CS 132, Jens Palsberg Dec 15, 2006

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 B {
  int i;
  int p(int a) { i = i-a;
     if (i>0) { return this.q(i-1); }
     return i;
  }
  int q(int a) { i = 2*i+3*a;
     if (i<200) { this.p(i-30); }
     return i;
  }
}

class C extends B {
  int j;
  void r(int a) {
     int x;
     j=j+1;
     x = new C().p(-10);
     i = i+4*a+j;
     System.out.println(x+i);
  }
}
```

Show the symboltable at the point where the type checker will try to type check the expression `i+4*a+j`.

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 `r` 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 `r` in class `C`. Assume that the compiler does not map any temporaries, variables, or arguments to registers. If there are major design choices, then include a brief discussion of those choices.

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.

```
      y = 0
      a = 1
 L1:  if y < 0 then goto L2
      a = y
 L2:  i = a + 2
      y = i + 1
      if i < 100 then goto L1
      return i
```

5. [**Register allocation**] In the following program, there are 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.

```
a = 1
b = 10
c = a + b
d = b + c
e = a + c
f = b + d
d = c + f
d = d + f
b = d + e
return d+f
```

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.

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.