Andrey Kupriyanov
Bernd Finkbeiner

Reactive Systems Group
Saarland University

Causality-based LTL Model Checking without Automata

Causality

A relationship between two events, when the first event is recognized as a necessary prerequisite for the occurrence of the second. Unrolling of the program causality relation can be exponentially more succinct than its state space.

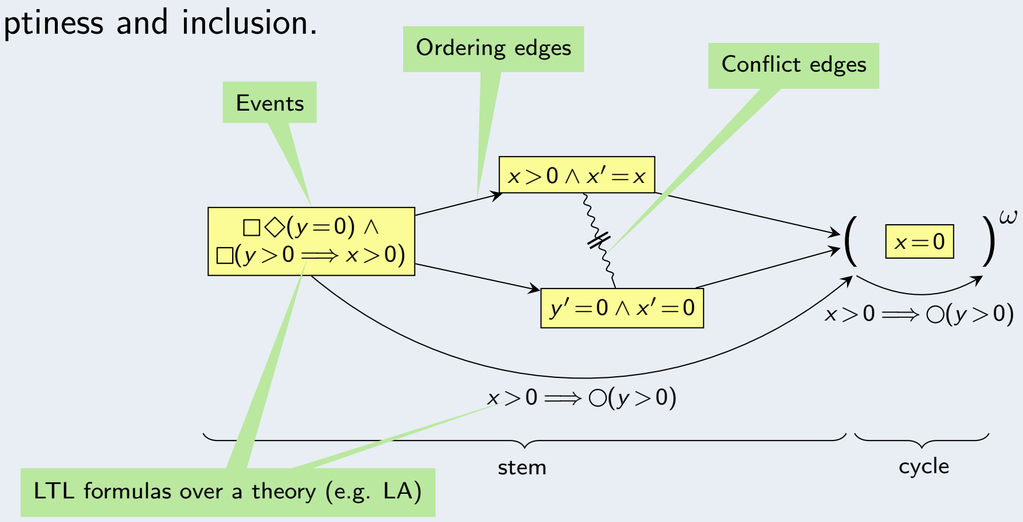
Our Approach

We consider the problem of verifying LTL properties for infinite-state concurrent programs. Our approach can be seen as a *proof by contradiction* technique. We assume that the property is violated, and follow causal consequences from that assumption.

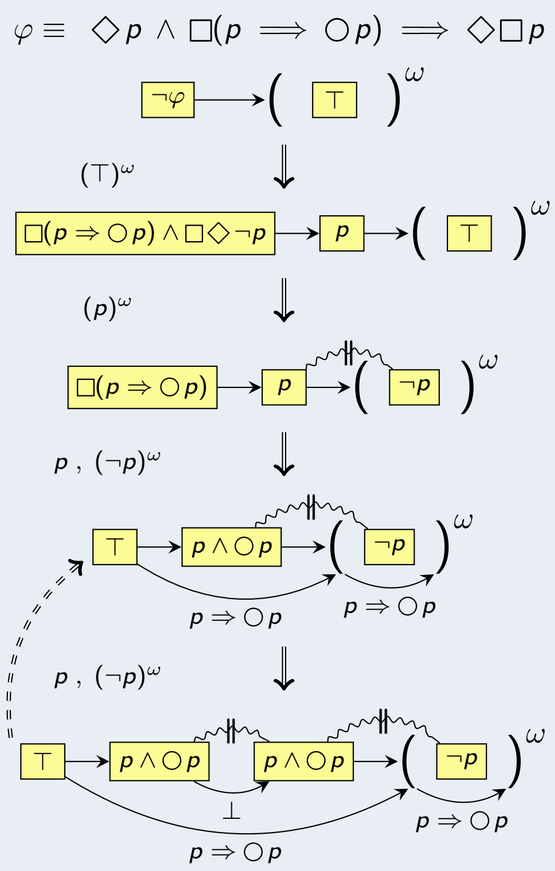
- We represent sets of potential counterexample runs as ***concurrent traces***.
- The search is organized into a ***trace tableau***, where vertices are labeled with concurrent traces. The root trace represents all possible counterexamples.
- We unroll the trace tableau by applying causality-based ***proof rules***.
- The property is proven when all tableau leaves are either contradictory, or can be covered by the premises of other vertices in the tableau.

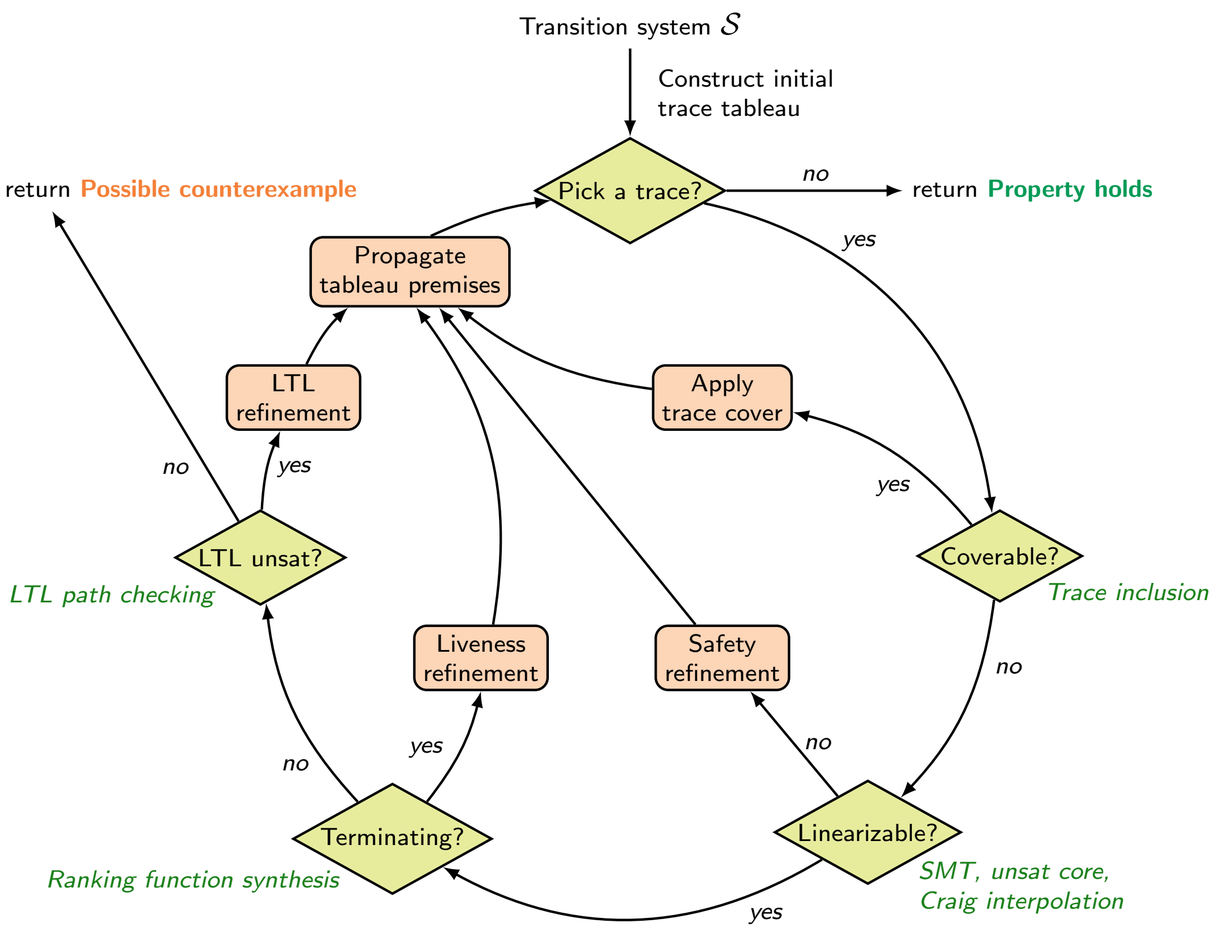
Concurrent Traces

Compactly represent sets of program runs, by specifying events that should *necessarily* occur in the run, and the partial order between them. An arbitrary number of other events can happen in between if they satisfy the edge constraints. Concurrent traces enjoy efficient algorithms for testing language emptiness and inclusion.



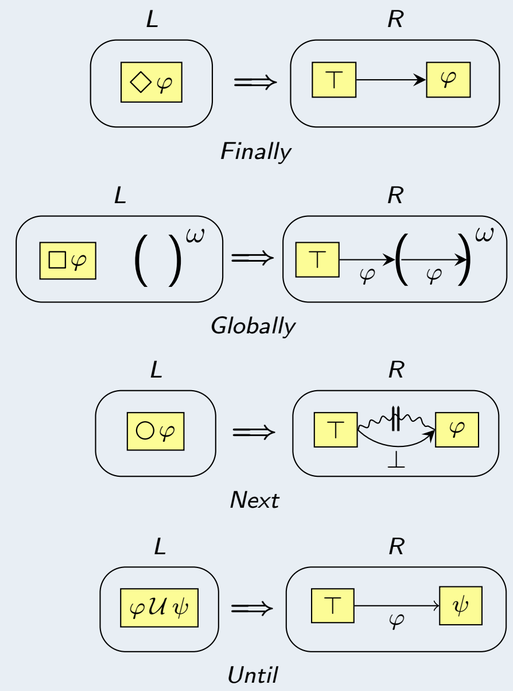
Trace Tableau





Proof Rules

Proof rule is a set $\tau : \{\tau_1, \ldots, \tau_n\}$ of trace transformations $\tau_i : (L \rightarrow R_i)$. A proof rule should preserve the set of system runs: $\mathcal{L}(L) = \cup_i \mathcal{L}(R_i)$.



Safety/Reachability	LTL Model Checking	Liveness/Termination
$\pm\ \forall\pi.\Box\Phi\ /\ \exists\pi.\Diamond\Phi$	✓ Full LTL	$\pm\ \forall\pi.\Diamond\Phi\ /\ \exists\pi.\Box\Phi$
✓ Infinite-state	✓ Infinite-state	✓ Infinite-state
✓ Multi-threading	✓ Multi-threading	✓ Multi-threading
$T_1 \parallel \ldots \parallel T_n \models$ $\Box\neg(at_{l_3} \wedge at_{m_3})$	$T_1 \parallel \ldots \parallel T_n \models$ $\Box\Diamond(at_{l_2} \wedge r>0) \Longrightarrow \Box\Diamond at_{l_3}$	$T_1 \parallel \ldots \parallel T_n \models$ $at_{l_2} \Longrightarrow \Diamond at_{l_3}$
[CONCUR 2013]		[CAV 2014]

Results

- **Safety**: our causality-based algorithm is the first that is able to analyze in polynomial time the class of multi-threaded programs with binary locks and arbitrary control flow.
- **Termination**: we have developed the first termination prover, *Arctor*, that scales to a large number of concurrent threads (see table).
- **LTL Model Checking**: implementation is in progress. Preliminary experience shows that in some cases our approach can achieve exponentially better results compared to automata-based model checking.

	Terminator		T2		AProVE		**Arctor** [a]	
Threads	Time(s)	Mem.(MB)	Time(s)	Mem.(MB)	Time(s)	Mem.(MB)	Time(s)	Mem.(MB)
1	3.37	26	2.42	38	3.17	237	**0.002**	**2.3**
2	1397	1394	3.25	44	6.79	523	**0.002**	**2.6**
3	×	MO	U(29.2)	253	U(26.6)	1439	**0.002**	**2.6**
4	×	MO	U(36.6)	316	U(71.2)	1455	**0.003**	**2.7**
5	×	MO	U(30.7)	400	U(312)	1536	**0.007**	**2.7**
10	×	MO	Z3-TO	×	×	MO	**0.027**	**3.0**
20	×	MO	Z3-TO	×	×	MO	**0.30**	**4.2**
40	×	MO	Z3-TO	×	×	MO	**4.30**	**12.7**
60	×	MO	Z3-TO	×	×	MO	**20.8**	**35**
80	×	MO	Z3-TO	×	×	MO	**67.7**	**145**
100	×	MO	Z3-TO	×	×	MO	**172**	**231**

[a] **Arctor** : **A**bstraction **R**efinement of **C**oncurrent **T**emporal **Or**derings (react.uni-saarland.de/tools/arctor/)

Publications

[CONCUR 2013] A. Kupriyanov and B. Finkbeiner. Causality-based Verification of Multi-threaded Programs. In *CONCUR 2013*, vol. 8052 of LNCS, pp. 257-272. Springer, 2013.

[CAV 2014] A. Kupriyanov and B. Finkbeiner. Causal Termination of Multi-threaded Programs. In *CAV 2014*, vol. 8859 of LNCS, pp. 812-828. Springer, 2014.

Contact: Andrey Kupriyanov
kupriyanov@cs.uni-saarland.de
react.uni-saarland.de/people/kupriyanov.html

Reactive Systems

UNIVERSITÄT DES SAARLANDES