

# Detecting argument selection defects

**Andrew Rice**<sup>\*</sup>, Eddie Aftandilian<sup>†</sup>, Ciera Jaspan<sup>†</sup>,

Emily Johnston<sup>†</sup>, Michael Pradel<sup>‡</sup>, and Yulissa Arroyo-Paredes<sup>§</sup>

<sup>\*</sup>University of Cambridge and Google, <sup>†</sup>Google, <sup>‡</sup>TU Darmstadt, <sup>§</sup>Barnard College of Columbia University

# What's probably wrong with this code?

```
Bitmap create(int width, int height) { ... }  
  
void doSomething(int width, int height) {  
    ...  
    Bitmap b = create(height, width);  
    ...  
}
```

# What's probably wrong with this code?

```
Bitmap create(int width, int height) { ... }  
  
void doSomething(int width, int height) {  
  
    ...  
  
    Bitmap b = create(width, height);  
  
    ...  
  
}
```

# Not all swaps are defects

```
Bitmap create(int width, int height) { ... }  
  
void doSomething(int width, int height) {  
    ...  
    if (rotate) { b = create(height, width); }  
    else { b = create(width, height); }  
    ...  
}
```

# We found instances in mature software projects

ConcurrentHashMap in OpenJDK ([JDK-8176402](#))

ASM MethodWriter ([ASM bug 317796](#))

ServerPreparedStatement in the MySQL JDBC driver  
([MySQL bug 85885](#))

SAXDocumentSerializer in OpenJDK ([JDK-8178411](#))

“Definitely embarrassing.”

-- Doug Lea, java.util.concurrent lead

# Building an argument selection defect checker for Google

# Parts of the algorithm

1. Extract names from expressions
2. Distance function
3. Decide when to suggest a different arrangement of arguments
4. Define heuristics

True positive rate

without heuristics: 10%

with heuristics: 85%

# Heuristic 1: Low information names

[a-z][a-z]?[0-9]\*

label

arg[0-9]

param[0-9]

value

str[0-9]

key



## Heuristic 2: Duplicate call

```
if (rotated) {  
    i = new Bitmap(height, width);  
}  
else {  
    i = new Bitmap(width, height);  
}
```

```
int something(int x, int y) {  
    if (x < y) {  
        return something(y, x);  
    }  
    ...  
}
```

# Heuristic 3: Enclosed by reverse

```
void reverse() {  
    return from(end,start);  
}
```

backwards?  
complement  
endian  
flip  
invert|se  
landscape|portrait  
opposite  
reciprocal  
reversed?  
rotat(e|ed|ion)?  
swap(ped)?  
transposed?  
undo

## Heuristic 4: Comment on argument

```
target(/*first = */second, /*second = */first);
```

```
target(second /*first*/, first /*second*/);
```

```
target(second, // first  
         first); // second
```

# Results

# Deployment at Google

Implemented a static check in Error Prone [Aftandilian 2012],  
Google's open-source static analysis tool for Java

Ran checker over checked-in code to find existing instances

Integrated checker into Google's code review system using  
Tricorder [Sadowski 2015]

# Analysis of checked-in code

Ran checker over large Java codebases

- 200 MLoC Google-authored
- 10 MLoC non-Google-authored

Found a total of 2,305 true positives/bugs with loose thresholds

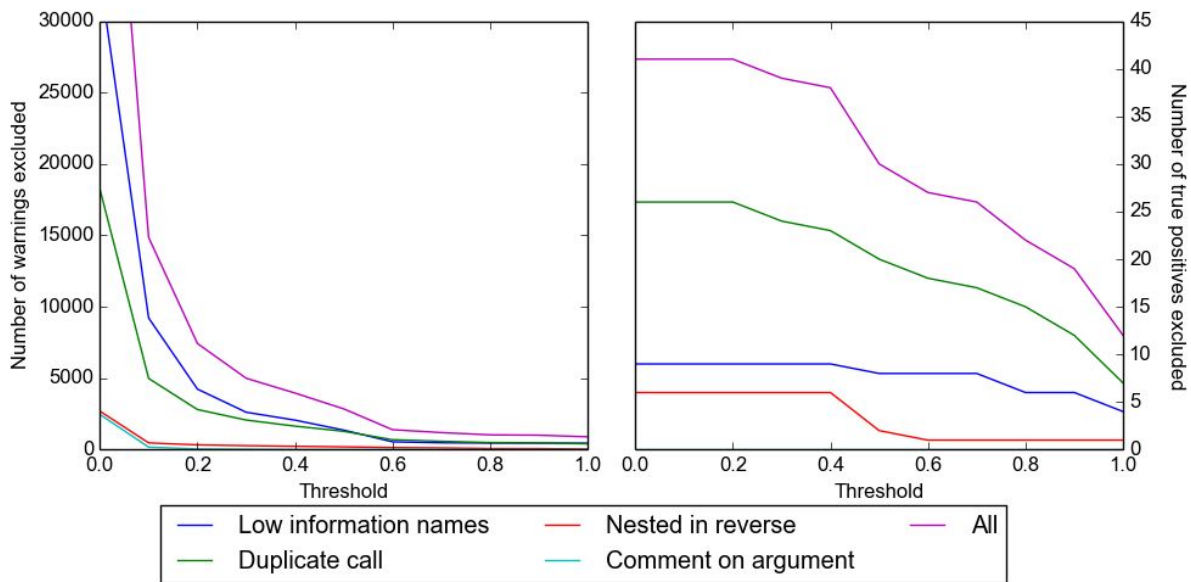
Set thresholds to minimize false positives

- 459 true positives, 78 false → 85% true positive rate

Many bugs found were not very impactful (“survivor effect”):

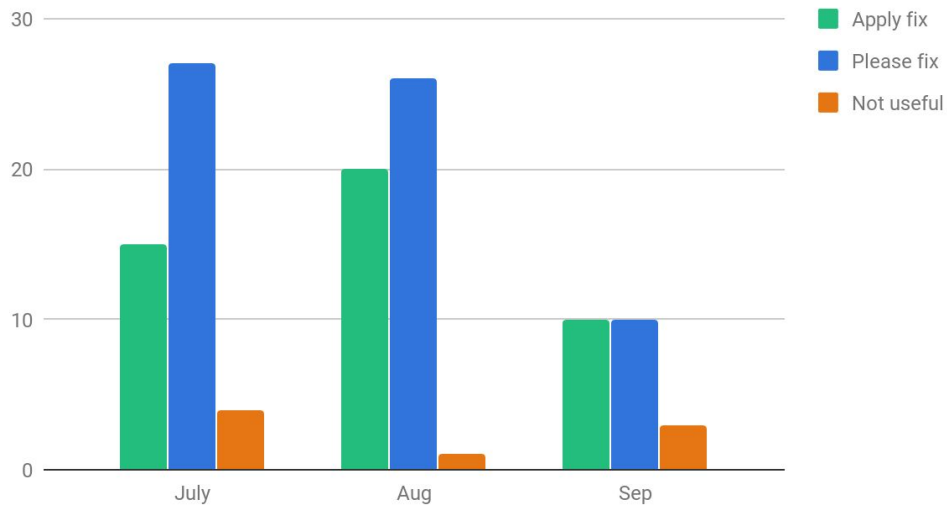
- Equal-and-opposite bugs
- Rarely taken code paths

# Heuristics



# Code review users like the suggestions

Apply fix, Please fix and Not useful



Only 7% not useful  
rate over last 3  
months



▼ **bsisko** nice catch! :-)

17:45, 19 Apr

[Reply](#) [Reply with quote](#)

▶ **jlpicard** That's one slick error prone

▶ **kjaneway** +1 The unit tests don't seem to discover this.

▼ **jtkirk** Added a unit test of toString() which will catch this error.

10:25, 9 May

[Reply](#) [Reply with quote](#)

# Try it out

Andrew Rice, Edward Aftandilian, Ciera Jaspan, Emily Johnston, Michael Pradel, and Yulissa Arroyo-Paredes. 2017. Detecting argument selection defects. Proc. ACM Program. Lang. 1, OOPSLA, Article 104 (October 2017), 22 pages. DOI: <https://doi.org/10.1145/3133928>

Error Prone: [github.com/google/error-prone](https://github.com/google/error-prone)

Artifact available at <https://doi.org/10.1145/3133928> (link in the paper)

