

第八章 中间代码生成

1. 设有表达式 $A*(B*C-A) \leq B+C*D$

- (1) 写出逆波兰式(后缀式)中间代码。
- (2) 写出三元式中间代码。
- (3) 写出多元式中间代码。
- (4) 画出树。

(答案)

i $ABC*A-*BCD*+ \leq$

ii (1) $(*BC)$

(2) $(- (1) A)$

(3) $(* A (2))$

(4) $(* C D)$

(5) $(+ B (4))$

(6) $(\leq (3) (5))$

iii $(* B C t1)$

$(- t1 A t2)$

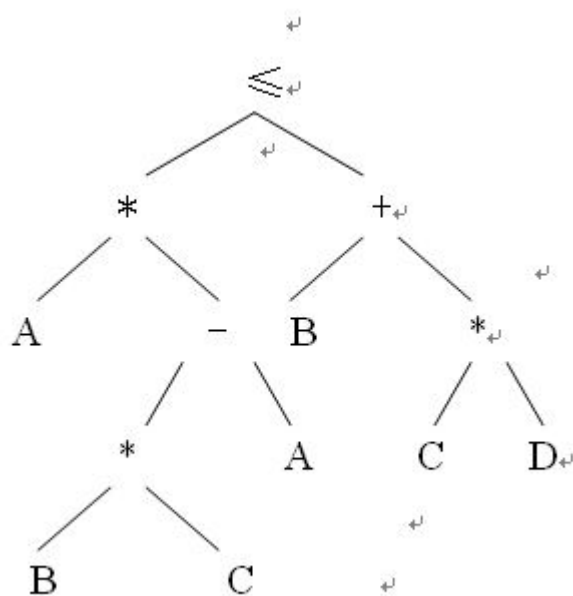
$(* A t2 t3)$

$(* C D t4)$

$(+ B t4 t5)$

$(\leq t3 t5 t6)$

iv



(关闭)

2. 试写出下列语句的四元式中间代码:

- (1) if $x > 0$ then $x := 0$ else $x := 1$
- (2) while $x > 0$ do $x := x - 1$
- (3) if $x > 0$ then if $x < 0$ then $x := x + 1$ else $x := 1$ else $x := 1$
- (4) while $x > 0$ do while $y > 0$ DO begin $y := y - x$; $x := x - 1$ end

(答案)

- | | |
|----------------------|--------------------|
| (1) 1)(GT, x, 0, t1) | (2) 1)(LABEL, L1) |
| 2)(JUMP0, t1, L1) | 2)(GT, x, 0, t1) |
| 3)(ASSIG, 0, x) | 3)(JUMP0, t1, L2) |
| 4)(JUMP, L2) | 4)(SUBI, x, 1, t2) |
| 5)(LABEL, L1) | 5)(ASSIG, t2, x) |
| 6)(ASSIG, 1, x) | 6)(JUMP, L1) |
| 7)(LABEL, L2) | 7)(LABEL, L2) |
| (3) 1)(GT, x, 0, t1) | (4) 1)(LABEL, L1) |
| 2)(JUMP0, t1, L2) | 2)(GT, x, 0, t1) |
| 3)(LT, x, 0, t2) | 3)(JUMP0, t1, L3) |
| 4)(JUMP0, t2, L1) | 4)(LABEL, L2) |
| 5)(ADDI, x, 1, t3) | 5)(GT, y, 0, t2) |
| 6)(ASSIG, t3, x) | 6)(JUMP0, t2, L1) |
| 7)(JUMP, L3) | 7)(SUBI, y, x, t3) |
| 8)(LABEL, L1) | 8)(ASSIG, t3, y) |
| 9)(ASSIG, 1, x) | 9)(SUBI, x, 1, t4) |
| 10)(JUMP, L3) | 10)(ASSIG, t4, x) |
| 11)(LABEL, L2) | 11)(JUMP, L2) |
| 12)(ASSIG, 1, x) | 12)(LABEL, L3) |
| 13)(LABEL, L3) | |

(关闭)

3. 试写出下列赋值语句的四元式中间代码, 其中个数组的下界为 1, 上界为 10。

- (1) $A[i+1] := A[i-1] + A[i]$
- (2) $R.f := R.x + A[i]$
- (3) $B[i].x := R1.D[j].y$
- (4) $A[A[i]] := R2.R.f$

(5) $D[i+1][j] := D[i+1][j]+1$

(答案)

此处的 Size 是 A 的基元的大小

①(i+, i, 1, T1)

(i-, T1, 1, T2)

(i*, T2, Size, T3)

([], A, T3, T4)

(i-, i, 1, T5)

(i-, T5, 1, T6)

(i*, T6, Size, T7)

([], A, T7, T8)

(i-, i, 1, T9)

(i*, T9, Size, T10)

([], A, T10, T11)

(+, T8, T11, T12)

(=, T12, -, T4)

②(? , R, Offset(f), T1)

(? , R, Offset(x), T2)

(i-, i, 1, T3)

(i*, T3, Size, T4)

(+, T2, T4, T5)

(=, T5, -, T1)

③(i-, i, 1, T1)

(i*, T1, Size, T2)

([], B, T2, T3,)

(? , T3, Offset(x), T4)

(? , R1, Offset(D), T5)

(i-, j, 1, T6)

(i*, T6, Size, T7)

([], T5, T7, T8)

(? , T8, Offset(y), T9)

(=, T9, -, T4)

④(i-, i, 1, T1)

(i*, T1, Size, T2)

([], A, T2, T3,)

(i-, T3, 1, T4)

(i*, T4, Size, T5)

([], A, T5, T6,)

(? , R2, Offset(R), T7)

(? , T7, Offset(f), T8)

(=, T8, -, T6)

⑤(i+, i, 1, T1)

(i-, T1, 1, T2)

(i*, T2, Size*10, T3)

([], D, T3, T4,)

(i-, j, 1, T5)

(i*, T5, Size, T6)

([], T4, T6, T7)

(i+, i, 1, T8)

(i-, T8, 1, T9)

(i*, T9, Size*10, T10)

([], D, T10, T11)

(i-, j, 1, T12)

(i*, T12, Size, T13)

([], T11, T13, T14)

(i+, T14, 1, T15)

(=, T15, -, T7)

(关闭)

4. 写出下列过程调用语句的多元式代码，其中 f、g 和 h 的形参均为赋值形参变量。

(1) $f(x*(x+1), i+1)$

(2) $f(g(x-1), x)$

(3) $f(g(h(x)), g(x))$

(答案)

- | | |
|---------------------|----------------------|
| (1) 1)(ADDI,x,1,t1) | (2) 1) (SUBI,x,1,t1) |
| 2)(MULI,t1,x,t2) | 2)(VALACT,t1,β1,1) |
| 3)(ADDI,i,1,t3) | 3)(CALL,g,true,t2) |
| 4)(VALACT,t2,β1,1) | 4)(VALACT,t2,β1,1) |
| 5)(VALACT,t3,β2,1) | 5)(VALACT,x,β2,1) |
| 6)(CALL,f,true) | 6)(CALL,f,true) |
- (3) 1)(VALACT,x,β1,1)
- 2)(CALL,h,true,t1)
- 3)(VALACT,t1,β1,1)
- 4)(CALL,g,true,t2)
- 5)(VALACT,x,β1,1)
- 6)(CALL,g,true,t3)
- 7)(VALACT,t2,β1,1)
- 8)(VALACT,t3,β1,1)
- 9)(CALL,f,true)

(关闭)

5. 设有表达式 $A-A*B-D*(A-B*D)/A$ ，写出生成多元式的大致过程(用 LR 尾动作文法)。

(答案)

LR 尾动作文法

[1] $Z \rightarrow E$ //push(E.Val)

[2] $E \rightarrow T$ //push(T.Val)

[3] $E \rightarrow E-T$ #action(-)

[4] $T \rightarrow P$ //push(P.Val)

[5] $T \rightarrow T * P$ #action(*)

[6] $T \rightarrow T / P$ #action(/)

[7] $P \rightarrow \langle id \rangle$ //push(id.Val)

[8] $P \rightarrow (E)$ //push(E.Val)

#action(ω): { Quad[j] := (ω , Sem(top-1), Sem(top), T[k]);

Sem[top-1] := T[k];

j := j + 1; //j 第一个可用的四元式编号

k := k + 1; //k 第一个可用的临时变量编号

}

Sem 语义栈

#	$A - A * B - D * (A - B * D) / A$	
#A	$E - A * B - D * (A - B * D) / A$	
#AA	$E - T * B - D * (A - B * D) / A$	
#AAB	$E - T * P - D * (A - B * D) / A$	
#AT ₁	$E - T_1 - D * (A - B * D) / A$	(i*, A, B, ...)
#T ₂	$E - D * (A - B * D) / A$	(i-, A, T ₁ , ...)
#T ₂ D	$E - T * (A - B * D) / A$	
#T ₂ DA	$E - T * (E - B * D) / A$	
#T ₂ DAB	$E - T * (E - T * D) / A$	
#T ₂ DABD	$E - T * (E - T * P) / A$	
#T ₂ DAT ₃	$E - T * (E - T) / A$	(i*, B, D, ...)
#T ₂ DT ₄	$E - T * (E) / A$	(i-, A, T ₃ , ...)
#T ₂ DT ₄	$E - T * P / A$	
#T ₂ T ₅	$E - T / A$	(I*, D, T ₄ , ...)
#T ₂ T ₅ A	$E - T / P$	
#T ₂ T ₆	$E - T$	(i/, T ₅ , A, ...)
#T ₇	E	(i-, T ₂ , T ₆ , ...)
#T ₇	Z	

(关闭)

6. 对于下列语句写出多元式的大致生成过程(用 LR 尾动作文法):

- (1) if $x > 0$ then $x := 0$ else $x := 1$
- (2) while $x > 0$ do begin $x := x - 1$ end
- (3) $A[A[i]] := R1.R2.X$

(答案)

(无)

(关闭)

7. 给出在产生中间代码时进行常表达式节省的算法。

(答案)

先引入 VVL(Variable Value List)

引入操作 Replace(A): 若 A 是常量, 返回 A 的值; 若不是, A 直接返回。

VVL 表单元

Var Name Value

方法: i 清空 VVL 表。

- ii 设当前要产生的四元式为 (ω, A, B, T) , 进行下面处理, 变为 $(\omega, \text{Replace}(A), \text{Replace}(B), T)$, 但不真的产生。

iii 若 ω 为 +, -, *, / , ...等常量表达式节省类运算符则 goto4, 否则 goto6。

iv 若 ω 是赋值, 则 goto5, 否则

① 若 $\text{Replace}(A)$, $\text{Replace}(B)$ 均为常量, 则计算 $\text{Replace}(A)$, ω , $\text{Replace}(B)$ 的结果。并填入 VVL 中, 产生四元式, goto6。

② 若 $\text{Replace}(A)$, $\text{Replace}(B)$ 中有一个不为常量, 则产生四元式, goto6。

v 若 ω 为赋值, $(\omega, A, -, T)$

① 若 T 不是引用型变量, 若 A 是常量, 或 VVL 表中有 (A, V) 项, 则将

$(T, A.\text{Value})$ 填入 VVL 中, 且产生四元式; 若 A 不是常量, 且 T 存在于 VVL 中, 则删除 T。

② 若 T 是引用型变量, 则产生四元式, 且结束基本块。

vi 继续语法制导工作, 若未超出基本块, goto2; 若超出了则 goto1。

(关闭)

8. 给出在产生中间代码时同时进行常表达式和公共表达式节省的算法。

[\(答案\)](#)

(无)

[\(关闭\)](#)

9. 假设有表达式 $x+3+4+5$ ，或 $3+x+4+5$ ，若按从左到右的方法处理，则将会产生如下中间代码：

$(+,x,3,t1)(+,t1,4,t2)(+,t2,5,t3)$

试给出对类似上述形能进行常表达式优化的一种处理方法。它对上例将均产生中间代码：

$(+,3,4,t1)(+,t1,5,t2)(+,x,t2,t3)$

[\(答案\)](#)

设置一个队列 Q ，在中间代码产生过程中，若四元式 $Quad = (+, A, B, T)$ 中， A, B 中至少有一个不是常量，常量的项压入 Q 中；若 A, B 均为常量，则产生四元式；待输入全部使用后，从 Q 中弹出队头，再构造四元式。

[\(关闭\)](#)

10. 同 9 题，但要求产生中间代码 $(+,x,12,t1)$ 。

[\(答案\)](#)

(无)

[\(关闭\)](#)

