自由条件组合及其应用

刘　扬　（抚顺石油学院计算中心）

一、引言

在程序设计过程中,经常要使用条件语句,尤其在对数据库文件或库文件的操作过程中更是如此。因此,如何正确有效地完成条件的自由组合并实施有条件操作,是程序设计中值得探讨的一个问题。本文结合应用软件开发的实际情况,探讨自由条件组合的方式和应用。

二、自由条件组合的方式

从条件语句的存在方式上,可将其划分为两类,即固定条件组合和自由条件组合。

固定条件组合是将由单一条件所组成的逻辑表达式固定到程序中来实现的,这种条件组合在程序执行过程中是不变的;自由条件组合也是由单一条件所组成的逻辑表达式,但它不固定到程序之中。自由条件组合是在程序设计的基础上允许程序的使用者自由进行条件的组合,只要条件逻辑表达式符合语法规则就可以正确执行。下面通过具体实例详细讨论有关自由条件组合的一些问题。

已知的事故基本信息库 SGXX.DBF 结构如下:

字段名(含义)	类型	长度	小数位数
SO1(事故编号)	C	11	—
SO2(企业名称)	C	12	—
SO3(发生时间)	D	8	—
SO4(事故原因)	C	4	—
SO5(经济损失)	N	8	2
SO6(死亡人数)	N	3	0
SO7(重要人数)	N	3	0
SO8(简要说明)	C	40	—

通过关系运算符将字段名与具体数据联接到一起即构成单一条件表达式。关系运算符共有六种:“=”、“#”、“>”、“>=”、“<”和“<=”。例如,“S07>5”即为单一条件表达式。

条件组合是逻辑运算符联接单一条件表达式的过程。逻辑运算符有“与”、“或”和“非”三种,而在实际的条件组合应用过程中通常只用前两种,这样做的原因有两条:第一、含有“非”运算符的逻辑表达式可以转化为不含“非”运算符的逻辑表达式,而且这种表达式更符合我们的思维习惯;第二、转化以后的逻辑表达式较转化以前更为简单。

例如,由反演律可知,表达式

.NOT.(S03='0001'.AND.S04<>.OR.S07>5)

等价于

S03#'0001'.OR.S04='1000'.AND.S07<=5

"与"和"或" 运算的优先级是相等的,但是,我们可以通过括号来改变它们的运算顺序.从下述内容中可以看到,正是由于它可以改变"与""或"运算的顺序,所以又赋予它更加丰富的内涵.对于单一条件线性"与"组合(如图 1 所示)和线性"或"组合(如图 2 所示)来说,运算符合结合律,所以在表达式中不使用括号;对于单一条件的线性"与""或"组合(如图 3 所示)来说,表达式不使用括号;而对于单一条件非线性"与""或"组合(如图 4 所示)来说,表达式中必须使用括号.

例如,表达式(S03='001'.AND. S04='100') .OR. (S03='220'. AND. S04='200') 之中的括号无法省略,否则,含义就改变了.至此,可以确定条件组合的四个基本元素:

.单一条件表达式

.逻辑运算符"与"

.逻辑运算符"或"

.括号

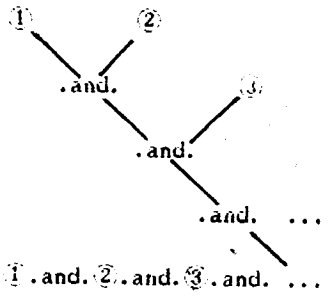

图 1 线性"与"组合

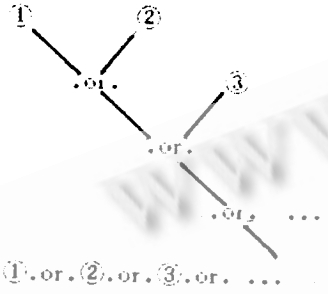

图 2 张性"或"组合

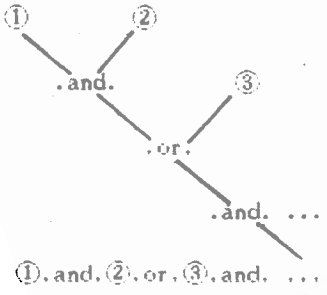

图 3 线性"与""或"组合

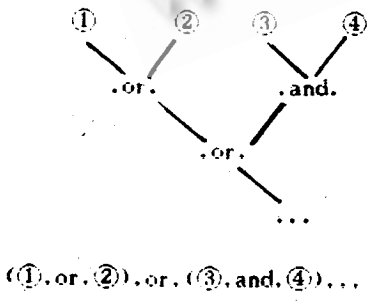

图 4 非线性"与""或"组合

自由条件组合是在程序设计的基础上实现的,它允许程序的使用者自由进行条件组合,也就是说,允许用户用所操作数据库(或文件)的任何字段构成单一条件,每个字段出现的次数不限,并且通过"与"/"或"运算符和括号构成所期望的条件表达式.

三、具体实现及其应用

对于任意给定的数据库文件,首先使用 COPY TO <数据结构文件名> STUCTURE EXTENDEN 命令生成数据库文件怕对应的结构库文件;再使用 MODIFY STRUCTURE 命令为结构库增加一个用以存放数据文件字段含义的字段名 FIELD MN(该字段的内容用作形成单一条件的提示信息),对于前文述及的事故基本信息库文件 SGXX.DBF 来说,修改后的结构库文件 *SGXX JG.DBF)为:

FIELE NAME	FIELD TYPE	FIELD LEN	FIELD DEC	FIELD MN
S01	C	11	–	事故编号
S02	C	12	–	企业名称
S03	D	8	–	发生时间
S04	C	4	–	事故原因
S05	N	8	2	经济损失
S06	N	8	0	死亡人数
S07	N	3	0	重伤人数
S08	C	40	–	简要说明

进行自由条件组合的程序包括两部分:一个用于条件组装(ZYTJZH.PRG(原程序附后));一个用于字段显示与选择(XSSJJG.PRG)。

调用程序 ZYTJZH.PRG 的命令格式为:

```
...
public tj1,sjjg    && tj1为ZYTJZH.RPG组装的条
tj1=''             &&件表达式;sjjg为数据文件
sjjg='SGXX JG' &&所对应的结构库文件名
do zytjzh with tj1,sjjg
...
```

由 ZYTJZH.PRG 组装的条件表达式有可能不符合语法规则,所以在执行条件语句时最好有一个相应的错误处理程序。

调用程序 XSSJJG.PRG 的命令格式为:

do xssjjg with adm,hzzdm,zdlx,zdcd,zdxsw,sjjg

其中,ZDM、HZZDM、ZDLX、XDCD 和 ZDXSW 是由结构库文件带回的相应字段的字段名、字段含义,字段类型、字段长度和字段小数位等。

附 ZYTJZH.PRG 原程序清单:

```
* ZYTJZH.PRG
para tj1,sjjg
set color to b+/w
  24,18 say '→←移动 ENTER-选择 ESC-退出'
row1=2
col1=20
do while .t.
 store "to zhm,zdix,hzzdm,dmhz,gsc,in
 store 0 to zdcd,zdxsw,ch1,ch2,ch3,ch4
 set color to w+/b
    row1,col1-4 clear to row1+2,col1+43
    row1+1,col1-2 prompt '1-'
    row1+1,col1+7 prompt '2-并且'
    row1+1,col1+17 prompt '3-或者'
    row1+1,col1-27 prompt '4-('
    row1+1,col1-36 prompt '5-)'
 menu to ch2
 do case
  case ch2=0
   exit case ch2=1
 case ch2=2
  tj1=tj1+'.and.'
  loop
 case ch2=3
  tj1=tj1"'.or.'
  loop
 case ch2=4
  tj1=tj1+'('
  loop
 8case ch2=5
  tj1=tj1+'('
  loop
 endcase
 set colot to w+/b
    12-2,8 clear to 12-2+12,10+17
 do xssjjg with zdm,hzzdm,zdlx,zdcd,zdxw,sjjg
 set color to w+/b
    row1+6,20 say hzzdm
 set color to w+/b
    roow1+6,col1+20 prompt '='
    roow1+6,col1+20 prompt '<>'
    roow1+6,col1+20 prompt '>'
    roow1+6,col1+20 prompt '>='
    roow1+6,col1+20 prompt '<'
    roow1+6,col1+20 prompt '<='
 menu to ch1
 do case
  case ch1=0
   tj1="
   loop
  case ch1=1
   r-s='=' &&r-s 为关系运算符变量
  case ch1=2
   r-s='<>'
  case ch1=3
        r-s='>'
  case ch1=4
   r-s='>='
      case ch1=5
  r-s='<'
   case ch1=6
  r-s='<='
  endcase
   tj1=tj1+trim(zdm)+r-s
   if trim(zdix)='D'
   in=date()
   set color to w+/b
      row1+6,40 say r-s get in
   read
   in=trim(dtoc(in))
  else
   if trim(zdlx)='N'
   in=0
   if zdxsw=0
    gsc=repl('9',zdcd) &&gsc 为格式场变量,它由
    else                    &&zdlx,zdcd 和 zdxsw 得来
    gsc=repl('9',zdcd-zdxsw-1)+'.'+;
      repl('9',zdxsw)
   endif
   set color to w+/b
      row1+6,40 say r-s get in pict "&gsc"
   read
   in=ltrim(trim(str(in)))
  else
   if trim(zdlx)='C'
    in=space(30)
    gsc=repl('N',zdcd)
    set color to w+/b
       row1+6,40 say r-s get in pict "&gsc"
    read in='"'+trim(in)+'"'
    endif
   endif
  endif
 tj1=tj1+trim(in)
 loop
 enddo
return
```