例题 7.1.1 将下列语句翻译为逆波兰表示（后缀式），三元式和间接三元式序列和四元式表示：

a:=(b+c)*e+(b+c)/f

解题思路:

把中缀式转换后缀式的简单方法: 按中缀式中各运算符的优先规则, 从最先执行的部分开始写，一层层套。 如对 a≤b+c∧a>d∨a+b≠e ,先把 b+c 写为 bc+，然后把 a≤套上去，成为 abc+≤; 再把 a>d 表示为 ad>, 然后把∧套上去, 成为 abc+≤ad>∧, 依此类推。

四元式的由 4 个部分组成：算符 op、第一和第二运算量 arg1 和 arg2，以及运算结果 result。运算量和运算结果有时指用户自定义的变量，有时指编译程序引进的临时变量。如果 op 是一个算术或逻辑算符，则 result 总是一个新引进的临时变量，用于存放运算结果。

三元式只需三个域：op、arg1 和 arg2。与四元式相比, 三元式避免了临时变量的填入, 而是通过计算这个临时变量值的语句的位置来引用这个临时变量。我们很容易把一个算术表达式或一个赋值句表示为四元式序列或三元式序列。

间接三元式是指用一张间接码表辅以三元式表的办法来表示中间代码。间接码表按运算的先后顺序列出有关三元式在三元表中的位置, 对于相同的三元式无需重复出现在三元表中。

解答

逆波兰表示为：bc+e*bc+f/+:=

三元式序列为:

(1) (+, b, c)
(2) (*,(1),e)
(3) (+, b, c)
(4) (/,(3),f)
(5) (+,(2),(4))
(6) (:=,a,(5))

间接三元式表示为：

三元式表	间接码表
(1) (+, b, c)	(1)
(2) (*,□, e)	(2)
(3) (/,□, f)	(1)
(4) (+,□,□)	(3)
(5) (:=,a,□)	(4)

四元式表示为:

(1) (+, b, c, T1)
(2) (*, T1,e, T2)
(3) (+, b, c, T3)
(4) (/, T3, f, T4)
(5) (+, T2,T4, T5)
(6) (:=,T5, -, a)

例题 7.1.2 利用回填技术把语句

while a>0 or b>0 do

if c>0 and d<0 then x:=y+1;

翻译为三地址代码。

解题思路:

把表达式或赋值语句翻译为三地址代码是容易理解的, 如 x:=y*z+1 翻译为:

T1:=y*z

T2:=T1+1

x:=T2

while 语句和 if 语句的翻译涉及到布尔表达式, 我们一并讨论。 产生布尔表达式三地址代码的语义规则如表 6.3 所示。 按表 6.3 的定义, 每个形如 A relop B 的表达式(其中 relop 为任一关系运算符)将翻译为如下两条转移指令:

if A relop B goto …

goto …

因此, 假定表达式的待确定的真假出口已分别为 Ltrue 和 Lfalse, 则 a>0 or b>0 将被翻译为

if a>o goto Ltrue

goto L1

L1: if b>0 goto Ltrue

goto Lfalse

而 c>0 and d<0 将被翻译为

```
    if  c>0  goto  L3
      goto  Lfalse
L3:  if  d<0  goto  Ltrue
      goto  Lfalse
```

有关 if 和 while 语句的属性文法如表 6.4 所示。

应用表 6.3 和表 6.4 不难生成含 if 和 while 的语句的三地址代码

解答:

所求三地址代码为:

```
L0:  if  a>o  goto  L2
      goto  L1
L1:  if  b>0  goto L2
      goto  Lnext
L2:  if  c>0  goto  L3
      goto  L0
L3:  if  d<0  goto  L4
      goto  L0
L4:  T1:=y + 1
      x:= T1
      goto  L0
Lnext:
```

例题 7. 1.3 把语句

```
while x>y do
```

if x>0 then x:=x-1

else y:=y+1;

翻译为四元式序列。

解题思路:

因为三地址语句可看成中间代码的一种抽象形式, 而四元式是三地址代码语句的具体实现。因此, 上题介绍的语义规则及翻译方法可用于产生四元式。

我们也可通过适合语法制导翻译的语义子程序(或称翻译模式)来理解和翻译四元式。

If 语句和 While 语句的翻译模式如图 6.8 所示。根据此翻译模式, 可以把含 if 和 while 的语句翻译为四元式序列。

解答:

(1) (j>, x, y, 3)

(2) (j, -, -, 11)

(3) (j>, x, 0, 5)

(4) (j, -, -, 8)

(5) (-, x, 1, T1)

(6) (:=,T1,-, x)

(7) (j, -, -, 1)

(8) (+, y, 1, T2)

(9) (:=,T2,-, y)

(10) (j, -, -, 1)