Assignment – 3

Interprocedural Context Sensitive Sign Analysis

Name: Shradha Agarwal

Roll Number: MT20123

How to execute the code

1) Create a package named -> CallGraphCreation
2) Add the given java files
3) Add the given jar files
4) Versions – JDK 1.7.0 java compliance level 1.7
5) Main class is in InterprocAnalysis.java and the test file is Sample_test.java

Implementation

1) This code is in continuation with the previous assignment on intraprocedural analysis.
2) The function **Handle_Invoke(Stmt u)** is created for the function call statements.
3) For all the parameters, their required signs are extracted and put in func_par_sign.
4) Now the function is called and the return variable sign's value is stored in return_sign.
5) In the flowThrough function, initially it is checked if the RHS is a function call. If it is, then the function name along with parameters are stored in the rhs_array for the unary RHS. For binary expressions, rhs_array stores both the operands along with the operator.
6) Now, the resultant sign for the function calls is extracted from the function Handle_Invoke(). In the case of binary expressions, after extracting both the operand's signs, the final sign for the LHS variable is calculated by using the operator table given in the Anders Moller's book.
7) When the return statement is reached, the return variable's sign is stored in the func_ret_sign hash map.
8) For the identity statements, the mapping of the parameters with their required signs is done.
9) Rest code is same as in Intra sign analysis.