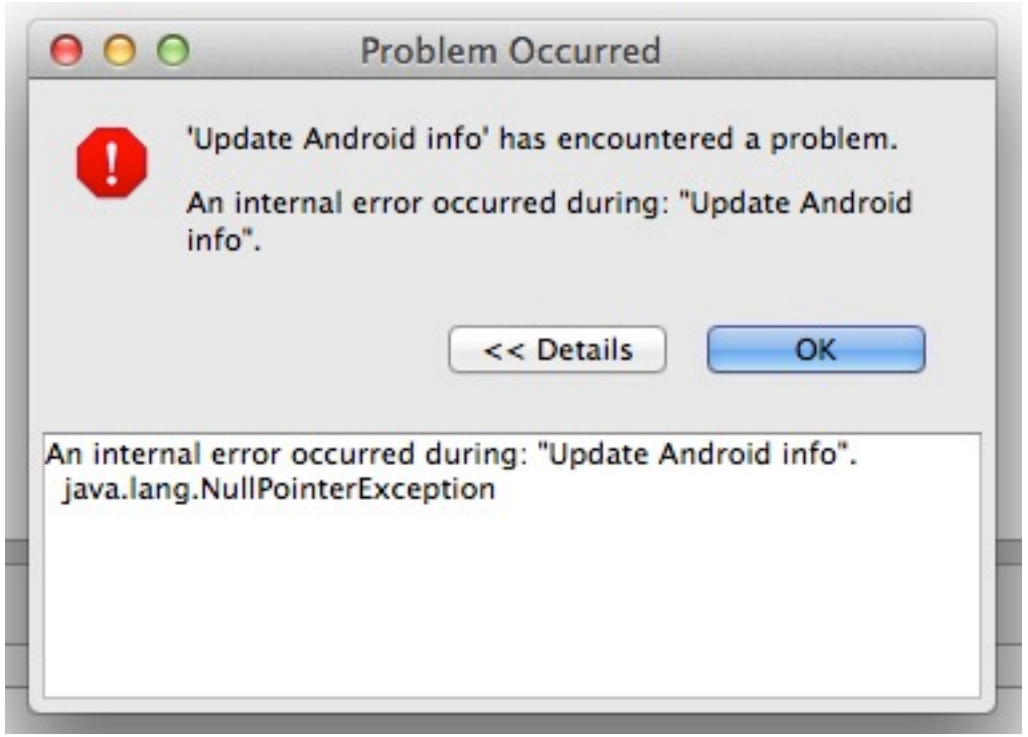
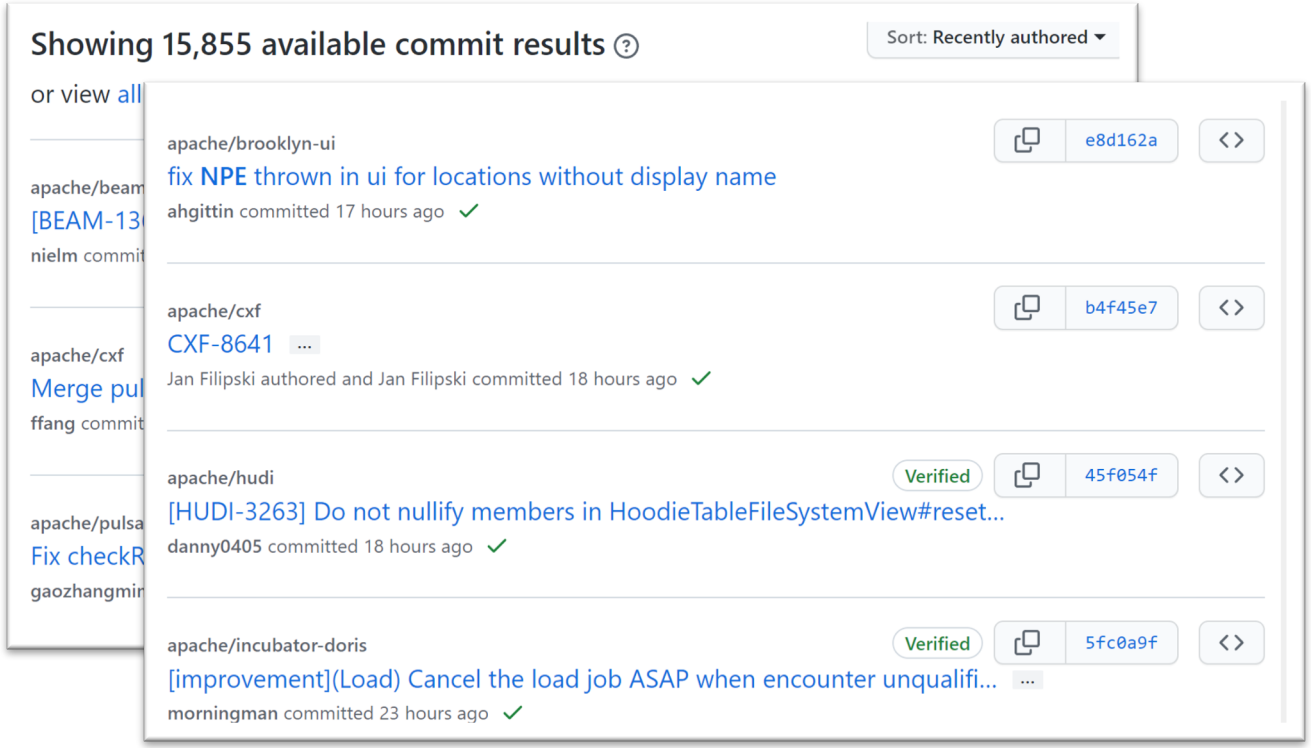
NPEX: Fixing Null Pointer Exception Without Tests

Junhee Lee, Seongjoon Hong, Hakjoo Oh
Korea University
22.02.11 @ 소프트웨어재난연구센터 워크샵

Java Null Pointer Exceptions (NPE)



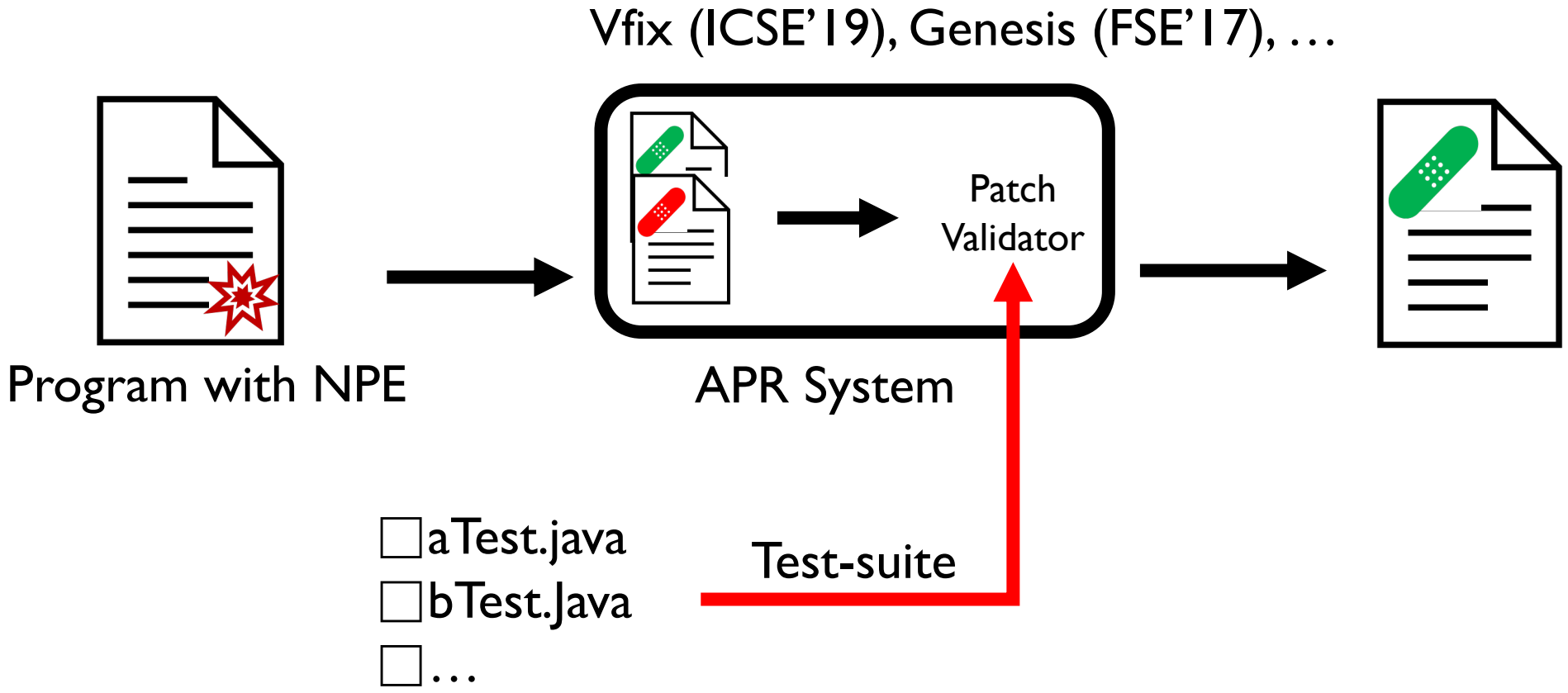
Common source of
Java application crashes



Daily NPE fixing commits from
Apache Foundation's GitHub Repositories

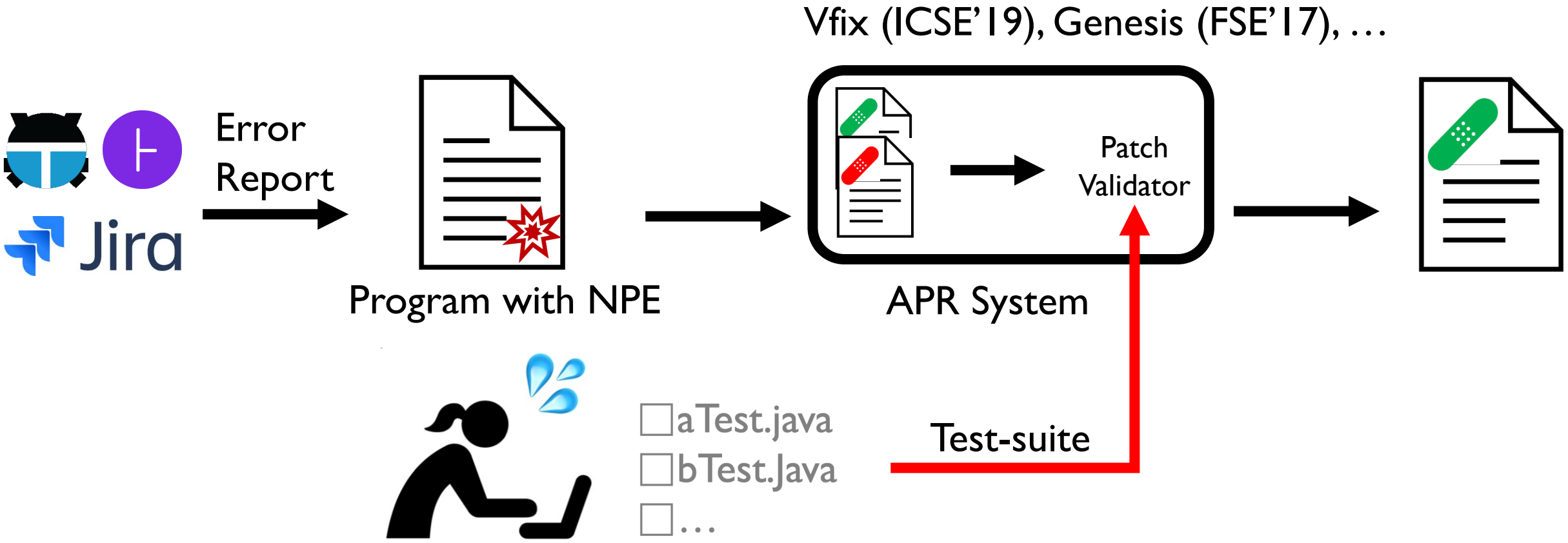
Left image from: https://jira.appcelerator.org/secure/attachment/47954/Screen%20Shot%202014-05-09%20at%2011.23.48%20AM.png

Existing Techniques for NPE Repair



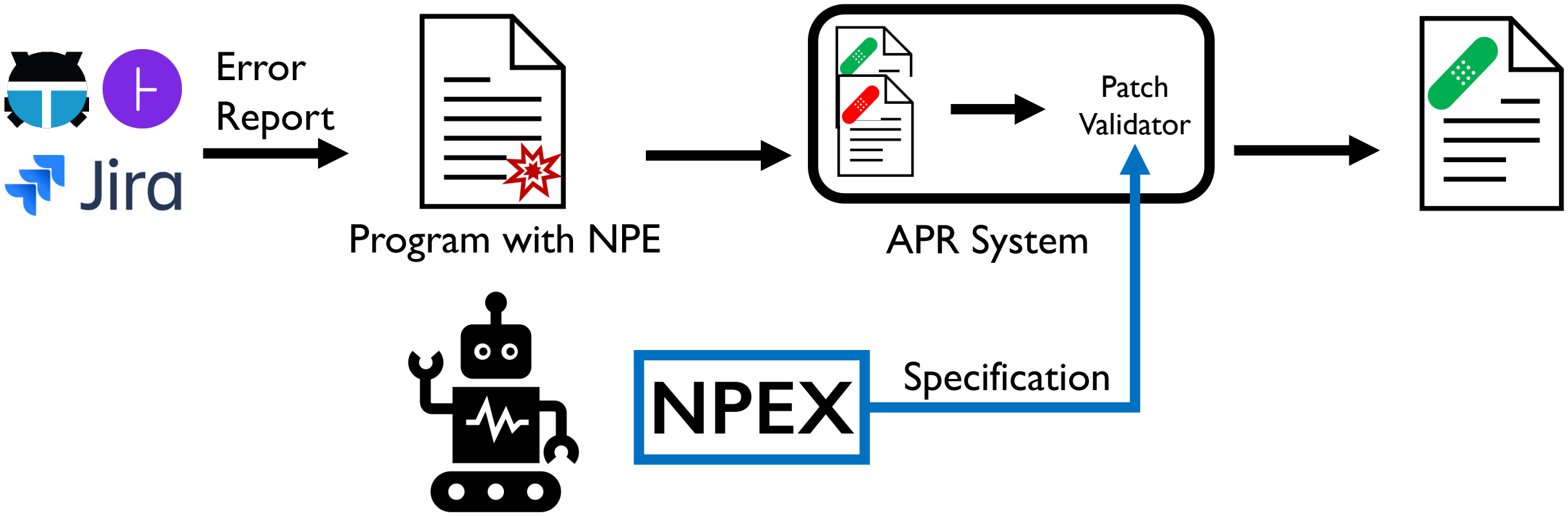
Highly depends on developer written test-suite

Existing Techniques for NPE Repair



Test-suite is not available when NPE is reported

Our Goal: Fixing NPE without Tests by Inferring Specification



Example: NPE in Apache Commons-Lang

```
public static String m(Object obj, String valueIfNull) {
    Class cls = obj.getClass();
    String name = cls.getCanonName();
    return name == null? valueIfNull : name;
}
```

Example: NPE in Apache Commons-Lang

```
public static String m(Object obj, String valueIfNull) {
    Class cls = obj.getClass();
    String name = cls.getCanonName();
    return name == null? valueIfNull : name;
}
```

```
If (obj == null)
    return valueIfNull;
Class cls = …;
```

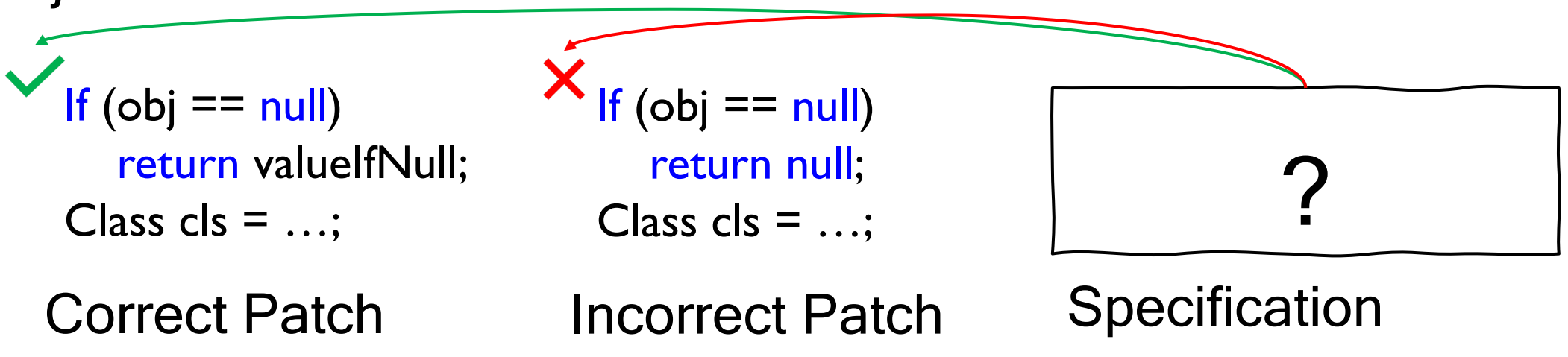
Correct Patch

```
If (obj == null)
    return null;
Class cls = …;
```

Incorrect Patch

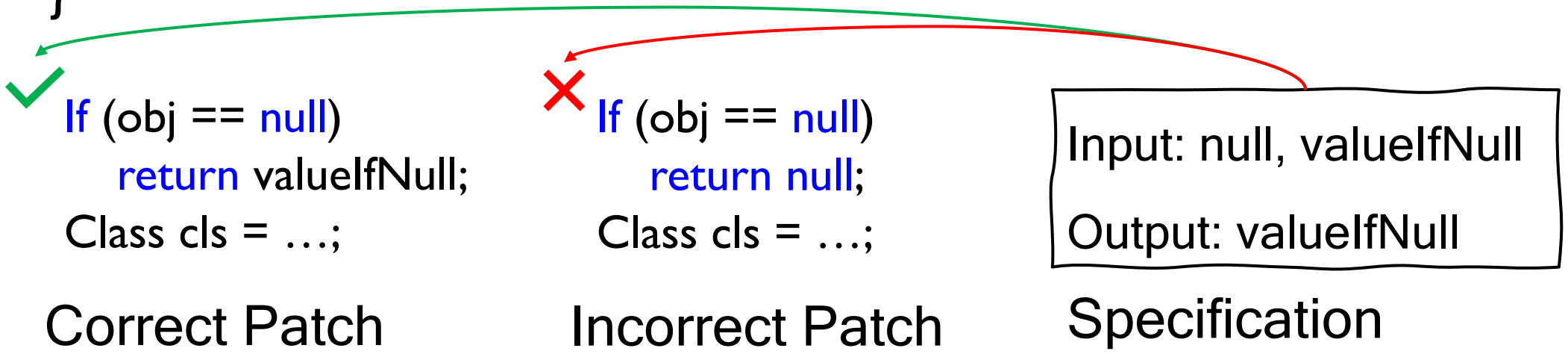
Problem: Inferring Specification

```
public static String m(Object obj, String valueIfNull) {
    Class cls = obj.getClass();
    String name = cls.getCanonName();
    return name == null? valueIfNull : name;
}
```



Problem: Inferring Specification

```
public static String m(Object obj, String valueIfNull) {
    Class cls = obj.getClass();
    String name = cls.getCanonName();
    return name == null? valueIfNull : name;
}
```



Our Idea (1) Learning Null Handling Model

Learn how developer handles NPE

Codebase

```
if (x == null ? true : x.isEmpty()) {
   throw new IOException(…);

new Object[] {
   x == null ? null : x.getClass(), … }

log.info("Conflicting …" +
x != null ? x.getCanonName() : null);

                …
```

Model : $Exp_{NPE} \rightarrow Exp_{NonNPE}$

```
null.isEmpty()          ↦  true
null.getClass()         ↦  null
null.getCanonName()     ↦  null

                …
```

Our Idea (1) Learning Null Handling Model

Learn how developer handles NPE

Codebase

```
if (x == null ? true : x.isEmpty()) {
   throw new IOException(…);

new Object[] {
   x == null ? null : x.getClass(), … }

log.info("Conflicting …" +
x != null ? x.getCanonName() : null);

                …
```

Model : $Exp_{NPE} \rightarrow Exp_{NonNPE}$

```
null.isEmpty()          ↦  true
null.getClass()         ↦  null
null.getCanonName()     ↦  null

                …
```

Our Idea (1) Learning Null Handling Model

Learn how developer handles NPE

Codebase

```
if (x == null ? true : x.isEmpty()) {
   throw new IOException(…);

new Object[] {
   x == null ? null : x.getClass(), … }

log.info("Conflicting …" +
x != null ? x.getCanonName() : null);

                …
```

Model : $Exp_{NPE} \rightarrow Exp_{NonNPE}$

```
null.isEmpty()          ↦  true
null.getClass()         ↦  null
null.getCanonName()     ↦  null

                …
```

Our Idea (1) Learning Null Handling Model

Learn how developer handles NPE

Codebase

```
if (x == null ? true : x.isEmpty()) {
   throw new IOException(…);

new Object[] {
   x == null ? null : x.getClass(), … }

log.info("Conflicting …" +
x != null ? x.getCanonName() : null);

                …
```

Model : $Exp_{NPE} \rightarrow Exp_{NonNPE}$

```
null.isEmpty()          ↦  true
null.getClass()         ↦  null
null.getCanonName()     ↦  null

                …
```

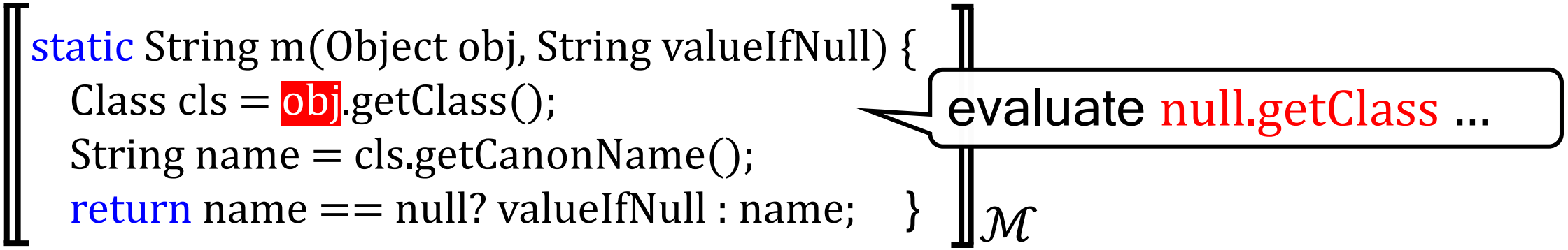
Our Idea (2) Spec. Inference by Model

Using the model, infer non-NPE output for NPE input

Our Idea (2) Spec. Inference by Model

Using the model, infer non-NPE output for NPE input

- Symbolically execute m with NPE input (null, valueIfNull)



Our Idea (2) Spec. Inference by Model

Using the model, infer non-NPE output for NPE input

- Symbolically execute m with NPE input (null, valueIfNull)
  1. Interpret null.getClass() as null

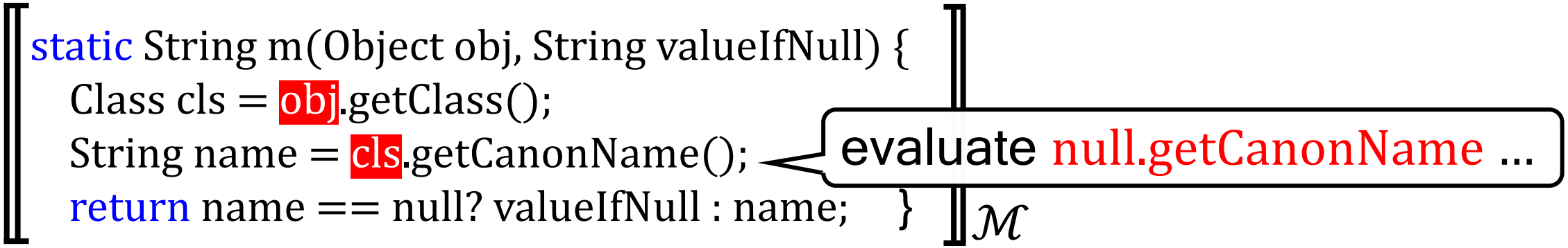
Using learned model
null.getClass() ↦ null

$[\![$
```
static String m(Object obj, String valueIfNull) {
    Class cls = obj.getClass();
    String name = cls.getCanonName();
    return name == null? valueIfNull : name;    }
```
$]\!]_{\mathcal{M}}$

Our Idea (2) Spec. Inference by Model

Using the model, infer non-NPE output for NPE input

- Symbolically execute m with NPE input (null, valueIfNull)
  1. Interpret null.getClass() as null
  2. Interpret null.getCanonName() as null



Our Idea (2) Spec. Inference by Model

Using the model, infer non-NPE output for NPE input

- Symbolically execute m with NPE input (null, valueIfNull)
  1. Interpret null.getClass() as null
  2. Interpret null.getCanonName() as null

$$[\![\text{static String m(Object obj, String valueIfNull) \{ Class cls = obj.getClass(); String name = cls.getCanonName(); return name == null? valueIfNull : name; \}}]\!]_{\mathcal{M}} =$$

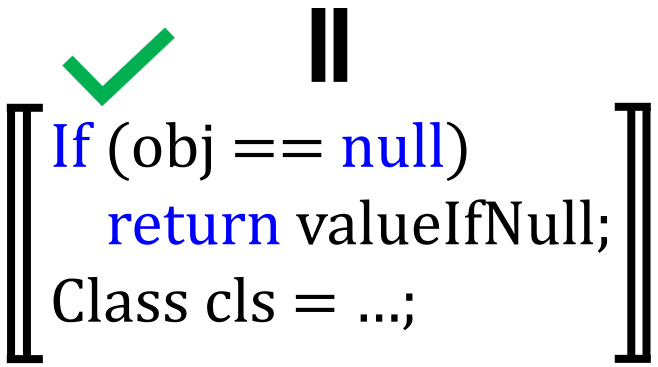
Pre-Condition	Post-State
obj = null	return = valueIfNull

Patch Validation by Spec.

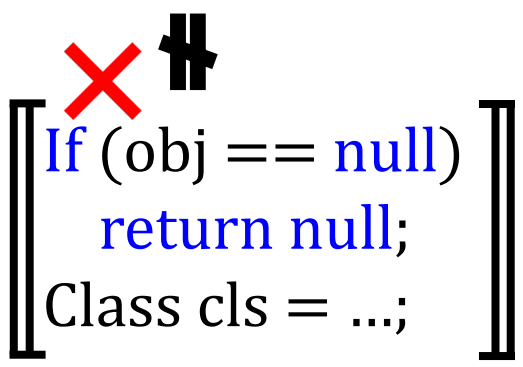
Check semantic equivalence

Pre-Condition	Post-State
obj = null	return = valueIfNull
obj ≠ null ∧ g(f(obj)) = null	return = valueIfNull
obj ≠ null ∧ g(f(obj)) ≠ null	return = g(f(obj))

f = getClass
g = getCanonName



Correct Patch



Incorrect Patch

Challenge in Learning Null Model

- Conflicts in same null expression
  - e.g., multiple, but different handling for null.toString()

```
return retType != null ? retType.toString() : null;
```

```
this.f = obj != null ? obj.toString() : "";
```

- No null handling for custom method invocation
  - e.g., null.getCustomerInfo()

Our Solution

Design Features

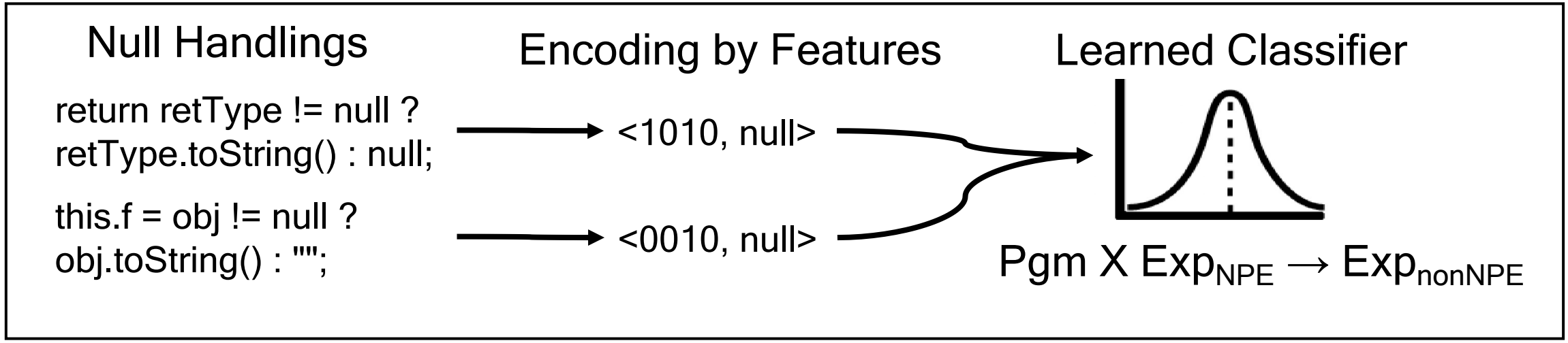
Context Features (for precision)

- Syntactic features of NPE expression (e.g., is field assignment?)

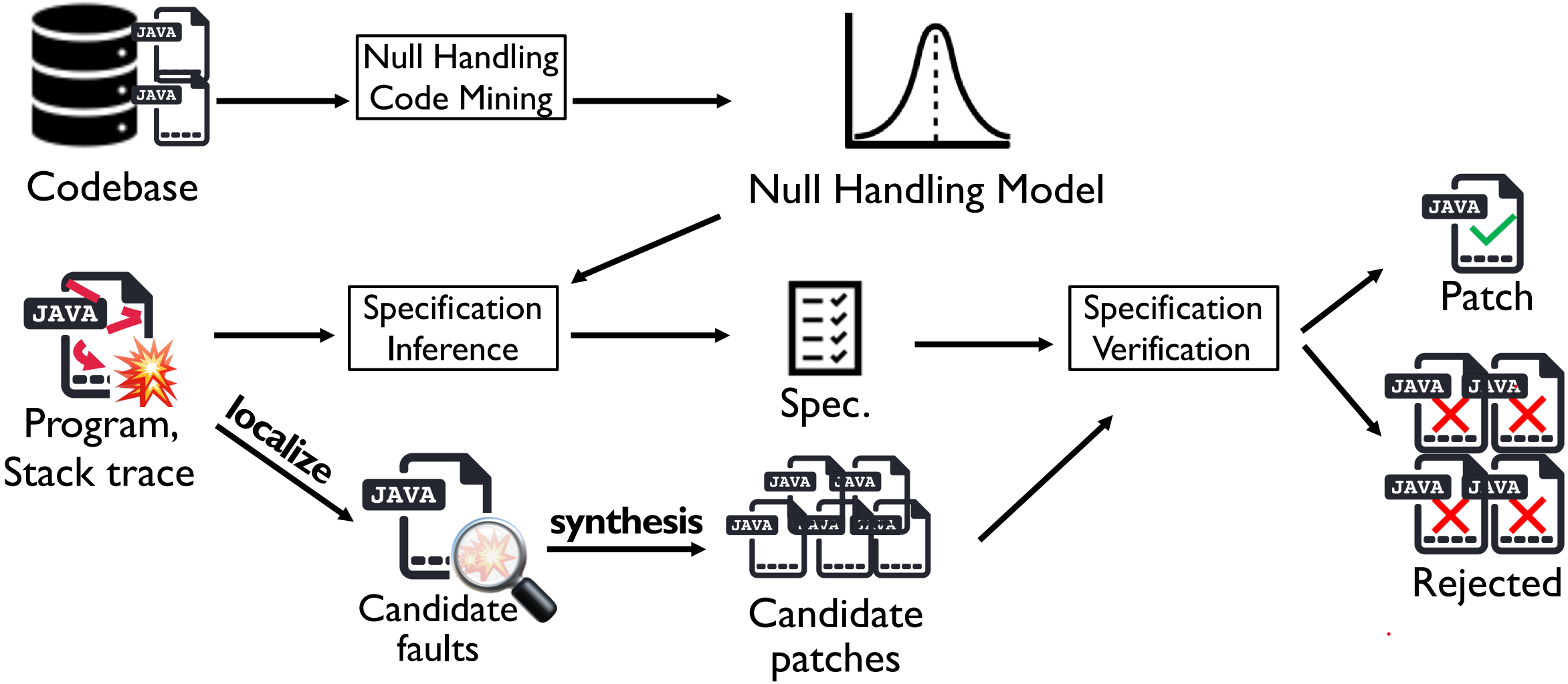
Name Features (for generalization)

- Top 20 words in methods: "get", 'set", "is", ...

Overview of Learning



NPEX System Overview



Evaluation Setup

- Benchmarks: 119 NPE bugs
  - VFix: 30
  - Genesis: 16
  - BEARS: 14
  - Ours: 59

- Tool Setup
  - NPEX: no test-case (stack-trace only)
  - VFix (ICSE'19): with test-case
  - Genesis (FSE'17): with test-case

- Patch Correctness Criteria
  - Semantic equivalence with developers' fix

RQ1: Comparison with Other Tools for Fixing NPE

Benchmarks		NPEX		Genesis		VFix	
Sources	#Bugs	#Run	FixRate	#Run	FixRate	#Run	FixRate
VFix	30	30	63%	0	-	30	67%
Genesis	16	16	63%	16	50%	2	50%
Bears	14	14	43%	14	21%	2	0%
Ours	59	59	44%	59	16%	21	14%
Total	119	119	**51%**	89	**23%**	55	**44%**

RQ1: Comparison with Other Tools for Fixing NPE (vs Genesis)

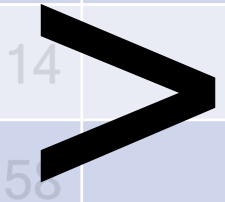
Benchmarks		NPEX		Genesis		VFix	
Sources	#Bugs	#Run	FixRate	#Run	FixRate	#Run	FixRate
VFix	30	30	63%	0	-	30	67%
Genesis	16	16	63%	16	50%	2	50%
Bears	14	14	43%	14	21%	2	0%
Ours	59	59	44%	58	16%	21	14%
Total	119	119	**51%**	89	**23%**	55	44%

>

NPEX outperformed Genesis without tests

RQ1: Comparison with Other Tools for Fixing NPE (vs VFix)

Benchmarks		NPEX		Genesis		VFix	
Sources	#Bugs	#Run	FixRate	#Run	FixRate	#Run	FixRate
VFix	30	30	63%	0	0%	30	67%
Genesis	16	16	63%	16	31%	2	50%
Bears	14	14	43%	14	21%	2	0%
Ours	59	59	44%	55	15%	21	14%
Total	119	119	**51%**	61	23%	55	**44%**



RQ2: Effectiveness of Spec. Inference

Benchmarks		NPEX		$NPEX_{Base}$	
Sources	#Bugs	#Run	FixRate	#Run	FixRate
VFix	30	30	63%	30	23%
Genesis	16	16	63%	16	31%
Bears	14	14	43%	14	21%
Ours	59	59	44%	59	15%
Total	119	119	**51%**	119	**24%**

Uses test-cases without spec. inference

Specification Inference is effective

Case Study: Inferred Spec. vs Test-Suite

NPEX often inferred better specification than test-suite

```
List getMembers(Class c, ...) {
   List members = new List();
   while (c != Object.class) {
      members.add(cl.getFields());
      c = c.getSuperclass();
   }
   return members;
}
```

NPE in Aries-jpa

	Input (condition)	Output
Test	c != null & c.superclass = null & c.field = {}	Empty list
NPEX-inferred	c = null	Empty list
	c != null & c.superclass = null	Any

Test-suite vs NPEX-Inferred Spec.

Case Study: Inferred Spec. vs Test-Suite

NPEX often inferred better spec

```
List getMembers(Class c, ...) {
   List members = new List();
   while (c != Object.class) {
      members.add(cl.getFields());
      c = c.getSuperclass();
   }
   return members;
}
```

NPE in Aries-jpa

```
interface class I {}
@Test void test() {
   assert(getMembers(I.class) == []); }
```

	Input (Condition)	Output
Test	c != null & c.superclass = null & c.field = {}	Empty list
NPEX-inferred	c = null	Empty list
	c != null & c.superclass = null	Any

Test-suite vs NPEX-Inferred Spec.

Case Study: Inferred Spec. vs Test-Suite

NPEX often inferred better specification than test-suite

```
List getMembers(Class c, ...) {
   List members = new List();
   while (c != Object.class) {
      if (c == null)
         return new List();
      members.add(cl.getFields());
      c = c.getSuperclass();
   }
   return members;                }
```

VFix & Genesis's patch

(test-overifitted)

	Input (condition)	Output
Test	c != null & c.superclass = null & c.field = {}	Empty list
NPEX-inferred	c = null	Empty list
	c != null & c.superclass = null	Any

Test-suite vs NPEX-Inferred Spec.

Case Study: Inferred Spec. vs Test-Suite

NPEX often inferred better specification than test-suite

```
List getMembers(Class c, …) {
   List members = new List();
   while (c != Object.class) {
      if (c == null)
         break;
      members.add(cl.getFields());
      c = c.getSuperclass();
   }
   return members;                }
```

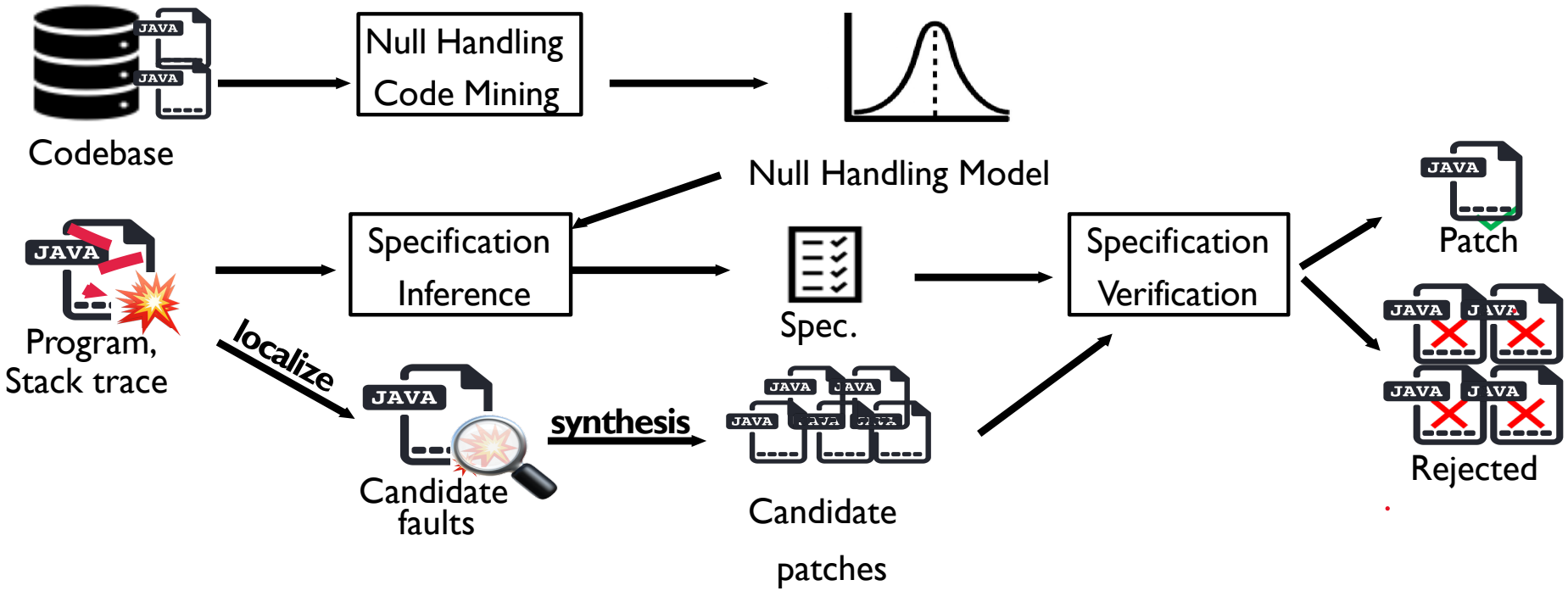
NPEX's patch (correct)

	Input (condition)	Output
Test	c != null & c.superclass = null & c.field = {}	Empty list
NPEX-inferred	c = null	Empty list
	c != null & c.superclass = null	Any

Test-suite vs NPEX-Inferred Spec.

Summary

- Fixing NPE requires specification, but test cases are typically unavailable.
- We present a method for fixing NPE without tests by inferring specification



Limitation of NPEX

NPEX assumes that no other faults exists than missing null handling

```
IterReader(Iterable iterable) {



    this(iterable.iterator()); // NPE here
}
IterReader(Iterator iterator) {
    if (iterator == null)
        throw new NPE(); // Another Fault
    this.iterator = iterator;
}
```

```
IterReader(Iterable iterable) {
    if (iterable == null)
        throw new IllegalArgExn();
    this(iterable.iterator());
}
IterReader(Iterable iterable) {
    if (iterator == null)
        throw new IllegalArgExn();
    this.iterator = iterator;
}
```

Incorrectly Inferred Specification

$$iterable = null \rightarrow return = NPE$$