了这个乘积。我们拿一个具体的数值作例子看看这个计算过程。假定已求出2^{14}=16,384，则A(1)=384，A(2)=16；2^{15}这样求：A(1)=A(1)*2=384*2=768，A(2)=A(2)*2=16*2=32，则2^{15}=32,768；2^{16}这样求：A(1)=A(1)*2=1536，减去1000，A(1)=536，A(2)=A(2)*2=64，加上1000，即加上1，A(2)=65，则2^{16}=65，536；……；当A(2)超过三位数时，则从A(2)减去1000(即1000000)，增加A(3)=1。变量说明：

A：数组，用来分段装2的幂；

N：乘以2的次数；

J：数组元素的下标。

程序步骤：

1)定义数组A，A(1)赋初值1；

2)让N从1到100循环，做

2.1)让J从1到11循环，做

2.1.1)判断A(J)是否等于0，

2.1.1.1）如果A（J）≠0，则做

2.1.1.1.1）A（J）=A（J）*2；

2.1.1.1.2）如果A（J-1）≥10000，则做A（J-1）=A（J-1）-1000，A(J)=A（J）+1；

2.1.1.1.3）继续J的下一个值；

2.1.1.2）如果A（J）=0，则做

2.1.1.2.1）如果A（J-1）≥1000，则做A（J-1）=A（J-1）-1000，A（J）=A（J）+1；

2.1.1.2.2）继续N的下一个值。

3）下标从大到小地输出数组A的值，这就是2^{100}的精确值。

根据这个算法，当用BASIC语言来编写程序时，对N和J的控制可以用循环语句完成，但还要想办法对J进行进一步地控制，即避免对A的数组元素为零时的乘法运算，而数组元素A（J）何时为零是不易事先估计的。因此，当A（J）=0时，只好用GOTO语句跳出。

BASIC程序清单：

```
10  DIM A(11)（2^100是31位数，占用11个数组元素）
20  A(1)=1
30  FOR N=1 TO 100（乘100次2）
（40行～80行，给每一个非零的A（J）乘以2）
40  FOR J=1 TO 11
50  IF A(J)=0 THEN 90
60  A(J)=A(J)*2
70  IF A(J-1)>=1000 THEN A(J-1)=A(J-1)-1000:A(J)=A(J)
    +1（进位）
80  NEXT J
90  IF A(J-1)>=1000 THEN A(J-1)=A(J-1)-1000:A(J)=
    A(J)+1（A（J）为零时也要考虑A（J-1）是否进位）
100 NEXT N
（110行～150行，输出结果）
110 FOR J=11 TO 1 STEP-1
120 IF A(J)<100 THEN PRINT 0;
130 IF A(J)<10 THEN PRINT 0;
140 PRINT A(J),
150 NEXT J
160 END
```

在APPLE Ⅱ微机上的运行结果：

```
RUN
0012676506002282294014967032O5376
```

在上面的程序中，50行使用了一个GOTO，用以跳出对J的循环。

如果这个程序改用PASCAL语言编写，例如用WHILE a[j]<　>0 DO循环控制数组的下标变量j，很容易地就可避开GOTO语句。

PASCAL程序清单：

```
PROGRAM poweroftwo(output);
VAR a:ARRAY[0..11] OF integer; i, j:integer;
BEGIN
  a[1]:=1;
  FOR i:=1 TO 100 DO
    BEGIN
      j:=1;
      WHILE a[j]<>0 DO
        BEGIN
          a[j]:=a[j]*2;
          IF a[j-1]>=1000 THEN
            BEGIN
              a[j-1]:=a[j-1]-1000;
              a[j]:=a[j]+1
            END;
          j:=j+1
        END;
      IF a[j-1]>=1000 THEN
        BEGIN
          a[j-1]:=a[j-1]-1000;
          a[j]:=a[j]+1
        END
    END;
  FOR j:=11 DOWNTO 1 DO
    BEGIN
      IF a[j]<100 THEN write(0:1);
      IF a[j]<10 THEN write(0:1);
      write(a[j]:1)
    END
END.
```

在APPLE Ⅱ微机上通过了这个程序，运行结果同上。

参 考 文 献

〔1〕李启炎、宋秋杰：《PASCAL程序设计语言》，同济大学出版社，1985年。

〔2〕仲萃豪、冯玉琳："程序设计方法学"，《计算机研究与发展》，第20卷第3期。

〔3〕Dijkstra, E. W.: "A Disciplice of Programming", 1976.

〔4〕David Gries: "The Science of Programming", 1982。

〔5〕Wirth, N.: "Algorithms+Data Structures=Programs", 1976.

GOTO STATEMENT IN PASCAL

Luo Haipeng Kuang Zhihui

(*Guangxi Computation Centre*)

ABSTRACT

In this paper, the writer introduces the background of the argument in GOTO statement and informally proves that GOTO statement can be removed in PASCAL computer language. To the end of this paper, the writer gives an example of removing "GOTO" statement in programming.