

一、(18 分) 给按要求给出如下基本块 (Basic Block) 的局部优化 (Local Optimization):

1、(6 分) 为以下基本块(basic block) 消除公共子表达式, 并在原指令的右边空白处写上优化后调整的指令(不必重写那些不变的指令)。做此题时请勿应用其他的优化技术。

```
g = a + n
t = b * c
u = a + n
t = a * a
a = 2 + a
h = b * c
```

2、(6 分) 针对以下代码片段反复施用复制传播 (copy propagation, 如 $a = b; c = a;$ 可优化为 $c = b;$)、常量折叠 (constant folding)以及代数化简(algebraic simplification, 即利用代数恒等式), 并在原指令的右边空白处写上优化后调整的指令 (不必重写那些不变的指令)。做此题时请勿应用其他的优化技术。

```
c = 3
o = a
m = b + o
p = c * c
i = x * 0
n = m * 2
e = c + p
r = e + n
```

3、(6 分) 在以下基本块中的所有程序点 (program point, 见括号) 处填写活跃变量 (live variable) 集。提示:在基本块入口与出口处的活跃变量集分别是 $\{x\}$ 与 $\{s, e\}$, 且基本块中不包含死代码。

```
                {x}
b = x * 0
                {}
n = b
                {}
a = b * 2
                {}
e = a + n
                {}
s = 2 + 3
                {s,e}
```

二、(34 分) 给定如下中间代码的基本块 (Basic Block):

```
d = b * c
e = a + b
f = a - c
b = b * c
a = e - d
```

1、(18 分) 构造该基本块的有向无环图 (Directed Acyclic Graph, 简称 DAG)。

2、(16 分) 分别有如下假设:

1) 假设#1: 仅变量 a 在基本块的出口 (exit) 是活跃的 (live);

2) 假设#2: 变量 f 和 a 在基本块的出口均是活跃的。

试分上述 2 种不同的假设情况, 分别基于你构造出来的 DAG 对基本块进行优化。

三、(48 分) 给定如下中间代码片段:

```
1:      x = 0
2:      y = 0
3: L0:   if n / 2 goto L1
4:      x = x + n
5:      y = y + 1
6:      goto L2
7: L1:   y = y + n
8:      c = 4 / 2
9:      t1 = x * c
10:     t2 = c - 1
11:     x = x + t2
12: L2:  n = n - 1
13:     if n > 0 goto L0
14:     return x
```

1、(18分) 将上述代码片段划分基本块 (Basic Block), 并画出该代码片段的流图 (Flow Graph)。你可以直接画出流图, 在图中的每一结点中用 $n-m$ 表示该基本块由第 n 至 m 条指令组成。

2、(20分) 为实现代码片段 7-11 的局部优化 (Local Optimization), 请将此段代码转换为一个有向无环图 (Directed Acyclic Graph, 简称 DAG)。

3、(10分) 对代码片段 7-11, 指出其中的两种代码优化方法。