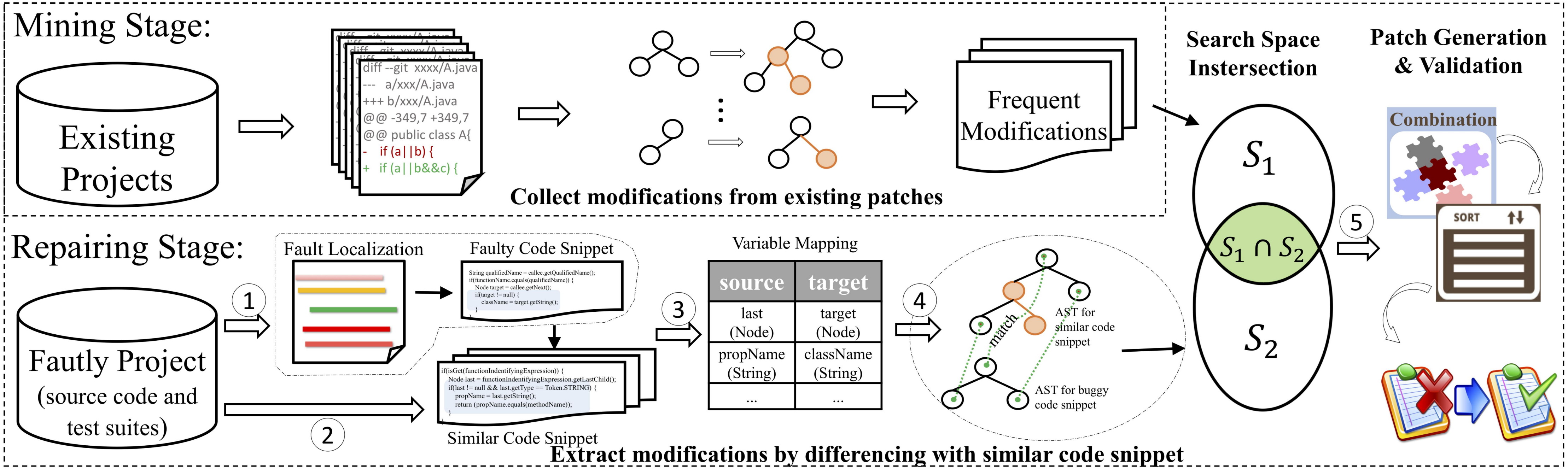
Tool: SimFix

Shaping Program Repair Space with Existing Patches and Similar Code (ISSTA’18)

Chenglin Li

About SimFix

SimFix is an **automatic program repair** technique, which leverages exisiting patches from other projects and similar code snippets in the same project to generate patches. The following figure is the workflow of our approach.


Mining Stage:
Existing Projects
Collect modifications from existing patches
Frequent Modifications
Search Space Instersection
S_1
$S_1 \cap S_2$
S_2
Patch Generation & Validation
Combination
SORT
Repairing Stage:
Fautly Project (source code and test suites)
①
②
③
④
⑤
Fault Localization
Faulty Code Snippet
Similar Code Snippet
Variable Mapping
match
AST for similar code snippet
AST for buggy code snippet
Extract modifications by differencing with similar code snippet

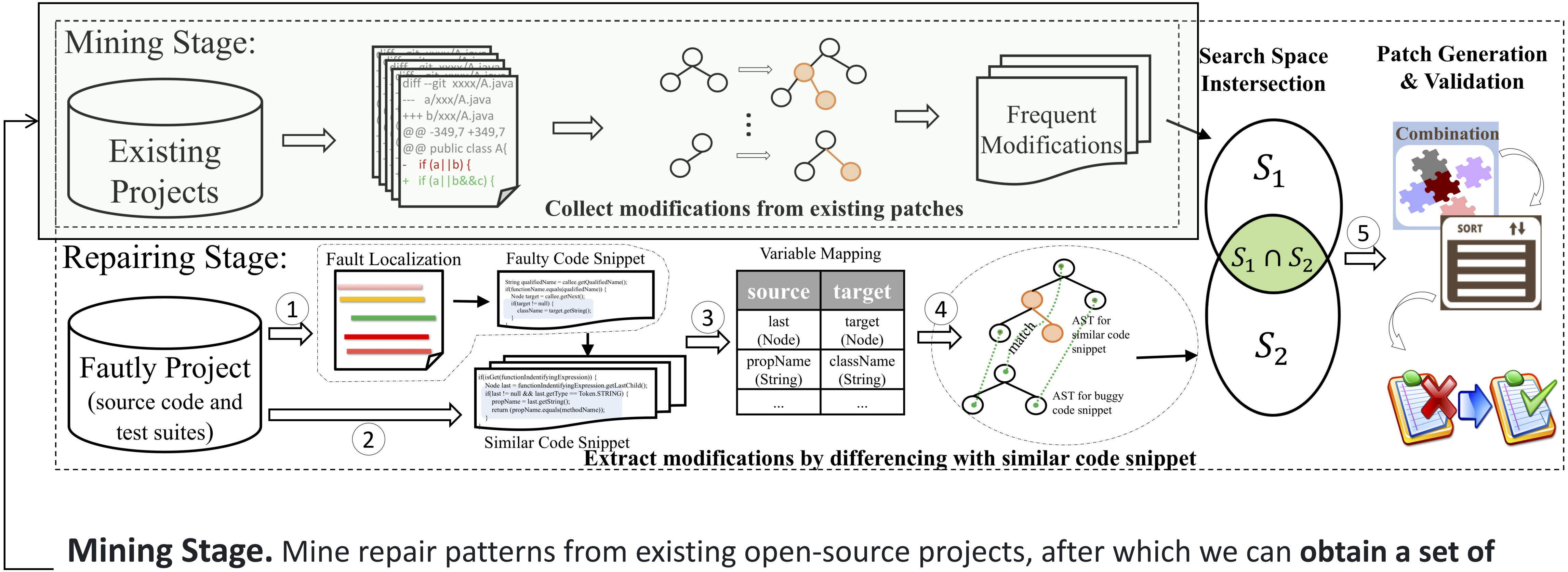
```
diff --git  xxxx/A.java
---   a/xxx/A.java
+++ b/xxx/A.java
@@ -349,7 +349,7
@@ public class A{
-    if (a||b) {
+    if (a||b&&c) {
```

```
String qualifiedName = callee.getQualifiedName();
if(functionName.equals(qualifiedName)) {
   Node target = callee.getNext();
   if(target != null) {
      className = target.getString();
   }
}
```

```
if(isGet(functionIndentifyingExpression)) {
   Node last = functionIndentifyingExpression.getLastChild();
   if(last != null && last.getType == Token.STRING) {
      propName = last.getString();
      return (propName.equals(methodName));
   }
}
```

source	target
last (Node)	target (Node)
propName (String)	className (String)
...	...

Idea of SimFix



Mining Stage:
Existing Projects
Frequent Modifications
Collect modifications from existing patches
Repairing Stage:
Fautly Project (source code and test suites)
①
②
Fault Localization
Faulty Code Snippet
Similar Code Snippet
③
Variable Mapping
④
match
AST for similar code snippet
AST for buggy code snippet
Extract modifications by differencing with similar code snippet
Search Space Instersection
S_1
$S_1 \cap S_2$
S_2
⑤
Patch Generation & Validation
Combination
SORT

```
diff --git xxxx/A.java
--- a/xxx/A.java
+++ b/xxx/A.java
@@ -349,7 +349,7
@@ public class A{
-   if (a||b) {
+   if (a||b&&c) {
```

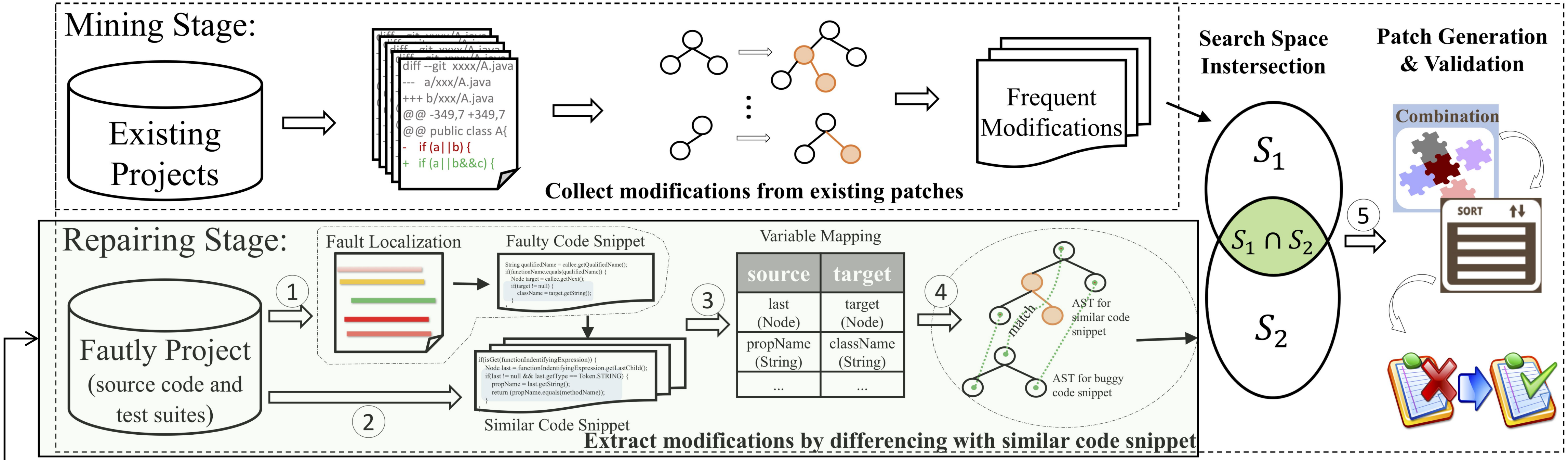
```
String qualifiedName = callee.getQualifiedName();
if(functionName.equals(qualifiedName)) {
  Node target = callee.getNext();
  if(target != null) {
    className = target.getString();
  }
```

```
if(isGet(functionIndentifyingExpression)) {
  Node last = functionIndentifyingExpression.getLastChild();
  if(last != null && last.getType == Token.STRING) {
    propName = last.getString();
    return (propName.equals(methodName));
  }
}
```

source	target
last (Node)	target (Node)
propName (String)	className (String)
...	...

Mining Stage. Mine repair patterns from existing open-source projects, after which we can **obtain a set of frequent repair patterns**. Those patterns **can be reused for other repairing** scenarios as well.

Idea of SimFix



Mining Stage:
Existing Projects
Collect modifications from existing patches
Frequent Modifications
Search Space Instersection
S_1
$S_1 \cap S_2$
S_2
Patch Generation & Validation
Combination
SORT
5
Repairing Stage:
Fautly Project (source code and test suites)
1
2
Fault Localization
Faulty Code Snippet
Similar Code Snippet
3
Variable Mapping
4
match
AST for similar code snippet
AST for buggy code snippet
Extract modifications by differencing with similar code snippet

```
diff --git  xxxx/A.java
---   a/xxx/A.java
+++ b/xxx/A.java
@@ -349,7 +349,7
@@ public class A{
-    if (a||b) {
+    if (a||b&&c) {
```

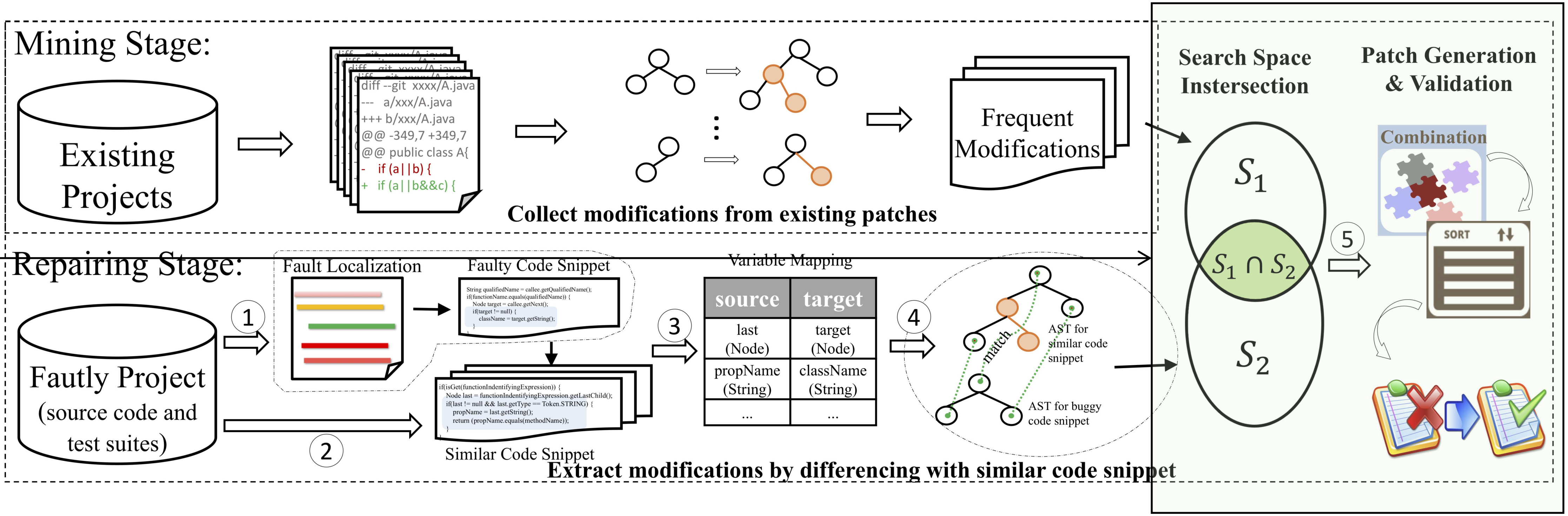
```
String qualifiedName = callee.getQualifiedName();
if(functionName.equals(qualifiedName)) {
  Node target = callee.getNext();
  if(target != null) {
    className = target.getString();
  }
}
```

```
if(isGet(functionIndentifyingExpression)) {
  Node last = functionIndentifyingExpression.getLastChild();
  if(last != null && last.getType == Token.STRING) {
    propName = last.getString();
    return (propName.equals(methodName));
  }
}
```

source	target
last (Node)	target (Node)
propName (String)	className (String)
...	...

- **Fault localization** involves ranking candidate faulty statements and extracting their code snippets.
- **Identifying and evaluating similarities between the faulty snippet and candidates** using three metrics, producing a list ordered by similarity.
- Variable mapping **establishes relationships between variables in faulty and similar snippets** via a mapping table.

Idea of SimFix


Mining Stage:
Existing Projects
Collect modifications from existing patches
Frequent Modifications
Repairing Stage:
Fautly Project (source code and test suites)
①
②
Fault Localization
Faulty Code Snippet
Similar Code Snippet
③
Variable Mapping
④
match
AST for similar code snippet
AST for buggy code snippet
Extract modifications by differencing with similar code snippet
Search Space Intersection
S_1
$S_1 \cap S_2$
S_2
⑤
Patch Generation & Validation
Combination
SORT

```
diff --git  xxxx/A.java
---  a/xxx/A.java
+++ b/xxx/A.java
@@ -349,7 +349,7
@@ public class A{
-    if (a||b) {
+    if (a||b&&c) {
```

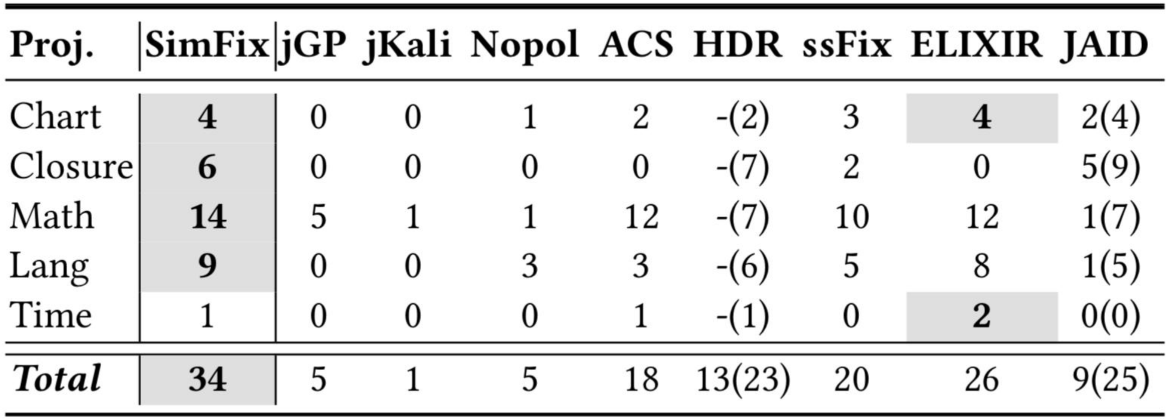
```
String qualifiedName = callee.getQualifiedName();
if(functionName.equals(qualifiedName)) {
  Node target = callee.getNext();
  if(target != null) {
    className = target.getString();
  }
}
```

```
if(isGet(functionIndentifyingExpression)) {
  Node last = functionIndentifyingExpression.getLastChild();
  if(last != null && last.getType == Token.STRING) {
    propName = last.getString();
    return (propName.equals(methodName));
  }
}
```

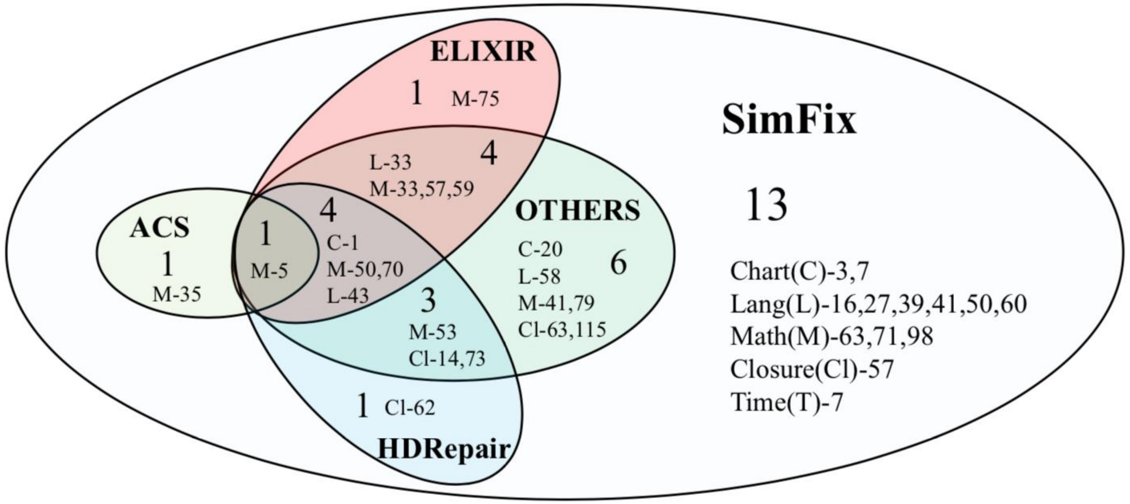
source	target
last (Node)	target (Node)
propName (String)	className (String)
...	...

Modification Extraction & Patch Generation & Validation : generate repair patches by applying extracted code modifications to the faulty code snippet with combining and ranking whose modifications, then using the test suite to validate the correctness of candidate patches until a correct patch found or timeout.

The performence of SimFix



Proj.	SimFix	jGP	jKali	Nopol	ACS	HDR	ssFix	ELIXIR	JAID
Chart	**4**	0	0	1	2	-(2)	3	**4**	2(4)
Closure	**6**	0	0	0	0	-(7)	2	0	5(9)
Math	**14**	5	1	1	12	-(7)	10	12	1(7)
Lang	**9**	0	0	3	3	-(6)	5	8	1(5)
Time	1	0	0	0	1	-(1)	0	**2**	0(0)
Total	**34**	5	1	5	18	13(23)	20	26	9(25)

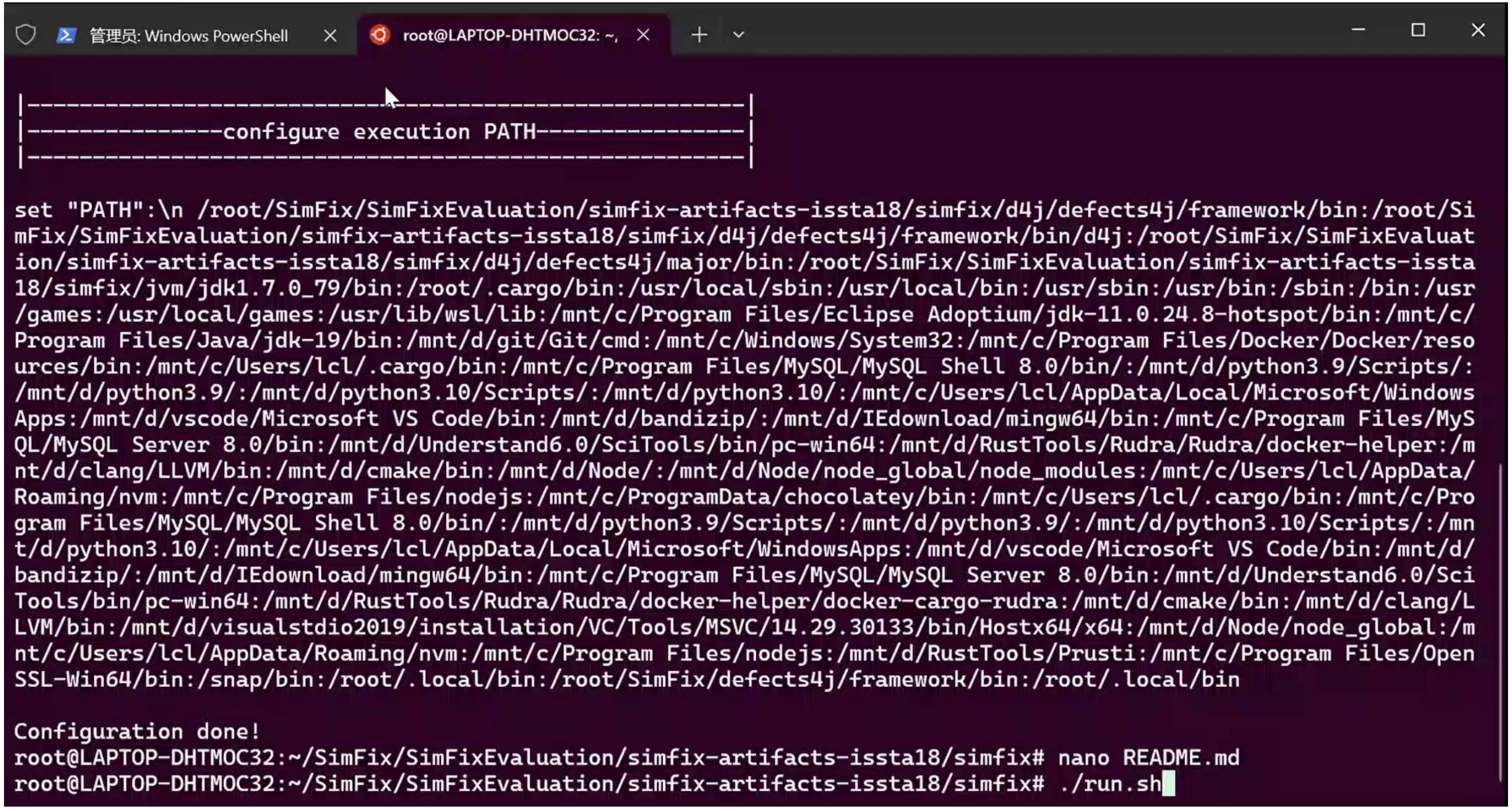

ELIXIR
1 M-75
L-33
M-33,57,59
4
ACS
1
M-35
1
M-5
4
C-1
M-50,70
L-43
3
M-53
Cl-14,73
OTHERS
C-20
L-58
M-41,79
Cl-63,115
6
1 Cl-62
HDRepair
SimFix
13
Chart(C)-3,7
Lang(L)-16,27,39,41,50,60
Math(M)-63,71,98
Closure(Cl)-57
Time(T)-7

Totally, SimFix successfully repair 34 bugs among 357 bugs in Defects4J v1.0 benchmark, better than other tools.

Limitation：

1. Time-consuming.
2. Limited usefulness.
3. Almost rely on other bug-fixing experience and the accurary of similarity calculation.

Show the tool demo with a example.



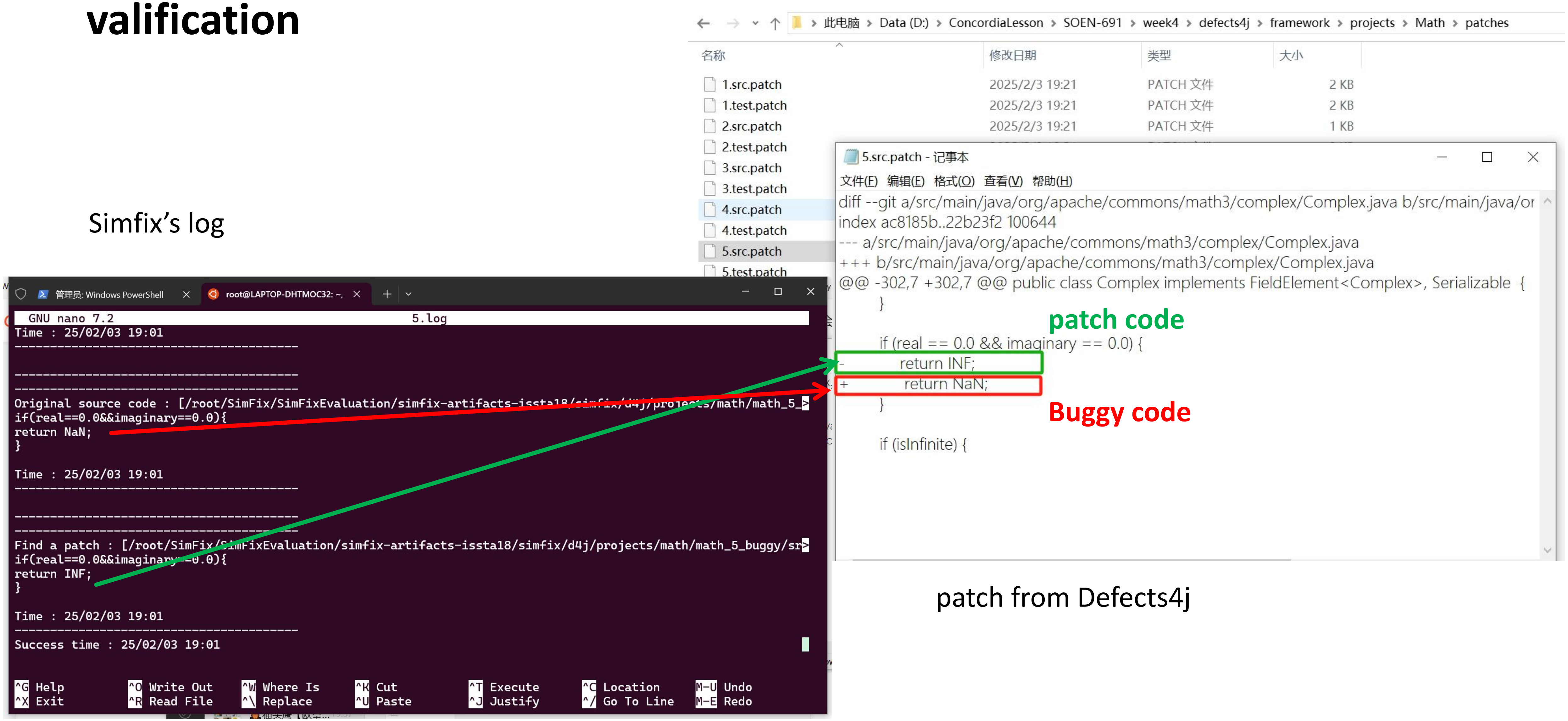
```
|-------------------------------------------------------|
|--------------configure execution PATH----------------|
|-------------------------------------------------------|

set "PATH":\n /root/SimFix/SimFixEvaluation/simfix-artifacts-issta18/simfix/d4j/defects4j/framework/bin:/root/Si
mFix/SimFixEvaluation/simfix-artifacts-issta18/simfix/d4j/defects4j/framework/bin/d4j:/root/SimFix/SimFixEvaluat
ion/simfix-artifacts-issta18/simfix/d4j/defects4j/major/bin:/root/SimFix/SimFixEvaluation/simfix-artifacts-issta
18/simfix/jvm/jdk1.7.0_79/bin:/root/.cargo/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr
/games:/usr/local/games:/usr/lib/wsl/lib:/mnt/c/Program Files/Eclipse Adoptium/jdk-11.0.24.8-hotspot/bin:/mnt/c/
Program Files/Java/jdk-19/bin:/mnt/d/git/Git/cmd:/mnt/c/Windows/System32:/mnt/c/Program Files/Docker/Docker/reso
urces/bin:/mnt/c/Users/lcl/.cargo/bin:/mnt/c/Program Files/MySQL/MySQL Shell 8.0/bin/:/mnt/d/python3.9/Scripts/:
/mnt/d/python3.9/:/mnt/d/python3.10/Scripts/:/mnt/d/python3.10/:/mnt/c/Users/lcl/AppData/Local/Microsoft/Windows
Apps:/mnt/d/vscode/Microsoft VS Code/bin:/mnt/d/bandizip/:/mnt/d/IEdownload/mingw64/bin:/mnt/c/Program Files/MyS
QL/MySQL Server 8.0/bin:/mnt/d/Understand6.0/SciTools/bin/pc-win64:/mnt/d/RustTools/Rudra/Rudra/docker-helper:/m
nt/d/clang/LLVM/bin:/mnt/d/cmake/bin:/mnt/d/Node/:/mnt/d/Node/node_global/node_modules:/mnt/c/Users/lcl/AppData/
Roaming/nvm:/mnt/c/Program Files/nodejs:/mnt/c/ProgramData/chocolatey/bin:/mnt/c/Users/lcl/.cargo/bin:/mnt/c/Pro
gram Files/MySQL/MySQL Shell 8.0/bin/:/mnt/d/python3.9/Scripts/:/mnt/d/python3.9/:/mnt/d/python3.10/Scripts/:/mn
t/d/python3.10/:/mnt/c/Users/lcl/AppData/Local/Microsoft/WindowsApps:/mnt/d/vscode/Microsoft VS Code/bin:/mnt/d/
bandizip/:/mnt/d/IEdownload/mingw64/bin:/mnt/c/Program Files/MySQL/MySQL Server 8.0/bin:/mnt/d/Understand6.0/Sci
Tools/bin/pc-win64:/mnt/d/RustTools/Rudra/Rudra/docker-helper/docker-cargo-rudra:/mnt/d/cmake/bin:/mnt/d/clang/L
LVM/bin:/mnt/d/visualstdio2019/installation/VC/Tools/MSVC/14.29.30133/bin/Hostx64/x64:/mnt/d/Node/node_global:/m
nt/c/Users/lcl/AppData/Roaming/nvm:/mnt/c/Program Files/nodejs:/mnt/d/RustTools/Prusti:/mnt/c/Program Files/Open
SSL-Win64/bin:/snap/bin:/root/.local/bin:/root/SimFix/defects4j/framework/bin:/root/.local/bin

Configuration done!
root@LAPTOP-DHTMOC32:~/SimFix/SimFixEvaluation/simfix-artifacts-issta18/simfix# nano README.md
root@LAPTOP-DHTMOC32:~/SimFix/SimFixEvaluation/simfix-artifacts-issta18/simfix# ./run.sh
```

valification

Simfix’s log



```
  GNU nano 7.2                                  5.log
Time : 25/02/03 19:01
----------------------------------------


----------------------------------------
----------------------------------------
Original source code : [/root/SimFix/SimFixEvaluation/simfix-artifacts-issta18/simfix/d4j/projects/math/math_5_>
if(real==0.0&&imaginary==0.0){
return NaN;
}

Time : 25/02/03 19:01
----------------------------------------


----------------------------------------
----------------------------------------
Find a patch : [/root/SimFix/SimFixEvaluation/simfix-artifacts-issta18/simfix/d4j/projects/math/math_5_buggy/sr>
if(real==0.0&&imaginary==0.0){
return INF;
}

Time : 25/02/03 19:01
----------------------------------------
Success time : 25/02/03 19:01

^G Help       ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo
^X Exit       ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line M-E Redo
```

管理员: Windows PowerShell
root@LAPTOP-DHTMOC32: ~,
此电脑 › Data (D:) › ConcordiaLesson › SOEN-691 › week4 › defects4j › framework › projects › Math › patches
5.src.patch - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

名称	修改日期	类型	大小
1.src.patch	2025/2/3 19:21	PATCH 文件	2 KB
1.test.patch	2025/2/3 19:21	PATCH 文件	2 KB
2.src.patch	2025/2/3 19:21	PATCH 文件	1 KB
2.test.patch			
3.src.patch			
3.test.patch			
4.src.patch			
4.test.patch			
5.src.patch			
5.test.patch			

```
diff --git a/src/main/java/org/apache/commons/math3/complex/Complex.java b/src/main/java/or
index ac8185b..22b23f2 100644
--- a/src/main/java/org/apache/commons/math3/complex/Complex.java
+++ b/src/main/java/org/apache/commons/math3/complex/Complex.java
@@ -302,7 +302,7 @@ public class Complex implements FieldElement<Complex>, Serializable {
         }

         if (real == 0.0 && imaginary == 0.0) {
-            return INF;
+            return NaN;
         }

         if (isInfinite) {
```

patch code

Buggy code

patch from Defects4j

Thanks