

HyperProb: A Model Checker for Probabilistic Hyperproperties

24th International Symposium on Formal methods '21

Oyendrila Dobe[†], Erika Ábrahám^{*}, Ezio Bartocci^{**}, Borzoo Bonakdarpour[†]

[†]Michigan State University (USA), ^{*}RWTH Aachen (Germany),

^{**}TU-Wien (Austria)

November 26, 2021

Hyperproperty [Clarkson, Schneider, 2010]

[Clarkson, Finkbeiner, Koleini, Micinski, Rabe, Sánchez, 2014]

- A **trace property** is a set of traces.

Hyperproperty [Clarkson, Schneider, 2010]

[Clarkson, Finkbeiner, Koleini, Micinski, Rabe, Sánchez, 2014]

- A **trace property** is a set of traces.
- Classical trace properties **cannot express relations** between traces!

Hyperproperty [Clarkson, Schneider, 2010]

[Clarkson, Finkbeiner, Koleini, Micinski, Rabe, Sánchez, 2014]

- A **trace property** is a set of traces.
- Classical trace properties **cannot express relations** between traces!
- A **hyperproperty** is a **set of sets of traces**.

Hyperproperty [Clarkson, Schneider, 2010]

[Clarkson, Finkbeiner, Koleini, Micinski, Rabe, Sánchez, 2014]

- A **trace property** is a set of traces.
- Classical trace properties **cannot express relations** between traces!
- A **hyperproperty** is a **set of sets of traces**.

HyperLTL property

I drink coffee every day at the same time.

$$\forall \pi. \forall \pi'. (\Box (coffee_{\pi} \Leftrightarrow coffee_{\pi'}))$$

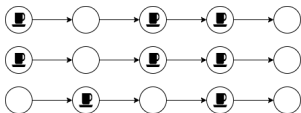


Figure: Example of set of traces

HyperPCTL [HyperPCTL for DTMCs by Ábrahám, Bonakdarpour, '18]

[HyperPCTL for MDPs by Ábrahám, Bartocci, Bonakdarpour, Dobe, '20]

- Express probabilistic **relations between traces**.

HyperPCTL [HyperPCTL for DTMCs by Ábrahám, Bonakdarpour, '18]

[HyperPCTL for MDPs by Ábrahám, Bartocci, Bonakdarpour, Dobe, '20]

- Express probabilistic **relations between traces**.
- Quantify ($Q_i \in \{\exists, \forall\}$) over schedulers and initial states:

$Q_{\hat{\sigma}_1} \hat{\sigma}_1 \dots Q_{\hat{\sigma}_m} \hat{\sigma}_m.$
scheduler quantification

$Q_{\hat{s}_1} \hat{s}_1(\hat{\sigma}_1) \dots Q_{\hat{s}_n} \hat{s}_n(\hat{\sigma}_n).$
state quantification

ψ
non-quantified HyperPCTL

HyperPCTL [HyperPCTL for DTMCs by Ábrahám, Bonakdarpour, '18]

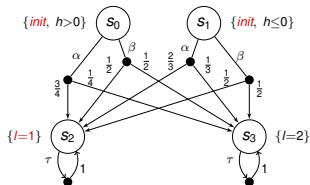
[HyperPCTL for MDPs by Ábrahám, Bartocci, Bonakdarpour, Dobe, '20]

- Express probabilistic **relations between traces**.
- Quantify ($Q_i \in \{\exists, \forall\}$) over schedulers and initial states:

$Q_{\hat{\sigma}_1} \hat{\sigma}_1 \dots Q_{\hat{\sigma}_m} \hat{\sigma}_m$
scheduler quantification

$Q_{\hat{s}_1} \hat{s}_1(\hat{\sigma}_1) \dots Q_{\hat{s}_n} \hat{s}_n(\hat{\sigma}_n)$
state quantification

ψ
non-quantified HyperPCTL



$$\forall \hat{\sigma}. \forall \hat{s}(\hat{\sigma}). \forall \hat{s}'(\hat{\sigma}). \left(\text{init}_{\hat{s}} \wedge \text{init}_{\hat{s}'} \wedge h_{\hat{s}} \neq h_{\hat{s}'} \right) \Rightarrow \left(\mathbb{P}(\diamond(l=1)_{\hat{s}}) = \mathbb{P}(\diamond(l=1)_{\hat{s}'})) \right)$$

Figure: Example of an MDP.

Restricted HyperPCTL model checking

Model checking HyperPCTL formulas for MDPs is undecidable



Restriction to non-probabilistic memoryless schedulers

Restricted HyperPCTL model checking

Model checking HyperPCTL formulas for MDPs is undecidable



Restriction to non-probabilistic memoryless schedulers

In HyperProb,

$$\mathcal{M} = (S, Act, Pr, AP, L)$$

satisfies

$$Q\hat{\sigma}.Q_1\hat{s}_1(\hat{\sigma}) \dots Q_n\hat{s}_n(\hat{\sigma}).\varphi^{nq}$$

iff

SMT encoding is satisfied

Currently, HyperProb allows **one scheduler quantifier**.

Overview of the tool

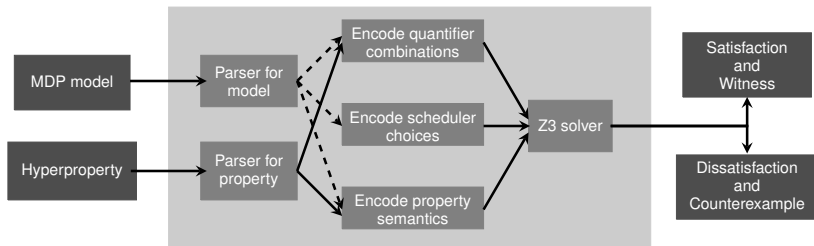


Figure: Dataflow inside the tool

- **Input:** Prism model (.nm file) and HyperPCTL formula (String).
- **Output:** Verification Result (Boolean) & Witness / Counterexample (sequence of actions).

Example output

```

oyendiladobe@MacBook-Pro-5 ~ % docker exec -it 05e196893961c30381f123cb28f83e13f7386e9e10d87d8b8e837bcfb4285aa9 /bin/sh
# cd ../home/HyperProb
# python source.py benchmark_files/mdp/TA/timing_attack2.nm 'AS sh . A s1 . A s2 . ~((start0(s1) & start1(s2)) & ~(P(F counter0(s1)) = P(F counter0(s2))))'
Model file is located at: /home/HyperProb/benchmark_files/mdp/TA/timing_attack2.nm
Total number of states: 24
Total number of transitions: 46
Total number of actions: 30

Encoded actions in the MDP...
Encoded quantifiers...
Encoded non-quantified formula...
Time to encode in seconds: 0.85
Checking...
The property DOES NOT hold!
The actions at the corresponding states of a counterexample are:
State 0 = 0
State 1 = 1
State 2 = 0
State 3 = 1
State 4 = 0
State 5 = 0
State 6 = 0
State 7 = 0
State 8 = 0
State 9 = 1
State 10 = 0
State 11 = 0
State 12 = 0
State 13 = 1
State 14 = 0
State 15 = 0
State 16 = 0
State 17 = 0
State 18 = 0
State 19 = 0
State 20 = 0
State 21 = 0
State 22 = 0
State 23 = 0

Time to encode in seconds: 0.85
Time required by z3 in seconds: 0.13

Number of variables: 3900
Number of formula checked: 6540

```

Figure: Sample execution

Evaluation

Case Study		Running time(s)						#SMT variables		#op	#st	#tr
		SE		SS		Total		N	O			
		N	O	N	O	N	O					
TA	$m = 2$	5	2	< 1	< 1	5	2	8088	2520	14	24	46
	$m = 4$	114	18	20	1	134	19	50460	14940		60	136
	$m = 6$	1721	140	865	45	2586	185	175728	51184		112	274
	$m = 8$	12585	952	TO	426	TO	1378	388980	131220		180	460
PW	$m = 2$	5	2	< 1	< 1	6	3	8088	2520	14	24	46
	$m = 4$	207	26	40	1	247	27	68670	20230		70	146
	$m = 6$	3980	331	1099	41	5079	372	274540	79660		140	302
	$m = 8$	26885	2636	TO	364	TO	3000	657306	221130		234	514
TS	$h = (0, 1)$	< 1	< 1	< 1	< 1	1	1	1379	441	28	7	13
	$h = (0, 15)$	60	8	1607	< 1	1667	8	34335	8085		35	83
	$h = (4, 8)$	12	3	17	< 1	29	3	12369	3087		21	48
	$h = (8, 15)$	60	8	1606	< 1	1666	8	34335	8085		35	83
	$h = (10, 20)$	186	19	13707	1	13893	20	52695	13095		45	108
PC	$s=(0)$	277	10	1996	5	2273	15	21220	6780	44	20	188
	$s=(0,1)$	822	13	5808	5	6630	18	21220	6780		20	340
	$s=(0..2)$	1690	15	TO	5	TO	20	21220	6780		20	494
	$s=(0..3)$	4631	16	TO	7	TO	23	21220	6780		20	648
	$s=(0..4)$	7353	22	TO	21	TO	43	21220	6780		20	802
	$s=(0..5)$	10661	19	TO	61	TO	80	21220	6780		20	956
	$s=(0..6)$	13320	18	TO	41	TO	59	21220	6780		20	1110

Table: **TA**: Timing attack. **PW**: Password leakage. **TS**: Thread scheduling. **PC**: Probabilistic conformance. **TO**: Timeout. **N**: Prototype presented in previous work. **O**: HyperProb. **SE**: SMT encoding. **SS**: SMT solving. **#op**: Formula size (number of operators). **#st**: Number of states. **#tr**: Number of transitions.

Summary

- 1 Presented a **SMT-based model checking algorithm** which is NP-complete (coNP-complete for universal quantifier) in the state set size of the input MDP.
- 2 Provided a docker container with the pre-installed dependencies, to run the tool.

`https://github.com/TART-MSU/HyperProb`