Assignment 2 Program Analysis

Name: Shradha Agarwal

Roll No.: MT20123

Project Name: ReachingDefinitionAnalysis1

Package name: bitvectorrd

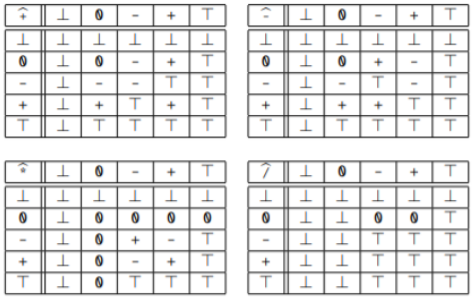
Source files, generated Jimple code and the output of the sign analysis can be found in the folder

Note: Modification of the code of reaching definition shared in the second tutorial

Analysis: forward flow analysis

Signs considered: + for positive, - for negative, 0 for zero, T for don't know, b for not a number

Referred Table:



Implementation:

1) Pairs stored for each block: all local variables and their signs
2) The symbol '#' is added at the beginning of the variables which are not integers. This is done to demarcate them and not killing them in the kill stage.
   For assignment statements:
3) The pairs in which the local variable matches with LHS is killed
4) For generation of the required pair, several cases for the RHS are checked.
5) The entire RHS is read character by character and the operands are extracted. This information is stored in rhs_array.
6) Now, for the unary expressions, the rhs_array size will be 1.
7) The varied cases for the unary expressions considered are variable = number; variable = integer array element; variable = 0; variable = - variable; variable = - number; variable = variable.
8) Now, for the binary expressions, the rhs_array size will be 3 (2 operands and 1 operator).
9) Initially, the sign of each operand is extracted in sign1 and sign2 respectively.
10) The varied cases for assigning the sign to an operand considered are number, negative number, zero, variable
11) The case where the killed variable was same as the LHS variable is handled by storing the variable's sign before killing it
12) For the operators +,-,*,/; the LHS sign is calculated using the if-else cases build by observing the Anders Moller's book.
13) Finally, the updated set of local variables is output for that block

14) For the merge points, the sets of the two branches are compared for each variable.
15) This is done with the help of dictionary for storing key-value pairs for one set and then comparing the signs for the other set.
16) If there is a local variable for which the signs don’t match, the resultant sign is changed to top.
17) Else there is no change.
18) The results can be checked for the provided test cases.