

# 编译原理 - 作业(4) : IR 生成与优化

截至时间: 2023.6.13/周二 14:19:59

提交方式: 超算习堂 (<https://easyhpc.net/course/164>)

## 1. 给定如下代码的基本块 (Basic Block) :

---

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

---

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

(2) 分别有如下假设:

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

## 2. 给定如下中间代码片段:

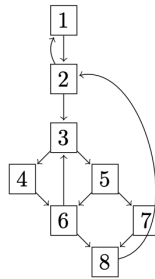
```
1:   x = 0
2:   y = 0
3:  L0: if n / 2 goto L1
4:   x = x + n
5:   y = y + 1
6:   goto L2
7:  L1: y = y + n
8:   c = 4 / 2
9:   t1 = x * c
10:  t2 = c - 1
11:  x = x + t2
12: L2: n = n - 1
13:  if n > 0 goto L0
14:  return x
```

(1) 为上述代码片断划分基本块 (basic block), 并画出该代码片断的控制流程图(control flow graph, 简称 CFG)。你可以直接画出 CFG, 在 CFG 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) 对第7-11条指令片段，列出两种代码优化方法。

(3) 假定所给代码片段（1-14行）来自于函数int Func(int n)，其中n是参数,x、y是局部变量。那么在最终生成的目标代码中，Func被调用时如何访问到n、x和y? 提示：函数调用的栈空间由\$sp（stack pointer，栈指针）和\$fp（frame pointer，帧指针）维护。

3. 给定如下控制流图(control flow graph): [注：需自行学习，详见龙书9.6 Loops in Flow Graphs或[链接](#)]



(1) 节点8的直接支配 (immediate dominator) ?

(2) 画出该CFG的支配树 (dominator tree) ;

(3) 列举出该CFG中的所有自然循环 (natural loops)，给出循环的头节点和其他节点。