CS 415 Compilers: Problem Set 2

Spring 2022

Due date: Wednesday, February 16, 11:59pm

Problem – Top-down and bottom-up register allocation

Assume the following ILOC code

```
loadI 1024 => r0
loadI 1 => r1         // a := 1
loadI 2 => r2         // b := 2
subI r2, 4 => r3      // c := b - 4
add r1, r2 => r4      // d := a + b
addI r4, 1 => r5      // e := d + 1
mult r3, r5 => r6     // c * e
sub r5, r6 => r7      // f := e - (c * e)
add r4, r5 => r8      // d + e
add r8, r7 => r9      // g := (d + e) + f
add r9, r1 => r10     // h := g + a
storeAI r10 => r0, 4 // printing requires value to be in memory
outputAI r0, 4       // print @h = 4 , h is only value in memory
```

1. Give the live ranges for all virtual registers (ignore r0). What is MAX_LIVE of the ILOC code?

2. Assume that there are two registers in the feasible set, and there is the dedicated machine register r0 that we do not consider for register allocation, i.e., is not part of the available register set. Note: It is your choice which two physical register you want to put aside as the feasible set.

   (a) Show the ILOC code that would be generated by the top-down algorithm discussed in EAC, i.e., with no live range consideration, for (MAX_LIVE - 1) available registers for allocation (k-F registers are available). Describe the heuristic that you are using.

   (b) Show the ILOC code that would be generated by the top-down algorithm discussed in class, i.e., with MAX_LIVE consideration, for (MAX_LIVE - 1) available registers for allocation (k-F registers are available). Describe the heuristic that you are using.

3. Show the ILOC code that would be generated by the bottom-up algorithm discussed in class for (MAX_LIVE - 1) available registers for allocation (no need for a feasible set). Describe the heuristic that you are using.