

作业 (0423)

● 5月7日上课前提交

● 1. 考虑右边的三地址代码:

- (1) 把这段代码序列划分为基本块
- (2) 为这段代码构造控制流图, 并额外添加入口结点 ENTRY 和出口结点 EXIT
- (3) 找出控制流图中的所有循环

```
(1)  s = 0
(2)  i = 0
(3)  t1 = s % 19
(4)  if t1 == 1 goto (19)
(5)  j = 0
(6)  if i < 50 goto (8)
(7)  return s
(8)  if j >= 100 goto (17)
(9)  if i == j goto (15)
(10) t2 = 100 * i
(11) t3 = t2 + j
(12) t4 = 4 * t3
(13) t5 = a[t4]
(14) s = s + t5
(15) j = j + 1
(16) goto (8)
(17) i = i + 1
(18) goto (3)
(19) return s
```

作业 (0423)

◎ 5月7日上课前提交

◎ 2. 按照课堂上给出的数组引用翻译的 SDT 给出下面语句 S 的三地址代码:

$$a[i*2+j][a[2][i+j][k]+i][i*4+k*2] = i*j+k*a[0][4][2*k];$$

其中数组 a 的类型是 $\text{int}[5][6][9]$ 。

作业 (0423)

◎ 5月7日上课前提交

◎ 3. 按照课堂上给出的控制流回填翻译的 SDT 给出下面语句 S 的三地址代码和 $S.nextlist$ (指令编号从 100 开始):

```
while ((a + b * c > x + y) && (m == n)) {  
    if (x <= y) {  
        while (a < b) { a = a + 10; b = c * m; }  
    } else a = b + c;  
}
```

作业 (0423)

◎ 5月7日上课前提交

◎ 4. 考虑一种基于栈的中间表示, 它的所有指令都隐式地操作一个全局的栈:

指令	作用
CONST(M)	把常数 M 压到栈顶
GET_VAR(x)	把程序变量 x 的值压到栈顶
SET_VAR(x)	弹栈一次, 把获取的值写入程序变量 x
ADD	弹栈两次, 依次获取值 b 和值 a , 计算 $a + b$ 的值并压到栈顶
SUB	弹栈两次, 依次获取值 b 和值 a , 计算 $a - b$ 的值并压到栈顶
MUL	弹栈两次, 依次获取值 b 和值 a , 计算 $a \times b$ 的值并压到栈顶
GOTO(L)	跳转到标号 L 的指令处
GOTO_IF_ZERO(L)	弹栈一次, 若获取的值为零则跳转到标号 L 的指令处

作业 (0423)

◎ 4. 以下是这种栈式中间表示的一些示例:

```
a = b - c * 2;  
d = a;
```

```
GET_VAR(b)  
GET_VAR(c)  
CONST(2)  
MUL  
SUB  
SET_VAR(a)  
GET_VAR(a)  
SET_VAR(d)
```

```
if (a == 0) b = b + 1;  
else c = c - 1;
```

```
GET_VAR(a)  
GOTO_IF_ZERO(L1)  
GOTO(L2)  
L1: GET_VAR(b)  
CONST(1)  
ADD  
SET_VAR(b)  
GOTO(L3)  
L2: GET_VAR(c)  
CONST(1)  
SUB  
SET_VAR(c)  
L3:
```

```
while (a == b)  
    c = c * c;
```

```
L1: GET_VAR(a)  
    GET_VAR(b)  
    SUB  
    GOTO_IF_ZERO(L2)  
    GOTO(L3)  
L2: GET_VAR(c)  
    GET_VAR(c)  
    MUL  
    SET_VAR(c)  
    GOTO(L1)  
L3:
```

作业 (0423)

◎ 4. 填写以下生成短路求值代码的 SDT:

产生规则	语义动作
$S \rightarrow ID = E ;$	{ $S.code = E.code \parallel \text{'SET_VAR(\{genvar(ID.lexeme)\})'}$; }
$E \rightarrow E_1 + E_2$	
$E \rightarrow -E_1$	
$E \rightarrow ID$	
$B \rightarrow \text{true}$	{ $B.code = \text{'GOTO(\{B.true\})'}$; }
$B \rightarrow E_1 == E_2$	
$B \rightarrow B_1 B_2$	
$S \rightarrow \text{if} ($ B S_1 else S_2	
$S \rightarrow \text{while} ($ B S_1	
$L \rightarrow$ L_1 S	{ $L_1.next = \text{genlabel}()$; } { $S.next = L.next$; } { $L.code = L_1.code \parallel \text{'\{L_1.next\}:'}$ $\parallel S.code$; } }