

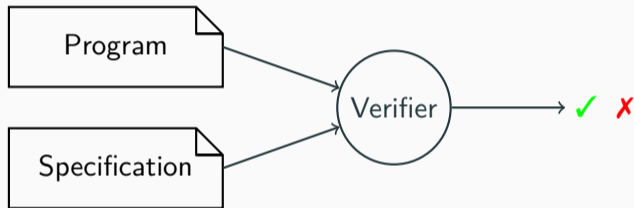
Witness-guided verification in trace abstraction

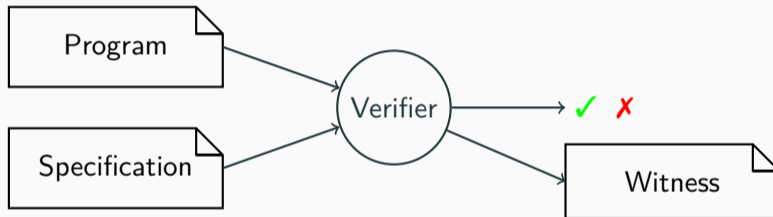
AVM 2025

Frank Schüssele

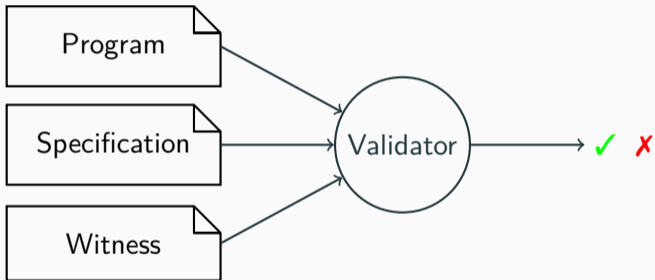
25th September 2025

University of Freiburg, Germany





Witness Validation



Correctness Witness

Violation Witness

Correctness Witness

- partial maps from program locations to invariants

Violation Witness

Correctness Witness

- partial maps from program locations to invariants

Violation Witness

- abstract counterexample

Correctness Witness

- partial maps from program locations to invariants
- valid if all invariants hold at location and program correct

Violation Witness

- abstract counterexample

Correctness Witness

- partial maps from program locations to invariants
- valid if all invariants hold at location and program correct

Violation Witness

- abstract counterexample
- valid if witness describes violation of specification

- validate verification tools

- validate verification tools ✓

- validate verification tools ✓
- speed up verification

- validate verification tools ✓
- speed up verification ?

Verification vs. Validation (of Correctness Witnesses)

	verification		validation			
witness			UAUTOMIZER		CPACHECKER	
	#	time (s)	#	time (s)	#	time (s)
✓	1402	30.2	1025	28.4	1305	30.3
✗	-	-	27	14.3	16	11.8

- What if we not interested in validity of invariants?

- What if we not interested in validity of invariants?
- Use invariants as hints, without checking

- What if we not interested in validity of invariants?
- Use invariants as hints, without checking

⇒ **Witness-guided verification**

- What if we not interested in validity of invariants?
- Use invariants as hints, without checking

⇒ **Witness-guided verification**

(already done for predicate abstraction, k-induction¹ and abstract interpretation²)

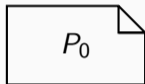
¹Jan Haltermann and Heike Wehrheim. “**CoVEGI: Cooperative Verification via Externally Generated Invariants**”. In: *FASE*. 2021.

²Simmo Saan et al. “**Correctness Witness Validation by Abstract Interpretation**”. In: *VMCAI*. 2024.

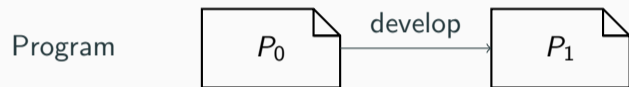
Use Case: Incremental Verification

Use Case: Incremental Verification

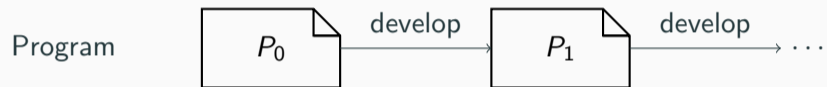
Program



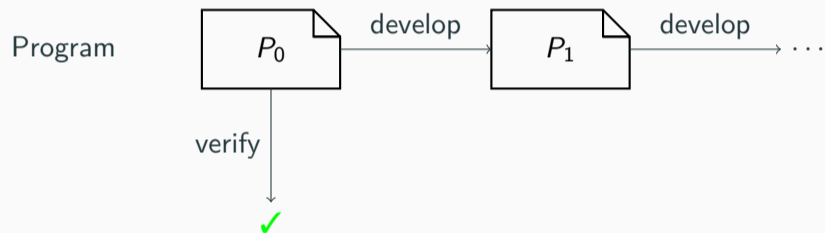
Use Case: Incremental Verification



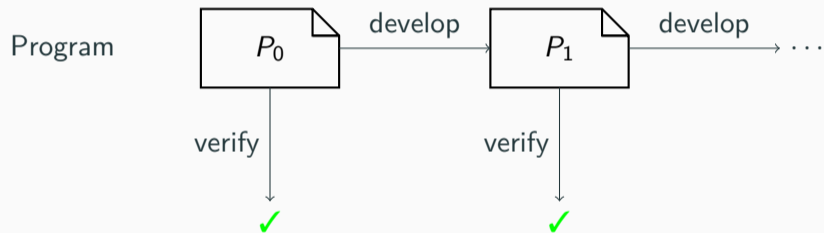
Use Case: Incremental Verification



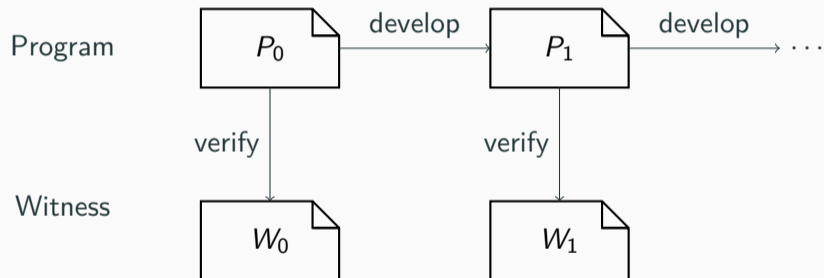
Use Case: Incremental Verification



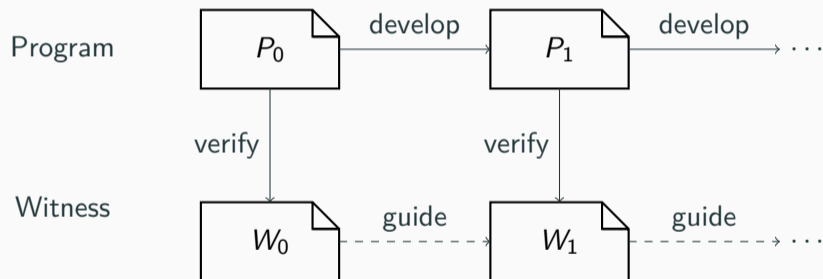
Use Case: Incremental Verification



Use Case: Incremental Verification



Use Case: Incremental Verification



Why is witness validation (of correctness witnesses) often slower than verification?

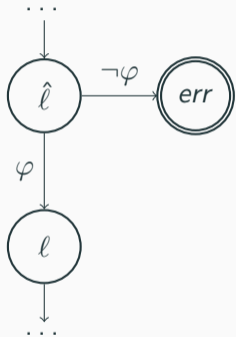
Witness validation through instrumentation

Witness validation through instrumentation

1. Instrument invariants as assertions

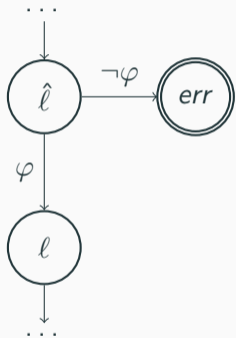
Witness validation through instrumentation

1. Instrument invariants as assertions



Witness validation through instrumentation

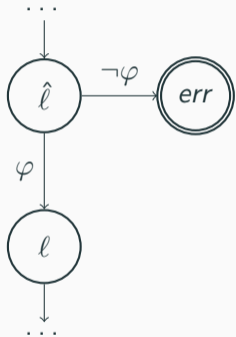
1. Instrument invariants as assertions



2. Verify instrumented program

Witness validation through instrumentation

1. Instrument invariants as assertions

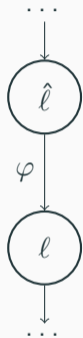


⇒ additional specification to prove

2. Verify instrumented program

Instrumentation for witness-guided verification

Instrumentation for witness-guided verification

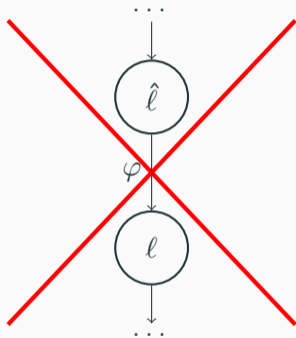


Instrumentation for witness-guided verification



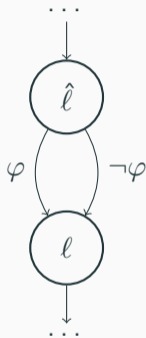
⇒ Only sound if φ is a valid invariant, would need validity check first

Instrumentation for witness-guided verification

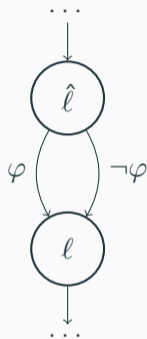


⇒ Only sound if φ is a valid invariant, would need validity check first

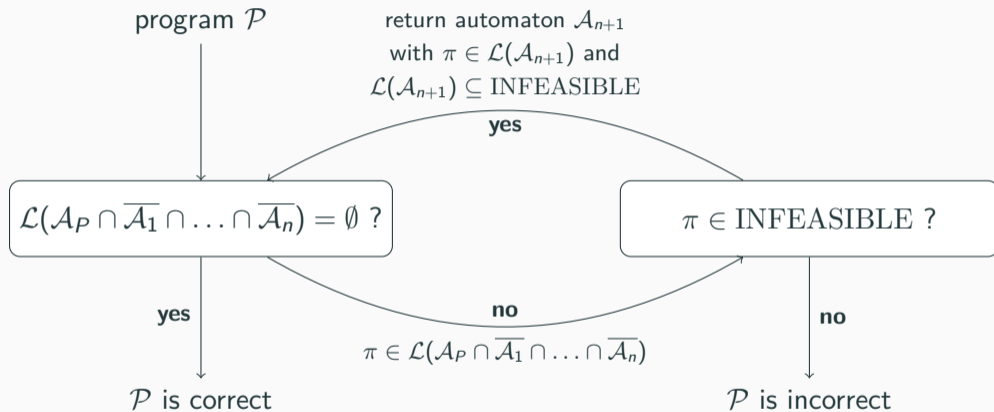
Instrumentation for witness-guided verification



Instrumentation for witness-guided verification



\Rightarrow Sound, but needs integrating in validation algorithm



³Matthias Heizmann, Jochen Hoenicke, and Andreas Podelski. “**Refinement of Trace Abstraction**”. In: *Static Analysis*. 2009.

Witness guided Trace abstraction

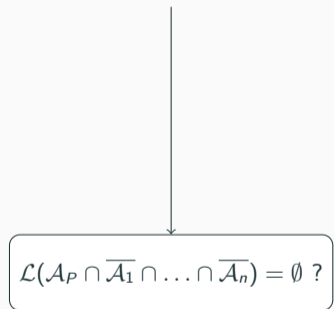
program \mathcal{P}
instrumented with witness



$$\mathcal{L}(\mathcal{A}_P \cap \overline{\mathcal{A}}_1 \cap \dots \cap \overline{\mathcal{A}}_n) = \emptyset ?$$

Witness guided Trace abstraction

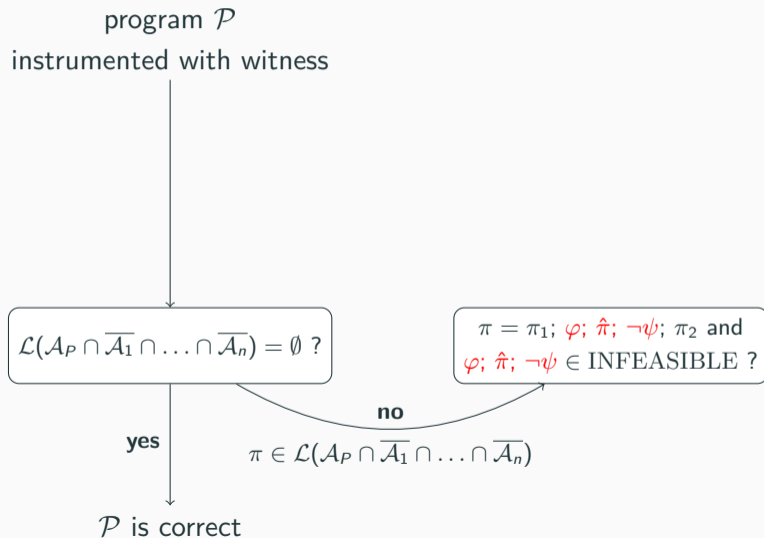
program \mathcal{P}
instrumented with witness



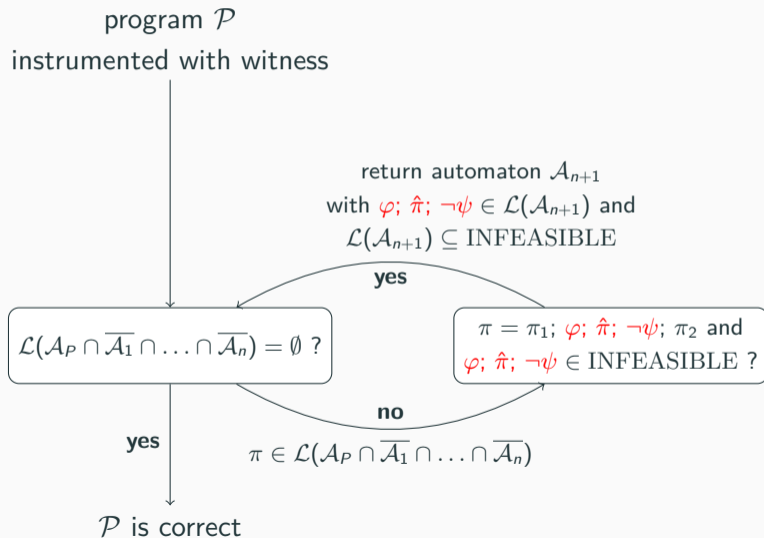
yes

\mathcal{P} is correct

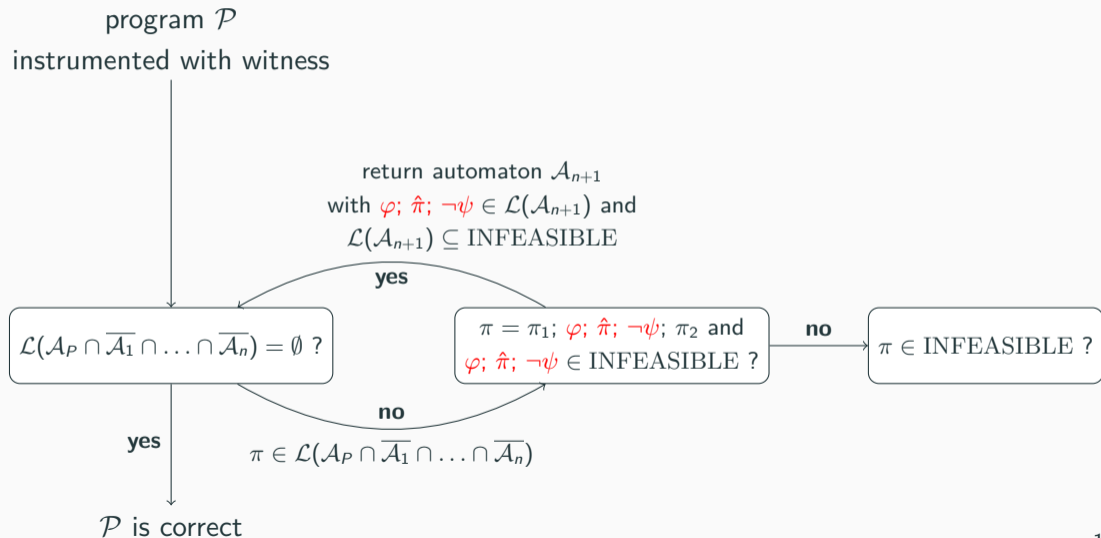
Witness guided Trace abstraction



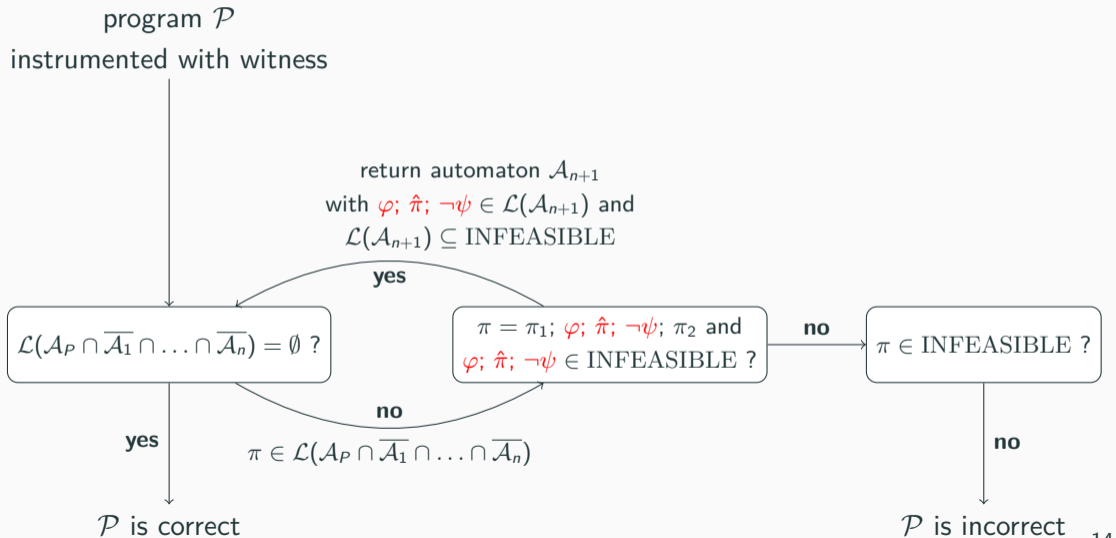
Witness guided Trace abstraction



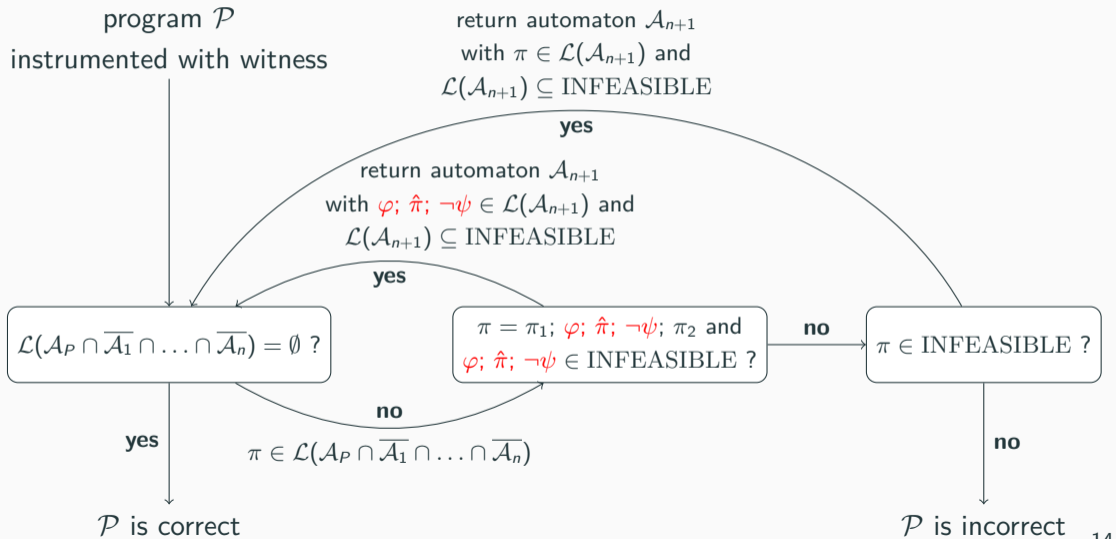
Witness guided Trace abstraction



Witness guided Trace abstraction



Witness guided Trace abstraction



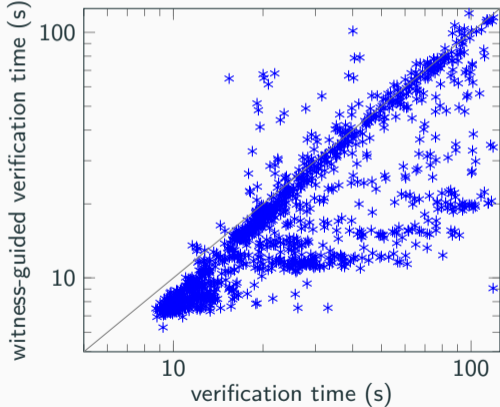
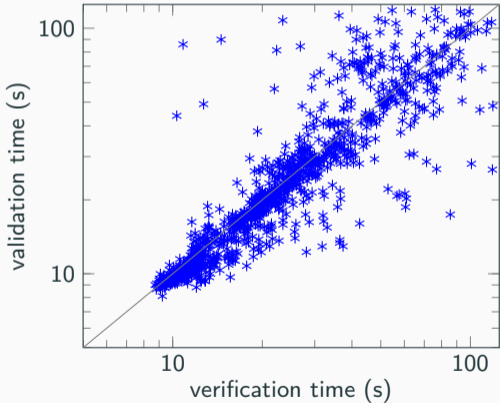
- ReachSafety category of SV-COMP 2025, where UAUTOMIZER and CPACHECKER proved correctness (1402 tasks)

- ReachSafety category of SV-COMP 2025, where UAUTOMIZER and CPACHECKER proved correctness (1402 tasks)
- comparison of UAUTOMIZER:
 - verification
 - validation (with witnesses from UAUTOMIZER and CPACHECKER)
 - witness-guided verification (with witnesses from UAUTOMIZER and CPACHECKER)

- ReachSafety category of SV-COMP 2025, where UAUTOMIZER and CPACHECKER proved correctness (1402 tasks)
- comparison of UAUTOMIZER:
 - verification
 - validation (with witnesses from UAUTOMIZER and CPACHECKER)
 - witness-guided verification (with witnesses from UAUTOMIZER and CPACHECKER)
- time limit 120 s, memory limit 8 GB
- AMD Ryzen Threadripper 3970X at 3.7 GHz limited to 2 cores

	verification		validation				witness-guided verification			
witness			UAUTOMIZER		CPACHECKER		UAUTOMIZER		CPACHECKER	
	#	time (s)	#	time (s)	#	time (s)	#	time (s)	#	time (s)
✓	1402	30.2	1025	28.4	1305	30.3	1115	22.7	1327	28.9
✗	-	-	27	14.3	16	11.8	-	-	-	-

UAUTOMIZER witnesses



CPACHECKER witnesses

