Logical Safety Analysis of Concurrent Cyber-physical Systems

Marvin Brieger
marvin.brieger@sosy.ifi.lmu.de

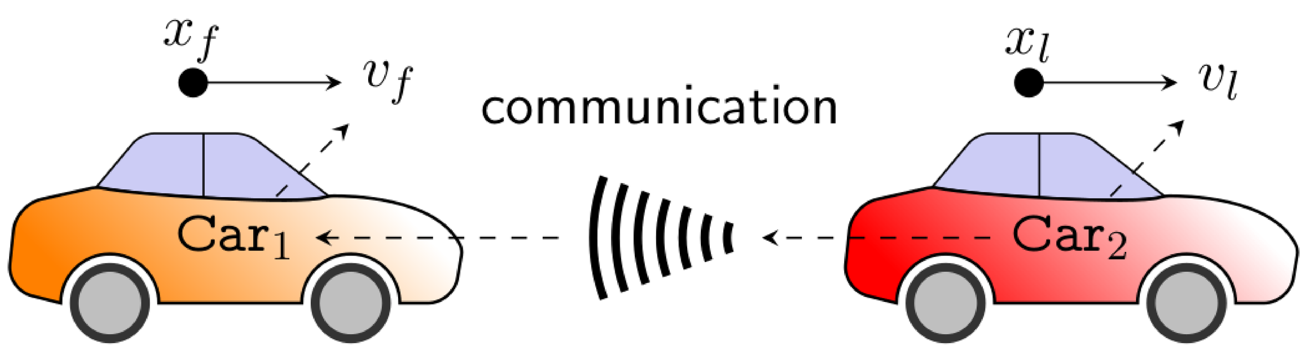
Supervisors: André Platzer (KIT) & Dirk Beyer (LMU)

Collaborators: Stefan Mitsch (CMU)

CONVEY

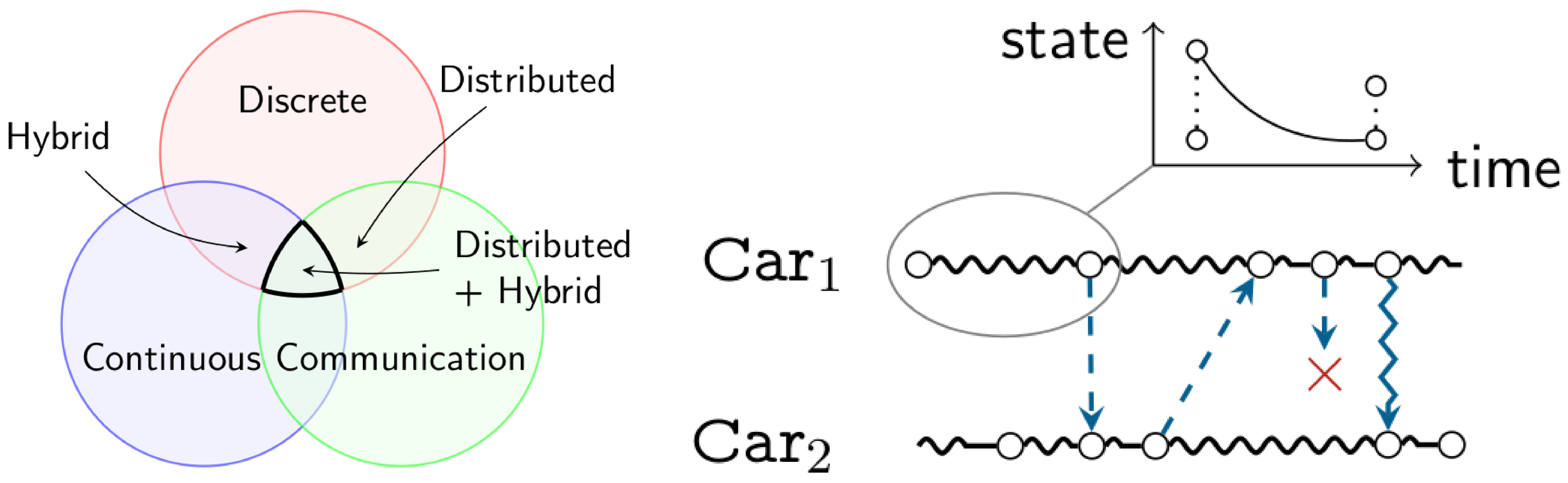
LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Distributed Cyber-physical Systems



Distributed CPS = Discrete + Continuous + Parallel

But more complex than: Parallelism + Hybrid!



Dynamic Logic of Communicating Hybrid Programs

$d\mathcal{L}_{\text{CHP}}$ = $d\mathcal{L}$ + CSP + AC-reasoning + symbols

Definition 1: Communicating Hybrid Programs

$a(|Y,\bar{z}|)$ | $x := \theta \mid \{x' = \theta\} \mid ?\chi \mid \alpha;\beta \mid \alpha \cup \beta \mid \alpha^*$ | $\mathsf{ch}(h)!\theta \mid \mathsf{ch}(h)?x \mid \alpha \parallel \beta$

hybrid programs — communication and parallelism

at most binds channels Y and variables $\bar{z}$

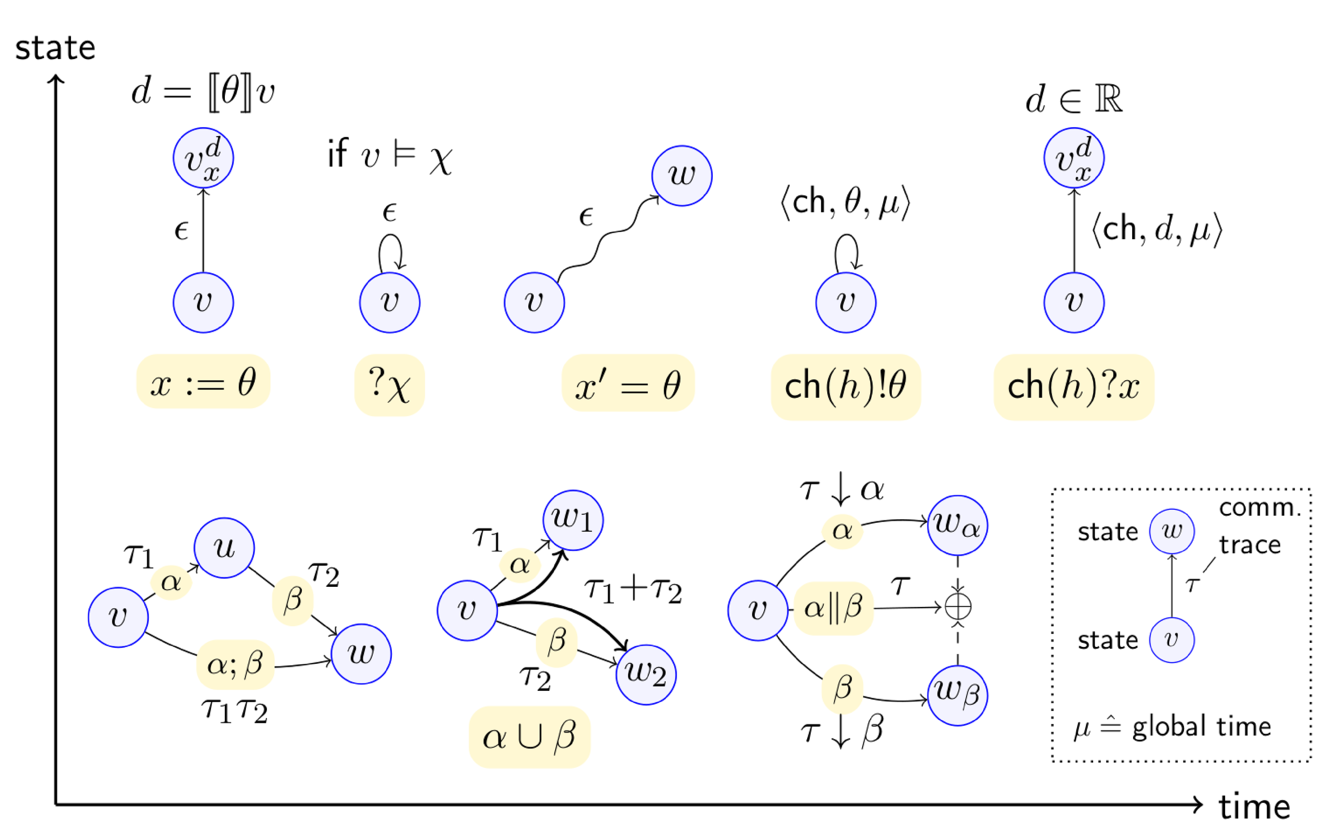
Definition 2: Dynamic assumption-commitment logic

$p(Y,\bar{e})$ | $e_1 \sim e_2 \mid \neg\varphi \mid \varphi \wedge \psi \mid \forall x\, \varphi \mid [\alpha]\psi$ | $[\alpha]_{\{\mathsf{A},\mathsf{C}\}}\psi$

first-order dynamic logic — assumption-commitment modality

at most depends on channels Y and variables $\bar{z}$

Semantics of Communicating Hybrid Programs



References

[1] M. Brieger et al. "Dynamic Logic of Communicating Hybrid Programs". In: *arXiv/CoRR* (2023).

[2] M. Brieger et al. "Uniform Substitution for Dynamic Logic with Communicating Hybrid Programs". In: *CADE (preprint in arXiv/CoRR)*. 2023.

Uniform Substitution

The proof rule US is the **single point of truth** for axiom instantiation!

Theorem 1: Uniform substitution is sound

A substitution σ maps symbols to terms, formulas, or programs.

$$\frac{\phi}{\sigma\phi}\ \text{US}$$ if for each operation $\otimes(e)$ and program constant $a(|Y,\bar{z}|)$ in ϕ:

(I) $\mathsf{FV}(\sigma|_{\Sigma(e)}) \cap \mathsf{BV}(\otimes(\cdot)) = \emptyset$ and $\mathsf{CN}(\sigma|_{\Sigma(e)}) \cap \mathsf{CN}(\otimes(\cdot)) = \emptyset$

(II) $\mathsf{BV}(\sigma a) \subseteq \mathsf{BV}(a(|Y,\bar{z}|))$ and $\mathsf{CN}(\sigma a) = \mathsf{CN}(a(|Y,\bar{z}|))$

Uniform substitution **is sound** if (parameters = variables + channels)

(I) it **never** puts a **free** parameter **into a context** where it is bound

(II) it **never binds** parameters **beyond the original** sets

If you bind a free parameter, you go to logic jail!

Axiomatization by Example

[:=] $[x := g^{\mathbb{R}}]p(x) \leftrightarrow p(g^{\mathbb{R}})$ [;]AC $[a;b]_{\{\mathrm{R},\mathrm{Q}\}}\mathrm{P} \leftrightarrow [a]_{\{\mathrm{R},\mathrm{Q}\}}[b]_{\{\mathrm{R},\mathrm{Q}\}}\mathrm{P}$

[?] $[?q_{\mathbb{R}}]p \leftrightarrow (q_{\mathbb{R}} \to p)$ $[]_{\top,\top}$ $[a]\mathrm{P} \leftrightarrow [a]_{\{\top,\top\}}\mathrm{P}$

[μ] $[\{\bar{x}' = g^{\mathbb{R}}(\bar{x},\mu)\}]p(\bar{x},\mu) \leftrightarrow [\{\mu' = 1, \bar{x}' = g^{\mathbb{R}}(\bar{x},\mu)\}]p(\bar{x},\mu)$

[ch!] $[\mathsf{ch}(h)!g^{\mathbb{R}}]p(\mathsf{ch},h) \leftrightarrow \forall h_0\, \big(h_0 = h \cdot \langle \mathsf{ch}, g^{\mathbb{R}}, \mu\rangle \to p(\mathsf{ch},h_0)\big)$

[ϵ]AC $[a(|\emptyset, V_{\mathbb{R}}|)]_{\{\mathrm{R},\mathrm{Q}\}}\mathrm{P} \leftrightarrow \mathrm{Q} \wedge (\mathrm{R} \to [a(|\emptyset, V_{\mathbb{R}}|)]\mathrm{P})$

$\mathrm{P} \equiv p(Y,\bar{z}) \mid \mathrm{R} \equiv r(Y,\bar{h}) \mid \mathrm{Q} \equiv q(Y,\bar{h}) \mid \sigma g^{\mathbb{R}} \in \mathbb{Q}[V_{\mathbb{R}}] \mid \sigma q_{\mathbb{R}} \in \mathrm{FOL}_{\mathbb{R}}$

Non-schematic Parallel Injection Axiom [||_]AC

$$\frac{[\alpha]\varphi \qquad [\beta]\psi}{[\alpha \parallel \beta](\varphi \wedge \psi)}\ (\star\star)$$

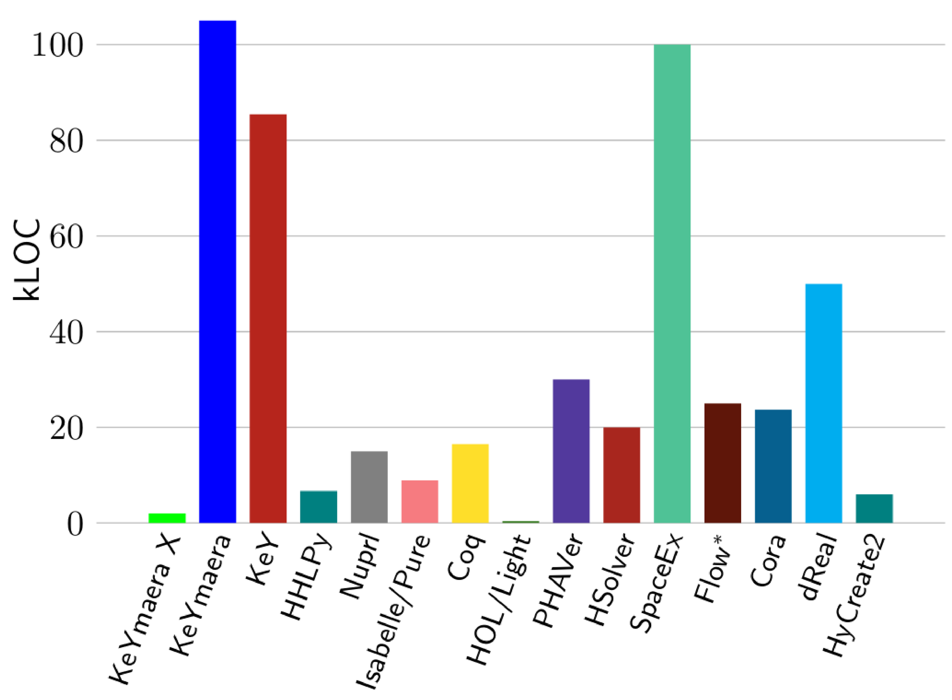
Everyone's favorite parallel proof rule

Requires algorithmic implementation :(

Instead, the parallel injection axiom drops a parallel subprogram if it **does not interfere** with the surronding contract:

$$[\,a(|Y_a,\bar{z}_a|)\,]\,p(Y,\bar{z}) \to [\,a(|Y_a,\bar{z}_a|) \parallel_{\text{wf}} b(|Y_b \cap (Y^{\complement} \cup Y_a), \bar{z}^{\complement}|)\,]\,p(Y,\bar{z})$$

Schematic vs. Flat Axioms



Schematic proof calculus
↓
Algorithmic implementation
↓
Large soundness-critical prover kernel

vs.

KeYmaera X