Compilers — Homework 4
Liveness analysis
revision 3

Due: Friday, 12 Oct 2012, 13:30

This is a written assignment. For this assignment, each person should submit their own solution. You are free to talk to other students, but (1) you must acknowledge any discussions and (2) you must write up your solutions on your own.

1 Liveness analysis

Given the following code fragments, compute the set of live variables at each program point (i.e., each point *between* two instructions). The first program is in Python. Compute liveness for all Python variables.

The remaining are in x86 pseudo-assembly. Compute liveness for all virtual registers and all physical registers, except `%esp` and `%ebp`. Assume `%eax` is live at the end of the code fragment. Refer to the Intel reference manual to learn which registers each instruction uses or defines. Be careful with multiplications and calls. Assume calls use all *physical* registers and define the *callee-saves* registers `%ebx`, `%esi`, and `%edi`.

(a)
```
x = 1
y = 2
z = x + y
r = x + z
z = z + 1
y = z * 2
r = z - r
return r
```

(b)
```
movl    $3, x
movl    $4, y
movl    y, %eax
imull   x
movl    %eax, w
movl    w, v
```

(c)
```
movl $3, x
movl $4, y
movl x, (%esp)
movl y, 4(%esp)
subl $4, %esp
call _f
movl %eax, z
addl $4, %esp
movl z, %eax
```

(d)
```
movl    $3, a
movl    $4, b
movl    a, %eax
movl    b, %ecx
imull   %ecx
movl    %eax, c
movl    c, %eax
addl    $7, %eax
movl    %eax, d
movl    b, %eax
movl    $9, e
subl    %eax, e
imull   d, e
movl    e, f
addl    b, f
movl    f, %eax
subl    c, %eax
```

2 Interference graphs

Using the liveness results you computed, draw the interference graphs for each of the code fragments above.

3 Coloring

Assume you have only three registers: `eax`, `ecx`, and `edx`. Color the interference graphs from the previous problem using these three registers using the simplification algorithm. Show each step of the algorithm. If the graph cannot be colored, rewrite the instruction sequence, spilling one of the variables. Redo the liveness analysis and interference graph construction and graph coloring. Repeat this process as necessary.