Manual Taller #1: SOOT

El objetivo de este documento es presentar el modo de utilización de la herramienta SOOT. La herramienta SOOT contiene un conjunto de algoritmos de ***dataflow analysis***.

Todos los pasos están pensados para una máquina linux. Tener en cuenta que si se tiene Windows o IOS los comandos pueden cambiar pero los pasos a seguir deberían ser los mismos.

A continuación explicaremos cómo configurar SOOT para luego ver cómo utilizarlo.

Configurando SOOT

- Instalar JAVA. Preferentemente Java 8
- Bajar el .jar de SOOT desde este link.
- Colocar en una carpeta que se llame "soot" el jar bajado.
- Setear ***$JRE*** a la dirección donde está el rt.jar. En mi caso es con el comando:

export JRE=/usr/lib/jvm/java-8-openjdk-amd64/jre/lib/rt.jar

Utilizando SOOT

Supongamos que queremos realizar un Reaching Definition Tagger sobre una clase. Para esto, lo primero es compilar el archivo java. Consideremos entonces que tenemos un archivo **A.java** en la carpeta soot creada. Vamos hasta la dirección de la carpeta soot y compilamos el archivo **A.java** utilizando javac

javac -g A.java

Esto creará un archivo **A.class** Ahora que tenemos el código compilado debemos ejecutar el análisis con el siguiente comando

java -cp .:soot-3.3.0-jar-with-dependencies.jar:. soot.Main -cp .:$JRE -f J A -print-tags -p jap.rdtagger enabled:true -p jb use-original-names:true -p jb.cp off -keep-line-number

¿Qué hace este comando?

- ***-cp*** setea el classpath del análisis. Es decir, el mismísimo classpath que tiene soot y el classpath que tiene el rt.jar (notar que hay dos flags -cp).
- ***-f*** establece el formato del output de SOOT. En este caso le pasamos el parámetro J para que genere un archivo en formato Jimple (formato recomendado por la cátedra).

- La ***A*** suelta es para decirle a SOOT que analice **A.class**.
- ***-print-tags*** le indica a SOOT que imprima en pantalla los tags luego de la línea
- ***-p*** indica que tipo de análisis se quiere realizar. Un manual con todos los tipos de análisis posibles puede verse aquí.
- En nuestro ejemplo utilizamos ***jap.rdtagger,*** este le pede a soot que realice un reaching definition tagger.
- El resto de los flags: ***use-original-names:true*** y ***jb.cp off***, deben mantenerse porque así es como mejor se comporta SOOT.

Entendiendo el output de SOOT

La herramienta debería crear un directorio *"sootOutput"* donde podrán encontrar los archivos en formato Jimple. Dependiendo del análisis que se corra se tendrá un output distinto pero explicaré levemente el output de un reaching definition tagger.

INSTRUCCIÓN
/*N*/
/* c has reching def: c = 1 */

Luego de la instrucción ***INSTRUCCIÓN*** de la línea ***N*** se tiene que la variable ***c*** está viva y vale ***1***.

Cada análisis tendrá su nomenclatura. Podrán darse cuenta solos que es lo que el análisis está diciendo.