Java Bytecode Specification and Verification

Lilian Burdy
INRIA Sophia-Antipolis
Lilian.Burdy@sophia.inria.fr

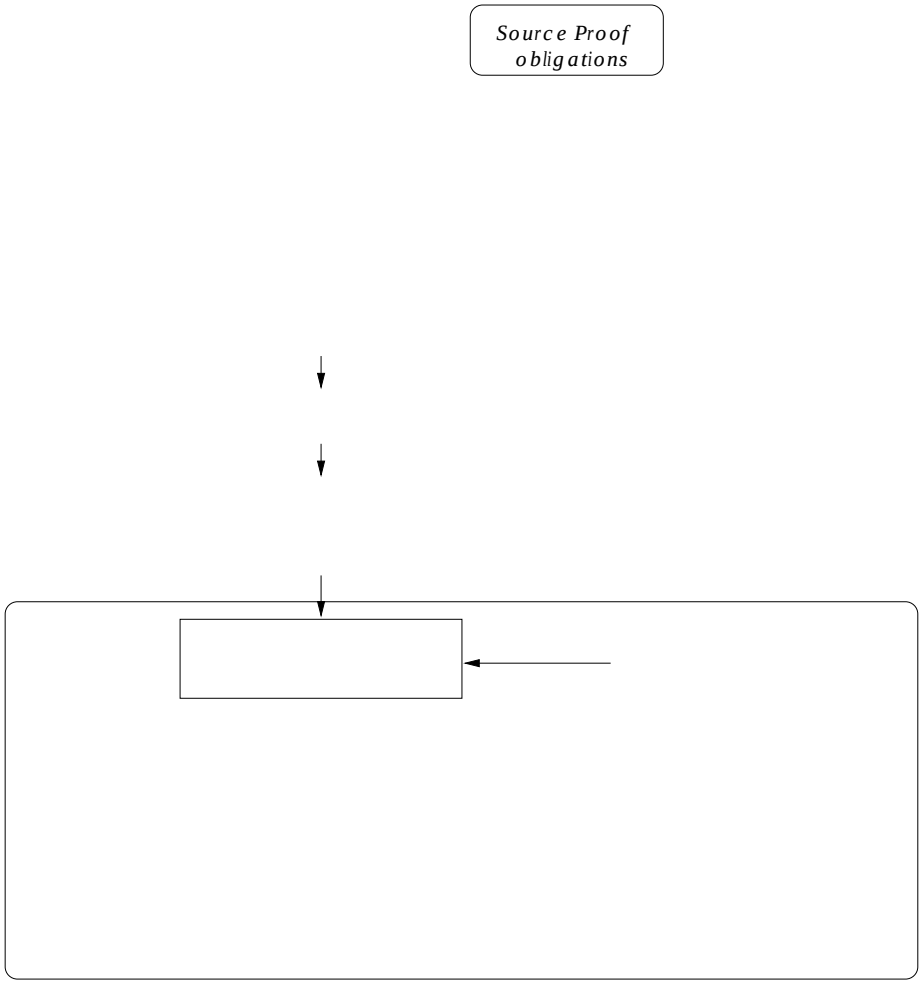
Mariela Pavlova
INRIA Sophia-Antipolis
Mariela.Pavlova@sophia.inria.fr

September 1, 2005

Abstract

We propose a framework for establishing the correctness of untrusted Java bytecode components w.r.t. to complex functional and/or security policies. To

untrusted code is accompanied by a proof for its safety w.r.t. to some safety property and the code receiver has just to generate the (prop)j 19.78989 0Td (the)Tj5.013.5956d (y)Tj -p1.8685


Source Proof
obligations

we introduce the BCSL language, the JML compiler and the bytecode weakest precondition calculus which underlines the bytecode veri cation condition generator.

3 Related Work

We now review works which treat very similar problematic.

The JVer tool [8] is a similar tool for verifying that downloaded Java bytecode programs do not abuse client computational

```
public class ListArray {
  Object[]
```

de ned attributes in the class le. For example, the speci cation of all the loops in a method are compiled to a unique method attribute: whose syntax is given in aFig4.

5.1

6 Comparison between source and bytecodes proofs

The purpose of this section is to

Hypothesis on bytecode:	Hypothesis on source level:
lv[2]_at_ins_20 len(#19(lv[0]))	i_at_ins_26 len(ListArray:l

References

[1] A. V. Aho, R. Sethi, and J. D. Ullman. Compilers-Principles, T[1] e[2]

[15] G. C. Necula and P. Lee. The design and implementation of