Augmenting Stack Overflow with API Usage Patterns Mined from GitHub

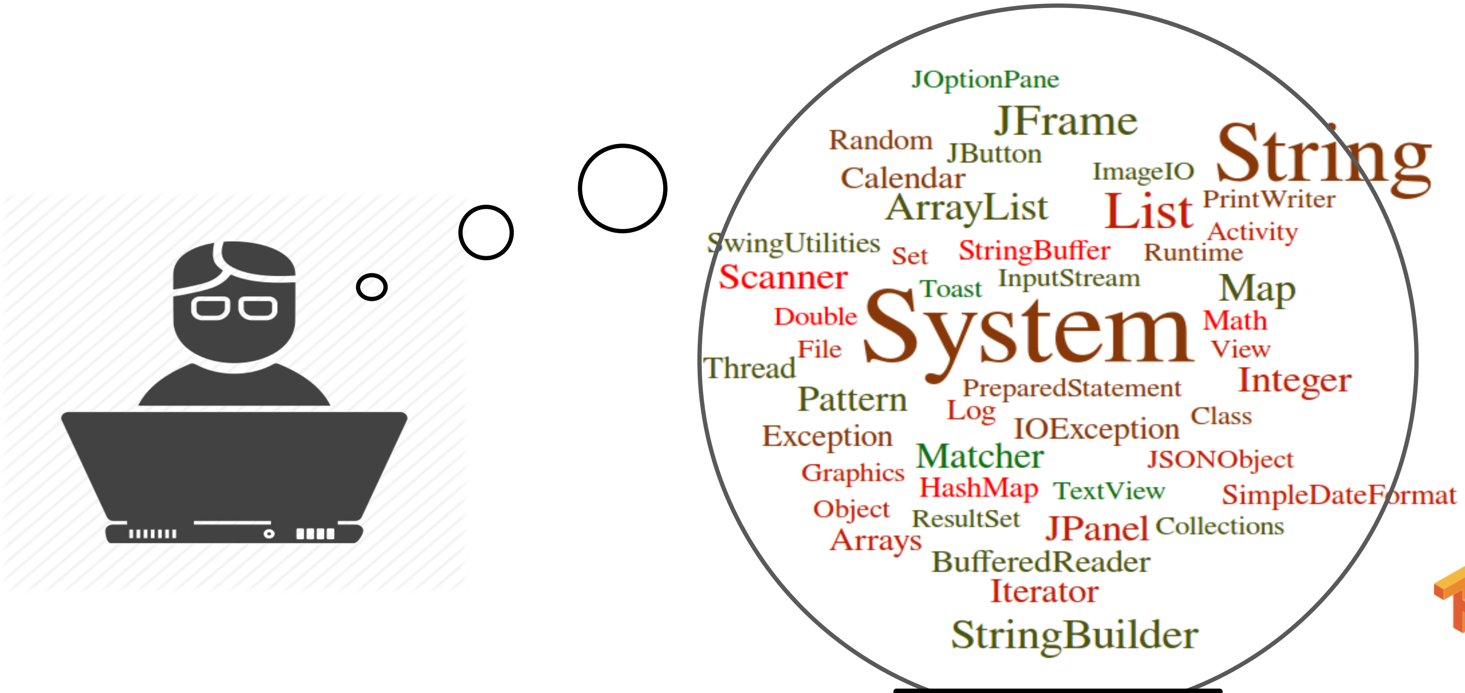
Anastasia Reinhart[1,2]* **Tianyi Zhang[1]** Mihir Marthur[1] Miryung Kim[1]

[1]University of California, Los Angeles

[2]George Fox University

* Work done as a research intern at UCLA.

Using APIs properly is becoming a key challenge



e.g., JDK APIs

Android SDK

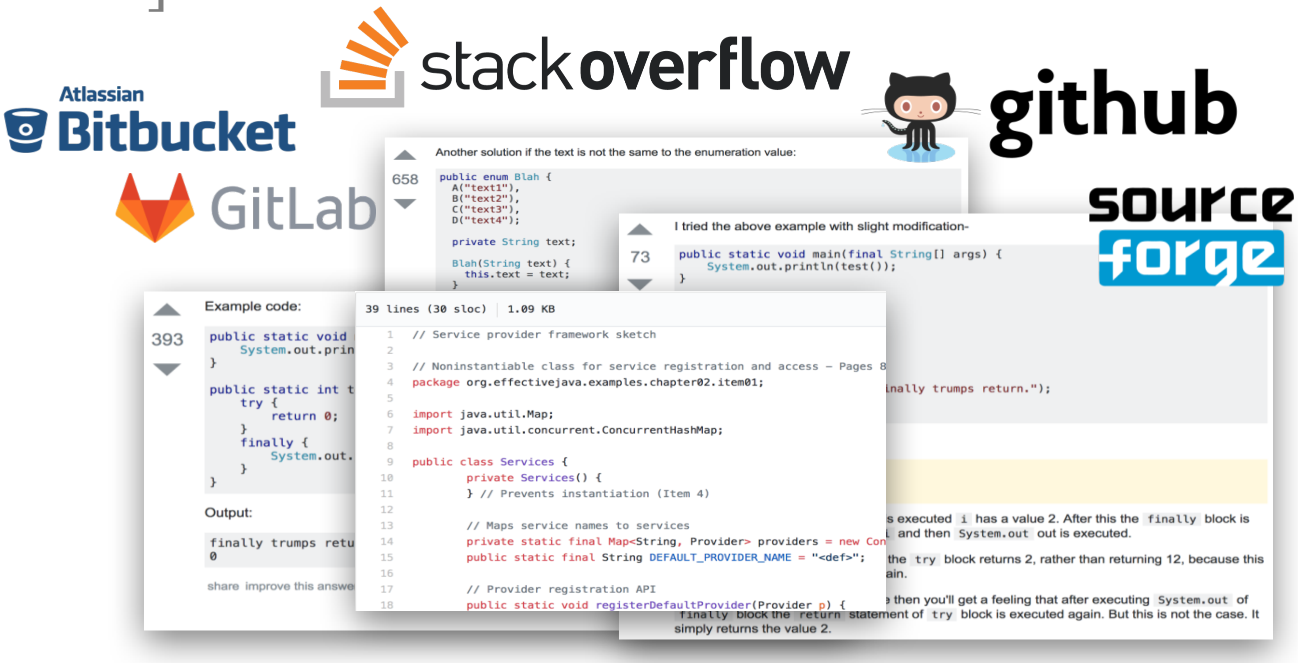
APACHE Spark

PyTorch

TensorFlow

The Status Quo of Learning APIs

Developers often search online for code examples to learn APIs [Sadowski et al. 2016]



Another solution if the text is not the same to the enumeration value:

658

```
public enum Blah {
  A("text1"),
  B("text2"),
  C("text3"),
  D("text4");

  private String text;

  Blah(String text) {
    this.text = text;
  }
```

I tried the above example with slight modification-

73

```
public static void main(final String[] args) {
    System.out.println(test());
}
```

```
inally trumps return.");
```

s executed i has a value 2. After this the finally block is
i and then System.out out is executed.

the try block returns 2, rather than returning 12, because this
ain.

e then you'll get a feeling that after executing System.out of
finally block the return statement of try block is executed again. But this is not the case. It
simply returns the value 2.

Example code:

393

```
public static void
    System.out.prin
}

public static int t
    try {
        return 0;
    }
    finally {
        System.out.
    }
}
```

Output:

```
finally trumps retu
0
```

share improve this answe

39 lines (30 sloc) | 1.09 KB

```
// Service provider framework sketch

// Noninstantiable class for service registration and access - Pages 8
package org.effectivejava.examples.chapter02.item01;

import java.util.Map;
import java.util.concurrent.ConcurrentHashMap;

public class Services {
        private Services() {
        } // Prevents instantiation (Item 4)

        // Maps service names to services
        private static final Map<String, Provider> providers = new Con
        public static final String DEFAULT_PROVIDER_NAME = "<def>";

        // Provider registration API
        public static void registerDefaultProvider(Provider p) {
```

The Limitation of Online Code Examples

- Programmers can only inspect a handful of search results. [Brandt et al., 2009, Starke et al., 2009, Duala-Ekoko and Robillard, 2012]
- Individual code examples may suffer from
  - insecure coding practices [Fischer et al., 2017]
  - unchecked obsolete usage [Zhou and Walker, 2016]
  - low readability [Treude and Robillard, 2017]

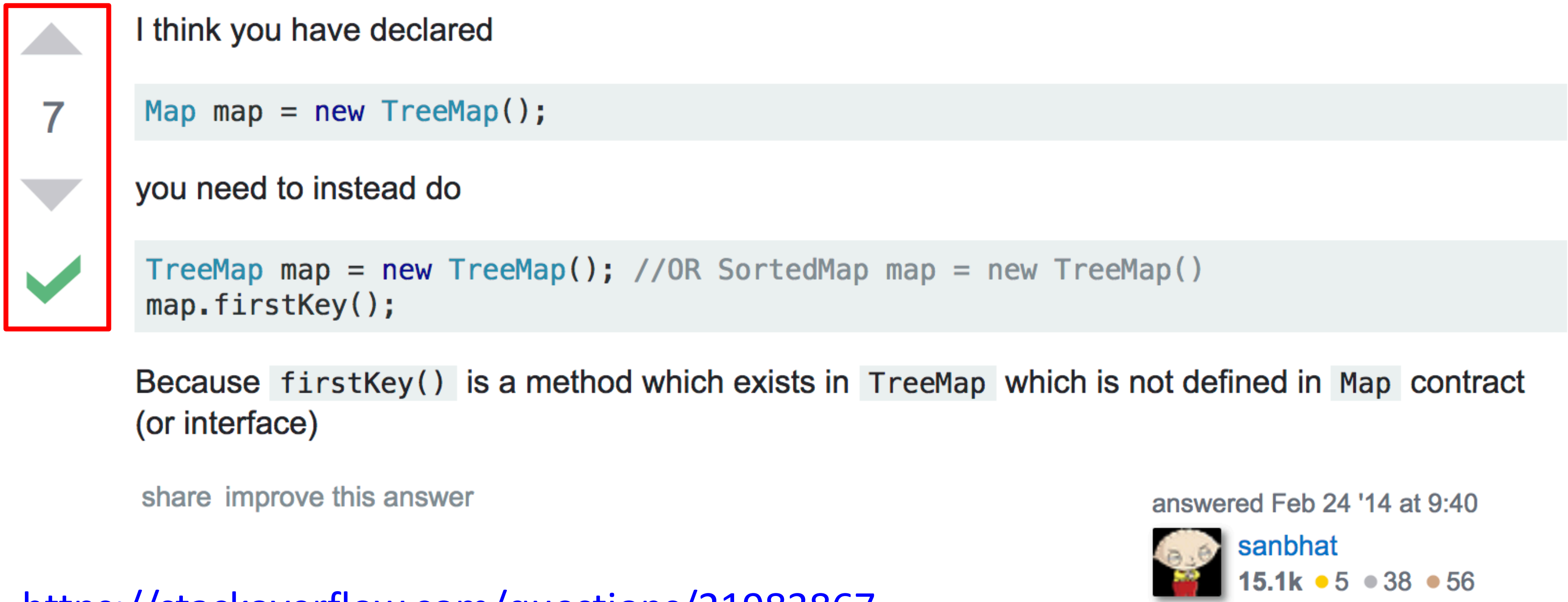
The Limitation of Online Code Examples

- Programmers can only inspect a handful of search results. [Brandt et al., 2009, Starke et al., 2009, Duala-Ekoko and Robillard, 2012]
  - insecure coding practices [Fischer et al., 2017]
  - unchecked obsolete usage [Zhou and Walker, 2016]

A recent study shows that 31% of SO posts have potential API usage violations.

Zhang et al., Are Online Code Examples Reliable? A Study of API Misuse on Stack Overflow, ICSE 2018
Dataset: http://web.cs.ucla.edu/~tianyi.zhang/examplecheck.html

Missing If Checks



7

✓

I think you have declared

```
Map map = new TreeMap();
```

you need to instead do

```
TreeMap map = new TreeMap(); //OR SortedMap map = new TreeMap()
map.firstKey();
```

Because `firstKey()` is a method which exists in `TreeMap` which is not defined in `Map` contract (or interface)

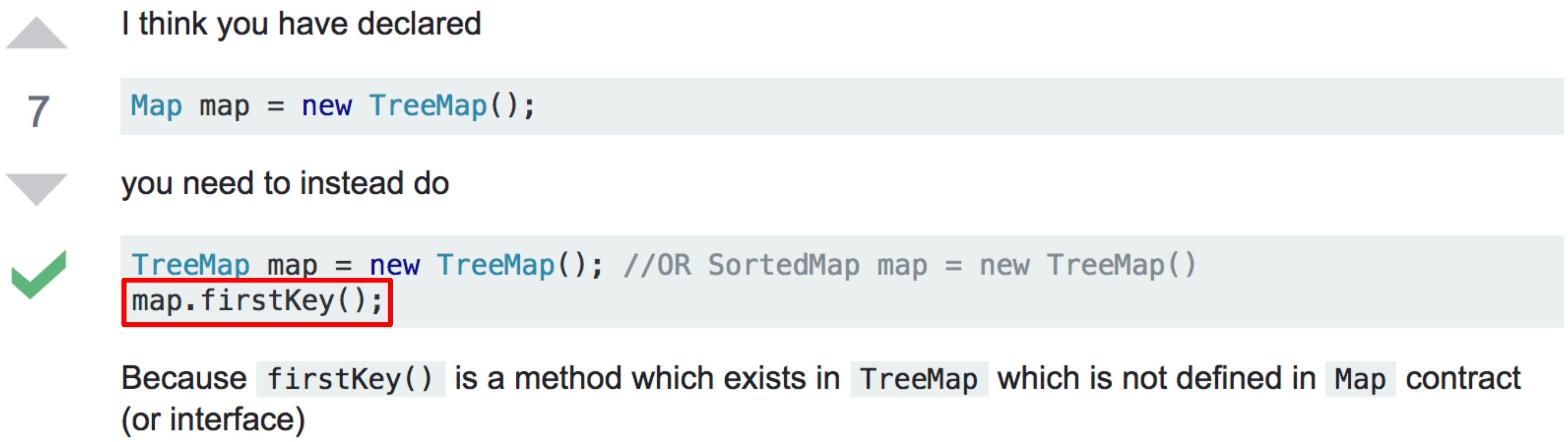
answered Feb 24 '14 at 9:40

sanbhat

15.1k ● 5 ● 38 ● 56

https://stackoverflow.com/questions/21983867

Missing If Checks



I think you have declared

```
Map map = new TreeMap();
```

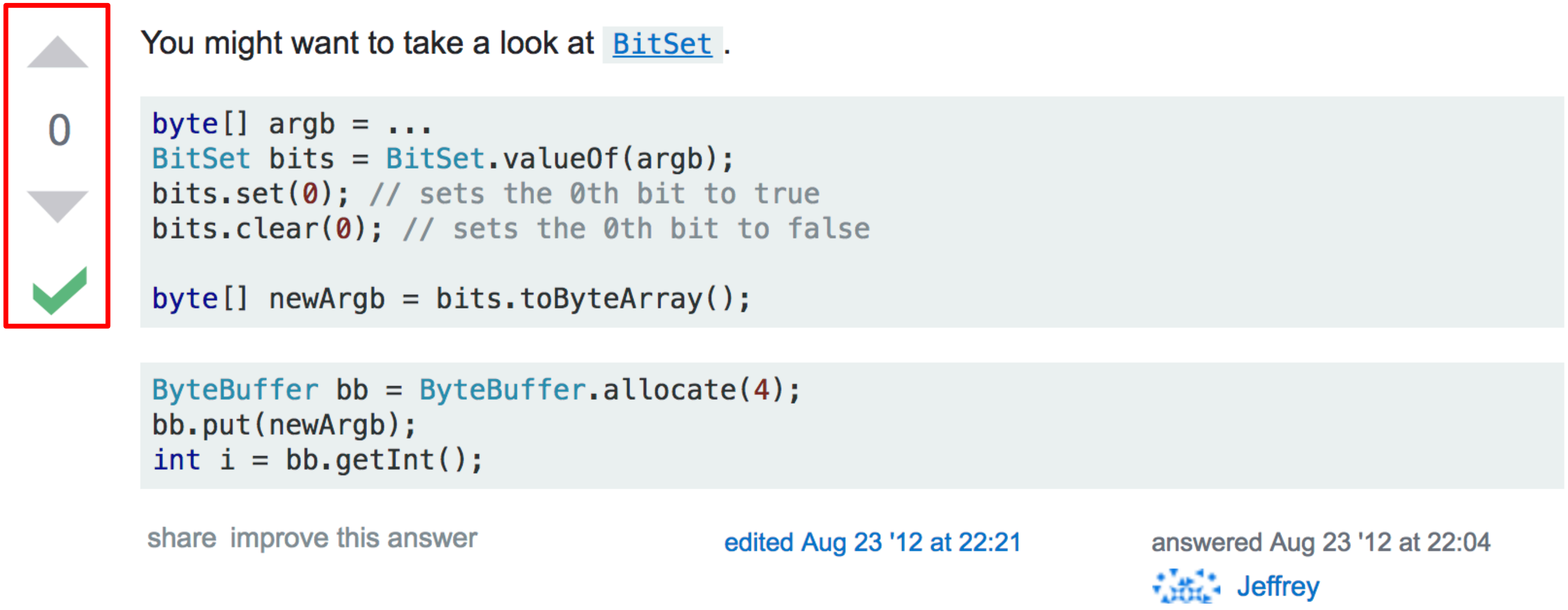
you need to instead do

```
TreeMap map = new TreeMap(); //OR SortedMap map = new TreeMap()
map.firstKey();
```

Because `firstKey()` is a method which exists in `TreeMap` which is not defined in `Map` contract (or interface)

This example throws NoSuchElementException. You should not call firstKey on an empty TreeMap.

Missing API Calls



You might want to take a look at BitSet .

```
byte[] argb = ...
BitSet bits = BitSet.valueOf(argb);
bits.set(0); // sets the 0th bit to true
bits.clear(0); // sets the 0th bit to false

byte[] newArgb = bits.toByteArray();
```

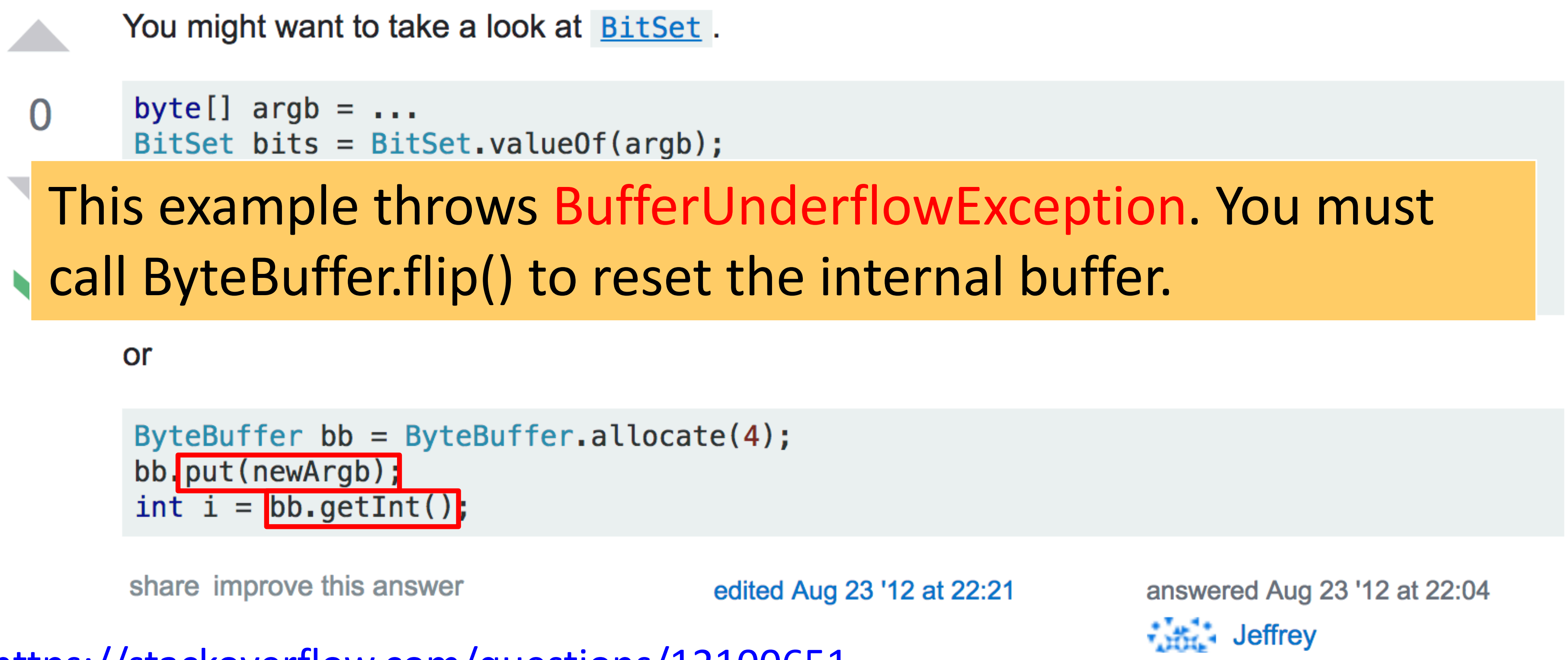
```
ByteBuffer bb = ByteBuffer.allocate(4);
bb.put(newArgb);
int i = bb.getInt();
```

edited Aug 23 '12 at 22:21

answered Aug 23 '12 at 22:04
Jeffrey

https://stackoverflow.com/questions/12100651

Missing API Calls



You might want to take a look at BitSet.

```
byte[] argb = ...
BitSet bits = BitSet.valueOf(argb);
```

This example throws BufferUnderflowException. You must call ByteBuffer.flip() to reset the internal buffer.

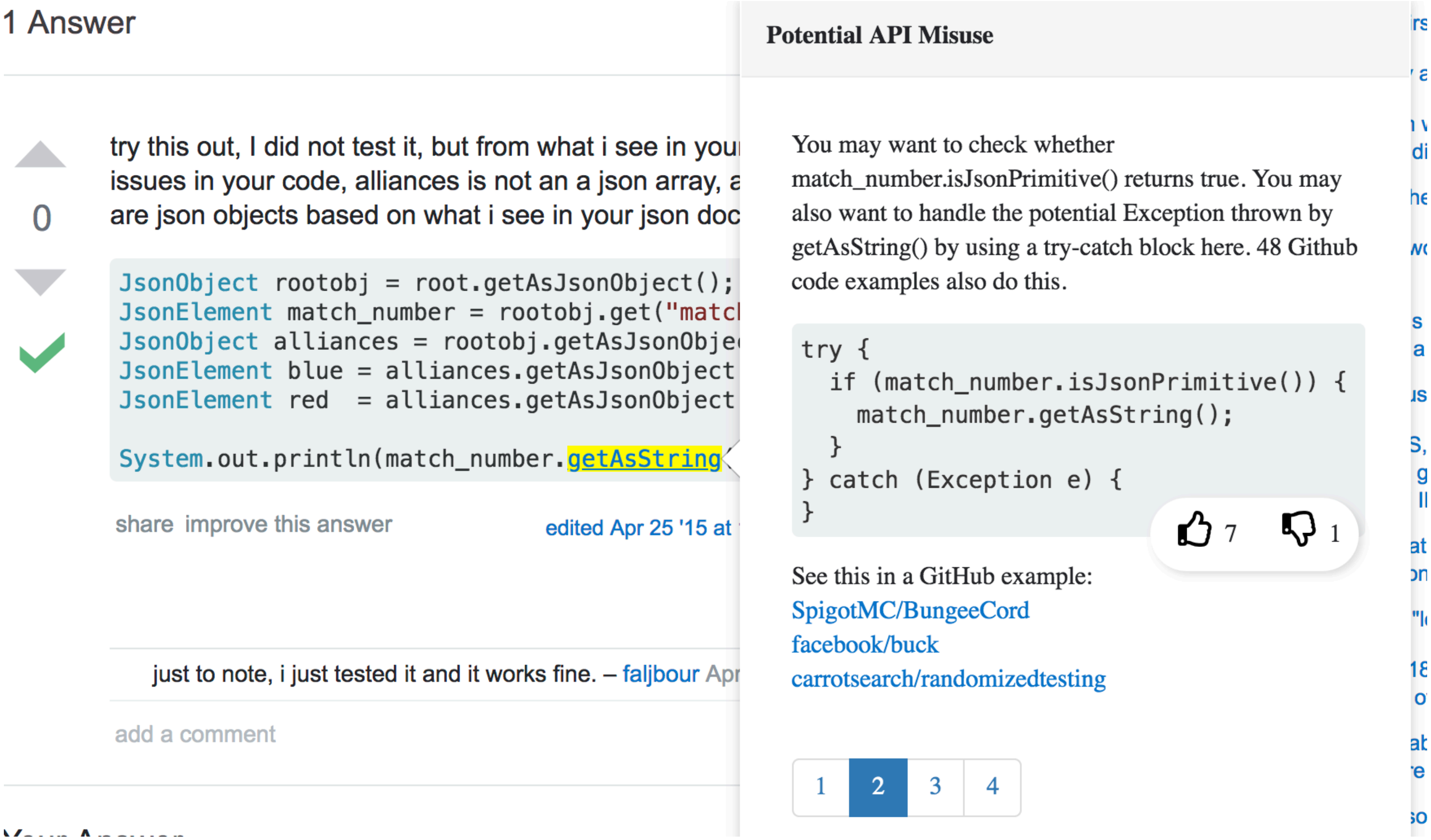
or

```
ByteBuffer bb = ByteBuffer.allocate(4);
bb.put(newArgb);
int i = bb.getInt();
```

share improve this answer
edited Aug 23 '12 at 22:21
answered Aug 23 '12 at 22:04
Jeffrey

https://stackoverflow.com/questions/12100651

ExampleCheck: Augmenting Stack Overflow with API Usage Patterns Mined from GitHub



1 Answer

0

try this out, I did not test it, but from what i see in you issues in your code, alliances is not an a json array, a are json objects based on what i see in your json doc

```
JsonObject rootobj = root.getAsJsonObject();
JsonElement match_number = rootobj.get("matc
JsonObject alliances = rootobj.getAsJsonObje
JsonElement blue = alliances.getAsJsonObject
JsonElement red  = alliances.getAsJsonObject

System.out.println(match_number.getAsString
```

share improve this answer edited Apr 25 '15 at

just to note, i just tested it and it works fine. – faljbour Apr

add a comment

Potential API Misuse

You may want to check whether match_number.isJsonPrimitive() returns true. You may also want to handle the potential Exception thrown by getAsString() by using a try-catch block here. 48 Github code examples also do this.

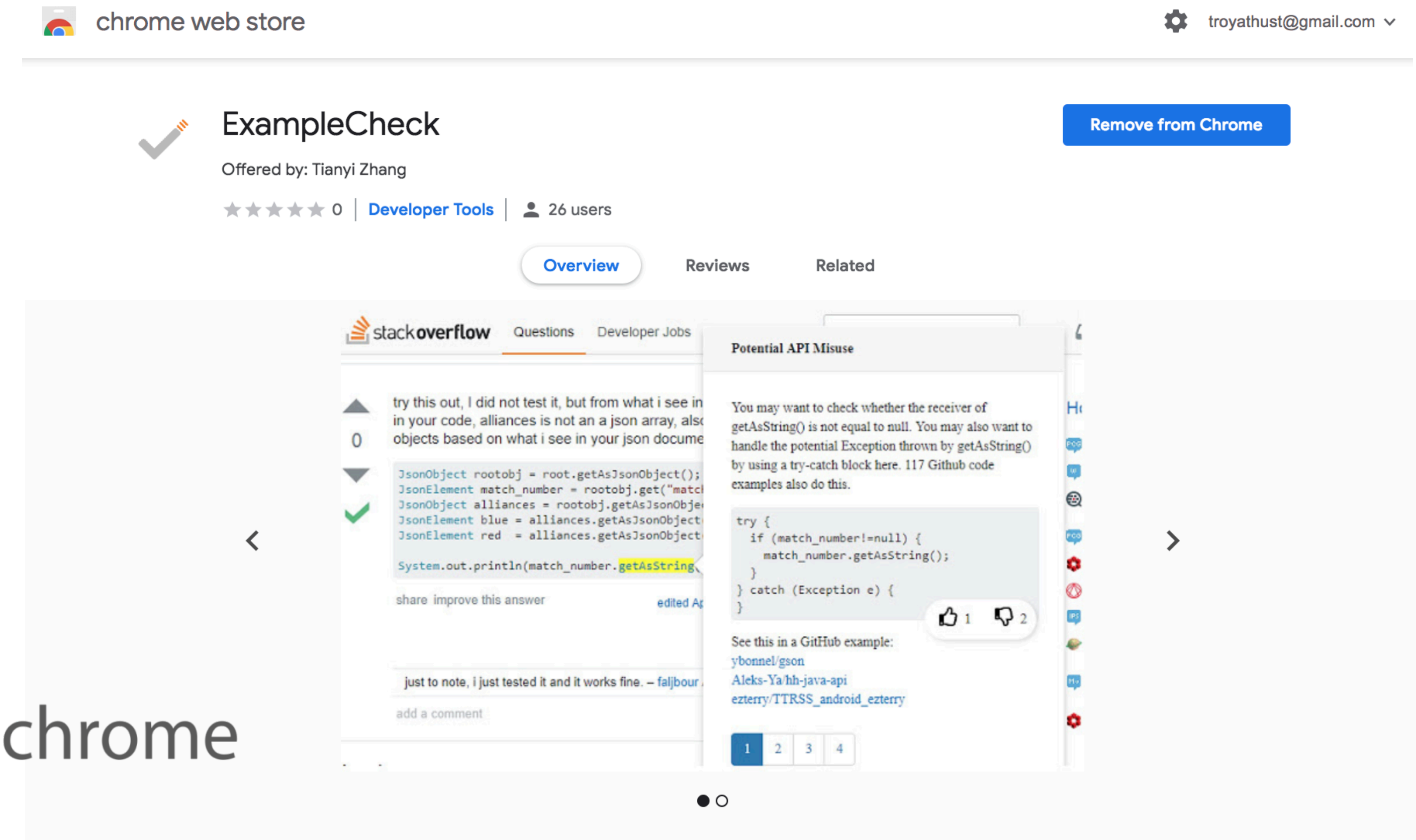
```
try {
  if (match_number.isJsonPrimitive()) {
    match_number.getAsString();
  }
} catch (Exception e) {
}
```

👍 7 👎 1

See this in a GitHub example:
SpigotMC/BungeeCord
facebook/buck
carrotsearch/randomizedtesting

1 2 3 4

Now available at Chrome Web Store!



chrome web store

troyathust@gmail.com

ExampleCheck

Offered by: Tianyi Zhang

☆☆☆☆☆ 0 | Developer Tools | 26 users

Remove from Chrome

Overview Reviews Related

try this out, I did not test it, but from what i see in
in your code, alliances is not an a json array, als
objects based on what i see in your json docume

```
JsonObject rootobj = root.getAsJsonObject();
JsonElement match_number = rootobj.get("matc
JsonObject alliances = rootobj.getAsJsonObje
JsonElement blue = alliances.getAsJsonObject
JsonElement red  = alliances.getAsJsonObject

System.out.println(match_number.getAsString
```

just to note, i just tested it and it works fine. – faljbour

Potential API Misuse

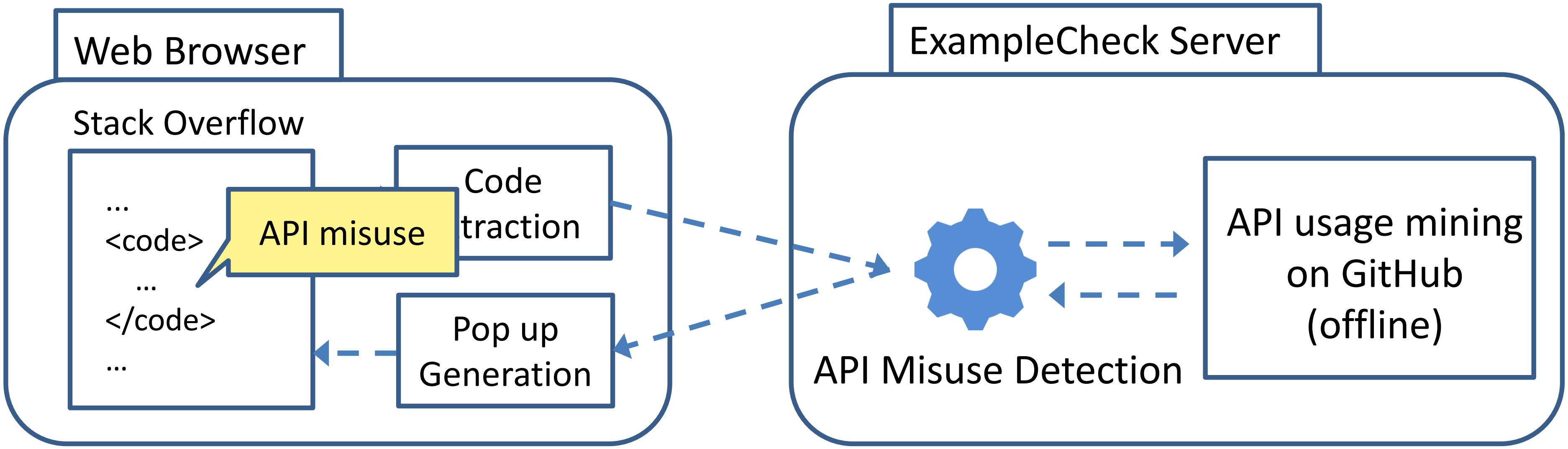
You may want to check whether the receiver of getAsString() is not equal to null. You may also want to handle the potential Exception thrown by getAsString() by using a try-catch block here. 117 Github code examples also do this.

```
try {
  if (match_number!=null) {
    match_number.getAsString();
  }
} catch (Exception e) {
}
```

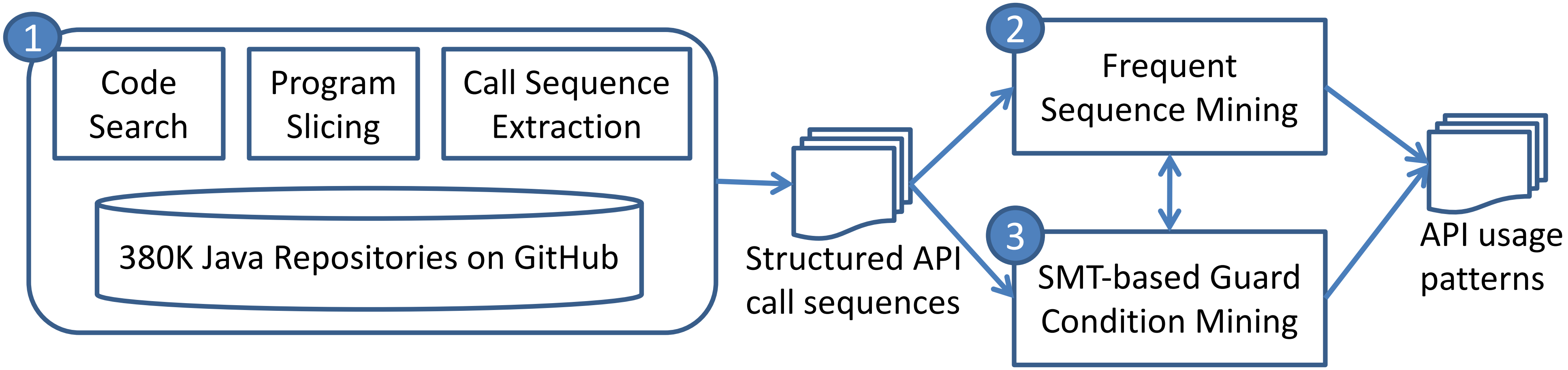
See this in a GitHub example:
ybonnel/gson
Aleks-Ya/hh-java-api
ezterry/TTRSS_android_ezterry

chrome

ExampleCheck Workflow

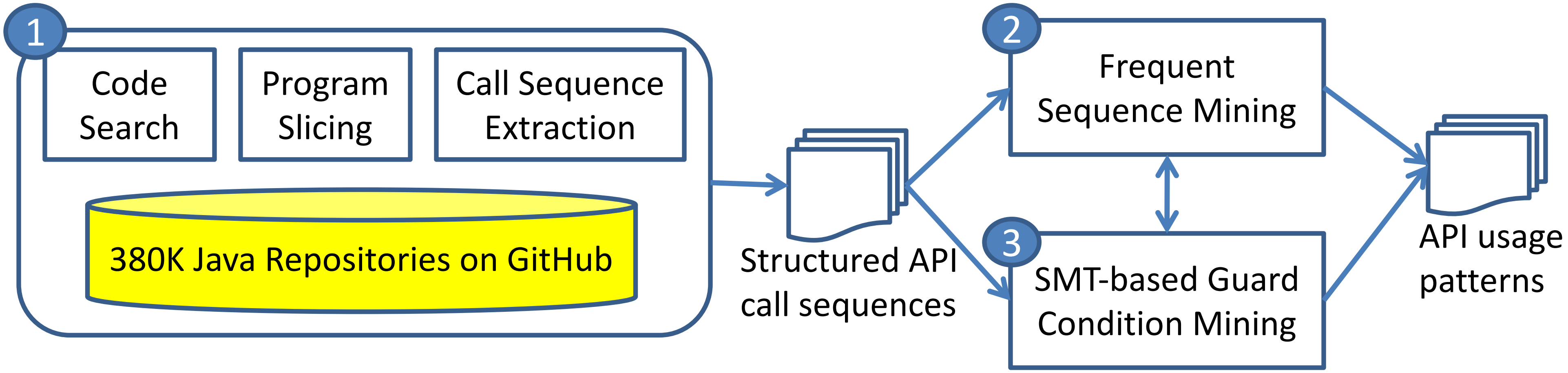


API Usage Mining from GitHub [ICSE 2018]



Insight 1: Mining a Large Code Corpus

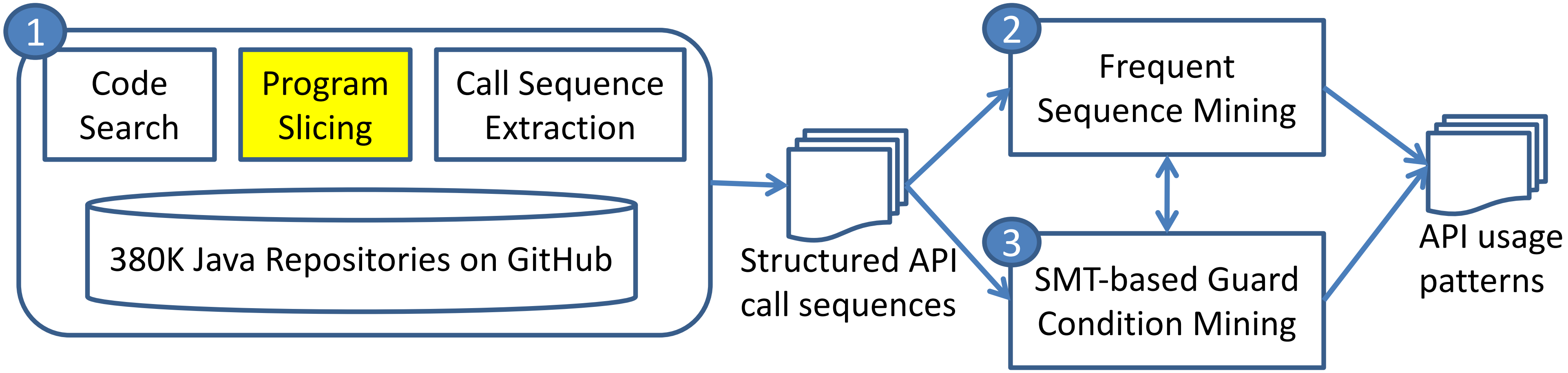
- Our code corpus includes 380K GitHub projects with at least 100 revisions and 2 contributors.



Dyer et al. Boa: A language and infrastructure for analyzing ultra-large-scale software repositories. ICSE 2013.

Insight 2: Removing Irrelevant Statements via Program Slicing

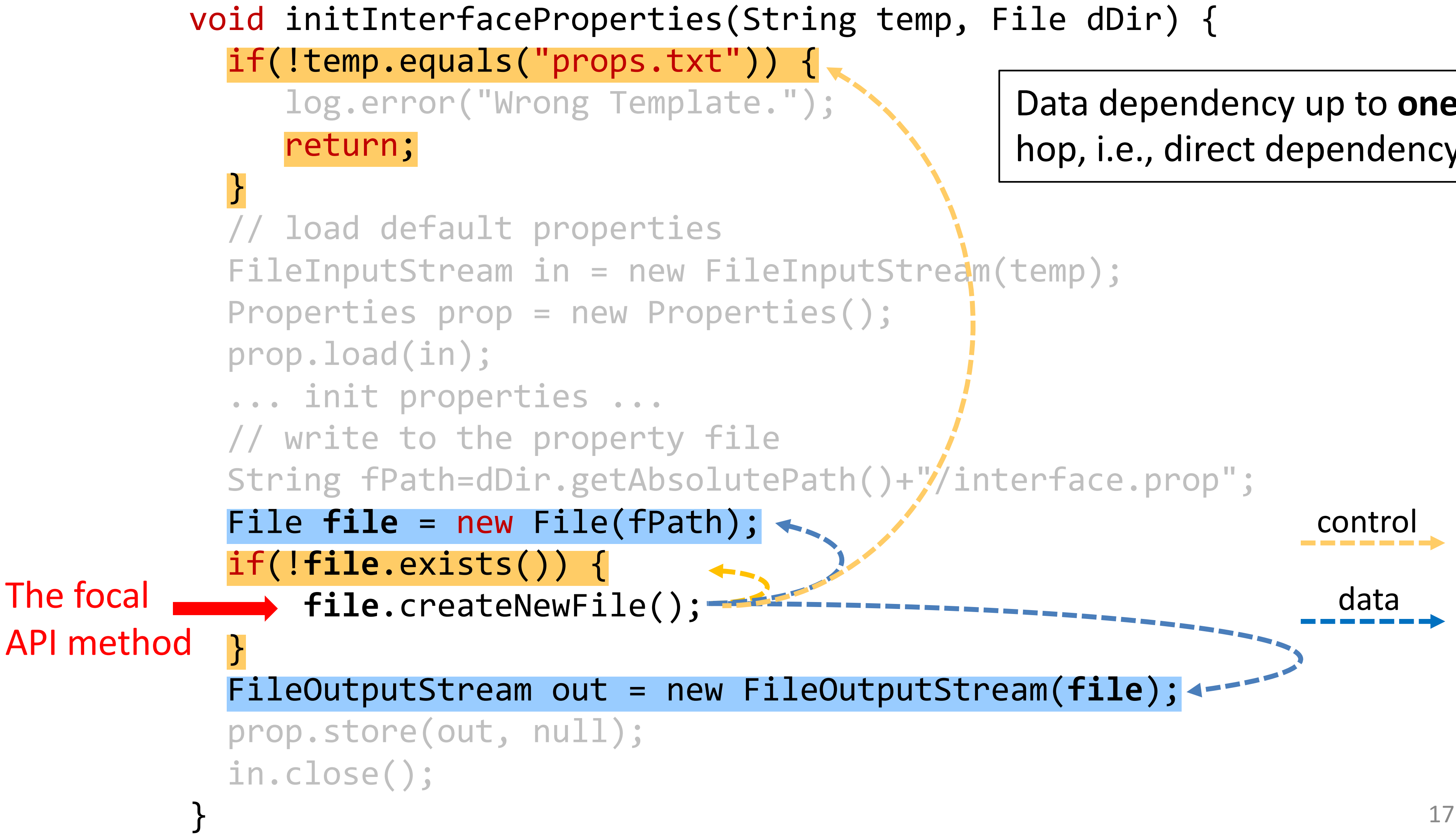
- We perform backward and forward slicing to identify data- and control-dependent statements to an API method of interest.



GitHub example of File.createNewFile

```
void initInterfaceProperties(String temp, File dDir) {
  if(!temp.equals("props.txt")) {
     log.error("Wrong Template.");
     return;
  }
  // load default properties
  FileInputStream in = new FileInputStream(temp);
  Properties prop = new Properties();
  prop.load(in);
  ... init properties ...
  // write to the property file
  String fPath=dDir.getAbsolutePath()+"/interface.prop";
  File file = new File(fPath);
  if(!file.exists()) {
      file.createNewFile();
  }
  FileOutputStream out = new FileOutputStream(file);
  prop.store(out, null);
  in.close();
}
```

The focal API method →

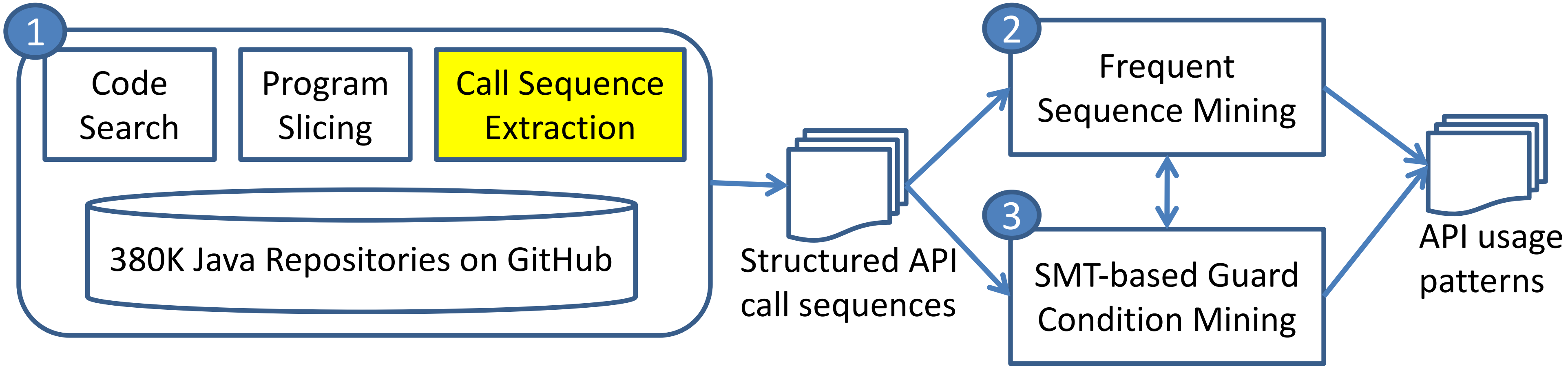


```
void initInterfaceProperties(String temp, File dDir) {
  if(!temp.equals("props.txt")) {
     log.error("Wrong Template.");
     return;
  }
  // load default properties
  FileInputStream in = new FileInputStream(temp);
  Properties prop = new Properties();
  prop.load(in);
  ... init properties ...
  // write to the property file
  String fPath=dDir.getAbsolutePath()+"/interface.prop";
  File file = new File(fPath);
  if(!file.exists()) {
    file.createNewFile();
  }
  FileOutputStream out = new FileOutputStream(file);
  prop.store(out, null);
  in.close();
}
```

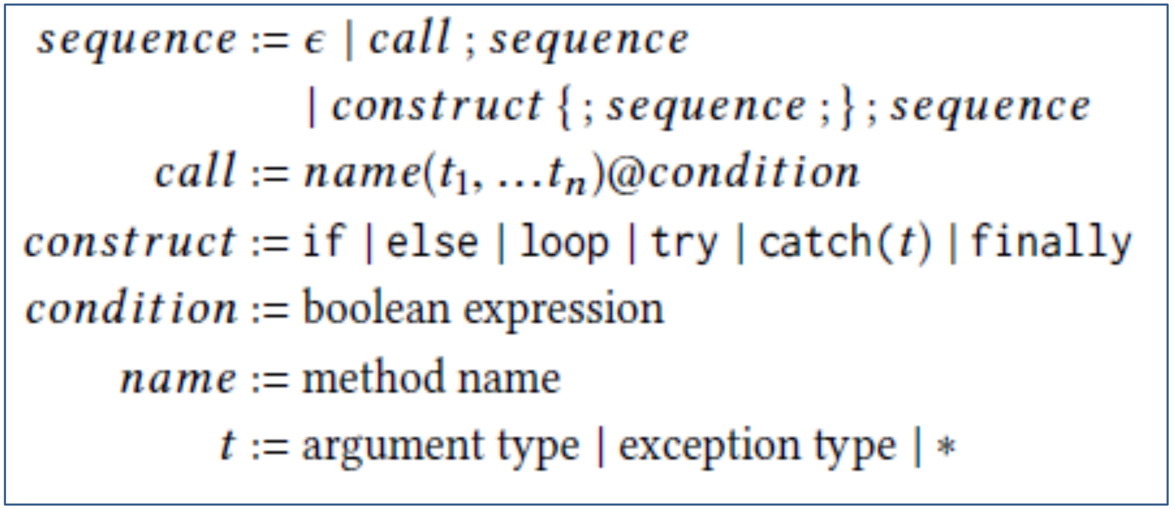
Data dependency up to **one** hop, i.e., direct dependency

Insight 3: Capture the Semantics of API Usage

- It is important to capture the temporal ordering, enclosing control structures, and appropriate guard conditions of API calls.



Insight 3: Capture the Semantics of API Usage

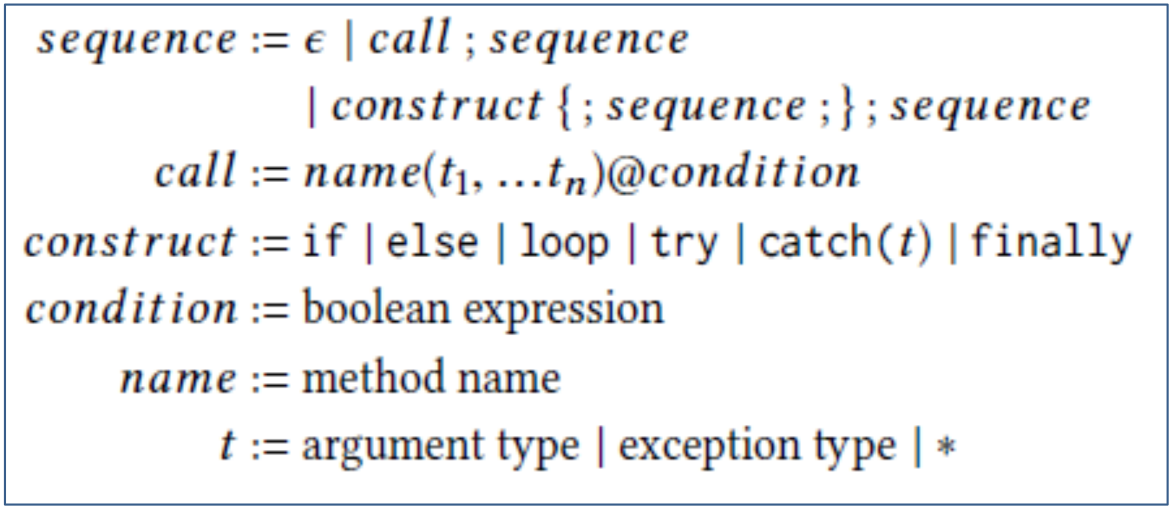


$$
\begin{aligned}
\mathit{sequence} &:= \epsilon \mid \mathit{call}\ ;\ \mathit{sequence} \\
&\quad \mid \mathit{construct}\ \{\ ;\ \mathit{sequence}\ ;\ \}\ ;\ \mathit{sequence} \\
\mathit{call} &:= \mathit{name}(t_1, \ldots t_n)@\mathit{condition} \\
\mathit{construct} &:= \mathtt{if} \mid \mathtt{else} \mid \mathtt{loop} \mid \mathtt{try} \mid \mathtt{catch}(t) \mid \mathtt{finally} \\
\mathit{condition} &:= \text{boolean expression} \\
\mathit{name} &:= \text{method name} \\
t &:= \text{argument type} \mid \text{exception type} \mid *
\end{aligned}
$$

Grammar of Structured
Call Sequences

new File (String); try {; new FileInputStream(File)@arg0.exists(); } catch (IOException) {; }

Insight 3: Capture the Semantics of API Usage

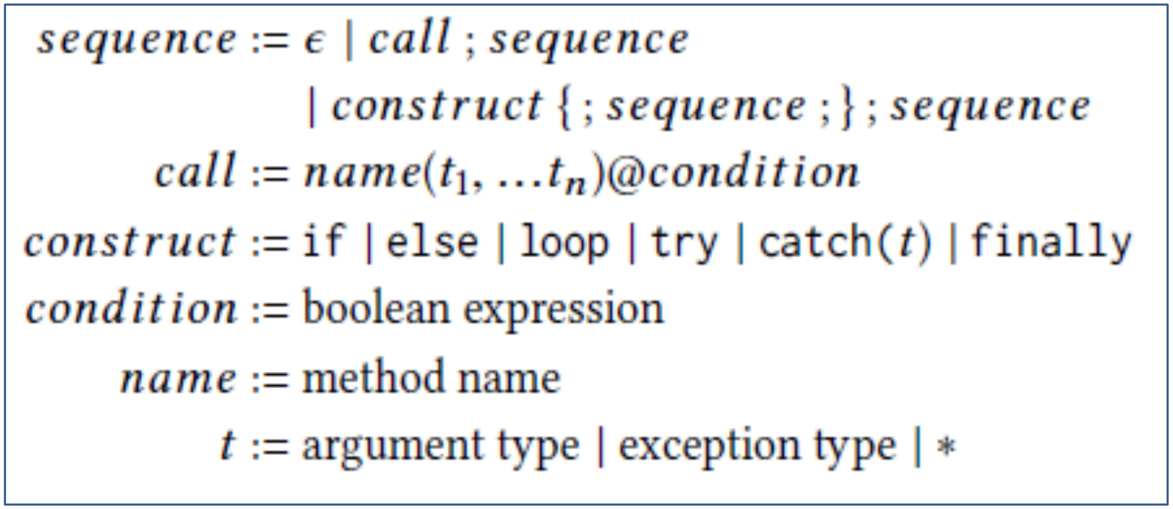


$$
\begin{aligned}
sequence &:= \epsilon \mid call \; ; sequence \\
&\quad \mid construct \; \{ \; ; sequence \; ; \} \; ; sequence \\
call &:= name(t_1, \ldots t_n)@condition \\
construct &:= \texttt{if} \mid \texttt{else} \mid \texttt{loop} \mid \texttt{try} \mid \texttt{catch}(t) \mid \texttt{finally} \\
condition &:= \text{boolean expression} \\
name &:= \text{method name} \\
t &:= \text{argument type} \mid \text{exception type} \mid *
\end{aligned}
$$

Grammar of Structured Call Sequences

new File (String); try {; new FileInputStream(File)@arg0.exists(); } catch (IOException) {; }

Insight 3: Capture the Semantics of API Usage

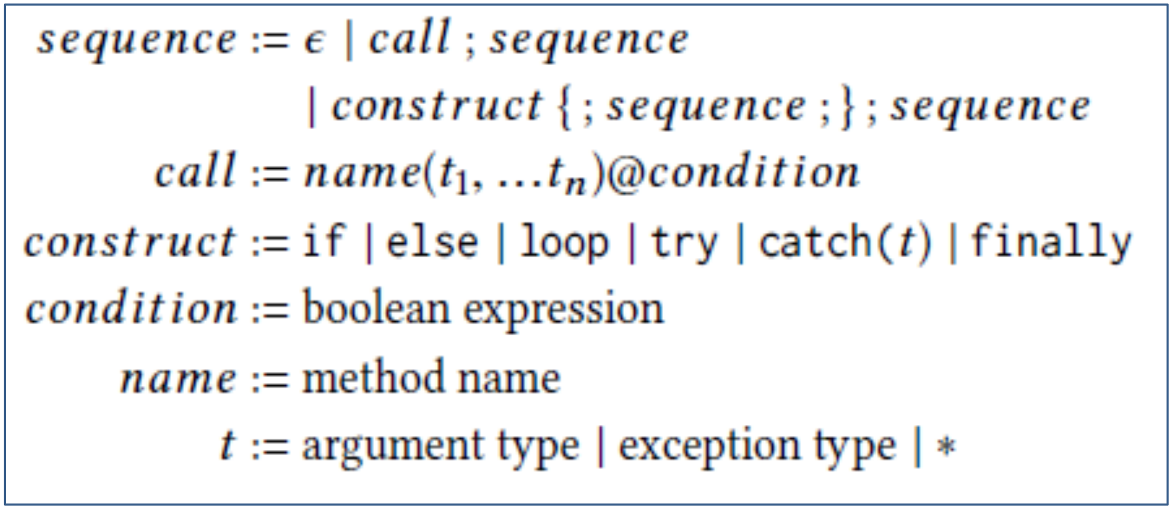


$$
\begin{aligned}
sequence &:= \epsilon \mid call\ ;\ sequence \\
&\quad \mid construct\ \{\ ;\ sequence\ ;\ \}\ ;\ sequence \\
call &:= name(t_1, \ldots t_n)@condition \\
construct &:= \texttt{if} \mid \texttt{else} \mid \texttt{loop} \mid \texttt{try} \mid \texttt{catch}(t) \mid \texttt{finally} \\
condition &:= \text{boolean expression} \\
name &:= \text{method name} \\
t &:= \text{argument type} \mid \text{exception type} \mid *
\end{aligned}
$$

Grammar of Structured Call Sequences

```
new File (String); try {; new FileInputStream(File)@arg0.exists(); } catch (IOException) {; }
```

Insight 3: Capture the Semantics of API Usage



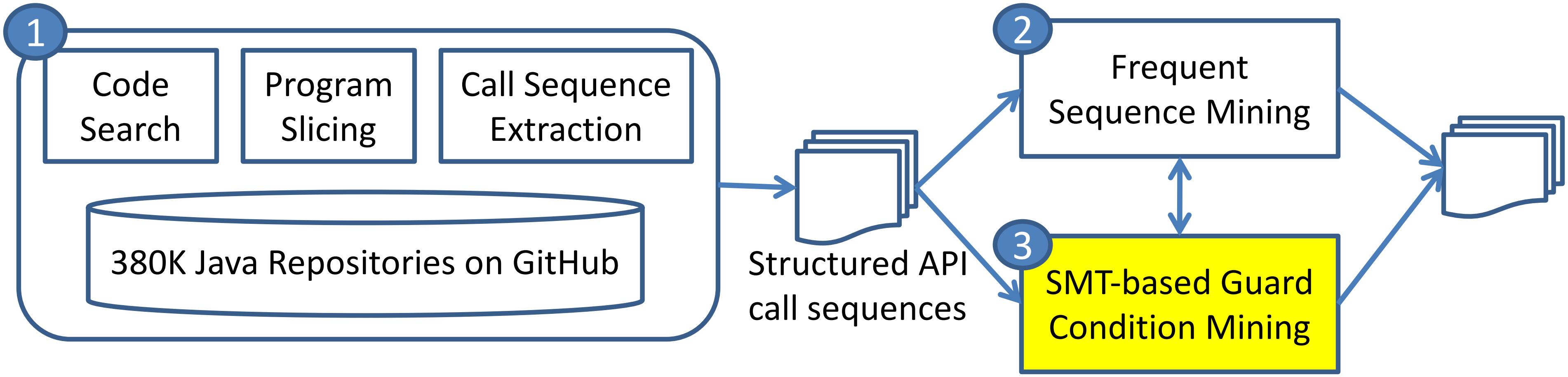
$$
\begin{aligned}
sequence &:= \epsilon \mid call \;;\; sequence \\
&\quad \mid construct \;\{\;;\; sequence \;;\; \}\;;\; sequence \\
call &:= name(t_1, \ldots t_n)@condition \\
construct &:= \texttt{if} \mid \texttt{else} \mid \texttt{loop} \mid \texttt{try} \mid \texttt{catch}(t) \mid \texttt{finally} \\
condition &:= \text{boolean expression} \\
name &:= \text{method name} \\
t &:= \text{argument type} \mid \text{exception type} \mid *
\end{aligned}
$$

Grammar of Structured Call Sequences

new File (String); try {; new FileInputStream(File)@arg0.exists(); } catch (IOException) {; }

Insight 4: SMT-based Guard Condition Mining

- GitHub developers may write the same predicate in different ways.


1
Code Search
Program Slicing
Call Sequence Extraction
380K Java Repositories on GitHub
Structured API call sequences
2
Frequent Sequence Mining
3
SMT-based Guard Condition Mining

Insight 4: SMT-based Guard Condition Mining

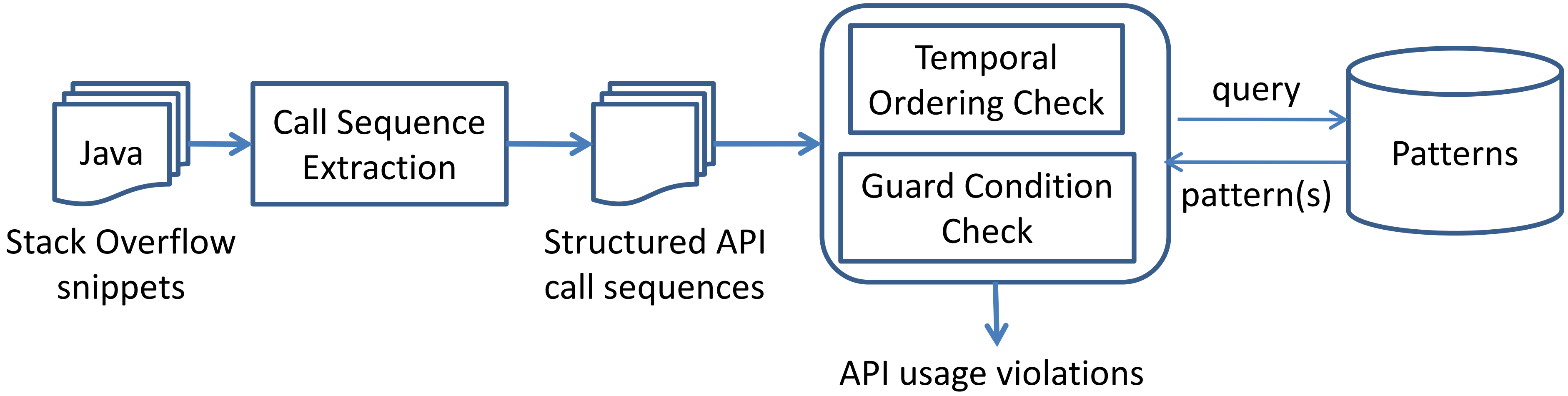
- We group guard conditions based on their logic equivalence.
- We use Z3 to prove the logic equivalence of guard conditions.
- $p \Leftrightarrow q$ is valid iff. $\neg((\neg p \vee q) \wedge (p \vee \neg q))$ is UNSAT.

Two equivalent but syntactically different guard conditions for `substring(int):`

```
arg0>=0 && arg0<=rcv.length() ⇔ arg0>-1 && arg0<rcv.length()+1
```

API Misuse Detection

- Contrast SO code snippets with mined API usage patterns automatically.



Extract Structured Call Sequence

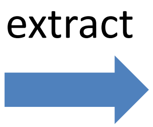
```
JsonObject obj =
  root.getAsJsonObject();
JsonElement match_number =
  obj.get("match_number");
...
System.out.println(
  match_number.getAsString());
```

SO code example [Post 29860000]

Extract Structured Call Sequence

```
JsonObject obj =
  root.getAsJsonObject();
JsonElement match_number =
  obj.get("match_number");
...
System.out.println(
  match_number.getAsString());
```

SO code example [Post 29860000]



extract

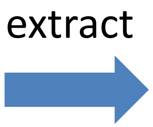
```
getAsJsonObject()@true;
get(String)@true;
...
getAsString()@true;
println(String)@true
```

Structured Call Sequence

Extract Structured Call Sequence

```
JsonObject obj =
  root.getAsJsonObject();
JsonElement match_number =
  obj.get("match_number");
...
System.out.println(
  match_number.getAsString());
```

SO code example [Post 29860000]



```
getAsJsonObject()@true;
get(String)@true;
...
getAsString()@true;
println(String)@true
```

Structured Call Sequence

API Usage Pattern for JsonElement.getAsString()

```
getAsJsonObject()@true;
get(String)@true;
...
getAsString()@true;
println(String)@true
```

Structured Call Sequence

```
try {;
getAsString()@rcv.isJsonPrimitive();
};
catch (Exception) {;
};
```

An API Usage Pattern of JsonElement.getAsString()

Temporal Ordering Check

```
getAsJsonObject()@true;
get(String)@true;
...
getAsString()@true;
println(String)@true
```

Structured Call Sequence

```
try {;
getAsString()@rcv.isJsonPrimitive();
};
catch (Exception) {;
};
```

An API Usage Pattern of JsonElement.getAsString()

missing a try-catch block!

Guard Condition Check

```
getAsJsonObject()@true;
get(String)@true;
...
getAsString()@true;
println(String)@true
```

Structured Call Sequence

```
try {;
getAsString()@rcv.isJsonPrimitive();
};
catch (Exception) {;
};
```

An API Usage Pattern of JsonElement.getAsString()

true ⇏ rcv.isJsonPrimitive() and thus incorrect guard condition!

Evaluation Results [ICSE 2018]

- 31% of 217K SO posts contain API usage violations.
- 72% of sampled posts with violations may cause program crashes, resource leaks, etc.
- Highly-voted posts are not necessarily more reliable in terms of correct API usage.

Live Demo

Summary

- Alert users about potential API usage violations in Stack Overflow using patterns mined from 380K GitHub projects
- Expand the scope of APIs beyond 100 Java and Android APIs
- Automate the end-to-end pipeline of API usage mining to keep API usage patterns up-to-date

Tool: https://chrome.google.com/webstore/detail/examplecheck/amliempebckaiaklimcpopomlnklkioe

Dataset: http://web.cs.ucla.edu/~tianyi.zhang/examplecheck.html

Q&A