Collaborators/funders:
Systems and Software Security / FM Research Group
ARM Centre of Excellence
PPGEE, PPGI – UFAM
Centre for Digital Trust and Society
UKRI, EPSRC, EU Horizon and industrial partners

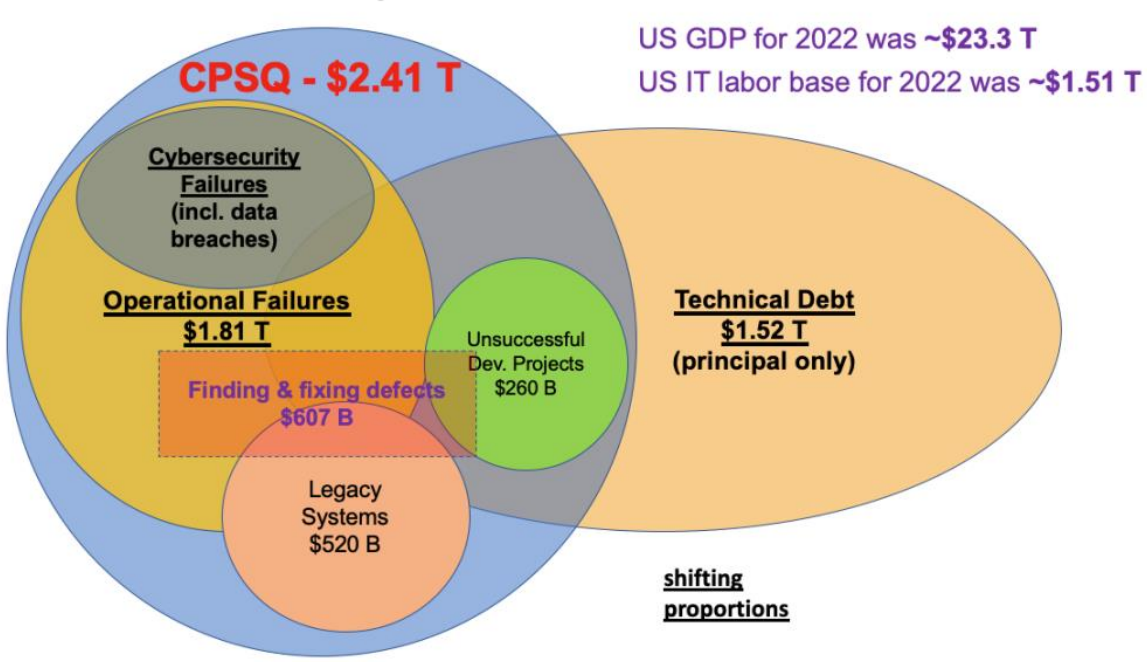
The University of Manchester

Workshop on Automated Software Verification, Testing and Repair

Lucas Cordeiro
lucas.cordeiro@manchester.ac.uk
https://ssvlab.github.io/lucasccordeiro/

How much could software errors cost your business?

Poor software quality cost US companies **$2.41 trillion in 2022**, while the **accumulated software Technical Debt** (TD) has grown to **~$1.52 trillion**



TD relies on temporary easy-to-implement solutions to achieve short-term results at the expense of efficiency in the long run

The cost of poor software quality in the US: A 2022 Report

CISQ
Consortium for Information & Software Quality™

Objective of this workshop

Discuss **automated testing, verification, and repair** techniques to establish a **robust foundation for building secure software systems**

- Introduce a **logic-based automated reasoning platform** to find and repair **software vulnerabilities**
- Explain **testing, verification, and repair** techniques to build **secure software systems**
- Present recent advancements towards a **hybrid approach** to protecting against **memory safety and concurrency vulnerabilities**

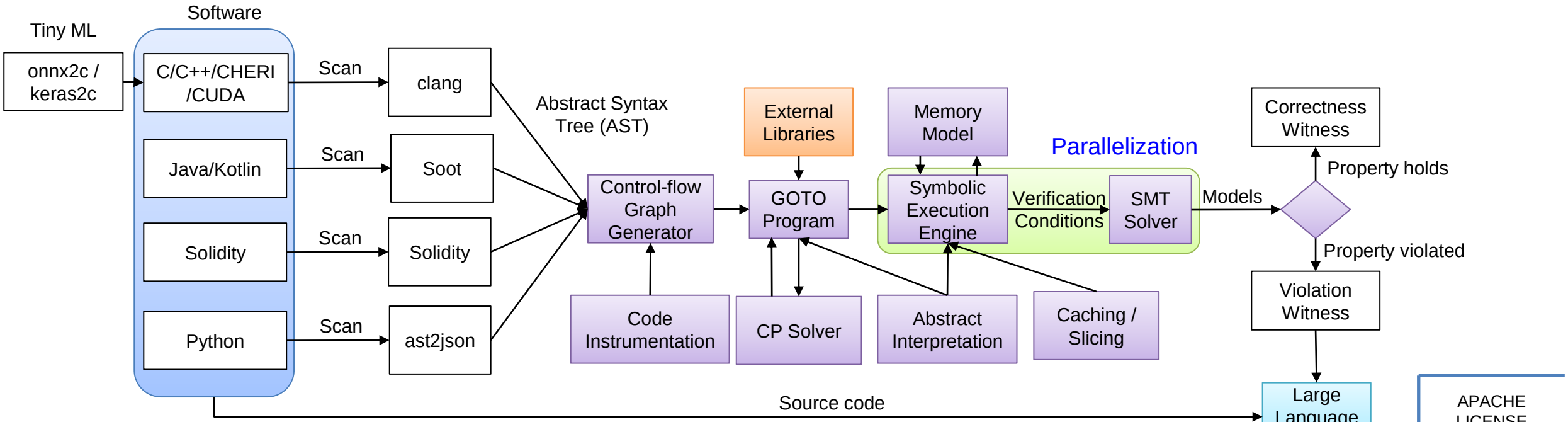
Research Questions

Given a **program** and a **safety/security specification**, can we automatically **verify** that the **program performs as specified**?

Can we leverage **program analysis/synthesis** to **discover and fix** more **software vulnerabilities** than existing state-of-the-art approaches?

ESBMC: An Automated Reasoning Platform

Logic-based automated reasoning for checking the **safety** and **security** of **AI and software systems**

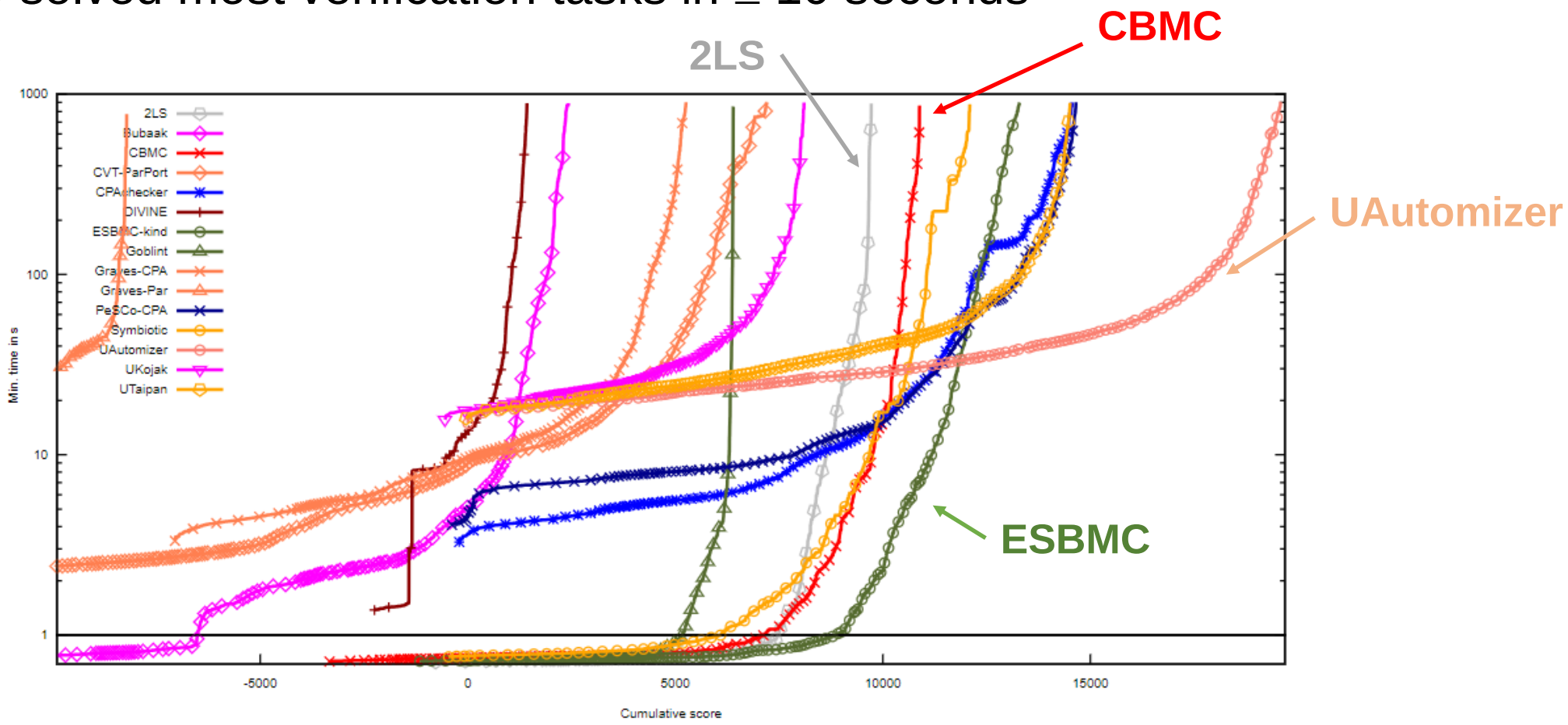


Combines BMC, *k*-induction, abstract interpretation, CP/SMT solving towards correctness proof and bug hunting

www.esbmc.org

Intl. Software Verification Competition (SV-Comp 2023)

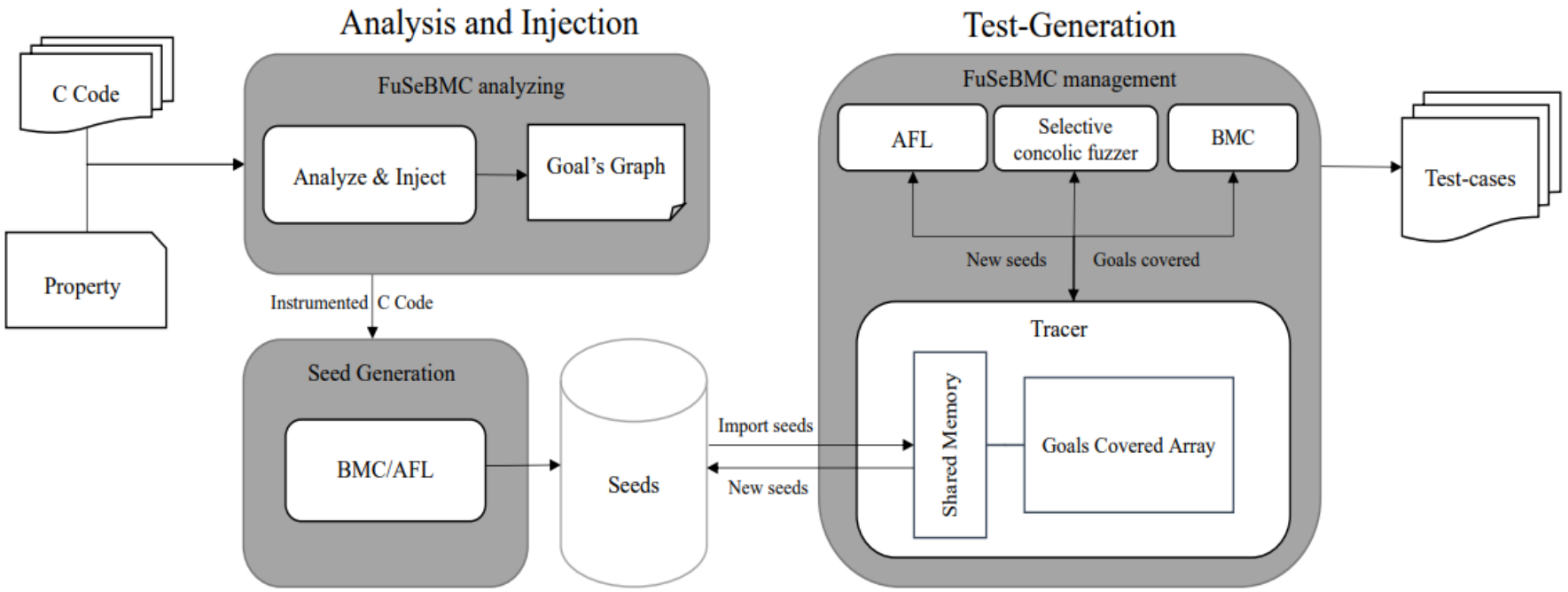
- SV-COMP 2023, 23805 verification tasks, max. score: 38644
- ESBMC solved most verification tasks in ≤ 10 seconds



Verification of the `Overall` Category

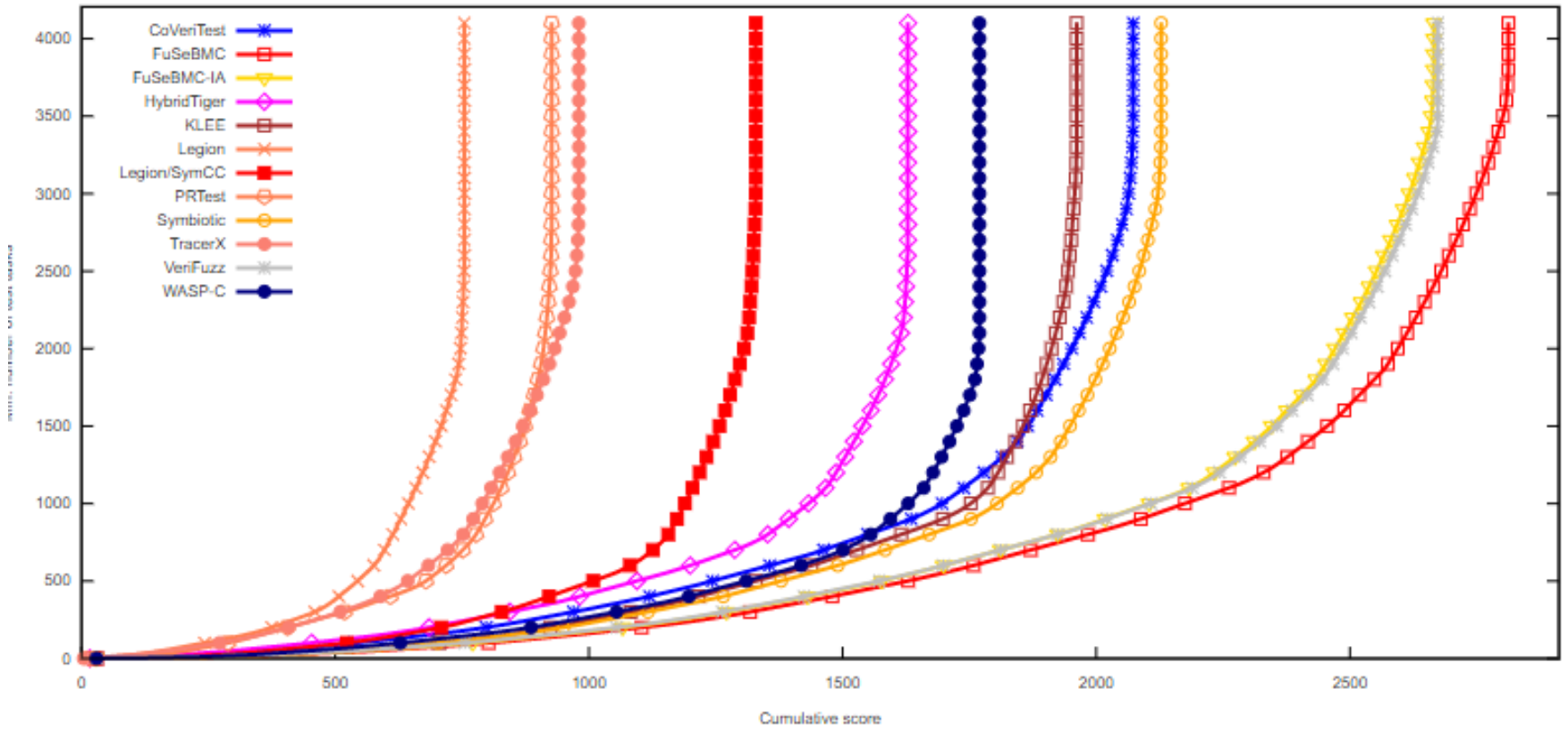
FuSeBMC v4 Framework

- Use **Clang** tooling infrastructure
- Employ three engines in its **reachability analysis**: **one BMC and two fuzzing engines**
- Use a **tracer** to coordinate the various engines



Alshmrany, K., Aldughaim, M., Bhayat, A., Cordeiro, L.: FuSeBMC v4: Smart Seed Generation for Hybrid Fuzzing - (Competition Contribution). FASE 2022: 336-340

Competition on Software Testing 2023: Results of the `Overall` Category



FuSeBMC achieved 3 awards: 1st place in `Cover-Error`, 1st place in `Cover-Branches`, and 1st place in `Overall`

Alshmrany, K., Aldughaim, M., Bhayat, A., Cordeiro, L.: FuSeBMC v4: Smart Seed Generation for Hybrid Fuzzing - (Competition Contribution). FASE 2022: 336-340

https://test-comp.sosy-lab.org/2023/

The Bitter Lesson by Rich Sutton
March 13, 2019

“The biggest lesson that can be read from 70 years of AI research is that general methods that leverage computation are ultimately the most effective, and by a large margin. The ultimate reason for this is Moore's law, or rather its generalization of continued exponentially falling cost per unit of computation”

“The two methods that seem to scale arbitrarily in this way are *search* and *learning*”

Impact: Awards and Industrial Deployment

- **Distinguished Paper Award** at ICSE'11
- **Best Paper Award** at SBESC'15
- **Most Influential Paper Award** at ASE'23
- **Best Tool Paper Award** at SBSeg'23
- **29 awards** from the international competitions on software verification (SV-COMP) and testing (Test-Comp) 2012-2023 at **TACAS/FASE**
  - Bug Finding and Code Coverage 
- **Intel** deploys **ESBMC** in production as one of its verification engines for **verifying firmware in C**
- **Nokia and ARM** have found **security vulnerabilities** in **C/C++ software**
- **Funded by government** (EPSRC, British Council, Royal Society, CAPES, CNPq, FAPEAM) and **industry** (Intel, Motorola, Samsung, Nokia, ARM)

(Real) Impact: Students and Contributors

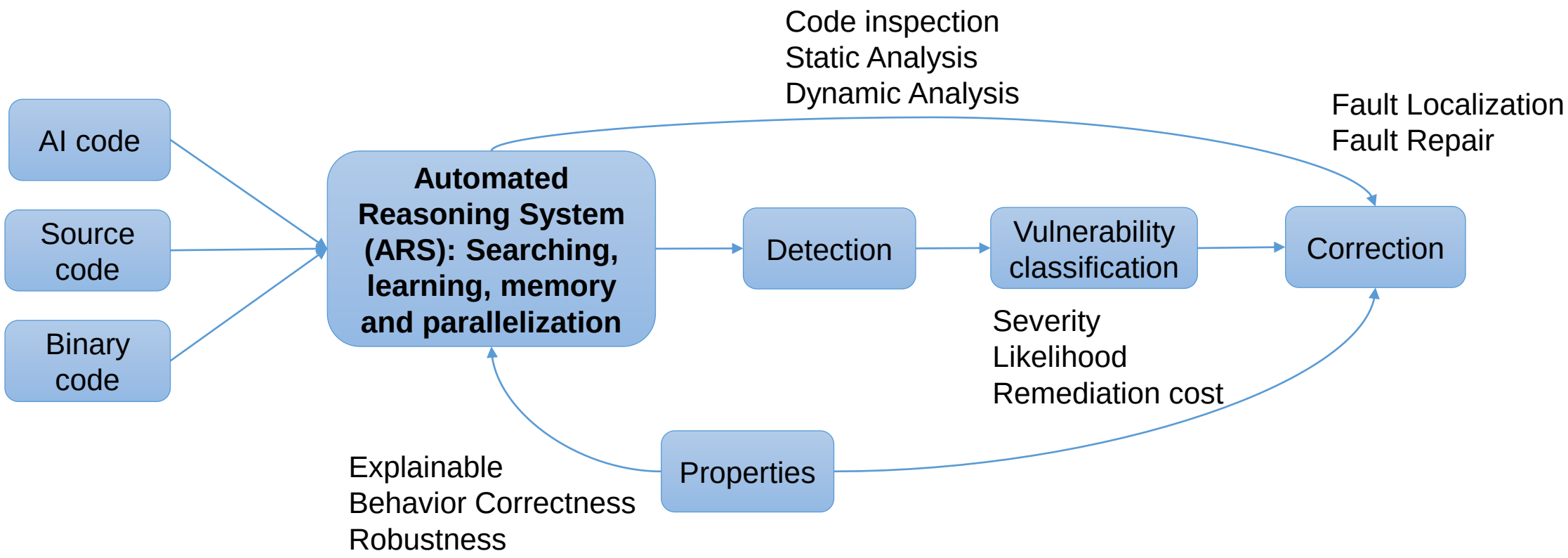
- 5 PhD theses
- 30+ MSc dissertations
- 30+ final-year projects

- GitHub:
  - 35 contributors
  - 21,580 commits
  - 195 stars
  - 81 forks

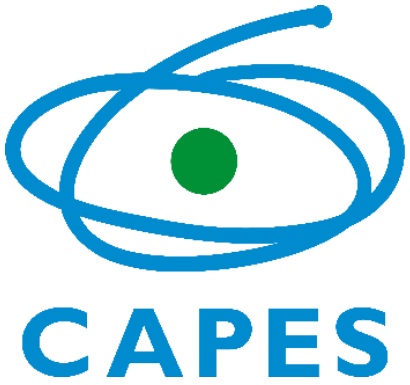
https://github.com/esbmc/esbmc

Vision: Automated Reasoning System for Secure SW and AI

Develop an automated reasoning system for safeguarding software and AI systems against security vulnerabilities in an increasingly digital and interconnected world



Acknowledgements



Agenda

- 10:00 - Arrival, biscuits, tea, and coffee (Mercury room)
- 10:30 – Welcome by Lucas
- 10:40 – ESBMC memory model by Fedor
- 11:00 – Formal Verification of Firmware by Rafael
- 11:20 – ESBMC-CHERI: Towards Verification of C Programs for CHERI Platforms with ESBMC by Franz
- 11:40 – Model Checking C++ Programs using Clang AST by Kunjian

Agenda

- 12:00 – Black-Box Cooperative Verification for Concurrent Programs by Fatimah
- 12:20 – Towards Self-Healing Software via Large Language Models and Formal Verification by Yiannus
- 12:40 – Coverage Checking and Optimizations in ESBMC by Chenfeng
- 13:00 – Lunchtime
- 14:00 – Intervals and Contractors in ESBMC by Mohanned

Agenda

- 14:20 – NeuroCodeBench: a plain C neural network benchmark for software verification by Edoardo
- 14:40 – The application of ESBMC to verify production code at ARM
- 15:00 – Coffee break
- 15:15 – Discussion
- 16:30 – Wrap up