CSE503: Program Analysis

Homework Assignment 1

IIIT-Delhi. 03rd August 2016. Due on 16th August 2016.

Problem Statement: You are required to perform a static taint analysis for programs written in the language Toy. The broad syntactic rules of this language have been specified at the end. You are free to make your own assumptions wherever necessary. **Please take care that your assumptions should not trivialize the problem.**

Input will be a source code which has to be read from a file. The output should state whether the variables which are returned from the function are tainted or not. This output should be accompanied with the line numbers where they are possibly tainted. List out your assumptions explicitly in a README file. **When in doubt, double check with your TAs if the assumptions are valid.**

Sample Input:

```
1. int compute(int x, int y) {
2.   int res = x*2;
3.   int a = 0;
4.   while (y != 0) {
5.          if (x >= 0) {
6.                 a = 2 * res + (y%2);
7.                 y = y / 2;
8.                 x--;
9.          }
10.                else {
11.                res = a * 2;
12.         }
13.  }
14.  res = 9;
15.  if(x > 0)
16.         return a;
17.  return res;
18.}
```

Sample Output:

```
Potentially tainted variables at the start of return statement:
{<a, 6>}
Variables which are untainted: <res>
```

Explanation:
Variable a may become tainted in line 6. Although res gets tainted at line 2 and 11, it is redefined to a new value (9) at line 14. Variable res becomes tainted at line 11 because a might be tainted at this line (because of the iterations of while loop).

Syntactic rules of Toy programs:

1. The program file should contain a function definition. You will be required to demonstrate the analysis on this function. Avoid using access modifiers or special keywords like 'final' and 'static' with these functions.
2. There can be one or more number of arguments in the function definition. All these passed arguments are to be taken as tainted variables.
3. Please avoid User Defined data types or arrays in this assignment. Also, consider only 'int' data type.
4. An assignment statement in a function can contain only variables on LHS and '=' as the assignment operator. Also, it can contain only variables or literals with one or more of these operators: +, -, /, * and any number of operands on the RHS.
5. Only one type of conditional statement: 'if-then-else' and only one type of loop: 'while' loop can be present.
6. Please note that a function can contain any number of these constructs (if-then-else and while) and can be nested with only one level.
7. Conditions inside while and if statements may have only one relational operator (<, <=, >, >=, ==, !=) and two operands, each one of which can be either a variable or a literal. You need not evaluate these conditions. You can safely assume that these conditions can be true or false.
8. **One or more number of return statements may appear anywhere** and they have their usual semantics.
9. You do not have to do error-checking and assume that the given code is syntactically correct.
10. You may assume that every statement in the program appears on a distinct line.