Empirical Study on Applying Program Analysis and Testing Tools to Student Code

Frederico Ramos, Filipe Marques, Nuno Santos, Pedro Adão, José Fragoso Santos

INESC-ID / Instituto Superior Técnico, Universidade Lisboa /Instituto Telecomunicações

KLEE Workshop 2022

Motivation

How well do existing testing and verification tools perform on student code?

- ➔ **RQ1**: Number of false positives (*Precision*)
- ➔ **RQ2**: Number of false negatives (*Recall*)
- ➔ **RQ3**: Resource usage (*Memory / Time*)

Goal: Make a case for the introduction of testing and verification tools in undergraduate courses

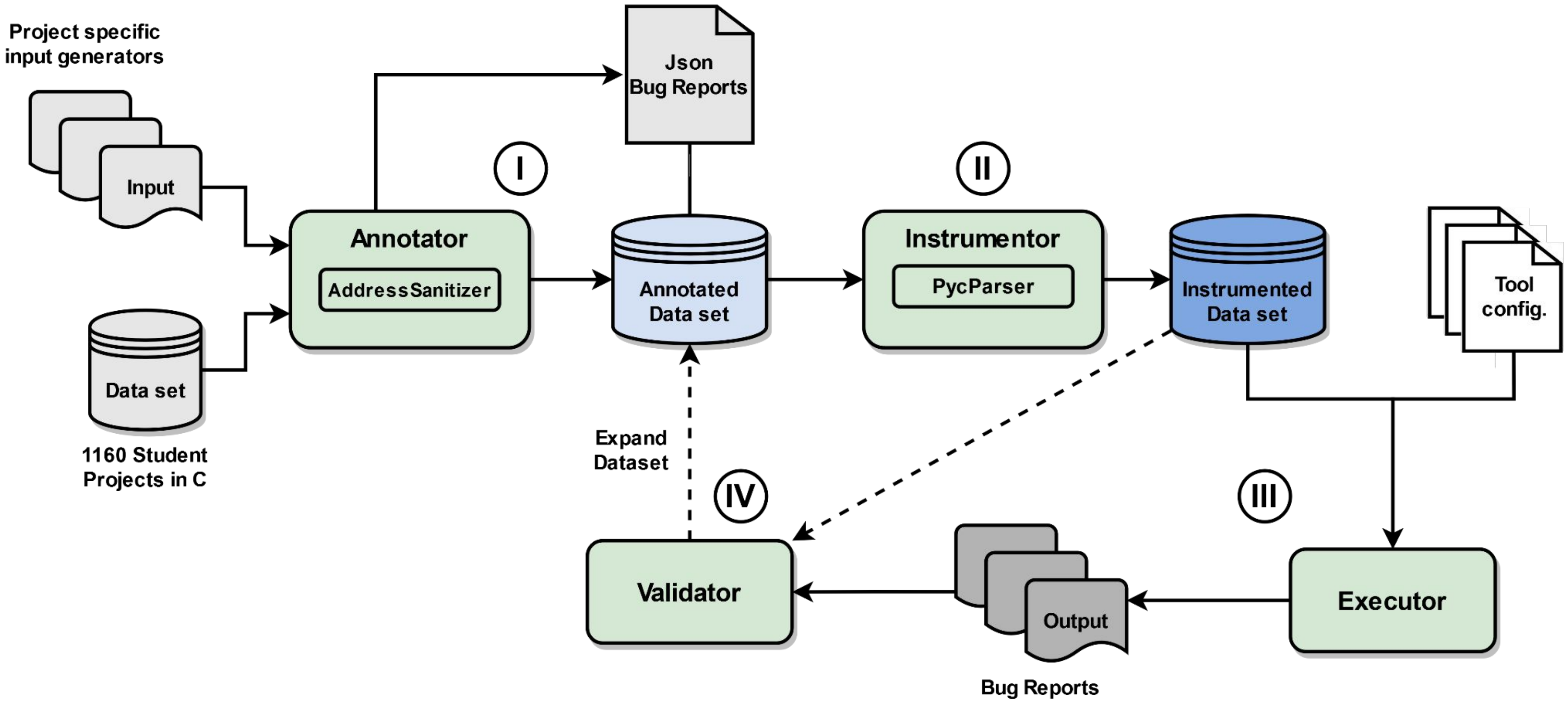
Contributions

- A curated data set consisting of **1160 student projects (405k LoC)** annotated with bug locations
  - Types of bugs detected: heap-overflows, stack-overflows invalid pointers, uninitialised-memory, stack-underflow, memory Leaks

- An empirical study characterizing how **9 state-of-the-art testing and verification tools** perform on our curated data set
  - Selected tools: FuseBMC, LibKluzzer, Verifuzz, Klee, Symbiotic, CPAchecker, Infer, Pulse

- **Preliminary results** obtained for: Infer, Pulse, KLEE, Symbiotic

Methodology



Tool Selection Criteria

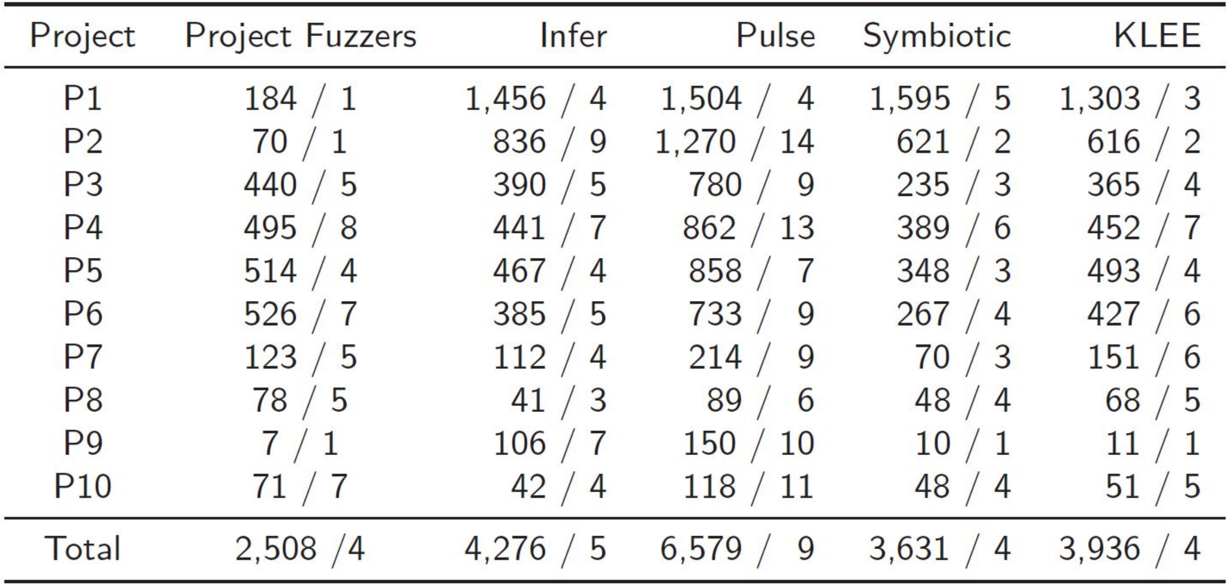
- **C1:** 5 best-performing tools in the *Cover-Error* category in **Test-Comp 2022**
  - FuseBMC, LibKluzzer, Verifuzz, Klee, Symbiotic

- **C2:** Winners of the categories *MemSafety*, *NoOverflows* and *SoftwareSystems* from **SV-Comp 2022**
  - Symbiotic, CPAchecker

- **C3:** Other high-profile static analysis tools
  - Infer, Pulse

If you want us to include your tool, contact us!

Preliminary Results



Project	Project Fuzzers	Infer	Pulse	Symbiotic	KLEE
P1	184 / 1	1,456 / 4	1,504 / 4	1,595 / 5	1,303 / 3
P2	70 / 1	836 / 9	1,270 / 14	621 / 2	616 / 2
P3	440 / 5	390 / 5	780 / 9	235 / 3	365 / 4
P4	495 / 8	441 / 7	862 / 13	389 / 6	452 / 7
P5	514 / 4	467 / 4	858 / 7	348 / 3	493 / 4
P6	526 / 7	385 / 5	733 / 9	267 / 4	427 / 6
P7	123 / 5	112 / 4	214 / 9	70 / 3	151 / 6
P8	78 / 5	41 / 3	89 / 6	48 / 4	68 / 5
P9	7 / 1	106 / 7	150 / 10	10 / 1	11 / 1
P10	71 / 7	42 / 4	118 / 11	48 / 4	51 / 5
Total	2,508 /4	4,276 / 5	6,579 / 9	3,631 / 4	3,936 / 4

All evaluated tools perform well, uncovering most of the memory bugs present in the dataset, with **high recall and precision**. Updated results in the poster

Thank You

Frederico Ramos

https://frediramos.github.io/

Collaborate with us

- ➔ The annotated dataset will be open-sourced
- ➔ Do you want us to include your tool in our study? Come talk to us!