# CSE 131b: Compiler Construction Final

Lingjia Tang
Lingjia@cs.ucsd.edu

June 14th 2013
Time: 80 Minutes

Start here: Please fill in the following information using a permanent pen before you do anything else! Your exam will not be graded if you use a pencil or erasable ink on this page.

Name ______________________

PID _________________________

This is a close book, close notes (no laptop, phone, Internet, etc) exam.

| | | |
|---|---|---|
| **1. Parameter Passing** | **20** | |
| **2. Activation Tree and Stack Frame** | **20** | |
| **3. Memory Layout for Objects + Vtable** | **20** | |
| **4. Three Address Code** | **20** | |
| **5. Local Analysis and Optimizations** | **20** | |
| **6. Register Allocation** | **20** | |
| **Total** | **120** | |
| **Extra Credit** | **10** | |

**Problem One: Parameter Passing (20 points)**

1a. What is printed if we are using call by value? Call by value result? Call by reference? (10 points)

```
int g;

void foo(int arg)
{
   g += 2;
   arg -= 1;
   printf("%d ", g);
}

main()
{
   g = 2;
   foo(g);
   printf("%d ", g);
}
```

1b. What does the array arr contain when we reach the CHECKPOINT if we are using call by value? Call by reference? (10 points)

```
int arr[2];
int g;

void foo2(int arg)
{
   g += 1;
   arg += 50;
}

void main(void)
{
   arr[0] = 100;
   arr[1] = 200;

   g = 0;

   foo2(arr[g]);
   /* CHECKPOINT */
}
```

**Problem Two: Activation Tree and Stack Frame (20 points)**

2a. Sketch the activation tree of the following code when factorial(5) is called. ( 5 points)

```
int factorial(int number) {
    if(number == 0) {
           return 1;
        }
        factorial_i(number, 1);
}

int factorial_i(int currentNumber, int sum) {
    if(currentNumber == 1) {
        return sum;
    } else {
        return factorial_i(currentNumber - 1,
sum*currentNumber);
    }
}
```

II. 2b. Fill in the content of the stack frame a) right before factorial_i(5,1) is called from factorial(5); and b) when factorial_i(5,1) is executing, including where frame pointer is pointing to. (15 points)

| |
|---|
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |

**Problem Three: Memory layout and virtual Table (20 Points)**

4a.. Sketch the memory layout for the following:

I. int array[3][5] C-style array. Specify where array[i][j] is stored. (5 points)

II. A and B (5 points)

```
class A
{
int i;
char c;
int j;
}

class B extends A
{
int k;
}
```

4b. In class we discussed using virtual table to implement dynamic dispatch. Please sketch the **vtable** for the following objects: (10 points)

```
class A {
    int x;
    void Callme() {
        Print("in A");
    }
    void SayHi(){
  Print("Hi");
    }
    A clone() {
  return new A;
    }
}

class B extends A {
   int y;
   void Callme() {
        Print("in B");
    }
   B clone() {
     return new B;
   }
}
```

**Problem Four: Three Address Code (20 points)**

I. What’s three address code? (5 points)

II. Write the TAC code for the following two functions: (15 points)

```
void Foo(int i, int j, int k)
{
     int x;

     x = i + i * l;
 }

void main()
{
   Foo(4,5,3);
}
```

**Problem Five: Local Optimization (20 points)**

I. What are compiler optimizations optimizing? (2 points)

II. Identify the basic blocks and sketch the control flow graph for the following code (6 points)

```
main:
   BeginFunc 24;
   _t0 = x * x;
   _t1 = y * y;
   m2 = _t0 + _t1;
_L0:
   _t2 = 5 < m2;
   IfZ _t2 Goto _L1;
   m2 = m2 – x;
   Goto _L0;
_L1:
   EndFunc;
```

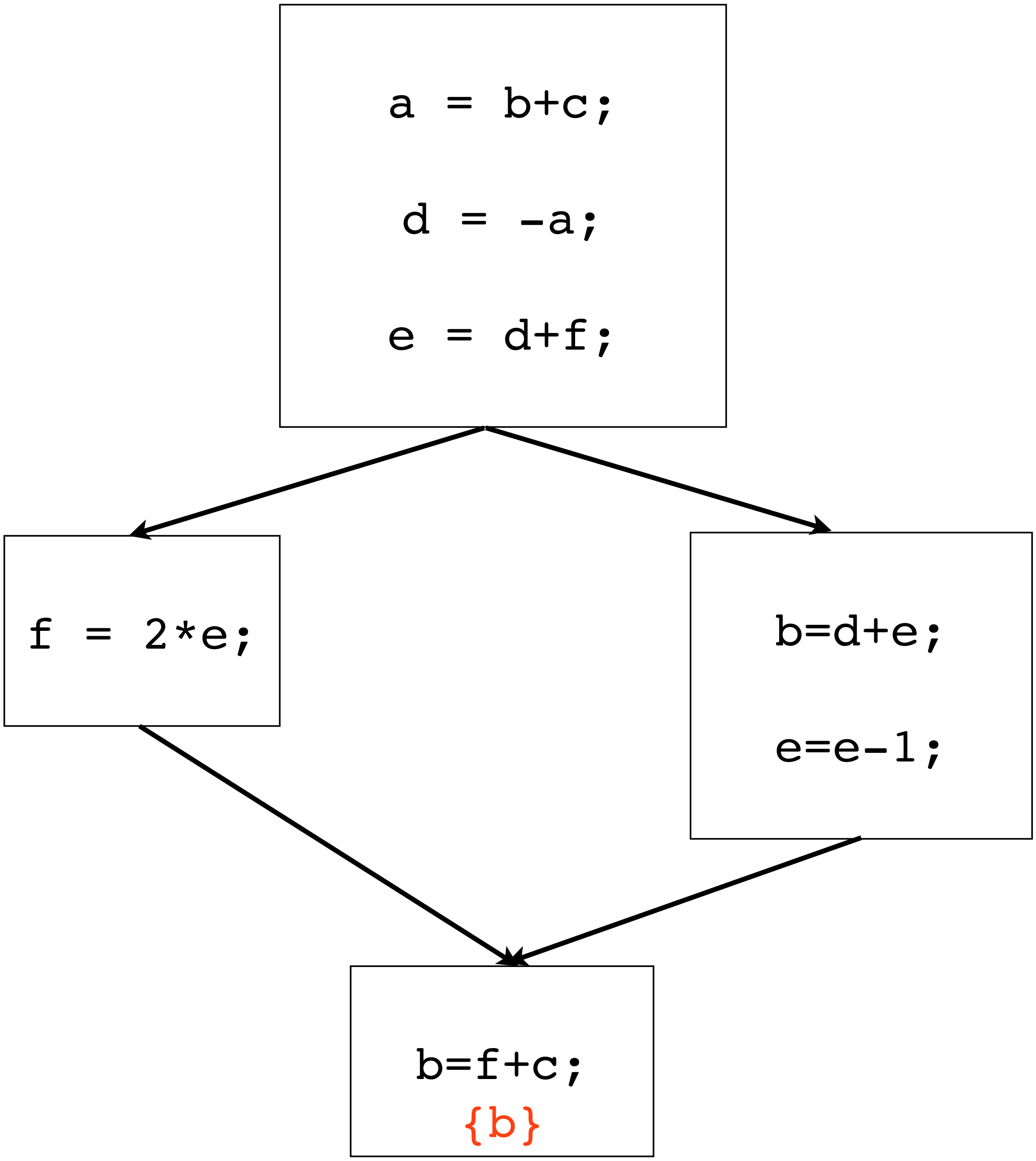
III. Available Analysis of the following code: (6 points)

```
 a = b + C;

 d = b;

d = b + c;

 d = b;

 c = b;

f = b + c;
```

IV. Apply Common Sub-expression Elimination based on the above Available Analysis (6 points)

**Problem Six: Register Allocation (20 points)**

I. Compute live variables for each point on the following CFG: (5 points)



```
a = b+c;
d = -a;
e = d+f;
```

```
f = 2*e;
```

```
b=d+e;
e=e-1;
```

```
b=f+c;
{b}
```

II. Sketch the Register Interference Graph for the above CFG: (5 points)

III. Color the above RIG with 4 colors using Chaitin’s algorithm. Show the steps in the algorithm (10 points)

**Extra Credit (10 points)**
Who gave a guest lecture this quarter?
What did he/she talk about?