Transfer of Methods Between Different Schools of Verification

Marek Jankola

marek.jankola@sosy.ifi.lmu.de

Supervisors: Dirk Beyer

Collaborators: Po-Chun Chien, Nian-Ze Lee, Marian Lingsch-Rosenfeld, Jan Strejček

CONVEY

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Motivation

The model checking problem $M \models \varphi$ varies in different contexts:

- M: transition system, C program, hardware circuit, etc.
- φ: general LTL formula, safety property, termination, etc.
- *The Goal*: answer the problem, provide proof/certificate, provide the shortest counterexample, etc.

Depending on the context, different methods were developed across the verification schools. Can we transfer some of these methods between the schools to solve the problem in a different context?

The Shortest Counterexample (FoSSaCS 2024)

If φ is a safety property, finding the shortest counterexample reduces to finding the shortest violation trace in the transition system. For specifications given by LTL or ω-automata, we need an infinite counterexample.

- The shortest counterexample: $cab(abab)^\omega \to c(ab)^\omega$
- If **Tight** automata are used for the specification, finding the shortest counterexample **reduces** to finding the shortest path.
- We proposed a more effective way to construct these automata and show how hard it is theoretically, to reduce this problem:

$$2^{\Omega(n)} \prec \Omega\left(\frac{n-1}{2}!\right) \longleftrightarrow \mathcal{O}(n! \cdot n^3) \prec \mathcal{O}((\sqrt{2}n)^{2n})$$

Hardware Circuit → C Program (FSE 2024)

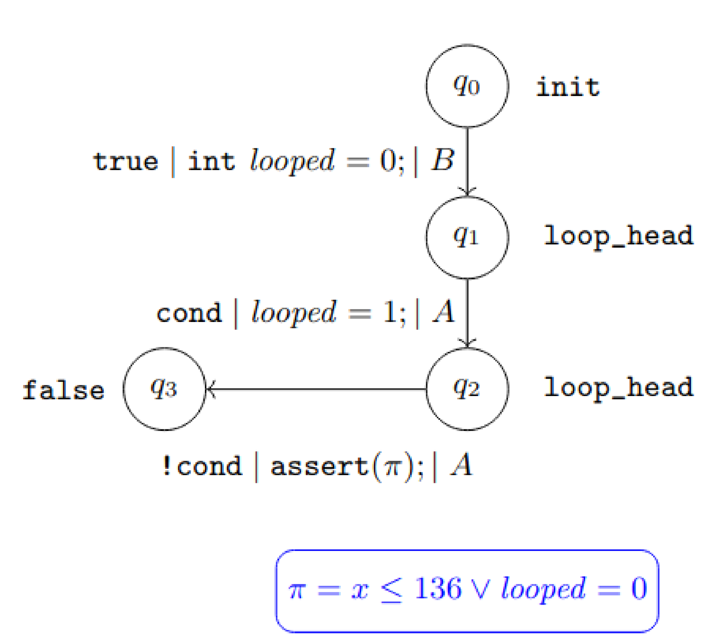
- Multiple verification algorithm were invented first for hardware and later adapted to software.
- There are two algorithms ISMC and DAR that were not adapted before.
- We made a systematic transferability study for ISMC and DAR:

	Hypothesis from ISMC Paper	Confirmed
H1.A	ISMC faster in finding bugs	✓
H1.B	ISMC faster in proving property if high unrolling bound	?
H1.C	ISMC overall faster	?

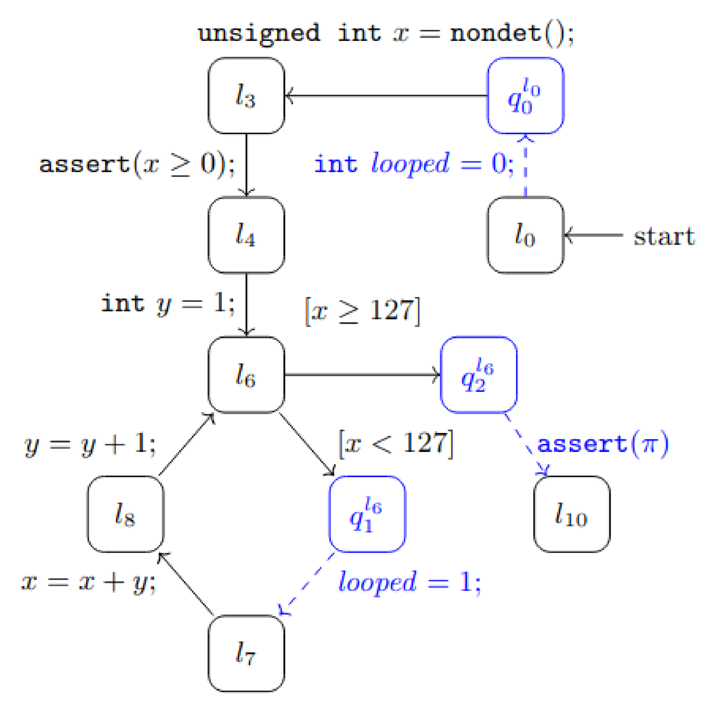
	Hypothesis from DAR Paper	Confirmed
H2.A	DAR performs more local phases than global	✓
H2.B	DAR faster in proving property	?
H2.C	DAR computes more interpolants	✓
H2.D	DAR's runtime more sensitive to sizes of interpolants	?
H2.E	DAR overall faster than IMC	?

Specifications → Reachability (SPIN 2025)

Numerous well-performing methods can verify a program against reachability property. Our modular framework based on instrumentation automata takes a control-flow of a program and transforms it in a way, that it can be verified against reachability.



(a) An instrumentation automaton



(b) The CFA after sequentialization

Tools with *-R* suffix are reachability analyzers. The comparison against termination verifiers:

Table 4: Summary of the results for transformed 377 termination tasks

Results (#Tasks)		UAutomizer	2ls	UAutomizer-R	CPAchecker-R
Correct	377	312	259	**333**	121
Proofs	264	250	189	**264**	55
Alarms	69	62	**70**	69	66

Reachability Witness ↔ Termination Witness

Reduction from termination to reachability via transformation is a known approach to termination analysis. We show that the construction and validation of termination witnesses can be reduced with a similar construction. $P = (X, S, R, Init)$ is a program and P' the transformed program:

Definition 1: Invariant

Formula $I(s)$ of P' is

- *1-step invariant* if $(\exists s' \in \mathscr{R} : R^+(s', s)) \implies I(s)$
- *safe* if $I(s) \implies saved = 0 \vee \bigvee_{x \in X} s(x) \neq s(x')$.

Definition 2: Transition Invariant

$T(s, s')$ is a transition invariant of P if $R^+ \subseteq T$. Program P is terminating iff there exists well-founded transition invariant.

Theorem 1: Transition Invariants ↔ Invariants

$I(\hat{x}_0, \ldots, \hat{x}_n, x_0, \ldots, x_n)$ is a safe 1-step invariant of P' iff it is a well-founded transition invariant of P.

CONVEY Evolving Systems

This work was supported by European Union's Horizon Europe program under the grant agreement No. 101087529, the Deutsche Forschungsgemeinschaft (DFG) – 378803395 (ConVeY) and Czech Science Foundation grant GA23-06506S