CS 132, Jens Palsberg Mar 15, 2010

Final Exam

Each of the six questions is worth five points, for a total of 30 points. Write your name and id number at the top of the first page you submit. Write the solutions for different questions on different sheets of paper. Don't submit this handout.

1. [**Type checking**] Consider the following Java program:

```
class C {
  int[] x;
  void run(int b) {
    x = new int[b];
    for (int i=0; i<b; i++) {
      x[i] = 3;
    }
    this.set(7,14);
  }

  void set(int j, int k) {
    x[j]=k;
  }
}
```

Show the symboltable at the point where the type checker will try to type check the statement `this.set(7,14);`.

2. [**Translation**] Consider again the Java program from question 1. Make a figure that illustrates a heap layout for a `C`-object. Present the result of translating the body of the method `run` in class `C` to intermediate code in the style of Piglet, with reasonably detailed comments in the code.

3. [**Stack layout**] Consider again the Java program from question 1. Make a figure that illustrates a MIPS stack layout for the method `run` in class `C`. Assume that any time there is a choice of where to represent data, the data is represented on the stack.

4. [**Liveness analysis**] Draw the control flow graph for the following program, and show the edges on which each of the variables `a`, `i`, and `y` are live.

```
    i = 1
    b = i + 2
    y = i + b
L1: if i > 20 then goto L2
    print b+y
L2: b = b+1
    if b > 10 then goto L3
    i = i+1
    y = b+1
    goto L1
L3: return y
```

5. [**Register allocation**] The following program uses six temporaries `a`, `b`, `c`, `d`, `e`, `f`. What is the fewest number of registers that is needed for this program, *without* spilling? Justify your answer by showing the interference graph and a coloring of the inference graph. Show also the program after register allocation; assume that the needed registers are available.

```
a = 1
b = a + 2
c = b + 3
d = a + c
e = b + 2
f = 4
b = d + 1
c = e + f
a = b + c
d = c + e
return a + d
```

6. [**Linear scan register allocation**] Consider again the program from question 5. First, assume that there are three registers available: `r1`, `r2`, `r3`. Make a figure that illustrates a run of the linear scan register allocation algorithm on this program. Show the program after the linear scan register allocation algorithm is done, including load and store instructions for handling spilled variables. Assume that registers `v1` and `v2` are available for short-lived transfers of data between registers and memory.

Second, assume that there are two register available: `s1`, `s2`. Make a figure that illustrates a run of the linear scan register allocation algorithm on the program. Show the program after the linear scan register allocation algorithm is done, including load and store instructions for handling spilled variables. Assume that registers `v1` and `v2` are available for short-lived transfers of data between registers and memory.