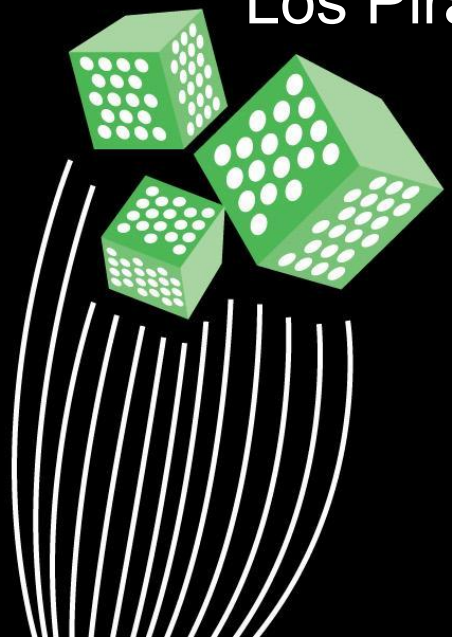
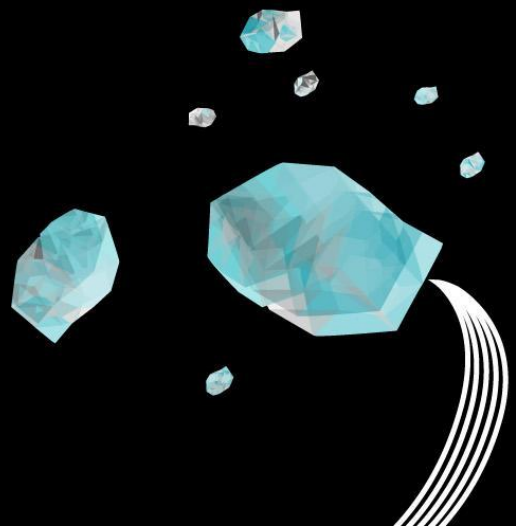
UNIVERSITY OF TWENTE.

Automatic Software Testing Techniques

Joris Diesvelt & Roeland Krak

Los Piratas Informaticos





Automatic Software Testing Techniques

- Fuzzing
- Symbolic Execution
- Concolic Execution

Fuzzing

- Black-box
- Provide input
  - Generative
  - Mutative
- Monitor unexpected behaviour
- Discover crashes and memory leaks

Symbolic Execution

- White-box
- Evaluate code symbolically
- Reason about complex program logic
- High code coverage

Challenges

- Constraint Solving
- Memory Modeling
- Concurrency
- Black-box function calls
- Path Explosion

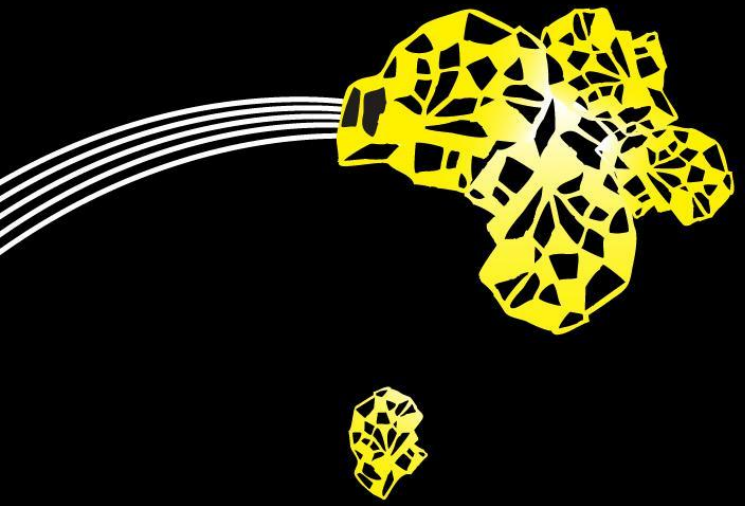
Path Explosion

- Restrict execution depth
- Partial concrete execution
- Pruning
  - RWset

Concolic Execution

- White-box
- Concrete and symbolic
- Reason about complex program logic
- High code coverage
- Alleviate imprecision
- Heavy use of constraint solver

UNIVERSITY OF TWENTE.

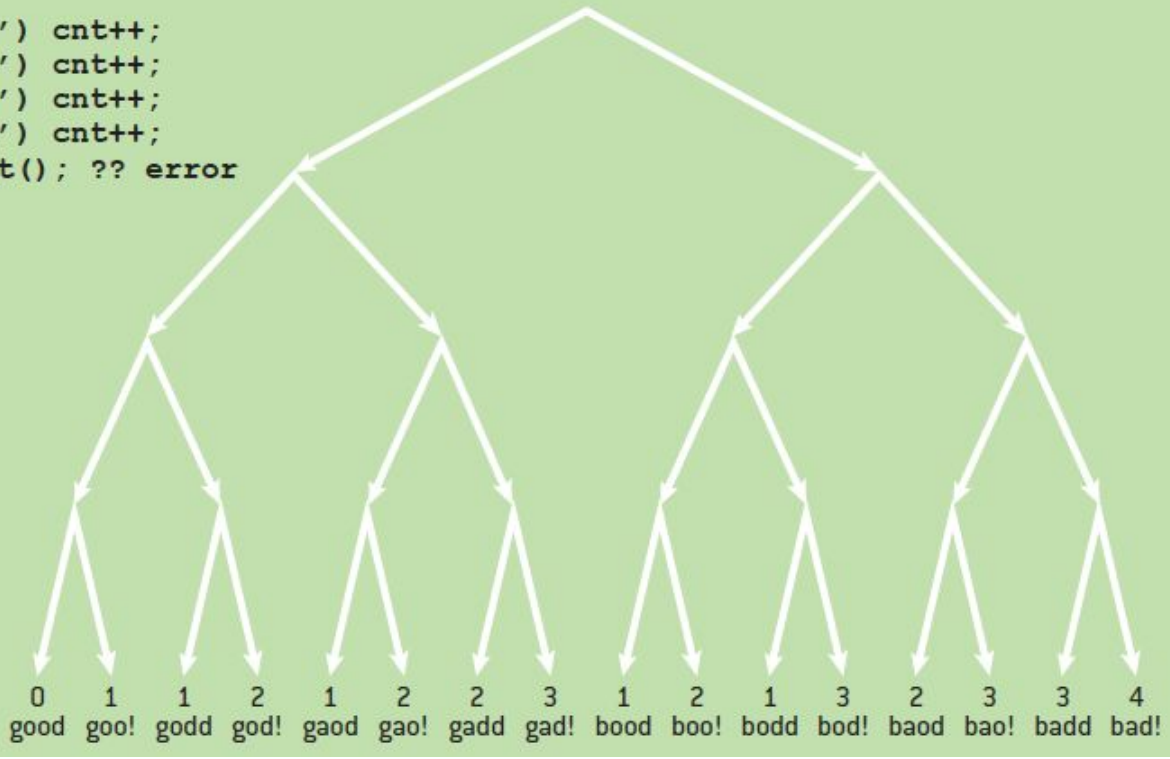


Questions

Execution Tree Example

Example of Program (Left) and Its Search Space (Right) with the Value of cnt at the End of Each Run

```
void top(char input[4] {
  int cnt=0;
  if (input[0] == 'b') cnt++;
  if (input[1] == 'a') cnt++;
  if (input[2] == 'd') cnt++;
  if (input[3] == '!') cnt++;
  if (cnt >= 4) abort(); ?? error
}
```



Godefroid, Patrice, Michael Y. Levin, and David Molnar. "SAGE: whitebox fuzzing for security testing." *Queue* 10.1 (2012): 20.