Università della Svizzera italiana

Faculty of Informatics

Exploiting Intrinsic Redundancy to Automatically Generate Test Oracles

Alberto Goffi and Andrea Mattavelli

Università della Svizzera italiana (USI), Switzerland

In collaboration with:

Antonio Carzaniga

Alessandra Gorla

Mauro Pezzè

Paolo Tonella

Redundancy

Software Redundancy

Software Redundancy

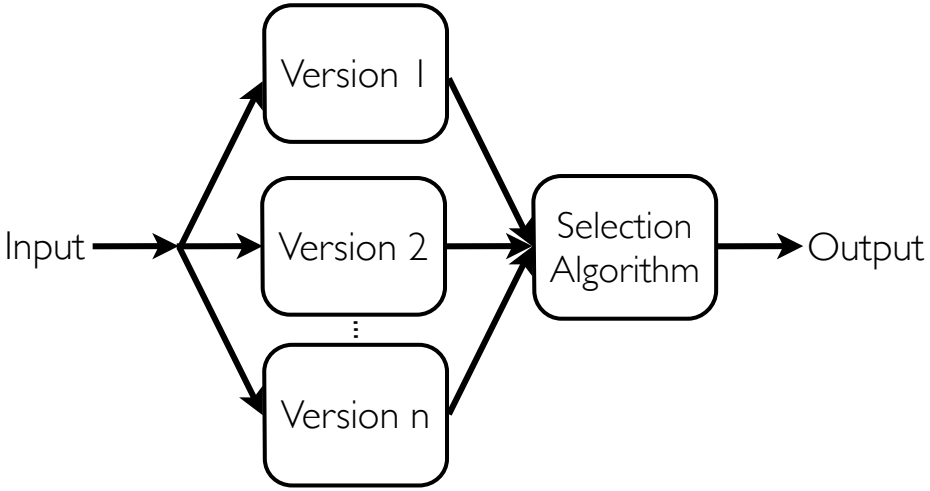
> A system is redundant when it is able to perform **equivalent functionalities** by executing **different code**.

Software Redundancy

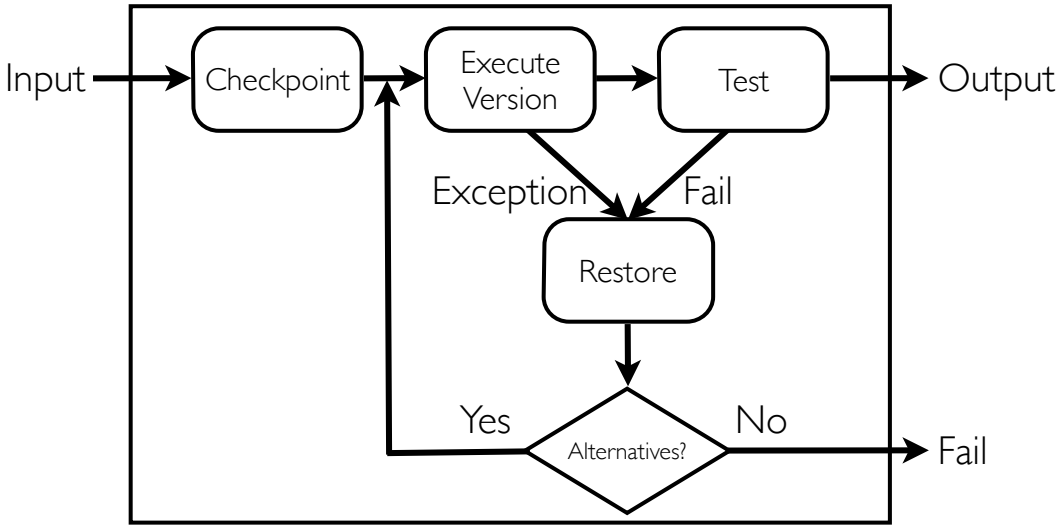
> A system is redundant when it is able to perform **equivalent functionalities** by executing **different code**.

- compute **identical results**
- lead to **identical states**

Software Redundancy

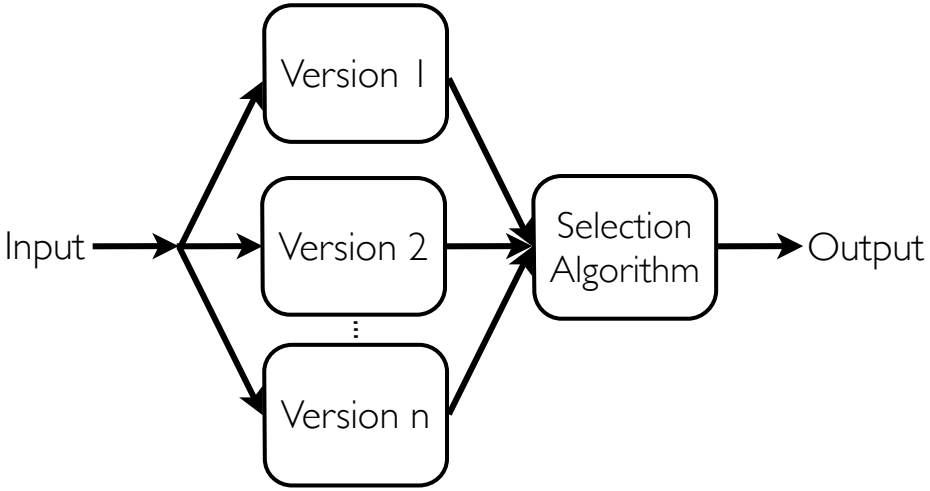


N-version

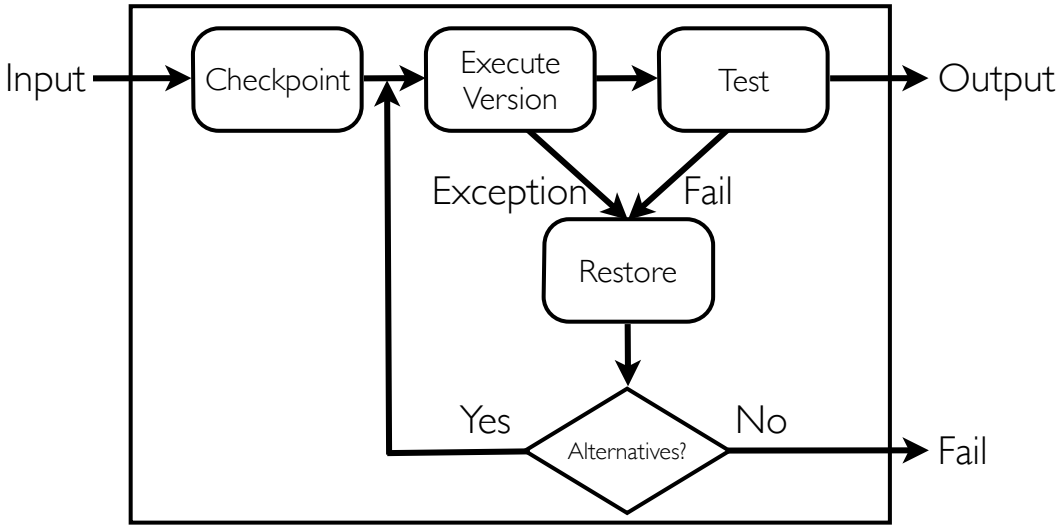


Recovery Blocks

Software Redundancy
Deliberate



N-version



Recovery Blocks

Software Redundancy
Intrinsic

Software Redundancy
Intrinsic

Joda-Time

```
DateTime t = new DateTime();
//...
//get the beginning of the day for time t
DateTime beginDay = t.millisOfDay().withMinimumValue();
```

Software Redundancy
Intrinsic

Joda-Time

```
DateTime t = new DateTime();
//...
//get the beginning of the day for time t
DateTime beginDay = t.millisOfDay().withMinimumValue();
                  = t.toDateMidnight().toDateTime();
                  = t.withTimeAtStartOfDay();
```

Software Redundancy
Intrinsic

Joda-Time

```
DateTime t = new DateTime();
//...
//get the beginning of the day for time t
DateTime beginDay = t.millisOfDay().withMinimumValue();
                  = t.toDateMidnight().toDateTime();
                  = t.withTimeAtStartOfDay();
```

Google Guava

```
MultiMap m = new MultiMap();
//…
//check if element is already in map
if (m.contains(x))
```

Software Redundancy
Intrinsic

Joda-Time

```
DateTime t = new DateTime();
//...
//get the beginning of the day for time t
DateTime beginDay = t.millisOfDay().withMinimumValue();
                  = t.toDateMidnight().toDateTime();
                  = t.withTimeAtStartOfDay();
```

Google Guava

```
MultiMap m = new MultiMap();
//…
//check if element is already in map
if (m.contains(x))
 if (m.elementSet().contains(x))
 if (m.count(x) > 0)
```

Software Redundancy
Intrinsic

Joda-Time

```
DateTime t = new DateTime();
//...
//get the beginning of the day for time t
DateTime beginDay = t.millisOfDay().withMinimumValue();
                  = t.toDateMidnight().toDateTime();
                  = t.withTimeAtStartOfDay();
```

2 LOC
(~0.1%)

Google Guava

```
MultiMap m = new MultiMap();
//…
//check if element is already in map
if (m.contains(x))
 if (m.elementSet().contains(x))
 if (m.count(x) > 0)
```

0 LOC
(0.0%)

Software Redundancy
Intrinsic

Joda-Time

SWT

4700+

equivalences

GraphStream

Exploiting Redundancy

Automatic repair

Test oracles

Security

Exploiting Redundancy

Automatic repair

Manual identification of equivalence

Test oracles

Security

Exploiting Redundancy

Automatic repair

Manual identification of equivalence

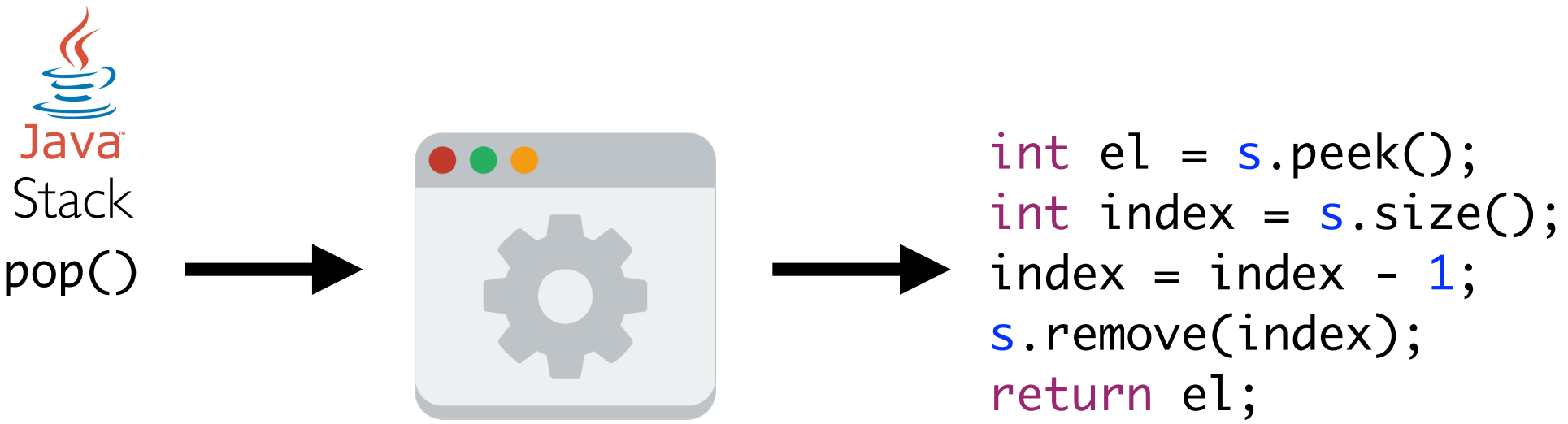
Test oracles

… is the main cost!

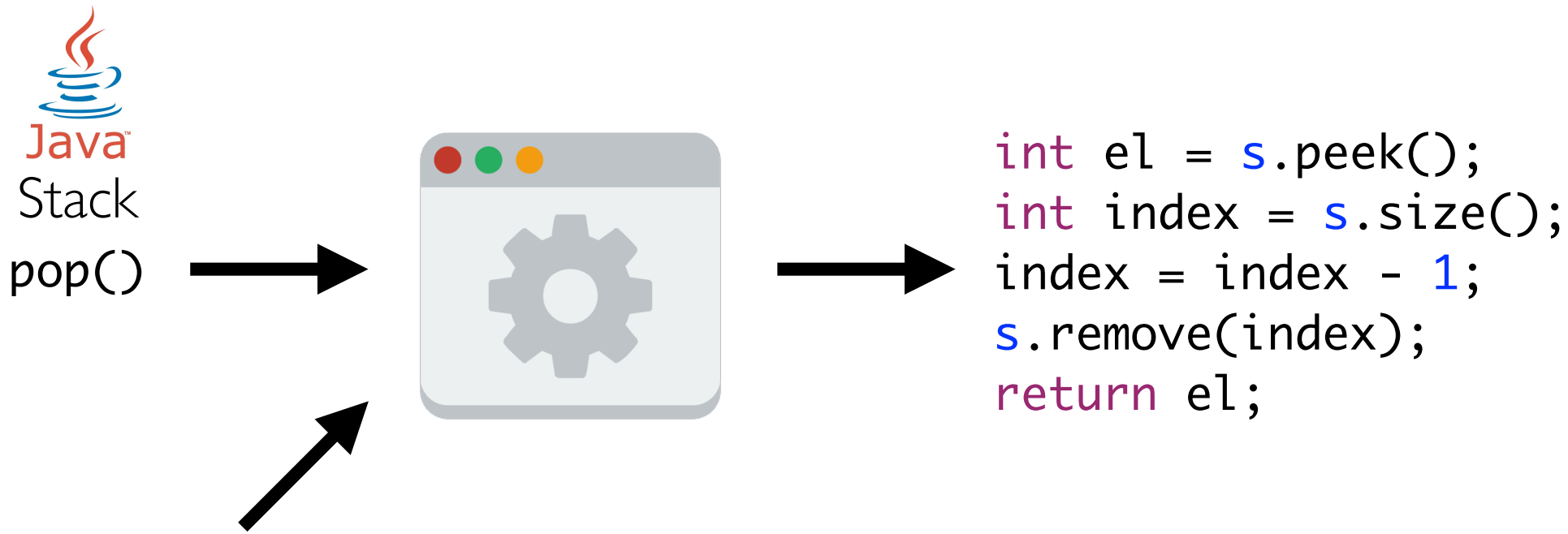
Security

Automatic Synthesis of Equivalent Method Sequences

Automatic Synthesis of Equivalences



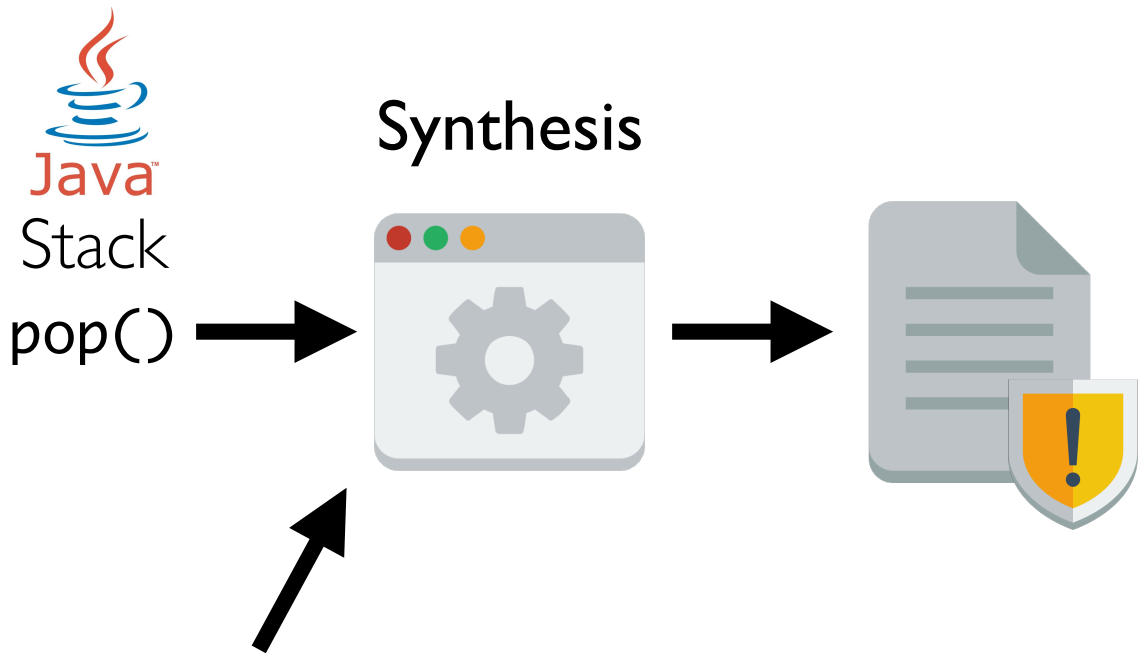
Automatic Synthesis of Equivalences




```
Stack s = new Stack();
s.push(1);
s.push(1);
Object ret = s.pop();
```

```
Stack s = new Stack();
s.push(-4);
Object ret = s.pop();
```

Automatic Synthesis of Equivalences

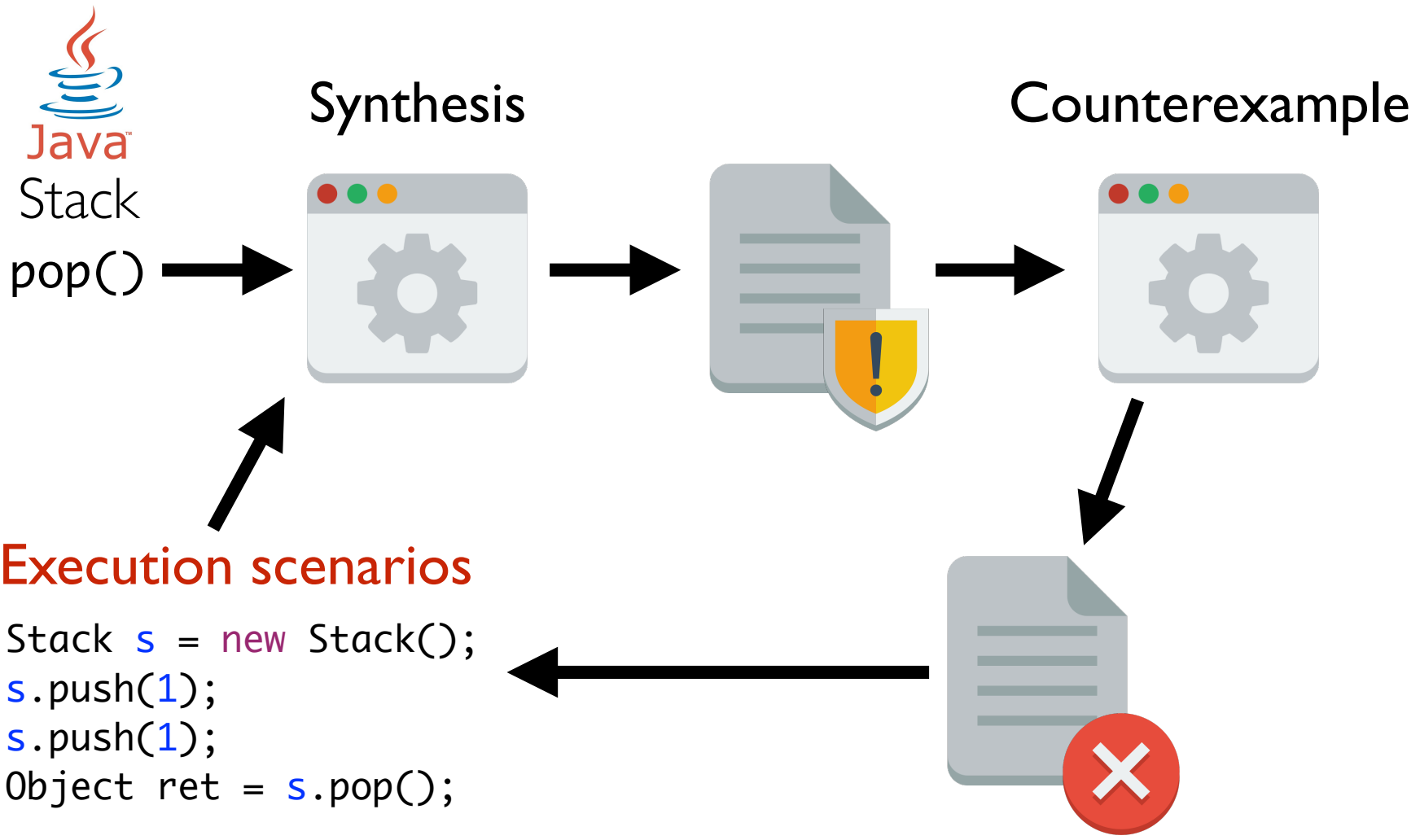



```
Stack s = new Stack();
s.push(1);
s.push(1);
Object ret = s.pop();

Stack s = new Stack();
s.push(-4);
Object ret = s.pop();
```

Automatic Synthesis of Equivalences


```
Stack s = new Stack();
s.push(1);
s.push(1);
Object ret = s.pop();

Stack s = new Stack();
s.push(-4);
Object ret = s.pop();
```

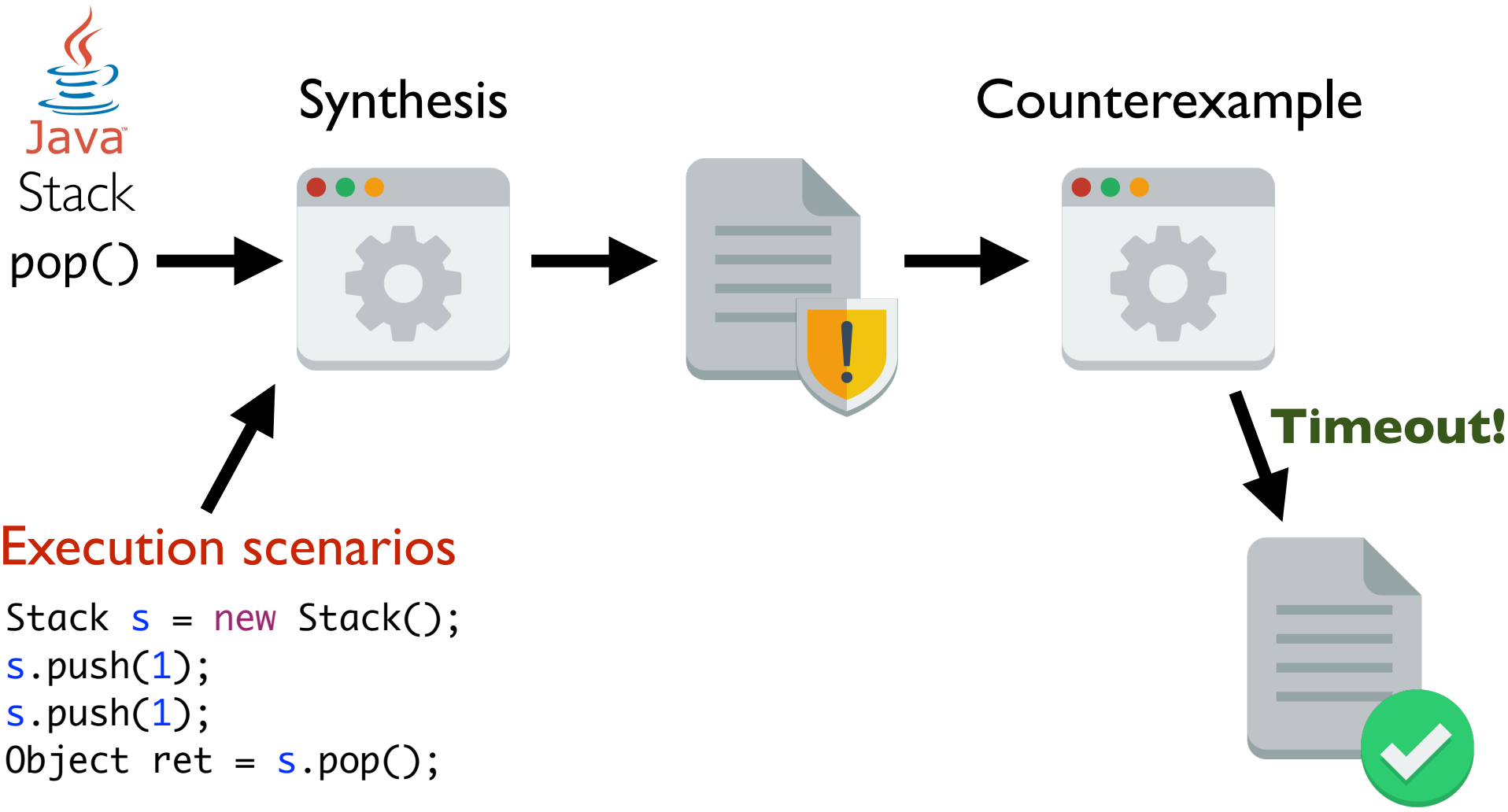
Automatic Synthesis of Equivalences



Automatic Synthesis of Equivalences

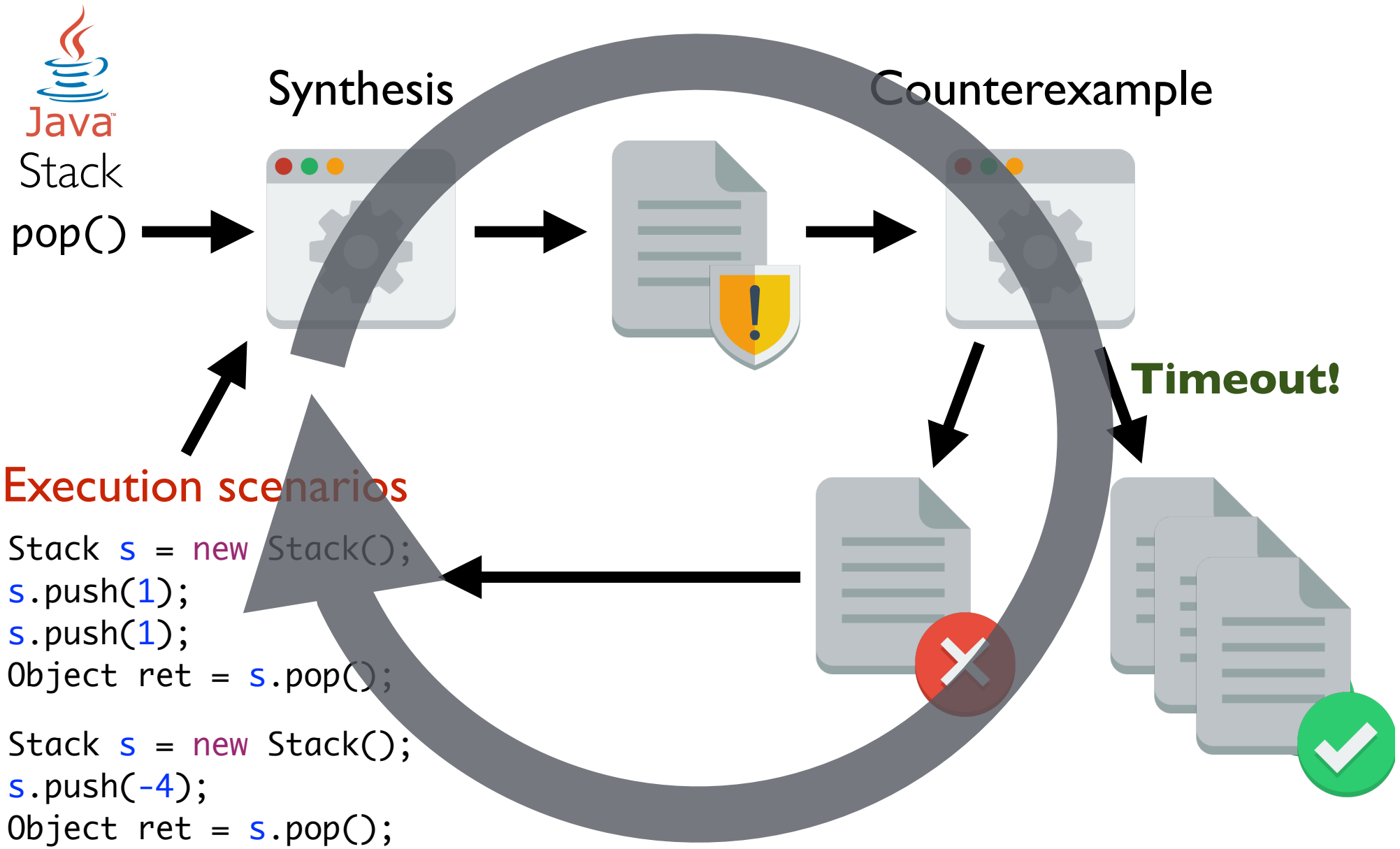

```
Stack s = new Stack();
s.push(1);
s.push(1);
Object ret = s.pop();

Stack s = new Stack();
s.push(-4);
Object ret = s.pop();
```

Automatic Synthesis of Equivalences

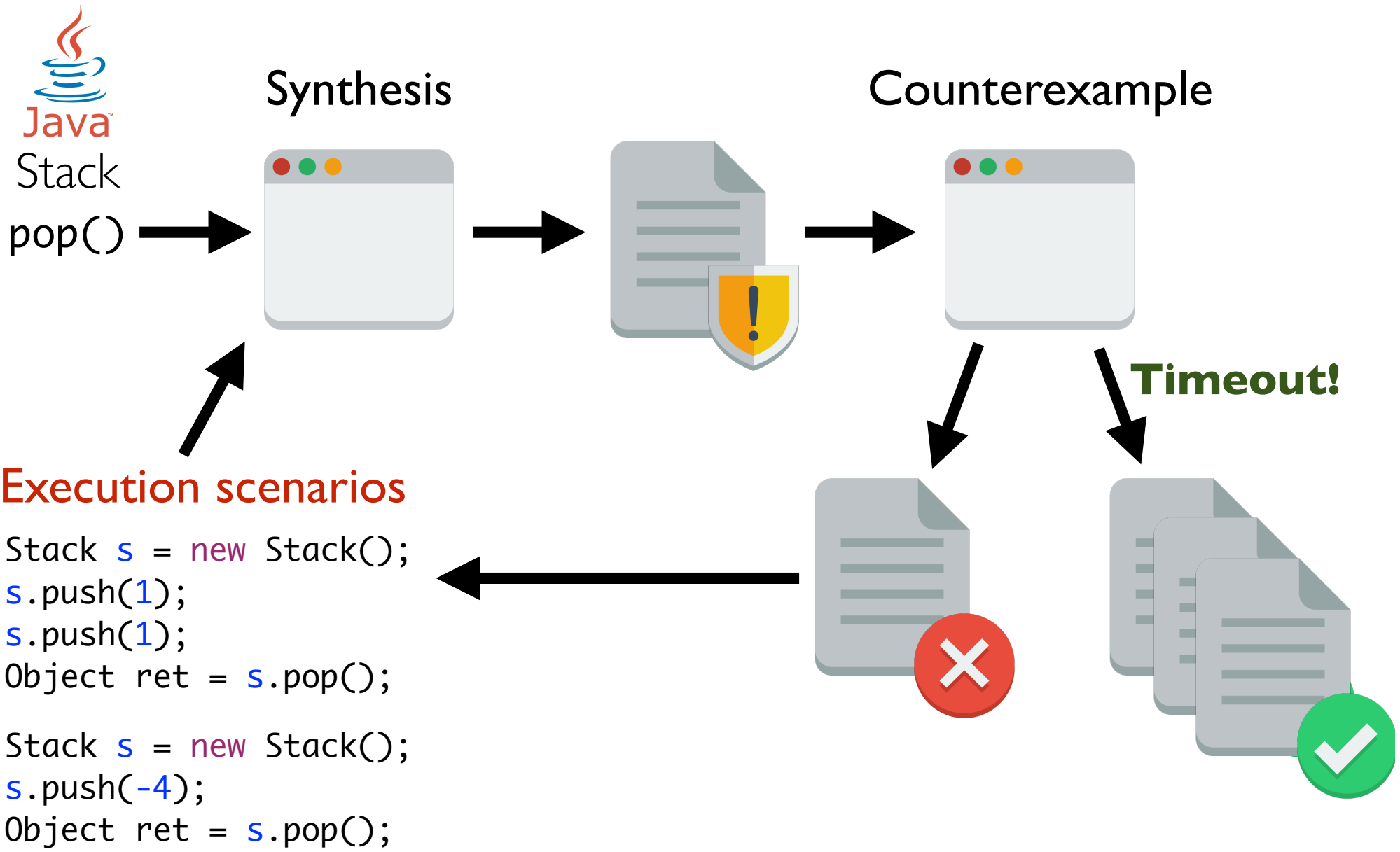



```
Stack s = new Stack();
s.push(1);
s.push(1);
Object ret = s.pop();

Stack s = new Stack();
s.push(-4);
Object ret = s.pop();
```

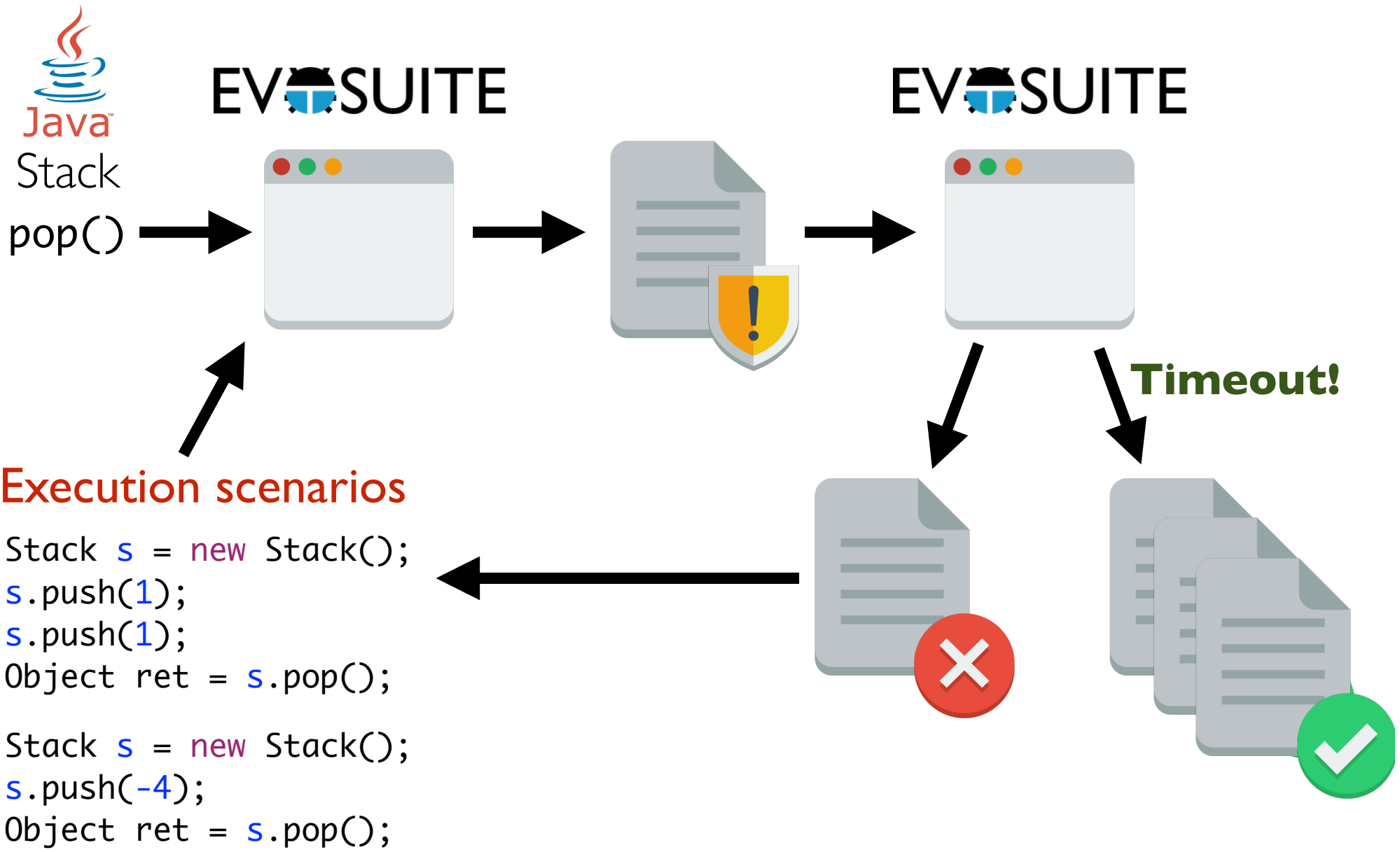
Automatic Synthesis of Equivalences



Search-based Synthesis of Equivalences



Search-based Synthesis of Equivalences



Equivalence Synthesis as TCG Problem

EVOSUITE

```
public void method_under_test() {
   if ( condition ) {
      // equivalent!
   }
}
```

Equivalence Synthesis as TCG Problem

EVOSUITE

```
public void method_under_test() {
   if ( condition ) {
      // equivalent!
   }
}
```

On **all** execution scenarios:

- compute **identical** results
- lead to **identical** object states

Equivalence Synthesis as TCG Problem

EVOSUITE

```
public void method_under_test() {
   if ( condition ) {
      // equivalent!
   }
}
```

pop()

Java
Stack

Execution scenarios

```
Stack s = new Stack();
s.push(1);
s.push(1);
int ret = s.pop();
```

```
Stack s = new Stack();
s.push(-4);
int ret = s.pop();
```

Equivalence Synthesis as TCG Problem

EVOSUITE

```
public void method_under_test() {
   if ( condition ) {
      // equivalent!
   }
}
```

pop()

Java Stack

```
Stack s = new Stack();
s.push(1);
s.push(1);
int ret = s.pop();
```

```
Stack s = new Stack();
s.push(1);
s.push(1);
int ret = ? ? ? ? ? ;
```

```
Stack s = new Stack();
s.push(-4);
int ret = s.pop();
```

```
Stack s = new Stack();
s.push(-4);
int ret = ? ? ? ? ? ;
```

Equivalence Synthesis as TCG Problem

EVOSUITE

```
public void method_under_test() {
  if ([    ]==[    ]&&[    ]==[    ]&&
      [    ]==[    ]&&[    ]==[    ]){
    // equivalent!
  }
}
```

pop()

Java Stack

```
Stack s = new Stack();
s.push(1);
s.push(1);
int ret = s.pop();
```

```
Stack s = new Stack();
s.push(1);
s.push(1);
int ret = ? ? ? ? ? ;
```

```
Stack s = new Stack();
s.push(-4);
int ret = s.pop();
```

```
Stack s = new Stack();
s.push(-4);
int ret = ? ? ? ? ? ;
```

Equivalence Synthesis as TCG Problem

EVOSUITE

```
public void method_under_test() {
  if (          ==          &&          ==          &&
      
          ==          &&          ==          ){
      // equivalent!
  }
}
```

remove(0)

```
Stack s = new Stack();
s.push(1);
s.push(1);
int ret = s.pop();
```

```
Stack s = new Stack();
s.push(1);
s.push(1);
int ret = ? ? ? ? ? ;
```

```
Stack s = new Stack();
s.push(-4);
int ret = s.pop();
```

```
Stack s = new Stack();
s.push(-4);
int ret = ? ? ? ? ? ;
```

Equivalence Synthesis as TCG Problem

EVOSUITE

```
public void method_under_test() {
  if (      ==      &&      ==      &&
        ==      &&      ==      ){
    // equivalent!
  }
}
```

remove(0)

```
Stack s = new Stack();
s.push(1);
s.push(1);
int ret = s.pop();
```

```
Stack s = new Stack();
s.push(1);
s.push(1);
int ret = s.remove(0);
```

```
Stack s = new Stack();
s.push(-4);
int ret = s.pop();
```

```
Stack s = new Stack();
s.push(-4);
int ret = s.remove(0);
```

Equivalence Synthesis as TCG Problem

EVOSUITE

```
public void method_under_test() {
  if ( [1] == [1] && [] == [] &&

       1 == 1 && -4 == -4 ){
    // equivalent!
  }
}
```

remove(0)

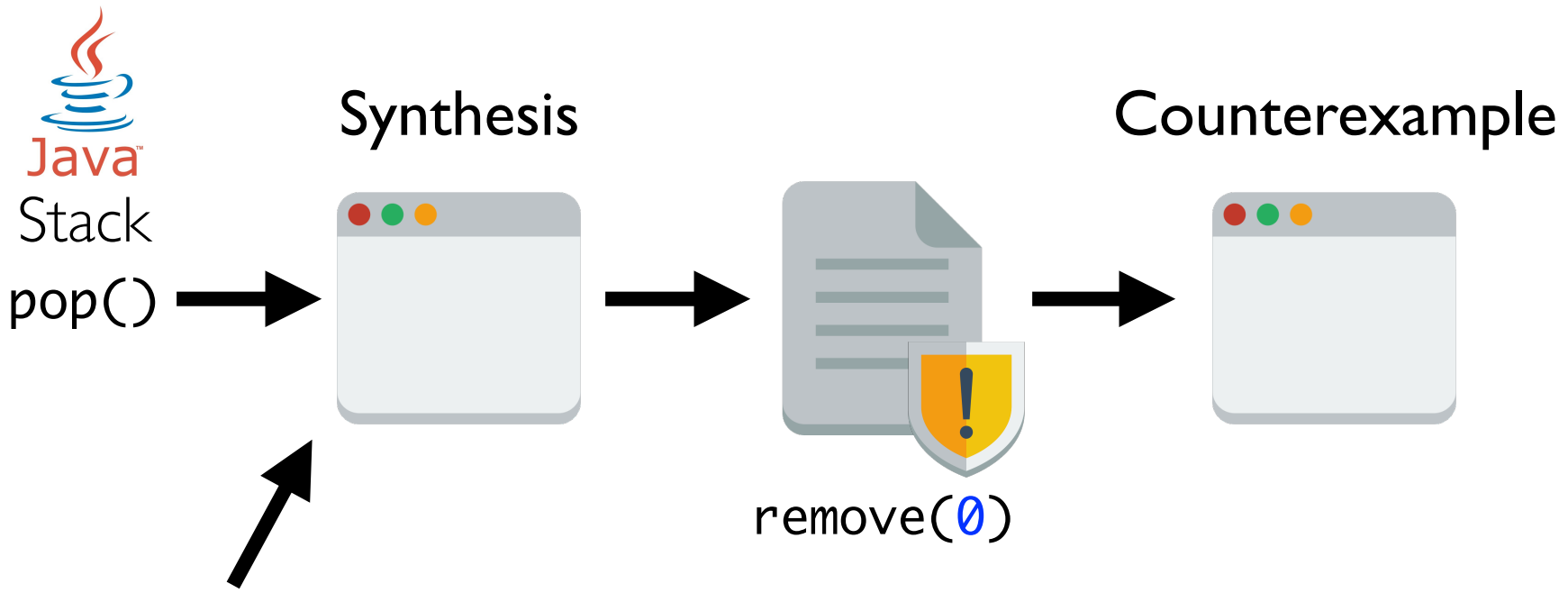
```
Stack s = new Stack();
s.push(1);
s.push(1);
int ret = s.pop();
```

```
Stack s = new Stack();
s.push(1);
s.push(1);
int ret = s.remove(0);
```

```
Stack s = new Stack();
s.push(-4);
int ret = s.pop();
```

```
Stack s = new Stack();
s.push(-4);
int ret = s.remove(0);
```

Search-based Synthesis of Equivalences




```
Stack s = new Stack();
s.push(1);
s.push(1);
Object ret = s.pop();

Stack s = new Stack();
s.push(-4);
Object ret = s.pop();
```

Counterexample as TCG Problem

EVOSUITE

```
public void method_under_test() {
   if ( condition ) {
      // counterexample
   }
}
```

Counterexample as TCG Problem

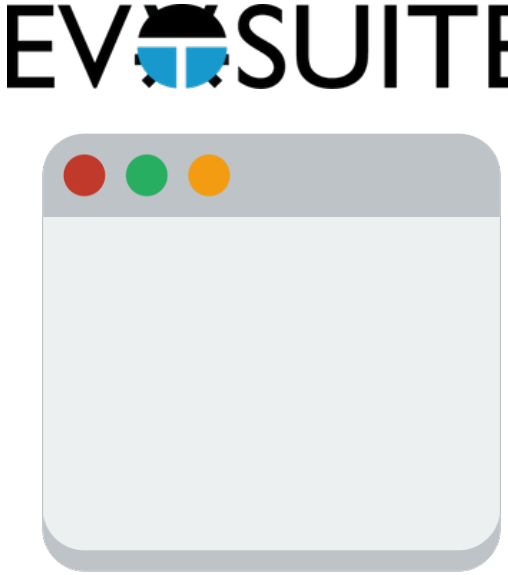
EVOSUITE

```
public void method_under_test() {
   if ( condition ) {
      // counterexample
   }
}
```

On **one** execution scenario:

- compute **different** results, or
- lead to **different** object states

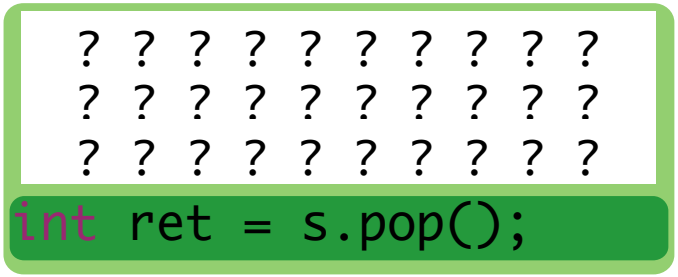
Counterexample as TCG Problem

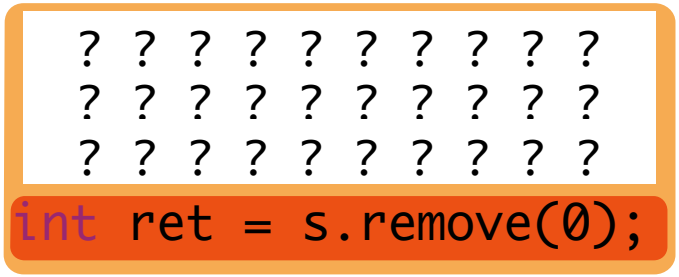



```
public void method_under_test() {
   if ( condition ) {
      // counterexample
   }
}
```

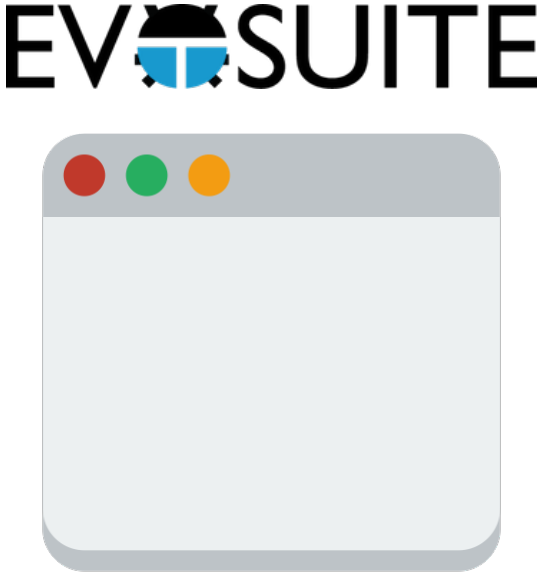
remove(0)

Java
Stack



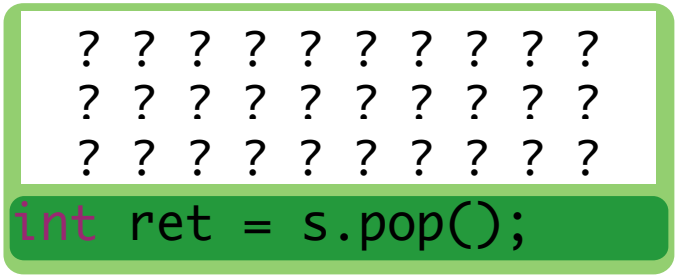


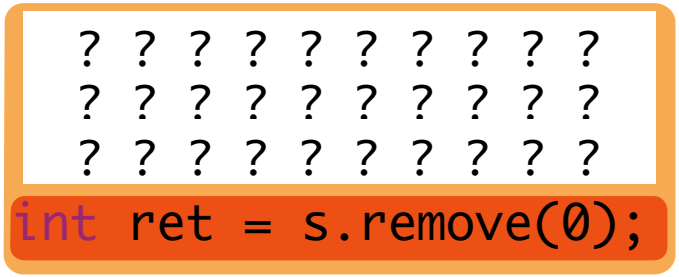
Counterexample as TCG Problem




```
public void method_under_test() {
  if ( [ ] != [ ] ||
       [ ] != [ ] ) {
    // counterexample
  }
}
```

remove(0)



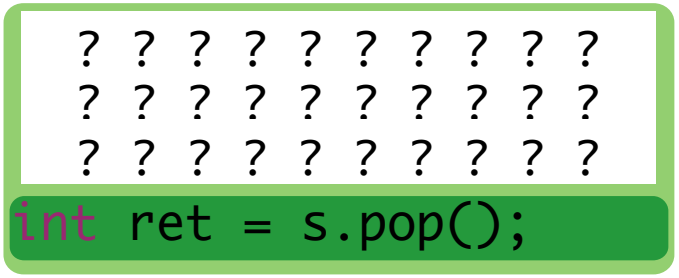


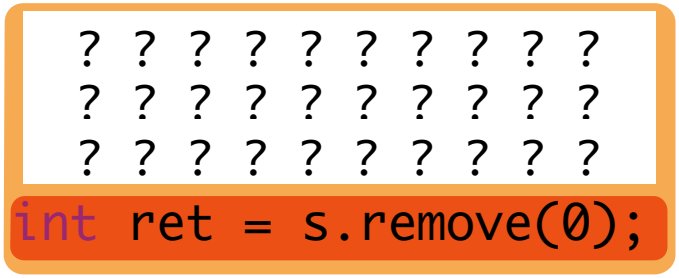
Stack

Counterexample as TCG Problem

EVOSUITE

```
public void method_under_test() {
  if ( [ ] != [ ] ||
       [ ] != [ ] ) {
    // counterexample
  }
}
```



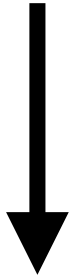



```
Stack s = new Stack();
s.push(0);
s.push(1);
```

Counterexample as TCG Problem

EVOSUITE

```
public void method_under_test() {
  if ( [ ] != [ ] ||
       [ ] != [ ] ) {
    // counterexample
  }
}
```



```
Stack s = new Stack();
s.push(0);
s.push(1);
int ret = s.pop();
```

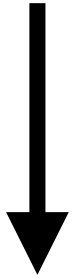
```
Stack s = new Stack();
s.push(0);
s.push(1);
int ret = s.remove(0);
```

```
Stack s = new Stack();
s.push(0);
s.push(1);
```

Counterexample as TCG Problem

EVOSUITE

```
public void method_under_test() {
  if ( [0] != [1] ||
       1  != 0    ) {
    // counterexample
  }
}
```

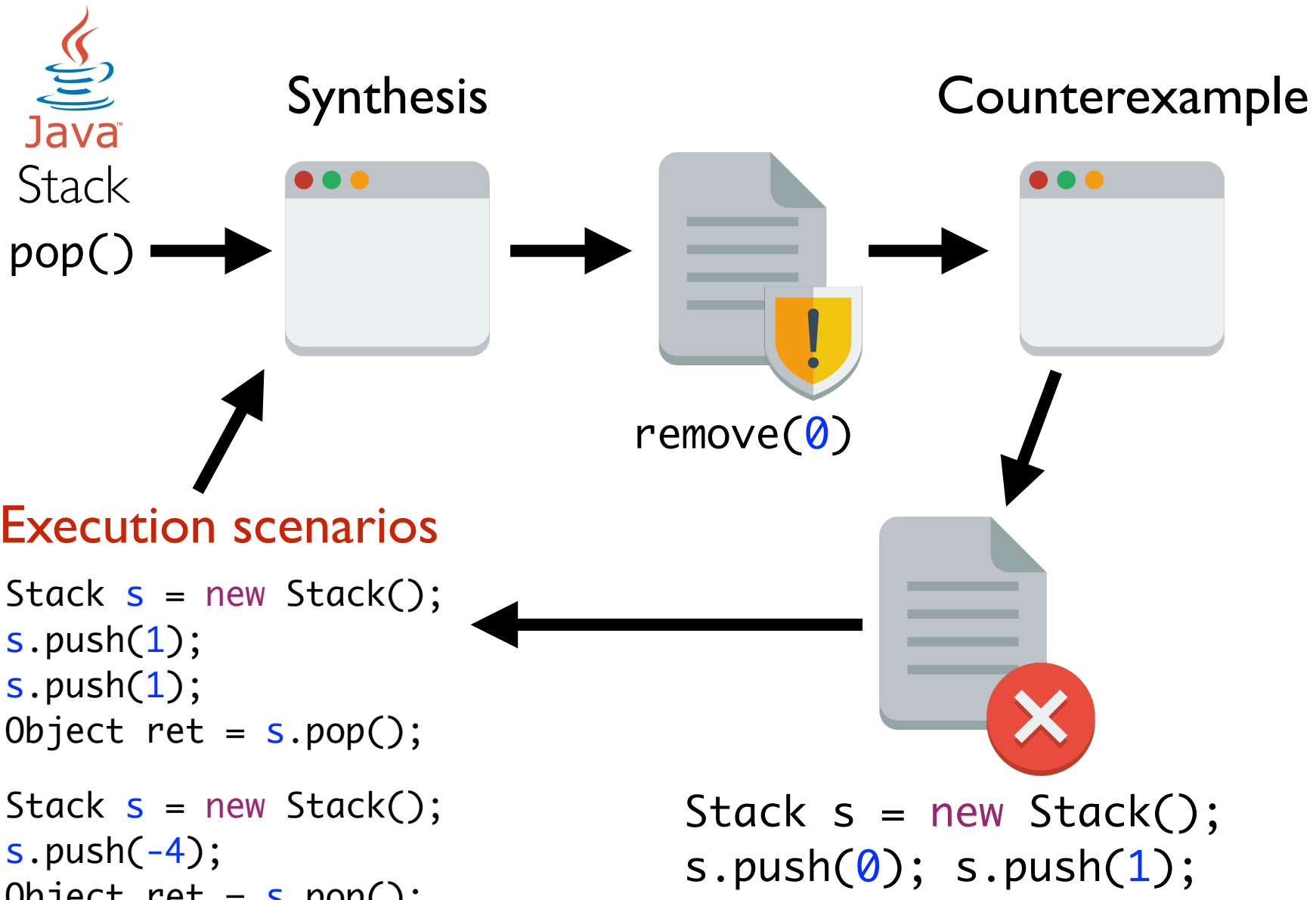


```
Stack s = new Stack();
s.push(0);
s.push(1);
int ret = s.pop();
```

```
Stack s = new Stack();
s.push(0);
s.push(1);
int ret = s.remove(0);
```

```
Stack s = new Stack();
s.push(0);
s.push(1);
```

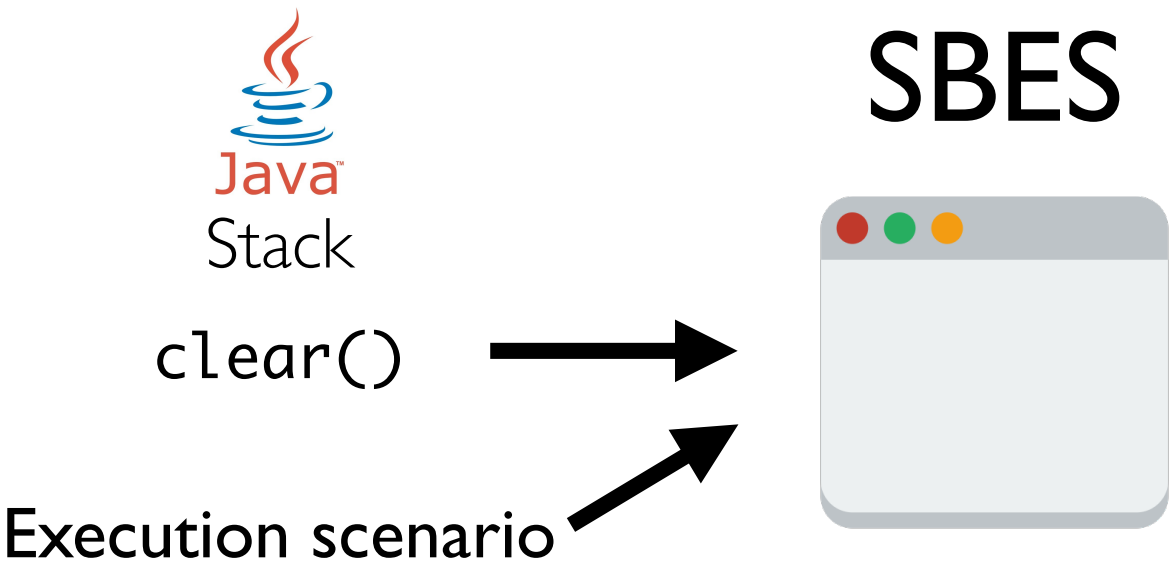
Search-based Synthesis of Equivalences



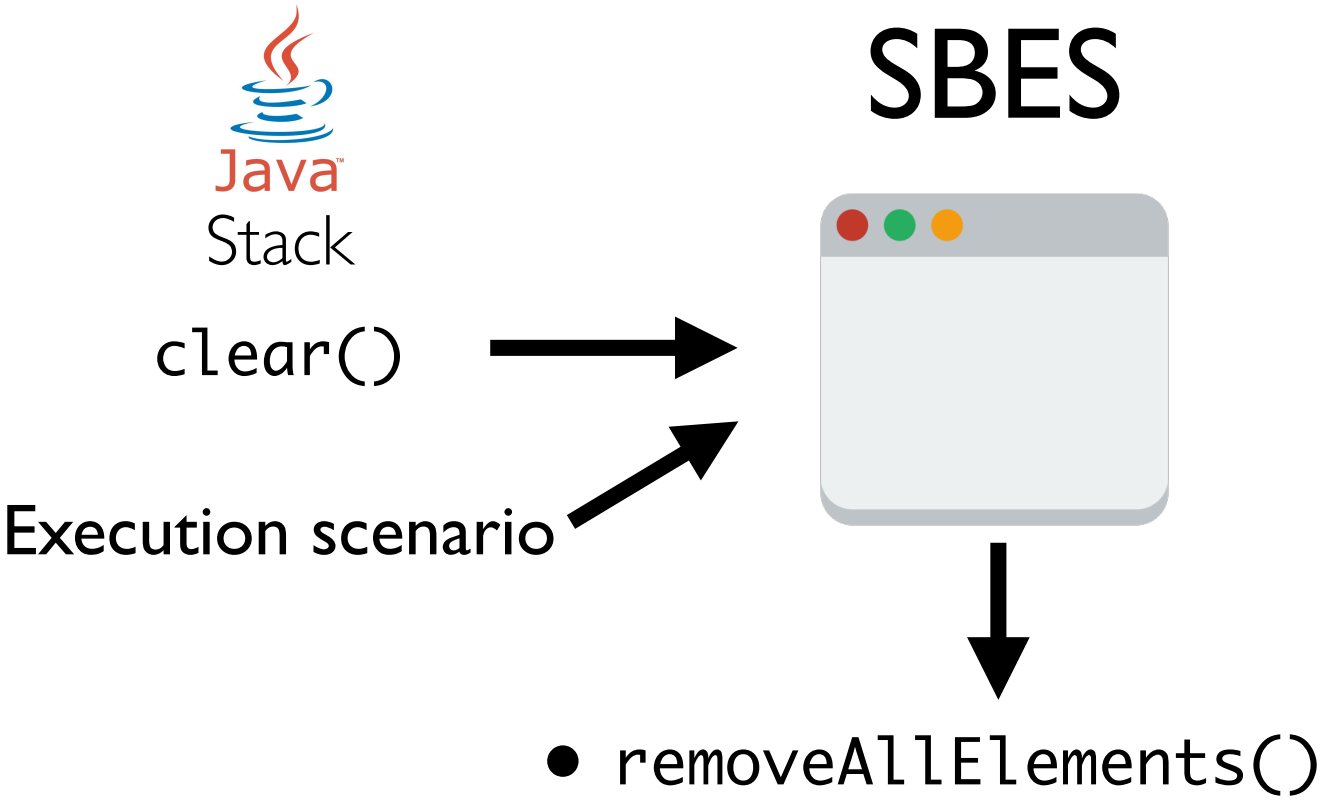
Search-based Synthesis of Equivalences

SBES

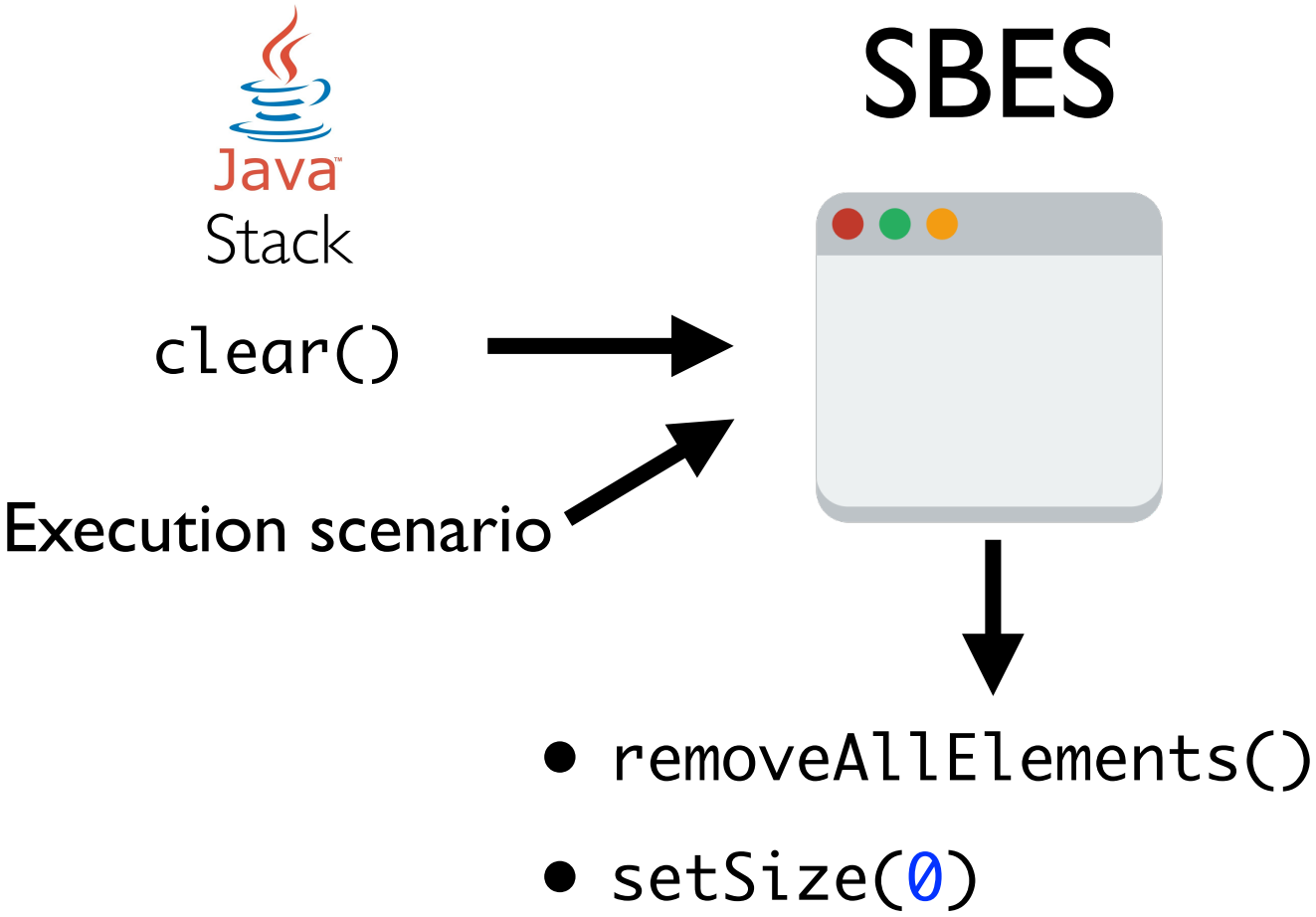
Search-based Synthesis of Equivalences



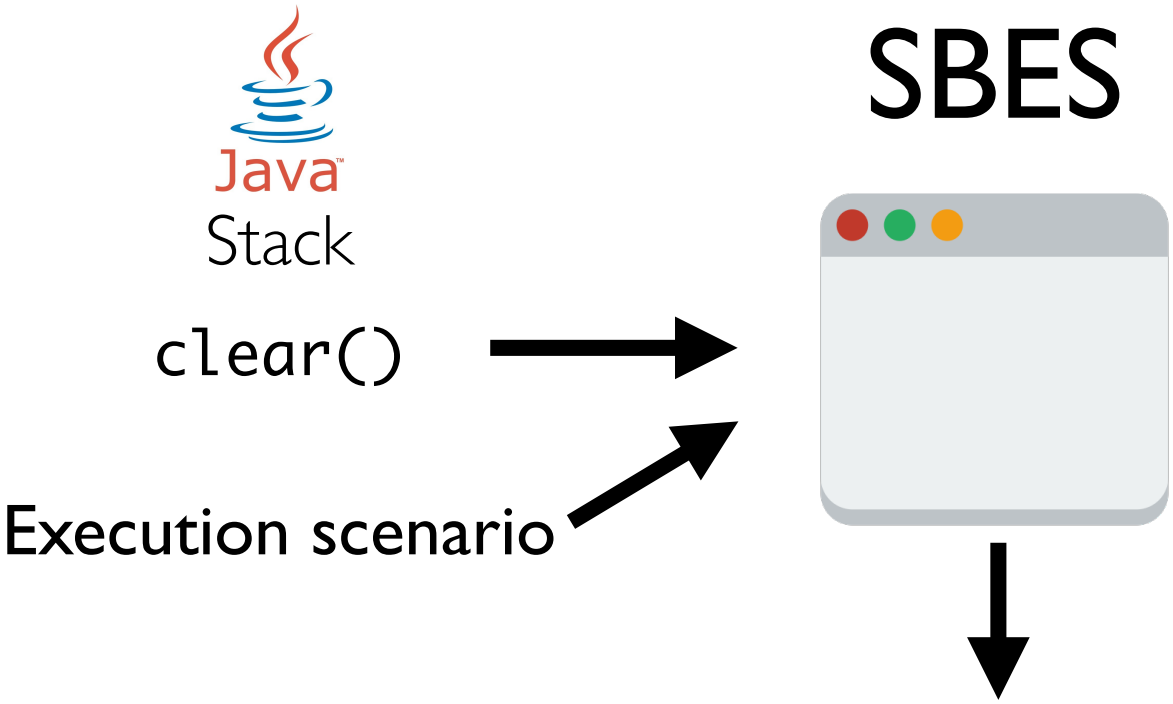
Search-based Synthesis of Equivalences



Search-based Synthesis of Equivalences



Search-based Synthesis of Equivalences



- `removeAllElements()`
- `setSize(0)`
- `Collection c = new Collection();`
  `retainAll(c)`

Evaluation

Evaluation

Stack

GraphStream

guava-libraries

23 classes

267 methods

Evaluation

Stack

31 equivalences

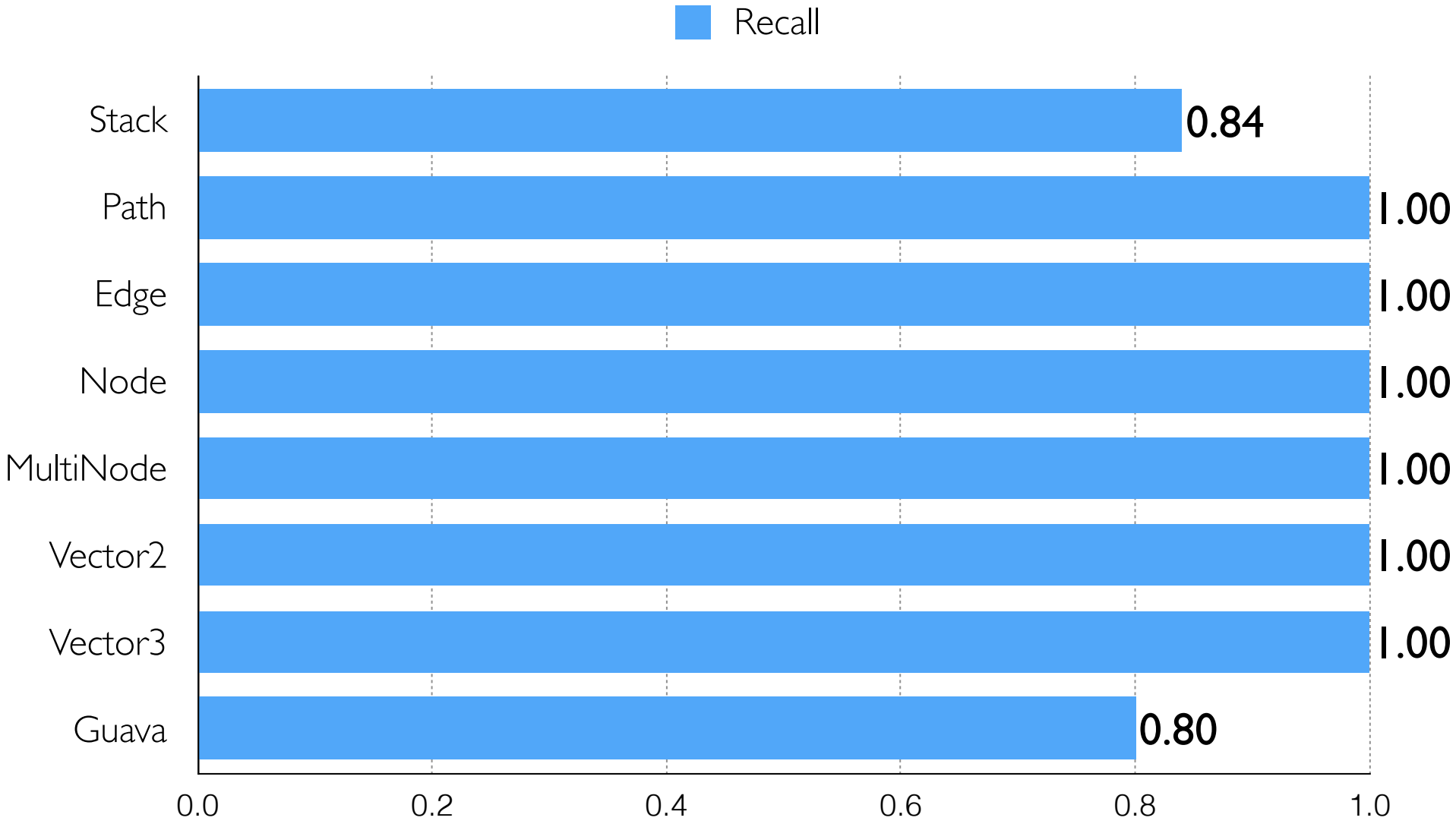
GraphStream

92 equivalences

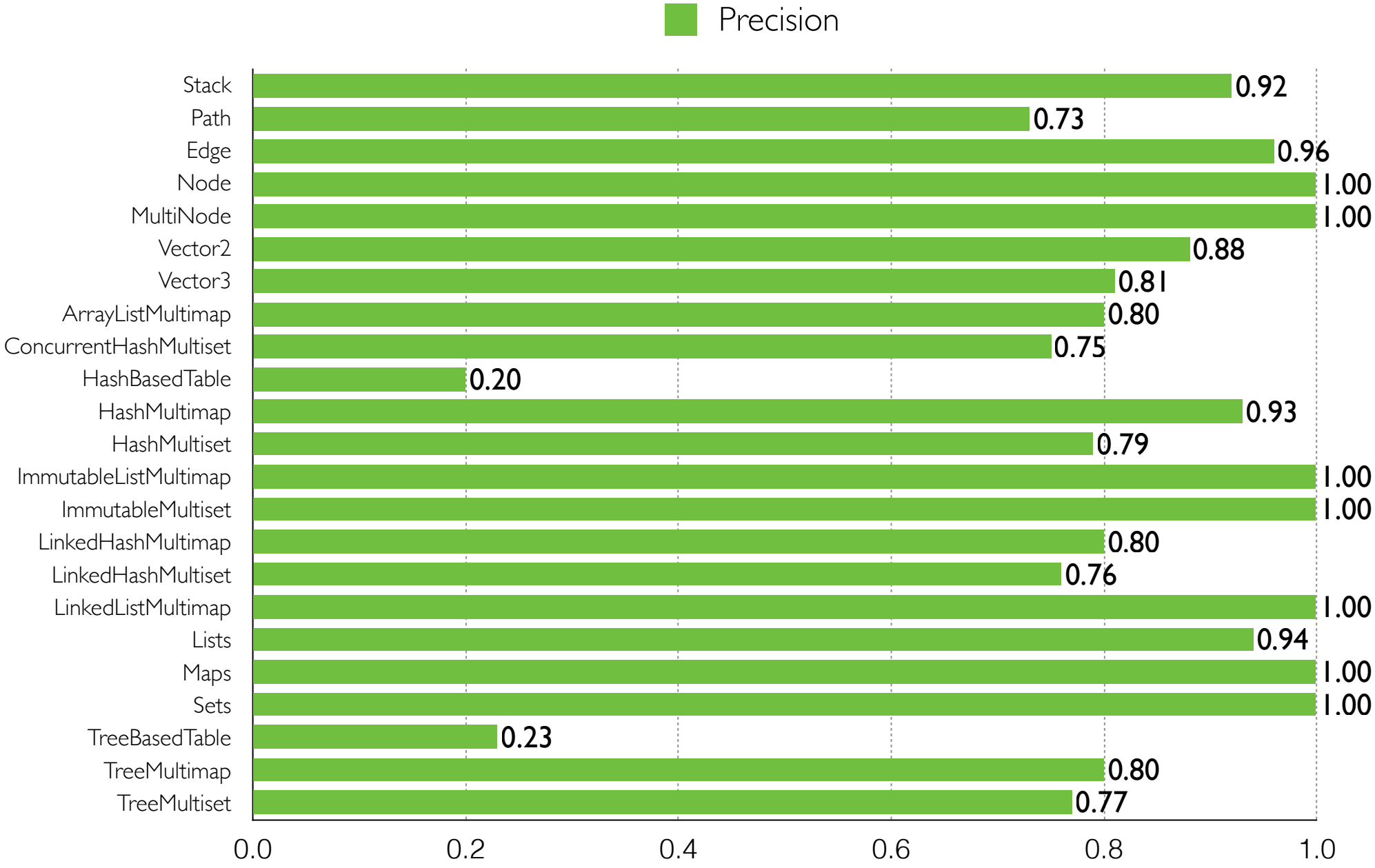
guava-libraries

188 equivalences

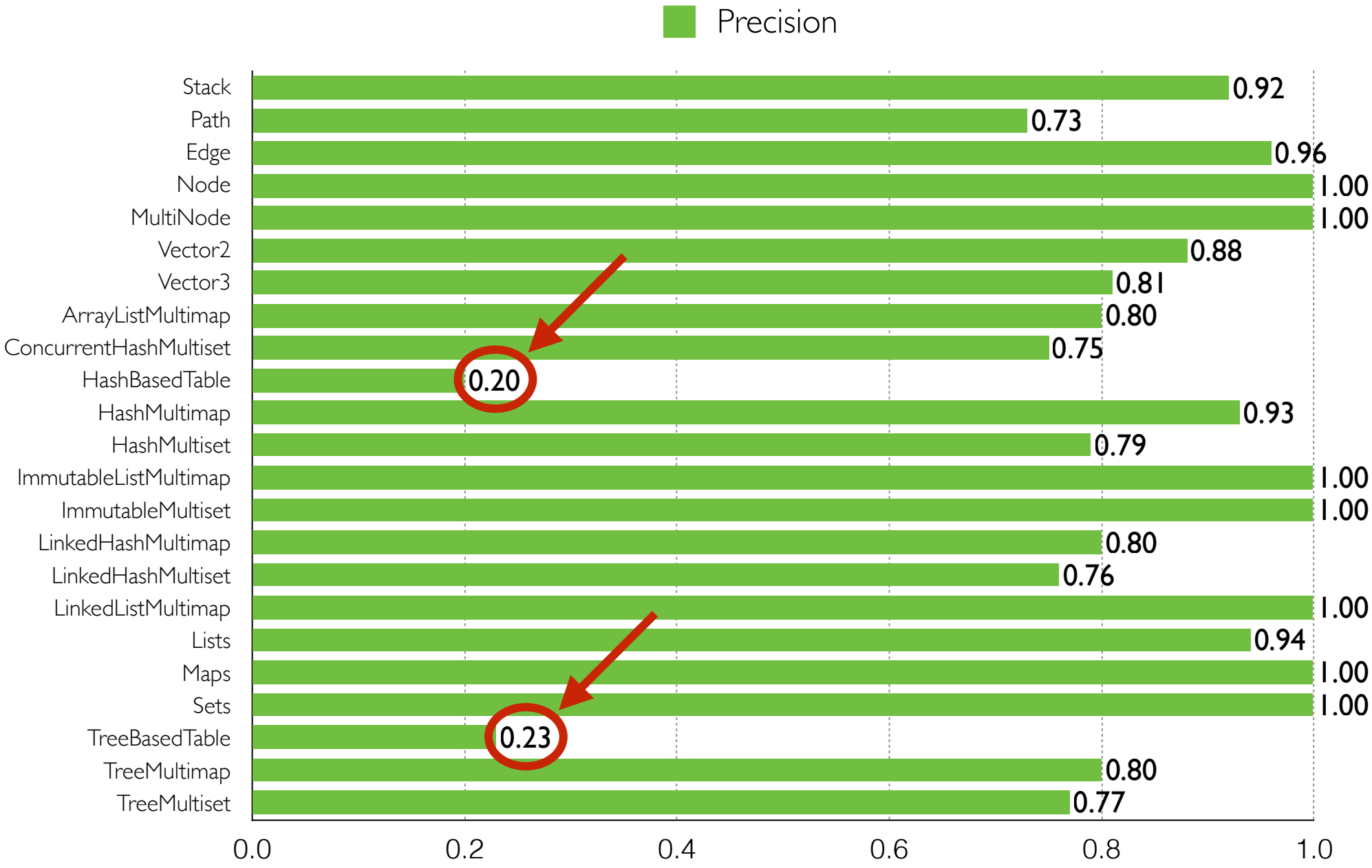
Evaluation


Recall
Stack
Path
Edge
Node
MultiNode
Vector2
Vector3
Guava
0.84
1.00
1.00
1.00
1.00
1.00
0.80
0.0
0.2
0.4
0.6
0.8
1.0

Evaluation


Precision
Stack 0.92
Path 0.73
Edge 0.96
Node 1.00
MultiNode 1.00
Vector2 0.88
Vector3 0.81
ArrayListMultimap 0.80
ConcurrentHashMultiset 0.75
HashBasedTable 0.20
HashMultimap 0.93
HashMultiset 0.79
ImmutableListMultimap 1.00
ImmutableMultiset 1.00
LinkedHashMultimap 0.80
LinkedHashMultiset 0.76
LinkedListMultimap 1.00
Lists 0.94
Maps 1.00
Sets 1.00
TreeBasedTable 0.23
TreeMultimap 0.80
TreeMultiset 0.77
0.0
0.2
0.4
0.6
0.8
1.0

Evaluation


Precision
Stack
Path
Edge
Node
MultiNode
Vector2
Vector3
ArrayListMultimap
ConcurrentHashMultiset
HashBasedTable
HashMultimap
HashMultiset
ImmutableListMultimap
ImmutableMultiset
LinkedHashMultimap
LinkedHashMultiset
LinkedListMultimap
Lists
Maps
Sets
TreeBasedTable
TreeMultimap
TreeMultiset
0.92
0.73
0.96
1.00
1.00
0.88
0.81
0.80
0.75
0.20
0.93
0.79
1.00
1.00
0.80
0.76
1.00
0.94
1.00
1.00
0.23
0.80
0.77
0.0
0.2
0.4
0.6
0.8
1.0

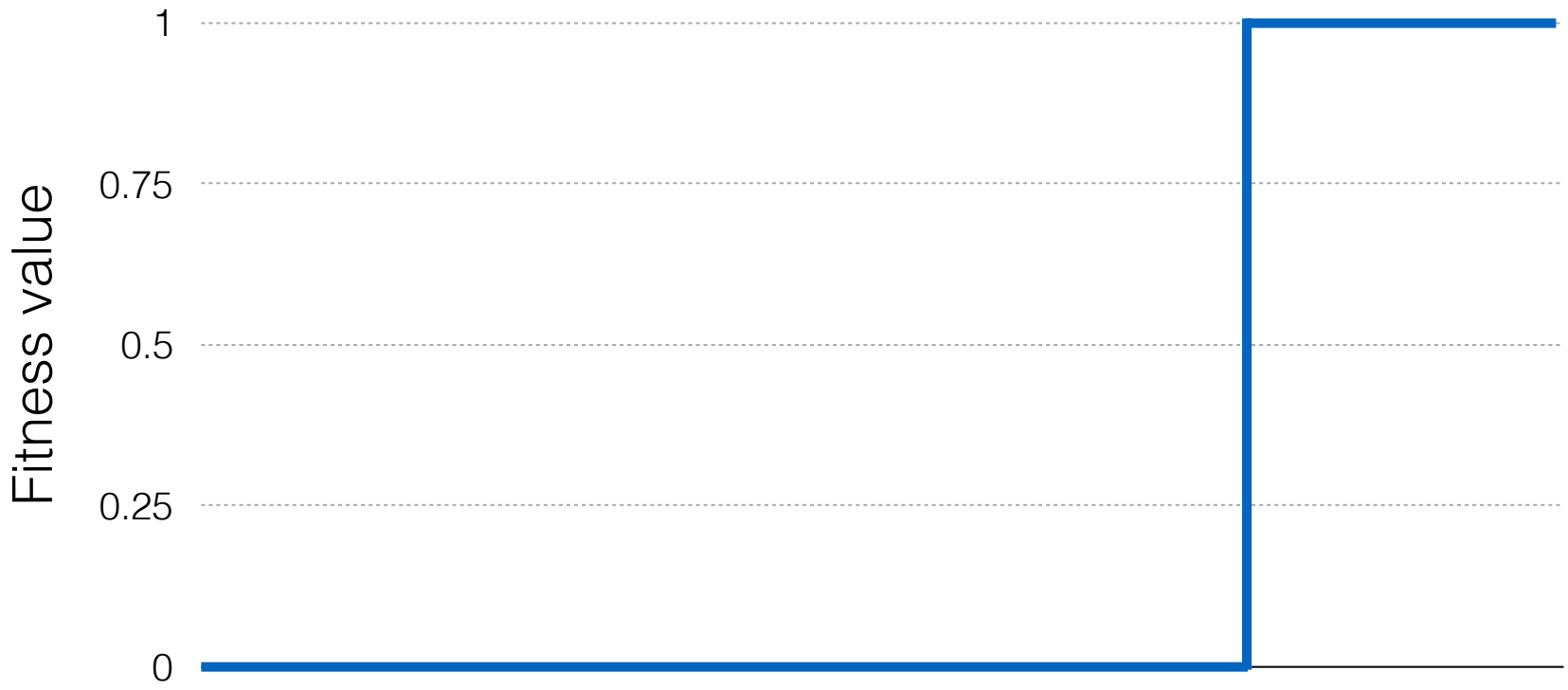
Effectiveness of Counterexamples

```
public void method_under_test() {
  if ( [redacted] != [redacted] ||
       [redacted] != [redacted] ) {
    // counterexample
  }
}
```

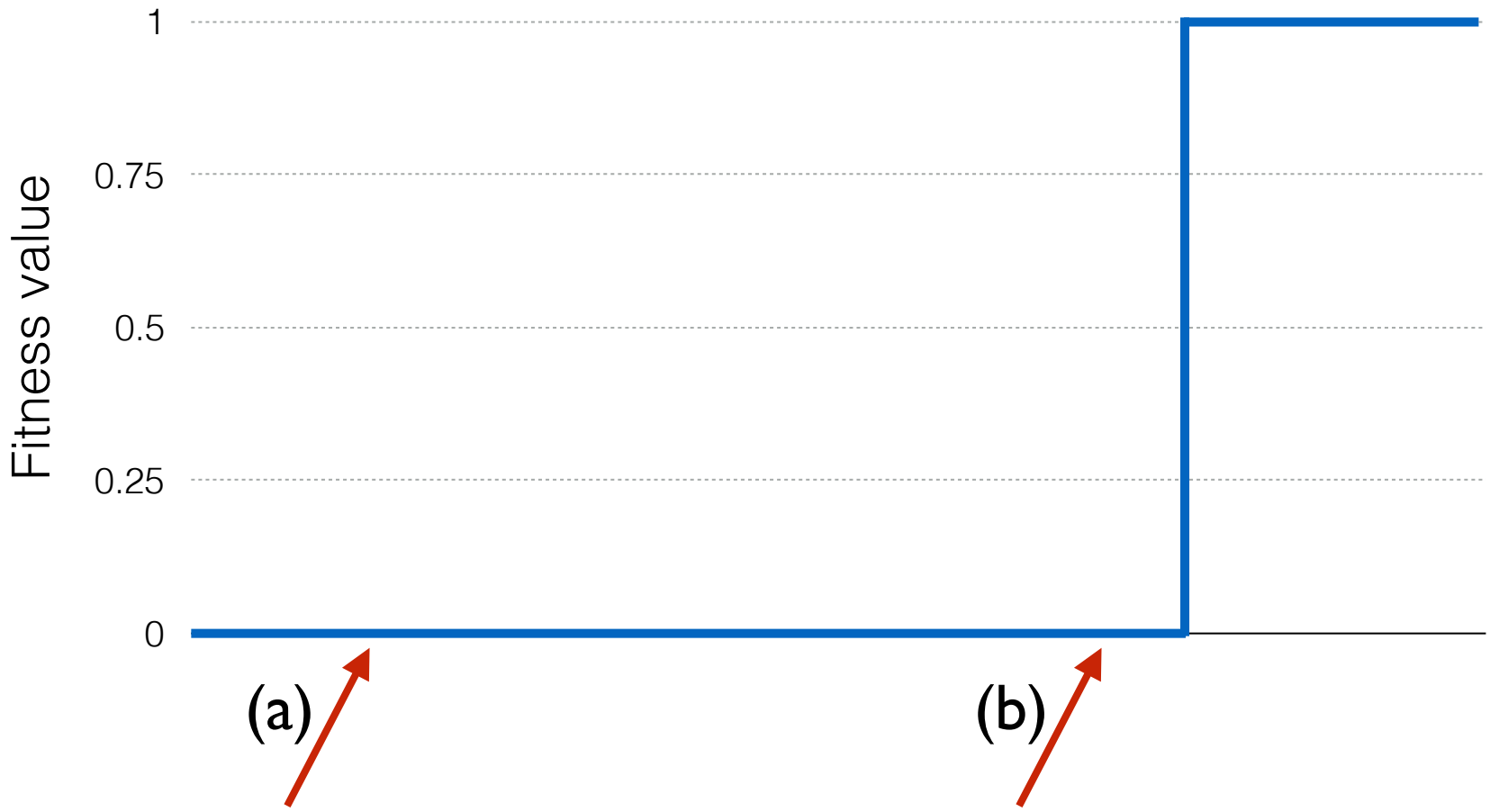
Effectiveness of Counterexamples

```
public void method_under_test() {
   clone = deepClone(this);
   boolean expect = this.original_method();
   boolean actual = clone.candidate_method();

   if (distance(this,clone) > 0 ||
         distance(expect,actual) > 0) {
      // equivalent!
   }
}
```

Effectiveness of Counterexamples



Effectiveness of Counterexamples



Effectiveness of Counterexamples

```
public void method_under_test() {
   clone = deepClone(this);
   boolean expect = this.original_method();
   boolean actual = clone.candidate_method();

   if (this != clone || expect != actual) {
      // equivalent!
   }
}
```

Effectiveness of Counterexamples

```
public void method_under_test() {
   clone = deepClone(this);
   boolean expect = this.original_method();
   boolean actual = clone.candidate_method();

   if (this != clone || expect != actual) {
      // equivalent!
   }
}
```

Difficult for search-based…

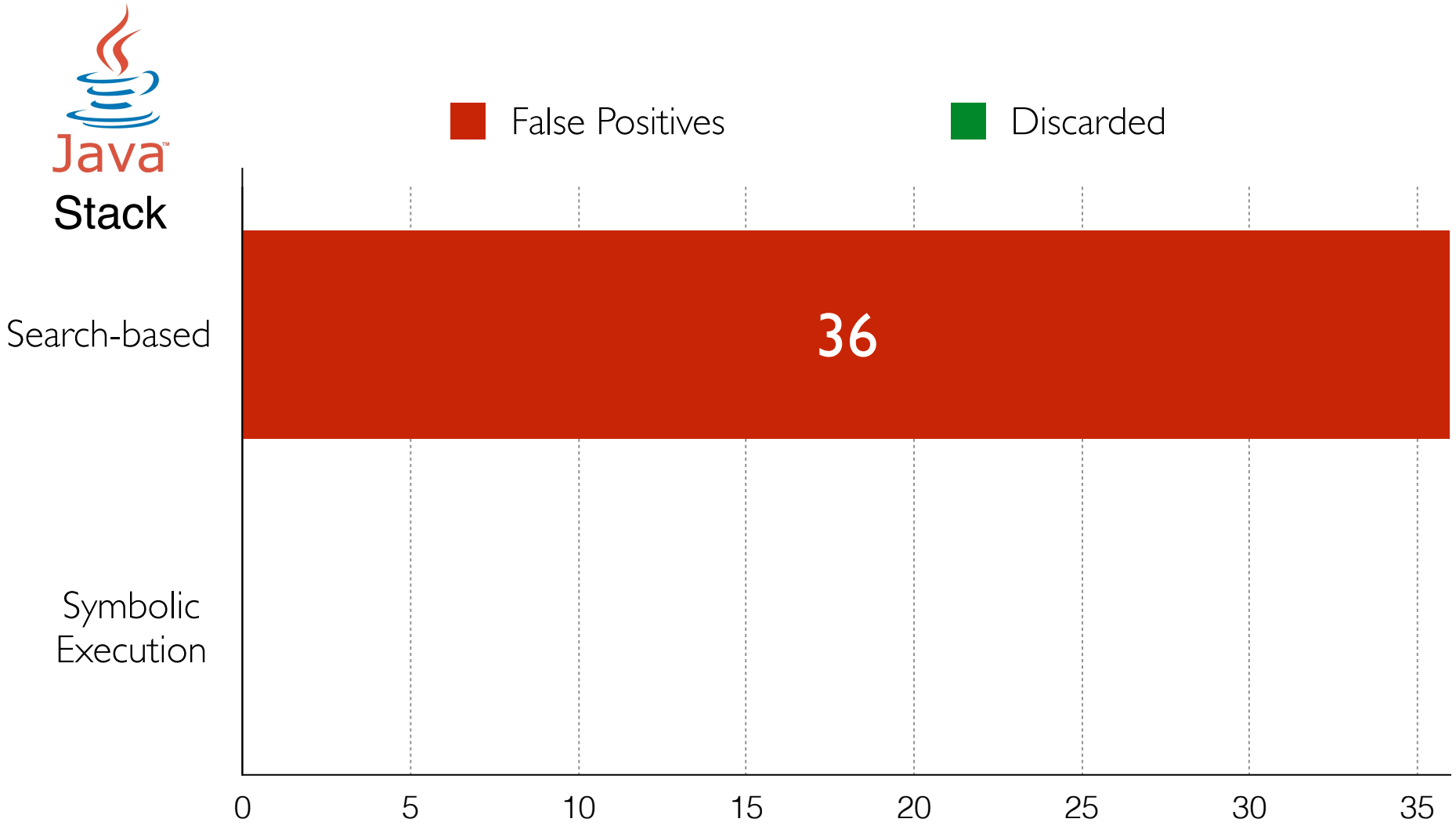
Effectiveness of Counterexamples

```
public void method_under_test() {
  clone = deepClone(this);
  boolean expect = this.original_method();
  boolean actual = clone.candidate_method();

  if (this != clone || expect != actual) {
    // equivalent!
  }
}
```

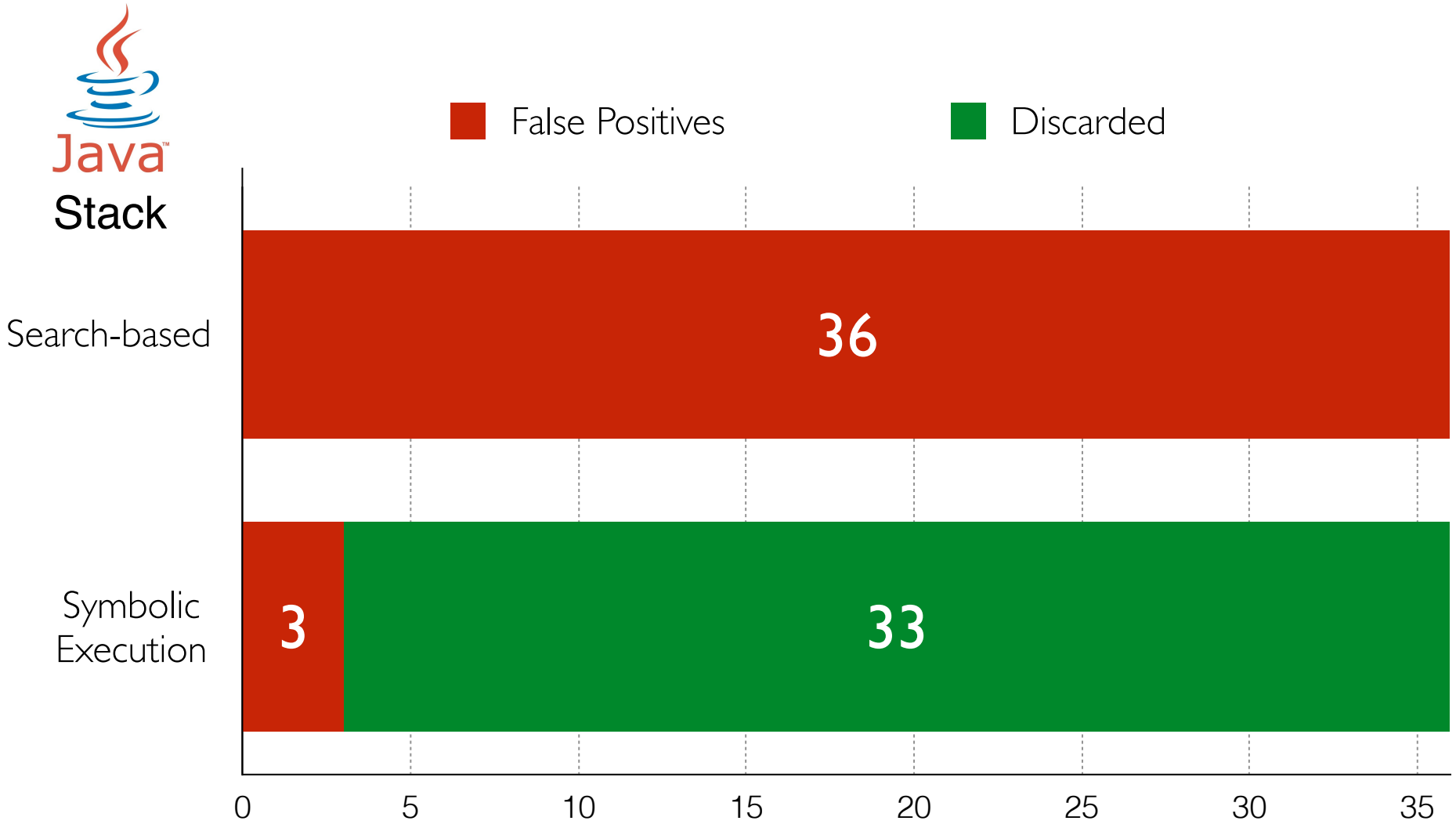
Difficult for search-based…

…well-suited for symbolic execution!

Effectiveness of Counterexamples



Effectiveness of Counterexamples



Software Redundancy
Intrinsic

Joda-Time

```
DateTime t = new DateTime();
//...
//get the beginning of the day for time t
DateTime beginDay = t.millisOfDay().withMinimumValue();
                  = t.toDateMidnight().toDateTime();
                  = t.withTimeAtStartOfDay();
```

2 LOC (~0.1%)

Google Guava

```
MultiMap m = new MultiMap();
//…
//check if element is already in map
if (m.contains(x))
 if (m.elementSet().contains(x))
 if (m.count(x) > 0)
```

0 LOC (0.0%)

Exploiting Redundancy

Automatic repair

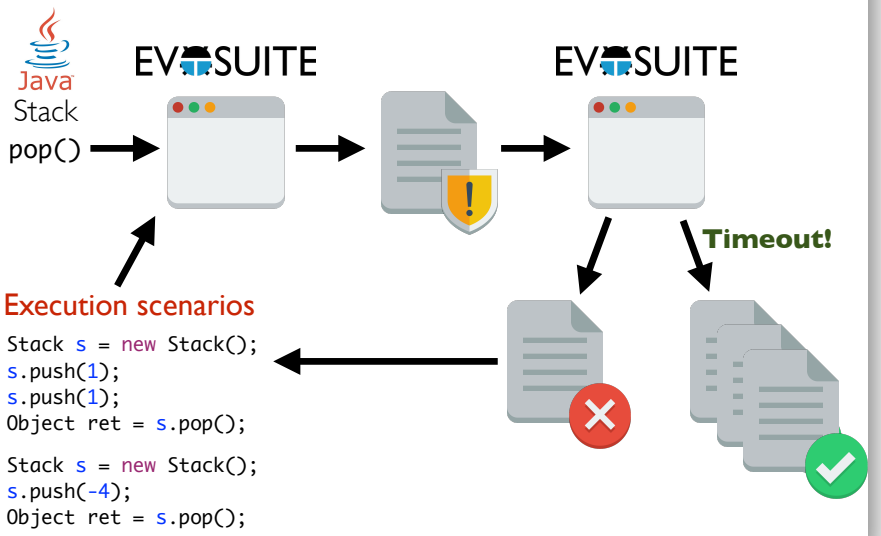
Manual identification of equivalence

Test oracles

… is the main cost!

Security

Search-based Synthesis of Equivalences



Evaluation

Java Stack — **31** equivalences

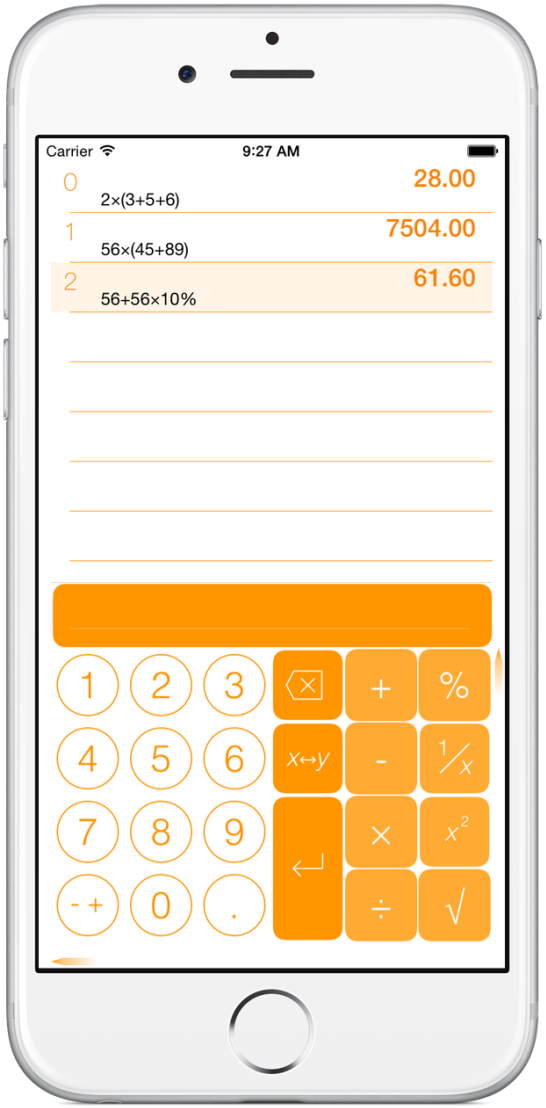
GraphStream — **92** equivalences

guava-libraries — **188** equivalences

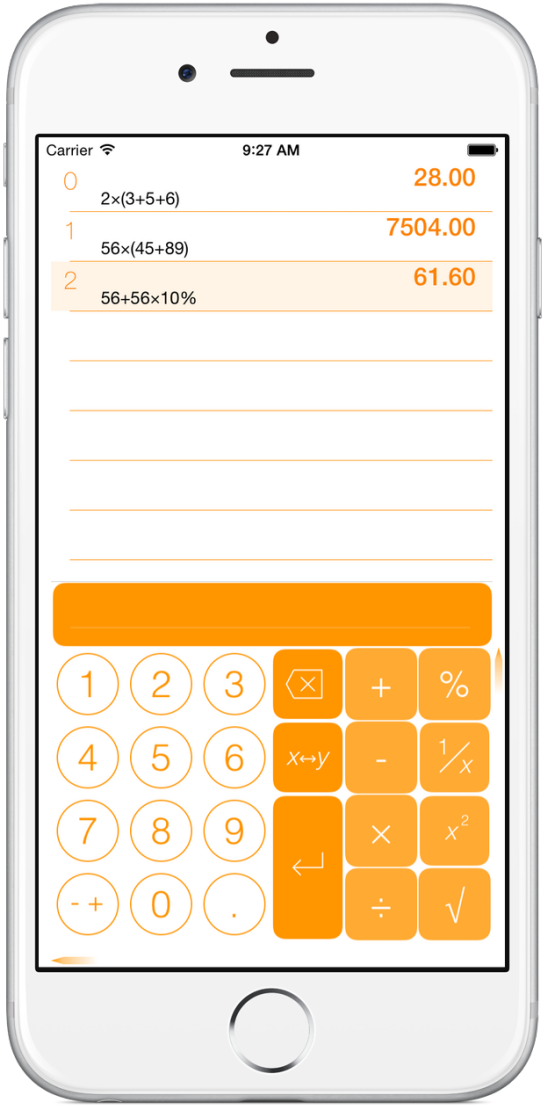

NISSAN ALTIMA
Crashlab
28 NOV 13
B13038
ANCAP
Crash testing for safety


Jeep
LRZ 346
TEST

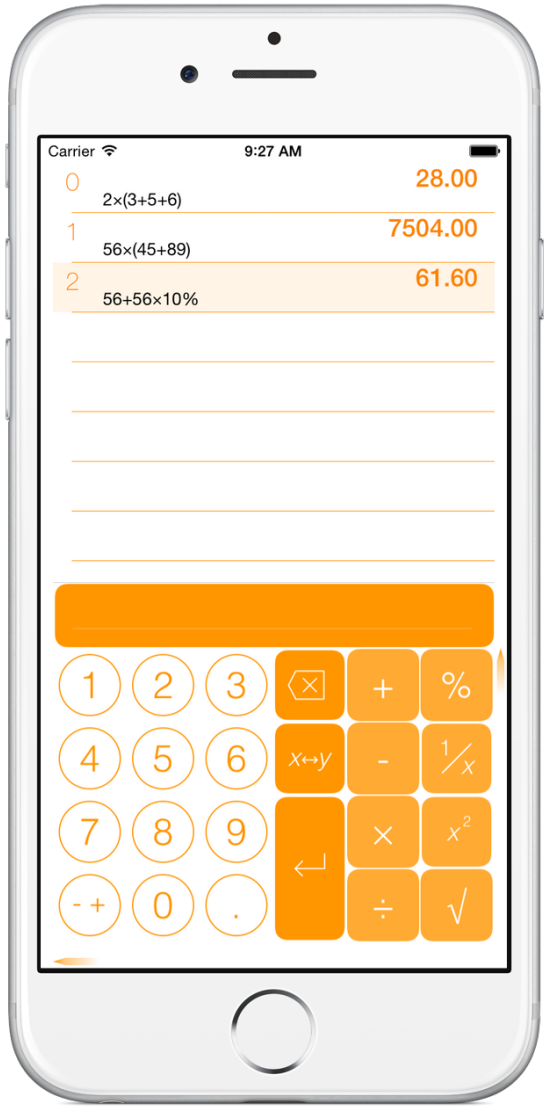
Software Testing



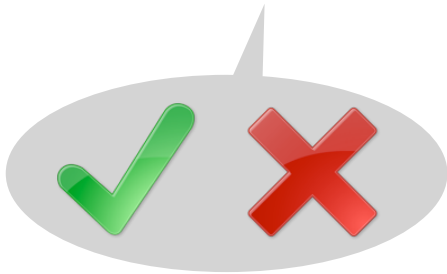
Software Testing

Input →  → Output

Software Testing

Input → 

→ Output



Software Testing

```
public void testPut() {
    Map map = …;
    map.put(1, “January”);
    map.put(2, “February”);
    …
    map.put(12, “December”);

}
```

Software Testing

```
public void testPut() {
    Map map = …;
    map.put(1, “January”);
    map.put(2, “February”);
    …
    map.put(12, “December”);
    assertEquals(map.size(), 12);
}
```

Software Testing

```
public void testPut() {
    Map map = …;
    map.put(1, “January”);
    map.put(2, “February”);
    …
    map.put(12, “December”);
    assertEquals(map.size(), 12);
}
```

Test input

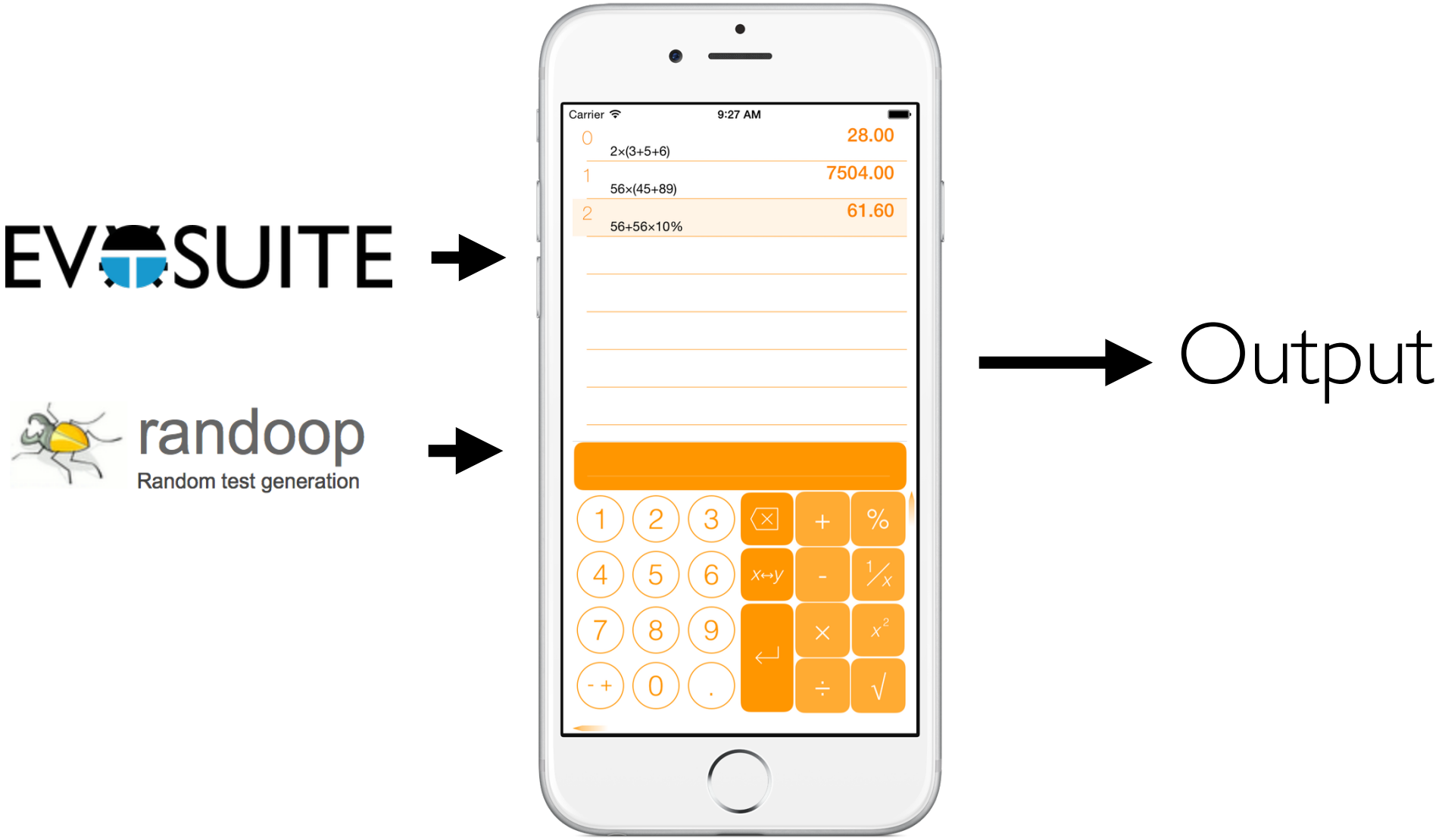
Software Testing

```
public void testPut() {
    Map map = …;
    map.put(1, “January”);
    map.put(2, “February”);
    …
    map.put(12, “December”);
    assertEquals(map.size(), 12);
}
```

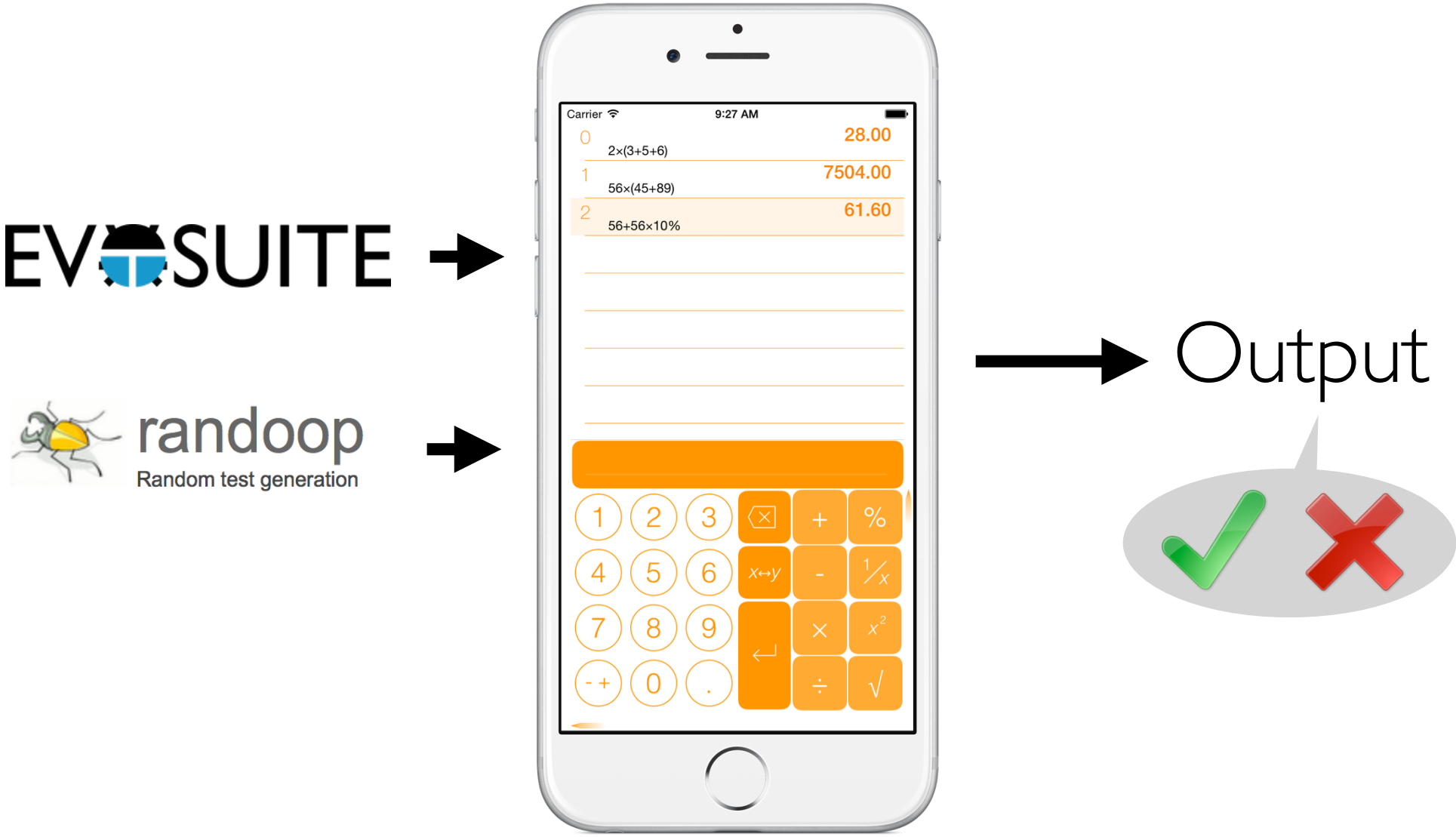
Test input

Test oracle

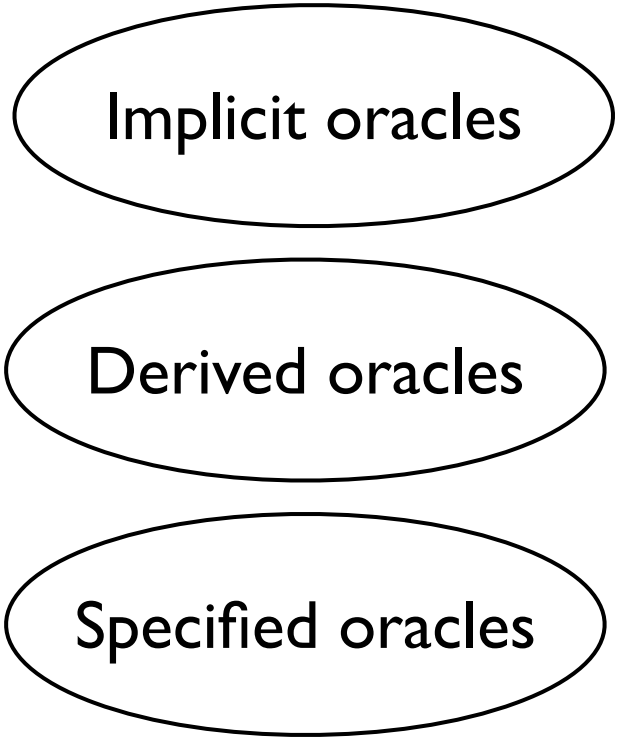
Software Testing



Software Testing

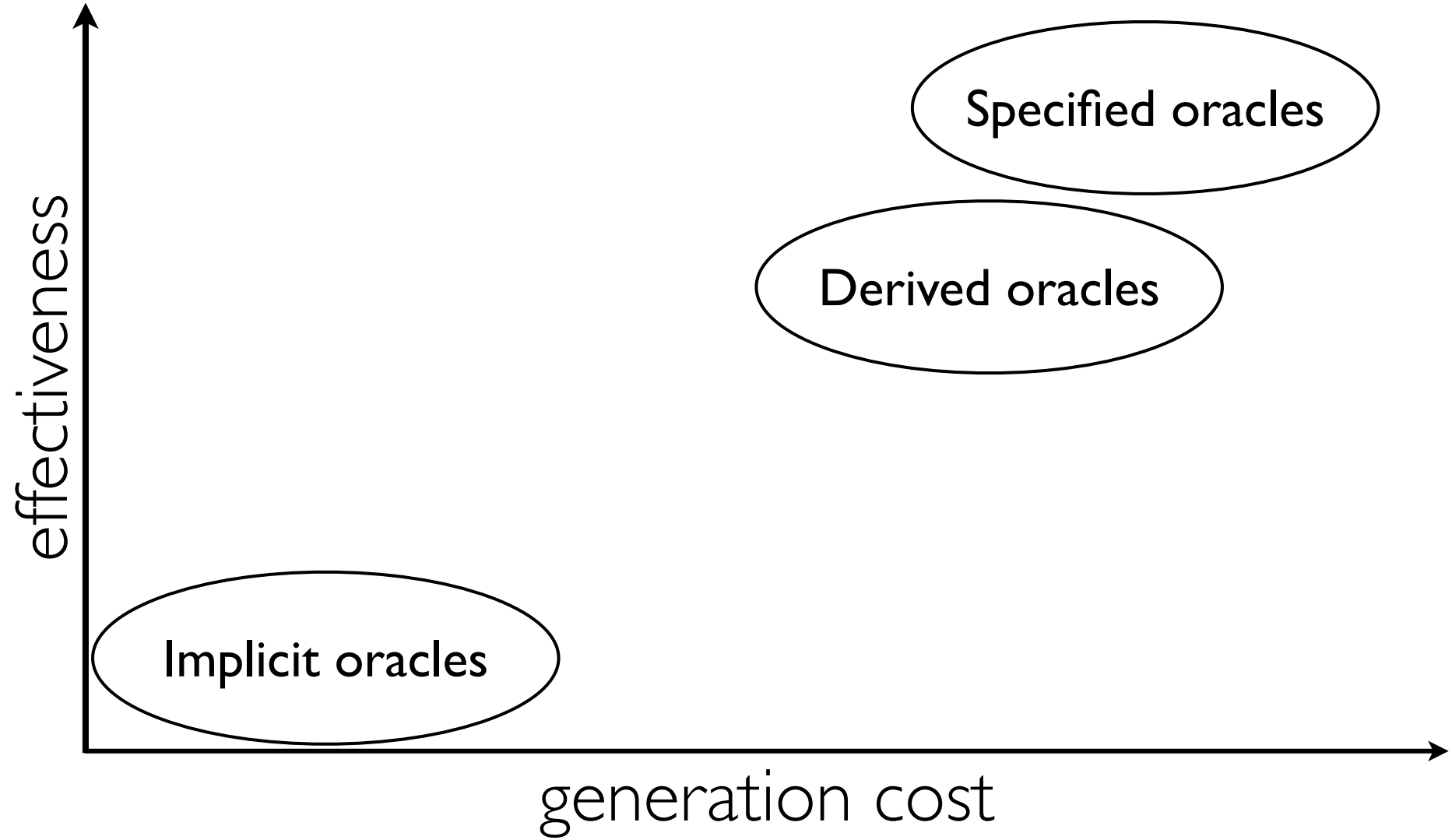


Automated Test Oracles

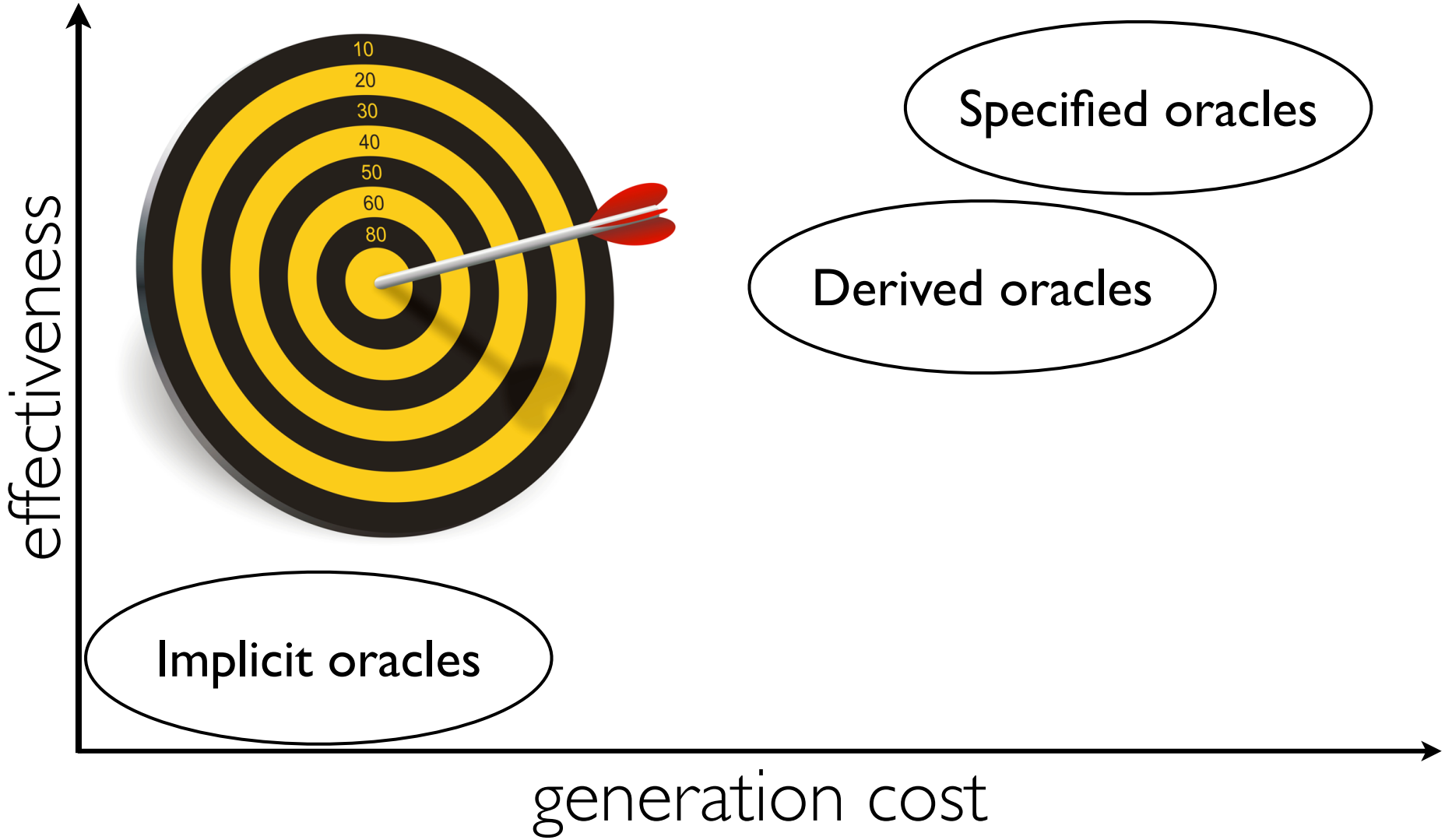


Barr et al. The Oracle Problem in Software Testing: A Survey. IEEE Transactions on Software Engineering. 2014.

Automated Test Oracles



Automated Test Oracles



Redundancy as Test Oracle

```
Map.put(key,value) ≡ Map.putAll(key, Arrays.asList(value))
```

Redundancy as Test Oracle

```
Map.put(key,value) ≡ Map.putAll(key, Arrays.asList(value))
```

```
res = map.put(1, “January”)
```

```
res = map.putAll(1,
    Arrays.asList(“January”))
```

Redundancy as Test Oracle

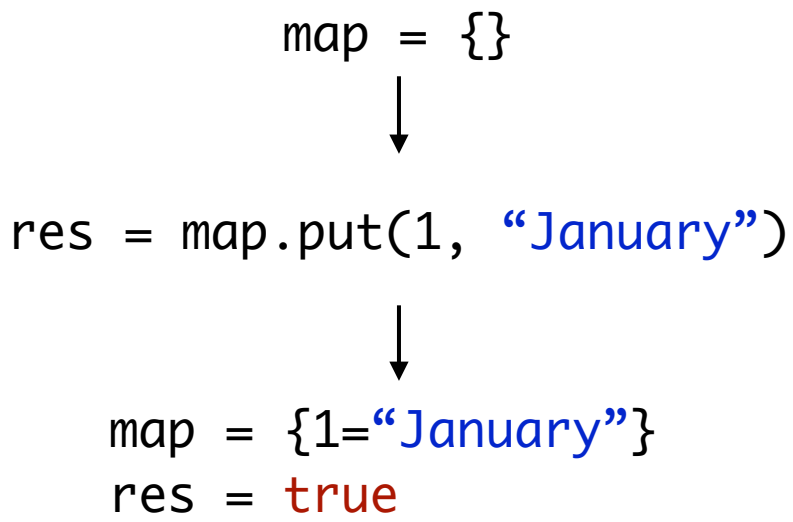
```
Map.put(key,value) ≡ Map.putAll(key, Arrays.asList(value))
```

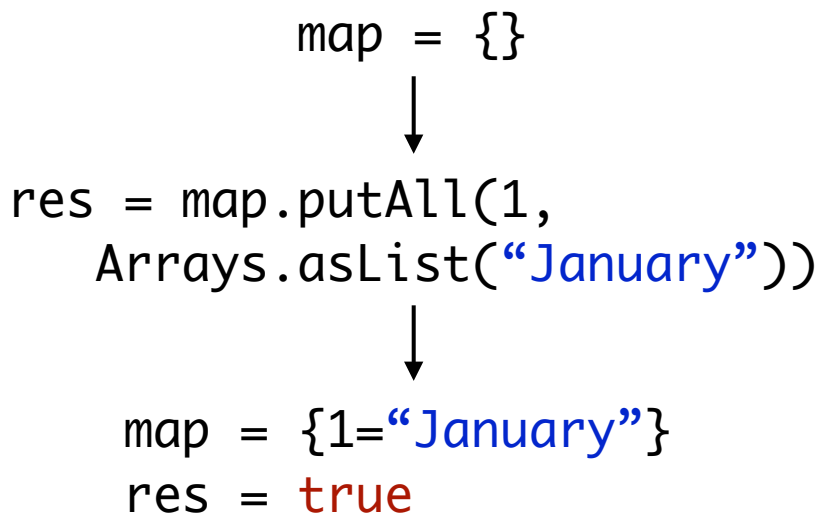
```
map = {}
   ↓
res = map.put(1, “January”)
```

```
map = {}
   ↓
res = map.putAll(1,
   Arrays.asList(“January”))
```

Redundancy as Test Oracle

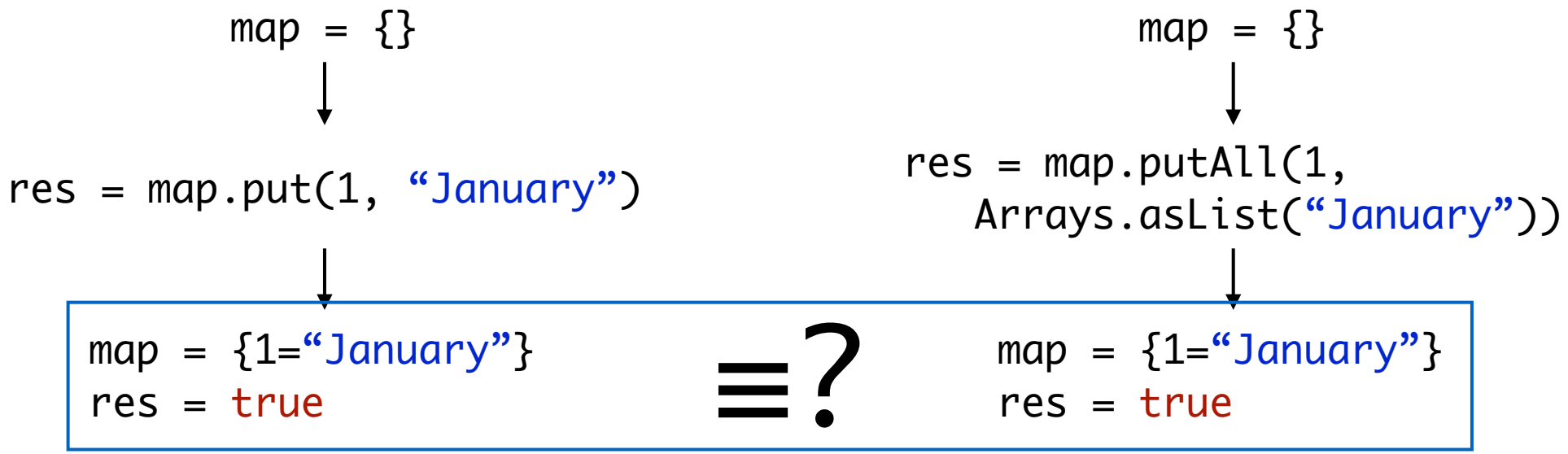
```
Map.put(key,value) ≡ Map.putAll(key, Arrays.asList(value))
```





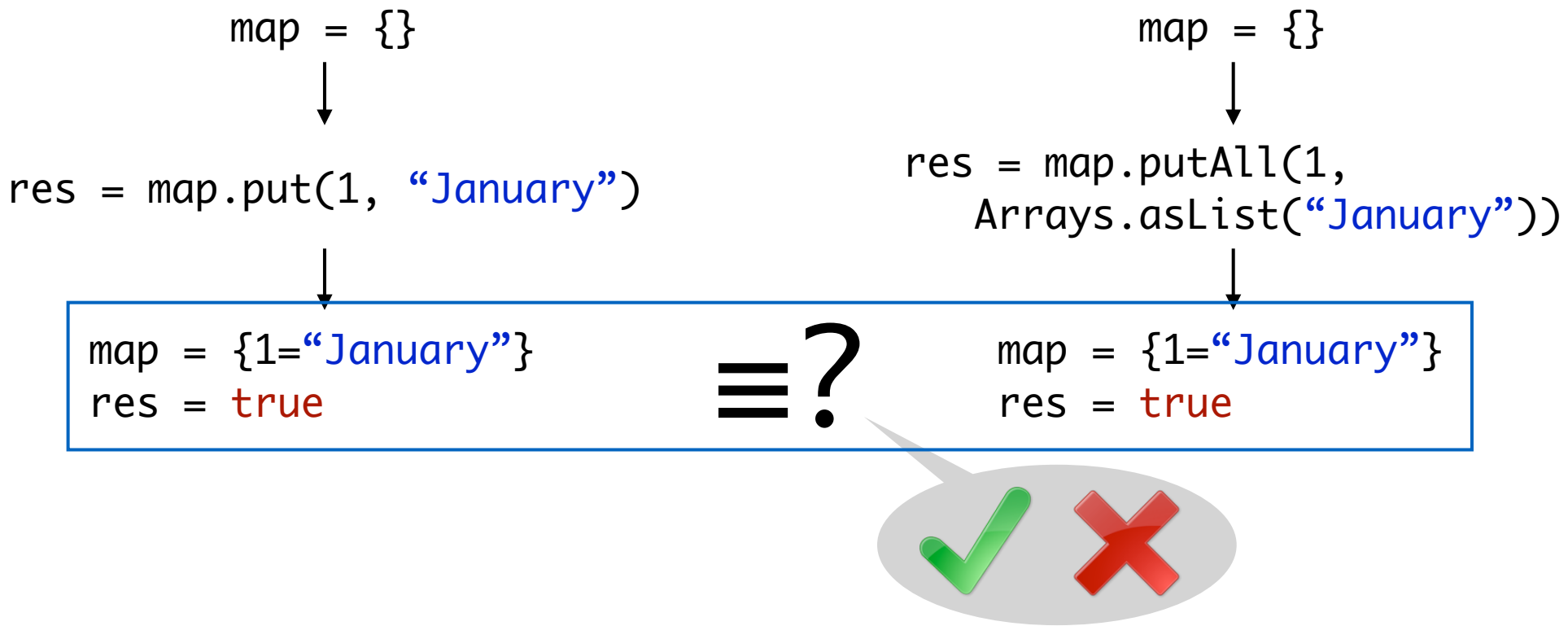
Redundancy as Test Oracle

```
Map.put(key,value) ≡ Map.putAll(key, Arrays.asList(value))
```



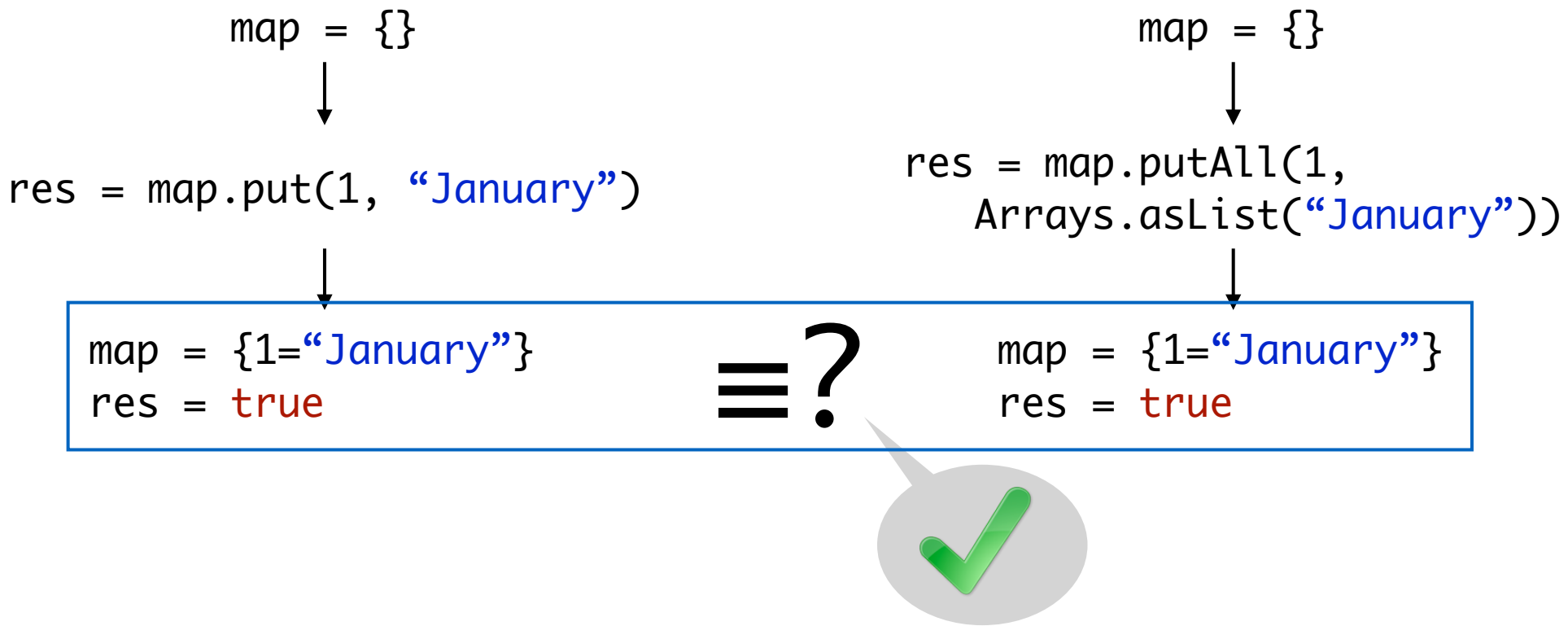
Redundancy as Test Oracle

```
Map.put(key,value) ≡ Map.putAll(key, Arrays.asList(value))
```



Redundancy as Test Oracle

```
Map.put(key,value) ≡ Map.putAll(key, Arrays.asList(value))
```



Redundancy as Test Oracle

```
Map.put(key,value) ≡ Map.putAll(key, Arrays.asList(value))
```

```
res = map.put(1, “January”)
```

```
res = map.putAll(1,
    Arrays.asList(“January”))
```

Redundancy as Test Oracle

```
Map.put(key,value) ≡ Map.putAll(key, Arrays.asList(value))
```

```
res = map.put(1, “January”)
```

```
res = map.putAll(1,
    Arrays.asList(“January”))
```

Redundancy as Test Oracle

```
Map.put(key,value) ≡ Map.putAll(key, Arrays.asList(value))
```

```
map = {}
   ↓
res = map.put(1, “January”)
```

```
map = {}
   ↓
res = map.putAll(1,
    Arrays.asList(“January”))
```

Redundancy as Test Oracle

```
Map.put(key,value) ≡ Map.putAll(key, Arrays.asList(value))
```

```
map = {}
   ↓
res = map.put(1, “January”)
   ↓
map = {1=“January”}
res = true
```

```
map = {}
   ↓
res = map.putAll(1,
    Arrays.asList(“January”))
   ↓
map = {}
res = true
```

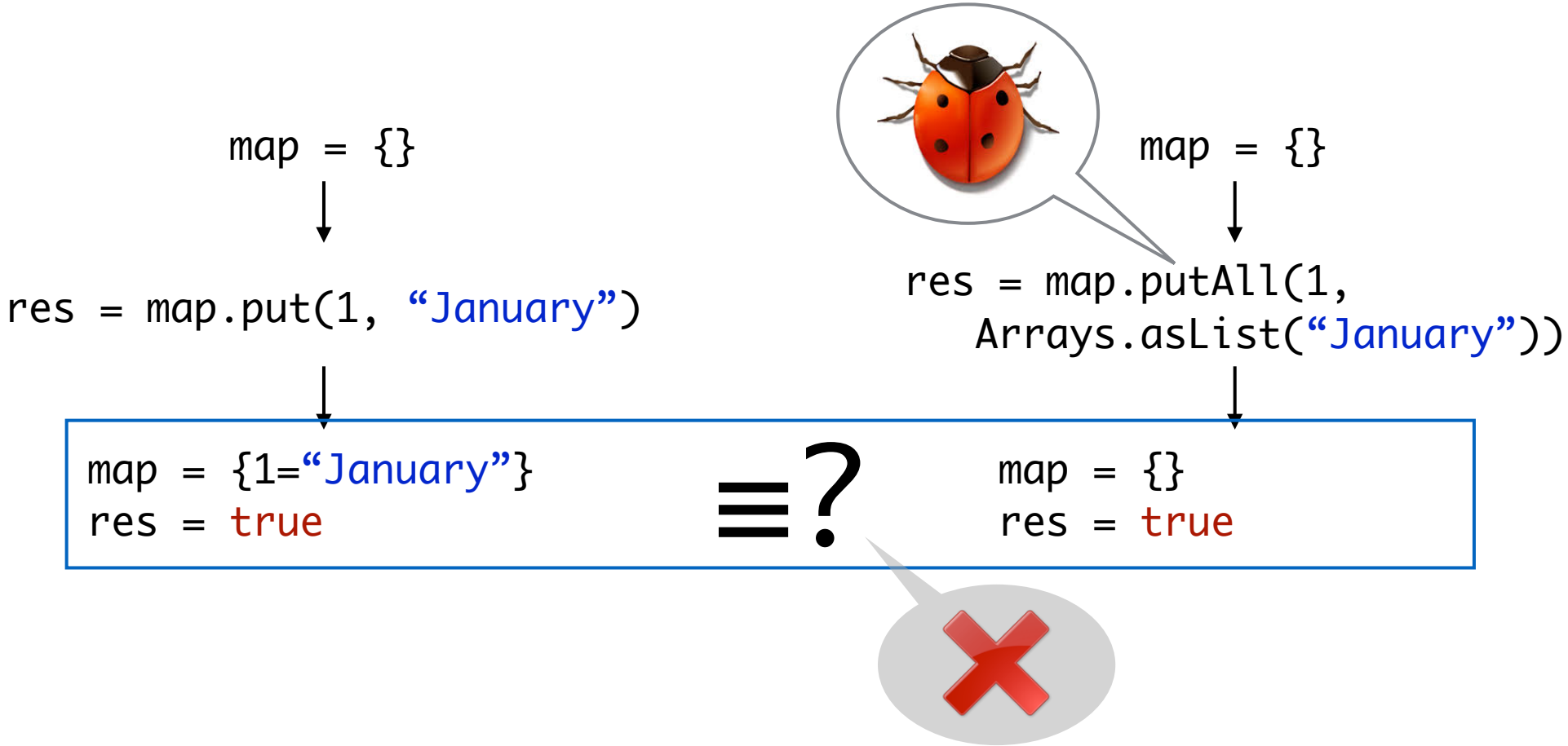
Redundancy as Test Oracle

```
Map.put(key,value) ≡ Map.putAll(key, Arrays.asList(value))
```

```
map = {}                          map = {}
  ↓                                 ↓
res = map.put(1, “January”)       res = map.putAll(1,
                                      Arrays.asList(“January”))
  ↓                                 ↓
map = {1=“January”}      ≡?       map = {}
res = true                        res = true
```

Redundancy as Test Oracle

```
Map.put(key,value) ≡ Map.putAll(key, Arrays.asList(value))
```



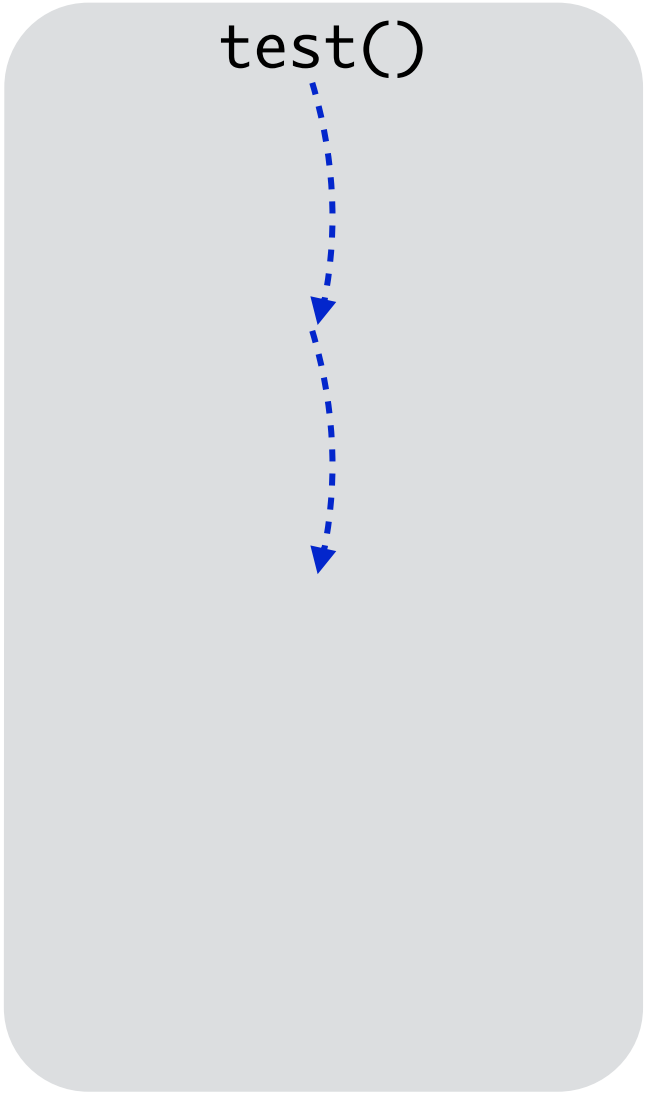
Redundancy as Test Oracle

```
test()
```

```
test() {
  …
  m();
  …
}
```

```
m ≡ m'
```

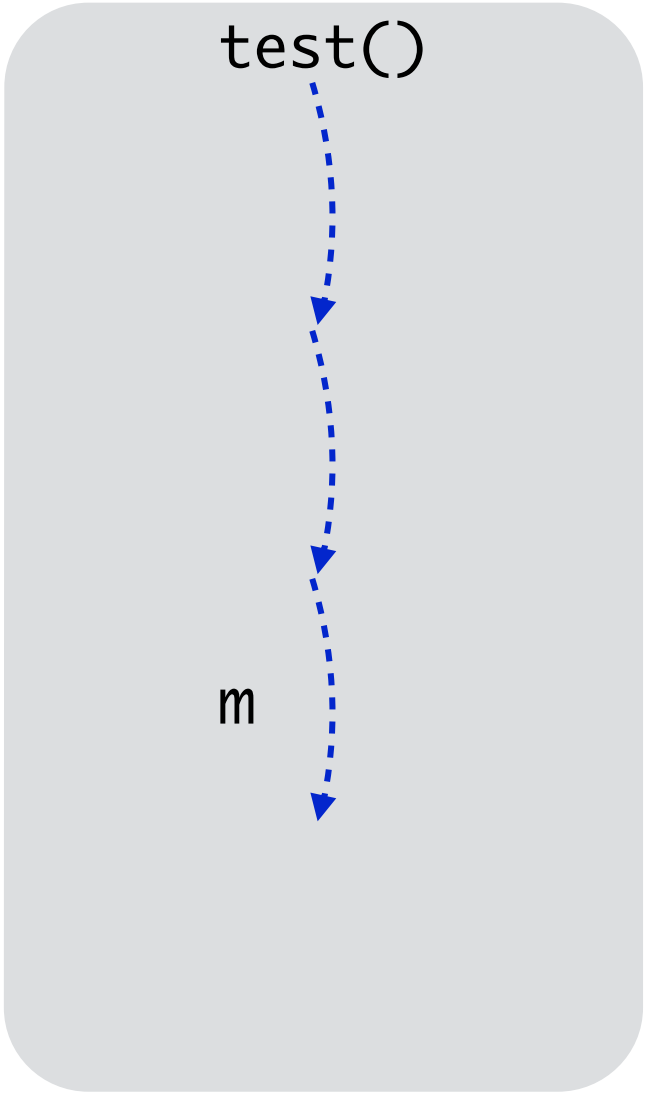
Redundancy as Test Oracle




```
test() {
  …
  m();
  …
}
```

```
m ≡ m'
```

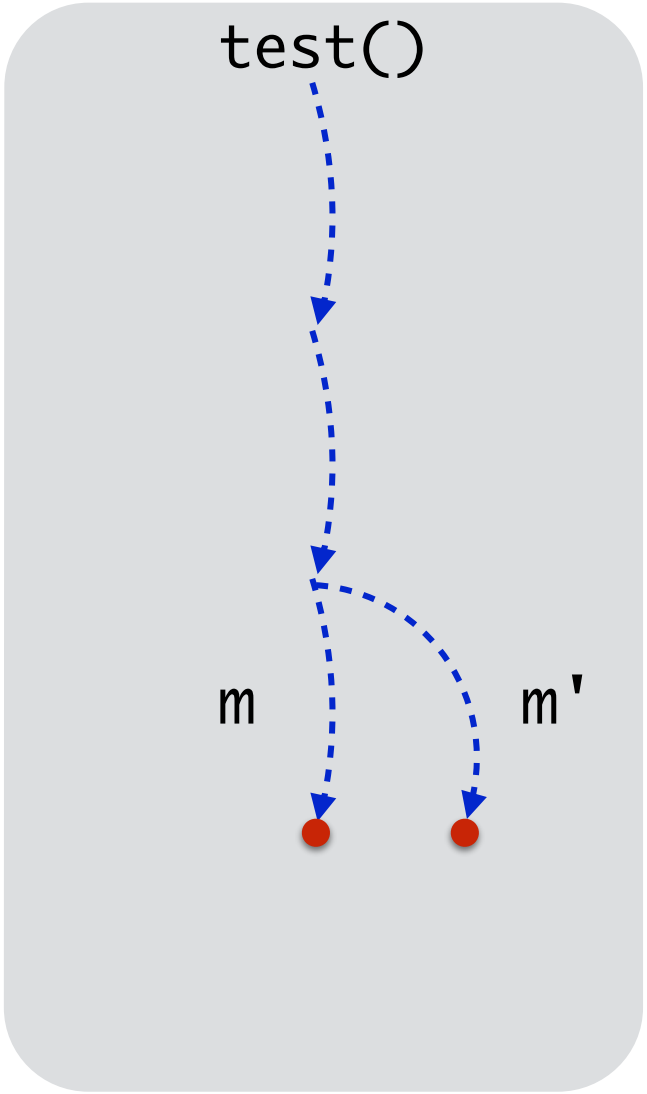
Redundancy as Test Oracle




```
test() {
  …
  m();
  …
}
```

```
m ≡ m'
```

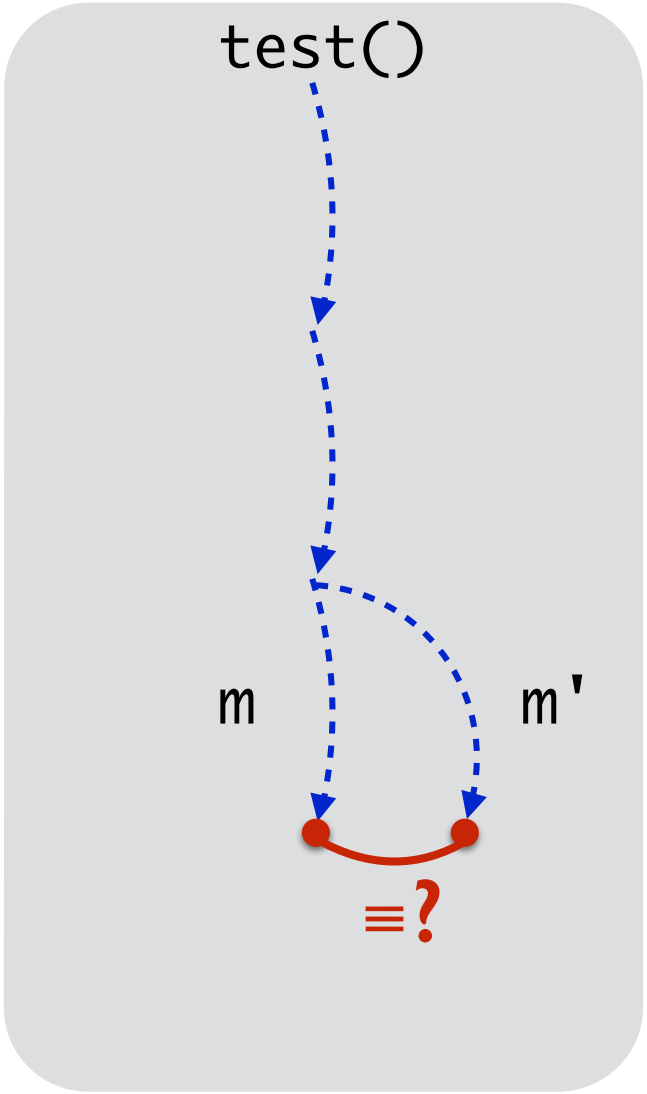
Redundancy as Test Oracle




```
test() {
  …
  m();
  …
}
```

```
m ≡ m'
```

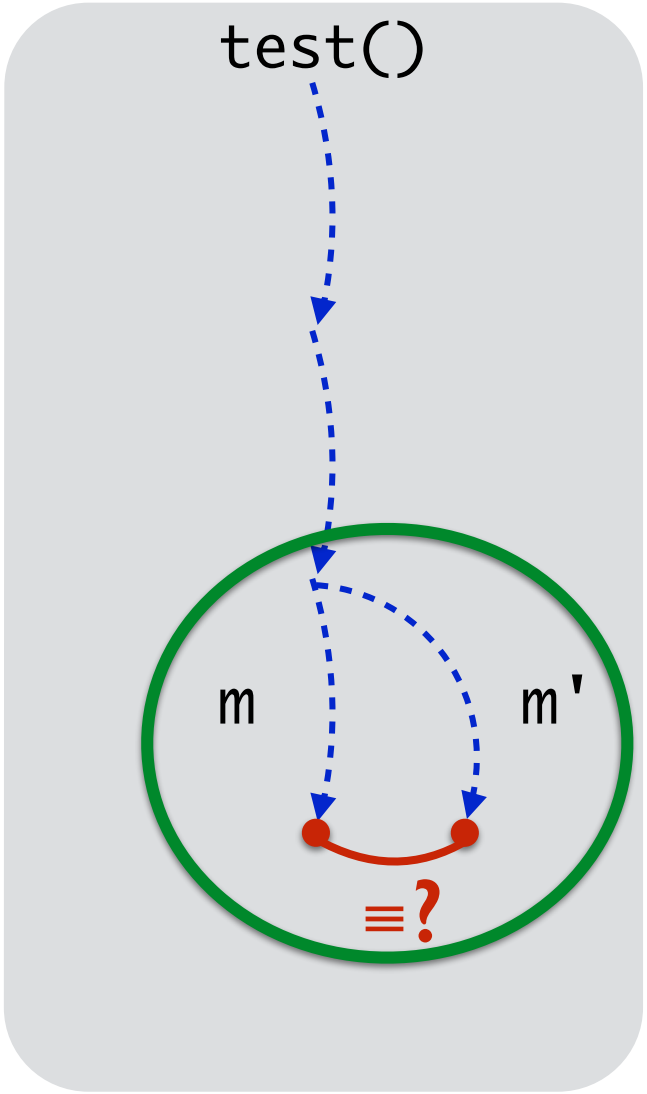
Redundancy as Test Oracle




```
test() {
  …
  m();
  …
}
```

```
m ≡ m'
```

Redundancy as Test Oracle

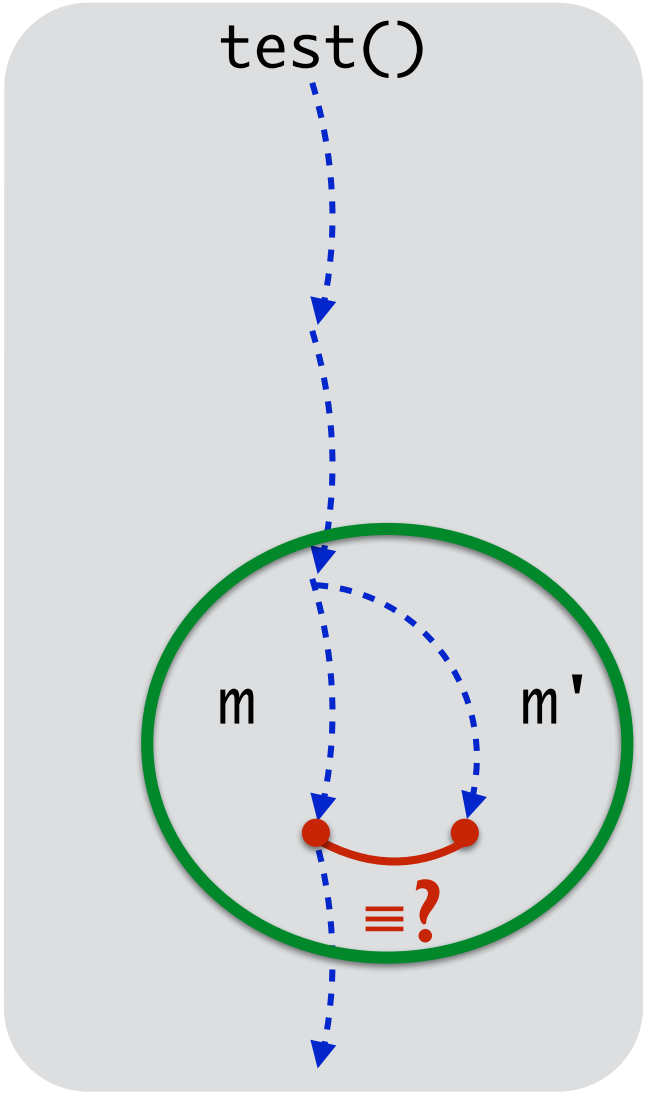



```
test() {
  …
  m();
  …
}
```

```
m ≡ m'
```

Cross-Checking Oracle

Redundancy as Test Oracle

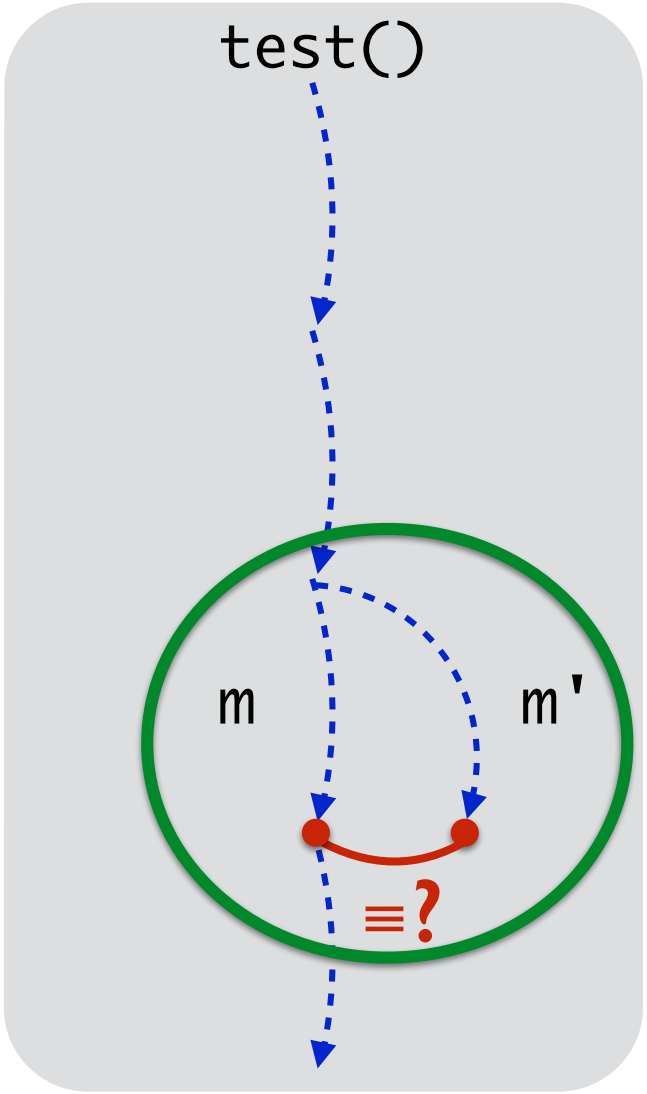



```
test() {
  …
  m();
  …
}
```

```
m ≡ m'
```

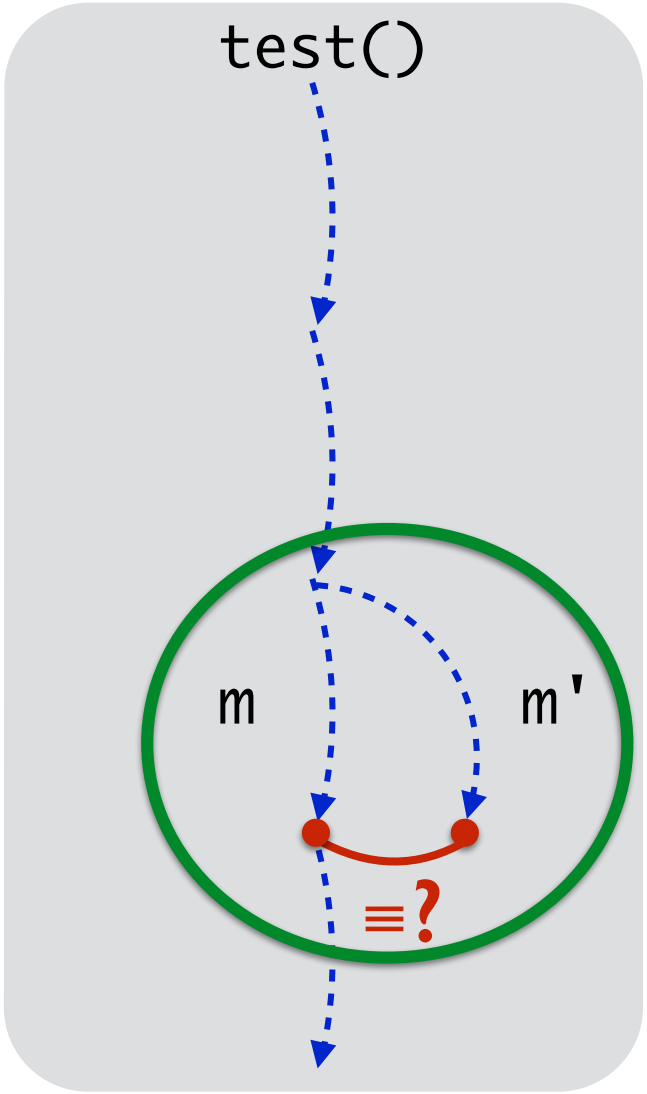
Cross-Checking Oracle

Cross-Checking Oracles



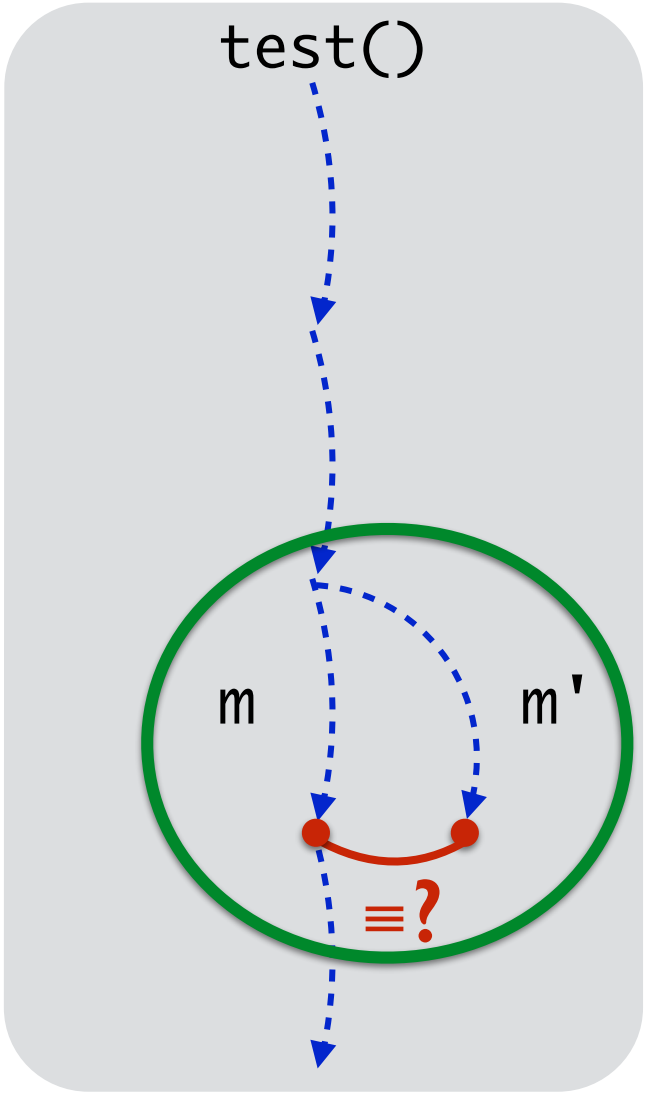
1. Automatic deployment

Cross-Checking Oracles



1. Automatic deployment
2. Cross-check execution

Cross-Checking Oracles



1. Automatic deployment
2. Cross-check execution
3. Equivalence check

Automatic Deployment

```
public class ArrayListMultimapTest {
  public void testPut() {
    Multimap<String, Integer> multimap = create();
    multimap.put(”foo”, 1);
    multimap.put(”foo”, 3);
    assertTrue(…);
    assertFalse(…);
  }
}
```

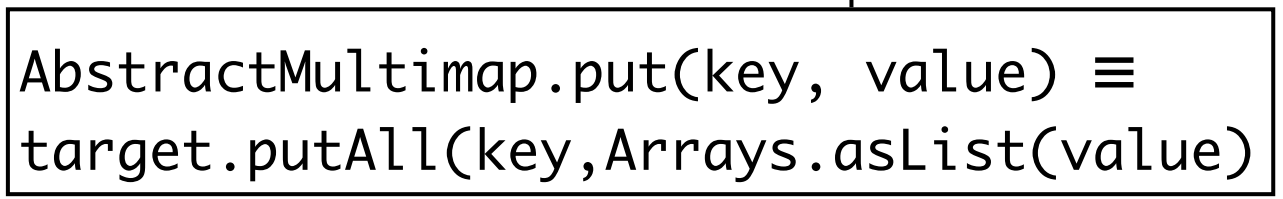
Automatic Deployment

```
public class ArrayListMultimapTest {
  public void testPut() {
    Multimap<String, Integer> multimap = create();
    multimap.put(”foo”, 1);
    multimap.put(”foo”, 3);
    assertTrue(…);
    assertFalse(…);
  }
}
```

```
AbstractMultimap.put(key, value) ≡
target.putAll(key,Arrays.asList(value)
```

Automatic Deployment

```
public class ArrayListMultimapTest {
  public void testPut() {
    Multimap<String, Integer> multimap = create();
    multimap.put(”foo”, 1);
    multimap.put(”foo”, 3);
    assertTrue(…);
    assertFalse(…);
  }
}
```



Automatic Deployment

```
public class ExampleOracle extends CrossCheckOracle {

  @PointCut("call(boolean com.google.collect.AbstractMultimap.put(Object, Object))")
  public boolean advice(Multimap<Object, Object> map, Object key, Object value) {
    this.target = map;
    this.key = key;
    this.value = value;
    return executeCrossCheck();
  }

  @Original
  boolean originalCall() {
    return target.put(key,value);
  }

  @Equivalent
  boolean equivalentCode() {
    return target.putAll(key,Arrays.asList(value));
  }
}
```

Automatic Deployment

```
public class ExampleOracle extends CrossCheckOracle {

  @PointCut("call(boolean com.google.collect.AbstractMultimap.put(Object, Object))")
  public boolean advice(Multimap<Object, Object> map, Object key, Object value) {
    this.target = map;
    this.key = key;
    this.value = value;
    return executeCrossCheck();
  }

  @Original
  boolean originalCall() {
    return target.put(key,value);
  }

  @Equivalent
  boolean equivalentCode() {
    return target.putAll(key,Arrays.asList(value));
  }
}
```

AbstractMultimap.put(key, value) ≡
target.putAll(key,Arrays.asList(value)

Automatic Deployment

```
public class ExampleOracle extends CrossCheckOracle {

  @PointCut("call(boolean com.google.collect.AbstractMultimap.put(Object, Object))")
  public boolean advice(Multimap<Object, Object> map, Object key, Object value) {
    this.target = map;
    this.key = key;
    this.value = value;
    return executeCrossCheck();
  }

  @Original
  boolean originalCall() {
    return target.put(key,value);
  }

  @Equivalent
  boolean equivalentCode() {
    return target.putAll(key,Arrays.asList(value));
  }
}
```

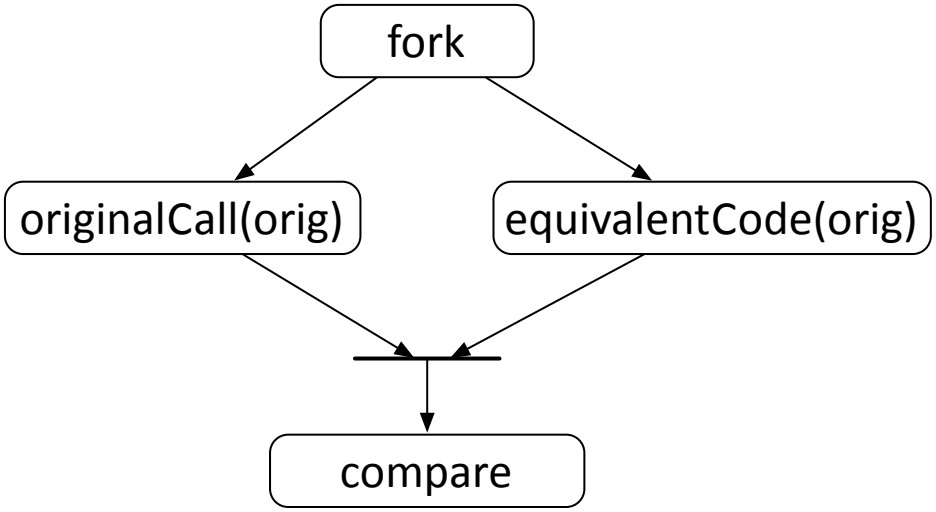
AbstractMultimap.put(key, value) ≡
target.putAll(key,Arrays.asList(value)

Automatic Deployment

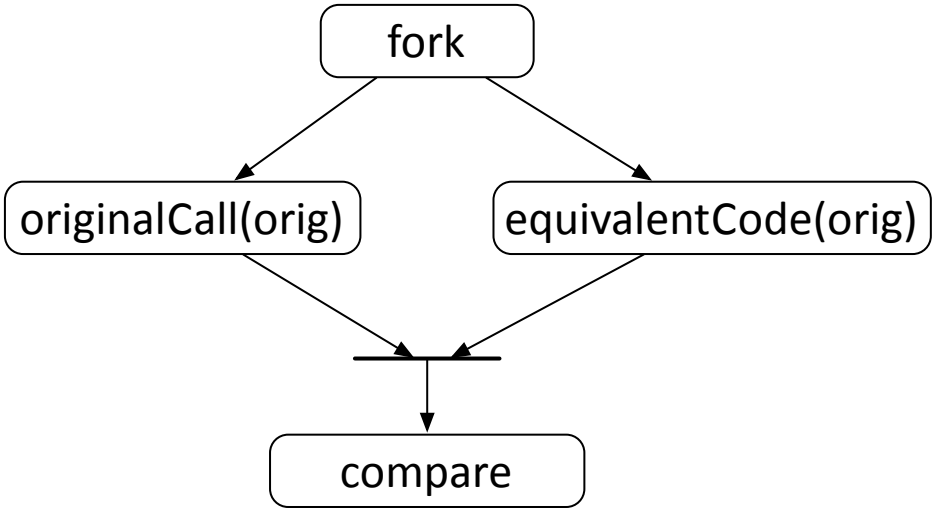
```
public class ExampleOracle extends CrossCheckOracle {

  @PointCut("call(boolean com.google.collect.AbstractMultimap.put(Object, Object))")
  public boolean advice(Multimap<Object, Object> map, Object key, Object value) {
    this.target = map;
    this.key = key;
    this.value = value;
    return executeCrossCheck();
  }

  @Original
  boolean originalCall() {
    return target.put(key,value);
  }

  @Equivalent
  boolean equivalentCode() {
    return target.putAll(key,Arrays.asList(value));
  }
}
```

AbstractMultimap.put(key, value) ≡
target.putAll(key,Arrays.asList(value)

Cross-Check Execution

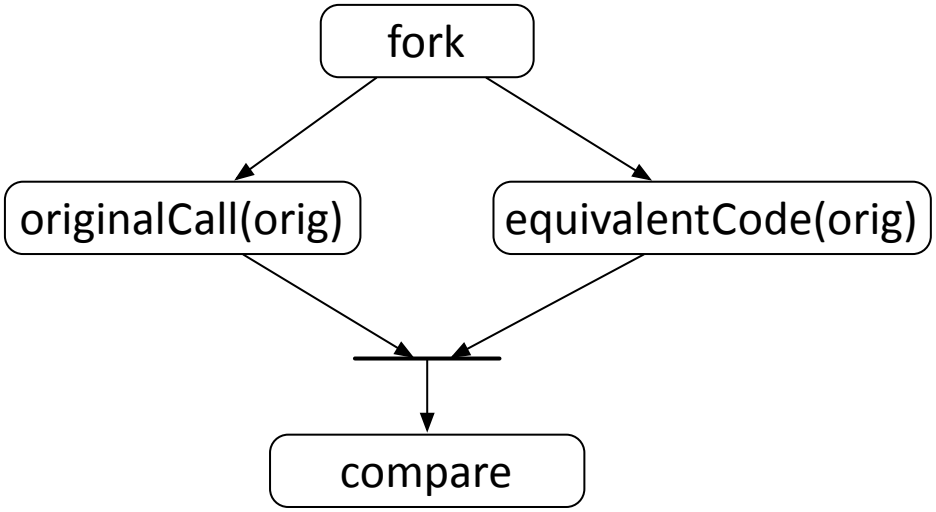


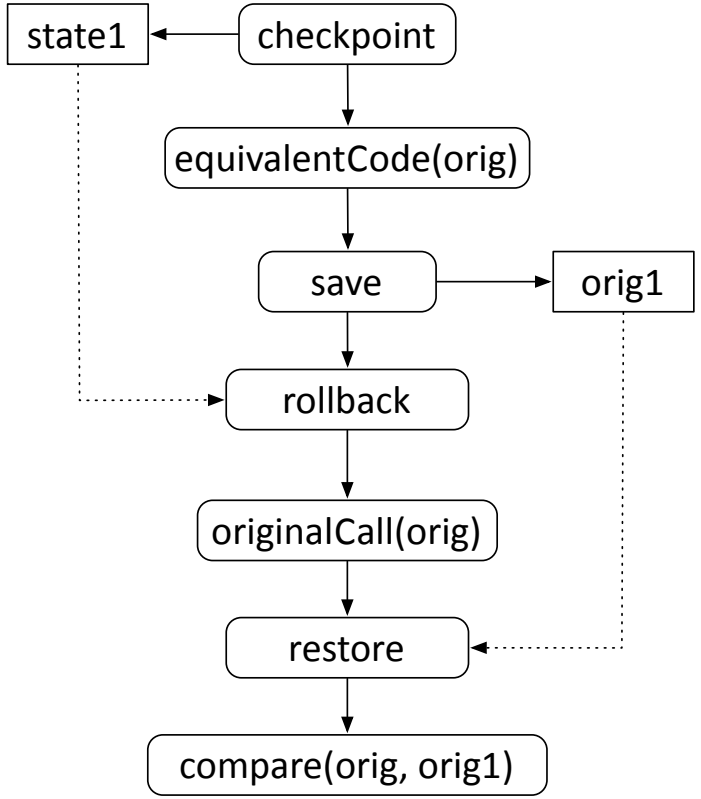
Cross-Check Execution



Fork the JVM?

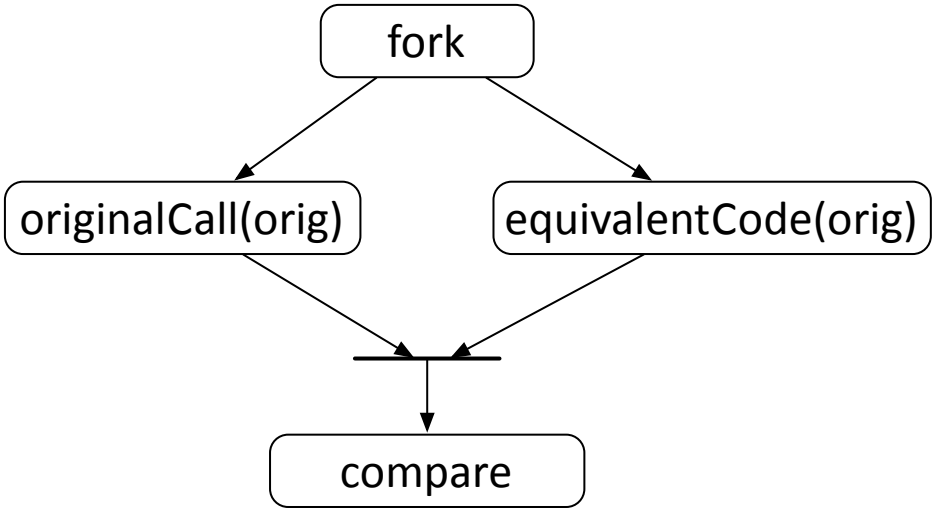
Cross-Check Execution

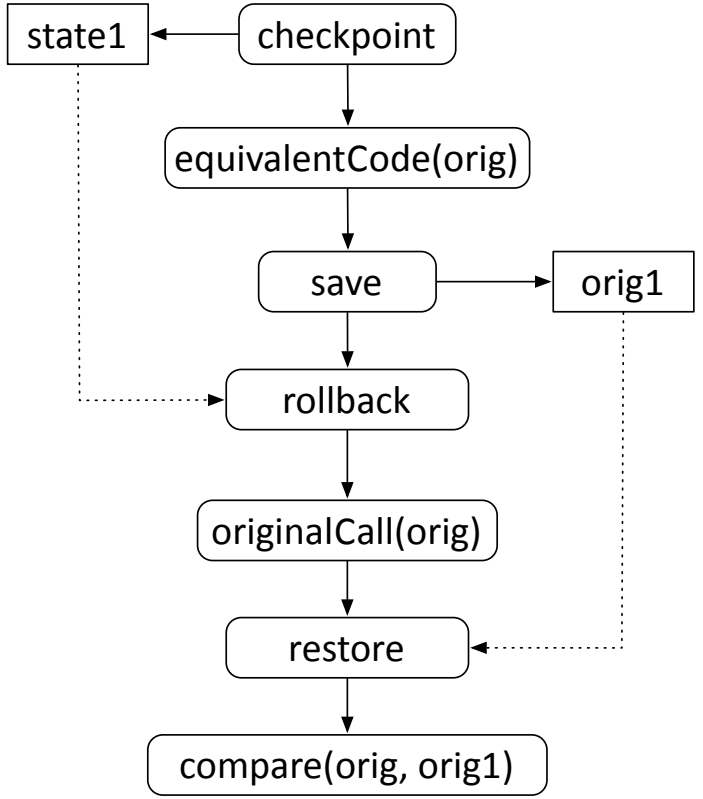




Fork the JVM?

Cross-Check Execution





Fork the JVM?

Store/load states?

Cross-Check Execution

```
public class ExampleOracle extends CrossCheckOracle {

  @PointCut("call(boolean com.google.collect.AbstractMultimap.put(Object, Object))")
  public boolean advice(Multimap<Object, Object> map, Object key, Object value) {
    ...
    return executeCrossCheck();
  }

  boolean executeCrossCheck() {
    Object target_orig = target;
    Object target_copy = deep_clone(target);

    Object orig_res = originalCall(target_orig);
    Object copy_res = equivalentCode(target_copy);

    return (orig_res ≡ copy_res) && (target_orig ≡ target_copy);
  }
}
```

Cross-Check Execution

```
public class ExampleOracle extends CrossCheckOracle {

  @PointCut("call(boolean com.google.collect.AbstractMultimap.put(Object, Object))")
  public boolean advice(Multimap<Object, Object> map, Object key, Object value) {
    ...
    return executeCrossCheck();
  }

  boolean executeCrossCheck() {
    Object target_orig = target;
    Object target_copy = deep_clone(target);

    Object orig_res = originalCall(target_orig);
    Object copy_res = equivalentCode(target_copy);

    return (orig_res ≡ copy_res) && (target_orig ≡ target_copy);
  }
}
```

Sequential execution

Cross-Check Execution

```
public class ExampleOracle extends CrossCheckOracle {

  @PointCut("call(boolean com.google.collect.AbstractMultimap.put(Object, Object))")
  public boolean advice(Multimap<Object, Object> map, Object key, Object value) {
    ...
    return executeCrossCheck();
  }

  boolean executeCrossCheck() {
    Object target_orig = target;
    Object target_copy = deep_clone(target);

    Object orig_res = originalCall(target_orig);
    Object copy_res = equivalentCode(target_copy);

    return (orig_res ≡ copy_res) && (target_orig ≡ target_copy);
  }
}
```

Deep-clone target object

Sequential execution

Cross-Check Execution

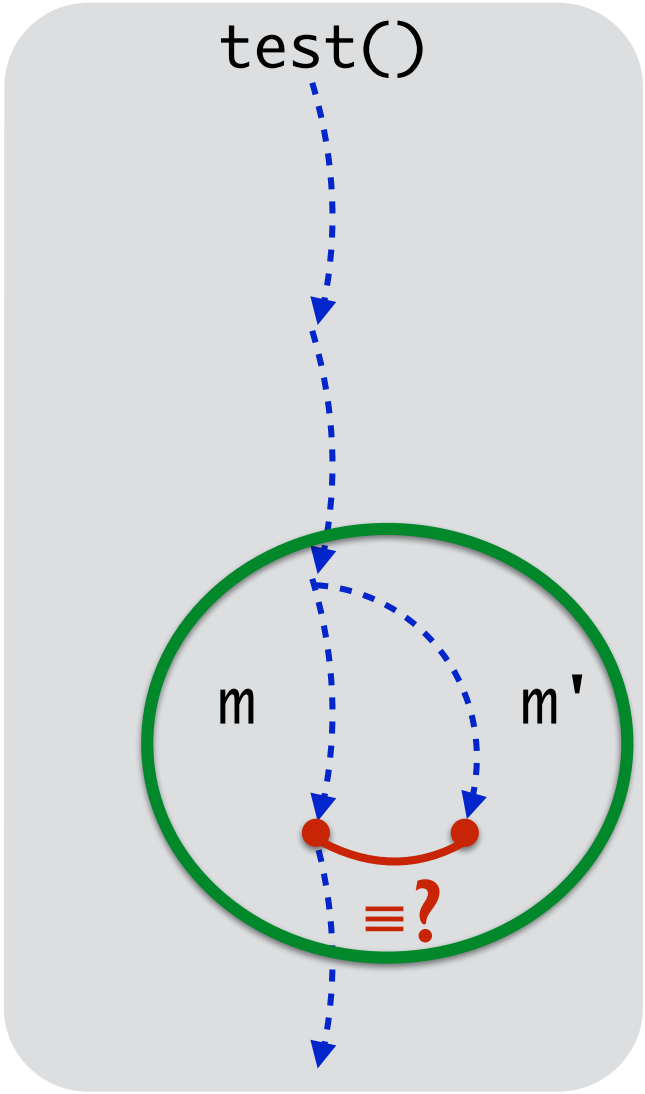
```
public class ExampleOracle extends CrossCheckOracle {

  @PointCut("call(boolean com.google.collect.AbstractMultimap.put(Object, Object))")
  public boolean advice(Multimap<Object, Object> map, Object key, Object value) {
    ...
    return executeCrossCheck();
  }

  boolean executeCrossCheck() {
    Object target_orig = target;
    Object target_copy = deep_clone(target);

    Object orig_res = originalCall(target_orig);
    Object copy_res = equivalentCode(target_copy);

    return (orig_res ≡ copy_res) && (target_orig ≡ target_copy);
  }
}
```

Deep-clone target object

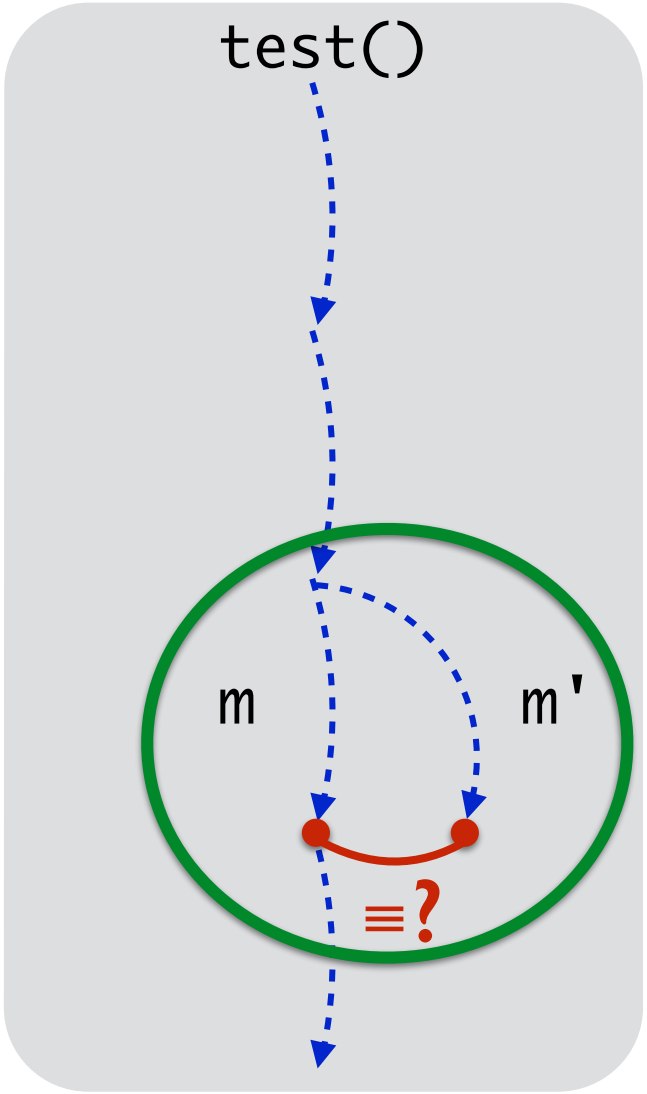
Sequential execution

Equivalence check

Equivalence Check

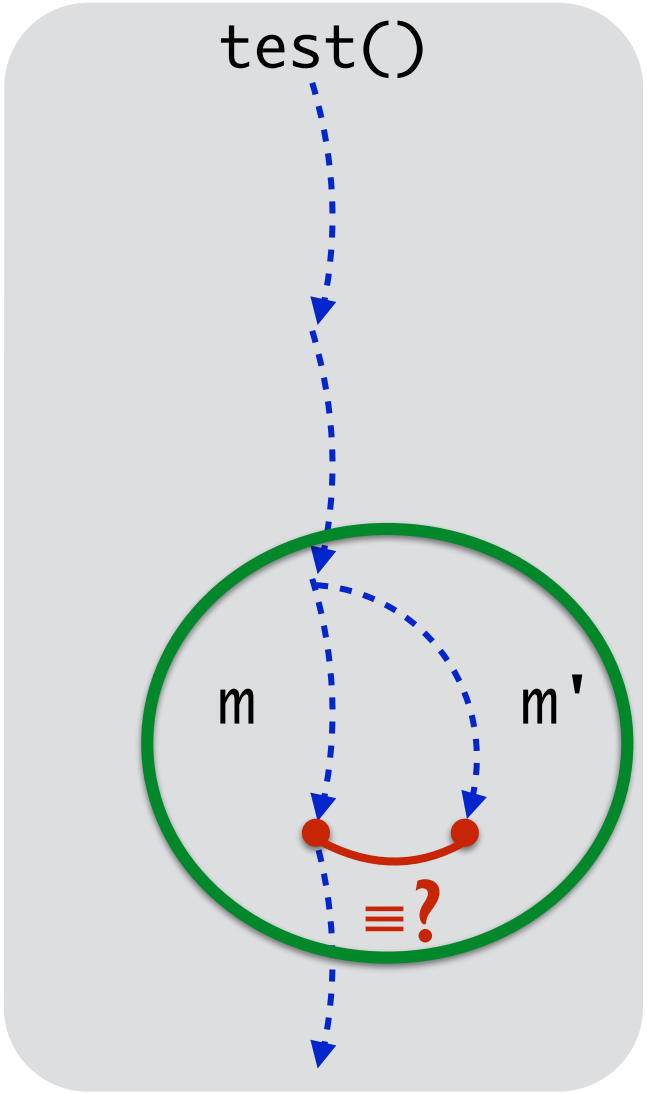


Equivalence Check



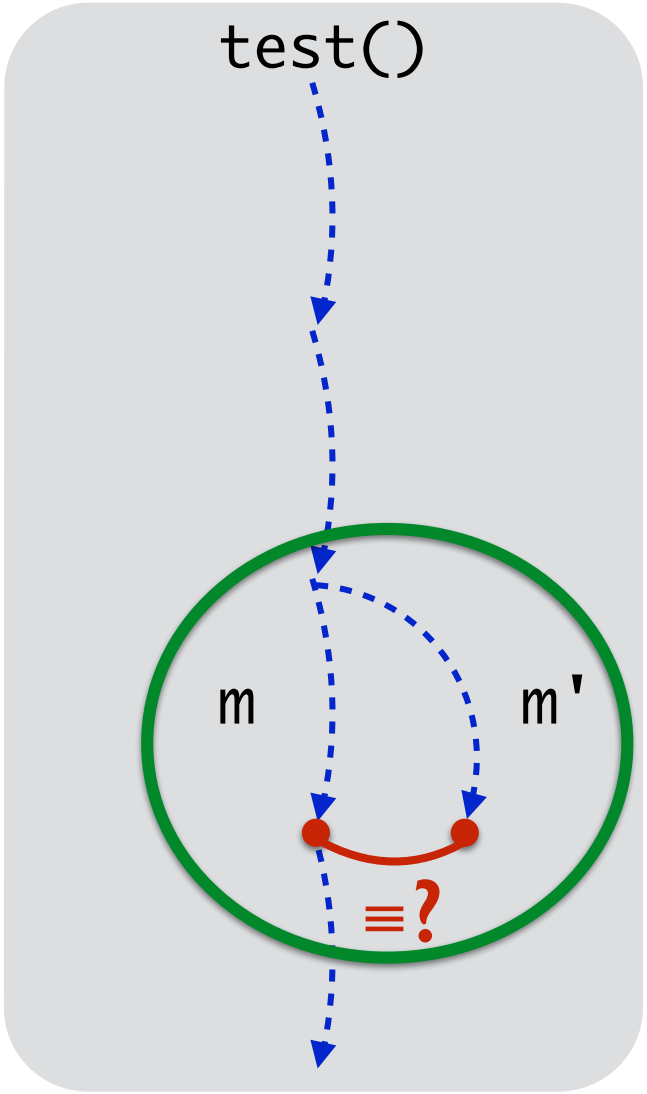

```
public boolean equals(Object object) {
    if (object == this) {
      return true;
    }
    if (object instanceof Multimap) {
      Multimap that = (Multimap) object;
      return this.map.equals(that.asMap());
    }
    return false;
}
```

Equivalence Check




```
public boolean equals(Object object) {
    if (object == this) {
      return true;
    }
    if (object instanceof Multimap) {
      Multimap that = (Multimap) object;
      return this.map.equals(that.asMap());
    }
    return false;
}


public class Object {
    ...
    public boolean equals(Object obj) {
        return (this == obj);
    }
}
```

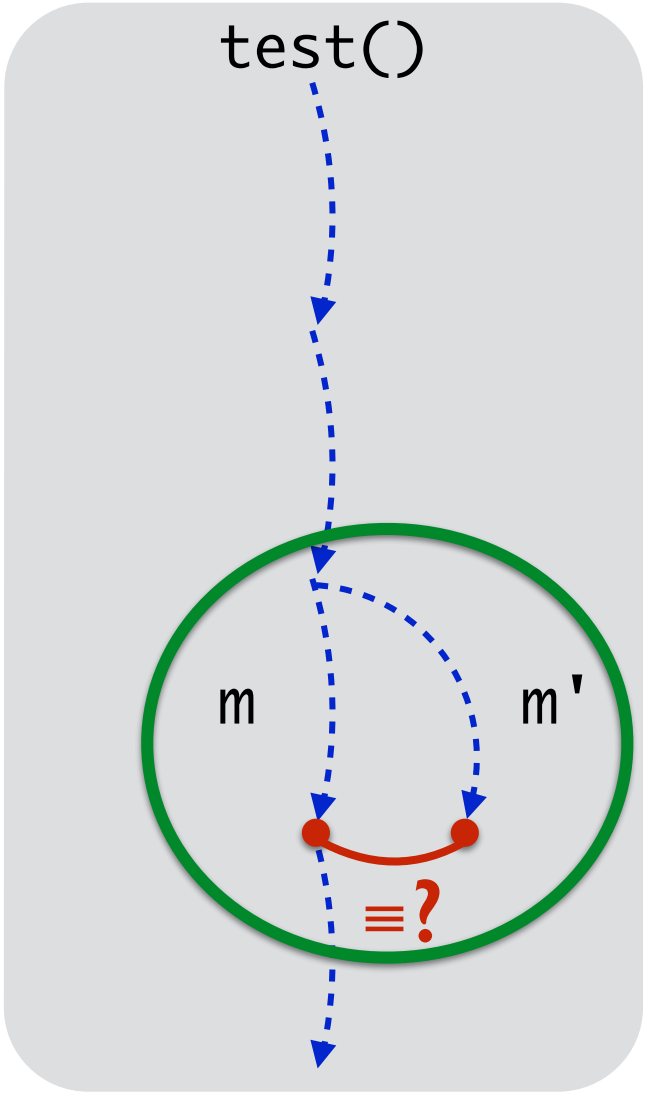
Equivalence Check




```
public boolean equals(Object object) {
    if (object == this) {
      return true;
    }
    if (object instanceof Multimap) {
      Multimap that = (Multimap) object;
      return this.map.equals(that.asMap());
    }
    return false;
}


public class Object {
    ...
    public boolean equals(Object obj) {
        return (this == obj);
    }
}
```

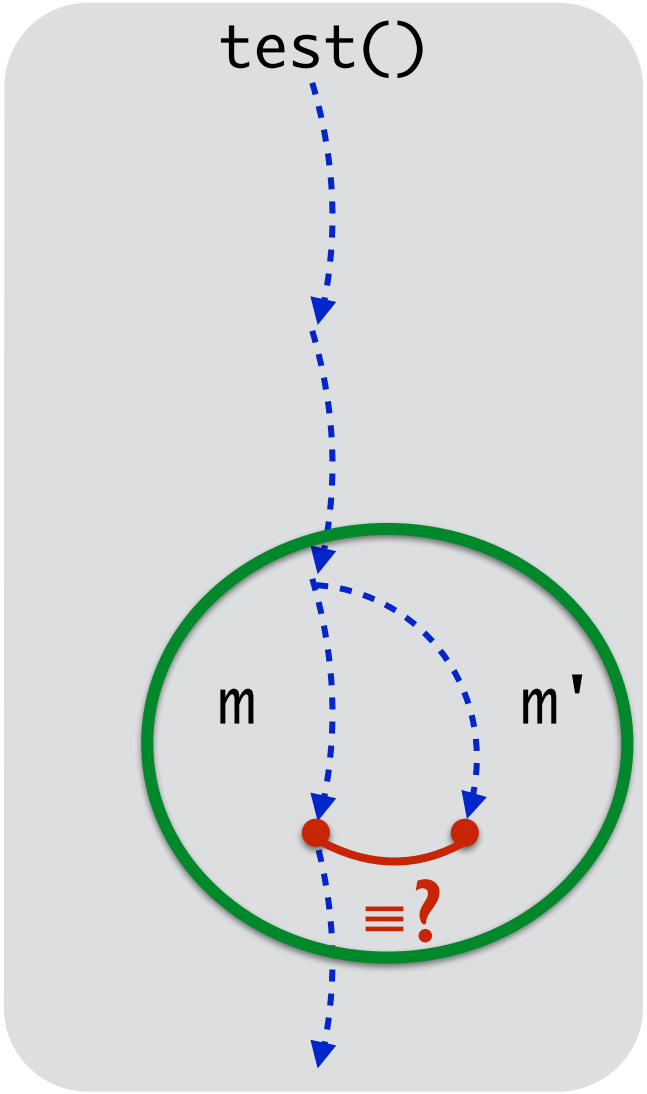
False positive!

Equivalence Check

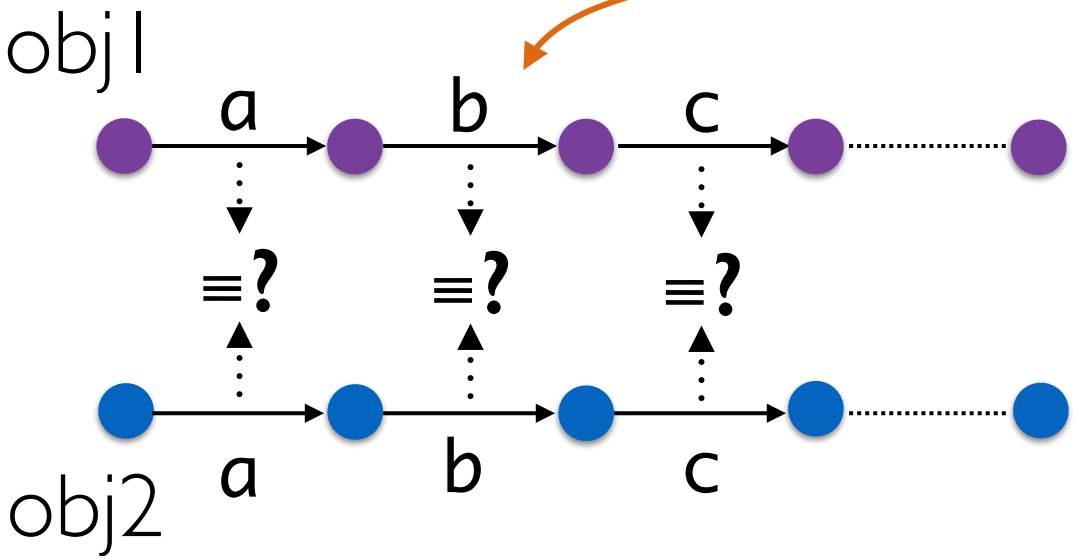



```
public boolean equals(Object object) {
    if (object == this) {
      return true;
    }
    if (object instanceof Multimap) {
      Multimap that = (Multimap) object;
      return this.map.equals(that.asMap());
    }
    return false;
}


public class Object {
    ...
    public boolean equals(Object obj) {
        return (this == obj);
    }
}
```

Equivalence Check




```
public boolean equals(Object object) {
    if (object == this) {
      return true;
    }
    if (object instanceof Multimap) {
      Multimap that = (Multimap) object;
      return this.map.equals(that.asMap());
    }
    return false;
}
```



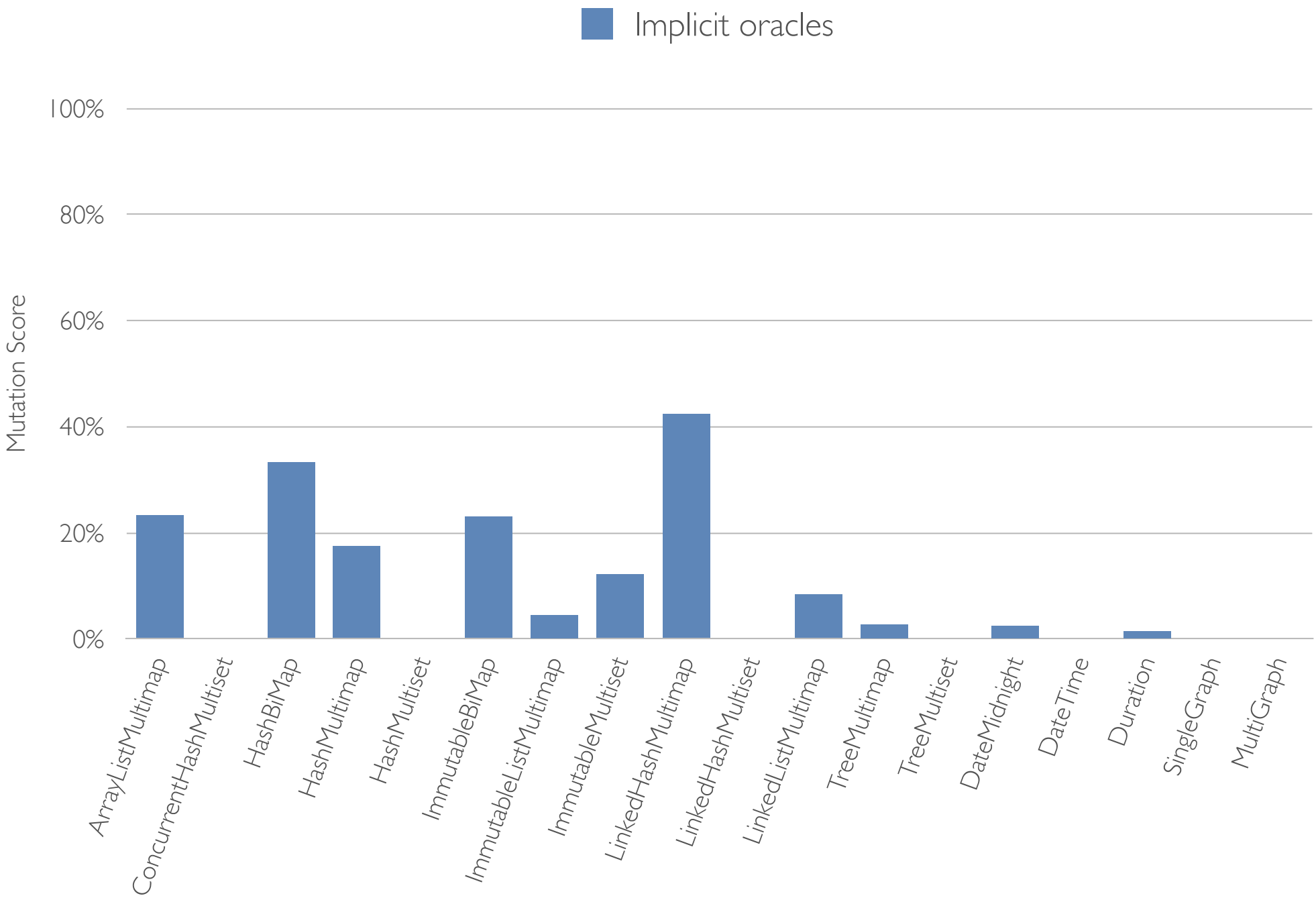
Evaluation

How **effective** are cross-checking oracles?

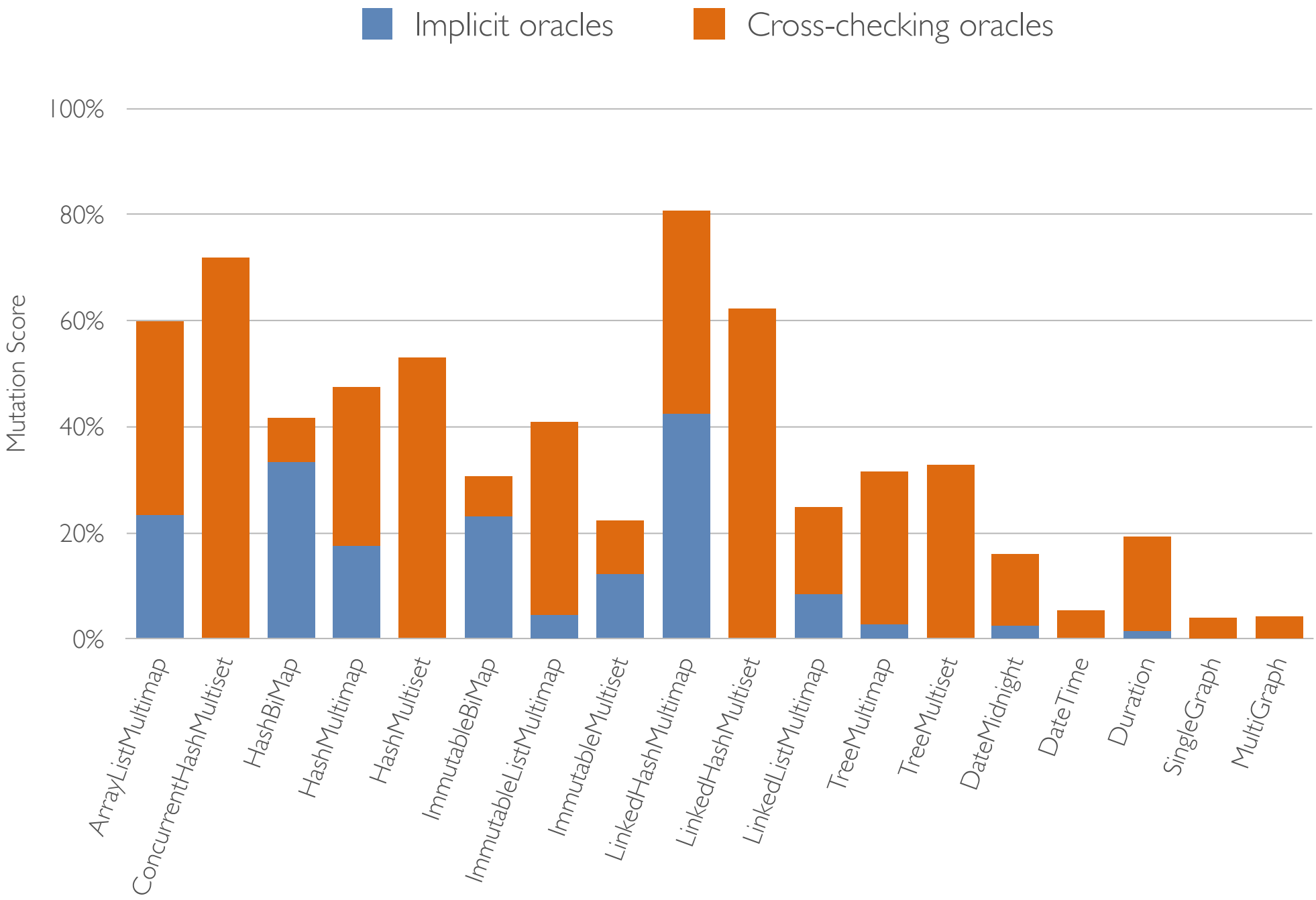
Evaluation

Subject Class		# Methods	# Equivalences
Guava	ArrayListMultimap	25	29
	ConcurrentHashMulti	27	32
	HashBiMap	20	16
	HashMultimap	24	29
	HashMultiset	26	30
	ImmutableBimap	25	19
	ImmutableListMultima	30	34
	ImmutableMultiset	32	50
	LinkedHashMultimap	24	30
	LinkedHashMultiset	26	31
	LinkedListMultimap	24	29
	TreeMultimap	26	28
	TreeMultiset	35	37
JodaTime	DateMidnight	118	20
	DateTime	153	27
	Duration	44	6
GraphStream	SingleGraph	107	41
	MultiGraph	107	41

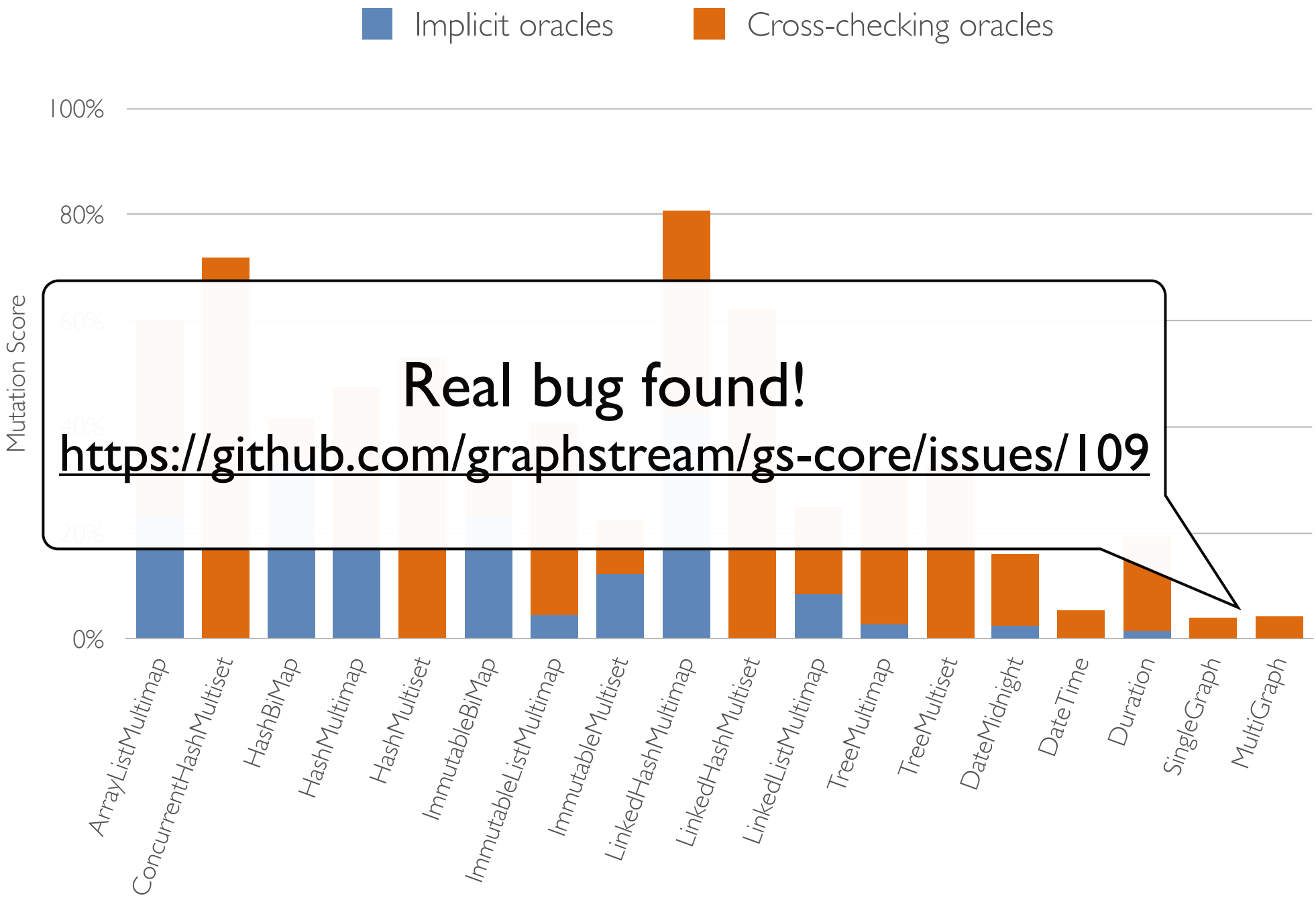
Cross-Checking & Implicit Oracles



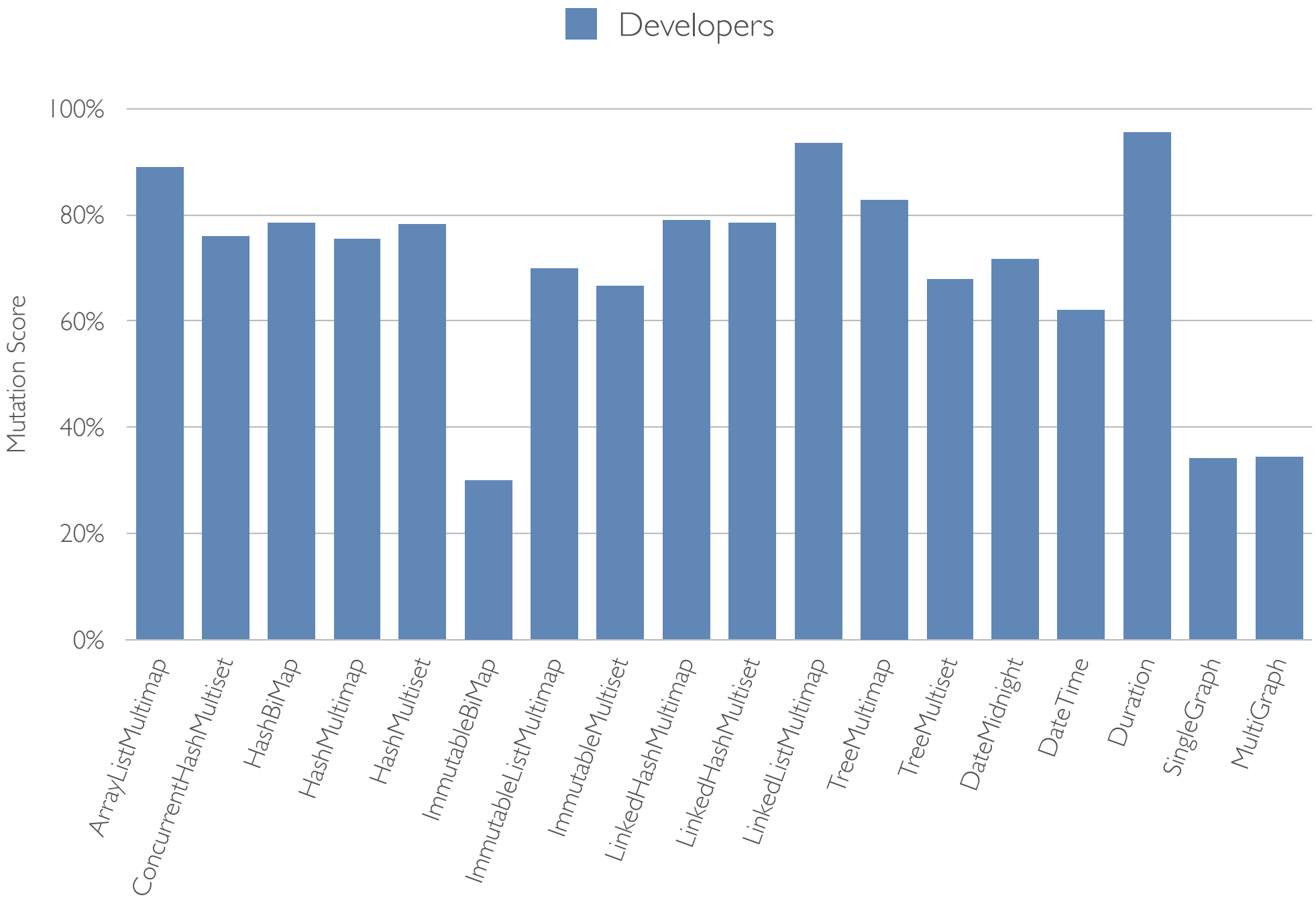
Cross-Checking & Implicit Oracles



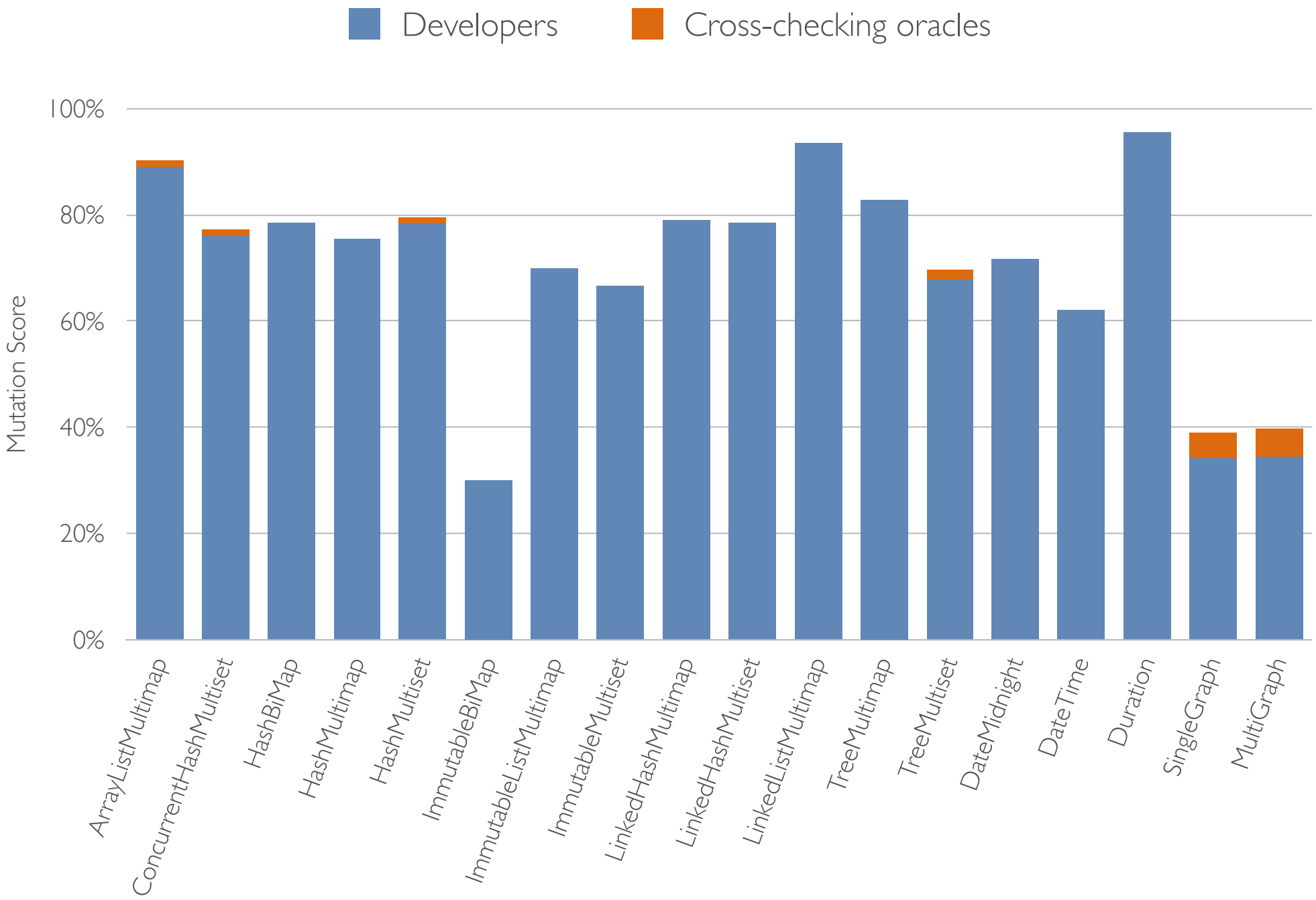
Cross-Checking & Implicit Oracles



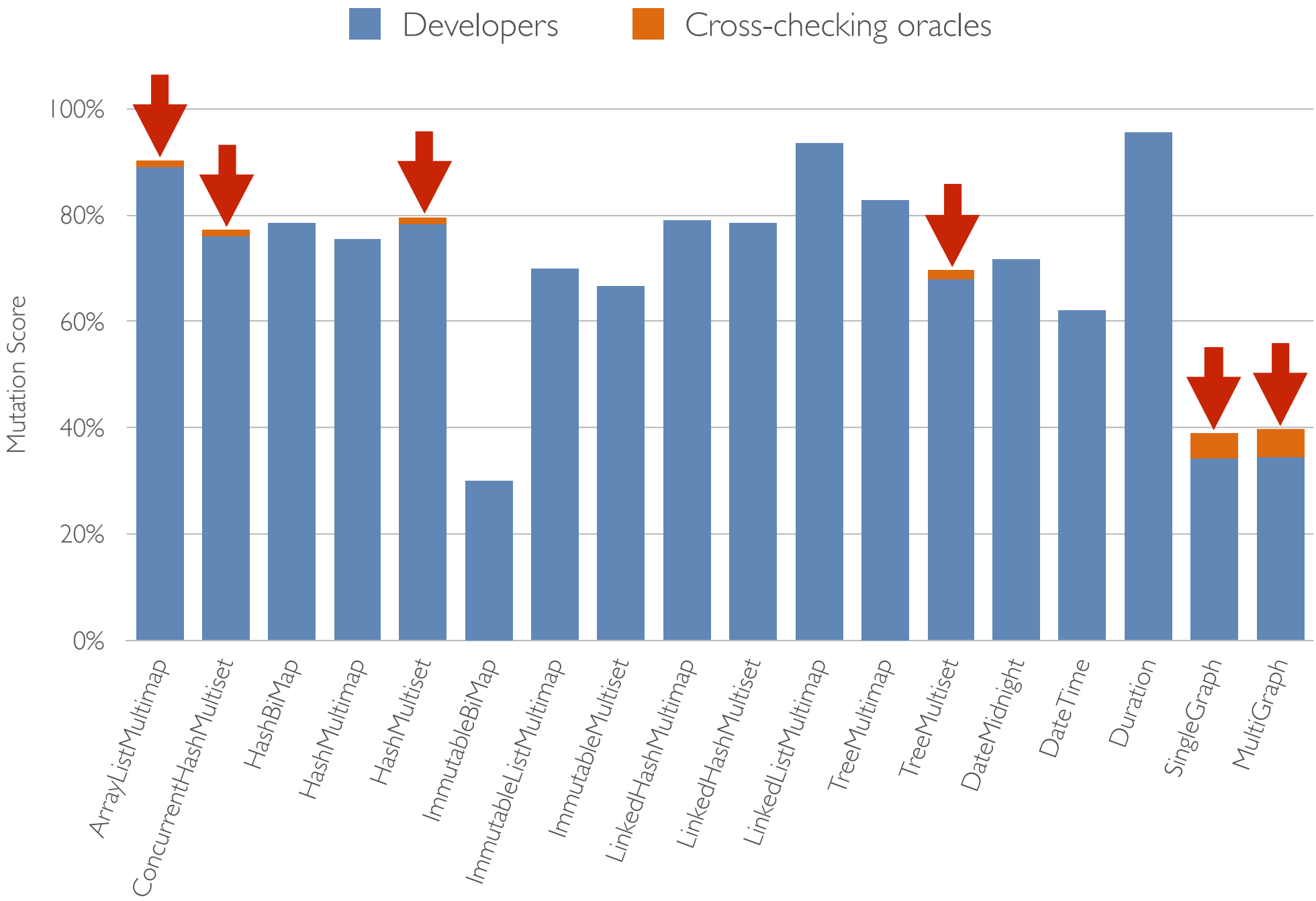
Cross-Checking & Hand-Written Oracles



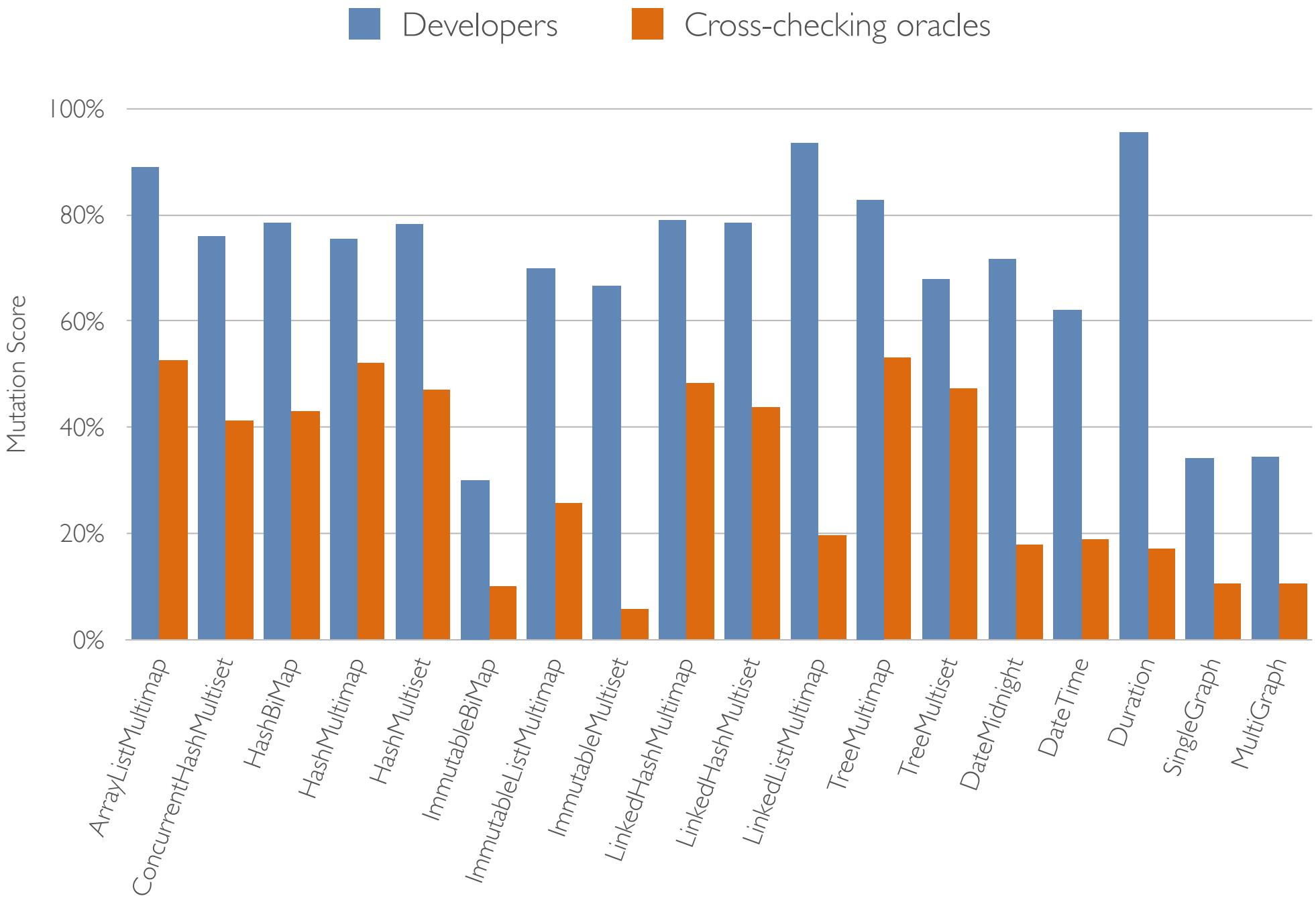
Cross-Checking & Hand-Written Oracles



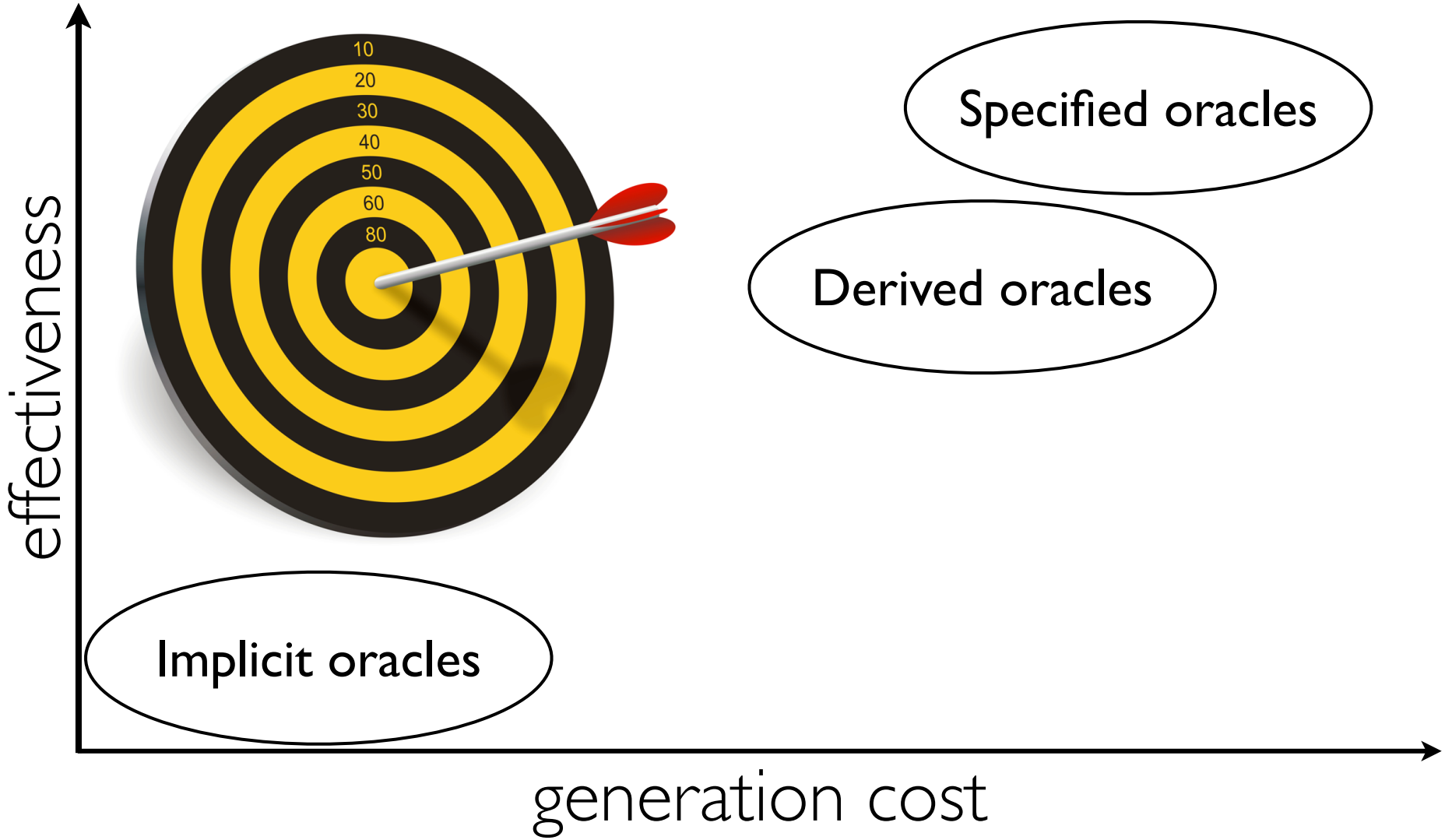
Cross-Checking & Hand-Written Oracles



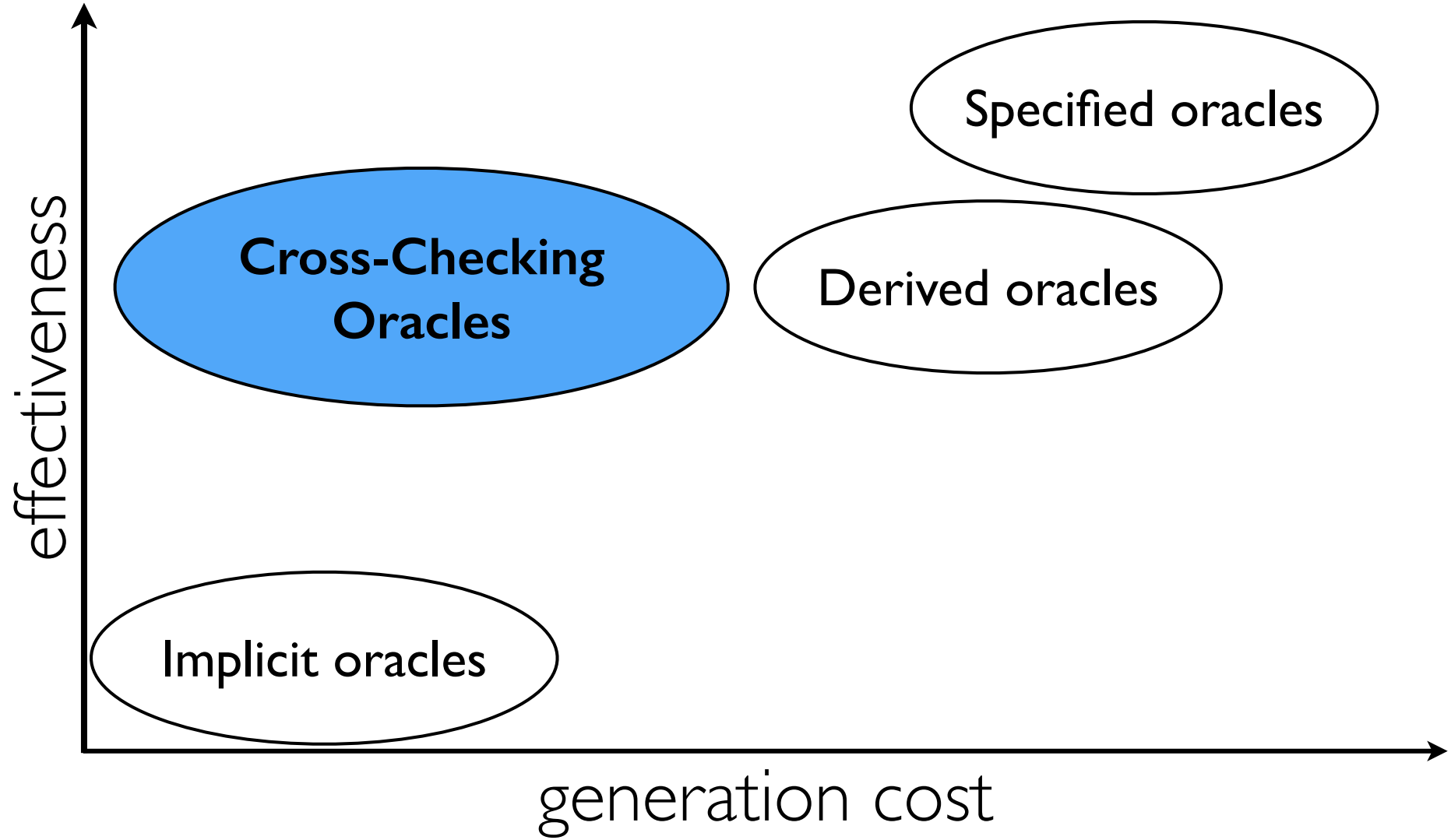
Cross-Checking & Hand-Written Oracles



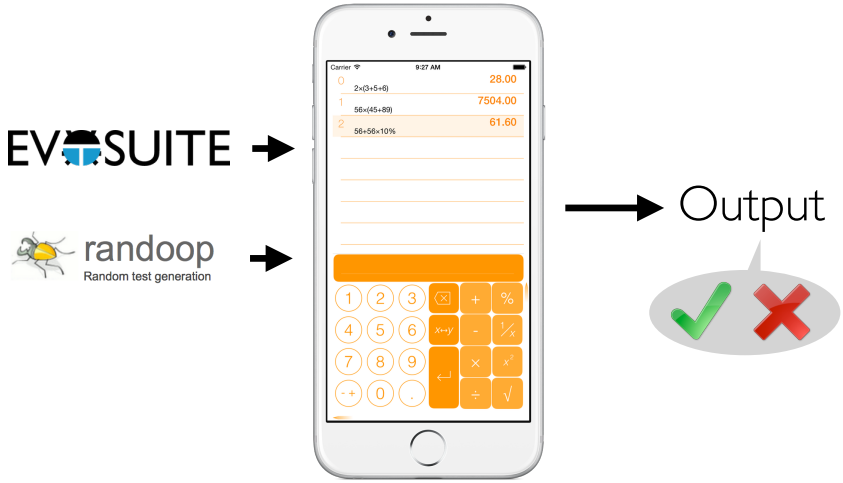
Automated Test Oracles



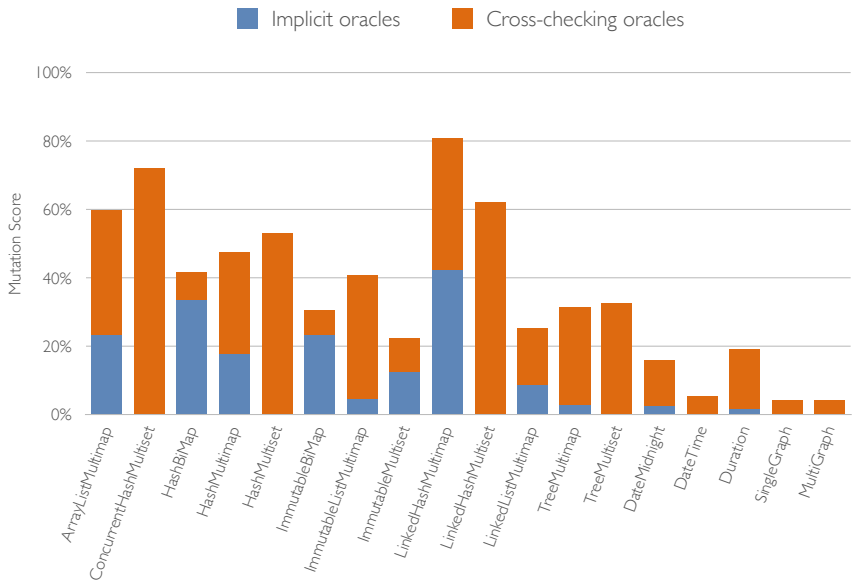
Automated Test Oracles



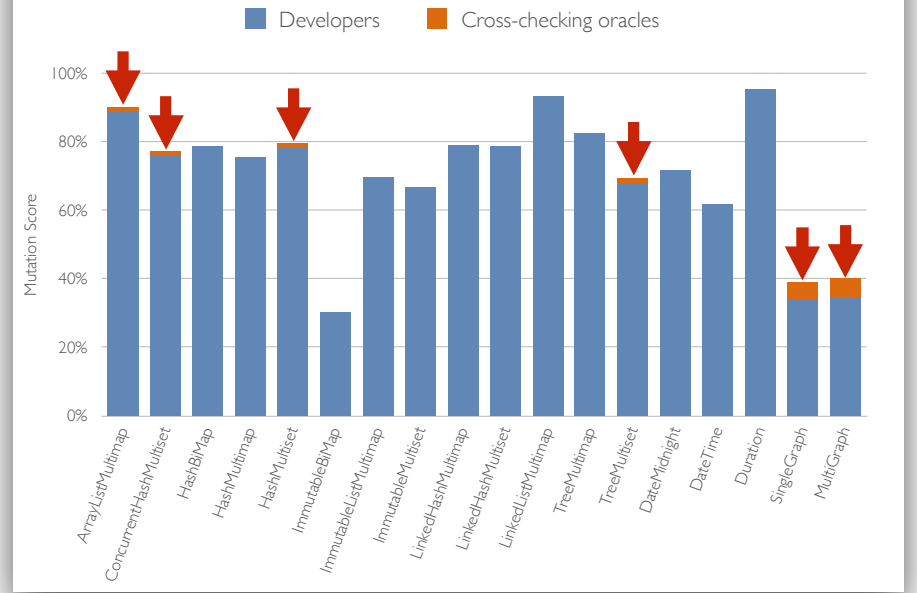
Software Testing



Cross-Checking & Implicit Oracles



Cross-Checking & Hand-Written Oracles



Automated Test Oracles