Assuring Autonomy in Contested Environments

Protecting Information

Y. Wang, S. Nalluri, and M. Pajic, “Hyperproperties for Robotics: Motion Planning via HyperLTL”, IEEE International Conference on Robotics and Automation (ICRA), 2020, to appear

Hyper-properties for Cyber-Physical Systems

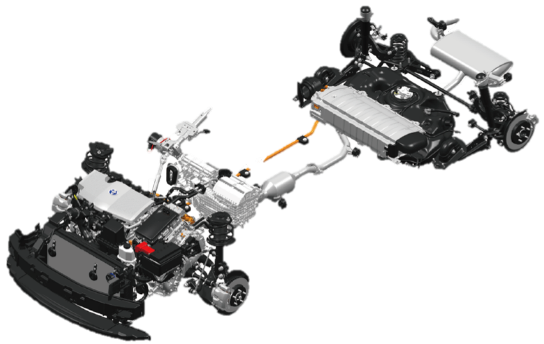
- CPS properties of interest commonly include properties such as
  - sensitivity to modeling errors,
  - probabilistic fairness, and
  - anomaly detectability

- These should capture a relationship between multiple simultaneous continuous-time runs

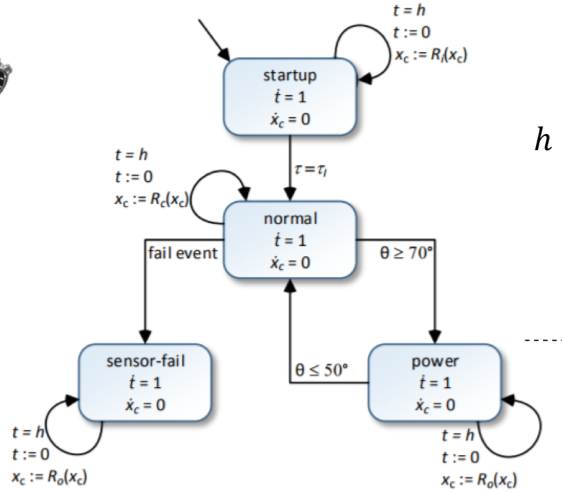
Example: System *Sensitivity* to Modeling Errors

Duke PRATT SCHOOL *of* ENGINEERING

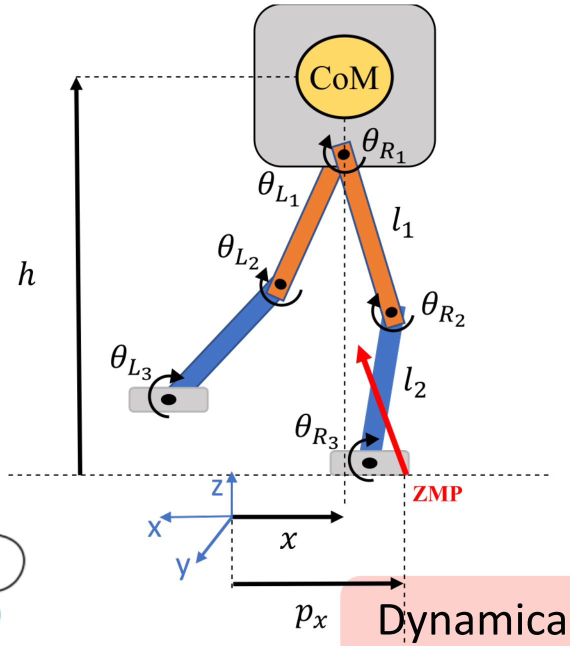
Toyota Powertrain Benchmark

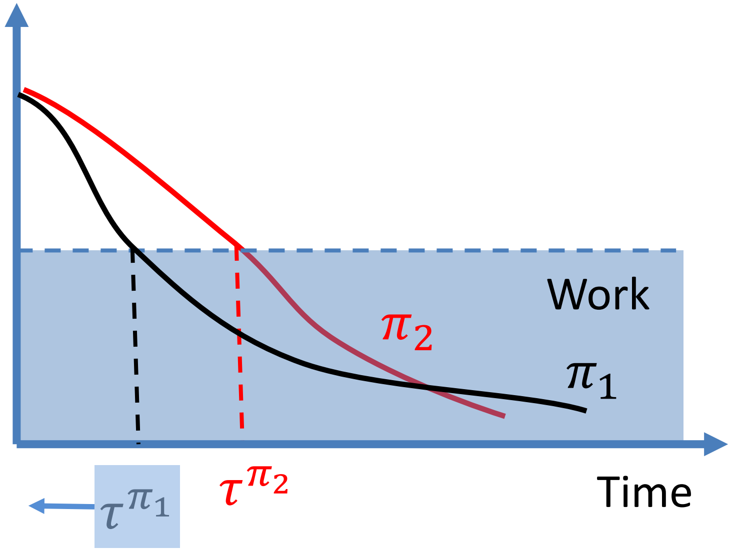


Embedded Controller



Walking Robot Benchmark With Reinforcement Learning





Dynamical response depends on system parameters

Y. Wang, M. Zarei, B. Bonakdarpour and M. Pajic, "Statistical Verification of Hyper-properties for Cyber-Physical Systems", *19th ACM SIGBED International Conference on Embedded Software* (**EMSOFT**), Oct 2019,
Best Paper Award Finalist

How does dynamical response change due to modeling errors or *wear-and-tear*?

- For example, start time change under probabilistic uncertainty?

Probabilistic hyperproperties*:* Sensitivity under probabilistic parameter change

$$\mathbf{Pr}_{\pi_1,\pi_2}(|\tau^{\pi_1} - \tau^{\pi_2}| \leq \delta) > 1 - \varepsilon$$

We need new logic to reason over ***multiple*** random paths!

HyperPSTL: Hyper Probabilistic Signal Temporal Logic

Duke PRATT SCHOOL of ENGINEERING

STL → Add reference to different paths → HyperSTL → Add probabilistic quantifications → HyperPSTL → Add probabilistic arithmetic → (full) HyperPSTL

Syntax:

$$\varphi ::= a^{\pi} \mid \varphi^{\pi} \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \mathcal{U}_{[t_1,t_2]}\varphi \mid p \bowtie p$$
$$p ::= \mathbb{P}^{\Pi}\varphi \mid \mathbb{P}^{\Pi}p \mid f(p,\dots,p)$$

- a ∈ AP, and AP is the finite set of *atomic propositions*,
- $t_1 < t_2$ with $t_1, t_2 \in \mathbb{Q}_{\infty}$,
- π is a path variable, and Π is a set of path variables,
- $\mathrm{fv}(\varphi) = \emptyset$
- $\mathbb{P}$ is the probability operator,
- $\bowtie \in \{<, >, =, \leq, \geq\}$,
- $f: \mathbb{R}^n \to \mathbb{R}$ is a n-ary elementary function, constants are viewed as 0-ary functions,

Semantics

For probabilistic arithmetic:

$$\mathcal{S} \vDash [\![p \bowtie p]\!]_{V_X} \Leftrightarrow \mathcal{S} \vDash [\![p]\!]_{V_X} \bowtie [\![p]\!]_{V_X}$$
$$\mathcal{S} \vDash [\![f(p,\dots,p)]\!]_{V_X} \Leftrightarrow \mathcal{S} \vDash f([\![p]\!]_{V_X},\dots,[\![p]\!]_{V_X})$$
$$\mathcal{S} \vDash [\![\mathbb{P}^{\Pi}(\varphi)]\!]_{V_X} \Leftrightarrow \mathcal{S} \vDash \mathbf{Pr}_{\boldsymbol{\sigma}\sim \mathrm{Path}^{|\Pi|}(X)}((\mathcal{S}, V_X[\Pi \to \boldsymbol{\sigma}]) \vDash \varphi)$$

$V_X[\Pi \to \boldsymbol{\sigma}]$ is a revision of the assignment V_X.

For AP and state formulas with $\mathrm{fv}(\varphi) = \emptyset$:

$$(\mathcal{S}, V_X) \vDash a^{\pi} \Leftrightarrow a \in L(V_X(\pi)(0))$$
$$(\mathcal{S}, V_X) \vDash \varphi^{\pi} \Leftrightarrow (\mathcal{S}, V_X(\pi)) \vDash \varphi$$
$$(\mathcal{S}, X) \vDash \varphi \Leftrightarrow \mathcal{S} \vDash [\![\varphi]\!]_{V_X}$$

For temporal operators:

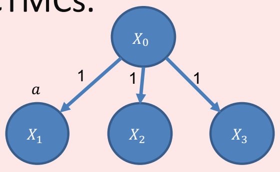
$$(\mathcal{S}, V_X) \vDash \varphi_1 \wedge \varphi_2 \Leftrightarrow (\mathcal{S}, V_X) \vDash \varphi_1 \text{ and } (\mathcal{S}, V_X) \vDash \varphi_2$$
$$(\mathcal{S}, V_X) \vDash \varphi_1 \mathcal{U}_{[t_1,t_2]}\varphi_2 \Leftrightarrow \exists t \in [t_1, t_2].$$
$$(\forall t' < t. (\mathcal{S}, V_X^{(t')}) \vDash \varphi_1) \wedge (\mathcal{S}, V_X^{(t)}) \vDash \varphi_2$$

where $V_X^{(t)}$ is the t-shift of the assignment V_X

Expressiveness:

Theorem: HyperPSTL strictly subsumes PSTL on CTMCs.

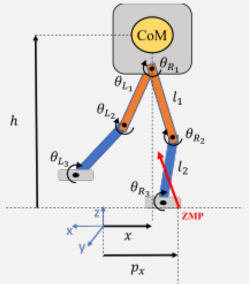
- CTMC has only 3 paths
- Satisfaction probability of any STL is $0, \frac{1}{3}, \frac{2}{3}, 1$, so $P(\varphi) = \frac{1}{9}$ is always false for any (φ)
- HyperPSTL $P^{(\pi_1,\pi_2)}(\boldsymbol{\mathcal{F}}(a^{\pi_1} \wedge a^{\pi_2})) = \frac{1}{9}$ is true



Sensitivity:

$$\mathbb{P}^{(\pi_1,\pi_2)}\left(\begin{array}{c}(\neg Q^{\pi_1} \wedge \neg Q^{\pi_2}) \\ \boldsymbol{\mathcal{U}}\left((Q^{\pi_1} \wedge \boldsymbol{\mathcal{F}}_{[0,\delta]}Q^{\pi_2}) \vee (Q^{\pi_2} \wedge \boldsymbol{\mathcal{F}}_{[0,\delta]}Q^{\pi_1})\right)\end{array}\right) > 1 - \varepsilon$$

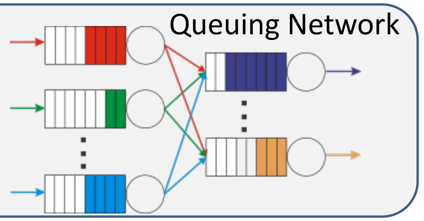
Toyota Powertrain Benchmark



Walking Robot Benchmark

Workload Fairness:

$$\mathbb{P}^{\pi_1}(|\mathbb{P}^{\pi_2}((\neg Q_i^{\pi_1} \wedge \neg Q_j^{\pi_2})\mathcal{U}(Q_i^{\pi_1} \wedge \hat{Y}_{[\tau,\infty)}Q_j^{\pi_2}))$$
$$- \mathbb{P}^{\pi_2}((\neg Q_i^{\pi_1} \wedge \neg Q_j^{\pi_2})\mathcal{U}(Q_j^{\pi_2} \wedge \hat{Y}_{[\tau,\infty)}Q_i^{\pi_1}))| \leq \delta)$$
$$\geq 1 - \varepsilon.$$



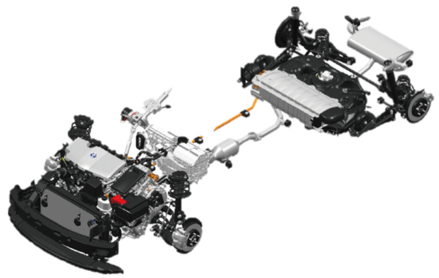
HyperSMC tool for SMC of HyperPSTL with desired confidence/significance level

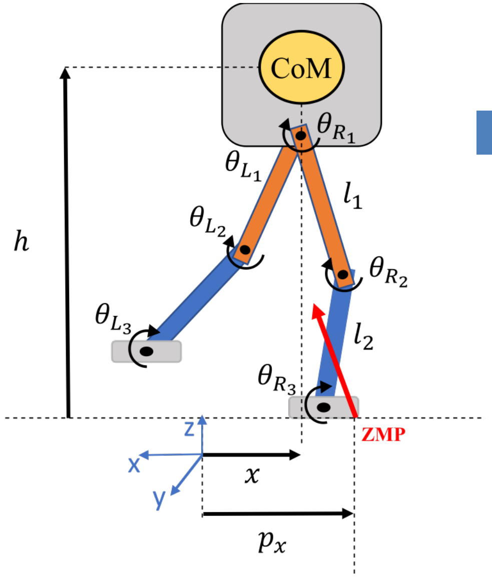
https://gitlab.oit.duke.edu/cpsl/hypersmc

Duke PRATT SCHOOL of ENGINEERING

Example: Sensitivity Verification for real-world CPS

$$\mathbb{P}^{(\pi_1,\pi_2)}\left((\neg Q^{\pi_1} \wedge \neg Q^{\pi_2})\ \boldsymbol{\mathcal{U}}\left(\left(Q^{\pi_1} \wedge \boldsymbol{\mathcal{F}}_{[0,\delta]} Q^{\pi_2}\right) \vee \left(Q^{\pi_2} \wedge \boldsymbol{\mathcal{F}}_{[0,\delta]} Q^{\pi_1}\right)\right)\right) > 1-\varepsilon$$





Walking Robot Benchmark With Reinforcement Learning Controller

δ	ε	α	Acc.	Sam.	Time (s)	Ans.
2.4	0.02	0.01	**1.00**	7.4e+01	3.0e-01	False
2.4	0.02	0.05	**0.99**	4.4e+01	1.4e-01	False
2.4	0.12	0.01	**1.00**	4.2e+01	1.2e-01	True
2.4	0.12	0.05	**1.00**	2.1e+01	7.0e-02	True
2.4	0.2	0.01	**1.00**	1.3e+01	4.0e-02	True
3.0	0.02	0.01	**1.00**	1.1e+01	2.4e-02	False
3.0	0.02	0.05	**1.00**	6.5e+00	1.1e-02	False
3.0	0.12	0.05	**0.98**	7.0e+01	4.3e-01	False
3.0	0.2	0.01	**1.00**	1.6e+02	5.5e-01	True
3.0	0.2	0.05	**0.98**	1.0e+02	2.9e-01	True

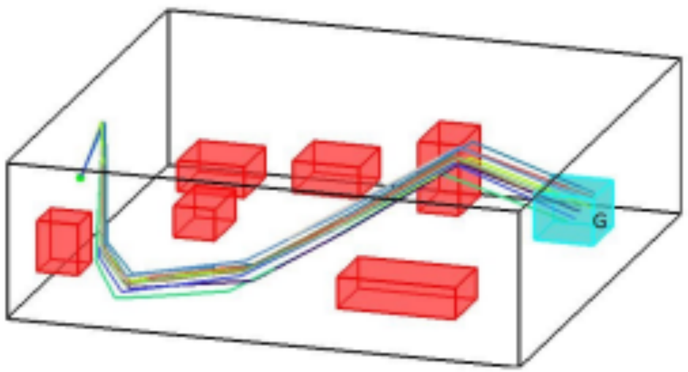
Toyota Powertrain Benchmark

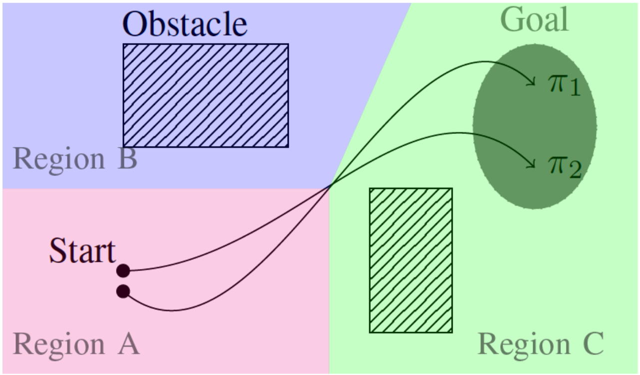
δ	ε	α	Acc.	Sam.	Time (s)	Ans.
0.15	0.95	0.05	**1.00**	5.9e+01	8.1e+00	True
0.15	0.95	0.01	**1.00**	9.0e+01	1.3e+01	True
0.15	0.99	0.05	**0.99**	6.6e+01	9.1e+00	False
0.15	0.99	0.01	**1.00**	9.7e+01	1.4e+01	False
0.20	0.95	0.05	**0.98**	5.9e+01	8.1e+00	True
0.20	0.95	0.01	**1.00**	9.0e+01	1.2e+01	True
0.20	0.99	0.05	**1.00**	3.0e+02	4.2e+01	True
0.20	0.99	0.01	**0.99**	4.6e+02	1.8e+02	True

SMC procedure extended to support verification of learning enabled controllers

[1] M. Zarei, Y. Wang, and M. Pajic, "Statistical Verification of Learning-Based Cyber-Physical Systems", *23rd ACM International Conference on Hybrid Systems: Computation and Control* (**HSCC**), 2020, **to appear**.

Hyperproperties for Motion Planning





Privacy-aware Motion Planning

$$\exists\pi_1\exists\pi_2.(\pi_1 \text{ and } \pi_1 \text{ are different paths}) \wedge (\pi_1 \text{ and } \pi_1 \text{ give identical observation}) \wedge (\pi_1 \text{ and } \pi_1 \text{ reach goal}).$$

$$\exists\pi_1\exists\pi_2.(\mathrm{sec}(\pi_1) \neq \mathrm{sec}(\pi_2)) \wedge (\mathrm{obs}(\pi_1) = \mathrm{obs}(\pi_2))$$

Optimality of Synthesized Plans

$$\exists\pi.\Big((\pi \text{ reaches goal}) \wedge (\forall\pi'.((\pi' \text{ reaches goal}) \Rightarrow (\pi \text{ reaches goal}))\Big)$$

$$\exists\pi_1\forall\pi_2.\ (\mathbf{s}_0{}^{\pi_1} \wedge \mathbf{s}_0{}^{\pi_2}) \wedge (\Diamond_T(g^{\pi_2} \Rightarrow \Diamond_T g^{\pi_1}));$$

$$\exists\pi_1\forall\pi_2.\ (\mathbf{s}_0{}^{\pi_1} \wedge \mathbf{s}_0{}^{\pi_2}) \wedge (\Diamond_T(g^{\pi_1} \Rightarrow \Diamond_T g^{\pi_2}))$$

Robustness of Synthesized Plans

$$\exists\pi\forall\pi'.(\pi \text{ is derived by disturbing } \pi') \wedge (\pi \text{ and } \pi' \text{ reach goal}).$$

$$\exists\pi_1\forall\pi_2.\ \mathrm{cls}_{\mathbf{s}_0}(\pi_1,\pi_2) \wedge \mathrm{cls}_{\mathtt{A}}(\pi_1,\pi_2) \Rightarrow (\varphi^{\pi_1} \wedge \varphi^{\pi_2})$$

HyperLTL

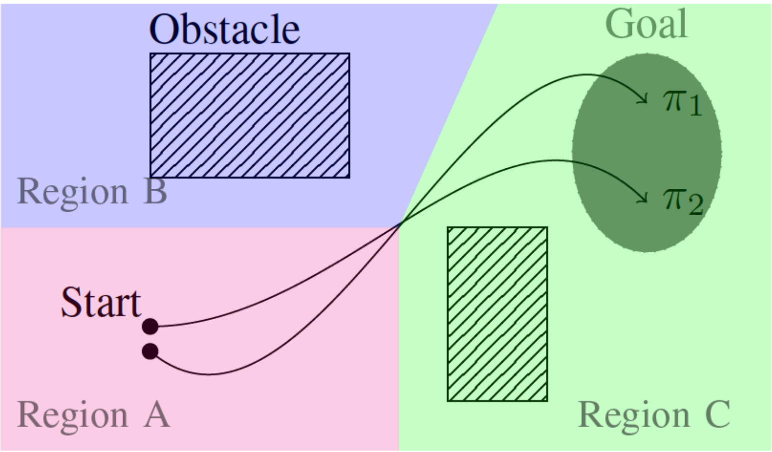
As a HyperLTL objective may contain multiple path variables, its satisfaction involves assigning concrete (infinite) paths to all these path

$$V : \Pi \to (2^{\mathsf{AP}})^{\omega}$$

The satisfaction relation for the HyperLTL path formulas is then defined for V by

$$
\begin{array}{lcl}
V \models \mathsf{a}^{\pi} & \Leftrightarrow & \mathsf{a} \in V(\pi)(0) \\
V \models \neg\varphi & \Leftrightarrow & V \not\models \varphi \\
V \models \varphi_1 \wedge \varphi_2 & \Leftrightarrow & V \models \varphi_1 \text{ and } V \models \varphi_2 \\
V \models \bigcirc\varphi & \Leftrightarrow & V^{(1)} \models \varphi \\
V \models \varphi_1 \,\mathcal{U}_T\, \varphi_2 & \Leftrightarrow & \exists t \leq T.\ \big(V^{(t)} \models \varphi_2 \text{ and} \\
 & & \quad (\forall t' < t.\ V^{(t')} \models \varphi_1)\big) \\
V \models \exists\pi.\ \psi & \Leftrightarrow & \text{there exists } \sigma \in (2^{\mathsf{AP}})^{\omega}, \\
 & & \quad \text{such that } V[\pi \mapsto \sigma] \models \psi \\
V \models \forall\pi.\ \psi & \Leftrightarrow & \text{for all } \sigma \in (2^{\mathsf{AP}})^{\omega}, \\
 & & \quad V[\pi \mapsto \sigma] \models \psi \text{ holds}
\end{array}
$$

HyperLTL for Motion Planning



Initial-state opacity for fixed strategy

$$\exists \pi_1 \exists \pi_2 . (\mathbf{s}_0{}^{\pi_1} \wedge (\neg \mathbf{s}_0{}^{\pi_2})) \wedge (\Box_T (\mathbf{a}^{\pi_1} = \mathbf{a}^{\pi_2})) \wedge ((\Diamond_T g^{\pi_1}) \wedge (\Diamond_T g^{\pi_2}))$$

Current-state opacity

$$\exists \pi_1 \exists \pi_2 . (\mathbf{s}_0{}^{\pi_1} \wedge \mathbf{s}_0{}^{\pi_2}) \wedge \wedge (\neg \Box_T (\mathbf{a}^{\pi_1} = \mathbf{a}^{\pi_2})) \wedge (\Box_T (o^{\pi_1} = o^{\pi_2}))$$

Shortest Path

$$\exists \pi_2 \forall \pi_1 .\ (\mathbf{s}_0{}^{\pi_1} \wedge \mathbf{s}_0{}^{\pi_2}) \wedge (\Diamond (g^{\pi_2} \Rightarrow \Diamond g^{\pi_1}))$$

Longest Path

$$\exists \pi_2 \forall \pi_1 .\ (\mathbf{s}_0{}^{\pi_1} \wedge \mathbf{s}_0{}^{\pi_2}) \wedge (\Diamond (g^{\pi_1} \Rightarrow \Diamond g^{\pi_2}))$$

Initial-state robustness

$$\exists \pi_1 \forall \pi_2 . (\mathbf{s}_0{}^{\pi_1} \wedge S_0^{\pi_2}) \wedge (\varphi^{\pi_1} \wedge \varphi^{\pi_2}) \wedge (\Box_T (\mathbf{a}^{\pi_1} = \mathbf{a}^{\pi_2}))$$

Action robustness

$$\exists \pi_1 \forall \pi_2 .\ (\mathbf{s}_0{}^{\pi_1} \wedge \mathbf{s}_0{}^{\pi_2}) \wedge (\varphi^{\pi_1} \wedge \varphi^{\pi_2})$$

Symbolic Synthesis from HyperLTL

Computing Required Time-Horizon

- A HyperLTL objective contains multiple paths => unlike with LTL formulas, the required time horizon may be different among the utilized path variables
- $H(\varphi, \pi)$ – the required time horizon for a path variable π in a HyperLTL objective φ

$$H(\mathsf{a}^{\pi}, \pi') = \begin{cases} 0 & \text{if } \pi' = \pi \\ -\infty & \text{otherwise.} \end{cases}$$

$$H(\bigcirc\varphi, \pi) = H(\varphi, \pi) + 1,$$

$$H(\varphi_1\, \mathcal{U}_T\, \varphi_2, \pi) = \max\{H(\varphi_1, \pi), H(\varphi_2, \pi)\} + T$$

$$H(\neg\varphi, \pi) = H(\varphi, \pi),$$

$$H(\varphi_1 \wedge \varphi_2, \pi) = \max\{H(\varphi_1, \pi), H(\varphi_2, \pi)\},$$

$$H(\exists\pi'.\ \psi, \pi) = H(\psi, \pi), \quad H(\forall\pi'.\ \psi, \pi) = H(\psi, \pi)$$

Example:

$$H(\exists\pi_2\forall\pi_1.\ (\mathsf{s_0}^{\pi_1} \wedge \mathsf{s_0}^{\pi_2}) \wedge (\Diamond_T(g^{\pi_2} \Rightarrow \Diamond_T g^{\pi_1})), \pi_1) = H(\Diamond_T(g^{\pi_2} \Rightarrow \Diamond_T g^{\pi_1}), \pi_1) =$$

$$H(\Diamond_T g^{\pi_1}, \pi_1) + T = 2T$$

Model Conversion for SMT-Based Synthesis

- Start from a Discrete-Transition System (DTS) $\mathcal{M}$
- A general HyperLTLf objective φ

$$\varphi = Q_1\pi_1 \ldots Q_n\pi_n \text{ where } Q_i \in \{\exists, \forall\} \text{ for } i \in \{1, \ldots, n\}$$

$$P_i = \bigwedge_{t \in [H_i]} (\mathbf{s}_i(t) = \mathrm{T}_{\mathcal{M}}(\mathbf{s}_i(t-1), \mathbf{a}_i(t-1)))$$

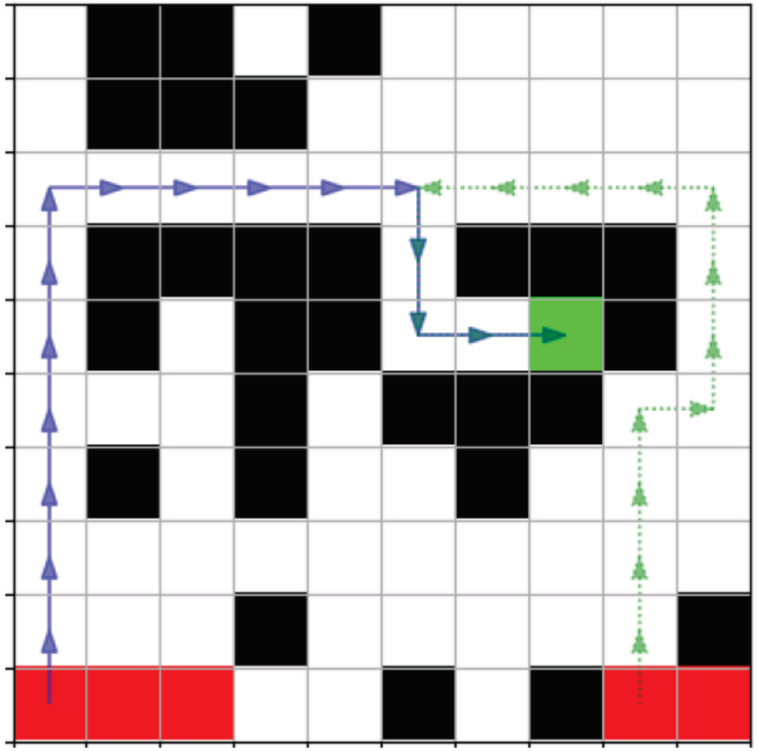
$$[\mathsf{Q}_i\pi_i] = \mathsf{Q}_i\mathbf{s}_i(0)\mathsf{Q}_i\mathbf{a}_i(0)\ldots\mathsf{Q}_i\mathbf{a}_i(H_1 - 1)$$

Resulting in a first-order formula:

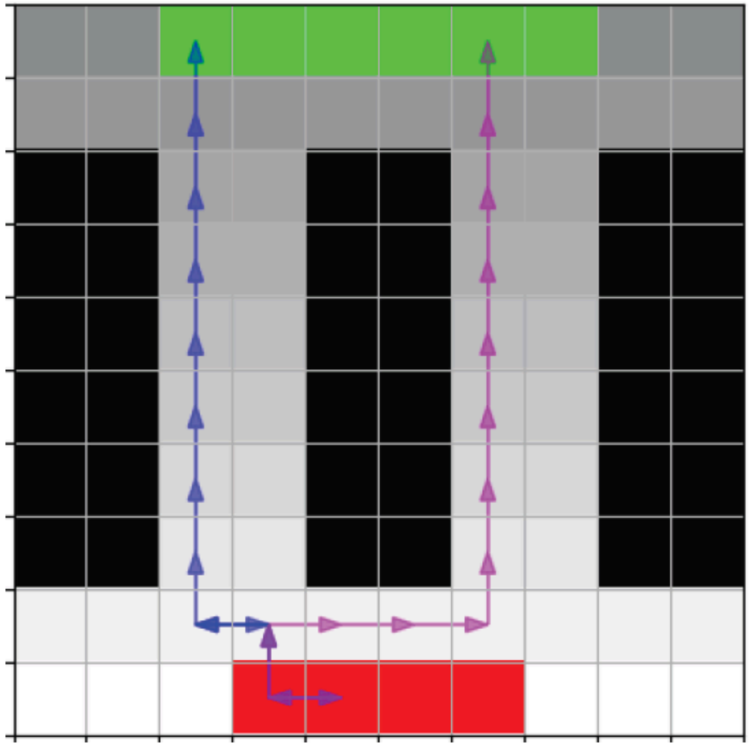
$$[\mathsf{Q}_1\pi_1]\ldots[\mathsf{Q}_n\pi_n].(\bigwedge_{i \in [n]} P_i) \wedge \varphi$$

Can be solved with an SMT solver (e.g., Z3)

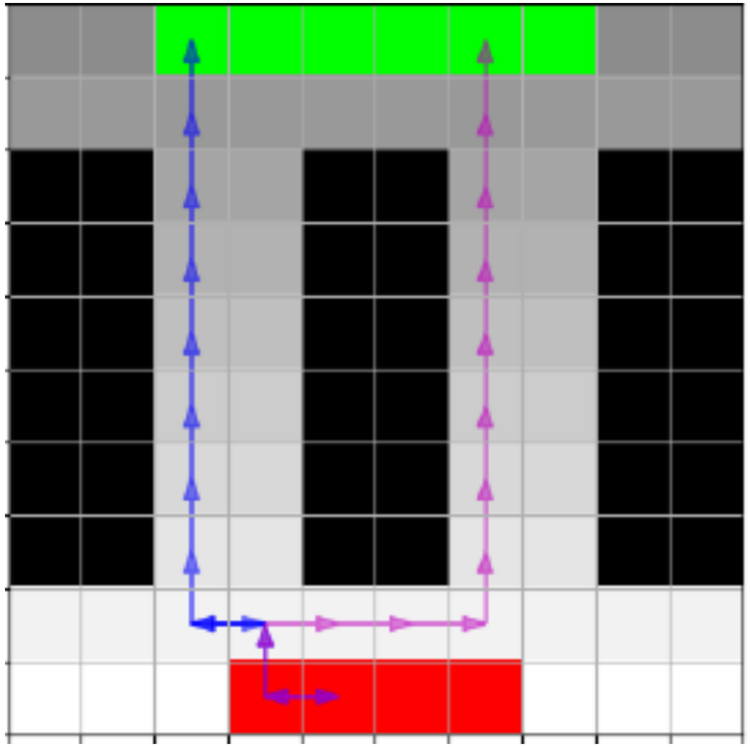
Symbolic Synthesis from HyperLTL [ICRA'20]



Shortest path



Current-state opacity



Initial-state opacity

MPHyper tool – Motion Planning from HyperLTL: https://gitlab.oit.duke.edu/cpsl/mp_hyper

Thank you