TD11 : Register Allocation - Thuesday, December 7

Exercise 1 Register allocation in a basic block

Consider the following assembly code :

```
Load b -> R1
Load b -> R2
Mult R1 R2 -> R2
Make 4 -> R3
Load a -> R4
Load c -> R5
Mult R4 R5 -> R5
Mult R5 R3 -> R3
Sub  R2 R3 -> R3
```

Question 1 What does this code compute and how many registers does it use ?

Question 2 Write this code in the SSA form.

Question 3 Do the register allocation with fixed ordering and a minimum number of registers. Use the greedy algorithm seen in course.

Exercise 2

We consider the program with the control flow graph given below. We consider that there are no live variables at the output of the program.

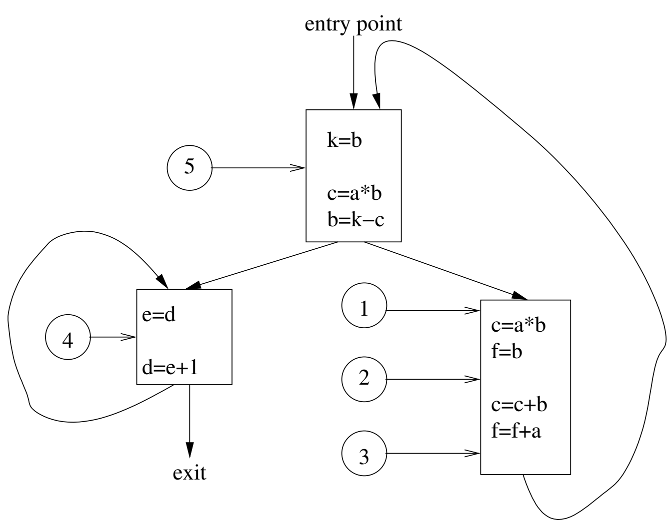

FIGURE 1 – Control flow graph

Question 4 Give the alive variables for points 1 to 5 in the program.

Question 5 Build the interference graph of the variables (nodes represent variables and edges represent the interference between the variables). Using the interference graph, explain the difference if the code would have been in SSA.

Question 6

In order to perform a register allocation, propose a coloring of the obtained graph. How many registers are there needed in order to execute the above code without needing to spill ?

Question 7 Consider now that we exchange the instructions : $f = b$ and $c = c + b$ so that the content of the basic block becomes :

```
c=a*b
c=c+b
f=b
f=f+a
```

What are now the alive variables before the instruction $f = b$? What about the number of colors needed to color the interference graph ?

Question 8 The instruction exchange operation performed above is legal, because it does not change the result of the program execution. Explain the conditions that the following two instructions must verify such that they may be exchanged while preserving the same semantics of the execution :

```
x1 = y1 op z1
x2 = y2 op z2
```

Question 9 One way to force 2 variables to share the same register is to join their nodes in the interference graph (this allows reducing the number of moves between registers). Explain in what conditions is it legal to join nodes in this way.

Exercise 3 Node deletion problem (on the interference graph)

Typically, *spilling* is done during the coloring phase : if the greedy algorithm is blocked, one or several variables are splilled in order to unblock it and lower the register pressure. In practice, on a basic block in SSA form, we can find, beforehand a sufficient set of nodes to be removed such that the resulting interference graph is k-colorable : it is enough to decrease the maximum number of variables alive simultaneously, MAXLIVE, to k (the number of registers). In other words, we have to remove live intervals until MAXLIVE is k.

Question 10 Consider a set of live intervals $\{I_1, ..., I_n\}$ on a basic block ; k the number of registers. Give a polynomial algorithm which eliminates a minimum number of intervals s.t. MAXLIVE becomes smaller than k.

Question 11 Spilling a variable which has many uses is usually more expensive than spilling a variable which has few uses. In fact we can ponderate the intervals with an approximation of the splilling cost. Is there a polynomial time algorithm for solving this problem ? Hint : Express this problem as an ILP problem. We know that ILP problems with totally unimodular matrices are solvable in polynomial time. See also, F. Bouchez, A. Darte and F. Rastello, "On the complexity of spill everywhere under SSA form", 2007.