CS 565, Jens Palsberg Apr 11, 2001

# Homework 7: Java Security Checker

The code of the OVM framework is on the CS 565 webpage. The goal of the homework is to implement a security checker that infers the confined types of a closed-world Java program or library in the sense of Bokovski and Vitek.

Your program gets two arguments. The first is a file which contains a list of classnames.[1] The second argument is the path to the classfiles. Your program should output the names of the confined classes in the same format as they are in the input file. You may assume that the bytecode is correct (verifies).

Together with the OVM framework you are provided with a skeleton for your code, consisting of three classes `Driver`, `ConfinementPolicy` and `ConfRewriter` in the package `ovm.tools.conf`. You may not modify files outside of that package.[2]

[1] This file can be generated using the `buildargs.sh` program.
[2] If you think you have to, contact grothoff@cs.purdue.edu first.

The definition of confined types as given by Bokovski and Vitek relies on declarations and operates on source code. Your task is to **infer** anonymous methods and confined types from the bytecode.

This has several consequences. Some of the constraints given by Bokovski and Vitek are either not applicable to bytecode or can be improved. Infering anonymity also allows more methods to be confined as a revised definition of anonymity may not require anonymity to be preserved in all subtypes.

It is part of the assignment to design an improved constraint system.

Your code must compile using `make` in the `ovm/tools/conf` directory. You may modify the makefile if needed. The TA will invoke your program using the **`run.sh`** script provided.

## Help!

You are encouraged to discuss definitions and share test-cases on the mailinglist. If you have problems with the OVM framework, contact `ovm@cs.purdue.edu`.

In the tar-file on the homepage you will find documentation on the verification process under documentation/ovm/ (type make in that directory to invoke latex).

## Where are the files?

| Class | Package | Description |
|---|---|---|
| Domain | ovm.domain | current context |
| Policy | ovm.domain | external part of the verifier |
| Repository | ovm.domain.repository | shared constants (Strings!) |
| OVMClass | ovm.domain.repository | class file |
| OVMSelector | ovm.domain.repository | method signature |
| ReferenceType | ovm.orep | type representation |
| Rewriter | ovm.tools.rewriter | Verification main |
| Instruction | ovm.tools.rewriter.instrutions | abstract instruction object |
| Stack | ovm.tools.rewriter.instructions | abstract VM state |
| StackElement | ovm.tools.rewriter.instructions | component of Stack |
| Driver | ovm.tools.conf | your main object |