



UNIVERSITÄT ZU LÜBECK
INSTITUTE FOR IT SECURITY

SWAT: Improvements for SV-Comp 2026

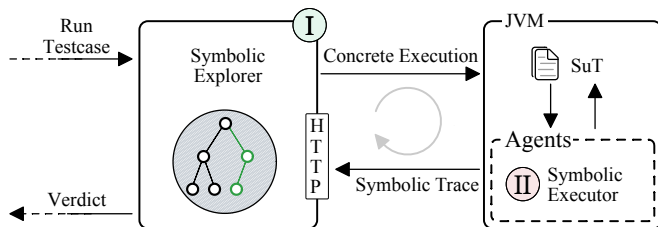
SV-Comp 2026, *Jan-Niclas Serr*

Nils Loose, Florian Sieck , Felix Mächtle , and Thomas Eisenbarth

April 13, 2026

Institute for IT Security

SWAT is a modular and instrumentation-based dynamic symbolic execution (DSE) framework for Java programs running on the standard JVM.



- **Symbolic Executor:** Java Agent that instruments target classes at load time and generates symbolic traces during execution.
- **Symbolic Explorer:** Decoupled component that solves path constraints and guides exploration through a dynamically assembled execution tree.

Contributions for SV-Comp 2026

1. Symbolic runtime exception modelling (`no-runtime-exception.prp`)
2. Migration to the theory of fixed-size bit-vectors
3. 124 new symbolic models for the Java standard library

Approach: Treat each potentially-throwing instruction as a branching instruction and symbolically model the condition under which the exceptional path is taken.

Example — IALOAD (load integer from array): May throw an `ArrayIndexOutOfBoundsException`. For symbolic index i and symbolic array length ℓ , we generate the exceptional path condition

$$\varphi_{exc} = (i < 0) \vee (i \geq \ell)$$

Its negation $\neg\varphi_{exc}$ guards the normal execution path.

Previously: SWAT relied on the *unbounded theory of arithmetic* for symbolic reasoning on all integral primitive types.

- Required explicit bound handling for fixed-width Java types.
- Limited support for bit-based operations.
- Complicated datatype translation between integer and floating-point primitives.

For SV-Comp 2026: Migration of the symbolic executor to the *theory of fixed-size bit-vectors*.

Contribution 3: New Symbolic Models

SWAT's instrumentation scope is configurable and (usually) targets the namespace of the system under analysis.

- A narrow scope enables more efficient execution.
- But it requires *symbolic models* for invocations of uninstrumented methods, including those from the Java Standard Library.

Motivation: The Java track invokes 425 unique method signatures from the Java Standard Library. This year's new `autostub` verification tasks explicitly stress a verifier's capabilities on library methods.

Contribution: 124 new symbolic models added, covering:

- The wrapper classes of all primitive types.
- Various commonly used utility classes.

Try SWAT!

Any Questions?



n.loose@uni-luebeck.de



github.com/SWAT-project/SWAT



Institute for IT Security, Uni Lübeck



GitHub