André Platzer

Assistant Professor

Computer Science

Carnegie Mellon University

Logical Analysis of Hybrid Systems

The KeYmaera Approach

TA: Jan-David Quesel, Ph.D. student, University of Oldenburg, Germany

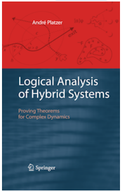
Pointers and Resources with More Information

- These slides are based on invited tutorial at LICS 2012 ▸ slides

  André Platzer.
  Logics of dynamical systems.
  LICS, 2012.
  Invited tutorial

- Read LICS 2012 invited tutorial write-up ▸ paper ▸ extended
- More details in the book ▸ book

  André Platzer.
  Logical Analysis of Hybrid Systems:
  Proving Theorems for Complex Dynamics.
  Springer, 2010.



- Verification tool KeYmaera and more ▸ KeYmaera
- Many more references in bibliography on slides, paper, web ▸ web

http://symbolaris.com/

Outline

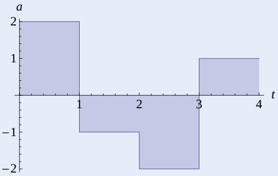
Outline

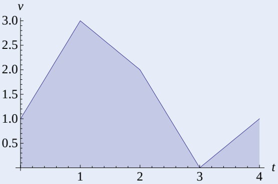
How can we design computers that are guaranteed to interact correctly with the physical world?

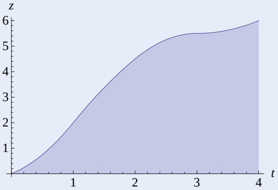
Challenge (Hybrid Systems)

Fixed rule describing state evolution with both

- Continuous dynamics (differential equations)
- Discrete dynamics (control decisions)







Challenge (Hybrid Systems)

Fixed rule describing state evolution with both

- Continuous dynamics (differential equations)
- Discrete dynamics (control decisions)

1. More than computers: no `NullPointerException` $\not\Rightarrow$ safe

Hybrid Systems Analysis: Car Control

Challenge (Hybrid Systems)

Fixed rule describing state evolution with both

- Continuous dynamics (differential equations)
- Discrete dynamics (control decisions)

1. More than computers: no `NullPointerException` $\not\Rightarrow$ safe
2. More than physics: braking control $v^2 \leq 2b(MA - z)$ $\not\Rightarrow$ safe

Hybrid Systems Analysis: Car Control

Challenge (Hybrid Systems)

Fixed rule describing state evolution with both

- Continuous dynamics (differential equations)
- Discrete dynamics (control decisions)

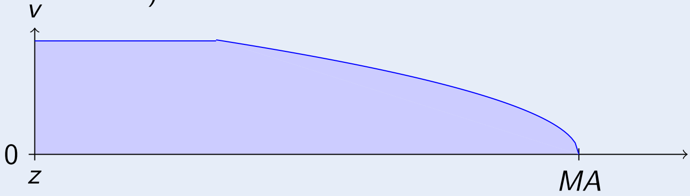
1. More than computers: no `NullPointerException` $\not\Rightarrow$ safe
2. More than physics: braking control $v^2 \leq 2b(MA - z) \not\Rightarrow$ safe
3. Joint dynamics requires: $SB \geq \frac{v^2}{2b} + \frac{a^2\varepsilon^2}{2b} + \frac{a}{b}\varepsilon v + \frac{a}{2}\varepsilon^2 + \varepsilon v \ldots$

Hybrid Systems Analysis: Car Control

Challenge (Hybrid Systems)

Fixed rule describing state evolution with both

- Continuous dynamics (differential equations)
- Discrete dynamics (control decisions)

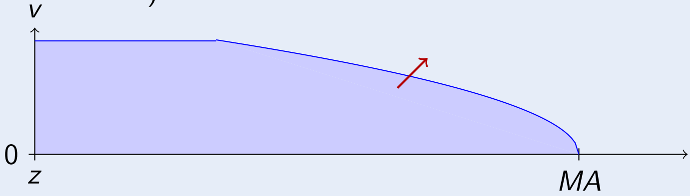


Hybrid Systems Analysis: Car Control

Challenge (Hybrid Systems)

Fixed rule describing state evolution with both

- Continuous dynamics (differential equations)
- Discrete dynamics (control decisions)

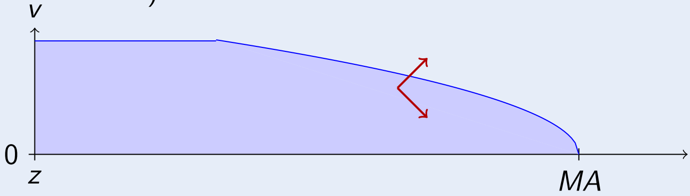


Hybrid Systems Analysis: Car Control

Challenge (Hybrid Systems)

Fixed rule describing state evolution with both

- Continuous dynamics (differential equations)
- Discrete dynamics (control decisions)

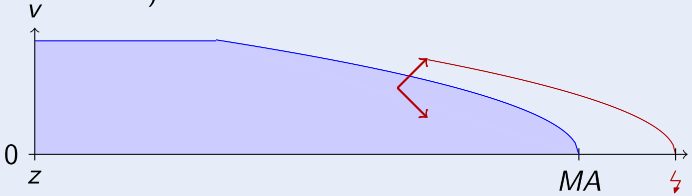


Hybrid Systems Analysis: Car Control

Challenge (Hybrid Systems)

Fixed rule describing state evolution with both

- Continuous dynamics (differential equations)
- Discrete dynamics (control decisions)

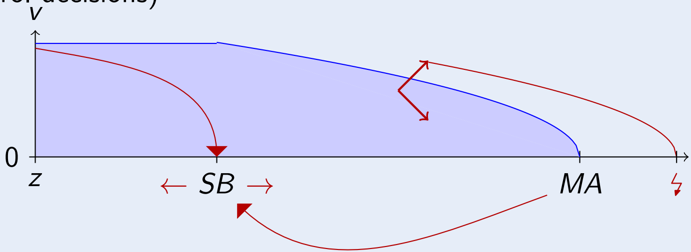


Hybrid Systems Analysis: Car Control

Challenge (Hybrid Systems)

Fixed rule describing state evolution with both

- Continuous dynamics (differential equations)
- Discrete dynamics (control decisions)

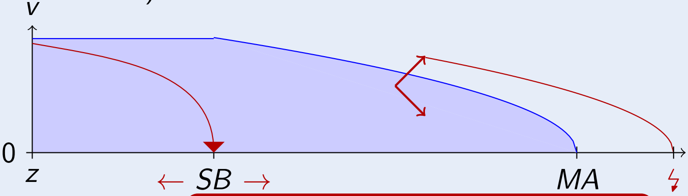


Hybrid Systems Analysis: Car Control

Challenge (Hybrid Systems)

Fixed rule describing state evolution with both

- Continuous dynamics (differential equations)
- Discrete dynamics (control decisions)

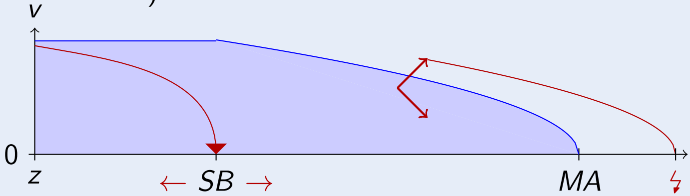


$$SB \geq \frac{v^2}{2b} + \frac{a^2\varepsilon^2}{2b} + \frac{a}{b}\varepsilon v + \frac{a}{2}\varepsilon^2 + \varepsilon v$$

Challenge (Hybrid Systems)

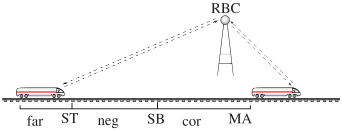
Fixed rule describing state evolution with both

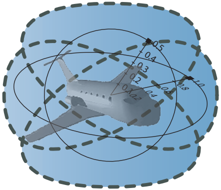
- Continuous dynamics (differential equations)
- Discrete dynamics (control decisions)

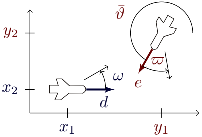


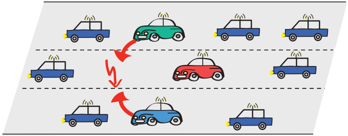
$\forall MA\, \exists SB$ "Car always safe"

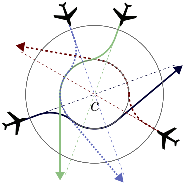
Hybrid Systems Analysis is Important for . . .

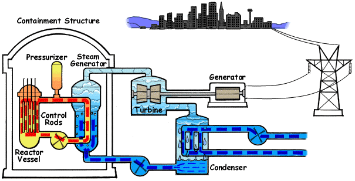


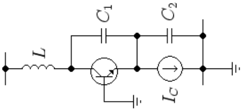












Outline

Outline

Differential Dynamic Logic for Hybrid Systems

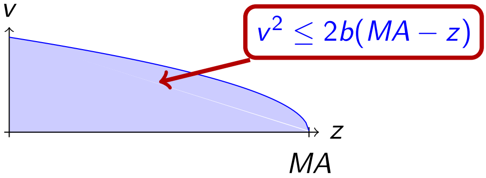
differential dynamic logic

$d\mathcal{L} = \quad \mathrm{DL} + \mathrm{HP}$

Differential Dynamic Logic for Hybrid Systems

differential dynamic logic

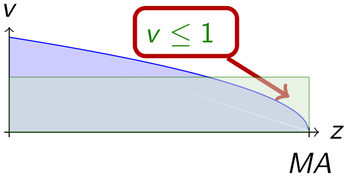
$d\mathcal{L} = \text{FOL}_{\mathbb{R}}$



Differential Dynamic Logic for Hybrid Systems

differential dynamic logic

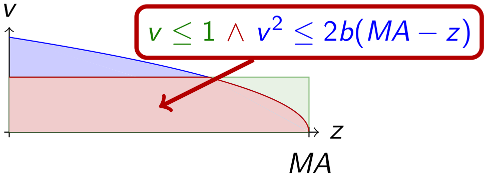
$d\mathcal{L} = \mathrm{FOL}_{\mathbb{R}}$



Differential Dynamic Logic for Hybrid Systems

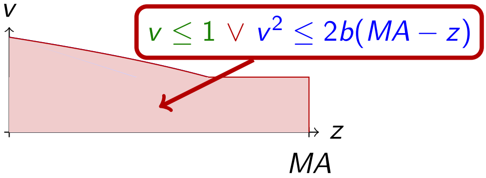
differential dynamic logic

$\mathrm{d}\mathcal{L} = \mathrm{FOL}_{\mathbb{R}}$



differential dynamic logic

$d\mathcal{L} = \mathrm{FOL}_{\mathbb{R}}$

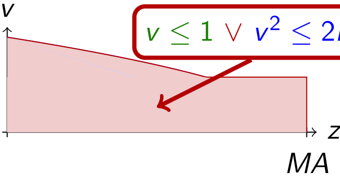


differential dynamic logic

$d\mathcal{L} = \mathrm{FOL}_{\mathbb{R}}$

$\forall MA\, \exists SB\, \ldots$

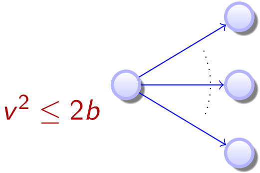
$\forall t{\geq}0\, \ldots$



$v \leq 1 \lor v^2 \leq 2b(MA - z)$

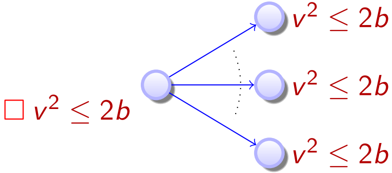
differential dynamic logic

$d\mathcal{L} = \text{FOL}_{\mathbb{R}} +$



differential dynamic logic

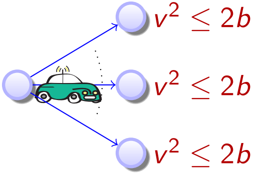
$d\mathcal{L} = \text{FOL}_{\mathbb{R}} + \text{ML}$



differential dynamic logic

$\mathsf{d}\mathcal{L} = \mathrm{FOL}_{\mathbb{R}} + \mathrm{DL}$

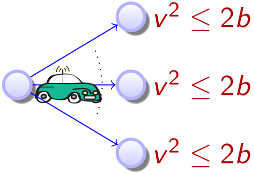
[] $v^2 \leq 2b$



differential dynamic logic

$d\mathcal{L} = \text{FOL}_{\mathbb{R}} + \text{DL} + \text{HP}$

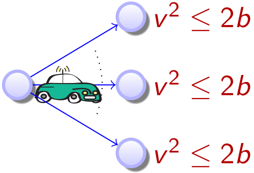
$[z'' = a]\, v^2 \leq 2b$



differential dynamic logic

$d\mathcal{L} = FOL_{\mathbb{R}} + DL + HP$

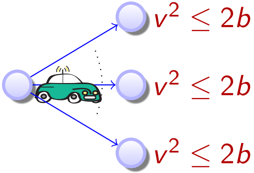
$[\texttt{if}(z > SB)\, a := -b;\ z'' = a]\, v^2 \leq 2b$



differential dynamic logic

$d\mathcal{L} = FOL_{\mathbb{R}} + DL + HP$

$[\underbrace{\mathtt{if}(z > SB)\, a := -b;\ z'' = a}_{\text{hybrid program}}]\, v^2 \leq 2b$

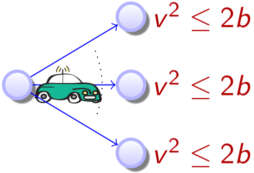


Differential Dynamic Logic for Hybrid Systems

differential dynamic logic

$d\mathcal{L} = \text{FOL}_{\mathbb{R}} + \text{DL} + \text{HP}$

$$\mathcal{C} \rightarrow [\underbrace{\texttt{if}(z > SB)\, a := -b;\ z'' = a}_{\text{hybrid program}}]\, v^2 \leq 2b$$



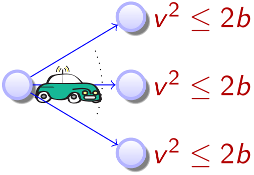
Differential Dynamic Logic for Hybrid Systems

differential dynamic logic

$\mathsf{d}\mathcal{L} = \mathrm{FOL}_{\mathbb{R}} + \mathrm{DL} + \mathrm{HP}$

$$\mathcal{C} \rightarrow [\underbrace{\texttt{if}(z > SB)\, a := -b;\ z'' = a}_{\text{hybrid program}}]\, v^2 \leq 2b$$

Initial condition

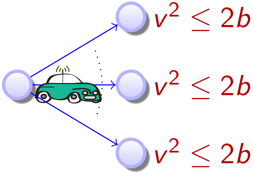


Differential Dynamic Logic for Hybrid Systems

differential dynamic logic

$d\mathcal{L} = \mathrm{FOL}_{\mathbb{R}} + \mathrm{DL} + \mathrm{HP}$

$$\mathcal{C} \rightarrow [\underbrace{\mathtt{if}(z > SB)\, a := -b;\ z'' = a}_{\text{hybrid program}}]\, v^2 \leq 2b$$



Initial condition

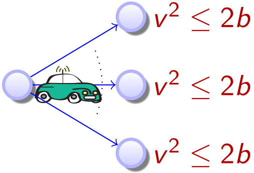
System dynamics

Differential Dynamic Logic for Hybrid Systems

differential dynamic logic

$$d\mathcal{L} = \mathrm{FOL}_{\mathbb{R}} + \mathrm{DL} + \mathrm{HP}$$

$$\mathcal{C} \to [\underbrace{\mathtt{if}(z > SB)\, a := -b;\ z'' = a}_{\text{hybrid program}}]\, v^2 \le 2b$$



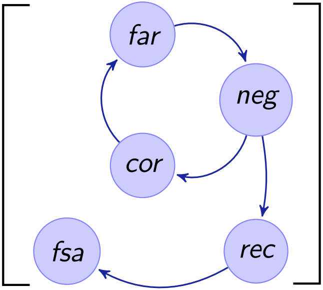
Initial condition

System dynamics

Post condition

differential dynamic logic

$d\mathcal{L} = \text{FOL}_{\mathbb{R}} + \text{DL} + \text{HP}$

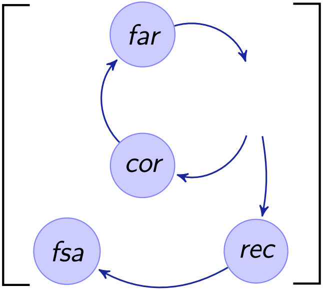


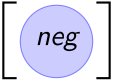
$v^2 \leq 2b..$

Differential Dynamic Logic for Hybrid Systems

differential dynamic logic

$d\mathcal{L} = \text{FOL}_{\mathbb{R}} + \text{DL} + \text{HP}$



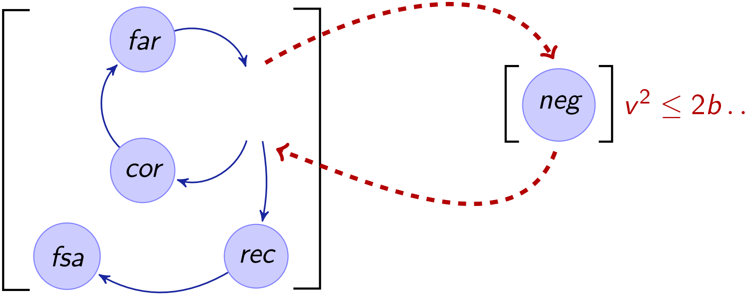


$v^2 \leq 2b\,..$

Differential Dynamic Logic for Hybrid Systems

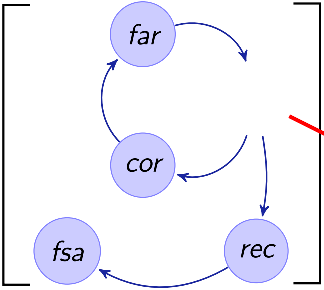
differential dynamic logic

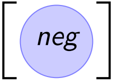
$d\mathcal{L} = \text{FOL}_{\mathbb{R}} + \text{DL} + \text{HP}$



differential dynamic logic

$d\mathcal{L} = FOL_{\mathbb{R}} + DL + HP$





$v^2 \leq 2b\,..$

Not Compositional

Differential Dynamic Logic d$\mathcal{L}$: Syntax

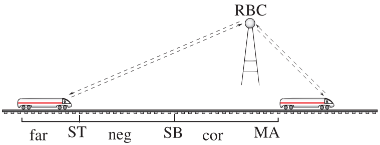
Definition (Hybrid program α)

$x' = f(x)$	(continuous evolution)	
$x := f(x)$	(discrete jump)	jump & test
$?H$	(conditional execution)	jump & test
$\alpha; \beta$	(seq. composition)	Kleene algebra
$\alpha \cup \beta$	(nondet. choice)	Kleene algebra
α^*	(nondet. repetition)	Kleene algebra

Definition (Hybrid program α)

$x' = f(x)$	(continuous evolution)	
$x := f(x)$	(discrete jump)	jump & test
$?H$	(conditional execution)	
$\alpha; \beta$	(seq. composition)	
$\alpha \cup \beta$	(nondet. choice)	Kleene algebra
α^*	(nondet. repetition)	

$$ETCS \equiv (ctrl; drive)^*$$
$$ctrl \equiv (?MA - z \leq SB; a := -b)$$
$$\cup\ (?MA - z \geq SB; a := \ldots)$$
$$drive \equiv \quad z'' = a$$
$$\& v \geq 0 \land \tau \leq \varepsilon$$



Differential Dynamic Logic d$\mathcal{L}$: Syntax

Definition (Hybrid program α)

$x' = f(x)$	(continuous evolution)	
$x := f(x)$	(discrete jump)	jump & test
$?H$	(conditional execution)	
$\alpha; \beta$	(seq. composition)	Kleene algebra
$\alpha \cup \beta$	(nondet. choice)	
α^*	(nondet. repetition)	

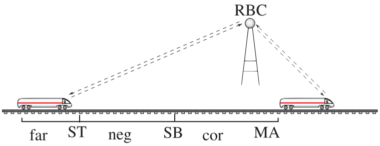
$$ETCS \equiv (ctrl; drive)^*$$

$$ctrl \equiv (?MA - z \leq SB; a := -b)$$

$$\cup (?MA - z \geq SB; a := \dots)$$

$$drive \equiv \tau := 0; z' = v, v' = a, \tau' = 1$$

$$\& \; v \geq 0 \wedge \tau \leq \varepsilon$$

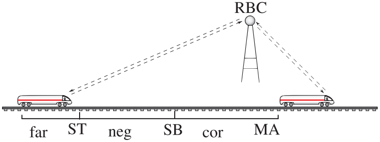


Differential Dynamic Logic d$\mathcal{L}$: Syntax

Definition (Hybrid program α)

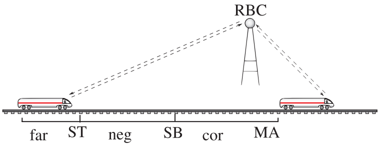
$x' = f(x) \,\&\, H$	(continuous evolution)	
$x := f(x)$	(discrete jump)	jump & test
$?H$	(conditional execution)	
$\alpha; \beta$	(seq. composition)	Kleene algebra
$\alpha \cup \beta$	(nondet. choice)	
α^*	(nondet. repetition)	

$$ETCS \equiv (ctrl; drive)^*$$
$$ctrl \equiv (?MA - z \leq SB; a := -b)$$
$$\cup (?MA - z \geq SB; a := \dots)$$
$$drive \equiv \tau := 0; z' = v, v' = a, \tau' = 1$$
$$\& \, v \geq 0 \wedge \tau \leq \varepsilon$$

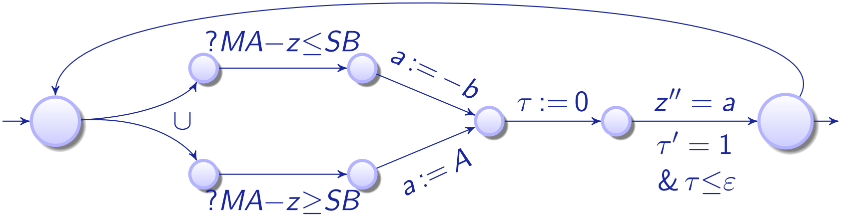


Branching Transitions in Hybrid Programs

$$
\begin{aligned}
ETCS &\equiv (ctrl\,;drive)^* \\
ctrl &\equiv (?MA - z \leq SB; a := -b) \\
&\quad \cup (?MA - z \geq SB; a := A) \\
drive &\equiv \tau := 0; z' = v, v' = a, \tau' = 1 \\
&\quad \& \, v \geq 0 \land \tau \leq \varepsilon
\end{aligned}
$$



Branching Transitions in Hybrid Programs



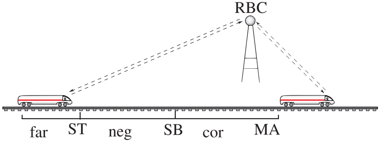
$$ETCS \equiv (ctrl\,;drive)^*$$

$$ctrl \equiv (?MA - z \leq SB; a := -b)$$

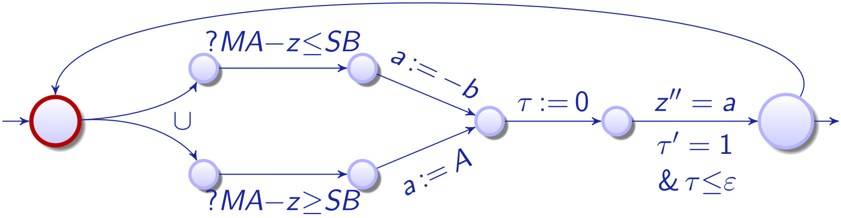
$$\cup\,(?MA - z \geq SB; a := A)$$

$$drive \equiv \tau := 0; z' = v, v' = a, \tau' = 1$$

$$\&\, v \geq 0 \land \tau \leq \varepsilon$$



Branching Transitions in Hybrid Programs



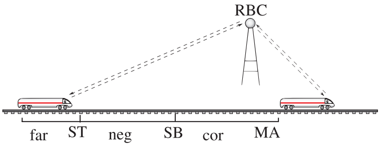
$$ETCS \equiv (ctrl\,;drive)^*$$

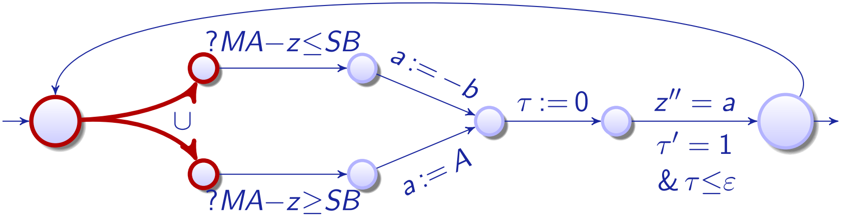
$$ctrl \equiv (?MA - z \le SB; a := -b)$$

$$\cup\,(?MA - z \ge SB; a := A)$$

$$drive \equiv \tau := 0; z' = v, v' = a, \tau' = 1$$

$$\& \, v \ge 0 \land \tau \le \varepsilon$$





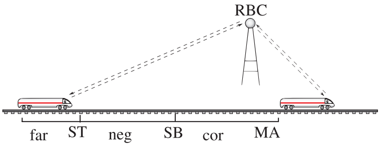
$$ETCS \equiv (ctrl\,;drive)^*$$

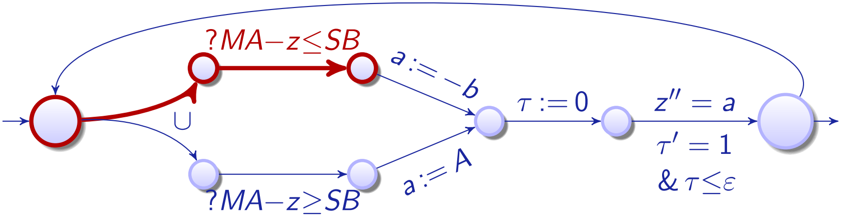
$$ctrl \equiv (?MA - z \leq SB; a := -b)$$

$$\cup\ (?MA - z \geq SB; a := A)$$

$$drive \equiv \tau := 0; z' = v, v' = a, \tau' = 1$$

$$\&\ v \geq 0 \land \tau \leq \varepsilon$$





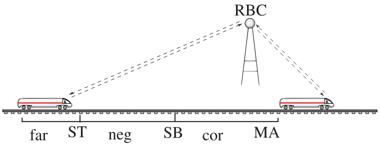
$ETCS \equiv (ctrl\,; drive)^*$

$ctrl \equiv (?MA - z \leq SB; a := -b)$

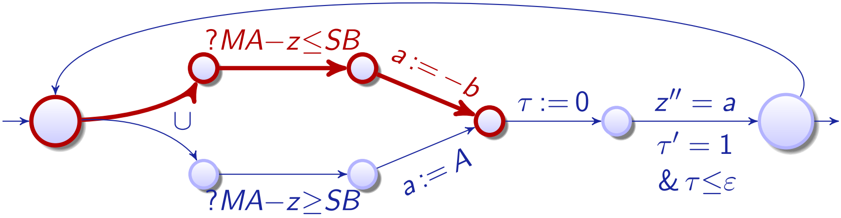
$\cup\, (?MA - z \geq SB; a := A)$

$drive \equiv \tau := 0; z' = v, v' = a, \tau' = 1$

$\&\, v \geq 0 \land \tau \leq \varepsilon$



Branching Transitions in Hybrid Programs



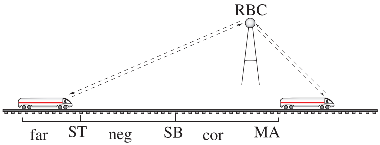
$$ETCS \equiv (\textcolor{red}{ctrl}\,; drive)^*$$

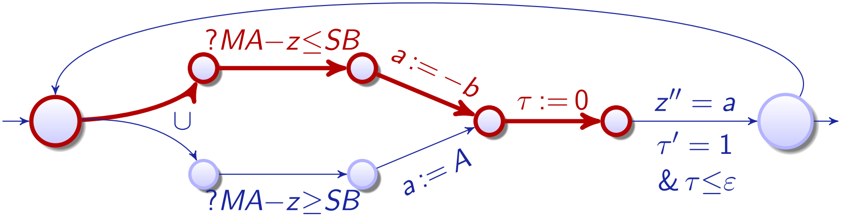
$$ctrl \equiv (?MA - z \leq SB;\ \textcolor{red}{a := -b})$$

$$\cup\,(?MA - z \geq SB;\ a := A)$$

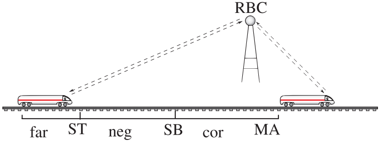
$$drive \equiv \tau := 0;\ z' = v, v' = a, \tau' = 1$$

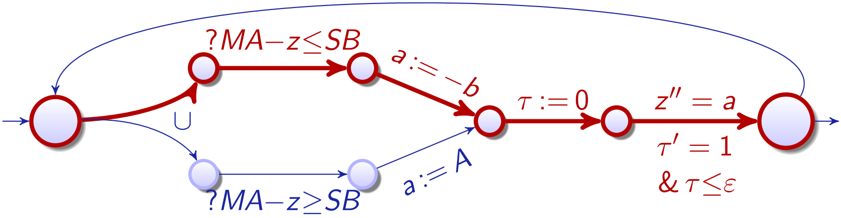
$$\&\ v \geq 0 \land \tau \leq \varepsilon$$





$$ETCS \equiv (ctrl\,;\, drive)^*$$
$$ctrl \equiv (?MA - z \leq SB;\, a := -b)$$
$$\cup\, (?MA - z \geq SB;\, a := A)$$
$$drive \equiv \tau := 0;\, z' = v,\, v' = a,\, \tau' = 1$$
$$\&\, v \geq 0 \land \tau \leq \varepsilon$$





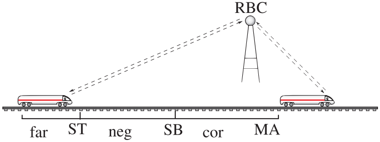
$$ETCS \equiv (ctrl\,;\, drive)^*$$

$$ctrl \equiv (?MA - z \leq SB;\, a := -b)$$

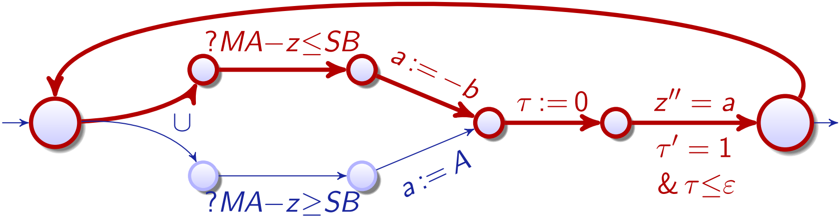
$$\cup\, (?MA - z \geq SB;\, a := A)$$

$$drive \equiv \tau := 0;\, z' = v,\, v' = a,\, \tau' = 1$$

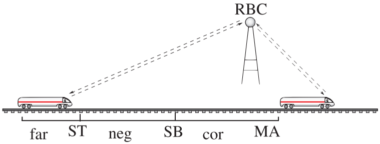
$$\&\, v \geq 0 \land \tau \leq \varepsilon$$



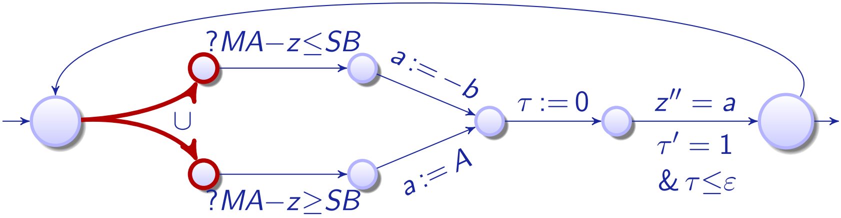
Branching Transitions in Hybrid Programs



$$ETCS \equiv (ctrl\,;drive)^*$$
$$ctrl \equiv (?MA - z \leq SB; a := -b)$$
$$\cup (?MA - z \geq SB; a := A)$$
$$drive \equiv \tau := 0; z' = v, v' = a, \tau' = 1$$
$$\& \, v \geq 0 \wedge \tau \leq \varepsilon$$



Branching Transitions in Hybrid Programs



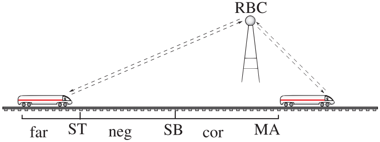
$$ETCS \equiv (ctrl\,; drive)^*$$

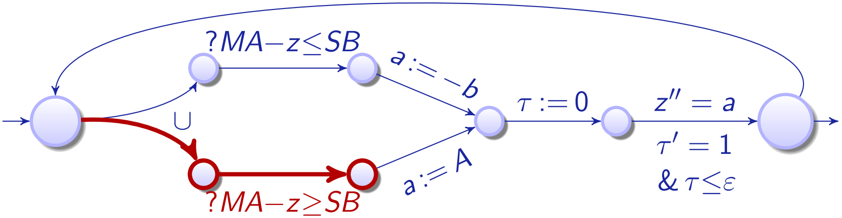
$$ctrl \equiv (?MA - z \leq SB; a := -b)$$

$$\cup\ (?MA - z \geq SB; a := A)$$

$$drive \equiv \tau := 0; z' = v, v' = a, \tau' = 1$$

$$\&\ v \geq 0 \land \tau \leq \varepsilon$$





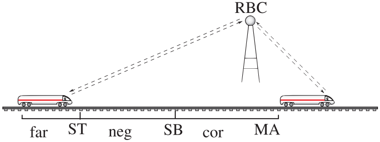
$$ETCS \equiv (ctrl\,;drive)^*$$

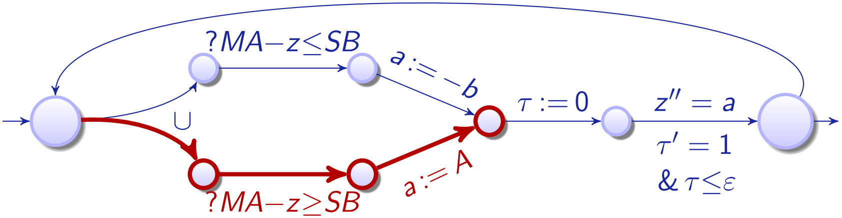
$$ctrl \equiv (?MA - z \le SB;\ a := -b)$$

$$\cup\ (?MA - z \ge SB;\ a := A)$$

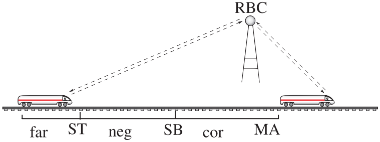
$$drive \equiv \tau := 0;\ z' = v, v' = a, \tau' = 1$$

$$\&\ v \ge 0 \land \tau \le \varepsilon$$

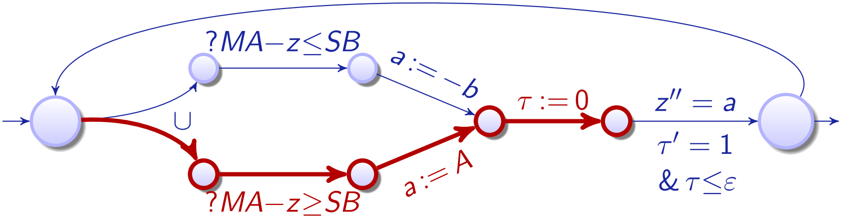




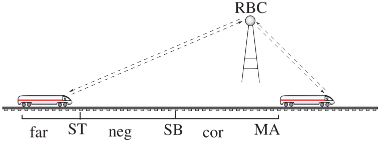
$$
\begin{aligned}
ETCS &\equiv (ctrl\,; drive)^* \\
ctrl &\equiv (?MA - z \leq SB; a := -b) \\
&\quad \cup (?MA - z \geq SB; a := A) \\
drive &\equiv \tau := 0; z' = v, v' = a, \tau' = 1 \\
&\quad \& \, v \geq 0 \land \tau \leq \varepsilon
\end{aligned}
$$

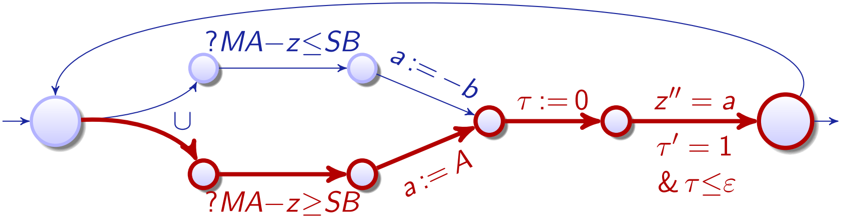


Branching Transitions in Hybrid Programs

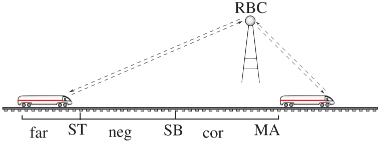


$$ETCS \equiv (ctrl\,;\,drive)^*$$
$$ctrl \equiv (?MA - z \le SB;\, a := -b)$$
$$\cup\, (?MA - z \ge SB;\, a := A)$$
$$drive \equiv \tau := 0;\, z' = v,\, v' = a,\, \tau' = 1$$
$$\&\, v \ge 0 \land \tau \le \varepsilon$$

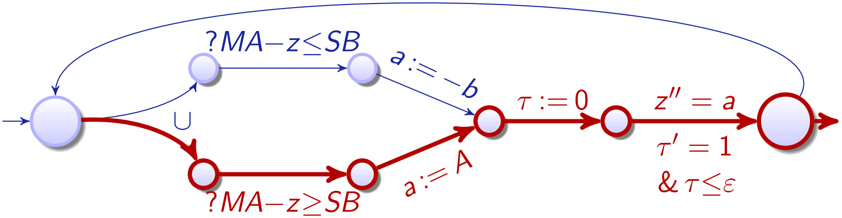




$$ETCS \equiv (ctrl\,;\, drive)^*$$
$$ctrl \equiv (?MA - z \leq SB;\, a := -b)$$
$$\cup\,(?MA - z \geq SB;\, a := A)$$
$$drive \equiv \tau := 0;\, z' = v, v' = a, \tau' = 1$$
$$\&\ v \geq 0 \wedge \tau \leq \varepsilon$$

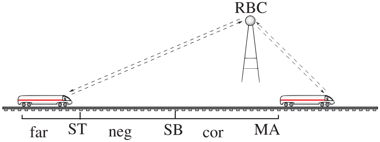


Branching Transitions in Hybrid Programs

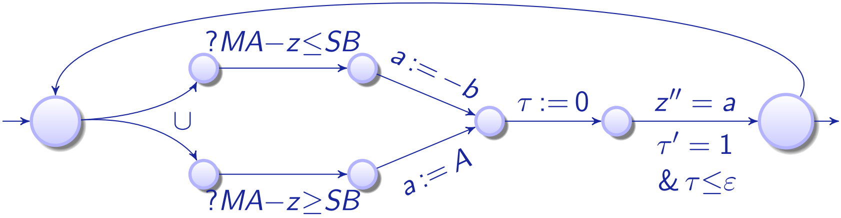


$$ETCS \equiv (ctrl\,;drive)^*$$

$$ctrl \equiv (?MA - z \leq SB; a := -b)$$
$$\cup\, (?MA - z \geq SB; a := A)$$
$$drive \equiv \tau := 0; z' = v, v' = a, \tau' = 1$$
$$\&\, v \geq 0 \wedge \tau \leq \varepsilon$$



Branching Transitions in Hybrid Programs



$\texttt{if}(H)\,\alpha\,\texttt{else}\,\beta \equiv$

$\texttt{while}(H)\,\alpha \equiv$

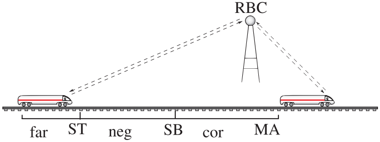
$$ETCS \equiv (ctrl\,;drive)^*$$

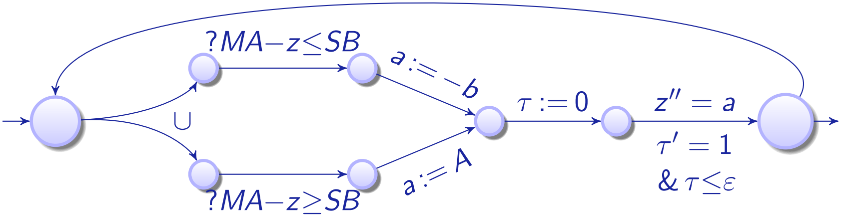
$$ctrl \equiv (?MA - z \leq SB; a := -b)$$

$$\cup\,(?MA - z \geq SB; a := A)$$

$$drive \equiv \tau := 0; z' = v, v' = a, \tau' = 1$$

$$\&\, v \geq 0 \land \tau \leq \varepsilon$$

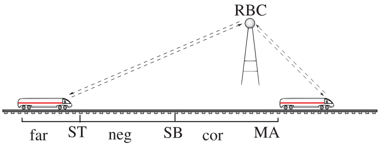


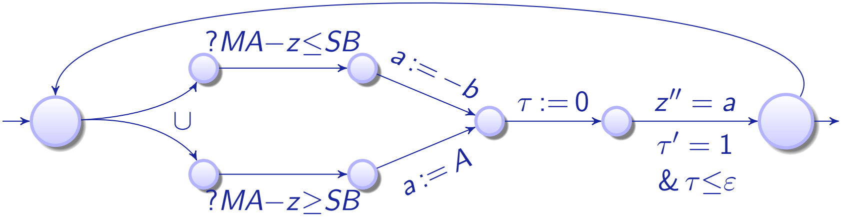


$\texttt{if}(H)\,\alpha\,\texttt{else}\,\beta \equiv (?H;\alpha) \cup (?\neg H;\beta)$

$\texttt{while}(H)\,\alpha \equiv$

$$ETCS \equiv (ctrl\,; drive)^*$$
$$ctrl \equiv (?MA - z \leq SB; a := -b)$$
$$\cup\,(?MA - z \geq SB; a := A)$$
$$drive \equiv \tau := 0; z' = v, v' = a, \tau' = 1$$
$$\&\, v \geq 0 \land \tau \leq \varepsilon$$





$$\texttt{if}(H)\,\alpha\,\texttt{else}\,\beta \equiv (?H;\alpha) \cup (?\neg H;\beta)$$

$$\texttt{while}(H)\,\alpha \equiv (?H;\alpha)^*;?\neg H$$

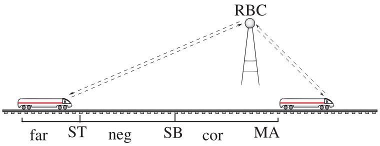
$$ETCS \equiv (ctrl\,;drive)^*$$

$$ctrl \equiv (?MA - z \leq SB; a := -b)$$

$$\cup\,(?MA - z \geq SB; a := A)$$

$$drive \equiv \tau := 0; z' = v, v' = a, \tau' = 1$$

$$\&\, v \geq 0 \land \tau \leq \varepsilon$$



Differential Dynamic Logic d$\mathcal{L}$: Syntax

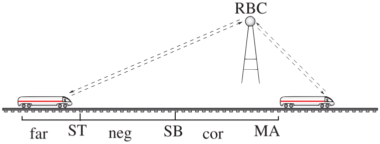
Definition (d$\mathcal{L}$ Formula ϕ)

$$\theta_1 \geq \theta_2 \mid \neg\phi \mid \phi \wedge \psi \mid \phi \vee \psi \mid \phi \rightarrow \psi \mid \forall x\, \phi \mid \exists x\, \phi \mid [\alpha]\phi \mid \langle\alpha\rangle\phi$$

with terms θ_1, θ_2 of nonlinear real arithmetic $(+, \cdot)$

$$SB \geq \ldots \;\rightarrow\; [(ctrl\,; drive)^*]\, z \leq MA$$

All trains respect MA
RBC partitions MA
$\Rightarrow$ system collision free



Differential Dynamic Logic d$\mathcal{L}$: Semantics

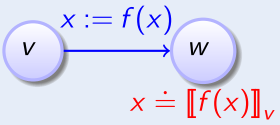
Definition (Hybrid program α)

$$
\begin{aligned}
\rho(x := \theta) &= \{(v, w) \;:\; w = v \text{ except } [\![x]\!]_w = [\![\theta]\!]_v\} \\
\rho(?H) &= \{(v, v) \;:\; v \models H\} \\
\rho(x' = f(x)) &= \{(\varphi(0), \varphi(r)) \;:\; \varphi \models x' = f(x) \text{ for some duration } r\} \\
\rho(\alpha \cup \beta) &= \rho(\alpha) \cup \rho(\beta) \\
\rho(\alpha; \beta) &= \rho(\beta) \circ \rho(\alpha) \\
\rho(\alpha^*) &= \bigcup_{n \in \mathbb{N}} \rho(\alpha^n)
\end{aligned}
$$

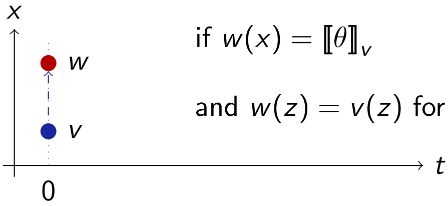
Definition (d$\mathcal{L}$ Formula ϕ)

$$
\begin{aligned}
v \models \theta_1 \geq \theta_2 &\quad \text{iff} \quad [\![\theta_1]\!]_v \geq [\![\theta_2]\!]_v \\
v \models [\alpha]\phi &\quad \text{iff} \quad w \models \phi \text{ for all } w \text{ with } (v, w) \in \rho(\alpha) \\
v \models \langle\alpha\rangle\phi &\quad \text{iff} \quad w \models \phi \text{ for some } w \text{ with } (v, w) \in \rho(\alpha) \\
v \models \forall x\, \phi &\quad \text{iff} \quad w \models \phi \text{ for all } w \text{ that agree with } v \text{ except for } x \\
v \models \exists x\, \phi &\quad \text{iff} \quad w \models \phi \text{ for some } w \text{ that agrees with } v \text{ except for } x \\
v \models \phi \wedge \psi &\quad \text{iff} \quad v \models \phi \text{ and } v \models \psi
\end{aligned}
$$

Definition (Hybrid programs α: transition semantics)



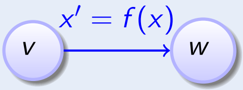
Example



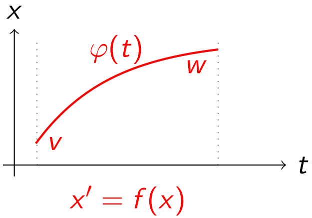
if $w(x) = [\![\theta]\!]_v$

and $w(z) = v(z)$ for $z \neq x$

Definition (Hybrid programs α: transition semantics)

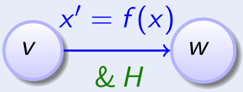


Example

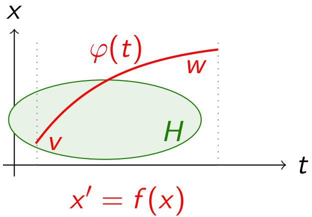


Differential Dynamic Logic d$\mathcal{L}$: Transition Semantics

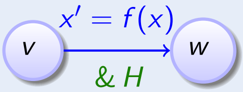
Definition (Hybrid programs α: transition semantics)



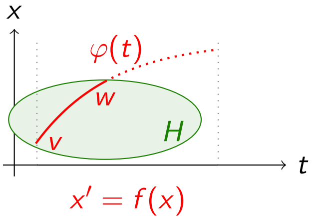
Example



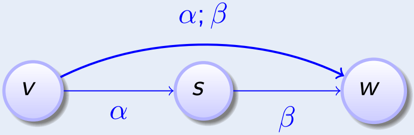
Definition (Hybrid programs α: transition semantics)



Example



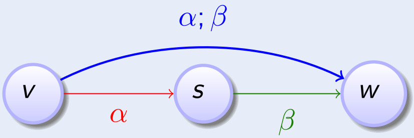
Definition (Hybrid programs α: transition semantics)



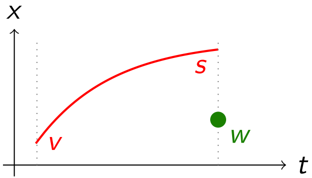
Example

Differential Dynamic Logic d$\mathcal{L}$: Transition Semantics

Definition (Hybrid programs α: transition semantics)

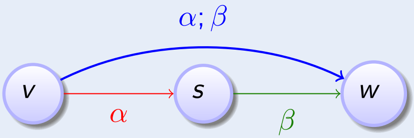


Example

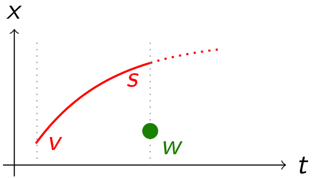


Differential Dynamic Logic d$\mathcal{L}$: Transition Semantics

Definition (Hybrid programs α: transition semantics)

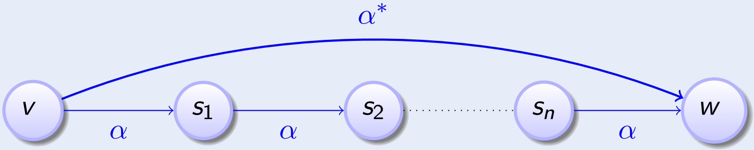


Example



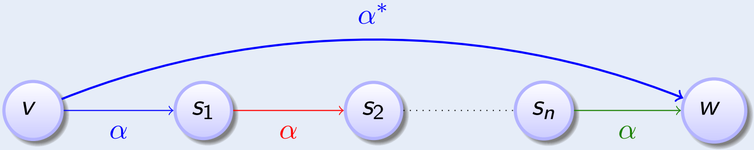
Differential Dynamic Logic d$\mathcal{L}$: Transition Semantics

Definition (Hybrid programs α: transition semantics)

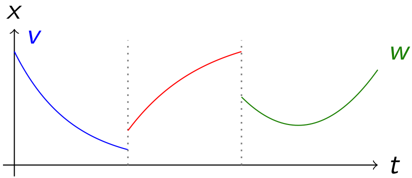


Example

Definition (Hybrid programs α: transition semantics)

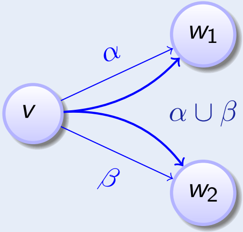


Example



Differential Dynamic Logic $d\mathcal{L}$: Transition Semantics

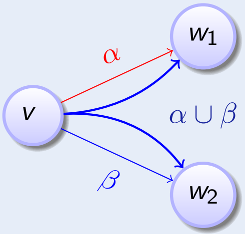
Definition (Hybrid programs α: transition semantics)



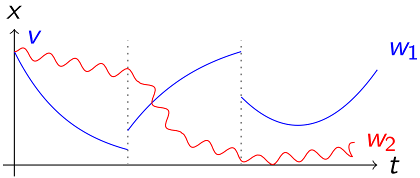
Example

Differential Dynamic Logic d$\mathcal{L}$: Transition Semantics

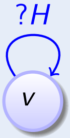
Definition (Hybrid programs α: transition semantics)



Example

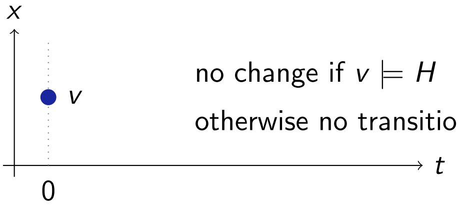


Definition (Hybrid programs α: transition semantics)



if $v \models H$

Example



no change if $v \models H$

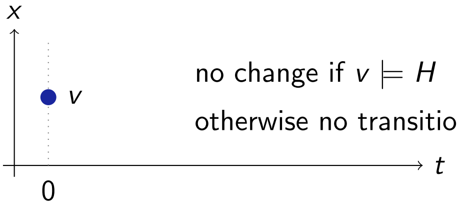
otherwise no transition

Differential Dynamic Logic d$\mathcal{L}$: Transition Semantics

Definition (Hybrid programs α: transition semantics)

if $v \not\models H$

Example

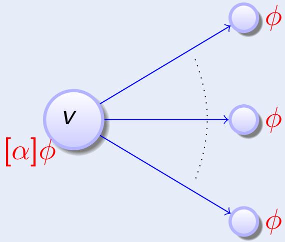


no change if $v \models H$

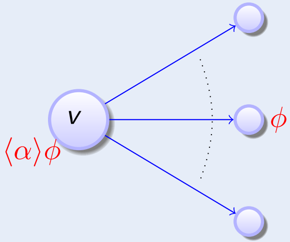
otherwise no transition

Differential Dynamic Logic d$\mathcal{L}$: Semantics

Definition (Formulas ϕ)

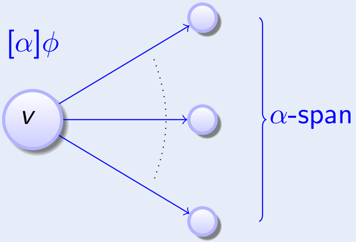


Definition (Formulas ϕ)



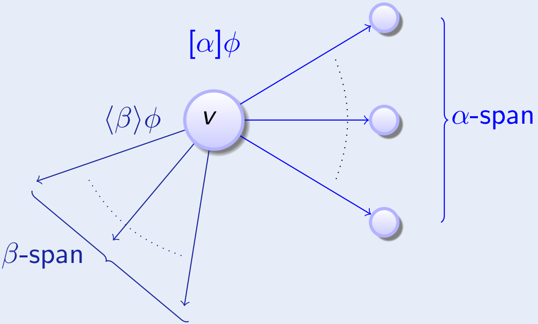
Differential Dynamic Logic d$\mathcal{L}$: Semantics

Definition (Formulas ϕ)



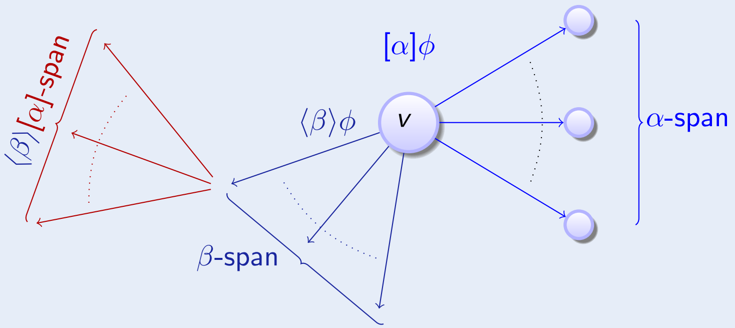
Differential Dynamic Logic d$\mathcal{L}$: Semantics

Definition (Formulas ϕ)



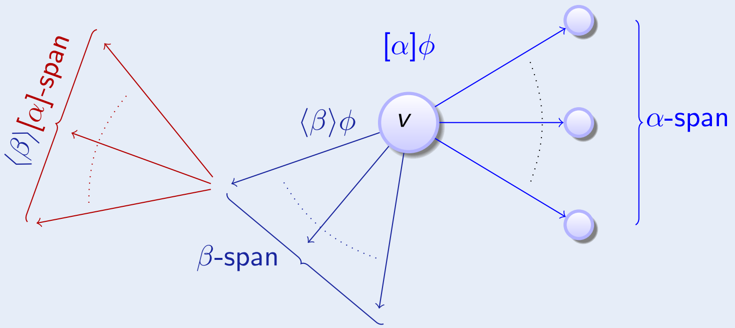
Differential Dynamic Logic d$\mathcal{L}$: Semantics

Definition (Formulas ϕ)



Differential Dynamic Logic d$\mathcal{L}$: Semantics

Definition (Formulas ϕ)

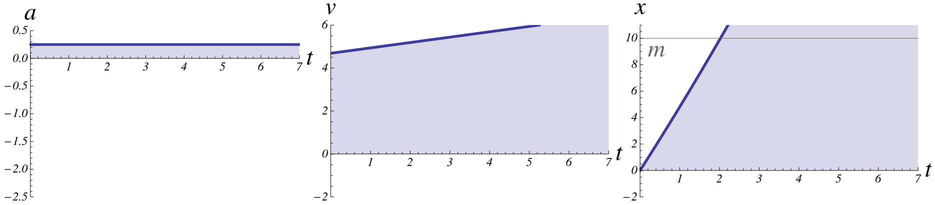


compositional semantics $\Rightarrow$ compositional proofs!

Ex: Car Control

Example (Single car car_s)

$$x' = v, v' = a$$

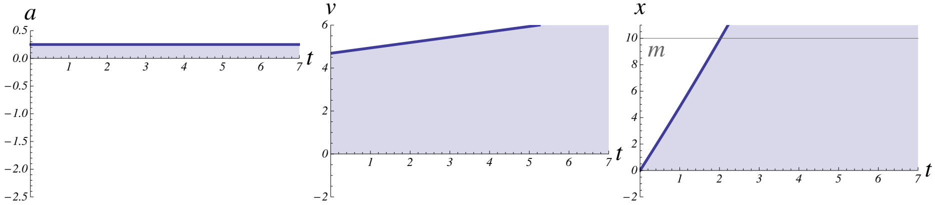


Ex: Car Control

Control decision: accelerate or brake

Example (Single car car_s)

$$(a := A \cup a := -b);\ x' = v, v' = a$$

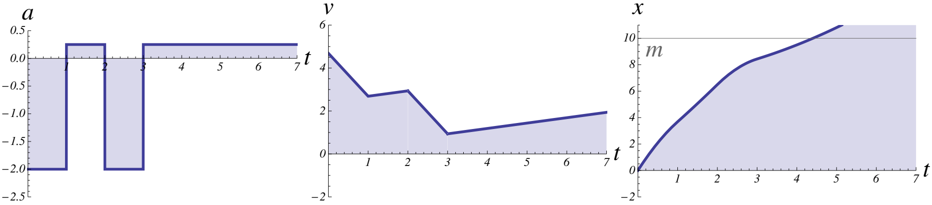


Ex: Car Control

Repeat control decisions

Example (Single car car_s)

$$\left(\left(a := A \cup a := -b\right);\ x' = v, v' = a\right)^*$$

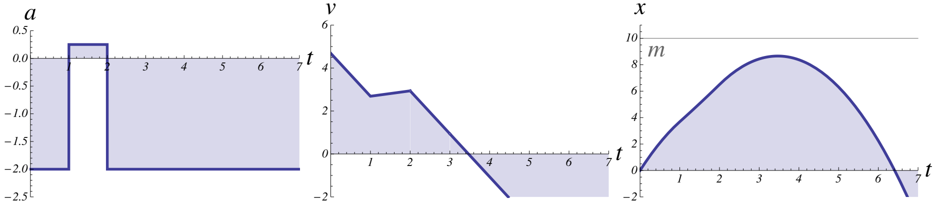


Ex: Car Control

Repeat control decisions

Example (Single car car_s)

$$\left(\left(a := A \;\cup a := -b\right);\; x' = v, v' = a\right)^*$$

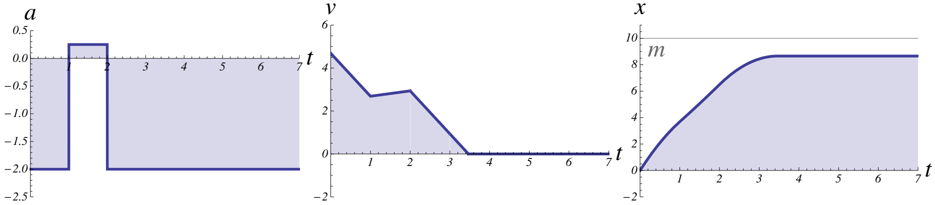


Ex: Car Control

Velocity bound $v \geq 0$

Example (Single car car_s)

$$\big((\ a := A\ \cup a := -b);\ x' = v, v' = a \,\&\, v \geq 0\big)^*$$

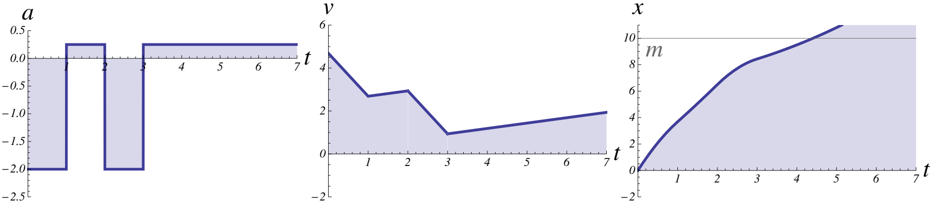


Ex: Car Control

Accelerate not always safe

Example (Single car car_s)

$$\left(\left(\, a := A \;\cup a := -b\right);\; x' = v, v' = a \,\&\, v \geq 0\right)^*$$

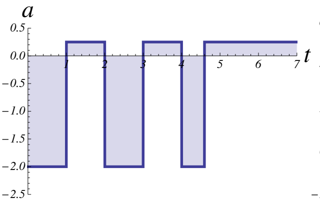


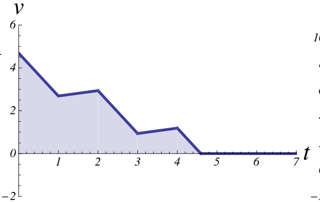
Ex: Car Control

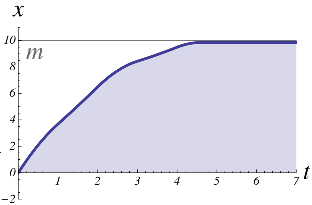
Accelerate condition $?H$

Example (Single car car_s)

$$(((?H;\, a := A) \cup a := -b);\;\; x' = v, v' = a \,\&\, v \geq 0)^*$$





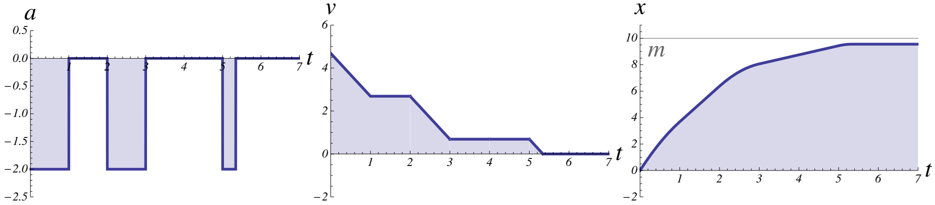


Ex: Car Control

Accelerate condition $?H$ depends on A

Example (Single car car_s)

$$\left(((?H;\, a := 0) \cup a := -b);\;\; x' = v, v' = a \,\&\, v \geq 0\right)^*$$



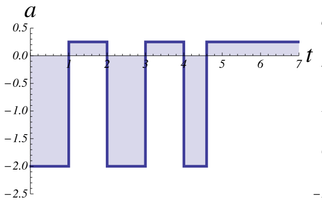
Example (Single car car_e)

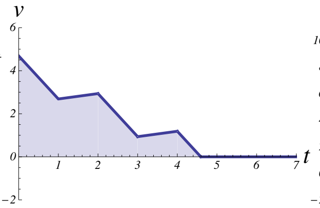
$$(((?H; a := A) \cup a := -b);\ x' = v, v' = a \,\&\, v \geq 0)^*$$

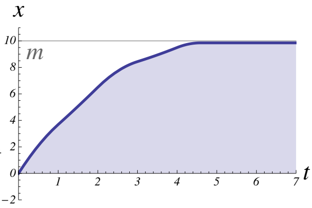
Accelerate condition tests proximity to m

Example (Single car car_e)

$$(((?m-x \geq 2; a := A) \cup a := -b);\ x' = v, v' = a \,\&\, v \geq 0)^*$$



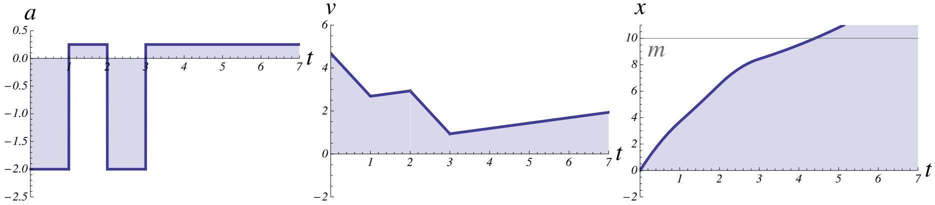




Miss event $m - x \leq 2 \Rightarrow$ crash

Example (Single car car_e)

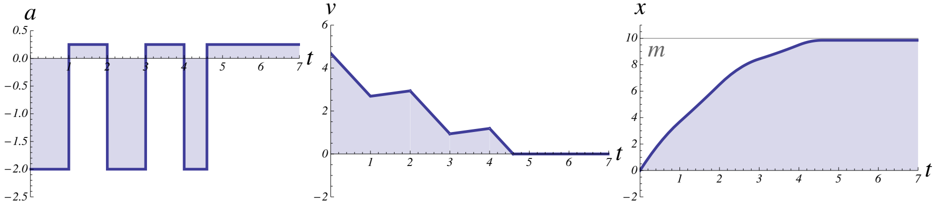
$$(((?m-x \geq 2; a := A) \cup a := -b);\ x' = v, v' = a \,\&\, v \geq 0)^*$$



Guard for event $1 \le m - x \le 2$

Example (Single car car_e event-triggered)

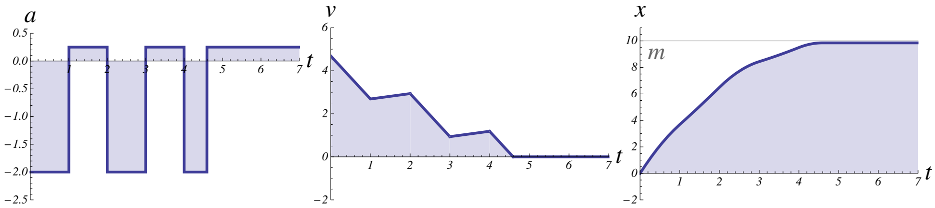
$$\Big(\big((?m{-}x \ge 2;\, a := A) \cup a := -b\big);\ x' = v, v' = a \,\&\, v \ge 0 \land m{-}x \ge 1\Big)^*$$



Guard for event $1 \le m - x \le 2$ hard to implement

Example (Single car car_e event-triggered)

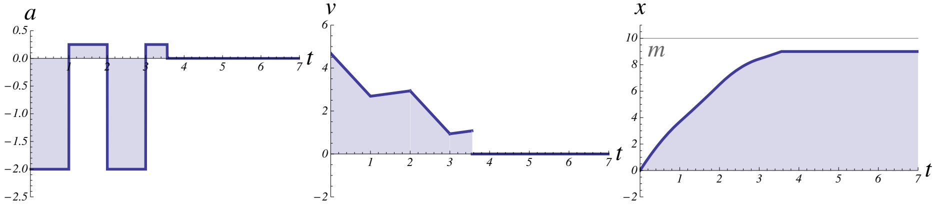
$$\left(((?m{-}x \ge 2;\, a := A) \cup a := -b);\; x' = v, v' = a \,\&\, v \ge 0 \land m{-}x \ge 1\right)^*$$



Careful with evolution domains!

Example (Single car car_e event-triggered)

$$\Big(\big((?m-x \geq 2; a := A) \cup a := -b\big);\ x' = v, v' = a \,\&\, v \geq 0 \land m-x \geq 1\Big)^*$$



Ex: Car Control & Sensors — time-triggered

Example (Single car car_ε)

$$(((?H; a := A) \cup a := -b);\ x' = v, v' = a \,\&\, v \geq 0)^*$$

Example (Single car car_ε time-triggered)

$$\big(((?H; a := A) \cup a := -b);\; x' = v, v' = a \,\&\, v \geq 0 \wedge t \leq \varepsilon\big)^*$$

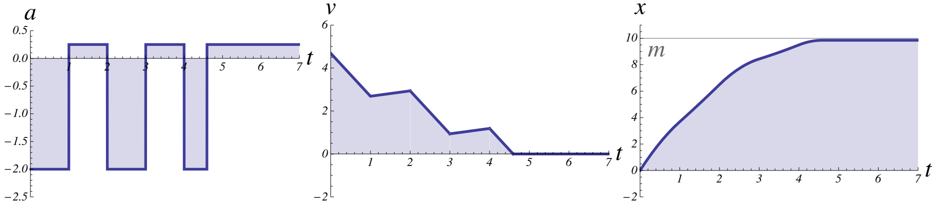
Example (Single car car_ε time-triggered)

$$\big(((?H; a := A) \cup a := -b);\ x' = v, v' = a, t' = 1 \,\&\, v \geq 0 \land t \leq \varepsilon\big)^*$$

Trigger control every $\leq \varepsilon$ time units

Example (Single car car_ε time-triggered)

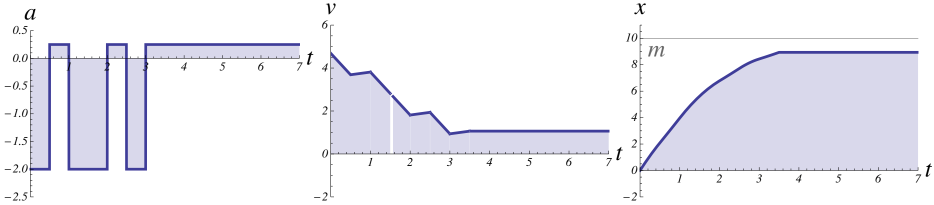
$$(((?H; a := A) \cup a := -b);\; t := 0;\; x' = v, v' = a, t' = 1 \,\&\, v \geq 0 \wedge t \leq \varepsilon)^*$$



Really faster ⇒ inefficient

Example (Single car car_ε time-triggered)

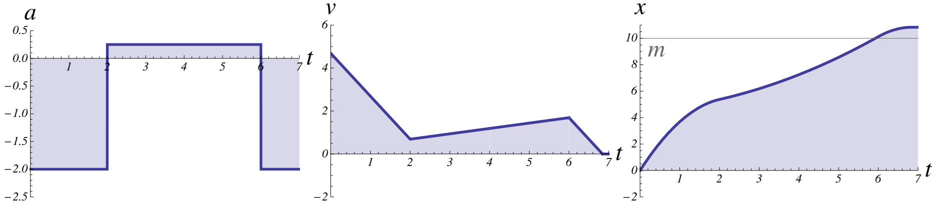
$$(((?H; a := A) \cup a := -b);\ t := 0;\ x' = v, v' = a, t' = 1 \,\&\, v \geq 0 \land t \leq \varepsilon)^*$$



Really slower $\Rightarrow$ crash

Example (Single car car_ε time-triggered)

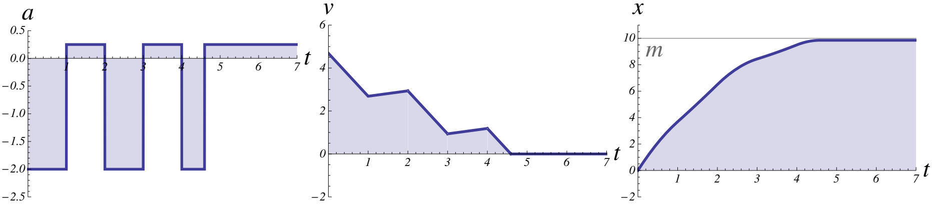
$$(((?H; a := A) \cup a := -b);\ t := 0;\ x' = v, v' = a, t' = 1 \,\&\, v \geq 0 \wedge t \leq \varepsilon)^*$$



Accelerate condition $?H$ depends on ...

Example (Single car car_ε time-triggered)

$$\left(((?H; a := A) \cup a := -b);\ t := 0;\ x' = v, v' = a, t' = 1 \,\&\, v \geq 0 \land t \leq \varepsilon\right)^*$$

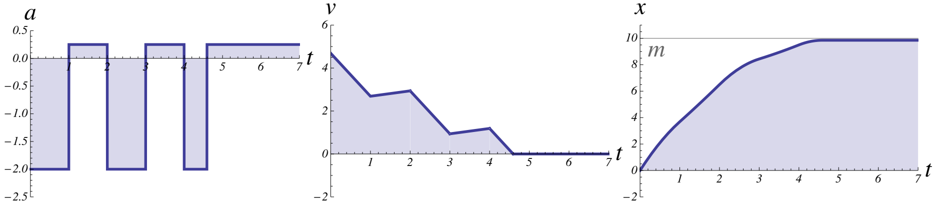


Accelerate condition $?H$ depends on …

$$H \equiv 2b(m-x) \geq v^2 + (A+b)(A\varepsilon^2 + 2\varepsilon v)$$

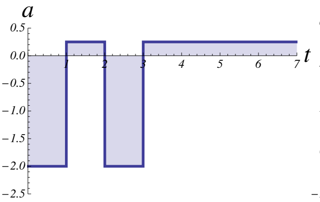
Example (Single car car_ε time-triggered)

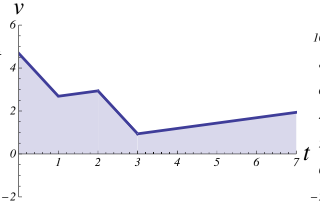
$$(((?H; a := A) \cup a := -b);\ t := 0;\ x' = v, v' = a, t' = 1 \,\&\, v \geq 0 \land t \leq \varepsilon)^*$$

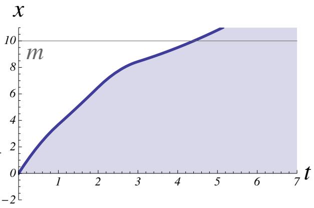


Example (Single car car_s)

$$\left(((?m - x \geq 2;\, a := A) \cup a := -b);\;\; x' = v, v' = a \,\&\, v \geq 0\right)^*$$





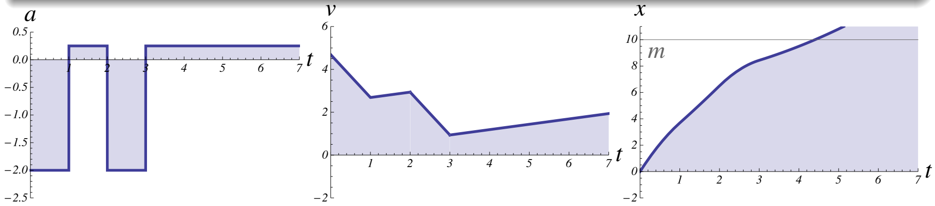


Ex: Car Control Properties

Example (Single car car_s)

$$\left(\left((?m - x \geq 2; a := A) \cup a := -b\right);\ x' = v, v' = a \,\&\, v \geq 0\right)^*$$

Example (Drives forward)



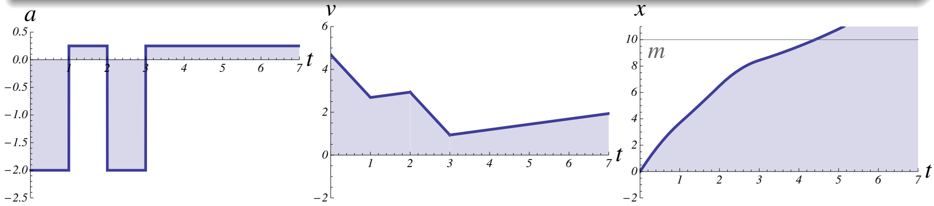
Ex: Car Control Properties

Example (Single car car_s)

$$(((?m-x \geq 2; a := A) \cup a := -b);\ x' = v, v' = a \,\&\, v \geq 0)^*$$

Example (Drives forward)

$$v \geq 0 \land A \geq 0 \land b > 0 \rightarrow [car_s]v \geq 0$$



Ex: Car Control Properties

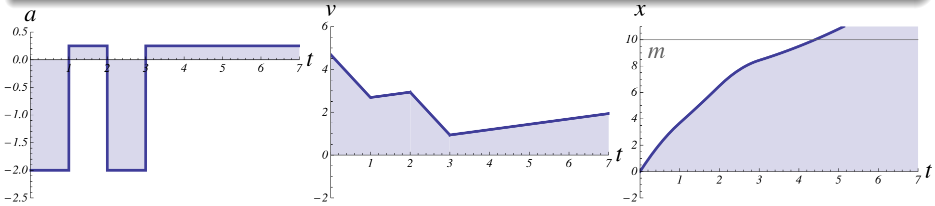
True initially, preserved by definition

Example (Single car car_s)

$$(((?m - x \geq 2; a := A) \cup a := -b);\ x' = v, v' = a \,\&\, v \geq 0)^*$$

Example (Drives forward)

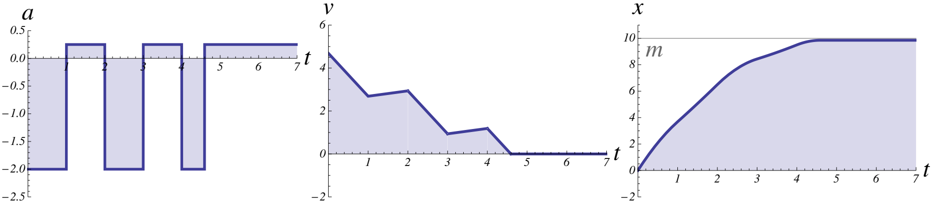
$$v \geq 0 \land A \geq 0 \land b > 0 \rightarrow [car_s] v \geq 0$$



$H \equiv 2b(m-x) \geq v^2 + (A+b)(A\varepsilon^2 + 2\varepsilon v)$

Example (Single car car_ε event-triggered)

$$(((?H; a := A) \cup a := -b);\ t := 0;\ x' = v, v' = a, t' = 1 \,\&\, v \geq 0 \wedge t \leq \varepsilon)^*$$

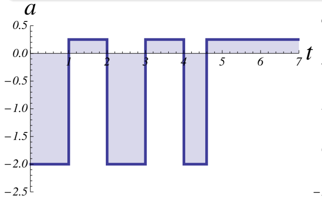


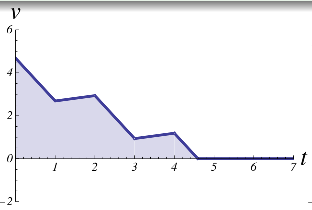
$H \equiv 2b(m-x) \geq v^2 + (A+b)(A\varepsilon^2 + 2\varepsilon v)$

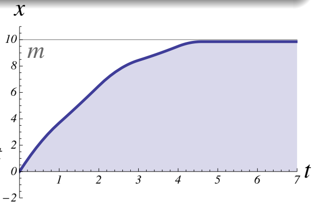
Example (Single car car_ε event-triggered)

$$(((?H; a := A) \cup a := -b);\ t := 0;\ x' = v, v' = a, t' = 1 \,\&\, v \geq 0 \wedge t \leq \varepsilon)^*$$

Example (Stays before traffic light m)







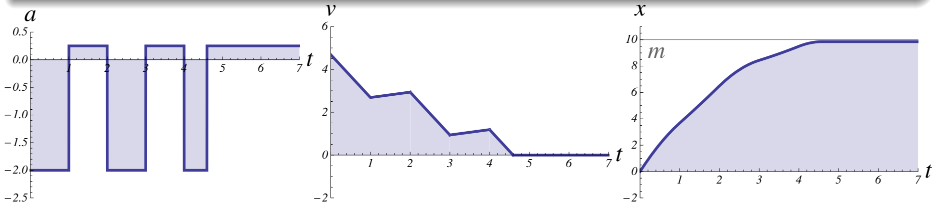
$H \equiv 2b(m - x) \geq v^2 + (A + b)(A\varepsilon^2 + 2\varepsilon v)$

Example (Single car car_ε event-triggered)

$$(((?H; a := A) \cup a := -b);\ t := 0;\ x' = v, v' = a, t' = 1 \,\&\, v \geq 0 \wedge t \leq \varepsilon)^*$$

Example (Stays before traffic light m)

$$v^2 \leq 2b(m - x) \wedge A \geq 0 \wedge b > 0 \rightarrow [car_\varepsilon]x \leq m$$

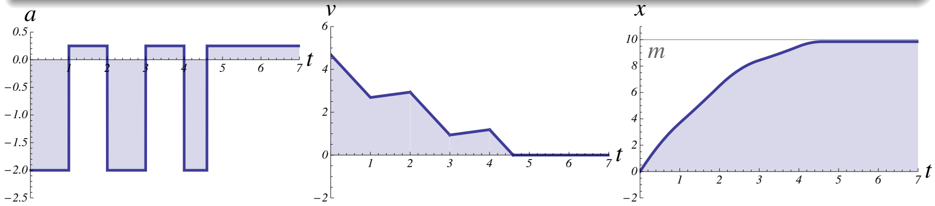


$H \equiv 2b(m-x) \geq v^2 + (A+b)(A\varepsilon^2 + 2\varepsilon v)$

Example (Single car car_ε event-triggered)

$$(((?H; a := A) \cup a := -b);\ t := 0;\ x' = v, v' = a, t' = 1\ \&\ v \geq 0 \wedge t \leq \varepsilon)^*$$

Example (Live, can move everywhere)



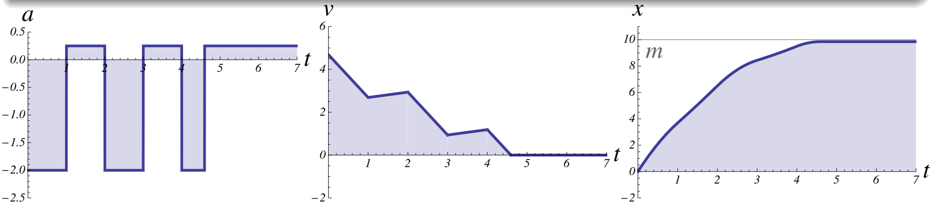
$$H \equiv 2b(m-x) \geq v^2 + (A+b)(A\varepsilon^2 + 2\varepsilon v)$$

Example (Single car car_ε event-triggered)

$$(((?H; a := A) \cup a := -b);\ t := 0;\ x' = v, v' = a, t' = 1 \,\&\, v \geq 0 \land t \leq \varepsilon)^*$$

Example (▶ Live, can move everywhere)

$$\varepsilon > 0 \land A > 0 \land b > 0 \rightarrow \forall p\, \exists m\, \langle car_\varepsilon \rangle\, x \geq p$$



Ex: Car Control Properties

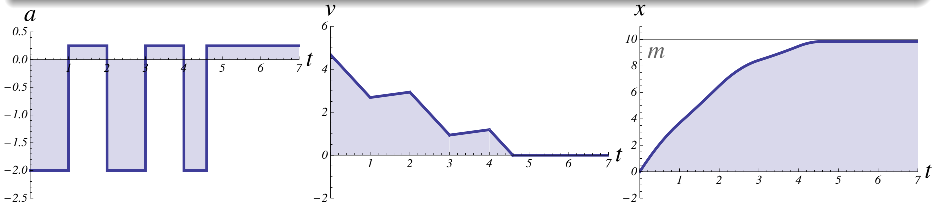
$$H \equiv 2b(m-x) \geq v^2 + (A+b)\left(A\varepsilon^2 + 2\varepsilon v\right)$$

Example (Single car car_ε event-triggered)

$$\left(((?H; a := A) \cup a := -b);\ t := 0;\ x' = v, v' = a, t' = 1 \,\&\, v \geq 0 \land t \leq \varepsilon\right)^*$$

Example (Stays before traffic light m)

$$v^2 \leq 2b(m-x) \land A \geq 0 \land b > 0 \rightarrow [car_\varepsilon]x \leq m$$



Ex: Car Control Properties

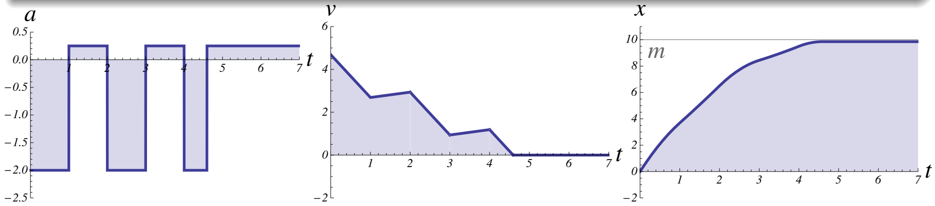
$$H \equiv 2b(m-x) \geq v^2 + (A+b)\left(A\varepsilon^2 + 2\varepsilon v\right)$$

Example (Single car car_ε event-triggered)

$$(((?H; a := A) \cup a := -b);\ t := 0;\ x' = v, v' = a, t' = 1\ \&\ v \geq 0 \land t \leq \varepsilon)^*$$

Example (Stays before traffic light m if braking would)

$$[x' = v, v' = -b]x \leq m \land v \geq 0 \land A \geq 0 \land b > 0 \rightarrow [car_\varepsilon]x \leq m$$



Example (Controllability equivalence)

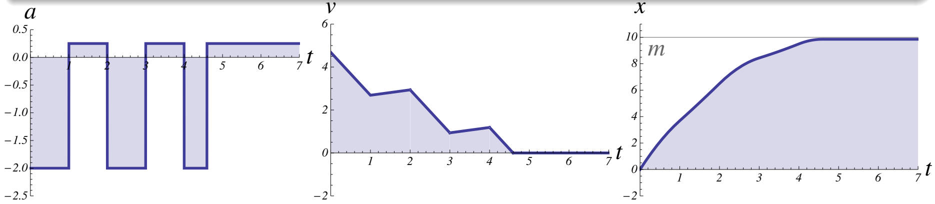
$$v \geq 0 \land b > 0 \rightarrow \left(v^2 \leq 2b(m - x) \leftrightarrow [x' = v, v' = -b]x \leq m\right)$$

Example (Single car car_ε event-triggered)

$$\left(((?H; a := A) \cup a := -b);\ t := 0;\ x' = v, v' = a, t' = 1 \,\&\, v \geq 0 \land t \leq \varepsilon\right)^*$$

Example (Stays before traffic light m if braking would)

$$[x' = v, v' = -b]x \leq m \land v \geq 0 \land A \geq 0 \land b > 0 \rightarrow [car_\varepsilon]x \leq m$$



Example ()

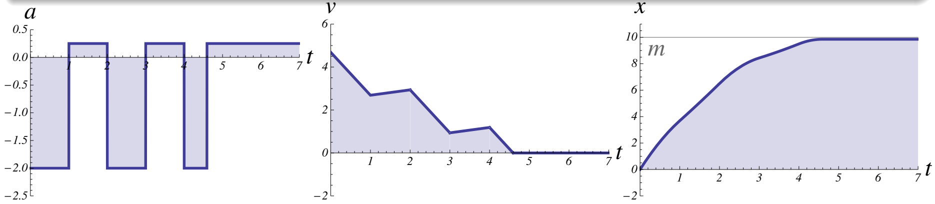
$H \equiv$

Example (Single car car_ε event-triggered)

$$(((?H; a := A) \cup a := -b);\ t := 0;\ x' = v, v' = a, t' = 1 \,\&\, v \geq 0 \land t \leq \varepsilon)^*$$

Example (Stays before traffic light m if braking would)

$$[x' = v, v' = -b]x \leq m \land v \geq 0 \land A \geq 0 \land b > 0 \rightarrow [car_\varepsilon]x \leq m$$



Example (Model-predictive control)

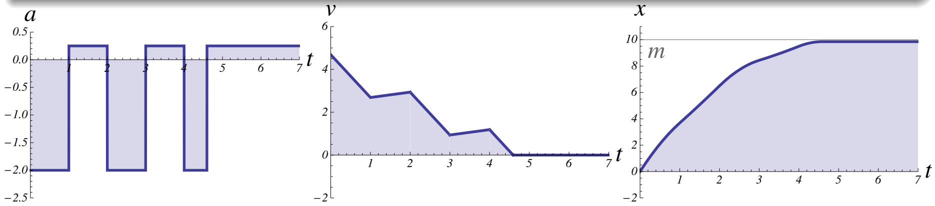
$$H \equiv [t := 0; x' = v, v' = A, t' = 1 \,\&\, v \geq 0 \land t \leq \varepsilon][x' = v, v' = -b]x \leq m$$

Example (Single car car_ε event-triggered)

$$\big(((?H; a := A) \cup a := -b);\ t := 0;\ x' = v, v' = a, t' = 1 \,\&\, v \geq 0 \land t \leq \varepsilon\big)^*$$

Example (Stays before traffic light m if braking would)

$$[x' = v, v' = -b]x \leq m \land v \geq 0 \land A \geq 0 \land b > 0 \rightarrow [car_\varepsilon]x \leq m$$



Example (Model-predictive control equivalence)

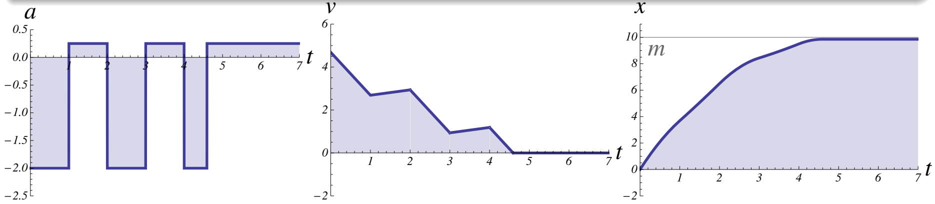
$$H \equiv [t := 0; x' = v, v' = A, t' = 1 \,\&\, v \geq 0 \wedge t \leq \varepsilon][x' = v, v' = -b]x \leq m$$
$$\leftrightarrow 2b(m - x) \geq v^2 + (A + b)(A\varepsilon^2 + 2\varepsilon v)$$

Example (Single car car_ε event-triggered)

$$(((?H; a := A) \cup a := -b);\ t := 0;\ x' = v, v' = a, t' = 1 \,\&\, v \geq 0 \wedge t \leq \varepsilon)^*$$

Example (Stays before traffic light m if braking would)

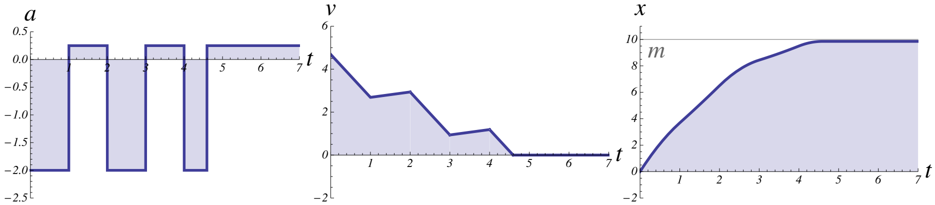
$$[x' = v, v' = -b]x \leq m \wedge v \geq 0 \wedge A \geq 0 \wedge b > 0 \rightarrow [car_\varepsilon]x \leq m$$



Example (dℒ-based model-predictive control design trafo)

$$\_\_\_\_\_\_\_\_ \land\ v \geq 0 \land A \geq 0 \land b > 0 \rightarrow$$

$$[((\ (?\_\_\_\_\_\_\_\_;\ a := A) \cup a := -b);\ t := 0;\ x' = v, v' = a, t' = 1 \,\&\, v \geq 0 \land t \leq \varepsilon)^*]\, x \leq m$$



Example (d$\mathcal{L}$-based model-predictive control design trafo)

$$\underline{\qquad ??? \qquad} \land v \geq 0 \land A \geq 0 \land b > 0 \rightarrow$$

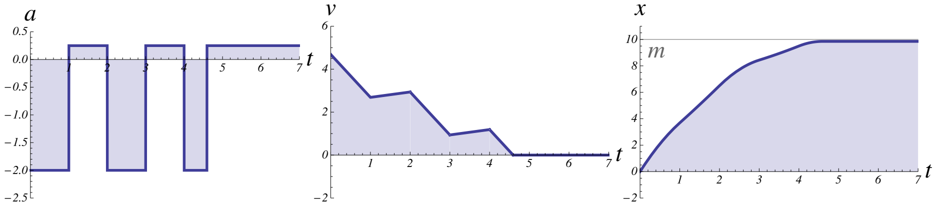
$$\Big[\Big(\Big($$

$$(?\underline{\qquad\qquad\qquad\qquad\qquad\qquad};$$

$$a := A)$$

$$\cup\, a := -b);$$

$$t := 0;\; x' = v, v' = a, t' = 1 \,\&\, v \geq 0 \land t \leq \varepsilon\Big)^*\Big]\, x \leq m$$

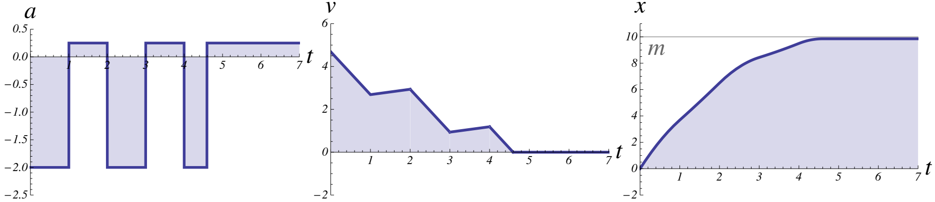


Example (d$\mathcal{L}$-based model-predictive control design trafo)

$$\frac{[x' = v, v' = -b]x \leq m}{} \land v \geq 0 \land A \geq 0 \land b > 0 \rightarrow$$

$$\Big[\Big(\big(\,(?\frac{}{}\;;\; a := A) \cup a := -b\big);$$

$$t := 0;\; x' = v, v' = a, t' = 1 \,\&\, v \geq 0 \land t \leq \varepsilon\Big)^*\Big]\, x \leq m$$



Example (d$\mathcal{L}$-based model-predictive control design trafo)

$$\underline{[x' = v, v' = -b]x \le m} \land v \ge 0 \land A \ge 0 \land b > 0 \to$$

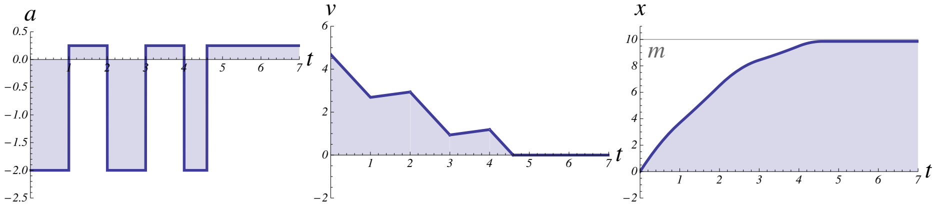
$$\Big[\big(\big($$

$$\big(?\ \underline{\quad ??? \quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad};$$

$$a := A\big)$$

$$\cup\ a := -b\big);$$

$$t := 0;\ x' = v, v' = a, t' = 1\ \&\ v \ge 0 \land t \le \varepsilon\big)^*\Big]\, x \le m$$



Example (d$\mathcal{L}$-based model-predictive control design trafo)

$$\underline{[x' = v, v' = -b]x \leq m} \land v \geq 0 \land A \geq 0 \land b > 0 \rightarrow$$

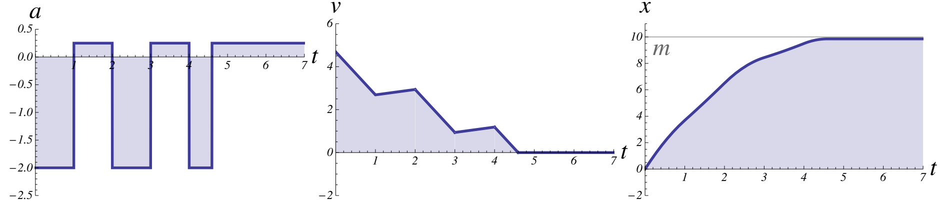
$$\Big[\big(\big($$

$$(?\underline{[t := 0; x' = v, v' = A, t' = 1 \,\&\, v \geq 0 \land t \leq \varepsilon][x' = v, v' = -b]x \leq m};$$

$$a := A)$$

$$\cup\, a := -b);$$

$$t := 0;\; x' = v, v' = a, t' = 1 \,\&\, v \geq 0 \land t \leq \varepsilon\big)^*\Big]\, x \leq m$$



Example (d$\mathcal{L}$-based model-predictive control design trafo)

$$\frac{[x' = v, v' = -b]x \le m}{} \land v \ge 0 \land A \ge 0 \land b > 0 \to$$

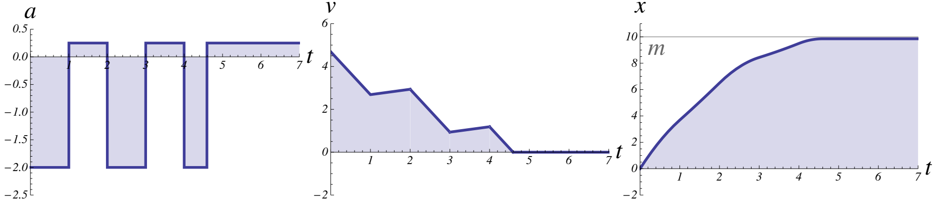
$$\Big[\Big(\Big($$

$$(?\underline{[t := 0; x' = v, v' = A, t' = 1 \,\&\, v \ge 0 \land t \le \varepsilon][x' = v, v' = -b]x \le m};$$

$$a := A)$$

$$\cup\ a := -b);$$

$$t := 0;\ x' = v, v' = a, t' = 1 \,\&\, v \ge 0 \land t \le \varepsilon\Big)^*\Big]\, x \le m$$



Example (d$\mathcal{L}$-based model-predictive control design trafo)

$$\underline{v^2 \le 2b(m-x)} \land v \ge 0 \land A \ge 0 \land b > 0 \to$$

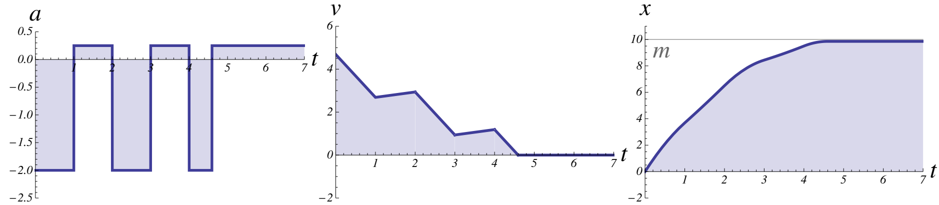
$$\Big[\Big(\Big($$

$$(?\underline{[t:=0; x'=v, v'=A, t'=1 \,\&\, v \ge 0 \land t \le \varepsilon][x'=v, v'=-b]x \le m};$$

$$a := A)$$

$$\cup\, a := -b);$$

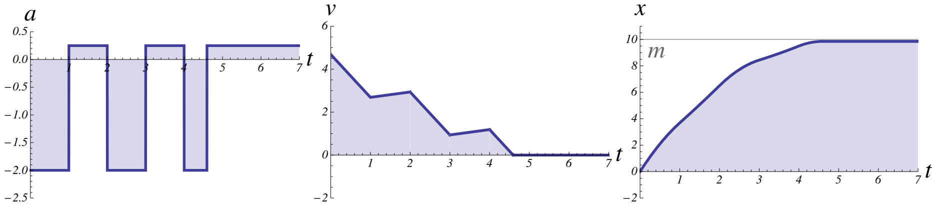
$$t:=0;\ x'=v, v'=a, t'=1 \,\&\, v \ge 0 \land t \le \varepsilon\Big)^*\Big]\, x \le m$$



Example (d$\mathcal{L}$-based model-predictive control design trafo)

$$\underline{v^2 \le 2b(m-x)} \land v \ge 0 \land A \ge 0 \land b > 0 \to$$

$$\Big[\big(\big($$
$$(?\underline{[t := 0; x' = v, v' = A, t' = 1 \,\&\, v \ge 0 \land t \le \varepsilon][x' = v, v' = -b]x \le m};$$
$$a := A)$$
$$\cup\ a := -b);$$
$$t := 0;\ x' = v, v' = a, t' = 1 \,\&\, v \ge 0 \land t \le \varepsilon\big)^*\Big]\, x \le m$$

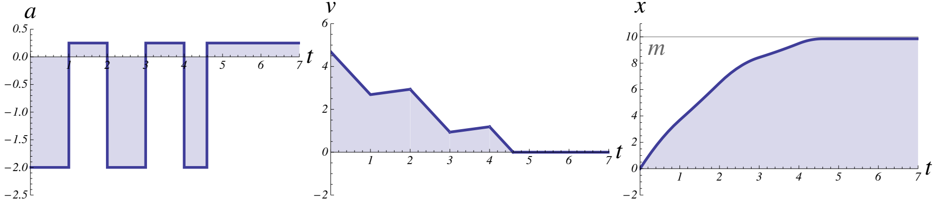


Example (dℒ-based model-predictive control design trafo)

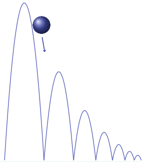
$$\underline{v^2 \le 2b(m-x)} \land v \ge 0 \land A \ge 0 \land b > 0 \rightarrow$$

$$[((\ (?\underline{2b(m-x) \ge v^2 + (A+b)(A\varepsilon^2 + 2\varepsilon v)};\ a := A) \cup a := -b);$$

$$t := 0;\ x' = v, v' = a, t' = 1 \,\&\, v \ge 0 \land t \le \varepsilon)^*]\, x \le m$$



Ex: Bouncing Ball Properties



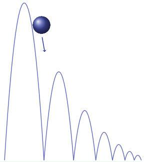
$$\texttt{if}(H)\,\alpha\,\texttt{else}\,\beta \equiv (?H;\alpha) \cup (?\neg H;\beta)$$
$$\texttt{while}(H)\,\alpha \equiv (?H;\alpha)^*;?\neg H$$

Example (Bouncing ball)

$$\big(\,h' = v, v' = -g\,\&\,h \geq 0;$$
$$\quad \text{if } (h = 0) \text{ then}$$
$$\qquad v := -cv$$
$$\quad \text{fi}\big)^*$$

Example (Bouncing ball height?)

$$h = H \land h \geq 0 \land g > 0 \rightarrow [ball](0 \leq h \leq H)$$



$$\texttt{if}(H)\,\alpha\,\texttt{else}\,\beta \equiv (?H;\alpha) \cup (?\neg H;\beta)$$

$$\texttt{while}(H)\,\alpha \equiv (?H;\alpha)^*;?\neg H$$

Example (Bouncing ball)

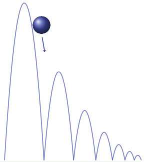
$$\big(\, h' = v, v' = -g \,\&\, h \geq 0;$$
$$\quad \text{if } (h = 0) \text{ then}$$
$$\qquad v := -cv$$
$$\quad \text{fi}\big)^*$$

Example (Bouncing ball height?)

$$h = H \land h \geq 0 \land g > 0 \rightarrow [ball](0 \leq h \leq H)$$

Not if $c > 1$ anti-damping

Ex: Bouncing Ball Properties



$$\texttt{if}(H)\,\alpha\,\texttt{else}\,\beta \equiv (?H;\alpha) \cup (?\neg H;\beta)$$

$$\texttt{while}(H)\,\alpha \equiv (?H;\alpha)^*;?\neg H$$

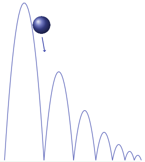
Example (Bouncing ball)

$$\big(\,h' = v, v' = -g \,\&\, h \geq 0;$$
$$\quad \text{if } (h = 0) \text{ then}$$
$$\qquad v := -cv$$
$$\quad \text{fi}\big)^*$$

Example (Bouncing ball height?)

$$1 > c \geq 0 \land h = H \land h \geq 0 \land g > 0 \rightarrow [ball](0 \leq h \leq H)$$

Ex: Bouncing Ball Properties



$$\texttt{if}(H)\,\alpha\,\texttt{else}\,\beta \equiv (?H;\alpha) \cup (?\neg H;\beta)$$

$$\texttt{while}(H)\,\alpha \equiv (?H;\alpha)^*; ?\neg H$$

Example (Bouncing ball)

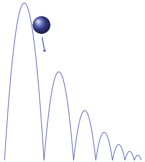
$\big(\, h' = v, v' = -g \,\&\, h \geq 0;$
 if $(h = 0)$ then
  $v := -cv$
 fi$\big)^*$

Example (Bouncing ball height?)

$$1 > c \geq 0 \land h = H \land h \geq 0 \land g > 0 \rightarrow [ball](0 \leq h \leq H)$$

Not if $v > 0$ climbing, initially

Ex: Bouncing Ball Properties



$$\texttt{if}(H)\,\alpha\,\texttt{else}\,\beta \equiv (?H;\alpha) \cup (?\neg H;\beta)$$
$$\texttt{while}(H)\,\alpha \equiv (?H;\alpha)^*;\,?\neg H$$

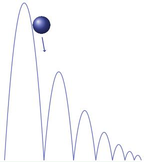
Example (Bouncing ball)

$$\Big(\, h' = v, v' = -g \,\&\, h \geq 0;$$
$$\text{if } (h = 0) \text{ then}$$
$$v := -cv$$
$$\text{fi}\Big)^*$$

Example (Bouncing ball height?)

$$v \leq 0 \land 1 > c \geq 0 \land h = H \land h \geq 0 \land g > 0 \rightarrow [ball](0 \leq h \leq H)$$

Ex: Bouncing Ball Properties



$$\texttt{if}(H)\,\alpha\,\texttt{else}\,\beta \equiv (?H;\alpha) \cup (?\neg H;\beta)$$

$$\texttt{while}(H)\,\alpha \equiv (?H;\alpha)^*;\,?\neg H$$

Example (▶ Bouncing ball)

$$\Big(\, h' = v, v' = -g \,\&\, h \geq 0;$$
$$\quad \text{if } (h = 0) \text{ then}$$
$$\qquad v := -cv$$
$$\quad \text{fi}\Big)^*$$

Example (Bouncing ball height?)

$$v \leq 0 \land 1 > c \geq 0 \land h = H \land h \geq 0 \land g > 0 \rightarrow [ball](0 \leq h \leq H)$$

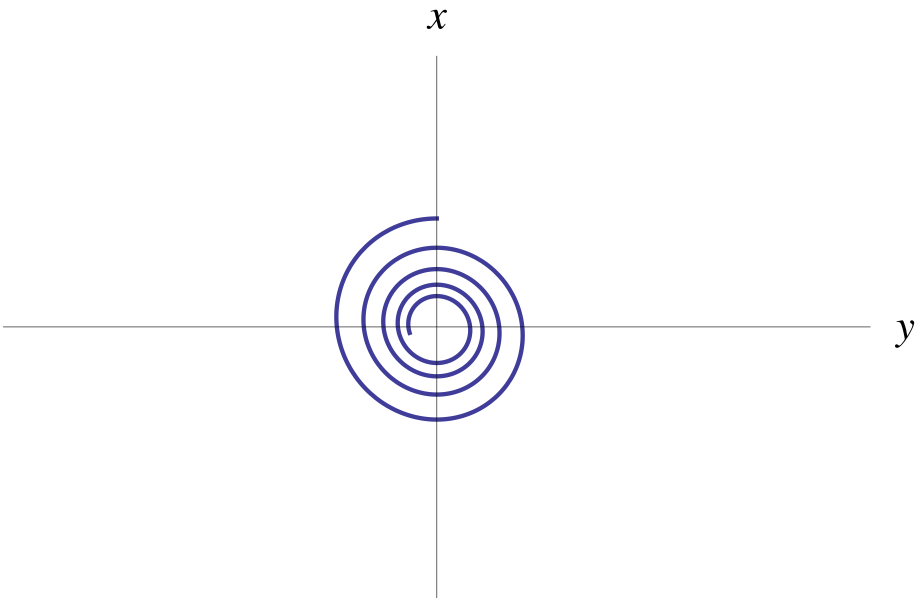
Not if $v \ll 0$ dribbling, initially

Arbitrarily Nested Modalities and Nested Quantifiers

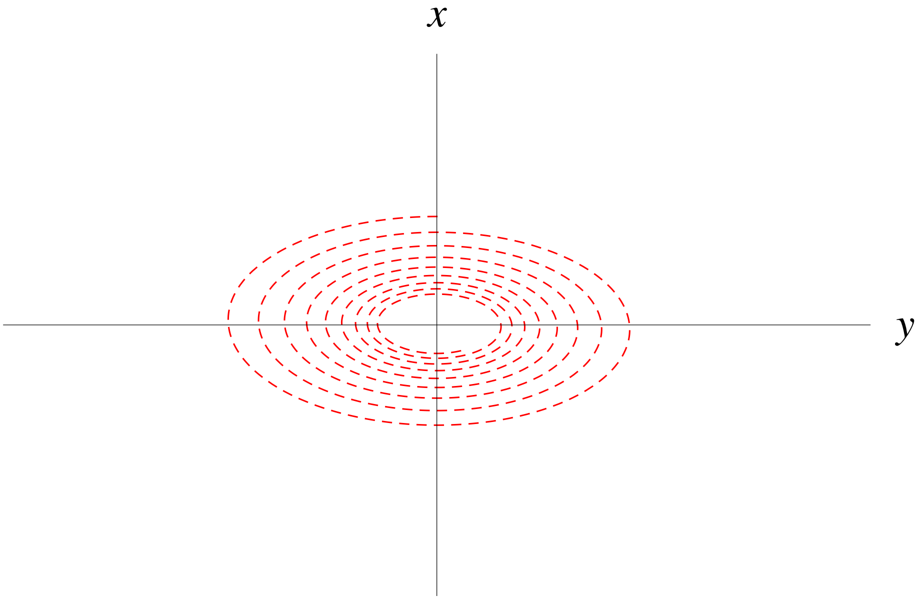
Example (Nest boxes and be happy)

- $[RBC]\text{partitioned} \;\rightarrow\; \exists SB\, \langle \text{Train} \rangle [RBC]\text{safe}$
- $([\text{Train}]\text{safe}) \leftrightarrow \frac{v^2}{2b} \leq m - z \ldots$
- $[\text{rbc}](M \rightarrow [\text{spd}]\langle SB := * \rangle [\text{atp}; \text{drive}]\text{safe})$
- $[\text{aircraft}_1]\langle \text{aircraft}_2 \rangle \text{separate}$

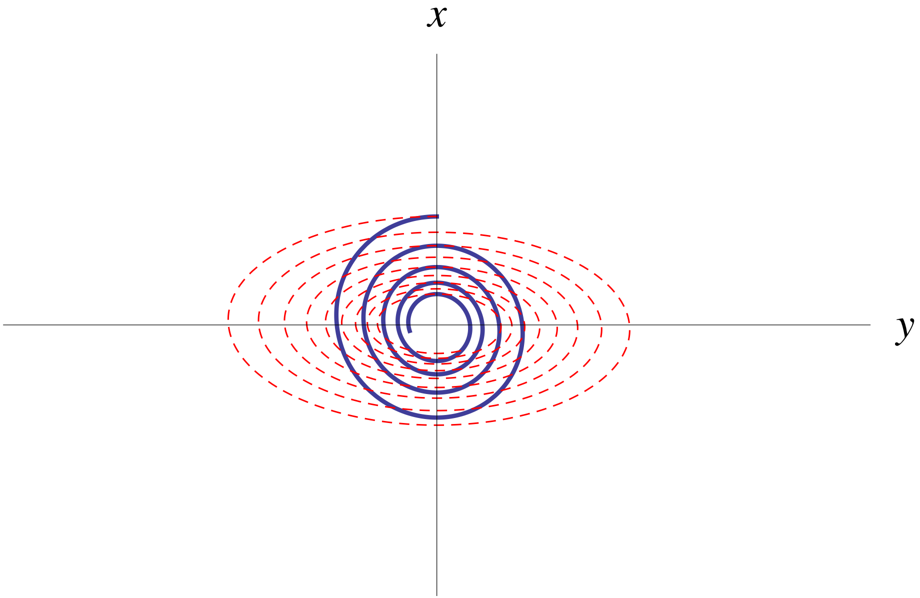
Compositionality in Hybrid Systems



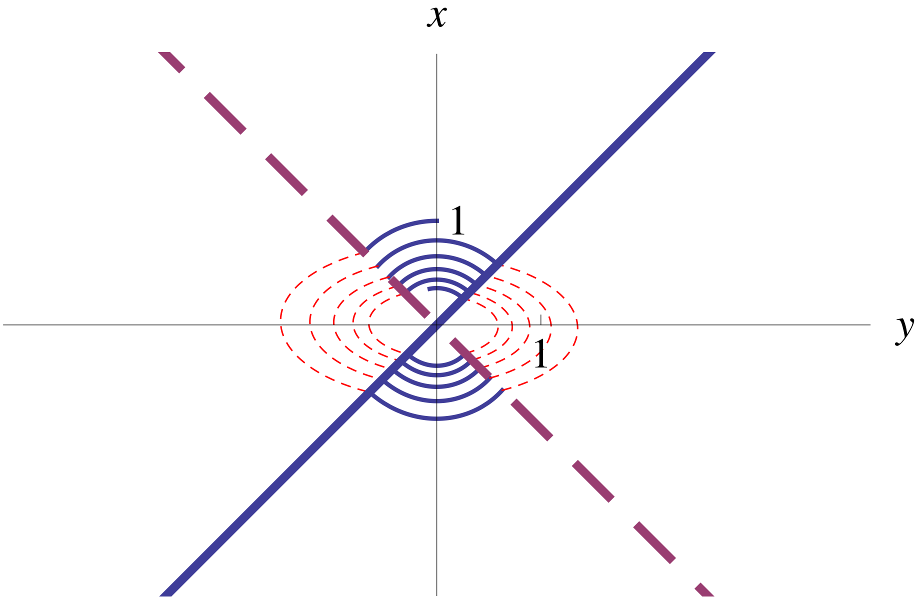
Compositionality in Hybrid Systems



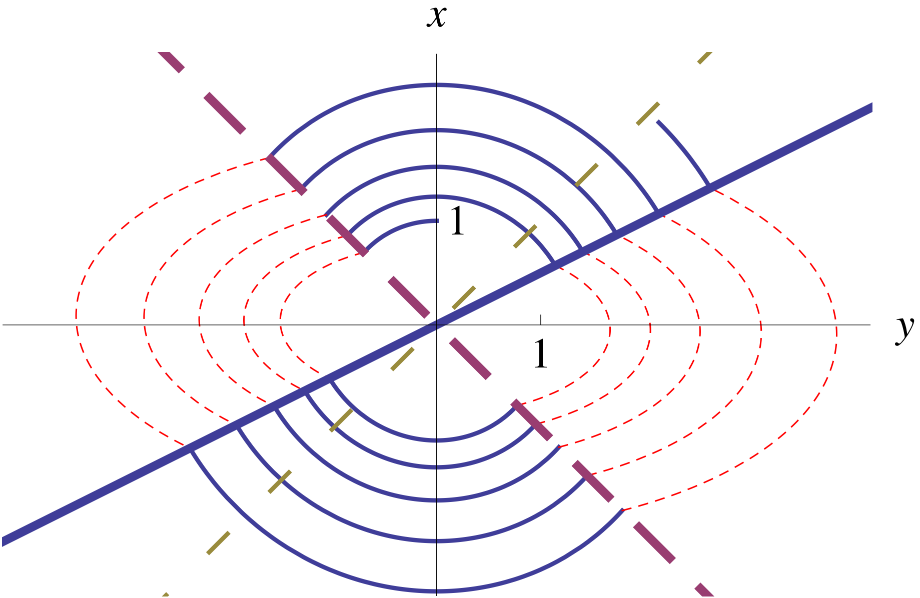
Compositionality in Hybrid Systems



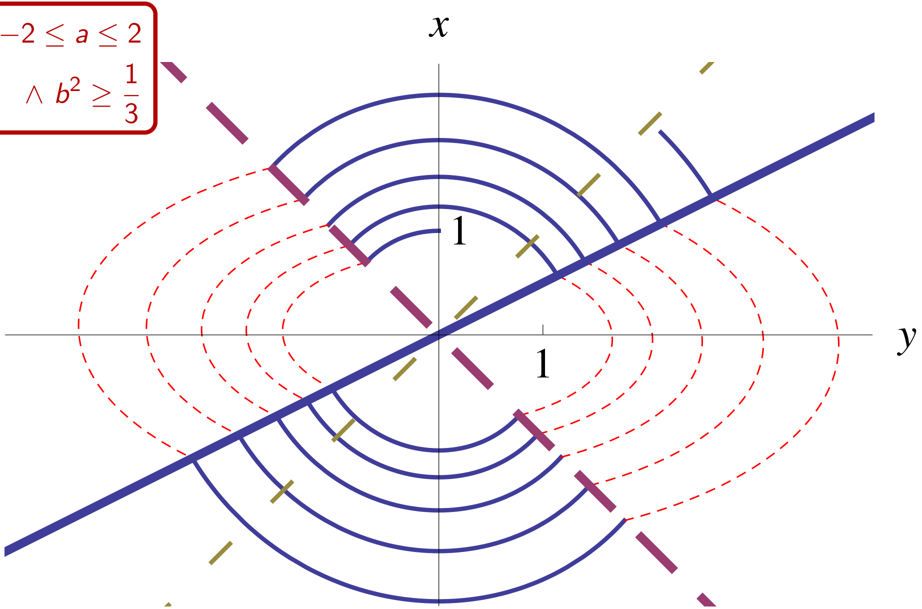
Compositionality in Hybrid Systems



Compositionality in Hybrid Systems



$$-2 \le a \le 2 \land b^2 \ge \frac{1}{3}$$



Outline

Differential Dynamic Logic d$\mathcal{L}$: Axiomatization

$$([:=]) \quad [x := \theta][(x)]\phi x \leftrightarrow [(x)]\phi\theta$$

$$([?]) \quad [?H]\phi \leftrightarrow (H \rightarrow \phi)$$

$$(['])\quad [x' = f(x)]\phi \leftrightarrow \forall t{\geq}0\, [x := y(t)]\phi \qquad (y'(t) = f(y))$$

$$([\cup]) \quad [\alpha \cup \beta]\phi \leftrightarrow [\alpha]\phi \wedge [\beta]\phi$$

$$([;]) \quad [\alpha; \beta]\phi \leftrightarrow [\alpha][\beta]\phi$$

$$([^*]) \quad [\alpha^*]\phi \leftrightarrow \phi \wedge [\alpha][\alpha^*]\phi$$

$$(K) \quad [\alpha](\phi \rightarrow \psi) \rightarrow ([\alpha]\phi \rightarrow [\alpha]\psi)$$

$$(I) \quad [\alpha^*](\phi \rightarrow [\alpha]\phi) \rightarrow (\phi \rightarrow [\alpha^*]\phi)$$

$$(C) \quad [\alpha^*]\forall v{>}0\, (\varphi(v) \rightarrow \langle\alpha\rangle\varphi(v-1)) \rightarrow \forall v\, (\varphi(v) \rightarrow \langle\alpha^*\rangle\exists v{\leq}0\, \varphi(v))$$

Differential Dynamic Logic d$\mathcal{L}$: Axiomatization

(G) $$\frac{\phi}{[\alpha]\phi}$$

(MP) $$\frac{\phi \to \psi \quad \phi}{\psi}$$

(∀) $$\frac{\phi}{\forall x\, \phi}$$

(G) $$\frac{\phi}{[\alpha]\phi}$$

(MP) $$\frac{\phi \to \psi \quad \phi}{\psi}$$

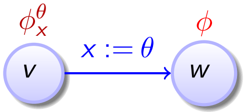
(∀) $$\frac{\phi}{\forall x\, \phi}$$

(B) $\forall x\, [\alpha]\phi \to [\alpha]\forall x\, \phi \qquad (x \notin \alpha)$

(V) $\phi \to [\alpha]\phi \qquad (FV(\phi) \cap BV(\alpha) = \emptyset)$

Proof Calculus for Differential Dynamic Logic

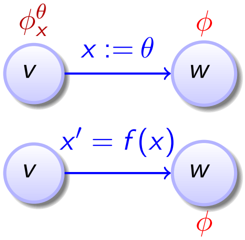
$$\frac{\phi_x^\theta}{[x := \theta]\phi}$$



Proof Calculus for Differential Dynamic Logic

$$\frac{\phi_x^\theta}{[x := \theta]\phi}$$

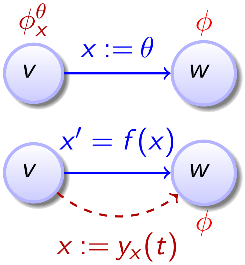
$$\frac{\forall t{\geq}0\, [x := y_x(t)]\phi}{[x' = f(x)]\phi}$$



Proof Calculus for Differential Dynamic Logic

$$\frac{\phi_x^\theta}{[x := \theta]\phi}$$

$$\frac{\forall t{\geq}0\, [x := y_x(t)]\phi}{[x' = f(x)]\phi}$$

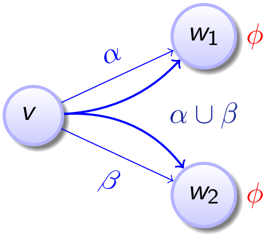


Proof Calculus for Differential Dynamic Logic

compositional semantics $\Rightarrow$ compositional rules!

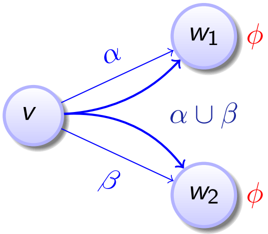
Proof Calculus for Differential Dynamic Logic

$$\frac{[\alpha]\phi \land [\beta]\phi}{[\alpha \cup \beta]\phi}$$

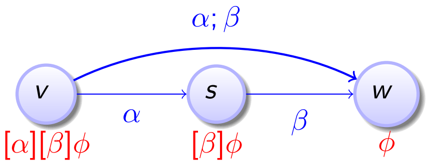


Proof Calculus for Differential Dynamic Logic

$$\frac{[\alpha]\phi \wedge [\beta]\phi}{[\alpha \cup \beta]\phi}$$

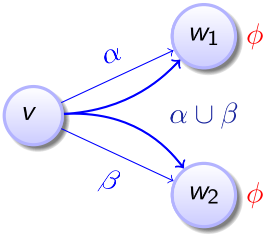


$$\frac{[\alpha][\beta]\phi}{[\alpha; \beta]\phi}$$

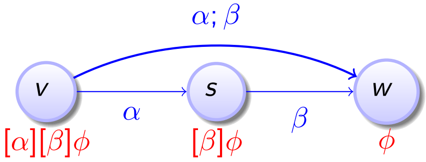


Proof Calculus for Differential Dynamic Logic

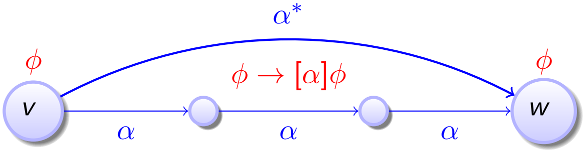
$$\frac{[\alpha]\phi \land [\beta]\phi}{[\alpha \cup \beta]\phi}$$



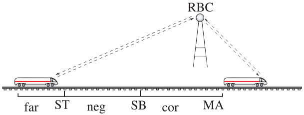
$$\frac{[\alpha][\beta]\phi}{[\alpha; \beta]\phi}$$



$$\frac{\phi \quad (\phi \to [\alpha]\phi)}{[\alpha^*]\phi}$$

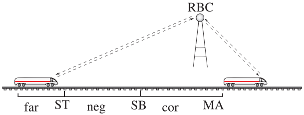


Deduction Modulo Real Arithmetic



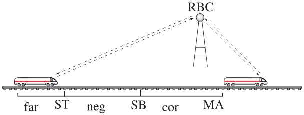
$$v \geq 0 \land z < MA \rightarrow \langle z' = v, v' = -b \rangle\ z > MA$$

Deduction Modulo Real Arithmetic



$$\dfrac{\dfrac{v \geq 0, z < MA \rightarrow \exists t \geq 0 \, \langle z := -\frac{b}{2}t^2 + vt + z \rangle z > MA}{v \geq 0, z < MA \rightarrow \langle z' = v, v' = -b \rangle z > MA}}{v \geq 0 \land z < MA \rightarrow \langle z' = v, v' = -b \rangle \; z > MA}$$

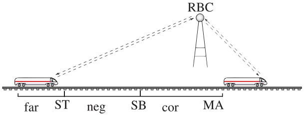
Deduction Modulo Real Arithmetic



Collins/Tarski QE not applicable!

$$\dfrac{\dfrac{v \geq 0, z < MA \rightarrow \exists t{\geq}0\, \langle z := -\frac{b}{2}t^2 + vt + z\rangle z > MA}{v \geq 0, z < MA \rightarrow \langle z' = v, v' = -b\rangle z > MA}}{v \geq 0 \land z < MA \rightarrow \langle z' = v, v' = -b\rangle\; z > MA}$$

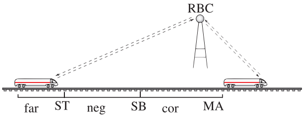
Deduction Modulo (Side Deduction)



$$\frac{}{v \geq 0, z < MA \to t \geq 0 \land \langle z := -\frac{b}{2}t^2 + vt + z\rangle z > MA}$$

start side

$$\cfrac{\cfrac{\cfrac{}{v \geq 0, z < MA \to \exists t{\geq}0\, \langle z := -\frac{b}{2}t^2 + vt + z\rangle z > MA}}{v \geq 0, z < MA \to \langle z' = v, v' = -b\rangle z > MA}}{v \geq 0 \land z < MA \to \langle z' = v, v' = -b\rangle\ z > MA}$$

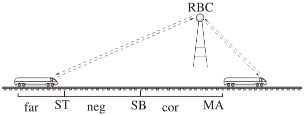


$$\dfrac{v \geq 0, z < MA \rightarrow t \geq 0 \qquad \dfrac{v \geq 0, z < MA \rightarrow -\frac{b}{2}t^2 + vt + z > MA}{v \geq 0, z < MA \rightarrow \langle z := -\frac{b}{2}t^2 + vt + z\rangle z > M}}{v \geq 0, z < MA \rightarrow t \geq 0 \wedge \langle z := -\frac{b}{2}t^2 + vt + z\rangle z > MA}$$

start
side

$$\dfrac{\dfrac{\dfrac{}{v \geq 0, z < MA \rightarrow \exists t{\geq}0\, \langle z := -\frac{b}{2}t^2 + vt + z\rangle z > MA}}{v \geq 0, z < MA \rightarrow \langle z' = v, v' = -b\rangle z > MA}}{v \geq 0 \wedge z < MA \rightarrow \langle z' = v, v' = -b\rangle\ z > MA}$$

Deduction Modulo (Side Deduction)



$$v \geq 0, z < MA \rightarrow -\frac{b}{2}t^2 + vt + z > MA$$

$$v \geq 0, z < MA \rightarrow t \geq 0 \qquad v \geq 0, z < MA \rightarrow \langle z := -\frac{b}{2}t^2 + vt + z\rangle z > M$$

$$v \geq 0, z < MA \rightarrow t \geq 0 \wedge \langle z := -\frac{b}{2}t^2 + vt + z\rangle z > MA$$

QE

start side

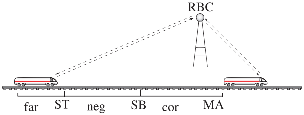
$$v \geq 0, z < MA \rightarrow \text{QE}(\exists t\,(\ldots t \geq 0 \wedge -\frac{b}{2}t^2 + vt + z > MA))$$

$$v \geq 0, z < MA \rightarrow \exists t {\geq} 0\, \langle z := -\frac{b}{2}t^2 + vt + z\rangle z > MA$$

$$v \geq 0, z < MA \rightarrow \langle z' = v, v' = -b\rangle z > MA$$

$$v \geq 0 \wedge z < MA \rightarrow \langle z' = v, v' = -b\rangle\ z > MA$$

Deduction Modulo (Side Deduction)



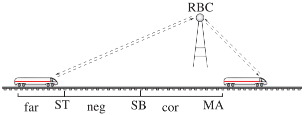
$$\dfrac{v \geq 0, z < MA \to -\frac{b}{2}t^2 + vt + z > MA}{v \geq 0, z < MA \to \langle z := -\frac{b}{2}t^2 + vt + z\rangle z > M}$$

$$\dfrac{v \geq 0, z < MA \to t \geq 0 \qquad v \geq 0, z < MA \to \langle z := -\frac{b}{2}t^2 + vt + z\rangle z > M}{v \geq 0, z < MA \to t \geq 0 \wedge \langle z := -\frac{b}{2}t^2 + vt + z\rangle z > MA}$$

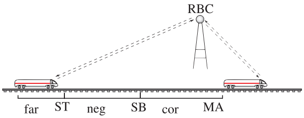
QE

start side

$$\dfrac{\dfrac{\dfrac{v \geq 0, z < MA \to v^2 > 2b(MA - z)}{v \geq 0, z < MA \to \exists t \geq 0\, \langle z := -\frac{b}{2}t^2 + vt + z\rangle z > MA}}{v \geq 0, z < MA \to \langle z' = v, v' = -b\rangle z > MA}}{v \geq 0 \wedge z < MA \to \langle z' = v, v' = -b\rangle\, z > MA}$$



$$\frac{\displaystyle\frac{v \geq 0, z < MA \to \exists t{\geq}0\, \langle z := -\tfrac{b}{2}t^2 + vt + z\rangle z{>}MA}{v \geq 0, z < MA \to \langle z' = v, v' = -b\rangle z > MA}}{v \geq 0 \land z < MA \to \langle z' = v, v' = -b\rangle z > MA}$$



$$v \geq 0, z < MA \rightarrow -\frac{b}{2}T^2 + vT + z > MA$$

$$v \geq 0, z < MA \rightarrow T \geq 0 \qquad v \geq 0, z < MA \rightarrow \langle z := -\frac{b}{2}T^2 + vT + z\rangle z > M$$

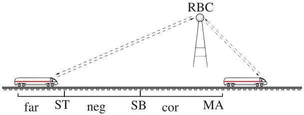
$$v \geq 0, z < MA \rightarrow T \geq 0 \wedge \langle z := -\frac{b}{2}T^2 + vT + z\rangle z > MA$$

$$v \geq 0, z < MA \rightarrow \exists t{\geq}0\, \langle z := -\frac{b}{2}t^2 + vt + z\rangle z{>}MA$$

$$v \geq 0, z < MA \rightarrow \langle z' = v, v' = -b\rangle z > MA$$

$$v \geq 0 \wedge z < MA \rightarrow \langle z' = v, v' = -b\rangle z > MA$$

Deduction Modulo (Free Variables for Automation)



$$v \geq 0, z < MA \rightarrow \quad \exists T\, (\dots T \geq 0 \land -\tfrac{b}{2}T^2 + vT + z > MA)$$

$$v \geq 0, z < MA \rightarrow -\tfrac{b}{2}T^2 + vT + z > MA$$

$$v \geq 0, z < MA \rightarrow T \geq 0 \qquad v \geq 0, z < MA \rightarrow \langle z := -\tfrac{b}{2}T^2 + vT + z\rangle z > M$$

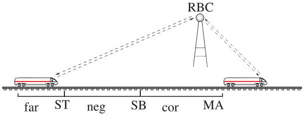
$$v \geq 0, z < MA \rightarrow T \geq 0 \land \langle z := -\tfrac{b}{2}T^2 + vT + z\rangle z > MA$$

$$v \geq 0, z < MA \rightarrow \exists t \geq 0\, \langle z := -\tfrac{b}{2}t^2 + vt + z\rangle z > MA$$

$$v \geq 0, z < MA \rightarrow \langle z' = v, v' = -b\rangle z > MA$$

$$v \geq 0 \land z < MA \rightarrow \langle z' = v, v' = -b\rangle z > MA$$

Deduction Modulo (Free Variables for Automation)



$$v \geq 0, z < MA \rightarrow \mathrm{QE}\big(\exists T\,(\ldots T \geq 0 \wedge -\tfrac{b}{2}T^2 + vT + z > MA)\big)$$

$$v \geq 0, z < MA \rightarrow -\tfrac{b}{2}T^2 + vT + z > MA$$

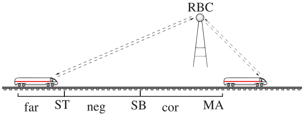
$$v \geq 0, z < MA \rightarrow T \geq 0 \qquad v \geq 0, z < MA \rightarrow \langle z := -\tfrac{b}{2}T^2 + vT + z\rangle z > M$$

$$v \geq 0, z < MA \rightarrow T \geq 0 \wedge \langle z := -\tfrac{b}{2}T^2 + vT + z\rangle z > MA$$

$$v \geq 0, z < MA \rightarrow \exists t \geq 0\, \langle z := -\tfrac{b}{2}t^2 + vt + z\rangle z > MA$$

$$v \geq 0, z < MA \rightarrow \langle z' = v, v' = -b\rangle z > MA$$

$$v \geq 0 \wedge z < MA \rightarrow \langle z' = v, v' = -b\rangle z > MA$$



$$v \geq 0, z < MA \rightarrow v^2 > 2b(MA - z)$$

$$v \geq 0, z < MA \rightarrow -\frac{b}{2}T^2 + vT + z > MA$$

$$v \geq 0, z < MA \rightarrow T \geq 0 \qquad v \geq 0, z < MA \rightarrow \langle z := -\frac{b}{2}T^2 + vT + z\rangle z > M$$

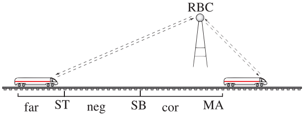
$$v \geq 0, z < MA \rightarrow T \geq 0 \wedge \langle z := -\frac{b}{2}T^2 + vT + z\rangle z > MA$$

$$v \geq 0, z < MA \rightarrow \exists t \geq 0\, \langle z := -\frac{b}{2}t^2 + vt + z\rangle z > MA$$

$$v \geq 0, z < MA \rightarrow \langle z' = v, v' = -b\rangle z > MA$$

$$v \geq 0 \wedge z < MA \rightarrow \langle z' = v, v' = -b\rangle z > MA$$

Deduction Modulo (Free Variables for Automation)



- For requantification, not for unification

$$v \geq 0, z < MA \to \mathrm{QE}\big(\exists T\, (\ldots T \geq 0 \land -\tfrac{b}{2} T^2 + vT + z > MA)\big)$$

$$v \geq 0, z < MA \to -\tfrac{b}{2} T^2 + vT + z > MA$$

$$v \geq 0, z < MA \to T \geq 0 \qquad v \geq 0, z < MA \to \langle z := -\tfrac{b}{2} T^2 + vT + z \rangle z > M$$

$$v \geq 0, z < MA \to T \geq 0 \land \langle z := -\tfrac{b}{2} T^2 + vT + z \rangle z > MA$$

$$v \geq 0, z < MA \to \exists t \geq 0\, \langle z := -\tfrac{b}{2} t^2 + vt + z \rangle z > MA$$

$$v \geq 0, z < MA \to \langle z' = v, v' = -b \rangle z > MA$$

$$v \geq 0 \land z < MA \to \langle z' = v, v' = -b \rangle z > MA$$

$$\cfrac{\cfrac{\cfrac{}{(X < S)}}{\forall s\,(X < s)}}{\exists x\,\forall s\,(x < s)}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\overline{\text{QE}(\forall S \exists X\,(X < S))}}{(X < S)}}{\forall s\,(X < s)}}{\exists x\,\forall s\,(x < s)}}{}$$

Deduction Modulo (Free Variables for Automation)

$$\dfrac{\dfrac{\dfrac{\overline{\mathrm{QE}(\forall S \exists X (X < S))} \qquad \overline{\mathrm{QE}(\exists X \forall S (X < S))}}{(X < S)}}{\forall s\,(X < s)}}{\exists x\,\forall s\,(x < s)}$$

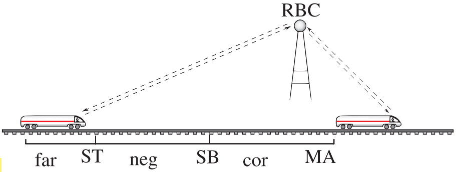
$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{true}{\mathrm{QE}(\forall S \exists X (X < S))} \qquad \dfrac{false}{\mathrm{QE}(\exists X \forall S (X < S))}}{(X < S)}}{\forall s\, (X < s)}}{\exists x\, \forall s\, (x < s)}}{false!}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{true}{\mathrm{QE}(\forall S \exists X (X < S))} \qquad \dfrac{false}{\mathrm{QE}(\exists X \forall S (X < S))}}{(X < S)}}{\forall s\, (X < s)}}{\exists x\, \forall s\, (x < s)}}{false!}$$

Skolemisation $S(X)$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\mathit{false}}{\mathrm{QE}(\exists X \forall S\,(X < S))}}{(X < S(X))}}{\forall s\,(X < s)}}{\exists x\,\forall s\,(x < s)}}{\mathit{false}!}$$

```
/* initial state characterization */
(v^2 <= 2*b*(m-z) & b>0 & A>=0) ->
\[(
  SB := (v^2)/(2*b) + ((A/b)+1) * ((A/2)*ep^2+ep*v);
  ((?m-z <= SB; a:= -b) ++ (?m-z >= SB; a:=A));
  t:=0;
  {z'=v, v'=a, t'=1, (v >= 0&t <= ep)}  /* drive */
 )*                     /* repeat these transitions */
\] (z < m)                /* safety / postcondition */
```



ETCS Train Control [bug]

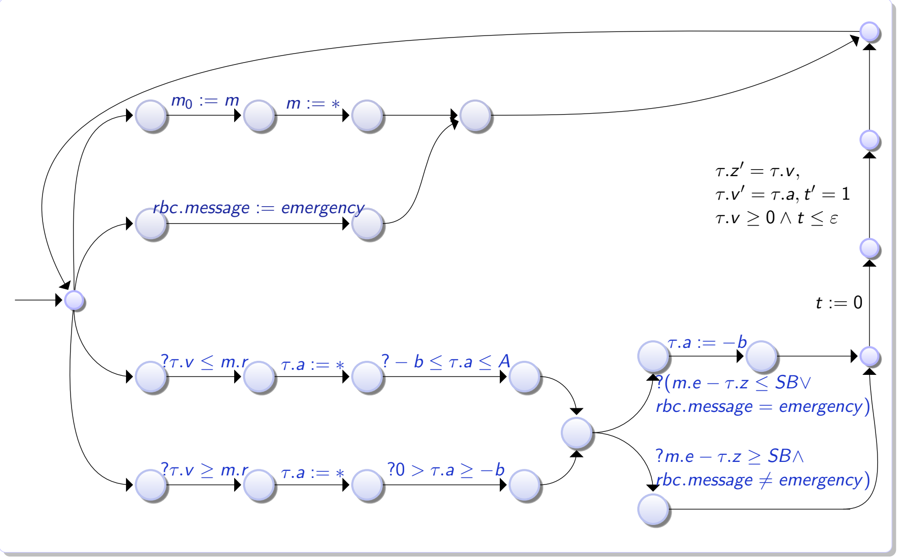
Read from the informal specification...

$$
\begin{aligned}
ETCS_{skel} &: (train \cup RBC)^* \\
train &: spd;\ atp;\ drive \\
spd &: \quad (?\tau.v \leq m.r;\ \tau.a := *;\ ?-b \leq \tau.a \leq A) \\
&\quad \cup (?\tau.v \geq m.r;\ \tau.a := *;\ ?-b \leq \tau.a \leq 0) \\
atp &: \mathtt{if}(m.e - \tau.z \leq SB \vee RBC.message = emergency)\, \tau.a := -b \\
drive &: t := 0;\ (\tau.z' = \tau.v, \tau.v' = \tau.a, t' = 1 \,\&\, \tau.v \geq 0 \wedge t \leq \varepsilon) \\
RBC &: (RBC.message := emergency)\ \cup\ (m := *;\ ?m.r > 0)
\end{aligned}
$$

Parametric Skeleton of ETCS

As transition system...

▸ ETCS Train Control [safety]



Soundness

Theorem (Soundness)

d$\mathcal{L}$ *calculus is sound, i.e., all provable* d$\mathcal{L}$ *formulas are valid:*

$$\vdash \phi \text{ implies } \vDash \phi$$

What about the converse?

Soundness

Theorem (Soundness)

d$\mathcal{L}$ *calculus is sound, i.e., all provable* d$\mathcal{L}$ *formulas are valid:*

$$\vdash \phi \text{ implies } \vDash \phi$$

What about the converse?

$$(s := s + 2n + 1; n := n + 1)^* \quad \rightsquigarrow \quad s = n^2$$

Soundness

Theorem (Soundness)

d$\mathcal{L}$ *calculus is sound, i.e., all provable* d$\mathcal{L}$ *formulas are valid:*

$$\vdash \phi \text{ implies } \vDash \phi$$

What about the converse?

$$(s := s + 2n + 1; n := n + 1)^* \quad \leadsto \quad s = n^2$$
$$x' = 5 \quad \leadsto \quad x(t) = 5t + x_0$$

Soundness

Theorem (Soundness)

d$\mathcal{L}$ *calculus is sound, i.e., all provable* d$\mathcal{L}$ *formulas are valid:*

$$\vdash \phi \textit{ implies } \vDash \phi$$

What about the converse?

$$\begin{array}{lcl}
(s := s + 2n + 1; n := n + 1)^* & \leadsto & s = n^2 \\
x' = 5 & \leadsto & x(t) = 5t + x_0 \\
x' = x & \leadsto & x(t) = x_0 e^t
\end{array}$$

Soundness

Theorem (Soundness)

d$\mathcal{L}$ *calculus is sound, i.e., all provable* d$\mathcal{L}$ *formulas are valid:*

$$\vdash \phi \text{ implies } \vDash \phi$$

What about the converse?

$$\begin{array}{lcl}
(s := s + 2n + 1; n := n + 1)^* & \leadsto & s = n^2 \\
x' = 5 & \leadsto & x(t) = 5t + x_0 \\
x' = x & \leadsto & x(t) = x_0 e^t \\
x'' = -x & \leadsto & x(t) = x_0 \cos t + x_0' \sin t
\end{array}$$

Incompleteness

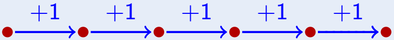
Theorem

Discrete fragment and continuous fragment of $\mathrm{d}\mathcal{L}$ *characterize* $\mathbb{N}$

Proof.

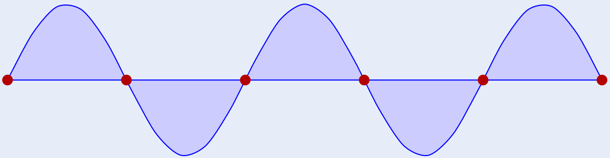
Discrete fragment:

$$\langle (x := x+1)^* \rangle\ x = n$$



Continuous fragment:

$$\langle s'' = -s, \tau' = 1 \rangle (s = 0 \land \tau = n) \qquad \leadsto s = \sin$$



□

Complete Proof Theory of Hybrid Systems

Theorem (Relative Completeness) (J.Autom.Reas. 2008)

d$\mathcal{L}$ *calculus is a sound & complete axiomatization of hybrid systems relative to differential equations.* ▸ Proof 15p

Complete Proof Theory of Hybrid Systems

Theorem (Continuous Relative Completeness) (J.Autom.Reas. 2008)

d$\mathcal{L}$ *calculus is a sound & complete axiomatization of hybrid systems relative to differential equations.*

▸ Proof 15p

Theorem (Discrete Relative Completeness) (LICS'12)

d$\mathcal{L}$ *calculus is a sound & complete axiomatization of hybrid systems relative to discrete dynamics.*

▸ Proof +10p

Complete Proof Theory of Hybrid Systems

Theorem (Continuous Relative Completeness) (J.Autom.Reas. 2008)

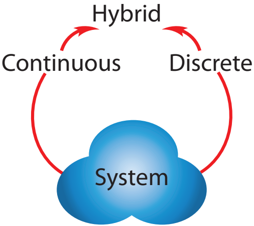
d$\mathcal{L}$ *calculus is a sound & complete axiomatization of hybrid systems relative to differential equations.*

▸ Proof 15p

Theorem (Discrete Relative Completeness) (LICS'12)

d$\mathcal{L}$ *calculus is a sound & complete axiomatization of hybrid systems relative to discrete dynamics.*

▸ Proof +10p



Complete Proof Theory of Hybrid Systems

Theorem (Continuous Relative Completeness) (J.Autom.Reas. 2008)

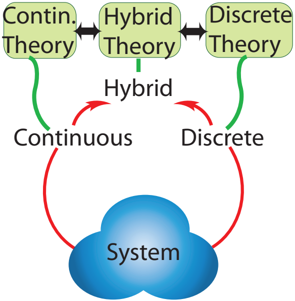
d$\mathcal{L}$ *calculus is a sound & complete axiomatization of hybrid systems relative to differential equations.*

▸ Proof 15p

Theorem (Discrete Relative Completeness) (LICS'12)

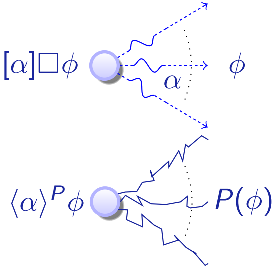
d$\mathcal{L}$ *calculus is a sound & complete axiomatization of hybrid systems relative to discrete dynamics.*

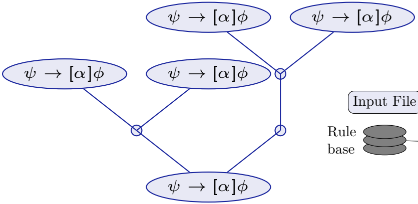
▸ Proof +10p

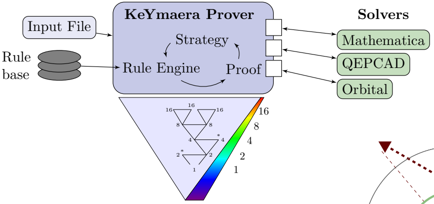


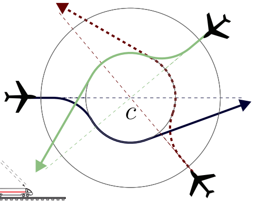
Outline

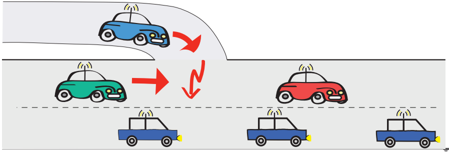

¬F
F
F

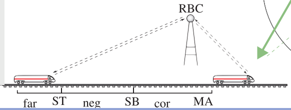

$[\alpha]\Box\phi$
α
ϕ
$\langle\alpha\rangle^P\phi$
$P(\phi)$


$\psi \to [\alpha]\phi$
$\psi \to [\alpha]\phi$
$\psi \to [\alpha]\phi$
$\psi \to [\alpha]\phi$
$\psi \to [\alpha]\phi$

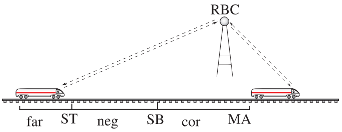

KeYmaera Prover
Input File
Rule base
Strategy
Rule Engine
Proof
Solvers
Mathematica
QEPCAD
Orbital
16
8
4
2
1

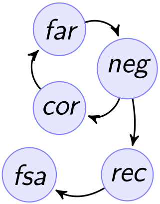

c

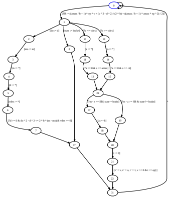


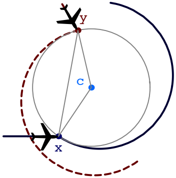

RBC
far
ST
neg
SB
cor
MA

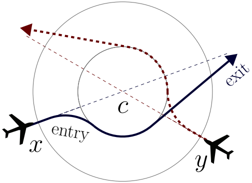
Successful Hybrid Systems Proofs

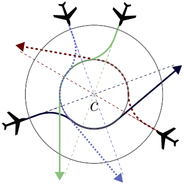


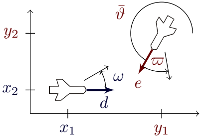


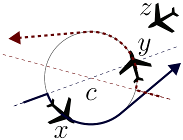


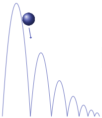


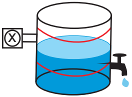




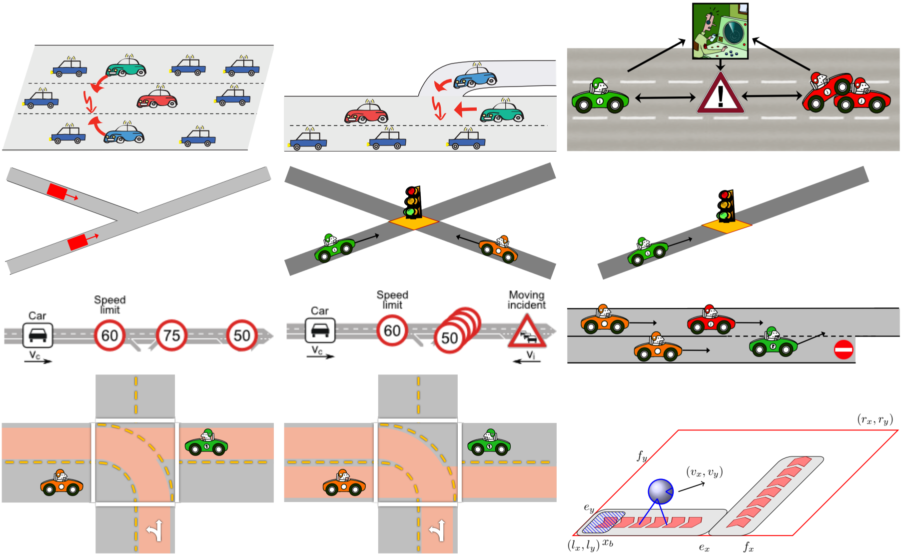






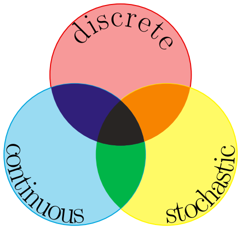


Successful Hybrid Systems Proofs



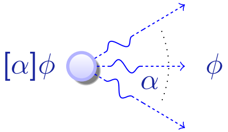
Outline

Logical Analysis of Hybrid Systems



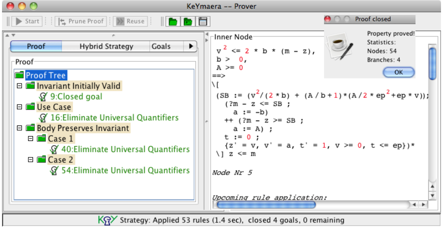
differential dynamic logic

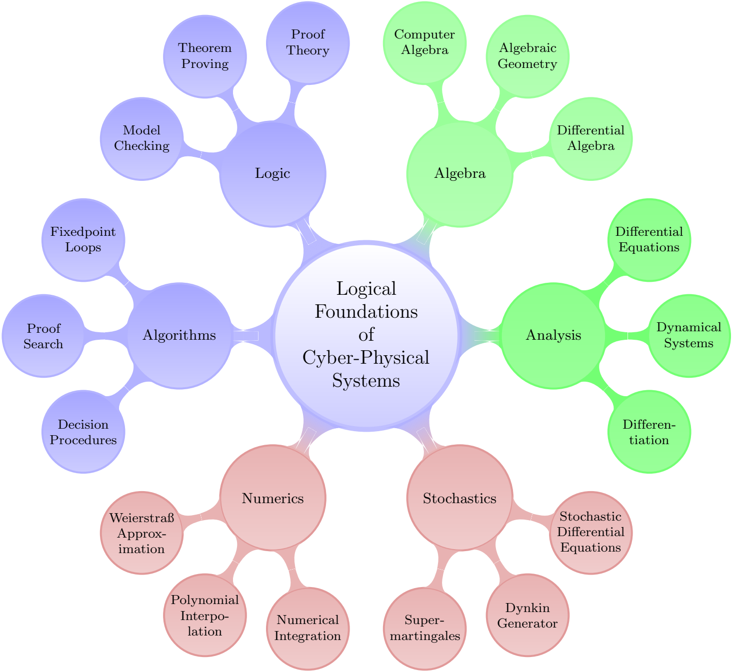
$d\mathcal{L} = \mathrm{DL} + \mathrm{HP}$

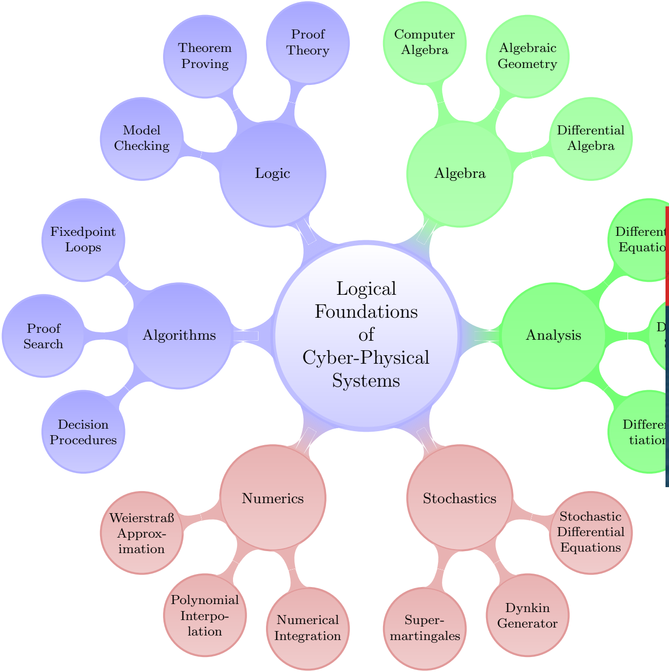
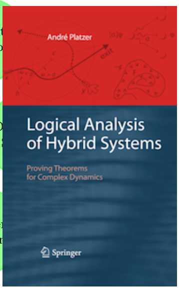


- Logic for hybrid systems
- Logic + distributed hybrid systems
- Logic + stochastic hybrid systems
- Compositional proofs
- Sound & complete / ODE
- Differential invariants

KeYmaera




Logical Foundations of Cyber-Physical Systems
Logic
Model Checking
Theorem Proving
Proof Theory
Algebra
Computer Algebra
Algebraic Geometry
Differential Algebra
Analysis
Differential Equations
Dynamical Systems
Differen-tiation
Stochastics
Stochastic Differential Equations
Dynkin Generator
Super-martingales
Numerics
Weierstraß Approx-imation
Polynomial Interpo-lation
Numerical Integration
Algorithms
Fixedpoint Loops
Proof Search
Decision Procedures


Logical Foundations of Cyber-Physical Systems
Logic
Theorem Proving
Proof Theory
Model Checking
Algebra
Computer Algebra
Algebraic Geometry
Differential Algebra
Algorithms
Fixedpoint Loops
Proof Search
Decision Procedures
Analysis
Numerics
Weierstraß Approx-imation
Polynomial Interpo-lation
Numerical Integration
Stochastics
Stochastic Differential Equations
Super-martingales
Dynkin Generator

André Platzer
Logical Analysis of Hybrid Systems
Proving Theorems for Complex Dynamics
Springer

André Platzer.
Logics of dynamical systems.
In LICS [11], pages 13–24.

André Platzer.
The complete proof theory of hybrid systems.
In LICS [11], pages 541–550.

André Platzer.
Logical Analysis of Hybrid Systems: Proving Theorems for Complex Dynamics.
Springer, Heidelberg, 2010.

André Platzer.
Differential dynamic logic for hybrid systems.
J. Autom. Reas., 41(2):143–189, 2008.

André Platzer.
Differential-algebraic dynamic logic for differential-algebraic programs.
J. Log. Comput., 20(1):309–352, 2010.
Advance Access published on November 18, 2008.

André Platzer and Edmund M. Clarke.
Computing differential invariants of hybrid systems as fixedpoints.
Form. Methods Syst. Des., 35(1):98–120, 2009.
Special issue for selected papers from CAV'08.

André Platzer and Jan-David Quesel.
KeYmaera: A hybrid theorem prover for hybrid systems.
In Alessandro Armando, Peter Baumgartner, and Gilles Dowek, editors, *IJCAR*, volume 5195 of *LNCS*, pages 171–178. Springer, 2008.

André Platzer.
Differential dynamic logic for verifying parametric hybrid systems.
In Nicola Olivetti, editor, *TABLEAUX*, volume 4548 of *LNCS*, pages 216–232. Springer, 2007.

André Platzer.
The structure of differential invariants and differential cut elimination.
Logical Methods in Computer Science, 2012.
To appear.

André Platzer.

A differential operator approach to equational differential invariants.
In Lennart Beringer and Amy Felty, editors, *ITP*, volume 7406 of *LNCS*, pages 28–48. Springer, 2012.

Proceedings of the 27th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2012, June 25–28, 2012, Dubrovnik, Croatia. IEEE Computer Society, 2012.

João P. Hespanha and Ashish Tiwari, editors.
Hybrid Systems: Computation and Control, 9th International Workshop, HSCC 2006, Santa Barbara, CA, USA, March 29-31, 2006, Proceedings, volume 3927 of *LNCS*. Springer, 2006.

André Platzer.
Quantified differential dynamic logic for distributed hybrid systems.
In Anuj Dawar and Helmut Veith, editors, *CSL*, volume 6247 of *LNCS*, pages 469–483. Springer, 2010.

André Platzer.
A complete axiomatization of quantified differential dynamic logic for distributed hybrid systems.

Logical Methods in Computer Science, 2012.
Special issue for selected papers from CSL'10.

André Platzer.
Quantified differential invariants.
In Emilio Frazzoli and Radu Grosu, editors, *HSCC*, pages 63–72. ACM, 2011.

Akash Deshpande, Aleks Göllü, and Pravin Varaiya.
SHIFT: A formalism and a programming language for dynamic networks of hybrid automata.
In Panos J. Antsaklis, Wolf Kohn, Anil Nerode, and Shankar Sastry, editors, *Hybrid Systems*, volume 1273 of *LNCS*, pages 113–133. Springer, 1996.

Fabian Kratz, Oleg Sokolsky, George J. Pappas, and Insup Lee.
R-Charon, a modeling language for reconfigurable hybrid systems.
In Hespanha and Tiwari [12], pages 392–406.

Zhou Chaochen, Wang Ji, and Anders P. Ravn.
A formal description of hybrid systems.

In Rajeev Alur, Thomas A. Henzinger, and Eduardo D. Sontag, editors, *Hybrid Systems*, volume 1066 of *LNCS*, pages 511–530. Springer, 1995.

Pieter J. L. Cuijpers and Michel A. Reniers.
Hybrid process algebra.
J. Log. Algebr. Program., 62(2):191–245, 2005.

D. A. van Beek, Ka L. Man, Michel A. Reniers, J. E. Rooda, and Ramon R. H. Schiffelers.
Syntax and consistent equation semantics of hybrid Chi.
J. Log. Algebr. Program., 68(1-2):129–210, 2006.

William C. Rounds.
A spatial logic for the hybrid π-calculus.
In Rajeev Alur and George J. Pappas, editors, *HSCC*, volume 2993 of *LNCS*, pages 508–522. Springer, 2004.

Jan A. Bergstra and C. A. Middelburg.
Process algebra for hybrid systems.
Theor. Comput. Sci., 335(2-3):215–280, 2005.

José Meseguer and Raman Sharykin.
Specification and analysis of distributed object-based stochastic hybrid systems.
In Hespanha and Tiwari [12], pages 460–475.

André Platzer.
Stochastic differential dynamic logic for stochastic hybrid programs.
In Nikolaj Bjørner and Viorica Sofronie-Stokkermans, editors, *CADE*, volume 6803 of *LNCS*, pages 431–445. Springer, 2011.

André Platzer and Edmund M. Clarke.
The image computation problem in hybrid systems model checking.
In Alberto Bemporad, Antonio Bicchi, and Giorgio Buttazzo, editors, *HSCC*, volume 4416 of *LNCS*, pages 473–486. Springer, 2007.

Outline Background

Outline Background

	Op	Par	T	Cl	Tec	Aut	Cex	Dim	
HenzingerH94, HyTech	✓	×	✓	×	✓	✓	✓		LHA
LafferrierePY99	✓	×	✓	×	✓		✓		forgetful reset
Fränzle99	✓	×	✓	×	✓		✓	×	robust systems
CKrogh03, CheckMate	✓	×	✓	×	✓	✓	✓		polyhedral
Frehse05, PHAVer	✓	×	✓	×	✓	✓	✓	8	LHA (+affine)
MysorePM05	✓	×	✓	×	✓	•	✓	4	bounded prefix
TomlinPS98,MBT05	○	×	×	×	○	○	•	4	HJB numPDE
RatschanS07, HSolver	✓	×		×	✓	✓	×	4	interval
MannaS98, STeP	✓			×	✓	○	×	7	inv↦VCG, flat
ÁbrahámSH01, PVS	•			×	•	○	×	≈9	HA↪PVS, -“-
ZhouRH92, EDC	×	•	✓	..	×	×	×	×	no maths
DavorenN00, Lμ	×	×		✓	○	×	×	×	prop. H-semantics
RönkköRS03, HGC	✓	×	×	×	×	×	×	×	HGC↪HOL
SSManna04	•	○		×	✓		×	4/1	equational system
CTiwari05	•	○		×	✓		×	6/0	linear, -”-
PrajnaJP07, barrier	•	×		×	•		×	3	needs 10000-dim
d$\mathcal{L}$ & dTL	✓	✓	✓	✓	✓	•	×	28	expr., compos.

	Dom	Op	Base	Modal	Quant	Cmpl	Aut
DL	$\mathbb{N}$		$\mathrm{FOL}_{(\mathbb{N})}$		FV+unify	$/\mathbb{N}$	
$d\mathcal{L}$	$\mathbb{R}$	x'	$\mathrm{FOL}_{\mathbb{R}}$	ODE	FV+requant+QE	/ODE	IBC

Outline Background

Proof (Soundness).

- $x' = f(x)$
- Side deductions
- Free variables & Skolemisation

∎

Return

Incompleteness

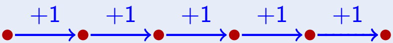
Theorem

Discrete fragment and continuous fragment of $d\mathcal{L}$ *characterize* $\mathbb{N}$

Proof.

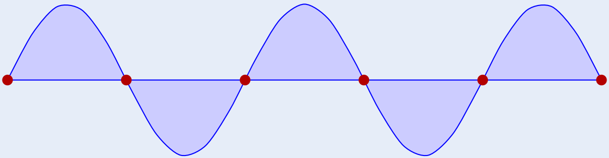
Discrete fragment:

$$\langle (x := x + 1)^* \rangle\ x = n$$



Continuous fragment:

$$\langle s'' = -s, \tau' = 1 \rangle (s = 0 \land \tau = n) \qquad \rightsquigarrow s = \sin$$



□

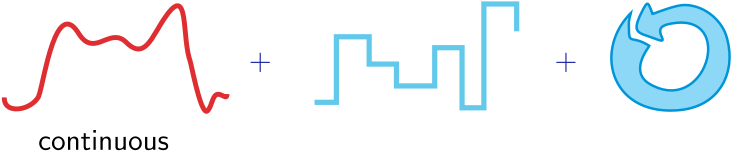
Incomplete! But are we missing proof rules?

Incomplete! But are we missing proof rules?

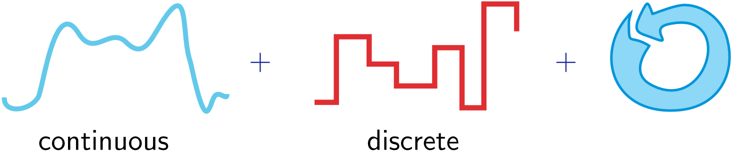
Relativity

Cook,Harel: discrete-DL/$data_{\mathbb{N}}$ hybrid-d$\mathcal{L}$/$data_{\mathbb{R}}$??

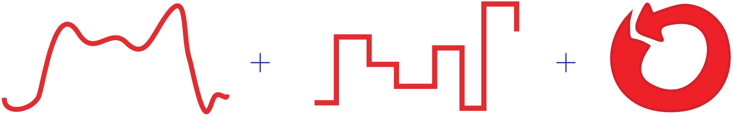
Sources of Incompleteness



Sources of Incompleteness

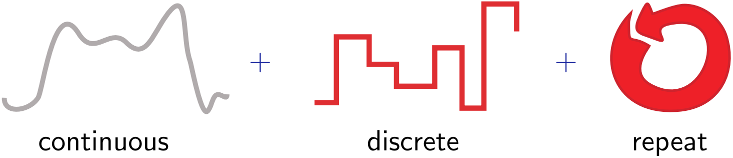


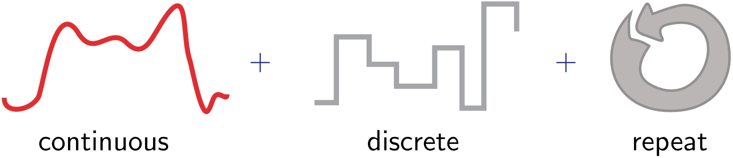
continuous + discrete + repeat



continuous + discrete + repeat

Relative Completeness




continuous
+
discrete
+
repeat

Relative Completeness

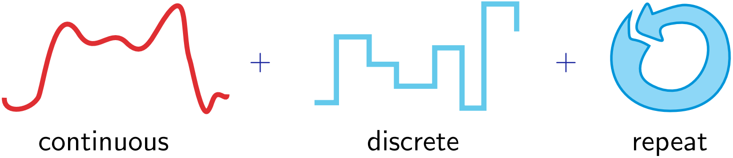
Theorem (Relative Completeness)

d$\mathcal{L}$ calculus is complete relative to first-order logic of differential equations.

$$\vDash \phi \quad \text{iff} \quad Taut_{\mathrm{FOD}} \vdash \phi$$

where $\quad \mathrm{FOD} = \mathrm{FOL}_{\mathbb{R}} + [x_1' = \theta_1, \ldots, x_n' = \theta_n]F$

Proof Outline 15p



$\Downarrow$

Relative Completeness

Theorem (Relative Completeness)

d$\mathcal{L}$ calculus is complete relative to first-order logic of differential equations.

$$\vDash \phi \quad \text{iff} \quad Taut_{\mathrm{FOD}} \vdash \phi$$

where $\mathrm{FOD} = \mathrm{FOL}_{\mathbb{R}} + [x_1' = \theta_1, \ldots, x_n' = \theta_n]F$

Proof Outline 15p

$\Downarrow$

Relativity

Cook,Harel: discrete-DL/data P.: hybrid-d$\mathcal{L}$/differential equations

Theorem (Relative Completeness)

d$\mathcal{L}$ calculus is complete relative to first-order logic of differential equations.

$$\vDash \phi \quad \text{iff} \quad \mathit{Taut}_{\mathrm{FOD}} \vdash \phi$$

where $\mathrm{FOD} = \mathrm{FOL}_{\mathbb{R}} + [x_1' = \theta_1, \ldots, x_n' = \theta_n]F$ Proof Outline 15p

$\Downarrow$

Corollary (Proof-theoretical Alignment)

verification of hybrid systems = verification of dynamical systems!

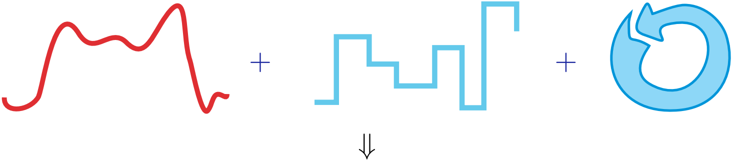
Relative Completeness

Theorem (Relative Completeness)

d$\mathcal{L}$ calculus is complete relative to first-order logic of differential equations.

$$\vDash \phi \quad \text{iff} \quad Taut_{\text{FOD}} \vdash \phi$$

where $\quad \text{FOD} = \text{FOL}_{\mathbb{R}} + [x_1' = \theta_1, \ldots, x_n' = \theta_n]F$ ▸ Proof Outline 15p



Corollary (Deductive Power)

d$\mathcal{L}$ calculus is *supremal hybrid* verification technique

Relative Completeness Proof

$$\vDash \phi \quad \text{iff} \quad Taut_{\text{FOD}} \vdash \phi$$

$$\text{where} \quad \text{FOD} = \text{FOL}_{\mathbb{R}} + [x_1' = \theta_1, \ldots, x_n' = \theta_n]F$$

Proof (Relative Completeness, 10 pages)

Return

1. Strong invariants and variants expressible in $d\mathcal{L}$
2. $d\mathcal{L}$ expressible in FOD
3. valid $d\mathcal{L}$ formulas $d\mathcal{L}$-derivable from corresponding FOD axioms
4. finite FOD formula characterising unbounded hybrid repetition
5. FOD characterises $\mathbb{R}$-Gödel encoding
6. First-order expressible & program rendition: $\forall \phi \ \exists F \in \text{FOD} \quad \vDash \phi \leftrightarrow F$
7. Propositionally & first-order complete
8. Relative complete for first-order safety $F \rightarrow [\alpha]G$
9. Relative complete for first-order liveness $F \rightarrow \langle \alpha \rangle G$

□

Relative Completeness Proof

$$\vDash \phi \quad \text{iff} \quad Taut_{\mathsf{FOD}} \vdash \phi$$

$$\text{where} \quad \mathsf{FOD} = \mathsf{FOL}_{\mathbb{R}} + [x_1' = \theta_1, \dots, x_n' = \theta_n]F$$

Proof (Relative Completeness, 10 pages)

Return

1. Strong invariants and variants expressible in d$\mathcal{L}$
2. d$\mathcal{L}$ expressible in FOD
3. valid d$\mathcal{L}$ formulas d$\mathcal{L}$-derivable from corresponding FOD axioms
4. finite FOD formula characterising unbounded hybrid repetition
5. FOD characterises $\mathbb{R}$-Gödel encoding
6. First-order expressible & program rendition: $\forall \phi \; \exists F \in \mathsf{FOD} \quad \vDash \phi \leftrightarrow F$
7. Propositionally & first-order complete
8. Relative complete for first-order safety $F \rightarrow [\alpha]G$
9. Relative complete for first-order liveness $F \rightarrow \langle\alpha\rangle G$

□

Relative Completeness Proof

$$\vDash \phi \quad \text{iff} \quad Taut_{\mathsf{FOD}} \vdash \phi$$

$$\text{where} \quad \mathsf{FOD} = \mathsf{FOL}_{\mathbb{R}} + [x_1' = \theta_1, \ldots, x_n' = \theta_n]F$$

Proof (Relative Completeness, 10 pages)

Return.

1. Strong invariants and variants expressible in d$\mathcal{L}$
2. d$\mathcal{L}$ expressible in FOD
3. valid d$\mathcal{L}$ formulas d$\mathcal{L}$-derivable from corresponding FOD axioms
4. finite FOD formula characterising unbounded hybrid repetition
5. FOD characterises $\mathbb{R}$-Gödel encoding
6. First-order expressible & program rendition: $\forall\phi \; \exists F \in \mathsf{FOD} \quad \vDash \phi \leftrightarrow F$
7. Propositionally & first-order complete
8. Relative complete for first-order safety $F \rightarrow [\alpha]G$
9. Relative complete for first-order liveness $F \rightarrow \langle\alpha\rangle G$

□

Relative Completeness Proof

$$\vDash \phi \quad \text{iff} \quad Taut_{\mathsf{FOD}} \vdash \phi$$

$$\text{where} \quad \mathsf{FOD} = \mathsf{FOL}_{\mathbb{R}} + [x_1' = \theta_1, \dots, x_n' = \theta_n]F$$

Proof (Relative Completeness, 10 pages)

Return.

1. Strong invariants and variants expressible in d$\mathcal{L}$
2. d$\mathcal{L}$ expressible in FOD
3. valid d$\mathcal{L}$ formulas d$\mathcal{L}$-derivable from corresponding FOD axioms
4. finite FOD formula characterising unbounded hybrid repetition
5. FOD characterises $\mathbb{R}$-Gödel encoding
6. First-order expressible & program rendition: $\forall\phi\ \exists F \in \mathsf{FOD} \quad \vDash \phi \leftrightarrow F$
7. Propositionally & first-order complete
8. Relative complete for first-order safety $F \to [\alpha]G$
9. Relative complete for first-order liveness $F \to \langle\alpha\rangle G$

□

Relative Completeness Proof

$$\vDash \phi \quad \text{iff} \quad Taut_{\mathrm{FOD}} \vdash \phi$$

$$\text{where} \quad \mathrm{FOD} = \mathrm{FOL}_{\mathbb{R}} + [x_1' = \theta_1, \dots, x_n' = \theta_n]F$$

Proof (Relative Completeness, 10 pages)

◂ Return.

1. Strong invariants and variants expressible in d$\mathcal{L}$
2. d$\mathcal{L}$ expressible in FOD
3. valid d$\mathcal{L}$ formulas d$\mathcal{L}$-derivable from corresponding FOD axioms
4. finite FOD formula characterising unbounded hybrid repetition
5. FOD characterises $\mathbb{R}$-Gödel encoding
6. First-order expressible & program rendition: $\forall \phi \; \exists F \in \mathrm{FOD} \quad \vDash \phi \leftrightarrow F$
7. Propositionally & first-order complete
8. Relative complete for first-order safety $F \rightarrow [\alpha]G$
9. Relative complete for first-order liveness $F \rightarrow \langle \alpha \rangle G$

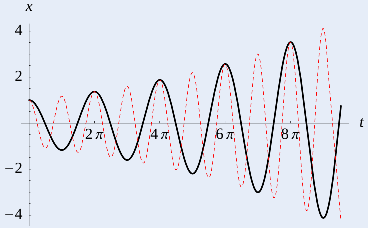
□

where FOD $= \text{FOL}_{\mathbb{R}} + [x_1' = \theta_1, \ldots, x_n' = \theta_n]F$

Proof ($\mathbb{R}$-Gödel encoding)

Return

FOD characterises constructive bijection $\mathbb{R} \to \mathbb{R}^2$

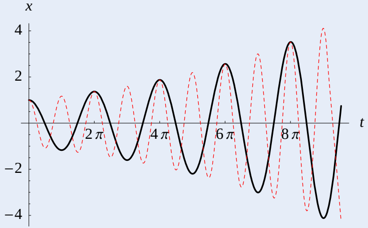


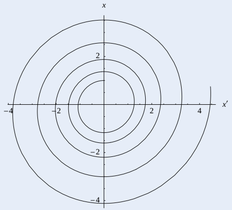
where $\quad \text{FOD} = \text{FOL}_{\mathbb{R}} + [x_1' = \theta_1, \dots, x_n' = \theta_n]F$

Proof ($\mathbb{R}$-Gödel encoding)

◂ Return

FOD characterises constructive bijection $\mathbb{R} \to \mathbb{R}^2$





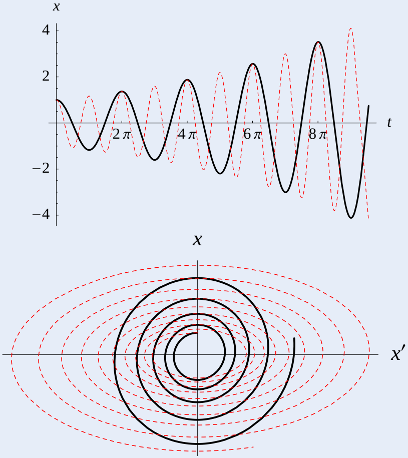
Relative Completeness Proof

where $\quad \text{FOD} = \text{FOL}_{\mathbb{R}} + [x_1' = \theta_1, \ldots, x_n' = \theta_n]F$

Proof ($\mathbb{R}$-Gödel encoding)

◂ Return

FOD characterises constructive bijection $\mathbb{R} \to \mathbb{R}^2$

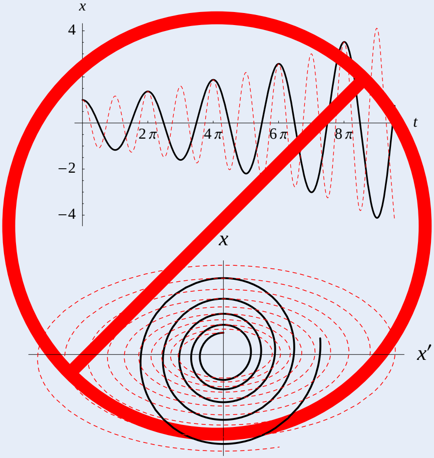


where $\text{FOD} = \text{FOL}_{\mathbb{R}} + [x_1' = \theta_1, \ldots, x_n' = \theta_n]F$

Proof ($\mathbb{R}$-Gödel encoding)

Return.

FOD characterises constructive bijection $\mathbb{R} \to \mathbb{R}^2$ not differentiable!



Relative Completeness Proof

where $\quad \text{FOD} = \text{FOL}_{\mathbb{R}} + [x_1' = \theta_1, \ldots, x_n' = \theta_n]F$

Proof ($\mathbb{R}$-Gödel encoding)

Return

FOD characterises constructive bijection $\mathbb{R} \to \mathbb{R}^2$

$$\sum_{i=1}^{\infty} \frac{a_i}{2^i} = 0.a_1 a_2 \ldots$$

$$\sum_{i=1}^{\infty} \frac{b_i}{2^i} = 0.b_1 b_2 \ldots$$

$$\sum_{i=0}^{\infty} \left(\frac{a_i}{2^{2i+1}} + \frac{b_i}{2^{2i+2}} \right) = 0.a_1 b_1 a_2 b_2 \ldots$$

$\square$

where $\quad \mathrm{FOD} = \mathrm{FOL}_{\mathbb{R}} + [x_1' = \theta_1, \ldots, x_n' = \theta_n]F$

Proof ($\mathbb{R}$-Gödel encoding)

Return

FOD characterises constructive bijection $\mathbb{R} \to \mathbb{R}^2$

$$\sum_{i=1}^{\infty} \frac{a_i}{2^i} = 0.a_1 a_2 \ldots$$

$$\sum_{i=1}^{\infty} \frac{b_i}{2^i} = 0.b_1 b_2 \ldots$$

$$\sum_{i=0}^{\infty} \left(\frac{a_i}{2^{2i+1}} + \frac{b_i}{2^{2i+2}} \right) = 0.a_1 b_1 a_2 b_2 \ldots$$

$$2^n = z \quad \leftrightarrow \quad \langle x := 1; \tau := 0; x' = x \ln 2 \land \tau' = 1 \rangle (\tau = n \land x = z)$$

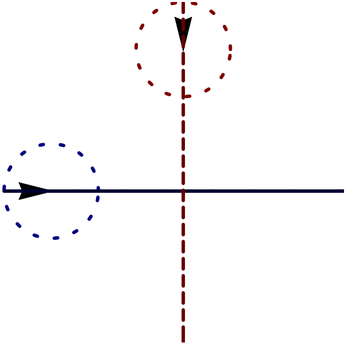
$$\ln 2 = z \quad \leftrightarrow \quad \langle x := 1; \tau := 0; x' = x \land \tau' = 1 \rangle (x = 2 \land \tau = z)$$

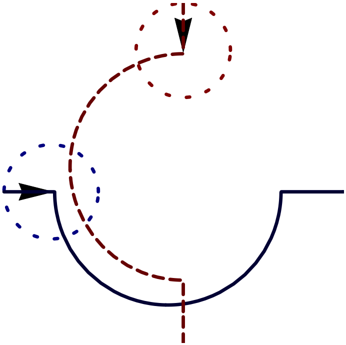
□

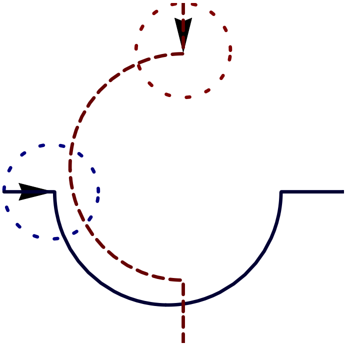
Outline Background

Outline Background

Air Traffic Control

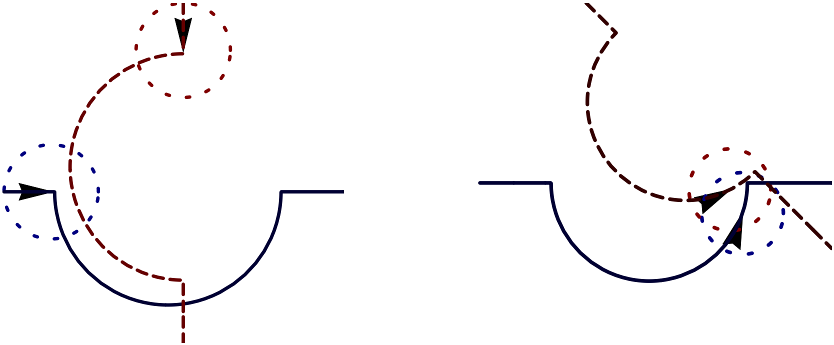






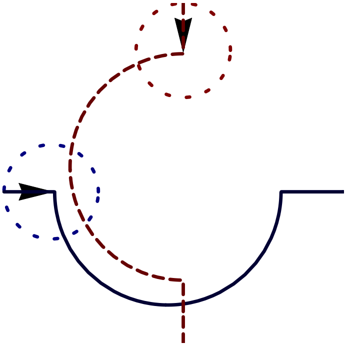
Verification?

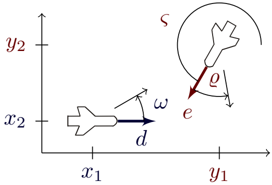
looks correct



Verification?

looks correct NO!



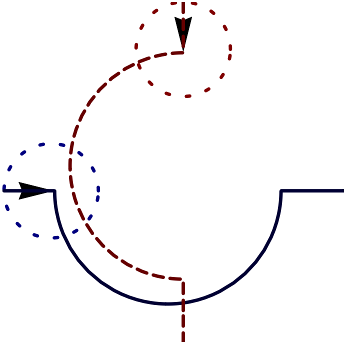


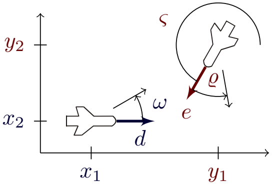
$$\begin{bmatrix} x_1' = -v_1 + v_2 \cos\vartheta + \omega x_2 \\ x_2' = v_2 \sin\vartheta - \omega x_1 \\ \vartheta' = \varpi - \omega \end{bmatrix}$$

Verification?

looks correct NO!

Air Traffic Control



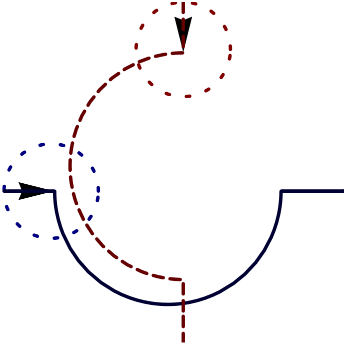


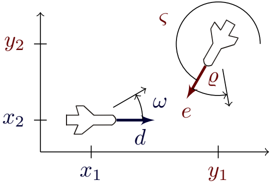
$$\begin{bmatrix} x_1' = -v_1 + v_2 \cos\vartheta + \omega x_2 \\ x_2' = v_2 \sin\vartheta - \omega x_1 \\ \vartheta' = \varpi - \omega \end{bmatrix}$$

Example ("Solving" differential equations)

$$x_1(t) = \frac{1}{\omega\varpi}\big(x_1\omega\varpi \cos t\omega - v_2\omega \cos t\omega \sin\vartheta + v_2\omega \cos t\omega \cos t\varpi \sin\vartheta - v_1\varpi \sin t\omega$$
$$+ x_2\omega\varpi \sin t\omega - v_2\omega \cos\vartheta \cos t\varpi \sin t\omega - v_2\omega\sqrt{1 - \sin\vartheta^2} \sin t\omega$$
$$+ v_2\omega \cos\vartheta \cos t\omega \sin t\varpi + v_2\omega \sin\vartheta \sin t\omega \sin t\varpi\big) \ldots$$

Air Traffic Control





$$\begin{bmatrix} x_1' = -v_1 + v_2 \cos\vartheta + \omega x_2 \\ x_2' = v_2 \sin\vartheta - \omega x_1 \\ \vartheta' = \varpi - \omega \end{bmatrix}$$

Example ("Solving" differential equations)

$$\forall t \geq 0 \quad \frac{1}{\omega\varpi}\Big(x_1\omega\varpi\cos t\omega - v_2\omega\cos t\omega\sin\vartheta + v_2\omega\cos t\omega\cos t\varpi\sin\vartheta - v_1\varpi\sin t\omega$$
$$+ x_2\omega\varpi\sin t\omega - v_2\omega\cos\vartheta\cos t\varpi\sin t\omega - v_2\omega\sqrt{1 - \sin\vartheta^2}\sin t\omega$$
$$+ v_2\omega\cos\vartheta\cos t\omega\sin t\varpi + v_2\omega\sin\vartheta\sin t\omega\sin t\varpi\Big)\ldots$$

```
\forall R ts2.
  (   0 <= ts2 & ts2 <= t2_0
   ->       (  (om_1)^-1
             * (omb_1)^-1
             * (  om_1 * omb_1 * x1 * Cos(om_1 * ts2)
                + om_1 * v2 * Cos(om_1 * ts2) * (1 + -1 * (Cos(u))^2)^(1 / 2)
                + -1 * omb_1 * v1 * Sin(om_1 * ts2)
                + om_1 * omb_1 * x2 * Sin(om_1 * ts2)
                + om_1 * v2 * Cos(u) * Sin(om_1 * ts2)
                + -1 * om_1 * v2 * Cos(omb_1 * ts2) * Cos(u) * Sin(om_1 * ts2)
                + om_1 * v2 * Cos(om_1 * ts2) * Cos(u) * Sin(omb_1 * ts2)
                + om_1 * v2 * Cos(om_1 * ts2) * Cos(omb_1 * ts2) * Sin(u)
                + om_1 * v2 * Sin(om_1 * ts2) * Sin(omb_1 * ts2) * Sin(u)))
         ^2
       +    (  (om_1)^-1
             * (omb_1)^-1
             * (  -1 * omb_1 * v1 * Cos(om_1 * ts2)
                + om_1 * omb_1 * x2 * Cos(om_1 * ts2)
                + omb_1 * v1 * (Cos(om_1 * ts2))^2
                + om_1 * v2 * Cos(om_1 * ts2) * Cos(u)
                + -1 * om_1 * v2 * Cos(om_1 * ts2) * Cos(omb_1 * ts2) * Cos(u)
                + -1 * om_1 * omb_1 * x1 * Sin(om_1 * ts2)
                +   -1
                  * om_1
                  * v2
                  * (1 + -1 * (Cos(u))^2)^(1 / 2)
                  * Sin(om_1 * ts2)
                + omb_1 * v1 * (Sin(om_1 * ts2))^2
                + -1 * om_1 * v2 * Cos(u) * Sin(om_1 * ts2) * Sin(omb_1 * ts2)
                + -1 * om_1 * v2 * Cos(omb_1 * ts2) * Sin(om_1 * ts2) * Sin(u)
                + om_1 * v2 * Cos(om_1 * ts2) * Sin(omb_1 * ts2) * Sin(u)))
         ^2
      >= (p)^2),
 t2_0 >= 0,
 x1^2 + x2^2 >= (p)^2
==>
```

```
\forall R t7.
  (   t7 >= 0
   ->       (  (om_3)^-1
             * (    om_3
                 * (  (om_1)^-1
                    * (omb_1)^-1
                    * (  om_1 * omb_1 * x1 * Cos(om_1 * t2_0)
                       +   om_1
                         * v2
                         * Cos(om_1 * t2_0)
                         * (1 + -1 * (Cos(u))^2)^(1 / 2)
                       + -1 * omb_1 * v1 * Sin(om_1 * t2_0)
                       + om_1 * omb_1 * x2 * Sin(om_1 * t2_0)
                       + om_1 * v2 * Cos(u) * Sin(om_1 * t2_0)
                       +   -1
                         * om_1
                         * v2
                         * Cos(omb_1 * t2_0)
                         * Cos(u)
                         * Sin(om_1 * t2_0)
                       +   om_1
                         * v2
                         * Cos(om_1 * t2_0)
                         * Cos(u)
                         * Sin(omb_1 * t2_0)
                       +   om_1
                         * v2
                         * Cos(om_1 * t2_0)
                         * Cos(omb_1 * t2_0)
                         * Sin(u)
                       +   om_1
                         * v2
                         * Sin(om_1 * t2_0)
                         * Sin(omb_1 * t2_0)
                         * Sin(u)))
```

```
    * Cos(om_3 * t5)
+     v2
    * Cos(om_3 * t5)
    *   (  1
         +   -1
           * (Cos(-1 * om_1 * t2_0 + omb_1 * t2_0 + u + Pi / 4))^2)
     ^(1 / 2)
+ -1 * v1 * Sin(om_3 * t5)
+   om_3
  * (  (om_1)^-1
     * (omb_1)^-1
     * (  -1 * omb_1 * v1 * Cos(om_1 * t2_0)
        + om_1 * omb_1 * x2 * Cos(om_1 * t2_0)
        + omb_1 * v1 * (Cos(om_1 * t2_0))^2
        + om_1 * v2 * Cos(om_1 * t2_0) * Cos(u)
        +   -1
          * om_1
          * v2
          * Cos(om_1 * t2_0)
          * Cos(omb_1 * t2_0)
          * Cos(u)
        + -1 * om_1 * omb_1 * x1 * Sin(om_1 * t2_0)
        +   -1
          * om_1
          * v2
          * (1 + -1 * (Cos(u))^2)^(1 / 2)
          * Sin(om_1 * t2_0)
        + omb_1 * v1 * (Sin(om_1 * t2_0))^2
        +   -1
          * om_1
          * v2
          * Cos(u)
          * Sin(om_1 * t2_0)
          * Sin(omb_1 * t2_0)
```

```
                + -1
                  * om_1
                  * v2
                  * Cos(omb_1 * t2_0)
                  * Sin(om_1 * t2_0)
                  * Sin(u)
                + om_1
                  * v2
                  * Cos(om_1 * t2_0)
                  * Sin(omb_1 * t2_0)
                  * Sin(u)))
          * Sin(om_3 * t5)
        + v2
          * Cos(-1 * om_1 * t2_0 + omb_1 * t2_0 + u + Pi / 4)
          * Sin(om_3 * t5)
        + v2
          * (Cos(om_3 * t5))^2
          * Sin(-1 * om_1 * t2_0 + omb_1 * t2_0 + u + Pi / 4)
        + v2
          * (Sin(om_3 * t5))^2
          * Sin(-1 * om_1 * t2_0 + omb_1 * t2_0 + u + Pi / 4)))
 ^2
+ ( (om_3)^-1
    * ( -1 * v1 * Cos(om_3 * t5)
        + om_3
          * ( (om_1)^-1
             * (omb_1)^-1
             * ( -1 * omb_1 * v1 * Cos(om_1 * t2_0)
                + om_1 * omb_1 * x2 * Cos(om_1 * t2_0)
                + omb_1 * v1 * (Cos(om_1 * t2_0))^2
                + om_1 * v2 * Cos(om_1 * t2_0) * Cos(u)
                + -1
                  * om_1
                  * v2
                  * Cos(om_1 * t2_0)
                  * Cos(omb_1 * t2_0)
```

```
        + -1 * om_1 * omb_1 * x1 * Sin(om_1 * t2_0)
        +   -1
          * om_1
          * v2
          * (1 + -1 * (Cos(u))^2)^(1 / 2)
          * Sin(om_1 * t2_0)
        + omb_1 * v1 * (Sin(om_1 * t2_0))^2
        +   -1
          * om_1
          * v2
          * Cos(u)
          * Sin(om_1 * t2_0)
          * Sin(omb_1 * t2_0)
        +   -1
          * om_1
          * v2
          * Cos(omb_1 * t2_0)
          * Sin(om_1 * t2_0)
          * Sin(u)
        +   om_1
          * v2
          * Cos(om_1 * t2_0)
          * Sin(omb_1 * t2_0)
          * Sin(u)))
  * Cos(om_3 * t5)
+ v1 * (Cos(om_3 * t5))^2
+   v2
  * Cos(om_3 * t5)
  * Cos(-1 * om_1 * t2_0 + omb_1 * t2_0 + u + Pi / 4)
+   -1
  * v2
  * (Cos(om_3 * t5))^2
  * Cos(-1 * om_1 * t2_0 + omb_1 * t2_0 + u + Pi / 4)
```

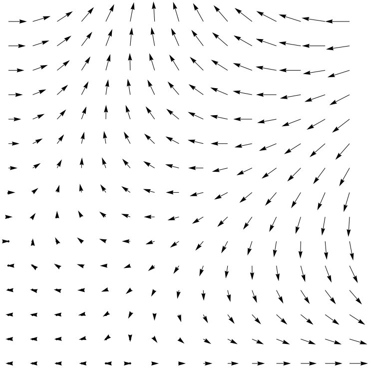
```
+   -1
  * om_3
  * (  (om_1)^-1
     * (omb_1)^-1
     * (  om_1 * omb_1 * x1 * Cos(om_1 * t2_0)
        +   om_1
          * v2
          * Cos(om_1 * t2_0)
          * (1 + -1 * (Cos(u))^2)^(1 / 2)
        + -1 * omb_1 * v1 * Sin(om_1 * t2_0)
        + om_1 * omb_1 * x2 * Sin(om_1 * t2_0)
        + om_1 * v2 * Cos(u) * Sin(om_1 * t2_0)
        +   -1
          * om_1
          * v2
          * Cos(omb_1 * t2_0)
          * Cos(u)
          * Sin(om_1 * t2_0)
        +   om_1
          * v2
          * Cos(om_1 * t2_0)
          * Cos(u)
          * Sin(omb_1 * t2_0)
        +   om_1
          * v2
          * Cos(om_1 * t2_0)
          * Cos(omb_1 * t2_0)
          * Sin(u)
        +   om_1
          * v2
          * Sin(om_1 * t2_0)
          * Sin(omb_1 * t2_0)
          * Sin(u)))
  * Sin(om_3 * t5)
```

```
        +   -1
          * v2
          *   (  1
                +   -1
                  * (Cos(-1 * om_1 * t2_0 + omb_1 * t2_0 + u + Pi / 4))^2)
           ^(1 / 2)
          * Sin(om_3 * t5)
        + v1 * (Sin(om_3 * t5))^2
        +   -1
          * v2
          * Cos(-1 * om_1 * t2_0 + omb_1 * t2_0 + u + Pi / 4)
          * (Sin(om_3 * t5))^2))
  ^2
>= (p)^2)
```

This is just one branch to prove for aircraft ...

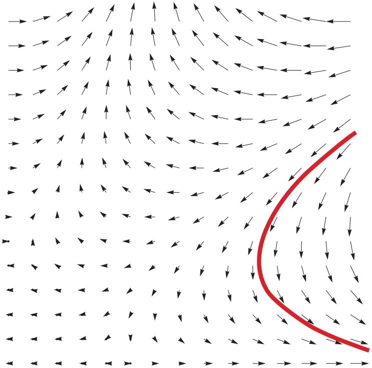
"Definition" (Differential Invariant)

"Formula that remains true in the direction of the dynamics"



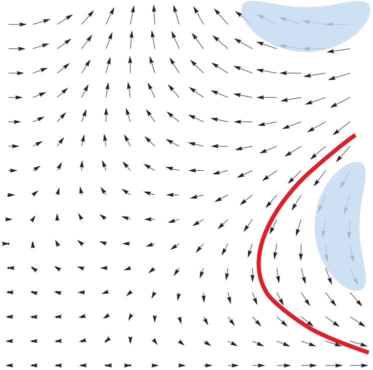
"Definition" (Differential Invariant)

"Formula that remains true in the direction of the dynamics"



"Definition" (Differential Invariant)

"Formula that remains true in the direction of the dynamics"



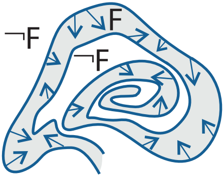
Differential Induction: Local Dynamics w/o Solutions

Definition (Differential Invariant) (J.Log.Comput. 2010)

F closed under total differentiation with respect to differential constraints

Definition (Differential Invariant) (J.Log.Comput. 2010)

F closed under total differentiation with respect to differential constraints



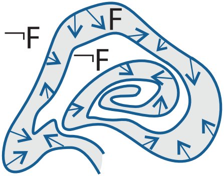
$$\frac{(\chi \rightarrow F')}{\chi \rightarrow F \rightarrow [x' = \theta \,\&\, \chi]F}$$

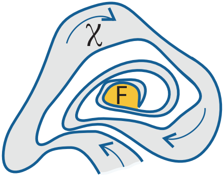
$$\frac{F \rightarrow [\alpha]F}{F \rightarrow [\alpha^*]F}$$

Differential Induction: Local Dynamics w/o Solutions

Definition (Differential Invariant) (J.Log.Comput. 2010)

F closed under total differentiation with respect to differential constraints

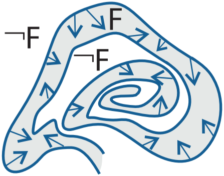


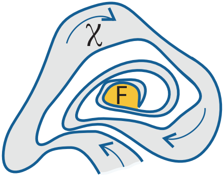


$$\frac{(\chi \rightarrow F')}{\chi \rightarrow F \rightarrow [x' = \theta \,\&\, \chi]F}$$

Definition (Differential Invariant) (J.Log.Comput. 2010)

F closed under total differentiation with respect to differential constraints



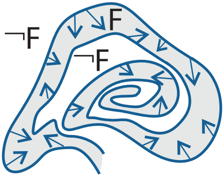


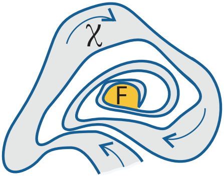
$$\frac{(\chi \to F')}{\chi \to F \to [x' = \theta \,\&\, \chi]F}$$

$$\frac{(\neg F \land \chi \to F'_{\gg})}{[x' = \theta \,\&\, \neg F]\chi \to \langle x' = \theta \,\&\, \chi\rangle F}$$

Definition (Differential Invariant) (J.Log.Comput. 2010)

F closed under total differentiation with respect to differential constraints





$$\frac{(\chi \to F')}{\chi \to F \to [x' = \theta \,\&\, \chi]F}$$

$$\frac{(\neg F \wedge \chi \to F'_{\gg})}{[x' = \theta \,\&\, \neg F]\chi \to \langle x' = \theta \,\&\, \chi\rangle F}$$

Total differential F' of *formulas*?

Ex: Nonlinear Dynamics

$$\frac{}{x^3 \geq -1 \rightarrow [x' = (x-3)^4 + a \,\&\, a \geq 0] x^3 \geq -1}$$

$$\dfrac{\overline{\quad a \geq 0 \rightarrow 2x^2 x' \geq 0 \quad}}{x^3 \geq -1 \rightarrow [x' = (x-3)^4 + a \,\&\, a \geq 0] x^3 \geq -1}$$

Ex: Nonlinear Dynamics

$$\dfrac{\dfrac{\overline{a \geq 0 \rightarrow 2x^2((x-3)^4 + a) \geq 0}}{a \geq 0 \rightarrow 2x^2 x' \geq 0}}{x^3 \geq -1 \rightarrow [x' = (x-3)^4 + a \,\&\, a \geq 0]x^3 \geq -1}$$

$$\dfrac{\dfrac{\dfrac{*}{a \geq 0 \rightarrow 2x^2((x-3)^4 + a) \geq 0}}{a \geq 0 \rightarrow 2x^2 x' \geq 0}}{x^3 \geq -1 \rightarrow [x' = (x-3)^4 + a \,\&\, a \geq 0] x^3 \geq -1}$$

Ex: Riccati

$$\overline{2x^3 \geq \frac{1}{4} \rightarrow [x' = x^2 + x^4] 2x^3 \geq \frac{1}{4}}$$

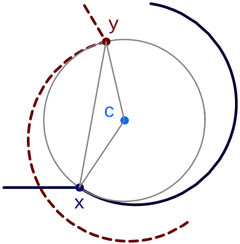
Ex: Riccati

$$\frac{\overline{\quad 6x^2x' \geq 0 \quad}}{2x^3 \geq \frac{1}{4} \rightarrow [x' = x^2 + x^4] 2x^3 \geq \frac{1}{4}}$$

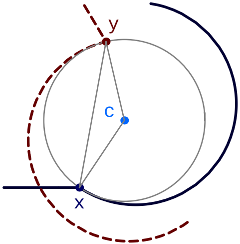
$$\dfrac{\dfrac{\overline{6x^2(x^2+x^4) \geq 0}}{6x^2 x' \geq 0}}{2x^3 \geq \frac{1}{4} \rightarrow [x' = x^2 + x^4] 2x^3 \geq \frac{1}{4}}$$

$$\dfrac{\dfrac{\dfrac{*}{6x^2(x^2+x^4)\geq 0}}{6x^2x'\geq 0}}{2x^3\geq \frac{1}{4}\rightarrow [x'=x^2+x^4]2x^3\geq \frac{1}{4}}$$

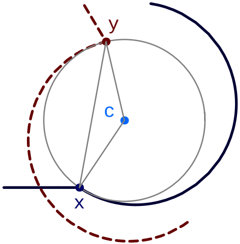
$$\overline{\qquad\qquad}$$
$$[x_1' = d_1, d_1' = -\omega d_2, x_2' = d_2, d_2' = \omega d_1, ..](x_1 - y_1)^2 + (x_2 - y_2)^2 \geq p^2$$



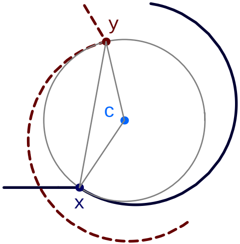
$$\frac{\frac{\partial\|x-y\|^2}{\partial x_1}x_1' + \frac{\partial\|x-y\|^2}{\partial y_1}y_1' + \frac{\partial\|x-y\|^2}{\partial x_2}x_2' + \frac{\partial\|x-y\|^2}{\partial y_2}y_2' \geq \frac{\partial p^2}{\partial x_1}x_1' \ldots}{[x_1' = d_1, d_1' = -\omega d_2, x_2' = d_2, d_2' = \omega d_1, ..](x_1 - y_1)^2 + (x_2 - y_2)^2 \geq p^2}$$



$$\frac{\frac{\partial\|x-y\|^2}{\partial x_1}x_1' + \frac{\partial\|x-y\|^2}{\partial y_1}y_1' + \frac{\partial\|x-y\|^2}{\partial x_2}x_2' + \frac{\partial\|x-y\|^2}{\partial y_2}y_2' \geq \frac{\partial p^2}{\partial x_1}x_1' \dots}{[x_1' = d_1, d_1' = -\omega d_2, x_2' = d_2, d_2' = \omega d_1, ..](x_1-y_1)^2 + (x_2-y_2)^2 \geq p^2}$$

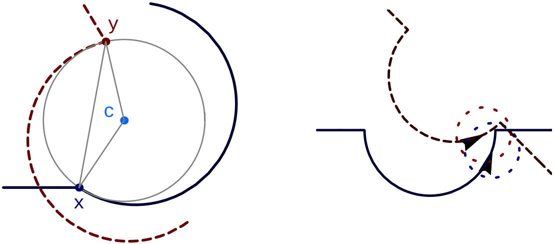


$$\frac{\dfrac{\partial\|x-y\|^2}{\partial x_1}d_1 + \dfrac{\partial\|x-y\|^2}{\partial y_1}e_1 + \dfrac{\partial\|x-y\|^2}{\partial x_2}d_2 + \dfrac{\partial\|x-y\|^2}{\partial y_2}e_2 \geq \dfrac{\partial p^2}{\partial x_1}d_1 \ldots}{[x_1' = d_1, d_1' = -\omega d_2, x_2' = d_2, d_2' = \omega d_1, ..](x_1 - y_1)^2 + (x_2 - y_2)^2 \geq p^2}$$



$$\frac{2(x_1 - y_1)(d_1 - e_1) + 2(x_2 - y_2)(d_2 - e_2) \geq 0}{\dfrac{\partial\|x-y\|^2}{\partial x_1} d_1 + \dfrac{\partial\|x-y\|^2}{\partial y_1} e_1 + \dfrac{\partial\|x-y\|^2}{\partial x_2} d_2 + \dfrac{\partial\|x-y\|^2}{\partial y_2} e_2 \geq \dfrac{\partial p^2}{\partial x_1} d_1 \ldots}$$

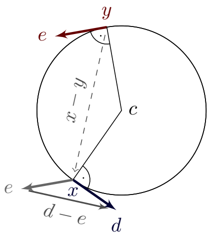
$$[x_1' = d_1, d_1' = -\omega d_2, x_2' = d_2, d_2' = \omega d_1, ..](x_1 - y_1)^2 + (x_2 - y_2)^2 \geq p^2$$

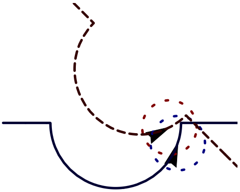


$$2(x_1 - y_1)(d_1 - e_1) + 2(x_2 - y_2)(d_2 - e_2) \geq 0$$

$$\frac{\partial\|x-y\|^2}{\partial x_1} d_1 + \frac{\partial\|x-y\|^2}{\partial y_1} e_1 + \frac{\partial\|x-y\|^2}{\partial x_2} d_2 + \frac{\partial\|x-y\|^2}{\partial y_2} e_2 \geq \frac{\partial p^2}{\partial x_1} d_1 \dots$$

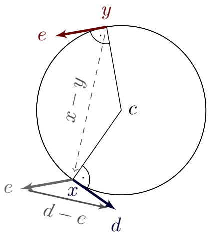
$$[x_1' = d_1, d_1' = -\omega d_2, x_2' = d_2, d_2' = \omega d_1, ..](x_1 - y_1)^2 + (x_2 - y_2)^2 \geq p^2$$

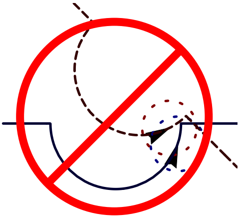




$$\frac{2(x_1-y_1)(d_1-e_1)+2(x_2-y_2)(d_2-e_2)\geq 0}{\frac{\partial\|x-y\|^2}{\partial x_1}d_1+\frac{\partial\|x-y\|^2}{\partial y_1}e_1+\frac{\partial\|x-y\|^2}{\partial x_2}d_2+\frac{\partial\|x-y\|^2}{\partial y_2}e_2\geq\frac{\partial p^2}{\partial x_1}d_1\dots}$$

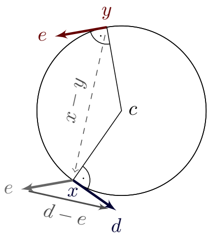
$$[x_1'=d_1,d_1'=-\omega d_2,x_2'=d_2,d_2'=\omega d_1,..](x_1-y_1)^2+(x_2-y_2)^2\geq p^2$$





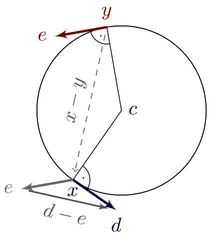
$$..\rightarrow[d_1'=-\omega d_2,e_1'=-\omega e_2,x_2'=d_2,d_2'=\omega d_1,..]d_1-e_1=-\omega(x_2-y_2)$$

$$\cfrac{\cfrac{\cfrac{2(x_1-y_1)(-\omega(x_2-y_2))+2(x_2-y_2)\omega(x_1-y_1)\geq 0}{2(x_1-y_1)(d_1-e_1)+2(x_2-y_2)(d_2-e_2)\geq 0}}{\frac{\partial\|x-y\|^2}{\partial x_1}d_1+\frac{\partial\|x-y\|^2}{\partial y_1}e_1+\frac{\partial\|x-y\|^2}{\partial x_2}d_2+\frac{\partial\|x-y\|^2}{\partial y_2}e_2\geq\frac{\partial p^2}{\partial x_1}d_1\ldots}}{[x_1'=d_1,d_1'=-\omega d_2,x_2'=d_2,d_2'=\omega d_1,..](x_1-y_1)^2+(x_2-y_2)^2\geq p^2}$$



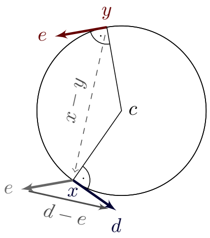
$$\overline{..\to[d_1'=-\omega d_2,e_1'=-\omega e_2,x_2'=d_2,d_2'=\omega d_1,..]d_1-e_1=-\omega(x_2-y_2)}$$

$$\frac{\dfrac{\dfrac{2(x_1-y_1)(-\omega(x_2-y_2))+2(x_2-y_2)\omega(x_1-y_1)\geq 0}{2(x_1-y_1)(d_1-e_1)+2(x_2-y_2)(d_2-e_2)\geq 0}}{\frac{\partial\|x-y\|^2}{\partial x_1}d_1+\frac{\partial\|x-y\|^2}{\partial y_1}e_1+\frac{\partial\|x-y\|^2}{\partial x_2}d_2+\frac{\partial\|x-y\|^2}{\partial y_2}e_2\geq\frac{\partial p^2}{\partial x_1}d_1\ldots}}{[x_1'=d_1,d_1'=-\omega d_2,x_2'=d_2,d_2'=\omega d_1,..](x_1-y_1)^2+(x_2-y_2)^2\geq p^2}$$



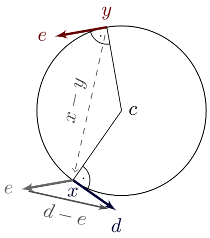
$$\frac{\dfrac{}{\frac{\partial(d_1-e_1)}{\partial d_1}d_1'+\frac{\partial(d_1-e_1)}{\partial e_1}e_1'=-\frac{\partial\omega(x_2-y_2)}{\partial x_2}x_2'-\frac{\partial\omega(x_2-y_2)}{\partial y_2}y_2'}}{..\rightarrow[d_1'=-\omega d_2,e_1'=-\omega e_2,x_2'=d_2,d_2'=\omega d_1,..]\color{red}{d_1-e_1=-\omega(x_2-y_2)}}$$

$$\frac{\dfrac{\dfrac{2(x_1-y_1)(-\omega(x_2-y_2))+2(x_2-y_2)\omega(x_1-y_1)\geq 0}{2(x_1-y_1)(d_1-e_1)+2(x_2-y_2)(d_2-e_2)\geq 0}}{\frac{\partial\|x-y\|^2}{\partial x_1}d_1+\frac{\partial\|x-y\|^2}{\partial y_1}e_1+\frac{\partial\|x-y\|^2}{\partial x_2}d_2+\frac{\partial\|x-y\|^2}{\partial y_2}e_2\geq\frac{\partial p^2}{\partial x_1}d_1\ldots}}{[x_1'=d_1,d_1'=-\omega d_2,x_2'=d_2,d_2'=\omega d_1,..](x_1-y_1)^2+(x_2-y_2)^2\geq p^2}$$



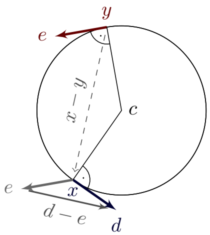
$$\frac{\dfrac{}{\frac{\partial(d_1-e_1)}{\partial d_1}d_1'+\frac{\partial(d_1-e_1)}{\partial e_1}e_1'=-\frac{\partial\omega(x_2-y_2)}{\partial x_2}x_2'-\frac{\partial\omega(x_2-y_2)}{\partial y_2}y_2'}}{..\to[d_1'=-\omega d_2,e_1'=-\omega e_2,x_2'=d_2,d_2'=\omega d_1,..]d_1-e_1=-\omega(x_2-y_2)}$$

$$\frac{\dfrac{\dfrac{2(x_1-y_1)(-\omega(x_2-y_2))+2(x_2-y_2)\omega(x_1-y_1)\geq 0}{2(x_1-y_1)(d_1-e_1)+2(x_2-y_2)(d_2-e_2)\geq 0}}{\frac{\partial\|x-y\|^2}{\partial x_1}d_1+\frac{\partial\|x-y\|^2}{\partial y_1}e_1+\frac{\partial\|x-y\|^2}{\partial x_2}d_2+\frac{\partial\|x-y\|^2}{\partial y_2}e_2\geq\frac{\partial p^2}{\partial x_1}d_1\dots}}{[x_1'=d_1,d_1'=-\omega d_2,x_2'=d_2,d_2'=\omega d_1,..](x_1-y_1)^2+(x_2-y_2)^2\geq p^2}$$



$$\frac{\frac{\partial(d_1-e_1)}{\partial d_1}(-\omega d_2)+\frac{\partial(d_1-e_1)}{\partial e_1}(-\omega e_2)=-\frac{\partial\omega(x_2-y_2)}{\partial x_2}d_2-\frac{\partial\omega(x_2-y_2)}{\partial y_2}e_2}{..\rightarrow[d_1'=-\omega d_2,e_1'=-\omega e_2,x_2'=d_2,d_2'=\omega d_1,..]d_1-e_1=-\omega(x_2-y_2)}$$

$$\frac{\dfrac{\dfrac{2(x_1-y_1)(-\omega(x_2-y_2))+2(x_2-y_2)\omega(x_1-y_1)\geq 0}{2(x_1-y_1)(d_1-e_1)+2(x_2-y_2)(d_2-e_2)\geq 0}}{\frac{\partial\|x-y\|^2}{\partial x_1}d_1+\frac{\partial\|x-y\|^2}{\partial y_1}e_1+\frac{\partial\|x-y\|^2}{\partial x_2}d_2+\frac{\partial\|x-y\|^2}{\partial y_2}e_2\geq\frac{\partial p^2}{\partial x_1}d_1\ldots}}{[x_1'=d_1,d_1'=-\omega d_2,x_2'=d_2,d_2'=\omega d_1,..](x_1-y_1)^2+(x_2-y_2)^2\geq p^2}$$



$$\frac{\dfrac{-\omega d_2+\omega e_2=-\omega(d_2-e_2)}{\frac{\partial(d_1-e_1)}{\partial d_1}(-\omega d_2)+\frac{\partial(d_1-e_1)}{\partial e_1}(-\omega e_2)=-\frac{\partial\omega(x_2-y_2)}{\partial x_2}d_2-\frac{\partial\omega(x_2-y_2)}{\partial y_2}e_2}}{..\rightarrow[d_1'=-\omega d_2,e_1'=-\omega e_2,x_2'=d_2,d_2'=\omega d_1,..]d_1-e_1=-\omega(x_2-y_2)}$$

$$\cfrac{\cfrac{\cfrac{2(x_1-y_1)(-\omega(x_2-y_2))+2(x_2-y_2)\omega(x_1-y_1)\geq 0}{2(x_1-y_1)(d_1-e_1)+2(x_2-y_2)(d_2-e_2)\geq 0}}{\frac{\partial\|x-y\|^2}{\partial x_1}d_1+\frac{\partial\|x-y\|^2}{\partial y_1}e_1+\frac{\partial\|x-y\|^2}{\partial x_2}d_2+\frac{\partial\|x-y\|^2}{\partial y_2}e_2\geq\frac{\partial p^2}{\partial x_1}d_1\ldots}}{[x_1'=d_1,d_1'=-\omega d_2,x_2'=d_2,d_2'=\omega d_1,..](x_1-y_1)^2+(x_2-y_2)^2\geq p^2}$$

Proposition (Differential cut saturation)

F differential invariant of $[x'=\theta\,\&\,H]\phi$, then

$$[x'=\theta\,\&\,H]\phi \quad \text{iff} \quad [x'=\theta\,\&\,H\wedge F]\phi$$

$$\cfrac{\cfrac{-\omega d_2+\omega e_2=-\omega(d_2-e_2)}{\frac{\partial(d_1-e_1)}{\partial d_1}(-\omega d_2)+\frac{\partial(d_1-e_1)}{\partial e_1}(-\omega e_2)=-\frac{\partial\omega(x_2-y_2)}{\partial x_2}d_2-\frac{\partial\omega(x_2-y_2)}{\partial y_2}e_2}}{..\to[d_1'=-\omega d_2,e_1'=-\omega e_2,x_2'=d_2,d_2'=\omega d_1,..]d_1-e_1=-\omega(x_2-y_2)}$$

$$\dfrac{\dfrac{\dfrac{[x_1' = d_1, d_1' = -\omega d_2, x_2' = d_2, d_2' = \omega d_1, ..](x_1 - y_1)^2 + (x_2 - y_2)^2 \geq p^2}{\frac{\partial\|x-y\|^2}{\partial x_1} d_1 + \frac{\partial\|x-y\|^2}{\partial y_1} e_1 + \frac{\partial\|x-y\|^2}{\partial x_2} d_2 + \frac{\partial\|x-y\|^2}{\partial y_2} e_2 \geq \frac{\partial p^2}{\partial x_1} d_1 \ldots}}{2(x_1 - y_1)(d_1 - e_1) + 2(x_2 - y_2)(d_2 - e_2) \geq 0}}{2(x_1 - y_1)(-\omega(x_2 - y_2)) + 2(x_2 - y_2)\omega(x_1 - y_1) \geq 0}$$

refine dynamics

by differential cut

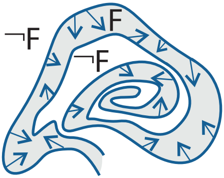
$$-\omega d_2 + \omega e_2 = -\omega(d_2 - e_2)$$

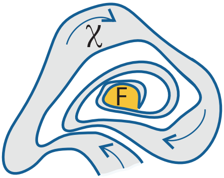
$$\frac{\partial(d_1 - e_1)}{\partial d_1}(-\omega d_2) + \frac{\partial(d_1 - e_1)}{\partial e_1}(-\omega e_2) = -\frac{\partial\omega(x_2 - y_2)}{\partial x_2} d_2 - \frac{\partial\omega(x_2 - y_2)}{\partial y_2} e_2$$

$$.. \to [d_1' = -\omega d_2, e_1' = -\omega e_2, x_2' = d_2, d_2' = \omega d_1, ..]\, d_1 - e_1 = -\omega(x_2 - y_2)$$

Definition (Differential Invariant) (J.Log.Comput. 2010)

F closed under total differentiation with respect to differential constraints

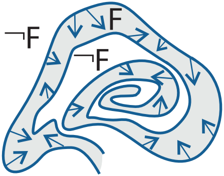


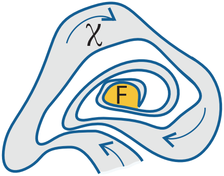


$$d_1 \geq d_2 \rightarrow [x := a^2 + 1;\ d_1' = -\omega d_2, d_2' = \omega d_1\ ]\, d_1 \geq d_2$$

Definition (Differential Invariant) (J.Log.Comput. 2010)

F closed under total differentiation with respect to differential constraints

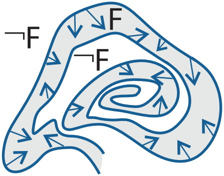


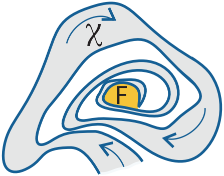


$$d_1 \geq d_2 \rightarrow [x := a^2 + 1;$$
$$(d_1' = -\omega d_2 \land d_2' = \omega d_1) \lor (d_1' \leq 2d_1)$$
$$]\, d_1 \geq d_2$$

Definition (Differential Invariant) (J.Log.Comput. 2010)

F closed under total differentiation with respect to differential constraints

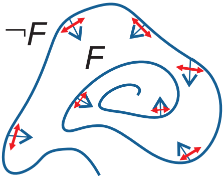


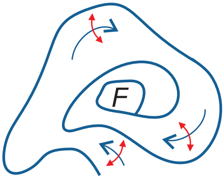


$$d_1 \geq d_2 \rightarrow [x := a^2 + 1;$$
$$\exists \omega\, (\omega \leq 1 \land d_1' = -\omega d_2 \land d_2' = \omega d_1) \lor (d_1' \leq 2d_1)$$
$$]\, d_1 \geq d_2$$

Definition (Differential Invariant) (J.Log.Comput. 2010)

F closed under total differentiation with respect to differential constraints



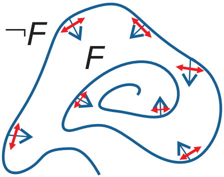


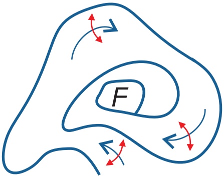
$$d_1 \geq d_2 \rightarrow [x := a^2 + 1;$$
$$\exists \omega\, (\omega \leq 1 \land d_1' = -\omega d_2 \land d_2' = \omega d_1) \lor (d_1' \leq 2d_1)$$
$$]\, d_1 \geq d_2$$

- quantified nondeterminism/disturbance

Definition (Differential Invariant) (J.Log.Comput. 2010)

F closed under total differentiation with respect to differential constraints





$$d_1 \geq d_2 \rightarrow [x := a^2 + 1;$$
$$\exists \omega\, (\omega \leq 1 \wedge d_1' = -\omega d_2 \wedge d_2' = \omega d_1) \vee (d_1' \leq 2d_1)$$
$$]\, d_1 \geq d_2$$

- quantified nondeterminism/disturbance

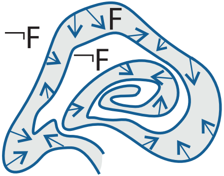
Differential Induction: Local Dynamics w/o Solutions

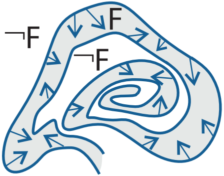
Definition (Differential Invariant) (J.Log.Comput. 2010)

F closed under total differentiation with respect to differential constraints

$$d_1 \geq d_2 \rightarrow [x > 0 \rightarrow \exists a\,(a < 5 \land x := a^2 + 1);$$
$$\exists \omega\,(\omega \leq 1 \land d_1' = -\omega d_2 \land d_2' = \omega d_1) \lor (d_1' \leq 2 d_1)$$
$$]\, d_1 \geq d_2$$

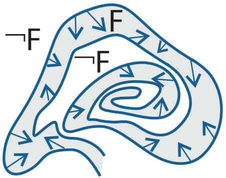
- discrete quantified nondeterminism/disturbance



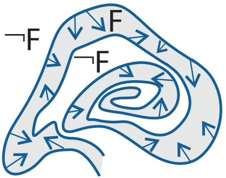


$$\frac{(H \rightarrow F')}{(H \rightarrow F) \rightarrow [x' = \theta \,\&\, H]F}$$

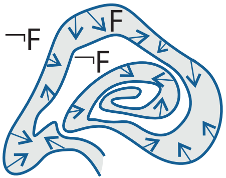
Assuming Differential Invariance

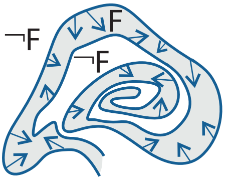


$$\frac{(H \to F')}{(H \to F) \to [x' = \theta \,\&\, H]F}$$



$$\frac{(F \land H \to F')}{(H \to F) \to [x' = \theta \,\&\, H]F}$$



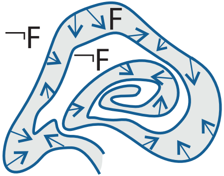


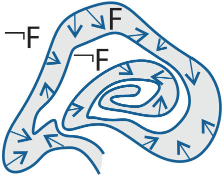
$$\frac{(H \to F')}{(H \to F) \to [x' = \theta \,\&\, H]F}$$

$$\frac{({\color{red}F} \land H \to F')}{(H \to F) \to [x' = \theta \,\&\, H]F}$$

Example (Restrictions)

$$\frac{}{x^2 - 6x + 9 = 0 \to [x' = y, y' = -x]x^2 - 6x + 9 = 0}$$



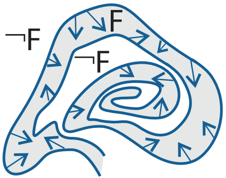


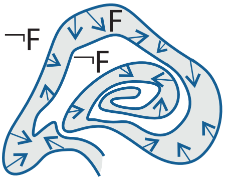
$$\frac{(H \to F')}{(H \to F) \to [x' = \theta \,\&\, H]F}$$

$$\frac{({\color{red}F} \land H \to F')}{(H \to F) \to [x' = \theta \,\&\, H]F}$$

Example (Restrictions)

$$\frac{x^2 - 6x + 9 = 0 \to y\frac{\partial(x^2-6x+9)}{\partial x} - x\frac{\partial(x^2-6x+9)}{\partial y} = 0}{x^2 - 6x + 9 = 0 \to [x' = y, y' = -x]x^2 - 6x + 9 = 0}$$

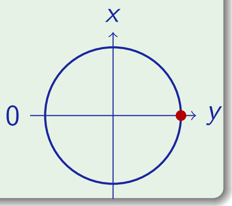




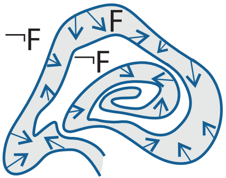
$$\frac{(H \to F')}{(H \to F) \to [x' = \theta \,\&\, H]F} \qquad \frac{(F \land H \to F')}{(H \to F) \to [x' = \theta \,\&\, H]F}$$

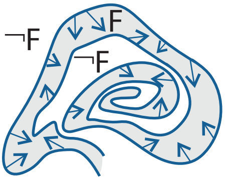
Example (Restrictions)

$$\frac{\dfrac{x^2 - 6x + 9 = 0 \to y2x - 6y = 0}{x^2 - 6x + 9 = 0 \to y\frac{\partial(x^2-6x+9)}{\partial x} - x\frac{\partial(x^2-6x+9)}{\partial y} = 0}}{x^2 - 6x + 9 = 0 \to [x' = y, y' = -x]x^2 - 6x + 9 = 0}$$



Assuming Differential Invariance



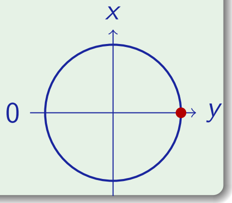


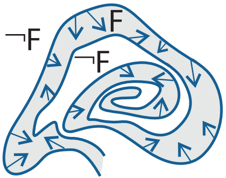
$$\frac{(H \to F')}{(H \to F) \to [x' = \theta \,\&\, H]F}$$

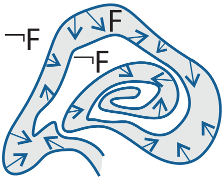
$$\frac{(F \wedge H \to F')}{(H \to F) \to [x' = \theta \,\&\, H]F}$$

Example (Restrictions are unsound!)

$$\cfrac{\cfrac{x^2 - 6x + 9 = 0 \to y2x - 6y = 0}{x^2 - 6x + 9 = 0 \to y\frac{\partial(x^2-6x+9)}{\partial x} - x\frac{\partial(x^2-6x+9)}{\partial y} = 0}}{x^2 - 6x + 9 = 0 \to [x' = y, y' = -x]x^2 - 6x + 9 = 0}$$





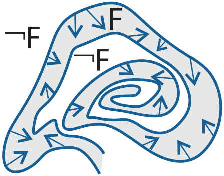


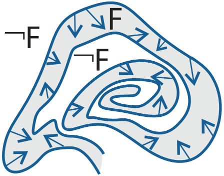
$$\frac{(H \to F')}{(H \to F) \to [x' = \theta \,\&\, H]F}$$

$$\frac{({\color{red}F} \land H \to F')}{(H \to F) \to [x' = \theta \,\&\, H]F}$$

Example (Restrictions)

$$\frac{(x^2 \leq 0 \to 2x \cdot 1 \leq 0)}{x^2 \leq 0 \to [x' = 1]x^2 \leq 0}$$



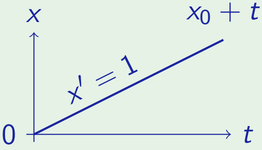


$$\frac{(H \to F')}{(H \to F) \to [x' = \theta \,\&\, H]F}$$

$$\frac{(F \land H \to F')}{(H \to F) \to [x' = \theta \,\&\, H]F}$$

Example (Restrictions are unsound!)

$$\frac{(x^2 \leq 0 \to 2x \cdot 1 \leq 0)}{x^2 \leq 0 \to [x' = 1]x^2 \leq 0}$$

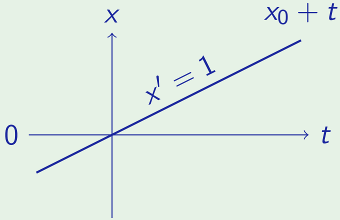


Example (Negative equations)

$$\dfrac{\dfrac{*}{\forall x\,(1 \neq 0)}}{x \neq 0 \rightarrow [x' = 1]x \neq 0}$$

Example (Negative equations)

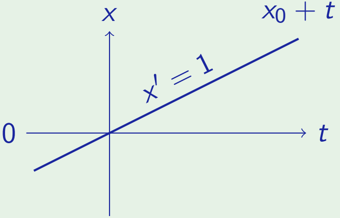
$$\dfrac{\dfrac{*}{\forall x\,(1 \neq 0)}}{x \neq 0 \rightarrow [x' = 1]x \neq 0}$$



Differential Invariance of Negative Equations

Example (Negative equations unsound! Transform)

$$\dfrac{\dfrac{*}{\forall x\,(1 \neq 0)}}{x \neq 0 \to [x' = 1]x \neq 0}$$



Disjunctive Differential Invariants

$$(F \land G)' \equiv$$

Disjunctive Differential Invariants

$$(F \land G)' \equiv F' \land G'$$

Disjunctive Differential Invariants

$$(F \land G)' \equiv F' \land G'$$
$$(F \lor G)' \equiv$$

Disjunctive Differential Invariants

$$(F \land G)' \equiv F' \land G'$$
$$(F \lor G)' \equiv F' \lor G' \text{ sound?}$$

$$(F \land G)' \equiv F' \land G'$$
$$(F \lor G)' \equiv F' \lor G' \text{ sound?}$$

Example (Provable)

$$d_1^2 + d_2^2 = v^2 \to [d_1' = -\omega d_2, d_2' = \omega d_1]\, d_1^2 + d_2^2 = v^2$$

Disjunctive Differential Invariants

$$(F \land G)' \equiv F' \land G'$$
$$(F \lor G)' \equiv F' \lor G' \text{ sound?}$$

Example (Provable)

$$d_1^2 + d_2^2 = v^2 \to [d_1' = -\omega d_2, d_2' = \omega d_1]\, d_1^2 + d_2^2 = v^2$$

Example (Consequence)

$$x_1 \geq 0 \lor d_1^2 + d_2^2 = v^2 \to [d_1' = -\omega d_2, d_2' = \omega d_1](x_1 \geq 0 \lor d_1^2 + d_2^2 = v^2)$$

$$(F \wedge G)' \equiv F' \wedge G'$$

$$(F \vee G)' \equiv F' \vee G' \text{ sound?}$$

Example (Provable)

$$d_1^2 + d_2^2 = v^2 \to [d_1' = -\omega d_2, d_2' = \omega d_1]\, d_1^2 + d_2^2 = v^2$$

Example (Unsound!)

$$x_1 \geq 0 \vee d_1^2 + d_2^2 = v^2 \to [d_1' = -\omega d_2, d_2' = \omega d_1](x_1 \geq 0 \vee d_1^2 + d_2^2 = v^2)$$

Disjunctive Differential Invariants

$$(F \land G)' \equiv F' \land G'$$
$$(F \lor G)' \equiv F' \land G' \text{ sound!}$$

Example (Provable)

$$d_1^2 + d_2^2 = v^2 \to [d_1' = -\omega d_2, d_2' = \omega d_1]\, d_1^2 + d_2^2 = v^2$$

Example (Unsound!)

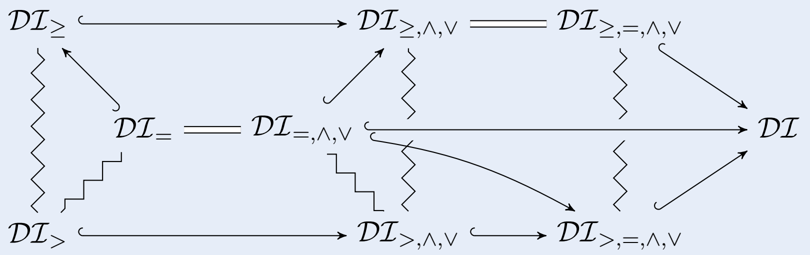
$$x_1 \geq 0 \lor d_1^2 + d_2^2 = v^2 \to [d_1' = -\omega d_2, d_2' = \omega d_1](x_1 \geq 0 \lor d_1^2 + d_2^2 = v^2)$$

Outline Background

Theorem (Closure properties of differential invariants) (LMCS 2012)

Closed under conjunction, differentiation, and propositional equivalences.

Theorem (Differential Invariance Chart) (LMCS 2012)



Differential Cuts

$$\frac{F \rightarrow [x' = \theta \,\&\, H]C \qquad F \rightarrow [x' = \theta \,\&\, (H \land C)]F}{F \rightarrow [x' = \theta \,\&\, H]F}$$

Ex: Differential Cuts

$$x^3 \geq -1 \wedge y^5 \geq 0 \rightarrow [x' = (x-3)^4 + y^5, y' = y^2] x^3 \geq -1$$

Ex: Differential Cuts

$$\frac{}{x^3 \geq -1 \land y^5 \geq 0 \rightarrow [x' = (x-3)^4 + y^5, y' = y^2] x^3 \geq -1}$$

$$\frac{}{y^5 \geq 0 \rightarrow [x' = (x-3)^4 + y^5, y' = y^2] y^5 \geq 0}$$

$$\frac{}{x^3 \geq -1 \land y^5 \geq 0 \rightarrow [x' = (x-3)^4 + y^5, y' = y^2] x^3 \geq -1}$$

$$\frac{\frac{}{5y^4 y' \geq 0}}{y^5 \geq 0 \rightarrow [x' = (x-3)^4 + y^5, y' = y^2] y^5 \geq 0}$$

Ex: Differential Cuts

$$x^3 \geq -1 \land y^5 \geq 0 \rightarrow [x' = (x-3)^4 + y^5, y' = y^2] x^3 \geq -1$$

$$5y^4 y^2 \geq 0$$

$$5y^4 y' \geq 0$$

$$y^5 \geq 0 \rightarrow [x' = (x-3)^4 + y^5, y' = y^2] y^5 \geq 0$$

Ex: Differential Cuts

$$\overline{x^3 \geq -1 \land y^5 \geq 0 \rightarrow [x' = (x-3)^4 + y^5, y' = y^2] x^3 \geq -1}$$

$$\cfrac{\cfrac{\cfrac{*}{5y^4 y^2 \geq 0}}{5y^4 y' \geq 0}}{y^5 \geq 0 \rightarrow [x' = (x-3)^4 + y^5, y' = y^2] y^5 \geq 0}$$

Ex: Differential Cuts

$$\dfrac{\dfrac{}{x^3 \geq -1 \rightarrow [x' = (x-3)^4 + y^5, y' = y^2 \,\&\, y^5 \geq 0] x^3 \geq -1} \triangleright}{x^3 \geq -1 \land y^5 \geq 0 \rightarrow [x' = (x-3)^4 + y^5, y' = y^2] x^3 \geq -1}$$

$$\dfrac{\dfrac{\dfrac{*}{5y^4 {\color{red}y^2} \geq 0}}{5y^4 {\color{red}y'} \geq 0}}{y^5 \geq 0 \rightarrow [x' = (x-3)^4 + y^5, y' = y^2] y^5 \geq 0}$$

$$\dfrac{\dfrac{y^5 \geq 0 \rightarrow 2x^2 x' \geq 0}{x^3 \geq -1 \rightarrow [x' = (x-3)^4 + y^5, y' = y^2 \,\&\, y^5 \geq 0] x^3 \geq -1}\ \triangleright}{x^3 \geq -1 \wedge y^5 \geq 0 \rightarrow [x' = (x-3)^4 + y^5, y' = y^2] x^3 \geq -1}$$

$$\dfrac{\dfrac{\dfrac{*}{5y^4 y^2 \geq 0}}{5y^4 y' \geq 0}}{y^5 \geq 0 \rightarrow [x' = (x-3)^4 + y^5, y' = y^2] y^5 \geq 0}$$

$$\dfrac{\dfrac{\dfrac{y^5 \geq 0 \rightarrow 2x^2((x-3)^4 + y^5) \geq 0}{y^5 \geq 0 \rightarrow 2x^2 x' \geq 0}}{x^3 \geq -1 \rightarrow [x' = (x-3)^4 + y^5, y' = y^2 \,\&\, y^5 \geq 0] x^3 \geq -1} \triangleright}{x^3 \geq -1 \land y^5 \geq 0 \rightarrow [x' = (x-3)^4 + y^5, y' = y^2] x^3 \geq -1}$$

$$\dfrac{\dfrac{\dfrac{*}{5y^4 y^2 \geq 0}}{5y^4 y' \geq 0}}{y^5 \geq 0 \rightarrow [x' = (x-3)^4 + y^5, y' = y^2] y^5 \geq 0}$$

Ex: Differential Cuts

$$\dfrac{\dfrac{\dfrac{\dfrac{*}{y^5 \geq 0 \to 2x^2((x-3)^4 + y^5) \geq 0}}{y^5 \geq 0 \to 2x^2 x' \geq 0}}{x^3 \geq -1 \to [x' = (x-3)^4 + y^5, y' = y^2 \,\&\, y^5 \geq 0]x^3 \geq -1} \;\triangleright}{x^3 \geq -1 \land y^5 \geq 0 \to [x' = (x-3)^4 + y^5, y' = y^2]x^3 \geq -1}$$

$$\dfrac{\dfrac{\dfrac{*}{5y^4 y^2 \geq 0}}{5y^4 y' \geq 0}}{y^5 \geq 0 \to [x' = (x-3)^4 + y^5, y' = y^2]y^5 \geq 0}$$

Differential Cuts

$$\frac{F \rightarrow [x' = \theta \,\&\, H]C \qquad F \rightarrow [x' = \theta \,\&\, (H \land C)]F}{F \rightarrow [x' = \theta \,\&\, H]F}$$

Differential Cuts

$$\frac{F \rightarrow [x' = \theta \,\&\, H] C \qquad F \rightarrow [x' = \theta \,\&\, (H \wedge C)] F}{F \rightarrow [x' = \theta \,\&\, H] F}$$

Theorem (Gentzen's Cut Elimination)

$$\frac{A \rightarrow B \vee C \quad A \wedge C \rightarrow B}{A \rightarrow B}$$ *cut can be eliminated*

Differential Cuts

$$\frac{F \rightarrow [x' = \theta \,\&\, H]C \qquad F \rightarrow [x' = \theta \,\&\, (H \wedge C)]F}{F \rightarrow [x' = \theta \,\&\, H]F}$$

Theorem (Gentzen's Cut Elimination)

$$\frac{A \rightarrow B \vee C \quad A \wedge C \rightarrow B}{A \rightarrow B}$$ *cut can be eliminated*

Theorem (No Differential Cut Elimination) (LMCS 2012)

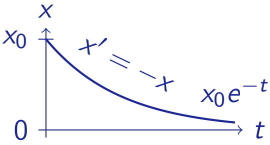
Deductive power with differential cut exceeds deductive power without.

$$\mathcal{DCI} > \mathcal{DI}$$

Ex: Exponentials

Counterexample ()

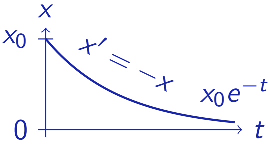
$$\overline{x > 0 \rightarrow [x' = -x]x > 0}$$



Ex: Exponentials

Counterexample ()

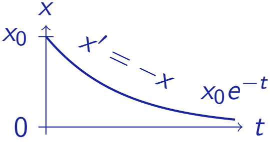
$$\dfrac{\overline{\quad x' > 0 \quad}}{x > 0 \rightarrow [x' = -x]x > 0}$$



Ex: Exponentials

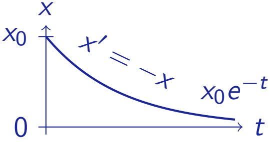
Counterexample ()

$$\dfrac{\dfrac{\dfrac{}{-x > 0}}{x' > 0}}{x > 0 \rightarrow [x' = -x]x > 0}$$



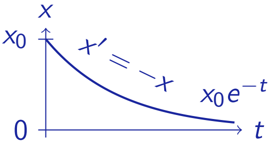
Counterexample (Cannot prove like this)

$$\dfrac{\dfrac{\dfrac{\text{not valid}}{-x > 0}}{x' > 0}}{x > 0 \rightarrow [x' = -x]x > 0}$$



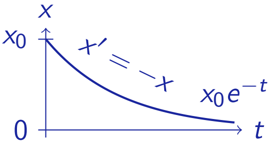
Example (Successful proof)

$$\frac{}{x > 0 \rightarrow [x' = -x]x > 0}$$



Example (Successful proof)

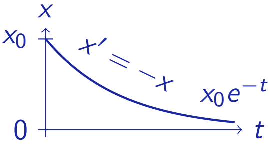
$$\dfrac{\dfrac{}{x > 0 \leftrightarrow \exists y\, xy^2 = 1} \qquad \dfrac{}{xy^2 = 1 \rightarrow [x' = -x, y' = \frac{y}{2}]xy^2 = 1}}{x > 0 \rightarrow [x' = -x]x > 0}$$



Differential Auxiliaries

Example (Successful proof)

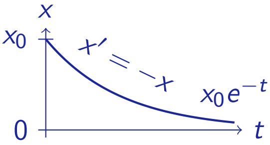
$$\dfrac{\dfrac{*}{x > 0 \leftrightarrow \exists y\, xy^2 = 1} \qquad \dfrac{}{xy^2 = 1 \rightarrow [x' = -x, y' = \frac{y}{2}]xy^2 = 1}}{x > 0 \rightarrow [x' = -x]x > 0}$$



Differential Auxiliaries

Example (Successful proof)

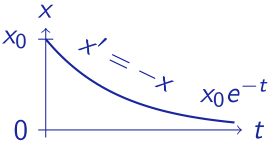
$$\dfrac{\dfrac{*}{x > 0 \leftrightarrow \exists y\, xy^2 = 1} \qquad \dfrac{\dfrac{}{x'y^2 + x2yy' = 0}}{xy^2 = 1 \rightarrow [x' = -x, y' = \frac{y}{2}]xy^2 = 1}}{x > 0 \rightarrow [x' = -x]x > 0}$$



Differential Auxiliaries

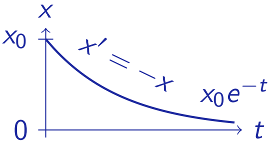
Example (Successful proof)

$$\dfrac{\dfrac{*}{x > 0 \leftrightarrow \exists y\, xy^2 = 1} \qquad \dfrac{\dfrac{\overline{-xy^2 + 2xy\frac{y}{2} = 0}}{x'y^2 + x2yy' = 0}}{xy^2 = 1 \rightarrow [x' = -x, y' = \frac{y}{2}]xy^2 = 1}}{x > 0 \rightarrow [x' = -x]x > 0}$$



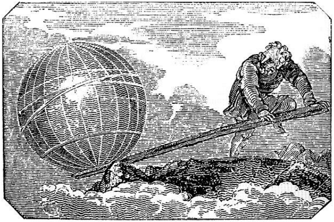
Example (Successful proof)

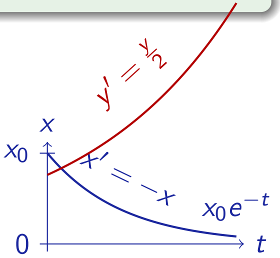
$$\dfrac{\dfrac{*}{x > 0 \leftrightarrow \exists y\, xy^2 = 1} \qquad \dfrac{\dfrac{\dfrac{*}{-xy^2 + 2xy\frac{y}{2} = 0}}{x'y^2 + x2yy' = 0}}{xy^2 = 1 \rightarrow [x' = -x, y' = \frac{y}{2}]xy^2 = 1}}{x > 0 \rightarrow [x' = -x]x > 0}$$



Example (Successful proof)

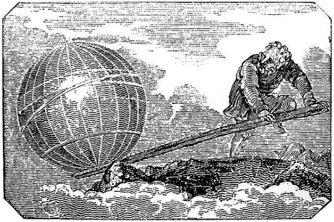
$$
\dfrac{\dfrac{*}{x > 0 \leftrightarrow \exists y\, xy^2 = 1} \quad \dfrac{\dfrac{\dfrac{*}{-xy^2 + 2xy\frac{y}{2} = 0}}{x'y^2 + x2yy' = 0}}{xy^2 = 1 \rightarrow [x' = -x, y' = \frac{y}{2}]xy^2 = 1}}{x > 0 \rightarrow [x' = -x]x > 0}
$$

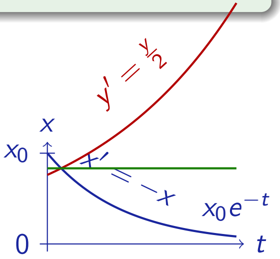




Example (Successful proof)

$$\dfrac{\dfrac{*}{x>0 \leftrightarrow \exists y\, xy^2=1} \qquad \dfrac{\dfrac{\dfrac{*}{-xy^2+2xy\frac{y}{2}=0}}{x'y^2+x2yy'=0}}{xy^2=1 \rightarrow [x'=-x,y'=\frac{y}{2}]xy^2=1}}{x>0 \rightarrow [x'=-x]x>0}$$





$$\frac{\phi \leftrightarrow \exists y\, \psi \quad \psi \rightarrow [x' = \theta, y' = \vartheta \,\&\, H]\psi}{\phi \rightarrow [x' = \theta \,\&\, H]\phi}$$

if $y' = \vartheta$ has solution $y : [0, \infty) \to \mathbb{R}^n$

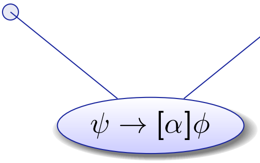
Theorem (Auxiliary Differential Variables) (LMCS 2012)

Deductive power with differential auxiliaries exceeds deductive power without.

$$\mathcal{DCI} + DA > \mathcal{DCI}$$

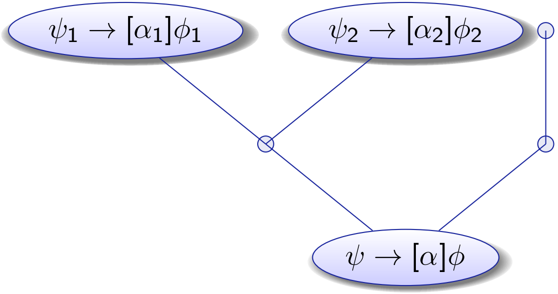
Outline Background

Differential Invariants as Fixedpoints



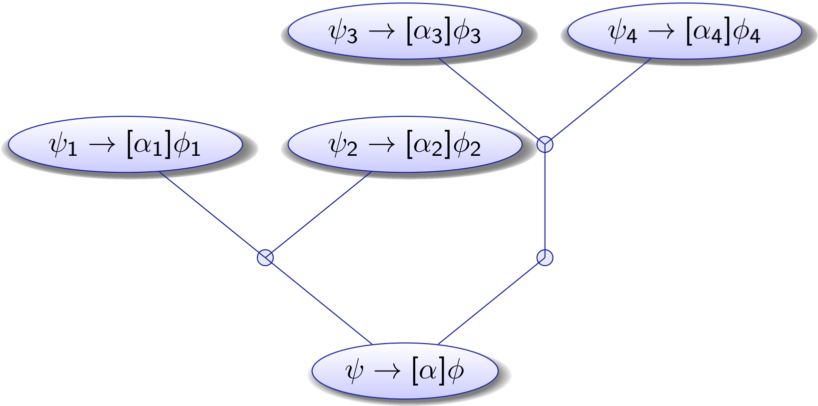
Details

Differential Invariants as Fixedpoints



for $\cup\,,;\,,:=$ do decompose

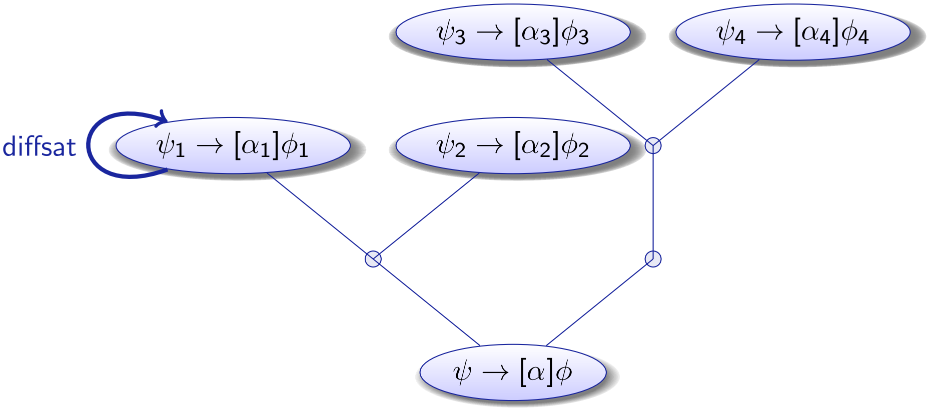
Details



for $\cup$, ; , := do decompose

Details

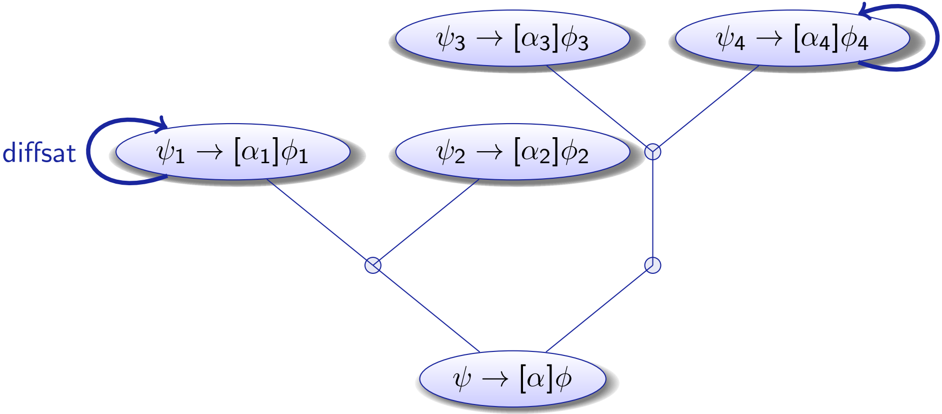
Differential Invariants as Fixedpoints



for $\cup\,;\,;\,:=$ do decompose

for $x' = \ldots$ do diffsat

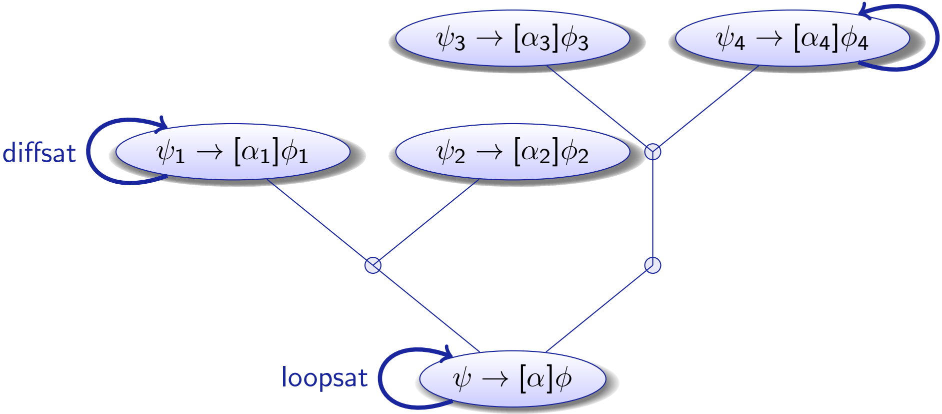
Details



for $\cup, ;, :=$ do decompose
for $x' = \dots$ do diffsat

Details

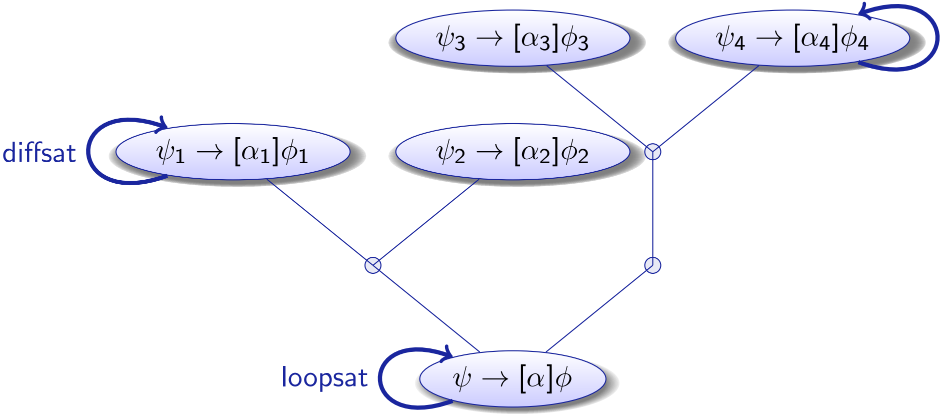
Differential Invariants as Fixedpoints



for $\cup\, , ;\, , :=$	do decompose
for $x' = \dots$	do diffsat
for α^*	do loopsat

Details

Differential Invariants as Fixedpoints



for $\cup\, ,\, ;\, ,\, :=$	do decompose	repeat until fixedpoint
for $x' = \dots$	do diffsat	
for α^*	do loopsat	

Details

Outline Background

Derivations and Differentiation

Definition (Syntactic total derivation $D : \mathrm{Trm} \to \mathrm{Trm}$)

$$
\begin{aligned}
D(r) &= 0 && \text{if } r \text{ is a (rigid) number symbol}\\
D(x^{(n)}) &= x^{(n+1)} && \text{if } x \in \Sigma \text{ is flexible, } n \geq 0\\
D(a+b) &= D(a) + D(b)\\
D(a \cdot b) &= D(a) \cdot b + a \cdot D(b)\\
D(a/b) &= (D(a) \cdot b - a \cdot D(b))/b^2\\
D(F) &\equiv \bigwedge_{i=1}^{m} D(F_i) && \{F_1, \dots, F_m\} \text{ all literals of } F\\
D(a \geq b) &\equiv D(a) \geq D(b) && \text{accordingly for } <, >, \leq, =
\end{aligned}
$$

$$
\begin{aligned}
\mathcal{P} &\equiv (x_1 - y_1)^2 + (x_2 - y_2)^2 \geq p^2\\
\Rightarrow D(\mathcal{P}) &\equiv 2(x_1 - y_1)(x_1' - y_1') + 2(x_2 - y_2)(x_2' - y_2') \geq 0
\end{aligned}
$$

Derivations and Differentiation

Definition (Syntactic total derivation $D : \mathrm{Trm} \to \mathrm{Trm}$)

$$
\begin{aligned}
D(r) &= 0 && \text{if } r \text{ is a (rigid) number symbol}\\
D(x^{(n)}) &= x^{(n+1)} && \text{if } x \in \Sigma \text{ is flexible, } n \geq 0\\
D(a+b) &= D(a) + D(b)\\
D(a \cdot b) &= D(a) \cdot b + a \cdot D(b)\\
D(a/b) &= (D(a) \cdot b - a \cdot D(b))/b^2\\
D(F) &\equiv \bigwedge_{i=1}^{m} D(F_i) && \{F_1, \dots, F_m\} \text{ all literals of } F\\
D(a \geq b) &\equiv D(a) \geq D(b) && \text{accordingly for } <, >, \leq, =
\end{aligned}
$$

$$
\begin{aligned}
\mathcal{P} &\equiv (x_1 - y_1)^2 + (x_2 - y_2)^2 \geq p^2\\
\Rightarrow D(\mathcal{P}) &\equiv 2(x_1 - y_1)(x_1' - y_1') + 2(x_2 - y_2)(x_2' - y_2') \geq 0
\end{aligned}
$$

Derivations and Differentiation

Syntactic derivation $D(\cdot)$ coincides with analytic differentiation:

Lemma (Derivation lemma)

Valuation is differential homomorphism: for all flows φ *all* $\zeta \in [0, r]$

$$\frac{\mathrm{d}\, [\![\theta]\!]_{\varphi(t)}}{\mathrm{d}t}(\zeta) = [\![D(\theta)]\!]_{\bar{\varphi}(\zeta)}$$

Theorem (Differential Invariant)

$$\frac{\chi \to F'}{\chi \to F \to [x' = \theta \,\&\, \chi]F} \quad \textit{sound for } F' \equiv D(F)^{\theta}_{x'}$$

Derivations and Differentiation

Syntactic derivation $D(\cdot)$ coincides with analytic differentiation:

Lemma (Derivation lemma)

Valuation is differential homomorphism: for all flows φ *all* $\zeta \in [0, r]$

$$\frac{\mathrm{d}\,[\![\theta]\!]_{\varphi(t)}}{\mathrm{d}t}(\zeta) = [\![D(\theta)]\!]_{\bar{\varphi}(\zeta)}$$

Locally understand differential equations by substitution:

Lemma (Differential substitution principle)

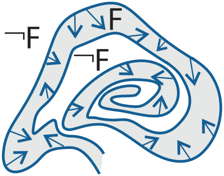
If $\varphi \models x_i' = \theta_i \wedge \chi$, *then* $\varphi \models \mathcal{D} \leftrightarrow (\chi \rightarrow \mathcal{D}_{x_i'}^{\theta_i})$ *for all* $\mathcal{D}$.

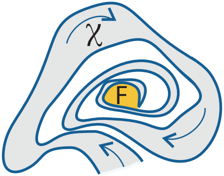
Theorem (Differential Invariant)

$$\frac{\chi \rightarrow F'}{\chi \rightarrow F \rightarrow [x' = \theta \,\&\, \chi]F} \quad \textit{sound for } F' \equiv D(F)_{x'}^{\theta}$$

Definition (Differential Invariant) (J.Log.Comput. 2010)

F closed under total differentiation with respect to differential constraints





$$\frac{(\chi \rightarrow F')}{\chi \rightarrow F \rightarrow [x' = \theta \,\&\, \chi]F}$$

$$\frac{(\neg F \land \chi \rightarrow F'_{\gg})}{[x' = \theta \,\&\, \neg F]\chi \rightarrow \langle x' = \theta \,\&\, \chi\rangle F}$$

$$\frac{}{\langle x' = a \rangle x \geq b}$$

Ex: Differential Variant for Progress Discovery

$$\frac{\overline{\exists \varepsilon{>}0\, \forall x\, (x \leq b \rightarrow x' \geq \varepsilon)}}{\langle x' = a \rangle x \geq b}$$

Ex: Differential Variant for Progress Discovery

$$\dfrac{\dfrac{\exists \varepsilon{>}0\, \forall x\, (x \leq b \rightarrow a \geq \varepsilon)}{\exists \varepsilon{>}0\, \forall x\, (x \leq b \rightarrow x' \geq \varepsilon)}}{\langle x' = a \rangle x \geq b}$$

Ex: Differential Variant for Progress Discovery

$$\cfrac{\cfrac{\cfrac{a > 0}{\exists \varepsilon{>}0\, \forall x\, (x \leq b \rightarrow a \geq \varepsilon)}}{\exists \varepsilon{>}0\, \forall x\, (x \leq b \rightarrow x' \geq \varepsilon)}}{\langle x' = a \rangle x \geq b}$$

$$\dfrac{\dfrac{\dfrac{b > 0}{\text{QE}(\exists d\,((\|d\|^2 \le b^2) \land (d_1 > 0 \land d_2 > 0)))}}{\dfrac{\|d\|^2 \le b^2 \qquad \dfrac{\dfrac{d_1 > 0 \land d_2 > 0}{\exists \epsilon{>}0\,\forall x_1, x_2\,(x_1 < p_1 \lor x_2 < p_2 \rightarrow d_1 \ge \epsilon \land d_2 \ge \epsilon)}}{\langle \mathcal{F}(0)\rangle(x_1 \ge p_1 \land x_2 \ge p_2)}}{\|d\|^2 \le b^2 \land \langle \mathcal{F}(0)\rangle(x_1 \ge p_1 \land x_2 \ge p_2)}}}{\dfrac{\exists d\,(\|d\|^2 \le b^2 \land \langle \mathcal{F}(0)\rangle(x_1 \ge p_1 \land x_2 \ge p_2))}{\forall p\,\exists d\,(\|d\|^2 \le b^2 \land \langle \mathcal{F}(0)\rangle(x_1 \ge p_1 \land x_2 \ge p_2))}}$$

$$\begin{aligned}
\mathcal{F}(0) &\equiv x_1' = d_1 \land x_2' = d_2 \\
F &\equiv x_1 \ge p_1 \land x_2 \ge p_2
\end{aligned}$$

Differential Variants for Flight Progress

$$
\dfrac{\dfrac{\dfrac{b > 0}{\mathrm{QE}(\exists d\, ((\|d\|^2 \le b^2) \land (d_1 > 0 \land d_2 > 0)))}}{\dfrac{\|d\|^2 \le b^2 \qquad \dfrac{\dfrac{d_1 > 0 \land d_2 > 0}{\exists \epsilon{>}0\, \forall x_1, x_2\, (x_1 < p_1 \lor x_2 < p_2 \rightarrow d_1 \ge \epsilon \land d_2 \ge \epsilon)}}{\langle \mathcal{F}(0) \rangle (x_1 \ge p_1 \land x_2 \ge p_2)}}{\|d\|^2 \le b^2 \land \langle \mathcal{F}(0) \rangle (x_1 \ge p_1 \land x_2 \ge p_2)}}}{\dfrac{\exists d\, (\|d\|^2 \le b^2 \land \langle \mathcal{F}(0) \rangle (x_1 \ge p_1 \land x_2 \ge p_2))}{\forall p\, \exists d\, (\|d\|^2 \le b^2 \land \langle \mathcal{F}(0) \rangle (x_1 \ge p_1 \land x_2 \ge p_2))}}
$$

$$
\begin{aligned}
\mathcal{F}(0) &\equiv x_1' = d_1 \land x_2' = d_2 \\
F &\equiv x_1 \ge p_1 \land x_2 \ge p_2 \\
F' &\equiv x_1' \ge 0 \land x_2' \ge 0
\end{aligned}
$$

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{b > 0}{\mathrm{QE}(\exists d\, ((\|d\|^2 \le b^2) \land (d_1 > 0 \land d_2 > 0)))}}{\|d\|^2 \le b^2 \qquad \dfrac{\dfrac{d_1 > 0 \land d_2 > 0}{\exists \epsilon{>}0\, \forall x_1, x_2\, (x_1 < p_1 \lor x_2 < p_2 \to d_1 \ge \epsilon \land d_2 \ge \epsilon)}}{\langle \mathcal{F}(0) \rangle (x_1 \ge p_1 \land x_2 \ge p_2)}}}{\|d\|^2 \le b^2 \land \langle \mathcal{F}(0) \rangle (x_1 \ge p_1 \land x_2 \ge p_2)}}{\exists d\, (\|d\|^2 \le b^2 \land \langle \mathcal{F}(0) \rangle (x_1 \ge p_1 \land x_2 \ge p_2))}}{\forall p\, \exists d\, (\|d\|^2 \le b^2 \land \langle \mathcal{F}(0) \rangle (x_1 \ge p_1 \land x_2 \ge p_2))}
$$

$$
\begin{aligned}
\mathcal{F}(0) &\equiv x_1' = d_1 \land x_2' = d_2 \\
F &\equiv x_1 \ge p_1 \land x_2 \ge p_2 \\
F' &\equiv x_1' \ge 0 \land x_2' \ge 0 \\
F' \ge \epsilon &\equiv x_1' \ge \epsilon \land x_2' \ge \epsilon
\end{aligned}
$$

Differential Variants for Flight Progress

$$
\dfrac{\dfrac{\dfrac{\dfrac{b > 0}{\mathrm{QE}(\exists d\,((\|d\|^2 \le b^2) \land (d_1 > 0 \land d_2 > 0)))}}{\|d\|^2 \le b^2 \qquad \dfrac{\dfrac{d_1 > 0 \land d_2 > 0}{\exists \epsilon{>}0\, \forall x_1, x_2\, (x_1 < p_1 \lor x_2 < p_2 \rightarrow d_1 \ge \epsilon \land d_2 \ge \epsilon)}}{\langle \mathcal{F}(0) \rangle (x_1 \ge p_1 \land x_2 \ge p_2)}}}{\|d\|^2 \le b^2 \land \langle \mathcal{F}(0) \rangle (x_1 \ge p_1 \land x_2 \ge p_2)}}{\dfrac{\exists d\,(\|d\|^2 \le b^2 \land \langle \mathcal{F}(0) \rangle (x_1 \ge p_1 \land x_2 \ge p_2))}{\forall p\, \exists d\,(\|d\|^2 \le b^2 \land \langle \mathcal{F}(0) \rangle (x_1 \ge p_1 \land x_2 \ge p_2))}}
$$

$$
\begin{aligned}
\mathcal{F}(0) &\equiv x_1' = d_1 \land x_2' = d_2 \\
F &\equiv x_1 \ge p_1 \land x_2 \ge p_2 \\
F' &\equiv x_1' \ge 0 \land x_2' \ge 0 \\
F' \ge \epsilon &\equiv x_1' \ge \epsilon \land x_2' \ge \epsilon
\end{aligned}
$$

Differential Variants for Flight Progress

$$\dfrac{\dfrac{\dfrac{\dfrac{b > 0}{\mathrm{QE}(\exists d\,((\|d\|^2 \le b^2) \land (d_1 > 0 \land d_2 > 0)))}\qquad \|d\|^2 \le b^2 \qquad \dfrac{\dfrac{d_1 > 0 \land d_2 > 0}{\exists \epsilon{>}0\, \forall x_1, x_2\, (x_1 < p_1 \lor x_2 < p_2 \to d_1 \ge \epsilon \land d_2 \ge \epsilon)}}{\langle \mathcal{F}(0)\rangle (x_1 \ge p_1 \land x_2 \ge p_2)}}{\|d\|^2 \le b^2 \land \langle \mathcal{F}(0)\rangle (x_1 \ge p_1 \land x_2 \ge p_2)}}{\exists d\,(\|d\|^2 \le b^2 \land \langle \mathcal{F}(0)\rangle (x_1 \ge p_1 \land x_2 \ge p_2))}}{\forall p\, \exists d\,(\|d\|^2 \le b^2 \land \langle \mathcal{F}(0)\rangle (x_1 \ge p_1 \land x_2 \ge p_2))}$$

$$\begin{aligned}
\mathcal{F}(0) &\equiv x_1' = d_1 \land x_2' = d_2 \\
F &\equiv x_1 \ge p_1 \land x_2 \ge p_2 \\
F' &\equiv d_1 \ge 0 \land d_2 \ge 0 \\
F' \ge \epsilon &\equiv d_1 \ge \epsilon \land d_2 \ge \epsilon
\end{aligned}$$

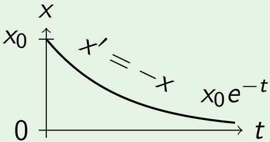
Differential Variants for Progress

Example (Progress)

$$\frac{\forall x\,(x > 0 \rightarrow -x < 0)}{\langle x' = -x \rangle x \leq 0}$$

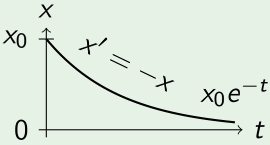
Example (Progress)

$$\frac{\forall x\, (x > 0 \to -x < 0)}{\langle x' = -x \rangle x \leq 0}$$



Example (Unsound without minimal progress!)

$$\frac{\forall x\,(x > 0 \rightarrow -x < 0)}{\langle x' = -x \rangle x \leq 0}$$

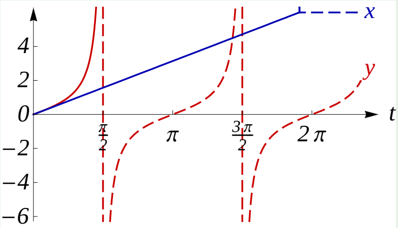


Example (Mixed dynamics)

$$\dfrac{\dfrac{*}{\exists \varepsilon{>}0\, \forall x \forall y\, (x < 6 \to 1 \geq \varepsilon)}}{\langle x' = 1 \land y' = 1 + y^2 \rangle x \geq 6}$$

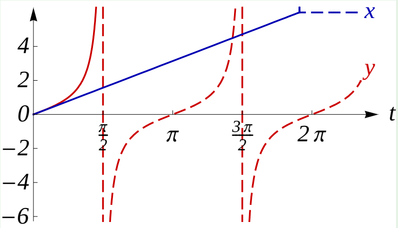
Example (Mixed dynamics)

$$\dfrac{\dfrac{*}{\exists\varepsilon{>}0\,\forall x\forall y\,(x < 6 \rightarrow 1 \geq \varepsilon)}}{\langle x' = 1 \land y' = 1 + y^2\rangle x \geq 6}$$



Example (Unsound without Lipschitz-continuity!)

$$\dfrac{\dfrac{*}{\exists \varepsilon{>}0\, \forall x \forall y\, (x < 6 \rightarrow 1 \geq \varepsilon)}}{\langle x' = 1 \land y' = 1 + y^2 \rangle x \geq 6}$$



Outline Background

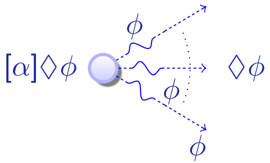
Temporal Modalities + Dynamic Modalities

problem	technique	Op	Par	T	closed
$ETCS \models z < MA$	TL-MC	✓	×	✓	×
$\models (\mathrm{Ax}(ETCS) \to z < MA)$	TL-calculus	×	...	✓	...
$\models [ETCS]\, z < MA$	DL-calculus	✓	✓	×	✓
$\models [ETCS]\Box\, z < MA$	dTL-calculus	✓	✓	✓	✓

problem	technique	Op	Par	T	closed
$ETCS \models z < MA$	TL-MC	✓	×	✓	×
$\models (\mathrm{Ax}(ETCS) \to z < MA)$	TL-calculus	×	...	✓	...
$\models [ETCS]\, z < MA$	DL-calculus	✓	✓	×	✓
$\models [ETCS]\Box\, z < MA$	dTL-calculus	✓	✓	✓	✓

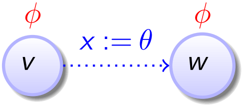
differential temporal dynamic logic

$$\mathrm{dTL} = \mathrm{TL} + \mathrm{DL} + \mathrm{HP}$$

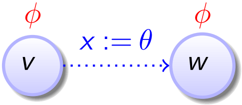


Proof Calculus for Temporal dTL

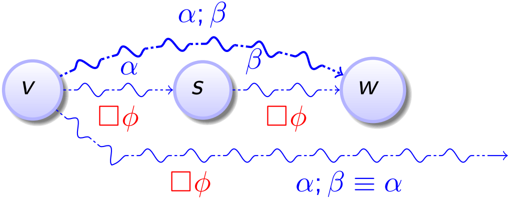
$$\frac{\phi \land [x := \theta]\phi}{[x := \theta]\Box\phi}$$



$$\frac{\phi \wedge [x := \theta]\phi}{[x := \theta]\Box\phi}$$

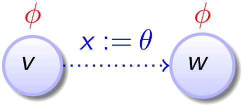


$$\frac{[\alpha]\Box\phi \wedge [\alpha][\beta]\Box\phi}{[\alpha;\beta]\Box\phi}$$

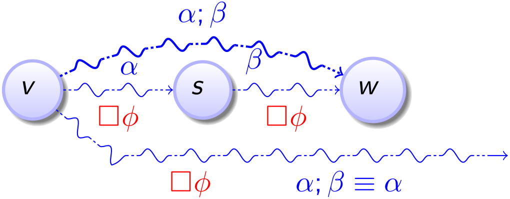


Proof Calculus for Temporal dTL

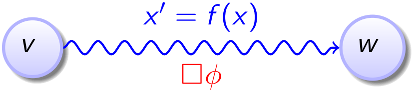
$$\frac{\phi \wedge [x := \theta]\phi}{[x := \theta]\Box\phi}$$



$$\frac{[\alpha]\Box\phi \wedge [\alpha][\beta]\Box\phi}{[\alpha;\beta]\Box\phi}$$



$$\frac{[x' = \theta]\phi}{[x' = \theta]\Box\phi}$$

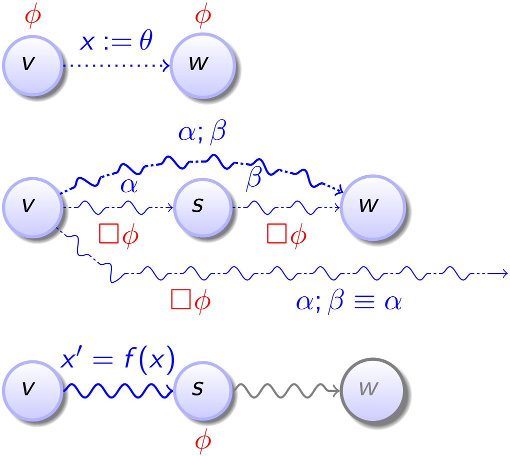


Proof Calculus for Temporal dTL

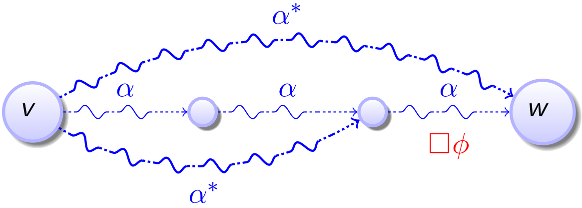
$$\frac{\phi \land [x := \theta]\phi}{[x := \theta]\Box\phi}$$

$$\frac{[\alpha]\Box\phi \land [\alpha][\beta]\Box\phi}{[\alpha; \beta]\Box\phi}$$

$$\frac{[x' = \theta]\phi}{[x' = \theta]\Box\phi}$$

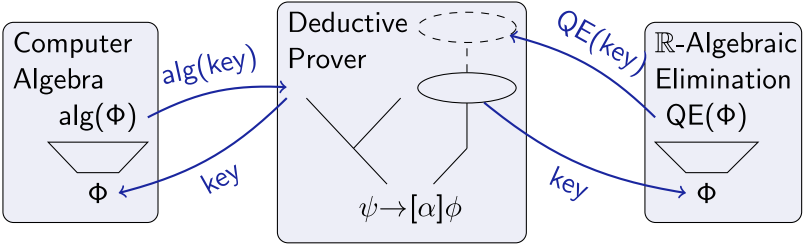


$$\frac{[\alpha^*][\alpha]\Box\phi}{[\alpha^*]\Box\phi}$$

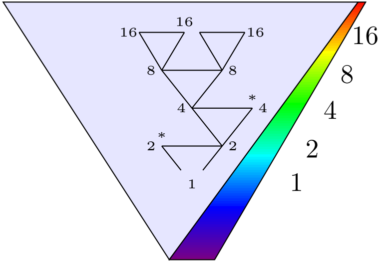


Outline Background

KeYmaera Verification Architecture

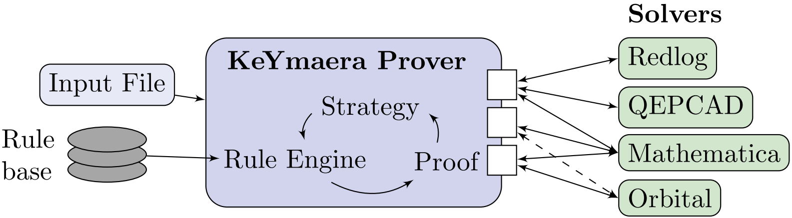


56 interactions?



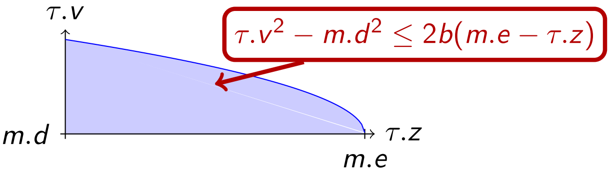
0–1 interactions!

KeYmaera Prover Architecture



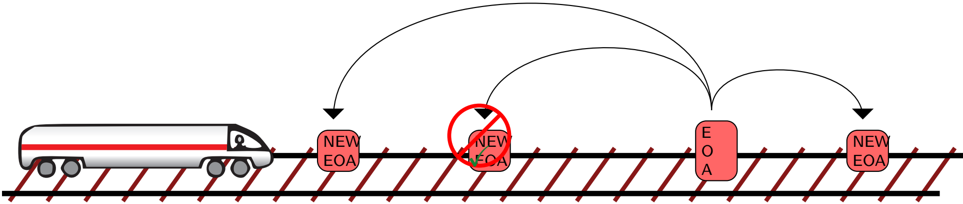
Outline Background

ETCS Controllability



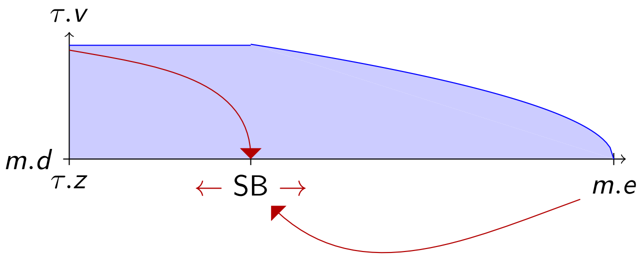
Proposition (Controllability)

$$[\tau.z' = \tau.v, \tau.v' = -b \,\&\, \tau.v \geq 0](\tau.z \geq m.e \rightarrow \tau.v \leq m.d)$$
$$\equiv \tau.v^2 - m.d^2 \leq 2b(m.e - \tau.z)$$



Proposition (RBC Controllability)

$$m.d \geq 0 \land b > 0 \rightarrow [m_0 := m;\ RBC] \Big($$
$$m_0.d^2 - m.d^2 \leq 2b(m.e - m_0.e) \land m_0.d \geq 0 \land m.d \geq 0 \leftrightarrow \forall \tau$$
$$((\langle m := m_0 \rangle \tau.v^2 - m.d^2 \leq 2b(m.e - \tau.z)) \rightarrow \tau.v^2 - m.d^2 \leq 2b(m.e - \tau.z)\Big)$$

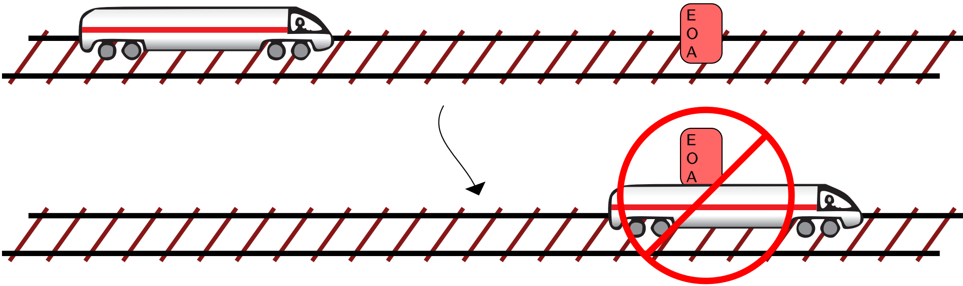


Proposition (Reactivity)

$$\left(\forall m.e\, \forall \tau.z \left(m.e - \tau.z \geq SB \land \tau.v^2 - m.d^2 \leq 2b(m.e - \tau.z) \rightarrow [\tau.a := A;\ drive]\, \tau.v^2 - m.d^2 \leq 2b(m.e - \tau.z)\right)\right)$$

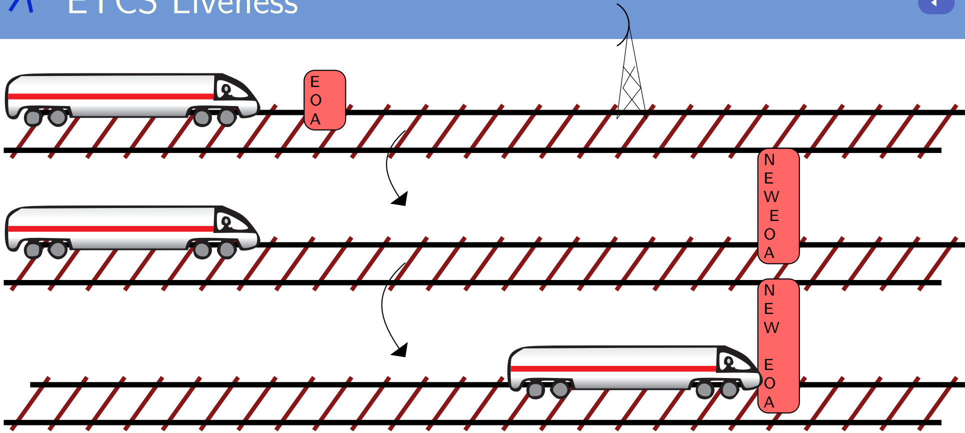
$$\equiv SB \geq \frac{\tau.v^2 - m.d^2}{2b} + \left(\frac{A}{b} + 1\right)\left(\frac{A}{2}\varepsilon^2 + \varepsilon\, \tau.v\right)$$

ETCS Safety



Proposition (Safety)

$$\tau.v^2 - m.d^2 \leq 2b(m.e - \tau.z) \rightarrow$$
$$[ETCS](\tau.z \geq m.e \rightarrow \tau.v \leq m.d)$$



Proposition (Liveness)

$$\tau.v > 0 \land \varepsilon > 0 \;\rightarrow\; \forall P \, \langle ETCS \rangle \, \tau.z \geq P$$

Safety Despite Disturbances

So far: no wind, friction, etc.

Direct control of the acceleration

Safety Despite Disturbances

So far: no wind, friction, etc.

Direct control of the acceleration

Issue

This is unrealistic!

Safety Despite Disturbances

So far: no wind, friction, etc.

Direct control of the acceleration

Issue

This is unrealistic!

Solution Take disturbances into account.

Theorem

ETCS is controllable, reactive, and safe in the presence of disturbances.

Safety Despite Disturbances

So far: no wind, friction, etc.

Direct control of the acceleration

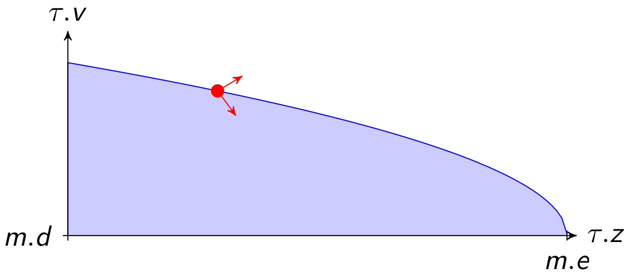
Issue

This is unrealistic!

Solution Take disturbances into account.

Theorem

ETCS is controllable, reactive, and safe in the presence of disturbances.



Safety Despite Disturbances

So far: no wind, friction, etc.
Direct control of the acceleration

Issue
This is unrealistic!

Solution Take disturbances into account.

Theorem
ETCS is controllable, reactive, and safe in the presence of disturbances.

Proof sketch
The system now contains $\tau.a - l \leq \tau.v' \leq \tau.a + u$ instead of $\tau.v' = \tau.a$.
$\leadsto$ We cannot solve the differential equations anymore.
$\leadsto$ Use differential invariants for approximation. For details see paper.

Platzer, A.:
Differential-algebraic dynamic logic for differential-algebraic programs.
J. Log. Comput., 35(1): 309–352, 2010.

Smooth Speed Control

So far

Almost completely non-deterministic control.

Smooth Speed Control

So far

Almost completely non-deterministic control.

Issue

This is unrealistic!

So far

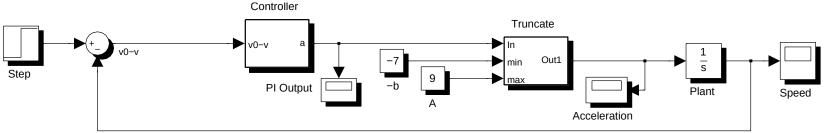
Almost completely non-deterministic control.

Issue

This is unrealistic!

Solution

Verify proportional-integral (PI) controllers used in trains.



So far

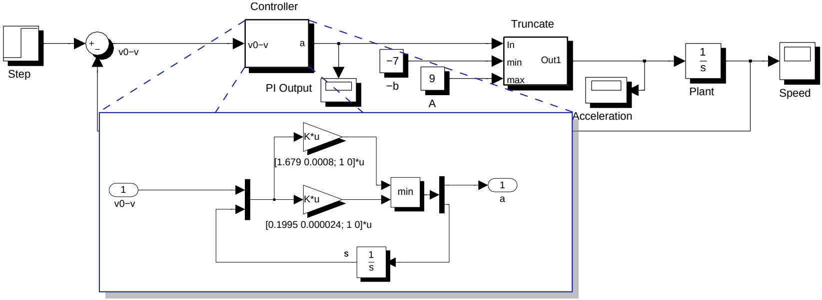
Almost completely non-deterministic control.

Issue

This is unrealistic!

Solution

Verify proportional-integral (PI) controllers used in trains.



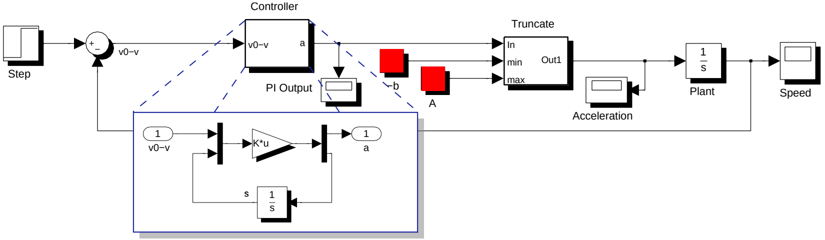
So far

Almost completely non-deterministic control.

Issue

This is unrealistic!

Solution Verify proportional-integral (PI) controllers used in trains.



Differential equation system

$$\tau.v' = \min\Big(A, \max\big(-b,\ \ell(\tau.v - m.r) - i\, s - c\, m.r\big)\Big) \land s' = \tau.v - m.r$$

Smooth Speed Control

So far

Almost completely non-deterministic control.

Issue

This is unrealistic!

Solution Verify proportional-integral (PI) controllers used in trains.

Theorem

The ETCS system remains safe when speed is controlled by a PI controller.

Proof sketch

Cannot solve differential equations really. Use differential invariants! For details see paper.

Platzer, A.:
Differential-algebraic dynamic logic for differential-algebraic programs.
J. Log. Comput., 35(1): 309–352, 2010.

Experimental Results (ETCS)

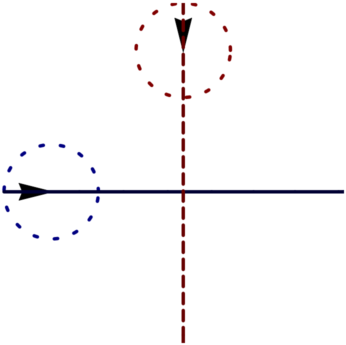
Case Study		Int	Time(s)	Mem(Mb)	Steps	Dim
controllability	train	0	0.6	6.9	14	5
controllability	RBC	0	0.5	6.4	42	12
controllability	RBC	0	0.9	6.5	82	12
reactivity		13	279.1	98.3	265	14
reactivity		0	103.9	61.7	47	14
safety		0	2052.4	204.3	153	14
liveness	essentials	4	35.2	92.2	62	10
liveness	simplified	6	9.6	23.5	134	13
controllability	disturbance	0	2.8	8.3	26	7
reactivity	disturbance	1	23.7	47.6	76	15
safety	disturbance	1	5805.2	34	218	16

provable automatically!

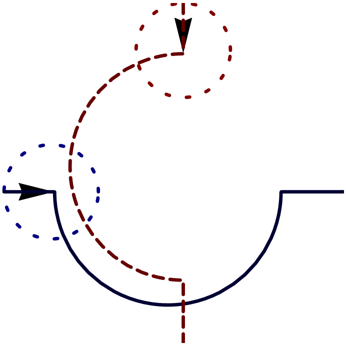
$$
\begin{aligned}
\text{spec} &: \tau.v^2 - \mathbf{m}.d^2 \le 2b(\mathbf{m}.e - \tau.p) \land \tau.v \ge 0 \land \mathbf{m}.d \ge 0 \land b > 0 \\
&\quad \to [\text{ETCS}](\tau.p \ge \mathbf{m}.e \to \tau.v \le \mathbf{m}.d) \\
\text{ETCS} &: (\text{train} \cup \text{rbc})^* \\
\text{train} &: \text{spd};\ \text{atp};\ \text{move} \\
\text{spd} &: (?\tau.v \le \mathbf{m}.r;\ \tau.a := *;\ ?-b \le \tau.a \le A) \\
&\quad \cup(?\tau.v \ge \mathbf{m}.r;\ \tau.a := *;\ ?0 > \tau.a \ge -b) \\
\text{atp} &: SB := \frac{\tau.v^2 - \mathbf{m}.d^2}{2b} + \left(\frac{A}{b} + 1\right)\left(\frac{A}{2}\varepsilon^2 + \varepsilon\,\tau.v\right); \\
&\quad (?(\mathbf{m}.e - \tau.p \le SB \lor rbc.message = emergency);\ \tau.a := -b) \\
&\quad \cup(?\mathbf{m}.e - \tau.p \ge SB \land rbc.message \ne emergency) \\
\text{move} &: t := 0;\ (\tau.p' = \tau.v, \tau.v' = \tau.a, t' = 1 \,\&\, \tau.v \ge 0 \land t \le \varepsilon) \\
\text{rbc} &: (rbc.message := emergency) \\
&\quad \cup \big(\mathbf{m}_0 := \mathbf{m}; \mathbf{m} := *; \\
&\qquad ?\mathbf{m}.r \ge 0 \land \mathbf{m}.d \ge 0 \land \mathbf{m}_0.d^2 - \mathbf{m}.d^2 \le 2b(\mathbf{m}.e - \mathbf{m}_0.e)\big)
\end{aligned}
$$

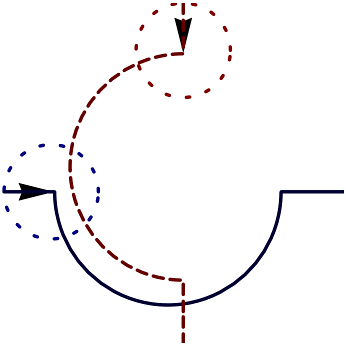
```
 state = 0,
 2 * b * (m - z) >= v ^ 2 - d ^ 2,
 v >= 0, d >= 0, v >= 0, ep >  0, b >  0, amax >  0, d >= 0
==>
    v <= vdes
 -> \forall R a_3;
      (   a_3 >= 0 & a_3 <= amax
       ->   (      m - z
               <= (amax / b + 1) * ep * v
                + (v ^ 2 - d ^ 2) / (2 * b)
                + (amax / b + 1) * amax * ep ^ 2 / 2
            -> \forall R t0;
                 (   t0 >= 0
                  -> \forall R ts0;  (0 <= ts0 & ts0 <= t0 -> -b * ts0 + v >= 0 & ts0 + 0 <= ep)
                  ->      2 * b * (m - 1 / 2 * (-b * t0 ^ 2 + 2 * t0 * v + 2 * z))
                       >= (-b * t0 + v) ^ 2
                        - d ^ 2
                     & -b * t0 + v >= 0
                     & d >= 0))
         & (      m - z
               >  (amax / b + 1) * ep * v
                + (v ^ 2 - d ^ 2) / (2 * b)
                + (amax / b + 1) * amax * ep ^ 2 / 2
            -> \forall R t2;
                 (   t2 >= 0
                  -> \forall R ts2;  (0 <= ts2 & ts2 <= t2 -> a_3 * ts2 + v >= 0 & ts2 + 0 <= ep)
                  ->      2 * b * (m - 1 / 2 * (a_3 * t2 ^ 2 + 2 * t2 * v + 2 * z))
                       >= (a_3 * t2 + v) ^ 2
                        - d ^ 2
                     & a_3 * t2 + v >= 0
                     & d >= 0)))
```

Outline Background



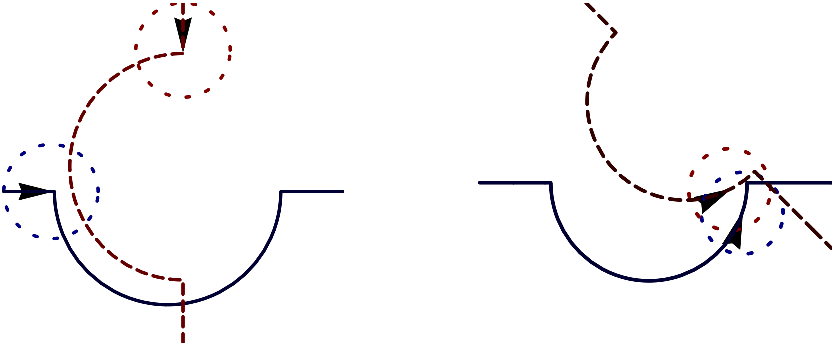
Air Traffic Control





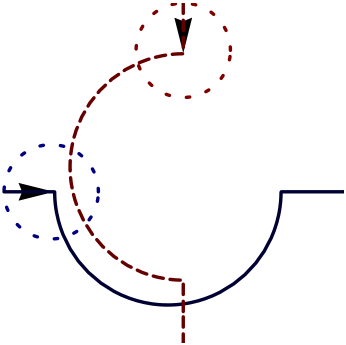
Verification?

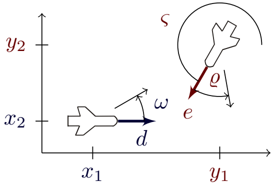
looks correct



Verification?

looks correct NO!

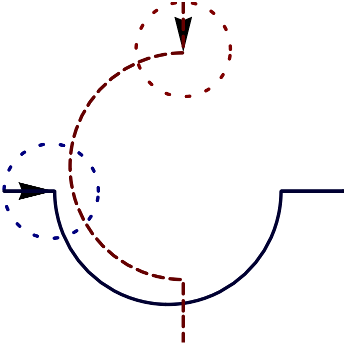


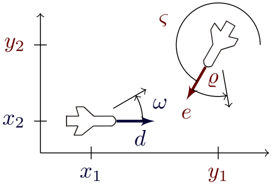


$$\begin{bmatrix} x_1' = -v_1 + v_2 \cos\vartheta + \omega x_2 \\ x_2' = v_2 \sin\vartheta - \omega x_1 \\ \vartheta' = \varpi - \omega \end{bmatrix}$$

Verification?

looks correct NO!





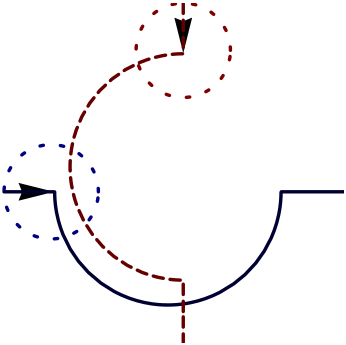
$$\begin{bmatrix} x_1' = -v_1 + v_2 \cos\vartheta + \omega x_2 \\ x_2' = v_2 \sin\vartheta - \omega x_1 \\ \vartheta' = \varpi - \omega \end{bmatrix}$$

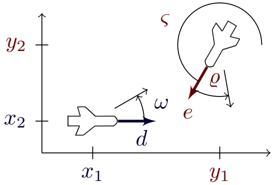
Example ("Solving" differential equations)

$$x_1(t) = \frac{1}{\omega\varpi}\big(x_1\omega\varpi\cos t\omega - v_2\omega\cos t\omega\sin\vartheta + v_2\omega\cos t\omega\cos t\varpi\sin\vartheta - v_1\varpi\sin t\omega$$

$$+ x_2\omega\varpi\sin t\omega - v_2\omega\cos\vartheta\cos t\varpi\sin t\omega - v_2\omega\sqrt{1 - \sin\vartheta^2}\sin t\omega$$

$$+ v_2\omega\cos\vartheta\cos t\omega\sin t\varpi + v_2\omega\sin\vartheta\sin t\omega\sin t\varpi\big)\ldots$$





$$\begin{bmatrix} x_1' = -v_1 + v_2 \cos\vartheta + \omega x_2 \\ x_2' = v_2 \sin\vartheta - \omega x_1 \\ \vartheta' = \varpi - \omega \end{bmatrix}$$

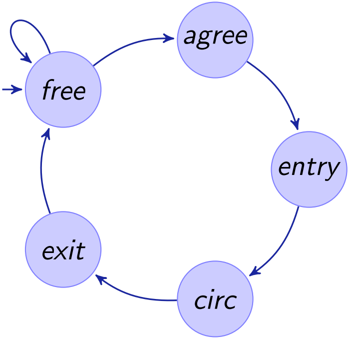
Example ("Solving" differential equations)

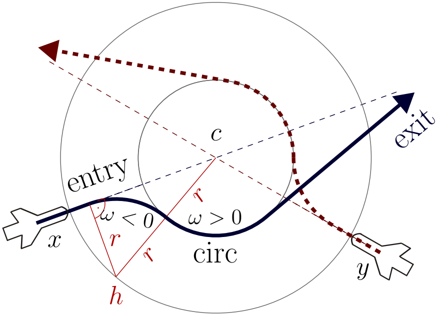
$$\forall t \geq 0 \quad \frac{1}{\omega\varpi}\Big(x_1\omega\varpi \cos t\omega - v_2\omega \cos t\omega \sin\vartheta + v_2\omega \cos t\omega \cos t\varpi \sin\vartheta - v_1\varpi \sin t\omega$$

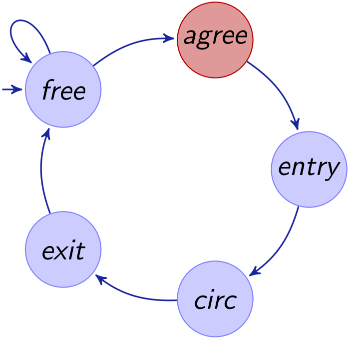
$$+ x_2\omega\varpi \sin t\omega - v_2\omega \cos\vartheta \cos t\varpi \sin t\omega - v_2\omega\sqrt{1 - \sin\vartheta^2} \sin t\omega$$

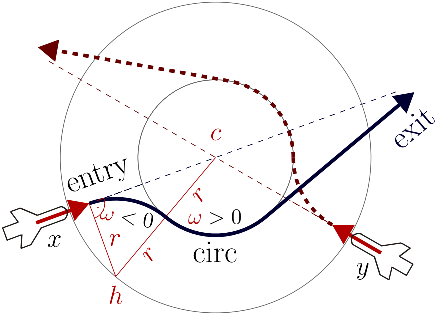
$$+ v_2\omega \cos\vartheta \cos t\omega \sin t\varpi + v_2\omega \sin\vartheta \sin t\omega \sin t\varpi\Big) \ldots$$

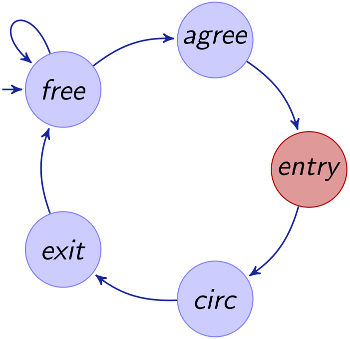
Flyable Roundabout Maneuver: Overview

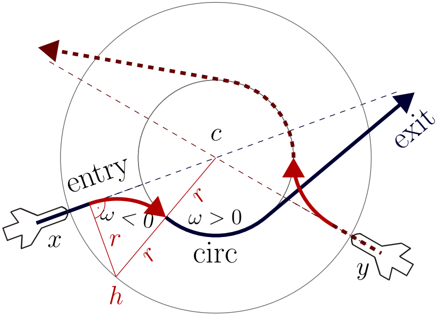


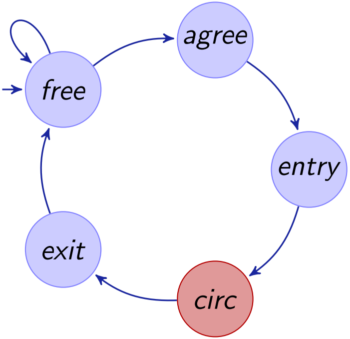


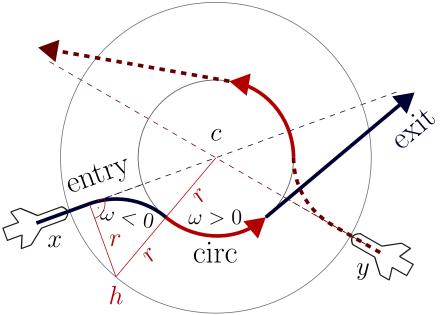

agree
free
entry
exit
circ


c
entry
exit
$\omega < 0$
$\omega > 0$
r
x
y
circ
h

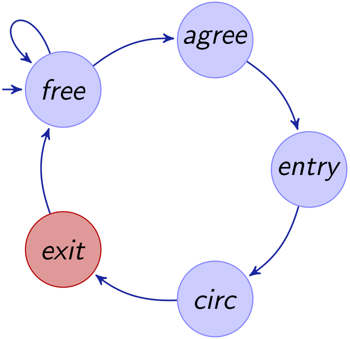

agree
free
entry
exit
circ

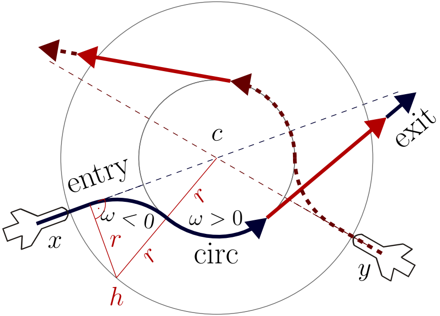

c
entry
exit
x
y
r
r
r
h
$\omega < 0$
$\omega > 0$
circ

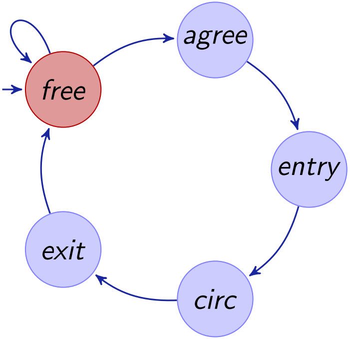

agree
free
entry
exit
circ

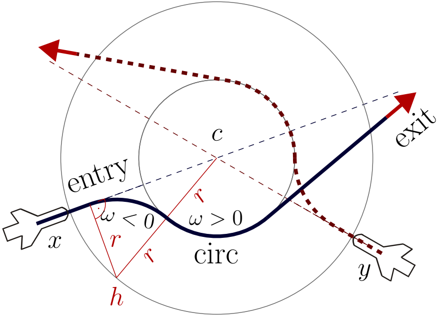

c
entry
exit
$\omega < 0$
$\omega > 0$
r
r
r
x
y
h
circ

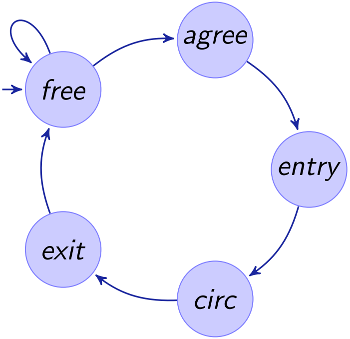
Flyable Roundabout Maneuver: Overview



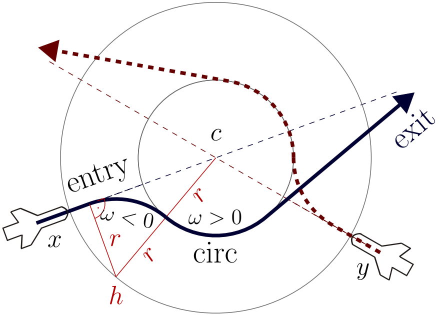



agree
free
entry
exit
circ

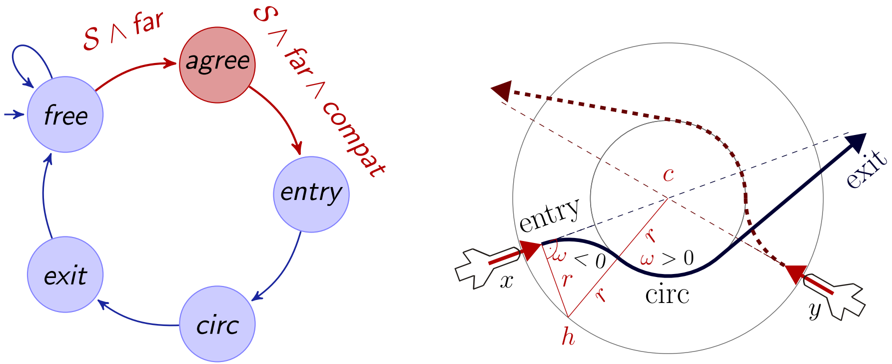

c
entry
exit
$\omega < 0$
$\omega > 0$
r
r
r
x
y
circ
h



agree
free
entry
exit
circ

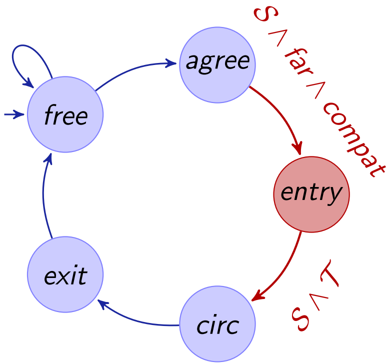


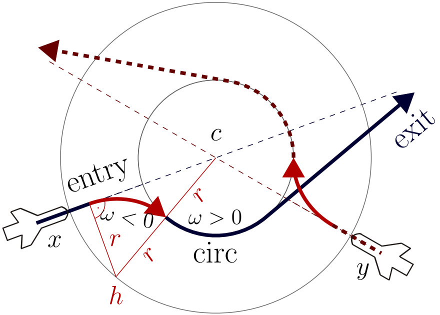
c
entry
exit
r
$\omega < 0$
$\omega > 0$
x
r
r
circ
y
h



Example (dℒ formula of verification subgoal)

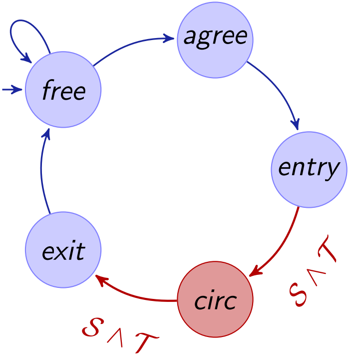
$$\mathit{safe} \land \mathit{far} \;\rightarrow\; [\mathit{agree}](\mathit{safe} \land \mathit{far} \land \mathit{compatible})$$

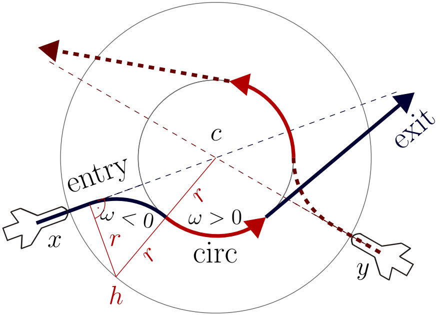




Example (d$\mathcal{L}$ formula of verification subgoal)

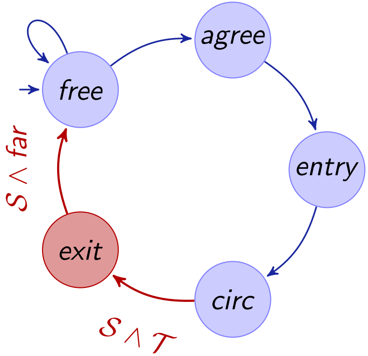
$$\mathit{safe} \wedge \mathit{far} \wedge \mathit{compatible} \rightarrow [\mathit{entry}](\mathit{safe} \wedge \mathit{tangential})$$

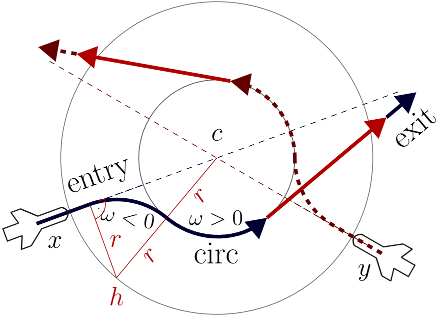




Example (d$\mathcal{L}$ formula of verification subgoal)

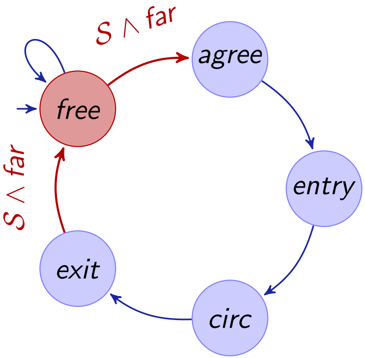
$$\mathit{safe} \wedge \mathit{tangential} \rightarrow [\mathit{circ}](\mathit{safe} \wedge \mathit{tangential})$$

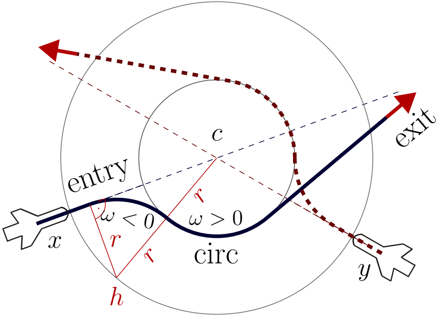




Example (d$\mathcal{L}$ formula of verification subgoal)

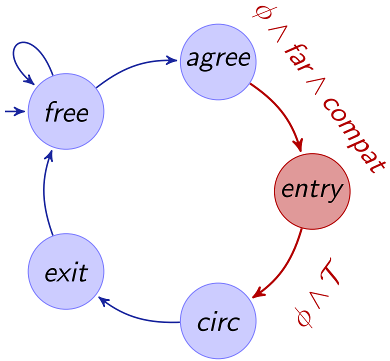
$$\mathit{safe} \land \mathit{tangential} \ \rightarrow\ [\mathit{exit}](\mathit{safe} \land \mathit{far})$$

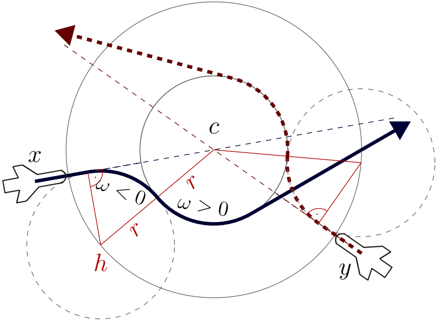




Example (d$\mathcal{L}$ formula of verification subgoal)

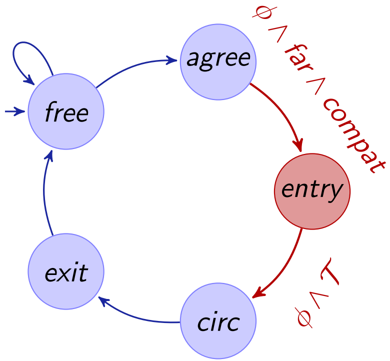
$$safe \land far \rightarrow [free](safe \land far)$$

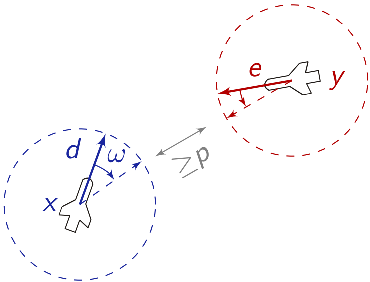




Example (d$\mathcal{L}$ formula of verification subgoal)

$$(r\omega)^2 = \|d\|^2 \wedge \|x - c\| = \sqrt{3}r \wedge \exists \lambda \geq 0\, (x + \lambda d = c) \wedge$$
$$\|h - c\| = 2r \wedge d = -\omega(x - h)^\perp$$
$$\rightarrow [\mathcal{F}(-\omega)\, \&\, \|x - c\| \geq r] \left(\|x - c\| \leq r \rightarrow d = \omega(x - c)^\perp\right)$$

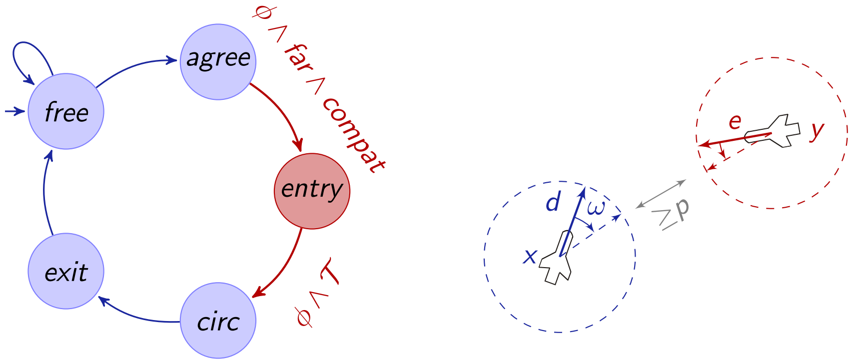




Example (d$\mathcal{L}$ formula of verification subgoal)

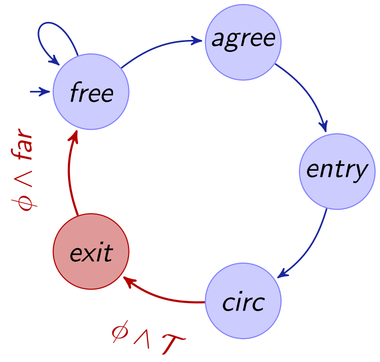
$$\|x - y\| \geq \sqrt{2}(p + 2bT) \wedge p \geq 0 \wedge \|d\|^2 \leq \|e\|^2 \leq b^2 \wedge b \geq 0 \wedge T \geq 0 \rightarrow [entry]\,(\|x - y\| \geq p)$$

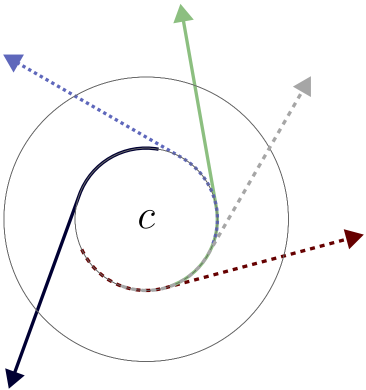
Flyable Roundabout Maneuver: Entry



Example (d$\mathcal{L}$ formula of verification subgoal)

$$x = z \land \|d\|^2 \le b^2 \land b \ge 0 \rightarrow [\tau := 0;\ \exists \omega\, \mathcal{F}(\omega) \land \tau' = 1]\,(\|x - z\|_\infty \le \tau b)$$

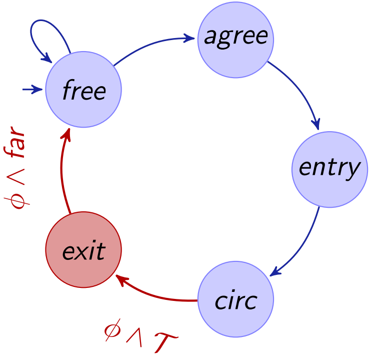


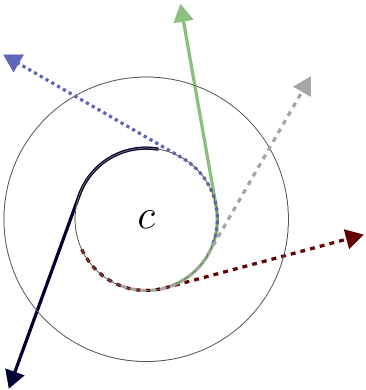


Example (d$\mathcal{L}$ formula of verification subgoal)

$$\mathcal{T} \wedge \|x - y\|^2 \geq p^2 \rightarrow [x' = d \wedge y' = e] (\|x - y\|^2 \geq p^2)$$

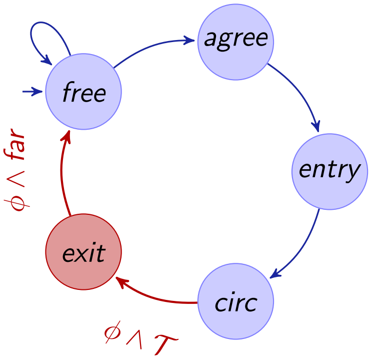
Flyable Roundabout Maneuver: Exit

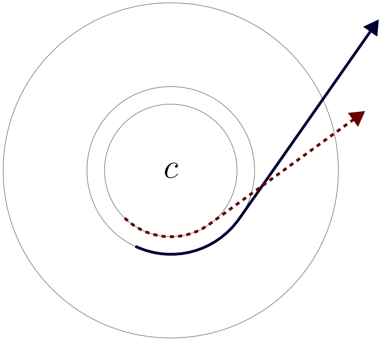




Example (d$\mathcal{L}$ formula of verification subgoal)

$$\mathcal{T} \wedge \|x - y\|^2 \geq p^2 \rightarrow [x' = d;\ y' = e]\,(\|x - y\|^2 \geq p^2)$$

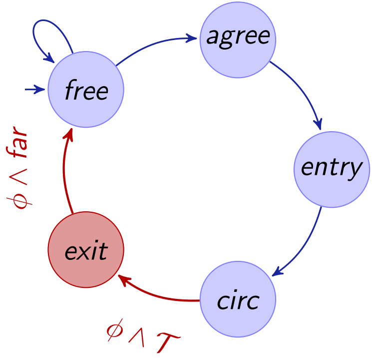


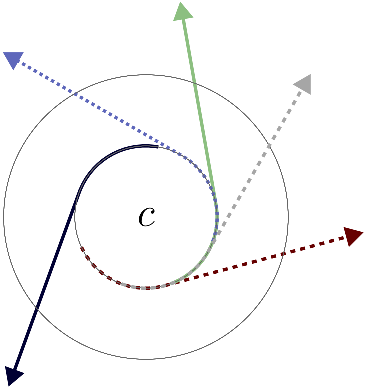


Example (d$\mathcal{L}$ formula of verification subgoal)

$$\mathcal{T} \wedge \|x - y\|^2 \geq p^2 \rightarrow [x' = d;\ y' = e]\,(\|x - y\|^2 \geq p^2)$$

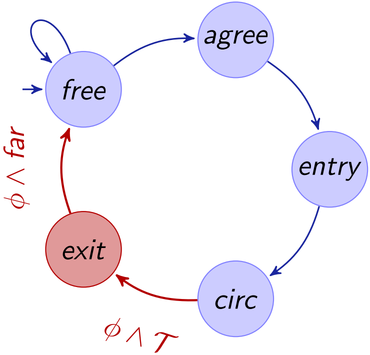
Flyable Roundabout Maneuver: Exit

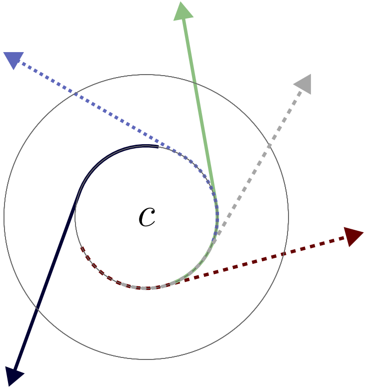




Example (d$\mathcal{L}$ formula of verification subgoal)

$$\mathcal{T} \wedge \|x - y\|^2 \geq p^2 \rightarrow [x' = d;\ y' = e]\,(\|x - y\|^2 \geq p^2)$$





Example (d$\mathcal{L}$ formula of verification subgoal)

$$\mathcal{T} \wedge d \neq e \rightarrow \forall a \, \langle x' = d \wedge y' = e \rangle \, (\|x - y\|^2 > a^2)$$

provable automatically!

$$
\begin{aligned}
\psi &\equiv \phi \to [trm^*]\phi \\
\phi &\equiv \|x-y\|^2 \geq p^2 \equiv (x_1-y_1)^2 + (x_2-y_2)^2 \geq p^2 \\
trm &\equiv free;\quad entry;\quad \mathcal{F}(\omega) \land \mathcal{G}(\omega) \\
free &\equiv \exists\omega\, \mathcal{F}(\omega) \land \exists\varpi\, \mathcal{G}(\varpi) \land \phi \\
entry &\equiv \exists u\, \omega := u;\ \exists c\, (d := \omega(x-c)^\perp \land e := \omega(y-c)^\perp)
\end{aligned}
$$

$$
\mathcal{F}(\omega) \equiv \begin{pmatrix} x_1' = v\cos\vartheta = d_1 \\ \land\, x_2' = v\sin\vartheta = d_2 \\ \land\, d_1' = v(-\sin\vartheta)\vartheta' = -\omega d_2 \\ \land\, d_2' = v(\cos\vartheta)\vartheta' = \omega d_1 \end{pmatrix} \qquad \mathcal{G}(\varpi) \equiv \begin{pmatrix} y_1' = e_1 \\ \land\, y_2' = e_2 \\ \land\, e_1' = -\varpi e_2 \\ \land\, e_2' = \varpi e_1 \end{pmatrix}
$$

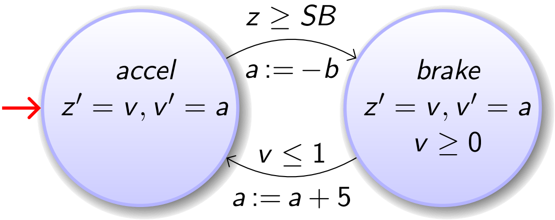
provable automatically!

$$
\begin{aligned}
\psi &\equiv \phi \to [trm^*]\phi \\
\phi &\equiv (x_1 - y_1)^2 + (x_2 - y_2)^2 \geq p^2 \land (y_1 - z_1)^2 + (y_2 - z_2)^2 \geq p^2 \\
&\quad \land (x_1 - z_1)^2 + (x_2 - z_2)^2 \geq p^2 \land (x_1 - u_1)^2 + (x_2 - u_2)^2 \geq p^2 \\
&\quad \land (y_1 - u_1)^2 + (y_2 - u_2)^2 \geq p^2 \land (z_1 - u_1)^2 + (z_2 - u_2)^2 \geq p^2 \\
trm &\equiv free;\ \ entry; \\
&\quad x_1' = d_1 \land x_2' = d_2 \land d_1' = -\omega_x d_2 \land d_2' = \omega_x d_1 \\
&\quad \land y_1' = e_1 \land y_2' = e_2 \land e_1' = -\omega_y e_2 \land e_2' = \omega_y e_1 \\
&\quad \land z_1' = f_1 \land z_2' = f_2 \land f_1' = -\omega_z f_2 \land f_2' = \omega_z f_1 \\
&\quad \land u_1' = g_1 \land u_2' = g_2 \land g_1' = -\omega_u g_2 \land g_2' = \omega_u g_1 \\
free &\equiv (\omega_x := *;\ \ \omega_y := *;\ \ \omega_z := *;\ \ \omega_u := *; \\
&\quad x_1' = d_1 \land x_2' = d_2 \land d_1' = -\omega_x d_2 \land d_2' = \omega_x d_1 \\
&\quad \land y_1' = e_1 \land y_2' = e_2 \land e_1' = -\omega_y e_2 \land e_2' = \omega_y e_1 \\
&\quad \land z_1' = f_1 \land z_2' = f_2 \land f_1' = -\omega_z f_2 \land f_2' = \omega_z f_1 \\
&\quad \land u_1' = g_1 \land u_2' = g_2 \land g_1' = -\omega_u g_2 \land g_2' = \omega_u g_1 \land \phi)^* \\
entry &\equiv \omega := *;\ c := *; \\
&\quad d_1 := -\omega(x_2 - c_2);\ d_2 := \omega(x_1 - c_1); \\
&\quad e_1 := -\omega(y_1 - c_1);\ e_2 := \omega(y_2 - c_2); \\
&\quad f_1 := -\omega(z_1 - c_1);\ f_2 := \omega(z_2 - c_2); \\
&\quad g_1 := -\omega(u_1 - c_1);\ g_2 := \omega(u_2 - c_2)
\end{aligned}
$$

Experimental Results (Air Traffic Control)

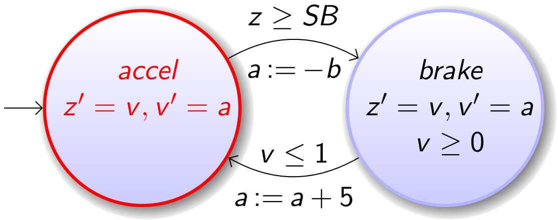
Case Study	Time(s)	Mem(Mb)	Steps	Dim
tangential roundabout (2a/c)	10.4	6.8	197	13
tangential roundabout (3a/c)	253.6	7.2	342	18
tangential roundabout (4a/c)	382.9	10.2	520	23
tangential roundabout (5a/c)	1882.9	39.1	735	28
bounded maneuver speed	0.5	6.3	14	4
flyable roundabout entry*	10.1	9.6	132	8
flyable entry feasible*	104.5	87.9	16	10
flyable entry circular	3.2	7.6	81	5
limited entry progress	1.9	6.5	60	8
entry separation	140.1	20.1	512	16
mutual negotiation successful	0.8	6.4	60	12
mutual negotiation feasible*	7.5	23.8	21	11
mutual far negotiation	2.4	8.1	67	14
simultaneous exit separation*	4.3	12.9	44	9
different exit directions	3.1	11.1	42	11

Outline Background



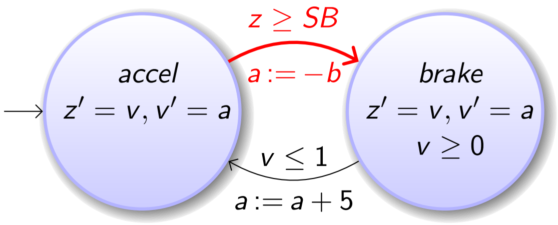
$\rightsquigarrow$

$$
\begin{aligned}
& q := accel; \\
& \big(\quad (?q = accel;\quad z' = v, v' = a) \\
& \cup\ (?q = accel \land z \geq SB;\quad a := -b;\quad q := brake;\quad ?v \geq 0) \\
& \cup\ (?q = brake;\quad z' = v, v' = a \,\&\, v \geq 0) \\
& \cup\ (?q = brake \land v \leq 1;\quad a := a + 5;\quad q := accel)\big)^*
\end{aligned}
$$



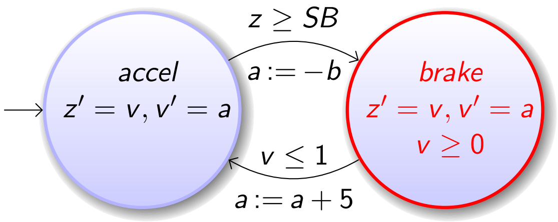
$\rightsquigarrow$

$$
\begin{aligned}
&q := accel;\\
&\big(\ \ (?q = accel;\ \ z' = v, v' = a)\\
&\cup\ (?q = accel \land z \geq SB;\ \ a := -b;\ \ q := brake;\ \ ?v \geq 0)\\
&\cup\ (?q = brake;\ \ z' = v, v' = a \,\&\, v \geq 0)\\
&\cup\ (?q = brake \land v \leq 1;\ \ a := a + 5;\ \ q := accel)\big)^*
\end{aligned}
$$

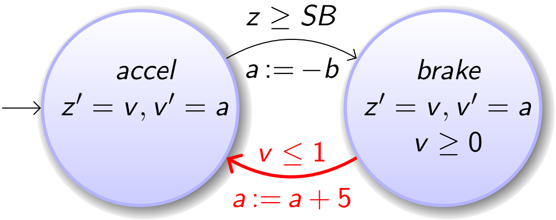


$\rightsquigarrow$

$$
\begin{aligned}
&q := accel;\\
&\big(\ \ (?q = accel;\ \ z' = v, v' = a)\\
&\cup\ (?q = accel \land z \geq SB;\ \ a := -b;\ \ q := brake;\ \ ?v \geq 0)\\
&\cup\ (?q = brake;\ \ z' = v, v' = a \,\&\, v \geq 0)\\
&\cup\ (?q = brake \land v \leq 1;\ \ a := a + 5;\ \ q := accel)\big)^*
\end{aligned}
$$



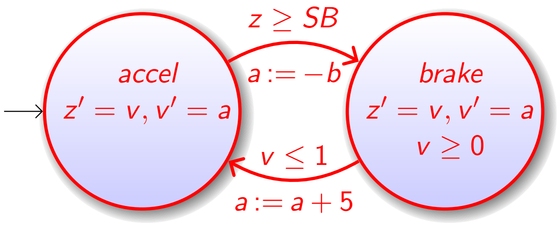
$$
\begin{aligned}
&q := accel;\\
&(\quad (?q = accel; \quad z' = v, v' = a)\\
&\cup\ (?q = accel \wedge z \geq SB; \quad a := -b; \quad q := brake; \quad ?v \geq 0)\\
&\cup\ (?q = brake; \quad z' = v, v' = a \,\&\, v \geq 0)\\
&\cup\ (?q = brake \wedge v \leq 1; \quad a := a + 5; \quad q := accel))^*
\end{aligned}
$$



$\rightsquigarrow$

$$
\begin{aligned}
& q := accel;\\
& (\quad (?q = accel;\quad z' = v, v' = a)\\
& \cup\ (?q = accel \land z \geq SB;\quad a := -b;\quad q := brake;\quad ?v \geq 0)\\
& \cup\ (?q = brake;\quad z' = v, v' = a \,\&\, v \geq 0)\\
& \cup\ (?q = brake \land v \leq 1;\quad a := a + 5;\quad q := accel)\big)^*
\end{aligned}
$$

Embedding Hybrid Automata as Hybrid Programs



$$\rightsquigarrow$$

$$\begin{aligned}
&q := accel;\\
&(\quad (?q = accel;\quad z' = v, v' = a)\\
&\cup\ (?q = accel \land z \geq SB;\quad a := -b;\quad q := brake;\quad ?v \geq 0)\\
&\cup\ (?q = brake;\quad z' = v, v' = a \,\&\, v \geq 0)\\
&\cup\ (?q = brake \land v \leq 1;\quad a := a + 5;\quad q := accel)\big)^*
\end{aligned}$$

Outline Background

Cyber-Physical Systems:

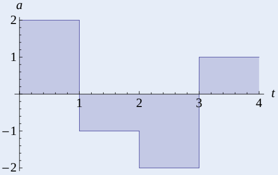
Q: I want to verify my car

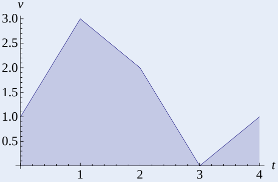
Challenge

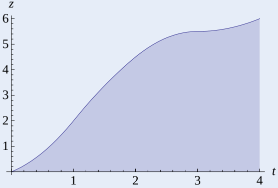
Q: I want to verify my car A: Hybrid systems

Challenge (Hybrid Systems)

- Continuous dynamics (differential equations)
- Discrete dynamics (control decisions)



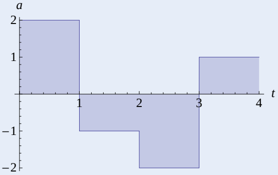


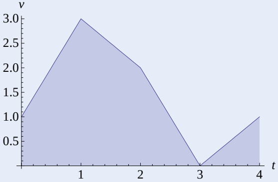


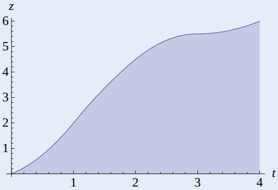
Q: I want to verify my car A: Hybrid systems Q: But there's a lot of cars!

Challenge (Hybrid Systems)

- Continuous dynamics (differential equations)
- Discrete dynamics (control decisions)



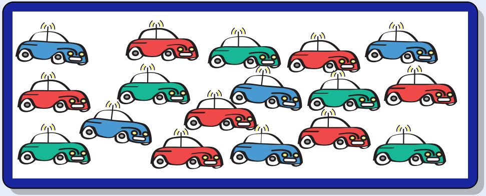




Cyber-Physical Systems:

Q: I want to verify a lot of cars

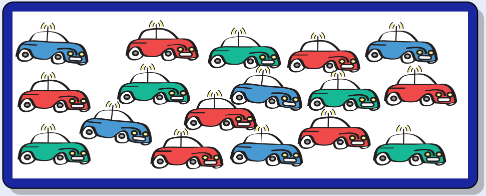
Challenge

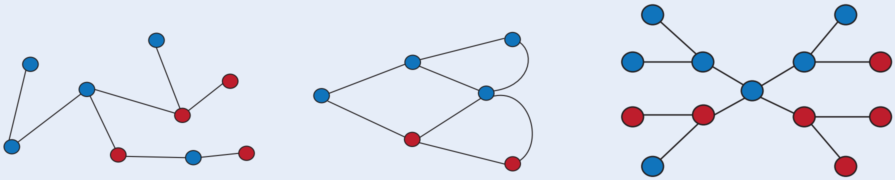


Q: I want to verify a lot of cars A: Distributed systems

Challenge (Distributed Systems)

- Local computation (finite state automaton)
- Remote communication (network graph)



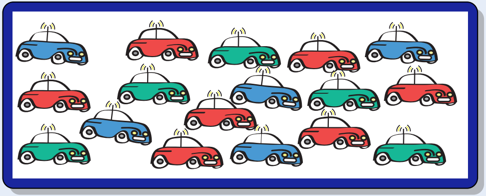


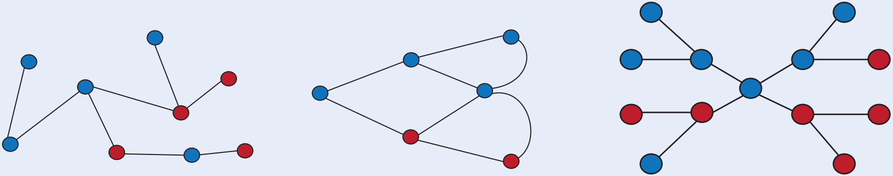
Cyber-Physical Systems: Distributed Systems

Q: I want to verify a lot of cars A: Distributed systems Q: But they move!

Challenge (Distributed Systems)

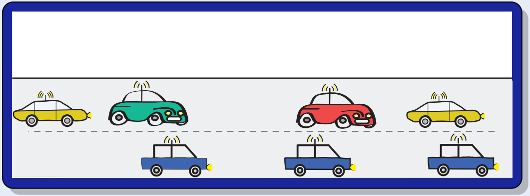
- Local computation (finite state automaton)
- Remote communication (network graph)





Q: I want to verify lots of moving cars

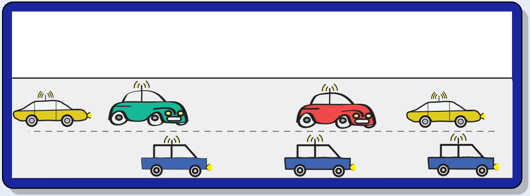
Challenge



Q: I want to verify lots of moving cars A: Distributed hybrid systems

Challenge (Distributed Hybrid Systems)

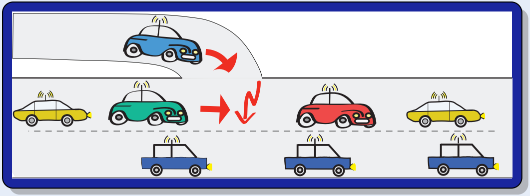
- Continuous dynamics (differential equations)
- Discrete dynamics (control decisions)
- Structural dynamics (communication/coupling)



Q: I want to verify lots of moving cars A: Distributed hybrid systems

Challenge (Distributed Hybrid Systems)

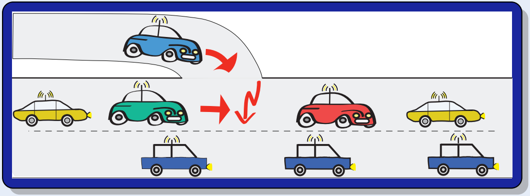
- Continuous dynamics (differential equations)
- Discrete dynamics (control decisions)
- Structural dynamics (communication/coupling)
- Dimensional dynamics (appearance)



Q: I want to verify lots of moving cars A: Distributed hybrid systems Q: How?

Challenge (Distributed Hybrid Systems)

- Continuous dynamics (differential equations)
- Discrete dynamics (control decisions)
- Structural dynamics (communication/coupling)
- Dimensional dynamics (appearance)

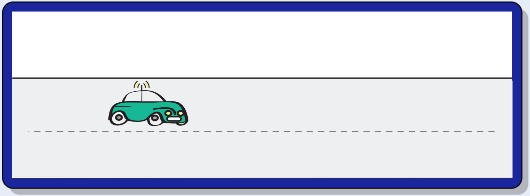


Model for Distributed Hybrid Systems

Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

- Continuous dynamics (differential equations)
- Discrete dynamics (control decisions)
- Structural dynamics (communication/coupling)

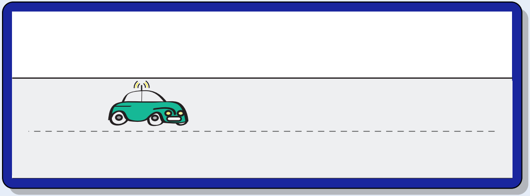


Model for Distributed Hybrid Systems

Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

- Continuous dynamics (differential equations)
  $x'' = a$
- Discrete dynamics (control decisions)
- Structural dynamics (communication/coupling)



Model for Distributed Hybrid Systems

Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

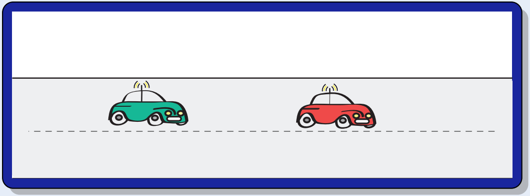
- Continuous dynamics (differential equations)

  $$x'' = a$$

- Discrete dynamics (control decisions)

  $a := \texttt{if} .. \texttt{then}\, a\, \texttt{else}\, -b\, \texttt{fi}$

- Structural dynamics (communication/coupling)

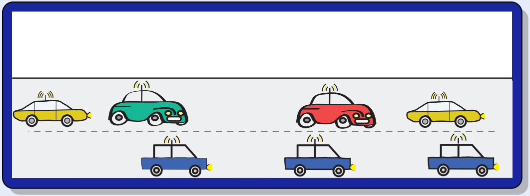


Model for Distributed Hybrid Systems

Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

- Continuous dynamics (differential equations)
  $$x'' = a$$
- Discrete dynamics (control decisions)
  $$a := \texttt{if} \,..\, \texttt{then}\, a \,\texttt{else}\, -b \,\texttt{fi}$$
- Structural dynamics (communication/coupling)

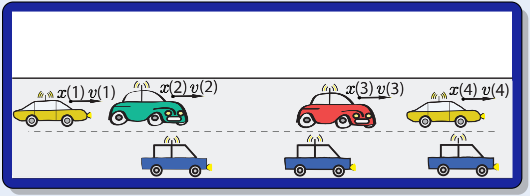


Model for Distributed Hybrid Systems

Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

- Continuous dynamics (differential equations)
  $x'' = a$
- Discrete dynamics (control decisions)
  $a := \texttt{if} .. \texttt{then}\, a\, \texttt{else} -b\, \texttt{fi}$
- Structural dynamics (communication/coupling)



Model for Distributed Hybrid Systems

Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

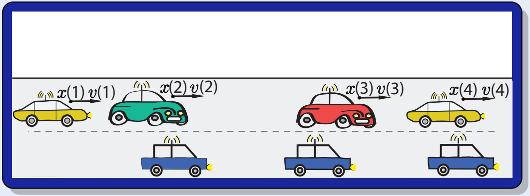
- Continuous dynamics (differential equations)

  $$x(i)'' = a(i)$$

- Discrete dynamics (control decisions)

  $$a(i) := \texttt{if} \, .. \, \texttt{then} \, a \, \texttt{else} \, -b \, \texttt{fi}$$

- Structural dynamics (communication/coupling)



Model for Distributed Hybrid Systems

Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

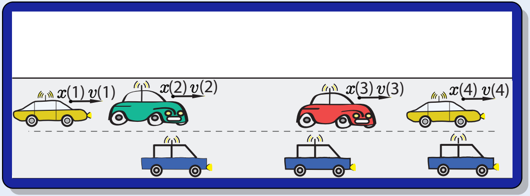
- Continuous dynamics (differential equations)

  $\forall i\, x(i)'' = a(i)$

- Discrete dynamics (control decisions)

$\forall i\, a(i) := \texttt{if} .. \texttt{then}\, a\, \texttt{else} -b\, \texttt{fi}$

- Structural dynamics (communication/coupling)



Model for Distributed Hybrid Systems

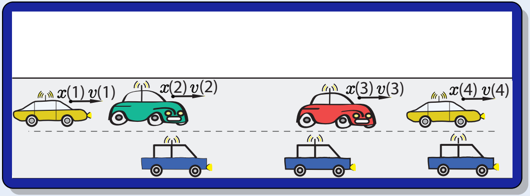
Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

- Continuous dynamics (differential equations)
  $\forall i\; x(i)'' = a(i)$
- Discrete dynamics (control decisions)

$\forall i\; a(i) := \texttt{if} .. \texttt{then}\; a\; \texttt{else}\; -b\; \texttt{fi}$

- Structural dynamics (communication/coupling)
  $\ell(i) := carInFrontOf(i)$



Model for Distributed Hybrid Systems

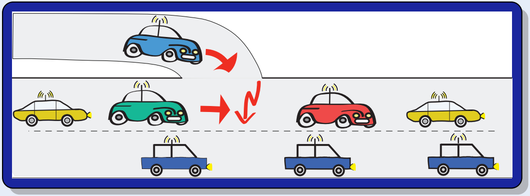
Q: How to model distributed hybrid systems A: Quantified Hybrid Programs

Model (Distributed Hybrid Systems)

- Continuous dynamics (differential equations)
  $\forall i\, x(i)'' = a(i)$
- Discrete dynamics (control decisions)

$\forall i\, a(i) := \texttt{if}\, ..\, \texttt{then}\, a\, \texttt{else}\, -b\, \texttt{fi}$

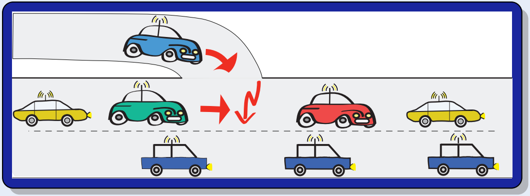
- Structural dynamics (communication/coupling)
  $\ell(i) := \mathit{carInFrontOf(i)}$
- Dimensional dynamics (appearance)



Model for Distributed Hybrid Systems

Q: How to model distributed hybrid systems A: Quantified Hybrid Programs

Model (Distributed Hybrid Systems)



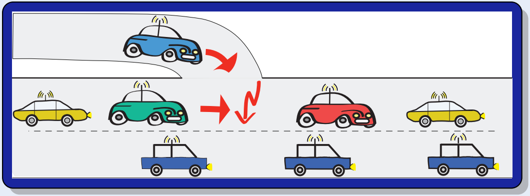
- Continuous dynamics (differential equations)
  $\forall i\, x(i)'' = a(i)$
- Discrete dynamics (control decisions)
  $\forall i\, a(i) := \texttt{if} .. \texttt{then}\, a\, \texttt{else}\, -b\, \texttt{fi}$
- Structural dynamics (communication/coupling)
  $\ell(i) := carInFrontOf(i)$
- Dimensional dynamics (appearance)
  $n := \texttt{new}\ Car$

Model for Distributed Hybrid Systems

Q: How to model distributed hybrid systems A: Quantified Hybrid Programs

Model (Distributed Hybrid Systems)

- Continuous dynamics (differential equations)
  $\forall i\, x(i)'' = a(i)$
- Discrete dynamics (control decisions)
  $\forall i\, a(i) := \mathtt{if} .. \mathtt{then}\, a\, \mathtt{else} -b\, \mathtt{fi}$
- Structural dynamics (communication/coupling)
  $\ell(i) := carInFrontOf(i)$
- Dimensional dynamics (appearance)
  $n := \mathtt{new}\ Car$



$\Rightarrow$ Communication
$d(i, \ell(i)) := d(i, \ell(i)) + 10$

Model for Distributed Hybrid Systems

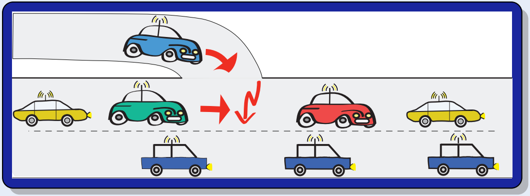
Q: How to model distributed hybrid systems A: Quantified Hybrid Programs

Model (Distributed Hybrid Systems)

- Continuous dynamics (differential equations)
  $\forall i\; x(i)'' = a(i)$
- Discrete dynamics (control decisions)

$\forall i\; a(i) := \texttt{if} .. \texttt{then}\; a\; \texttt{else}\; -b\; \texttt{fi}$

- Structural dynamics (communication/coupling)
  $\ell(i) := carInFrontOf(i)$
- Dimensional dynamics (appearance)
  $n := \texttt{new}\; Car$



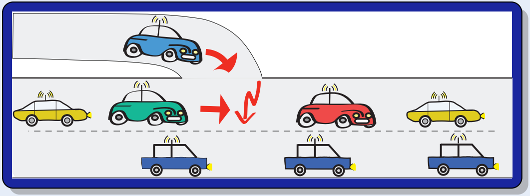
$\Rightarrow$ Communication

$\forall i\; d(i, \ell(i)) := d(i, \ell(i)) + 10$

Model for Distributed Hybrid Systems

Q: How to model distributed hybrid systems A: Quantified Hybrid Programs

Model (Distributed Hybrid Systems)



- Continuous dynamics (differential equations)
  $\forall i\, x(i)'' = a(i)$
- Discrete dynamics (control decisions)
  $\forall i\, a(i) := \mathtt{if}\ ..\ \mathtt{then}\ a\ \mathtt{else}\ -b\ \mathtt{fi}$
- Structural dynamics (communication/coupling)
  $\ell(i) := carInFrontOf(i)$
- Dimensional dynamics (appearance)
  $n := \mathtt{new}\ Car$

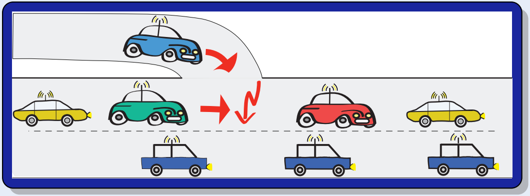
$\Rightarrow$ Communication
$\forall i\, d(i, \ell(i)) := d(i, \ell(i)) + 10$

$\Rightarrow$ Discrete structural dynamics
$\ell(i) := \ell(\ell(i))$

Model for Distributed Hybrid Systems

Q: How to model distributed hybrid systems A: Quantified Hybrid Programs

Model (Distributed Hybrid Systems)



- Continuous dynamics (differential equations)
  $\forall i\, x(i)'' = a(i)$
- Discrete dynamics (control decisions)
  $\forall i\, a(i) := \texttt{if .. then } a \texttt{ else } -b \texttt{ fi}$
- Structural dynamics (communication/coupling)
  $\ell(i) := carInFrontOf(i)$
- Dimensional dynamics (appearance)
  $n := \texttt{new } Car$

$\Rightarrow$ Communication
$$\forall i\, d(i,\ell(i)) := d(i,\ell(i)) + 10$$

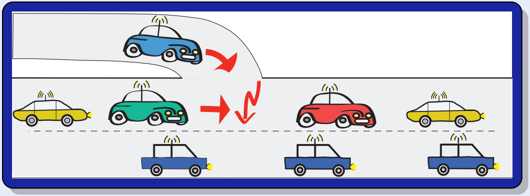
$\Rightarrow$ Discrete structural dynamics
$$\ell(i) := \ell(\ell(i))$$

$\Rightarrow$ Continuous structural dynamics
$$x(i)'' = a(i) + c(i,\ell(i))a(\ell(i))$$

Model for Distributed Hybrid Systems

Q: How to model distributed hybrid systems A: Quantified Hybrid Programs

Model (Distributed Hybrid Systems)



- Continuous dynamics (differential equations)
  $\forall i\, x(i)'' = a(i)$
- Discrete dynamics (control decisions)

$\forall i\, a(i) := \texttt{if} .. \texttt{then}\, a\, \texttt{else} -b\, \texttt{fi}$

- Structural dynamics (communication/coupling)
  $\ell(i) := carInFrontOf(i)$
- Dimensional dynamics (appearance)
  $n := \texttt{new}\ Car$

$\Rightarrow$ Communication
$\forall i\, d(i,\ell(i)) := d(i,\ell(i)) + 10$

$\Rightarrow$ Discrete structural dynamics
$\ell(i) := \ell(\ell(i))$

$\Rightarrow$ Continuous structural dynamics
$\forall i\, x(i)'' = a(i) + c(i,\ell(i))a(\ell(i))$

Related Work

Shift [16] The Hybrid System Simulation Programming Language

Hybrid CSP [18] Semantics in Extended Duration Calculus

HyPA [19] Translate fragment into normal form.

χ process algebra [20] Simulation, translation of fragments to PHAVER, UPPAAL

R-Charon [17] Modeling Language for Reconfigurable Hybrid Systems

Φ-calculus [21] Semantics in rich set theory

ACP_{hs}^{srt} [22] Modeling language proposal

OBSHS [23] Partial random simulation of objects

Related Work

Shift [16] The Hybrid System Simulation Programming Language

Hybrid CSP [18] Semantics in Extended Duration Calculus

HyPA [19] Translate fragment into normal form.

χ process algebra [20] Simulation, translation of fragments to PHAVER, UPPAAL

R-Charon [17] Modeling Language for Reconfigurable Hybrid Systems

Φ-calculus [21] Semantics in rich set theory

ACP^{srt}_{hs} [22] Modeling language proposal

OBSHS [23] Partial random simulation of objects

Definition (Quantified hybrid program α)

$\forall i : C\ x(i)' = \theta$	(quantified ODE)	
$\forall i : C\ x(i) := \theta$	(quantified assignment)	jump & test
$?\chi$	(conditional execution)	jump & test
$\alpha; \beta$	(seq. composition)	Kleene algebra
$\alpha \cup \beta$	(nondet. choice)	Kleene algebra
α^*	(nondet. repetition)	Kleene algebra

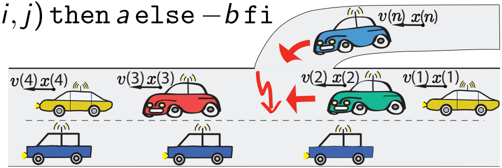
Definition (Quantified hybrid program α)

$\forall i : C\ x(i)' = \theta$	(quantified ODE)	
$\forall i : C\ x(i) := \theta$	(quantified assignment)	jump & test
$?\chi$	(conditional execution)	
$\alpha; \beta$	(seq. composition)	Kleene algebra
$\alpha \cup \beta$	(nondet. choice)	
α^*	(nondet. repetition)	

$$DCCS \equiv (ctrl; drive)^*$$

$$ctrl \equiv \forall i : C\ a(i) := \mathtt{if}\ \forall j : C\ far(i,j)\ \mathtt{then}\ a\ \mathtt{else}\ -b\ \mathtt{fi}$$

$$drive \equiv \forall i : C\ x(i)'' = a(i)$$



Quantified Differential Dynamic Logic Qd$\mathcal{L}$: Syntax

Definition (Quantified hybrid program α)

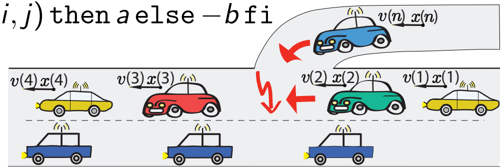
$\forall i : C \; x(i)' = \theta$	(quantified ODE)	
$\forall i : C \; x(i) := \theta$	(quantified assignment)	jump & test
$?\chi$	(conditional execution)	
$\alpha; \beta$	(seq. composition)	Kleene algebra
$\alpha \cup \beta$	(nondet. choice)	
α^*	(nondet. repetition)	

$$DCCS \equiv (appear; ctrl; drive)^*$$

$$appear \equiv n := \mathtt{new}\ C; \quad ?(\forall j : C \; far(j, n))$$

$$ctrl \equiv \forall i : C \; a(i) := \mathtt{if}\ \forall j : C \; far(i, j)\ \mathtt{then}\ a\ \mathtt{else}\ -b\ \mathtt{fi}$$

$$drive \equiv \forall i : C \; x(i)'' = a(i)$$

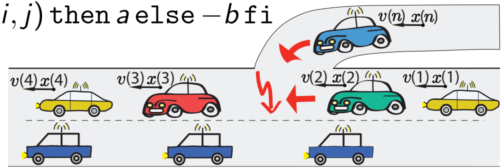


Definition (Quantified hybrid program α)

$\forall i : C\ x(i)' = \theta$	(quantified ODE)	
$\forall i : C\ x(i) := \theta$	(quantified assignment)	jump & test
$?\chi$	(conditional execution)	
$\alpha; \beta$	(seq. composition)	
$\alpha \cup \beta$	(nondet. choice)	Kleene algebra
α^*	(nondet. repetition)	

$$DCCS \equiv (appear; ctrl; drive)^*$$
$$appear \equiv n := \mathtt{new}\ C; \quad ?(\forall j : C\ far(j, n))$$
$$ctrl \equiv \forall i : C\ a(i) := \mathtt{if}\ \forall j : C\ far(i, j)\ \mathtt{then}\ a\ \mathtt{else}\ -b\ \mathtt{fi}$$
$$drive \equiv \forall i : C\ x(i)'' = a(i)$$

new C is definable!



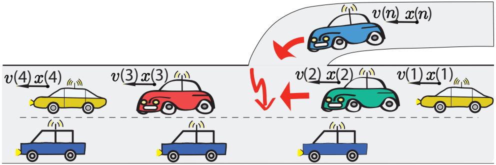
Definition (Qd$\mathcal{L}$ Formula ϕ)

$\neg, \wedge, \vee, \rightarrow, \forall x, \exists x, \quad =, \leq, \quad +, \cdot$ (ℝ-first-order part)

$[\alpha]\phi, \quad \langle\alpha\rangle\phi$ (dynamic part)

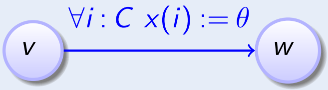
$$\forall i, j : C \; far(i,j) \rightarrow [(appear; ctrl; drive)^*] \; \forall i{\neq}j : C \; x(i) \neq x(j)$$

$$far(i,j) \equiv i \neq j \rightarrow x(i) < x(j) \wedge v(i) \leq v(j) \wedge a(i) \leq a(j) \\ \vee x(i) > x(j) \wedge v(i) \geq v(j) \wedge a(i) \geq a(j) \dots$$

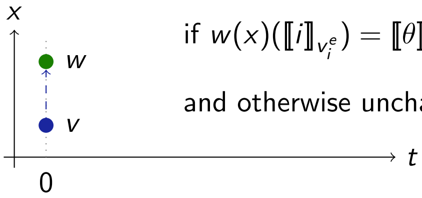


Quantified Differential Dynamic Logic Qd$\mathcal{L}$: Semantics

Definition (Quantified hybrid program α: transition semantics)



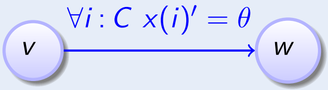
Example



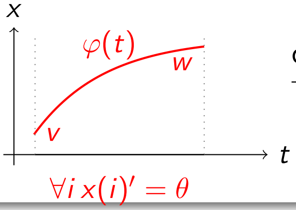
if $w(x)([\![i]\!]_{v_i^e}) = [\![\theta]\!]_{v_i^e}$ (for all e)

and otherwise unchanged

Definition (Quantified hybrid program α: transition semantics)



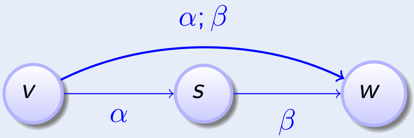
Example



$$\frac{d\, [\![x(i)]\!]_{\varphi(t)_i^e}}{dt}(\zeta) = [\![\theta]\!]_{\varphi(\zeta)_i^e}$$

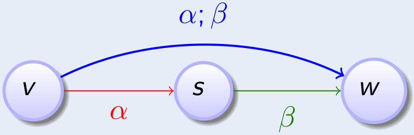
(for all e)

Definition (Quantified hybrid program α: transition semantics)

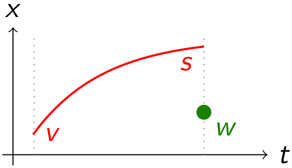


Example

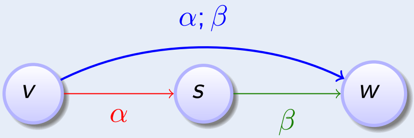
Definition (Quantified hybrid program α: transition semantics)



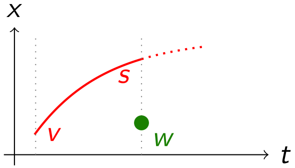
Example



Definition (Quantified hybrid program α: transition semantics)

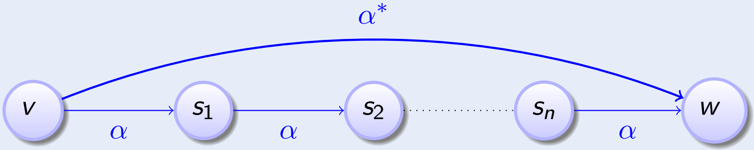


Example



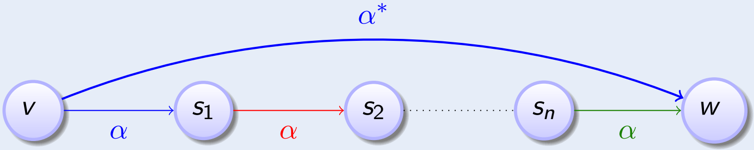
Quantified Differential Dynamic Logic Qd$\mathcal{L}$: Semantics

Definition (Quantified hybrid program α: transition semantics)

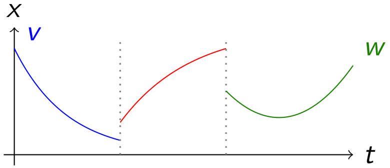


Example

Definition (Quantified hybrid program α: transition semantics)

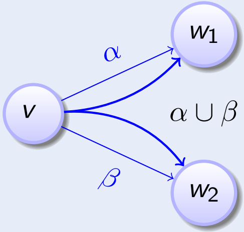


Example



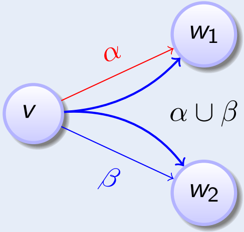
Quantified Differential Dynamic Logic Qd$\mathcal{L}$: Semantics

Definition (Quantified hybrid program α: transition semantics)

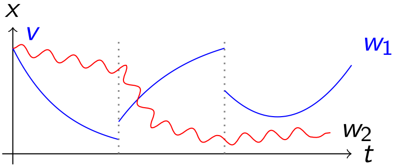


Example

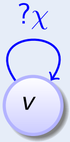
Definition (Quantified hybrid program α: transition semantics)



Example

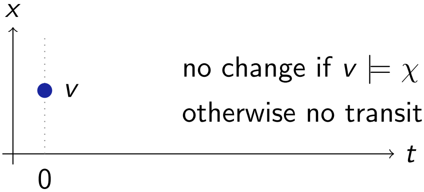


Definition (Quantified hybrid program α: transition semantics)



if $v \models \chi$

Example



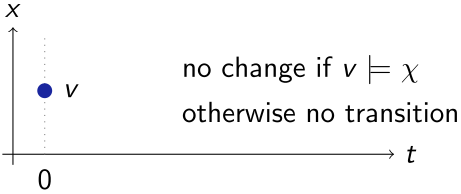
no change if $v \models \chi$

otherwise no transition

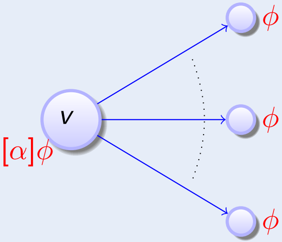
Definition (Quantified hybrid program α: transition semantics)

if $v \not\models \chi$

Example

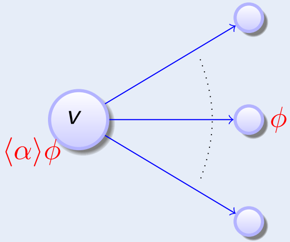


Definition (Qd$\mathcal{L}$ Formula ϕ)

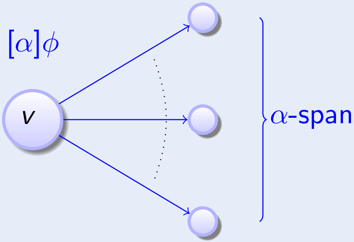


Quantified Differential Dynamic Logic Qd$\mathcal{L}$: Semantics

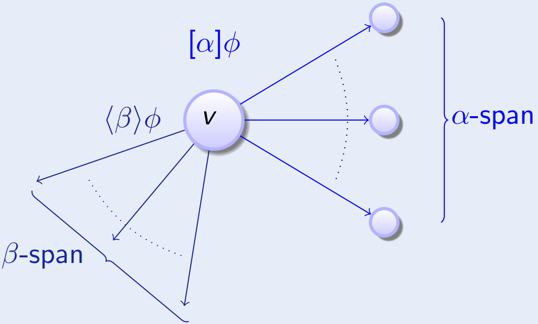
Definition (Qd$\mathcal{L}$ Formula ϕ)



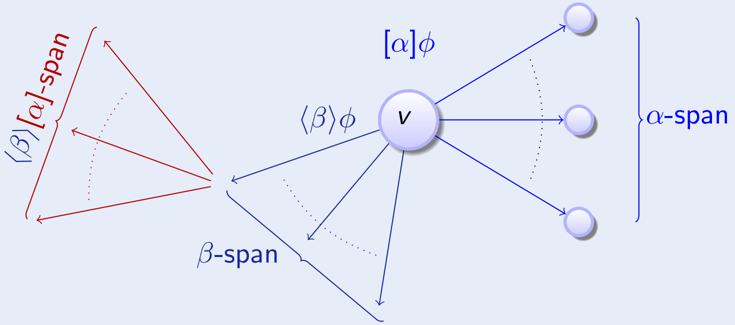
Definition (QdL Formula ϕ)



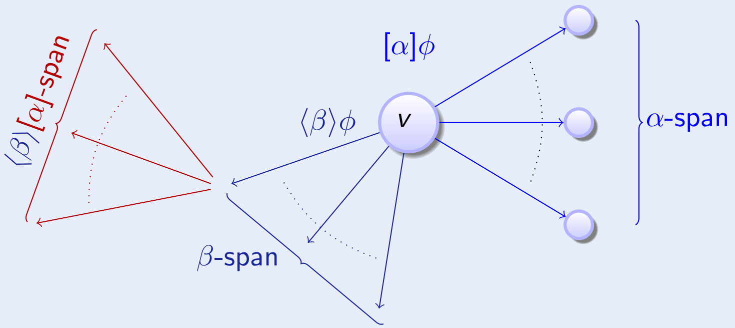
Definition (QdL Formula ϕ)



Definition (QdL Formula ϕ)

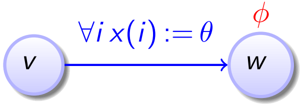


Definition ($\mathsf{Qd}\mathcal{L}$ Formula ϕ)

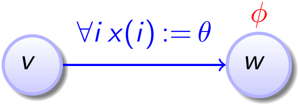


compositional semantics $\Rightarrow$ compositional calculus!

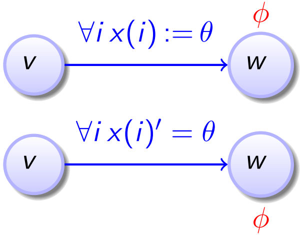
$$\frac{\forall i\,(i = \vec{u} \rightarrow \phi(\theta))}{\phi([\forall i\, x(i) := \theta]x(\vec{u}))}$$



$$\frac{\forall i\,(i = [\forall i\, x(i) := \theta]\vec{u} \rightarrow \phi(\theta))}{\phi([\forall i\, x(i) := \theta]x(\vec{u}))}$$



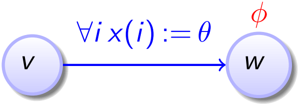
$$\frac{\forall i\,(i = [\forall i\, x(i) := \theta]\vec{u} \to \phi(\theta))}{\phi([\forall i\, x(i) := \theta]x(\vec{u}))}$$



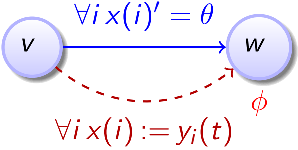
$$\frac{\exists t{\geq}0\,\langle\forall i\, x(i) := y_i(t)\rangle\phi}{\langle\forall i\, x(i)' = \theta\rangle\phi}$$

Proof Calculus: Quantified Differential Dynamic Logic

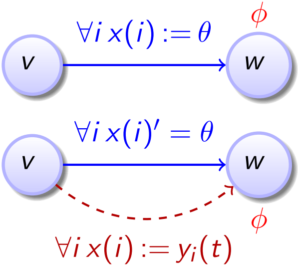
$$\frac{\forall i\, (i = [\forall i\, x(i) := \theta]\vec{u} \to \phi(\theta))}{\phi([\forall i\, x(i) := \theta]x(\vec{u}))}$$



$$\frac{\exists t{\ge}0\, \langle \forall i\, x(i) := y_i(t)\rangle \phi}{\langle \forall i\, x(i)' = \theta\rangle \phi}$$



$$\frac{\forall i\,(i = [\forall i\, x(i) := \theta]\vec{u} \to \phi(\theta))}{\phi([\forall i\, x(i) := \theta]x(\vec{u}))}$$

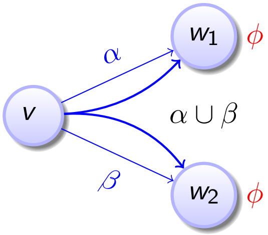


$$\frac{\exists t{\geq}0\, \langle \forall i\, x(i) := y_i(t)\rangle \phi}{\langle \forall i\, x(i)' = \theta \rangle \phi}$$

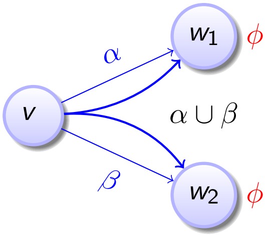
solve infinite-dimensional diff. eqn.?

compositional semantics $\Rightarrow$ compositional rules!

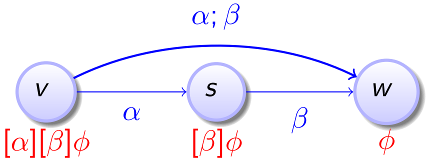
$$\frac{[\alpha]\phi \land [\beta]\phi}{[\alpha \cup \beta]\phi}$$



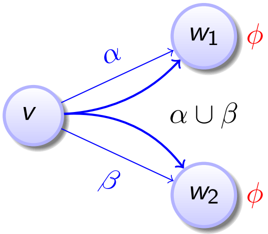
$$\frac{[\alpha]\phi \wedge [\beta]\phi}{[\alpha \cup \beta]\phi}$$



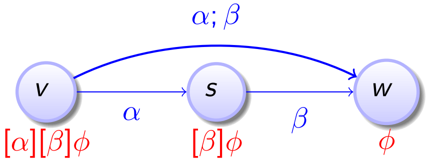
$$\frac{[\alpha][\beta]\phi}{[\alpha; \beta]\phi}$$



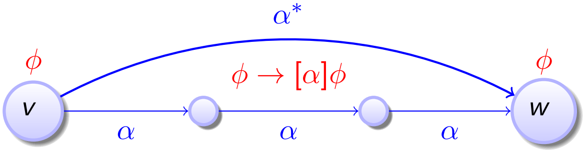
$$\frac{[\alpha]\phi \wedge [\beta]\phi}{[\alpha \cup \beta]\phi}$$



$$\frac{[\alpha][\beta]\phi}{[\alpha; \beta]\phi}$$

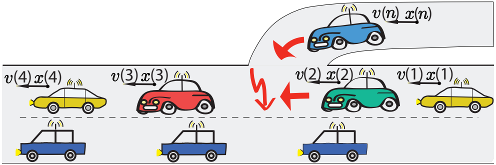


$$\frac{\phi \quad (\phi \to [\alpha]\phi)}{[\alpha^*]\phi}$$



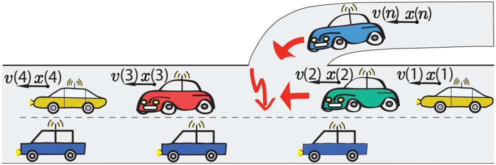
$$\overline{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}$$

$$\forall i{\neq}j\, x(i){\neq}x(j) \rightarrow [\forall i\, x(i)' = v(i), v(i)' = -b]\, \forall j{\neq}k\, x(j){\neq}x(k)$$



Deduction Modulo

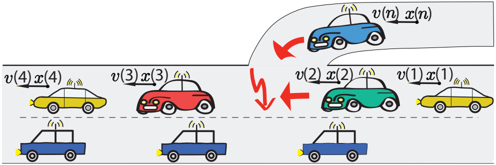
$$\frac{\forall i{\neq}j\, x(i){\neq}x(j) \rightarrow \forall t{\geq}0\, [\forall i\, x(i) := -\tfrac{b}{2}t^2 + v(i)t + x(i)]\, \forall j{\neq}k\, x(j){\neq}x(k)}{\forall i{\neq}j\, x(i){\neq}x(j) \rightarrow [\forall i\, x(i)' = v(i), v(i)' = -b]\, \forall j{\neq}k\, x(j){\neq}x(k)}$$



$$\forall i{\neq}j\, x(i){\neq}x(j) \rightarrow \forall t{\geq}0\, \forall j{\neq}k\, [\forall i\, x(i) := -\tfrac{b}{2}t^2 + v(i)t + x(i)]\, x(j){\neq}x(k)$$

$$\forall i{\neq}j\, x(i){\neq}x(j) \rightarrow \forall t{\geq}0\, [\forall i\, x(i) := -\tfrac{b}{2}t^2 + v(i)t + x(i)]\, \forall j{\neq}k\, x(j){\neq}x(k)$$

$$\forall i{\neq}j\, x(i){\neq}x(j) \rightarrow [\forall i\, x(i)' = v(i), v(i)' = -b]\, \forall j{\neq}k\, x(j){\neq}x(k)$$



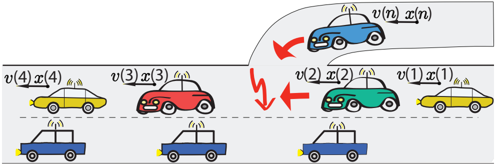
Deduction Modulo

$$\forall i{\neq}j\, x(i){\neq}x(j) \rightarrow \forall t{\geq}0\, \forall j{\neq}k\, (-\tfrac{b}{2}t^2 + v(j)t + x(j) \neq -\tfrac{b}{2}t^2 + v(k)t + x(k))$$

$$\forall i{\neq}j\, x(i){\neq}x(j) \rightarrow \forall t{\geq}0\, \forall j{\neq}k\, [\forall i\, x(i) := -\tfrac{b}{2}t^2 + v(i)t + x(i)]\, x(j){\neq}x(k)$$

$$\forall i{\neq}j\, x(i){\neq}x(j) \rightarrow \forall t{\geq}0\, [\forall i\, x(i) := -\tfrac{b}{2}t^2 + v(i)t + x(i)]\, \forall j{\neq}k\, x(j){\neq}x(k)$$

$$\forall i{\neq}j\, x(i){\neq}x(j) \rightarrow [\forall i\, x(i)' = v(i), v(i)' = -b]\, \forall j{\neq}k\, x(j){\neq}x(k)$$



Deduction Modulo

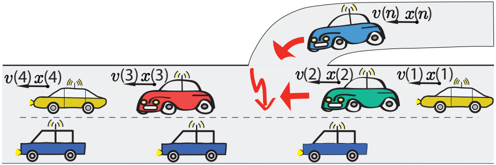
$$\forall i{\neq}j\, x(i){\neq}x(j) \rightarrow \forall j{\neq}k\, \forall t{\geq}0\, \left(-\tfrac{b}{2}t^2 + v(j)t + x(j) \neq -\tfrac{b}{2}t^2 + v(k)t + x(k)\right)$$

$$\forall i{\neq}j\, x(i){\neq}x(j) \rightarrow \forall t{\geq}0\, \forall j{\neq}k\, \left(-\tfrac{b}{2}t^2 + v(j)t + x(j) \neq -\tfrac{b}{2}t^2 + v(k)t + x(k)\right)$$

$$\forall i{\neq}j\, x(i){\neq}x(j) \rightarrow \forall t{\geq}0\, \forall j{\neq}k\, [\forall i\, x(i) := -\tfrac{b}{2}t^2 + v(i)t + x(i)]\, x(j){\neq}x(k)$$

$$\forall i{\neq}j\, x(i){\neq}x(j) \rightarrow \forall t{\geq}0\, [\forall i\, x(i) := -\tfrac{b}{2}t^2 + v(i)t + x(i)]\, \forall j{\neq}k\, x(j){\neq}x(k)$$

$$\forall i{\neq}j\, x(i){\neq}x(j) \rightarrow [\forall i\, x(i)' = v(i), v(i)' = -b]\, \forall j{\neq}k\, x(j){\neq}x(k)$$



$$\forall i{\neq}j\, x(i){\neq}x(j) \rightarrow \forall j{\neq}k\, (x(j) \leq x(k) \land v(j) \leq v(k) \lor x(j) \geq x(k) \land v(j) \geq v$$

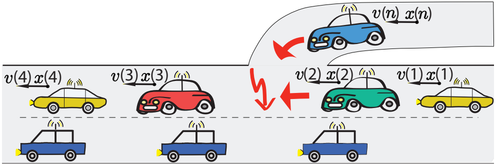
$$\forall i{\neq}j\, x(i){\neq}x(j) \rightarrow \forall j{\neq}k\, \forall t{\geq}0\, (-\tfrac{b}{2}t^2 + v(j)t + x(j) \neq -\tfrac{b}{2}t^2 + v(k)t + x(k))$$

$$\forall i{\neq}j\, x(i){\neq}x(j) \rightarrow \forall t{\geq}0\, \forall j{\neq}k\, (-\tfrac{b}{2}t^2 + v(j)t + x(j) \neq -\tfrac{b}{2}t^2 + v(k)t + x(k))$$

$$\forall i{\neq}j\, x(i){\neq}x(j) \rightarrow \forall t{\geq}0\, \forall j{\neq}k\, [\forall i\, x(i) := -\tfrac{b}{2}t^2 + v(i)t + x(i)]\, x(j){\neq}x(k)$$

$$\forall i{\neq}j\, x(i){\neq}x(j) \rightarrow \forall t{\geq}0\, [\forall i\, x(i) := -\tfrac{b}{2}t^2 + v(i)t + x(i)]\, \forall j{\neq}k\, x(j){\neq}x(k)$$

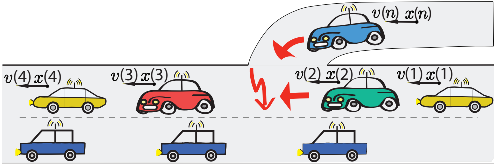
$$\forall i{\neq}j\, x(i){\neq}x(j) \rightarrow [\forall i\, x(i)' = v(i), v(i)' = -b]\, \forall j{\neq}k\, x(j){\neq}x(k)$$



Actual Existence and Creation

Actual Existence Function E(·)

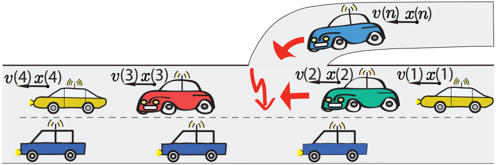
$$E(i) = \begin{cases} 0 & \text{if } i \text{ denotes a possible object} \\ 1 & \text{if } i \text{ denotes an actively existing objects} \end{cases}$$



Actual Existence Function $\mathrm{E}(\cdot)$

$$\mathrm{E}(i) = \begin{cases} 0 & \text{if } i \text{ denotes a possible object} \\ 1 & \text{if } i \text{ denotes an actively existing objects} \end{cases}$$

$$\frac{}{[n := \mathtt{new}\ C]\phi}$$

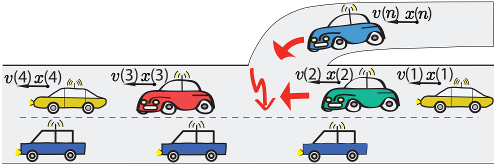


Actual Existence and Creation

Actual Existence Function $\mathrm{E}(\cdot)$

$$\mathrm{E}(i) = \begin{cases} 0 & \text{if } i \text{ denotes a possible object} \\ 1 & \text{if } i \text{ denotes an actively existing objects} \end{cases}$$

$$\frac{[(\forall j : C\ n := j);\qquad\qquad]\phi}{[n := \mathtt{new}\ C]\phi}$$

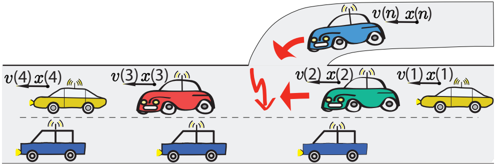


Actual Existence and Creation

Actual Existence Function E(·)

$$\mathsf{E}(i) = \begin{cases} 0 & \text{if } i \text{ denotes a possible object} \\ 1 & \text{if } i \text{ denotes an actively existing objects} \end{cases}$$

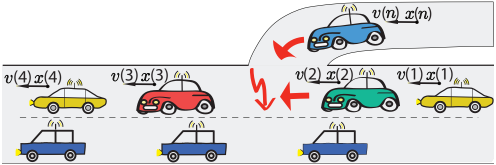
$$\frac{[(\forall j : C\ \ n := j);\ \ ?(\mathsf{E}(n) = 0);\qquad\qquad]\phi}{[n := \mathtt{new}\ C]\phi}$$



Actual Existence Function $\mathrm{E}(\cdot)$

$$\mathrm{E}(i) = \begin{cases} 0 & \text{if } i \text{ denotes a possible object} \\ 1 & \text{if } i \text{ denotes an actively existing objects} \end{cases}$$

$$\frac{[(\forall j : C\ n := j);\ ?(\mathrm{E}(n) = 0);\ \mathrm{E}(n) := 1]\phi}{[n := \mathtt{new}\ C]\phi}$$



Actual Existence Function $\mathrm{E}(\cdot)$

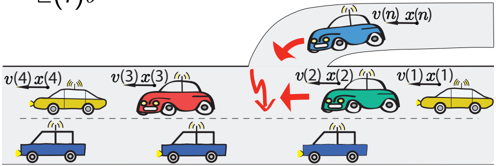
$$\mathrm{E}(i) = \begin{cases} 0 & \text{if } i \text{ denotes a possible object} \\ 1 & \text{if } i \text{ denotes an actively existing objects} \end{cases}$$

$$\frac{[(\forall j : C\ n := j);\ ?(\mathrm{E}(n) = 0);\ \mathrm{E}(n) := 1]\phi}{[n := \mathtt{new}\ C]\phi}$$

$$\forall i : C!\ \phi \equiv \forall i : C\ (\mathrm{E}(i) = 1 \to \phi)$$

$$\forall i : C!\ f(i) := \theta \equiv \forall i : C\ f(i) := (\mathtt{if}\ \mathrm{E}(i) = 1\ \mathtt{then}\ \theta\ \mathtt{else}\ f(i)\ \mathtt{fi})$$

$$\forall i : C!\ f(i)' = \theta \equiv \forall i : C\ f(i)' = \mathrm{E}(i)\theta$$



Soundness and Completeness

Theorem (Relative Completeness)

Qd$\mathcal{L}$ calculus is a sound & complete axiomatisation of distributed hybrid systems relative to quantified differential equations. Proof 16p.

André Platzer.
Quantified differential dynamic logic for distributed hybrid systems.
In Anuj Dawar and Helmut Veith, editors,
CSL, vol. 6247 of *LNCS*, 469–483. Springer, 2010.

Soundness and Completeness

Theorem (Relative Completeness)

Qd$\mathcal{L}$ *calculus is a sound & complete axiomatisation of distributed hybrid systems relative to quantified differential equations.* Proof 16p.

Corollary (Proof-theoretical Alignment)

proving distributed hybrid systems = proving dynamical systems!

André Platzer.
Quantified differential dynamic logic for distributed hybrid systems.
In Anuj Dawar and Helmut Veith, editors,
CSL, vol. 6247 of *LNCS*, 469–483. Springer, 2010.

Soundness and Completeness

Theorem (Relative Completeness)

Qd$\mathcal{L}$ calculus is a sound & complete axiomatisation of distributed hybrid systems relative to quantified differential equations.

Proof 16p.

Corollary (Proof-theoretical Alignment)

proving distributed hybrid systems = proving dynamical systems!

Corollary (Decomposition!)

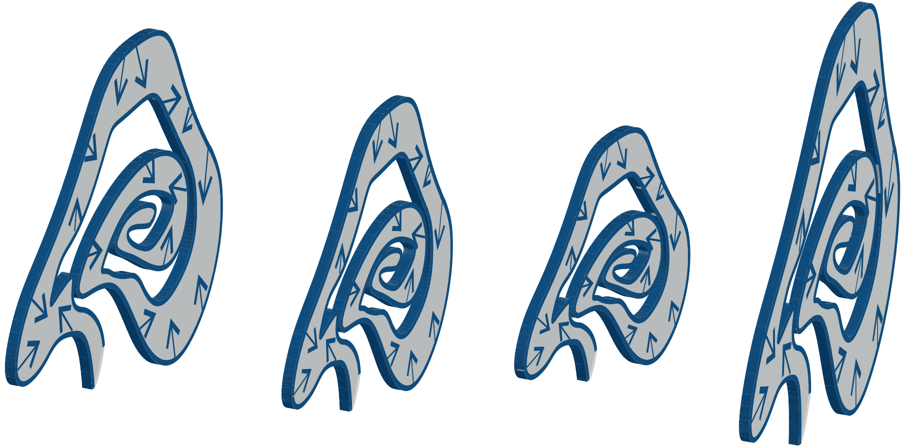
distributed hybrid systems can be verified by recursive decomposition

André Platzer.
Quantified differential dynamic logic for distributed hybrid systems.
In Anuj Dawar and Helmut Veith, editors,
CSL, vol. 6247 of *LNCS*, 469–483. Springer, 2010.

Quantified Differential Invariants

Definition (Quantified Differential Invariant)

Quantified formula F closed under total differentiation with respect to quantified differential constraints



Derivations and Differentiation

Definition (Syntactic total derivation D)

$$D(r) = 0 \quad \text{if } r \text{ a number symbol}$$
$$D(x(i)) = x(i)' \quad \text{if } x : C \to \mathbb{R},\ C \neq \mathbb{R}$$
$$D(a + b) = D(a) + D(b)$$
$$D(a \cdot b) = D(a) \cdot b + a \cdot D(b)$$
$$D(a/b) = (D(a) \cdot b - a \cdot D(b))/b^2$$

Definition (Syntactic total derivation D)

$$
\begin{aligned}
D(r) &= 0 && \text{if } r \text{ a number symbol}\\
D(x(i)) &= x(i)' && \text{if } x : C \to \mathbb{R},\ C \neq \mathbb{R}\\
D(a+b) &= D(a) + D(b)\\
D(a \cdot b) &= D(a) \cdot b + a \cdot D(b)\\
D(a/b) &= (D(a) \cdot b - a \cdot D(b))/b^2\\
D(a \geq b) &\equiv D(a) \geq D(b) && \text{accordingly for } >, =\\
D(F \land G) &\equiv D(F) \land D(G)\\
D(\forall i\, F) &\equiv \forall i\, D(F)
\end{aligned}
$$

Definition (Syntactic total derivation D)

$$
\begin{aligned}
D(r) &= 0 && \text{if } r \text{ a number symbol}\\
D(x(i)) &= x(i)' && \text{if } x : C \to \mathbb{R},\ C \neq \mathbb{R}\\
D(a+b) &= D(a) + D(b)\\
D(a \cdot b) &= D(a) \cdot b + a \cdot D(b)\\
D(a/b) &= (D(a) \cdot b - a \cdot D(b))/b^2\\
D(a \geq b) &\equiv D(a) \geq D(b) && \text{accordingly for } >, =\\
D(F \land G) &\equiv D(F) \land D(G)\\
D(\forall i\, F) &\equiv \forall i\, D(F)
\end{aligned}
$$

$$
\begin{aligned}
\mathcal{P} &\equiv \forall i,j : A\ (i = j \lor (x_1(i) - x_1(j))^2 + (x_2(i) - x_2(j))^2 \geq p^2)\\
\Rightarrow D(\mathcal{P}) &\equiv \forall i,j : A\ (i' = j' \land 2(x_1(i) - x_1(j))(x_1(i)' - x_1(j)')\\
&\qquad + 2(x_2(i) - x_2(j))(x_2(i)' - x_2(j)') \geq 0)
\end{aligned}
$$

Definition (Syntactic total derivation D)

$$
\begin{aligned}
D(r) &= 0 && \text{if } r \text{ a number symbol}\\
D(x(i)) &= x(i)' && \text{if } x : C \to \mathbb{R},\ C \neq \mathbb{R}\\
D(a+b) &= D(a) + D(b)\\
D(a \cdot b) &= D(a) \cdot b + a \cdot D(b)\\
D(a/b) &= (D(a) \cdot b - a \cdot D(b))/b^2\\
D(a \geq b) &\equiv D(a) \geq D(b) && \text{accordingly for } >, =\\
D(F \wedge G) &\equiv D(F) \wedge D(G)\\
D(\forall i\, F) &\equiv \forall i\, D(F)
\end{aligned}
$$

$$
\begin{aligned}
\mathcal{P} &\equiv \forall i,j : A\ \left(i = j \vee (x_1(i) - x_1(j))^2 + (x_2(i) - x_2(j))^2 \geq p^2\right)\\
\Rightarrow D(\mathcal{P}) &\equiv \forall i,j : A\ \left(i' = j' \wedge 2(x_1(i) - x_1(j))(x_1(i)' - x_1(j)')\right.\\
&\qquad\qquad \left. + 2(x_2(i) - x_2(j))(x_2(i)' - x_2(j)') \geq 0\right)
\end{aligned}
$$

Derivations and Differentiation

Syntactic derivation $D(\cdot)$ coincides with analytic differentiation:

Lemma (Derivation lemma)

Valuation is a differential homomorphism: for all flows φ *all* $\zeta \in [0, r]$

$$\frac{\mathrm{d}\,[\![\theta]\!]_{\varphi(t)}}{\mathrm{d}t}(\zeta) = [\![D(\theta)]\!]_{\bar{\varphi}(\zeta)}$$

Syntactic derivation $D(\cdot)$ coincides with analytic differentiation:

Lemma (Derivation lemma)

Valuation is a differential homomorphism: for all flows φ *all* $\zeta \in [0, r]$

$$\frac{\mathrm{d}\, [\![\theta]\!]_{\varphi(t)}}{\mathrm{d}t}(\zeta) = [\![D(\theta)]\!]_{\bar{\varphi}(\zeta)}$$

Locally understand QDE as quantified assignments:

Lemma (Quantified differential substitution principle)

If $\varphi \models \forall i : C\ f(i)' = \theta \,\&\, H$*, then* $\varphi \models \upsilon = [\forall i : C\ f(i)' := \theta]\upsilon$ *for all* υ*.*

Derivations and Differentiation

Syntactic derivation $D(\cdot)$ coincides with analytic differentiation:

Lemma (Derivation lemma)

Valuation is a differential homomorphism: for all flows φ all $\zeta \in [0, r]$

$$\frac{\mathrm{d}\, [\![\theta]\!]_{\varphi(t)}}{\mathrm{d}t}(\zeta) = [\![D(\theta)]\!]_{\bar{\varphi}(\zeta)}$$

Locally understand QDE as quantified assignments:

Lemma (Quantified differential substitution principle)

If $\varphi \models \forall i : C\ f(i)' = \theta \,\&\, H$, then $\varphi \models \upsilon = [\forall i : C\ f(i)' := \theta]\upsilon$ for all υ.

Theorem (Quantified Differential Invariant)

$$(QDI)\quad \frac{H \rightarrow [\forall i : C\ f(\vec{i})' := \theta]D(F)}{F \rightarrow [\forall i : C\ f(\vec{i})' = \theta \,\&\, H]F} \quad \text{is sound}$$

A Simple Proof with Quantified Differential Invariants

$$\frac{}{\forall i : C\ 2x(i)^3 \geq 1 \to [\forall i : C\ x(i)' = x(i)^2 + x(i)^4 + 2]\forall i : C\ 2x(i)^3 \geq 1}$$

A Simple Proof with Quantified Differential Invariants

$$\dfrac{[\forall i : C\ x(i)' := x(i)^2 + x(i)^4 + 2]\forall i : C\ 2(x(i)^3)' \geq 0}{\forall i : C\ 2x(i)^3 \geq 1 \rightarrow [\forall i : C\ x(i)' = x(i)^2 + x(i)^4 + 2]\forall i : C\ 2x(i)^3 \geq 1}$$

$$\dfrac{\dfrac{}{[\forall i : C\ x(i)' := x(i)^2 + x(i)^4 + 2]\forall i : C\ 6x(i)^2 x(i)' \geq 0}}{\dfrac{[\forall i : C\ x(i)' := x(i)^2 + x(i)^4 + 2]\forall i : C\ 2(x(i)^3)' \geq 0}{\forall i : C\ 2x(i)^3 \geq 1 \rightarrow [\forall i : C\ x(i)' = x(i)^2 + x(i)^4 + 2]\forall i : C\ 2x(i)^3 \geq 1}}$$

$$\forall i : C\ 6x(i)^2(x(i)^2 + x(i)^4 + 2) \geq 0$$

$$[\forall i : C\ x(i)' := x(i)^2 + x(i)^4 + 2]\forall i : C\ 6x(i)^2 x(i)' \geq 0$$

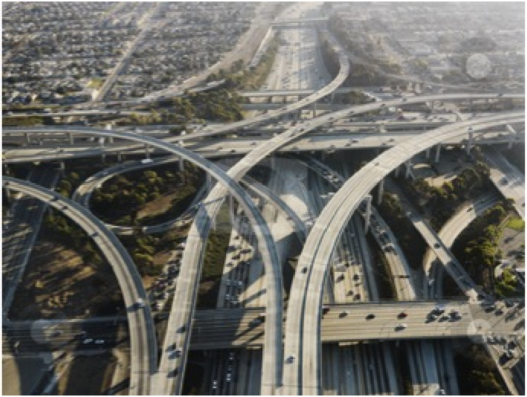
$$[\forall i : C\ x(i)' := x(i)^2 + x(i)^4 + 2]\forall i : C\ 2(x(i)^3)' \geq 0$$

$$\forall i : C\ 2x(i)^3 \geq 1 \rightarrow [\forall i : C\ x(i)' = x(i)^2 + x(i)^4 + 2]\forall i : C\ 2x(i)^3 \geq 1$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\mathit{true}}{\forall i : C\ 6x(i)^2(x(i)^2 + x(i)^4 + 2) \geq 0}}{[\forall i : C\ x(i)' := x(i)^2 + x(i)^4 + 2]\forall i : C\ 6x(i)^2 x(i)' \geq 0}}{[\forall i : C\ x(i)' := x(i)^2 + x(i)^4 + 2]\forall i : C\ 2(x(i)^3)' \geq 0}}{\forall i : C\ 2x(i)^3 \geq 1 \rightarrow [\forall i : C\ x(i)' = x(i)^2 + x(i)^4 + 2]\forall i : C\ 2x(i)^3 \geq 1}$$

Outline Background

Driver's License Test for Robotic Cars?



Driver's License Test for Robotic Cars?

Driver's License Test for Robotic Cars? Proof!

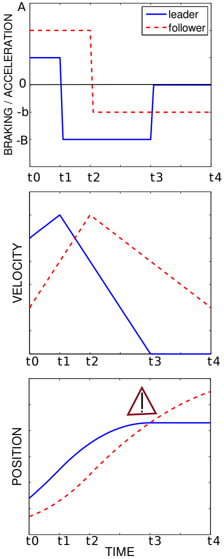
Car Control: Local Lane Control Challenge

Challenge: Local lane dynamics

- A car controller for a differential equation respects separation of local lane.

Challenge: Local lane dynamics

- A car controller for a differential equation respects separation of local lane.
- Follower car maintains safe distance to leader:

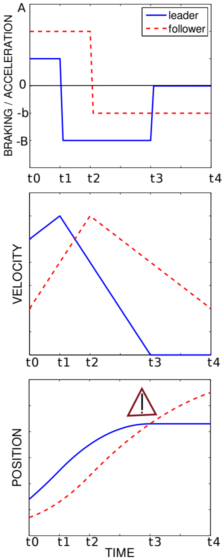


Car Control: Local Lane Control Challenge

Challenge: Local lane dynamics

- A car controller for a differential equation respects separation of local lane.
- Follower car maintains safe distance to leader:

$$f \ll \ell \to [(a_i := ctrl;\ x_i'' = a_i)^*]\, f \ll \ell$$

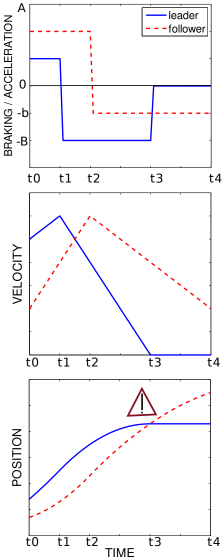


Challenge: Local lane dynamics

- A car controller for a differential equation respects separation of local lane.
- Follower car maintains safe distance to leader:

$$f \ll \ell \rightarrow [(a_i := ctrl;\ x_i'' = a_i)^*]\, f \ll \ell$$

$$f \ll \ell \equiv\ (x_f \le x_\ell) \land (f \ne \ell) \rightarrow$$
$$(x_\ell > x_f + \frac{v_f^2}{2b} - \frac{v_\ell^2}{2B}$$
$$\land\, x_\ell > x_f \land v_f \ge 0 \land v_\ell \ge 0)$$



$$f \ll \ell \rightarrow [\mathtt{llc}]\, f \ll \ell$$

Hybrid Program (Local lane control)

$$\mathtt{llc} \equiv (ctrl; dyn)^*$$

$$ctrl \equiv \ell_{ctrl} \;||\; f_{ctrl};$$

$$\ell_{ctrl} \equiv (a_\ell := *; \quad ?(-B \le a_\ell \le A))$$

$$f_{ctrl} \equiv \big(a_f := *; \quad ?(-B \le a_f \le -b)\big)$$

$$\cup \quad \big(?\mathbf{Safe}_\varepsilon; \quad a_f := *; \quad ?(-B \le a_f \le A)\big)$$

$$\cup \quad \big(?(v_f = 0); \; a_f := 0\big)$$

$$\mathbf{Safe}_\varepsilon \equiv x_f + \frac{v_f^2}{2b} + \left(\frac{A}{b} + 1\right)\left(\frac{A}{2}\varepsilon^2 + \varepsilon v_f\right) < x_\ell + \frac{v_\ell^2}{2B}$$

$$dyn \equiv (t := 0; \; x_f' = v_f, \; v_f' = a_f, \; x_\ell' = v_\ell, \; v_\ell' = a_\ell, t' = 1$$

$$\& \; v_f \ge 0 \;\wedge\; v_\ell \ge 0 \;\wedge\; t \le \varepsilon)$$

Challenge: Global lane dynamics

- All controllers for arbitrarily many differential equations respect separation globally on lane.

Challenge: Global lane dynamics

- All controllers for arbitrarily many differential equations respect separation globally on lane.
- Each car safe behind all others

Challenge: Global lane dynamics

- All controllers for arbitrarily many differential equations respect separation globally on lane.
- Each car safe behind all others

$$[(\forall i\, a(i) := ctrl;\ \forall i\, x(i)'' = a(i))^*]\, \forall i, j\, i \ll j$$

$$\forall i : C\ i \ll \ell(i) \rightarrow [\texttt{glc}](\forall i : C\ i \ll \ell^*(i))$$

Quantified Hybrid Program (Global lane control)

$$\texttt{glc} \equiv (ctrl^n; dyn^n)^*$$

$$ctrl^n \equiv \forall i : C\ (ctrl(i))$$

$$ctrl(i) \equiv (a(i) := *; ?(-B \leq a(i) \leq -b))$$

$$\cup\quad (?\mathbf{Safe}_\varepsilon(i);\ a(i) := *;\ ?(-B \leq a(i) \leq A))$$

$$\cup\quad (?(v(i) = 0);\ a(i) := 0)$$

$$\mathbf{Safe}_\varepsilon(i) \equiv x(i) + \frac{v(i)^2}{2b} + \left(\frac{A}{b} + 1\right)\left(\frac{A}{2}\varepsilon^2 + \varepsilon v(i)\right) < x(\ell(i)) + \frac{v(\ell(i))^2}{2B}$$

$$dyn^n \equiv t := 0;\ \forall i : C\ (dyn(i), t' = 1\ \&\ v(i) \geq 0 \land t \leq \varepsilon)$$

$$dyn(i) \equiv x(i)' = v(i), v(i)' = a(i)$$

$$i \ll \ell^*(i) \equiv [k := i;\ (k := \ell(k))^*] i \ll k$$

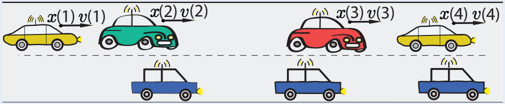
$$\forall i : C\ i \ll \ell(i) \to [\mathtt{glc}](\forall i : C\ i \ll \ell^*(i))$$

Quantified Hybrid Program (Global lane control)

$$\mathtt{glc} \equiv (ctrl^n; dyn^n)^*$$
$$ctrl^n \equiv \forall i : C\ (ctrl(i))$$
$$ctrl(i) \equiv (a(i) := *; ?(-B \le a(i) \le -b))$$
$$\cup\ (?\mathbf{Safe}_\varepsilon(i);\ a(i) := *;\ ?(-B \le a(i) \le A))$$
$$\cup\ (?(v(i) = 0);\ a(i) := 0)$$
$$\mathbf{Safe}_\varepsilon(i) \equiv x(i) + \frac{v(i)^2}{2b} + \left(\frac{A}{b} + 1\right)\left(\frac{A}{2}\varepsilon^2 + \varepsilon v(i)\right) < x(\ell(i)) + \frac{v(\ell(i))^2}{2B}$$
$$dyn^n \equiv t := 0;\ \forall i : C\ (dyn(i), t' = 1\ \&\ v(i) \ge 0 \land t \le \varepsilon)$$
$$dyn(i) \equiv x(i)' = v(i), v(i)' = a(i)$$
$$i \ll \ell^*(i) \equiv [k := i;\ (k := \ell(k))^*]i \ll k$$

$$\forall i : C \; i \ll \ell(i) \rightarrow [\texttt{glc}](\forall i : C \; i \ll \ell^*(i))$$

Quantified Hybrid Program (Global lane control)



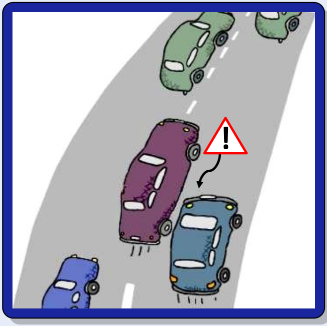
$i \ll \ell^*(i) \equiv [k := i; \; (k := \ell(k))^*] i \ll k$

Challenge: Local highway dynamics

- All controllers for arbitrarily many differential equations respect separation locally on highway.

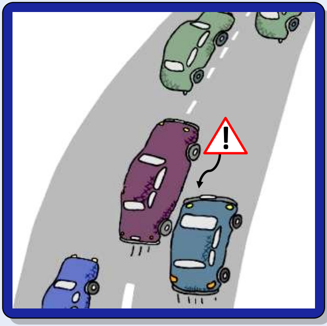
Challenge: Local highway dynamics

- All controllers for arbitrarily many differential equations respect separation locally on highway.
- For each lane: all controllers for the differential equations respect separation even if cars appear or disappear.



Challenge: Local highway dynamics

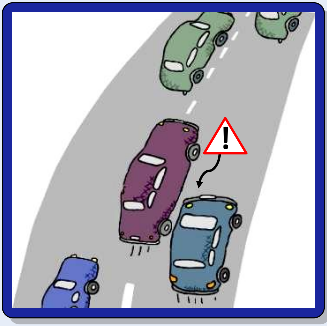
- All controllers for arbitrarily many differential equations respect separation locally on highway.
- For each lane: all controllers for the differential equations respect separation even if cars appear or disappear.
- Each car safe behind all others, even if new cars appear or disappear.



Car Control: Local Highway Control Challenge

Challenge: Local highway dynamics

- All controllers for arbitrarily many differential equations respect separation locally on highway.
- For each lane: all controllers for the differential equations respect separation even if cars appear or disappear.
- Each car safe behind all others, even if new cars appear or disappear.



$$[(n := \mathtt{new}\ C;\ \forall i\, a(i) := ctrl;\ \forall i\, x(i)'' = a(i))^*]\, \forall i,j\, i \ll j$$

Car Control: Local Highway Control Challenge

$$\forall i : C \; i \ll \ell(i) \rightarrow [\mathtt{glc}](\forall i : C \; i \ll \ell^*(i))$$

Quantified Hybrid Program (Local highway control)

$$\begin{aligned}
\mathtt{lhc} &\equiv (delete^*; create^*; ctrl^n; dyn^n)^* \\
create &\equiv n := new;\ ?(F(n) \ll n \wedge n \ll \ell(n)) \\
(n := new) &\equiv n := *;\ ?(\mathrm{E}(n) = 0);\ \mathrm{E}(n) := 1 \\
F(n) \ll n &\equiv \forall j : C\ (\ell(j) = n \rightarrow j \ll n) \\
delete &\equiv n := *;\ ?(\mathrm{E}(n) = 1);\ \mathrm{E}(n) := 0
\end{aligned}$$

$$\forall i : C\ i \ll \ell(i) \rightarrow [\mathtt{glc}](\forall i : C\ i \ll \ell^*(i))$$

Quantified Hybrid Program (Local highway control)

$$\begin{aligned}
\mathtt{lhc} &\equiv (\mathit{delete}^*; \mathit{create}^*; \mathit{ctrl}^n; \mathit{dyn}^n)^* \\
\mathit{create} &\equiv n := \mathit{new};\ ?(F(n) \ll n \land n \ll \ell(n)) \\
(n := \mathit{new}) &\equiv n := *;\ ?(\mathrm{E}(n) = 0);\ \mathrm{E}(n) := 1 \\
F(n) \ll n &\equiv \forall j : C\ (\ell(j) = n \rightarrow j \ll n) \\
\mathit{delete} &\equiv n := *;\ ?(\mathrm{E}(n) = 1);\ \mathrm{E}(n) := 0
\end{aligned}$$

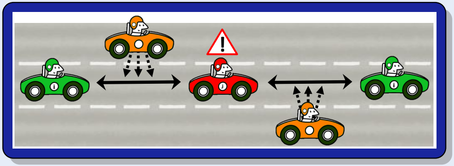
Car Control: Global Highway Control Challenge

Challenge: Global highway dynamics

- All controllers for arbitrarily many differential equations respect separation globally on highway.

Challenge: Global highway dynamics

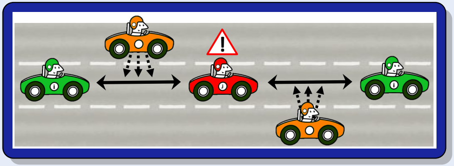
- All controllers for arbitrarily many differential equations respect separation globally on highway.
- All controllers for the differential equations respect separation even if cars switch lanes.



Car Control: Global Highway Control Challenge

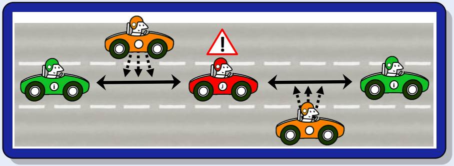
Challenge: Global highway dynamics

- All controllers for arbitrarily many differential equations respect separation globally on highway.
- All controllers for the differential equations respect separation even if cars switch lanes.
- On all lanes, all car safe behind all others on their lanes, even if cars switch lanes.



Car Control: Global Highway Control Challenge

Challenge: Global highway dynamics



- All controllers for arbitrarily many differential equations respect separation globally on highway.
- All controllers for the differential equations respect separation even if cars switch lanes.
- On all lanes, all car safe behind all others on their lanes, even if cars switch lanes.

$$[\forall l\, (n := \mathtt{new}\ C;\ \forall i\, a(i) := ctrl;\ \forall i\, x(i)'' = a(i))^*]\, \forall l\, \forall i, j\, i \ll j$$

$$\forall l : L \forall i : C_l\, i \ll \ell_l(i) \rightarrow$$
$$[(\forall l : L\, delete_l^*; \forall l : L\, new_l^*; \forall l : L\, ctrl_l^n; \forall l : L\, dyn_l^n)^*]\, \forall l : L \forall i : C_l\, i \ll \ell_l^*(i)$$

Quantified Hybrid Program (Global highway control)

$$\mathtt{ghc} \equiv (\forall l : L\, delete_l^*;\ \forall l : L\, new_l^*;\ \forall l : L\,,ctrl_l^n;\ \forall l : L\, dyn_l^n)^*$$

Car Control: Global Highway Control Challenge

$$\forall l : L \forall i : C_l\, i \ll \ell_l(i) \rightarrow$$
$$[(\forall l : L\, delete_l^*; \forall l : L\, new_l^*; \forall l : L\, ctrl_l^n; \forall l : L\, dyn_l^n)^*]\, \forall l : L \forall i : C_l\, i \ll \ell_l^*(i)$$

Quantified Hybrid Program (Global highway control)

$$\mathrm{ghc} \equiv (\forall l : L\, delete_l^*;\ \forall l : L\, new_l^*;\ \forall l : L\,, ctrl_l^n;\ \forall l : L\, dyn_l^n)^*$$

Outline Background

Cyber-Physical Systems:

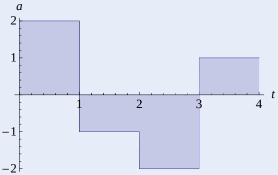
Q: I want to verify trains

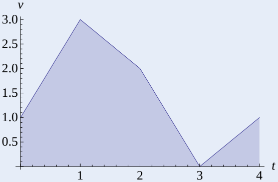
Challenge

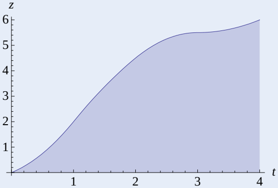
Q: I want to verify trains A: Hybrid systems

Challenge (Hybrid Systems)

- Continuous dynamics (differential equations)
- Discrete dynamics (control decisions)



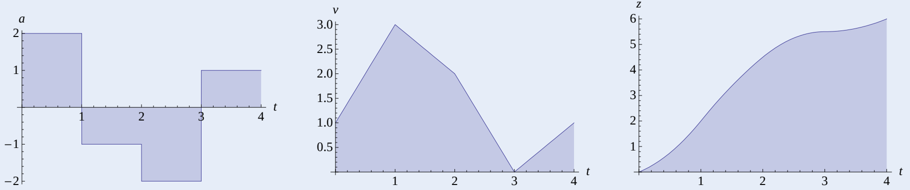




Q: I want to verify trains A: Hybrid systems Q: But there's uncertainties!

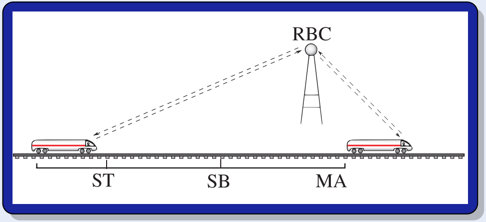
Challenge (Hybrid Systems)

- Continuous dynamics (differential equations)
- Discrete dynamics (control decisions)



Q: I want to verify uncertain trains

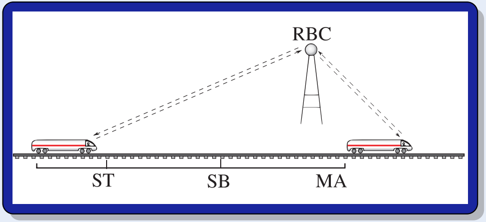
Challenge

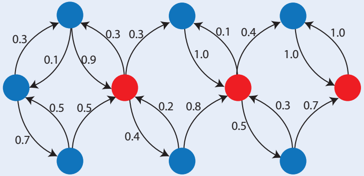


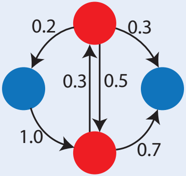
Q: I want to verify uncertain trains A: Markov chains

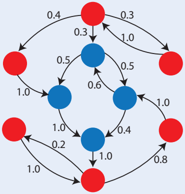
Challenge (Probabilistic Systems)

- Directed graph (Countable state space)
- Weighted edges (Transition probabilities)





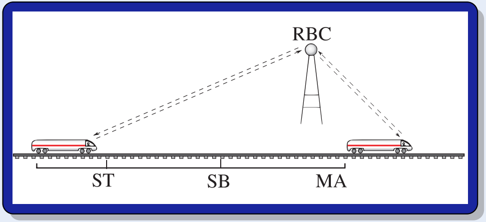


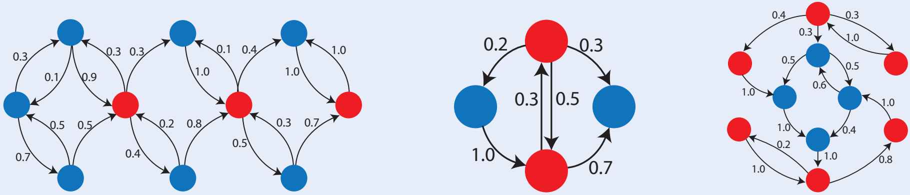


Q: I want to verify uncertain trains A: Markov chains Q: But trains move!

Challenge (Probabilistic Systems)

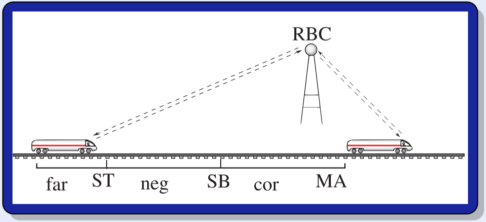
- Directed graph
  (Countable state space)
- Weighted edges
  (Transition probabilities)





Q: I want to verify uncertain systems

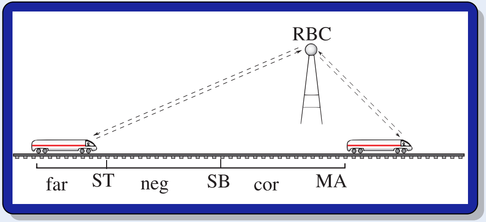
Challenge



Q: I want to verify uncertain systems A: Stochastic hybrid systems

Challenge (Stochastic Hybrid Systems)

- Continuous dynamics (differential equations)
- Discrete dynamics (control decisions)
- Stochastic dynamics (uncertainty)

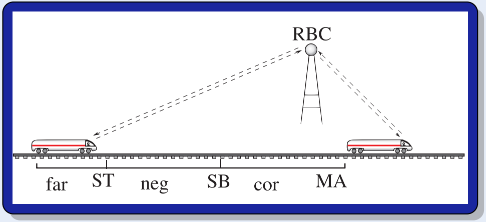


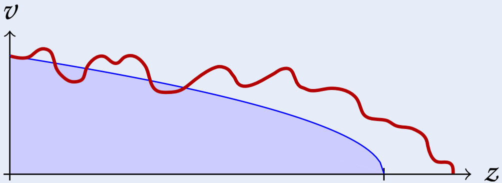
Q: I want to verify uncertain systems A: Stochastic hybrid systems

Challenge (Stochastic Hybrid Systems)

- Continuous dynamics (differential equations)
- Discrete dynamics (control decisions)
- Stochastic dynamics (uncertainty)

- Discrete stochastic (lossy communication)
- Continuous stochastic (wind, track)



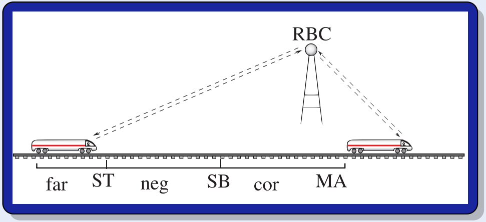


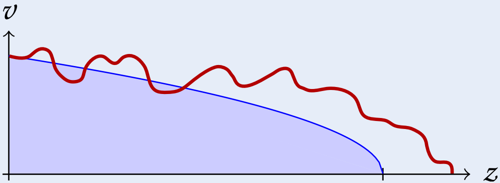
Q: I want to verify uncertain systems A: Stochastic hybrid systems Q: How?

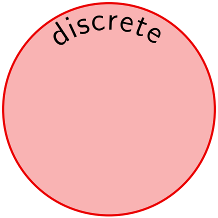
Challenge (Stochastic Hybrid Systems)

- Continuous dynamics (differential equations)
- Discrete dynamics (control decisions)
- Stochastic dynamics (uncertainty)

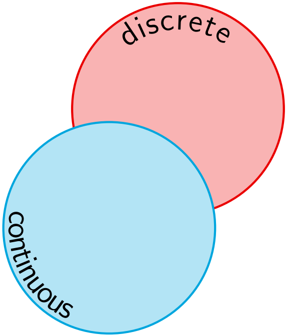
- Discrete stochastic (lossy communication)
- Continuous stochastic (wind, track)



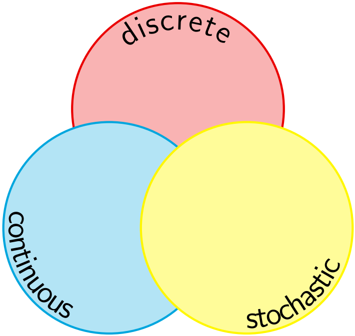



discrete

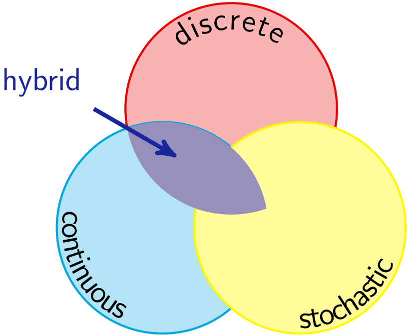
Cyber-Physical Systems

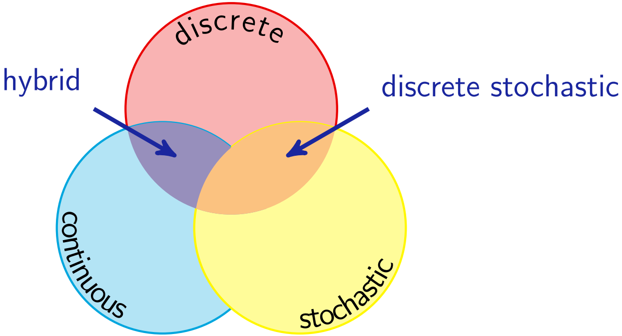


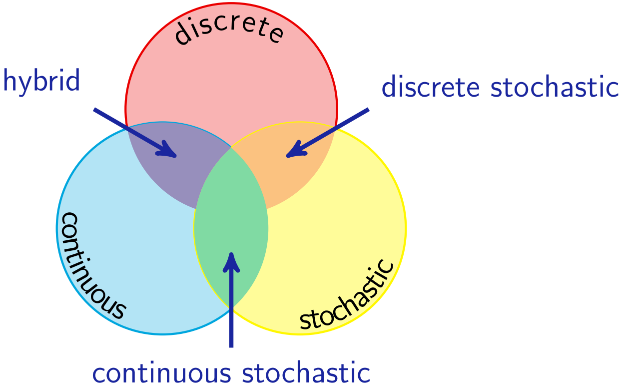
Cyber-Physical Systems

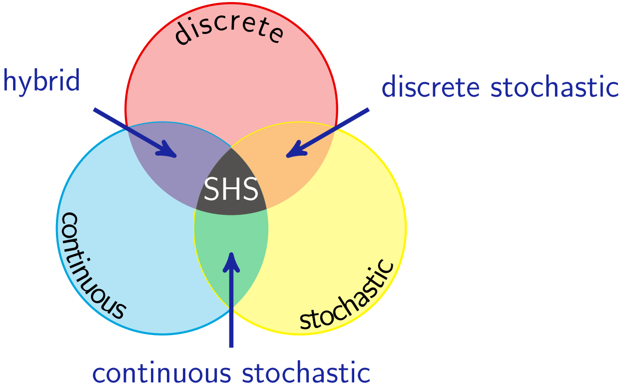


Cyber-Physical Systems



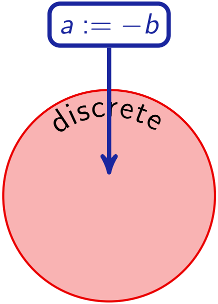

discrete
hybrid
discrete stochastic
continuous
stochastic


discrete
hybrid
discrete stochastic
continuous
stochastic
continuous stochastic

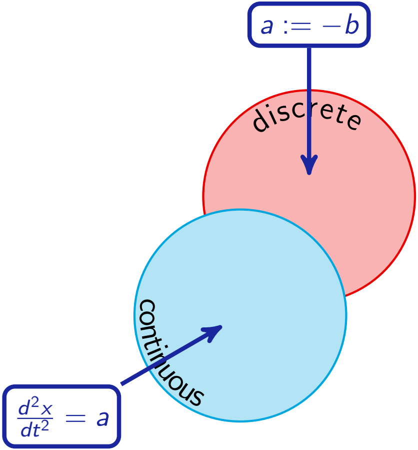


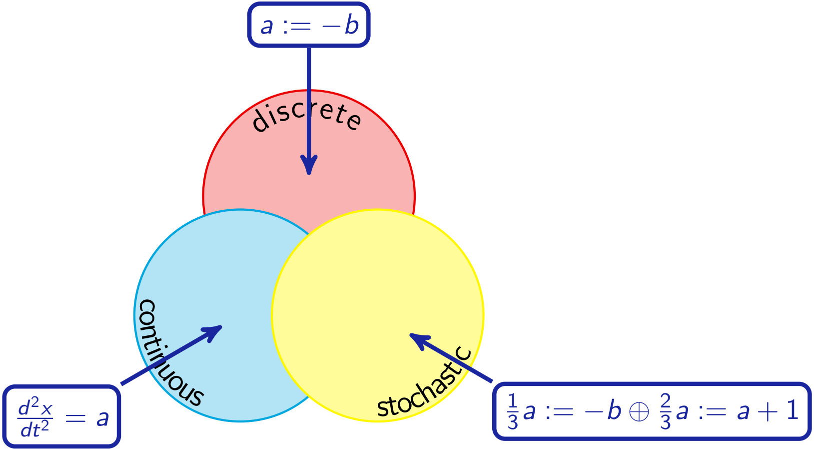
discrete
hybrid
discrete stochastic
SHS
continuous
stochastic
continuous stochastic

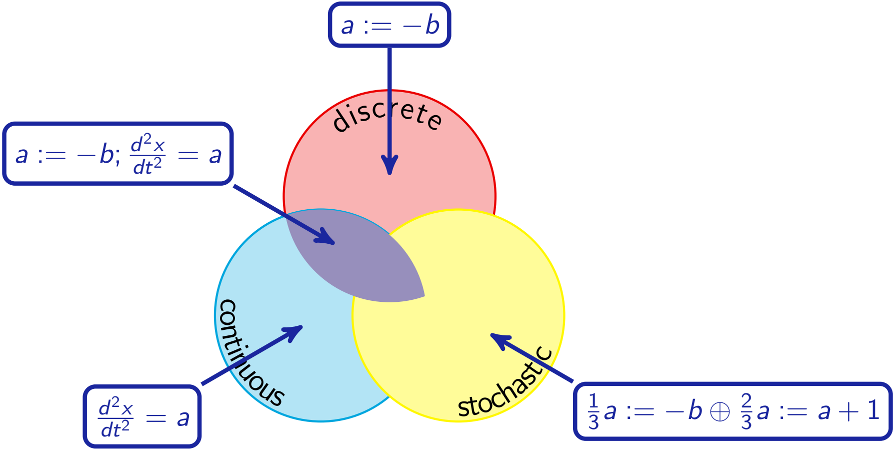
Model for Stochastic Hybrid Systems



Model for Stochastic Hybrid Systems

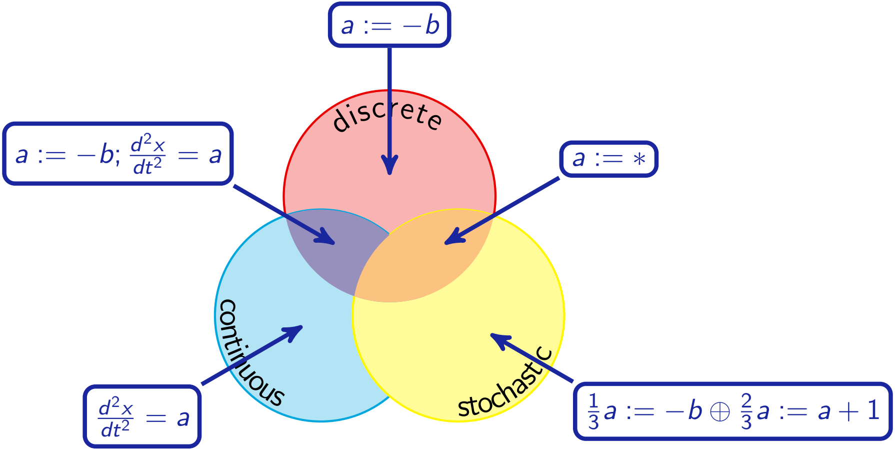



$a := -b$
discrete
continuous
stochastic
$\frac{d^2x}{dt^2} = a$
$\frac{1}{3} a := -b \oplus \frac{2}{3} a := a + 1$

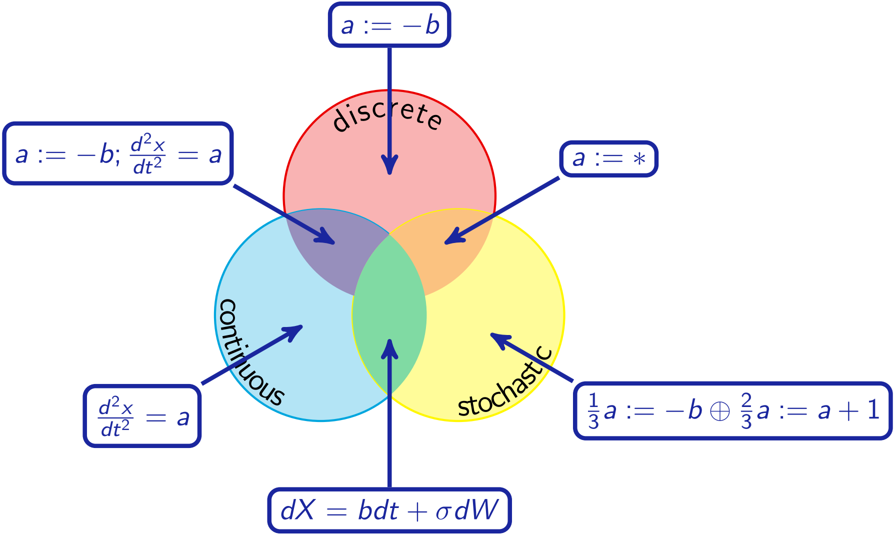


$a := -b$
discrete
$a := -b; \frac{d^2x}{dt^2} = a$
continuous
$\frac{d^2x}{dt^2} = a$
stochastic
$\frac{1}{3}a := -b \oplus \frac{2}{3}a := a + 1$

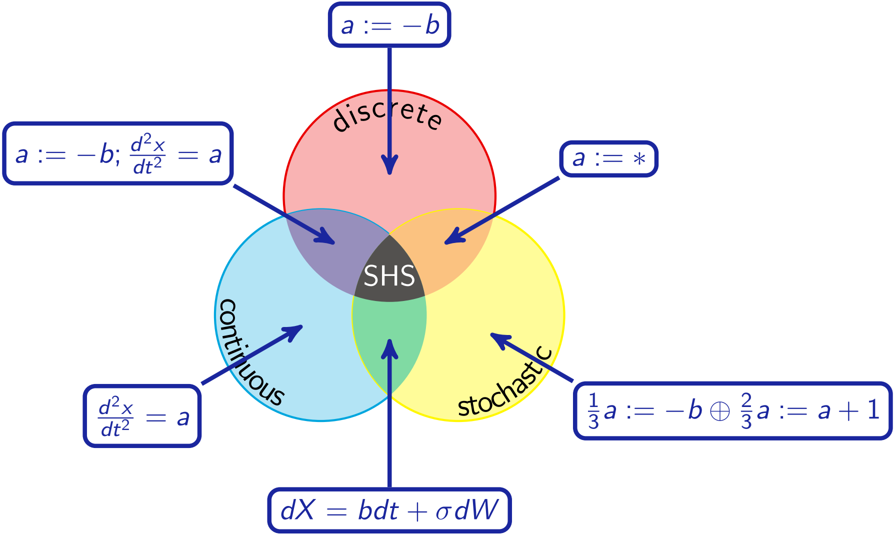
Model for Stochastic Hybrid Systems



Model for Stochastic Hybrid Systems

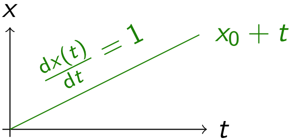


Model for Stochastic Hybrid Systems



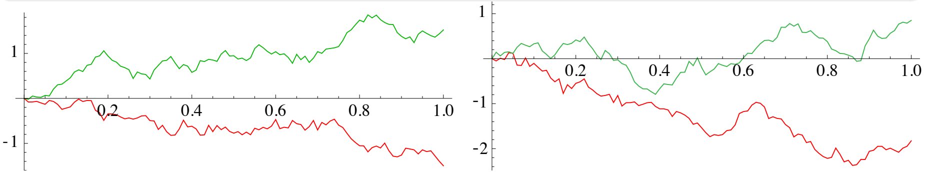
Definition (Ordinary differential equation (ODE))

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} = b(x(t)) \quad x(0) = x_0$$



Definition (Itō stochastic differential equation (SDE))

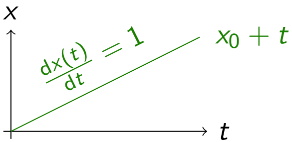
$$dX_t = b(X_t)dt + \sigma(X_t)dW_t \quad X_0 = Z$$



Stochastic Differential Equations (SDE)

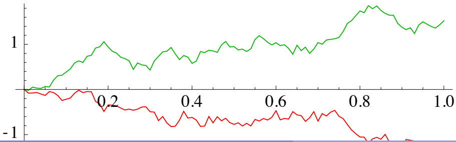
Definition (Ordinary differential equation (ODE))

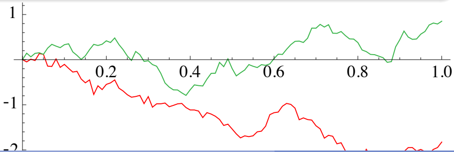
$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} = b(x(t)) \quad x(0) = x_0$$



Definition (Itō stochastic differential equation (SDE))

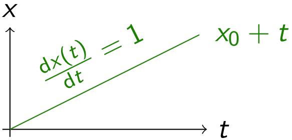
$$X_s = Z + \int_0^s dX_t = Z + \int_0^s b(X_t)dt + \int_0^s \sigma(X_t)dW_t$$





Definition (Ordinary differential equation (ODE))

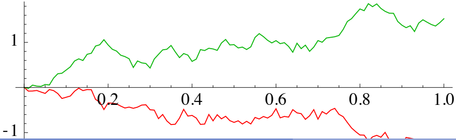
$$\frac{dx(t)}{dt} = b(x(t)) \quad x(0) = x_0$$

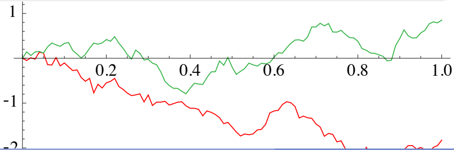


Calculus

Definition (Itō stochastic differential equation (SDE))

$$X_s = Z + \int_0^s dX_t = Z + \int_0^s b(X_t)dt + \int_0^s \sigma(X_t)dW_t$$

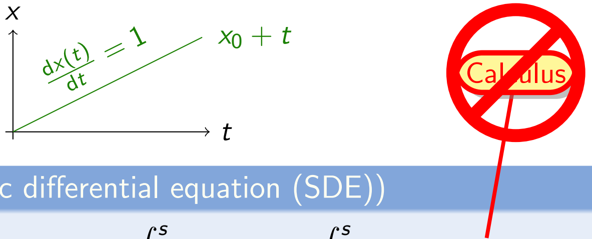




Stochastic Differential Equations (SDE)

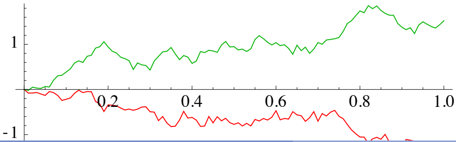
Definition (Ordinary differential equation (ODE))

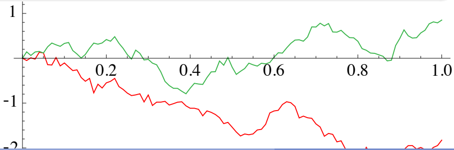
$$\frac{dx(t)}{dt} = b(x(t)) \quad x(0) = x_0$$



Definition (Itō stochastic differential equation (SDE))

$$X_s = Z + \int_0^s dX_t = Z + \int_0^s b(X_t)dt + \int_0^s \sigma(X_t)dW_t$$





Brownian Motion is Extremely Complex

Definition (Brownian motion W $\Rightarrow$ end of calculus)

1. $W_0 = 0$ (start at 0)
2. W_t almost surely continuous
3. $W_t - W_s \sim \mathcal{N}(0, t-s)$ (independent normal increments)

$\Rightarrow$ a.s. continuous everywhere but nowhere differentiable

$\Rightarrow$ a.s. unbounded variation, $\notin$ FV, nonmonotonic on every interval

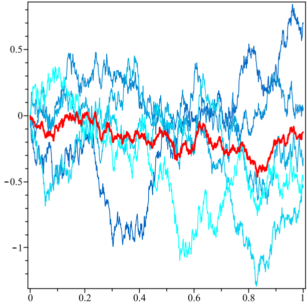
Brownian Motion is Extremely Complex

Definition (Brownian motion W $\Rightarrow$ end of calculus)

1. $W_0 = 0$ (start at 0)
2. W_t almost surely continuous
3. $W_t - W_s \sim \mathcal{N}(0, t-s)$ (independent normal increments)

$\Rightarrow$ a.s. continuous everywhere but nowhere differentiable

$\Rightarrow$ a.s. unbounded variation, $\notin$ FV, nonmonotonic on every interval

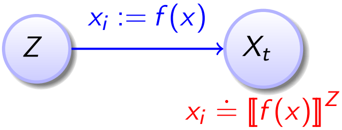


Stochastic Hybrid Programs: Syntax

Definition (Stochastic hybrid program α)

$x := \theta$	(assignment)	jump & test
$x := *$	(random assignment)	
$?H$	(conditional execution)	
$dx = bdt + \sigma dW \,\&\, H$	(SDE)	
$\alpha; \beta$	(seq. composition)	algebra
$\lambda\alpha \oplus \nu\beta$	(convex combination)	
α^*	(nondet. repetition)	

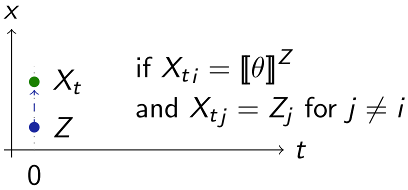
Stochastic Hybrid Program: Process Semantics

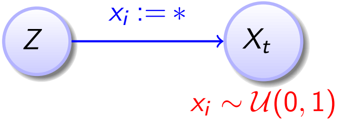


Definition (Stochastic hybrid program α: process semantics)

$$x_i := \theta = \hat{Y} \quad Y(\omega)_i = [\![\theta]\!]^{Z(\omega)} \text{ and } Y_j = Z_j \text{ (for } j \neq i\text{)}$$

$$(\!|x_i := \theta|\!)^Z = 0$$

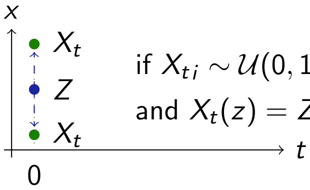




Definition (Stochastic hybrid program α: process semantics)

$x_i := * = \hat{U} \qquad U_i \sim \mathcal{U}(0,1)$ i.i.d. $\mathcal{F}_0$-measurable

$(\!|x_i := *|\!)^Z = 0$



if $X_{t\,i} \sim \mathcal{U}(0,1)$

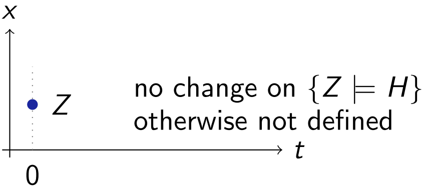
and $X_t(z) = Z(z)$ for $z \neq x$

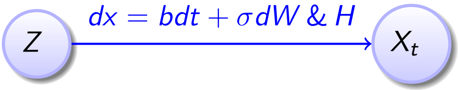
on $\{Z \models H\}$

Definition (Stochastic hybrid program α: process semantics)

$$?H = \hat{Z} \quad \text{on the event } \{Z \models H\}$$

$$(\!|?H|\!)^Z = 0$$

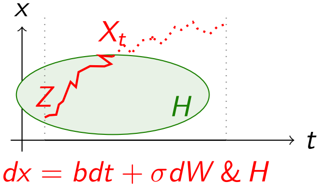


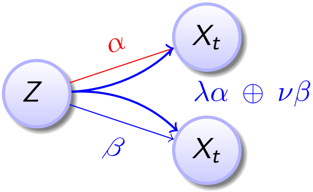


Definition (Stochastic hybrid program α: process semantics)

$dx = bdt + \sigma dW \,\&\, H$ solves $dX = [\![b]\!]^X dt + [\![\sigma]\!]^X dB_t, X_0 = Z$

$(\!|dx = bdt + \sigma dW \,\&\, H|\!)^Z = \inf\{t \geq 0 \;:\; X_t \notin H\}$

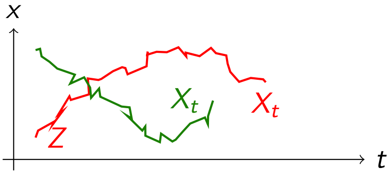


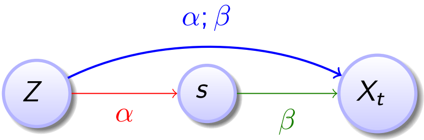


Definition (Stochastic hybrid program α: process semantics)

$$\lambda\alpha \oplus \nu\beta = \mathcal{I}_{U\leq\lambda}\alpha + \mathcal{I}_{U>\lambda}\beta = \begin{cases} \alpha & \text{on event } \{U \leq \lambda\} \\ \beta & \text{on event } \{U > \lambda\} \end{cases}$$

$(\!|\lambda\alpha \oplus \nu\beta|\!)^Z = \mathcal{I}_{U\leq\lambda}(\!|\alpha|\!)^Z + \mathcal{I}_{U>\lambda}(\!|\beta|\!)^Z$ with i.i.d. $U \sim \mathcal{U}(0,1)$, $\mathcal{F}_0$-meas

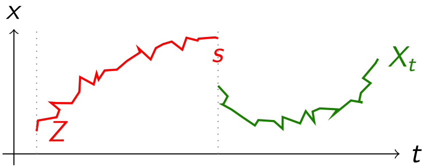


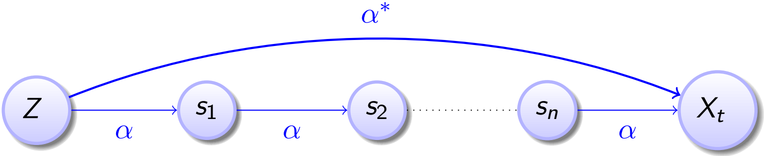


Definition (Stochastic hybrid program α: process semantics)

$$\alpha;\beta = \begin{cases} \alpha & \text{on event } \{t < (\!|\alpha|\!)^Z\} \\ \beta & \text{on event } \{t \geq (\!|\alpha|\!)^Z\} \end{cases}$$

$$(\!|\alpha;\beta|\!)^Z = (\!|\alpha|\!)^Z + (\!|\beta|\!)^\alpha$$

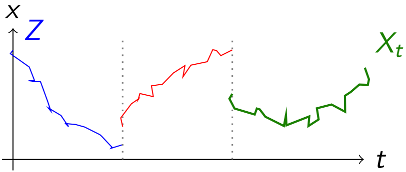


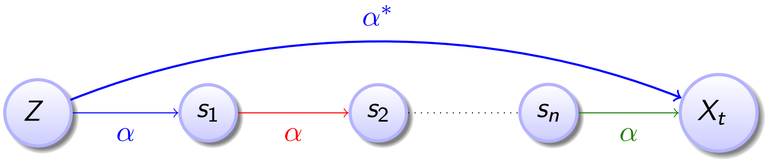


Definition (Stochastic hybrid program α: process semantics)

$$\alpha^* = \alpha^n \text{ on event } \{(\!|\alpha^n|\!)^Z > t\}$$

$$(\!|\alpha^*|\!)^Z = \lim_{n\to\infty} (\!|\alpha^n|\!)^Z$$

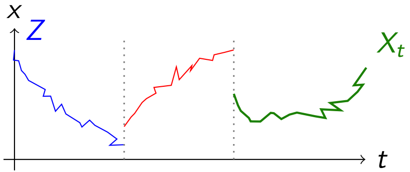




Definition (Stochastic hybrid program α: process semantics)

$$\alpha^* = \alpha^n \text{ on event } \{(\!|\alpha^n|\!)^Z > t\}$$

$$(\!|\alpha^*|\!)^Z = \lim_{n\to\infty} (\!|\alpha^n|\!)^Z \qquad \text{monotone!}$$



Stochastic Differential Dynamic Logic Sd$\mathcal{L}$: Syntax

Definition (Sd$\mathcal{L}$ term f)

F	(primitive measurable function, e.g., characteristic $\mathcal{I}_A$)
$\lambda f + \nu g$	(linear term)
Bf	(scalar term for boolean term B)
$\langle\alpha\rangle f$	(reachable)

Definition (Sd$\mathcal{L}$ formula ϕ)

$$\phi \ ::= \ f \leq g \mid f = g$$

Definition (Measurable semantics)

Definition (Measurable semantics)

$$[\![F]\!]^Z = F^\ell(Z) \text{ i.e., } [\![F]\!]^Z(\omega) = F^\ell(Z(\omega))$$

Definition (Measurable semantics)

$$[\![F]\!]^Z = F^\ell(Z) \text{ i.e., } [\![F]\!]^Z(\omega) = F^\ell(Z(\omega))$$

$$[\![\lambda f + \nu g]\!]^Z = \lambda [\![f]\!]^Z + \nu [\![g]\!]^Z$$

Stochastic Differential Dynamic Logic Sd$\mathcal{L}$: Semantics

Definition (Measurable semantics)

$$
\begin{aligned}
[\![F]\!]^Z &= F^\ell(Z) \text{ i.e., } [\![F]\!]^Z(\omega) = F^\ell(Z(\omega)) \\
[\![\lambda f + \nu g]\!]^Z &= \lambda [\![f]\!]^Z + \nu [\![g]\!]^Z \\
[\![Bf]\!]^Z &= [\![B]\!]^Z * [\![f]\!]^Z \text{ i.e., } [\![Bf]\!]^Z(\omega) = [\![B]\!]^Z(\omega)[\![f]\!]^Z(\omega)
\end{aligned}
$$

Definition (Measurable semantics)

$$[\![F]\!]^Z = F^\ell(Z) \text{ i.e., } [\![F]\!]^Z(\omega) = F^\ell(Z(\omega))$$

$$[\![\lambda f + \nu g]\!]^Z = \lambda[\![f]\!]^Z + \nu[\![g]\!]^Z$$

$$[\![Bf]\!]^Z = [\![B]\!]^Z * [\![f]\!]^Z \text{ i.e., } [\![Bf]\!]^Z(\omega) = [\![B]\!]^Z(\omega)[\![f]\!]^Z(\omega)$$

$$[\![\langle\alpha\rangle f]\!]^Z = \sup\{[\![f]\!]^\alpha \ : \ 0 \le t \le (\!|\alpha|\!)^Z\}$$

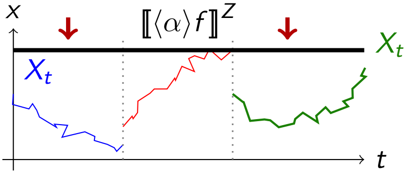
Definition (Measurable semantics)

$$[\![F]\!]^Z = F^\ell(Z) \text{ i.e., } [\![F]\!]^Z(\omega) = F^\ell(Z(\omega))$$

$$[\![\lambda f + \nu g]\!]^Z = \lambda[\![f]\!]^Z + \nu[\![g]\!]^Z$$

$$[\![Bf]\!]^Z = [\![B]\!]^Z * [\![f]\!]^Z \text{ i.e., } [\![Bf]\!]^Z(\omega) = [\![B]\!]^Z(\omega)[\![f]\!]^Z(\omega)$$

$$[\![\langle\alpha\rangle f]\!]^Z = \sup\{[\![f]\!]^{\alpha} \; : \; 0 \leq t \leq (\!|\alpha|\!)^Z\}$$



Well-definedness of Sd$\mathcal{L}$ Semantics

Theorem (Measurable)

$[\![f]\!]^Z$ is a random variable (i.e., measurable) for any random variable Z and Sd$\mathcal{L}$ term f.

Well-definedness of Sd$\mathcal{L}$ Semantics

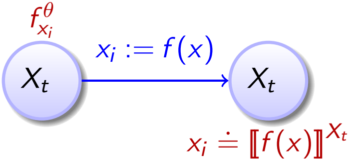
Theorem (Measurable)

$[\![f]\!]^Z$ *is a random variable (i.e., measurable) for any random variable Z and* Sd$\mathcal{L}$ *term f.*

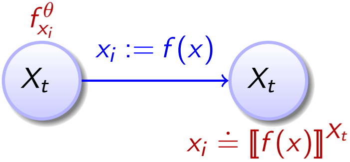
Corollary (Pushforward measure well-defined for Borel-measurable S)

$$S \mapsto P(([\![f]\!]^Z)^{-1}(S)) = P(\{\omega \in \Omega : [\![f]\!]^Z(\omega) \in S\}) = P([\![f]\!]^Z \in S)$$

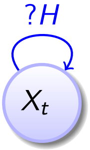
$$\langle x_i := \theta \rangle f \;=\; f_{x_i}^{\theta}$$



$\langle x_i := \theta \rangle f \;=\; f_{x_i}^{\theta}$

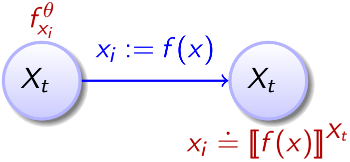


$\langle ?H \rangle f \;=\; Hf$

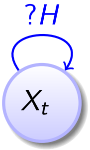


on $\{X_t \models H\}$

$\langle x_i := \theta \rangle f = f_{x_i}^{\theta}$



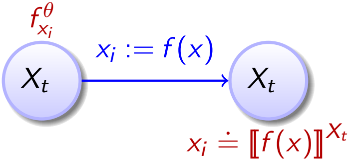
$\langle ?H \rangle f = Hf$



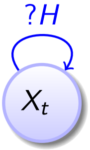
on $\{X_t \models H\}$

$\langle \alpha \rangle (\lambda f) = \lambda \langle \alpha \rangle f$

$$\langle x_i := \theta \rangle f = f_{x_i}^{\theta}$$



$$\langle ?H \rangle f = Hf$$



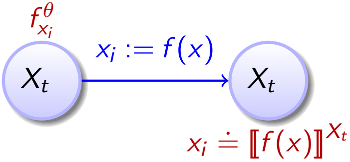
on $\{X_t \models H\}$

$$\langle \alpha \rangle (\lambda f) = \lambda \langle \alpha \rangle f$$

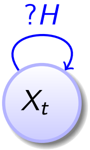
$$\langle \alpha \rangle (\lambda f + \nu g) \leq \lambda \langle \alpha \rangle f + \nu \langle \alpha \rangle g$$

Proof Calculus: Stochastic Differential Dynamic Logic

$$\langle x_i := \theta \rangle f \;=\; f_{x_i}^{\theta}$$



$$\langle ?H \rangle f \;=\; Hf$$



on $\{X_t \models H\}$

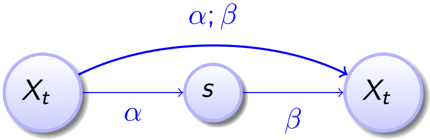
$$\langle \alpha \rangle (\lambda f) \;=\; \lambda \langle \alpha \rangle f$$

$$\langle \alpha \rangle (\lambda f + \nu g) \;\leq\; \lambda \langle \alpha \rangle f + \nu \langle \alpha \rangle g$$

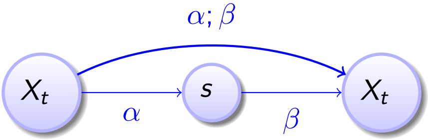
$$f \leq g \models \langle \alpha \rangle f \leq \langle \alpha \rangle g$$

Proof Calculus: Stochastic Differential Dynamic Logic

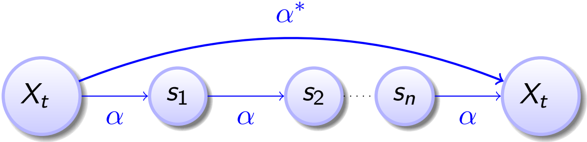
$$\langle \alpha; \beta \rangle f \ \leq \ \langle \alpha \rangle (f \sqcup \langle \beta \rangle f)$$



$$\langle\alpha;\beta\rangle f \;\le\; \langle\alpha\rangle(f \sqcup \langle\beta\rangle f)$$

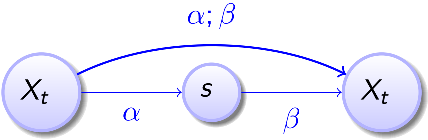


$$\langle\alpha\rangle f \le f \vDash \langle\alpha^*\rangle f \le f$$

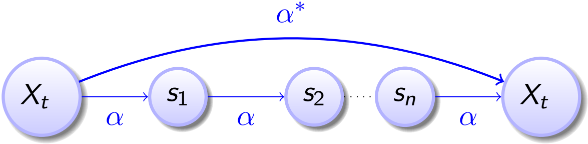


Proof Calculus: Stochastic Differential Dynamic Logic

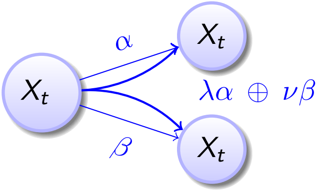
$\langle \alpha; \beta \rangle f \;\leq\; \langle \alpha \rangle (f \sqcup \langle \beta \rangle f)$



$\langle \alpha \rangle f \leq f \vDash \langle \alpha^* \rangle f \leq f$



$$P(\langle \lambda\alpha \oplus \nu\beta \rangle f \in S) = \lambda P(\langle \alpha \rangle f \in S) + \nu P(\langle \beta \rangle f \in S)$$



Soundness

Theorem (Soundness)

1. *Rules are globally sound pathwise, i.e.,* $f_i \leq g_i \vDash f \leq g$ *holds for each initial* Z *pathwise for each* $\omega \in \Omega$
2. $\langle \oplus \rangle$ *is sound in distribution*

Soundness

Theorem (Soundness)

1. *Rules are globally sound pathwise, i.e.,* $f_i \leq g_i \vDash f \leq g$ *holds for each initial* Z *pathwise for each* $\omega \in \Omega$
2. $\langle \oplus \rangle$ *is sound in distribution*

Theorem (Stochastic Differential Invariants)

Let $\lambda > 0$, $\phi \in C_C^2(\mathbb{R}^d, \mathbb{R})$ *compact support on* H *(e.g.,* H *bounded)*

$$\frac{\langle\alpha\rangle(H \rightarrow \phi) \leq \lambda p \quad H \rightarrow \phi \geq 0 \quad H \rightarrow Lf \leq 0}{P(\langle\alpha\rangle\langle dx = bdt + \sigma dW \,\&\, H\rangle\phi \geq \lambda) \leq p} \quad \textit{sound}$$

Soundness

Theorem (Stochastic Differential Invariants)

Let $\lambda > 0$, $\phi \in C_C^2(\mathbb{R}^d, \mathbb{R})$ *compact support on* H *(e.g.,* H *bounded)*

$$\frac{\langle\alpha\rangle(H \to \phi) \leq \lambda p \quad H \to \phi \geq 0 \quad H \to Lf \leq 0}{P(\langle\alpha\rangle\langle dx = bdt + \sigma dW \,\&\, H\rangle \phi \geq \lambda) \leq p} \quad \textit{sound}$$

Theorem (Dynkin for càdlàg strong Markov X_t and $\phi \in C_C^2(\mathbb{R}^d, \mathbb{R})$)

$$Af(x) := \lim_{t \searrow 0} \frac{E^x f(X_t) - f(x)}{t} \overset{E^x \tau < \infty}{\Rightarrow} E^x f(X_\tau) = f(x) + E^x \int_0^\tau Af(X_s) ds$$

Soundness

Theorem (Stochastic Differential Invariants)

Let $\lambda > 0$, $\phi \in C_C^2(\mathbb{R}^d, \mathbb{R})$ *compact support on* H *(e.g.,* H *bounded)*

$$\frac{\langle\alpha\rangle(H \to \phi) \leq \lambda p \quad H \to \phi \geq 0 \quad H \to Lf \leq 0}{P(\langle\alpha\rangle\langle dx = bdt + \sigma dW \,\&\, H\rangle\phi \geq \lambda) \leq p} \quad \textit{sound}$$

Theorem (Dynkin for càdlàg strong Markov X_t and $\phi \in C_C^2(\mathbb{R}^d, \mathbb{R})$)

$$Af(x) := \lim_{t \searrow 0} \frac{E^x f(X_t) - f(x)}{t} \overset{E^x\tau < \infty}{\Rightarrow} E^x f(X_\tau) = f(x) + E^x \int_0^\tau Af(X_s)ds$$

Theorem (Differential generator for SDE solution and $\phi \in C_C^2(\mathbb{R}^d, \mathbb{R})$)

$$A\phi = L\phi := b\nabla f + \frac{\sigma\sigma^T}{2}\nabla\nabla f$$

Soundness

Theorem (Stochastic Differential Invariants)

Let $\lambda > 0$, $\phi \in C_C^2(\mathbb{R}^d, \mathbb{R})$ *compact support on* H *(e.g.,* H *bounded)*

$$\frac{\langle\alpha\rangle(H \to \phi) \leq \lambda p \quad H \to \phi \geq 0 \quad H \to Lf \leq 0}{P(\langle\alpha\rangle\langle dx = bdt + \sigma dW \,\&\, H\rangle\phi \geq \lambda) \leq p} \quad \textit{sound}$$

Theorem (Dynkin for càdlàg strong Markov X_t and $\phi \in C_C^2(\mathbb{R}^d, \mathbb{R})$)

$$Af(x) := \lim_{t \searrow 0} \frac{E^x f(X_t) - f(x)}{t} \overset{E^x \tau < \infty}{\Rightarrow} E^x f(X_\tau) = f(x) + E^x \int_0^\tau Af(X_s) ds$$

Theorem (Differential generator for SDE solution and $\phi \in C_C^2(\mathbb{R}^d, \mathbb{R})$)

$$A\phi = L\phi := b\nabla f + \frac{\sigma\sigma^T}{2} \nabla\nabla f = \sum_i b_i \frac{\partial f}{\partial x_i} + \frac{1}{2} \sum_{i,j} (\sigma\sigma^T)_{i,j} \frac{\partial^2 f}{\partial x_i \partial x_j}$$

Theorem (Stochastic Differential Invariants)

Let $\lambda > 0$, $\phi \in C_C^2(\mathbb{R}^d, \mathbb{R})$ *compact support on H (e.g., H bounded)*

$$\frac{\langle\alpha\rangle(H \to \phi) \leq \lambda p \quad H{\to}\phi \geq 0 \quad H{\to}Lf \leq 0}{P(\langle\alpha\rangle\langle dx = bdt + \sigma dW \,\&\, H\rangle\phi \geq \lambda) \leq p} \quad \textit{sound}$$

Theorem (Dynkin for càdlàg strong Markov X_t and $\phi \in C_C^2(\mathbb{R}^d, \mathbb{R})$)

$$Af(x) := \lim_{t \searrow 0} \frac{E^x f(X_t) - f(x)}{t} \overset{E^x\tau<\infty}{\Rightarrow} E^x f(X_\tau) = f(x) + E^x \int_0^\tau Af(X_s)ds$$

$$A\phi(X_s) = L\phi(X_s) \leq 0 \text{ on } H \quad \Rightarrow \quad E^x\phi(X_\tau) \leq \phi(x) \forall x, \tau$$

$$\Rightarrow \quad P^x\text{-a.s. } E^x(\phi(X_t)|\mathcal{F}_s) = E^{X_s}\phi(X_{t-s}) \leq \phi(X_s)$$

$$\Rightarrow \quad X_t \text{ supermartingale}$$

Soundness

Theorem (Stochastic Differential Invariants)

Let $\lambda > 0$, $\phi \in C_C^2(\mathbb{R}^d, \mathbb{R})$ *compact support on* H *(e.g.,* H *bounded)*

$$\frac{\langle\alpha\rangle(H \to \phi) \le \lambda p \quad H \to \phi \ge 0 \quad H \to Lf \le 0}{P(\langle\alpha\rangle\langle dx = bdt + \sigma dW \,\&\, H\rangle\phi \ge \lambda) \le p} \quad \textit{sound}$$

Theorem (Dynkin for càdlàg strong Markov X_t and $\phi \in C_C^2(\mathbb{R}^d, \mathbb{R})$)

$$Af(x) := \lim_{t \searrow 0} \frac{E^x f(X_t) - f(x)}{t} \quad \overset{E^x \tau < \infty}{\Rightarrow} \quad E^x f(X_\tau) = f(x) + E^x \int_0^\tau Af(X_s) ds$$

Theorem (Doob maximal martingale ineq., càdlàg supermartingale)

$$\forall\ f \ge 0, \lambda > 0 \quad P\left(\sup_{t \ge 0} f(X_t) \ge \lambda \,\middle|\, \mathcal{F}_0\right) \le \frac{Ef(X_0)}{\lambda}$$

Soundness

Theorem (Stochastic Differential Invariants)

Let $\lambda > 0$, $\phi \in C_C^2(\mathbb{R}^d, \mathbb{R})$ *compact support on* H *(e.g.,* H *bounded)*

$$\frac{\langle\alpha\rangle(H \to \phi) \le \lambda p \quad H \to \phi \ge 0 \quad H \to Lf \le 0}{P(\langle\alpha\rangle\langle dx = bdt + \sigma dW \,\&\, H\rangle \phi \ge \lambda) \le p} \ \textit{sound}$$

Theorem (Dynkin for càdlàg strong Markov X_t and $\phi \in C_C^2(\mathbb{R}^d, \mathbb{R})$)

$$Af(x) := \lim_{t \searrow 0} \frac{E^x f(X_t) - f(x)}{t} \overset{E^x \tau < \infty}{\Rightarrow} E^x f(X_\tau) = f(x) + E^x \int_0^\tau Af(X_s) ds$$

Theorem (Doob maximal martingale ineq., càdlàg supermartingale)

$$\forall\ f \ge 0, \lambda > 0 \quad P\left(\sup_{t \ge 0} f(X_t) \ge \lambda \,|\, \mathcal{F}_0\right) \le \frac{Ef(X_0)}{\lambda} \le \frac{\lambda p}{\lambda} = p$$

Proof Example

$$\frac{\langle\alpha\rangle(H \to \phi) \le \lambda p \quad H \to \phi \ge 0 \quad H \to Lf \le 0}{P(\langle\alpha\rangle\langle dx = bdt + \sigma dW \,\&\, H\rangle\phi \ge \lambda) \le p}$$

$$\langle ?x^2 + y^2 \le \frac{1}{3}\rangle(H \to \phi) = \left(H \to x^2 + y^2 \le \frac{1}{3}\right)(x^2 + y^2) \le 1 * \frac{1}{3}$$

$$\phi \equiv x^2 + y^2 \ge 0 \qquad \text{with} \qquad H \equiv x^2 + y^2 < 10$$

$$\frac{L\phi = \frac{1}{2}\left(-x\frac{\partial\phi}{\partial x} - y\frac{\partial\phi}{\partial y} + y^2\frac{\partial^2\phi}{\partial x^2} - 2xy\frac{\partial^2\phi}{\partial x\partial y} + x^2\frac{\partial^2\phi}{\partial y^2}\right) \le 0}{P(\langle ?x^2 + y^2 \le \frac{1}{3}; dx = -\frac{x}{2}dt - ydW, dy = -\frac{y}{2}dt + xdW \,\&\, H\rangle x^2 + y^2 \ge 1)}$$

$$\le \qquad \text{(by ??)}$$

$$P(\langle ?x^2 + y^2 \le \frac{1}{3}\rangle\langle dx = -\frac{x}{2}dt - ydW, dy = -\frac{y}{2}dt + xdW \,\&\, H\rangle x^2 + y^2 \ge 1$$

$$\le \frac{1}{3}$$

Outline Background

Image Computation

Problem (Image Computation – generic)

Do transitions of system H reach bad state in B from an initial state in I?

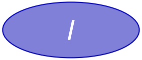
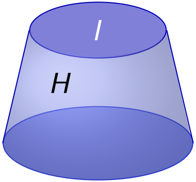


Image Computation

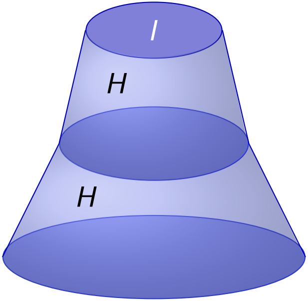
Problem (Image Computation – generic)

Do transitions of system H reach bad state in B from an initial state in I?



Problem (Image Computation – generic)

Do transitions of system H reach bad state in B from an initial state in I?



Problem (Image Computation – generic)

Do transitions of system H reach bad state in B from an initial state in I?

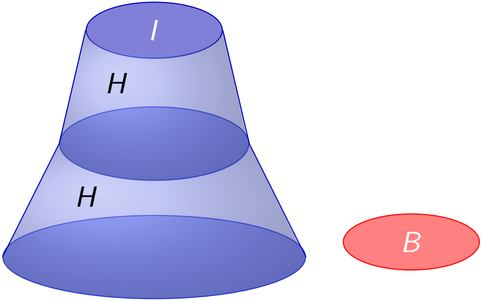
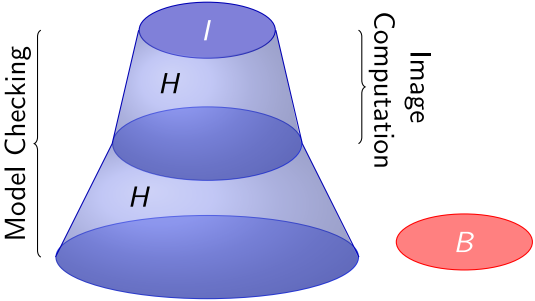


Image Computation

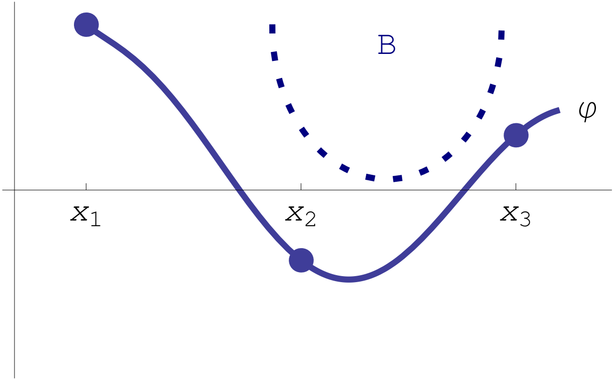
Problem (Image Computation – generic)

Do transitions of system H reach bad state in B from an initial state in I?



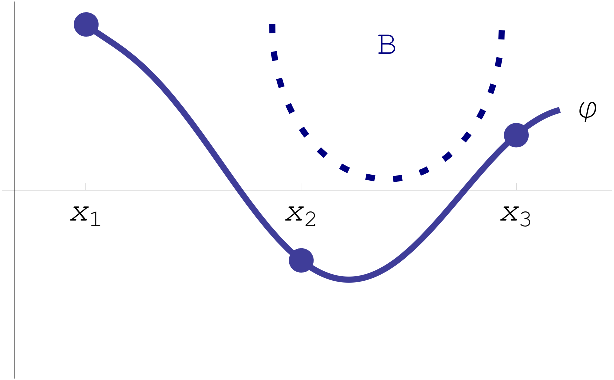
Problem (Image Computation – continuous transition)

Flow $\varphi : [0,\infty) \times \mathbb{R}^n \to \mathbb{R}^n$ reaches state B, i.e., $\exists t, x_0 : \varphi(t, x_0) \in B$?



Problem (Image Computation – continuous transition)

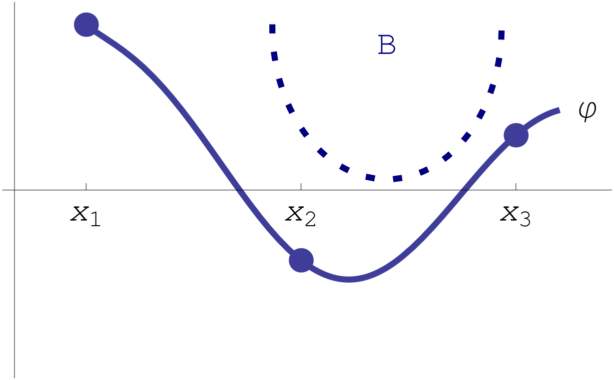
Flow $\varphi : [0,\infty) \times \mathbb{R}^n \to \mathbb{R}^n$ reaches state B, i.e., $\exists t, x_0 : \varphi(t, x_0) \in B$?



Idea: Sample points

Problem (Image Computation – continuous transition)

Flow $\varphi : [0, \infty) \times \mathbb{R}^n \to \mathbb{R}^n$ reaches state B, i.e., $\exists t, x_0 : \ \varphi(t, x_0) \in B$?

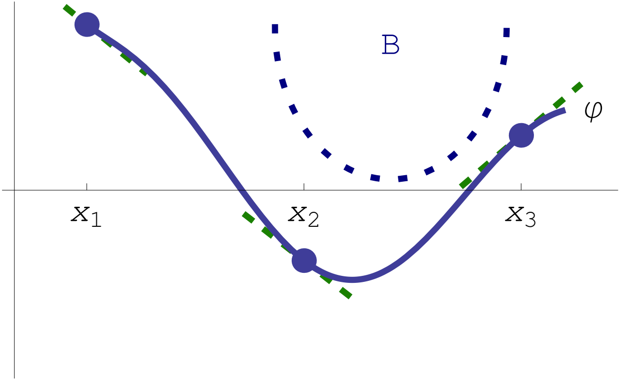


Idea: Sample points

too many!

Problem (Image Computation – continuous transition)

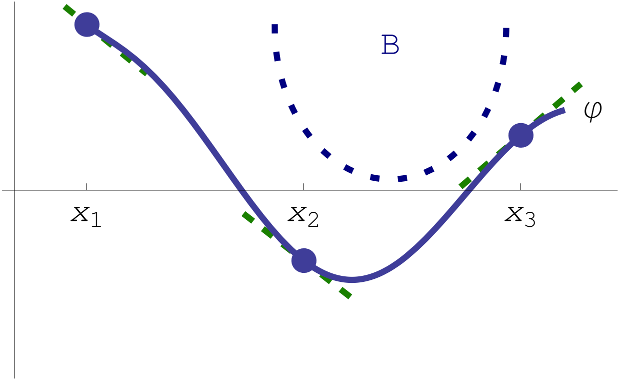
Flow $\varphi : [0,\infty) \times \mathbb{R}^n \to \mathbb{R}^n$ reaches state B, i.e., $\exists t, x_0 : \ \varphi(t, x_0) \in B$?



Idea: Sample points & derivatives

Problem (Image Computation – continuous transition)

Flow $\varphi : [0, \infty) \times \mathbb{R}^n \to \mathbb{R}^n$ reaches state B, i.e., $\exists t, x_0 : \varphi(t, x_0) \in B$?

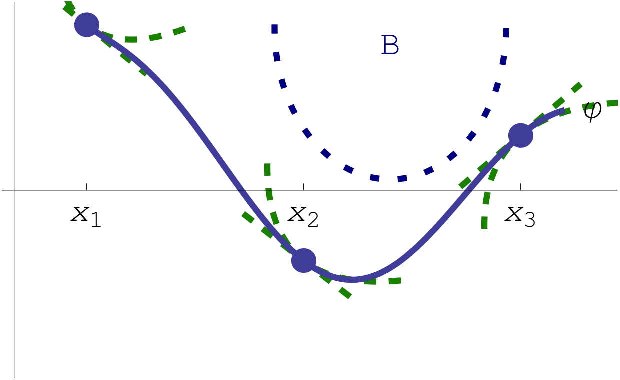


Idea: Sample points & derivatives

too many!

Problem (Image Computation – continuous transition)

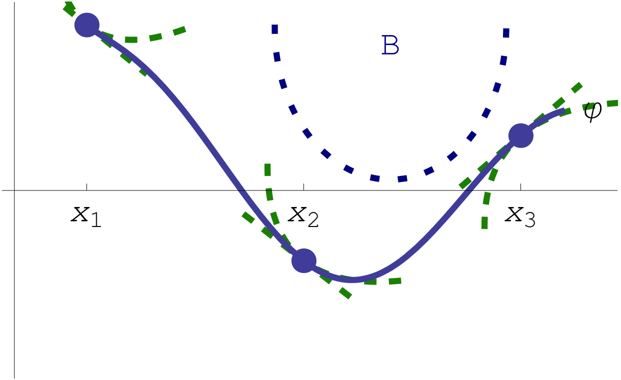
Flow $\varphi : [0,\infty) \times \mathbb{R}^n \to \mathbb{R}^n$ reaches state B, i.e., $\exists t, x_0 : \varphi(t, x_0) \in B$?



Idea: Sample points & derivatives 1&2

Problem (Image Computation – continuous transition)

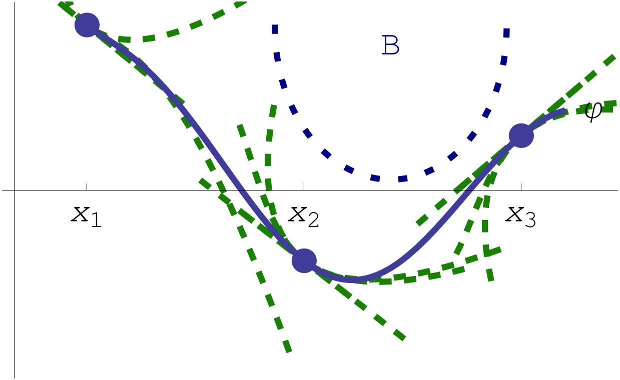
Flow $\varphi : [0, \infty) \times \mathbb{R}^n \to \mathbb{R}^n$ reaches state B, i.e., $\exists t, x_0 : \varphi(t, x_0) \in B$?



Idea: Sample points & derivatives 1&2 too many!

Problem (Image Computation – continuous transition)

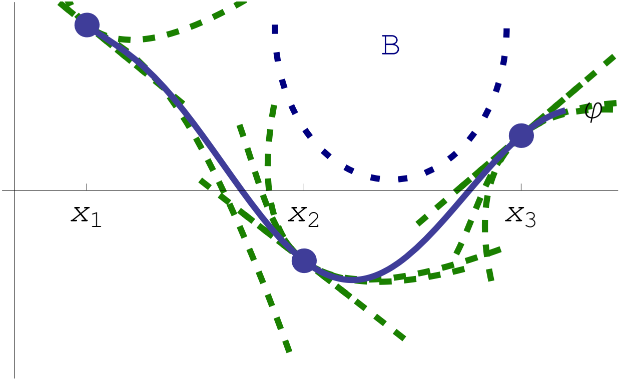
Flow $\varphi : [0,\infty) \times \mathbb{R}^n \to \mathbb{R}^n$ reaches state B, i.e., $\exists t, x_0 : \varphi(t, x_0) \in B$?



Idea: Sample points & derivatives 1&2&3

Problem (Image Computation – continuous transition)

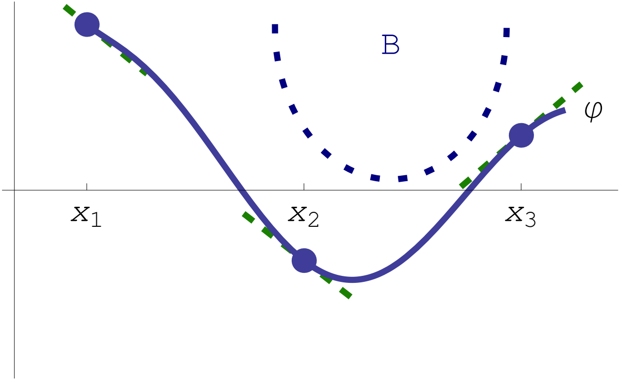
Flow $\varphi : [0,\infty) \times \mathbb{R}^n \to \mathbb{R}^n$ reaches state B, i.e., $\exists t, x_0 : \ \varphi(t, x_0) \in B$?



Idea: Sample points & derivatives 1&2&3 too many!

Problem (Image Computation – continuous transition)

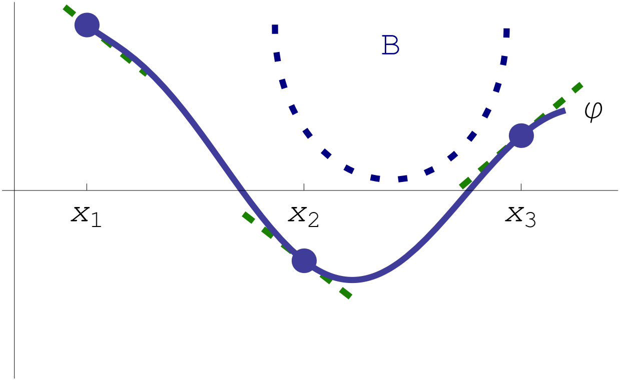
Flow $\varphi : [0,\infty) \times \mathbb{R}^n \to \mathbb{R}^n$ reaches state B, i.e., $\exists t, x_0 : \ \varphi(t, x_0) \in B$?



Idea: Sample points & X curve & blow up to regions & ...

Problem (Image Computation – continuous transition)

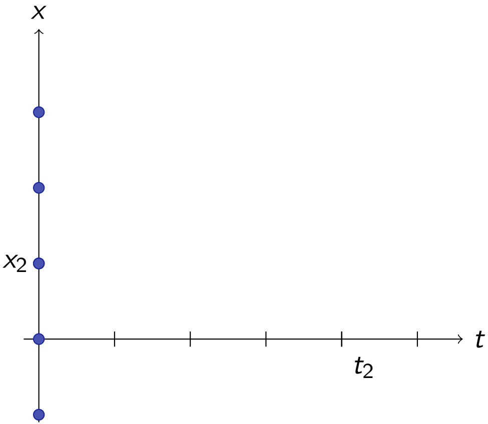
Flow $\varphi : [0,\infty) \times \mathbb{R}^n \to \mathbb{R}^n$ reaches state B, i.e., $\exists t, x_0 : \ \varphi(t, x_0) \in B$?



Idea: Sample points & X curve & blow up to regions & . . . too many!

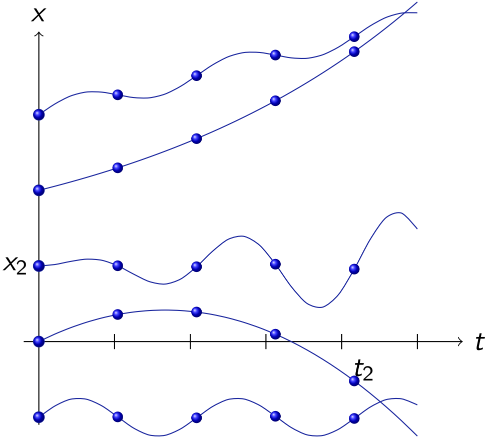
Problem (Image Computation – ODE transition)

Flow $\varphi : [0, \infty) \times \mathbb{R}^n \to \mathbb{R}^n$ solving $x' = f(x)$ reaches state B?



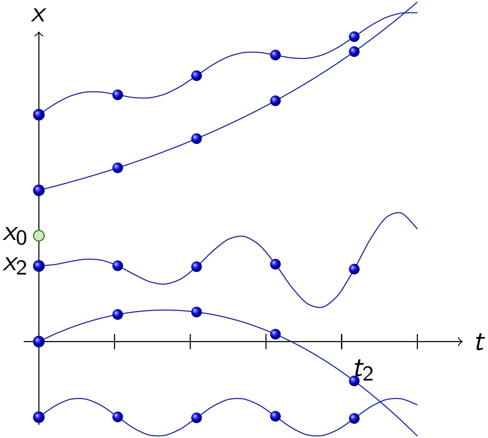
Problem (Image Computation – ODE transition)

Flow $\varphi : [0, \infty) \times \mathbb{R}^n \to \mathbb{R}^n$ solving $x' = f(x)$ reaches state B?



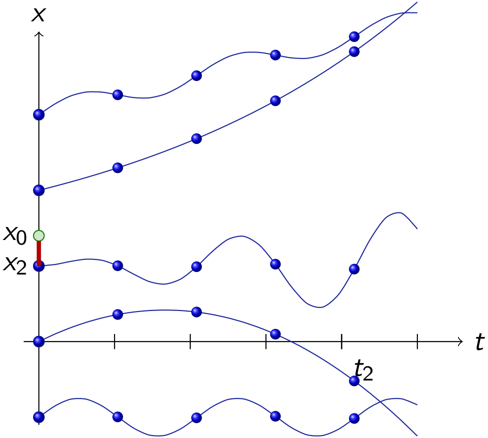
Problem (Image Computation – ODE transition)

Flow $\varphi : [0, \infty) \times \mathbb{R}^n \to \mathbb{R}^n$ solving $x' = f(x)$ reaches state B?



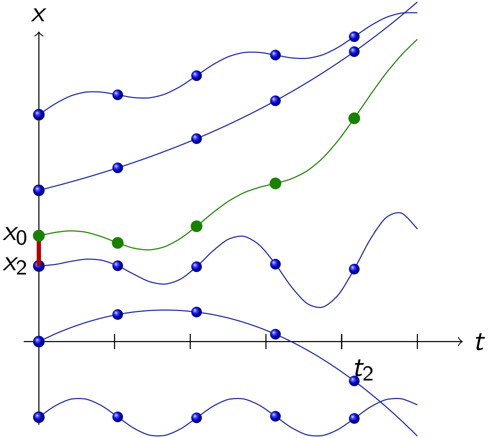
Problem (Image Computation – ODE transition)

Flow $\varphi : [0, \infty) \times \mathbb{R}^n \to \mathbb{R}^n$ solving $x' = f(x)$ reaches state B?



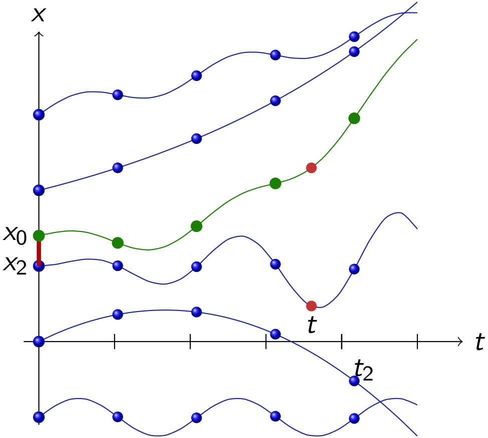
Problem (Image Computation – ODE transition)

Flow $\varphi : [0,\infty) \times \mathbb{R}^n \to \mathbb{R}^n$ solving $x' = f(x)$ reaches state B?



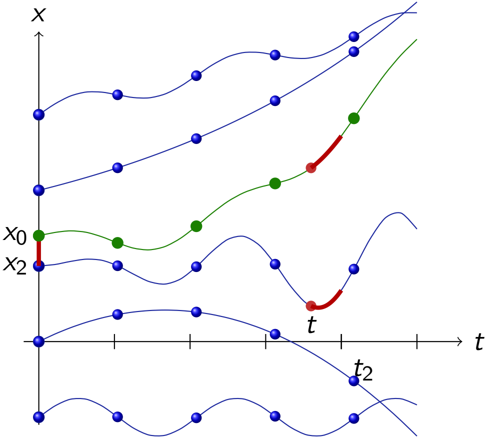
Problem (Image Computation – ODE transition)

Flow $\varphi : [0,\infty) \times \mathbb{R}^n \to \mathbb{R}^n$ solving $x' = f(x)$ reaches state B?



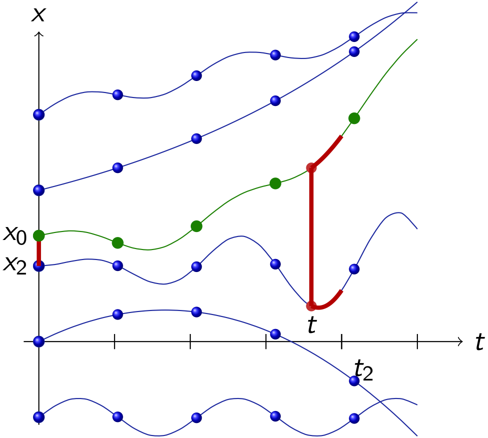
Problem (Image Computation – ODE transition)

Flow $\varphi : [0,\infty) \times \mathbb{R}^n \to \mathbb{R}^n$ solving $x' = f(x)$ reaches state B?



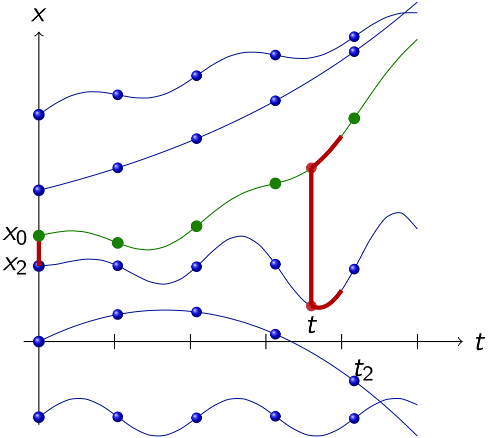
Problem (Image Computation – ODE transition)

Flow $\varphi : [0,\infty) \times \mathbb{R}^n \to \mathbb{R}^n$ solving $x' = f(x)$ reaches state B?



Problem (Image Computation – ODE transition)

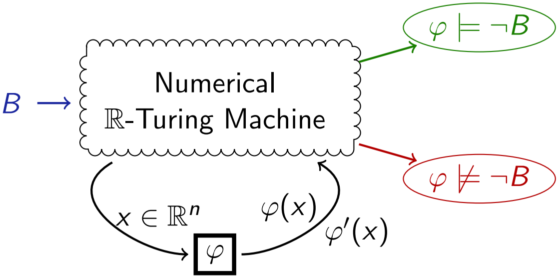
Flow $\varphi : [0,\infty) \times \mathbb{R}^n \to \mathbb{R}^n$ solving $x' = f(x)$ reaches state B?

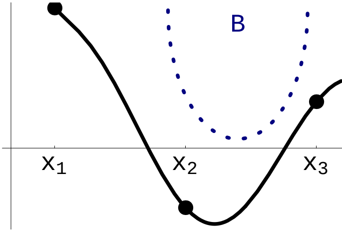


errors!

too many!

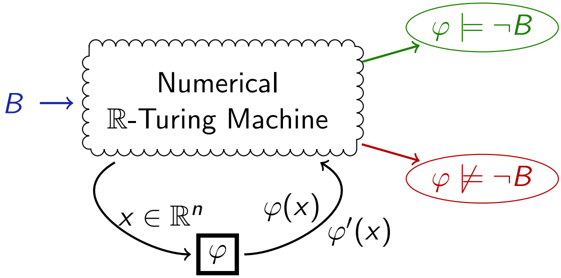
Numerical Image Computation

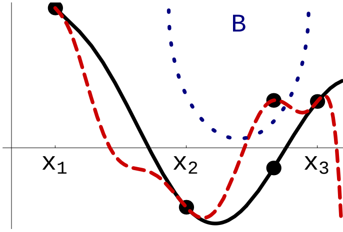




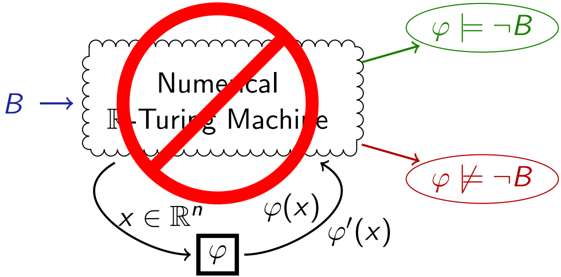
André Platzer and Edmund M. Clarke.
The image computation problem in hybrid systems model checking.
HSCC, vol. 4416 of *LNCS*, 473–486. Springer, 2007.

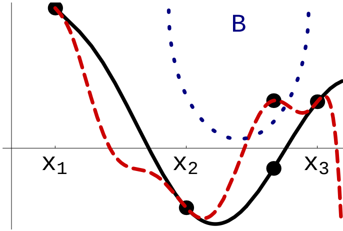
Numerical Image Computation





André Platzer and Edmund M. Clarke.
The image computation problem in hybrid systems model checking.
HSCC, vol. 4416 of *LNCS*, 473–486. Springer, 2007.





Proposition (Image computation undecidable numerically for. . .)

- *arbitrarily effective flow* $\varphi \in C^k(D \subseteq \mathbb{R}^n, \mathbb{R}^m)$*; D, B effective*
- *tolerate error* $\epsilon > 0$ *in decisions*

André Platzer and Edmund M. Clarke.
The image computation problem in hybrid systems model checking.
HSCC, vol. 4416 of *LNCS*, 473–486. Springer, 2007.