

编译原理 · hw5

计01 容逸朗 2020010869

A1

- 函数 p 访问 $d[i]$ 的 $Offset$ 为: $30 + 2*i$;
- $Offset = 27$ 的单元内容应保存数组 d 的起始位置, 即 30 ;
- 若 c 也是动态数组, 此时活动纪录如下所示 :

Offset	0	3	4	6	7	8	9	10	12	$12+2N_c$
内容	控制	a	b	N_c	$\&c$	N_d	$\&d$	e	$c[0]$	$d[0]$

- 此时存放 “指向 d 的指针” 单元的 $Offset$ 值是 9 , 内容是 $12 + 2N_c$ 。

A2

- 对应单元内容如下 :

单元	18	19	21	22	23
内容	0	13	x	0	18

注: 单元 21 中的 x 是 $q.x$

- 当前位置 Display 表内容 :

D[0]	D[1]	D[2]
0	22	13

各活动纪录所保存的 Display 表内容 :

单元	0	5	9	13	18	22
内容	-	-	5	-	9	18

- 若采用动态作用域, 则程序第二次执行至 q_{18} 时 a 的值为 r_{13} 所定义的 3

由于此时 $b=2$, 不满足跳转条件 $a < b$, 故程序不会执行 (L) 语句的跳转, 也不会调用 P 。

A3

- 1. 总大小为 412 + 4N
- 2. ① -412
② 408
③ 404
- 3. ④ 20
⑤ 44
- 4. L 取值大于等于 N+1 时会覆盖函数的返回值 s，
取值大于等于 N+103 时会覆盖函数的返回地址。

A4

- 1. 基本块 B4 的支配结点集合：{B1, B2, B4}
始于 B4 的回边有：B4→B2
基于此回边的自然循环包含：B2, B3, B4, B5, B6
- 2. 活跃变量信息如下所示：

	LiveUse	Def	LiveIn	LiveOut
B1	∅	{a}	{e}	{a,e}
B2	{a}	{c}	{a,e}	{a,e}
B3	{e}	{a}	{e}	{a}
B4	{a}	{b,c,d,e}	{a}	{a,c,d,e}
B5	{a,d}	∅	{a,c,d}	{a,c,d}
B6	{a,c}	{e}	{a,c}	{a}
B7	∅	∅	∅	∅

- 3. 到达-定值数据流分析表如下：

	Gen	Kill	In	Out
B1	{1}	∅	∅	{1}
B2	{2}	{9}	{1,4,5,7,8,9,14}	{1,2,4,5,7,8,14}
B3	{4}	{1,14}	{1,2,4,5,7,8,14}	{2,4,5,7,8}
B4	{5,7,8,9}	{2,11,13}	{1,2,4,5,7,8,9,11,13,14}	{1,4,5,7,8,9,14}
B5	{11}	{7}	{1,4,5,7,8,9,11,14}	{1,4,5,8,9,11,14}
B6	{13,14}	{1,4,6,8}	{1,4,5,8,9,11,14}	{5,9,11,13,14}
B7	∅	∅	{5,9,11,13,14}	{5,9,11,13,14}

- 4. 变量 a 在 (11) 的 UD 链：{(1), (4), (14)}
- 5. 变量 c 在 (2) 的 DU 链：{(3)}

A5

- 利用 DAG 化简合并得到：

```
1 | B := 3
2 | D := A + C
3 | H := D
4 | E := A * C
5 | I := E
6 | F := D + E
7 | J := F
8 | G := 3 * F
9 | K := 15
10 | L := F + 15
11 | M := L
```

1. 只有 G、L、M 在基本块后面还要被引用：

```
1 | D := A + C
2 | E := A * C
3 | F := D + E
4 | G := 3 * F
5 | L := F + 15
6 | M := L
```

2. 假设只有 L 在基本块后面还要被引用：

```
1 | D := A + C
2 | E := A * C
3 | F := D + E
4 | L := F + 15
```

A6

1. 函数定义如下：

$$f(a,b) = \begin{cases} \max(a,b), & a \neq b \\ a + 1, & a = b \end{cases}$$
$$g(a) = a$$

2. 需要寄存器数目的最小值 $n = 3$ ；

生成目标代码如下：

```
1 | LD R0, b
2 | LD R1, c
```

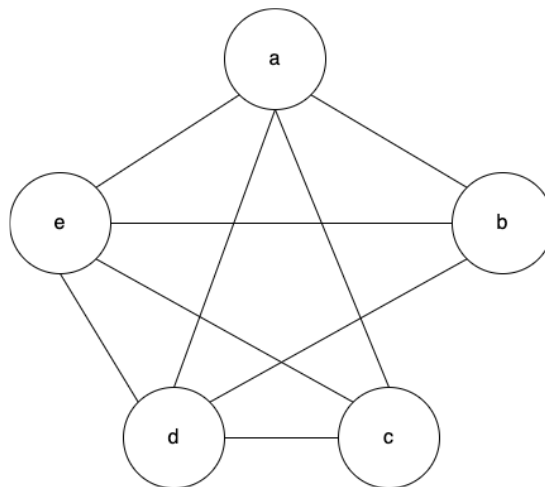
```

3  AND R0, R0, R1  # b∧c
4  LD  R1, a
5  OR  R0, R0, R1  # a∨(b∧c)
6  LD  R1, b
7  LD  R2, c
8  AND R1, R1, R2  # b∧c
9  LD  R2, a
10 NOT R2, R2      # ¬a
11 OR  R1, R1, R2  # ¬a∨(b∧c)
12 OR  R0, R0, R1  # (a∨(b∧c))∨(¬a∨(b∧c))
13 LD  R1, a
14 LD  R2, b
15 NOT R2, R2      # ¬b
16 AND R1, R1, R2  # (a∧¬b)
17 OR  R0, R0, R1  # (a∧¬b)∨((a∨(b∧c))∨(¬a∨(b∧c)))

```

A7

- 可分配物理寄存器最小数目为 4
- 对应寄存器相干图如下：



A8

1. (a) 前向

(b) $In(B_4) = \{Entry, B_1, B_2, B_3\}$, $Out(B_5) = \{Entry, B_1, B_2, B_3, B_5\}$.

(c) Out

2. (a) 后向

(b) $Gen(B_2) = \{a+b\}$, $Kill(B_3) = \{a-d, b*d\}$,

$In(B_4) = \{a+b, e+1, a-d, c-a\}$, $Out(B_1) = \{a+b\}$.

(c) 是；否