

Java Bytecode Specification and Verification

ABSTRACT

We propose a framework for establishing the correctness of Java bytecode. The framework is based on a formal semantics of Java bytecode and a set of verification conditions. The framework is implemented in the form of a tool that takes a Java bytecode file as input and produces a verification report. The tool is available for download at <http://www.javasoft.com/~javatech/java/bytecode/>.

Java
Weakest Precondition
Calculus

JML source identifiers are linked with their identifiers on bytecode level, namely with the corresponding indexes either from the constant pool or the array of

5. WEAKEST

$$\begin{aligned}
 & \text{wp}(\text{invoke } m; \text{ } ; \text{ }^{\text{exc}}) = \\
 & \quad \text{pre}(m) \wedge \\
 & \quad \bigwedge_{j=1::s} e_j : \quad \text{post}(m) \quad [! \vee [i] \quad \text{st}(c+i - \text{nArg}(m) \text{ })]_{i=0}^{\text{nArg}(m)} \\
 & \quad \quad \quad [n\text{result} \quad \text{fresh_var}] \\
 & \quad) \\
 & \quad [c \quad c \quad \text{nArg}(m)] \\
 & \quad [\text{st}(c) \quad \text{fresh_var}] \\
 & \quad \wedge k
 \end{aligned}$$

Hypothesis on bytecode:	Hypothesis on source level:
-------------------------	-----------------------------

