Parameterized Verification of Asynchronous Round-Based Distributed Algorithms

via Reduction to Finite-counter Systems

Nathalie Bertrand, Univ. Rennes, Inria, CNRS, IRISA, France
Pranav Ghorpade, University of Sydney, Australia
Sasha Rubin, University of Sydney, Australia

CONCUR 2026

Asynchronous round-based distributed algorithms

n processes, t faults with $t < n/2$.

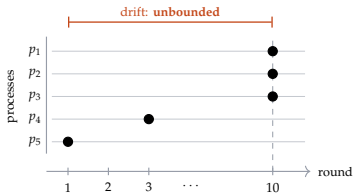
Each process in round $r \in \mathbb{N}$:

1. BROADCAST messages tagged r
2. WAIT to receive enough messages tagged r
3. MOVE within round r, or JUMP to a round $r' > r$

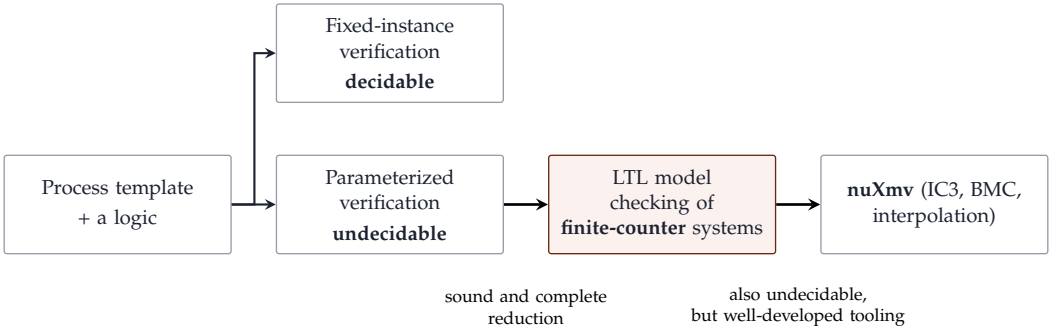
Shape of standard consensus (Ben-Or, Bracha) and leader election (Raft leader) algorithms

Notice

- these algorithms are parameterized
- each process is infinite-state
- under asynchronous communication, processes drift



Overview of results



Our reduction lets us perform parameterized verification using state-of-the-art model checkers.

Case studies using nuXmv

Algorithm	safety properties	liveness properties	time
Ben-Or, crash faults	agreement, validity	restricted termination	$< 4\,s$
Ben-Or, Byzantine faults	agreement, validity	restricted termination	$< 8\,s$
Bracha, Byzantine faults	agreement, validity	restricted termination	$< 15\,s$
Raft leader election	leader uniqueness	N/A	$2\,s$

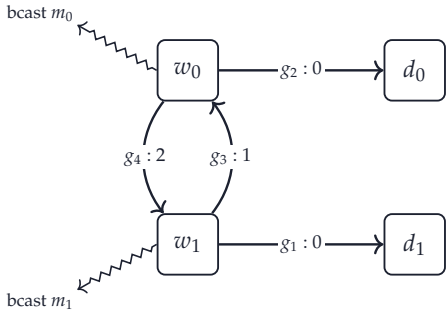
Every property verified in under 15 s.

Process templates for specifying algorithms

Process template captures the behavior of one process in one round.

It is a control graph with:

- finitely many control locations
- finitely many message types with a possible broadcast at each location
- transitions labeled with a guard and a jump length



$$g_1 \quad := \quad \#m_1 \quad > \quad \frac{n}{2}$$

HSCL for specifying correctness

HSCL = History State Count Logic

- Formulas are built over integer variables $\mathsf{count}_r(\ell)$
  – the number of processes that have visited location ℓ in round r.

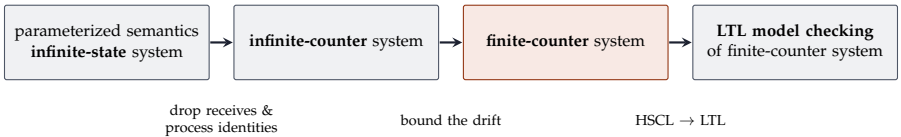
HSCL formulas are Boolean combinations of the following two types of atoms:

- $\forall_{\mathbf{r}}\ \mathsf{count}_r(\ell_0) \leq 1$ (round local atom)
- $\sum_{\mathbf{r}}\ \mathsf{count}_r(\ell_1) < n$ (cumulative atom)

HSCL does not observe:

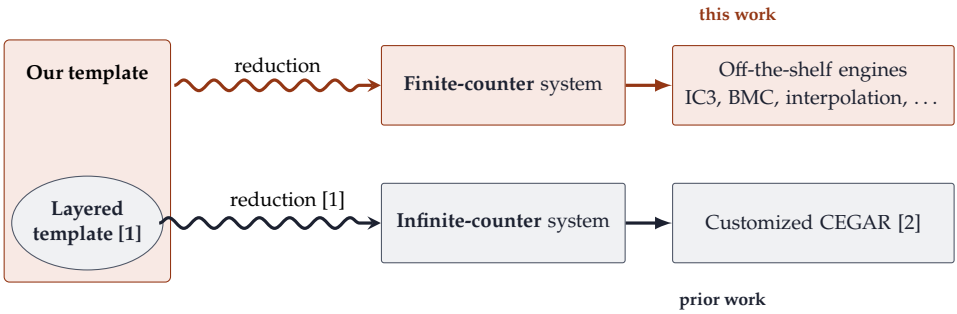
- which processes
- in what order
- which messages they received that enabled the transitions.

The reduction



- Treat receive steps as stuttering steps.
- Forget which process is in which state, instead store how many processes are in each state.
- Restrict verification to executions with bounded drift and prove it is complete.
- Translate HSCL formulas to LTL formulas over reduced system.

Comparison to closest prior work



[1] Bertrand, Thomas, and Widder CONCUR'21
[2] Thomas and Sankur TACAS'23

Conclusion

Summary

- We study parameterized verification of asynchronous round-based distributed algorithms
- Give a reduction to LTL model checking of finite-counter systems
- Show the reduction is practical using experimental evaluation

Future work

Generalize the reduction to

- more expressive logic (for almost-sure termination)
- more expressive template (for DAG-based consensus)

Nathalie Bertrand, Pranav Ghorpade, Sasha Rubin

Thank you!