A PRECISE FRAMEWORK FOR SOURCE-LEVEL CONTROL-FLOW ANALYSIS

IDRISS RIOUAK, CHRISTOPH REICHENBACH, GÖREL HEDIN, AND NIKLAS FORS Lund University, Sweden

MOTIVATIONS

Static program analysis plays a fundamental role in software development and helps developers to detect subtle bugs such as null pointer exceptions or security vulnerabilities.

In this poster, we present IntraJ: an instance of the *IntraCFG* framework that constructs precise intraprocedural control-flow graphs.

EXPERIMENTS

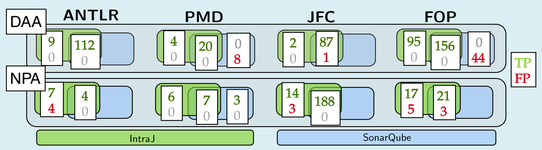
We compared the precision and the performance of IntraJ against SonarQube by implementing two dataflow analyses:

- Dead Assignment Analysis (*DAA*)
- Null Pointer Analysis (*NPA*)

Our benchmarks:

ANTLR	PMD	JFC	FOP
33K LOC	49K LOC	95K LOC	97K LOC

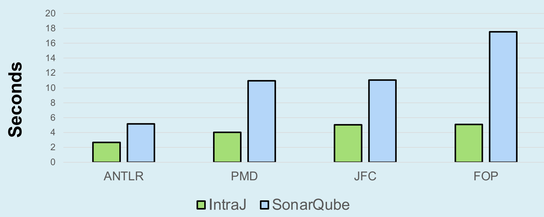
PRECISION



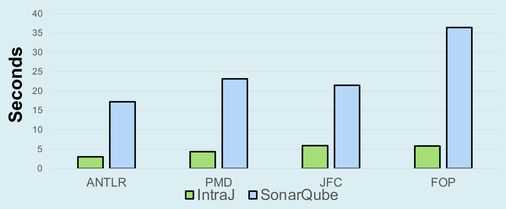
QUICK FACT:
PRECISE ANALYSES RESULTS

PERFORMANCE

Dead Assignment Analysis



Null-Pointer Analysis



QUICK FACT:
OVERALL FASTER THAN SONARQUBE

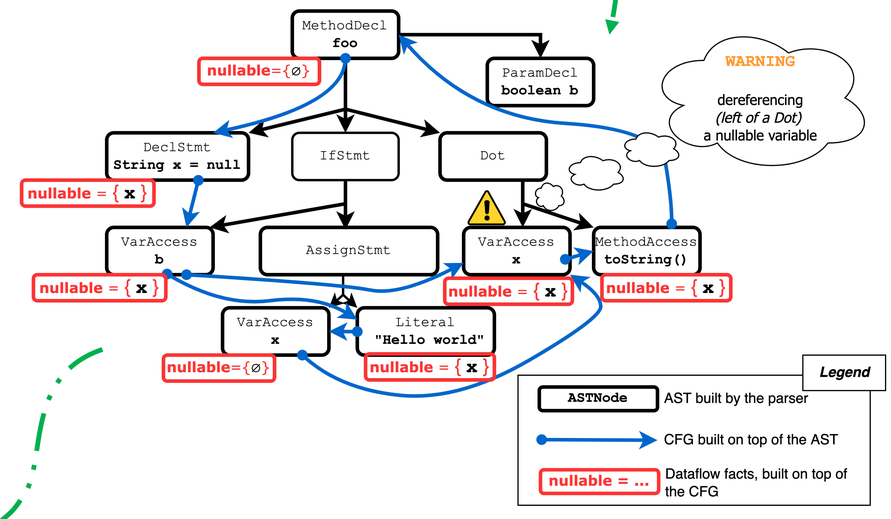
ADVANCED BUG DETECTION IN YOUR IDE

```
1 void foo(boolean b){
2   String x = null;
3   if (b) x = "Hello world!";
4   x.toString();
5 }
```

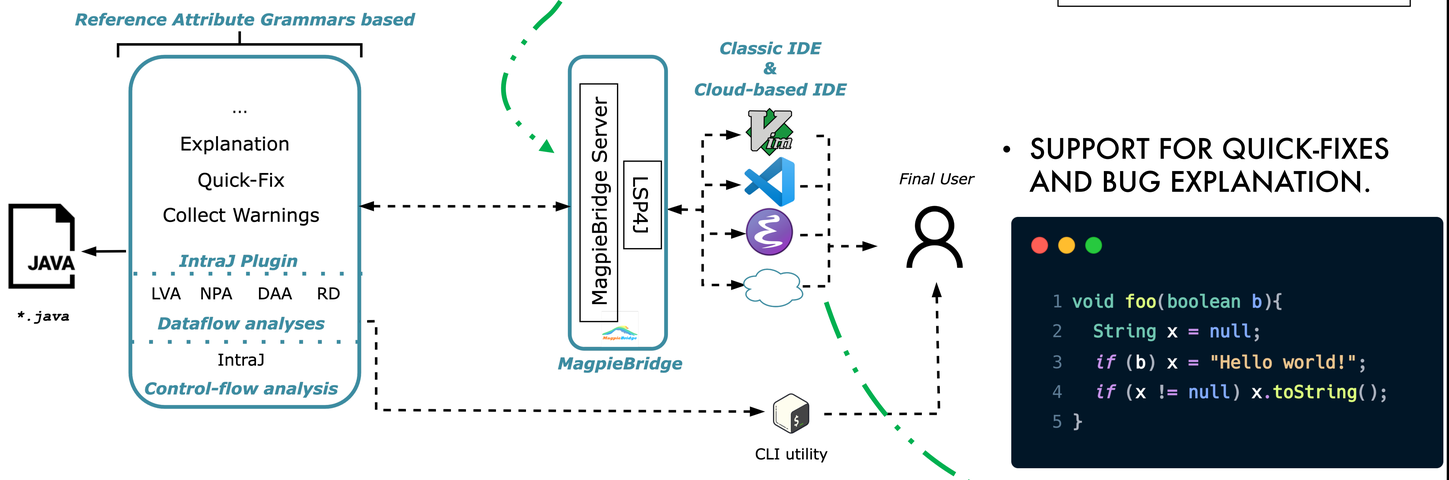
WARNING

A 'NullPointerException' could be thrown; 'x' is nullable here.

- ADVANCED BUG DETECTION REQUIRES CONTROL-FLOW AND DATAFLOW ANALYSES
- FAST ANALYSES ARE REQUIRED FOR IDE INTEGRATION



- THE CONTROL-FLOW GRAPH IS CONSTRUCTED ON TOP OF THE ABSTRACT SYNTAX TREE
- ONLY RELEVANT NODES ARE INCLUDED IN THE ANALYSIS



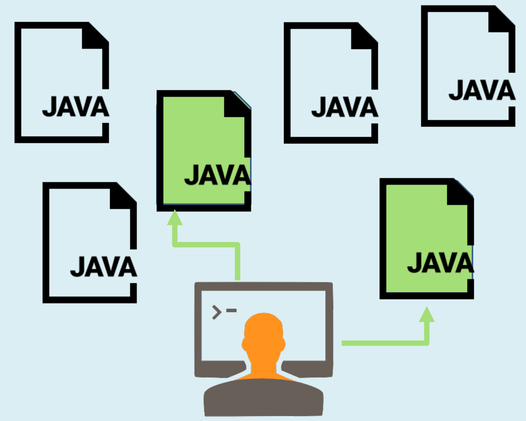
- SUPPORT FOR QUICK-FIXES AND BUG EXPLANATION.

```
1 void foo(boolean b){
2   String x = null;
3   if (b) x = "Hello world!";
4   if (x != null) x.toString();
5 }
```

WALLENBERG AUTONOMOUS SYSTEMS AND SOFTWARE PROGRAM

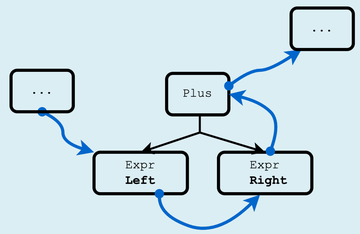
jastadd

ON-DEMAND EVALUATION



No need to analyse the entire project. The analyses are computed only on *open* and *dependent files*.

SPECIFICATION



Fully declarative specification using JastAdd2

- Easy to support new language constructs

```
1 Plus.firstNodes() = getLeft().firstNodes();
2 Plus.getLeft().nextNodes() = getRight().firstNodes();
3 Plus.getRight().nextNodes() = {this};
```

CONCLUSIONS

- HIGH PRECISION
- CONCISE SPECIFICATION
- COMPETITIVE WITH SONARQUBE
- LSP SUPPORT

FUTURE WORK

- INTERPROCEDURAL CFGs
- BETTER SUPPORT FOR QUICK-FIX AND BUG EXPLANATION

VSCode Extension | GitHub Repository | IntraJ Article