Homework 1 (due Apr. 9)

Your first homework assignment contains small exercises for practice, getting familiar with tools, and experimenting. You can work on this during the last hour of each class meeting, so you can ask for help directly. Submit via email.

Write Datalog programs to implement the following concepts:

1) Control Flow:
- Dominates(block1, block2): there is no path from the function entry to block2 that doesn't pass through block1
- PostDominates(block1, block2): there is no path from block2 to the function exit that doesn't pass through block1
- LoopHeadBlock(block): the block is the head of a structured loop. Hint: use Dominates.

- Controls(statement, block): the statement is a check (i.e., JUMPI instruction) that decides whether the block will be executed, directly or indirectly.

2) Data Flow:

- Flows(fromVar, toVar): there is flow of information from variable fromVar to toVar. This flow can be direct assignment (hint: see relations Statement_Defines and Statement_Uses), phi assignment, indirect assignment (i.e., function call with a variable as an actual argument is an inderect assignment to the corresponding formal argument variable of the function, and similarly for return variables), as well as any combination of the above.

3) Client Analysis:

- Reentrancy(callStmt, storeStmt): Find calls (CALL instruction) that could be vulnerable to reentrancy. You should at least write code to find CALL instructions that later lead to SSTORE instructions. However, this simple analysis will be imprecise. Can you think of ways to make it more precise? (Example: a check of whether the CALL instruction is due to the translation of a Solidity “transfer” statement, i.e., is given constant 2300/”0x8fc” gas and throws if the call fails.)

Demonstrate your analyses on at least one Solidity program that you write by hand (although you can also experiment with all the contract bytecodes given on the server).