Trace Reasoning for Automated Full First-order Program Verification

Pamina Georgiou, Bernhard Gleiss, Laura Kovács

Automated Program Reasoning Group

TU WIEN

MOTIVATION

Limitations of automated software verification

- restricted expressiveness
- imprecision (due to approximations)
- requiring expert knowledge for
  - finding invariants or
  - generating proofs

GOAL

Automated Theorem Proving (ATP) for Software Verification

Full First-Order Logic with Theories: Superposition-Based ATP

Program semantics that allow expressing program variables over program locations and **timepoints**: **RAPID** Framework.

Automating Inductive Reasoning over Loops with **Trace Lemmas**: necessary instantiations of the induction axiom to automate proof search

Verification Task

Imperative Program

Reachability or Relational Property

RAPID Framework

Custom Program Semantics

Automatically generated Trace Lemmas

Vampire Theorem Prover

Discharged Verification Conditions = hopefully a Proof

```
func main() {
  Int[] a;
  const Int alength;
  const Int v;

  Int i = 0;
  while (i < alength) {
    a[i] = v;
    i = i+1;
  }
}
```

Let's prove that the array *a* is initialized with some integer value *v*...

$$\forall pos_{\mathbb{I}} \,.\, (0 \leq pos \wedge pos < alength \wedge alength > 0) \rightarrow a(end, pos) = v$$

What the prover doesn't know is that due to the incrementation of *i* from *0* to *(alength - 1)*, every position of *a* in this range will be affected by exactly one of the loop iterations.

What the prover needs to know to show it is that *i* will have the value of every integer in this range at least once, formalized by the following **trace lemma:**

$$\forall x_{\mathbb{I}} \,.\, \Big((i(l_7(zero)) \leq x \wedge x < i(l_7(lastIt)) \wedge \forall it_{\mathbb{N}} \,.\, i(l_7(s(it))) = i(l_7(it)) + 1 \Big) \rightarrow \exists it_{\mathbb{N}} \,.\, i(l_7(it)) = x \wedge it < lastIt$$

CONCLUSION

PROGRAM SEMANTICS WITH TIMEPOINTS ARE POWERFUL: LOOP NESTING, RELATIONAL VERIFICATION.

ALLOW FOR **AUTOMATED INDUCTIVE REASONING** IN COMBINATION WITH TRACE LEMMAS.

TRACE LEMMAS: AUTOMATICALLY GENERATED INDUCTIVE PROPERTIES FOR PROGRAM VARIABLES INSTEAD OF PROGRAM-SPECIFIC LOOP INVARIANTS