§4.7 控制语句(flow-of-control statements)的翻译

1. 标号(label)和转移语句(goto statement)

标号定义的形式:

L : S

\*当这种语句被处理之后,标号L称为“定义了”.在符号表中,标号L的“地址”栏将登记上语句S的第一个四元式的地址(编号).

如果 goto L; 是向后转移的处理:

如果 goto L; 是向前转移的处理:

(见黑板)

1.建链

2.符号表建立入口.

翻译到 S→label S

label→i:

当用label→i:进行归约时,应做如下的语义动作:

(1)若i代表的标识符(假定为L)不在符号表中, 则把它填入,置“类型”为“标号”,“定义否”为“已”,“地址”为NXQ.

(2)若L已在符号表中, 但“类型”不是“标号”或“定义否”为“已”,则报告出错.

(3)若L已在符号表中,则把标志“未”改为“已”,然后把地址栏中的链头(记为q)取出,同时把NXQ填入其中, 最后执行BACKPATCH(q, NXQ).

1. 条件语句(控制语句)

(1) S→if E then S

(2) | if E then S else S

(3) | while E do S

(4) | begin L end

(5) | A

(6) L→L; S

(7) | S

1. 条件语句的代码结构

1. if语句: if E then S1 和 if E then S1 else S2

代码结构(见书 P258, P401)

2. while语句: while E do S1

代码结构(见书 P258, P401)

3.语句嵌套

设立语义变量S.CHAIN, L.CHAIN

例如: if E1 then if E2 then S1 else S2 else S3

代码结构(见黑板)

1. 条件语句的语义动作

S→CS(1)

{S.CHAIN := MERG(C.CHAIN, S(1).CHAIN) }

S→TPS(2)

{S.CHAIN := MERG(TP.CHAIN, S(2).CHAIN) }

S→WdS(1)

{BACKPATCH(S(1).CHAIN, Wd.QUAD);

GEN(j, ―,―,Wd.QUAD);

S.CHAIN := Wd.CHAIN }

S→begin L end

{S.CHAIN := L.CHAIN }

S→A

{S.CHAIN := 0 /\*空链\*/ }

L→LSS(1)

{L.CHAIN := S(1).CHAIN }

L→S

{ L.CHAIN := S.CHAIN }

C→if E then

{BACKPATCH(E.TC, NXQ);

C.CHAIN := E.FC }

TP→C S(1) else

{q := NXQ;

GEN(j, ―,―,0);

BACKPATCH(C.CHAIN, NXQ);

TP.CHAIN := MERG(S(1).CHAIN, q) }

W→while

{W.QUAD := NXQ }

Wd→W E do

{BACKPATCH(E.TC, NXQ);

Wd.CHAIN := E.FC;

Wd.QUAD := W.QUAD }

LS→L;

{BACKPATCH(L.CHAIN, NXQ) }

1. 例子

while A < B do

if C < D then X := Y + Z

的四元式代码如下:

(1) (j<, A, B, 3)

(2) (j, ―, ―, 0)←S.CHAIN

(3) (j<, C, D, 5)

(4) (j, ―, ―, 1)

(5) (+, Y, Z, T1)

(6) (:=, T1, ―, X)

(7) (j, ―, ―, 1)

1. 循环语句

1.语法

S→for i := E(1) to E(2) do S(1)

2. PASCAL的解释

i := E(1);

LIMIT := E(2);

goto OVER;

AGAIN: i := i + 1;

OVER: if i ≤ LIMIT then

begin S(1); goto AGAIN end;

1. 语义动作

F1→for i := E(1)

{GEN(:=, E(1).PLACE, ―, ENTRY(i));

F1.PLACE := ENTRY(i) }

F2→F1 to E(2)

{LIMIT := NEWTEMP;

GEN(:=, E(2).PLACE, ―, LIMIT);

OVER := NXQ+2;

GEN(j, ―, ―, OVER);

F2.QUAD := NXQ;

GEN(+, F1.PLACE, 1, F1.PLACE);

q := NXQ;

GEN(j≤, F1.PLACE, LIMIT, q+2);

F2.CHAIN := NXQ;

GEN(j, ―,―,0) }

S→F2 do S(1)

{BACKPATCH(S(1).CHAIN, F2.QUAD);

GEN(j, ―,―,F2.QUAD);

S.CHAIN := F2.CHAIN }

1. 例子:

for I := 1 to N do M := M + I

的四元式代码如下:

(1) (:=, 1, ―, I)

(2) (:=, N, ―, LIMIT)

(3) (j, ―, ―, 5)

(4) (+, I, 1, I)

(5) (j≤, I, LIMIT, 7)

(6) (j, ―, ―, 0)

(7) (+, M, I, T1)

(8) (:=, T1, ―, M)

(9) (j, ―, ―, 4)

七. 分叉语句(switch or case statement)

1. 语法:

Case E of

C1 : S1;

C2 : S2;

……

Cn-1: Sn-1;

otherwise: Sn

end

1. 中间代码结构

计算E的值放入T中的代码

Goto TEST

L1: 关于S1的中间代码

Goto NEXT

L2: 关于S2的中间代码

Goto NEXT

……

Ln-1: 关于Sn-1的中间代码

Goto NEXT

Ln: 关于Sn的中间代码

Goto NEXT

TEST: IF T = C1 Goto L1

IF T = C2 Goto L2

……

IF T = Cn-1 Goto Ln-1

Goto Ln

NEXT:

阅读: 讲义和教材 6.6.1, 6.6.2, 6.6.3, 6.7.3, 6.7.4, 6.8 节

作业11:

1. 用本课程所讲的方法,将下面的语句翻译成四元式序列:

while A < C ∧ B < D do

if A = 1 then C := C + 1 else

while A ≤ D do A := A + 2 .