HW 1: Control-flow Analysis

Summary:

Write a tool in Java that can analyze the bytecode of the main method of a class and produce its intra-procedural control-flow graph (CFG) as output. All bytecode instructions related to control-flow, with exception of JSR, must be accounted for. Label each conditional control-flow edge with the corresponding jump condition. Each node should be labeled with bytecode offset number it represents. Assume basic blocks of size one.

Rules:

1. Code must be written in Java.
2. Use the Soot library to process JVM bytecode. Recommended: use the complete package or pre-compiled jars.
   a. https://www.sable.mcgill.ca/soot/
   b. https://sable.github.io/soot/
3. Output CFG as a file viewable using dotty. (http://www.graphviz.org/)
4. Command line arguments to the tool will be (1) a path specifying the location of the class (i.e., bytecode version) file and, (2) a path specifying the name of the output graph, in that order. Paths may be relative or absolute.
5. Submission: Submit an executable jar, named CFG.jar and the source for the code you wrote.
6. You may help classmates to understand how to use and access Soot, but code and algorithms must be your own.

Hints:

1. Remember that the class path Soot uses will be different from Java's class path
2. Test thoroughly!

Evaluation:

Tool will be run on a set of Java classes and output graphs compared against the correct solutions. The command to run your jar will be java –jar CFG.jar (1) (2) where the (1) and (2) refer to the command line arguments specified above.

Assigned date: 8/24/16
Due date: 9/7/16
Credit: 7.5%