

Google &  
Download  
Tools

# Combining Static and Dynamic Analysis for Bug Finding

Christoph Csallner

Professor Yannis Smaragdakis

18 October 2007

*Goal: Automatic bug  
finding tool without false  
bug warnings*


# Combining Static and Dynamic Analysis for Bug Finding



Program Analysis for mass-market  
(+) **Fully automated**

# Combining Static and Dynamic Analysis for Bug Finding

```
/**  
 * @param p should be 1 or greater  
 */
```



Analyse program and specification  
(-) Hard: spec mostly informal

- (+) Abstract to analyze all paths
- (-) Even impossible ones



## Combining **Static** and Dynamic Analysis for Bug Finding

Model with superset of possible paths

→ Example: compiler

Warn about “bug” on impossible path

→ False bug warning

- (+) Run only possible paths
- (-) Cannot run all paths



## Combining Static and **Dynamic** Analysis for Bug Finding

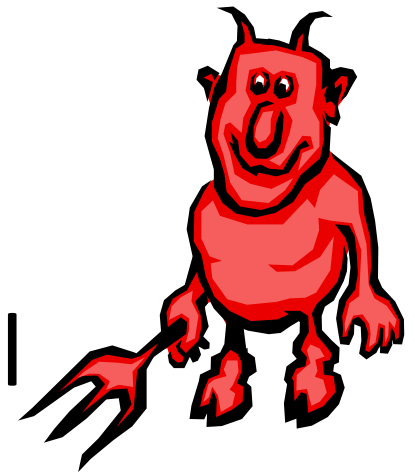
Real bug on missed path  
→ Missed bug



# Combining Static and Dynamic Analysis for Bug Finding



People have limited time  
→ False bug warnings are evil



# Many bug warnings, lot of noise

```
groovy.gdo.DataSet: add(java.util.Map) ...
groovy.gdo.DataSet.java:107: Warning: Possible type cast error (Cast)
Map Entry entry = (Map.Entry) iter.next();
-----
Execution trace information:
Reached top of loop after 0 iterations in "groovy.gdo.DataSet.java", line 106, col 8.
-----
groovy.gdo.GroovyResultSet: java:129: Warning: Possible type cast error (Cast)
Map Entry entry = (Map.Entry) iter.next();
-----
Execution trace information:
Reached top of loop after 0 iterations in "groovy.gdo.DataSet.java", line 106, col 8.
Reached top of loop after 0 iterations in "groovy.gdo.GroovyResultSet.java", line 128, col 12.
-----
[0.139 + 3602952 bytes] failed
groovy.gdo.GroovyResultSet: add(java.util.Map) ...
-----
groovy.gdo.GroovyResultSet.java:110: Warning: Possible type cast error (Cast)
Map Entry entry = (Map.Entry) iter.next();
-----
Execution trace information:
Reached top of loop after 0 iterations in "groovy.gdo.GroovyResultSet.java", line 109, col 8.
-----
[0.22 + 3830536 bytes] failed
groovy.lang.GString: getValues(int) ...
-----
groovy.lang.GString.java:87: Warning: Possible negative array index (IndexNegative)
return value[i];
-----
groovy.lang.GString.java:87: Warning: Array index possibly too large (IndexTooBig)
return value[i];
-----
[0.085 + 3867968 bytes] failed
groovy.lang.MetaClass: getMethod(java.lang.String) ...
-----
groovy.lang.MetaClass.java:152: Warning: Possible type cast error (Cast)
List answer = (List) methodIndex.getName();
-----
[0.811 + 48215192 bytes] failed
groovy.lang.MetaClass: getStaticMethods(java.lang.String) ...
-----
groovy.lang.MetaClass.java:163: Warning: Possible type cast error (Cast)
List answer = (List) staticMethodIndex.getName();
-----
[0.733 + 47945920 bytes] failed
groovy.lang.MetaClass: invokeMethod(java.lang.Object, java.lang.String, java.lang.Object) ...
-----
groovy.lang.MetaClass.java:219: Warning: Possible type cast error (Cast)
Method method = (Method) chooseMethod(methodName, methods, ...
-----
Execution trace information:
Executed else branch in "groovy.lang.MetaClass.java", line 213, col 8.
Executed then branch in "groovy.lang.MetaClass.java", line 218, col 32.
-----
groovy.lang.MetaClass.java:236: Warning: Possible type cast error (Cast)
method = (Method) chooseMethod(methodName, newStaticInstan
-----
Execution trace information:
Executed else branch in "groovy.lang.MetaClass.java", line 213, col 8.
Executed then branch in "groovy.lang.MetaClass.java", line 218, col 32.
Executed else branch in "groovy.lang.MetaClass.java", line 220, col 12.
Executed then branch in "groovy.lang.MetaClass.java", line 227, col 41.
Executed then branch in "groovy.lang.MetaClass.java", line 230, col 22.
Executed then branch in "groovy.lang.MetaClass.java", line 233, col 49.
-----
[3.095 + 54201552 bytes] failed
groovy.lang.MetaClass: invokeStaticMethod(java.lang.Object, java.lang.String, java.lang.Object) ...
-----
groovy.lang.MetaClass.java:269: Warning: Possible type cast error (Cast)
Method method = (Method) chooseMethod(methodName, methods, ...
-----
Execution trace information:
Executed then branch in "groovy.lang.MetaClass.java", line 268, col 32.
-----
[0.955 + 5382472 bytes] failed
groovy.lang.MetaClass: invokeConstructor(java.lang.Object) ...
-----
groovy.lang.MetaClass.java:287: Warning: Possible type cast error (Cast)
Constructor constructor = (Constructor) chooseMethod("cinv", ...
-----
[0.353 + 55822768 bytes] failed
-----
groovy.lang.MetaClass: getNewStaticInstanceMethod(java.lang.String) ...
-----
groovy.lang.MetaClass.java:288: Warning: Possible type cast error (Cast)
List newStaticInstanceMethods = (List) newStaticInstanceMethod
-----
[0.685 + 55779728 bytes] failed
groovy.lang.MetaClass: getProperty(java.lang.Object, java.lang.String) ...
-----
groovy.lang.MetaClass.java:309: Warning: Possible type cast error (Cast)
PropertyDescriptor descriptor = (PropertyDescriptor) propertyD
-----
[0.797 + 5627944 bytes] failed
groovy.lang.MetaClass: setProperty(java.lang.Object, java.lang.String, java.lang.Object) ...
-----
groovy.lang.MetaClass.java:357: Warning: Possible type cast error (Cast)
PropertyDescriptor descriptor = (PropertyDescriptor) propertyD
-----
groovy.lang.MetaClass.java:390: Warning: Possible type cast error (Cast)
Method addListenerMethod = (Method) listeners.getProperty();
-----
Execution trace information:
Executed else branch in "groovy.lang.MetaClass.java", line 359, col 8.
-----
groovy.lang.MetaClass.java:393: Warning: Array index possibly too large (IndexTooBig)
return listenerMethod = (Method) listeners.revisi
-----
Execution trace information:
Executed else branch in "groovy.lang.MetaClass.java", line 359, col 8.
Executed then branch in "groovy.lang.MetaClass.java", line 371, col 70.
-----
[3.515 + 57829496 bytes] failed
groovy.lang.MetaClassRegistry: getMetaClass(java.lang.Class) ...
-----
groovy.lang.MetaClassRegistry.java:86: Warning: Possible type cast error (Cast)
MetaClass answer = (MetaClass) metaClasses.get(theClass);
-----
[0.889 + 6876152 bytes] failed
groovy.lang.Range: subList(int, int) ...
-----
groovy.lang.Range.java:133: Warning: Possible type cast error (Cast)
return new Range((Comparable) getFromIndex(), (Comparable) get
-----
Execution trace information:
Executed else branch in "groovy.lang.Range.java", line 124, col 8.
Executed else branch in "groovy.lang.Range.java", line 127, col 8.
Executed else branch in "groovy.lang.Range.java", line 130, col 8.
-----
groovy.lang.Range.java:133: Warning: Possible type cast error (Cast)
return new Range((Comparable) getFromIndex(), (Comparable) get
-----
Execution trace information:
Executed else branch in "groovy.lang.Range.java", line 124, col 8.
Executed else branch in "groovy.lang.Range.java", line 127, col 8.
Executed else branch in "groovy.lang.Range.java", line 130, col 8.
-----
[0.356 + 69376168 bytes] failed
groovy.lang.Range: step(int, groovy.lang.Closure) ...
-----
groovy.lang.Range.java:154: Warning: Possible type cast error (Cast)
value = (Comparable) increment(value);
-----
Execution trace information:
Executed then branch in "groovy.lang.Range.java", line 149, col 23.
Reached top of loop after 0 iterations in "groovy.lang.Range.java", line 151, col 22.
Reached top of loop after 0 iterations in "groovy.lang.Range.java", line 153, col 35.
-----
groovy.lang.Range.java:164: Warning: Possible type cast error (Cast)
value = (Comparable) decrement(value);
-----
Execution trace information:
Executed else branch in "groovy.lang.Range.java", line 158, col 13.
Reached top of loop after 0 iterations in "groovy.lang.Range.java", line 161, col 12.
Reached top of loop after 0 iterations in "groovy.lang.Range.java", line 163, col 16.
-----
[0.318 + 69947400 bytes] failed
groovy.lang.Tuple: get(int) ...
-----
groovy.lang.Tuple.java:70: Warning: Possible negative array index (IndexNegative)
return contents[index];
-----
groovy.lang.Tuple.java:70: Warning: Array index possibly too large (IndexTooBig)
return contents[index];
-----
[0.111 + 7147344 bytes] failed
-----
[17.081 + 73462240 bytes] failed
-----
groovy.lang.Tuple: subList(int, int) ...
-----
groovy.lang.Tuple.java:113: Warning: Possible attempt to allocate array of negative length (Negative)
Object[] newContent = new Object[size];
-----
[0.075 + 73827664 bytes] failed
groovy.model.DefaultTableModel: getValues(int, int) ...
-----
groovy.model.DefaultTableModel.java:164: Warning: Possible type cast error (Cast)
DefaultTableModel column = (DefaultTableModel) columnModel.g
-----
Execution trace information:
Executed else branch in "groovy.model.DefaultTableModel.java", line 156, col 8.
Executed else branch in "groovy.model.DefaultTableModel.java", line 159, col 8.
-----
[0.152 + 73922424 bytes] failed
groovy.model.DefaultTableModel: setValueAt(java.lang.Object, int, int) ...
-----
groovy.model.DefaultTableModel.java:181: Warning: Possible type cast error (Cast)
DefaultTableModel column = (DefaultTableModel) columnModel.g
-----
Execution trace information:
Executed else branch in "groovy.model.DefaultTableModel.java", line 173, col 8.
Executed else branch in "groovy.model.DefaultTableModel.java", line 176, col 8.
-----
[0.162 + 7092888 bytes] failed
groovy.servlet.GroovyServlet: service(java.servlet.ServletRequest, java.servlet.ServletResponse) ...
-----
groovy.servlet.GroovyServlet.java:97: Warning: Possible type cast error (Cast)
HttpServletRequest httpRequest = (HttpServletRequest) ...
-----
groovy.servlet.GroovyServlet.java:98: Warning: Possible type cast error (Cast)
HttpServletResponse httpResponse = (HttpServletResponse) ...
-----
[1.678 + 77210128 bytes] failed
groovy.swing.html.TableLayoutRow: start() ...
-----
groovy.swing.html.TableLayoutRow.java:82: Warning: Possible type cast error (Cast)
TableLayoutCell cell = (TableLayoutCell) iter.next();
-----
Execution trace information:
Reached top of loop after 0 iterations in "groovy.swing.html.TableLayoutRow.java", line 81, col 8.
-----
[0.153 + 64710336 bytes] failed
groovy.util.GroovyMBean: invokeMethod(java.lang.String, java.lang.Object) ...
-----
groovy.util.GroovyMBean.java:110: Warning: Possible type cast error (Cast)
String[] signature = (String[]) operations.getMethod();
-----
[0.367 + 65075024 bytes] failed
groovy.util.Node: breadthFirst() ...
-----
groovy.util.Node.java:241: Warning: Possible type cast error (Cast)
Node childNode = (Node) iter.next();
-----
Execution trace information:
Reached top of loop after 0 iterations in "groovy.util.Node.java", line 232, col 8.
Reached top of loop after 0 iterations in "groovy.util.Node.java", line 240, col 8.
-----
[1.181 + 68470320 bytes] failed
groovy.util.OrderBy: compare(java.lang.Object, java.lang.Object) ...
-----
groovy.util.OrderBy.java:85: Warning: Possible type cast error (Cast)
Closure closure = (Closure) iter.next();
-----
Execution trace information:
Reached top of loop after 0 iterations in "groovy.util.OrderBy.java", line 84, col 8.
-----
[0.247 + 69643024 bytes] failed
groovy.util.XmlParser: endElement(java.lang.String, java.lang.String, java.lang.String) ...
-----
groovy.util.XmlParser.java:199: Warning: Possible type cast error (Cast)
parent = (Node) stack.get(stack.size() - 1);
-----
Execution trace information:
Executed then branch in "groovy.util.XmlParser.java", line 189, col 28.
Executed then branch in "groovy.util.XmlParser.java", line 192, col 31.
Executed then branch in "groovy.util.XmlParser.java", line 196, col 30.
Executed then branch in "groovy.util.XmlParser.java", line 198, col 34.
-----
groovy.util.XmlParser.java:202: Warning: Possible type cast error (Cast)
bodyText = (StringBuffer) bodyTexts.remove(bodyTexts.size) - ...
-----
Execution trace information:
Executed then branch in "groovy.util.XmlParser.java", line 189, col 28.
Executed then branch in "groovy.util.XmlParser.java", line 192, col 31.
Executed then branch in "groovy.util.XmlParser.java", line 196, col 30.
Executed then branch in "groovy.util.XmlParser.java", line 198, col 34.
-----
[17.081 + 73462240 bytes] failed
-----
org.codehaus.groovy.ast.stmt.BlockStatement: visit(org.codehaus.groovy.ast.GroovyCodeVisitor) ...
-----
org.codehaus.groovy.ast.stmt.BlockStatement.java:74: Warning: Possible type cast error (Cast)
Statement statement = (Statement) iter.next();
-----
Execution trace information:
Reached top of loop after 0 iterations in "org.codehaus.groovy.ast.stmt.BlockStatement.java", line 73, col 8.
-----
[0.189 + 74074352 bytes] failed
org.codehaus.groovy.ast.stmt.BlockStatement: getText() ...
-----
org.codehaus.groovy.ast.stmt.BlockStatement.java:105: Warning: Possible type cast error (Cast)
Statement statement = (Statement) iter.next();
-----
Execution trace information:
Reached top of loop after 0 iterations in "org.codehaus.groovy.ast.stmt.BlockStatement.java", line 99, col 23.
-----
[0.563 + 78370992 bytes] failed
org.codehaus.groovy.ast.stmt.SwitchStatement: getCaseStatements(int) ...
-----
org.codehaus.groovy.ast.stmt.SwitchStatement.java:105: Warning: Possible type cast error (Cast)
return (CaseStatement) caseStatements.get(id);
-----
Execution trace information:
Executed then branch in "org.codehaus.groovy.ast.stmt.SwitchStatement.java", line 104, col 53.
-----
[0.138 + 74215312 bytes] failed
org.codehaus.groovy.ast.stmt.TryCatchStatement: getCatchStatements(int) ...
-----
org.codehaus.groovy.ast.stmt.TryCatchStatement.java:96: Warning: Possible type cast error (Cast)
return (CatchStatement) catchStatements.get(id);
-----
Execution trace information:
Executed then branch in "org.codehaus.groovy.ast.stmt.TryCatchStatement.java", line 95, col 54.
-----
[0.184 + 73937376 bytes] failed
org.codehaus.groovy.classgen.Verifier: visitMethod(org.codehaus.groovy.ast.MethodNode) ...
-----
org.codehaus.groovy.classgen.Verifier.java:202: Warning: Possible type cast error (Cast)
Statement last = (Statement) list.get(id);
-----
Execution trace information:
Executed then branch in "org.codehaus.groovy.classgen.Verifier.java", line 189, col 34.
Executed else branch in "org.codehaus.groovy.classgen.Verifier.java", line 195, col 17.
Executed then branch in "org.codehaus.groovy.classgen.Verifier.java", line 195, col 58.
Executed then branch in "org.codehaus.groovy.classgen.Verifier.java", line 200, col 37.
-----
[1.006 + 79397184 bytes] failed
org.codehaus.groovy.runtime.ClassExtender: call(java.lang.String, java.lang.Object) ...
-----
org.codehaus.groovy.runtime.ClassExtender.java:88: Warning: Possible type cast error (Cast)
closure = (Closure) methods.getName();
-----
Execution trace information:
Executed then branch in "org.codehaus.groovy.runtime.ClassExtender.java", line 87, col 33.
-----
[1.116 + 79320160 bytes] failed
org.codehaus.groovy.runtime.DefaultGroovyMethods: query(java.sql.Connection, groovy.lang.Closure) ...
-----
org.codehaus.groovy.runtime.DefaultGroovyMethods.java:1075: Warning: Array index possibly too large (IndexTooBig)
StringBuffer buffer = new StringBuffer(text());
-----
Execution trace information:
Executed else branch in "org.codehaus.groovy.runtime.DefaultGroovyMethods.java", line 1074, col 31.
-----
[2.292 + 88370992 bytes] failed
org.codehaus.groovy.runtime.Invoker: asList(java.lang.Object) ...
-----
org.codehaus.groovy.runtime.Invoker.java:174: Warning: Possible type cast error (Cast)
return Arrays.asList(Object) value);
-----
Execution trace information:
Executed else branch in "org.codehaus.groovy.runtime.Invoker.java", line 170, col 13.
Executed else branch in "org.codehaus.groovy.runtime.Invoker.java", line 173, col 13.
Executed then branch in "org.codehaus.groovy.runtime.Invoker.java", line 173, col 45.
-----
[1.127 + 86713152 bytes] failed
-----
org.codehaus.groovy.runtime.Invoker: asCollection(java.lang.Object) ...
-----
org.codehaus.groovy.runtime.Invoker.java:205: Warning: Possible type cast error (Cast)
return Arrays.asList(Object) value);
-----
Execution trace information:
Executed else branch in "org.codehaus.groovy/runtime/Invoker.java", line 197, col 11.
Executed else branch in "org.codehaus.groovy/runtime/Invoker.java", line 200, col 11.
Executed then branch in "org.codehaus.groovy/runtime/Invoker.java", line 204, col 45.
-----
[0.261 + 87486472 bytes] failed
org.codehaus.groovy.runtime.Invoker: toString(java.lang.Object) ...
-----
org.codehaus.groovy/runtime/Invoker: toString(java.lang.Object) ...
-----
Execution trace information:
Executed else branch in "org.codehaus.groovy/runtime/Invoker.java", line 290, col 43.
Executed then branch in "org.codehaus.groovy/runtime/Invoker.java", line 292, col 11.
Reached top of loop after 0 iterations in "org.codehaus.groovy/runtime/Invoker.java", line 320, col 12.
Executed then branch in "org.codehaus.groovy/runtime/Invoker.java", line 320, col 43.
-----
org.codehaus.groovy.runtime.Invoker: invoke(java.lang.Object, Object[] array = (Object[]) arguments;
-----
Execution trace information:
Executed else branch in "org.codehaus.groovy/runtime/Invoker.java", line 290, col 13.
Executed then branch in "org.codehaus.groovy/runtime/Invoker.java", line 290, col 48.
-----
org.codehaus.groovy.runtime.Invoker: invoke(java.lang.Object, Map Entry entry = (Map.Entry) iter.next());
-----
Execution trace information:
Executed else branch in "org.codehaus.groovy/runtime/Invoker.java", line 290, col 13.
Executed else branch in "org.codehaus.groovy/runtime/Invoker.java", line 290, col 48.
-----
org.codehaus.groovy.runtime.Invoker: invoke(java.lang.Object, Map Entry entry = (Map.Entry) iter.next());
-----
Execution trace information:
Executed else branch in "org.codehaus.groovy/runtime/Invoker.java", line 290, col 13.
Executed else branch in "org.codehaus.groovy/runtime/Invoker.java", line 302, col 11.
Executed else branch in "org.codehaus.groovy/runtime/Invoker.java", line 318, col 11.
Reached top of loop after 0 iterations in "org.codehaus.groovy/runtime/Invoker.java", line 320, col 12.
Executed then branch in "org.codehaus.groovy/runtime/Invoker.java", line 320, col 47.
-----
[4.249 + 90848488 bytes] failed
org.codehaus.groovy.runtime.InvokerHelper: createMap(java.lang.Object) ...
-----
org.codehaus.groovy.runtime.InvokerHelper.java:246: Warning: Array index possibly too large (IndexTooBig)
answer.put(value[i+1], value[i+1]);
-----
Execution trace information:
Reached top of loop after 0 iterations in "org.codehaus.groovy/runtime/InvokerHelper.java", line 248, col 8.
-----
[0.558 + 96199576 bytes] failed
org.codehaus.groovy.runtime.InvokerHelper: createRange(java.lang.Object, java.lang.Object) ...
-----
org.codehaus.groovy.runtime.InvokerHelper.java:258: Warning: Possible type cast error (Cast)
return new Range((Comparable) from, (Comparable) to);
-----
Execution trace information:
Short circuited boolean operation in "org.codehaus.groovy/runtime/InvokerHelper.java", line 255, col 36.
Executed else branch in "org.codehaus.groovy/runtime/InvokerHelper.java", line 255, col 8.
-----
org.codehaus.groovy/runtime/InvokerHelper: invoke(java.lang.Object, Comparable) ...
-----
Execution trace information:
Reached top of loop after 0 iterations in "org.codehaus.groovy/runtime/InvokerHelper.java", line 255, col 8.
-----
[1.141 + 9071904 bytes] failed
org.codehaus.groovy.runtime.InvokerHelper: createScript(java.lang.Class, groovy.lang.ScriptContext) ...
-----
org.codehaus.groovy/runtime/InvokerHelper.java:283: Warning: Possible type cast error (Cast)
return (Script) constructor.newInstance(Object) ctor ...
-----
[0.38 + 93533120 bytes] failed
org.codehaus.groovy.syntax.Lexer: AbstractCharStream: b() ...
-----
org.codehaus.groovy.syntax.Lexer: AbstractCharStream.java:52: Warning: Possible division by zero (ZeroDiv)
pos % buf.length;
-----
Execution trace information:
Executed else branch in "org.codehaus.groovy/syntax/lexer/AbstractCharStream.java", line 45, col 8.
-----
[0.102 + 96514328 bytes] failed
org.codehaus.groovy.syntax.Lexer: AbstractCharStream: consume() ...
-----
org.codehaus.groovy/syntax/lexer/AbstractCharStream.java:73: Warning: Possible division by zero (ZeroDiv)
this.cur % buf.length;
-----
[1.285 + 92189184 bytes] failed
[1.406 + 92189184 bytes] total
51 warnings
runtime = 128 seconds.
```



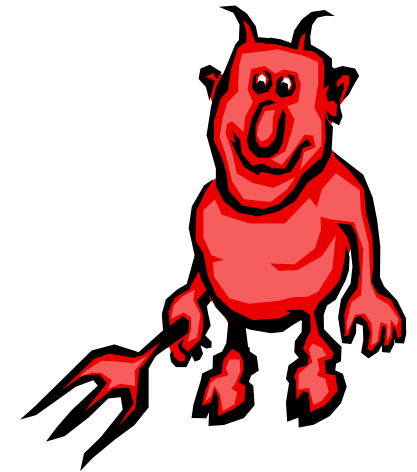


# False bug warnings: Show-stopper

- Flanagan et al. (ESC/Java people), 2002:
  - “[T]he tool has not reached the desired level of cost effectiveness. In particular, users complain about an annotation burden that is perceived to be heavy, and about **excessive warnings about non-bugs**, particularly on unannotated or partially-annotated programs.”
- Rutar et al., 2004:
  - > 9k NPE warnings in 170k non commented source stmt
  - “[..] **too many warnings to be easily useful by themselves.**”

# Two kinds of false bug warnings

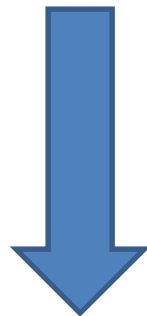
- *Language-level*, e.g.: Java semantics
  - E.g.: `if (0==1) {infeasible};`
  - Behavior cannot occur
  - Never ever



- *User-level*
  - E.g.: passing -1 into `m(int p) //p pos!`
  - Behavior can occur, but user does not care
  - We cannot read minds, but worth trying



*Goal: Automatic bug  
finding tool without false  
bug warnings*



Huge, unsolved problem  
(+) Commercial interest



# Sound(bug-finder) = Complete(correctness-prover)

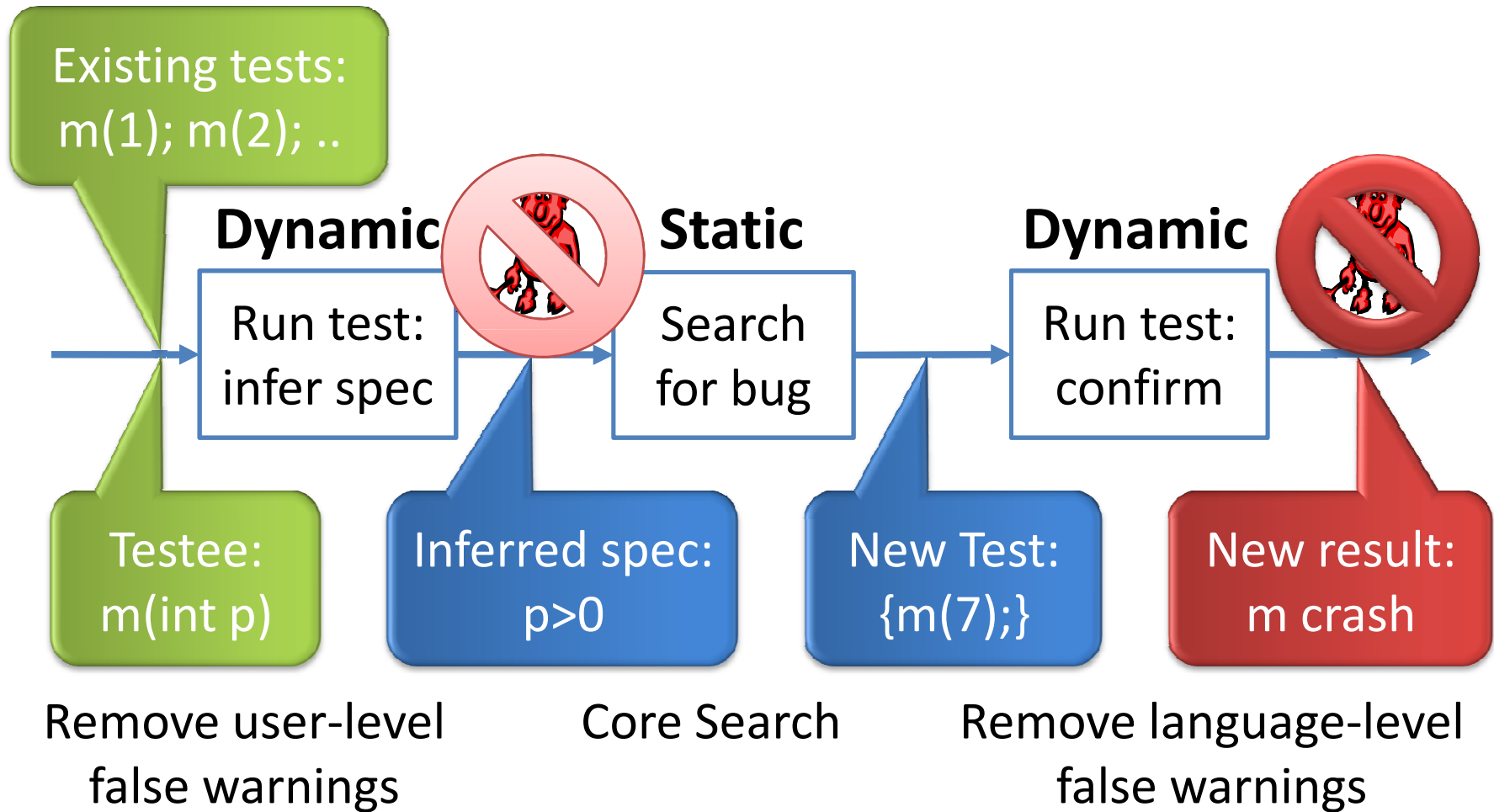
- *Sound*:  $\text{claim}(p) \Rightarrow p$ 
  - Bug finding: **No false bug warnings**
  - Prove correctness: **Find all bugs**
- *Complete*:  $p \Rightarrow \text{claim}(p)$ 
  - Bug finding: **Find all bugs**
  - Prove correctness: **No false bug warnings**
- Sound(bug-finding) = Complete(prove correctness)
- Sound(prove correctness) = Complete(bug-finding)



# Our tool-chain: DSD-Crasher

ISSTA 2006 paper: "DSD-Crasher" [distinguished paper award]

(ACM SIGSOFT International Symposium on Software Testing and Analysis)



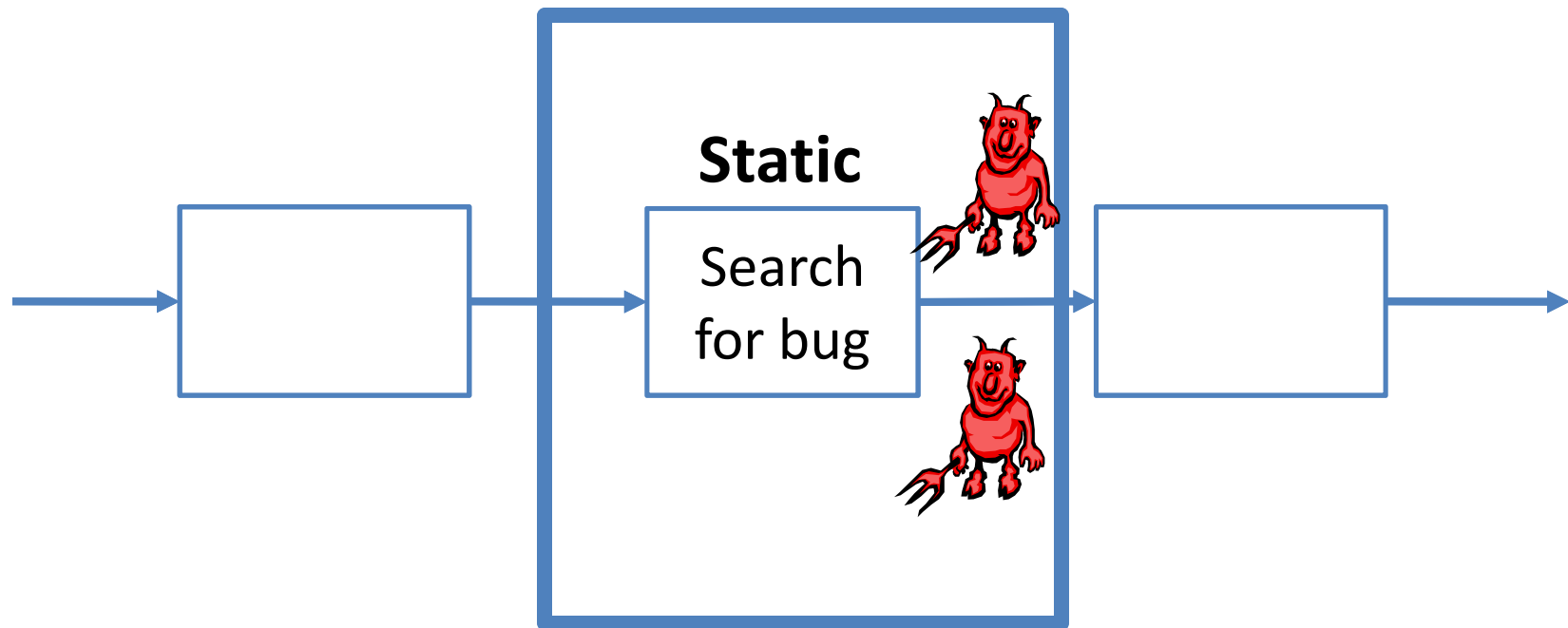
D-S-D

Christoph Csallner (12)

# ESC/Java: Static Analysis

## Core bug search component

Cormac Flanagan et al. (Compaq)



Core Search

# ESC/Java: Static Analysis

## Core bug search component

- Checks for potential invariant violations
  - User-defined in JML [Gary T. Leavens]
  - **Pre**conditions, **Post**conditions, class invariants
- Knows pre and post of primitive language ops
  - Pointer dereference, class cast, array access, etc.
- Checks each method in isolation
  - Call → check pre, assume post
  - Method body → its weakest pre

# ESC under the hood

- Weakest pre `wp (method body, true)`
  - States from which execution terminates normally
  - Remaining states lead execution to an error:  
Violate some pre or post
- We are interested in those that throw a runtime exception
  - Dereferencing null, Illegal type cast, Illegal array allocation or access, Division by zero
- ESC uses Simplify to derive abstract error



# Expected static analysis problem: False bug warnings

- Example:
- `public int get10() {return 10;}`
- `public int meth(int p) {  
    return p / get10();  
}`
- ESC warns of a division by zero in meth
  - Can never occur!

# Less expected: Output cryptic

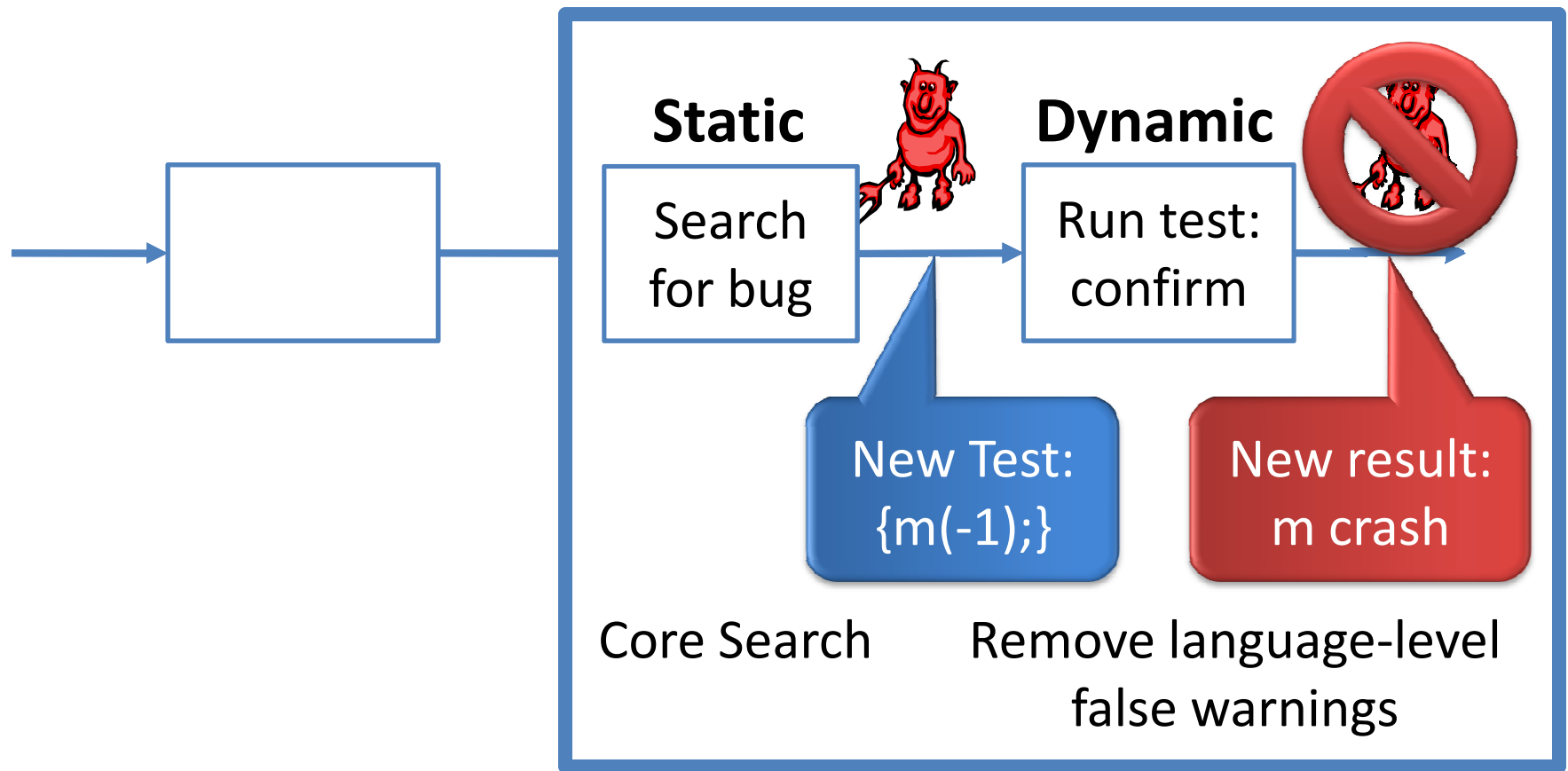
```
(arrayLength(firstArray:294.43) <= intLast)
(firstArray:294.43 < longFirst)
(tmp0!new!double[:297.27 < longFirst)
(secondArray:295.43 < longFirst)
(longFirst < intFirst)
(vAllocTime(tmp0!new!double[:297.27) < alloc<1>)
(0 < arrayLength(firstArray:294.43))
(null <= max(LS))
(vAllocTime(firstArray:294.43) < alloc)
(alloc <= vAllocTime(tmp0!new!double[:297.27))
(eClosedTime(elems@loopold) < alloc)
(vAllocTime(out:6..) < alloc)
(vAllocTime(secondArray:295.43) < alloc)
(intLast < longLast)
(1000001 <= intLast)
((intFirst + 1000001) <= 0)
(out:6.. < longFirst)
arrayLength(tmp0!new!double[:297.27) ==
  arrayLength(firstArray:294.43)
typeof(0) <: T_int
null.LS == @true
typeof(firstArray:294.43[0]) <: T_double
isNewArray(tmp0!new!double[:297.27) == @true
typeof(arrayLength(firstArray:294.43)) <: T_int
T_double[] <: arrayType
typeof(firstArray:294.43) == T_double[]
T_double[] <: T_double[]
elemtype(T_double[]) == T_double
T_double <: T_double
typeof(secondArray:295.43) <: T_double[]
typeof(secondArray:295.43) == T_double[]
arrayFresh(tmp0!new!double[:297.27 alloc alloc<1>
  elems@loopold
```

```
arrayShapeOne(arrayLength(firstArray:294.43))
  T_double[] F_0.0)
typeof(firstArray:294.43) <: T_double[]
typeof(tmp0!new!double[:297.27) == T_double[]
(null <= max(LS))
tmp0!new!double[:297.27[0] == firstArray:294.43[0]
arrayLength(secondArray:295.43) == 0
elems@loopold == elems
elems@pre == elems
state@pre == state
state@loopold == state
m@loopold:299.12 == 0
out@pre:6.. == out:6..
EC@loopold == EC
alloc@pre == alloc
ltypeof(out:6..) <: T_float
ltypeof(out:6..) <: T_byte
ltypeof(secondArray:295.43) <: T_boolean
ltypeof(firstArray:294.43) <: T_double
ltypeof(secondArray:295.43) <: T_short
ltypeof(tmp0!new!double[:297.27) <: T_boolean
ltypeof(tmp0!new!double[:297.27) <: T_short
ltypeof(firstArray:294.43) <: T_boolean
ltypeof(out:6..) <: T_char
ltypeof(secondArray:295.43) <: T_double
ltypeof(firstArray:294.43) <: T_float
ltypeof(out:6..) <: T_int
lisAllocated(tmp0!new!double[:297.27 alloc)
ltypeof(secondArray:295.43) <: T_long
ltypeof(firstArray:294.43) <: T_char
ltypeof(tmp0!new!double[:297.27) <: T_double
ltypeof(tmp0!new!double[:297.27) <: T_long
```

```
T_double[] != T_boolean
T_double[] != T_char
T_double[] != T_byte
T_double[] != T_long
T_double[] != T_short
T_double[] != T_int
T_double[] != T_float
ltypeof(firstArray:294.43) <: T_byte
ltypeof(out:6..) <: T_boolean
ltypeof(secondArray:295.43) <: T_float
ltypeof(out:6..) <: T_short
ltypeof(secondArray:295.43) <: T_byte
ltypeof(tmp0!new!double[:297.27) <: T_float
ltypeof(firstArray:294.43) <: T_long
ltypeof(tmp0!new!double[:297.27) <: T_byte
ltypeof(out:6..) <: T_double
ltypeof(firstArray:294.43) <: T_short
ltypeof(out:6..) <: T_long
typeof(out:6..) != T_boolean
typeof(out:6..) != T_char
typeof(out:6..) != T_byte
typeof(out:6..) != T_long
typeof(out:6..) != T_short
typeof(out:6..) != T_int
typeof(out:6..) != T_float
ltypeof(secondArray:295.43) <: T_char
ltypeof(secondArray:295.43) <: T_int
ltypeof(tmp0!new!double[:297.27) <: T_char
ltypeof(firstArray:294.43) <: T_int
ltypeof(tmp0!new!double[:297.27) <: T_int
bool$false != @true
secondArray:295.43 != null
firstArray:294.43 != null
T_double != typeof(out:6..)
T_double != T_double[]
tmp0!new!double[:297.27 != null
```

# Check 'n' Crash: Language-level sound bug warnings

ICSE 2005 paper: ``Check 'n' Crash''  
(ACM/IEEE International Conference on Software Engineering)



# Finding test cases that satisfy ESC's error conditions

- ESC uses Simplify automated theorem prover
  - Limitations: multiplication, float, large numbers, ...
- We use POOC int solver [Schlenker et al.]
- float and double: currently ignored
- Object reference: Keep equivalence relation
- Array: Reference to a special object
  - Int constraints on array length and indices
- Rest: JCrasher random test case generator

# Filling in random values: JCrasher

Software, Practice & Experience 2004 paper: "JCrasher"

- Predefined values, e.g.:

```
int ← 0
```

```
int ← 1
```

```
A ← null
```

- Crawl testee for method signatures:

```
A ← (B, int) //from method A B.m(int)
```

- Chain discovered rules recursively:

```
– (new B()) .m(0)
```

```
– (new B()) .m(1)
```

```
– ...
```

D-S-D

# Example: Student homework

- ```
public static void swapArrays
    (double[] fstArray, double[] sndArray) { //..
    for(int m=0; m<fstArray.length; m++) { //..
        fstArray[m]=sndArray[m]; /*..*/ } }
```

- Informal spec:

- Swap `fstArray` and `sndArray` elements
- If array lengths differ, just return

- Corresponding ESC/Java warning:

```
Array index possibly too large (IndexTooBig)
    fstArray[m]=sndArray[m];
                        ^
```

# Example: ESC's counterexample

```
(arrayLength(firstArray:294.43) <= intLast)
(firstArray:294.43 < longFirst)
(tmp0!new!double[:297.27 < longFirst)
(secondArray:295.43 < longFirst)
(longFirst < intFirst)
(vAllocTime(tmp0!new!double[:297.27) < alloc<1>)
(0 < arrayLength(firstArray:294.43))
(null <= max(LS))
(vAllocTime(firstArray:294.43) < alloc)
(alloc <= vAllocTime(tmp0!new!double[:297.27))
(eClosedTime(elems@loopold) < alloc)
(vAllocTime(out:6..) < alloc)
(vAllocTime(secondArray:295.43) < alloc)
(intLast < longLast)
(1000001 <= intLast)
((intFirst + 1000001) <= 0)
(out:6.. < longFirst)
arrayLength(tmp0!new!double[:297.27) ==
  arrayLength(firstArray:294.43)
typeof(0) <: T_int
null.LS == @true
typeof(firstArray:294.43[0]) <: T_double
isNewArray(tmp0!new!double[:297.27) == @true
typeof(arrayLength(firstArray:294.43)) <: T_int
T_double[] <: arrayType
typeof(firstArray:294.43) == T_double[]
T_double[] <: T_double[]
elemtype(T_double[]) == T_double
T_double <: T_double
typeof(secondArray:295.43) <: T_double[]
typeof(secondArray:295.43) == T_double[]
arrayFresh(tmp0!new!double[:297.27 alloc alloc<1>
  elems@loopold
```

D-S-D

```
arrayShapeOne(arrayLength(firstArray:294.43))
  T_double[] F_0.0)
typeof(firstArray:294.43) <: T_double[]
typeof(tmp0!new!double[:297.27) == T_double[]
(null <= max(LS))
tmp0!new!double[:297.27[0] == firstArray:294.43[0]
arrayLength(secondArray:295.43) == 0
elems@loopold == elems
elems@pre == elems
state@pre == state
state@loopold == state
m@loopold:299.12 == 0
out@pre:6.. == out:6..
EC@loopold == EC
alloc@pre == alloc
!typeof(out:6..) <: T_float
!typeof(out:6..) <: T_byte
!typeof(secondArray:295.43) <: T_boolean
!typeof(firstArray:294.43) <: T_double
!typeof(secondArray:295.43) <: T_short
!typeof(tmp0!new!double[:297.27) <: T_boolean
!typeof(tmp0!new!double[:297.27) <: T_short
!typeof(firstArray:294.43) <: T_boolean
!typeof(out:6..) <: T_char
!typeof(secondArray:295.43) <: T_double
!typeof(firstArray:294.43) <: T_float
!typeof(out:6..) <: T_int
!isAllocated(tmp0!new!double[:297.27 alloc)
!typeof(secondArray:295.43) <: T_long
!typeof(firstArray:294.43) <: T_char
!typeof(tmp0!new!double[:297.27) <: T_double
!typeof(tmp0!new!double[:297.27) <: T_long
```

Christoph Csallner (22)

```
T_double[] != T_boolean
T_double[] != T_char
T_double[] != T_byte
T_double[] != T_long
T_double[] != T_short
T_double[] != T_int
T_double[] != T_float
!typeof(firstArray:294.43) <: T_byte
!typeof(out:6..) <: T_boolean
!typeof(secondArray:295.43) <: T_float
!typeof(out:6..) <: T_short
!typeof(secondArray:295.43) <: T_byte
!typeof(tmp0!new!double[:297.27) <: T_float
!typeof(firstArray:294.43) <: T_long
!typeof(tmp0!new!double[:297.27) <: T_byte
!typeof(out:6..) <: T_double
!typeof(firstArray:294.43) <: T_short
!typeof(out:6..) <: T_long
typeof(out:6..) != T_boolean
typeof(out:6..) != T_char
typeof(out:6..) != T_byte
typeof(out:6..) != T_long
typeof(out:6..) != T_short
typeof(out:6..) != T_int
typeof(out:6..) != T_float
!typeof(secondArray:295.43) <: T_char
!typeof(secondArray:295.43) <: T_int
!typeof(tmp0!new!double[:297.27) <: T_char
!typeof(firstArray:294.43) <: T_int
!typeof(tmp0!new!double[:297.27) <: T_int
bool$false != @true
secondArray:295.43 != null
firstArray:294.43 != null
T_double != typeof(out:6..)
T_double != T_double[]
tmp0!new!double[:297.27 != null
```



# Process ESC's counterexample

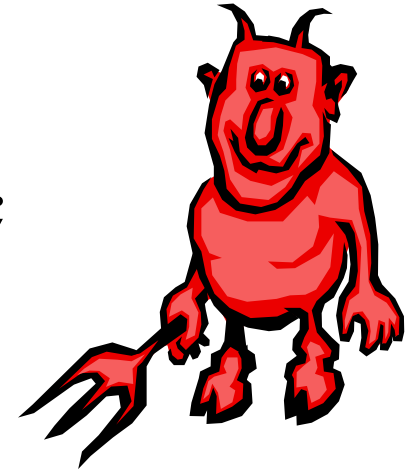
- Feed relevant constraints to constraint solvers
- Compile solutions into a JUnit test case:
- ```
public void test0() throws Throwable {  
    try {  
        double[] d1 = new double[]{1.0};  
        double[] d3 = new double[]{};  
        P1s1.swapArrays(d1, d3);  
    } catch (Exception e) {dispatchException(e);}  
}
```
- Observe test case execution
  - Use JCrasher's extensions of JUnit runtime



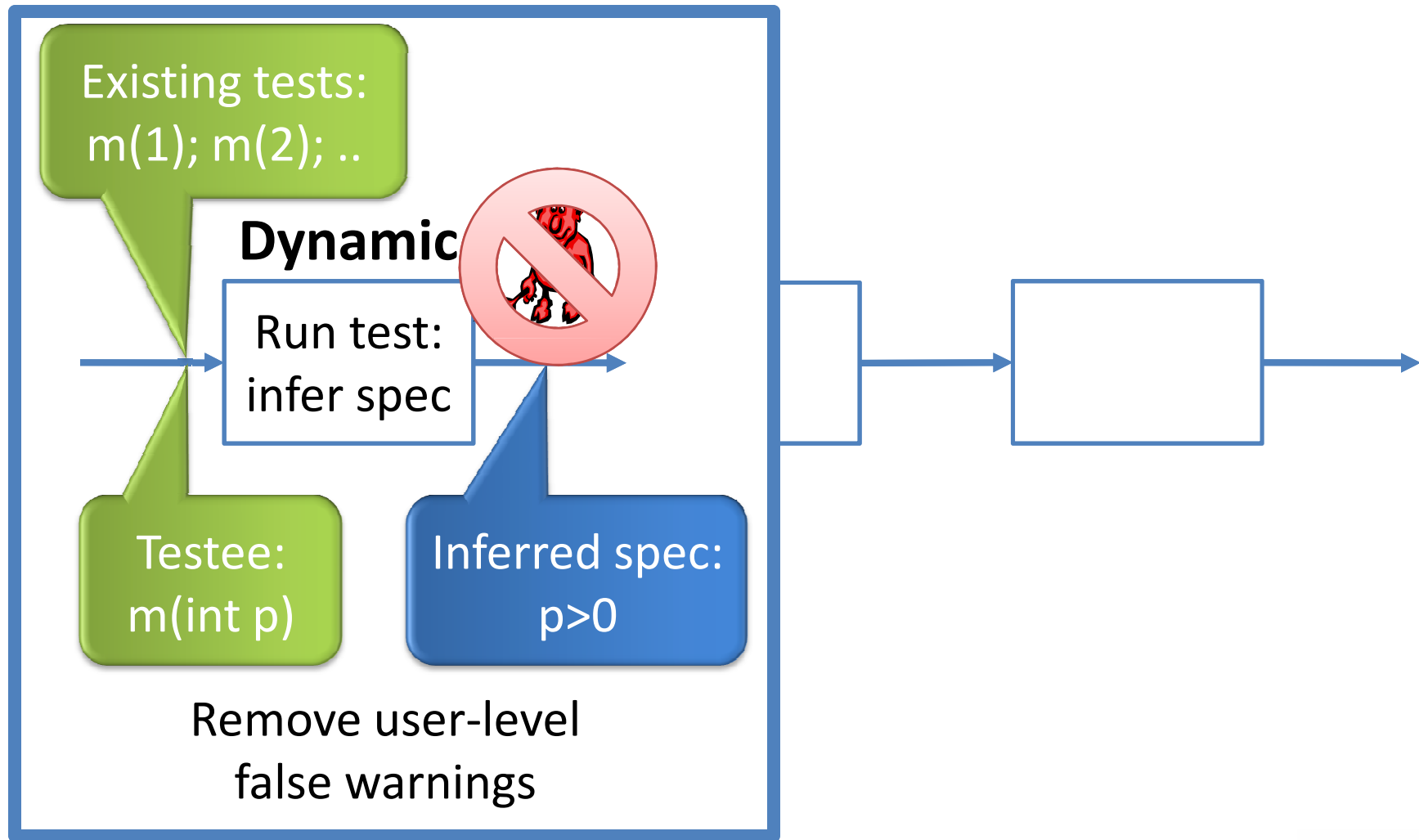


# Two kinds of false bug warnings

- *Language-level*, e.g.: Java semantics
  - E.g.: `if (false) {infeasible};`
  - Behavior cannot occur
  - Never ever
- *User-level*
  - E.g.: passing -1 into `m(int p) //p pos!`
  - Behavior can occur, but user does not care
  - We cannot read minds, but worth trying



# Front-end: User-level soundness



# Daikon: Infer invariants

Michael Ernst et al. (MIT)

1. Instrument testee bytecode, execute
  - Trace variable values at method entry & exit
2. Analyze each entry/exit value set
  - Instantiate invariant templates with values
  - Invariant invalidated by sample → Drop invariant
  - Invariant held for some samples, never invalidated → Assume: true invariant
3. Annotate testee's source code (with JML):
  - Preconditions, postconditions, class invariants

# Daikon: Configuration

- Concentrate on simple invariants, e.g.
  - `intVariable {=,≥,>, ≤,<} {intConstant, intVariable}`
- Ignore complex invariants, benefit unclear:
  - variable is one of `{const1, const2, ..}`
  - Elements of container structures
  - `float, double, String`

# Experiments Overview

- ESC/Java produces many false bug warnings
- Check 'n' Crash on its own discovers real bugs
  - In JBoss JMS, JABA, Groovy, student homeworks
- DSD-Crasher improves soundness
  - Fewer false bug reports
- DSD-Crasher improves on other tools with less sophisticated static analysis
  - Finds more bugs

# Improvement over Check 'n' Crash: Groovy experiments

- Use test suite that comes with Groovy

Tool	Runtime [min:s]	Exception reports
Check 'n' Crash	10:43	19
DSD-Crasher	30:32	11

- Using Daikon-inferred invariants
  - 12..18: ESC could statically **rule out false positives**
  - 19: ESC produces more complicated error condition, threw off constraint solver

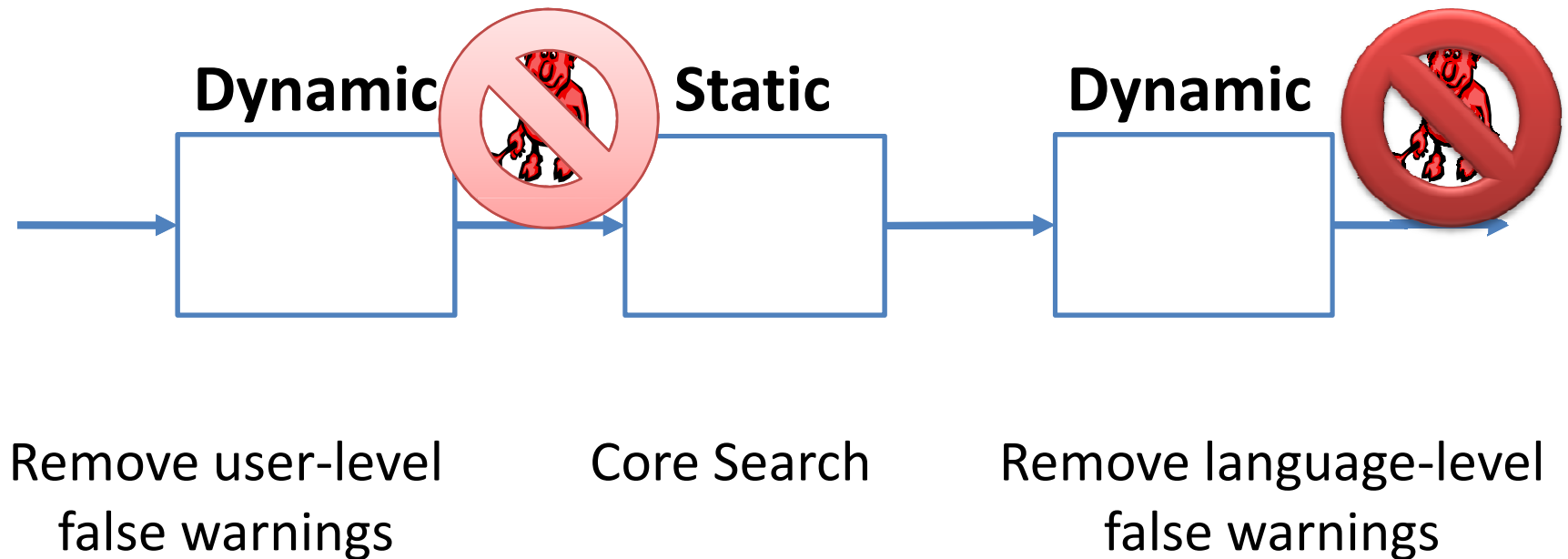
# Example improvement over Check 'n' Crash

- JBoss JMS: Check 'n' Crash warns of NegativeArraySizeException
- `setBytes(byte[] value, int length) //simplified`

```
{  
    byte[] bytes = new byte[length]; //..  
}
```
- Test case calls `setBytes` three times
- Daikon infers precondition:  
`length == daikon.Quant.size(value)`
- DSD-Crasher suppresses false positive

# Eclat: DSD combination with less static analysis

Carlos Pacheco et al. (MIT)





# DSD-Crasher: Deeper than Eclat

- From Groovy experiments
- DSD-Crasher benefits from deeper static analysis
- Eclat does more than we look for

Tool (timeout)	Exception reports	Runtime [min:s]
Eclat-default	0	7:01
Eclat-hybrid, 4 rounds	0	8:24
Eclat-exh, 2 rounds	2	10:02
Eclat-exh (500 s)	2	16:42
Eclat-exh (1200 s)	2	33:17
<b>DSD-Crasher</b>	<b>4</b>	<b>30:32</b>

(Eclat-exh = Eclat's exhaustive search)

# DSD-Crasher: Example Improvement Over Eclat

- Eclat is a Daikon+test generation combination
  - Static analysis mostly random
  - Does not find following bug for any settings
- JBoss JMS example
  - `byte[] getBytes(Object value) //simplified`

```
{  
    if (value == null) {return null;}  
    else if (value instanceof Byte[]) {  
        return (byte[]) value;  
    } //..  
}
```

# Selected related bug-finding work

- Dynamic (JCrasher)
  - Boyapati et al. (ISSTA '02), Pacheco et al. (ICSE '07)
- Static-Dynamic (Check 'n' Crash)
  - Beyer et al. (ICSE '04), Tomb et al. (ISSTA '07)
- Dynamic-Static-Dynamic (DSD-Crasher)
  - Xie et al. (ASE '03), Pacheco et al. (ECOOP '05)
- Concolic (Static and dynamic in parallel)
  - Godefroid et al. (PLDI '05), Sen et al. (FSE '05)

# Our tools are used

- Check 'n' Crash
  - Freie Universität Berlin: Lab course
  - North Carolina State University
    - Sarah Smith Heckman et al.: Rank output of bug finding tools (Short papers in ISSRE '05, ISSRE '06, ICSE '07)
  - UC Santa Cruz & RIACS/NASA Ames
    - Aaron Tomb et al.: Compare with advanced tool (ISSTA '07)
- JCrasher in teaching:
  - Drexel University: Homework assignment (SE320)
  - Universiteit Utrecht: Seminar (Program Verification)

# Examples of JCrasher in research

- University of Washington: David Notkin's group
  - Characterize test cases (ICFEM '04, ISSRE '05)
- North Carolina State University: Tao Xie's group
  - Minimize random tests (ASE '04)
  - Add assertions to random tests (ECOOP '06)
  - Web service robustness (ICWS '07)
- MIT: Michael Ernst's group
  - Compare with Eclat (ECOOP '05)
  - With Microsoft Research: Randoop comparison (ICSE '07)

# Our tool-chain: DSD-Crasher

## Dynamic-Static-Dynamic

