OMSCS 6340 - Fall 2016
Assignment 5 (100 points)
Due at: 8:00am EDT, October 10, 2016

Objective

The objective of this assignment is to implement two intra-procedural dataflow analyses, one forward (reaching definitions analysis) and one backward (liveness analysis).

Resources

- Chord Project Webpage: http://www.cc.gatech.edu/~naik/chord.html
- Chord Repository: https://bitbucket.org/pag-lab/jchord
- Chord User Guide: http://pag-www.gtisc.gatech.edu/chord/user_guide/index.html
- Chord Javadoc: http://pag-www.gtisc.gatech.edu/chord/javadoc/
- Download the zip file **dataflow.zip** from T-Square which when uncompressed should produce a directory named **dataflow/**.

Setup

A Linux or MacOS machine is preferred for running Chord. You will need to install the following software
on your machine if not already present:

- If the command “**javac**” is not found on your machine, then install a JVM, e.g. Oracle HotSpot from http://www.oracle.com/technetwork/java/javase/downloads/index.html. You will need a JVM that supports Java 6 or 7 (note that Chord does not support Java 8). You can find the version number by running “**javac -version**”.
- If the command **ant** is not found on your machine, then install Apache Ant (a Java build tool) from http://ant.apache.org/.

Build the given example Java program to be analyzed, called ArrayDemo, by running “**ant**” in directory
dataflow/cs6340/examples/array demo/. This should produce sub-directory **classes/** in that directory. You can create your own test cases similar to ArrayDemo in the **examples/** directory to test the analyses you will write.
Write your liveness analysis in file **cs6340/src/LivenessAnalysis.java** and your reaching definitions analysis in file **cs6340/src/ReachDefAnalysis.java**. Follow the instructions in those files for how to write the analyses.
To run the liveness analysis and print its output for the ArrayDemo example, run the following commands in directory **dataflow/cs6340/**:

```
ant compile
ant -Dchord.work.dir=examples/array_demo/ liveness
```

This should print output at **dataflow/cs6340/examples/array_demo/chord_output/log.txt** similar to:

```
Chord run initiated at: Mar 24, 2016 1:14:04 PM
OPT config = data.config
ENTER: liveness at Thu Mar 24 13:14:05 EDT 2016
********** liveness analysis *********
ENTER: RTA
Iteration: 0

Infinite loop discovered in run:()V@java.lang.ref.Reference$ReferenceHandler,
linking
BB6 to exit.
Infinite loop discovered in
decodeBuffer:(Ljava/io/InputStream;Ljava/io/OutputStream;)
V@sun.misc.CharacterDecoder, linking BB9 to exit.
Iteration: 1
Infinite loop discovered in run:()V@sun.security.provider.
SeedGenerator$ThreadedSeedGenerator,linking BB15 to exit.
Iteration: 2
Iteration: 3
Iteration: 4
LEAVE: RTA
Time: 00:00:08:735 hh:mm:ss:ms
Control flow graph for main:([Ljava/lang/String;)V@ArrayDemo:
BB0 (ENTRY) (in: <none>, out: BB2)
...
Exception handlers: []
Register factory: Registers: 26
********** ENTER Analysis Results **********
********** LEAVE Analysis Results **********
LEAVE: liveness
Exclusive time: 00:00:08:874 hh:mm:ss:ms
Inclusive time: 00:00:08:874 hh:mm:ss:ms
Chord run completed at: Mar 24, 2016 1:14:14 PM
Total time: 00:00:10:246 hh:mm:ss:ms
```

Similarly, to run the reaching definitions analysis and print its output for the ArrayDemo example, run this command:

```
ant -Dchord.work.dir=examples/array_demo/ reachdef
```

You can verify your solution by comparing your results against the sample results provided at:

dataflow/cs6340/examples/array_demo/results/liveness.txt, and **dataflow/cs6340/examples/array_demo/results/reachdef.txt.**

Submission Instructions

Upload your completed files **`LivenessAnalysis.java`** and **`ReachDefAnalysis.java`** to T-Square.