10th UCAAT User Conference on Advanced Automated Testing

ETSI The Standards People

Detecting Equivalent Mutants Through Fuzzing

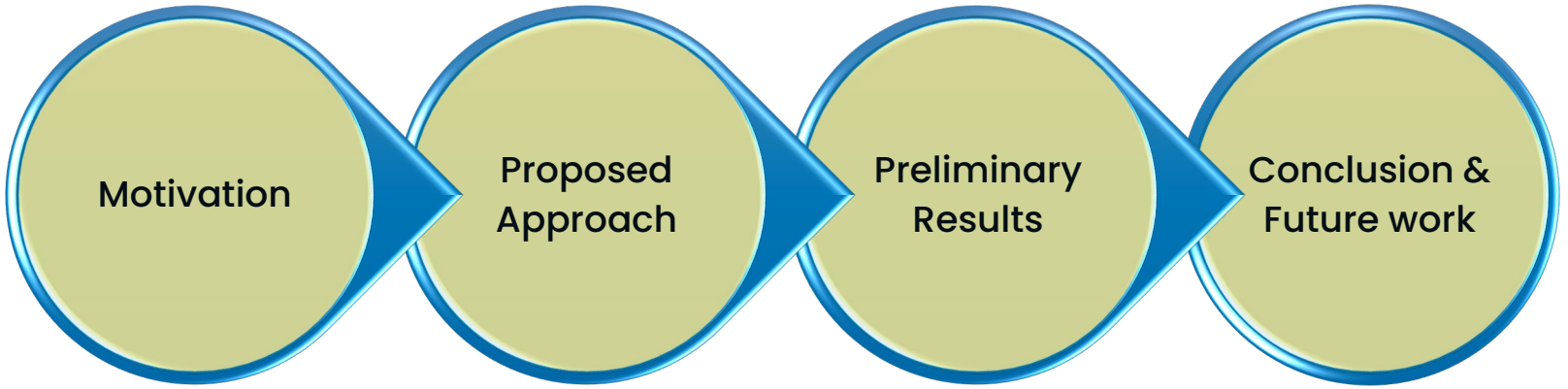
Zaheed Ahmed, Jens Grabowski

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN IN PUBLICA COMMODA SEIT 1737

SWT

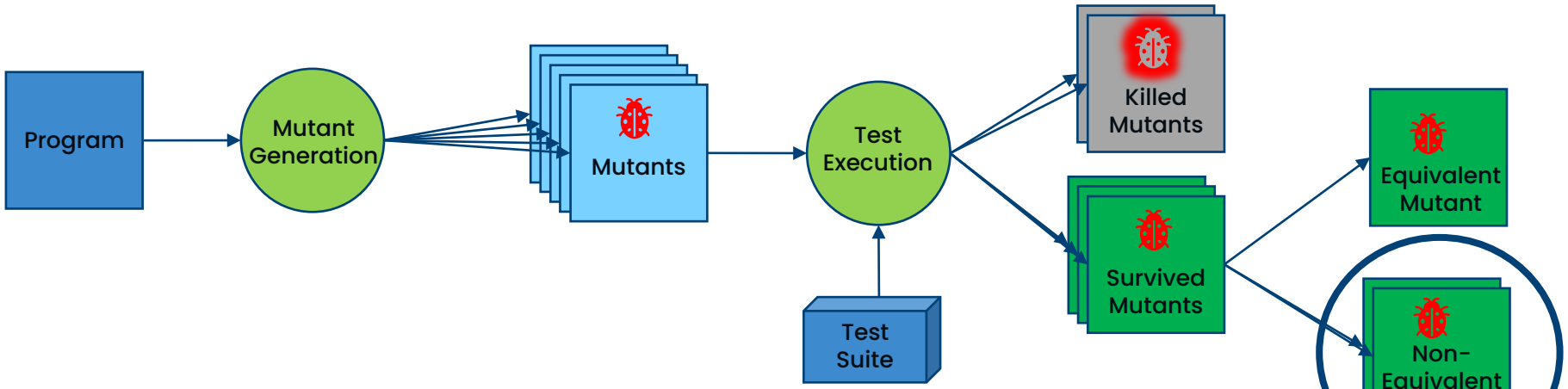
16/11/2023

Outline


Motivation
Proposed Approach
Preliminary Results
Conclusion & Future work

Motivation

- How do you assess the quality of a test suite?



Mutation Testing Process

Our approach tries to kill as many non-equivalent mutants as possible.

Motivation: Mutant

Original Program

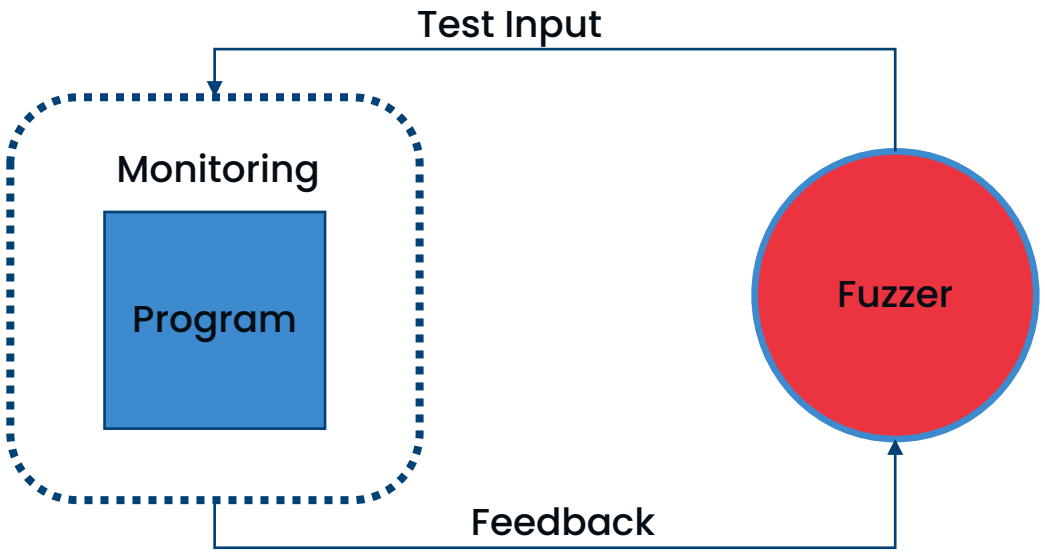
```
public void someLogic(int i) {
 if (i <= 100) {
  throw new IllegalArgumentException();
  }
 if (i > 100) {
  doSomething();
 }
}
```

Mutant

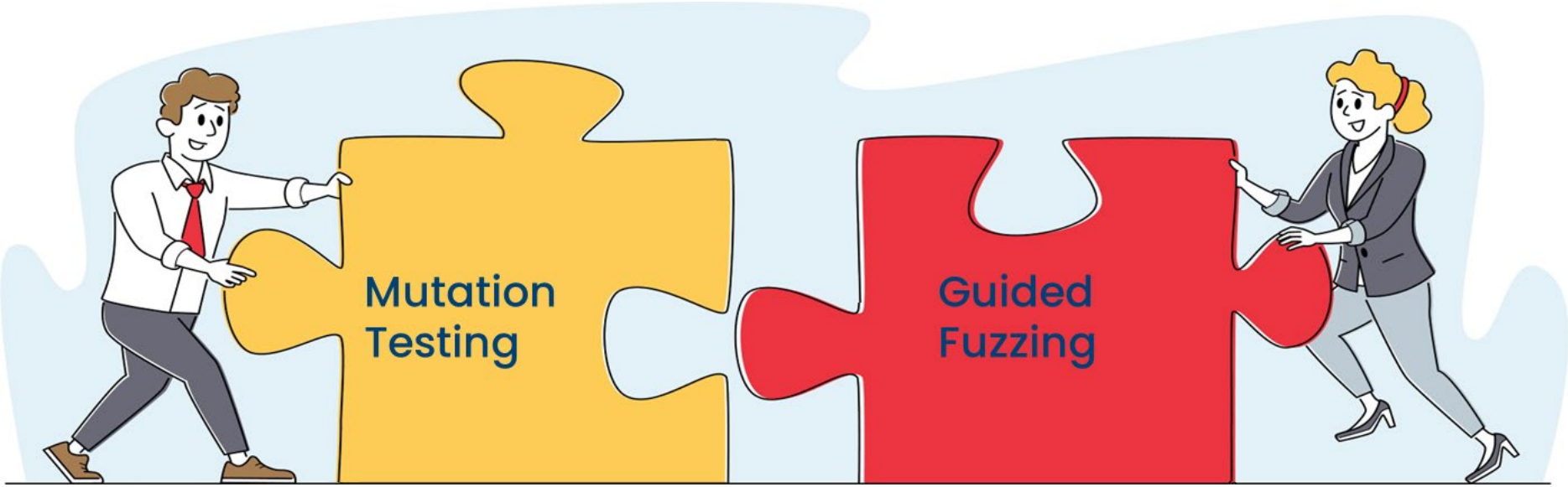
```
public void someLogic(int i) {
 if (i < 100) {
  throw new IllegalArgumentException();
  }
 if (i > 100) {
  doSomething();
 }
}
```

- Test suite = {(Input i=95, expected= Exception), (Input i=103, Expected = Something)}
- But what about Input i = 100?
- Is there a possibility to generate additional and relevant test data automatically?

Motivation: Fuzzing



Proposed Approach: Idea

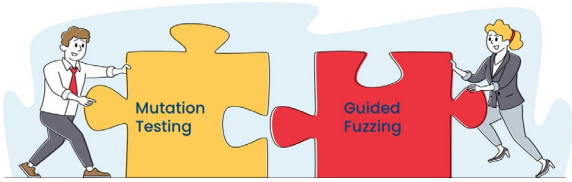


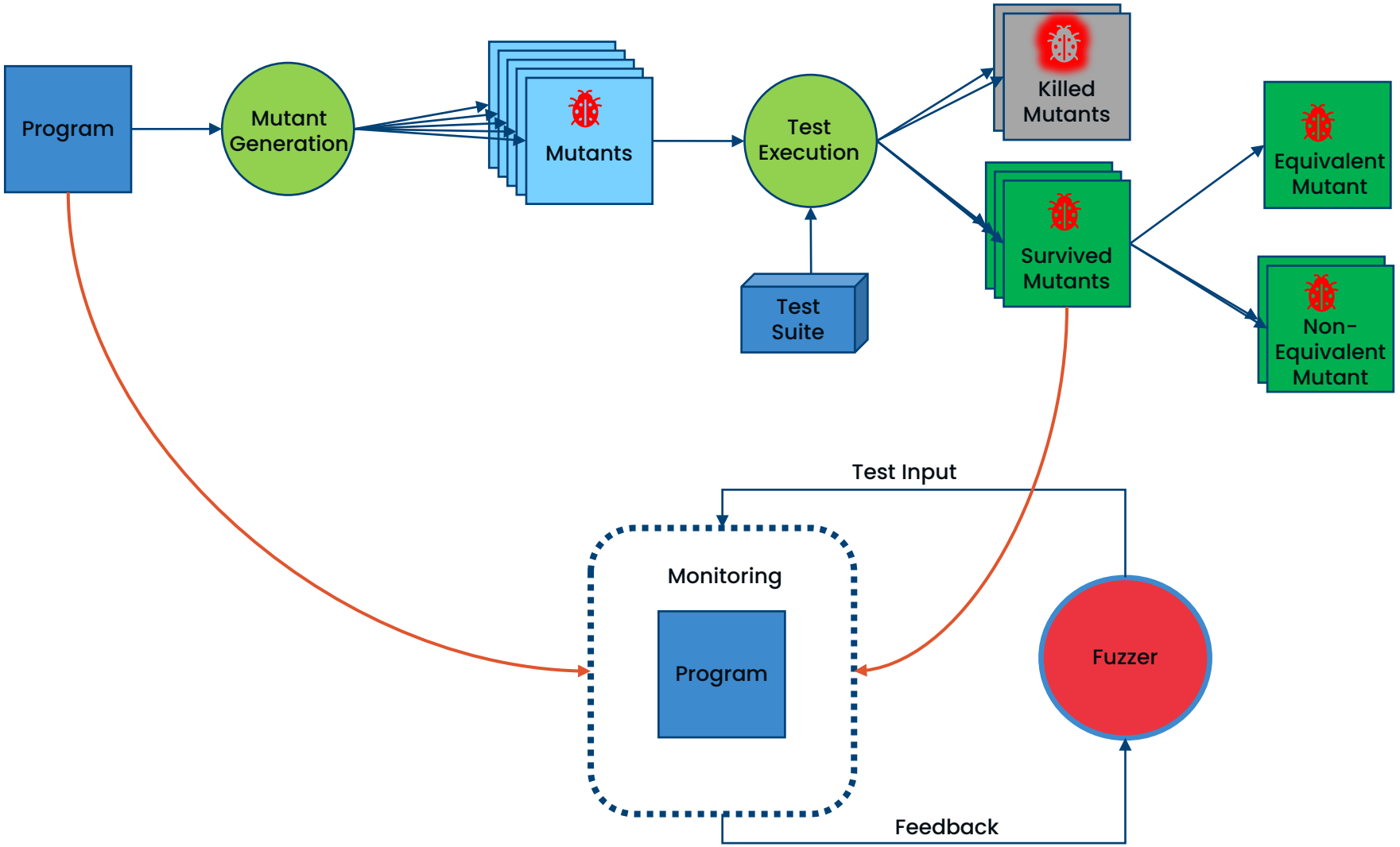
Adapted [1]

- Mutation testing: assess the tests quality by injecting new bugs
- Guided fuzzing: search for relevant test input

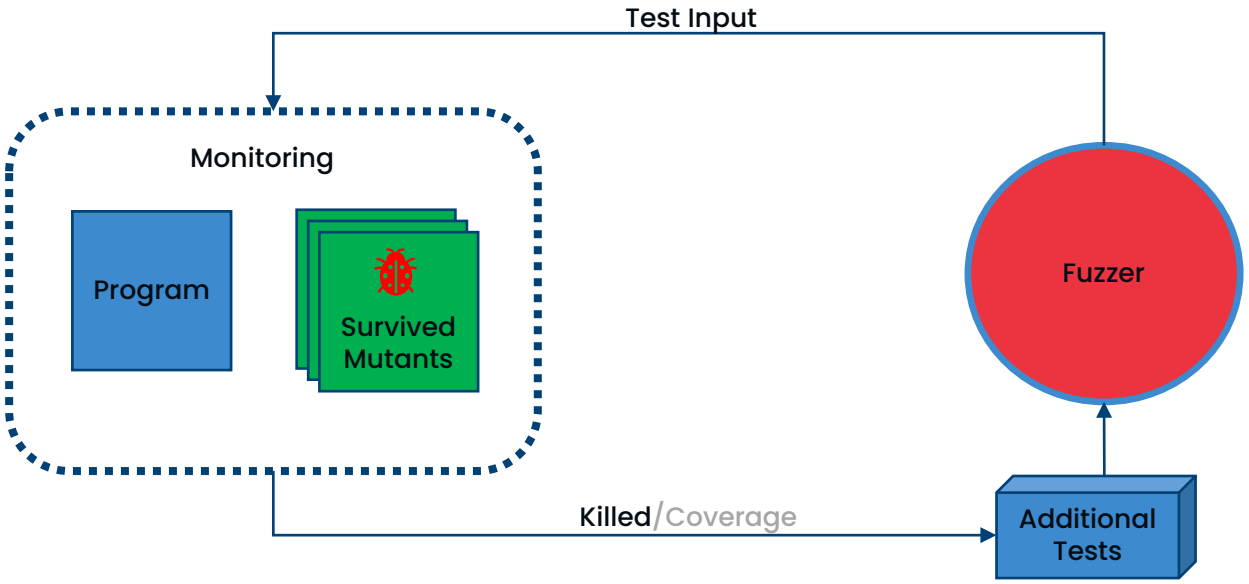
[1] Pekka Aho, Complementing GUI Testing Scripts with Smart Monkey Testing, 09UCAAT, 2022

Proposed Approach: Overview





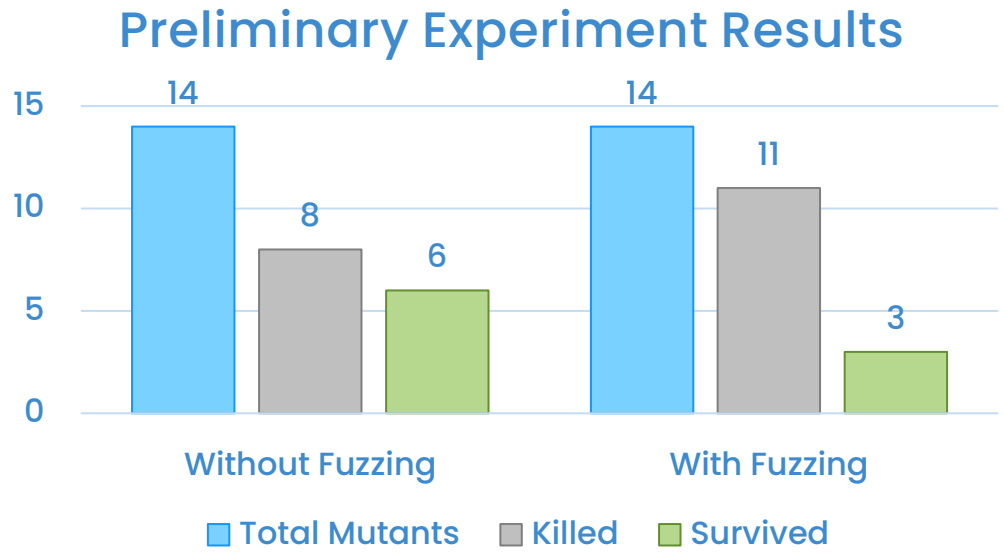
Proposed Approach: Overview



Preliminary Results

Experiment

- Tested on Bisect benchmark program
- Encouraging results



Conclusion & Future Work

Conclusion

- Our technique helps to automatically:
  - Detect equivalent mutants
  - Augment test suite

Future Work

- Conduct more experiments
- Add other adequacy criteria in fuzzing loop

Thank you!

zaheed.ahmed@cs.uni-goettingen.de