Midterm Exam
CS6013

12-Mar-2016

Read all the instructions and questions carefully. You can make any reasonable assumptions that you think are necessary; but state them clearly. There are total four questions, totalling 40 marks (+ 4 bonus marks). You will need approximately 30 minutes for answering a 10 marks question (plan your time accordingly). The bonus part may take more time. For questions with sub-parts, the division for the sub-parts are given in square brackets.

You will get an answer sheet with 12 pages (if you get a answer sheet with fewer pages then ask for a replacement sheet). Leave the first page empty. Start each question on a new page. Think about the question before you start writing and write briefly. Each question also specifies the maximum number of allowed pages for the question. If the answer for any question is spanning more than specified number of pages, we will strictly ignore the spill-over text. If you scratch/cross some part of the answer, you can use space from the next page.

1. [12 marks, 2 pages] **Code Generation**:
   Come up with a translation scheme to translate a program written in object oriented language (like MiniJava) to a procedural language (like C). The only OO features allowed in the language are inheritance, virtual functions, and overriding of thereof. The language uses the syntax of Java (like MiniJava). Use your scheme to generate code for the following MiniJava code (You are required mainly show the translated code. Do mention any salient features of your translation scheme).

```
class A {
  int x, y;
  int foo(){
    return x + y;
  }
  int bar(){
    return foo();
  }
}
class B extends A{

  int y, z;
  int foo(){
    return x + y;
  }
}
```

```
class Main(){
  public static void main (String args[]){
    class A x1, x2;

    x1 = new B();
    x2 = new A();
    x1.x = 5;
    x1.y = 6;
    x1.foo();
    x2.foo();
  }
}
```

2. [10 marks + 4 marks Bonus, 2 pages] **Available Expression Analysis**:
Consider the following code in three address code:

```
L1: t1 = q * r;
x = p + t1;
L2: t2 = q * r;
y = t2 / s;
L3: q = 5;
L4:
```

It can be easily seen that the expression `q * r` is available at `L2` and need not be recomputed. But this expression is not available at `L4`, as one of the constituent variables of the expression is reassigned at `L3`. Consider another piece of code.

```
b = e1;
t0 = b * c;
if (cond)
then
  c = e2;
  L1: t1 = b * c;
  t2 = t1 + x;
fi
L2: t3 = b * c;
```

Similar to the previous example, the expression `b*c` is available at L2. But if the line labelled `L1` is commented out, the expression `b*c` is not available at `L2`.
Present an iterative data flow analysis to compute the available expressions at each program point in a given function. Apply your algorithm on the above codes and show a trace [2 + 2 bonus marks, will take around 15 minutes].

3. [12 marks, 2 pages] **Structural Analysis**:
Perform structural analysis to identify constants in the following program:

```
i = 2;
j = 3;
k = 2;
l = 5;
n = input;
while (i < n)
  if (i % 2) {
    k = (j/2)*2;
  }
  l = l * 1;
  i = i + 1;
}
print i, j, k, l;
```

Show the control tree [3] and the flow functions for the nodes of the control tree [6]. Use the flow functions to compute the constants at each program point [3].

4. [6 marks, 1 page] **SSA**:
Why are the phi nodes inserted at the iterated dominance frontiers (IDF) of the set of definitions of each variable? Why are all the phi nodes (inserted at the beginning of a basic block) assumed to be evaluated in parallel and not in sequence? Illustrate/Answer these questions using examples [3 + 3].