UNDERSTANDING DIFFERENCES AMONG EXECUTIONS WITH VARIATIONAL TRACES

Jens Meinicke, Chu-Pan Wong, Christian Kästner, Gunter Saake

Carnegie Mellon University · isr institute for SOFTWARE RESEARCH · OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG · INF

DIFFERENTIAL ANALYSIS FOR 2^N EXECUTIONS

```
3  int netpollSetup(boolean ipv4, boolean flag) {
4    Integer err = null;
5    if (flag) {
6      err = -1;
7    }
8    return getErrValue(err, ipv4);
9  }
10 int getErrValue(Integer err, boolean ipv4) {
11   if (ipv4) {
12     return err;
13   }
14   return 1;
15 }
```

Can you find the bug?

Which inputs cause the fault?

Why do the inputs cause the fault?

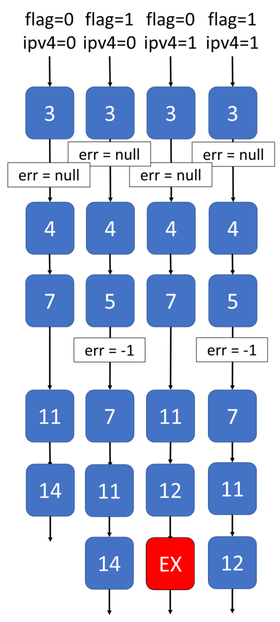
How do inputs interact?

VAREXJ: GENERATING VARIATIONAL TRACES USING VARIATIONAL EXECUTION

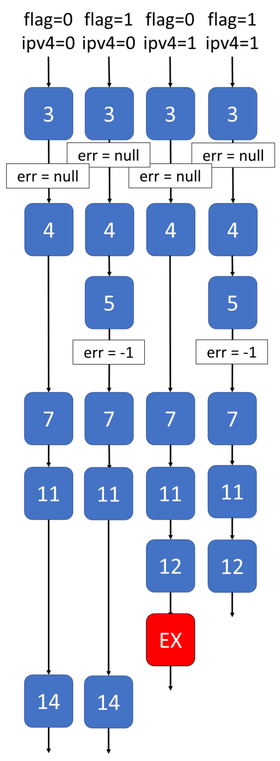
https://github.com/meinicke/varexj

Basic Steps for Differential Analysis

Record state and control-flow changes

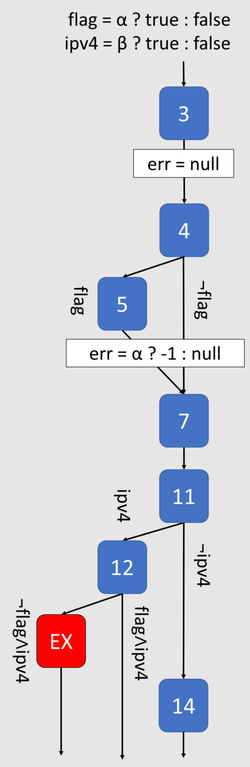


Align traces



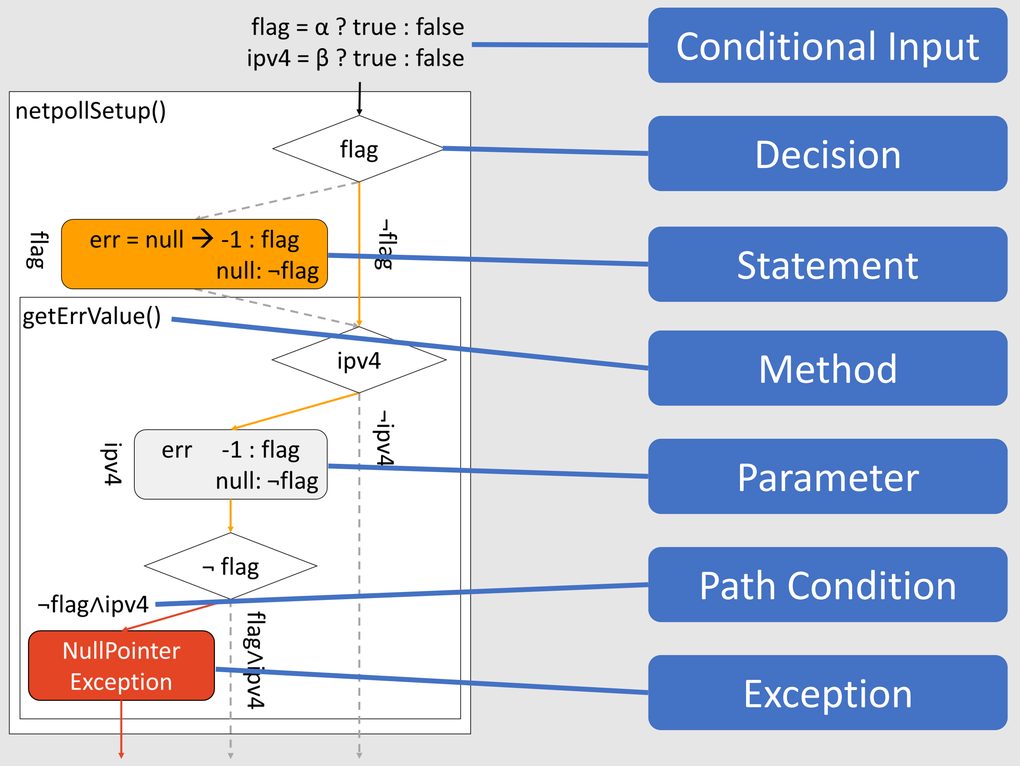
Differential Analysis with Variational Execution

Merge traces into single trace with conditions



Shared & Synchronized execution

Reduce traces by removing irrelevant statements

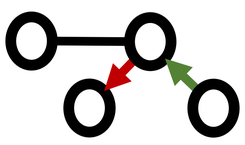


On-the-fly data reduction

FURTHER USE CASES

INFORMATION FLOW

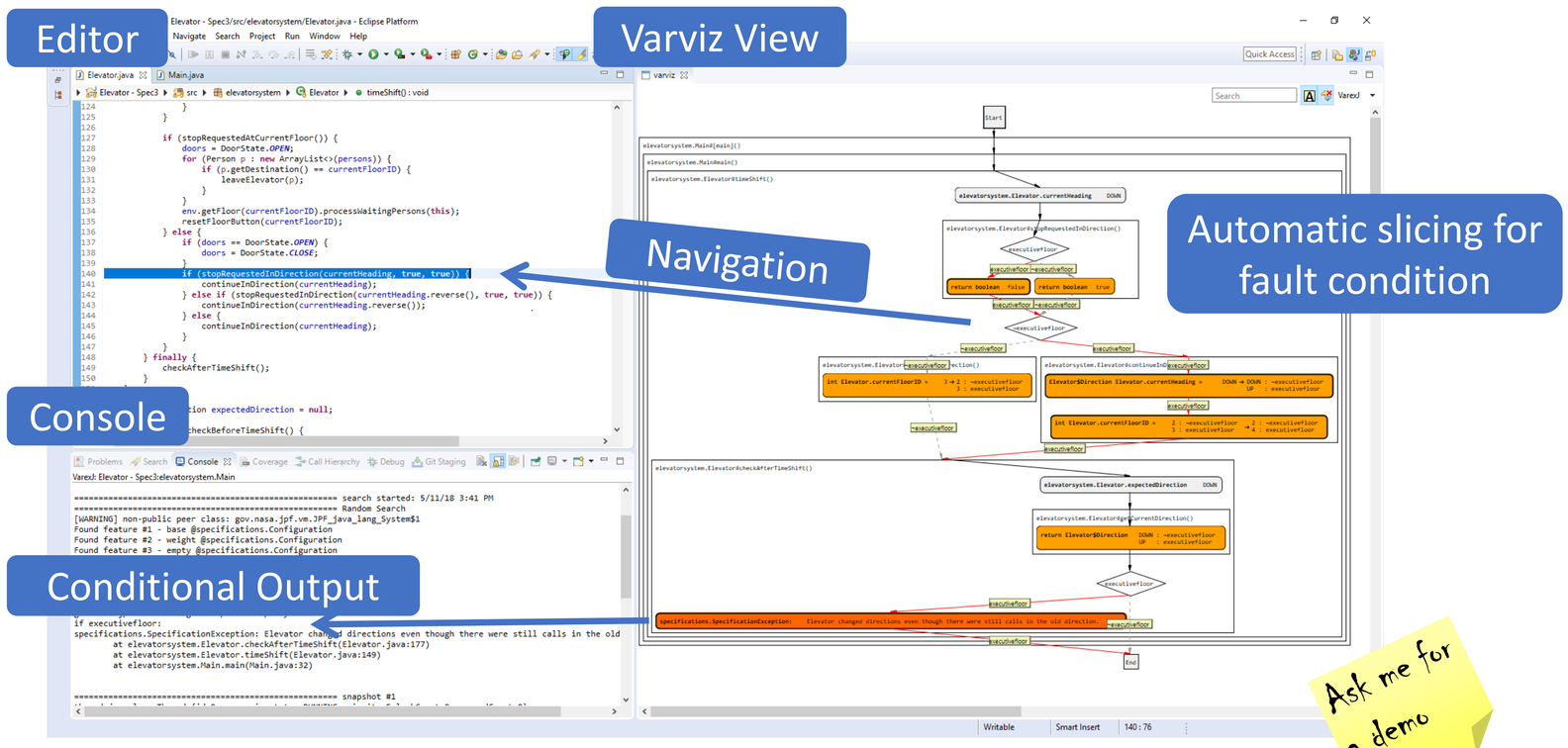
TRACKING LOAD-TIME OPTIONS



UNDERSTANDING FEATURE INTERACTIONS

VARVIZ: VARIATIONAL DEBUGGING

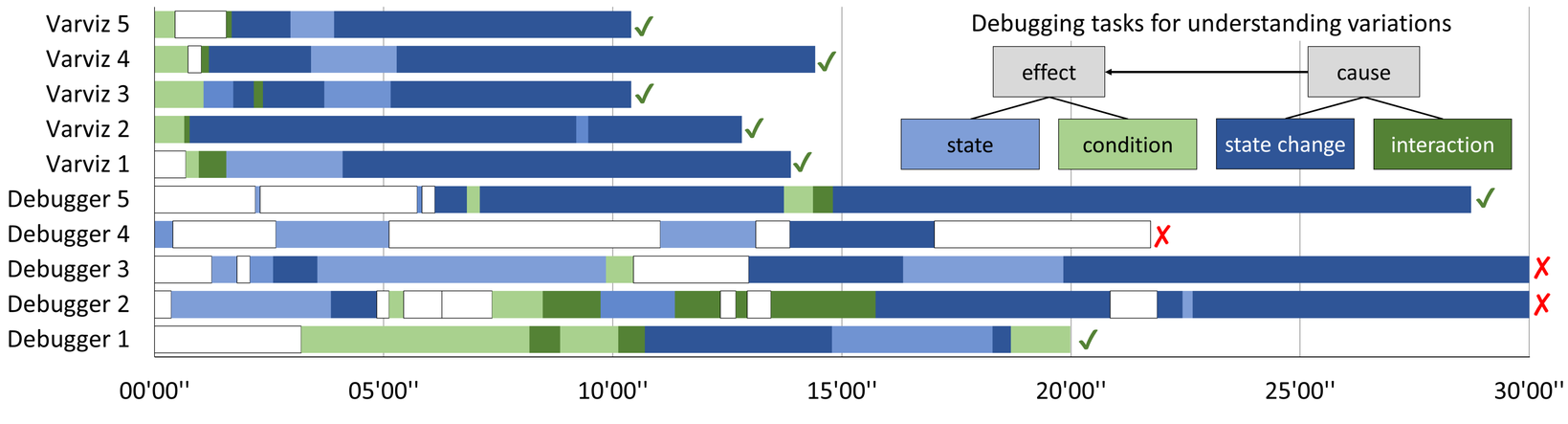
https://github.com/meinicke/varviz



HUMAN SUBJECT STUDY

ELEVATOR (6 OPTIONS, 20 CONFIGURATIONS, 259 LOC)

TIME SPEND ON DEBUGGING TASKS



EXPLORATION OF THE SOURCE CODE