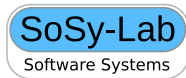


Conditional Testing

Off-the-Shelf Combination of Test-Case Generators

Thomas Lemberger
Joint work with Dirk Beyer

LMU Munich, Germany



A word cloud of software testing tools arranged in a semi-circle. The tools listed are: KLEE, PRTTest, AFL-fuzz, CoVeriTest, EXE, CPA/Tiger-MGP, Fair-fuzz, Verifuzz, DART, Shell, and CREST. The text is in various colors and orientations, with some words being larger than others.

KLEE PRTTest
AFL-fuzz
CoVeriTest EXE
CPA/Tiger-MGP
Fair-fuzz
Verifuzz DART
Shell CREST

- ▶ Automated test generation is at its peak
- ▶ But:
 - ▶ Different strengths and weaknesses
 - ▶ Every generator working on its own
 - ▶ Proprietary interfaces
- ▶ Lock-in effect

KLEE
AFL^{vs}fuzz

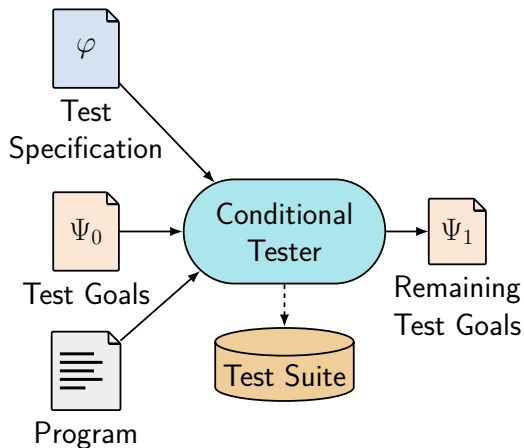
- ▶ Automated test generation is at its peak
- ▶ But:
 - ▶ Different strengths and weaknesses
 - ▶ Every generator working on its own
 - ▶ Proprietary interfaces
- ▶ Lock-in effect

AFL-fuzz
+
KLEE

```
int i = input();

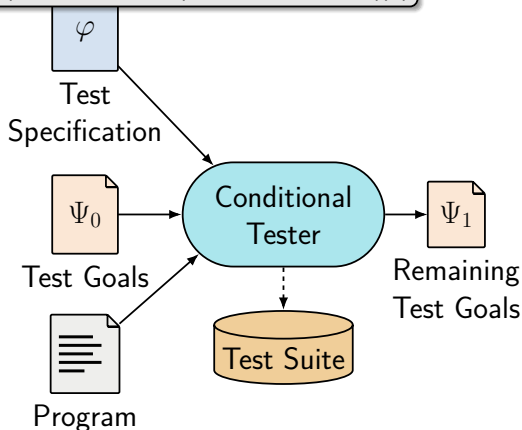
if(i != 1017) {
    while(i > 1017) {
        // branch 1.1
        // ...
        i--;
    }
} else {
    // branch 2
    // ...
}
```

Conditional Tester

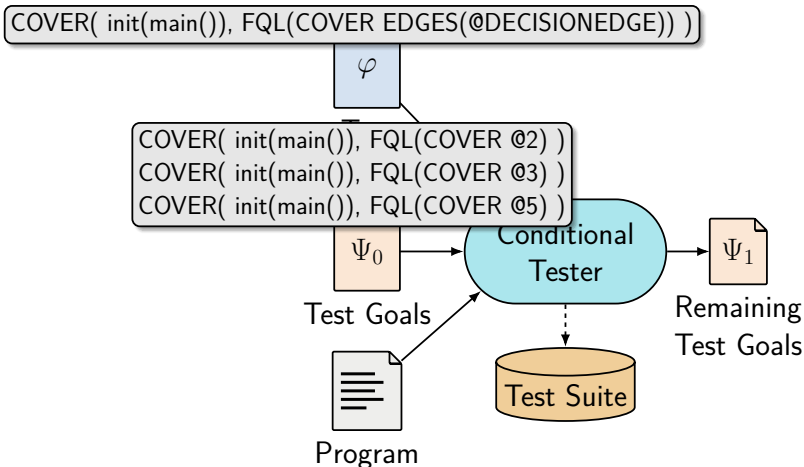


Conditional Tester

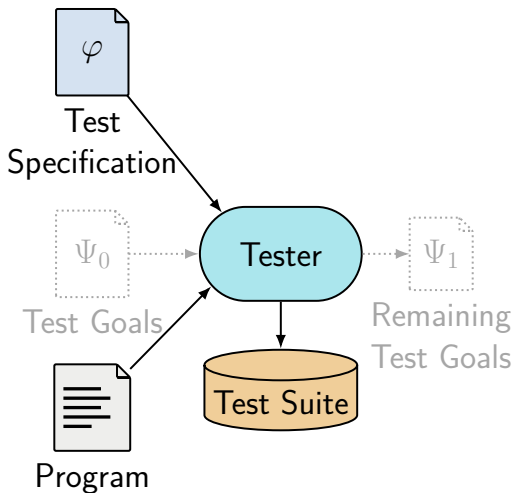
```
COVER( init(main()), FQL(COVER EDGES(@DECISIONEDGE)) )
```



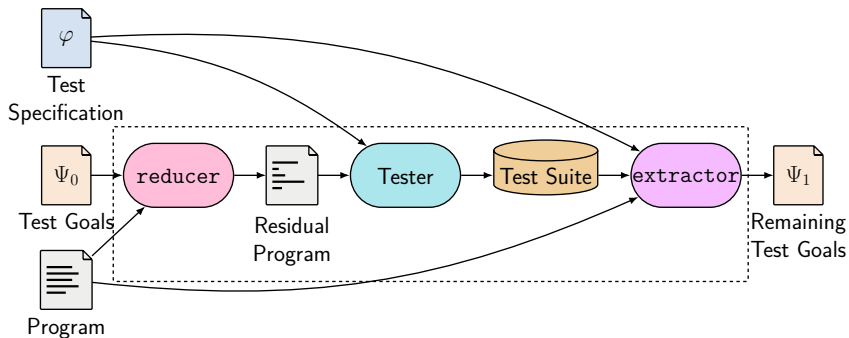
Conditional Tester



Traditional Tester



Traditional \Rightarrow Conditional



Reducer

- ▶ Input: Program P , test goals Ψ
- ▶ Output: Residual program P'
- ▶ P' *reachability-equivalent* to P with regard to Ψ

Reachability Equivalence [?]

Each program input that reaches a test goal of Ψ in P' reaches the same test goal in P

Reducer Examples

1. Identity
2. Pruning based on syntactic reachability

```
1 int i = input();
2
3 if(i != 1017) {
4     while(i > 1017) {
5         // branch 1.1
6         // ...
7         i--;
8     }
9 } else {
10 // branch 2
11 // ...
12 }
```

@10
→

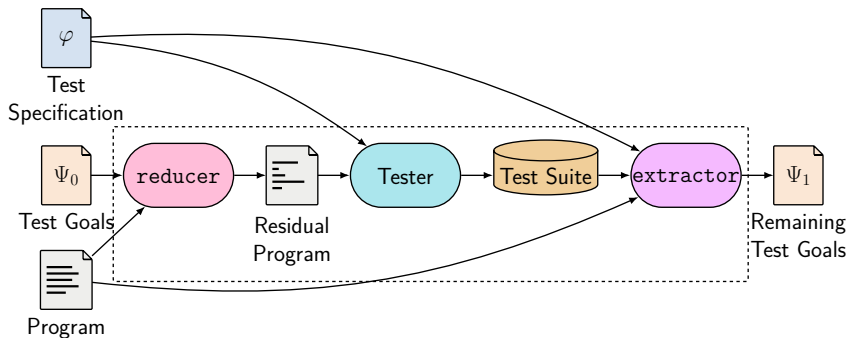
```
1 int i = input();
2
3 if(i != 1017) {
4
5
6
7
8
9 } else {
10 // branch 2
11 // ...
12 }
```

Test-Goal Extractor

- ▶ Input: Program P , test specification φ , test suite S
- ▶ Output: Test goals Ψ described by φ but not covered by S

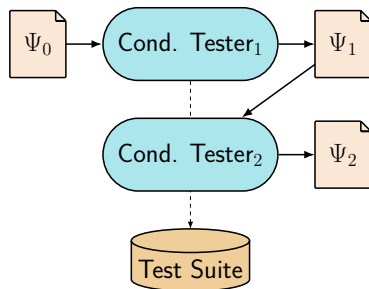
- ▶ Example: Test execution + coverage measurement
- ▶ (here: based on Test-Comp format)

Traditional \Rightarrow Conditional



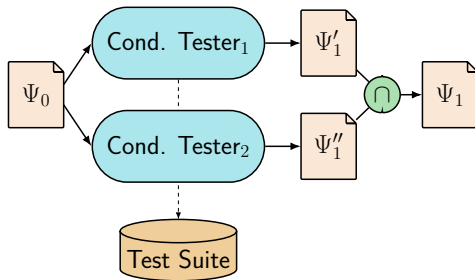
Combinations

► Sequential



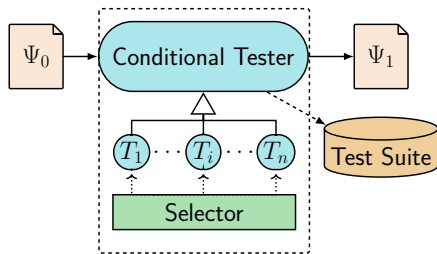
Combinations

- ▶ Sequential
- ▶ Portfolio



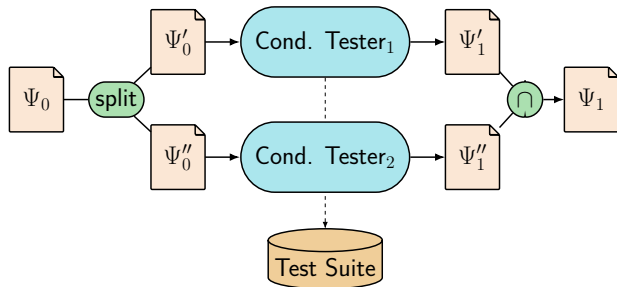
Combinations

- ▶ Sequential
- ▶ Portfolio
- ▶ Strategy selection



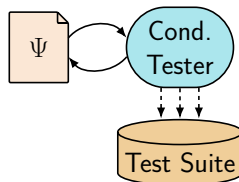
Combinations

- ▶ Sequential
- ▶ Portfolio
- ▶ Strategy selection
- ▶ Compositional

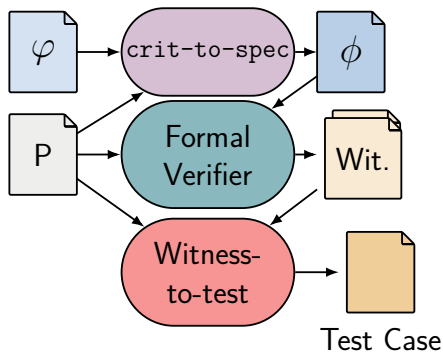


Combinations

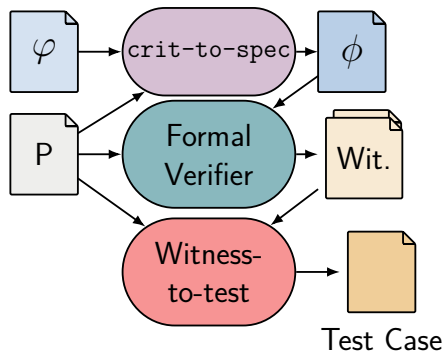
- ▶ Sequential
- ▶ Portfolio
- ▶ Strategy selection
- ▶ Compositional
- ▶ Cyclic



Tests from Formal Verification (I)



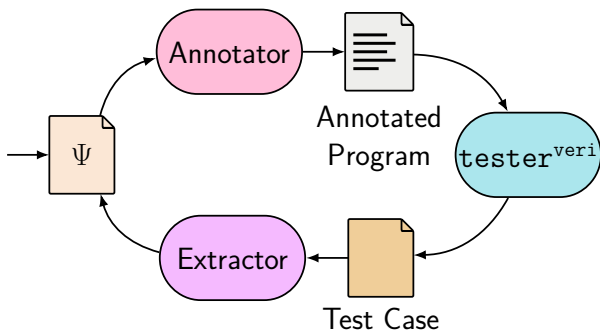
Tests from Formal Verification (I)



- ▶ Only one test per verifier-run
- ⇒ conditional testing

Tests from Formal Verification (II)

- ▶ Reducer: identity + annotate goals with `__VERIFIER_error`
- ▶ Apply cyclic tester



Evaluation (I)

- ▶ CPA-TIGER + CoVeriTEST + KLEE (300 s each)
- ▶ Sequential without (id) and with info exchange (prune)

Task	coverage (%)		
	id	→	prune
rewnifrev	0	+92	92
rewrev	8	+83	92
simple_array_index_value_2	17	+67	83
array_3-1	50	+38	88
Problem102_label43	33	+8	41
Problem102_label34	44	+4	49
Problem08_label46	47	+4	52
Problem08_label33	47	+4	51
Problem08_label16	45	+4	49
Problem102_label37	42	+3	45

Evaluation (II)

- ▶ CPA-TIGER + CoVeriTEST + KLEE (prune)
- ▶ CPA-TIGER + CoVeriTEST + KLEE (200 s each) + ESBMC (300 s) (vb)

Task	coverage (%)		
	prune	→	vb
Problem08_label59	6	+ 50	56
Problem08_label26	6	+ 50	55
Problem102_label47	7	+ 48	55
Problem102_label11	7	+ 48	55
Problem102_label53	7	+ 48	55
Problem09_label15	4	+ 37	41
Problem09_label06	4	+ 37	41
Problem09_label18	4	+ 36	40
Problem08_label34	25	+ 32	57
Problem103_label42	9	+ 30	39

