ISCAS 中国科学院软件研究所学术年会’2022
暨计算机科学国家重点实验室开放周

学术论文

Find Bugs in Static Bug Finders
基于差分测试的静态分析工具的缺陷检测

Junjie Wang, Yuchao Huang, Song Wang, Qing Wang

In 30th IEEE/ACM International Conference on Program Comprehension (ICPC 2022)

ACM Distinguished Paper Award

联系人：王俊杰，黄芋超，王青　　联系方式：{junjie, yuchao2019, wq}@iscas.ac.cn

Github: *https://github.com/wuchiuwong/Diff-Testing-01*

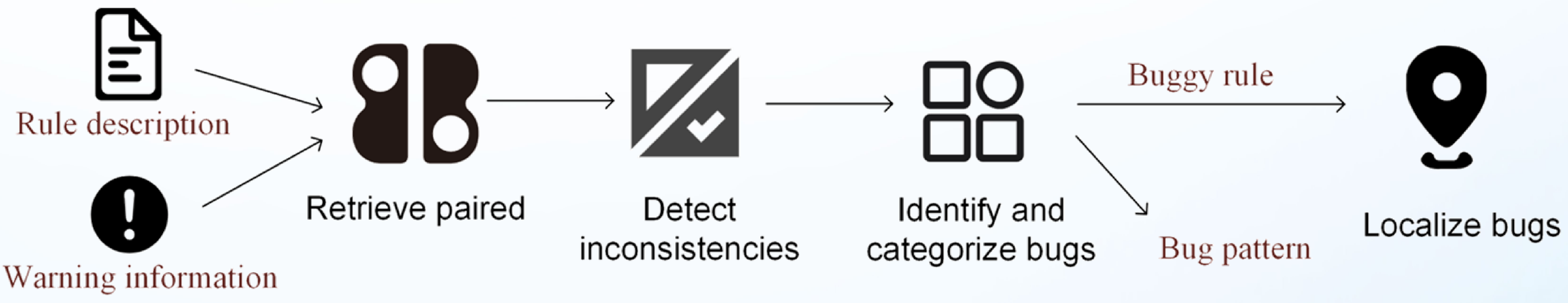
Background

- **Static Bug Finders (code analyzer)**
  - Leverage predefined heuristic static analysis rules
  - Scan source/binary code
  - Report violations as warnings

- **Researches on Static Bug Finders**
  - Filter out false positives reported
  - Design new static analysis rules
  - Less attention on incorrectness of existing rules

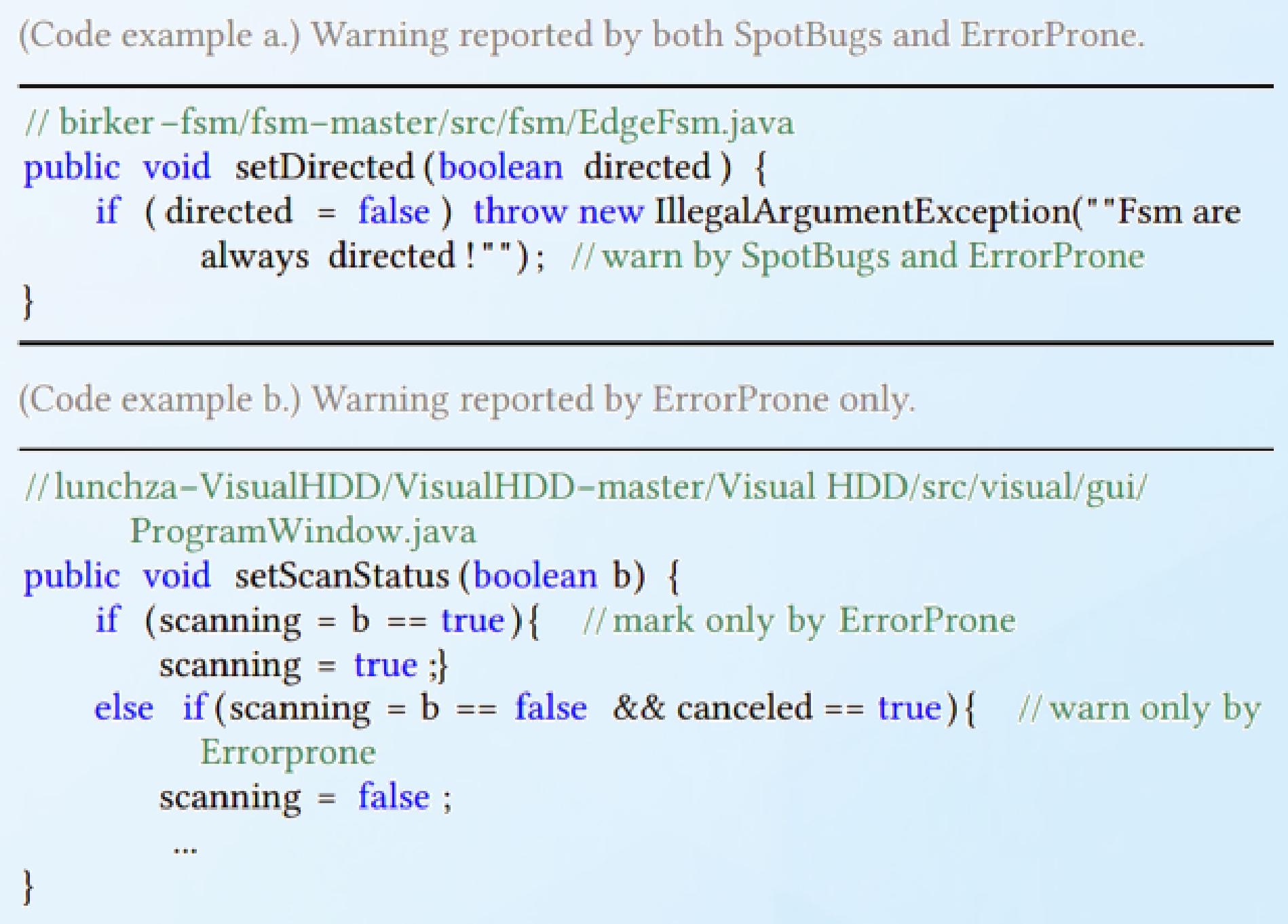
First Work targeting at Correctness of Static Bug Finders

- Differential testing approach to detect bugs in rules of four widely-used static bug finders
- A qualitative study about the bugs found
- A heuristic-based rule mapping method which combines the similarity in rules’ description and the overlap in warning information reported by the tools, to retrieve paired rules across static bug finders for differential testing
- 46 reported bugs in the static bug finders, among which 30 are fixed or confirmed



- **Retrieving Paired Rules**
  - Description Similarity: Term similarity, Semantic similarity, Code similarity
  - Filtering out less possible rule pairs based on warning information
  - Conduct a manual check to determine the paired rules
- **Detecting Inconsistencies**
  - Run on 2,728 open source Java projects
  - 73% rules are triggered
  - The median number of trigger times is 163

- **Identifying and Categorizing Bugs**
  - Manually check the detected inconsistencies and identifie the bugs
  - Examine the source code of the static bug finders to localize the bugs


(Code example a.) Warning reported by both SpotBugs and ErrorProne.

```
// birker-fsm/fsm-master/src/fsm/EdgeFsm.java
public void setDirected(boolean directed) {
    if (directed = false) throw new IllegalArgumentException(""Fsm are
            always directed!""); //warn by SpotBugs and ErrorProne
}
```

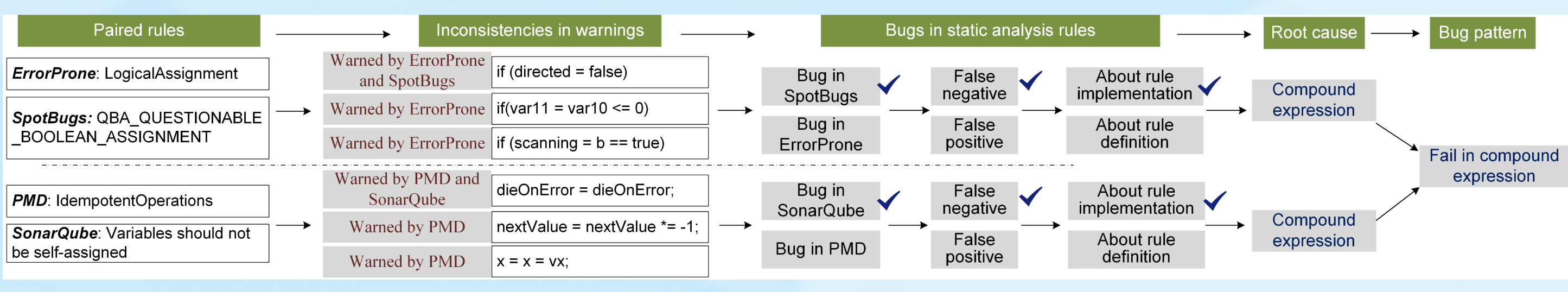
(Code example b.) Warning reported by ErrorProne only.

```
//lunchza-VisualHDD/VisualHDD-master/Visual HDD/src/visual/gui/
        ProgramWindow.java
public void setScanStatus(boolean b) {
    if (scanning = b == true){    //mark only by ErrorProne
        scanning = true;}
    else if(scanning = b == false && canceled == true){    //warn only by
            Errorprone
        scanning = false;
        ...
}
```

Results and Analysis

- **Paired rules**
  - 74 rule pairs from SonarQube and PMD, and 30 rule pairs from SpotBugs and ErrorProne are finally determined as having identical functionality
- **13 bug patterns**
  - 13 bug patterns based on bugs’ context and root causes, serve as checklist when designing and implementing rules
- **46 detected bugs**
  - 46 detected bugs about the implementation or design of static analysis rules, among which 30 are fixed/confirmed

Tool	False negative (about rule implementation)	False negative (about rule definition)	False Positive	Overall
SonarQube	6	3	1	10
PMD	15	4	6	25
SpotBugs	4	1	1	6
ErrorProne	4	1	0	5
Overall	29	9	8	46


Paired rules
Inconsistencies in warnings
Bugs in static analysis rules
Root cause
Bug pattern
ErrorProne: LogicalAssignment
SpotBugs: QBA_QUESTIONABLE_BOOLEAN_ASSIGNMENT
Warned by ErrorProne and SpotBugs
if (directed = false)
Warned by ErrorProne
if(var11 = var10 <= 0)
Warned by ErrorProne
if (scanning = b == true)
Bug in SpotBugs ✓
Bug in ErrorProne
False negative ✓
False positive
About rule implementation ✓
About rule definition
Compound expression
Fail in compound expression
PMD: IdempotentOperations
SonarQube: Variables should not be self-assigned
Warned by PMD and SonarQube
dieOnError = dieOnError;
Warned by PMD
nextValue = nextValue *= -1;
Warned by PMD
x = x = vx;
Bug in SonarQube ✓
Bug in PMD
False negative ✓
False positive
About rule implementation ✓
About rule definition
Compound expression