

# Automatic Generation of Oracles for Exceptional Behaviors



**Alberto Goffi** — [alberto.goffi@usi.ch](mailto:alberto.goffi@usi.ch)

USI Università della Svizzera italiana (Lugano, Switzerland)

**Alessandra Gorla**

IMDEA Software Institute (Madrid, Spain)

**Michael D. Ernst**

University of Washington (Seattle, Washington, USA)

**Mauro Pezzè**

USI Università della Svizzera italiana (Lugano, Switzerland)

# The Oracle Problem

Automatically generated test cases **need oracles**

# The Oracle Problem

Automatically generated test cases **need oracles**

Developers **do not write** formal **specifications**

# The Oracle Problem

Automatically generated test cases **need oracles**

Developers **do not write** formal **specifications**

Developers **do write Javadoc** comments

# The Oracle Problem

Automatically generated test cases **need oracles**

Developers **do not write** formal **specifications**

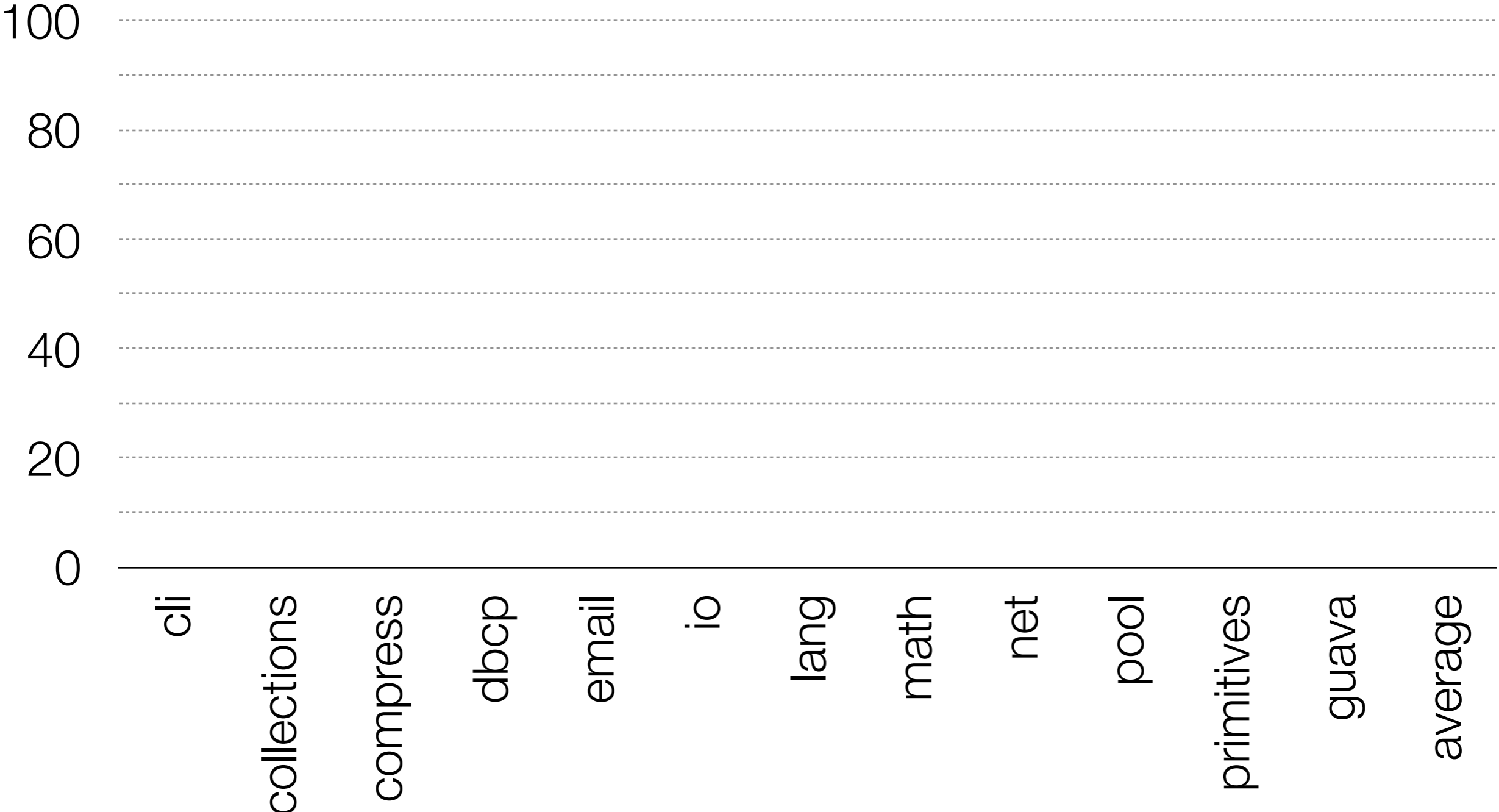
Developers **do write Javadoc** comments



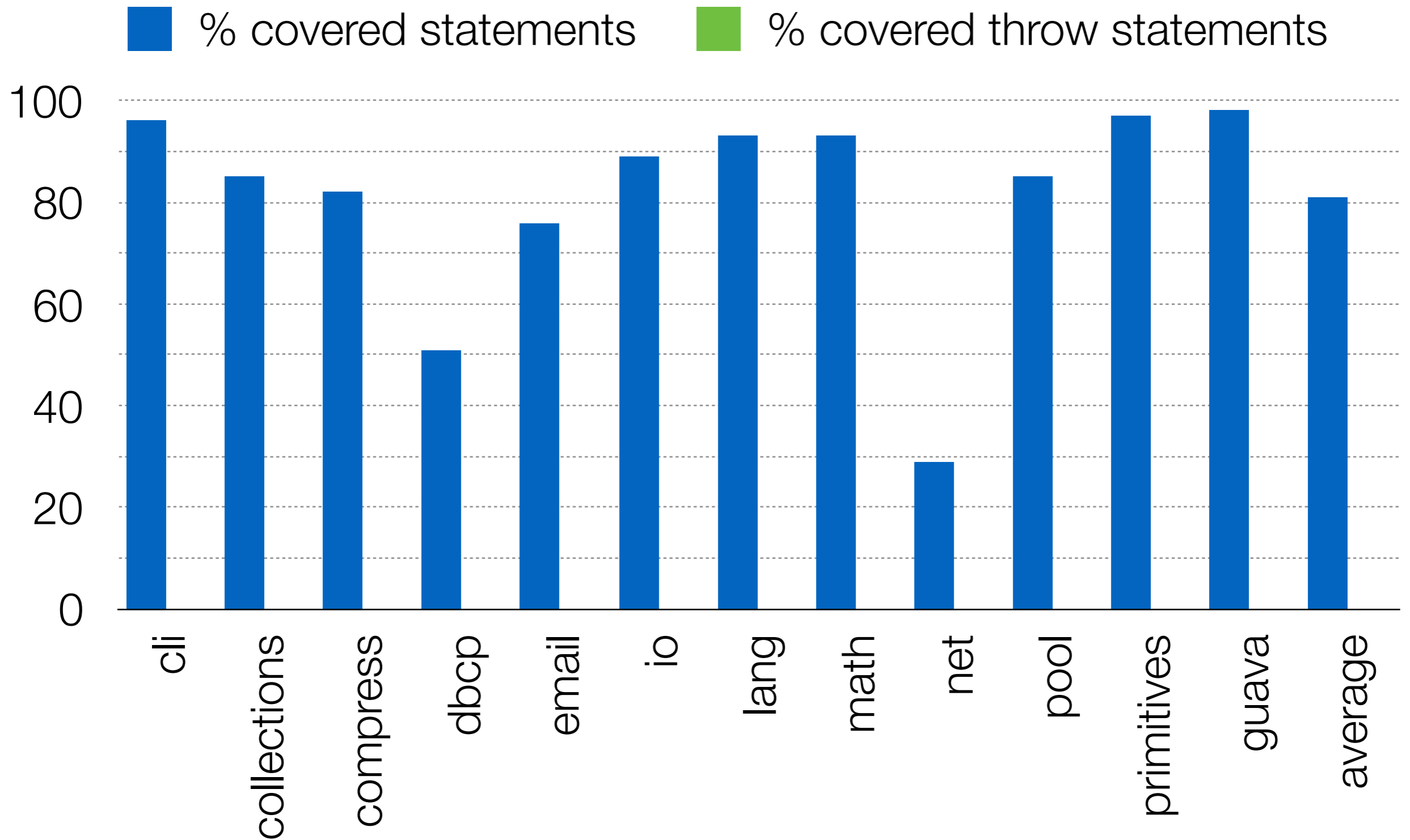
Automatically generate **oracles**  
**from Javadoc comments**

# Testers under-test exceptions

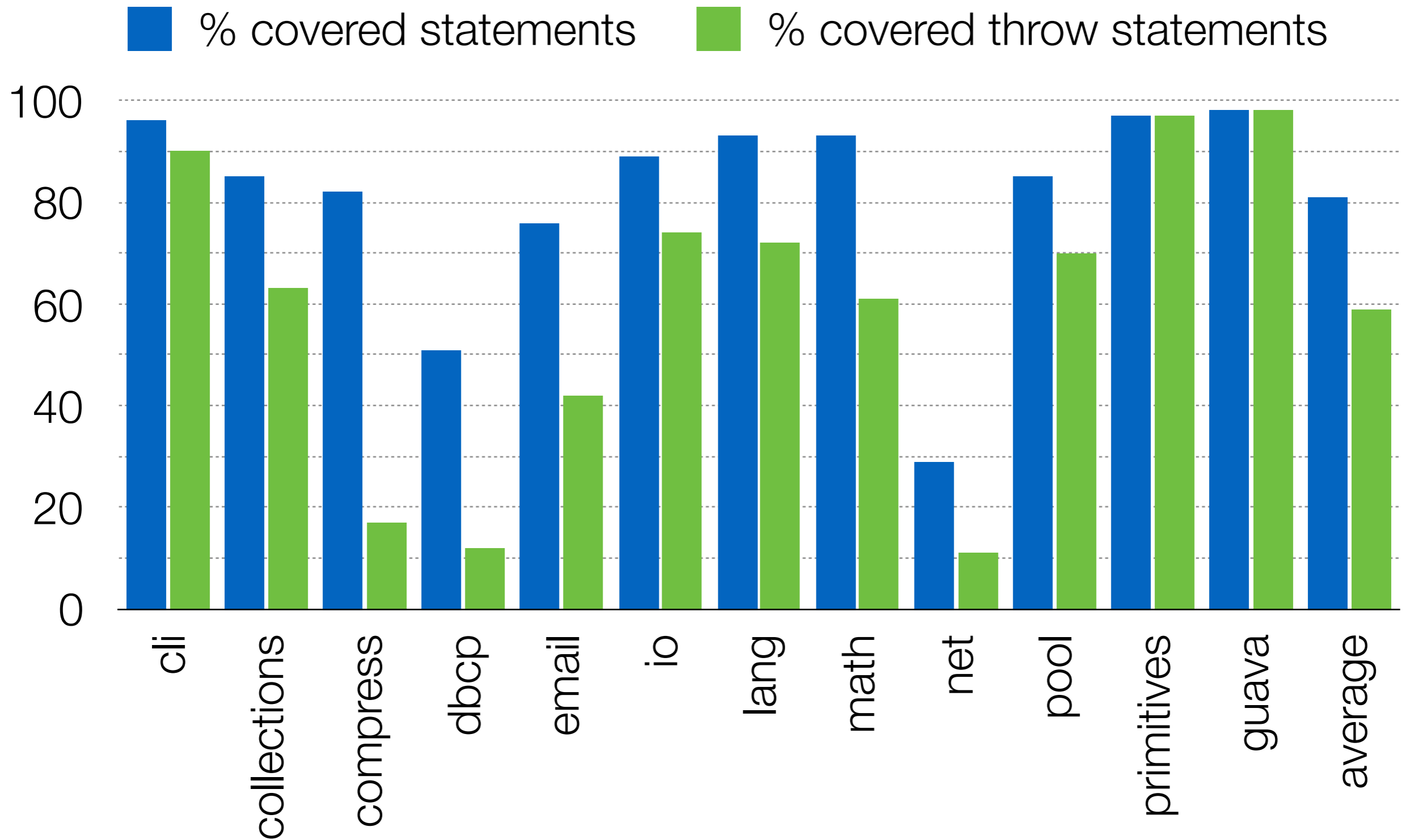
■ % covered statements    ■ % covered throw statements



# Testers under-test exceptions

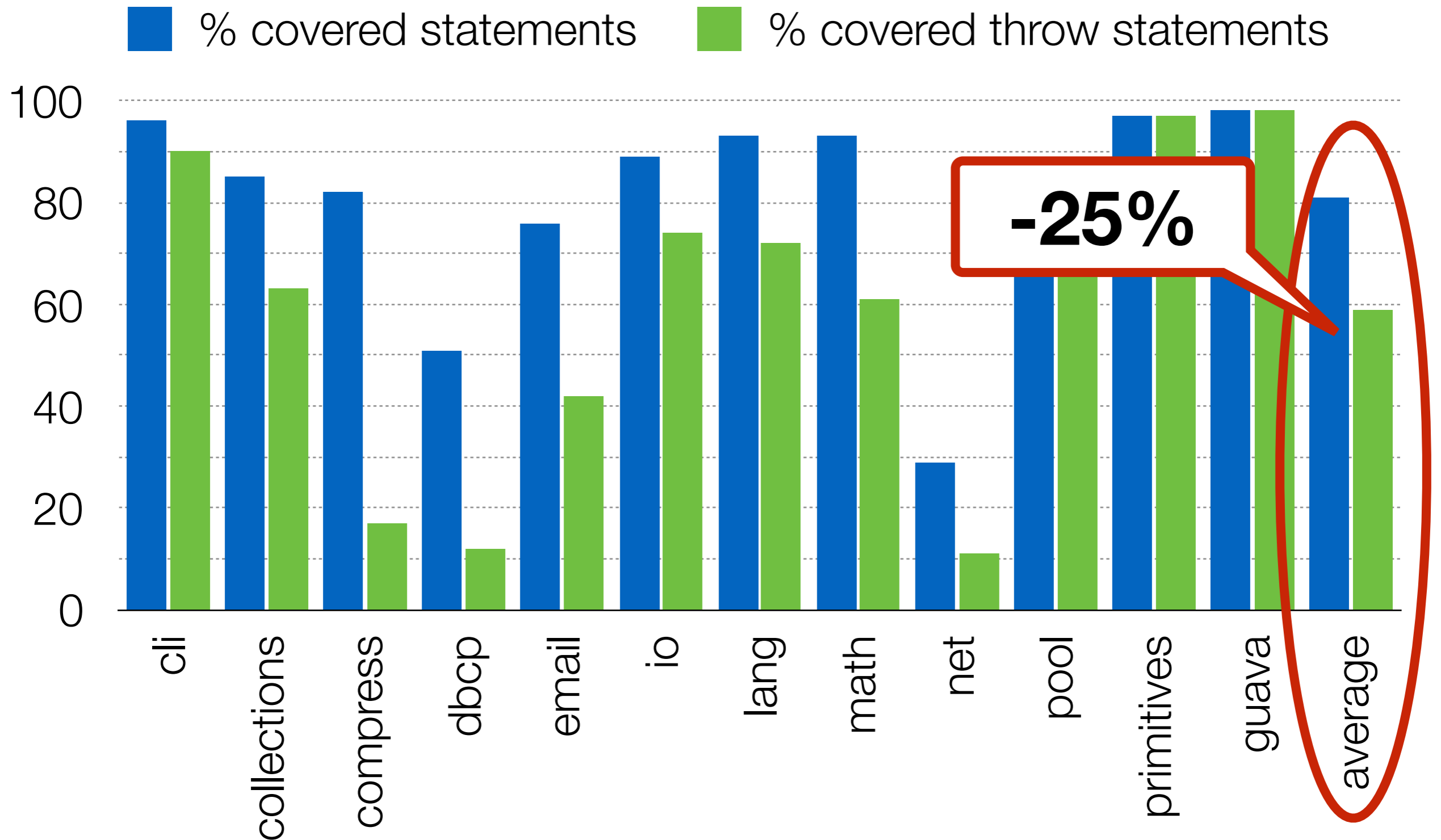


# Testers under-test exceptions





# Testers under-test exceptions



...and automatic testing does *not* help

```
public class FilterIterator implements Iterator {  
    public FilterIterator(Iterator i, Predicate p) {...}  
    public Object next() {...}  
    ...  
}
```

...and automatic testing does *not* help

```
public class FilterIterator implements Iterator {  
    public FilterIterator(Iterator i, Predicate p) {...}  
    public Object next() {...}  
    ...  
}
```

```
public void test {  
    FilterIterator i = new FilterIterator(null, null);  
    i.next();  
}
```

# What does an exception mean?

```
public class FilterIterator implements Iterator {  
    public FilterIterator(Iterator i, Predicate p) {...}  
    public Object next() {...}  
    ...  
}
```

```
public void test {  
    FilterIterator i = new FilterIterator(null, null);  
    i.next();  
}
```



**NullPointerException**

# What does an exception mean?

```
public class FilterIterator implements Iterator {  
    public FilterIterator(Iterator i, Predicate p) {...}  
    public Object next() {...}  
    ...  
}
```

```
public void test {  
    FilterIterator i = new FilterIterator(null, null);  
    i.next();  
}
```



**NullPointerException**

1. Expected behavior
2. Illegal input
3. Implementation bug



# What does an exception mean?

```
public class FilterIterator implements Iterator {
    public FilterIterator(Iterator i, Predicate p) {...}

    /**
     * @throws NullPointerException if either
     * the iterator or predicate are null
     */
    public Object next() {...}
}
```

```
public void test {
    FilterIterator i = new FilterIterator(null, null);
    i.next();
}
```



**NullPointerException**

1. Expected behavior
2. Illegal input
3. Implementation bug



# What does an exception mean?

```
public class FilterIterator implements Iterator {
    public FilterIterator(Iterator i, Predicate p) {...}

    /**
     * @throws NullPointerException if either
     * the iterator or predicate are null
     */
    public Object next() {...}
}
```

```
public void test {
    FilterIterator i = new FilterIterator(null, null);
    i.next();
}
```



**NullPointerException**

**1. Expected behavior**

2. Illegal input

3. Implementation bug

# What does an exception mean?

```
public class FilterIterator implements Iterator {
    public FilterIterator(Iterator i, Predicate p) {...}

    /**
     * @throws NullPointerException if either
     * the iterator or predicate are null
     */
    public Object next() {...}
}
```

```
public void test {
    FilterIterator i = new FilterIterator(null, null);
    i.next();
}
```

FALSE ALARM



**NullPointerException**

**1. Expected behavior**

2. Illegal input

3. Implementation bug



# Consider another implementation

```
public class FilterIterator implements Iterator {
    public FilterIterator(Iterator i, Predicate p) {...}

    /**
     * @throws NullPointerException if either
     * the iterator or predicate are null
     */
    public Object next() {...}
}
```

```
public void test {
    FilterIterator i = new FilterIterator(null, null);
    i.next();
}
```



**No Exceptions**

1. Expected behavior
2. Illegal input
3. Implementation bug



# Consider another implementation

```
public class FilterIterator implements Iterator {
    public FilterIterator(Iterator i, Predicate p) {...}

    /**
     * @throws NullPointerException if either
     * the iterator or predicate are null
     */
    public Object next() {...}
}
```

```
public void test {
    FilterIterator i = new FilterIterator(null, null);
    i.next();
}
```



**No Exceptions**

1. Expected behavior
2. Illegal input
- 3. Implementation bug**

# Consider another implementation

```
public class FilterIterator implements Iterator {  
    public FilterIterator(Iterator i, Predicate p) {...}  
  
    /**  
     * @throws NullPointerException if either  
     * the iterator or predicate are null  
     */  
    public Object next() {...}  
}
```

```
public void test {  
    FilterIterator i = new FilterIterator(new Iterator(), null);  
    i.next();  
}
```

Missed Alarm



**No Exceptions**

1. Expected behavior
2. Illegal input
- 3. Implementation bug**

Test

```
public void test {  
    FilterIterator i =  
        new FilterIterator(null, null);  
    i.next();  
}
```

Test

```
public void test {  
    FilterIterator i =  
        new FilterIterator(null, null);  
    i.next();  
}
```

Documentation

```
/**  
 * @throws NullPointerException if either  
 * the iterator or predicate are null  
 */  
public Object next() {...}
```

Test

+

Documentation

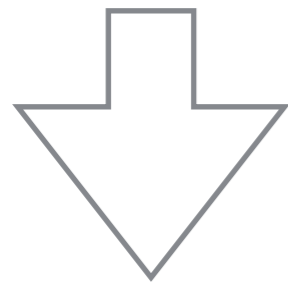
```
public void test {  
    FilterIterator i =  
        new FilterIterator(null, null);  
    i.next();  
}
```

```
/**  
 * @throws NullPointerException if either  
 * the iterator or predicate are null  
 */  
public Object next() {...}
```

Test

+

Documentation



Test + Oracle

```
public void test {  
    FilterIterator i =  
        new FilterIterator(null, null);  
    i.next();  
}
```

```
/**  
 * @throws NullPointerException if either  
 * the iterator or predicate are null  
 */  
public Object next() {...}
```

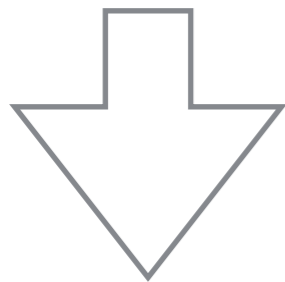


```
public void test {  
    FilterIterator i =  
        new FilterIterator(null, null);  
    if (i.getIterator() == null ||  
        i.getPredicate() == null) {  
        try {  
            i.next();  
            fail("Expected NPE not thrown");  
        } catch (NullPointerException e) {  
            // Expected exception  
        }  
    } else {  
        i.next();  
    }  
}
```

Test

+

Documentation



Test + Oracle

```
public void test {  
    FilterIterator i =  
        new FilterIterator(null, null);  
    i.next();  
}
```

```
/**  
 * @throws NullPointerException if either  
 * the iterator or predicate are null  
 */  
public Object next() {...}
```



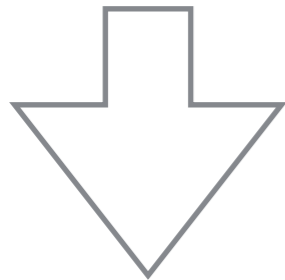
```
public void test {  
    FilterIterator i =  
        new FilterIterator(null, null);  
    if (i.getIterator() == null ||  
        i.getPredicate() == null) {  
        try {  
            i.next();  
            fail("Expected NPE not thrown");  
        } catch (NullPointerException e) {  
            // Expected exception  
        }  
    } else {  
        i.next();  
    }  
}
```



Test

+

Documentation



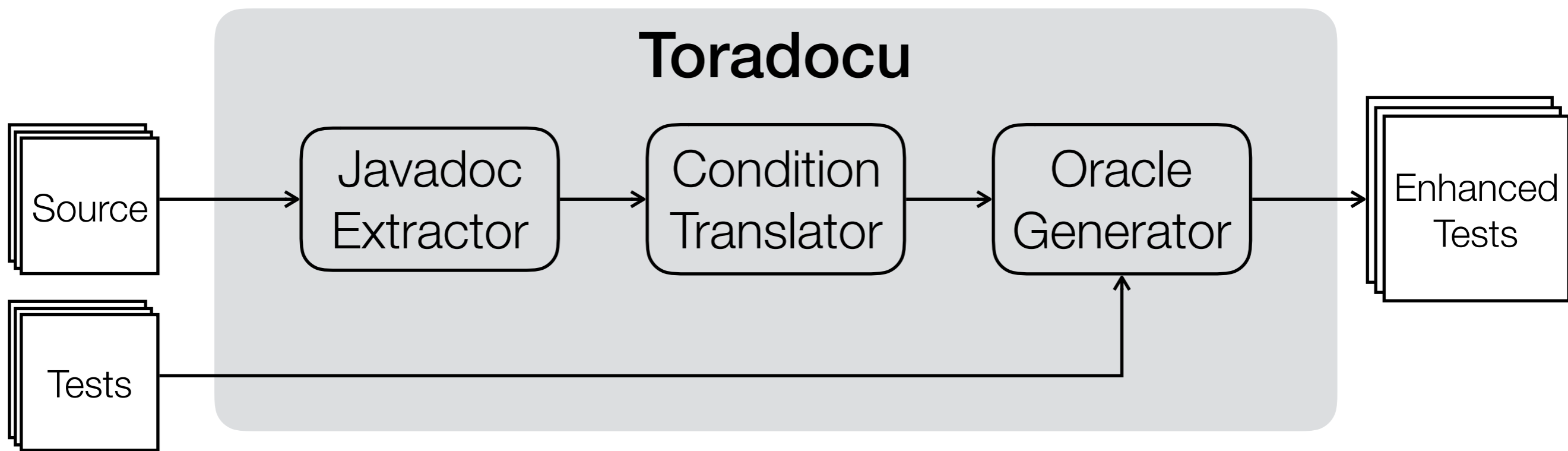
Test + Oracle

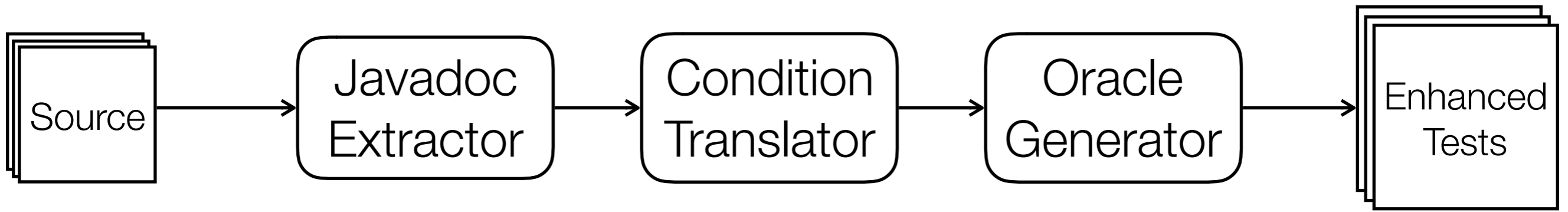
```
public void test {
    FilterIterator i =
        new FilterIterator(null, null);
    i.next();
}
```

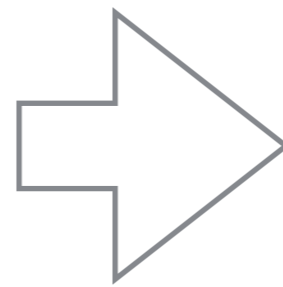
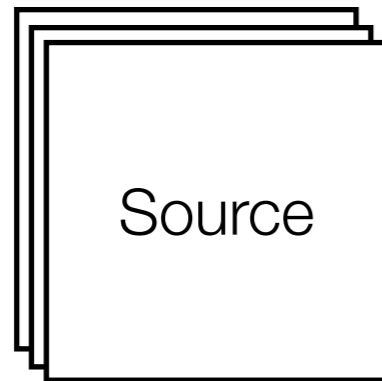
```
/**
 * @throws NullPointerException if either
 * the iterator or predicate are null
 */
public Object next() {...}
```



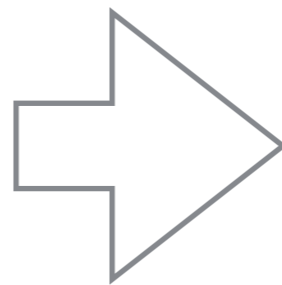
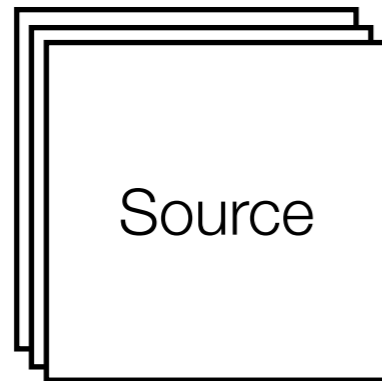
```
public void test {
    FilterIterator i =
        new FilterIterator(null, null);
    if (i.getIterator() == null ||
        i.getPredicate() == null) {
        try {
            i.next();
            fail("Expected NPE not thrown");
        } catch (NullPointerException e) {
            // Expected exception
        }
    } else {
        i.next();
    }
}
```





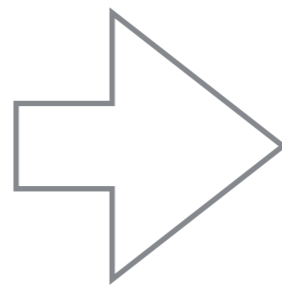


1. Method Signature
2. Expected Exceptions
3. Conditions (Text)

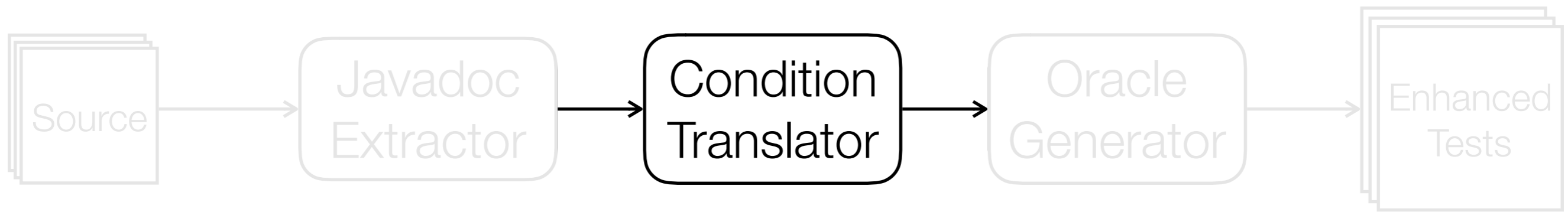


1. Method Signature
2. Expected Exceptions
3. Conditions (Text)

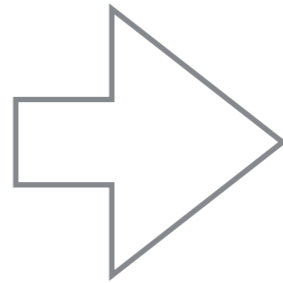
```
/**  
 * @throws NullPointerException  
 * if either the iterator  
 * or predicate are null  
 */  
public Object next() {...}
```



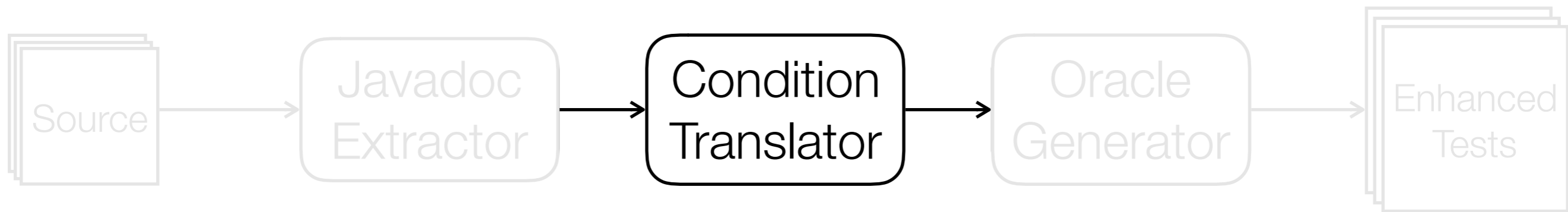
1. `FilterIterator.next()`
2. `NullPointerException`
3. "if either the iterator or predicate are null"



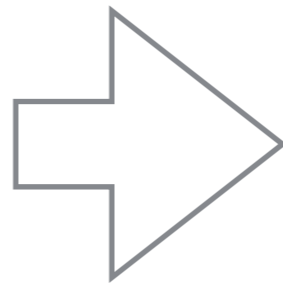
1. Method Signature
2. Expected Exceptions
3. Conditions (Text)



1. Method Signature
2. Expected Exceptions
3. **Conditions (Code)**

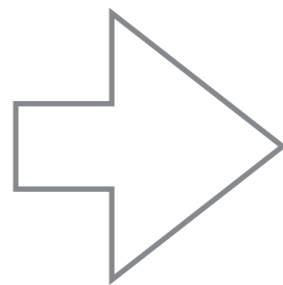


1. Method Signature
2. Expected Exceptions
3. Conditions (Text)

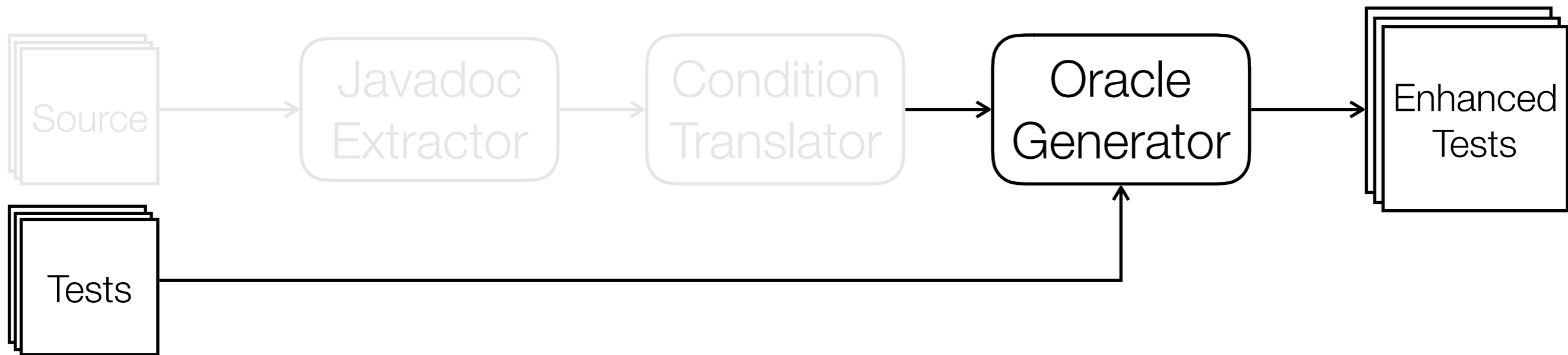


1. Method Signature
2. Expected Exceptions
3. **Conditions (Code)**

1. `FilterIterator.next()`
2. `NullPointerException`
3. "if either the iterator or predicate are null"

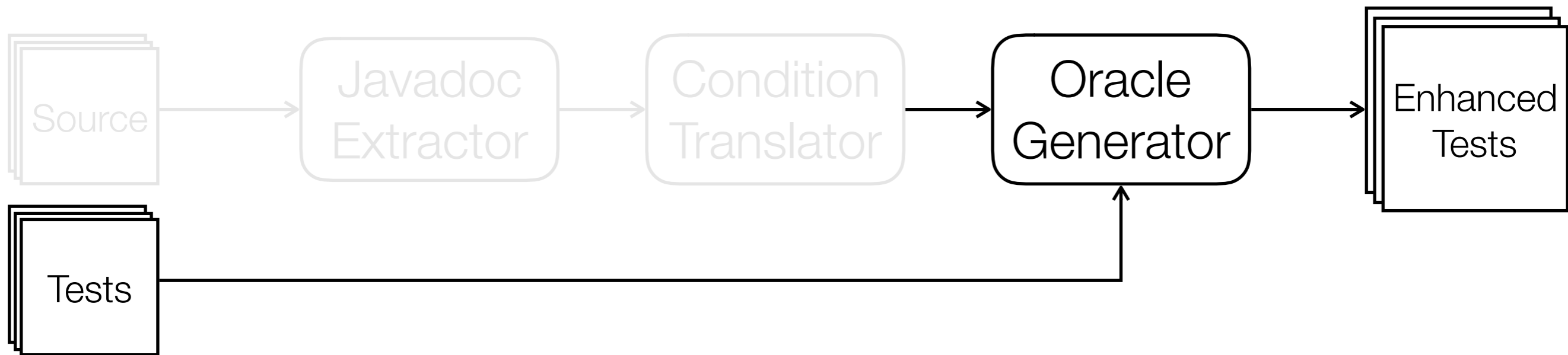


1. `FilterIterator.next()`
2. `NullPointerException`
3. **`getIterator() == null || getPredicate() == null`**



1. `FilterIterator.next()`
2. `NullPointerException`
3. `getIterator() == null ||  
getPredicate() == null`





Test

+

Oracle

=

Test + Oracle

```

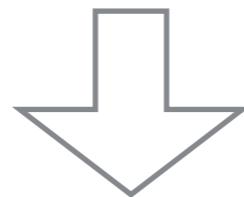
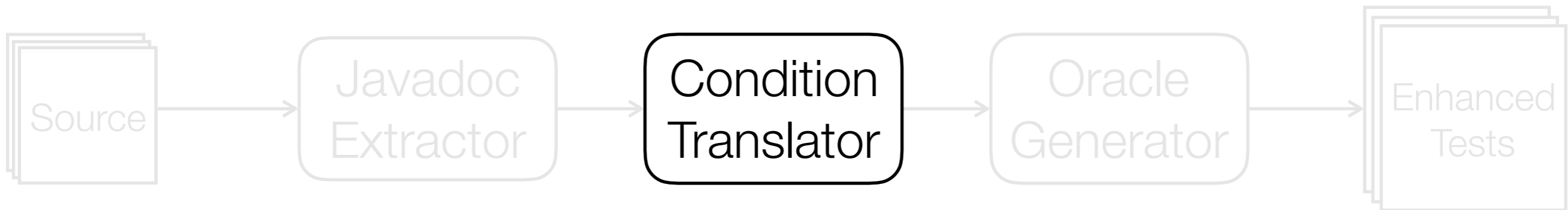
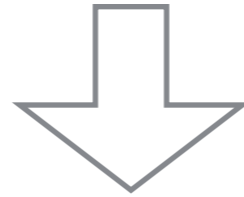
public void test {
    FilterIterator i = new FilterIterator(null, null);
    i.next();
}
  
```

1. `FilterIterator.next()`
2. `NullPointerException`
3. `getIterator() == null || getPredicate() == null`

```

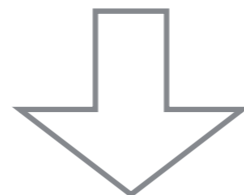
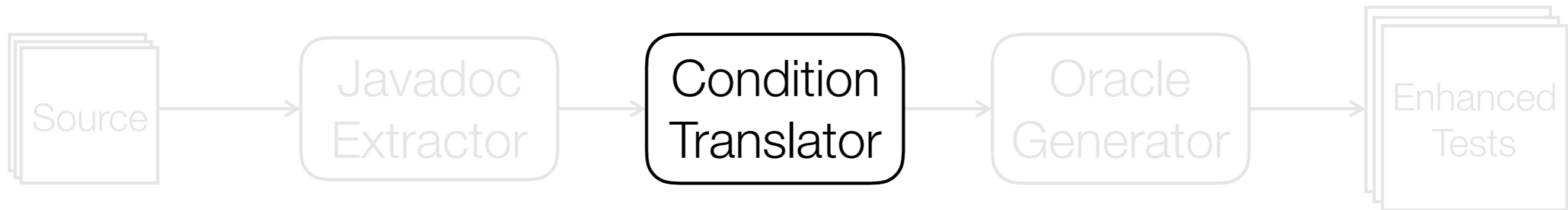
public void test {
    FilterIterator i = new FilterIterator(null, null);
    if (i.getIterator()==null || i.getPredicate()==null) {
        try {
            i.next();
            fail("Expected NPE not thrown");
        } catch(NullPointerException e) {
            // Expected exception
        }
    } else { i.next(); }
}
  
```

Natural Language Condition



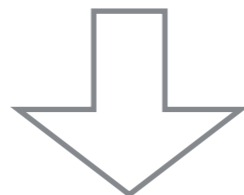
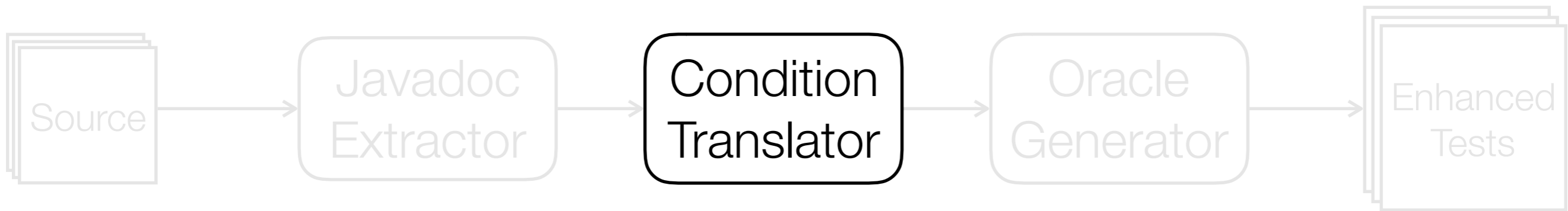
Java boolean condition

`"if the array is empty"`



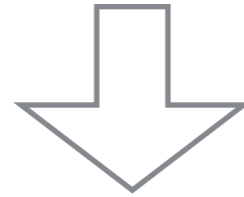
`array.length == 0`

`"if foo is positive"`

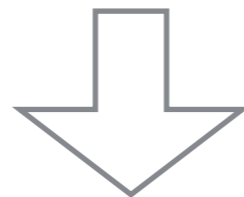
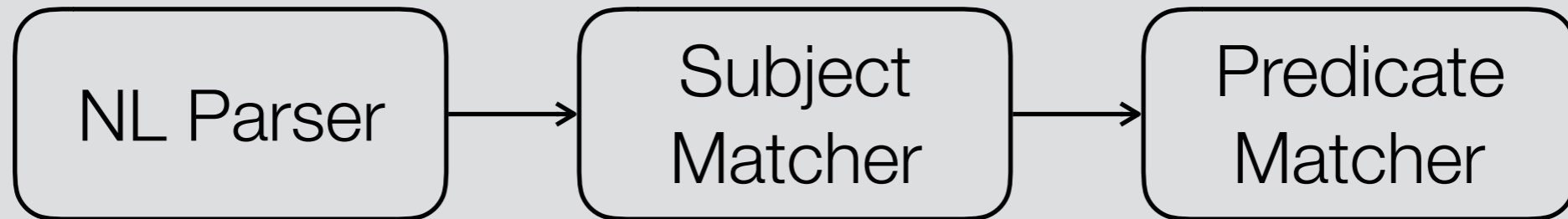


`foo > 0`

Natural Language Condition



## Condition Translator



Java Condition



`"if either the iterator or predicate are null"`



"if either the **iterator** or predicate **are null**"

Proposition

Subject

Predicate

---

1

"iterator"

"are null"



"if either the **iterator** or **predicate** **are null**"

Proposition	Subject	Predicate
1	"iterator"	"are null"
2	"predicate"	"are null"

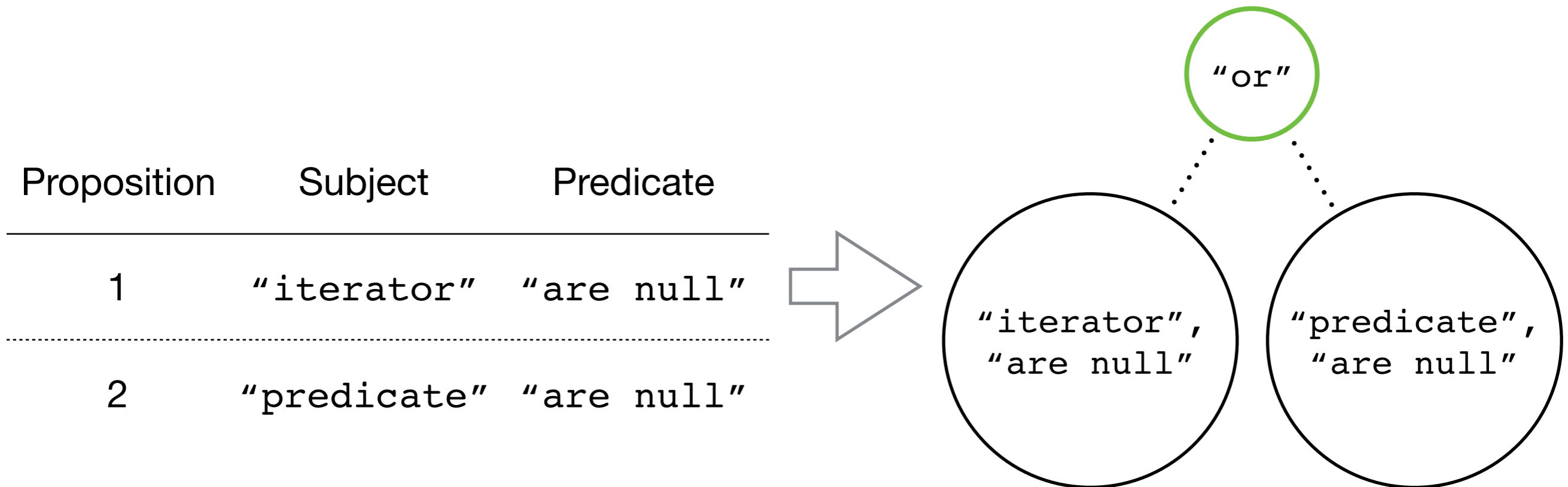


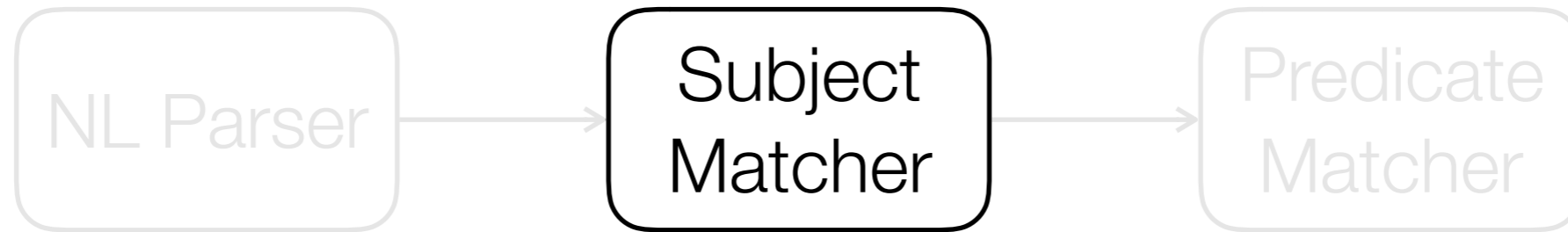


"if either the `iterator` `or` `predicate` are null"

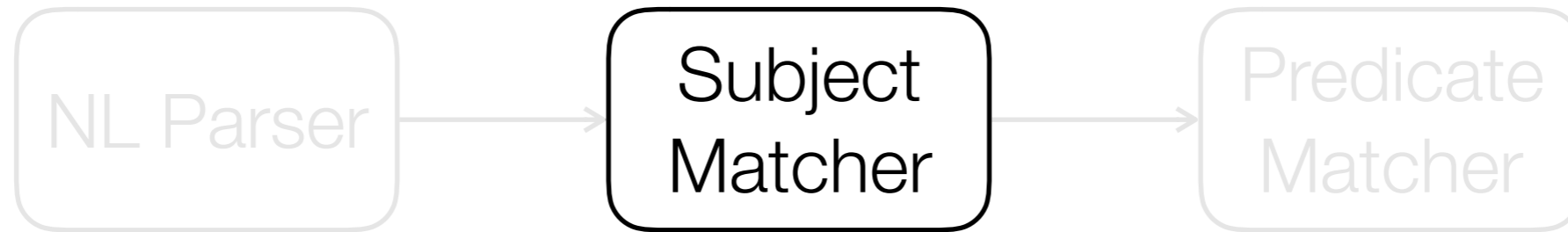


"if either the **iterator** **or** **predicate** are null"

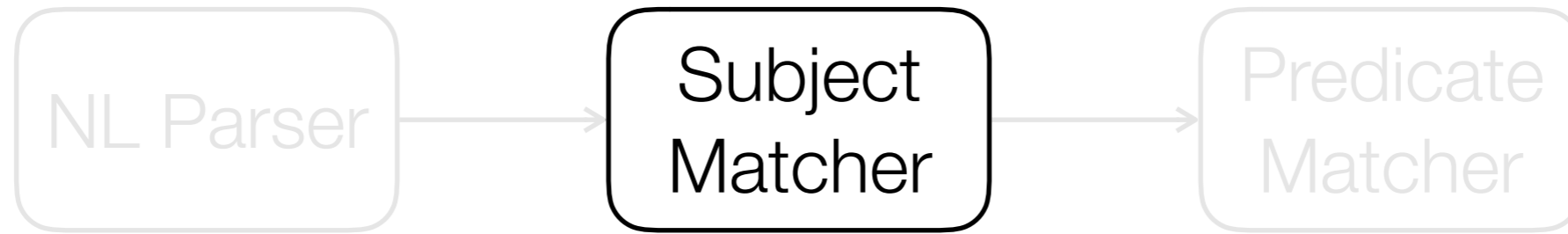




Proposition	Subject	Predicate
1	<code>"iterator"</code>	<code>"are null"</code>
2	<code>"predicate"</code>	<code>"are null"</code>



**Subject: "iterator"**



Subject: "iterator"

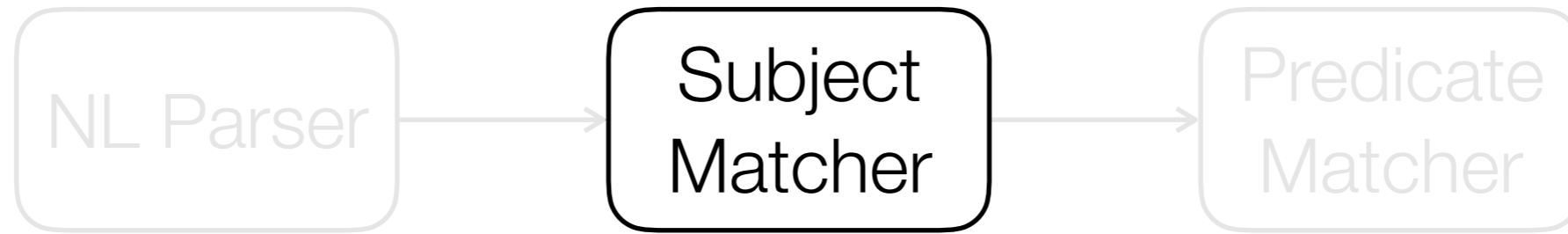
## Candidates

Formal Parameters

Class Name

Non-void Nullary Methods

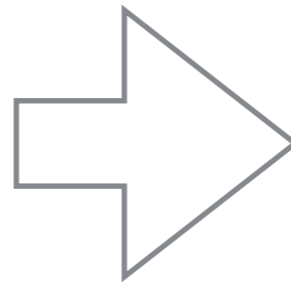
Fields



Subject: "iterator"

## Candidates

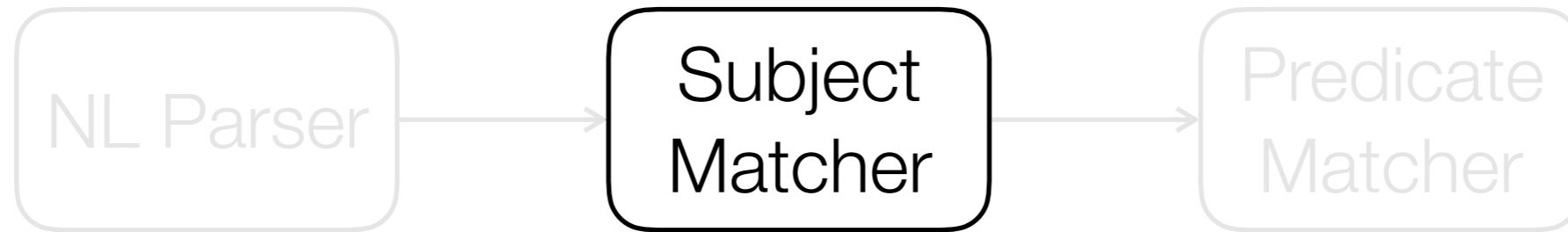
Formal Parameters  
Class Name  
Non-void Nullary Methods  
Fields



Candidate

**FilterIterator**  
getIterator  
getPredicate  
next  
...

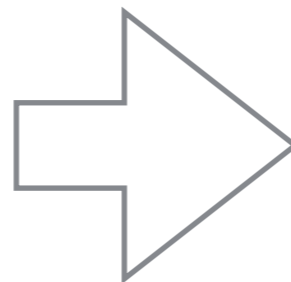
```
public class FilterIterator implements Iterator {  
    public FilterIterator(Iterator i, Predicate p) {...}  
    public Iterator getIterator() {...}  
    public Predicate getPredicate() {...}  
    public Object next() {...}  
    ...  
}
```



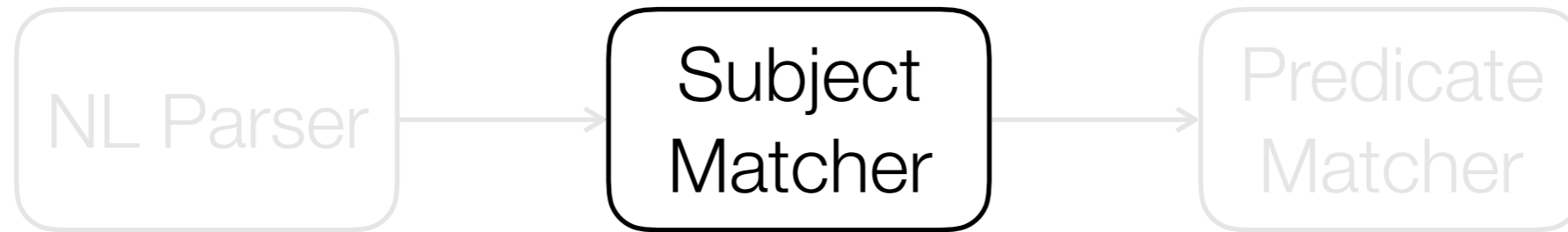
Subject: "iterator"

### Candidates

Formal Parameters  
Class Name  
Non-void Nullary Methods  
Fields



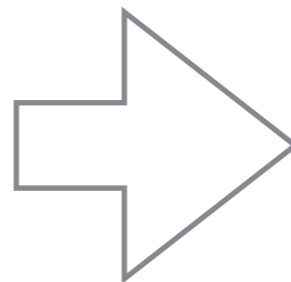
Candidate	Distance
<b>FilterIterator</b>	6
<b>getIterator</b>	3
<b>getPredicate</b>	9
<b>next</b>	6
...	



Subject: "iterator" => **getIterator()**

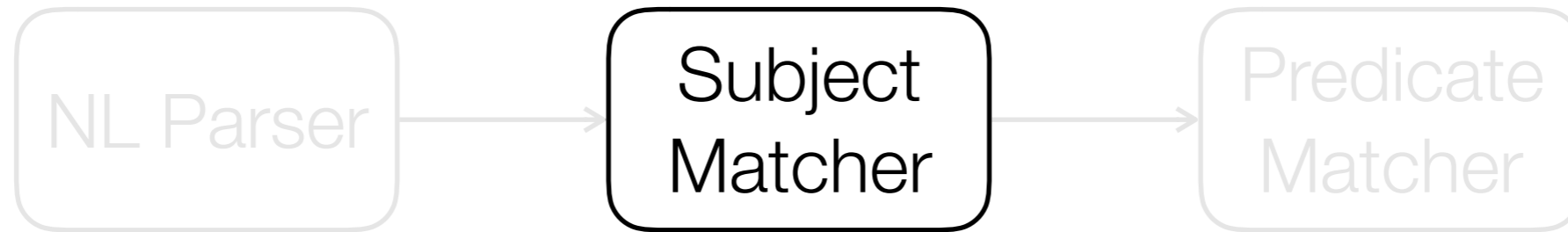
### Candidates

Formal Parameters  
Class Name  
Non-void Nullary Methods  
Fields

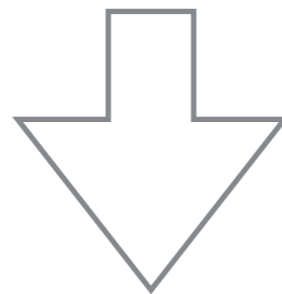


Candidate	Distance
FilterIterator	6
<b>getIterator</b>	<b>3</b>
getPredicate	9
next	6
...	





Proposition	Subject	Predicate
1	<code>"iterator"</code>	<code>"are null"</code>
2	<code>"predicate"</code>	<code>"are null"</code>



Proposition	Subject	Predicate
1	<code>FilterIterator.getIterator()</code>	<code>"are null"</code>
2	<code>FilterIterator.getPredicate()</code>	<code>"are null"</code>



Proposition

Subject

Predicate

1

`FilterIterator.getIterator()`

`"are null"`

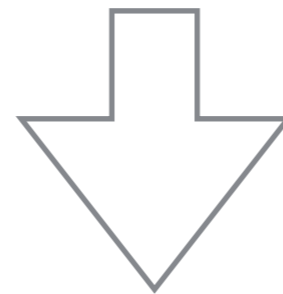
2

`FilterIterator.getPredicate()`

`"are null"`



Proposition	Subject	Predicate
1	<code>FilterIterator.getIterator()</code>	<code>"are null"</code>
2	<code>FilterIterator.getPredicate()</code>	<code>"are null"</code>



Proposition	Subject	Predicate
1	<code>FilterIterator.getIterator()</code>	<code>"==null"</code>
2	<code>FilterIterator.getPredicate()</code>	<code>"==null"</code>



```
/**  
 * @throws IllegalArgumentException if employees is empty  
 */  
public void sortByAge(List employees) {...}
```

Proposition	Subject	Predicate
1	employees	"is empty"



```

/**
 * @throws IllegalArgumentException if employees is empty
 */
public void sortByAge(List employees) {...}
  
```

Proposition	Subject	Predicate
1	employees	"is empty"
	Candidate	Distance
	clear	7
	<b>isEmpty</b>	<b>1</b>
	iterator	6
	size	7
	...	



```

/**
 * @throws IllegalArgumentException if employees is empty
 */
public void sortByAge(List employees) {...}
  
```

Proposition	Subject	Predicate
1	employees	"is empty"

↓

Condition	Translation
1	employees.isEmpty()



`"if either the iterator or predicate are null"`

Condition

Translation

---

1

`FilterIterator.getIterator()==null`

---

2

`FilterIterator.getPredicate()==null`

# Positive conditions

`"if either the iterator or predicate are null"`

Condition

Translation

---

1	<code>(FilterIterator.getIterator()==null)==<b>true</b></code>
---	--

---

2	<code>(FilterIterator.getPredicate()==null)==<b>true</b></code>
---	---



# Positive conditions

“if either the iterator or predicate **are** null”

Condition	Translation
1	<code>(FilterIterator.getIterator()==null)==true</code>
2	<code>(FilterIterator.getPredicate()==null)==true</code>

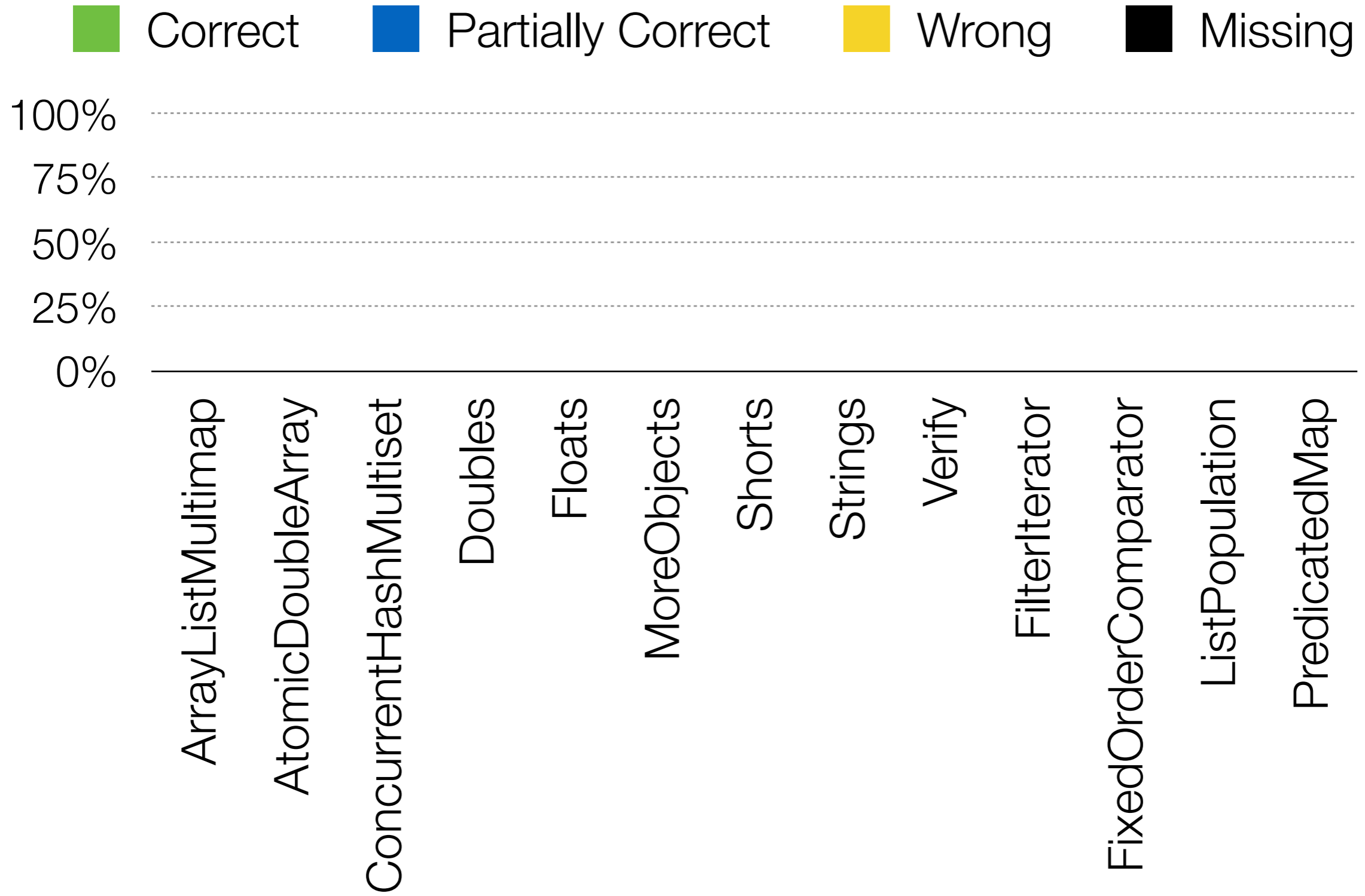
# Negative conditions

“if either the iterator or predicate **aren't** null”

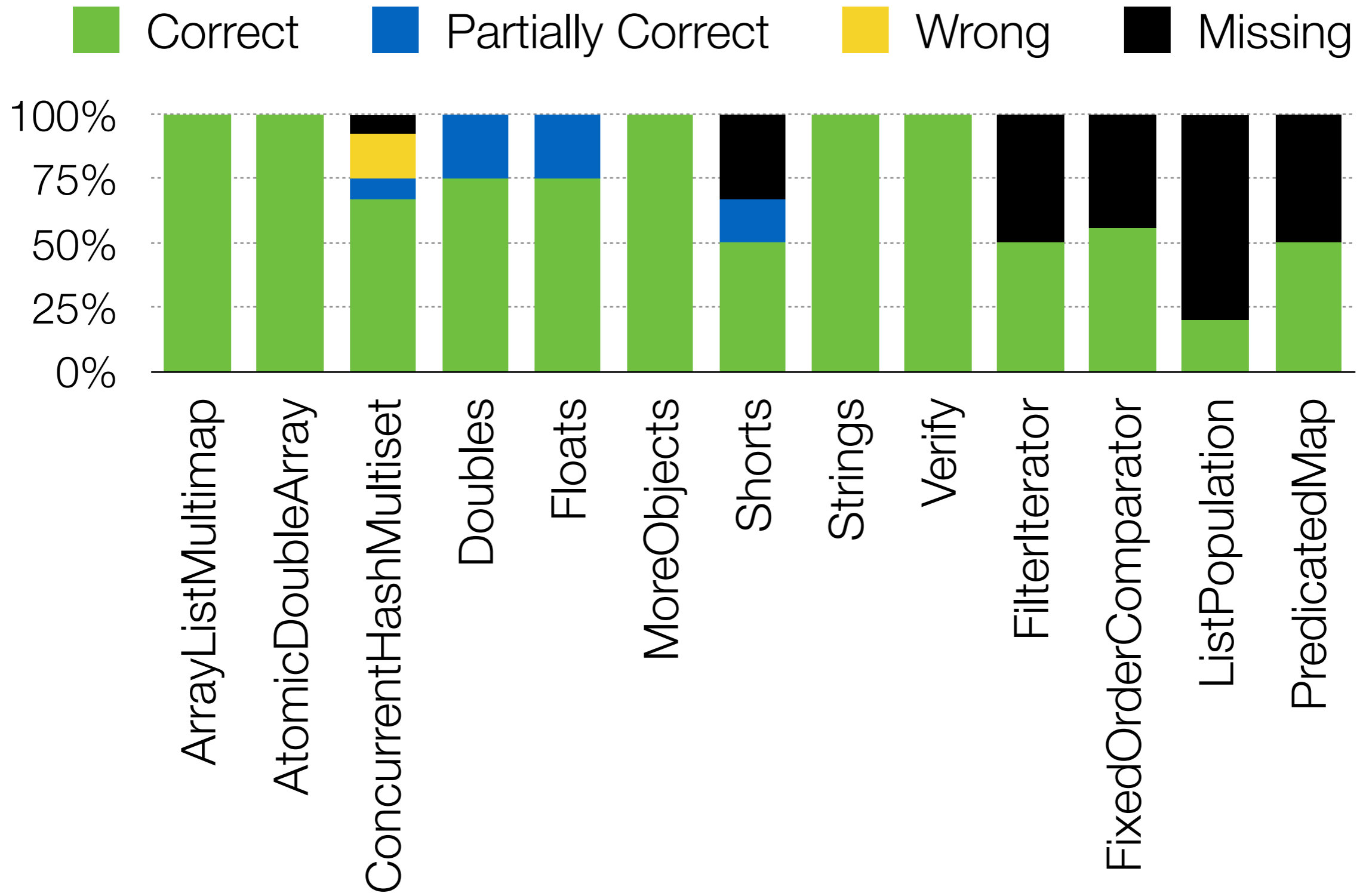
Condition	Translation
1	<code>(FilterIterator.getIterator()==null)==false</code>
2	<code>(FilterIterator.getPredicate()==null)==false</code>

Evaluation

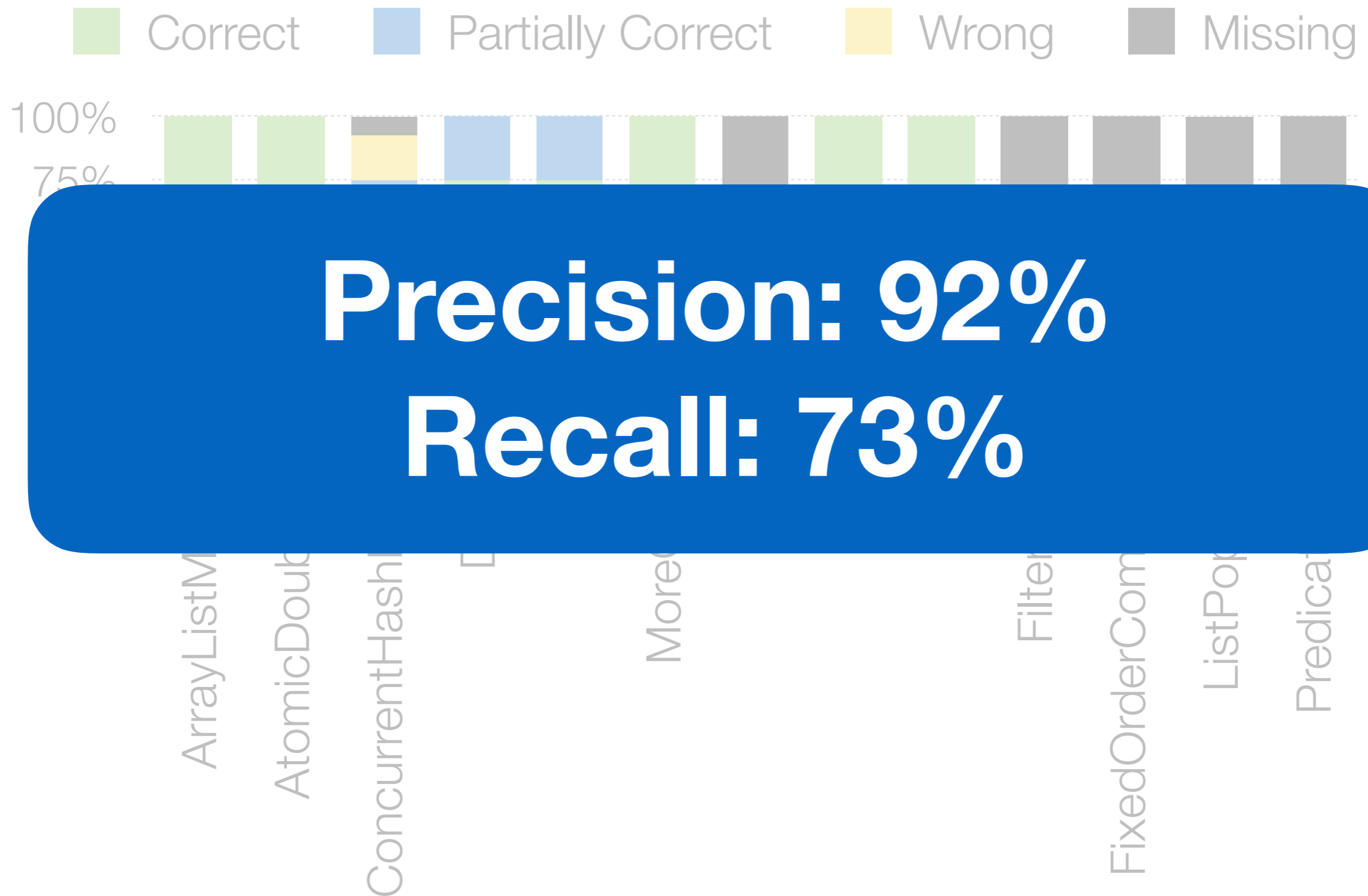
# From NL to Java Conditions



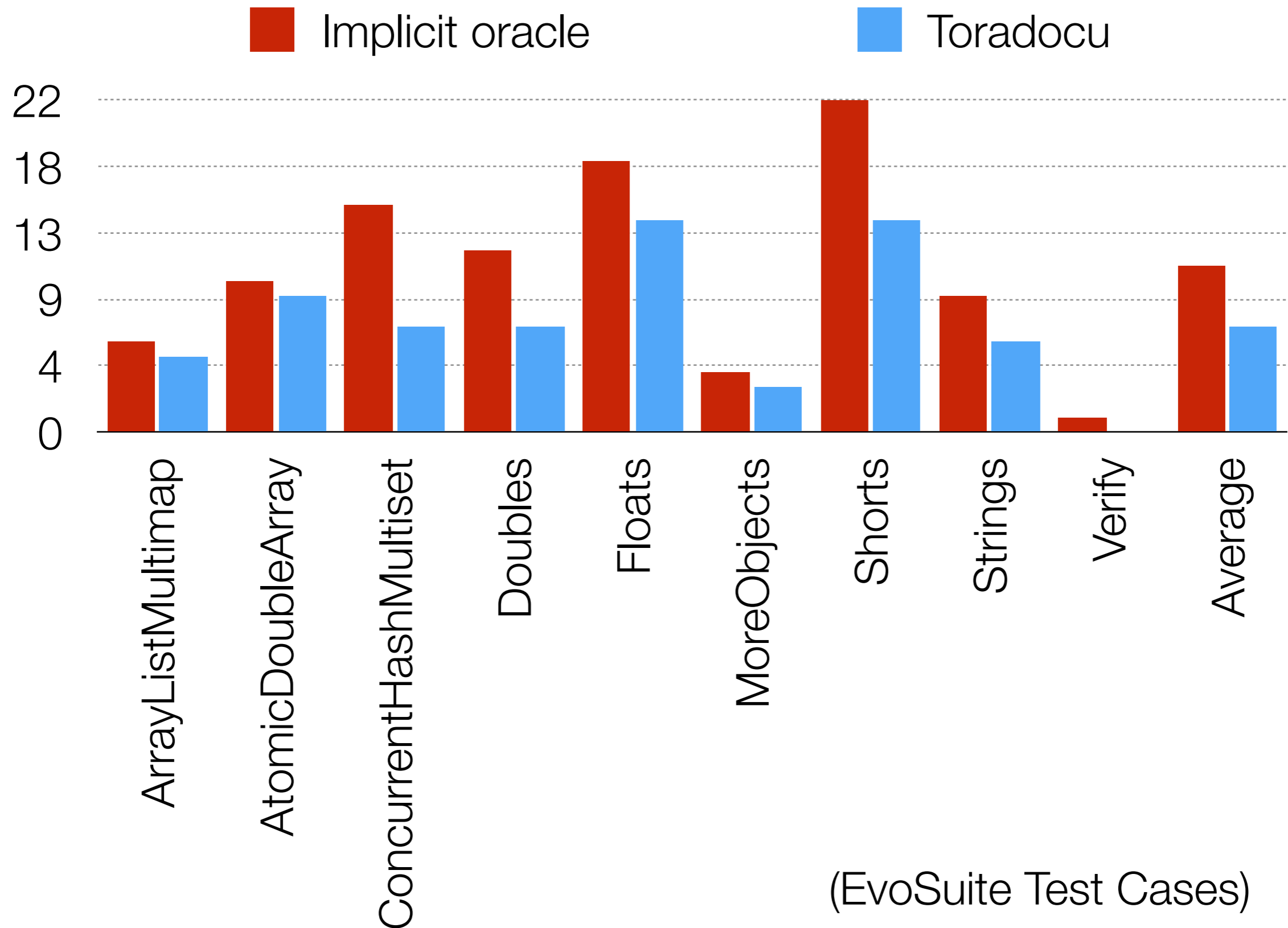
# From NL to Java Conditions



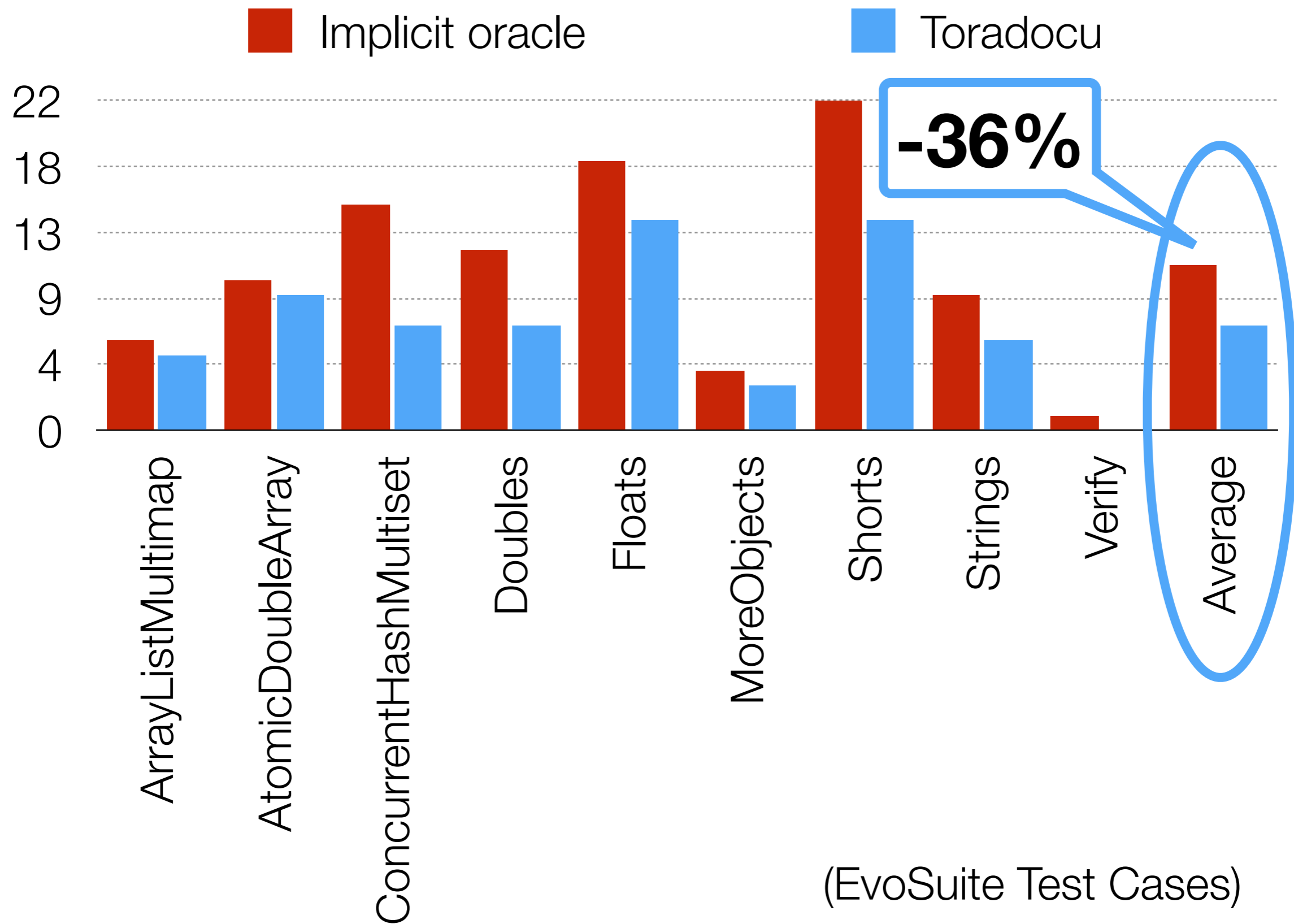
# From NL to Java Conditions



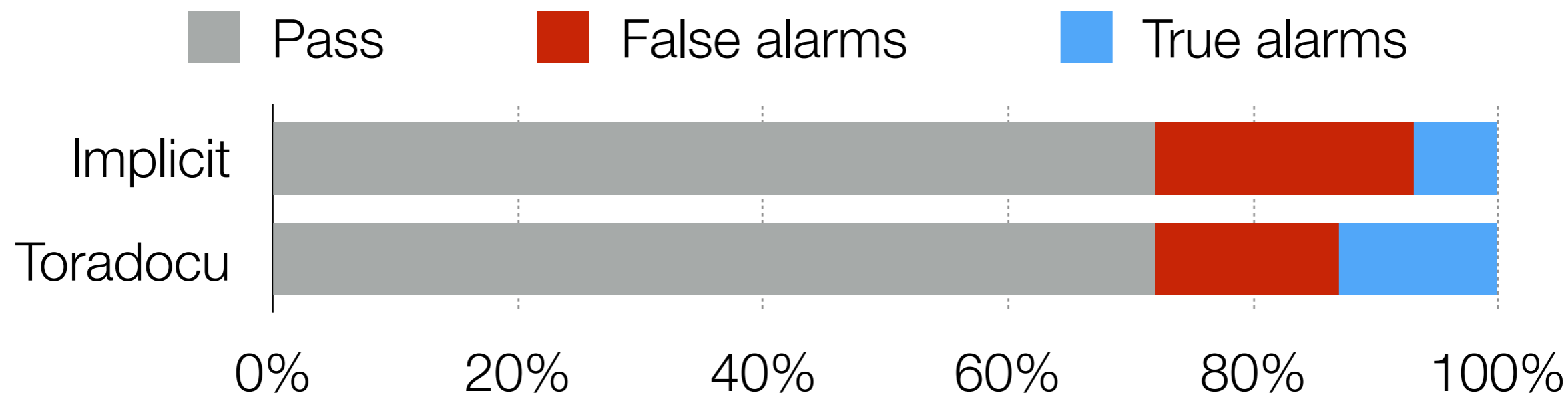
# False Alarms — Google Guava



# False Alarms — Google Guava

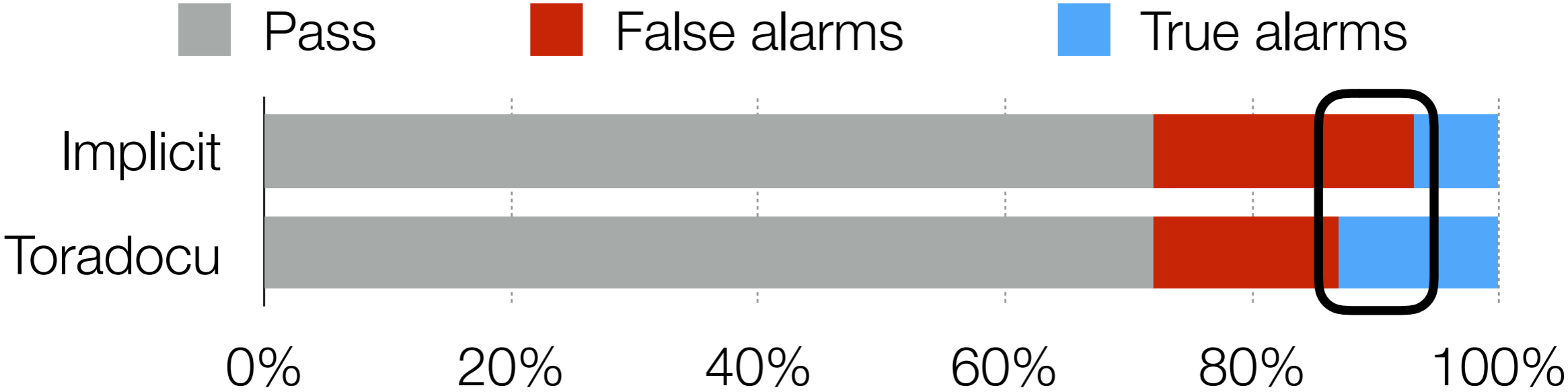


# True Alarms — Apache Commons

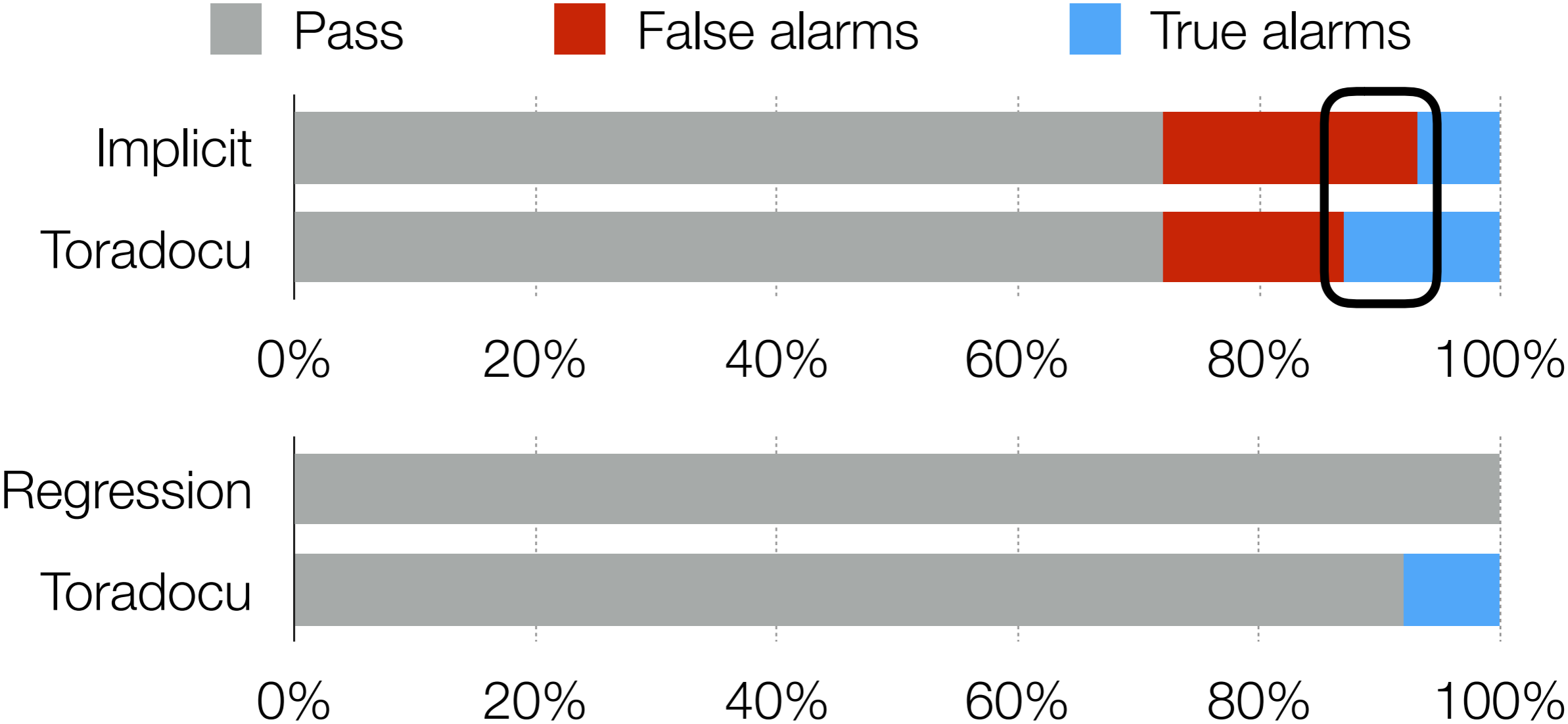




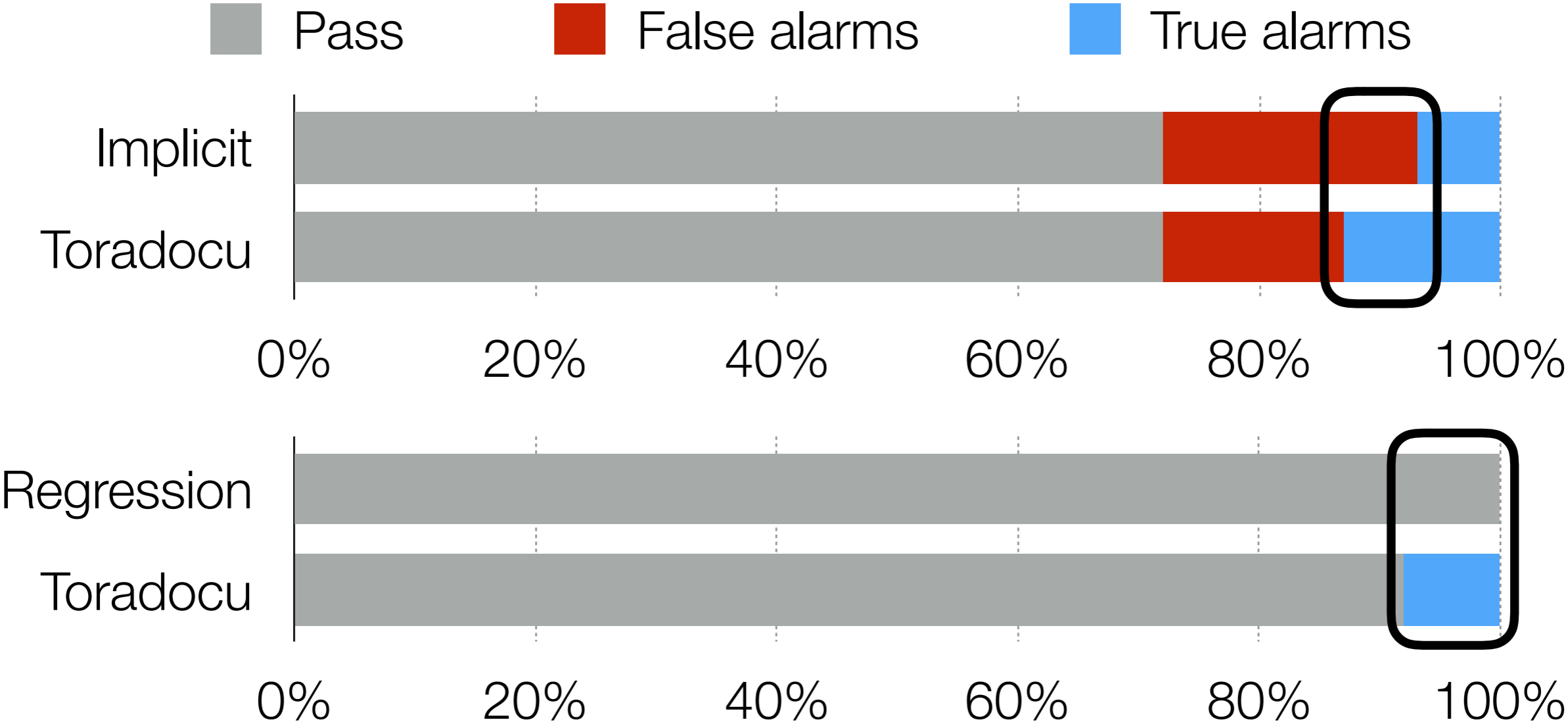
# True Alarms — Apache Commons



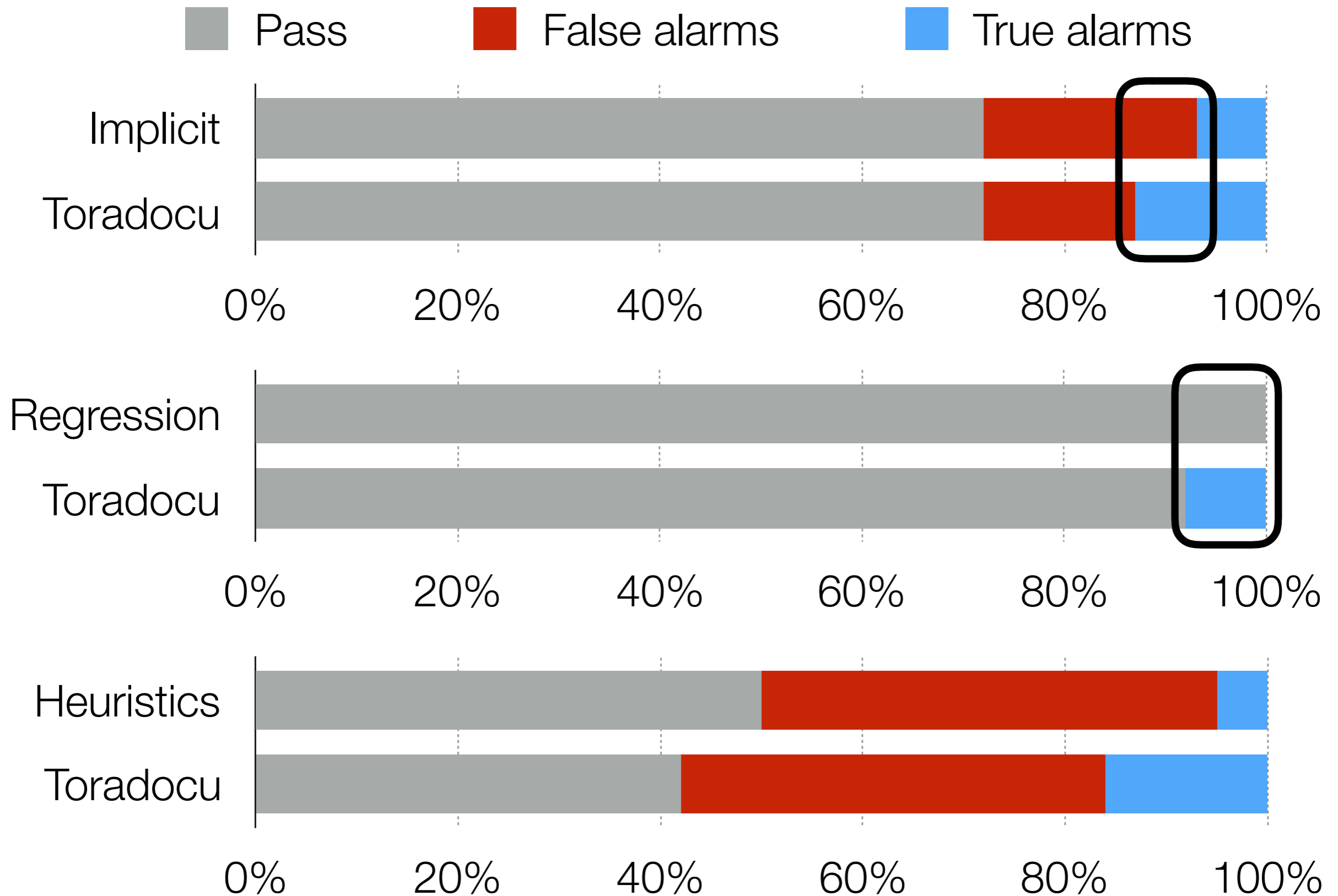
# True Alarms — Apache Commons



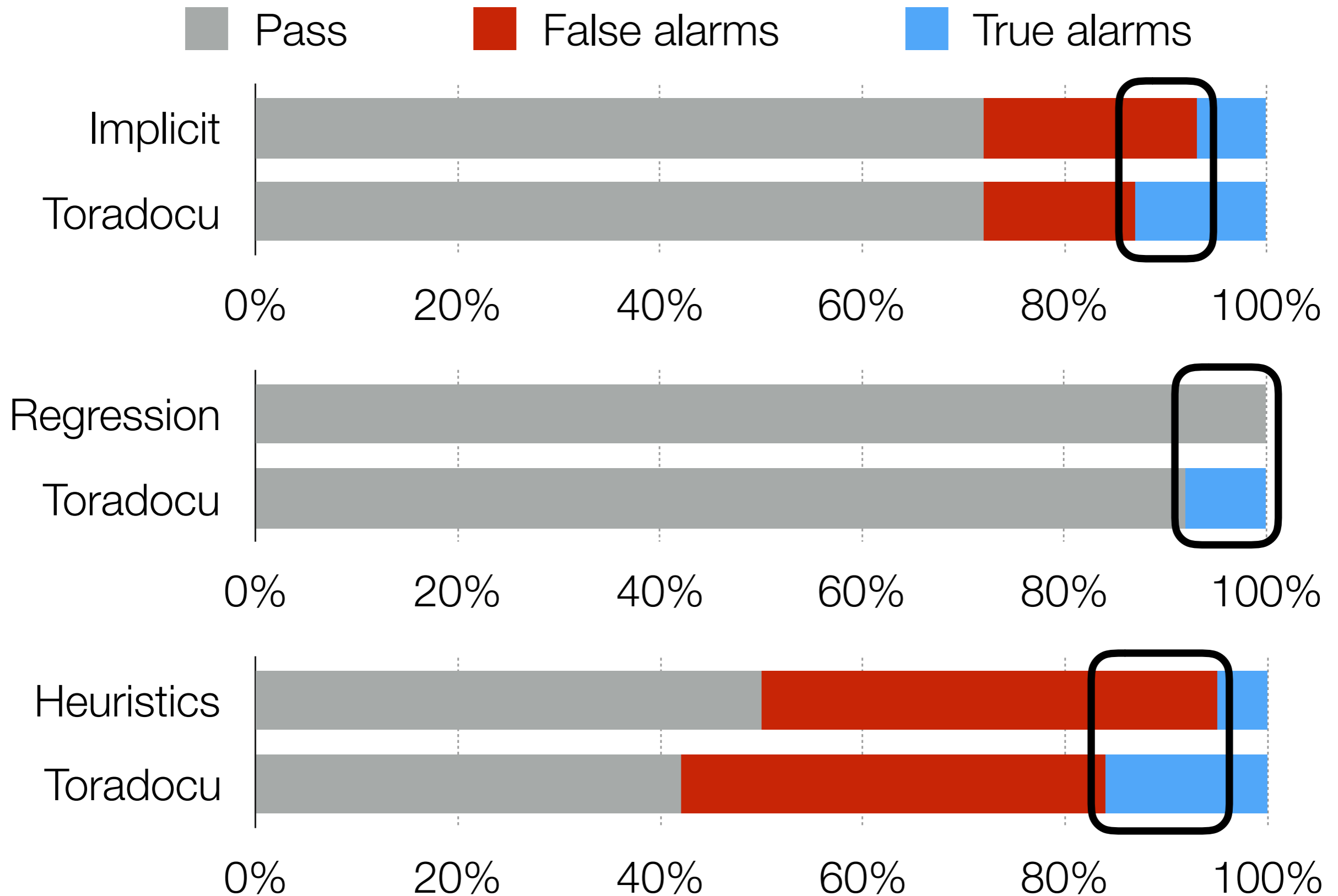
# True Alarms — Apache Commons



# True Alarms — Apache Commons



# True Alarms — Apache Commons



# What's next?

## **Oracles for exceptions on arrays/containers elements**

*@throws* IllegalArgumentException if the array contains null

# What's next?

## **Oracles for exceptions on arrays/containers elements**

*@throws* IllegalArgumentException if the array contains null

## **input values**

*@param* foo must not be null

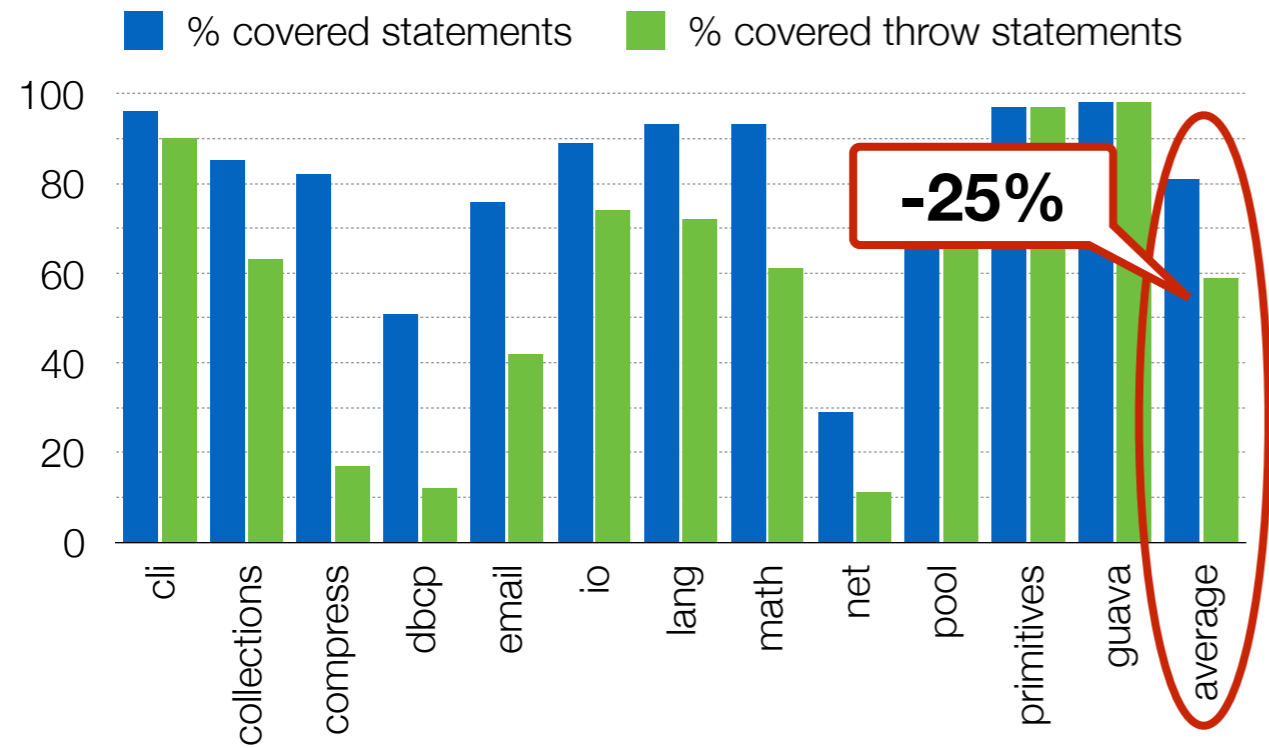
```
void sort(@NotNull int[] a)
```

## **return values**

*@return* the size of the array

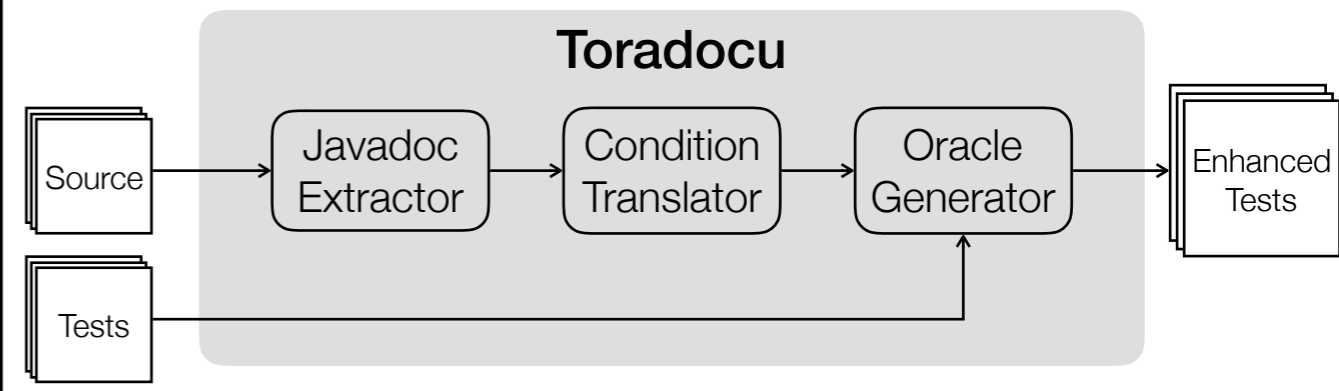
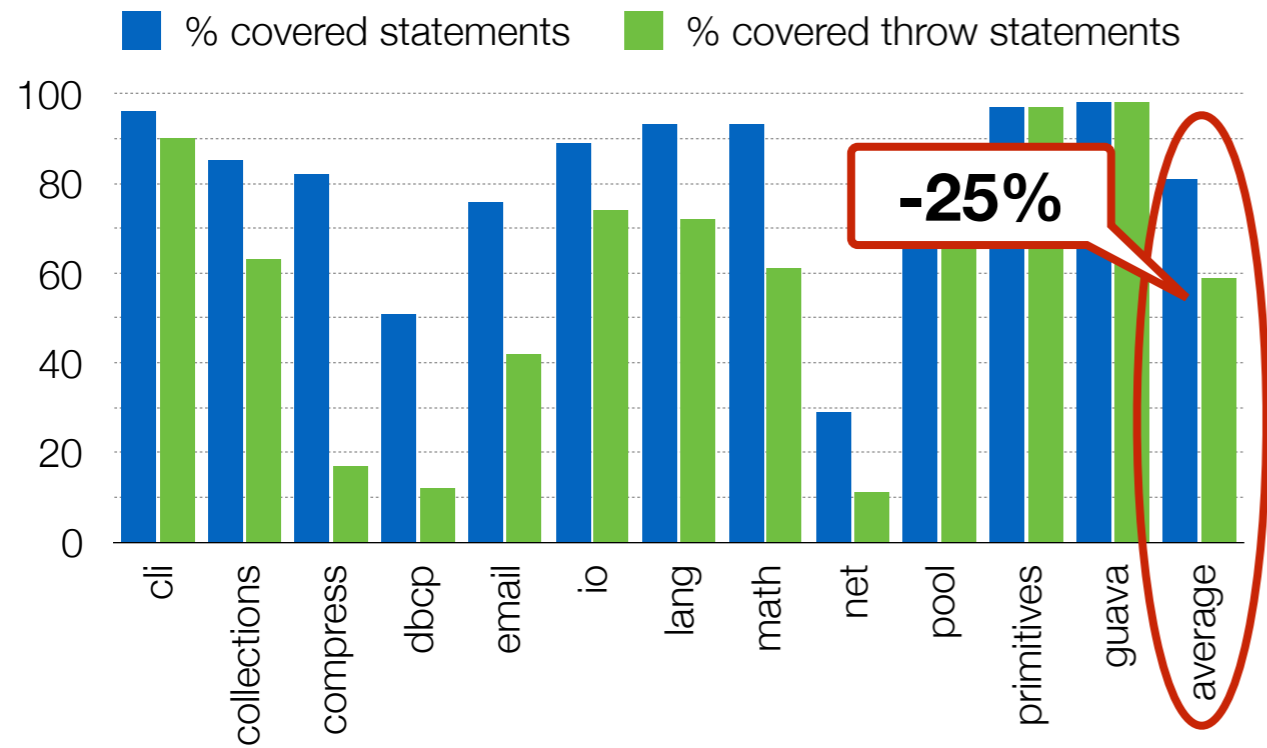
**...more**

# Testers under-test exceptions

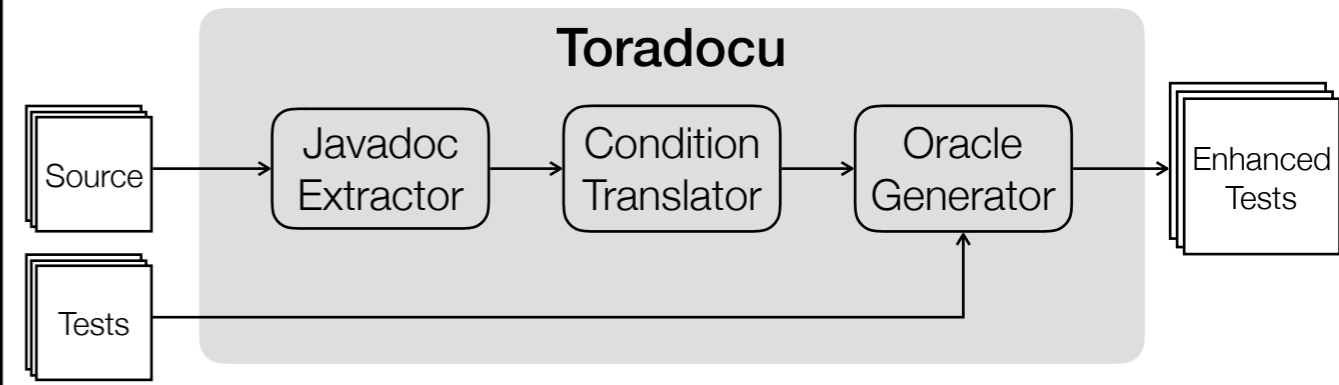
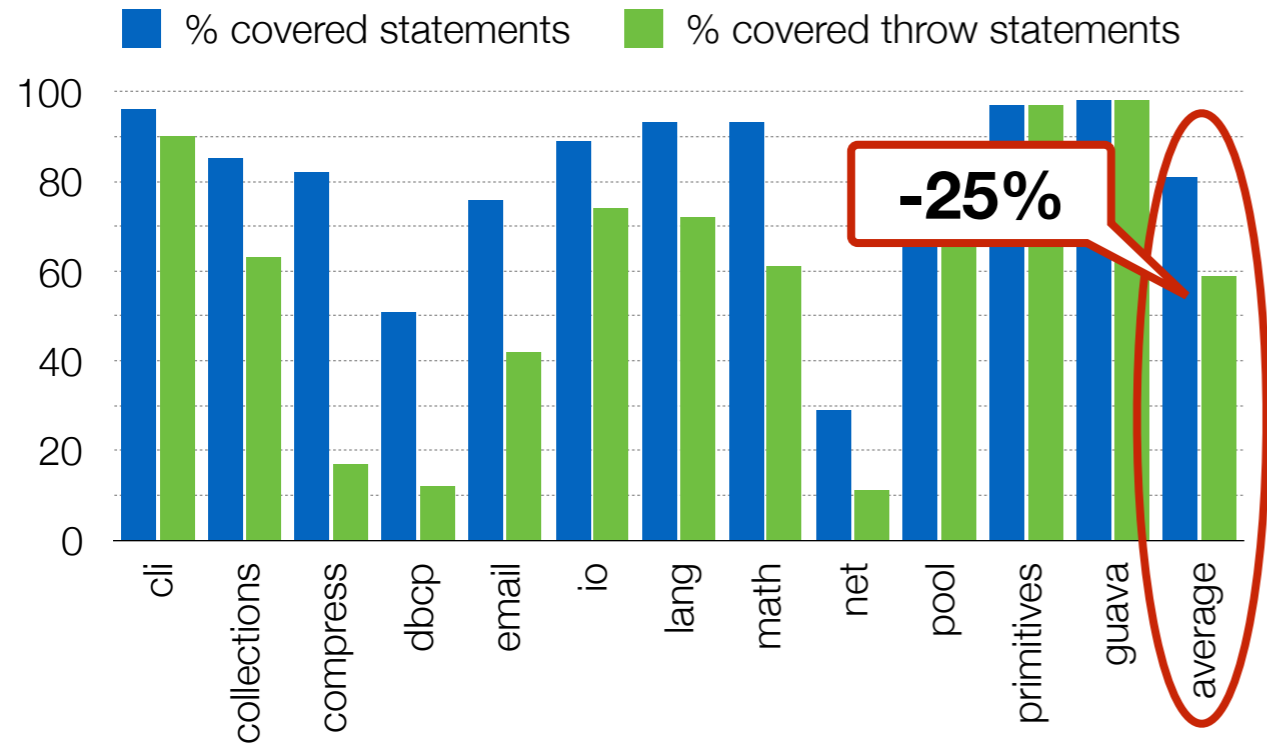




# Testers under-test exceptions



# Testers under-test exceptions



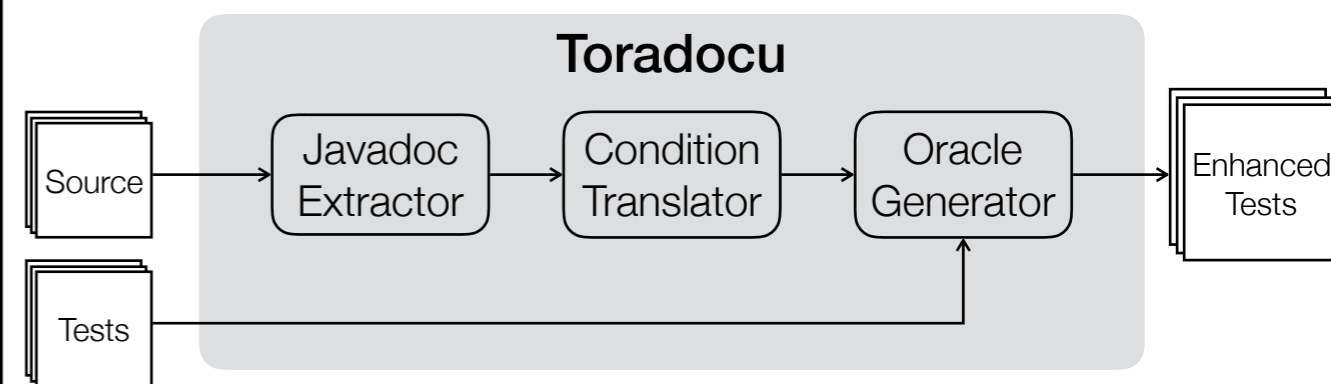
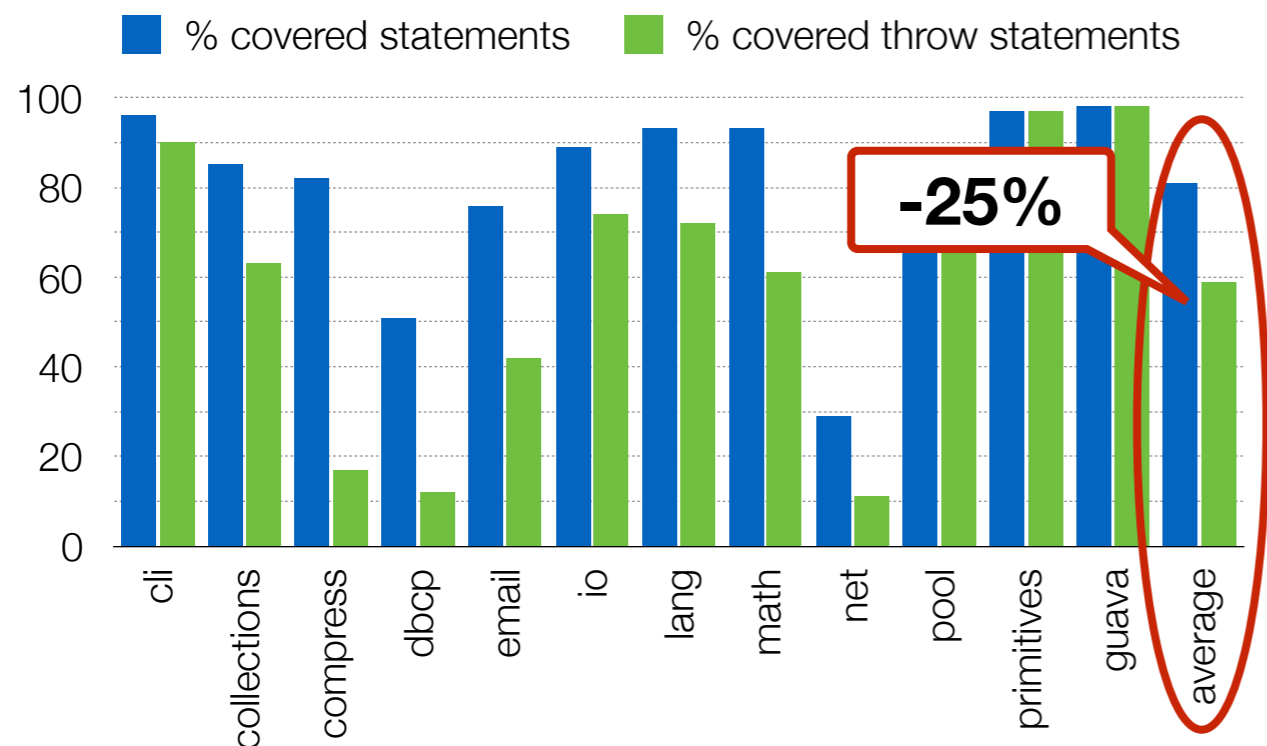
# From NL to Java Conditions

■ Correct 
 ■ Partially Correct 
 ■ Wrong 
 ■ Missing

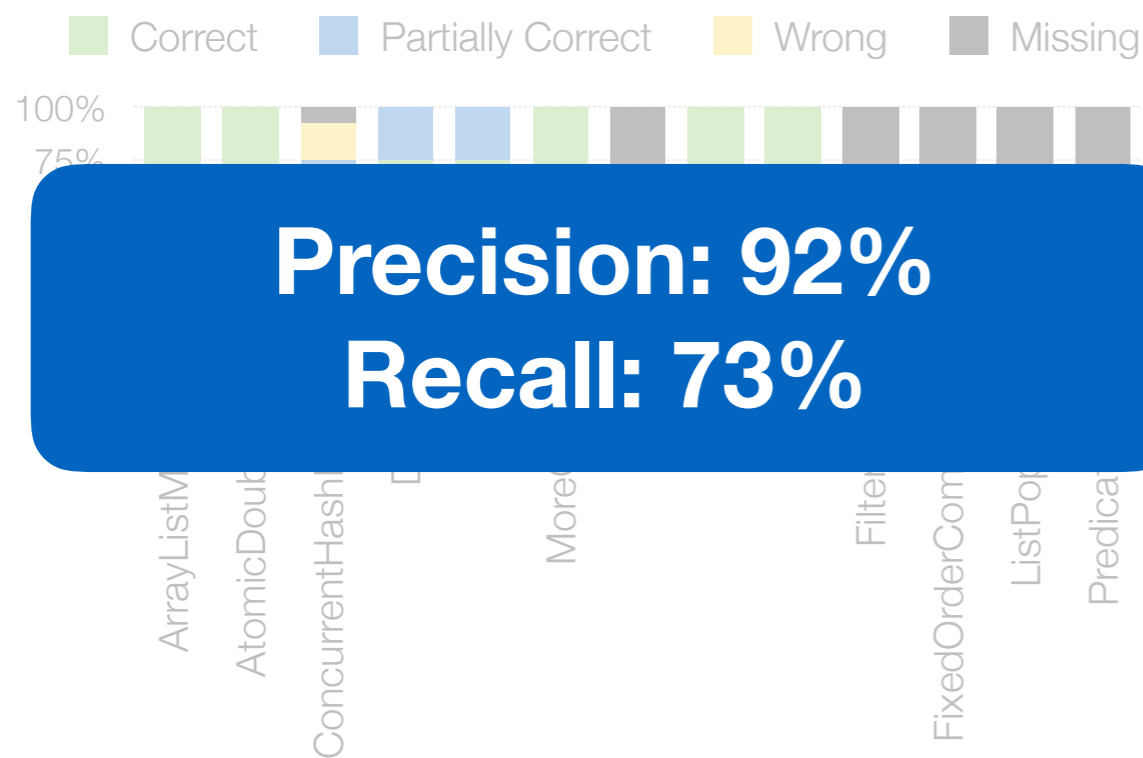


**Precision: 92%**  
**Recall: 73%**

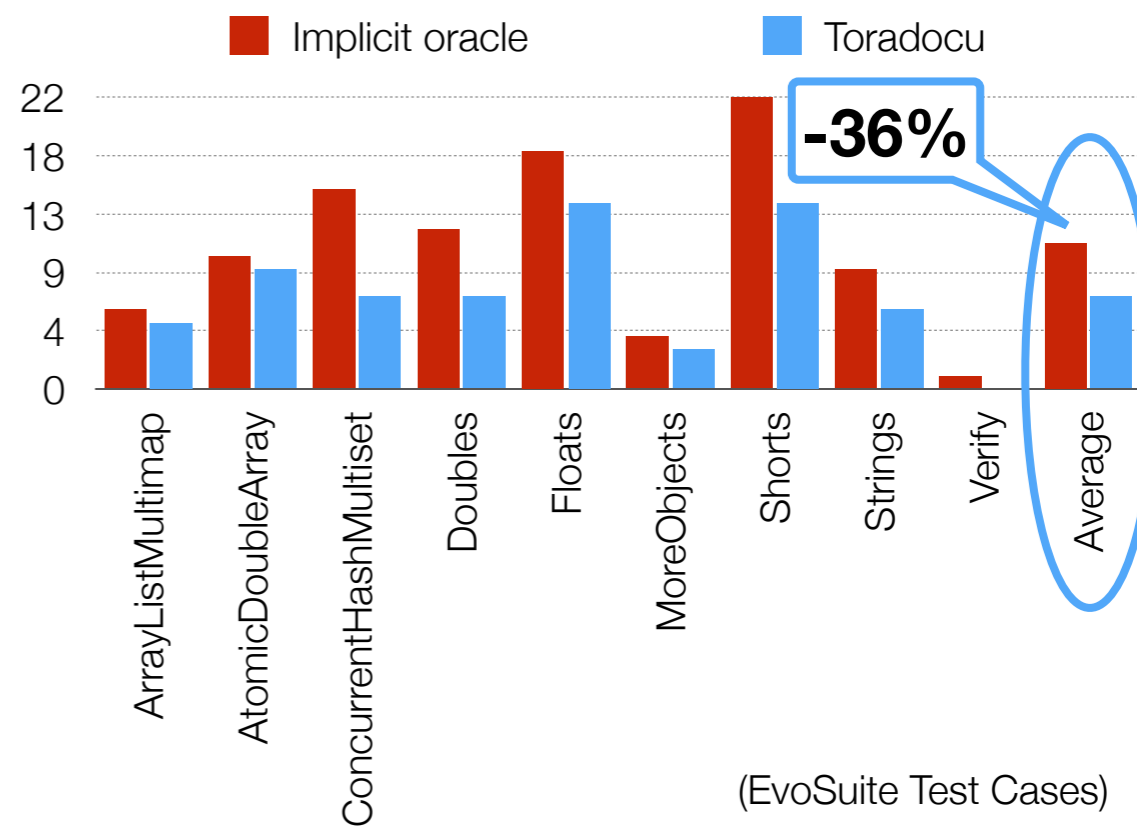
# Testers under-test exceptions



# From NL to Java Conditions

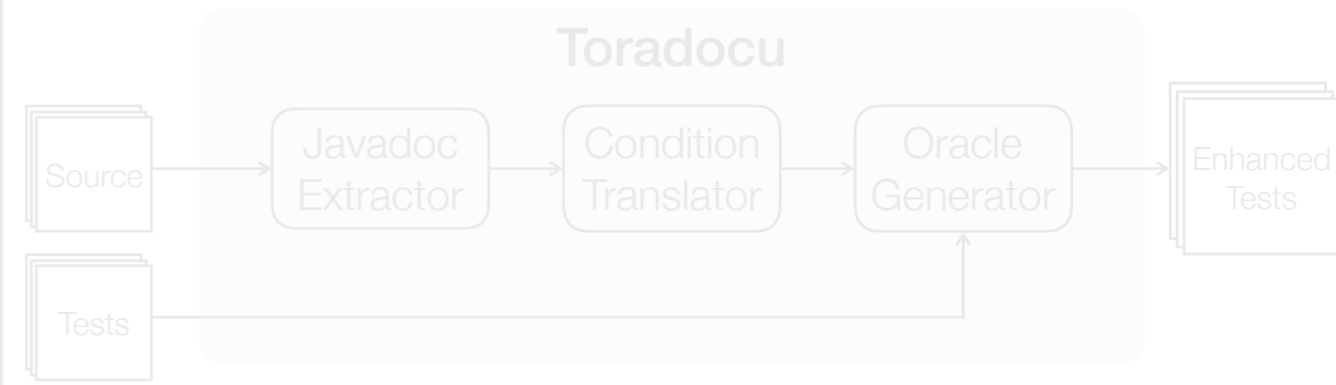
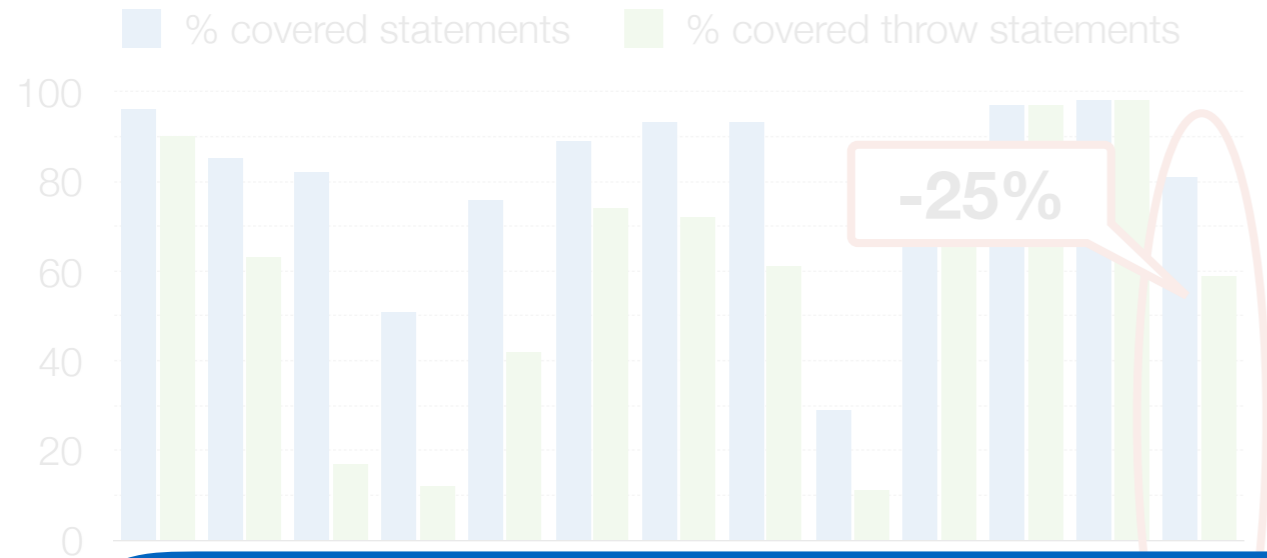


# False Alarms — Google Guava

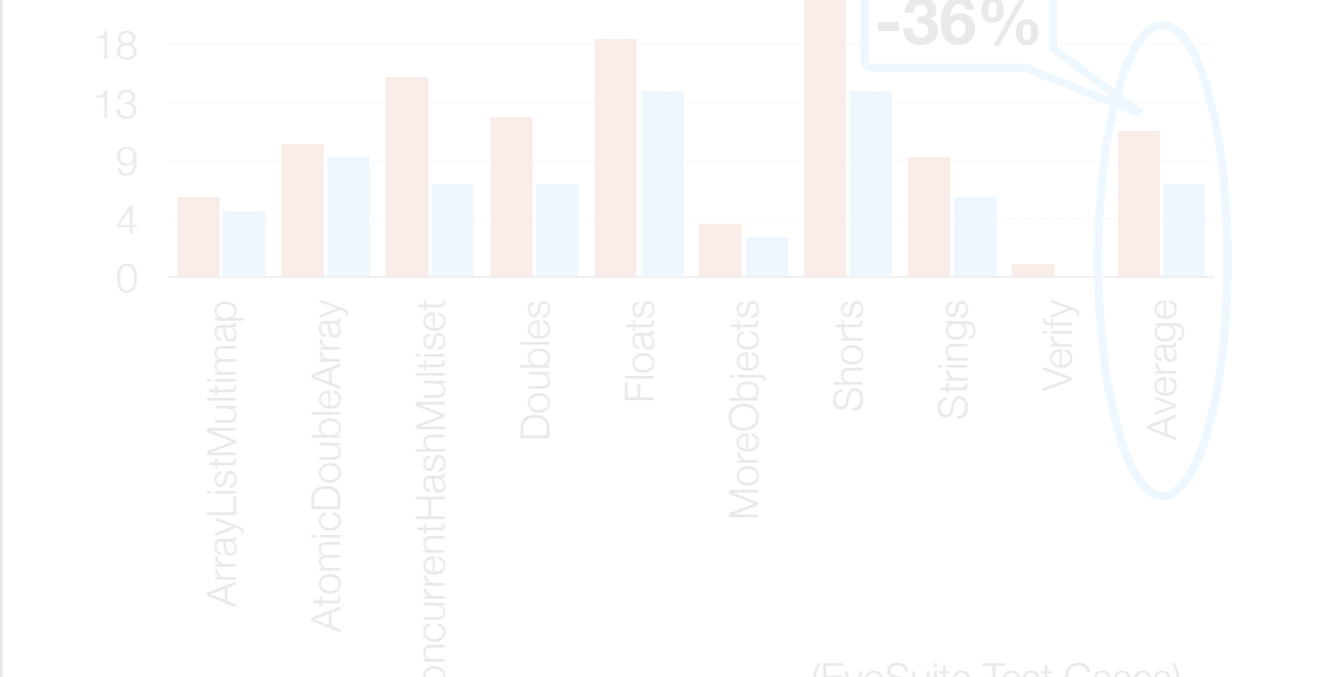
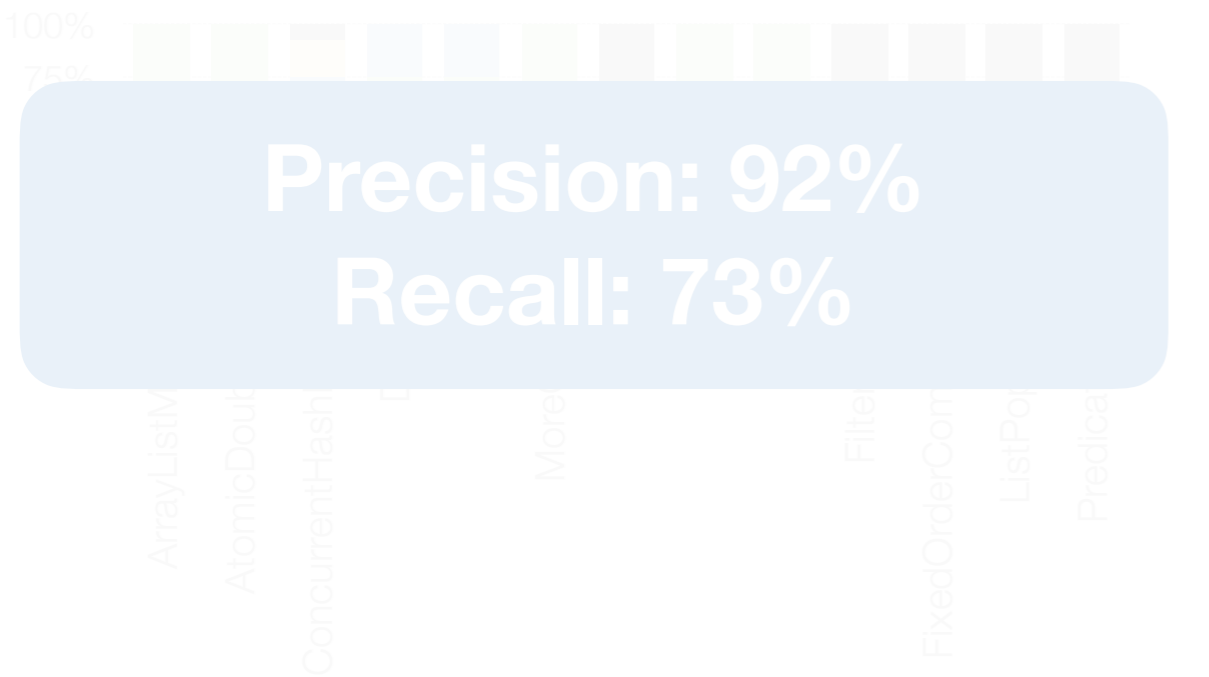


(EvoSuite Test Cases)

# Testers under-test exceptions



[github.com/albertogoffi/toradocu](https://github.com/albertogoffi/toradocu)  
 alberto.goffi@usi.ch





# Related Work

## Heuristics

JCrasher, Crash'n'Check (Csallner, and Smaragdakis. ICSE '05)

Randoop (Pacheco, Lahiri, Ernst, and Ball. ICSE '07)

## Specifications

ASTOOT (Doong, and Frankl. TOSEM '94)

Models, contracts, ...

## Properties

Cross-checking oracles (Carzaniga, Goffi, Gorla, Mattavelli, and Pezzè. ICSE '14)

Metamorphic testing (Chen, Kuo, Tse, and Zhou. STEP '13)

Symmetric testing (Gotlieb. ISSRE '03)

## Natural Language Documentation

@tComment (Tan, Marinov, Tan, and Leavens. ICST '12)

aComment (Tan, Zhou, and Padioleau. ICSE '11)

iComment (Tan, Yuan, Krishna, and Zhou. SOSPP '07)

# When does Toradocu fail?

## **Conditions on the elements in a container**

- if collection or any of its elements is null
- if any element in the arrays is null
- if notes has fewer than 2 elements

## **Specific checks**

- if the runtime type of the specified array is not a supertype of the runtime type of every element in this collection
- if  $x > 0$

## **Missing predicate translation (No output)**

- if the charset is not valid
- if closed

# Examples

@throws NullPointerException if the predicate or either closure is null

```
.. ifClosure(Predicate predicate, Closure trueClosure, Closure falseClosure) {...}
```

```
==> predicate == null || trueClosure == null || falseClosure == null
```

@throws UnsupportedOperationException if the comparator is locked

```
.. checkLocked() {...}
```

```
==> target.isLocked()
```