

Part 5: Cooperative Verification by Tests from Witnesses: Execution-Based Validation of Verification Results

Dirk Beyer¹

Joint work with Matthias Dangl¹, Thomas Lemberger¹ and
Michael Tautschnig²

Published in Proc. TAP 2018



¹LMU Munich, Germany



²Queen Mary University of London, UK

Software contains bugs.

Software contains bugs.
⇒ Automatic verification.

Software contains bugs.
⇒ Automatic verification.
But software contains bugs.

Software contains bugs.
⇒ Automatic verification.

But software contains bugs.
⇒ Automatic validation of results.

Software contains bugs.

⇒ Automatic verification.

But software contains bugs.

⇒ Automatic validation of results.

But software contains bugs.

Software contains bugs.

⇒ Automatic verification.

But software contains bugs.

⇒ Automatic validation of results.

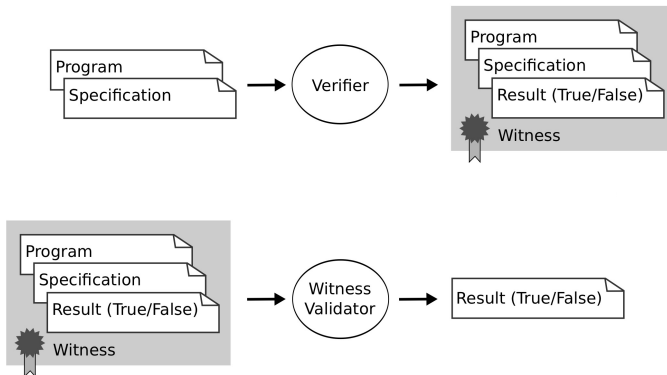
But software contains bugs.

⇒ Execution as proof.

1. Overview Witness Validation
2. Execution-based Witness Validation
3. Experimental Results

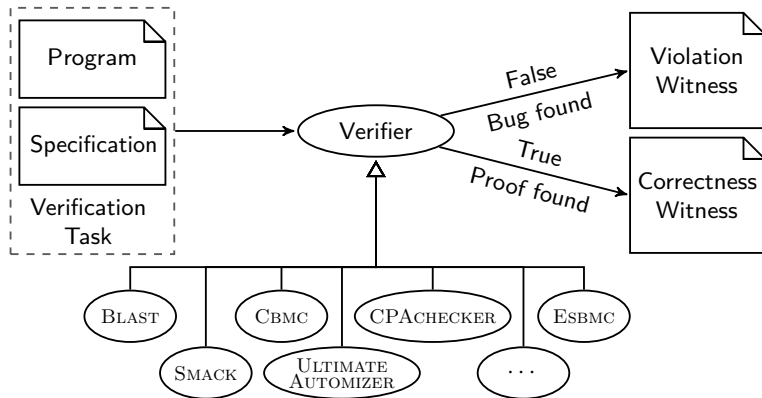
Witness Validation

- ▶ Problem: Confidence in verifiers
- ▶ Approach: Witness validation



[Beyer/Dangl/Dietsch/Heizmann/Stahlbauer FSE'15]

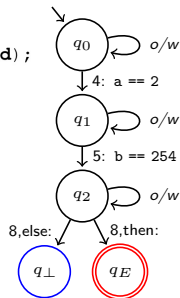
Witness Creation



Violation Witness Format: Witness Automaton

- ▶ Automaton
- ▶ Describes set of error paths
- ▶ State-space + source-code guards

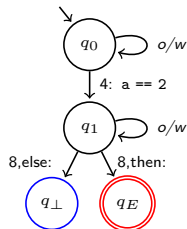
```
1 extern void __VERIFIER_error(void);  
2 extern unsigned char __VERIFIER_nondet_uchar(void);  
3 int main(void) {  
4   unsigned char a = __VERIFIER_nondet_uchar();  
5   unsigned char b = __VERIFIER_nondet_uchar();  
6   unsigned char sum = a + b;  
7   unsigned char mean = sum / 2;  
8   if (mean < a / 2) {  
9     __VERIFIER_error();  
10  }  
11  return 0;  
12 }
```



Witness Refinement

► Problem 1: Abstract witnesses

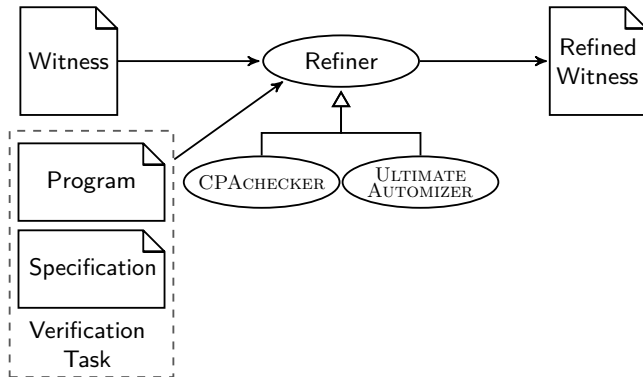
```
1 extern void __VERIFIER_error(void);
2 extern unsigned char __VERIFIER_nondet_uchar(void);
3 int main(void) {
4   unsigned char a = __VERIFIER_nondet_uchar();
5   unsigned char b = __VERIFIER_nondet_uchar();
6   unsigned char sum = a + b;
7   unsigned char mean = sum / 2;
8   if (mean < a / 2) {
9     __VERIFIER_error();
10  }
11  return 0;
12 }
```



b == ?

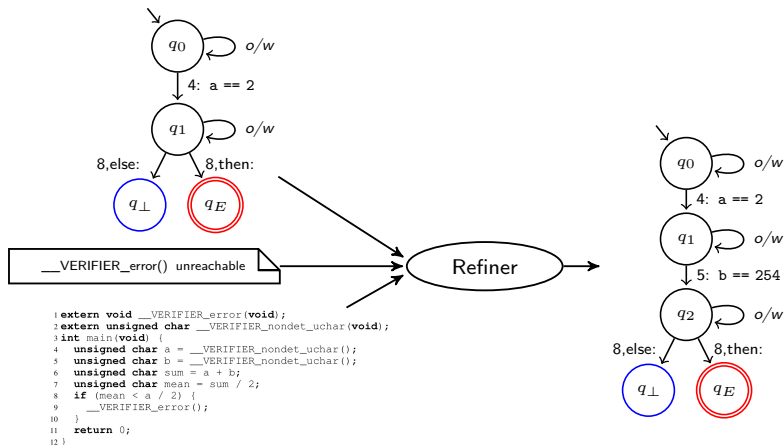
Witness Refinement

- ▶ Problem 1: Abstract witnesses
- ▶ Solution: Witness refinement



Witness Refinement

- ▶ Problem 1: Abstract witnesses
- ▶ Solution: Witness refinement

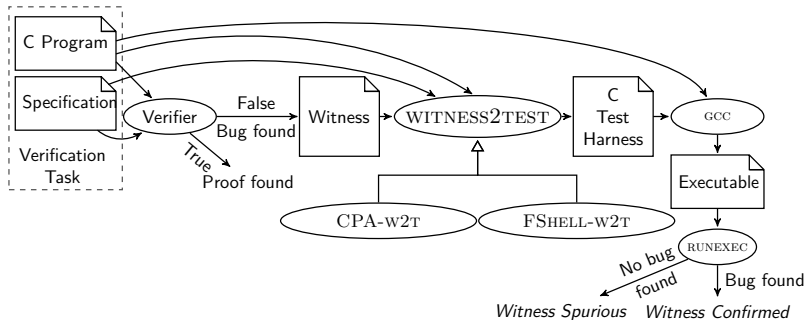


Witness Validation

- ▶ Existing validators are model checkers
- ▶ Problem 2: Confidence in validators
- ▶ Problem 3: Found errors difficult to debug
- ▶ Solution: Executable counterexamples

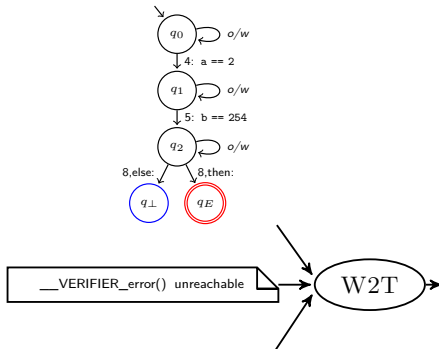
Execution-based Witness Validation

Execution-based Witness Validation



Execution-based Witness Validation

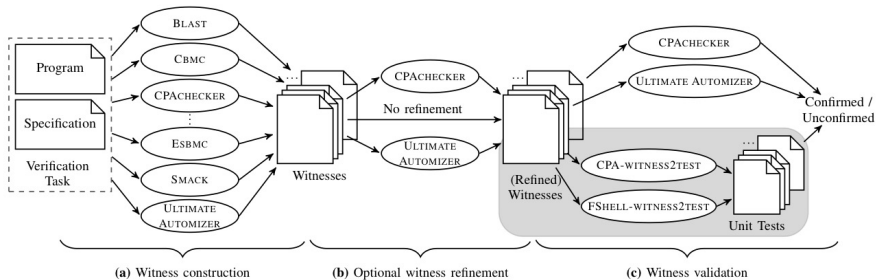
- ▶ Build executable counterexample from witness
- ▶ State-space guards $\mapsto$ input variables/functions



```
1 extern void __VERIFIER_error(void);
2 extern unsigned char __VERIFIER_nondet_uchar(void);
3 int main(void) {
4     unsigned char a = __VERIFIER_nondet_uchar();
5     unsigned char b = __VERIFIER_nondet_uchar();
6     unsigned char sum = a + b;
7     unsigned char mean = sum / 2;
8     if (mean < a / 2) {
9         __VERIFIER_error();
10    }
11    return 0;
12 }
```

```
1 #include <stdlib.h>
2 void __VERIFIER_error() { exit(107); }
3 unsigned char __VERIFIER_nondet_uchar() {
4     static unsigned int test_vector_index = 0;
5     unsigned char retval;
6     switch (test_vector_index) {
7         case 0: retval = 2U; break;
8         case 1: retval = 254U; break;
9     }
10    ++test_vector_index;
11    return retval;
12 }
```

Full Workflow



Experimental Results

Experiments

- ▶ Implementations: CPA-W2T and FShell-W2T
- ▶ Witness Refiner: CPACHECKER
- ▶ Benchmark set:
 - ▶ 18 965 witnesses
 - ▶ From 21 verifiers
 - ▶ From 5 692 verification tasks (1 490 false tasks)

Validation Performance

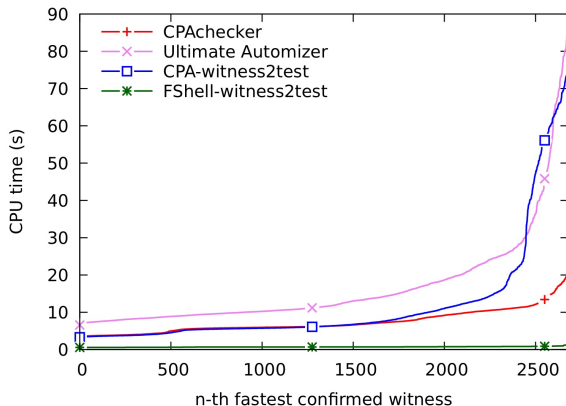
- ▶ 18 965 witnesses in total
- ▶ Not only increase of confidence,
but also increase of overall effectivity

TABLE III: Validation results of static/dynamic validators

	Static	Dynamic	Union
Confirmed results	12 671	8 702	14 434
Incorrectly confirmed results	21	6	27

Time Performance

- ▶ Time comparison over 2 680 witnesses that all validators confirmed



Verdict

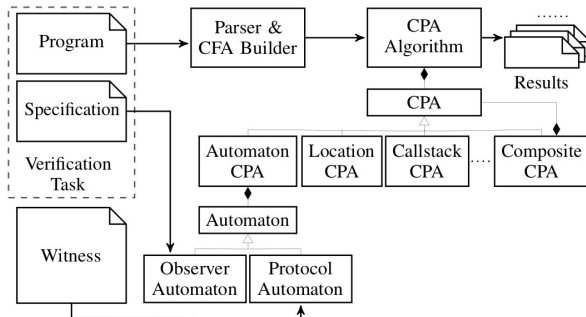
- ▶ Validate **more** witnesses
- ▶ Validate witnesses **faster**
- ▶ Provide **debuggable** counterexamples
- ▶ Provide **executable** tests
- ▶ Increase **confidence** in results

Thank You!

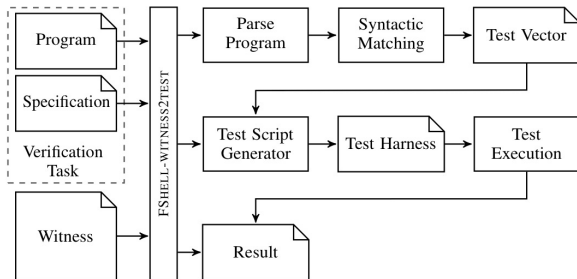
Future Work

- ▶ Floating point benchmarks / binary representations
- ▶ Concurrent programs
- ▶ Undefined behavior

Architecture CPA-w2t



Architecture FShell-w2t



Experiment Environment

- ▶ Machines:
 - ▶ Intel Xeon E3-1230 v5 CPU, 8 units, 3.4 GHz
 - ▶ 33 GB RAM
 - ▶ Ubuntu 16.04
- ▶ Limits verifiers:
 - ▶ 4 processing units
 - ▶ 7 GB RAM
 - ▶ 15 min CPU time
- ▶ Limits validators:
 - ▶ 2 processing units
 - ▶ 4 GB RAM
 - ▶ 1.5 min CPU time

Verification Tasks

TABLE I: Subject verification tasks from the SV-BENCHMARKS repository

Sub-category	Number of verification tasks
ReachSafety-Arrays	135
ReachSafety-BitVectors	50
ReachSafety-ControlFlow	94
ReachSafety-ECA	1 149
ReachSafety-Floats	172
ReachSafety-Heap	173
ReachSafety-Loops	156
ReachSafety-ProductLines	597
ReachSafety-Recursive	98
ReachSafety-Sequentialized	273
Systems_DeviceDriversLinux64_ReachSafety	2 795
Total	5 692

Witness Set

TABLE II: Number of violation witnesses produced by verifiers from the subject verification tasks

Verifier	Ref.	Produced witnesses	Refined witnesses	Total witnesses
2LS	[39]	992	384	1 376
BLAST	[41]	778	202	980
CBMC	[31]	831	467	1 298
CEAGLE		619	426	1 045
CPA-BAM-BNB	[2]	851	175	1 026
CPA-KIND	[9]	263	193	456
CPA-SEQ	[19]	883	767	1 650
DEPTHK	[37]	1 159	305	1 464
ESBMC	[34]	653	148	801
ESBMC-FALSI	[34]	981	395	1 376
ESBMC-INCR	[34]	970	392	1 362
ESBMC-KIND	[20]	847	352	1 199
FORESTER	[27]	51	0	51
PREDATORHP	[30]	86	61	147
SKINK	[13]	30	25	55
SMACK	[36]	871	632	1 503
SYMBIOTIC	[15]	927	411	1 338
SYMDIVINE	[29]	247	223	470
ULTIMATE AUTOMIZER	[25]	514	70	584
UKOJAK	[35]	309	67	376
UTAIKAN	[22]	338	70	408
Total		13 200	5 765	18 965