

Static Analysis for Java-like Programs

Sukyoung Ryu

Department of Computer Science
Korea Advanced Institute of Science and Technology

January 7, 2010

Static Program Analysis (at KAIST)

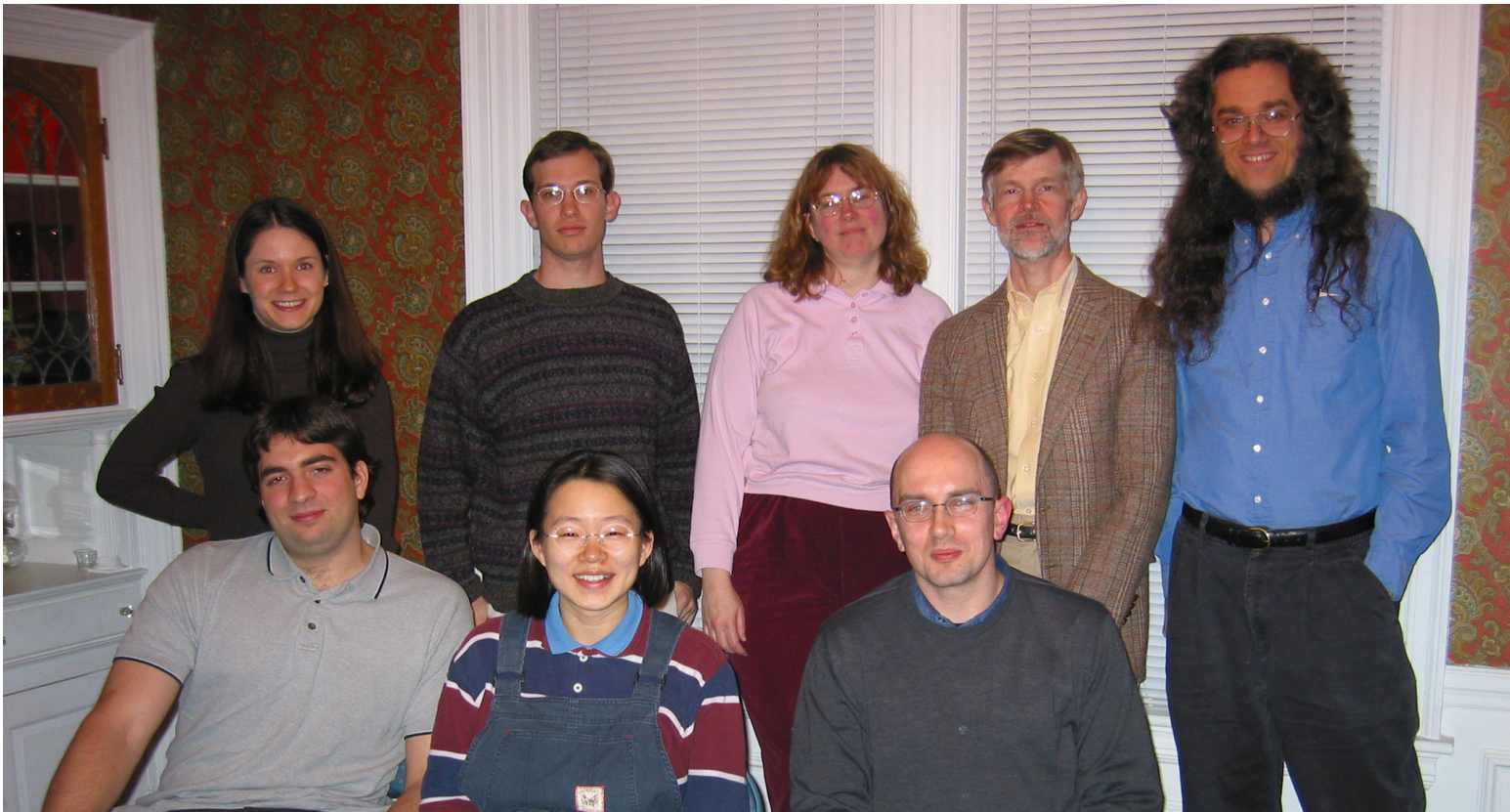


Exception Analyses

- Exception analysis for ML programs
 - > Exceptions and functions are intermingled.
 - > Decoupled two analyses:
control flow analysis and exception analysis
 - > Exception analyzer:
<http://cm.bell-labs.com/cm/cs/what/smlnj/links.html>
- Exception analysis for multithreaded Java programs
 - > concurrency analysis and exception analysis
- Rigorous, safe, and practical exception analyses
- A systematic development of an analysis

Debugging Everywhere

(at Harvard)

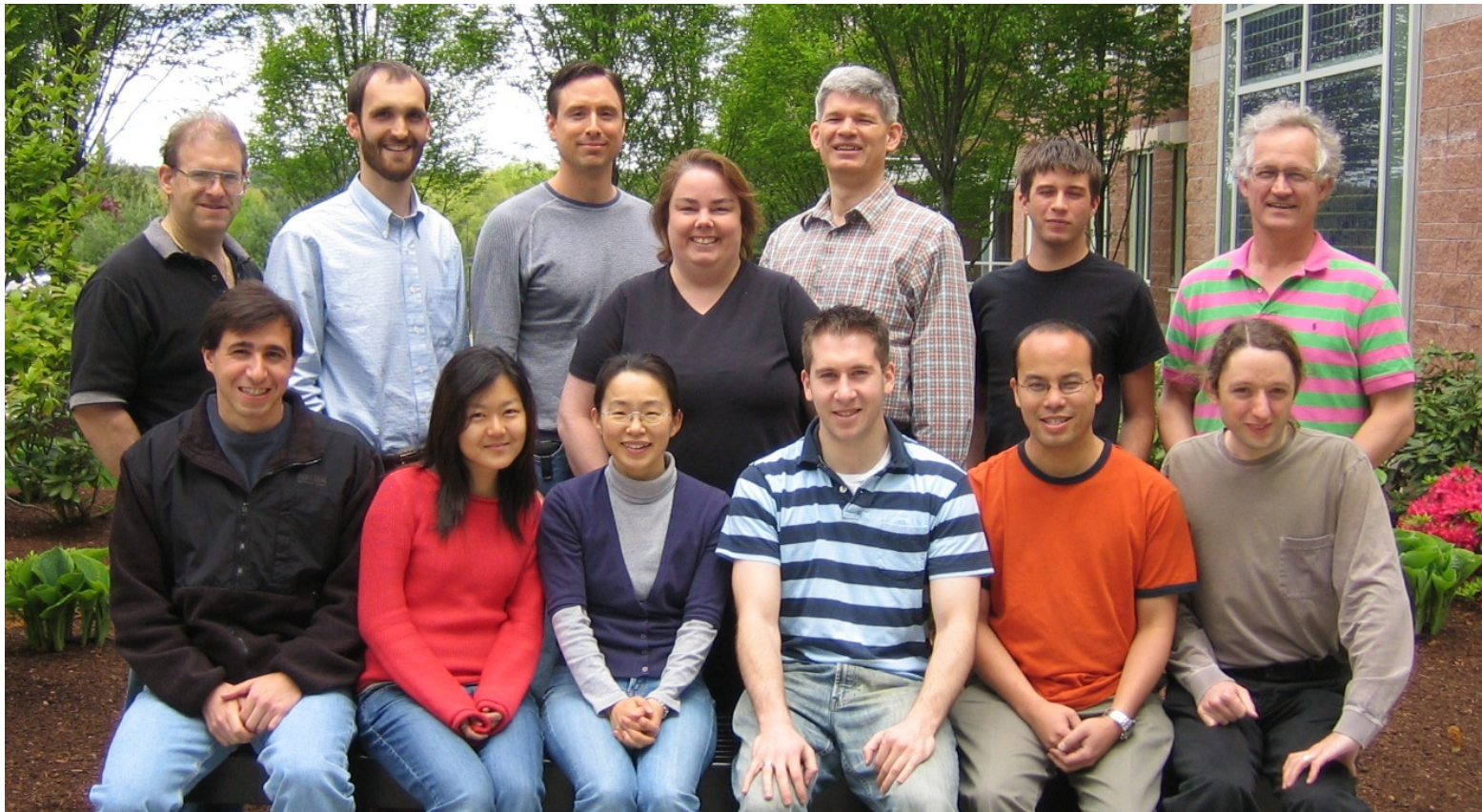


Source-Level Debugger ldb

- for multiple languages
C, Java, OCaml, and SML
- for multiple platforms
Sparc, Mips, and x86
- with modest programming effort
contract between a compiler and ldb
- without sacrificing runtime performance

<http://www.etaps05.inf.ed.ac.uk/Programme/CC.html>

Fortress Programming Language (at Sun Labs.)



Project Fortress

- A **multicore language** for scientists and engineers
- Run your **whiteboard in parallel!**

$$v_{\text{norm}} = \underline{\underline{v / \|v\|}}$$

$$\sum_{\underline{k \leftarrow 1:n}} \underline{a_k} \underline{x^k}$$

$$C = \underline{A \cup B}$$

$$y = \underline{\underline{3x \sin x}} \underline{\underline{\cos 2x}} \underline{\underline{\log \log x}}$$

- “Growing a Language”

Guy L. Steele Jr., keynote talk, OOPSLA 1998

Static Analysis for Java-like Programs

What to Not Expect

- Exhaustive survey of Java static analysis tools
- An apple to apple comparison of Java and C analysis tools
- Fortress sales

What Java™ Did for C

- Catch “stupid mistakes”: static type system
- Automatic storage management: garbage collection
- Platform independence: JVM
- Extensive libraries
- Security model, including type safety
- Dynamic compilation

Java-like Languages

- Scala <http://www.scala-lang.org>
- Fortress <http://projectfortress.sun.com>
- X10 <http://x10-lang.org>
- Clojure <http://clojure.org>
- Groovy <http://groovy.codehaus.org>
- JRuby <http://jruby.org>
- Jython <http://www.jython.org>
- ...

Languages on the JVM

- Scala <http://www.scala-lang.org>
- Fortress <http://projectfortress.sun.com>
- X10 <http://x10-lang.org>
- Clojure <http://clojure.org>
- Groovy <http://groovy.codehaus.org>
- JRuby <http://jruby.org>
- Jython <http://www.jython.org>
- ...

Why JVM for Other Languages

- Available for many hardware and software platforms
- Extremely high performance (especially HotSpot)
- Huge universe of Java libraries

Static Analysis Tools

- Commercial tools

Sparrow, CodeSonar, Coverity, KlocWork, PolySpace, Purify, Lint, PREFIX, PRefast, QAC, Safer C, GoAnna, Fortify, VeraCode, SLAM

- Open-source or noncommercial tools

FindBugsTM, clang, BLAST, Jlint, JPF, Splint, Calysto, Saturn, mygcc, ESC, LC-Lint, Vault, Astree, CGS, C-Kit, Uno, Orion

Static Analysis for C Programs

- Memory-related errors
 - > buffer overflow
 - > read outside array bounds
 - > memory leaks
 - > null pointer dereferences
- Compared to Java programs
 - > more bugs to find
 - > a lot scarier bugs
 - > not as good free tools

Static Analysis for Java Programs

- Violations of reasonable programming practices
 - > Shouldn't have infinite recursive loop.
 - > Shouldn't throw NullPointerException.
 - > All statements should be reachable.
 - > Shouldn't allow SQL injection.
- Compared to C programs
 - > static type system
 - > bytecode verifier
 - > good free tools, notably FindBugs

FindBugs

- An open-source static analysis tool
<http://findbugs.sourceforge.net>
- Analyzes classfiles; source files used only for display
- Looks for bug patterns, inspired by real problems in real code
- Built into the standard software development processes of Google and eBay

Kinds of Bugs

- Errors: Some things are always wrong.
 - > SQL injection
 - > infinite recursive loop
- Warnings: Some things are merely error prone.
 - > duplicate branches
 - > switch case falls through
- Guidelines: Some things are for code quality.
 - > confusing method name

What Matters

- At Google, null pointer exceptions aren't considered to be a serious problem in server code.
 - > But at eBay, they are.
- Both eBay and Google have developed their own prioritized lists of which issues they care about.
 - > They are significantly different.

FindBugs: Some Lessons

- Static analysis typically finds mistakes
 - > but some mistakes don't matter
 - > need to find important bugs.
- The bugs that *matter* depend on context.
- Concurrency is tricky.

FindBugs: Low-Hanging Fruits

- Some detectors are simple but specific: looking for ignored return values is easy.
- Some are harder: finding uses of `.equals` to compare two objects of different types (requires a type analysis.)
- FindBugs does lots of simple analyses, very little interprocedural code analysis.
- You don't have to be clever to find stupid mistakes; being stupid works pretty well.
- But clever can find more.

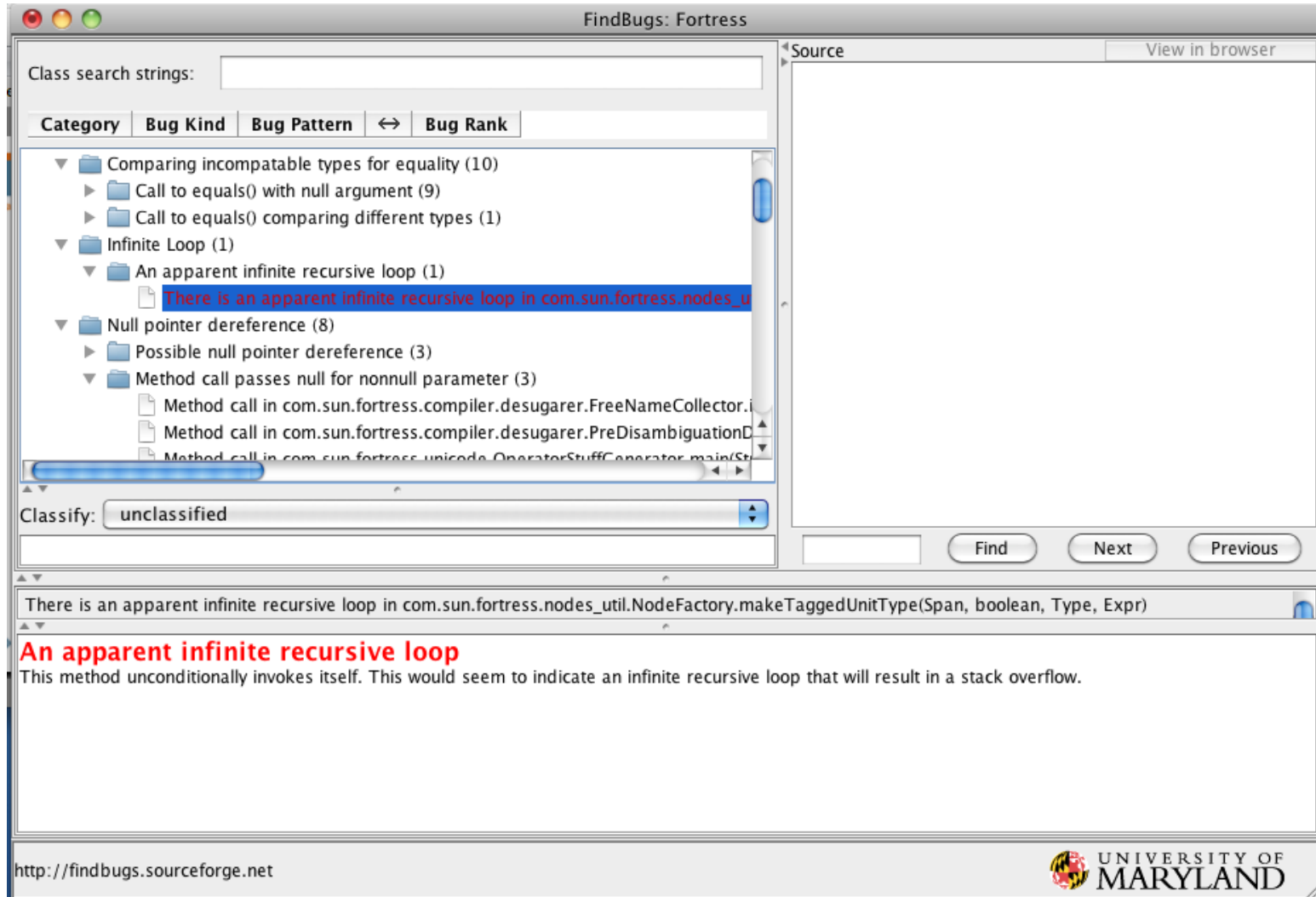
FindBugs: Bug Categories

- Correctness
- Bad practice
- Dodgy code
- Multithreaded correctness
- Potential performance problems
- Malicious code vulnerability
- Experimental
- Security
- Internationalization

FindBugs: Bug Categories for Fortress

- Correctness 45
- Bad practice 480
- Dodgy code 596
- Multithreaded correctness 15
- Potential performance problems 133
- Malicious code vulnerability 65
- Experimental 4
- Security
- Internationalization

FindBugs: Bug Categories for Fortress



The screenshot shows the FindBugs: Fortress application window. The main panel displays a tree view of bug categories:

- Comparing incompatible types for equality (10)
 - Call to equals() with null argument (9)
 - Call to equals() comparing different types (1)
- Infinite Loop (1)
 - An apparent infinite recursive loop (1)
 - There is an apparent infinite recursive loop in com.sun.fortress.nodes_u...
- Null pointer dereference (8)
 - Possible null pointer dereference (3)
 - Method call passes null for nonnull parameter (3)
 - Method call in com.sun.fortress.compiler.desugarer.FreeNameCollector.i...
 - Method call in com.sun.fortress.compiler.desugarer.PreDisambiguationD...
 - Method call in com.sun.fortress.unicode.OperatorStuffGenerator.main/St...

The 'Classify:' dropdown is set to 'unclassified'. At the bottom, a detailed view of the selected bug is shown:

There is an apparent infinite recursive loop in com.sun.fortress.nodes_util.NodeFactory.makeTaggedUnitType(Span, boolean, Type, Expr)

An apparent infinite recursive loop
 This method unconditionally invokes itself. This would seem to indicate an infinite recursive loop that will result in a stack overflow.

At the bottom left, the URL <http://findbugs.sourceforge.net> is displayed. At the bottom right, the University of Maryland logo is visible.

FindBugs: Correctness Bugs in Fortress

- Correctness 45
 - > Infinite recursive loop 1
 - > Bad casts of object references 2
 - * Impossible cast 1
 - * instanceof will always return false 1
 - > Bad use of return value from method 3
 - * Exception created and dropped rather than thrown 3
 - > Redundant comparison to null 9
 - * Nullcheck of value previously dereferenced 9
 - > ...

FindBugs: Bad Practice in Fortress

- Bad practice 480
 - > Bad use of return value from method 5
 - * Method ignores exceptional return value 5
 - > Null pointer dereference 47
 - * Method with Boolean return type returns explicit null 38
 - * equals() method does not check for null argument 9
 - > Checking String equality using == or != 5
 - * Comparison of String objects using == or != 5
 - > Dropped or ignored exceptions 5
 - * Method might ignore exceptions 5
 - > ...

FindBugs: Multithreaded Bugs

- Multithreaded bugs 15
 - > Constructor invokes `Thread.start()` 1
 - > Inconsistent synchronization 2
 - > Lock not released on all paths 3
 - * Method does not release lock on all exception paths 3
 - > Possible double check of field 5
 - > Static use of type `Calendar` or `DateFormat` 4
 - * Call to static `DateFormat` 2
 - * Static `DateFormat` 2

FindBugs: Performance Problems

- Performance problems 133
 - > Inner class could be made static 8
 - * Should be a static inner class 8
 - > Private method is never called 7
 - > Questionable Boxing of primitive value 12
 - * Method invokes inefficient Number constructor; use static valueOf instead 12
 - > String concatenation in loop using + operator 44
 - * Method concatenates strings using + in a loop 44
 - > Unread field 24
 - > ...

FindBugs: Bug Patterns

- Some big, broad and common patterns
 - > Dereferencing a null pointer
 - > An impossible checked cast
 - > Methods whose return value should not be ignored
- Lots of small, specific bug patterns, that together find lots of bugs
 - > Every **Programming Puzzler**
 - > Every chapter in **Effective Java**
 - > Most postings to <http://thedailywtf.com>

FindBugs: Analysis Techniques

- Local pattern matching
 - > If you invoke `String.toLowerCase()`, don't ignore the return value.
- Intraprocedural dataflow analysis
 - > Null pointer, type case errors
- Interprocedural method summaries
 - > This method always dereferences its parameter.
- Context-sensitive interprocedural analysis
 - > Interprocedural flow of untrusted data
 - * SQL injection, cross site scripting

FindBugs: More Bugs

- Where is the best place to expend effort to find more bugs?
 - > Use more sophisticated analysis to find more subtle errors
 - > Build more shallow and general bug detectors
 - > Write application-specific bug detectors

More Free Tools for Java Programs

- Checkstyle <http://checkstyle.sourceforge.net>
- PMD <http://pmd.sourceforge.net>
- Hammurapi <http://www.hammurapi.biz>
- Soot <http://www.sable.mcgill.ca/soot>
- Squale <http://www.squale.org>

Commercial Tools for Java Programs

- KlocWork <http://www.klocwork.com>
- Fortify Software SCA <http://www.fortify.com>
- Coverity Prevent <http://coverity.com>
- SureLogic Fluid <http://www.surelogic.com>
- Parasoft JTest <http://www.parasoft.com/jsp/home.jsp>

Static Analysis for Scala Programs

- More features while preserving backward compatibility for Java
 - > Type erasure semantics
- Issues to map the source-level new features down to JVM
 - > Compiling generics through user-directed type specialization [ICOOOLPS 2009](#)
 - > Implementing first-class polymorphic delimited continuations by a type-directed selective CPS-transform [ICFP 2009](#)

Static Analysis for Fortress Programs

- Mind-changing semantics
 - > Parallelism by default
 - > Advanced type system
- Issues to map the source-level new features down to JVM
 - > Encoding Fortress type system in Java bytecode
 - > Implementing checks for various static guarantees
- Issues to improve performance to take advantage of multicores
 - > purity analysis
 - > unboxed value analysis
 - > contention management for transactional memory

Sukyong Ryu

`sryu@cs.kaist.ac.kr`

`http://plrg.kaist.ac.kr`