CS6004: Code Optimization for Object-Oriented Languages

Spring 2024 (Due: March 31^{st}, 2024)

Assignment PA3: Collect Me If You Can.

---

1 Assignment Objective

Use Soot's existing liveness analysis to find out the place where an object can be garbage collected safely.

2 Detailed Specification

Garbage collection is used in programming languages like Java to automatically free up the heap memory. If a memory location cannot be referenced after a certain statement, the memory can be freed up after that statement using a garbage collector. Your objectives in this assignment are to do:

- **Interprocedural points-to information**: You need to generate flow-sensitive interprocedural points-to information for each method. In the last assignment you had generated intraprocedural points-to information where at each callsite you had conservative result. Now at each callsite you need to map the effects of callee(s) back in the caller to get the interprocedural points-to information.
- **Find the line at which an object can be collected**: Using Soot's existing liveness information, find out at which line number an object can be garbage collected. Print results in lexicographic order.
- For simplicity, you can assume that there will not be any recursion or global variable in the testcases.
- You can also ignore the results for constructors and libraries while printing the final result.
- We will provide a "**pa3-starter.zip**" file that you can expand upon for the assignment.

3 Sample Input Program

```
 1 class Node {
 2   Node f;
 3   Node g;
 4   Node() {}
 5 }
 6 
 7 public class Test {
 8   public static Node global;
 9   public static void main(String[] args) {
10     Node a = foo();
11   }
12   public static Node foo(){
13     Node x = new Node(); // Object O13
14     Node y = new Node(); // Object O14
15     y.f = new Node();    // Object O15
16     y = new Node();      // Object O16
17     bar(x, y);
18     Node z = y.f;
19     Node a = x.f;
20     return x;
21   }
22   public static void bar(Node p1, Node p2){
23     Node v = new Node(); // Object O23
24     p1.f = v;
25   }
26 }
```

Expected Output:

```
Test:bar
Test:foo 14:15 15:15 16:18
Test:main 13:10 23:10
```

4 Submission

Your submission must be named `rollnum-pa3.zip`, where rollnum is your roll-number in small letters. Upon unzipping the submission, we should get a directory named `rollnum-pa3`. The main class inside this directory should be named `PA3.java`. `PA3.java` should contain all the flags passed to soot along with invoking the escape analysis. Your submission should consist only of Java files (that is, no jars or class files). We will test the code using the version of Soot uploaded on piazza.

For evaluating the assignment, for a testcase Test.java, we would run the following commands:

- `javac -g Test.java`
- `javac -cp .:sootclasses-trunk-jar-with-dependencies.jar PA3.java`
- `java -cp .:sootclasses-trunk-jar-with-dependencies.jar PA3`

If the printed output matches the expected output, you will get the marks for the corresponding testcase.

5 Plagiarism Warning

You are allowed to discuss publicly on piazza, but are supposed to do the assignment completely individually. If plagiarism is found:

- First instance: 0 marks in the assignment
- Second instance: FR grade in the course
- Third instance: report to institutional committee

-*-*-*- Do the assignment honestly, enjoy learning the course. -*-*-*-