A PRECISE FRAMEWORK FOR SOURCE-LEVEL CONTROL-FLOW ANALYSIS

IDRISS RIOUAK[1], CHRISTOPH REICHENBACH[1], GÖREL HEDIN[1] AND NIKLAS FORS[1]
[1]Lund University, Sweden

IDRISS RIOUAK
IDRISS.RIOUAK@CS.LTH.SE
GITHUB.COM/IDRISSRIO
@RIOUAKIDRISS

YOU CAN DETECT BUGS EARLIER IN YOUR DEVELOPMENT CYCLE

MOTIVATIONS

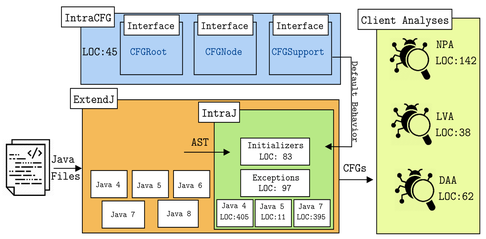
Static program analysis plays a fundamental role in software development and may help developers detect subtle bugs such as null pointer exceptions or security vulnerabilities. In this poster we present IntraCFG, a language-independent framework for constructing precise *intraprocedural* control-flow graphs (CFGs) superimposed on the Abstract Syntax Tree (AST). Source-level dataflow analysis permits easier integration with the IDEs and Cloud tools as the reports can be directly linked to the source code and do not require producing the Intermediate Representation.

OUR METHOD

We build the CFGs on top of the AST using Reference Attribute Grammars (RAGs).

Highlights of our approach:
- Fully declarative specification using *JastAdd2*
- *Handles implicit control flow*
- Heavily exploit *on-demand evaluation*

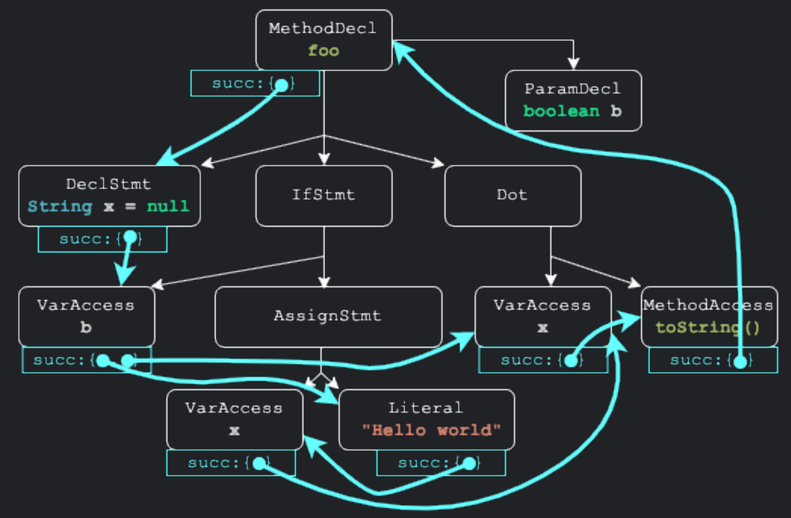
THE FRAMEWORK

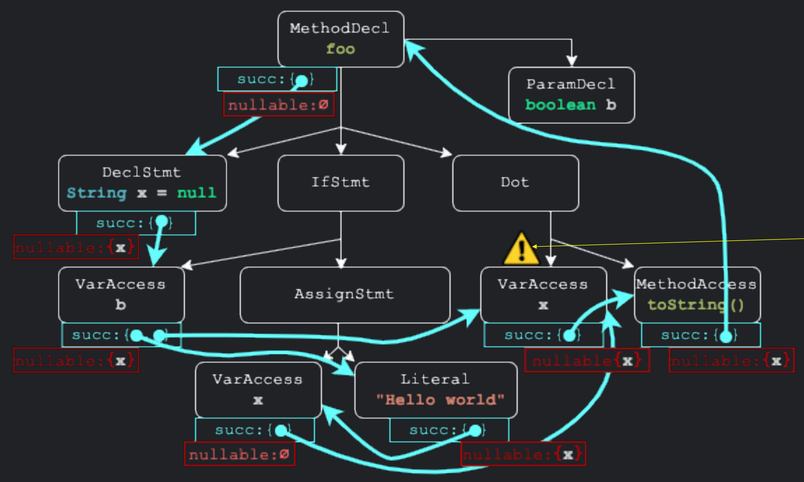


IntraCFG provides client APIs for the successor and predecessor relations, and default behaviour that simplifies the CFG construction for a specific language.

We used IntraCFG to construct precise CFGs for Java 8, extending the ExtendJ Java compiler.

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



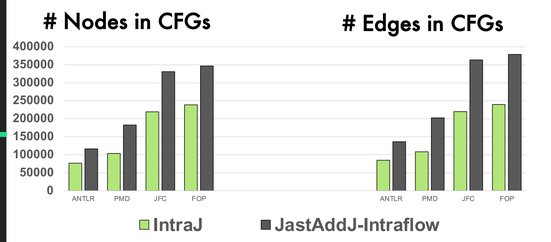


EXPERIMENTS

We compared the results of IntraJ with:
- JastAddJ-Intraflow (JJI): a RAG based framework
- SonarQube: a highly tuned static analyser

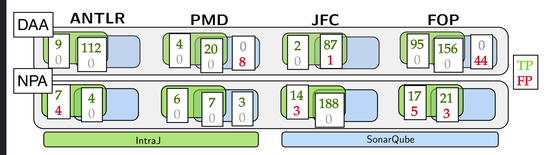
We used as benchmarks:

CFG SIZE REDUCTION W.R.T. JJI

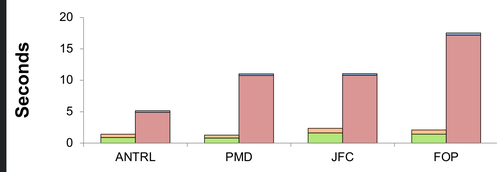


PRECISION AND PERFORMANCE

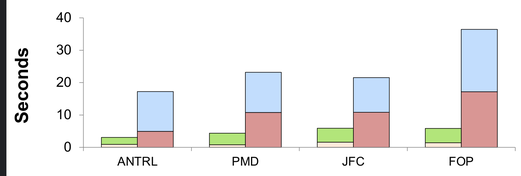
We compared the precision and the performance of IntraJ against SonarQube by implementing two dataflow analyses:
- Dead Assignment Analysis (*DAA*)
- Null Pointer Analysis (*NPA*)



DEAD ASSIGNMENT ANALYSIS



NULL POINTER EXCEPTION ANALYSIS



CONCLUSIONS

- HIGH-PRECISION
- CONCISE SPECIFICATION
- 30% FEWER CFG NODES INVOLVED IN THE ANALYSIS
- COMPETITIVE WITH SONARQUBE

WASP WALLENBERG AI, AUTONOMOUS SYSTEMS AND SOFTWARE PROGRAM

LUND UNIVERSITY