On the **Probabilistic** Symbolic Analysis of Programs

Antonio Filieri, Corina S. Pasareanu

Mateus Borges, Marcelo d’Amorin, Matt Dwyer, Jaco Geldenhuys, Kasper Luckow, Willem Visser
public class OnBoardAbordExecutive {

    public void checkSafety(int pressure, int altitude, int spinSpeed){
        int discountedPressure = pressure - altitude/2;
        ...

        if(discountedPressure > 80 && spinSpeed>72){
            abort();
        }

        ...
        return;
    }

    ...
}
Sensors → Software

Commands →

Probabilistic Usage Profile + Abstract Execution → Probability of
Probabilistic Usage Profile

Arbitrarily accurate discretization
Probabilistic Usage Profile + Abstract Execution → Probability of √ ❌
Abstract Execution

More precisely **Symbolic Execution**

discountedPressure <= 80

spinSpeed <= 72

PC: discountedPressure > 80 && spinSpeed <= 72

discountedPressure > 80

spinSpeed > 72
Probabilistic Usage Profile + Abstract Execution

Pr( PCs | UP )

…and the confidence on such result
Initial contribution [ICSE 2013]:

• General **white-box** methodology for **finite domains** using **integer model counting**, with explicit measure of **confidence**

• Handling linear integer constraints with **polytopes analysis** and our **divide and conquer** strategy

• Bounded execution for **loops and recursion**, **multithreading**

• Based on **Korat** for data structures
In the last year

- Dealing with floating-point numbers and non-linear constraints [PLDI 2014]
- Approximate incremental analysis [FSE2014]
- Synthesis of optimal schedulers for multithreading [ASE2014?]
- Improved support for data structures
- Parallelization
The boiling pot

- Nondeterminism
- Strings
- Dynamic discretization of continuous CDF
- Distribution-aware statistical methods
- Probabilistic loop invariants
- “Usage-coverage” criteria
- Errors and bug ranking (prioritization)
- Usage profile inference
- Automatic data-structures and code selection
- Quantitative information flow analysis [SPIN2014?]