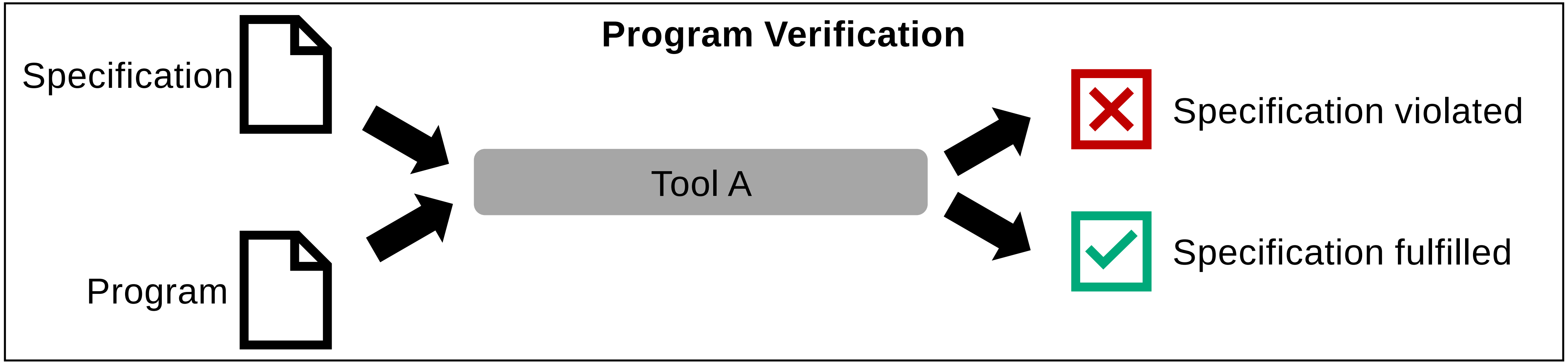
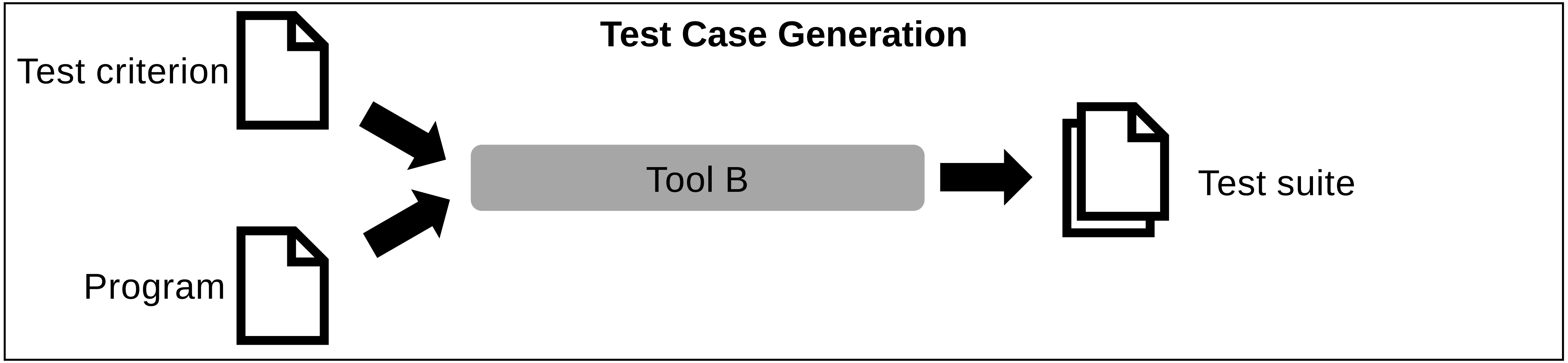
Carl von Ossietzky
Universität
Oldenburg

Information Exchange between Over- and Under-approximating Software Analyses

Jan Haltermann and Heike Wehrheim
CPAchecker Workshop, 06.10.2022

Motivation


Program Verification
Specification
Program
Tool A
Specification violated
Specification fulfilled

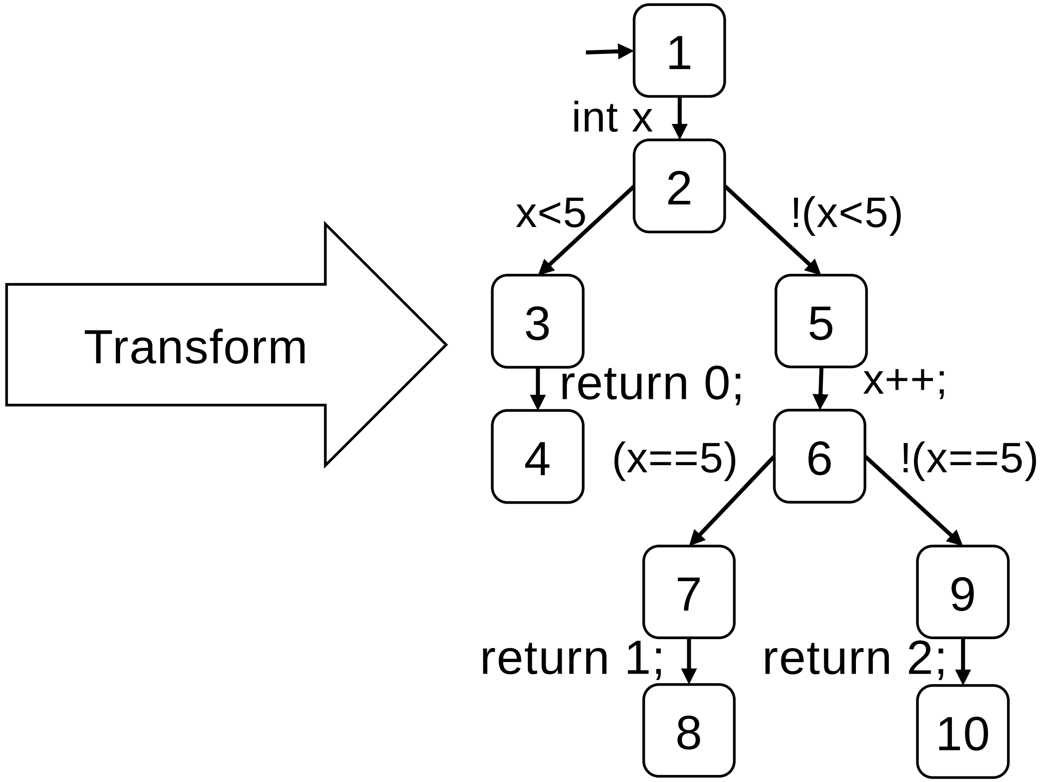

Test Case Generation
Test criterion
Program
Tool B
Test suite

Test Case Generation

– Goal: Generate input values s.t. branches (3,5,7,9) reached

```c
int main(int x){
  if (x < 5)  {
    return 0;
  } else {
    x++;
    if (x == 5) {
      return 1;
    } else  {
      return 2;
    }
  }
}
```

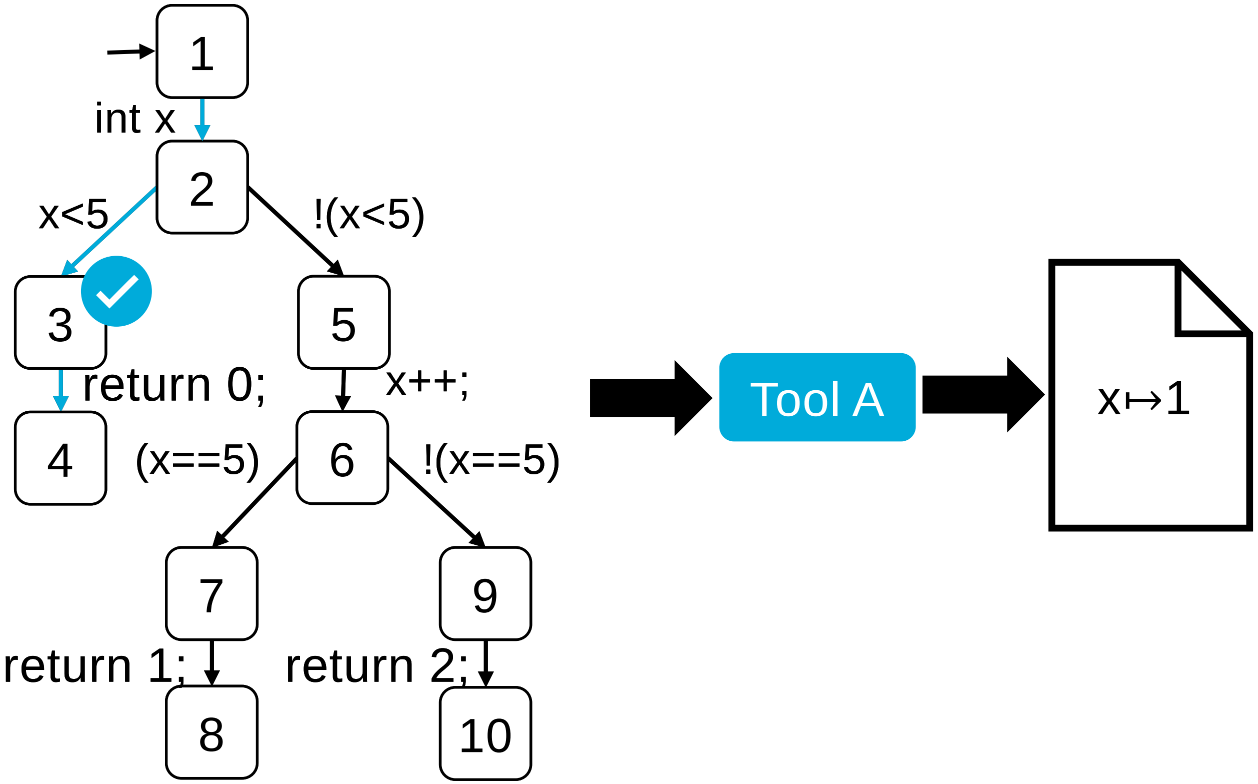
Program

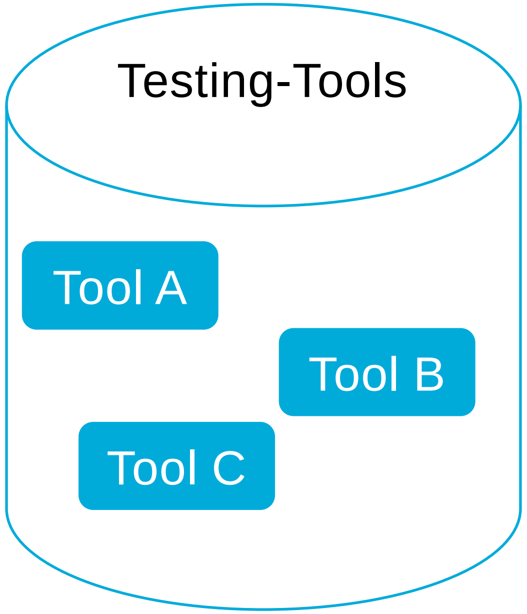


CFA

Test Case Generation

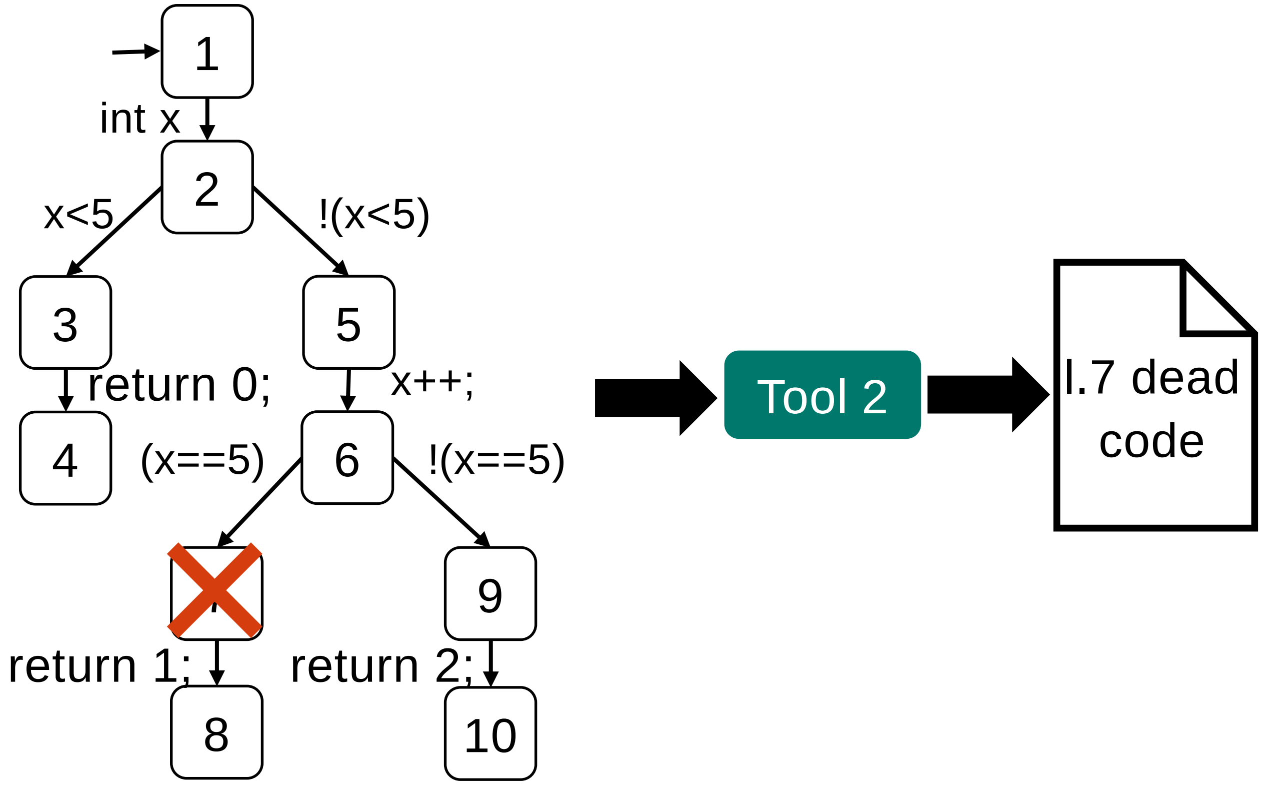
– Goal: Generate input values s.t. branches (3,5,7,9) reached

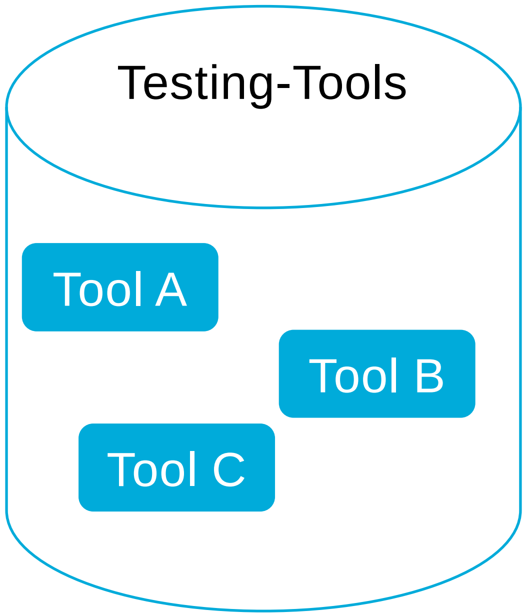


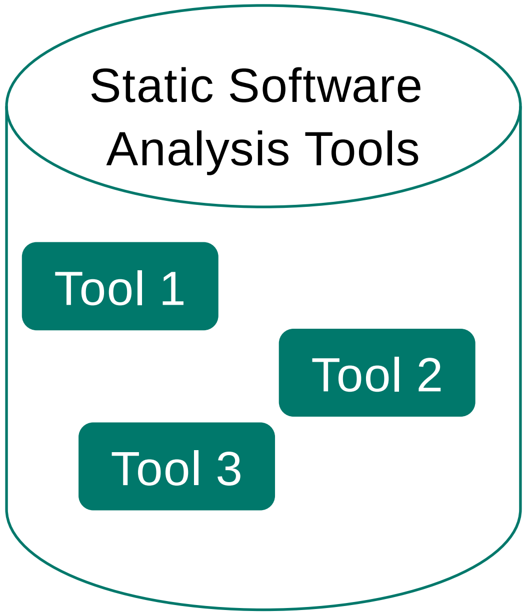


Test Case Generation

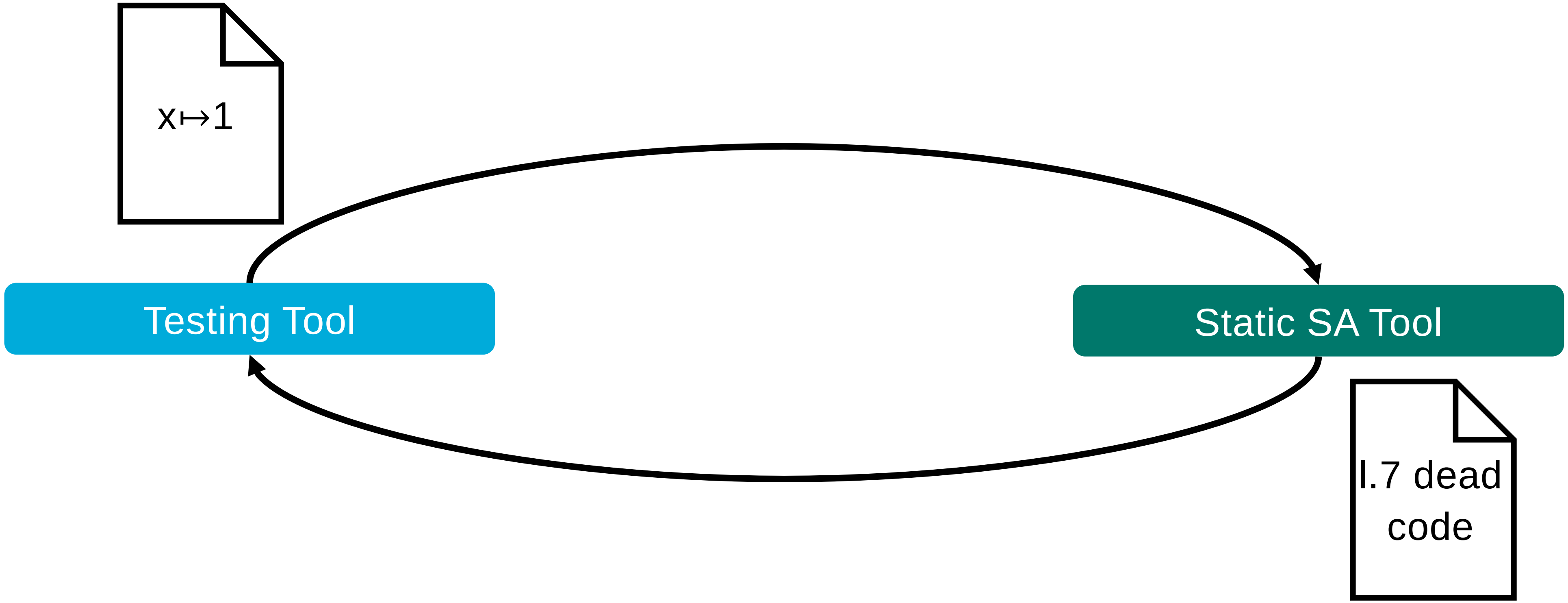
– Goal: Generate input values s.t. branches (3,5,7,9) reached







Cooperative Software Validation



Different tasks and information computed - How to exchange them?

Over- and Underapproximating Software Analyses

Underapproximative (UA):

- Analyze at most all feasible paths
- Cannot prove unreachability of target nodes
- Definite information for reachability of target nodes

Testing Tool

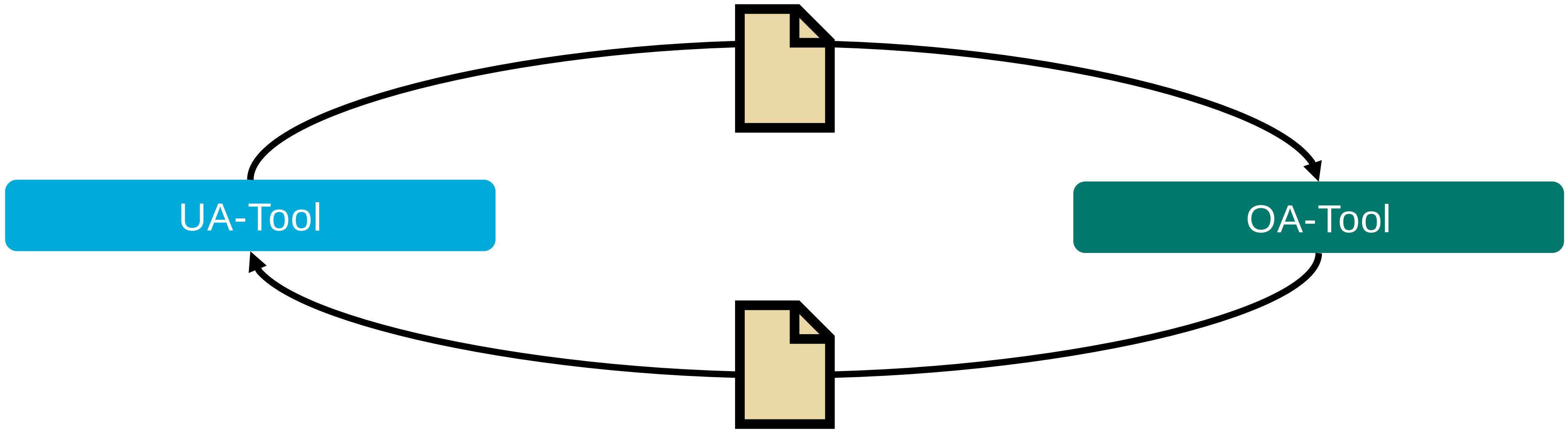
Overapproximation (OA):

- Analyze at least all feasible paths
- Path to target nodes may be spurious
- Definite information for unreachability of target nodes

Static SA Tool

Carl von Ossietzky Universität Oldenburg

Goal: Uniformed Verification Artifact

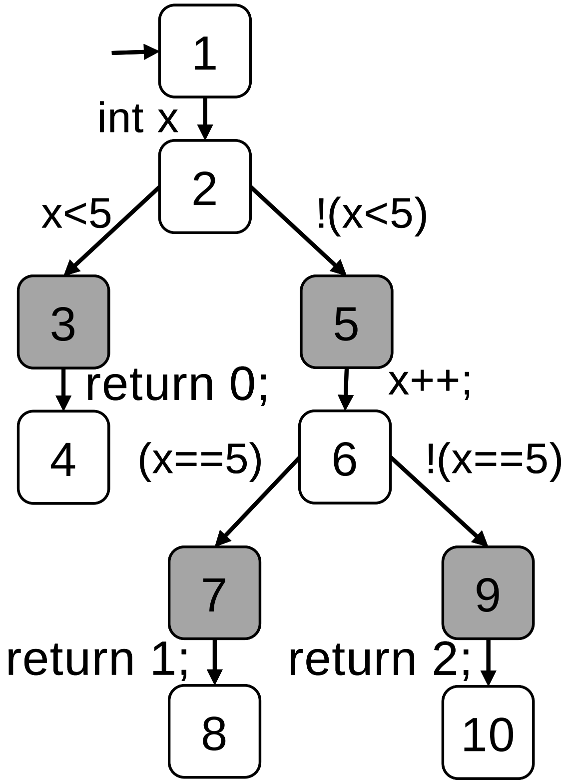


- **Advantage**: Format usable independent of context
- Artifact is required to encode:
  - Feasible path guaranteed reaching a target nodes
  - Infeasible path or path guaranteed reaching no target node
  - Candidates for the former

→ No existing format fulfills all requirements

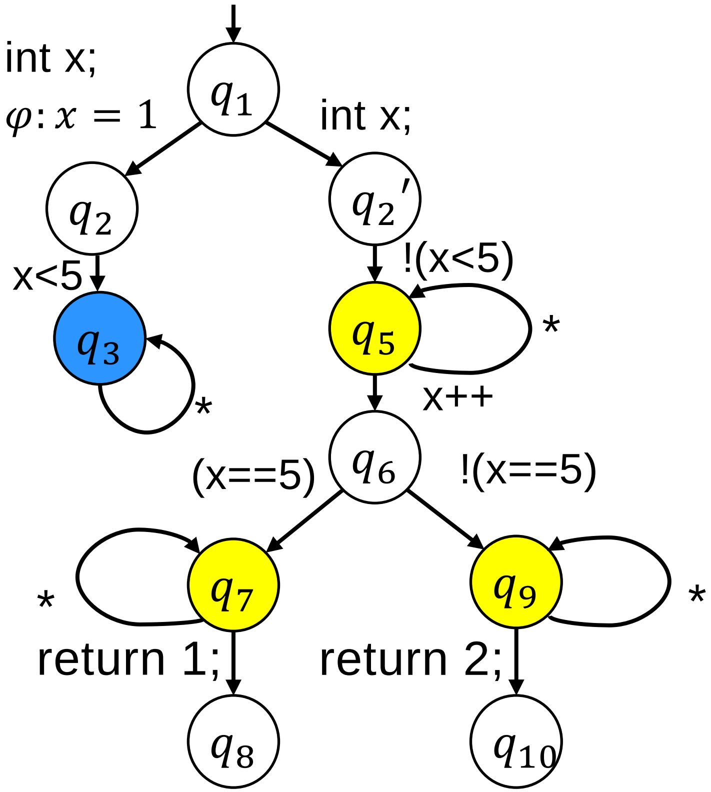
Target Node

Target Nodes encode the task for the tool → For test case generation, L = {3,5,7,9}



CFA

Generalized Information Exchange Automaton (GIA)

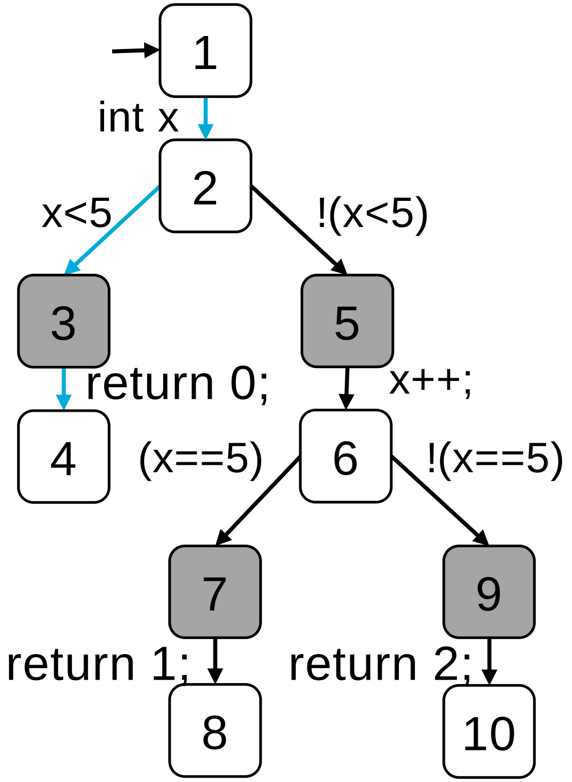


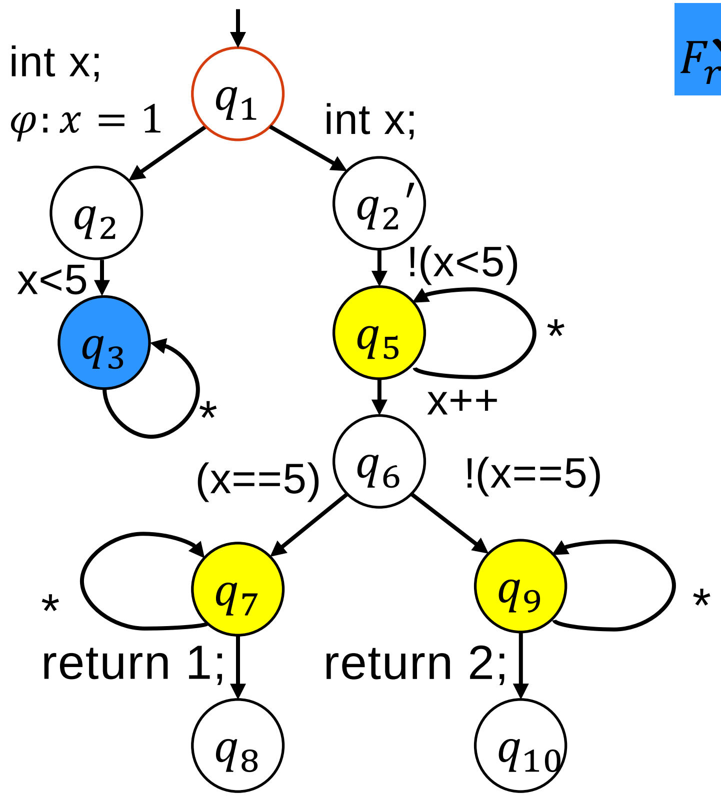
$F_{rt}^{✓} = \{q_3\}$, $F_{ut}^{✗} = \emptyset$, $F_{cand}^{?} = \{q_5, q_7, q_9\}$

A GIA has three pairwise disjoint sets of *accepting states:*

- $F_{rt}^{✓}$ (for reachable targets),
- $F_{ut}^{✗}$ (for unreachable targets) and
- $F_{cand}^{?}$ (for candidates)

Semantics of a GIA – Covered Paths





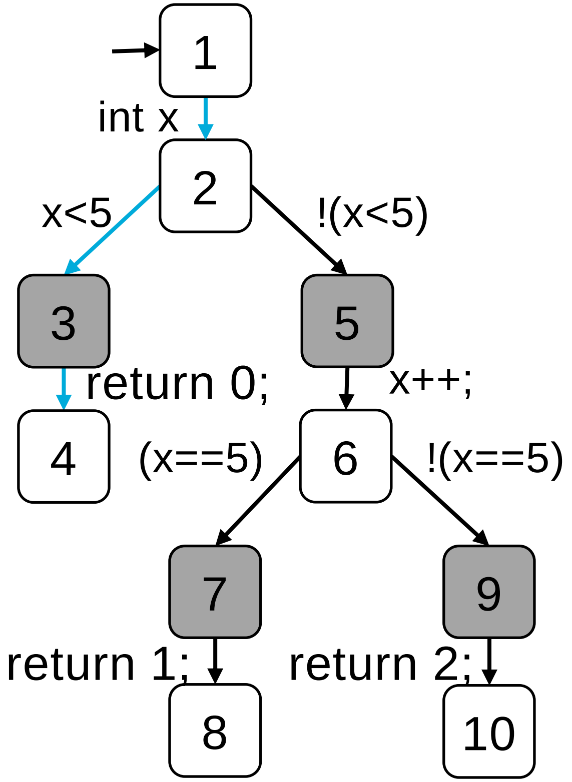
$F_{rt}^{✓} = \{q_3\}, F_{ut}^{✗} = \emptyset$

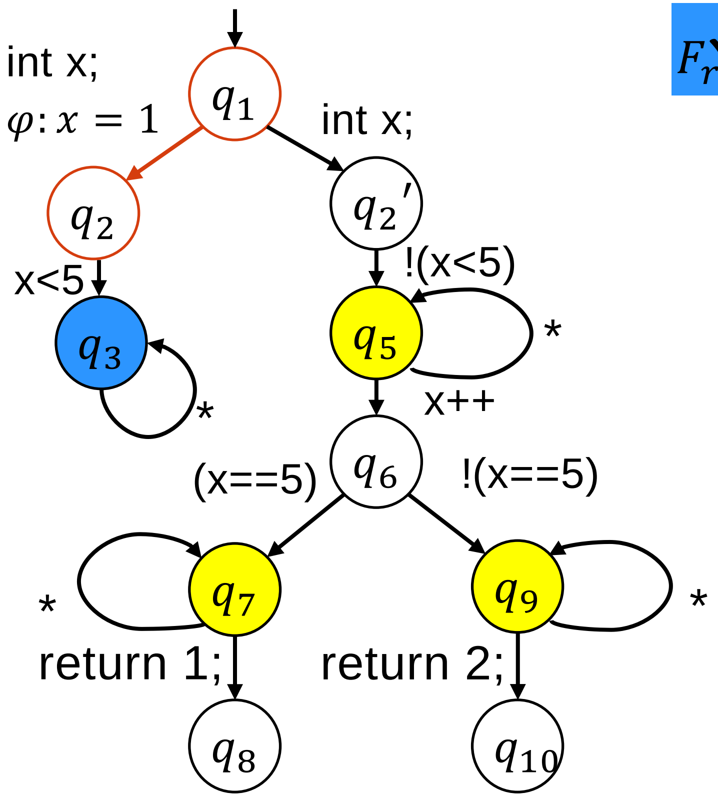
$$\pi_1 = \langle (x \mapsto 1), \ell_1 \rangle \xrightarrow{int\ x} \langle (x \mapsto 1), \ell_2 \rangle \xrightarrow{x<5} \langle (x \mapsto 1), \ell_3 \rangle \xrightarrow{return\ 0} \langle (x \mapsto 1), \ell_4 \rangle$$

$$\rho_1 = (q_1, t)$$

Semantics of a GIA – Covered Paths

$F_{rt}^{\checkmark} = \{q_3\}, F_{ut}^{\times} = \emptyset$

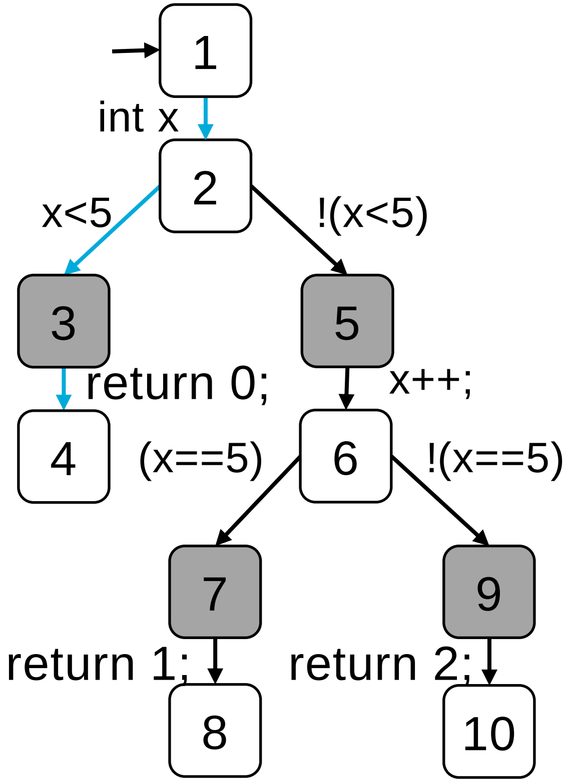


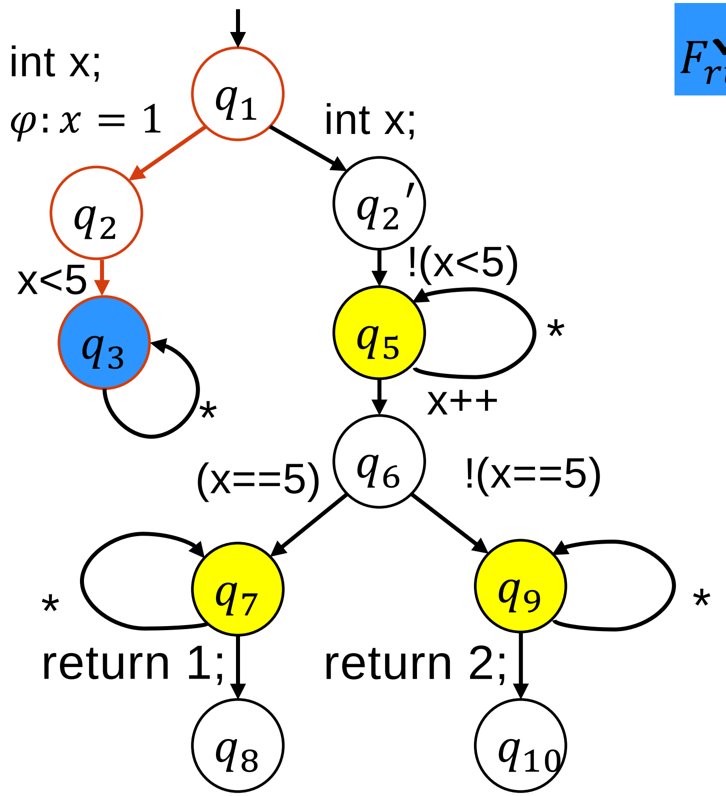


$$\pi_1 = \langle (x \mapsto 1), \ell_1 \rangle \xrightarrow{int\ x} \langle (x \mapsto 1), \ell_2 \rangle \xrightarrow{x<5} \langle (x \mapsto 1), \ell_3 \rangle \xrightarrow{return\ 0} \langle (x \mapsto 1), \ell_4 \rangle$$

$$\rho_1 = (q_1, t) \xrightarrow{(int\ x,\ x=1)} (q_2, t)$$

Semantics of a GIA – Covered Paths



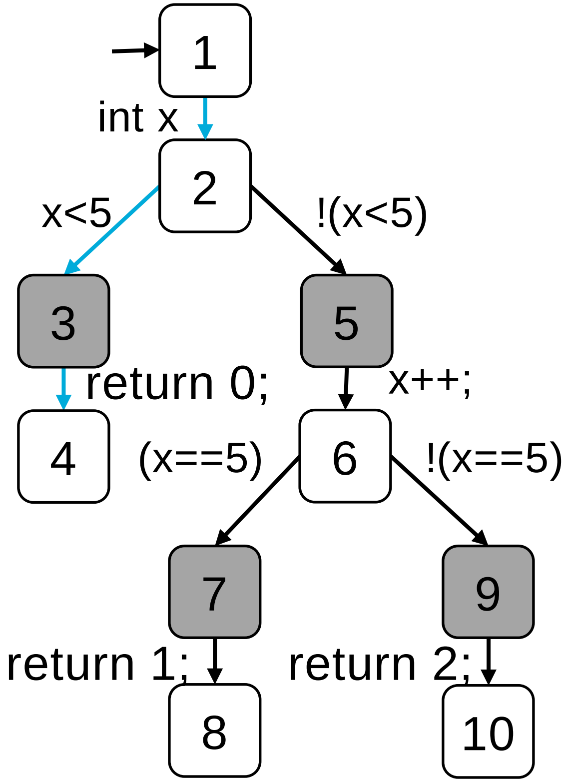


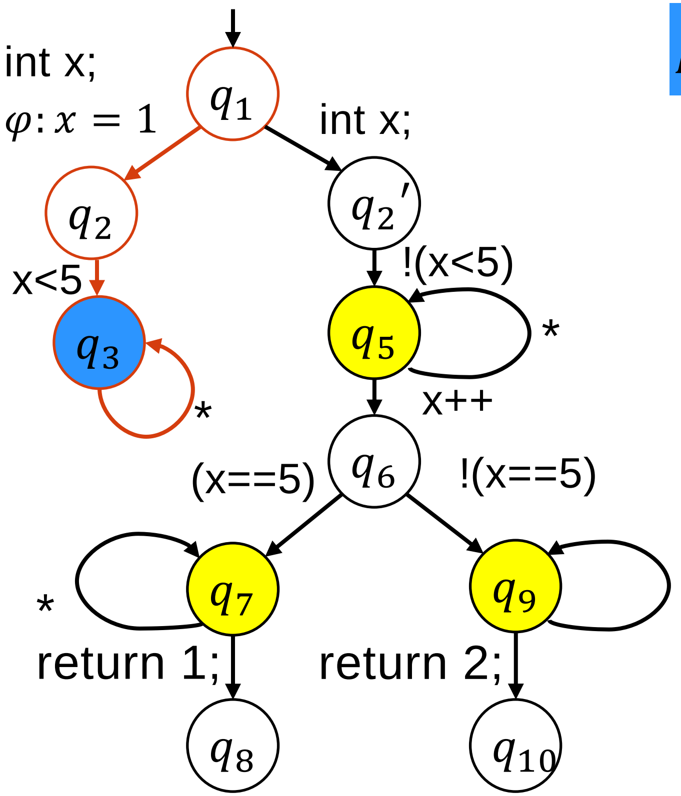
$F_{rt}^{✓} = \{q_3\}$, $F_{ut}^{✗} = \emptyset$

$$\pi_1 = \langle (x \mapsto 1), \ell_1 \rangle \xrightarrow{int\ x} \langle (x \mapsto 1), \ell_2 \rangle \xrightarrow{x<5} \langle (x \mapsto 1), \ell_3 \rangle \xrightarrow{return\ 0} \langle (x \mapsto 1), \ell_4 \rangle$$

$$\rho_1 = (q_1, t) \xrightarrow{(int\ x,\ x=1)} (q_2, t) \xrightarrow{(x<5, true)} (q_3, t)$$

Semantics of a GIA – Covered Paths





F_{rt} ✓ $= \{q_3\}$, F_{ut} ✗ $= \emptyset$

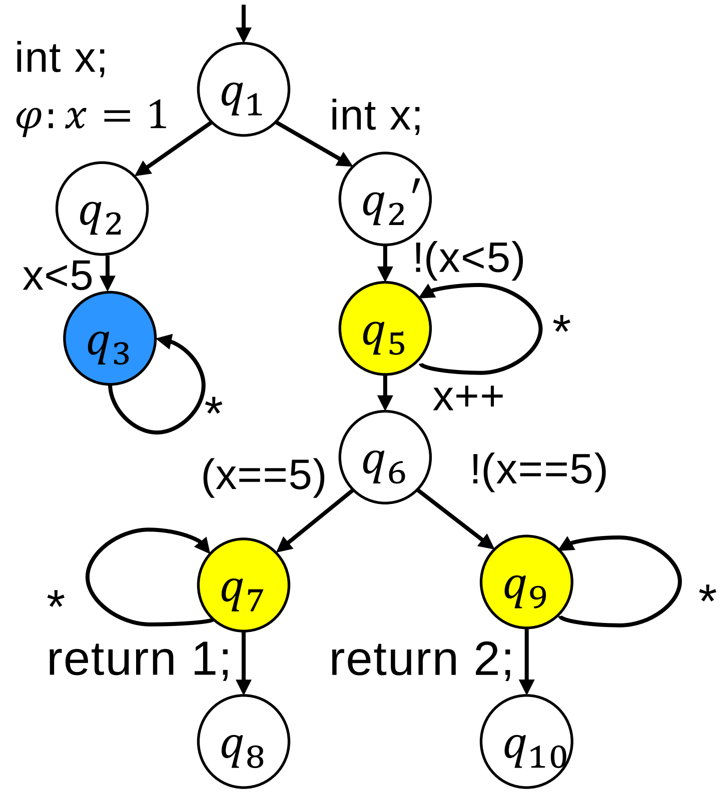
→ A covers π_1

$$\pi_1 = \langle (x \mapsto 1), \ell_1 \rangle \xrightarrow{int\ x} \langle (x \mapsto 1), \ell_2 \rangle \xrightarrow{x<5} \langle (x \mapsto 1), \ell_3 \rangle \xrightarrow{return\ 0} \langle (x \mapsto 1), \ell_4 \rangle$$

$$\rho_1 = (q_1, t) \xrightarrow{(int\ x,\ x=1)} (q_2, t) \xrightarrow{(x<5, true)} (q_3, t) \xrightarrow{*} (q_3, t)$$

(ρ_1 is a run of A)

Semantics of a GIA – Language and Correctness

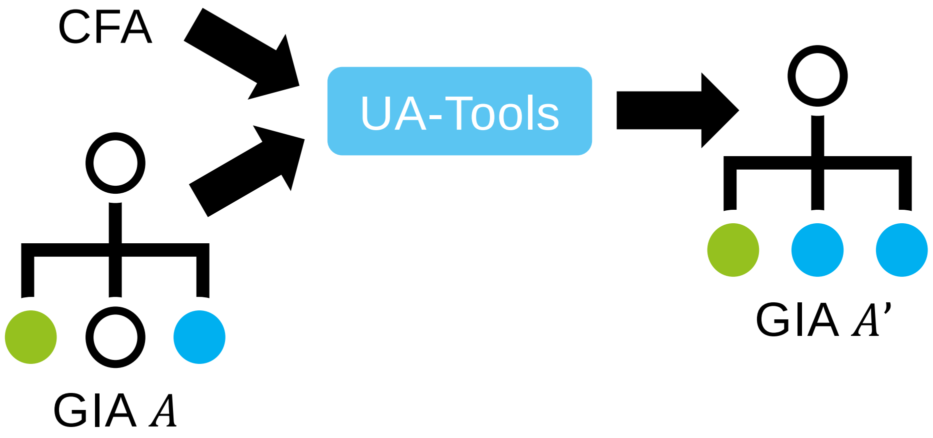


Language of A:

- $P_{rt}^{\checkmark}$ =paths covered and last state in $F_{rt}^{\checkmark}$
- $P_{ut}^{\times}$ =paths covered and last state in $F_{ut}^{\times}$
- $P_{cand}^{?}$=paths covered and last state in $F_{cand}^{?}$

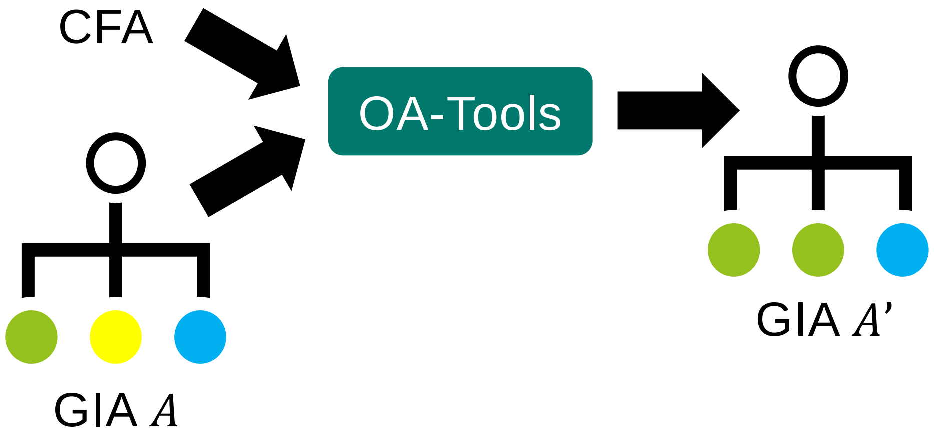
Sound OA- and UA-Tools

Underapproximating tool:



$P_{rt}^{\checkmark}(A) \subseteq P_{rt}^{\checkmark}(A')$ and $P_{ut}^{\times}(A) = P_{ut}^{\times}(A')$

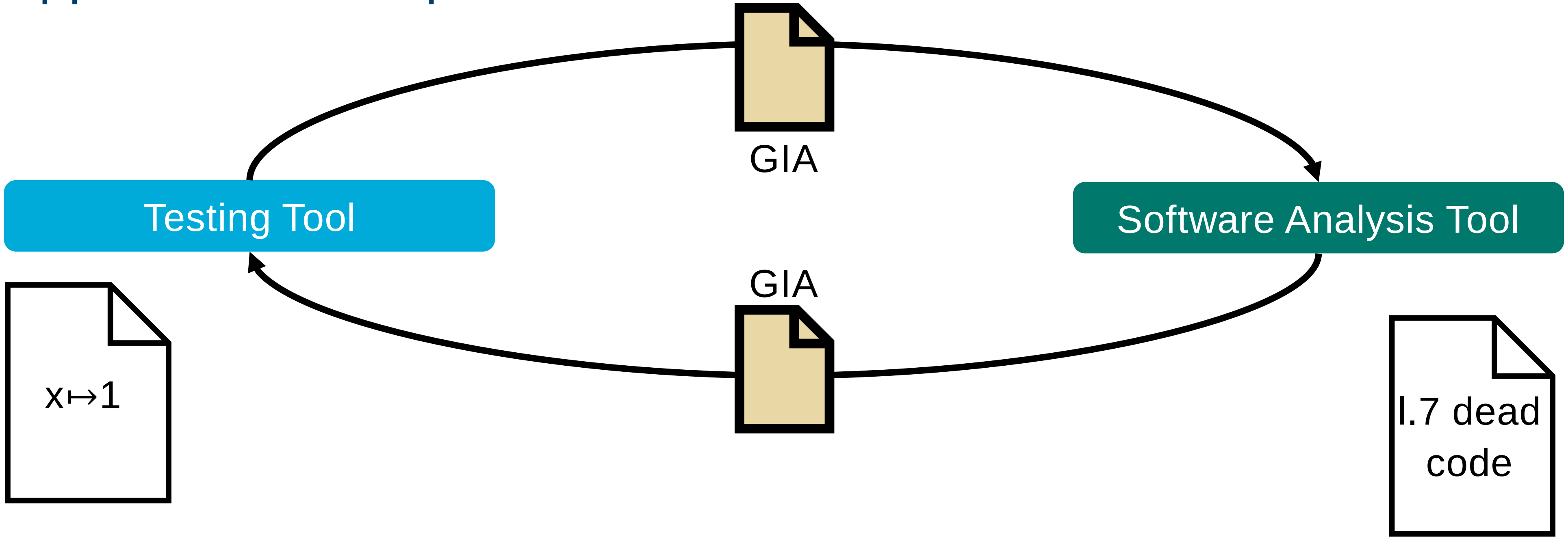
Overapproximating tool:



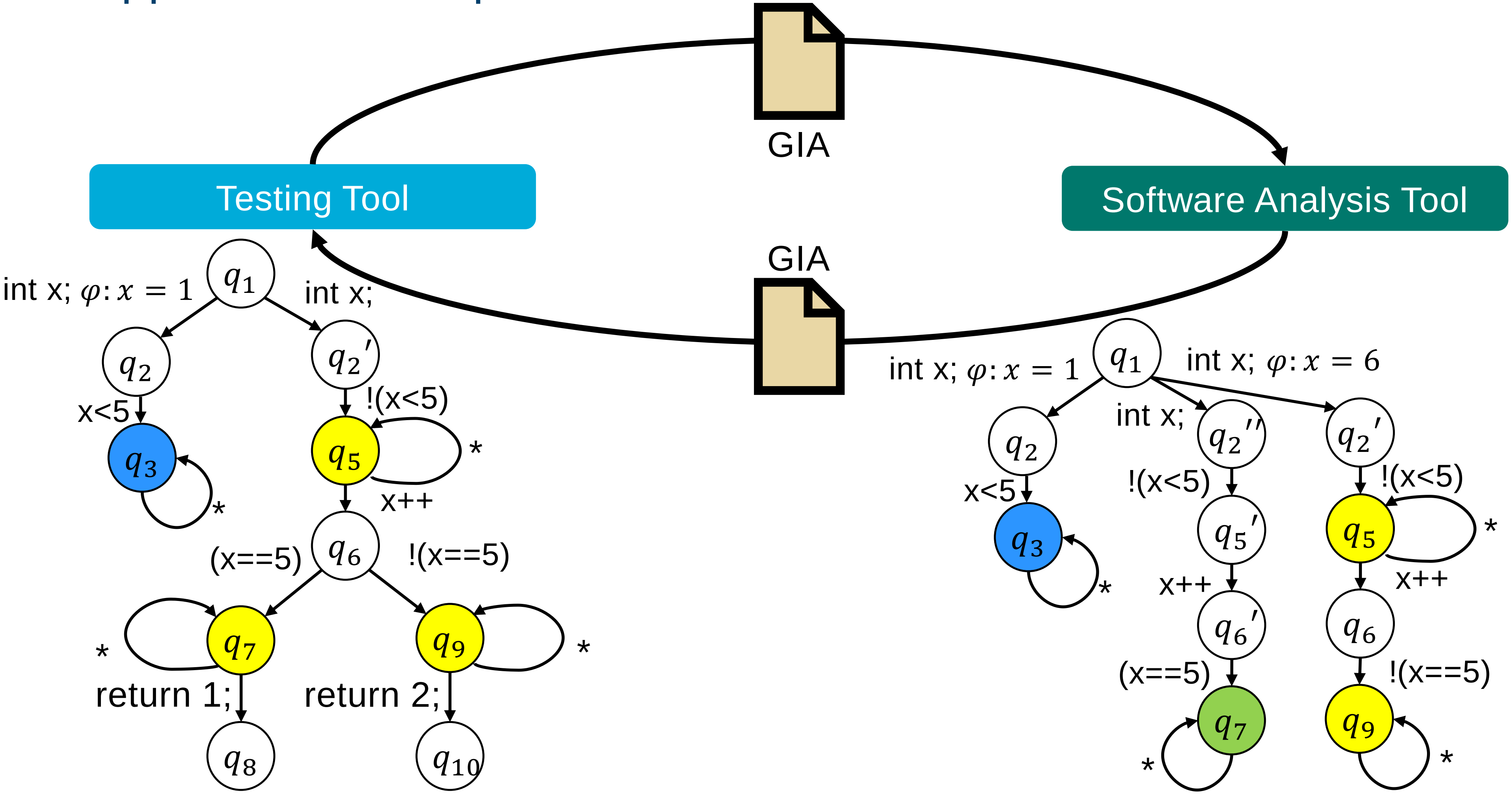
$P_{rt}^{\checkmark}(A) = P_{rt}^{\checkmark}(A')$ and $P_{ut}^{\times}(A) \subseteq P_{ut}^{\times}(A')$

Carl von Ossietzky Universität Oldenburg

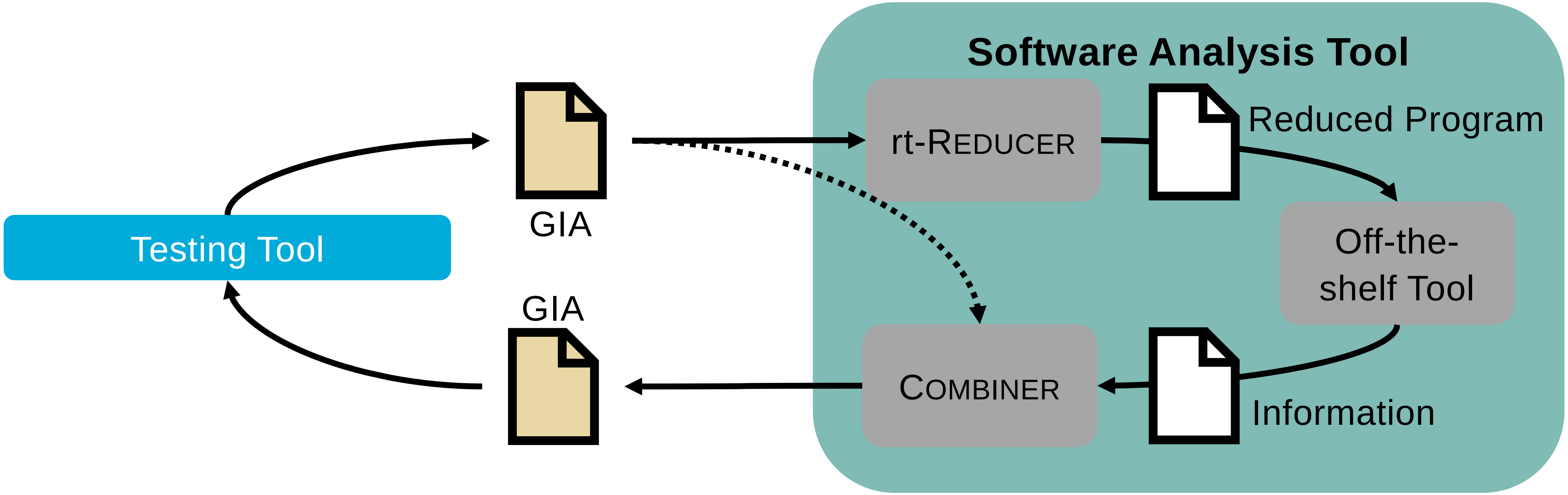
Applications: Cooperative Test Case Generation



Applications: Cooperative Test Case Generation



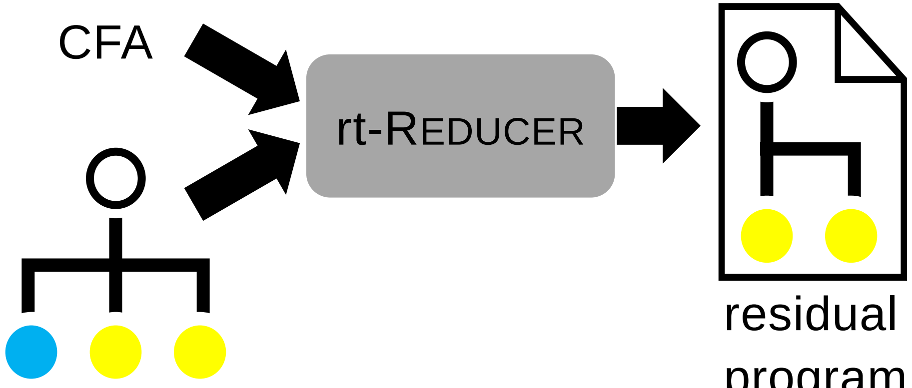
Using Off-the-shelf Tools



rt-REDUCER and COMBINER

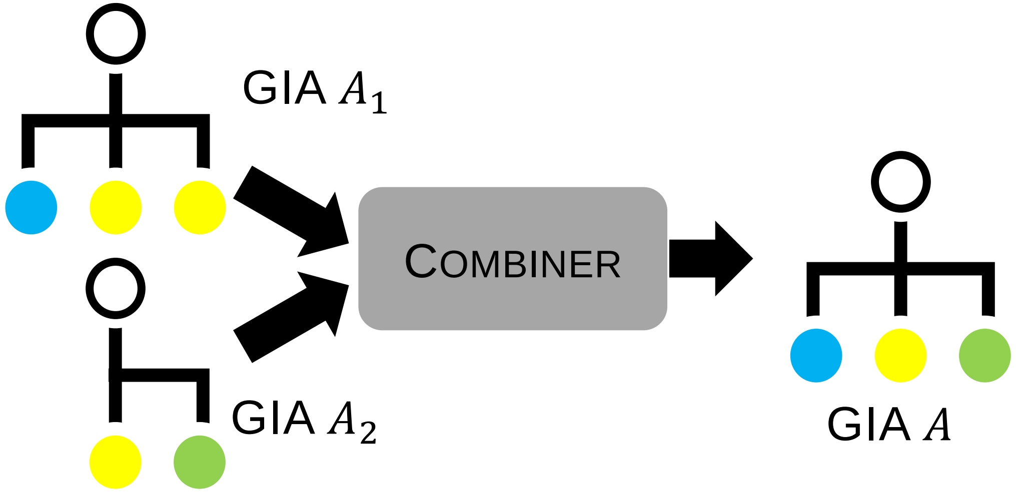
rt-REDUCER

- Reduces program w.r.t. reachable target nodes
- Generates a residual program
- Based on existing reducer [1]



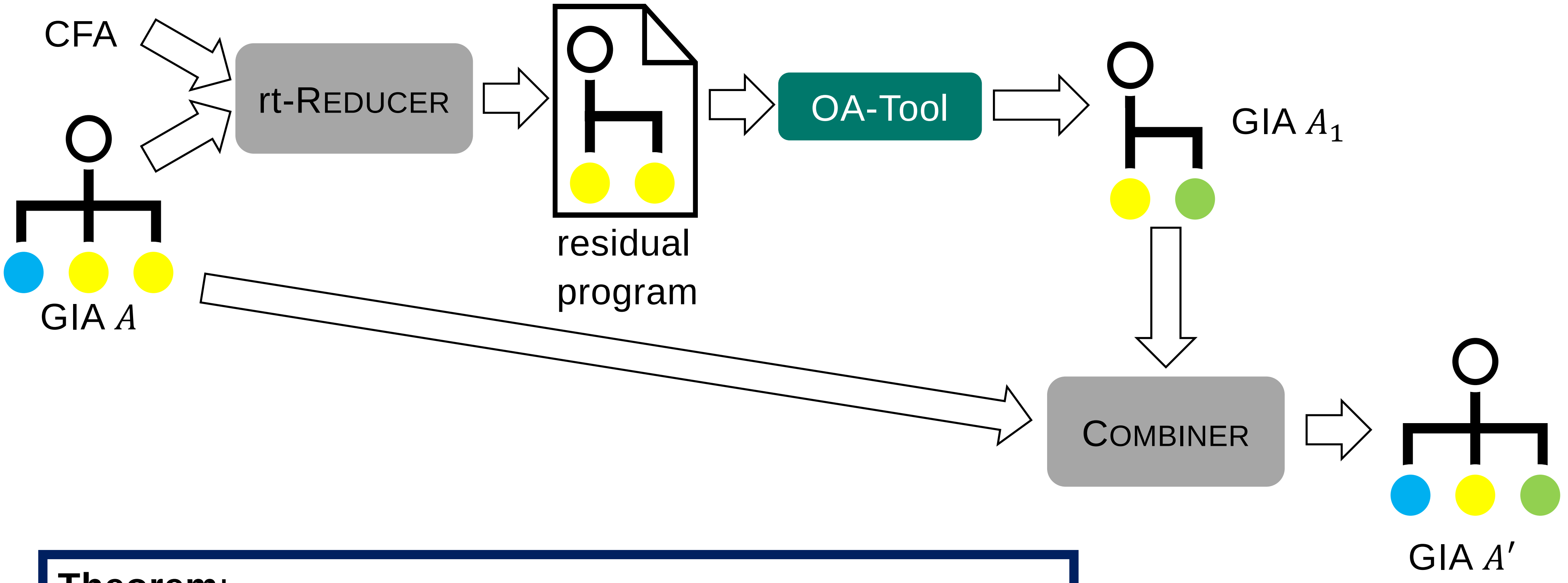
COMBINER

- Combines two GIA
- No information is lost nor added
- Retains more precise information for $P_{cand}(A)$



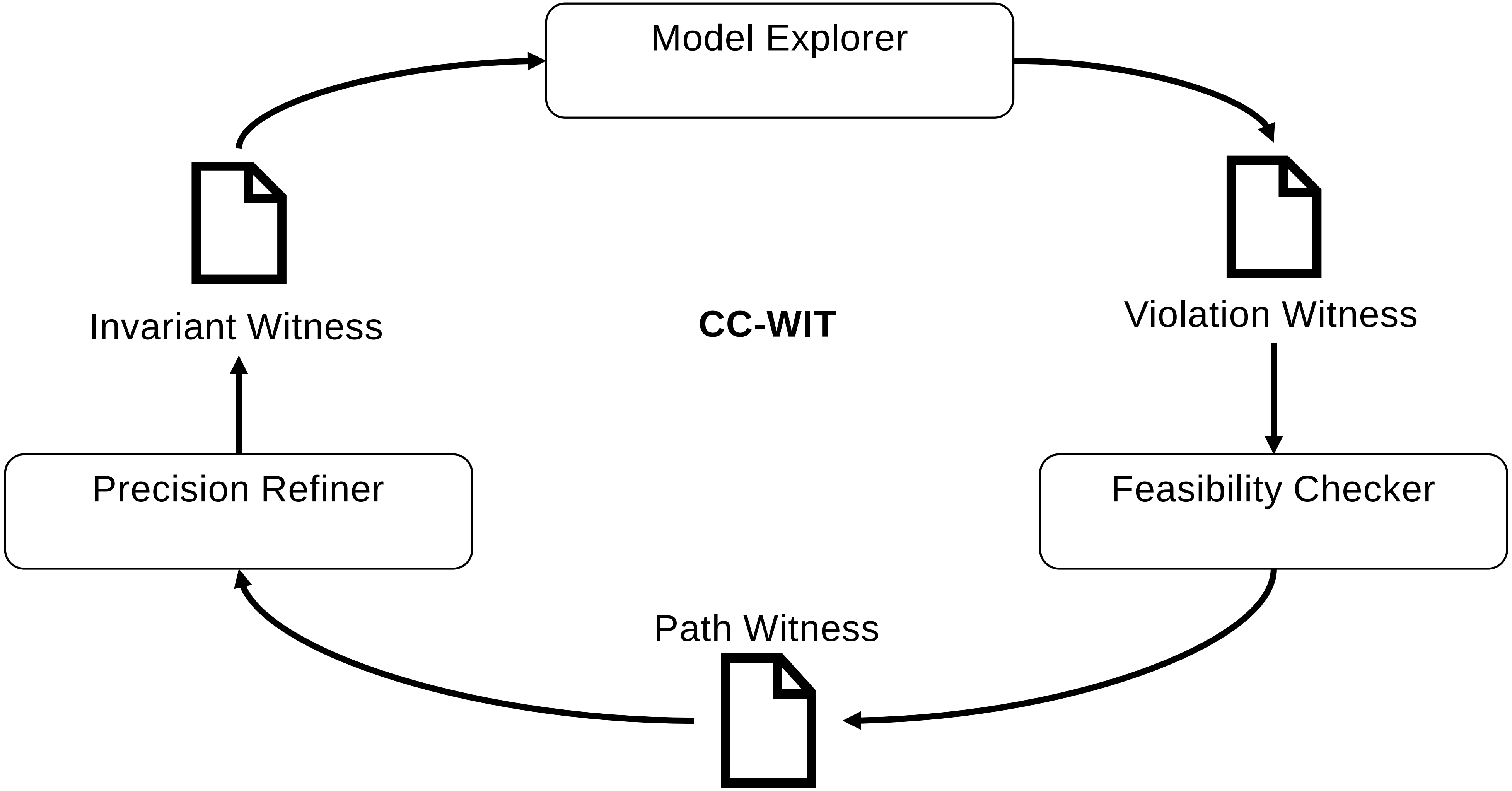
[1] Beyer, D., Jakobs, M., Lemberger, T., Wehrheim, H.: Reducer-based construction of conditional verifiers. In: Proc. ICSE18.

Application of REDUCER and COMBINER retains all information



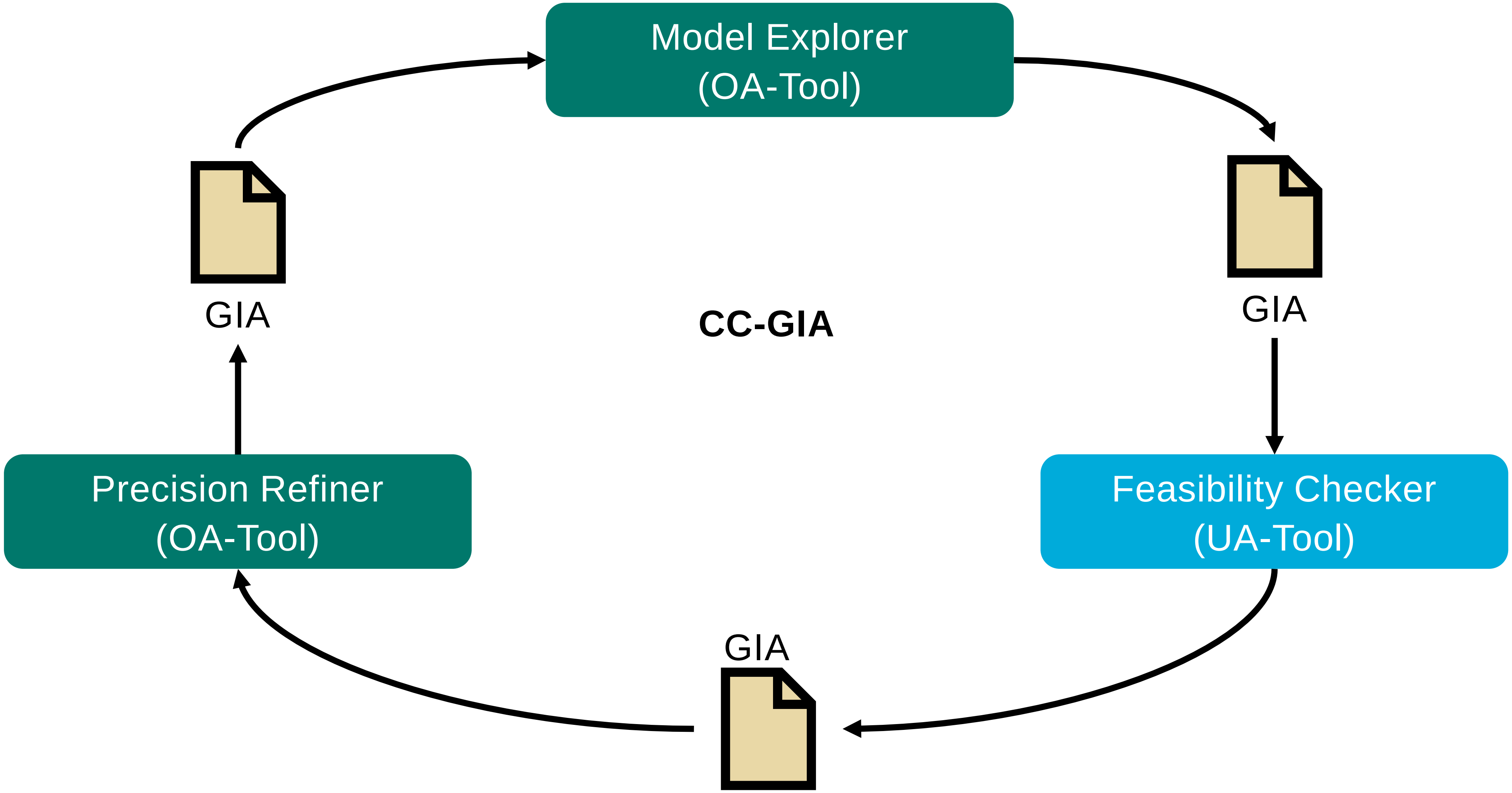
Theorem:
$P_{rt}^{\checkmark}(A') = P_{rt}^{\checkmark}(A) \wedge P_{ut}^{\times}(A') \supseteq P_{ut}^{\times}(A)$ if OA-Tools is sound

Usecase of GIAs: Cooperative Verification using C-CEGAR



[2] Beyer, D., Haltermann, J., Lemberger, T., Wehrheim, H.: Decomposing Software Verification into Off-the-Shelf Components: An Application to CEGAR. In: Proc. ICSE22.

Usecase of GIAs: Cooperative Verification using C-CEGAR

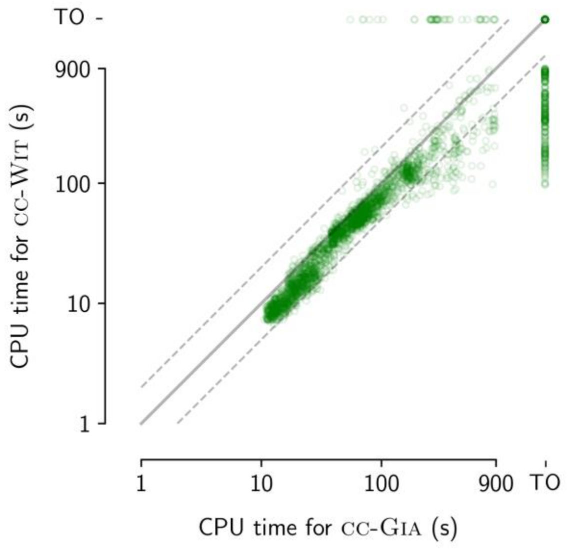


Carl von Ossietzky Universität Oldenburg

Implementation and Evaluation

- C-CEGAR using GIA implemented in CPACHECKER and COVERITEAM
- Goal: Show GIA are feasible as exchange format and exemplify advantages
- Dataset used: SV-Benchmarks as in SV-COMP'22

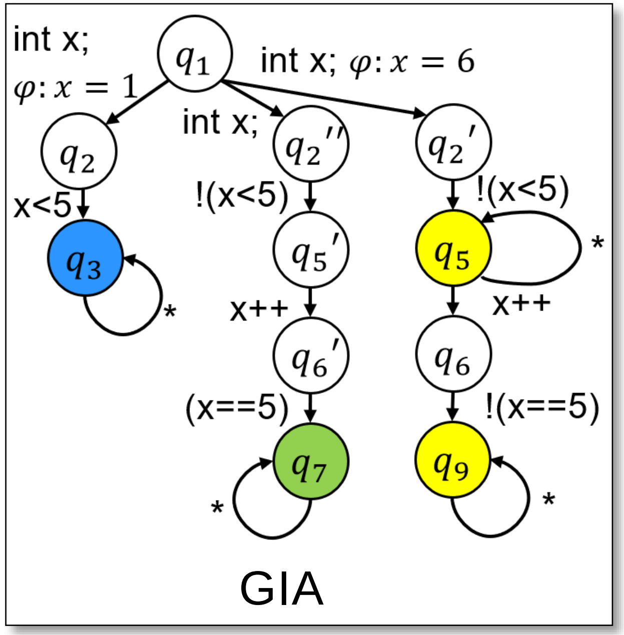
Result	CC-GIA	CC-WIT
Correct overall	2641	2819
Correct proof	2068	2100
Correct alarm	573	719
Add. Solved	**114**	-
incorrect	5	7

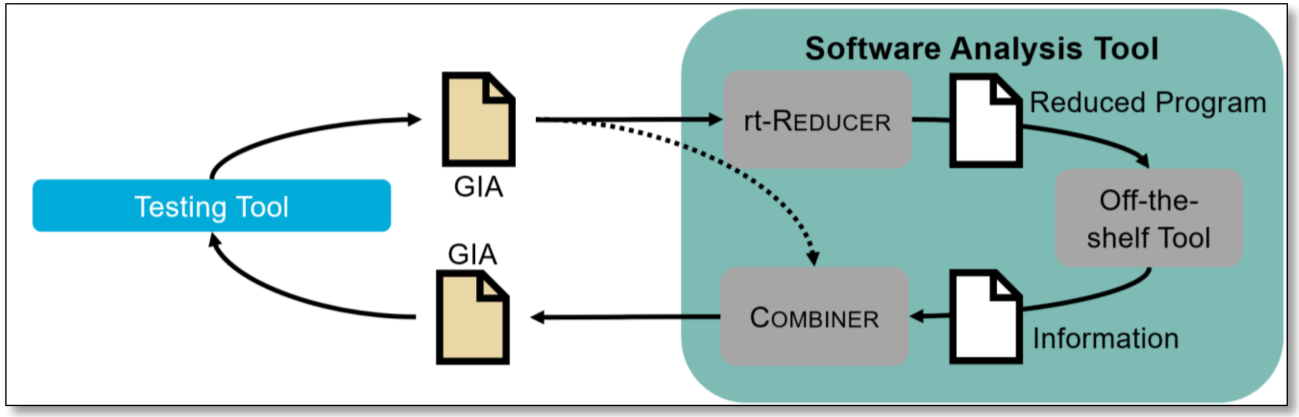


→ Flexible, precises and practically applicable for C-CEGAR

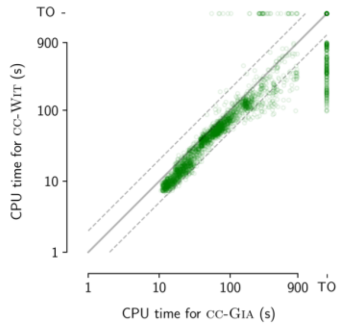
Carl von Ossietzky Universität Oldenburg

Summary and Future Work





Result	CC-GIA	CC-WIT
Correct overall	2641	2819
Correct proof	2068	2100
Correct alarm	573	719
Add. Solved	114	-
incorrect	5	7



Future work

Application of GIA in different settings as:

- Cooperative test case generation
- Conditional model checking
- Parallelization