HyperLasso: Bounded Model Checking of $\forall^+\exists^+$-Liveness Hyperproperties

Alcino Cunha Hugo Pacheco ⌜Nuno Macedo⌟
University of Minho & INESC TEC, Portugal

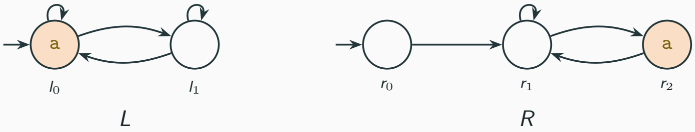
HYPER 2026 (adapted from CAV 2026)

Hyperproperties: relating multiple executions

Some properties relate multiple traces: non-interference, robustness, synthesis, ...
HyperLTL = LTL + explicit quantification over traces:

$$\varphi ::= \forall\pi \cdot \varphi \mid \exists\pi \cdot \varphi \mid \psi \qquad\qquad \psi ::= \mathsf{a}_\pi \mid \neg\psi \mid \psi \wedge \psi \mid \mathbf{X}\,\psi \mid \psi\,\mathbf{U}\,\psi$$

Running example (quantifiers may range over *different* models):



$\forall\pi_1 : L \cdot \exists\pi_2 : R \cdot \mathbf{F}(\mathsf{a}_{\pi_1} \wedge \mathsf{a}_{\pi_2})$ "some counterpart eventually sees a *simultaneously*"

Refute by finding a witness of the dual $\exists\pi_1 : L \cdot \forall\pi_2 : R \cdot \mathbf{G}(\neg\mathsf{a}_{\pi_1} \vee \neg\mathsf{a}_{\pi_2})$

✗ false: $l_0(l_1)^\omega$ sees a only at step 0, but traces of R need ≥ 2 steps to reach a.

Bounded model checking

A model is a Kripke structure $K = \langle S, S_0, \delta, L\rangle$. BMC is symbolic: the initial-state predicate S_0 and transition predicate δ are over Boolean state vectors.

Bug finding by unrolling: search for a length-k path of K that violates φ

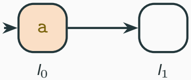
$$\underbrace{[\![K]\!]_k}_{\text{model}} \wedge \underbrace{{}_0[\![\neg\varphi]\!]_k}_{\text{property}} \rightsquigarrow \text{SAT}$$

Theorem (infinite inference for LTL)

$\not\models_k \varphi \;\Rightarrow\; \not\models \varphi$ every bounded counter-example is a real bug of K

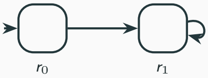
A bounded path can **refute** a property over all traces:

✗ **safety** $\exists\pi : L \cdot \mathbf{F}\,\neg \mathtt{a}_\pi$ has a witness



a finite **prefix** $l_0 l_1$ of L reaches $\neg\mathtt{a}$

✗ **liveness** $\exists\pi : R \cdot \mathbf{G}\,\neg \mathtt{a}_\pi$ has a witness



an infinite **lasso** $r_0(r_1)^\omega$ of R never reaches $\mathtt{a}$

HyperQB: BMC for HyperLTL, without loops

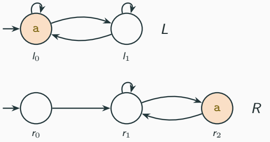
[Hsu et al., TACAS'21]

Trace quantifiers $\Rightarrow$ one copy of the symbolic unrolling per trace $\rightsquigarrow$ QBF
HyperQB keeps **finite-prefix** semantics (no loops) $\Rightarrow$ conclusive only for some $\forall\exists$-liveness

Consider refuting the dual properties:

Valid properties, any models

under *pessimistic* semantics

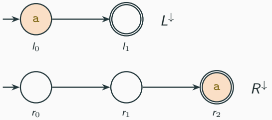


$$\text{Orig} : \forall\pi_1 : L \cdot \exists\pi_2 : R \cdot \mathbf{F}(a_{\pi_1} \vee a_{\pi_2})$$
$$\text{Dual} : \exists\pi_1 : L \cdot \forall\pi_2 : R \cdot \mathbf{G}\,\neg(a_{\pi_1} \vee a_{\pi_2})$$

✓ **no witness** for any k: a in first state l_0.

Invalid properties, terminating models

under *optimistic halting* semantics



$$\text{Orig} : \forall\pi_1 : L^{\downarrow} \cdot \exists\pi_2 : R^{\downarrow} \cdot \mathbf{F}(a_{\pi_1} \wedge a_{\pi_2})$$
$$\text{Dual} : \exists\pi_1 : L^{\downarrow} \cdot \forall\pi_2 : R^{\downarrow} \cdot \mathbf{G}\,\neg(a_{\pi_1} \wedge a_{\pi_2})$$

✗ **witness** for $k = 2$: traces eventually halt without a at l_1.

HyperQB 2.0: BMC for HyperLTL, with loops

[Hsu et al., TACAS'23 & CAV'26]

For HyperLTL, refuting liveness still needs infinite **lasso** counter-examples.

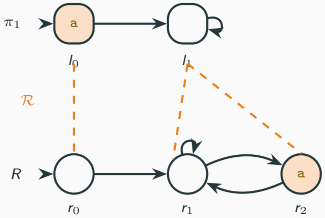
> *"[...] the model under $\forall$ would have to be explored to the maximal lasso length [...], which is doubly-exponential. Therefore, this naive approach would be highly inefficient."*

HyperQB 2.0 instead recovers loops via **simulation** between the states of the two models.

$$\exists \pi_1 : L \cdot \forall \pi_2 : R \cdot \mathbf{G}\, \neg(\mathtt{a}_{\pi_1} \wedge \mathtt{a}_{\pi_2})$$

Our running example falls into their **EA-simulation** fragment.
Guess a **lasso** π_1 of L and a relation $\mathcal{R} \subseteq S_L \times S_R$:

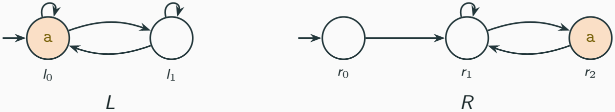
- $(\pi_1(0), r_0) \in \mathcal{R}$ for every initial state r_0 of R.
- closed under every successor of R (wrapping at t's loop)
- $\neg(\mathtt{a}_{\pi_1} \wedge \mathtt{a}_{\pi_2})$ on every pair of $\mathcal{R}$



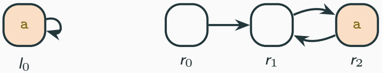
✓ **true** with $\pi_1 = l_0(l_1)^\omega$ and $\mathcal{R} = \{(l_0, r_0), (l_1, r_1), (l_1, r_2)\}$.
Limitations: R is unrolled over its explicit state space, only for $\mathbf{G}$.

Our problem: BMC for HyperLTL, in general

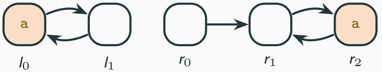
Find a k-loop witness of $\exists\pi_1 \cdot \forall\pi_2 \cdot \mathbf{G}(\neg \mathtt{a}_{\pi_1} \vee \neg \mathtt{a}_{\pi_2})$.
But the inner $\forall$ now ranges only over k*-bounded* counterparts, an under-approximation:



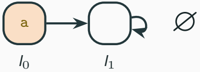
- $k = 0$: $\pi_1 = (l_0)^\omega$; R has *no* 0-loops
  - $\forall\pi_2$ holds vacuously $\Rightarrow$ bug?
  - ✓ spurious: $\mathbf{G}(\neg \mathtt{a}_{\pi_1} \vee \neg \mathtt{a}_{\pi_2})$ does not hold for the $(2,1)$-loop $r_0(r_1 r_2)^\omega$ (it needs $k = 2$)



- $k = 1$: $\pi_1 = (l_0 l_1)^\omega$
  - ✓ spurious again: same counterpart (both see a at step 2)

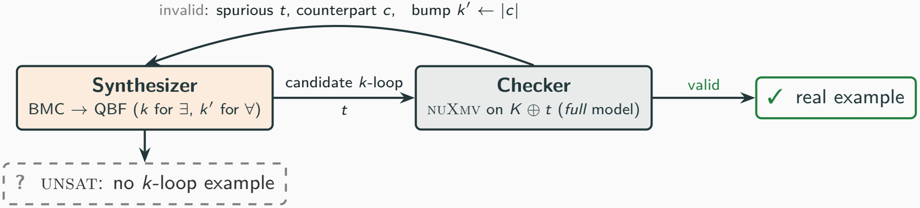


- $k = 1$: $\pi_1 = l_0(l_1)^\omega$
  - ✓ a true bug: a only at step 0, unmatchable by any trace from R



Our approach: synthesize, then check

Key observation: with a *single* quantifier alternation $\exists^+\forall^+\neg\psi$, once the $\exists$-traces are fixed, what remains is $\forall^+\neg\psi \Rightarrow$ checkable by a complete LTL model checker via a product construction.



CEGAR-flavoured loop; on the example:

1. synthesizer ($k{=}1, k'{=}1$) proposes $(l_0 l_1)^\omega$ $\Rightarrow$ checker deems invalid, counterpart $r_0(r_1 r_2)^\omega$
2. synthesizer ($k{=}1, k'{=}2$) proposes $l_0(l_1)^\omega$ $\Rightarrow$ checker deems valid

Our approach: guarantees

Theorem (infinite inference)

Every k-loop returned by the procedure witnesses both $\models_k \exists^+\forall^+\neg\psi$ *and* $\models \exists^+\forall^+\neg\psi$.

Why a returned lasso unrolls to any $k' > k$, and longer universal counterparts were already discharged (the checker is complete).

Takeaway BMC's value proposition is restored (every reported counter-example is real)

Note Lasso-shaped traces capture all relevant infinite paths for HyperLTL.

Lemma (bounded inference)

If no witness is returned, then $\not\models_k \exists^+\forall^+\neg\psi$.

Why growing the inner bound $k' \geq k$ only *adds* universal counterparts, so discarded candidates stay discarded.

Takeaway Monotonic confidence as k grows (bounded completeness *for bound* k).

Note Identifying a completeness threshold remains orthogonal and generally infeasible.

The synthesizer: reduction to QBF

General schema = k copies of each model's state variables + a loopback index per trace:

$$[\![\exists \pi_{1\dots n} \dots \forall \pi_{n+1\dots m} \cdot \psi]\!]_k \coloneqq \exists \overline{x_{\pi_{1\dots n}}} \cdot \exists l_{1\dots n} \dots \forall \overline{x_{\pi_{n+1\dots m}}} \cdot \forall l_{n+1\dots m} \cdot$$

$$\underbrace{\bigvee_{0 \le i_1, \dots, i_m \le k} \Big(\bigwedge_{1 \le j \le m} l_j = i_j}_{\text{loop selection}} \wedge \underbrace{\bigwedge_{1 \le j \le n} [\![\pi_j]\!]_k^{i_j}}_{\exists \text{ models}} \wedge \Big(\underbrace{\bigwedge_{n < j \le m} [\![\pi_j]\!]_k^{i_j}}_{\forall \text{ models}} \rightarrow \underbrace{{}_0[\![\psi]\!]_k^{\{\pi_1 \mapsto i_1, \dots, \pi_m \mapsto i_m\}}}_{\text{property}} \Big)\Big)$$

The whole encoding for the example at $k = 1$:

$$\exists x_L^0\, x_L^1 \cdot \exists l_1 \cdot \forall x_R^0\, x_R^1 \cdot \forall l_2 \cdot$$

$$\bigvee_{i_1, i_2 \in \{0,1\}} \Big(l_1 = i_1 \wedge l_2 = i_2 \wedge [\![\pi_1]\!]_1^{i_1} \wedge \Big([\![\pi_2]\!]_1^{i_2} \rightarrow {}_0[\![\mathbf{G}(\neg \mathtt{a}_{\pi_1} \vee \neg \mathtt{a}_{\pi_2})]\!]_1^{\{\pi_1 \mapsto i_1,\, \pi_2 \mapsto i_2\}} \Big)\Big)$$

where the models unroll and close a loop back to state i, e.g.

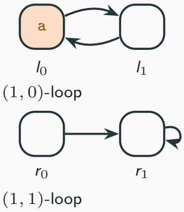
$$[\![\pi_1]\!]_1^0 = I_L(x_L^0) \wedge R_L(x_L^0, x_L^1) \wedge R_L(x_L^1, x_L^0)$$

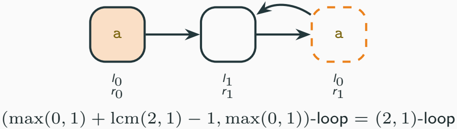
and the body expands over the *combined* lasso, e.g. for $(i_1, i_2) = (0, 1)$:

$$(\neg \mathtt{a}_L^0 \vee \neg \mathtt{a}_R^0) \wedge (\neg \mathtt{a}_L^1 \vee \neg \mathtt{a}_R^1) \wedge \underbrace{(\neg \mathtt{a}_L^0 \vee \neg \mathtt{a}_R^1)}_{\text{indices from different periods}}$$

The synthesizer: aligning lassos of different periods

Evaluating ψ over a *pair* of lassos = walking both loops synchronously (up to k^m for m traces): a combined lasso with loopback $\underline{l} = \max(i, j)$ and last index $\underline{k} = \underline{l} + \mathrm{lcm}(\text{periods}) - 1$





Key insight: do not create state variables beyond k

States of the combined path beyond k are virtual (dashed): the formula indexes back into the depth-k variables by modular arithmetic over each loop's period,

$$ {}_i[\![\mathtt{a}_\pi]\!]^\Lambda_k := \begin{cases} \mathtt{a}_\pi^i & i < \Lambda(\pi) \\ \mathtt{a}_\pi^{\underline{i}} & \text{otherwise} \end{cases} \qquad \underline{i} := \Lambda(\pi) + \big((i - \Lambda(\pi)) \bmod (k + 1 - \Lambda(\pi))\big) $$

with temporal operators unfolding up to $\underline{k}$, and $\mathbf{X}$ wrapping back to $\underline{l}$.

The checker: an off-the-shelf LTL model checker

Candidate = concrete lassos $t_1, \ldots, t_n$ for the $\exists$-traces.
Verify $\forall\pi_1 \ldots \forall\pi_m \cdot \psi$ where:

- $\exists$-models are pinned to their candidate
  - $K_{\pi_i} \oplus t_i$ adds an index variable + invariants forcing the single trace t_i
- all models composed into one; negated body as the LTLSPEC
- solved by NUXMV (*complete*, no bound)

A counter-example $c \Rightarrow$ candidate invalid

- $|c|$ is then used as the new inner bound k'

Checking candidate $(l_0 l_1)^\omega$:

```
MODULE main
VAR L_l : 0..1;  R_r : 0..2;
    L_s : 0..1;  -- lasso index
ASSIGN
  init(L_l) := 0; ...
  init(R_r) := 0; ...
  init(L_s) := 0;
  next(L_s) := case L_s < 1 : L_s+1;
                    TRUE : 0; esac;
INVAR -- pin L to the candidate
  (L_s = 0 & L_l = 0) |
  (L_s = 1 & L_l = 1);
DEFINE L_a := (L_l = 0);
       R_a := (R_r = 2);
LTLSPEC G (!L_a | !R_a)
```

A benchmark for liveness hyperproperties

Existing suites contain mostly safety properties or toy examples $\Rightarrow$ new public benchmark: liveness hyperproperties with a single alternation, over non-terminating models of parametric size.

Verification ($\forall^+\exists^+$ with liveness)	
`gni`	generalized non-interference in a conference system (decisions eventually made)
`rc/ser`	DB isolation-level equivalence, Read Committed vs Serializability
`cni`	non-interference in a multi-threaded PIN-masking program, fair scheduler
`selfstab`	Herman's self-stabilization on a ring (eventually a single token)
`robust`	robustness of Lamport's Bakery (a halted process cannot starve the others)
Synthesis ($\exists^+\forall^+$ with safety)	
`spec`	semaphore controller synthesis (Mealy machine as frozen variables)
`plan`	robot path planning avoiding e moving enemies on an $n \times n$ board

Results

Setup: AutoHyper (explicit-state, complete tool for any model/property) with its best automata backend per problem, witness mode; $300s$ timeout; MacBook Pro M1.

	Result	Magnitude	AutoHyper	HyperLasso
`gni_any_3x3`	✗	$(10^9)^3$	⌛	3.8
`rc_ser_3x2`	✗	$10^6 \times 10^6$	⌛	0.4
`cni_any_2`	✗	$(10^3)^2$	⌛	65.5
`selfstab_gf_7`	✗	$(10^3)^2$	⌛	0.7
`robust_crit_3`	✗	$(10^3)^2$	⌛	8.7
`spec2_4x4`	✓	$10^{14} \times 10^3$	⌛	2.6
`plan_4x3`	✓	$(10^2)^4$	⌛	1.7
`selfstab_lf_3`	✓	$(10^1)^2$	**0.8**	117.8 ?

RQ1 — versus AutoHyper HyperLasso wins consistently beyond minimal scopes. AutoHyper wins `selfstab_lf_3`. AutoHyper times out out from $\sim 10^5$ states (e.g. `selfstab_lf_5`).

RQ2 — are spurious candidates real? Yes, in 6 of 7 problem families; but ≤ 2 iterations ever needed, checker time negligible, one k' jump usually suffices.

? Incomplete results: valid answers are only "valid up to k", and a small k can render some problems vacuous (e.g. fairness in `selfstab` needs the token to circle the ring once).

Takeaways

1. **Bounded refutation is unsound for $\forall\exists$-liveness**: over reactive models, bounded counter-examples can be spurious.
2. **Synthesize, then check**: a QBF lasso-synthesizer + a complete LTL checker (self-composition, one alternation) restore infinite inference.
3. **Symbolic scales**: HyperLasso consistently outperforms the explicit-state AutoHyper on models beyond $\sim 10^5$ states.

Future work: arbitrary quantifier alternations; completeness results beyond bug-finding; a proper CEGAR loop.

`https://github.com/haslab/HyperLasso`
`https://doi.org/10.5281/zenodo.19850821`

Thank you!