

# Modelling Events for CPS

---

Enguerrand Prebet

9<sup>th</sup> February 2026

Karlsruhe Institute of Technology

Safe CPS aren't blind.

“In this situation, what decisions should I take?”

Safe CPS aren't blind.

“In this situation, what decisions should I take?”

Today: “In which situation, can/should I take a decision?”

→ How does it affect the first question?

# Event-triggered systems

Optimistic setting: continuous sensing

Define precisely when the controller need to be ran.

## Example (Bouncing Ball)

$$0 \leq x \leq 5 \wedge v \leq 0 \wedge 0 < g \wedge 0 \leq c \leq 1 \wedge 0 \leq f \rightarrow$$
$$[(\{x' = v, v' = -g \ \& \ x \geq 0\};$$
$$\mathbf{if}(x = 0) v := -cv)^*](0 \leq x \leq 5)$$

# Event-triggered systems

Optimistic setting: continuous sensing

Define precisely when the controller need to be ran.

## Example (Bouncing Ball)

$$0 \leq x \leq 5 \wedge v \leq 0 \wedge 0 < g \wedge 0 \leq c \leq 1 \wedge 0 \leq f \rightarrow$$
$$[(\{x' = v, v' = -g \& x \geq 0\};$$
$$\mathbf{i f}(x = 0) v := -cv; \mathbf{i f}(4 \leq x \leq 5) v := -fv)^*](0 \leq x \leq 5)$$

Optimistic setting: continuous sensing

Define precisely when the controller need to be ran.

## Example (Bouncing Ball)

$$\begin{aligned} &0 \leq x \leq 5 \wedge v \leq 0 \wedge 0 < g \wedge 0 \leq c \leq 1 \wedge 0 \leq f \rightarrow \\ &[(\{x' = v, v' = -g \& x \geq 0 \wedge x \leq 5\}; \\ &\mathbf{i}f(x = 0) v := -cv; \mathbf{i}f(4 \leq x \leq 5) v := -fv)^*](0 \leq x \leq 5) \end{aligned}$$

# Event-triggered systems

Optimistic setting: continuous sensing

Define precisely when the controller need to be ran.

## Example (Bouncing Ball)

$$\begin{aligned} &0 \leq x \leq 5 \wedge v \leq 0 \wedge 0 < g \wedge 0 \leq c \leq 1 \wedge 0 \leq f \rightarrow \\ &[(\{x' = v, v' = -g \& x \geq 0 \wedge x \leq 5\}; \\ &\mathbf{i}f(x = 0) v := -cv; \mathbf{i}f(4 \leq x \leq 5) v := -fv)^*](0 \leq x \leq 5) \end{aligned}$$

The law of physics are **non-negotiable!**

# Event-triggered systems

Optimistic setting: continuous sensing

Define precisely when the controller need to be ran.

## Example (Bouncing Ball)

$$0 \leq x \leq 5 \wedge v \leq 0 \wedge 0 < g \wedge 0 \leq c \leq 1 \wedge 0 \leq f \rightarrow$$
$$[\{x' = v, v' = -g \wedge x \geq 0 \wedge x \leq 5\} \cup \{x' = v, v' = -g \wedge x \geq 5\}];$$
$$\mathbf{if}(x = 0) v := -cv; \mathbf{if}(4 \leq x \leq 5) v := -fv)^*(0 \leq x \leq 5)$$

The law of physics are **non-negotiable!**

# Event-triggered systems

Optimistic setting: continuous sensing

Define precisely when the controller need to be ran.

## Example (Bouncing Ball)

$$0 \leq x \leq 5 \wedge v \leq 0 \wedge 0 < g \wedge 0 \leq c \leq 1 \wedge 0 \leq f \rightarrow$$
$$[\{x' = v, v' = -g \wedge x \geq 0 \wedge x \leq 5\} \cup \{x' = v, v' = -g \wedge x \geq 5\});$$
$$\mathbf{if}(x = 0) v := -cv; \mathbf{if}(4 \leq x \leq 5 \wedge \underbrace{v > 0}_{\text{only tap down}}) v := -fv)^*(0 \leq x \leq 5)$$

The law of physics are **non-negotiable!**

## Time-triggered system

Software are discrete not continuous.

Only time causes the controller to act.

$$\begin{aligned} &0 \leq x \leq 5 \wedge v \leq 0 \wedge 0 < g \wedge 0 \leq c \leq 1 \wedge 0 \leq f \rightarrow \\ &[(\mathbf{if}(x = 0) v := -cv; \\ &\{x' = v, v' = -g, t' = 1 \ \& \ x \geq 0\})^*](0 \leq x \leq 5) \end{aligned}$$

## Time-triggered system

Software are discrete not continuous.

Only time causes the controller to act.

$$\begin{aligned} &0 \leq x \leq 5 \wedge v \leq 0 \wedge 0 < g \wedge 0 \leq c \leq 1 \wedge 0 \leq f \rightarrow \\ &[(\mathbf{if}(x = 0) v := -cv; \\ &t := 0; \{x' = v, v' = -g, t' = 1 \ \& \ x \geq 0 \wedge t \leq 1\})^*](0 \leq x \leq 5) \end{aligned}$$

## Time-triggered system

Software are discrete not continuous.

Only time causes the controller to act.

$$\begin{aligned} &0 \leq x \leq 5 \wedge v \leq 0 \wedge 0 < g \wedge 0 \leq c \leq 1 \wedge 0 \leq f \rightarrow \\ &[(\mathbf{if}(x = 0) v := -cv; \mathbf{if}(4 \leq x \leq 5) v := -fv; \\ &t := 0; \{x' = v, v' = -g, t' = 1 \ \& \ x \geq 0 \wedge t \leq 1\})^*](0 \leq x \leq 5) \end{aligned}$$

## Time-triggered system

Software are discrete not continuous.

Only time causes the controller to act.

$$\begin{aligned} &0 \leq x \leq 5 \wedge v \leq 0 \wedge 0 < g \wedge 0 \leq c \leq 1 \wedge 0 \leq f \rightarrow \\ &[(\mathbf{if}(x = 0) v := -cv; \mathbf{if}(4 \leq x \leq 5) v := -fv; \\ &t := 0; \{x' = v, v' = -g, t' = 1 \ \& \ x \geq 0 \wedge t \leq 1\})^*](0 \leq x \leq 5) \end{aligned}$$

Now need to take delay into account

## Time-triggered system

Software are discrete not continuous.

Only time causes the controller to act.

$$0 \leq x \leq 5 \wedge v \leq 0 \wedge 0 < g \wedge 0 \leq c \leq 1 \wedge 0 \leq f \rightarrow \\ [(\mathbf{if}(x = 0) v := -cv; \mathbf{if}(x + v - \frac{g}{2} > 5) v := -fv; \\ t := 0; \{x' = v, v' = -g, t' = 1 \ \& \ x \geq 0 \wedge t \leq 1\})^*](0 \leq x \leq 5)$$

Now need to take delay into account

## Time-triggered system

Software are discrete not continuous.

Only time causes the controller to act.

$$0 \leq x \leq 5 \wedge v \leq 0 \wedge 0 < g \wedge 0 \leq c \leq 1 \wedge 0 \leq f \rightarrow \\ \left[ \left( \mathbf{if}(x = 0) v := -cv; \mathbf{if}(x + v - \frac{g}{2} > 5 \vee \dots) v := -fv; \right. \right. \\ \left. \left. t := 0; \{x' = v, v' = -g, t' = 1 \ \& \ x \geq 0 \wedge t \leq 1\}^* \right) (0 \leq x \leq 5) \right]$$

Now need to take delay into account

The safety condition needs to be **always** true, not only at the end.

→ Tricky to get right

## Time-triggered system

Software are discrete not continuous.

Only time causes the controller to act.

$$0 \leq x \leq 5 \wedge v \leq 0 \wedge 0 < g \wedge 0 \leq c \leq 1 \wedge 0 \leq f \rightarrow$$
$$\left[ \left( \mathbf{if}(x = 0) v := -cv; \mathbf{if}(x + v - \frac{g}{2} > 5 \vee (x + \frac{v^2}{2g} > 5 \wedge v - g < 0)) v := -fv; \right. \right.$$
$$\left. t := 0; \{x' = v, v' = -g, t' = 1 \ \& \ x \geq 0 \wedge t \leq 1\}^* \right] (0 \leq x \leq 5)$$

Now need to take delay into account

The safety condition needs to be **always** true, not only at the end.

→ Tricky to get right

# Summary

	Event-triggered	Time-triggered
Sensing	Continuous sensing	Discrete sensing
Easier to	Verify	Implement
More info	Chapter 08[3]	Chapter 09[3]

Physics matters!

# Summary

	Event-triggered	Time-triggered
Sensing	Continuous sensing	Discrete sensing
Easier to	Verify	Implement
More info	Chapter 08[3]	Chapter 09[3]

Physics matters!

Use verified event-triggered model as basis for time-triggered model

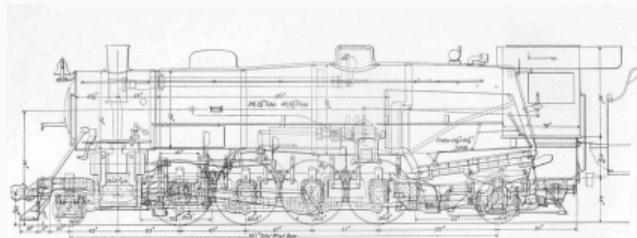
# Relating CPS via Differential Refinement Logic

---

Enguerrand Prebet

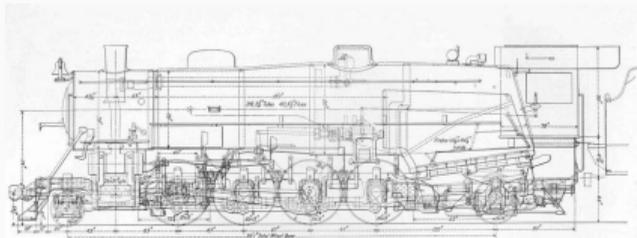
9<sup>th</sup> February 2026

Karlsruhe Institute of Technology



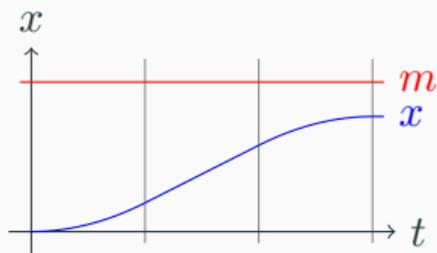
[train]safe

# Cyber-Physical Systems

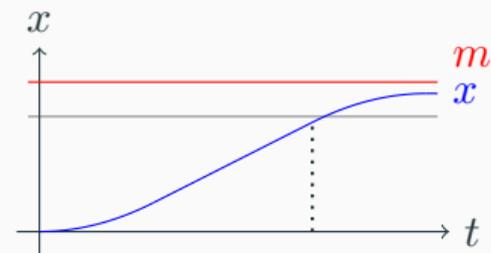


[train]safe

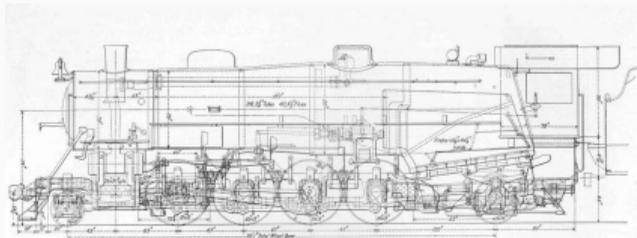
Time-triggered



Event-triggered

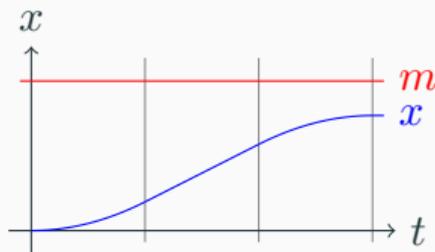


# Cyber-Physical Systems

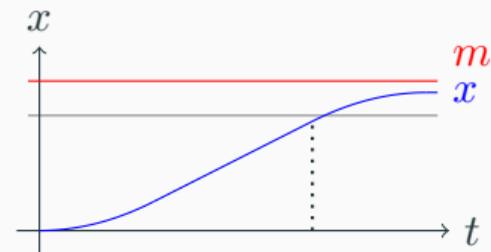


[train]safe

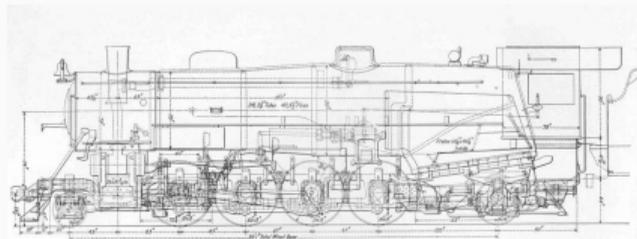
Time-triggered  
Implementable



Event-triggered

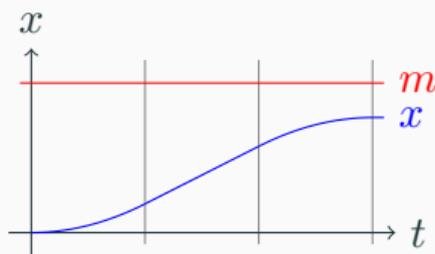


# Cyber-Physical Systems

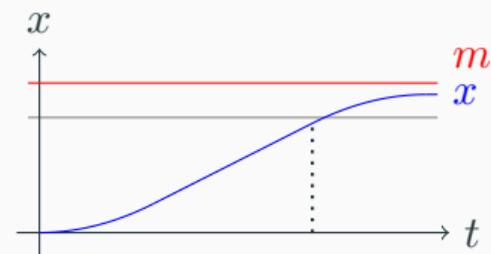


[train]safe

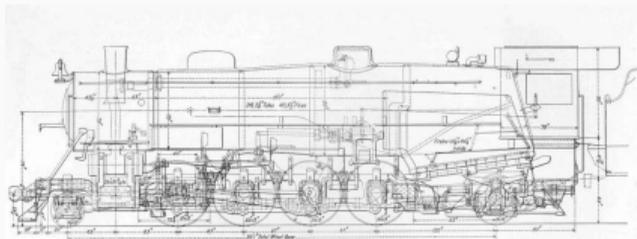
Time-triggered  
Implementable



Event-triggered  
Verifiable

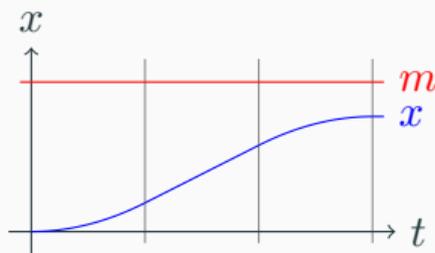


# Cyber-Physical Systems

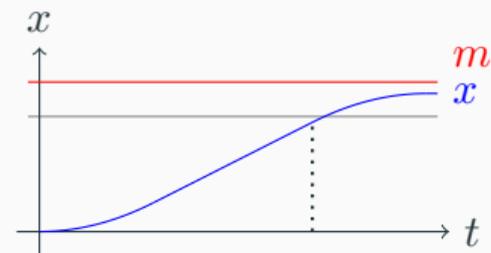


$[\text{train}]_{\text{safe}}$

Time-triggered  
Implementable

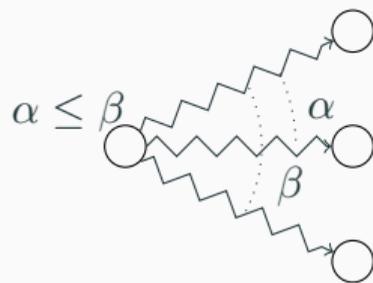
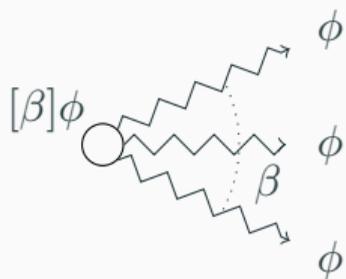


Event-triggered  
Verifiable



$[\text{time}]_{\text{safe}} \leftarrow [\text{event}]_{\text{safe}} ?$

[time]safe  $\leftarrow$  [event]safe ?



Relating the two systems...  $\rightarrow$  refinement [2]  
...in a trusted way.  $\rightarrow$  uniform substitution

# Differential Refinement Logic (dRL)

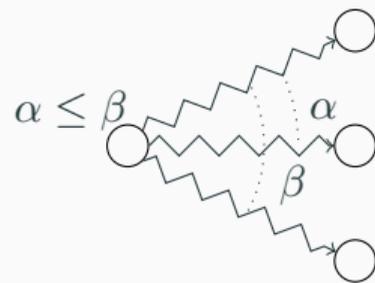
dL  $\subseteq$  dRL

Term:  $\theta, \eta ::= x \mid \theta + \eta \mid \theta \cdot \eta \mid (\theta)'$

Formula:  $\phi, \psi ::= \theta \leq \eta \mid \neg\phi \mid \phi \wedge \psi \mid \forall x \phi \mid [\alpha]\phi \mid \alpha \leq \beta \mid \alpha = \beta$

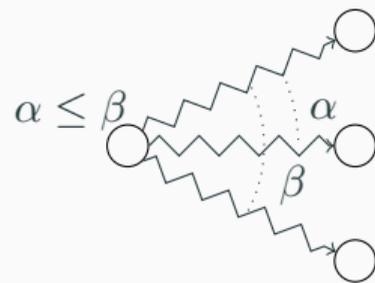
Program:  $\alpha, \beta ::= ?\psi \mid x := \theta \mid x := * \mid \alpha \cup \beta \mid \alpha; \beta \mid \alpha^* \mid x' = \theta \& \psi$

$\nu \in \llbracket \alpha \leq \beta \rrbracket$  iff  $(\nu, \omega) \in \llbracket \beta \rrbracket$  for all  $(\nu, \omega) \in \llbracket \alpha \rrbracket$



Quiz time:

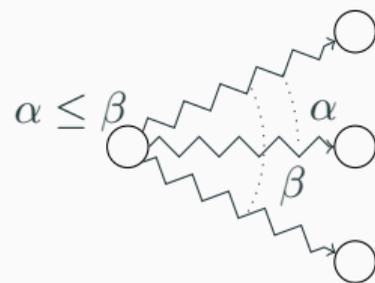
$\nu \in \llbracket \alpha \leq \beta \rrbracket$  iff  $(\nu, \omega) \in \llbracket \beta \rrbracket$  for all  $(\nu, \omega) \in \llbracket \alpha \rrbracket$



Quiz time:

$$?x = 0 \leq ?x \geq 0$$

$\nu \in \llbracket \alpha \leq \beta \rrbracket$  iff  $(\nu, \omega) \in \llbracket \beta \rrbracket$  for all  $(\nu, \omega) \in \llbracket \alpha \rrbracket$

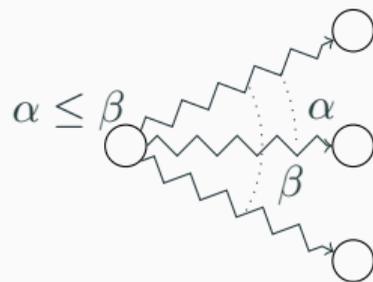


Quiz time:

$$?x = 0 \leq ?x \geq 0$$

✓

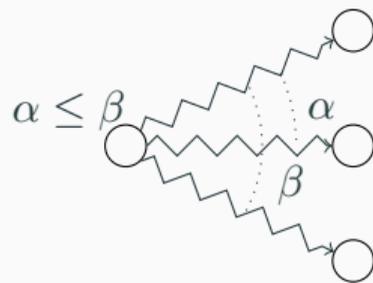
$\nu \in \llbracket \alpha \leq \beta \rrbracket$  iff  $(\nu, \omega) \in \llbracket \beta \rrbracket$  for all  $(\nu, \omega) \in \llbracket \alpha \rrbracket$



Quiz time:

$$\begin{array}{l} ?x = 0 \leq ?x \geq 0 \\ x := 2x \leq (?x \geq 0; ?x \leq 0 \cup x := 2x) \end{array} \quad \checkmark$$

$\nu \in \llbracket \alpha \leq \beta \rrbracket$  iff  $(\nu, \omega) \in \llbracket \beta \rrbracket$  for all  $(\nu, \omega) \in \llbracket \alpha \rrbracket$

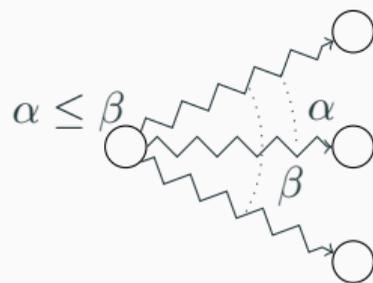


Quiz time:

$$?x = 0 \leq ?x \geq 0 \quad \checkmark$$

$$x := 2x \leq (?x \geq 0; ?x \leq 0 \cup x := 2x) \quad \checkmark$$

$\nu \in \llbracket \alpha \leq \beta \rrbracket$  iff  $(\nu, \omega) \in \llbracket \beta \rrbracket$  for all  $(\nu, \omega) \in \llbracket \alpha \rrbracket$



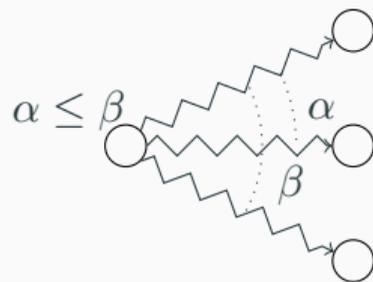
Quiz time:

$$?x = 0 \leq ?x \geq 0 \quad \checkmark$$

$$x := 2x \leq (?x \geq 0; ?x \leq 0 \cup x := 2x) \quad \checkmark$$

$$(?x \geq 0; ?x \leq 0 \cup x := 2x) \leq x := 2x$$

$\nu \in \llbracket \alpha \leq \beta \rrbracket$  iff  $(\nu, \omega) \in \llbracket \beta \rrbracket$  for all  $(\nu, \omega) \in \llbracket \alpha \rrbracket$



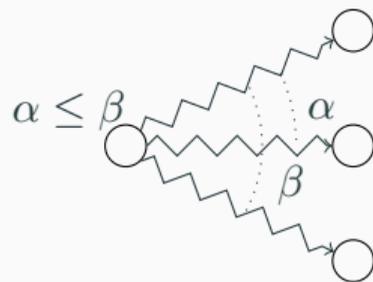
Quiz time:

$$?x = 0 \leq ?x \geq 0 \quad \checkmark$$

$$x := 2x \leq (?x \geq 0; ?x \leq 0 \cup x := 2x) \quad \checkmark$$

$$(?x \geq 0; ?x \leq 0 \cup x := 2x) \leq x := 2x \quad \checkmark!$$

$\nu \in \llbracket \alpha \leq \beta \rrbracket$  iff  $(\nu, \omega) \in \llbracket \beta \rrbracket$  for all  $(\nu, \omega) \in \llbracket \alpha \rrbracket$



Quiz time:

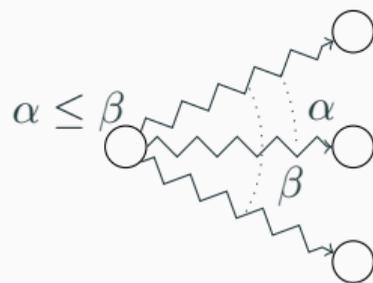
$$?x = 0 \leq ?x \geq 0 \quad \checkmark$$

$$x := 2x \leq (?x \geq 0; ?x \leq 0 \cup x := 2x) \quad \checkmark$$

$$(?x \geq 0; ?x \leq 0 \cup x := 2x) \leq x := 2x \quad \checkmark$$

$$x := y = x := 0$$

$\nu \in \llbracket \alpha \leq \beta \rrbracket$  iff  $(\nu, \omega) \in \llbracket \beta \rrbracket$  for all  $(\nu, \omega) \in \llbracket \alpha \rrbracket$



Quiz time:

$$?x = 0 \leq ?x \geq 0 \quad \checkmark$$

$$x := 2x \leq (?x \geq 0; ?x \leq 0 \cup x := 2x) \quad \checkmark$$

$$(?x \geq 0; ?x \leq 0 \cup x := 2x) \leq x := 2x \quad \checkmark$$

$$x := y = x := 0 \quad \text{iff } y = 0$$

## Using refinement

dL: reasoning according to the structure of the HP

dRL: allows changing the structure itself

$$\text{(Congr)} \quad \frac{\alpha = \beta}{C(\alpha) \leftrightarrow C(\beta)}$$

$$\alpha \leq \beta \rightarrow ([\alpha]\phi \leftarrow [\beta]\phi)$$

# dRL Calculus and Uniform Substitution

---

# dRL Axioms: the very basics

All dL axioms...

Structural rewriting:

$$(\leq_{\text{refl}}) \quad \alpha \leq \alpha$$

$$(\cup_{\text{idemp}}) \quad \alpha \cup \alpha = \alpha$$

$$(\cup_{\text{assoc}}) \quad (\alpha \cup \beta) \cup \gamma = \alpha \cup (\beta \cup \gamma)$$

$$(\text{id-l}) \quad ?\text{true}; \alpha = \alpha$$

$$(\text{dist-l}) \quad (\alpha; \beta) \cup (\alpha; \gamma) = \alpha; (\beta \cup \gamma)$$

$$(\text{annih-l}) \quad ?\text{false}; \alpha = ?\text{false}$$

$$(\text{and}) \quad ?\phi; ?\psi = ?(\phi \wedge \psi)$$

$$(\text{unfold-l}) \quad ?\text{true} \cup (\alpha; \alpha^*) = \alpha^*$$

$$(\cup_{\text{id}}) \quad \alpha \cup ?\text{false} = \alpha$$

$$(\cup_{\text{comm}}) \quad \beta \cup \alpha = \alpha \cup \beta$$

$$(\text{;assoc}) \quad \alpha; (\beta; \gamma) = (\alpha; \beta); \gamma$$

$$(\text{id-r}) \quad \alpha; ?\text{true} = \alpha$$

$$(\text{dist-r}) \quad (\alpha; \gamma) \cup (\beta; \gamma) = (\alpha \cup \beta); \gamma$$

$$(\text{annih-r}) \quad \alpha; ?\text{false} = ?\text{false}$$

$$(\text{or}) \quad ?\phi \cup ?\psi = ?(\phi \vee \psi)$$

$$(\text{unfold-r}) \quad ?\text{true} \cup (\alpha^*; \alpha) = \alpha^*$$

Kleene Algebra (with tests) [1]

## (Some) Axioms of dRL

dRL-specific axioms:

$$([\leq]) \quad \alpha \leq \beta \rightarrow ([\alpha]\phi \leftarrow [\beta]\phi) \qquad (:) \quad \alpha; \beta \leq \gamma; \delta \leftarrow \alpha \leq \gamma \wedge [\alpha]\beta \leq \delta$$

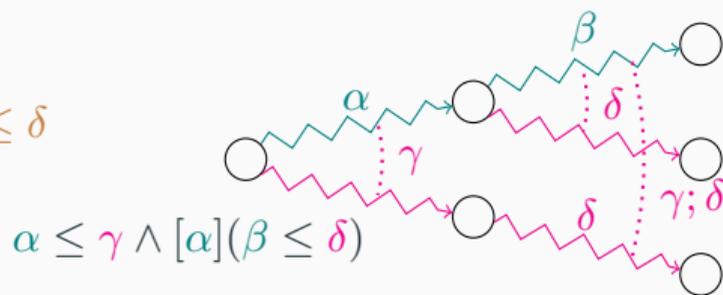
$$(\cup_l) \quad \alpha \cup \beta \leq \gamma \leftrightarrow \alpha \leq \gamma \wedge \beta \leq \gamma \qquad (\cup_r) \quad \alpha \leq \beta \cup \gamma \leftarrow \alpha \leq \beta \vee \alpha \leq \gamma$$

$$(\leq_t) \quad \alpha \leq \beta \leftarrow (\alpha \leq \gamma \wedge \gamma \leq \beta) \qquad (\text{unloop}) \quad \alpha^* \leq \beta^* \leftarrow [\alpha^*](\alpha \leq \beta)$$

$$(\text{loop}_r) \quad \alpha; \beta^* \leq \alpha \leftarrow \alpha; \beta \leq \alpha \qquad (\text{loop}_l) \quad \alpha^*; \beta \leq \beta \leftarrow [\alpha^*]\alpha; \beta \leq \beta$$

$$(\text{ODE}) \quad x' = \theta \& \phi \leq x' = \eta \& \psi \leftrightarrow [x' = \theta \& \phi](x' = \eta \wedge \psi)$$

$$(;)\quad \alpha; \beta \leq \gamma; \delta \leftarrow \alpha \leq \gamma \wedge [\alpha]\beta \leq \delta$$



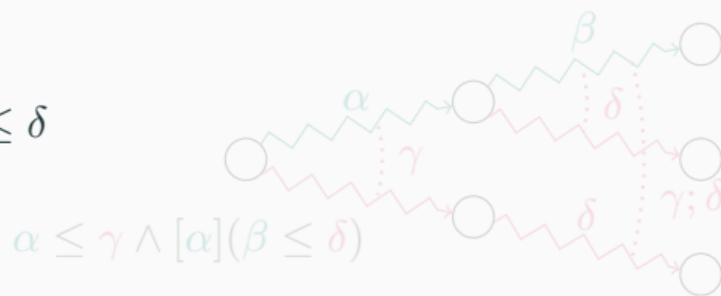
$$(\text{unloop})\quad \alpha^* \leq \beta^* \leftarrow [\alpha^*](\alpha \leq \beta)$$

$$[\alpha^*](\alpha \leq \beta) \rightarrow \underbrace{(\alpha \leq \beta \wedge [\alpha]\alpha \leq \beta \wedge [\alpha; \alpha]\alpha \leq \beta \wedge \dots)}_{\alpha; \alpha \leq \beta; \beta}$$

$$(\text{ODE})\quad x' = \theta \& \phi \leq x' = \eta \& \psi \leftrightarrow [x' = \theta \& \phi](x' = \eta \wedge \psi)$$

# Axioms soundness

$$(;)\quad \alpha; \beta \leq \gamma; \delta \leftarrow \alpha \leq \gamma \wedge [\alpha] \beta \leq \delta$$

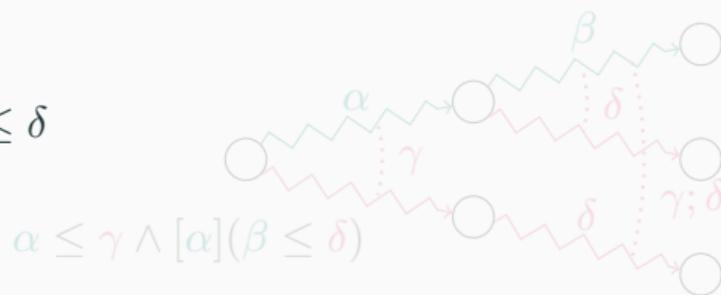


$$(\text{unloop})\quad \alpha^* \leq \beta^* \leftarrow [\alpha^*](\alpha \leq \beta)$$

$$[\alpha^*](\alpha \leq \beta) \rightarrow (\alpha \leq \beta \wedge [\alpha] \alpha \leq \beta \wedge [\alpha; \alpha] \alpha \leq \beta \wedge \dots)$$

$$(\text{ODE})\quad x' = \theta \& \phi \leq x' = \eta \& \psi \leftrightarrow [x' = \theta \& \phi](x' = \eta \wedge \psi)$$

$$(;)\quad \alpha; \beta \leq \gamma; \delta \leftarrow \alpha \leq \gamma \wedge [\alpha] \beta \leq \delta$$

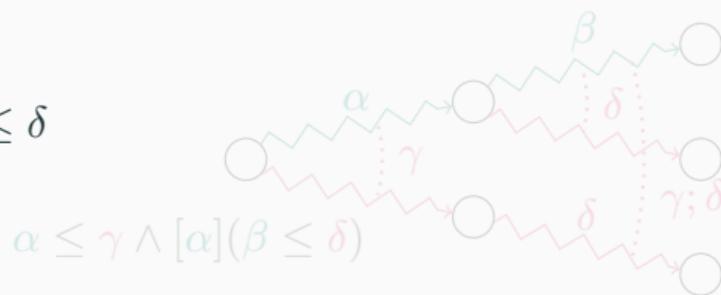


$$(\text{unloop})\quad \alpha^* \leq \beta^* \leftarrow [\alpha^*](\alpha \leq \beta)$$

$$[\alpha^*](\alpha \leq \beta) \rightarrow \underbrace{(\alpha \leq \beta \wedge [\alpha] \alpha \leq \beta \wedge [\alpha; \alpha] \alpha \leq \beta \wedge \dots)}_{\alpha; \alpha \leq \beta; \beta}$$

$$(\text{ODE})\quad x' = \theta \& \phi \leq x' = \eta \& \psi \leftrightarrow [x' = \theta \& \phi](x' = \eta \wedge \psi)$$

$$(;)\quad \alpha; \beta \leq \gamma; \delta \leftarrow \alpha \leq \gamma \wedge [\alpha] \beta \leq \delta$$



$$(\text{unloop})\quad \alpha^* \leq \beta^* \leftarrow [\alpha^*](\alpha \leq \beta)$$

$$[\alpha^*](\alpha \leq \beta) \rightarrow \underbrace{(\alpha \leq \beta \wedge [\alpha] \alpha \leq \beta \wedge [\alpha; \alpha] \alpha \leq \beta \wedge \dots)}_{\alpha; \alpha \leq \beta; \beta}$$

$$(\text{ODE})\quad x' = \theta \& \phi \leq x' = \eta \& \psi \leftrightarrow [x' = \theta \& \phi](x' = \eta \wedge \psi)$$

## Relating event and time

$$\begin{array}{c} \overbrace{((a := c \cup a := *; ?\text{Safe}_E); t := 0; (\{x' = f(x), t' = 1 \& E(x) \geq 0\} \\ \cup \{x' = f(x), t' = 1 \& E(x) \leq 0\}))}^{\text{ctrl}_E} \\ \vee \\ \underbrace{((a := c \cup a := *; ?\text{Safe}_T(T)); t := 0; \{x' = f(x), t' = 1 \& t \leq T\})}_{\text{ctrl}_T} \underbrace{\quad}_{\text{ode}} \end{array}^*$$

## Relating event and time

$$\begin{array}{c} \text{ctrl}_E \\ \underbrace{\left( (a := c \cup a := *; ?\text{Safe}_E); t := 0; \left( \{x' = f(x), t' = 1 \ \& \ E(x) \geq 0\} \right. \right. \\ \left. \left. \cup \{x' = f(x), t' = 1 \ \& \ E(x) \leq 0\} \right) \right)^* \\ \vee \\ \underbrace{\left( (a := c \cup a := *; ?\text{Safe}_T(T)); t := 0; \{x' = f(x), t' = 1 \ \& \ t \leq T\} \right)^*}_{\text{ctrl}_T} \quad \underbrace{\hspace{10em}}_{\text{ode}} \end{array}$$

General proof tree structure on board

# Uniform Substitution

Axioms are formulas, instantiations are recovered by uniform substitution.

Term:  $\theta, \eta ::= \dots \mid f(\theta)$

$\llbracket \cdot \rrbracket : \mathcal{I} \rightarrow \mathcal{S} \rightarrow \mathbb{R}$

Formula:  $\phi, \psi ::= \dots \mid p(\theta)$

$\llbracket \cdot \rrbracket : \mathcal{I} \rightarrow \mathcal{P}(\mathcal{S})$

Program:  $\alpha, \beta ::= \dots \mid a$

$\llbracket \cdot \rrbracket : \mathcal{I} \rightarrow \mathcal{P}(\mathcal{S} \times \mathcal{S})$

## (Some) Axioms of dRL with constants

dRL-specific axioms:

$$([\leq]) \quad \alpha \leq \beta \rightarrow ([\alpha]\phi \leftarrow [\beta]\phi) \qquad (:) \quad \alpha; \beta \leq \gamma; \delta \leftarrow \alpha \leq \gamma \wedge [\alpha]\beta \leq \delta$$

$$(\cup_l) \quad \alpha \cup \beta \leq \gamma \leftrightarrow \alpha \leq \gamma \wedge \beta \leq \gamma \qquad (\cup_r) \quad \alpha \leq \beta \cup \gamma \leftarrow \alpha \leq \beta \vee \alpha \leq \gamma$$

$$(\leq_t) \quad \alpha \leq \beta \leftarrow (\alpha \leq \gamma \wedge \gamma \leq \beta) \qquad (\text{unloop}) \quad \alpha^* \leq \beta^* \leftarrow [\alpha^*](\alpha \leq \beta)$$

$$(\text{loop}_r) \quad \alpha; \beta^* \leq \alpha \leftarrow \alpha; \beta \leq \alpha \qquad (\text{loop}_l) \quad \alpha^*; \beta \leq \beta \leftarrow [\alpha^*]\alpha; \beta \leq \beta$$

$$(\text{ODE}) \quad x' = \theta \ \& \ \phi \leq x' = \eta \ \& \ \psi \leftrightarrow [x' = \theta \ \& \ \phi](x' = \eta \ \& \ \psi)$$

## (Some) Axioms of dRL with constants

dRL-specific axioms:

$$([\leq]) \quad a \leq b \rightarrow ([a]\phi \leftarrow [b]\phi) \qquad (;) \quad a; b \leq c; d \leftarrow a \leq c \wedge [a]b \leq d$$

$$(\cup_l) \quad a \cup b \leq c \leftrightarrow a \leq c \wedge b \leq c \qquad (\cup_r) \quad a \leq b \cup c \leftarrow a \leq b \vee a \leq c$$

$$(\leq_t) \quad a \leq b \leftarrow (a \leq c \wedge c \leq b) \qquad (\text{unloop}) \quad a^* \leq b^* \leftarrow [a^*](a \leq b)$$

$$(\text{loop}_r) \quad a; b^* \leq a \leftarrow a; b \leq a \qquad (\text{loop}_l) \quad a^*; b \leq b \leftarrow [a^*]a; b \leq b$$

$$(\text{ODE}) \quad x' = f(x) \ \& \ p(x) \leq x' = g(x) \ \& \ q(x) \leftrightarrow [x' = f(x) \ \& \ p(x)](x' = g(x) \ \& \ q(x))$$

## Soundness of Uniform Substitution

$$(US) \quad \frac{\phi}{\sigma(\phi)} \text{ if } \sigma(\phi) \text{ defined}$$

“Do not introduce new **free** variables in a context where they are **bound**.”

Additional case:  $\sigma(\alpha \leq \beta) = \sigma(\alpha) \leq \sigma(\beta)$

And... that's it?

The free variables of a refinement is not defined.

$$FV(\alpha \leq \beta) = FV(\alpha) \cup FV(\beta) ?$$

The free variables of a refinement is not defined.

$$FV(\alpha \leq \beta) = FV(\alpha) \cup FV(\beta) ?$$

$$(?y = 1 \leq x := 0) \leftrightarrow (y = 1 \rightarrow x = 0)$$

The free variables of a refinement is not defined.

$$FV(\alpha \leq \beta) = FV(\alpha) \cup FV(\beta) \cup ((BV(\alpha) \cup BV(\beta)) \setminus (MBV(\alpha) \cap MBV(\beta)))$$

$$(?y = 1 \leq x := 0) \leftrightarrow (y = 1 \rightarrow x = 0)$$

- Hybrid system refinement
  - CPS proofs no longer bound by the HP shape
  - transfer event-triggered systems' safety to time-triggered systems
- Uniform substitution-style calculus
  - easily implemented in KeYmaera X

# References

-  Dexter Kozen.  
**Kleene algebra with tests.**  
*ACM Trans. Program. Lang. Syst.*, 19(3):427–443, 1997.
-  Sarah M. Loos and André Platzer.  
**Differential refinement logic.**  
In *LICS*, pages 505–514, 2016.
-  André Platzer.  
**Logical Foundations of Cyber-Physical Systems.**  
Springer, 2018.
-  André Platzer and Yong Kiam Tan.  
**Differential equation invariance axiomatization.**  
*J. ACM*, 67(1):6:1–6:66, 2020.
-  Enguerrand Prebet and André Platzer.  
**Uniform substitution for differential refinement logic.**  
In Chris Benzmüller, Marijn Heule, and Renate Schmidt, editors, *IJCAR*, LNCS. Springer, 2024.