

How Code Search Drives Software Engineering

Dr. Kathryn (Katie) Stolee

Associate Professor

North Carolina State University

Search

Repositories463

Code6M

Commits7K

Issues48K

Discussions394

Packages3

Marketplace0

6,279,643 code results

Sort: Best match

comanje/comanje.nl

pages/python.md

```
114 ## replace broken words based on lookup table
115 ```python
116 #!/usr/bin/env python3
117
118 import re, csv
119 from collections import defaultdict
120 import os.path
```

```
7 * { name: "Jane Doe",
8 *   collaborators: ["John Doe", "Karen Smith"]
9 * },
10 * { name: "Skittles the Cat",
11 *   collaborators: []
12 * }
13 * ]
14 */
15 function collaborators_map(json: any): Map<string, Set<string>> {
16   const map = new Map<string, Set<string>>();
17   for (const item of json) {
18     const name = item.name;
19     const collaborators = item.collaborators;
20     const set = new Set<string>(collaborators);
21     map.set(name, set);
22   }
23   return map;
24 }
```

Copilot

Google

Google Search

I'm Feeling Lucky

I'd like Python code

Certainly! You can u
list. Here's the Pyth

chromium

An open-source browser to help move the web forward.

Search projects

Project Home

Downloads

Wiki

Issues

Code Search

Search code

Search Code

Search via regular expression, e.g. `^java/.*\.java$`

Search Options		In Search Box
Language	<div>Any language</div>	lang:c++
File Path	<input type="text"/>	file:(code [^or]g)search
Class	<input type="text"/>	class:HashMap
Function	<input type="text"/>	function:toString
Symbol	<input type="text"/>	symbol:std::vector
Case Sensitive	<div>No</div>	case:yes
Exact	<div>No</div>	exact:yes

We studied Google Developers in 2013

- Code search is performed **throughout the development lifecycle**
- Search queries happens ~**12x** per day

In 2022, we also found (yet unpublished observations):

- Searching for examples was **less successful** than searches for other purposes



FIND IT



THINK IT



TEST IT



FIX IT



Finding Code Examples

- Empirical Investigations into Developer Behavior
 - 85% of developers search for code at least weekly [TOSEM 2014]
 - Average of 12 queries per day [FSE 2015]
 - Code searches require more effort than information search [MSR 2018]
- Innovations:
 - Behavior-based code search via static analysis [TOSEM 2014]
 - Behavior-based code search via dynamic analysis [ICSE 2020] [FSE 2021]



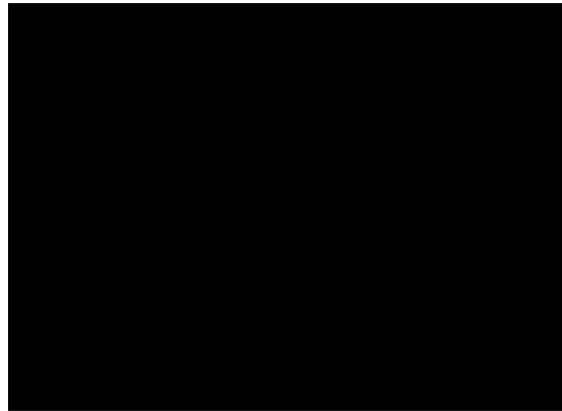
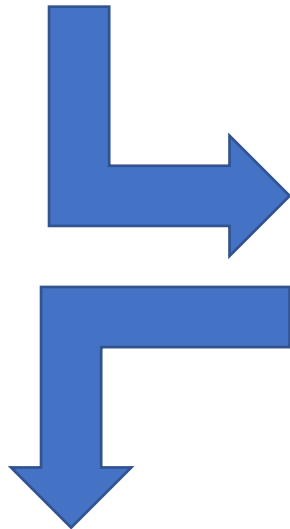


A different kind of search

```
▶ blackbox([6, 2, 3, 4])  
[→ 2
```

← Example input

← Example output



```
def min_py2(a):  
    return sorted(a)[0]
```



Specification

Code

Matching

Indexing

Result(s)



Code Search via Symbolic Execution

K. T. Stolee, S. Elbaum, M. B. Dwyer: Code search with input/output queries: Generalizing, ranking, and assessment. JSS 2016.

K. T. Stolee, S. Elbaum, D. Dobos: Solving the Search for Source Code. TOSEM 2014.

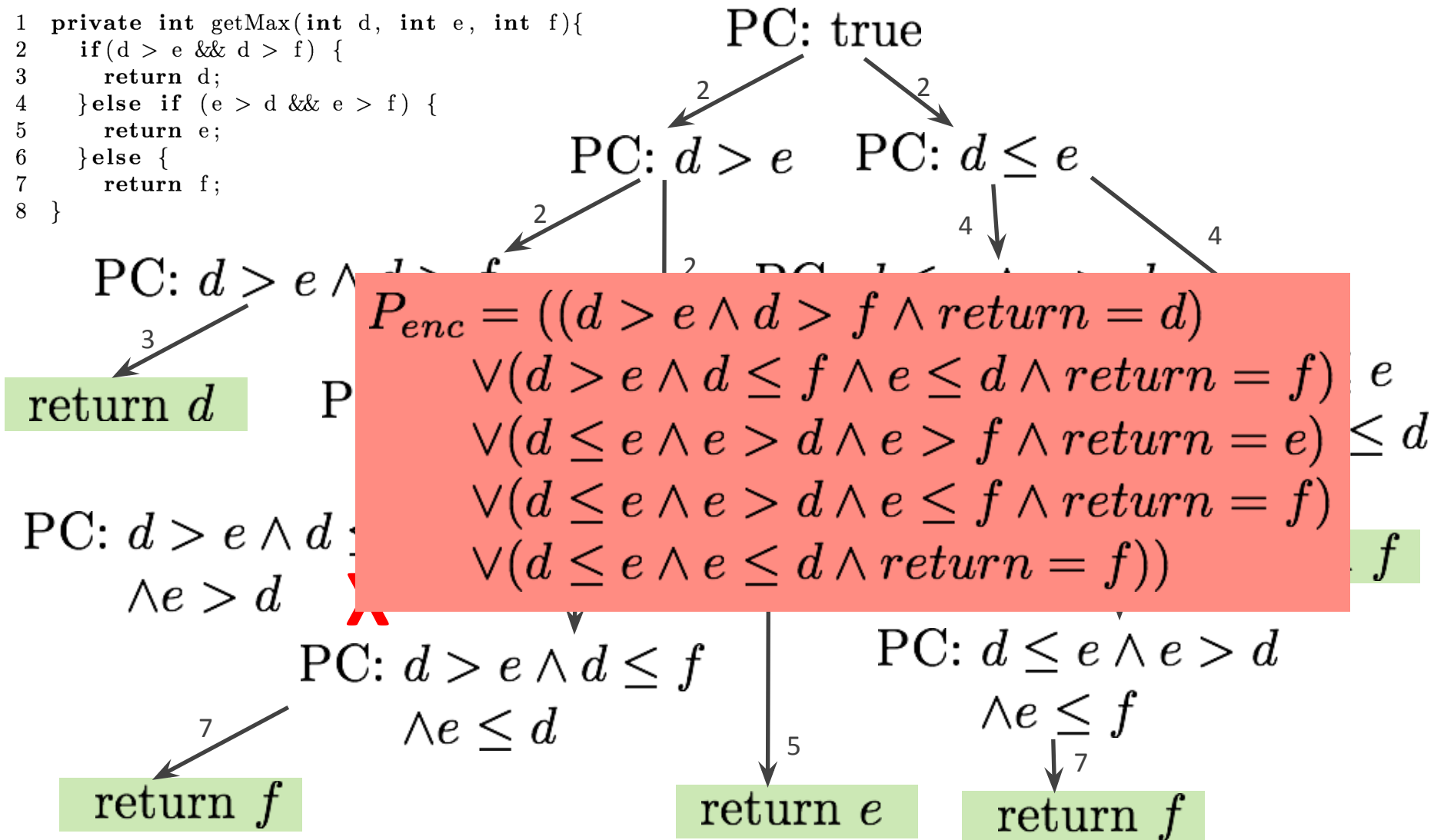


Symbolic Execution

```

1 private int getMax(int d, int e, int f){
2     if(d > e && d > f) {
3         return d;
4     }else if (e > d && e > f) {
5         return e;
6     }else {
7         return f;
8     }

```





SMT Solvers

Satisfiability **M**odulo **T**heory solvers determine if a logical formula is satisfiable

Facts

$a \geq 0$

$b = 2$

$c = 2$

$c = a * b$

Assertions

`(assert (\geq a 0))`

`(assert (= b 2))`

`(assert (= c 2))`

`(assert (= (* a b) c))`

Result: **sat** $a \mapsto 1$



SMT Solvers

Satisfiability **M**odulo **T**heory solvers determine if a logical formula is satisfiable

Facts

Assertions

`a >= 0`

`(assert (>= a 0))`

`b = ?`

`(assert (= b ?))`

`c = 2`

`(assert (= c 2))`

`c = a * b`

`(assert (= (* a b) c))`

Result: **sat** $a \mapsto \mathbb{Z} \wedge b \mapsto \mathbb{Z}$



SMT Solvers

Satisfiability **M**odulo **T**heory solvers determine if a logical formula is satisfiable

Facts	Assertions
<code>a = 0</code>	<code>(assert (= a 0))</code>
<code>b = ?</code>	<code>(assert (= b ?))</code>
<code>c = 2</code>	<code>(assert (= c 2))</code>
<code>c = a * b</code>	<code>(assert (= (* a b) c))</code>

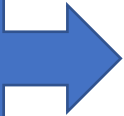
Result: **unsat**



SMT Matching

```
1 private int getMax(int d, int e, int f){  
2   if(d > e && d > f) {  
3     return d;  
4   }else if (e > d && e > f) {  
5     return e;  
6   }else {  
7     return f;  
8   }
```

$$P_{enc} = ((d > e \wedge d > f \wedge return = d) \\ \vee (d > e \wedge d \leq f \wedge e \leq d \wedge return = f) \\ \vee (d \leq e \wedge e > d \wedge e > f \wedge return = e) \\ \vee (d \leq e \wedge e > d \wedge e \leq f \wedge return = f) \\ \vee (d \leq e \wedge e \leq d \wedge return = f))$$

Query	Input	Output	Result
	3, 4, 3	4	sa
	3, 0, 0	3	fa
			†

This is a result!



SMT Matching

Encoding

```
private int getsum(int a, int b, int c){  
    return a + b + c;  
}
```

Potential Search Result

```
(declare-fun a () Int)  
(declare-fun b () Int)  
(declare-fun c () Int)  
(declare-fun return () Int)  
(assert (= return (+ (+ a b) c)))  
(assert (and (= a 3) (= b 4)  
              (= c 0)))  
(assert (= return 7))
```

Query

Input

Output

Result

3, 4, 3

7

unsa

3, 4, 0

7

sa

†

Not a Result!



Why not just execute the code?



SMT Matching

```
private int getsum(int a, int b, int c){  
    return a + b + c;  
}
```

Potential Search Result

Encoding

Relax Encoding

```
(declare-fun a () Int)  
(declare-fun b () Int)  
(declare-fun c () Int)  
(declare-fun return () Int)  
(assert (= return (+ (+ a b) ?)))  
(assert (and (= a 3) (= b 4)  
              (= c 0)))  
(assert (= return 7))
```

Query

Input

Output

Result

3, 4, 3

7

unsat

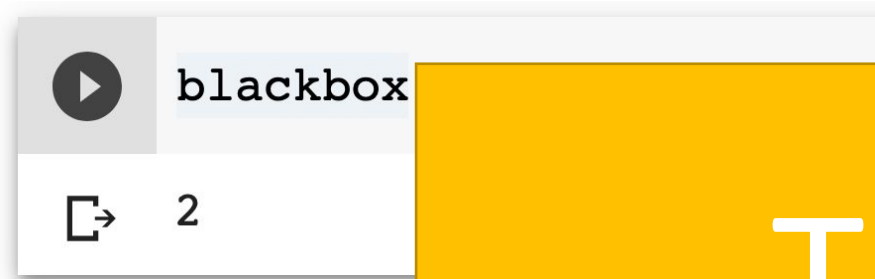
3, 4, 0

7

sat

†

This is a result (with modification)!



This gets expensive!

```
def min_py1(arr):  
    least = arr[0]  
    for a in arr[1:]:  
        if a < least:  
            least = a  
    return least
```

```
def min_py2(a):  
    return sorted(a)[0]
```

```
    a) {  
length; i++) {  
    i] < m)
```



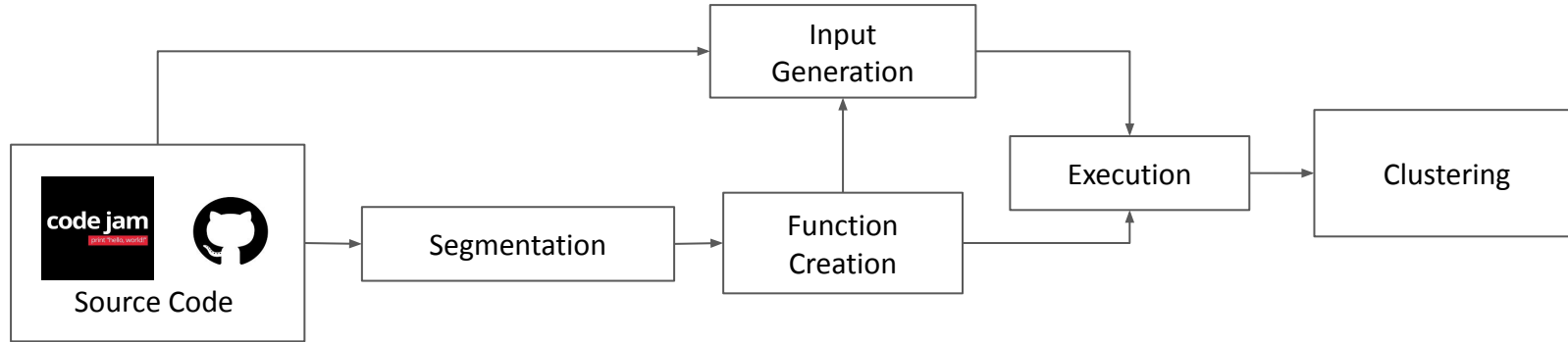
Cross-Language Code Search

G. Mathew, K. T. Stolee: Cross-language code search using static and dynamic analyses. ESEC/SIGSOFT FSE 2021.

G. Mathew, C. Parnin, K. T. Stolee: SLACC: simion-based language agnostic code clones. ICSE 2020.

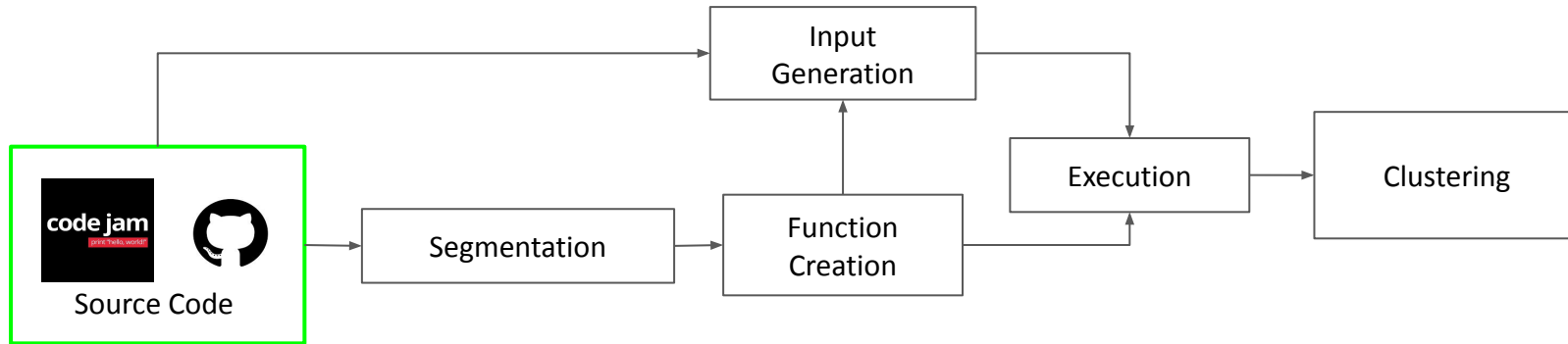


Cross-Language Behavioral Clones





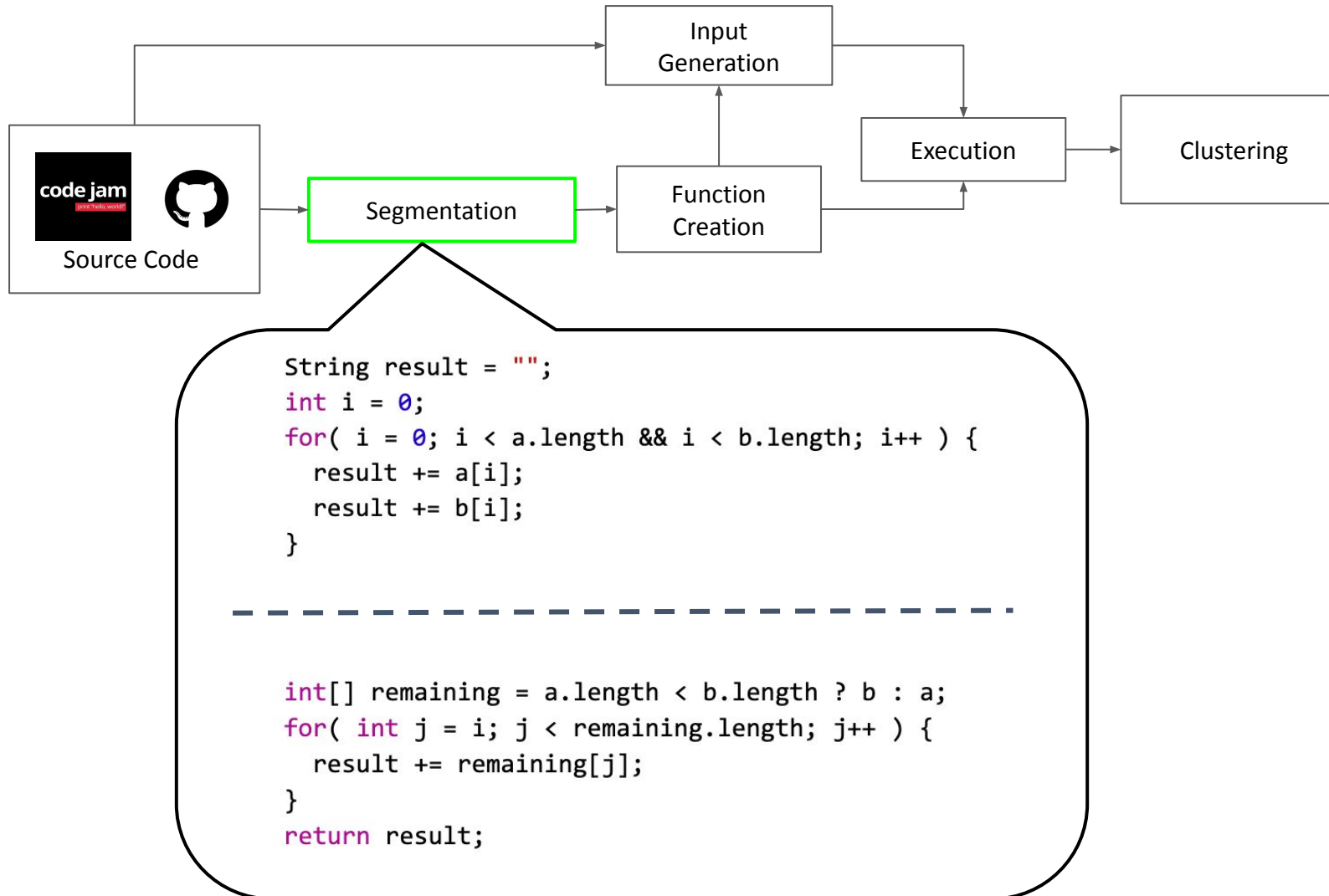
Cross-Language Behavioral Clones



```
public String interleave(int[] a, int[] b) {  
    String result = "";  
    int i = 0;  
    for( i = 0; i < a.length && i < b.length; i++ ) {  
        result += a[i];  
        result += b[i];  
    }  
    int[] remaining = a.length < b.length ? b : a;  
    for( int j = i; j < remaining.length; j++ ) {  
        result += remaining[j];  
    }  
    return result;  
}
```

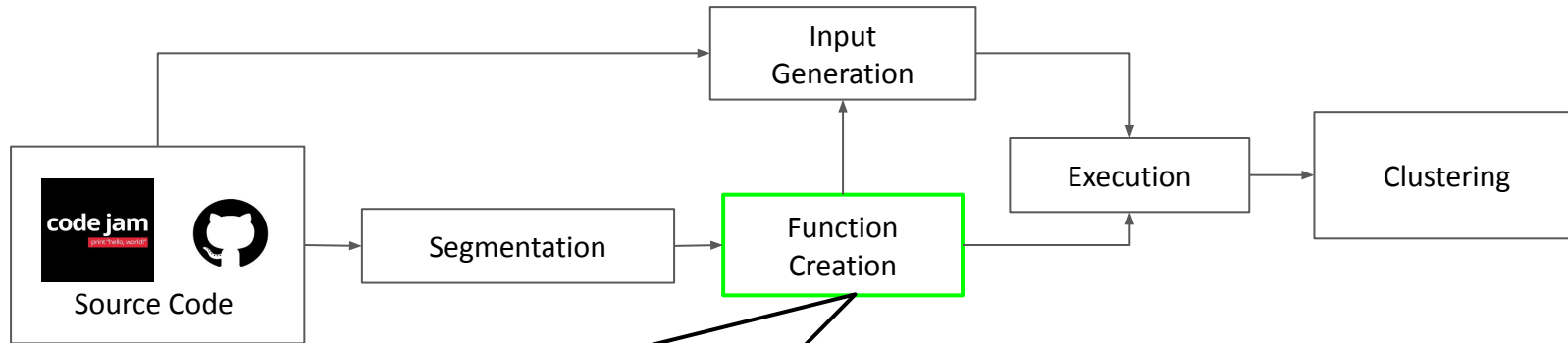


Cross-Language Behavioral Clones





Cross-Language Behavioral Clones

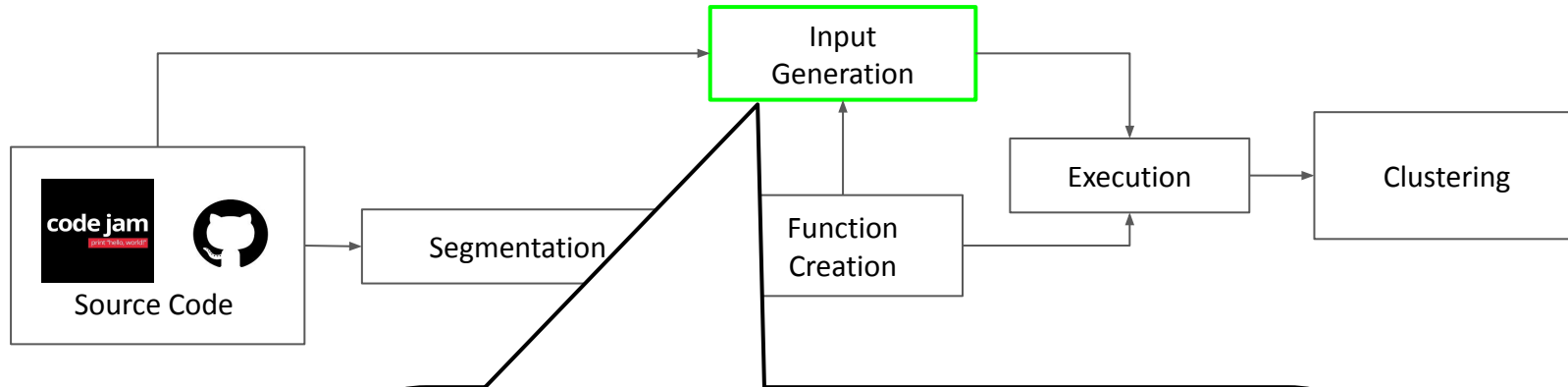


```
public String func_b15f(int[] a, int[] b) {  
    String result = "";  
    int i = 0;  
    for( i = 0; i < a.length && i < b.length; i++ ) {  
        result += a[i];  
        result += b[i];  
    }  
    return result;  
}
```

```
public String func_ea72(int[] a, int[] b, int i, String result) {  
    int[] remaining = a.length < b.length ? b : a;  
    for( int j = i; j < remaining.length; j++ ) {  
        result += remaining[j];  
    }  
    return result;  
}
```



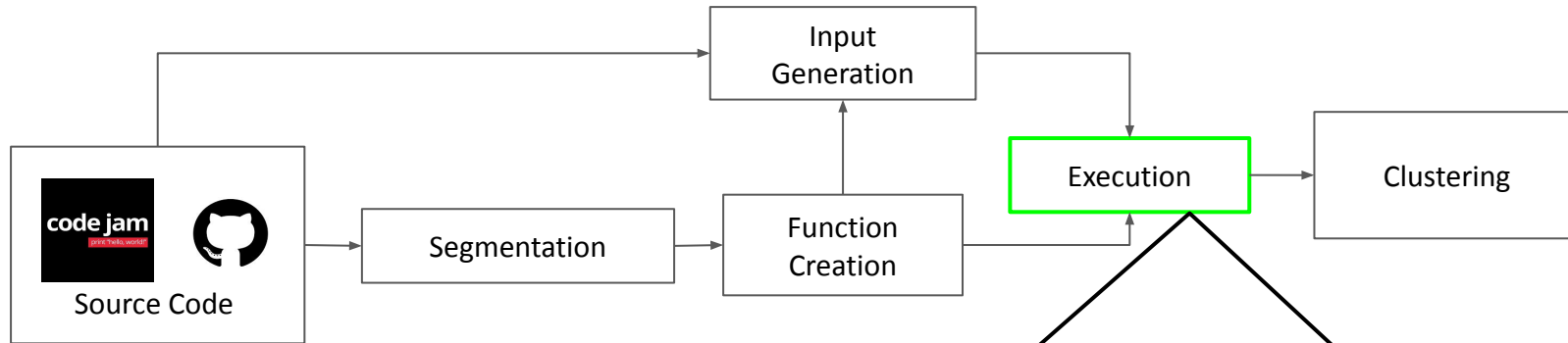
Cross-Language Behavioral Clones



```
int[], int[]
• [1,2,3], [45, 16]
• [ ], [3,2,1]
⋮
256 times
⋮
• [4, 5, 6, 7, 8, 99], [ ]
-----
int[], int[], int, String
• [1,2,3], [45, 16], 0, "Hello
World"
⋮
256 times
⋮
```



Cross-Language Behavioral Clones



```
func_b15f(int[] a, int[] b)
```

- `func_b15f([1,2,3], [45, 16])` \Rightarrow 1452163

- `func_b15f([], [3,2,1])` \Rightarrow 321

⋮

256 times

⋮

- `func_b15f([4, 5, 6, 7, 8, 99], [])` \Rightarrow 4567899

- `func_b15f([3], [2])` \Rightarrow 32

```
func_ea72(int[] a, int[] b, int i, String result)
```

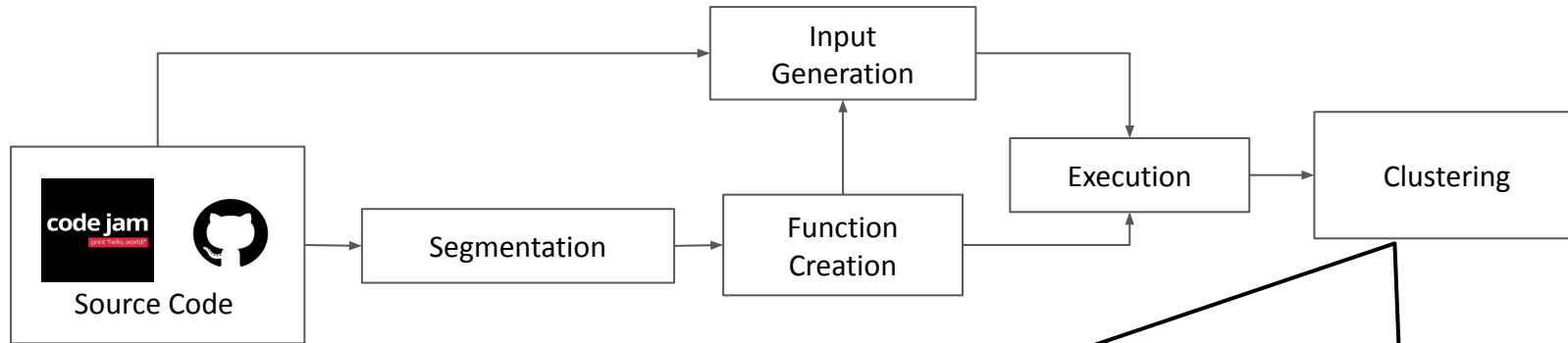
- `func_ea72([1,2,3], [45, 16], 0, "Hello World")` \Rightarrow
Hello World123

⋮

256 times



Cross-Language Behavioral Clones



```
public String func_b15f(int[] a, int[] b) {  
    String result = "";  
    int i = 0;  
    for( i = 0; i < a.length && i < b.length; i++ ) {  
        result += a[i];  
        result += b[i];  
    }  
    return result;  
}
```

```
def func_9f34(l1, l2):  
    result = ""  
    for (e1, e2) in zip(l1, l2):  
        result += str(e1)  
        result += str(e2)  
    return result
```

```
def func_15e8(l1, l2):  
    from itertools import chain  
    return "".join([str(x)  
        for x in chain.from_iterable(zip(l1, l2))])
```

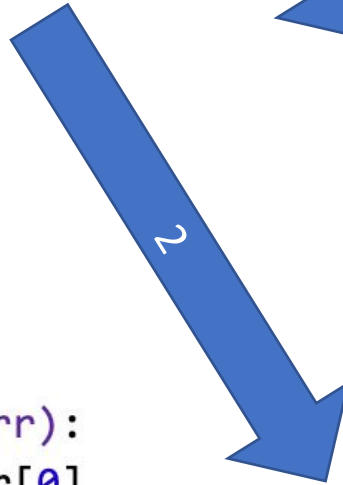
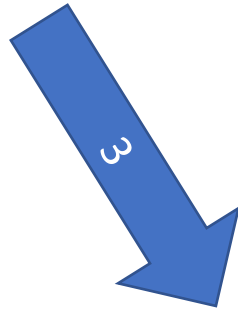
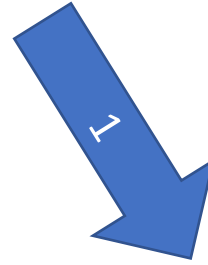


What does this mean for **search**?



```
blackbox([6, 2, 3, 4])
```

```
2
```



```
def min_py1(arr):  
    least = arr[0]  
    for a in arr[1:]:  
        if a < least:  
            least = a  
    return least
```

```
public Integer minJ(int[] a) {  
    Integer m = null;  
    for (int i=0; i < a.length; i++) {  
        if (m == null || a[i] < m)  
            m = a[i];  
    }  
    return m;  
}
```

```
def min_py2(a):  
    return sorted(a)[0]
```



```
blackbox([6, 2, 3, 4])
```



2



```
def min_py1(arr):  
    least = arr[0]  
    for a in arr[1:]:  
        if a < least:  
            least = a  
    return least
```

```
public Integer minJ(int[] a) {  
    Integer m = null;  
    for (int i=0; i < a.length; i++) {  
        if (m == null || a[i] < m)  
            m = a[i];  
    }  
    return m;  
}
```

```
def min_py2(a):  
    return sorted(a)[0]
```

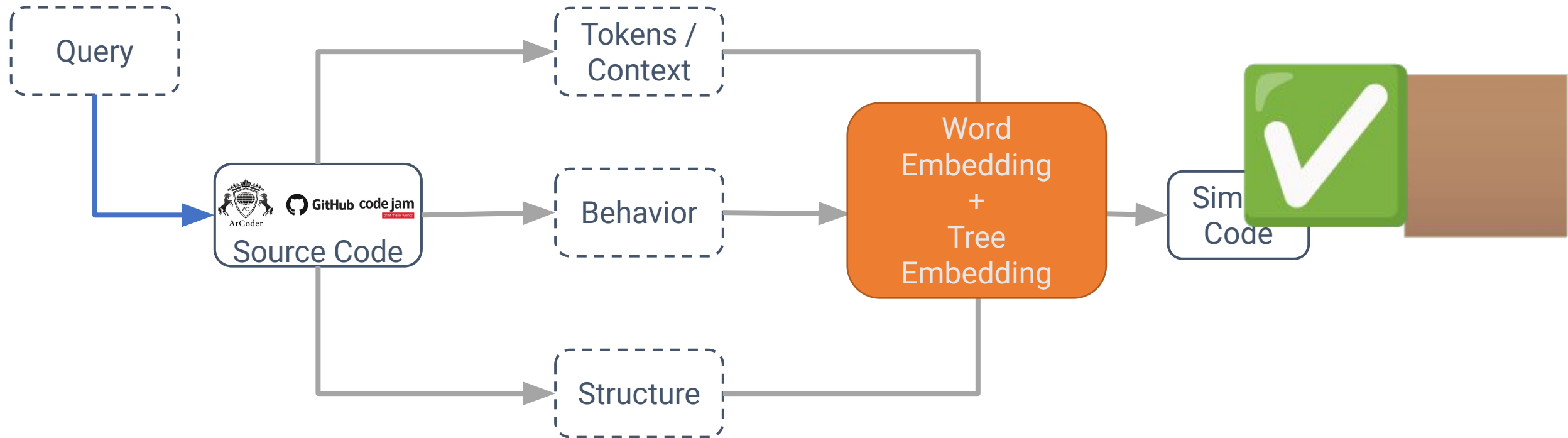




What about **code-to-code search**?

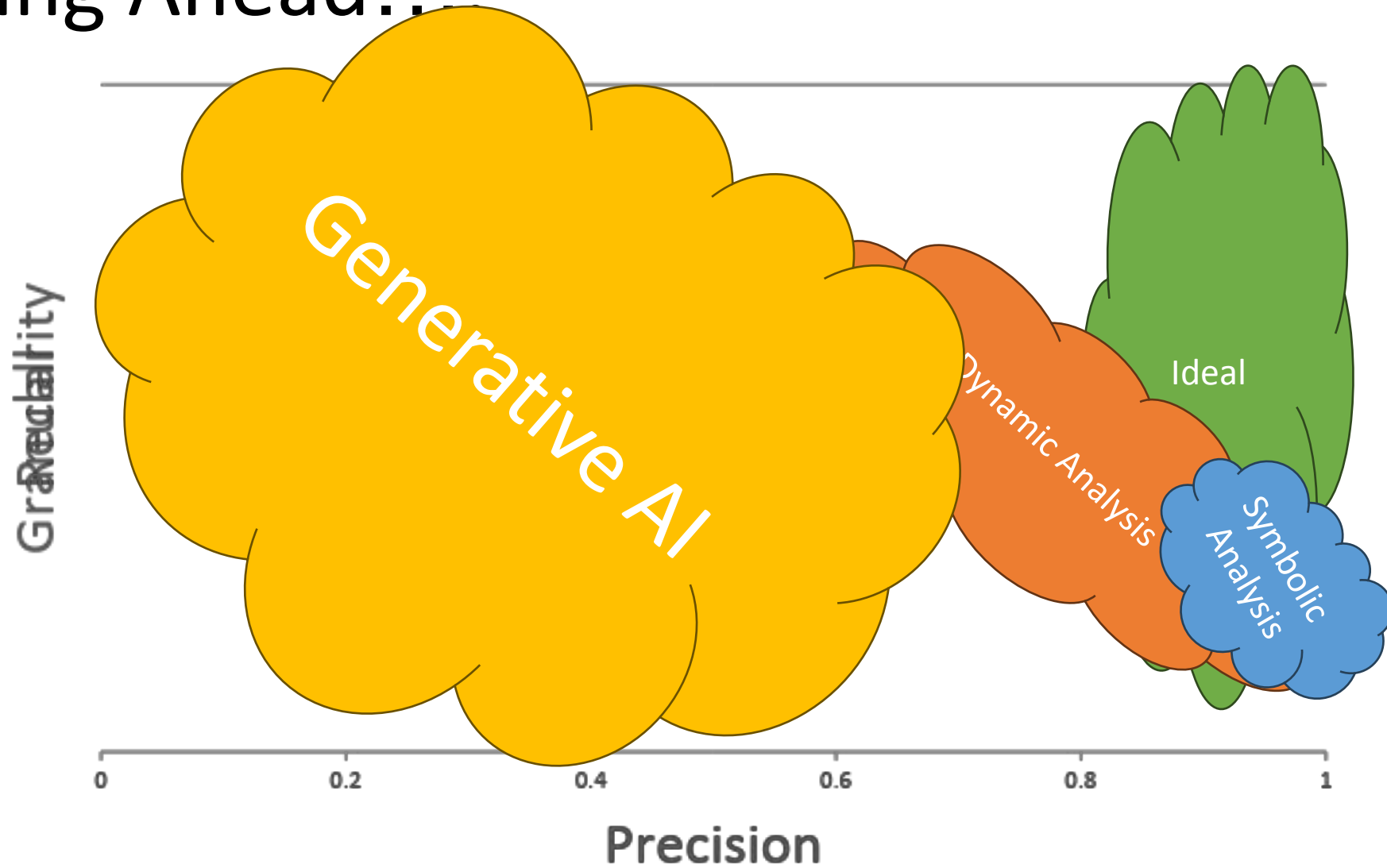


Code-to-code Search





Looking Ahead...





Looking Ahead...



I'd like Python code that, given an input of [6,2,3,4], produces the output 2



Certainly!
list. Here

python

```
def find_smallest_number(numbers):
```

```
    smallest = numbers[0]
```

```
    for num in numbers:
```

```
        if num < smallest:
```

```
            smallest = num
```

```
    print(smallest)
```



What is another way to do this?



Another way to find the smallest number in a list is by using a simple loop. Here's an alternative implementation using a loop:



python

Copy code

```
def find_smallest_number(numbers):
```

```
    smallest = numbers[0] # Initialize smallest as the first element of the
```

```
    for num in numbers:
```

```
        if num < smallest:
```

```
            smallest = num
```




FIND IT



THINK IT



TEST IT



FIX IT



Understanding Code

- Empirical Investigations into Code Comprehension
 - Regular expression representation significantly impacts understandability [ASE 2017]
 - Comparing similar code algorithms is difficult and error-prone for developers [VL/HCC 2022]
 - Code review of refactorings is *very hard* for students [under review]





Understanding Code

- Empirical Investigations into Code Comprehension
 - Regular expression representation significantly impacts understandability [ASE 2017]
 - **Comparing similar code algorithms is difficult and error-prone for developers [VL/HCC 2022]**
 - Code review of refactorings is *very hard* for students [under review]

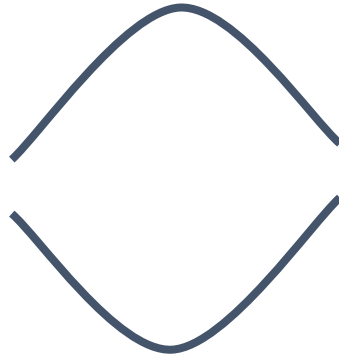




Comparative Comprehension

The cognitive activity of understanding how algorithms behave relative to each other

```
def sumup(numbers):  
    accumulator = 0  
    for value in numbers:  
        accumulator += value  
    return accumulator
```



```
def sumup(x):  
    s = 0  
    i = 0  
    while i < len(x):  
        s += i  
        i += 1  
    return s
```

Controlled Experiment



```
1 public static boolean isAnagram(String str1,
2                               String str2) {
3     if (str1.length() != str2.length())
4         return false;
5
6     int[] count1 = new int[256];
7     int[] count2 = new int[256];
8
9     for (int i = 0; i < str1.length(); i++) {
10         ++count1[str1.charAt(i)];
11         ++count2[str2.charAt(i)];
12     }
13
14     for (int i = 0; i < 256; i++)
15         if (count1[i] != count2[i])
16             return false;
17
18     return true;
19 }
```

```
1 def isAnagram(s, t):
2     hash1 = [0]*256
3     hash2 = [0]*256
4
5     for char in s:
6         hash1[ord(char)] += 1
7     for char in t:
8         hash2[ord(char)] += 1
9
10    return hash1 == hash2
11
12
13
14
15
16
17
18
19
```

- 4 independent dimensions of variation
 - Behavior (same or not)
 - Language (same or not)
 - Structures (similar AST or not)
 - Meaningful names (original or obfuscated)

Controlled Experiment



Interviews

n=16



Undergraduate students



Graduate students



Professionals

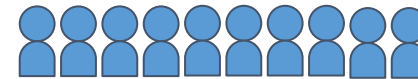


Survey

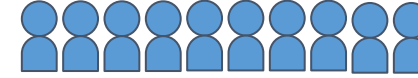
n=95



Unknown



Graduate students



Professionals



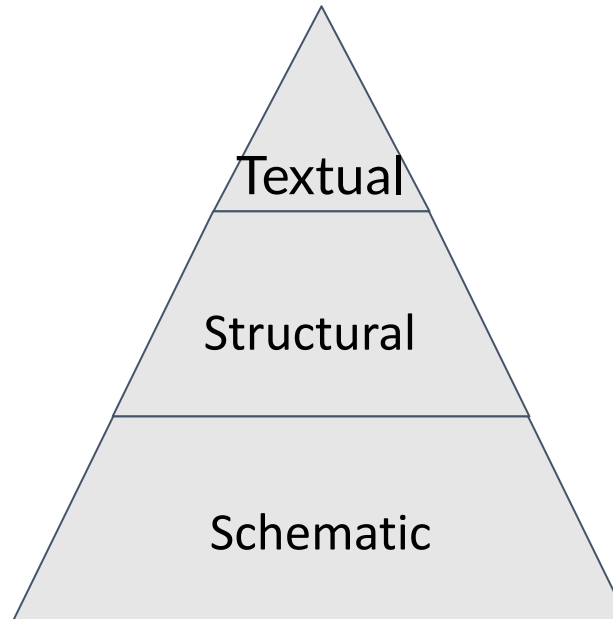


Comparison Accuracy

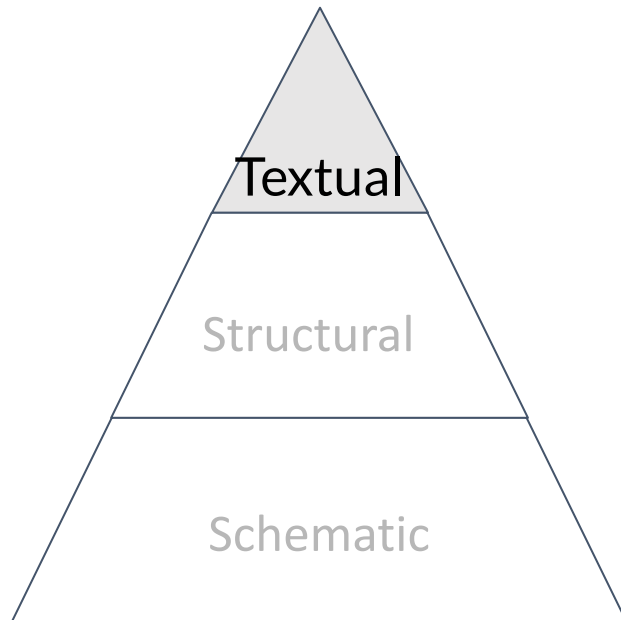
Overall correctness: 292 of 439 – 66.5%

Correctness (%) for...			
Clone Truth Language Structure Names (Meaningful Obf.)	Similarity	Dissimilarity	
	85.3	46.7	***
	70.9	62.7	*
	75.0	59.9	*
	66.8	66.2	

Comparison Strategies



Comparison Strategies

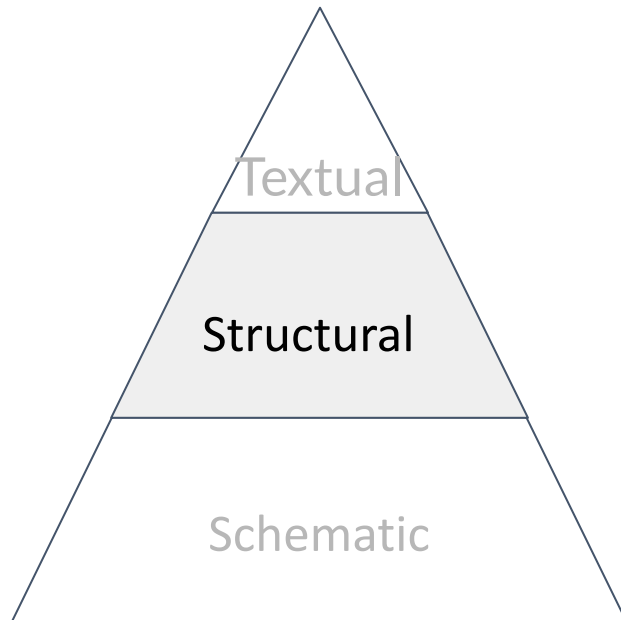


```
1 def camel_case(string):
2     a = list(string)
3     for i in range(0, len(a)):
4         if i==0 or a[i-1]=='_':
5             a[i] = a[i].upper()
6     return ''.join(a).replace('_', '')
```

```
1 def camel_case(string):
2     a = list(string)
3     for i in range(len(a)):
4         if i==0 or a[i-1]=='_':
5             a[i] = a[i].upper()
6     return ''.join([c for c in a if c != "_"])
```

“I didn't even need to [understand the logic] because they were so similar.” - P4

Comparison Strategies

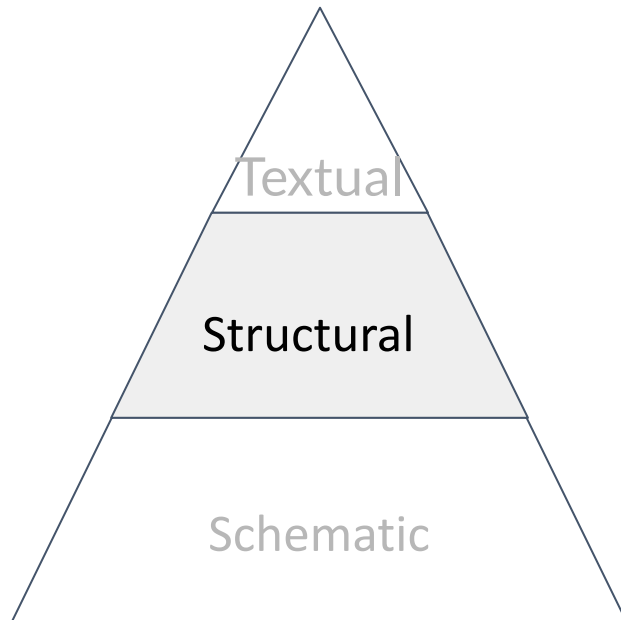


```
1 public static String removeDuplicates(String orig)
2 {
3     if (orig.length() == 0) return "";
4     String result = "" + orig.charAt(0);
5     for (int i = 1; i < orig.length(); i++)
6         if (orig.charAt(i-1) != orig.charAt(i))
7             result+=orig.charAt(i);
8     return result;
9 }
```

```
1 def removeadjacent(orig):
2     if len(orig) == 0: return orig
3     res = ""
4     for i in range(1, len(orig)):
5         if orig[i-1] != orig[i]:
6             res += orig[i]
7     return orig[0] + res
8
9
```

U4 on cross-language deduplicators

Comparison Strategies

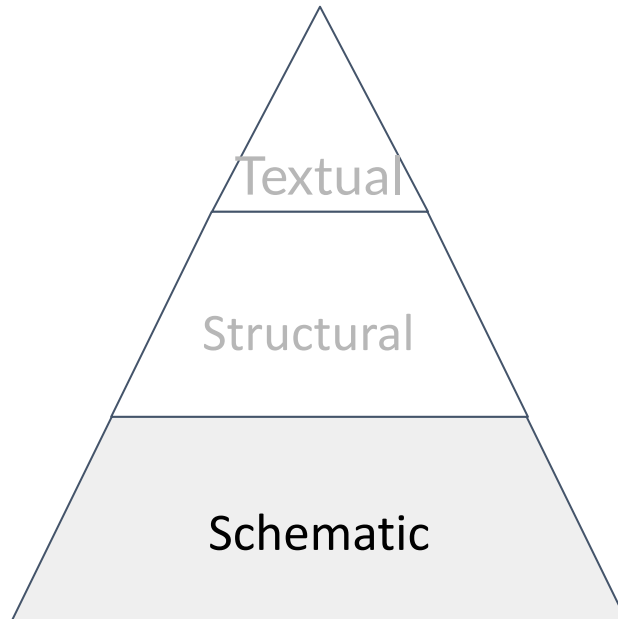


```
1 public static String removeDuplicates(String orig)
2 {
3     if (orig.length() == 0) return "";
4     String result = "" + orig.charAt(0);
5     for (int i = 1; i < orig.length(); i++)
6         if (orig.charAt(i-1) != orig.charAt(i))
7             result += orig.charAt(i);
8     return result;
9 }

1 def removeadjacent(orig):
2     if len(orig) == 0: return orig
3     res = ""
4     for i in range(1, len(orig)):
5         if orig[i-1] != orig[i]:
6             res += orig[i]
7     return orig[0] + res
8
9
```

U4 on cross-language deduplicators

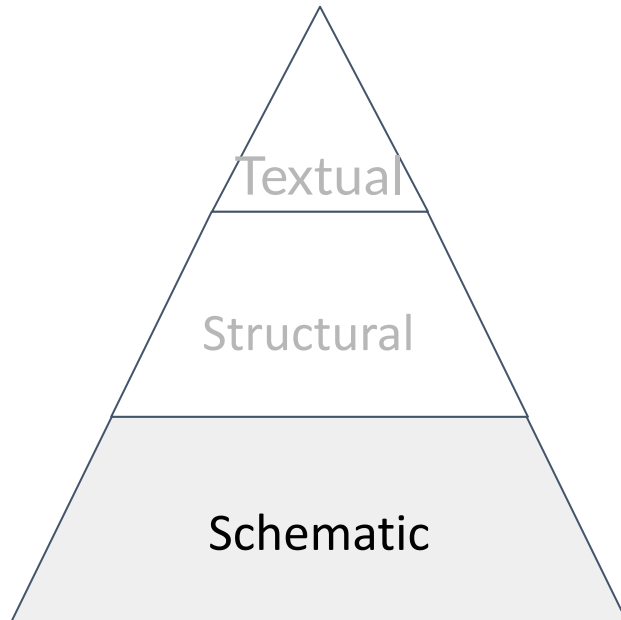
Comparison Strategies



```
1 def to_camel_case(text):
2     cap = True
3     newText = ''
4     for t in text:
5         if t == '_':
6             cap = True
7             continue
8         else:
9             if cap == True:
10                 t = t.upper()
11                 newText = newText + t
12                 cap = False
13     return newText
```

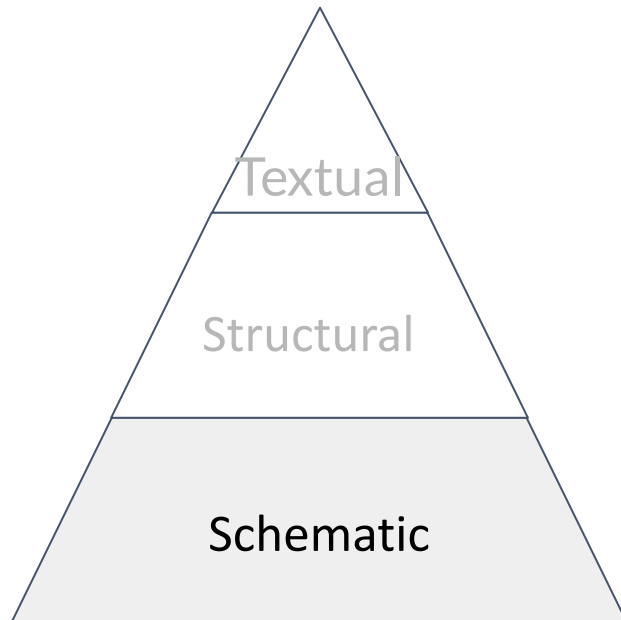
```
1 def UnderscoreToCamelCase(under_score):
2     segments = under_score.split('_')
3     return segments[0] + ''.join([s[0].upper() + s[1:] \
4         for s in segments[1:] if len(s) > 0])
5
6
7
8
9
10
11
12
13
```

Comparison Strategies



```
1 def UnderscoreToCamelCase(under_score):  
2     segments = under_score.split('_')  
3     return segments[0] + ''.join([s[0].upper() + s[1:] \br/>4         for s in segments[1:] if len(s) > 0])  
5  
6  
7  
8  
9  
10  
11  
12  
13
```

Comparison Strategies



```
1 def to_camel_case(text):
2     cap = True
3     newText = ''
4     for t in text:
5         if t == '_':
6             cap = True
7             continue
8         else:
9             if cap == True:
10                 t = t.upper()
11                 newText = newText + t
12                 cap = False
13     return newText
```



Looking Ahead...

Conversation 0 Commits 1 Checks 0 Files changed 1

 jamiddl2 commented on Nov 29, 2021 • edited by ktstolee ▾ Member ...

Adjusting some for-loops within getEntryByDateRange.

Instructions:

(fill out the Google form)

✓ 8 

iTrust2/src/main/java/edu/ncsu/csc/iTrust2/controllers/api/APILogEntryController.

java 

☐ Viewed ...

↑

@@ -110,8 +110,7 @@ public class APILogEntryController extends APICo

110	110	if (user == null user.getRoles() == null user.getRo
111	111	visible = new ArrayList<LogEntry>();
112	112	
113	-	for (int i = 0; i < entries.size(); i++) {
114	-	final LogEntry le = entries.get(i);
	113	+ for (final LogEntry le : entries) {



Looking Ahead...



I'd like Python code that, given an input of [6,2,3,4], produces the output 2



Certainly
list. Here

python

```
def f
```

```
S
```

```
r
```

input

output

```
print
```



What is another way to do this?



Another
alternativ

python

```
def f
```



What is an input that makes the code examples behave differently?



█

```
smallest = numbers[0] # Initialize smallest as the first element of the
```

```
for num in numbers:
    if num < smallest:
        smallest = num
```




FIND IT



THINK IT



TEST IT



FIX IT



Testing Code

- Empirical Investigations into Code Repositories
 - Only 17% of regular expressions are tested at all [FSE 2018]
 - Students believe code coverage is the most important outcome for test suites [ITiCSE 2021]
- Innovations:
 - A static checklist for testing is as effective as coverage tools for second-year students [ITiCSE 2022]



SHF: Small: Supporting Regular Expression Testing, Search, Repair, Comprehension, and Maintenance \$500k [completed]

IUSE: EHR: Improving Software Testing Education through Lightweight Explicit Testing Strategies and Feedback \$150k (lead PI, \$300k total) [active]



Testing Code

- Empirical Investigations into Code Repositories
 - Only 17% of regular expressions are tested at all [FSE 2018]
 - **Students believe code coverage is the most important outcome for test suites [ITiCSE 2021]**
- Innovations:
 - **A static checklist for testing is as effective as coverage tools for second-year students [ITiCSE 2022]**



SHF: Small: Supporting Regular Expression Testing, Search, Repair, Comprehension, and Maintenance \$500k [completed]

IUSE: EHR: Improving Software Testing Education through Lightweight Explicit Testing Strategies and Feedback \$150k (lead PI, \$300k total) [active]



Test Case Checklist

Each test case *should*:

- ☐ be executable (i.e., it has an `@Test` annotation and can be run via “Run as JUnit Test”)
- ☐ have at least one assert statement or assert an exception is thrown. Example assert statements include: `assertTrue`, `assertEquals` ([click for tutorials](#)). For asserting an exception is thrown, there are different approaches: `try{...; fail();} catch {assertThat...;}`, `@Test(expected = exception.class)` in JUnit 4, or `assertThrows` in JUnit 5 ([click for tutorials](#)).
- ☐ evaluate/test only one method

Each test case *could*:

- ☐ be descriptively named and commented
- ☐ If there is redundant setup code in multiple test cases, extract it into a common method (e.g., using `@Before`)
- ☐ If there are too many assert statements in a single test case (e.g., more than 5), you might split it up so each test evaluate behavior.

Test Suite Checklist

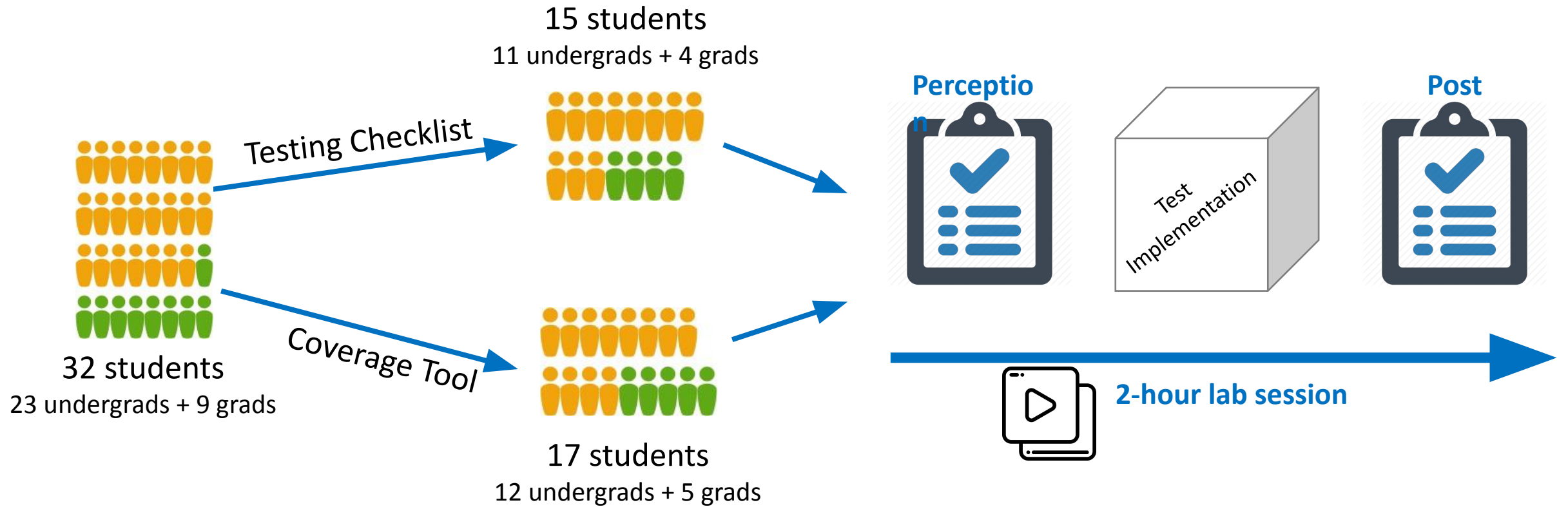
The test suite *should*:

- ☐ have at least one test for each requirement
- ☐ appropriately use the setup and teardown code (e.g., `@Before`, which runs before each `@Test`)
- ☐ contain a fault-revealing test for each bug in the code (i.e., a test that fails)
- ☐ For each requirement, contain test cases for:
 - ☐ Valid inputs
 - ☐ Boundary cases
 - ☐ Invalid inputs
 - ☐ Expected exceptions

To improve the test suite, you *could*:

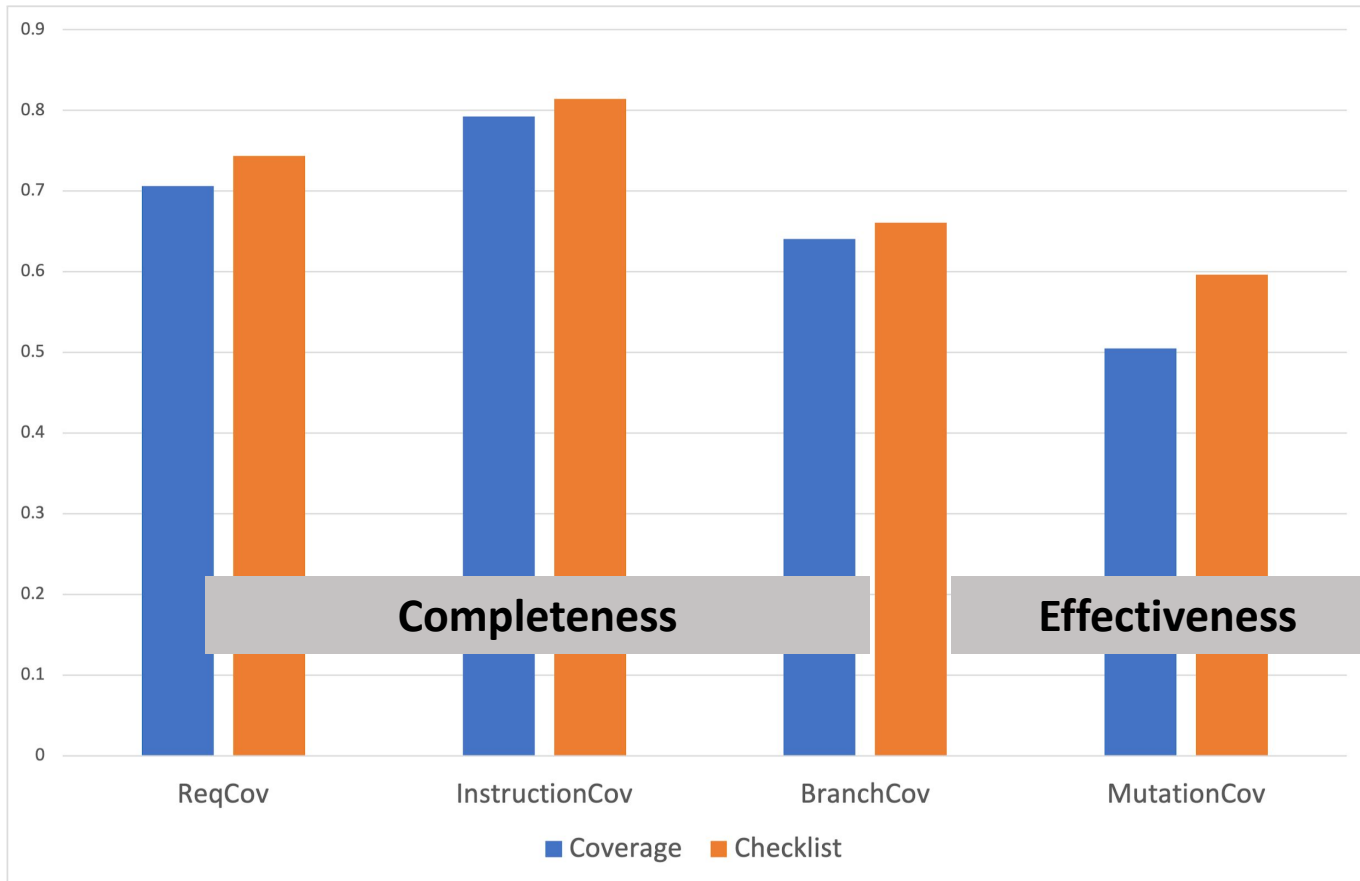
- ☐ measure code coverage using an appropriate tool, such as EcEmma ([installation](#), [tutorial](#)). Inspect uncovered code and w appropriate.

Methodology





Checklists vs. Coverage Tools

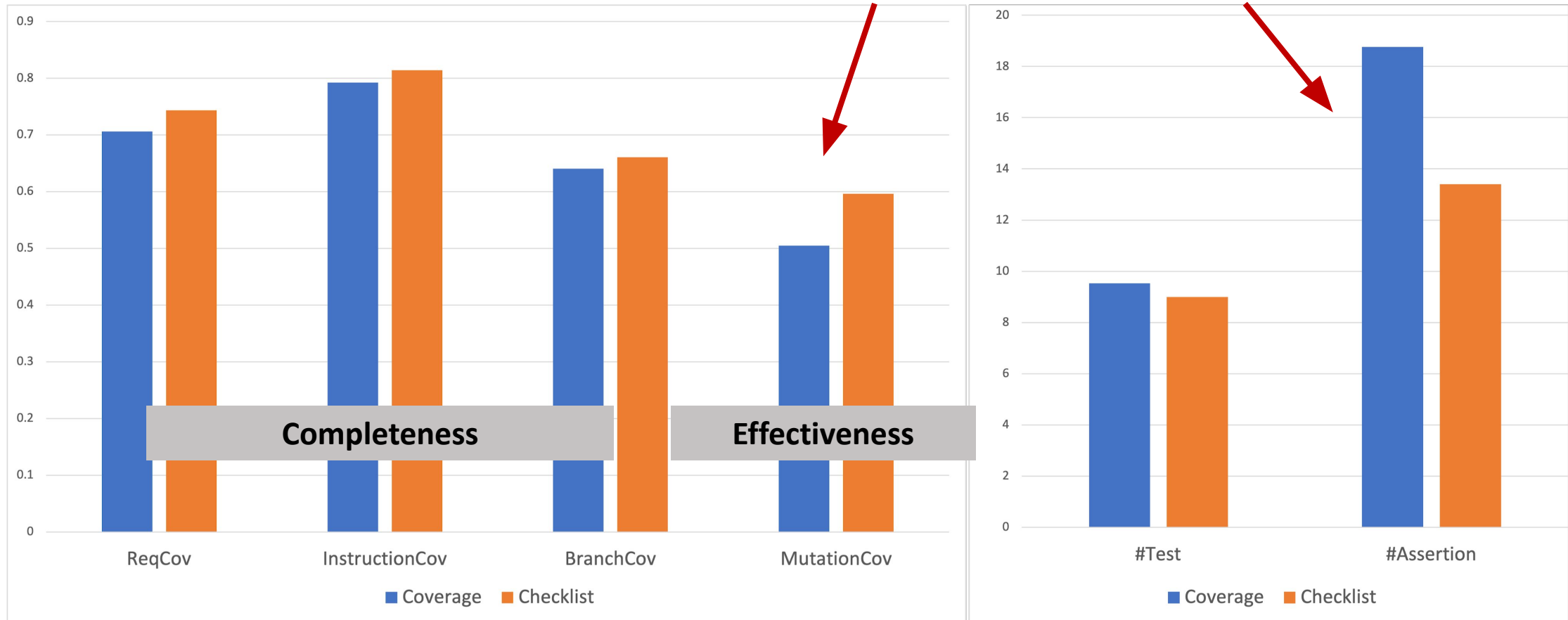


Tool support does not need to be sophisticated to be effective!



Checklists vs. Coverage Tools

fewer assertions, but higher mutation coverage





Looking Ahead...



generate tests for a bowling scorekeeper



1. Test: Calculate total score

- Input: Player A rolls [5, 3, 8, 2, 10, 4, 6, 10, 9, 0, 5, 5, 10, 8, 1]
- Expected Output: Total score = 142

2. Test: Calculate total score with spare bonus

- Input: Player A rolls [5, 3, 8, 2, 10, 4, 6, 10, 9, 1, 5, 5, 10, 8, 1]
- Expected Output: Total score = 152

3. Test: Calculate total score with strike bonus

- Input: Player A rolls [5, 3, 8, 2, 10, 4, 6, 10, 10, 10, 5, 5, 10, 8, 1]
- Expected Output: Total score = 201

4. Test: Calculate total score with consecutive strikes (turkey)

- Input: Player A rolls [10, 10, 10, 5, 3, 8, 2, 10, 4, 6, 10, 9, 1, 5, 5, 10]

- Are the tests correct?
- Are the tests complete?
- What happens if the requirements change?



FIND IT



THINK IT



TEST IT



FIX IT



Automated Program Repair

- Innovations:
 - Program Repair fueled by semantic search creates patches that are less prone to over-fitting [ASE 2015]
 - Semantic search is promising for producing high-quality real-world defect repairs [TSE 2021]

SHF: Medium: Collaborative Research: Semi and Fully Automated Program Repair and Synthesis via Semantic Code Search \$387k (co-PI, \$1.2m total) [completed]



SHF: EAGER: Collaborative Research: Demonstrating the Feasibility of Automatic Program Repair Guided by Code Search. \$87k (co-PI, \$287; total) [completed]



Automated Program Repair

- Innovations:
 - **Program Repair fueled by semantic search creates patches that are less prone to over-fitting [ASE 2015]**
 - **Semantic search is promising for producing high-quality real-world defect repairs [TSE 2021]**

SHF: Medium: Collaborative Research: Semi and Fully Automated Program Repair and Synthesis via Semantic Code Search \$387k (co-PI, \$1.2m total) [completed]

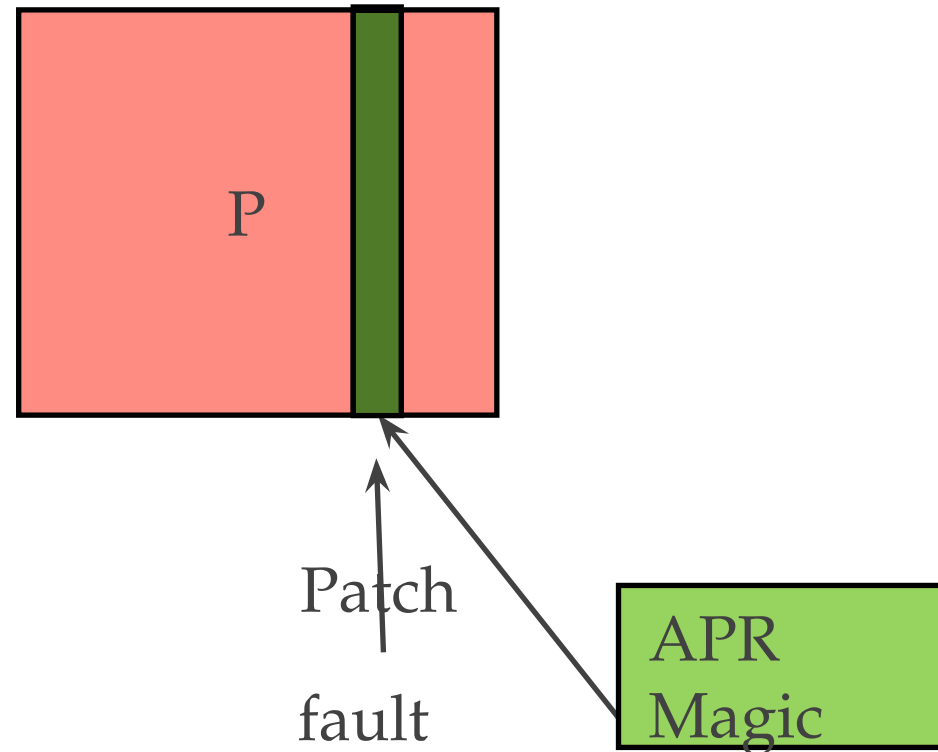


SHF: EAGER: Collaborative Research: Demonstrating the Feasibility of Automatic Program Repair Guided by Code Search. \$87k (co-PI, \$287; total) [completed]



Automated Program Repair

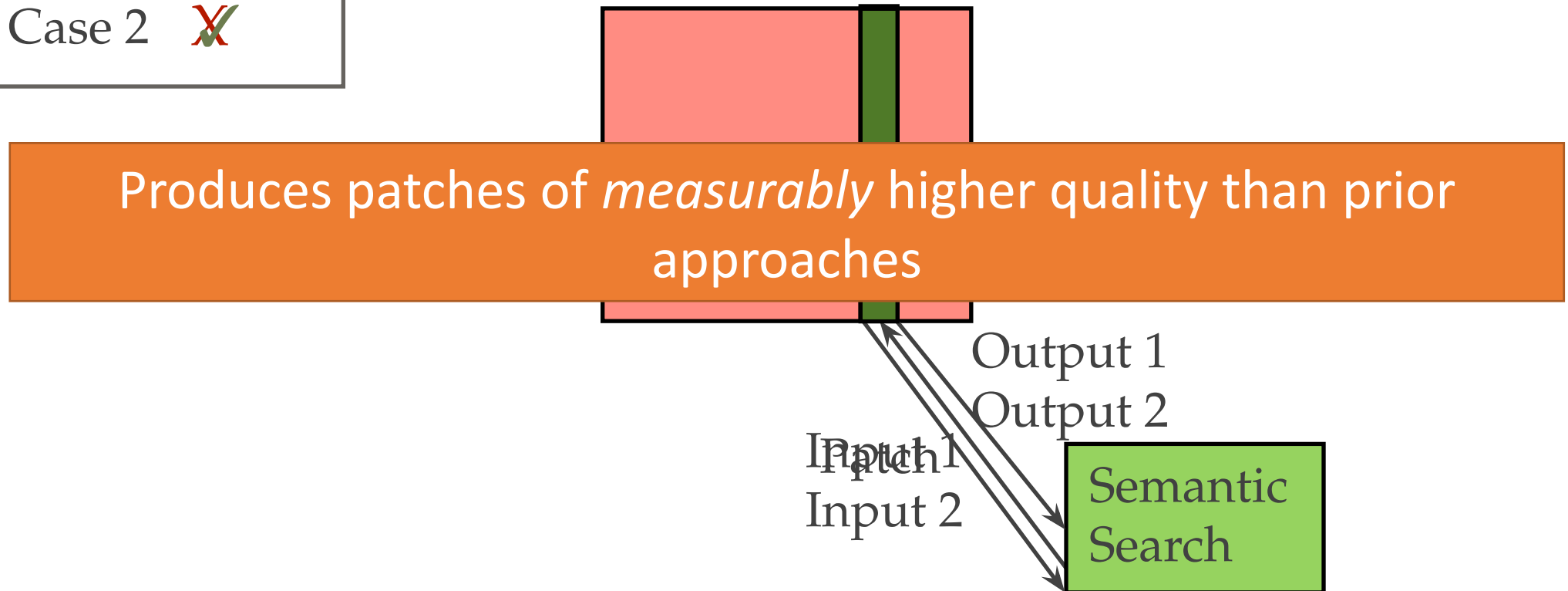
Test Suite	
Test Case 1	✓
Test Case 2	✗





My Automated Program Repair

Test Suite	
Test Case 1	✓
Test Case 2	✗





Can it patch **real** bugs?



Python bug #69223

Developer
Patch

```
    }  
+   if (timeout < 0) {  
+       PyErr_SetString(PyExc_ValueError ,  
+           "timeout must be non-negative");  
+       return NULL;  
+   }  
    seconds = (long)timeout;
```

SearchRepair
Patch

```
if ( timeout < 0  
    PyErr_SetString(PyExc_ValueError ,  
        "read length must be positive");  
    return NULL;  
}  
seconds = (long)timeout;
```



Looking Ahead...



please fix this code:

```
if (timeout < 0) {  
    PyErr_SetString(PyExc_ValueError ,  
        "timeout must be non-negative"); + return NULL;  
}  
seconds = (long)timeout;
```



The code you provided seems to be a snippet from a C extension module for Python. It checks if a variable `timeout` is less than zero and raises a `ValueError` with the message "timeout must be non-negative" if that condition is true.

The issue with the code is that there is an extra character, namely `+`, on the line before `return NULL;`. It seems to be a formatting mistake. To fix the code, remove that extra character, and the code will work as intended.

Here's the corrected code:

```
c  
Copy code  
  
if (timeout < 0) {  
    PyErr_SetString(PyExc_ValueError, "timeout must be non-negative");  
    return NULL;  
}  
seconds = (long)timeout;
```

After making this correction, the code should compile and execute without any issues.

ChatGPT

find another



Certainly! Here's an alternative solution that addresses the issue while providing a clear structure to the code:

```
c  
Copy code  
  
if (timeout < 0) {  
    PyErr_SetString(PyExc_ValueError, "timeout must be non-negative");  
    return NULL;  
} else {  
    seconds = (long)timeout;  
    // Rest of the code goes here  
}
```




FIND IT



THINK IT







TEST IT

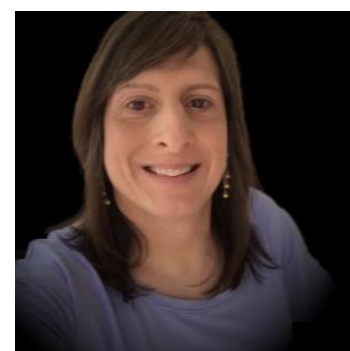


FIX IT

Big Results

-  **Code search via analysis** leads to high precision and has promise for a variety of applications
-  **Program Repair via code search** leads to patches that are higher quality
-  **Comparative comprehension** is challenging when behaviors are close but not exact
-  **Testing via checklist** is as effective as coverage tools for early students

Teamwork makes it happen.



.... And
more!

Thanks!

ktstolee@ncsu.edu



Thanks!

ktstolee@ncsu.edu

