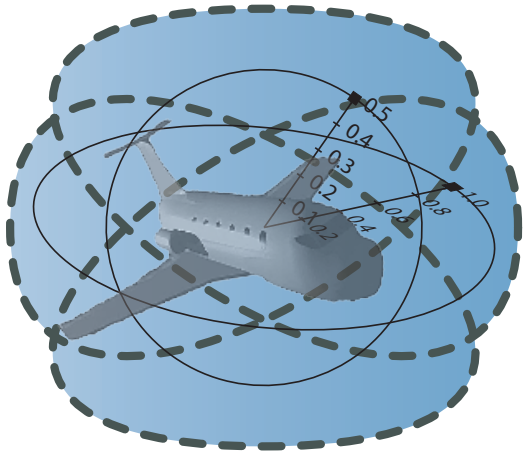
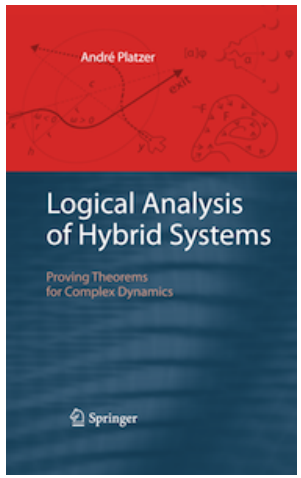
Logic and Compositional Verification of Hybrid Systems

André Platzer

aplatzer@cs.cmu.edu
Logical Systems Lab
Carnegie Mellon University, Pittsburgh, PA

http://symbolaris.com/





Outline

How can we build computerized controllers for physical systems that are guaranteed to meet their design goals?

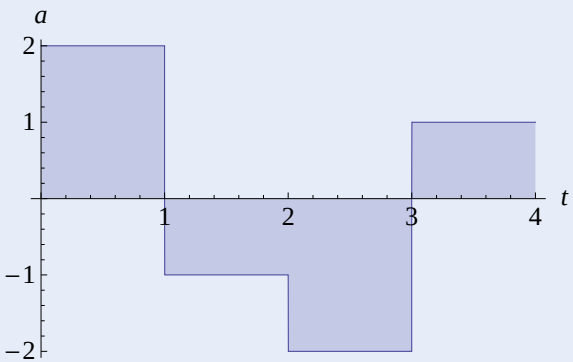
Hybrid Systems Analysis: Train Control

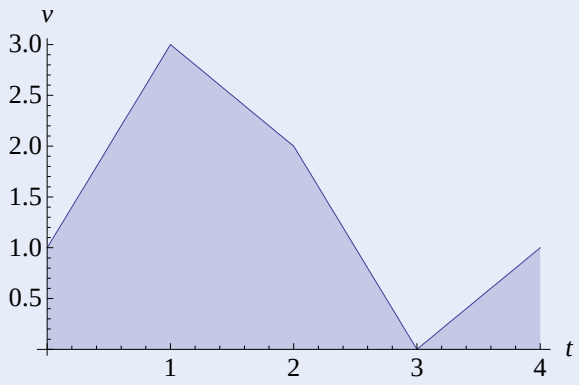
"Time is defined so that motion looks simple" [Henri Poincaré]

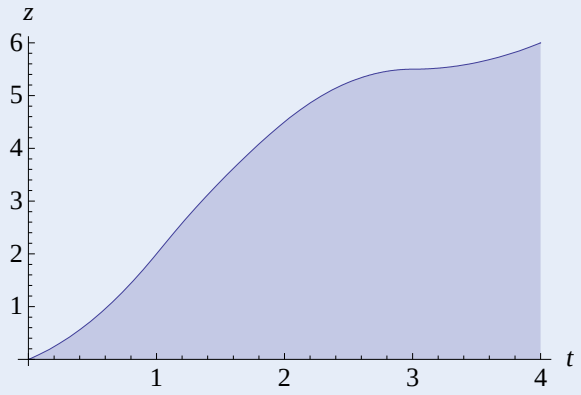
Challenge

Hybrid Systems

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







Hybrid Systems Analysis: Train Control

"Time is defined so that motion looks simple" [Henri Poincaré]

Challenge

Hybrid Systems

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

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

Hybrid Systems Analysis: Train Control

"Time is defined so that motion looks simple" [Henri Poincaré]

Challenge

Hybrid Systems

- 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: Train Control

"Time is defined so that motion looks simple" [Henri Poincaré]

Challenge

Hybrid Systems

- 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: Train Control

"Time is defined so that motion looks simple" [Henri Poincaré]

Challenge

Hybrid Systems

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

Hybrid Systems Analysis: Train Control

"Time is defined so that motion looks simple" [Henri Poincaré]

Challenge

Hybrid Systems

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

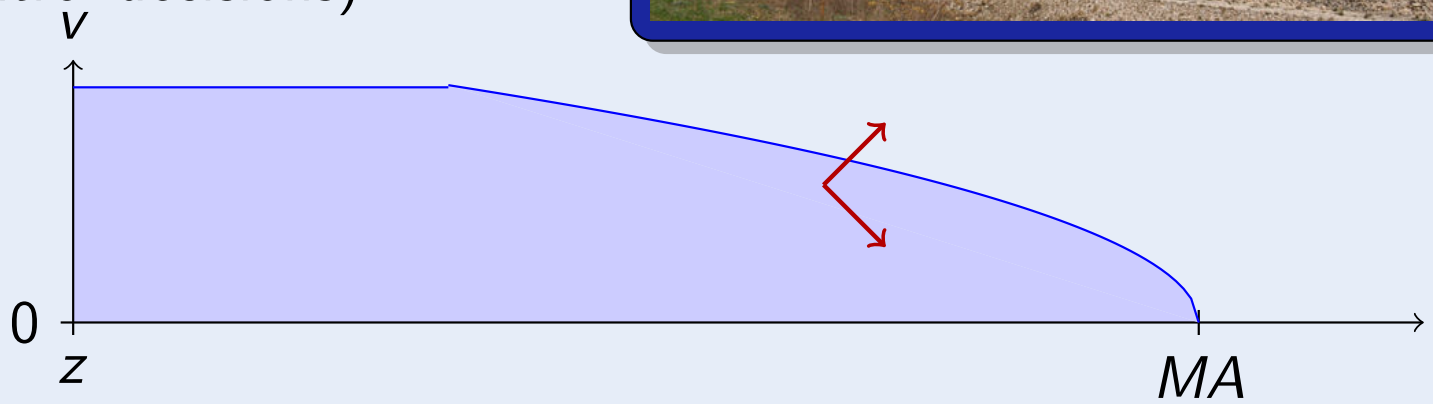
Hybrid Systems Analysis: Train Control

"Time is defined so that motion looks simple" [Henri Poincaré]

Challenge

Hybrid Systems

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



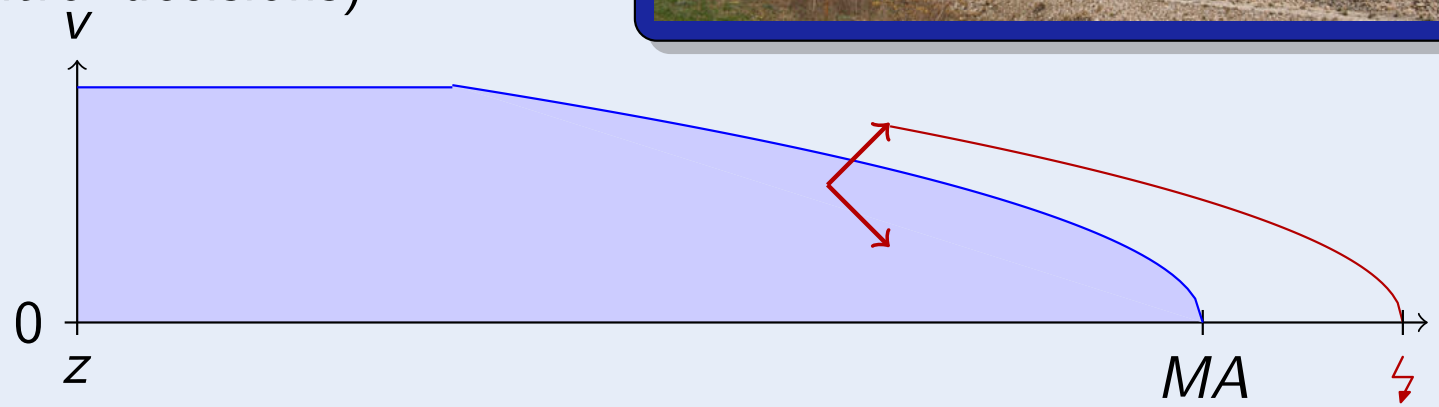
Hybrid Systems Analysis: Train Control

"Time is defined so that motion looks simple" [Henri Poincaré]

Challenge

Hybrid Systems

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



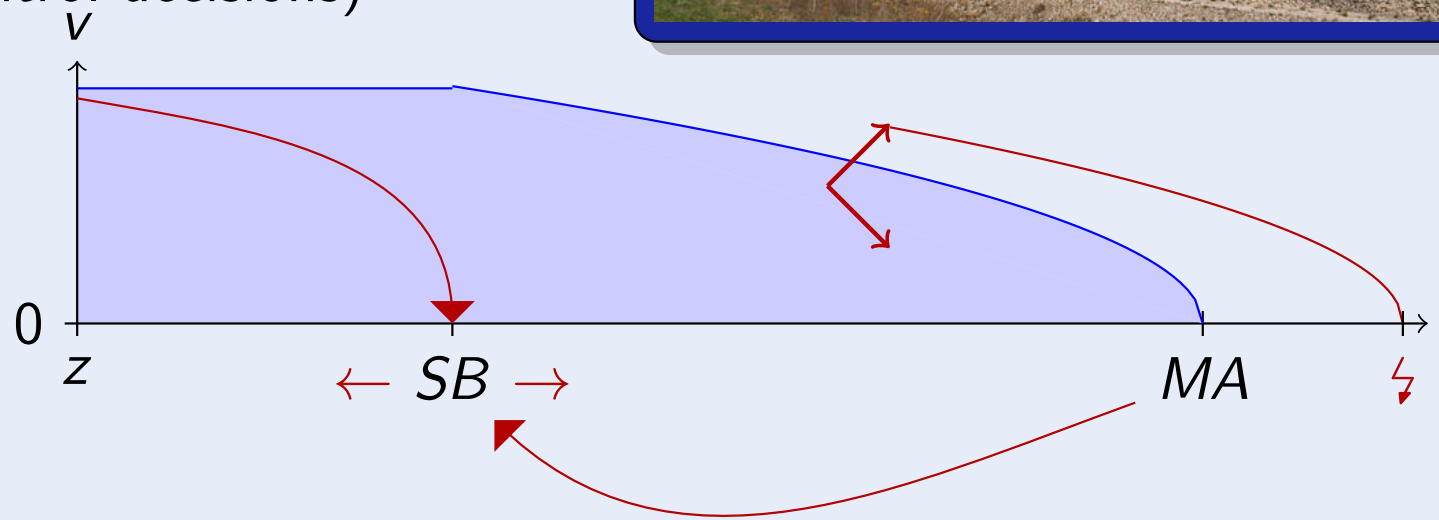
Hybrid Systems Analysis: Train Control

"Time is defined so that motion looks simple" [Henri Poincaré]

Challenge

Hybrid Systems

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



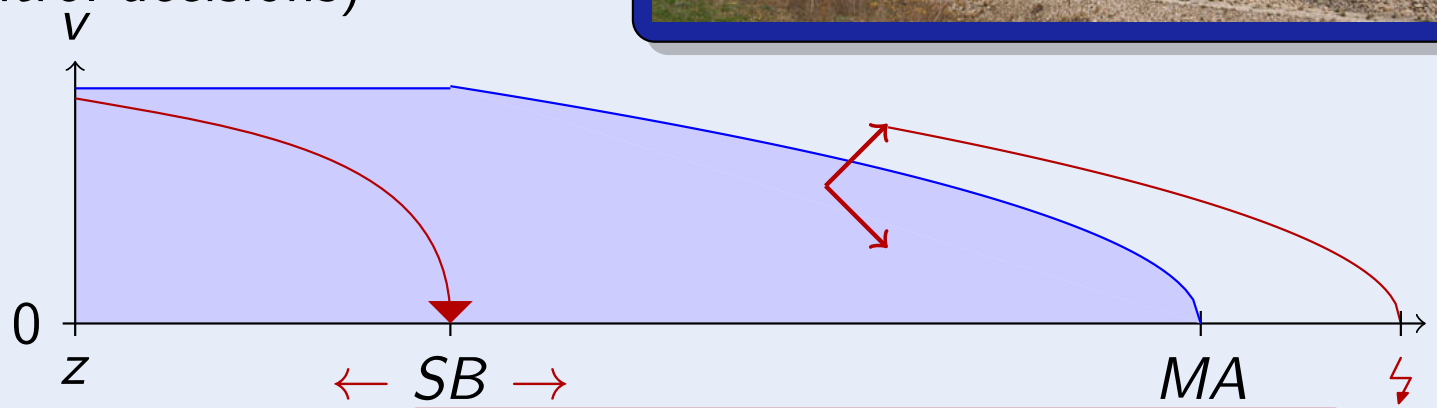
Hybrid Systems Analysis: Train Control

"Time is defined so that motion looks simple" [Henri Poincaré]

Challenge

Hybrid Systems

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



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

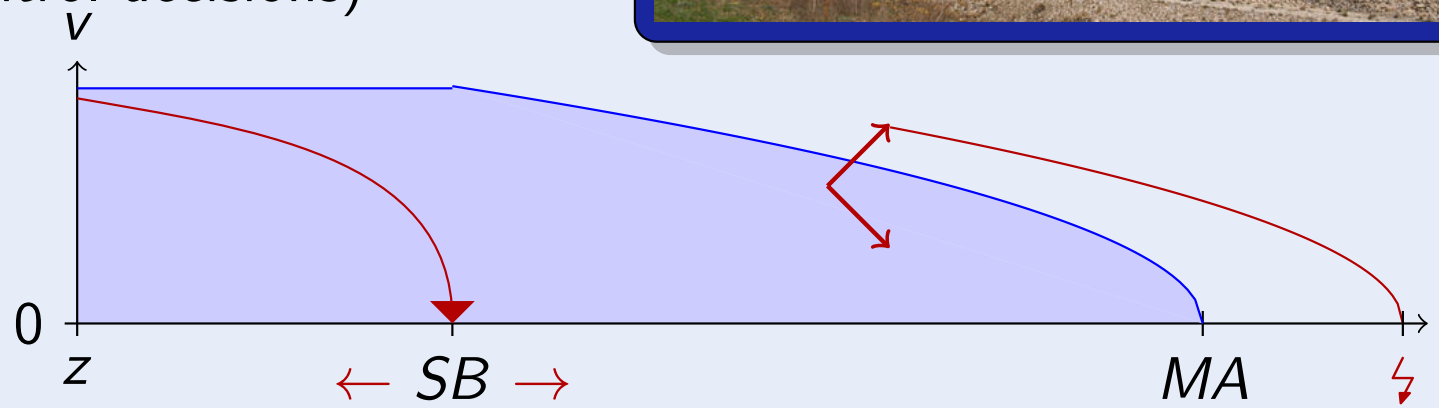
Hybrid Systems Analysis: Train Control

"Time is defined so that motion looks simple" [Henri Poincaré]

Challenge

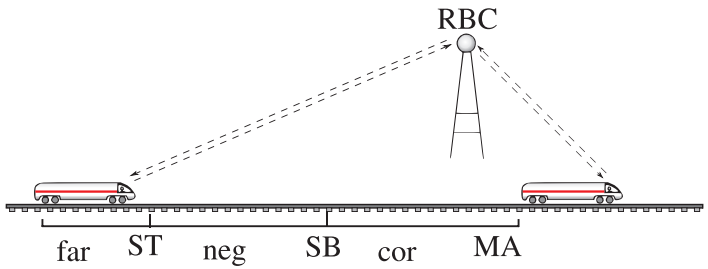
Hybrid Systems

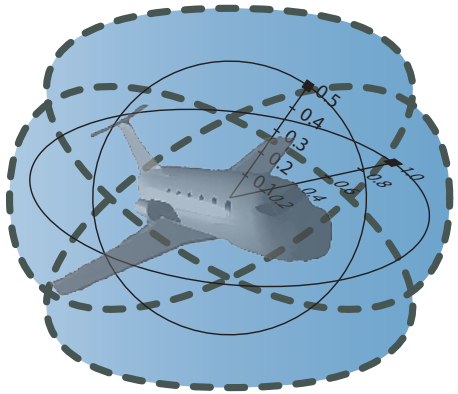
- Continuous dynamics (differential equations)
- Discrete dynamics (control decisions)

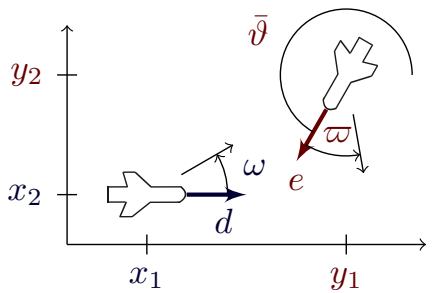


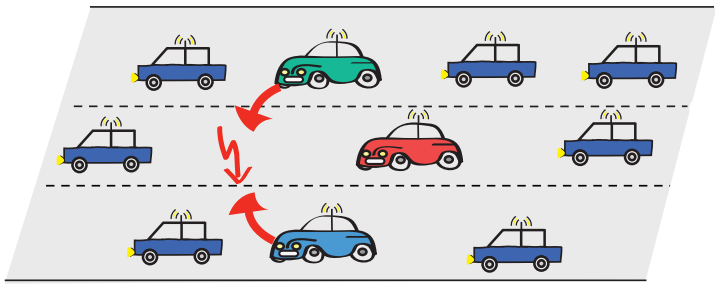
$\forall MA\, \exists SB$ "train always safe"

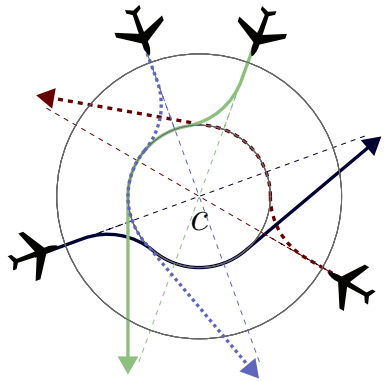
Hybrid Systems Analysis is Important for . . .

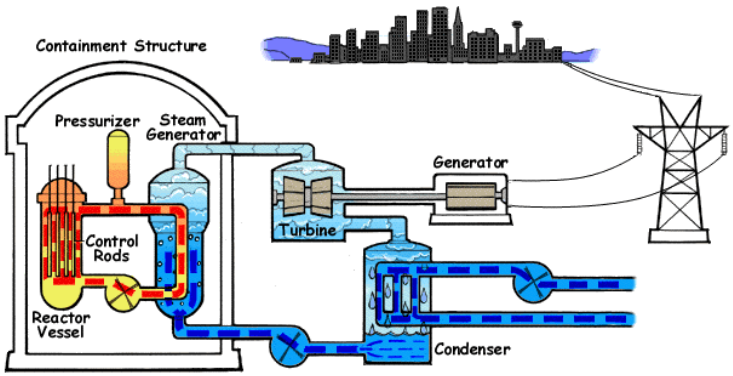












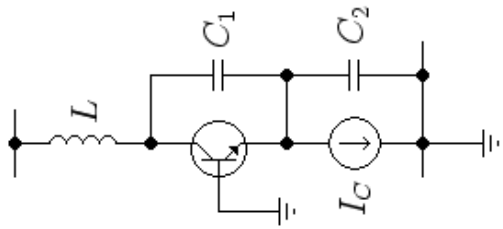


Image Computation

Problem (Image Computation – generic)

Do transitions of system H reach bad state in B from an initial state in I?

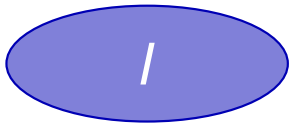


Image Computation

Problem (Image Computation – generic)

Do transitions of system H reach bad state in B from an initial state in I?

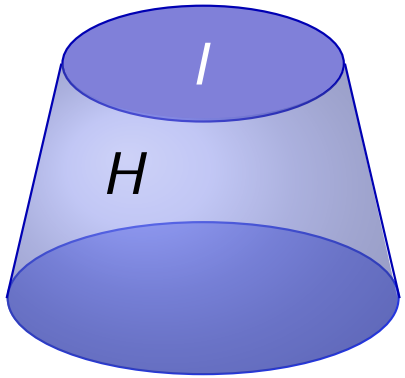


Image Computation

Problem (Image Computation – generic)

Do transitions of system H reach bad state in B from an initial state in I?

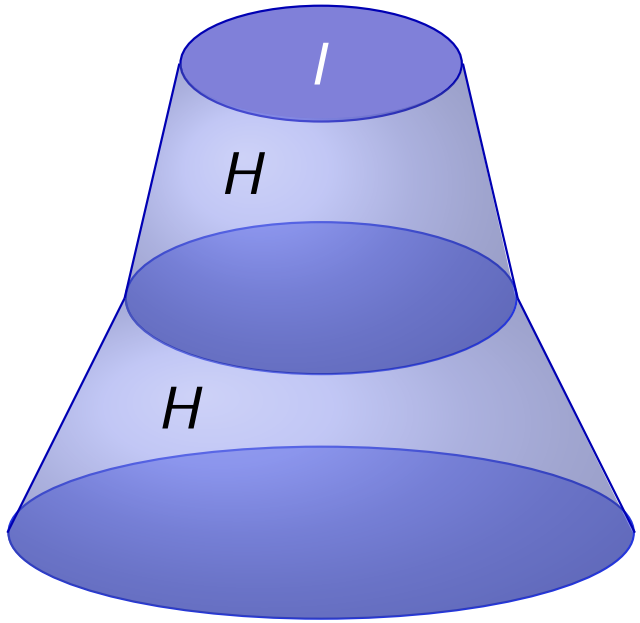


Image Computation

Problem (Image Computation – generic)

Do transitions of system H reach bad state in B from an initial state in I?

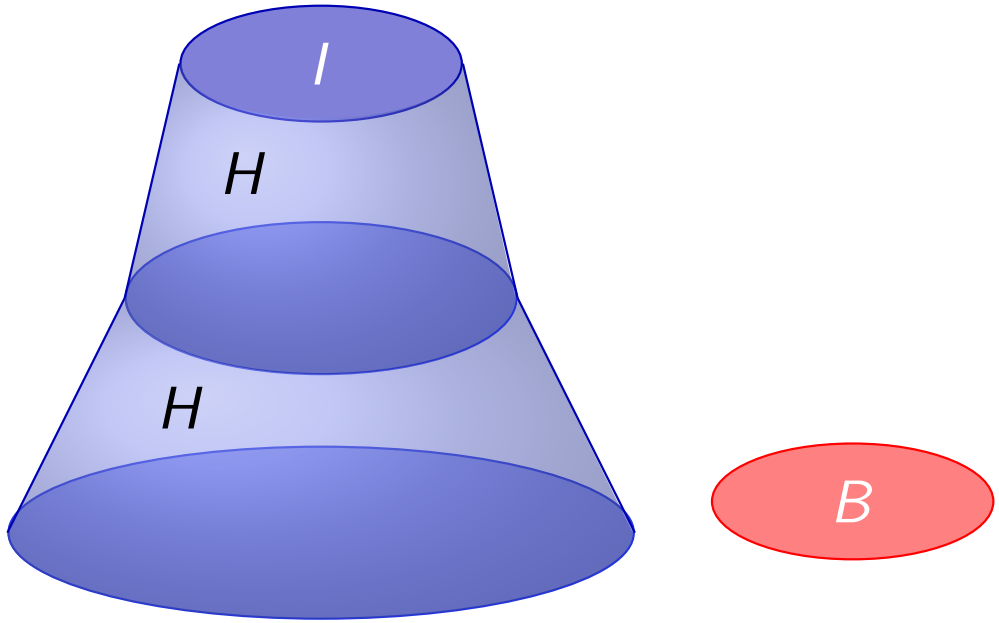


Image Computation

Problem (Image Computation – generic)

Do transitions of system H reach bad state in B from an initial state in I?

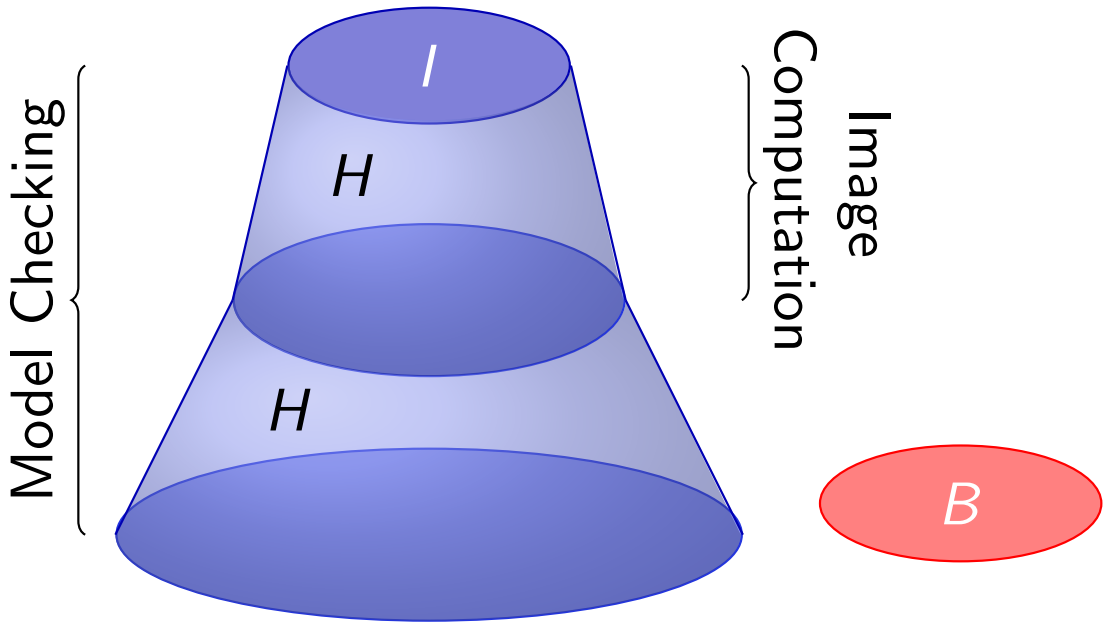


Image Computation: Continuous Systems

Problem (Image Computation – continuous transition)

Flow $\varphi : [0, \infty) \times \mathbb{R}^n \to \mathbb{R}^n$ reaches state B, i.e., $\exists t, x_0 : \ \varphi(t, x_0) \in B$?

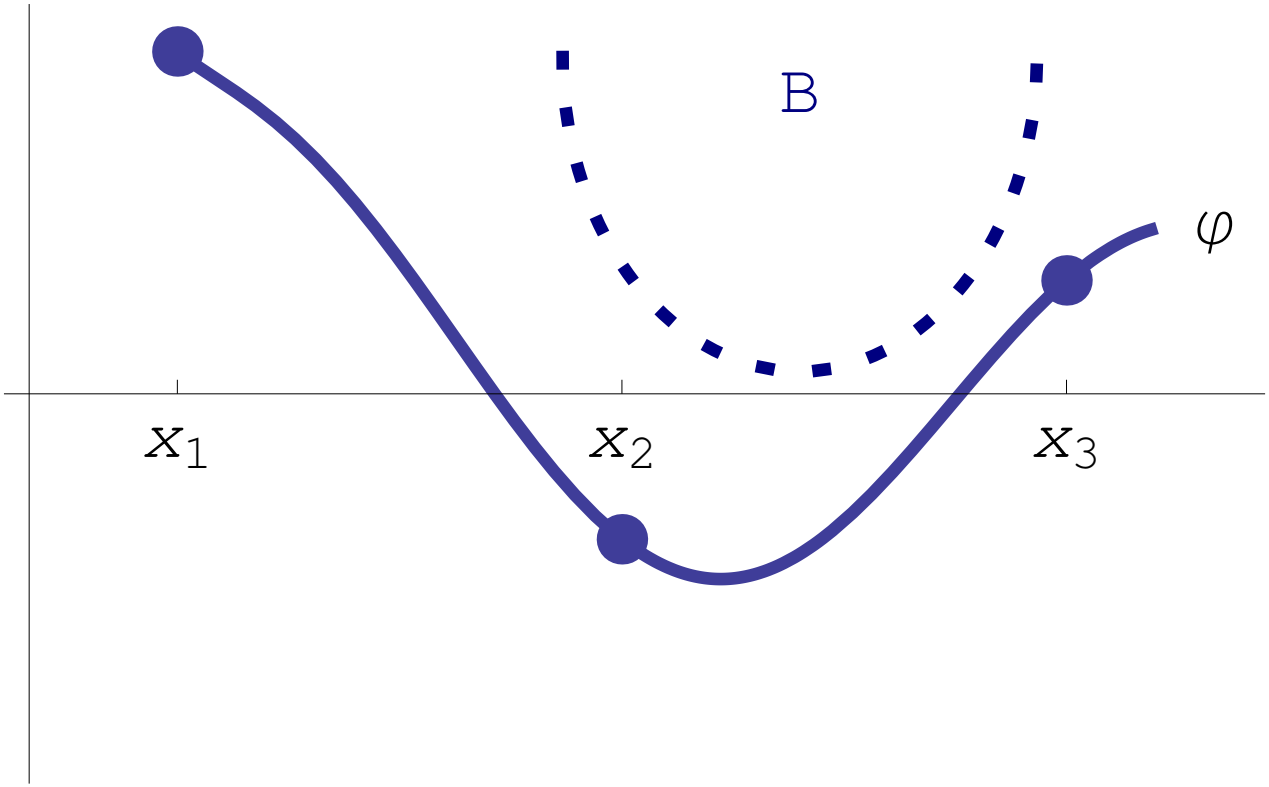
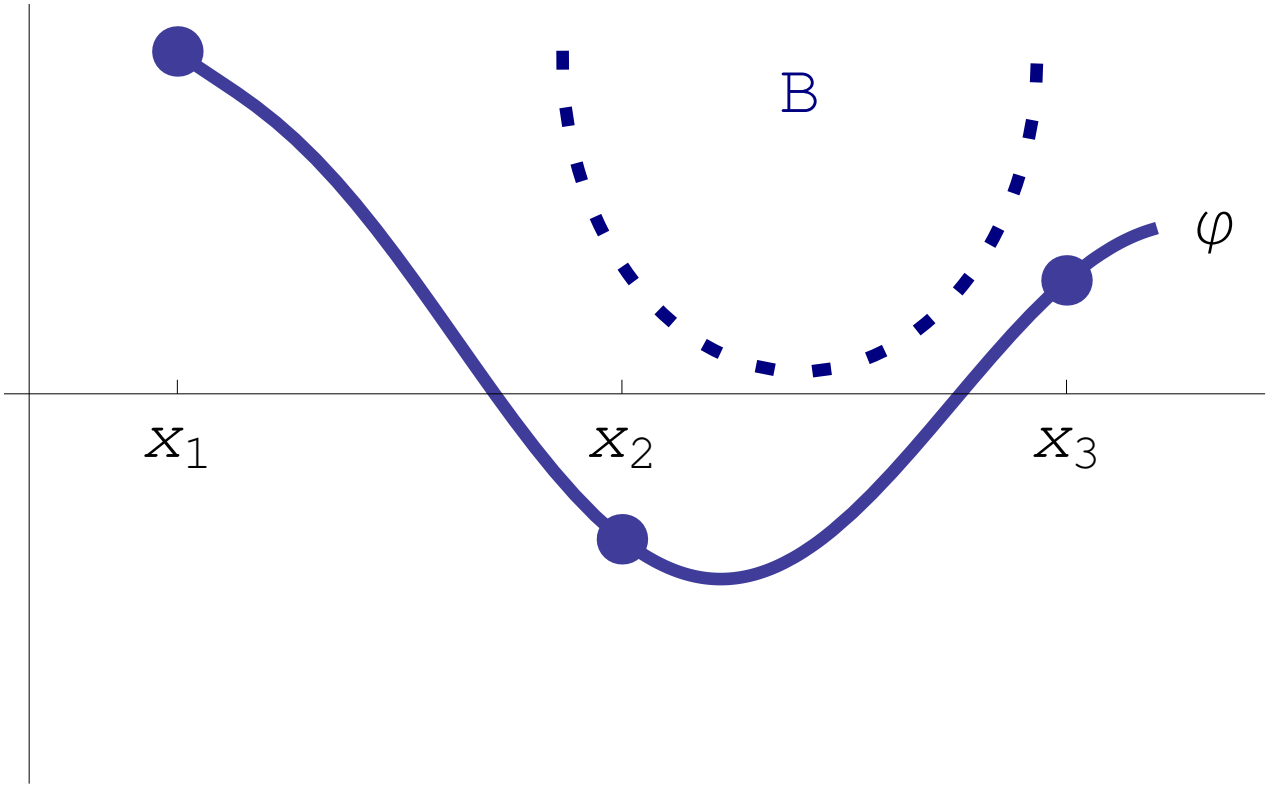


Image Computation: Continuous Systems

Problem (Image Computation – continuous transition)

Flow $\varphi : [0, \infty) \times \mathbb{R}^n \to \mathbb{R}^n$ reaches state B, i.e., $\exists t, x_0 : \ \varphi(t, x_0) \in B$?

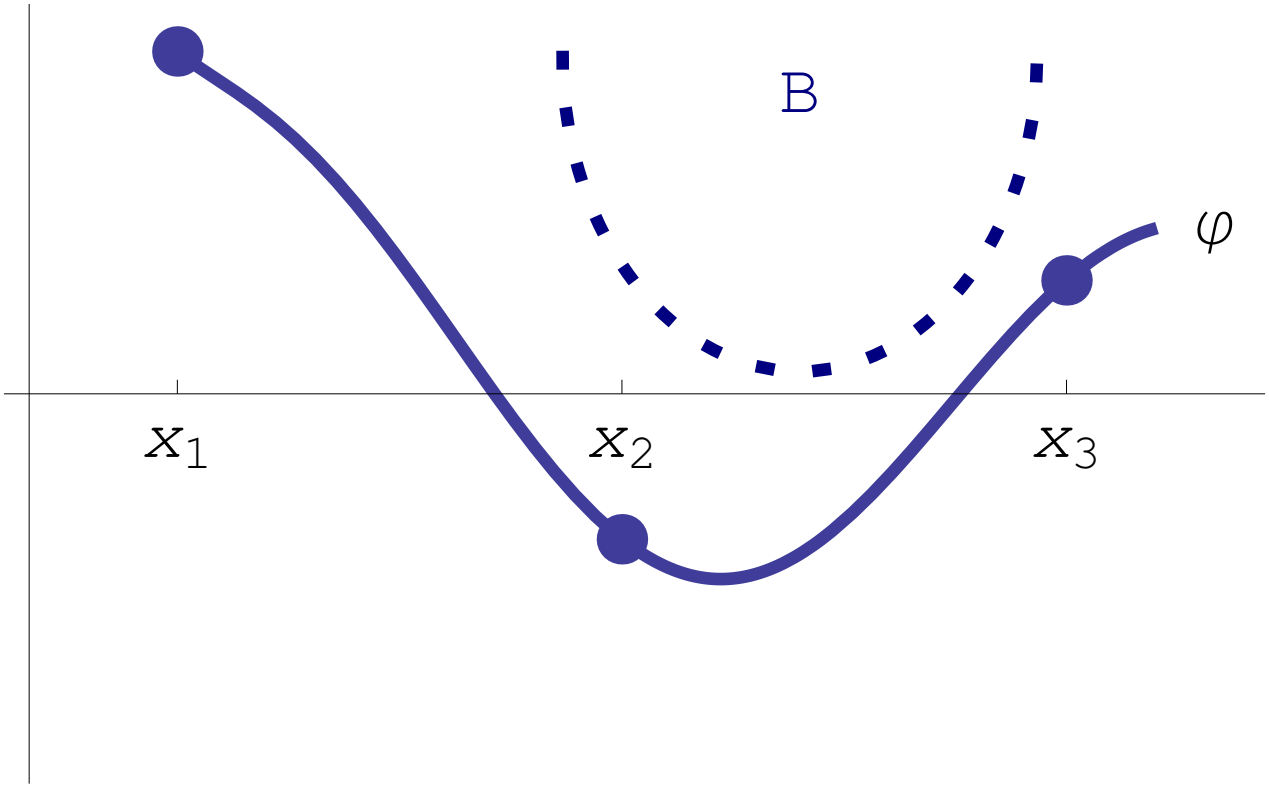


Idea: Sample points

Image Computation: Continuous Systems

Problem (Image Computation – continuous transition)

Flow $\varphi : [0, \infty) \times \mathbb{R}^n \to \mathbb{R}^n$ reaches state B, i.e., $\exists t, x_0 : \ \varphi(t, x_0) \in B$?



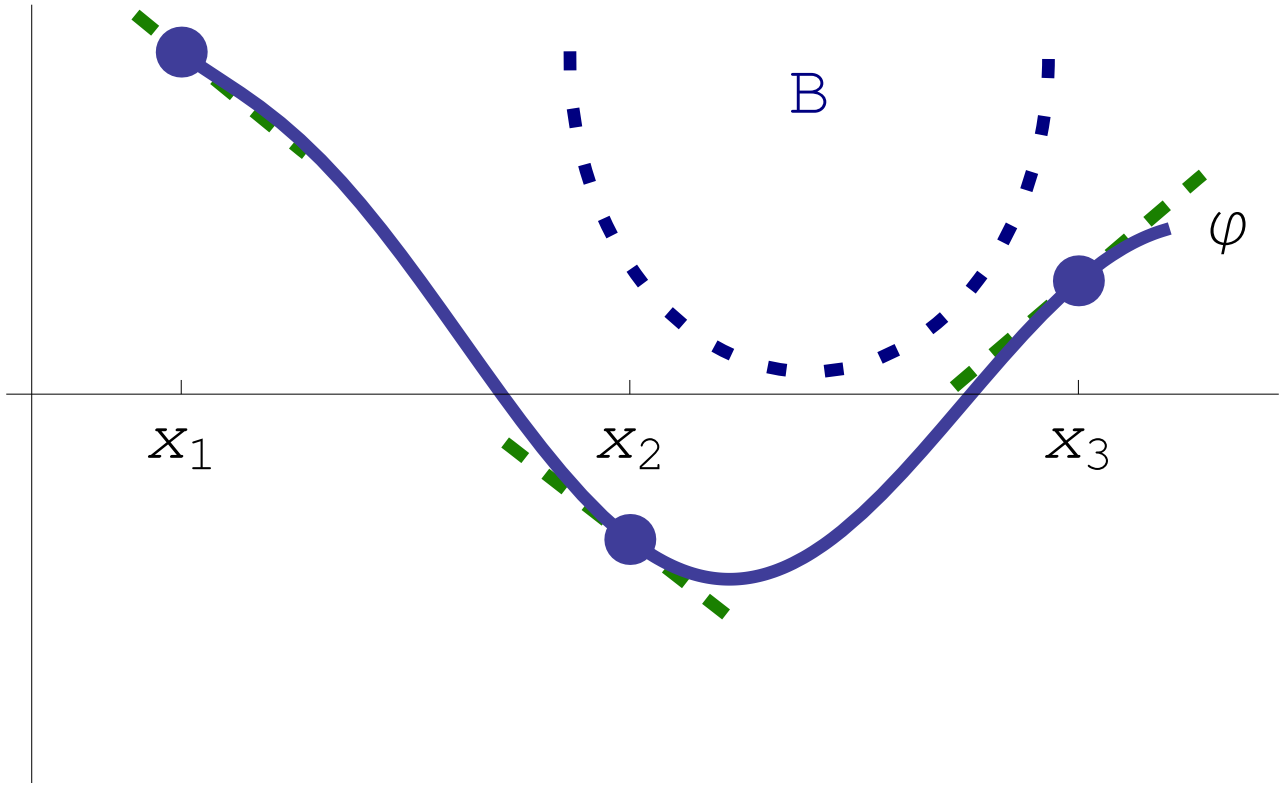
Idea: Sample points

too many!

Image Computation: Continuous Systems

Problem (Image Computation – continuous transition)

Flow $\varphi : [0, \infty) \times \mathbb{R}^n \to \mathbb{R}^n$ reaches state B, i.e., $\exists t, x_0 : \ \varphi(t, x_0) \in B$?

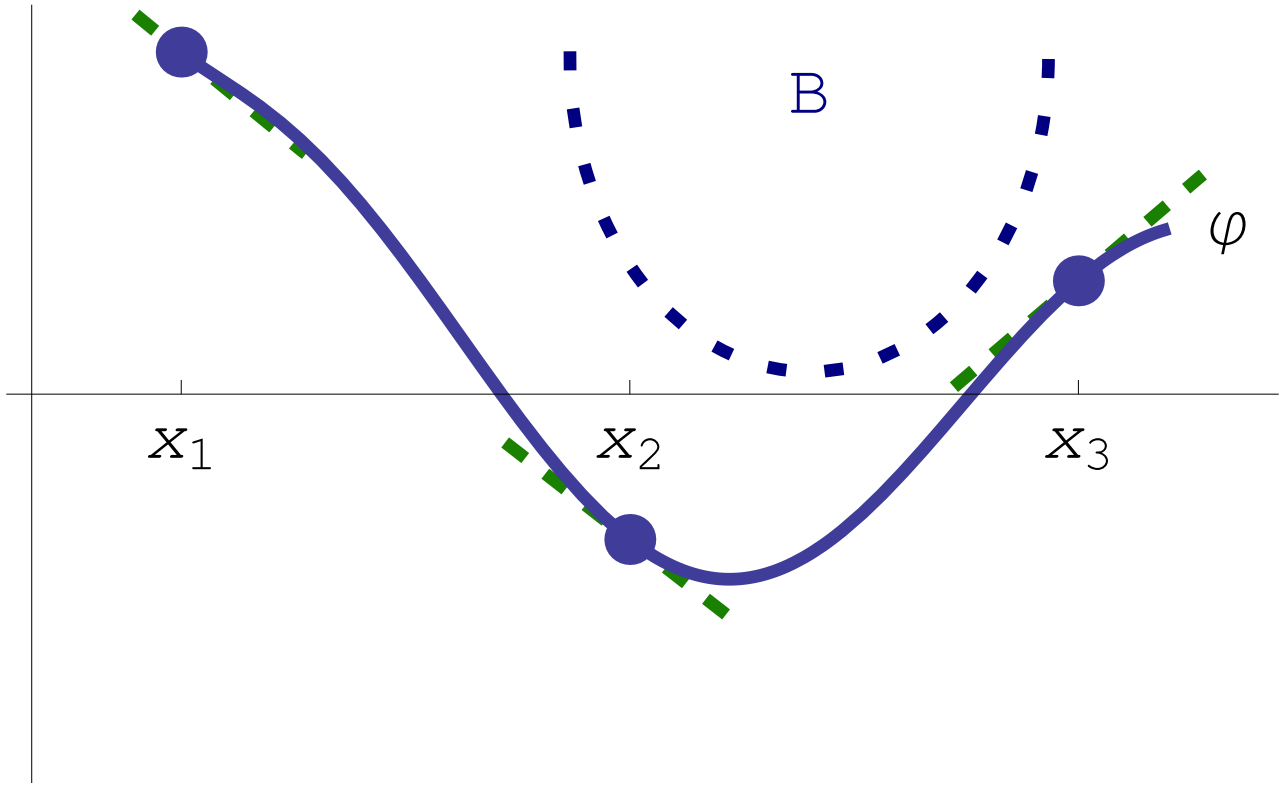


Idea: Sample points & derivatives

Image Computation: Continuous Systems

Problem (Image Computation – continuous transition)

Flow $\varphi : [0, \infty) \times \mathbb{R}^n \to \mathbb{R}^n$ reaches state B, i.e., $\exists t, x_0 : \ \varphi(t, x_0) \in B$?

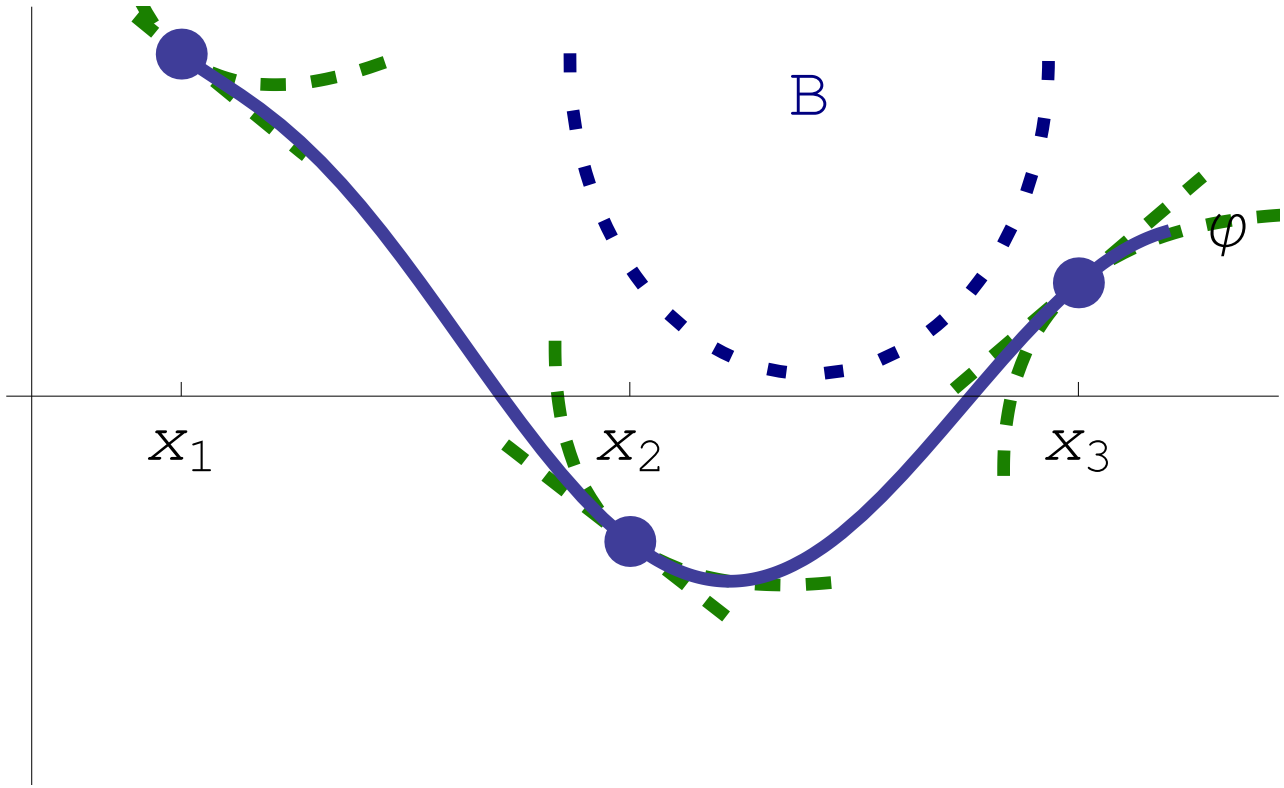


Idea: Sample points & derivatives

too many!

Problem (Image Computation – continuous transition)

Flow $\varphi : [0, \infty) \times \mathbb{R}^n \to \mathbb{R}^n$ reaches state B, i.e., $\exists t, x_0 : \ \varphi(t, x_0) \in B$?

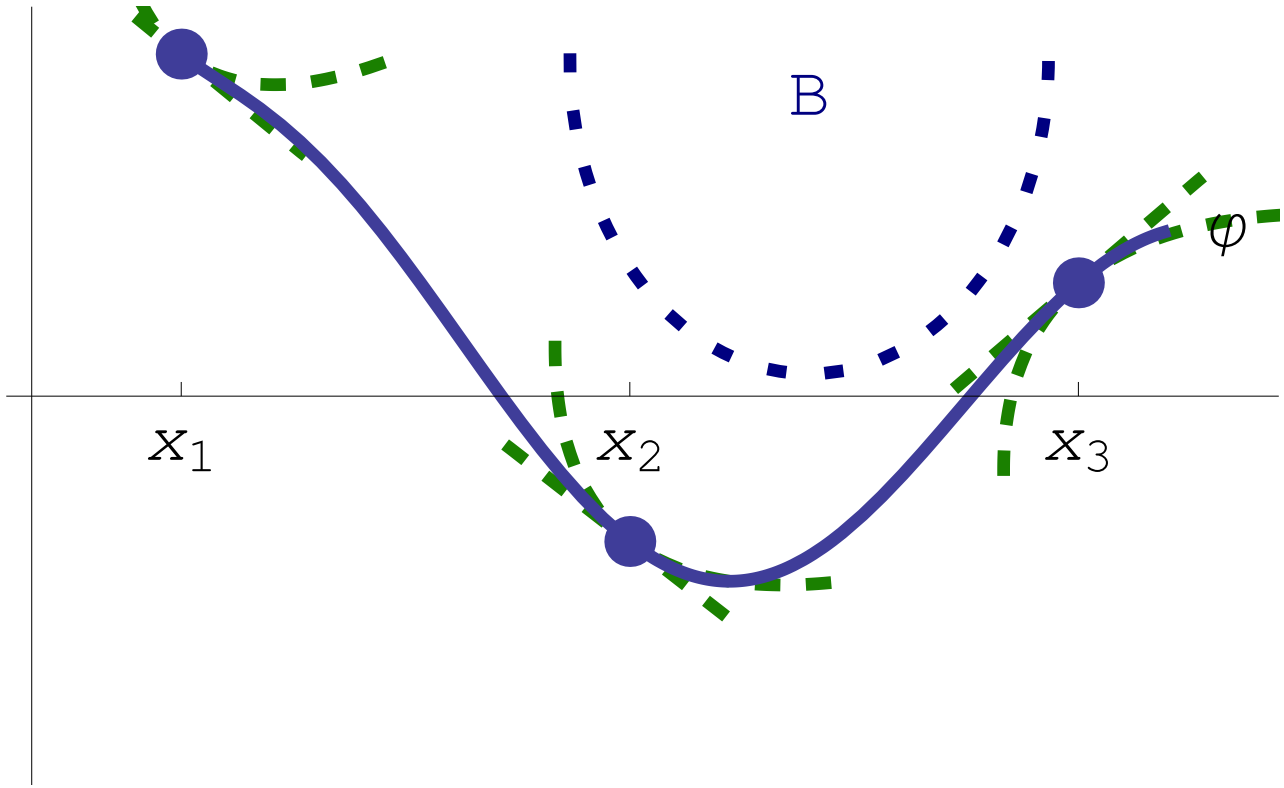


Idea: Sample points & derivatives 1&2

Image Computation: Continuous Systems

Problem (Image Computation – continuous transition)

Flow $\varphi : [0, \infty) \times \mathbb{R}^n \to \mathbb{R}^n$ reaches state B, i.e., $\exists t, x_0 : \ \varphi(t, x_0) \in B$?



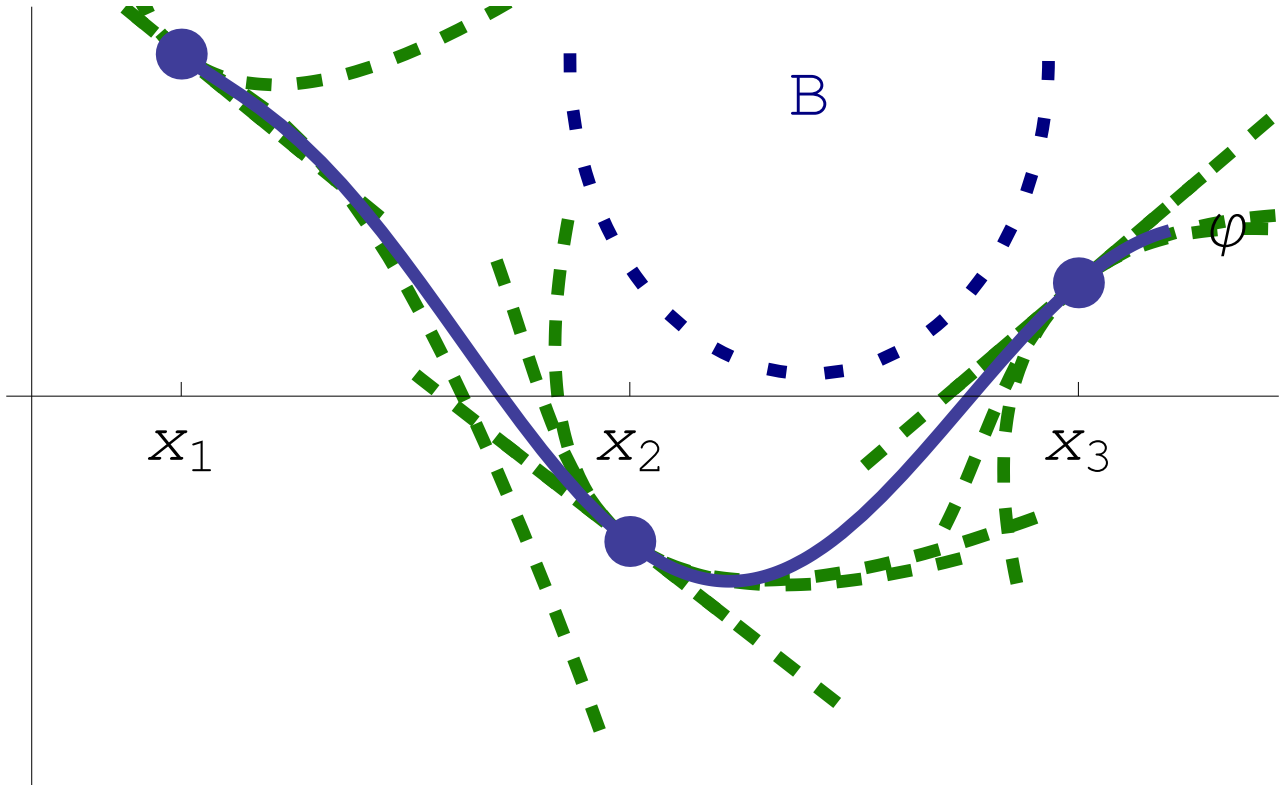
Idea: Sample points & derivatives 1&2

too many!

Image Computation: Continuous Systems

Problem (Image Computation – continuous transition)

Flow $\varphi : [0, \infty) \times \mathbb{R}^n \to \mathbb{R}^n$ reaches state B, i.e., $\exists t, x_0 : \ \varphi(t, x_0) \in B$?

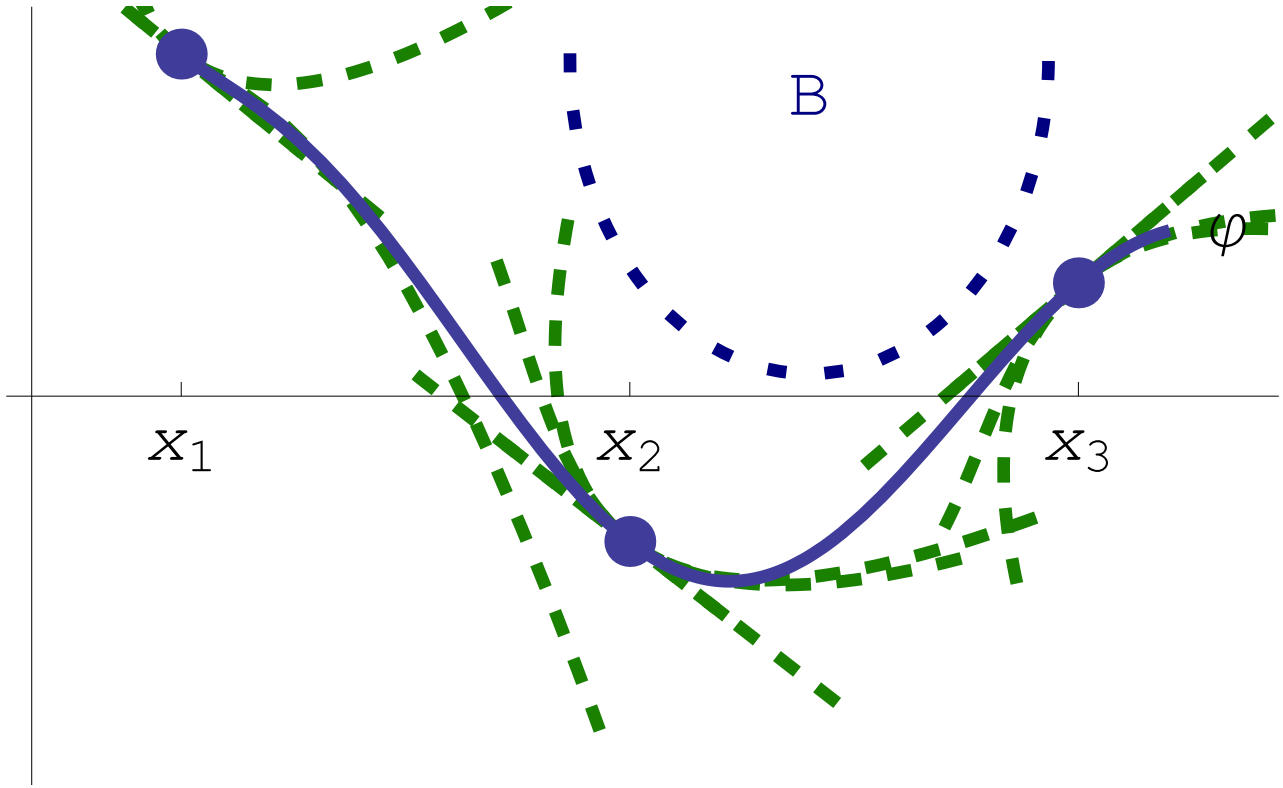


Idea: Sample points & derivatives 1&2&3

Image Computation: Continuous Systems

Problem (Image Computation – continuous transition)

Flow $\varphi : [0, \infty) \times \mathbb{R}^n \to \mathbb{R}^n$ reaches state B, i.e., $\exists t, x_0 : \ \varphi(t, x_0) \in B$?

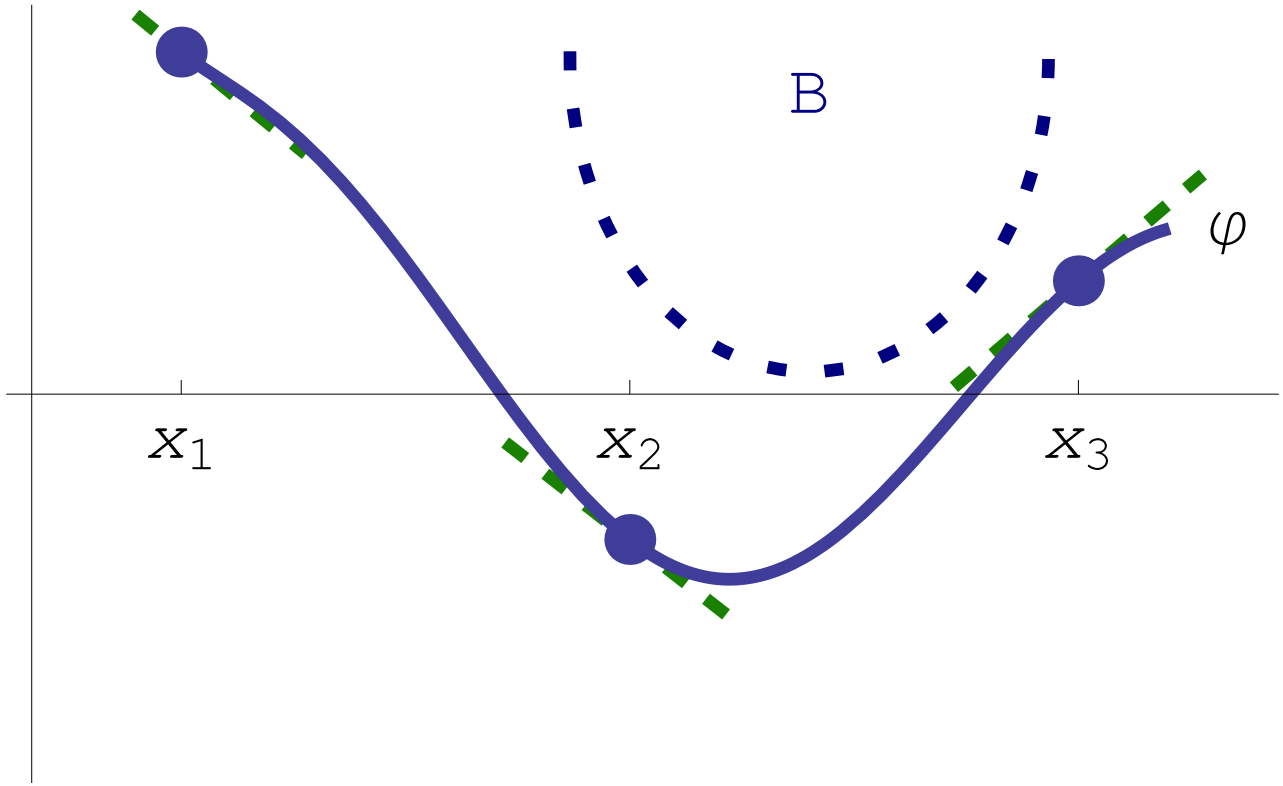


Idea: Sample points & derivatives 1&2&3 too many!

Image Computation: Continuous Systems

Problem (Image Computation – continuous transition)

Flow $\varphi : [0, \infty) \times \mathbb{R}^n \to \mathbb{R}^n$ reaches state B, i.e., $\exists t, x_0 : \ \varphi(t, x_0) \in B$?

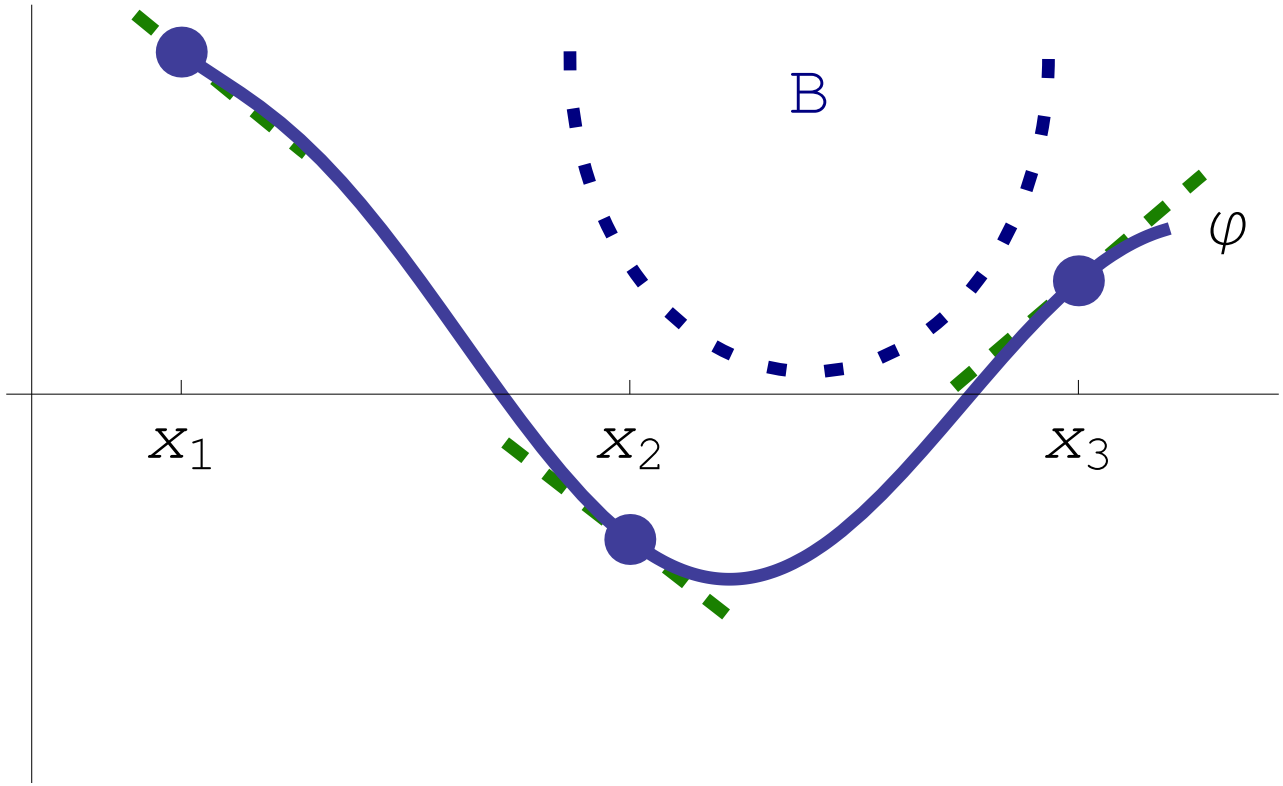


Idea: Sample points & X curve & blow up to regions & ...

Image Computation: Continuous Systems

Problem (Image Computation – continuous transition)

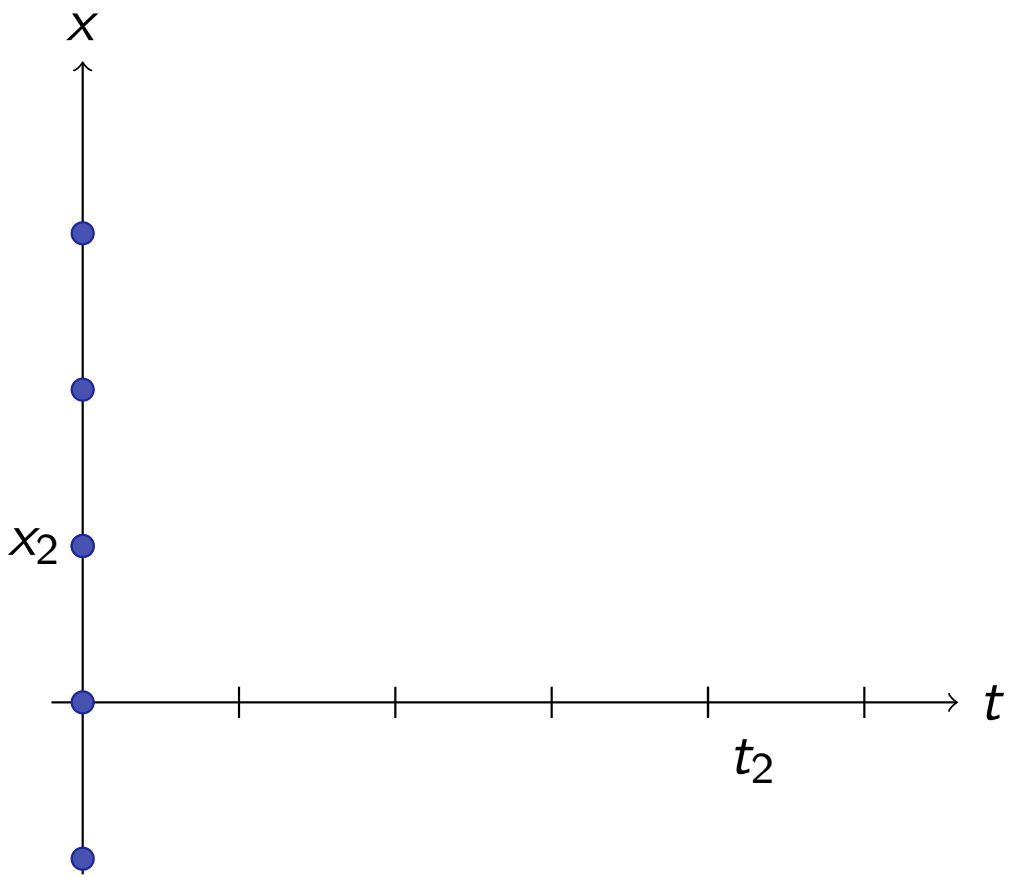
Flow $\varphi : [0, \infty) \times \mathbb{R}^n \to \mathbb{R}^n$ reaches state B, i.e., $\exists t, x_0 : \ \varphi(t, x_0) \in B$?



Idea: Sample points & X curve & blow up to regions & ... too many!

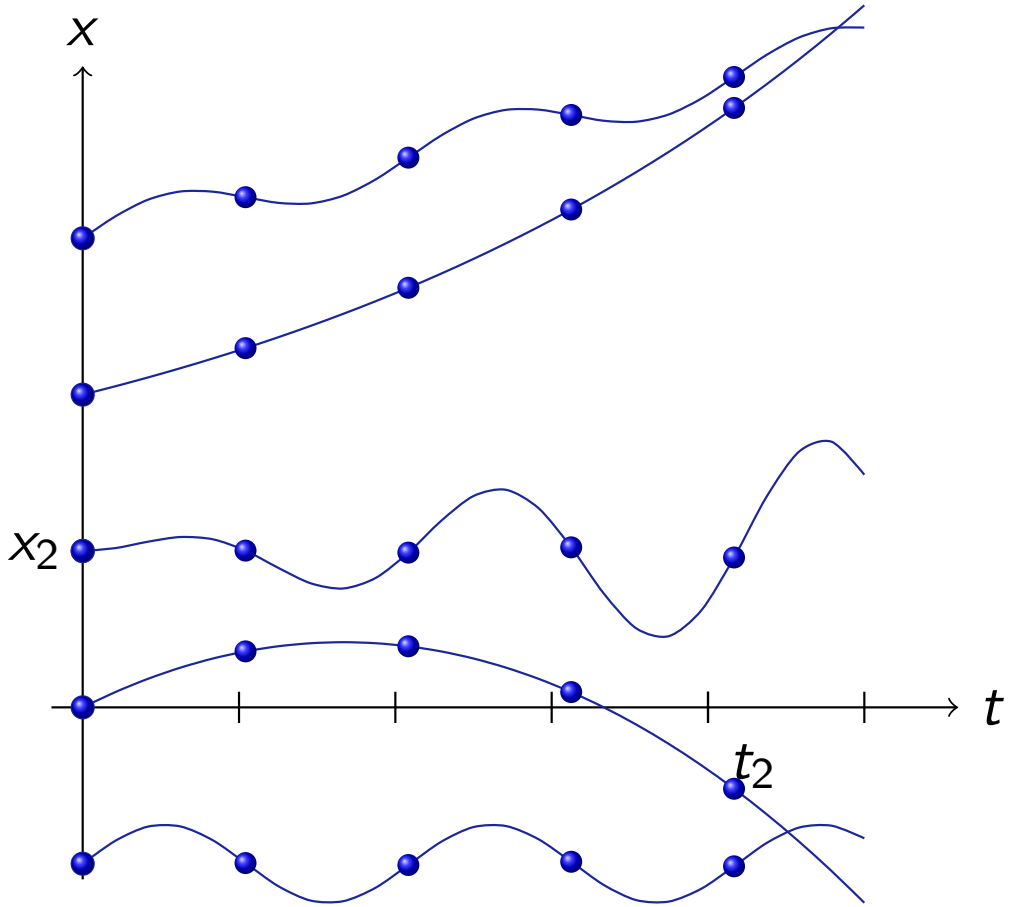
Problem (Image Computation – ODE transition)

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?

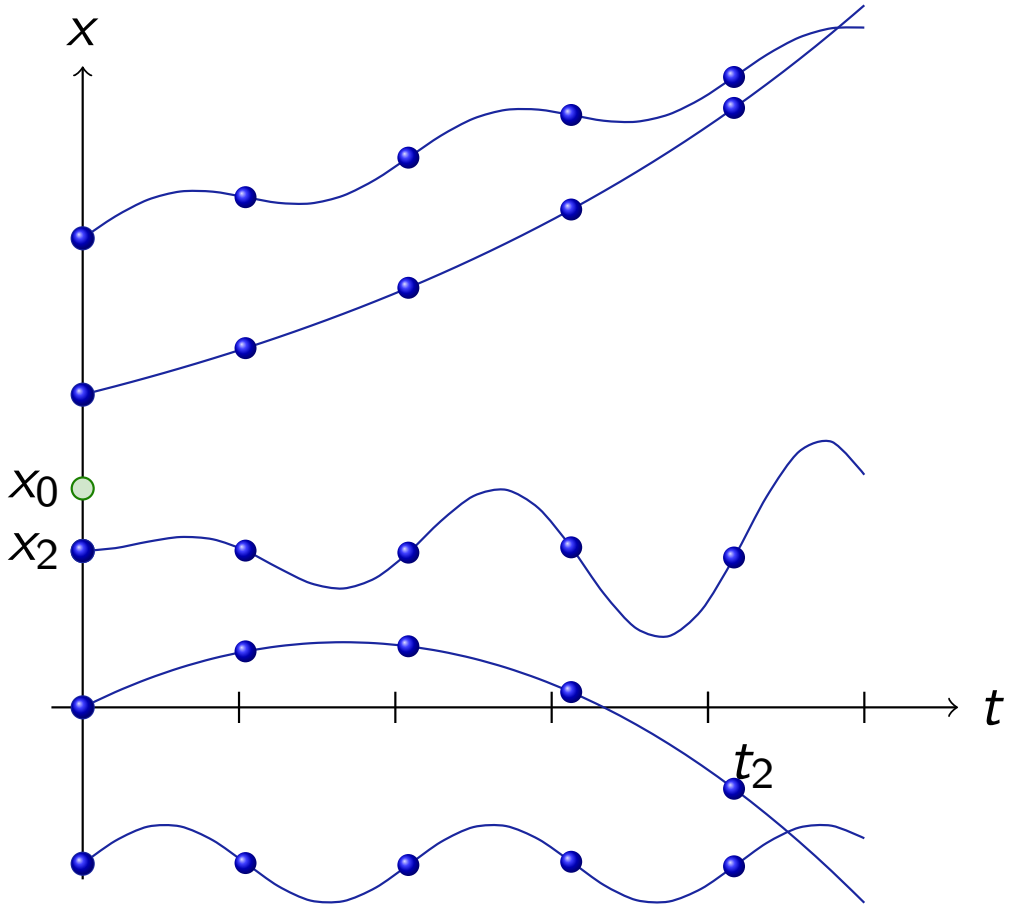
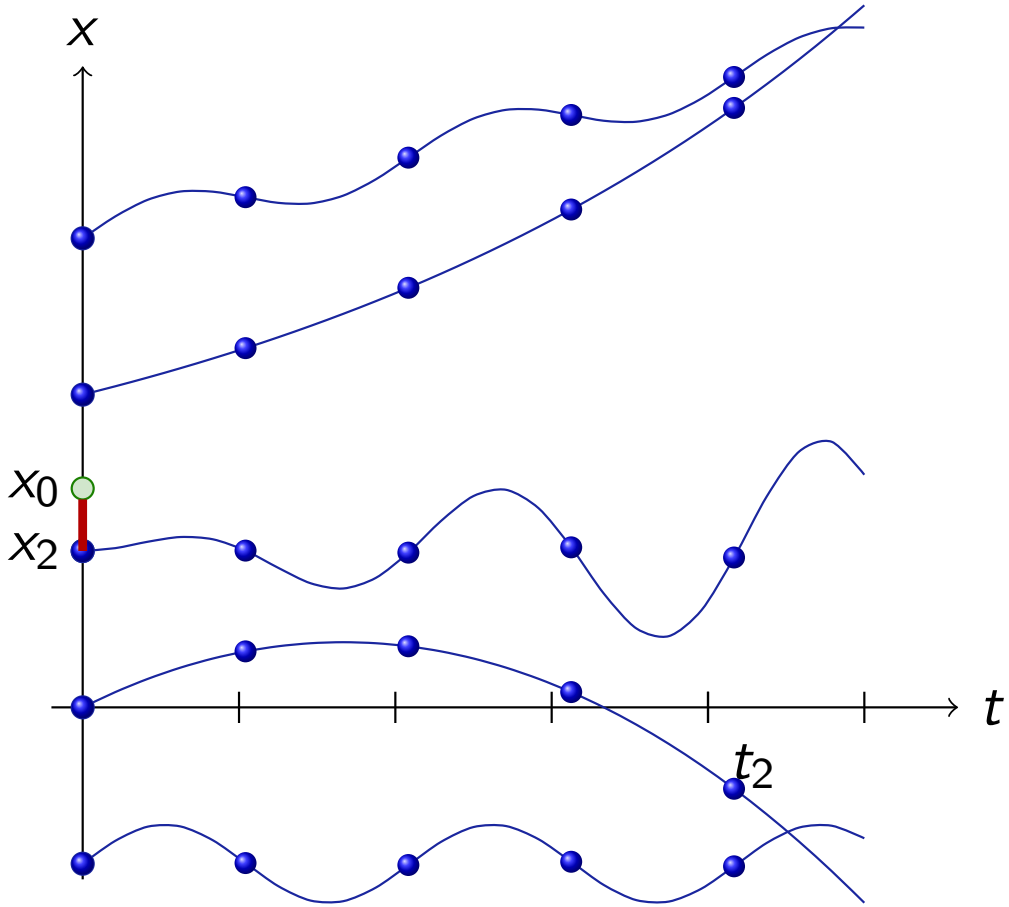


Image Computation: Continuous Systems

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?

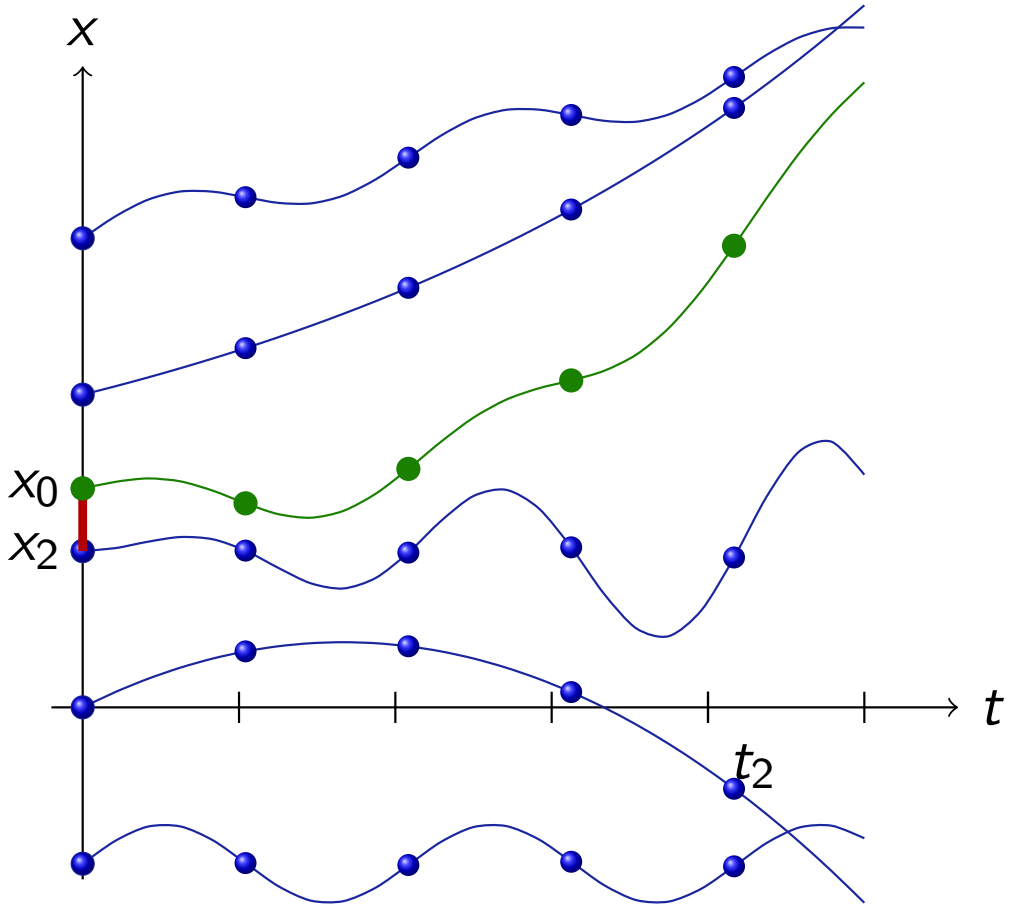


Image Computation: Continuous Systems

Problem (Image Computation – ODE transition)

Flow $\varphi : [0, \infty) \times \mathbb{R}^n \to \mathbb{R}^n$ solving $x' = f(x)$ reaches state B?

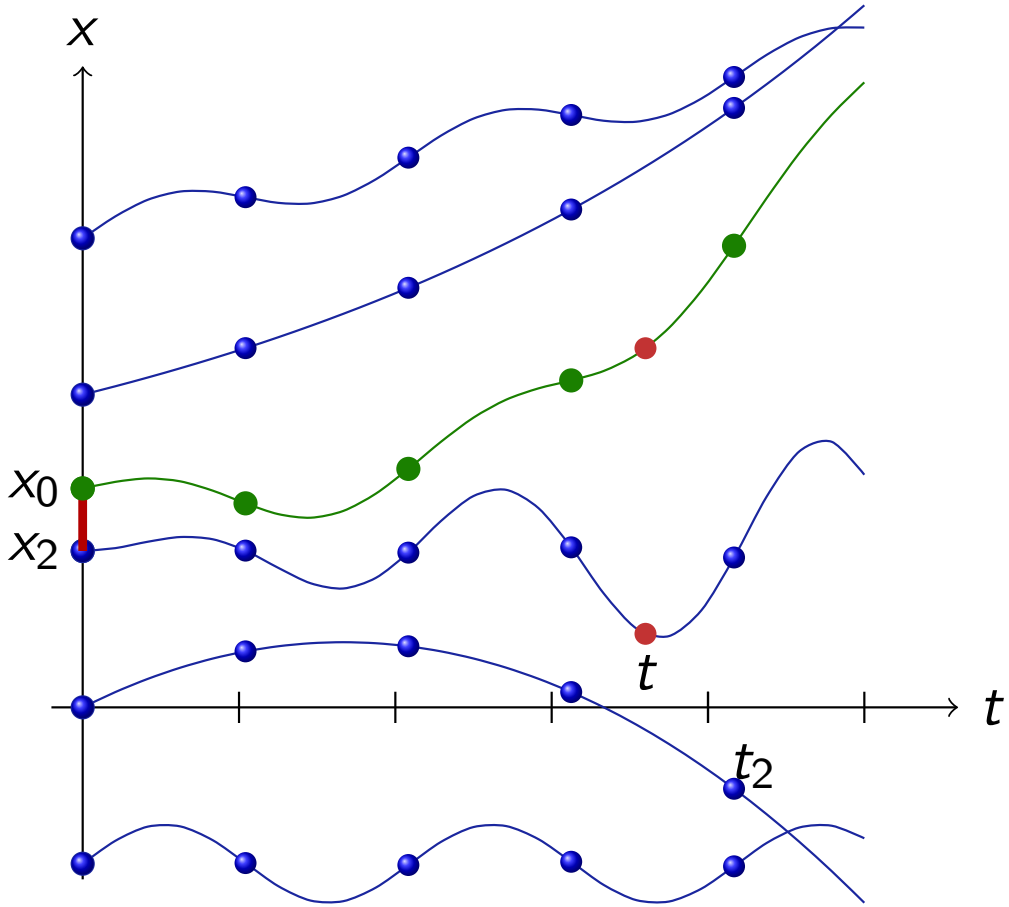


Image Computation: Continuous Systems

Problem (Image Computation – ODE transition)

Flow $\varphi : [0, \infty) \times \mathbb{R}^n \to \mathbb{R}^n$ solving $x' = f(x)$ reaches state B?

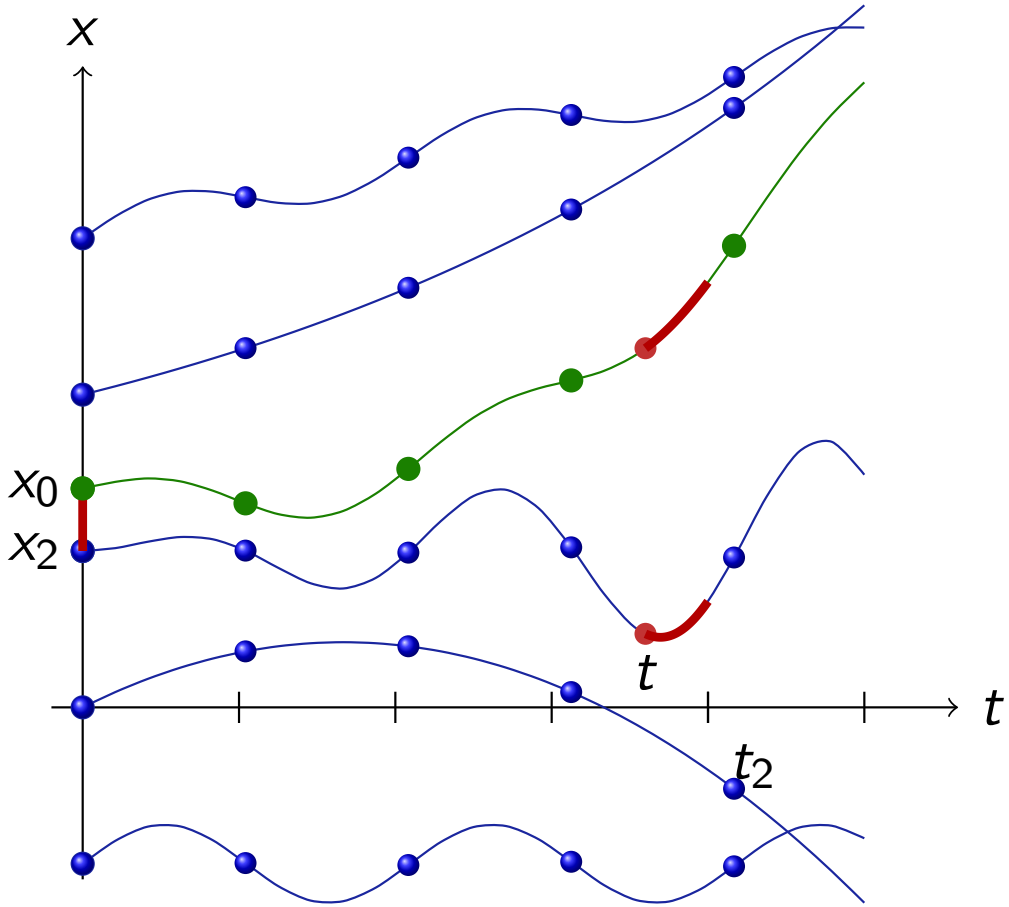


Image Computation: Continuous Systems

Problem (Image Computation – ODE transition)

Flow $\varphi : [0, \infty) \times \mathbb{R}^n \to \mathbb{R}^n$ solving $x' = f(x)$ reaches state B?

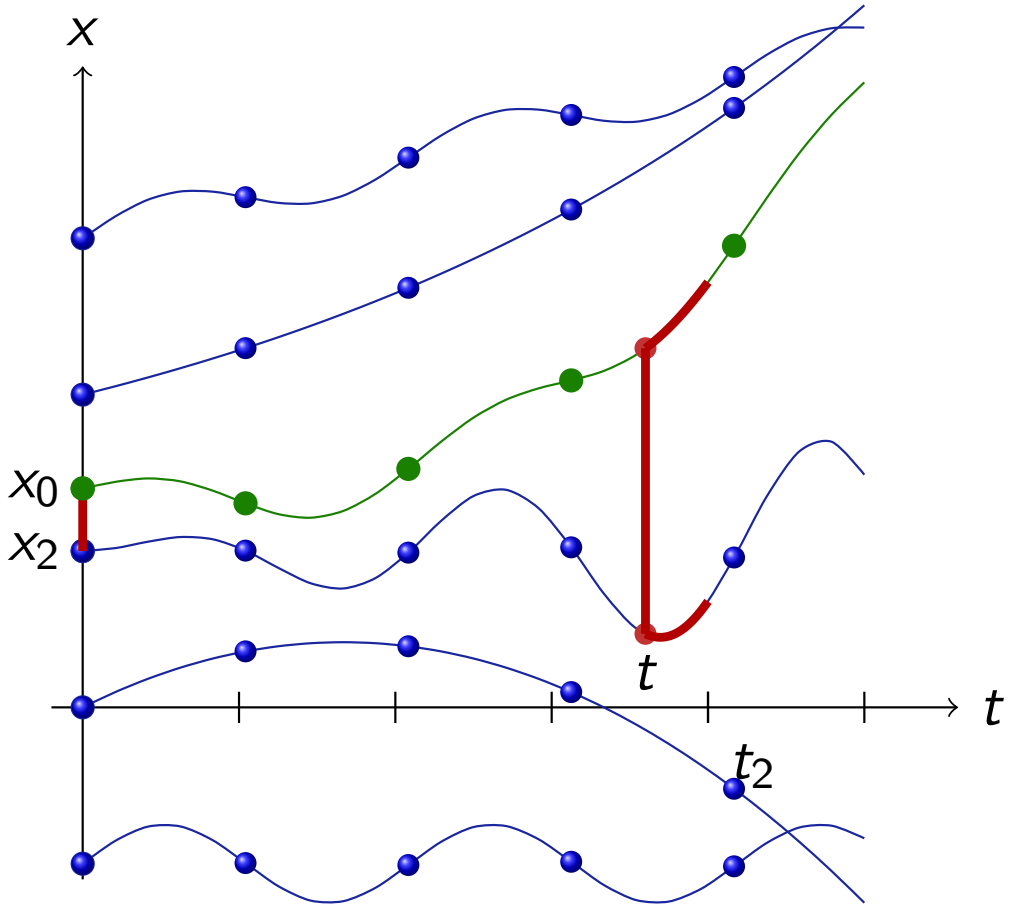
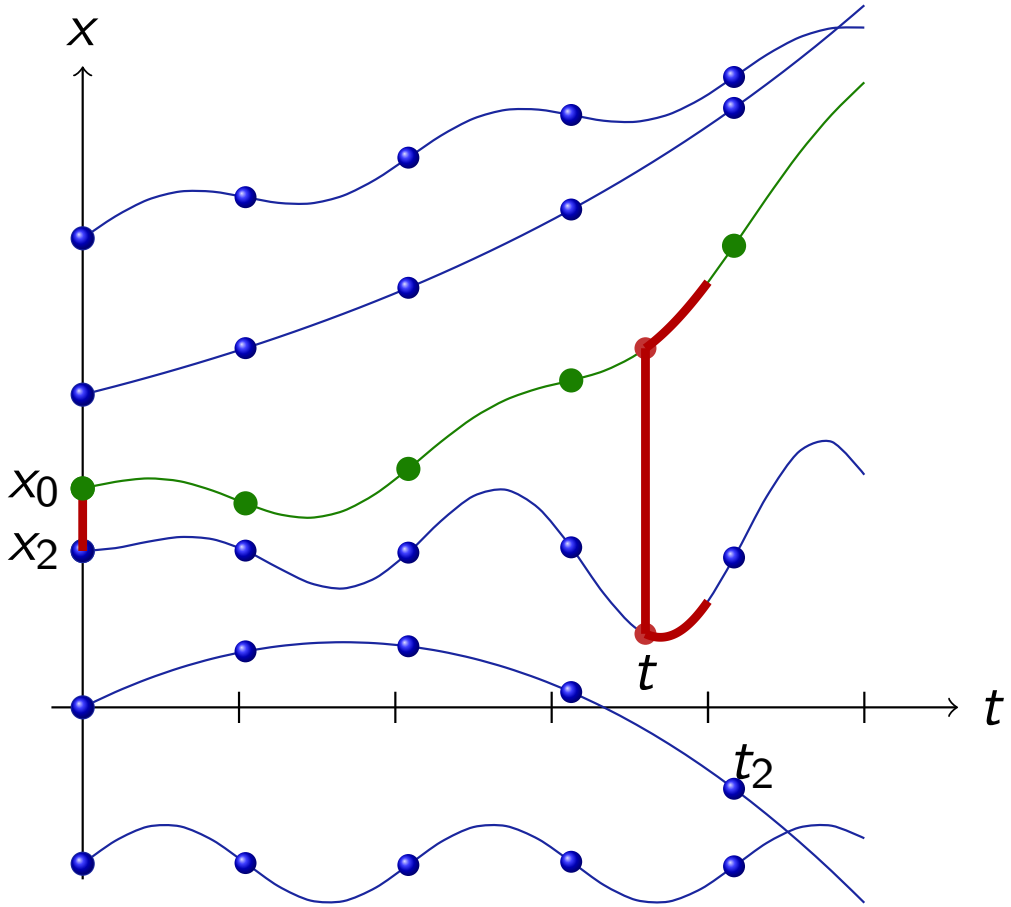


Image Computation: Continuous Systems

Problem (Image Computation – ODE transition)

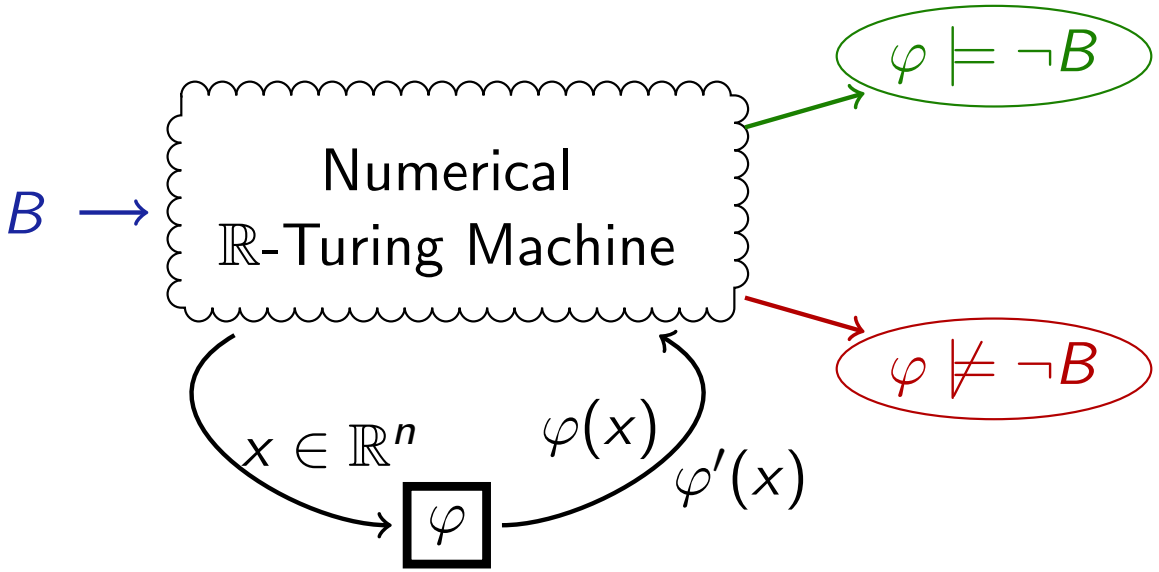
Flow $\varphi : [0, \infty) \times \mathbb{R}^n \to \mathbb{R}^n$ solving $x' = f(x)$ reaches state B?

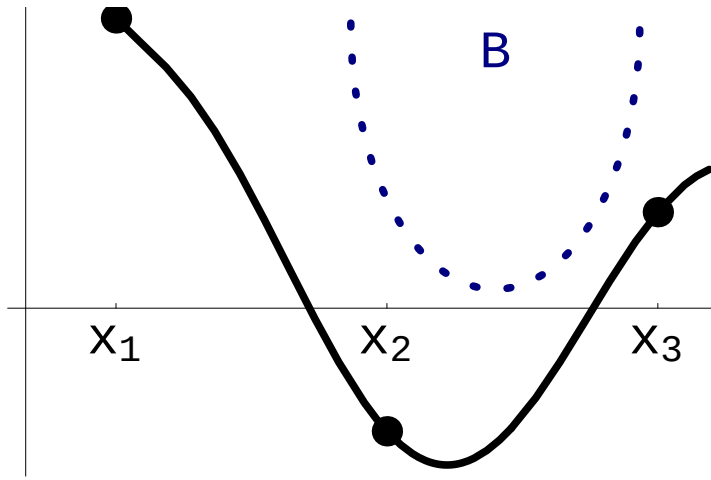


errors!

too many!

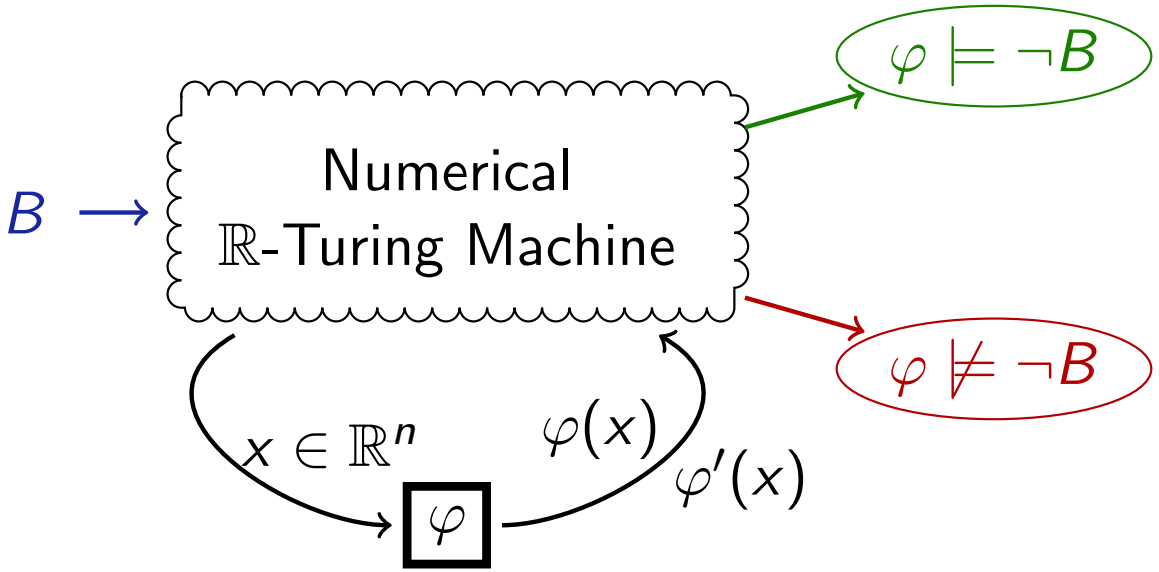
Numerical Image Computation

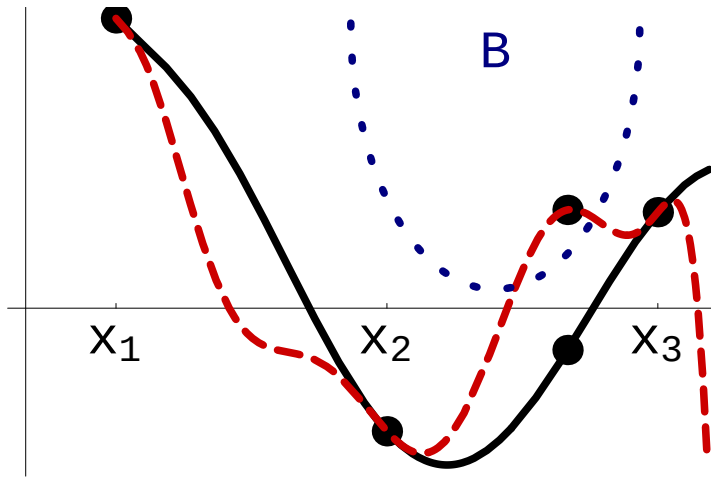




André Platzer and Edmund M. Clarke.
The image computation problem in hybrid systems model checking.
HSCC, vol. 4416 of *LNCS*, 473–486. Springer, 2007.

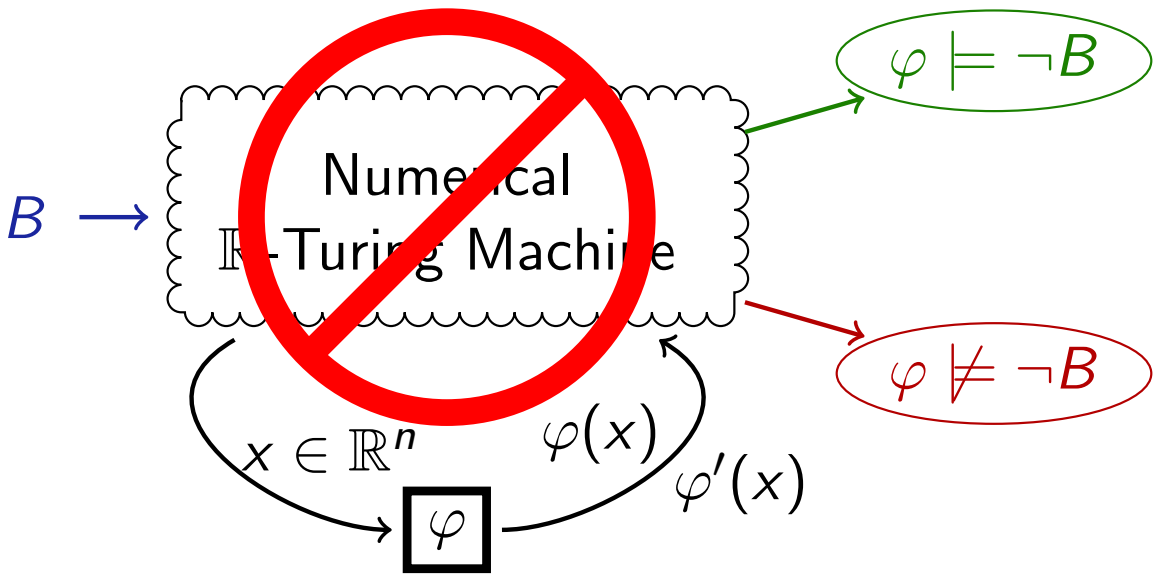
Numerical Image Computation

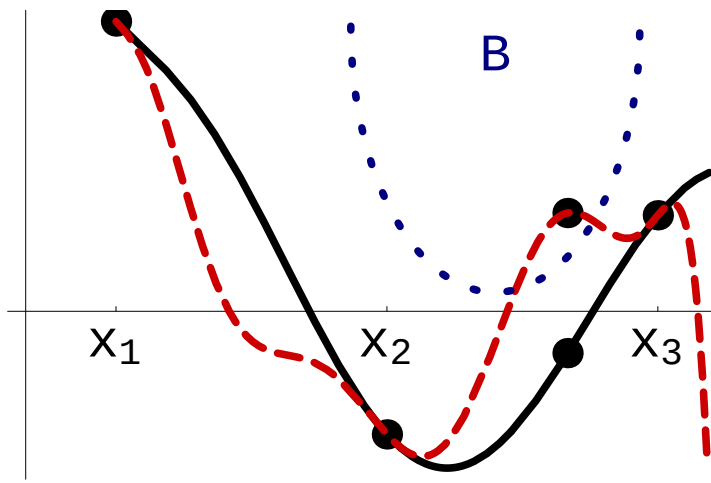




André Platzer and Edmund M. Clarke.
The image computation problem in hybrid systems model checking.
HSCC, vol. 4416 of *LNCS*, 473–486. Springer, 2007.

Numerical Image Computation





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.

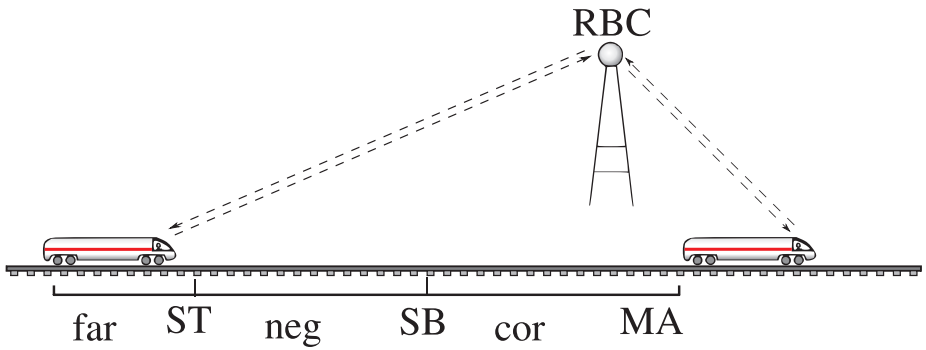
Outline

Outline (Symbolic Approach)

dℒ Design: The Logic of Hybrid Systems

differential dynamic logic

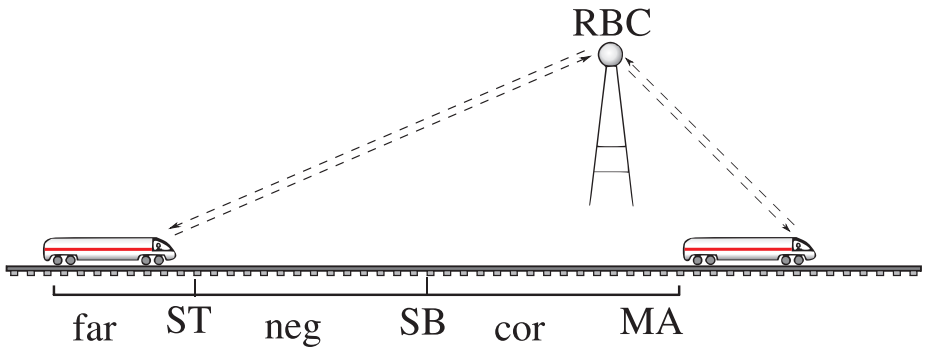
$$\mathsf{d}\mathcal{L} = \mathsf{DL} + \mathsf{HP}$$

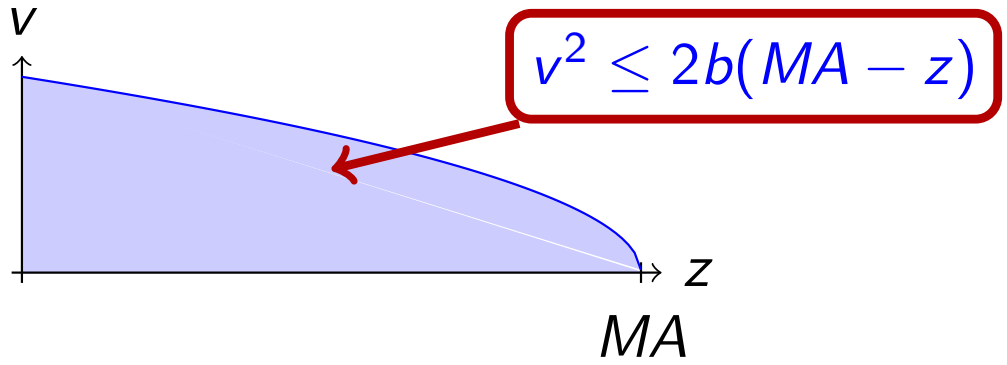


dℒ Design: Regions in First-Order Logic

differential dynamic logic

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

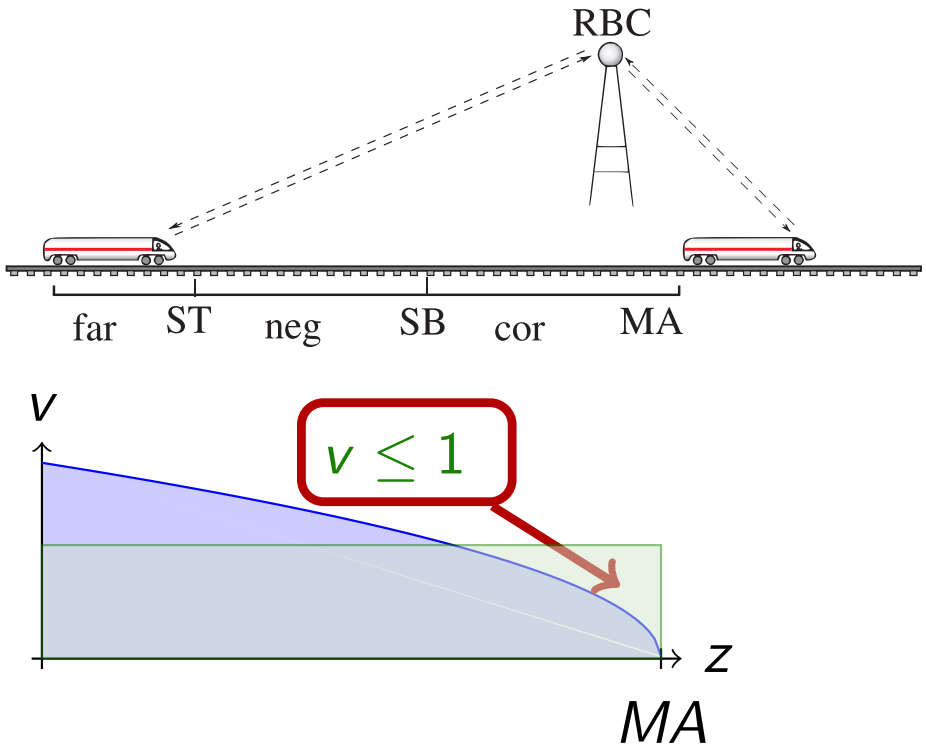




dℒ Design: Regions in First-Order Logic

differential dynamic logic

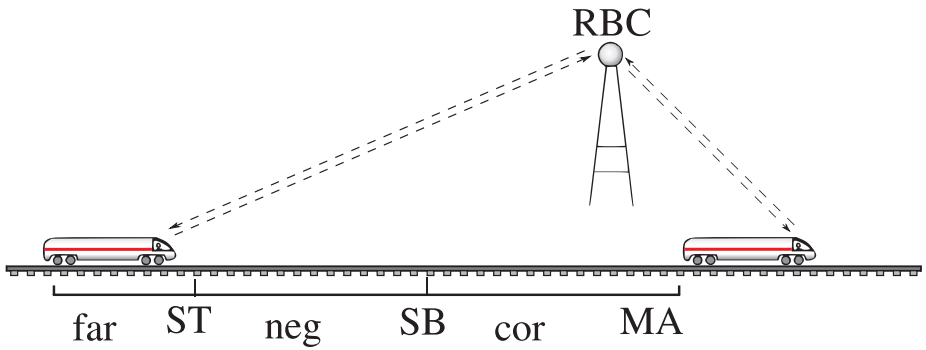
$d\mathcal{L} = \text{FOL}_{\mathbb{R}}$

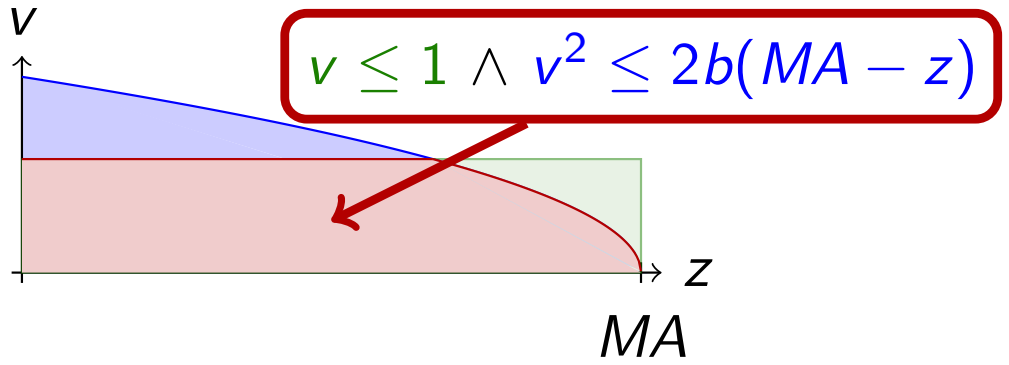


dℒ Design: Regions in First-Order Logic

differential dynamic logic

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

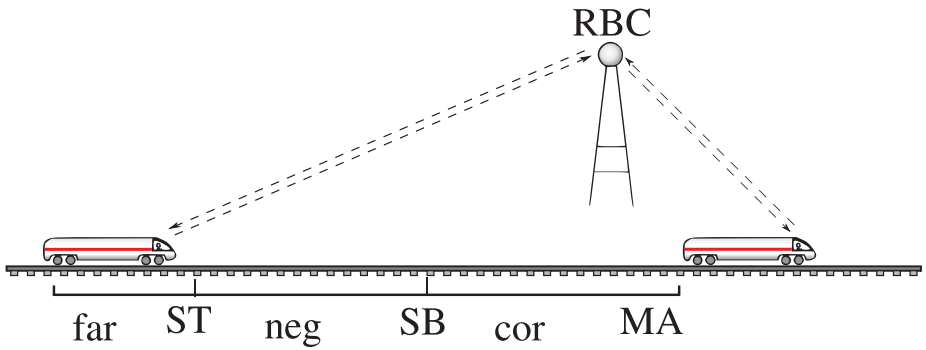


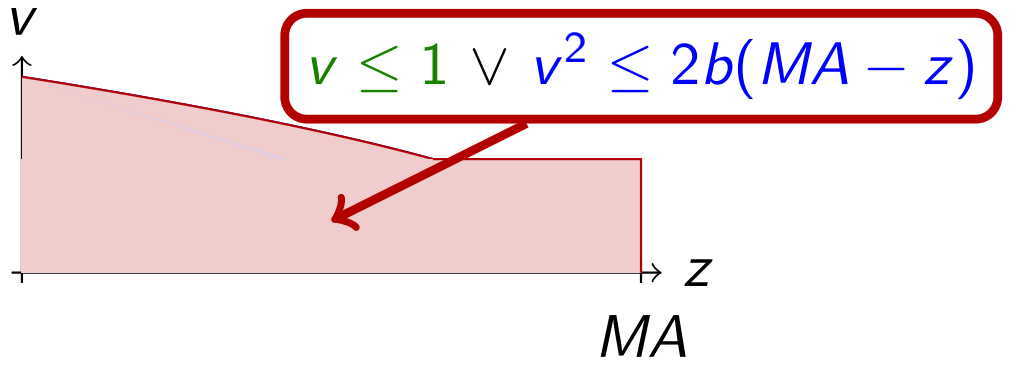


dℒ Design: Regions in First-Order Logic

differential dynamic logic

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





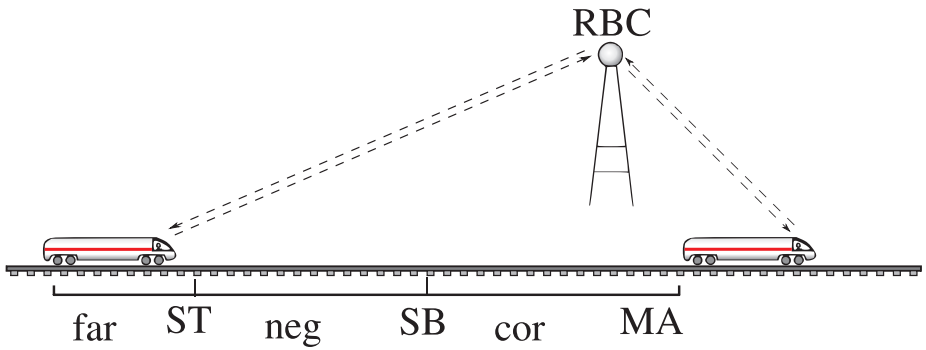
dℒ Design: Regions in First-Order Logic

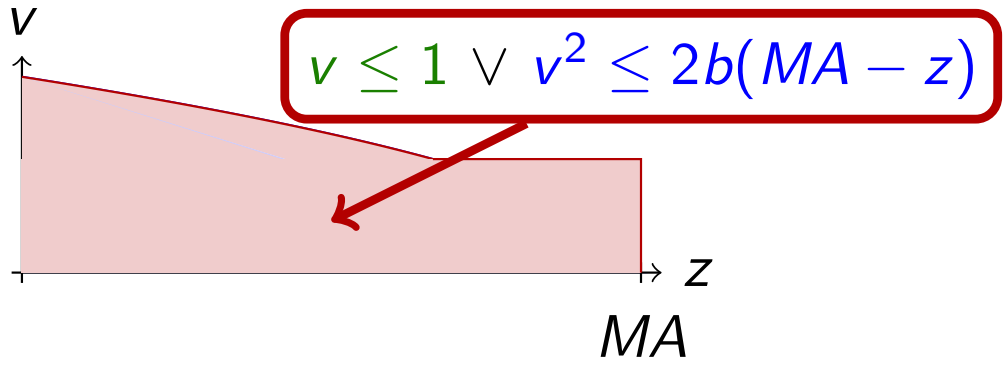
differential dynamic logic

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

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

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

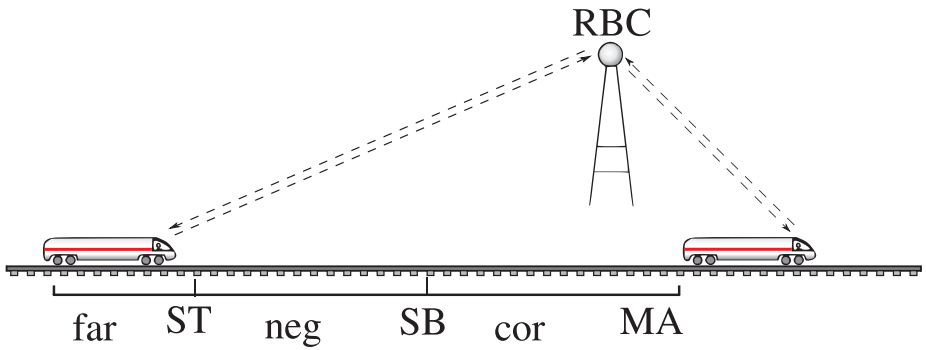


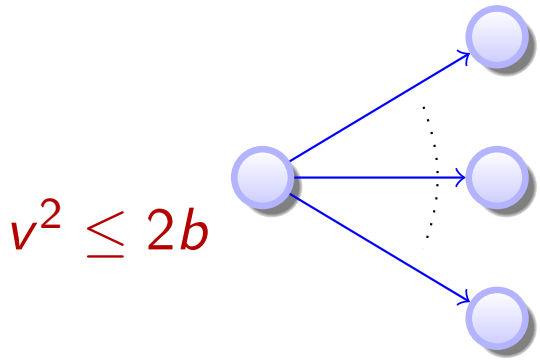


dℒ Design: State Transitions in Dynamic Logic

differential dynamic logic

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

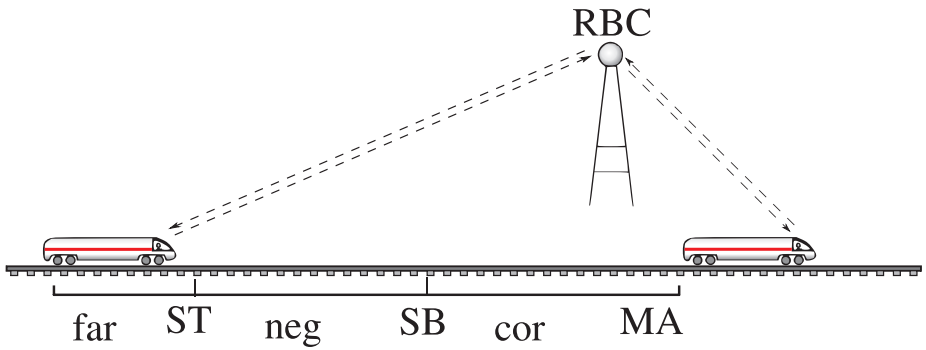


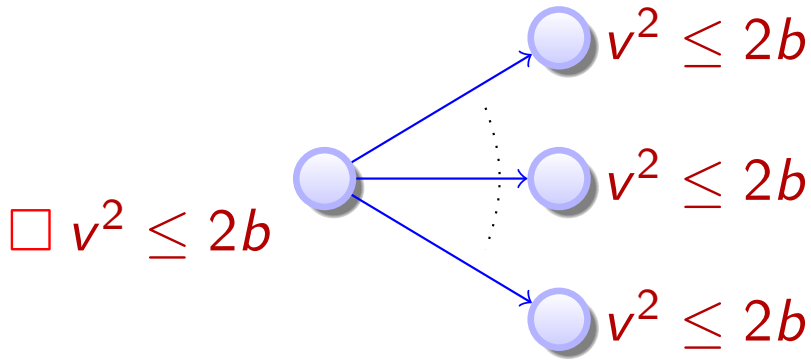


dℒ Design: State Transitions in Dynamic Logic

differential dynamic logic

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

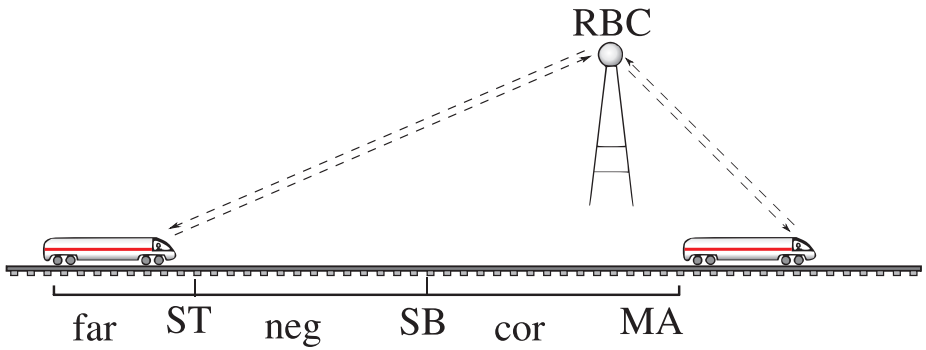


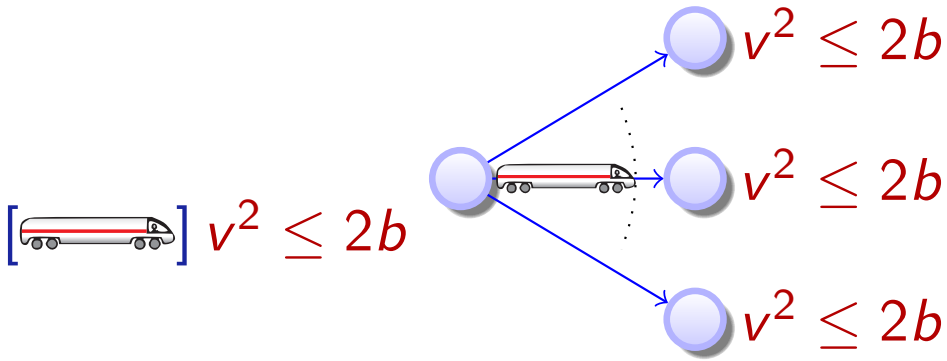


dℒ Design: State Transitions in Dynamic Logic

differential dynamic logic

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

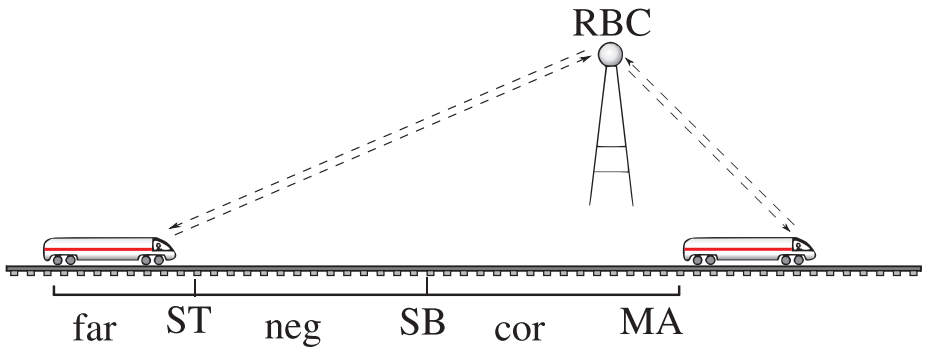




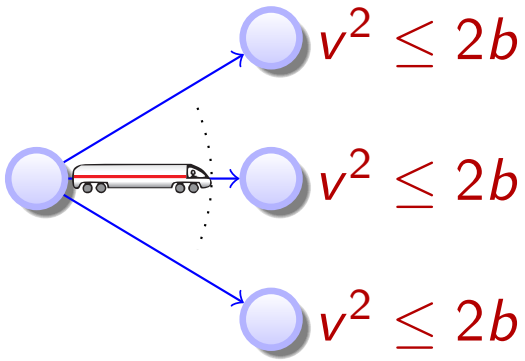
dℒ Design: Hybrid Programs as Uniform Model

differential dynamic logic

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



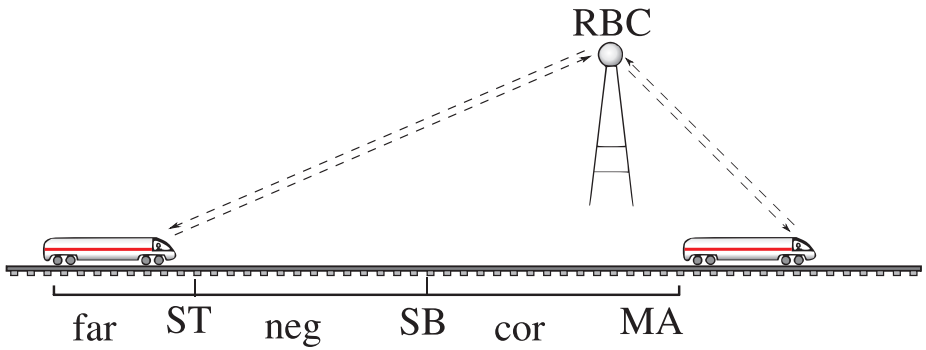
$[z'' = a]\, v^2 \leq 2b$



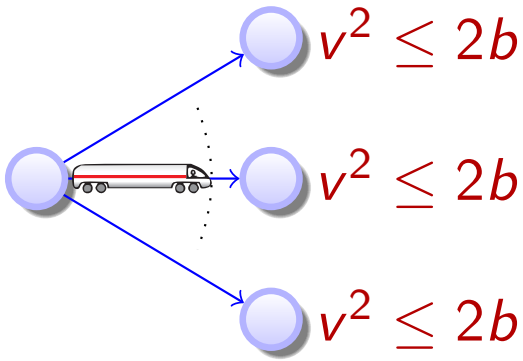
dℒ Design: Hybrid Programs as Uniform Model

differential dynamic logic

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



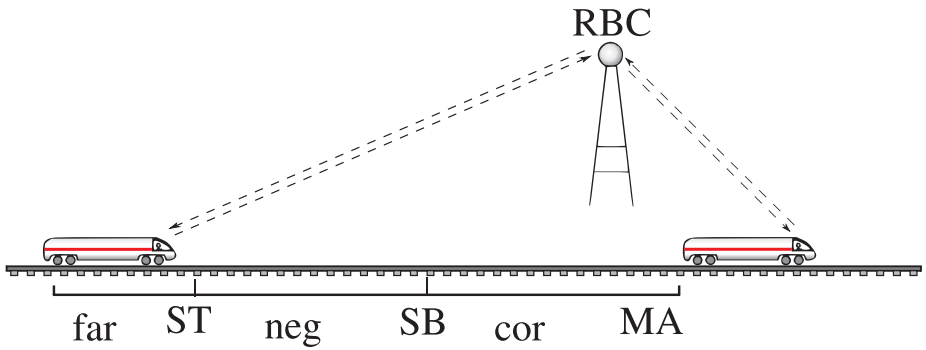
$$[\texttt{if}(z > SB)\, a := -b;\ z'' = a]\, v^2 \leq 2b$$



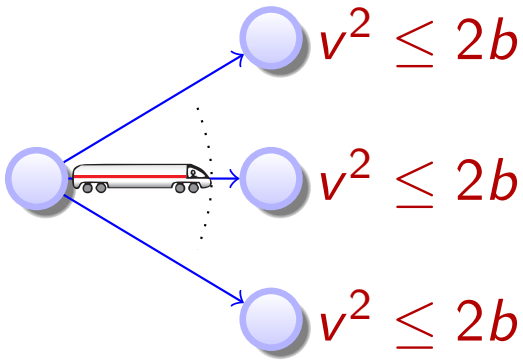
dℒ Design: Hybrid Programs as Uniform Model

differential dynamic logic

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



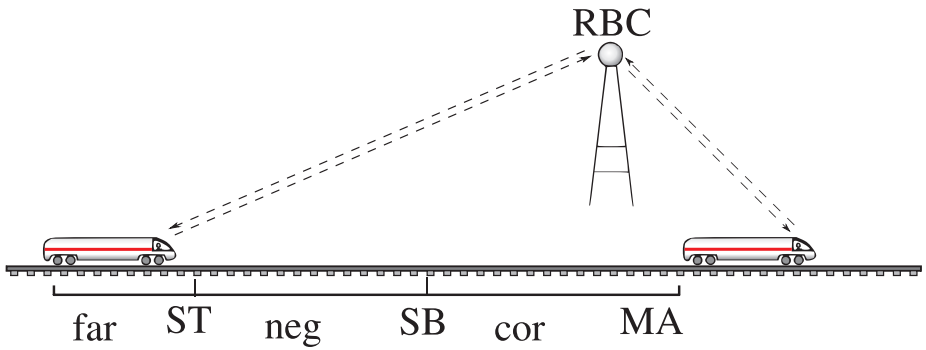
$$[\underbrace{\texttt{if}(z > SB)\, a := -b;\ z'' = a}_{\text{hybrid program}}]\, v^2 \leq 2b$$



dℒ Design: Hybrid Programs as Uniform Model

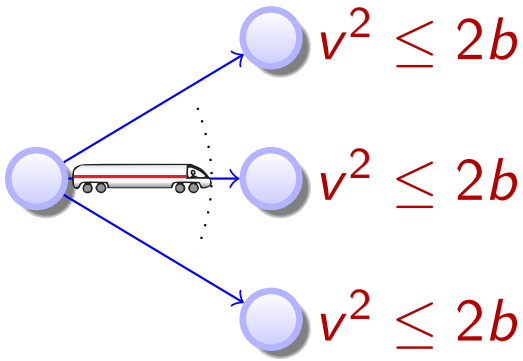
differential dynamic logic

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



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

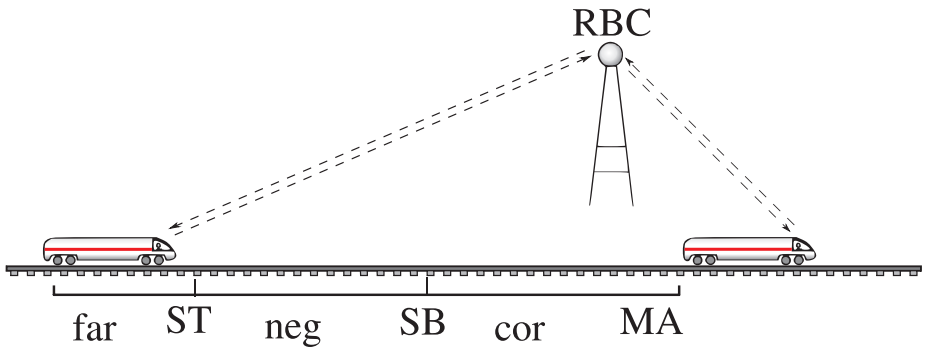
Initial condition



dℒ Design: Hybrid Programs as Uniform Model

differential dynamic logic

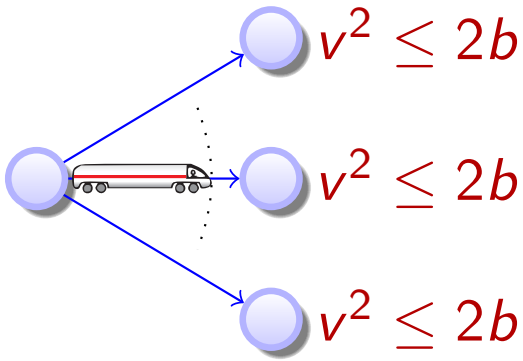
$$\mathrm{d}\mathcal{L} = \mathrm{FOL}_{\mathbb{R}} + \mathrm{DL} + \mathrm{HP}$$



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

Initial condition

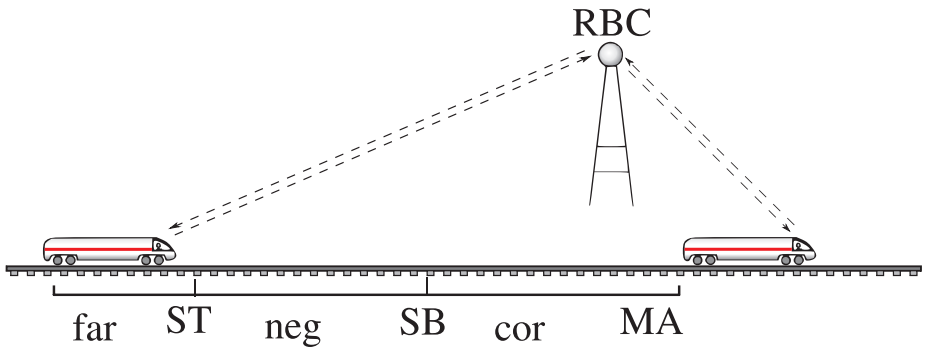
System dynamics

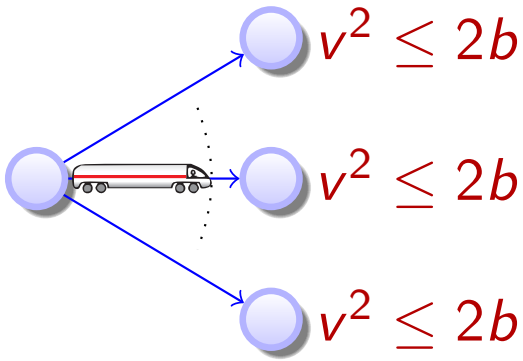


dℒ Design: Hybrid Programs as Uniform Model

differential dynamic logic

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





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

Initial condition

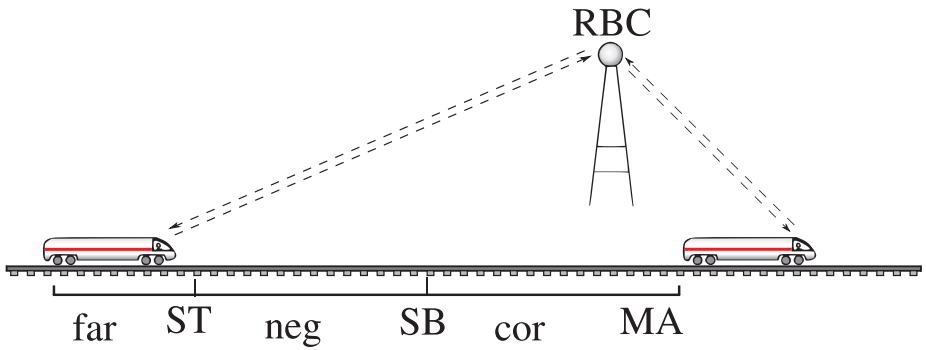
System dynamics

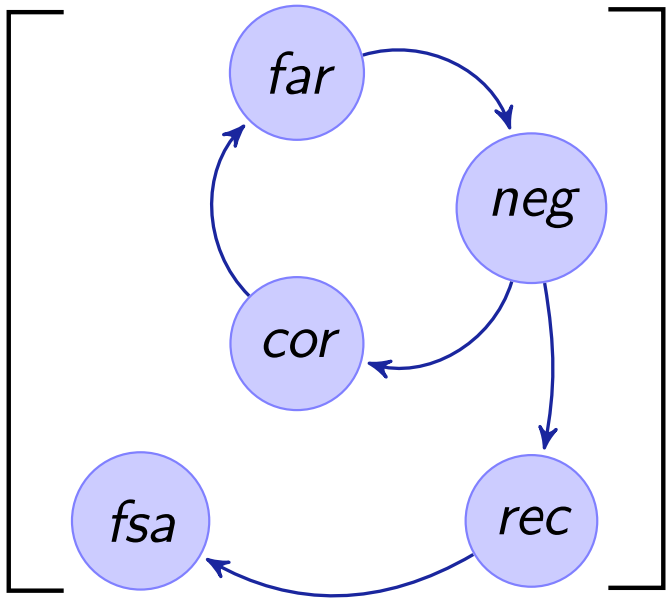
Post condition

dℒ Design: What about Hybrid Automata?

differential dynamic logic

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



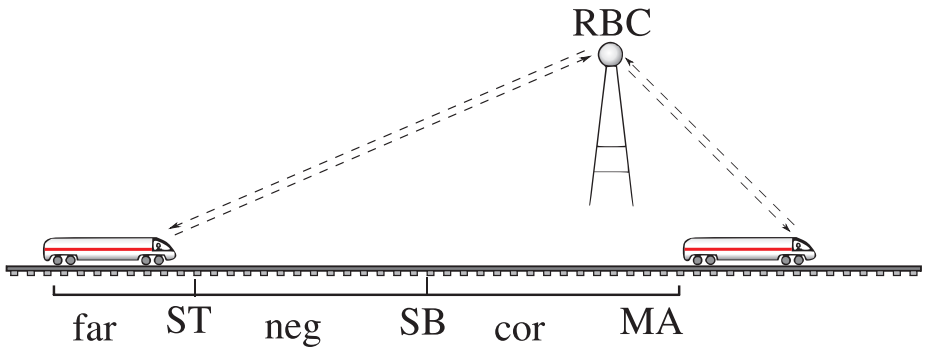


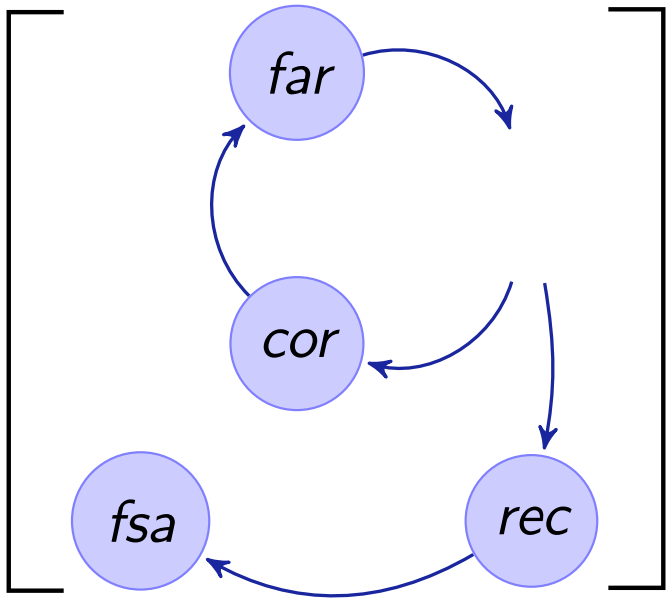
$v^2 \leq 2b\,..$

dℒ Design: What about Hybrid Automata?

differential dynamic logic

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



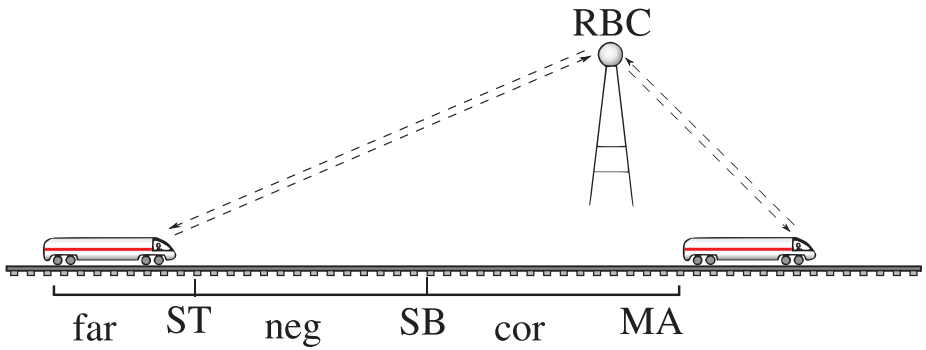


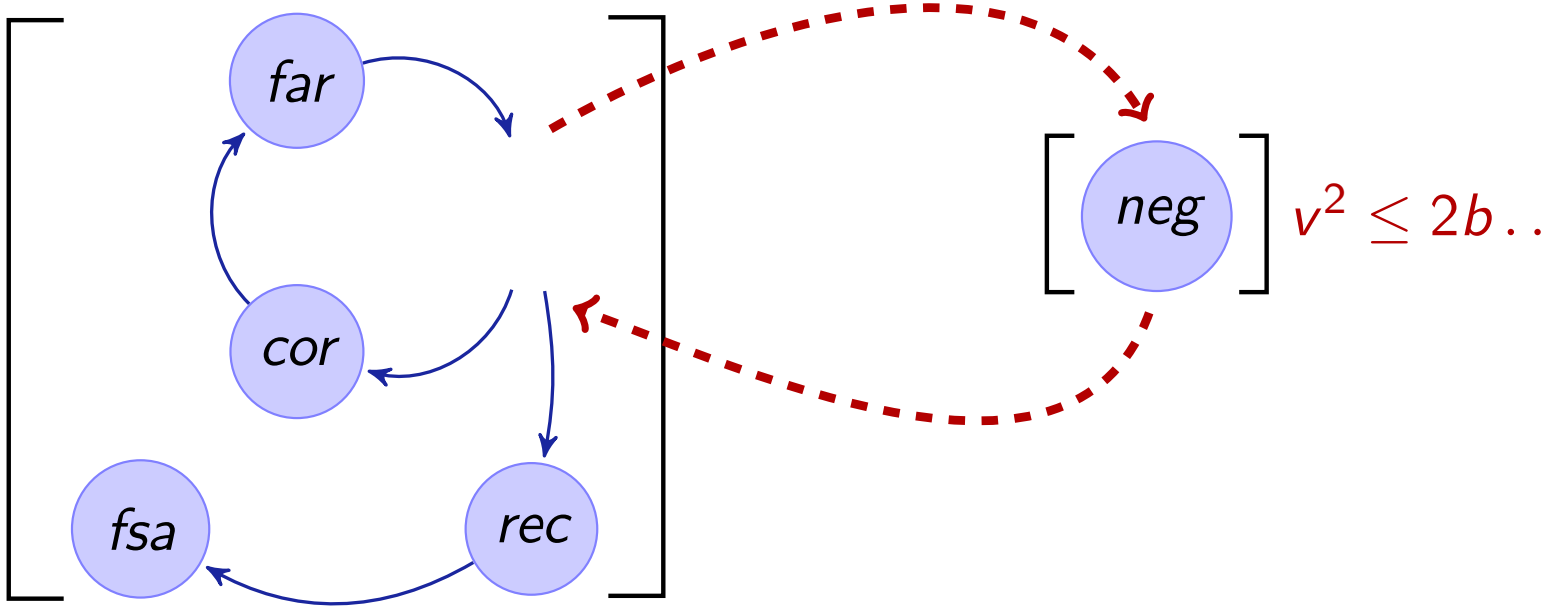
$$[\,\textit{neg}\,]\; v^2 \leq 2b\,..$$

dℒ Design: What about Hybrid Automata?

differential dynamic logic

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

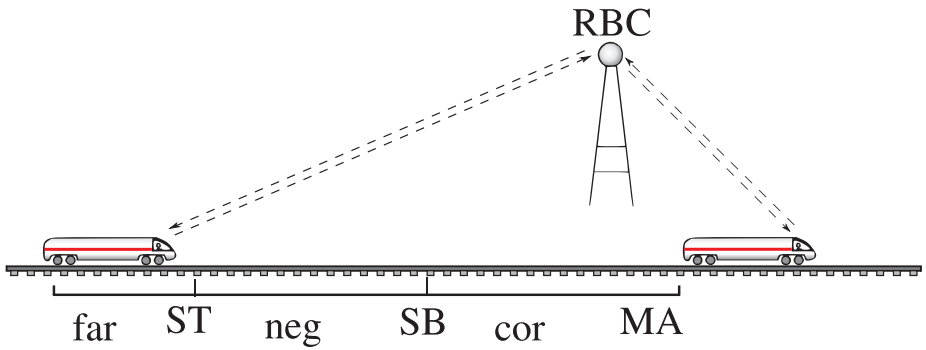


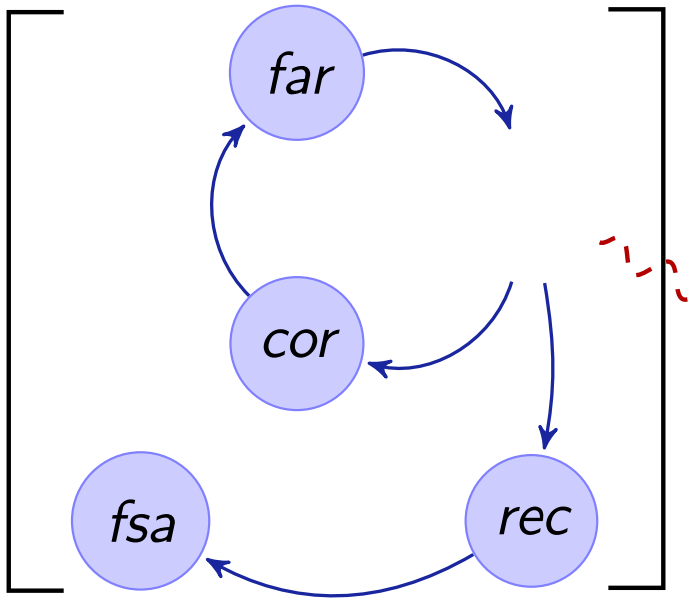


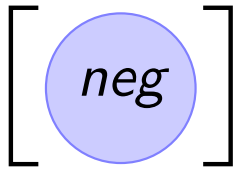
dℒ Design: What about Hybrid Automata?

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

Differential Dynamic Logic dℒ: Syntax

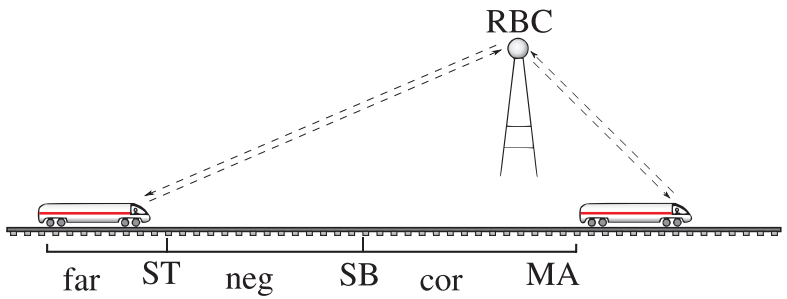
Definition (Hybrid program α)

$x' = f(x)$	(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

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

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

$$drive \equiv z'' = a \wedge v \geq 0 \wedge \tau \leq \varepsilon$$



Differential Dynamic Logic dℒ: Syntax

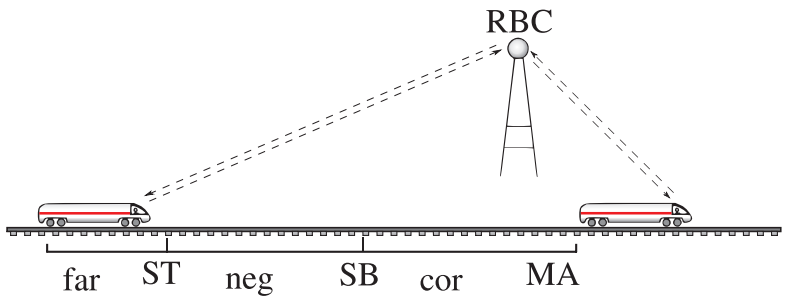
Definition (Hybrid program α)

$x' = f(x)$	(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

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

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

$$drive \equiv \tau := 0; z' = v, v' = a, \tau' = 1 \wedge v \geq 0 \wedge \tau \leq \varepsilon$$



Differential Dynamic Logic dℒ: Syntax

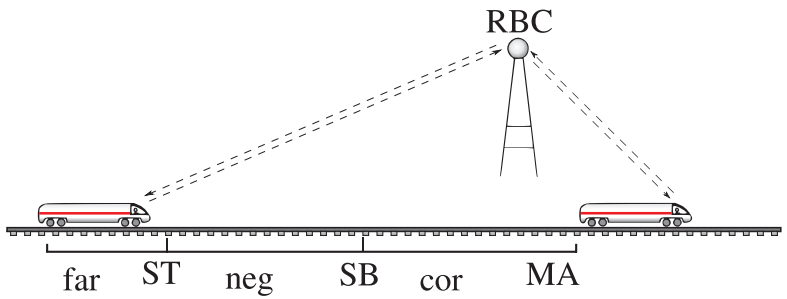
Definition (Hybrid program α)

$x' = f(x) \wedge H$	(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

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

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

$$drive \equiv \tau := 0; z' = v, v' = a, \tau' = 1 \wedge v \geq 0 \wedge \tau \leq \varepsilon$$



Branching Transitions in Hybrid Programs

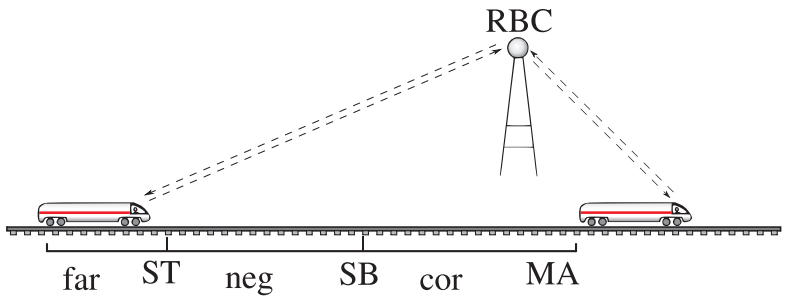
$$ETCS \equiv (ctrl\,;drive)^*$$

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

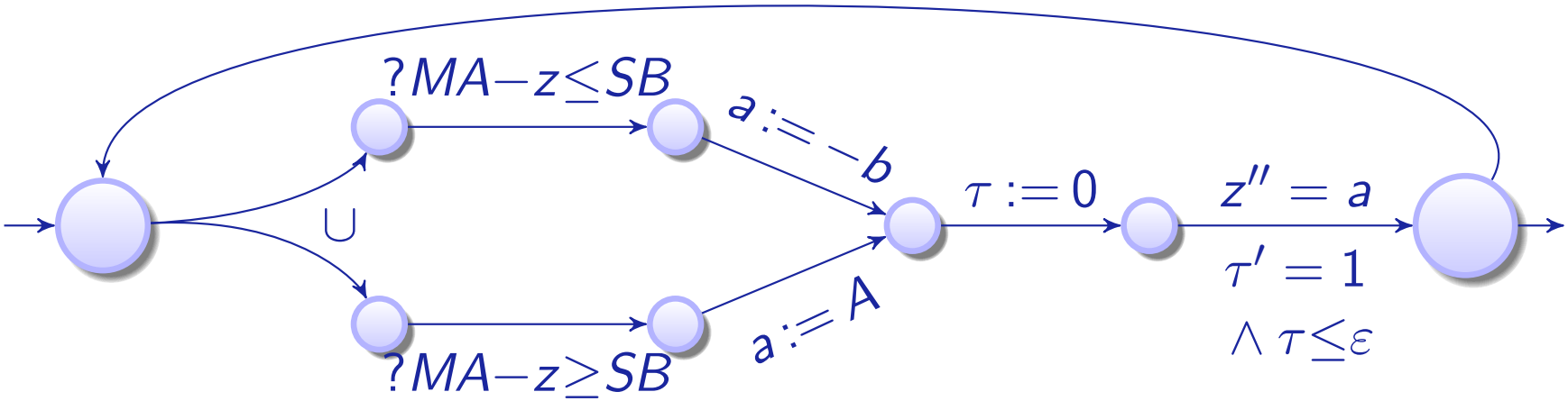
$$\cup\,(?MA - z \geq SB; a := A)$$

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

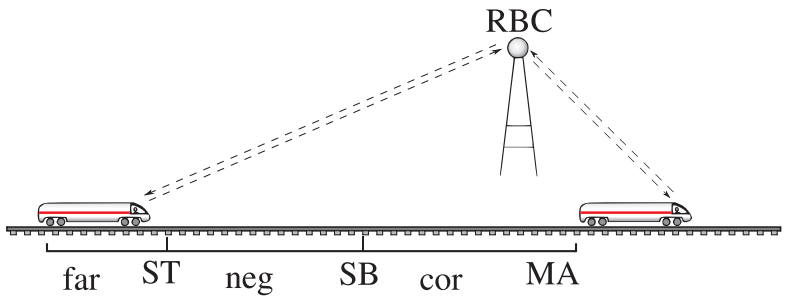
$$\wedge\, v \geq 0 \wedge \tau \leq \varepsilon)$$



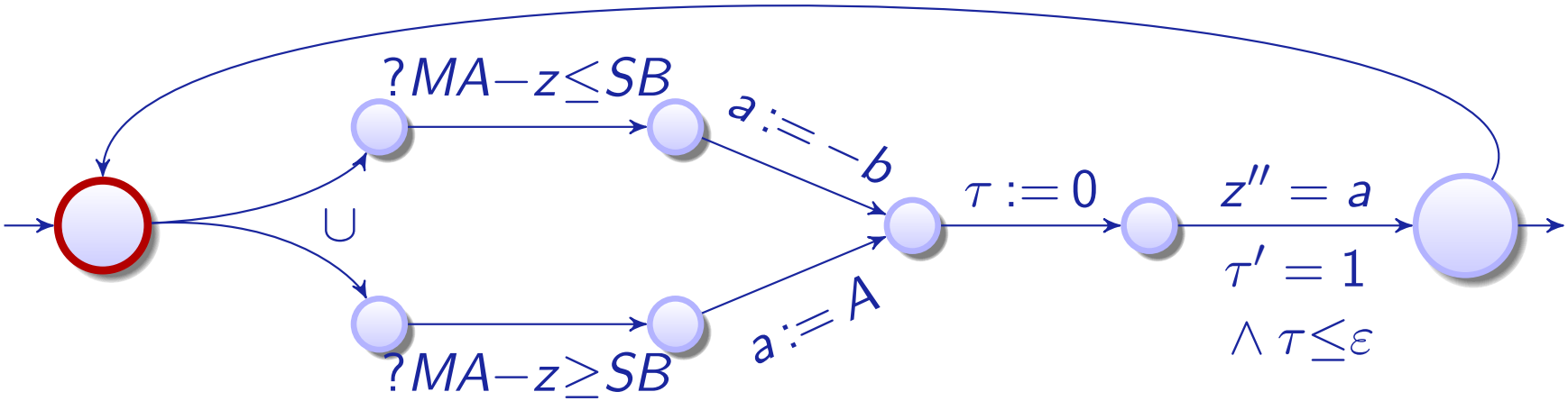
Branching Transitions in Hybrid Programs



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



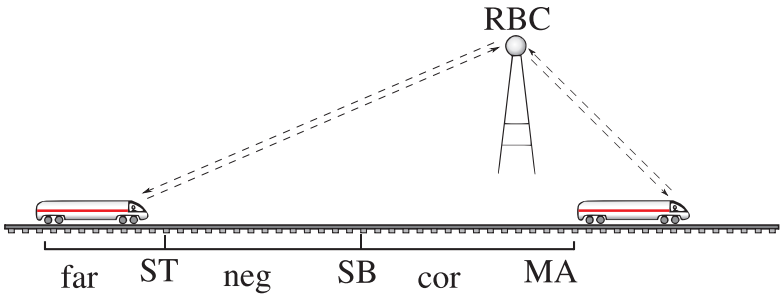
Branching Transitions in Hybrid Programs



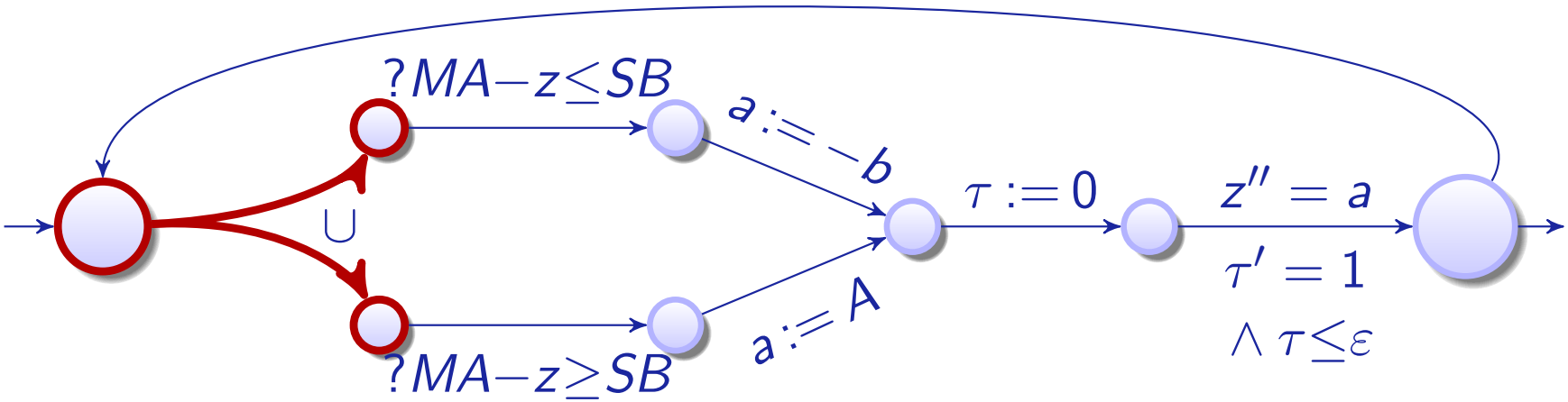
$$ETCS \equiv (ctrl\,;drive)^*$$

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

$$drive \equiv \tau := 0; (z' = v, v' = a, \tau' = 1 \wedge v \geq 0 \wedge \tau \leq \varepsilon)$$



Branching Transitions in Hybrid Programs



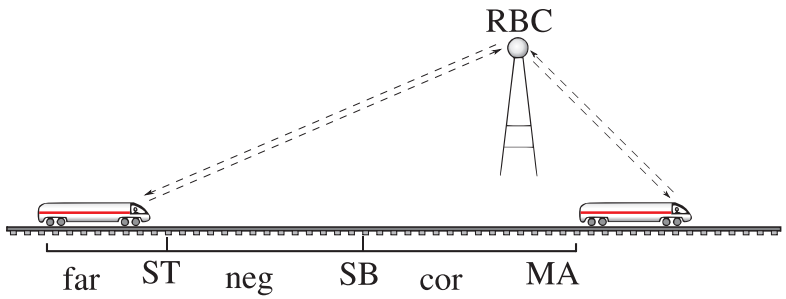
$$ETCS \equiv (ctrl\,;drive)^*$$

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

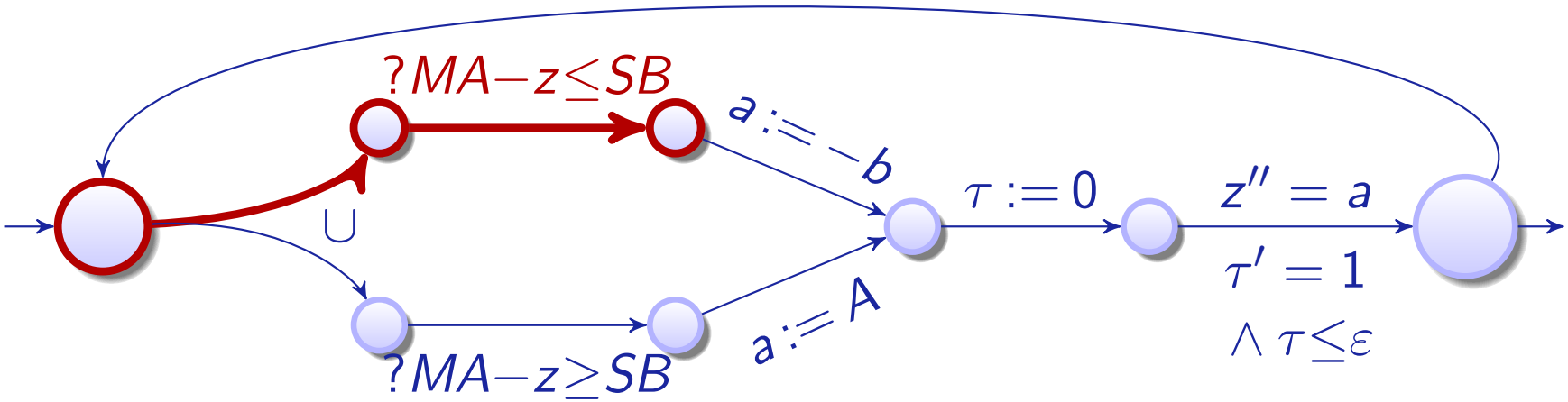
$$\cup\,(?MA - z \geq SB; a := A)$$

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

$$\wedge\, v \geq 0 \wedge \tau \leq \varepsilon)$$



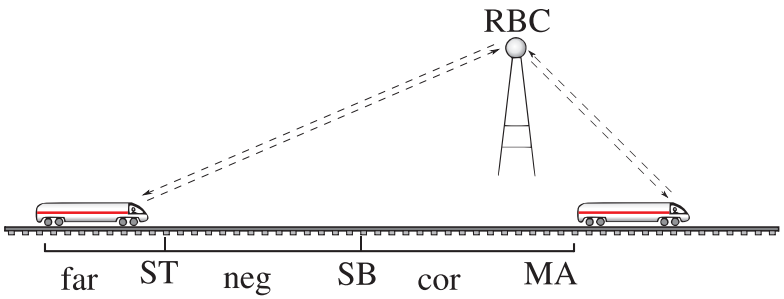
Branching Transitions in Hybrid Programs



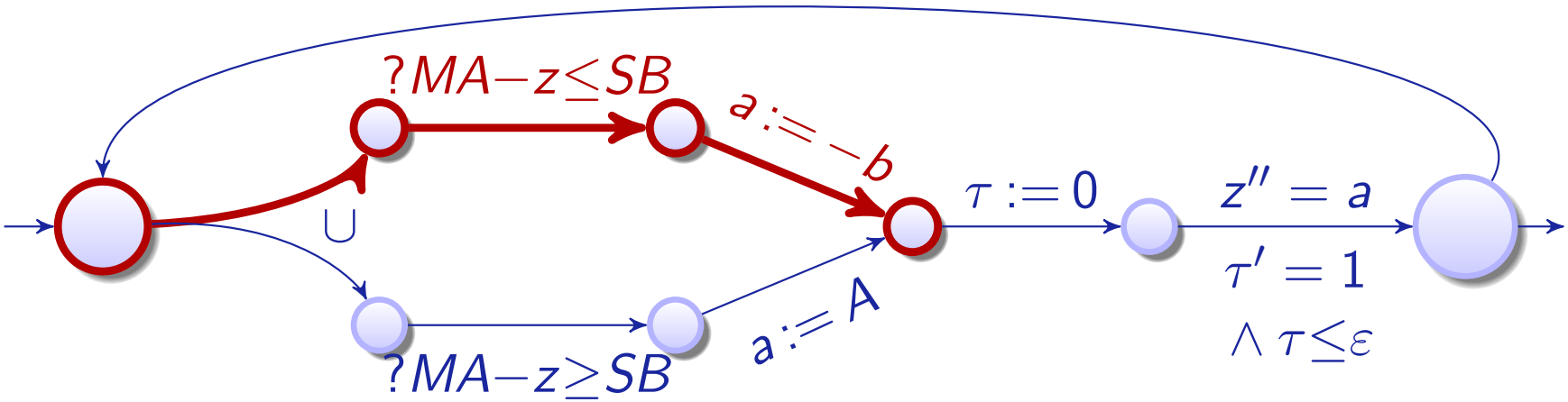
$$ETCS \equiv (ctrl\,;drive)^*$$

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

$$drive \equiv \tau := 0; (z' = v, v' = a, \tau' = 1 \wedge v \geq 0 \wedge \tau \leq \varepsilon)$$



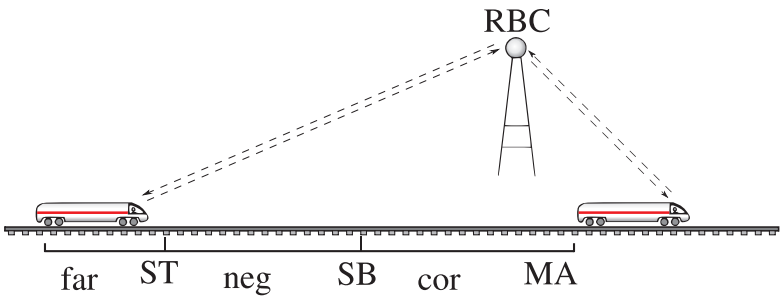
Branching Transitions in Hybrid Programs



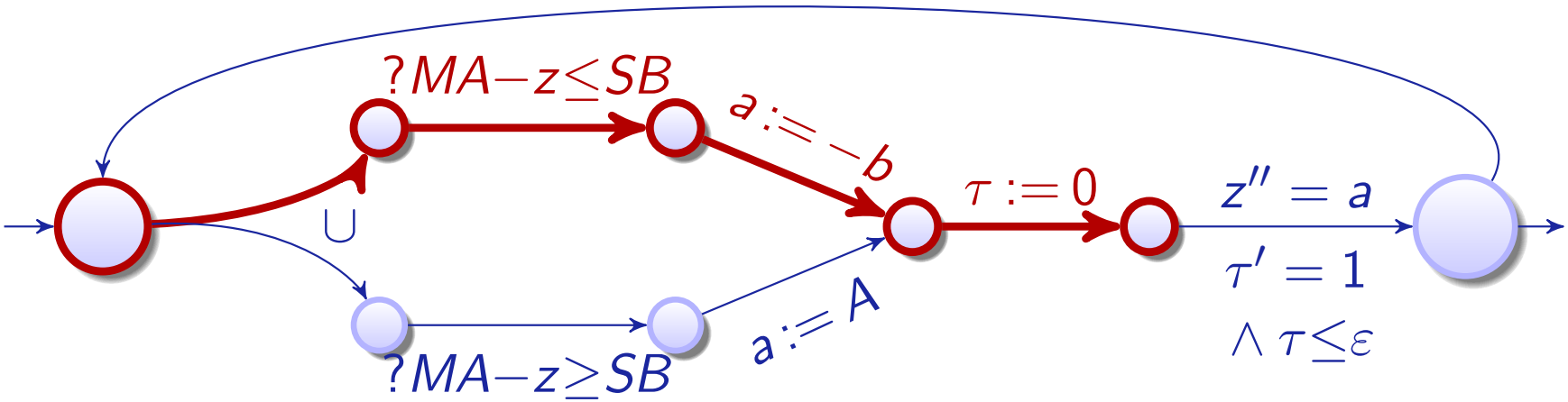
$$ETCS \equiv (ctrl\,;drive)^*$$

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

$$drive \equiv \tau := 0; (z' = v, v' = a, \tau' = 1 \wedge v \geq 0 \wedge \tau \leq \varepsilon)$$



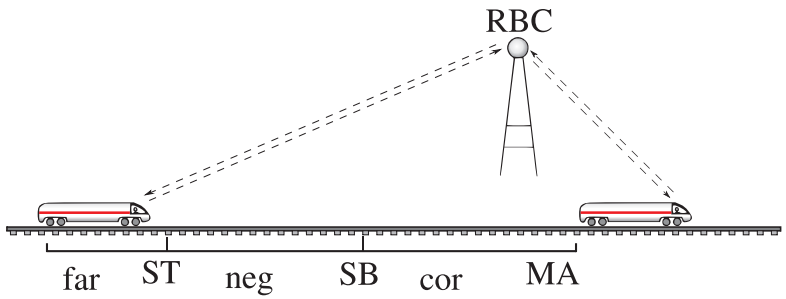
Branching Transitions in Hybrid Programs



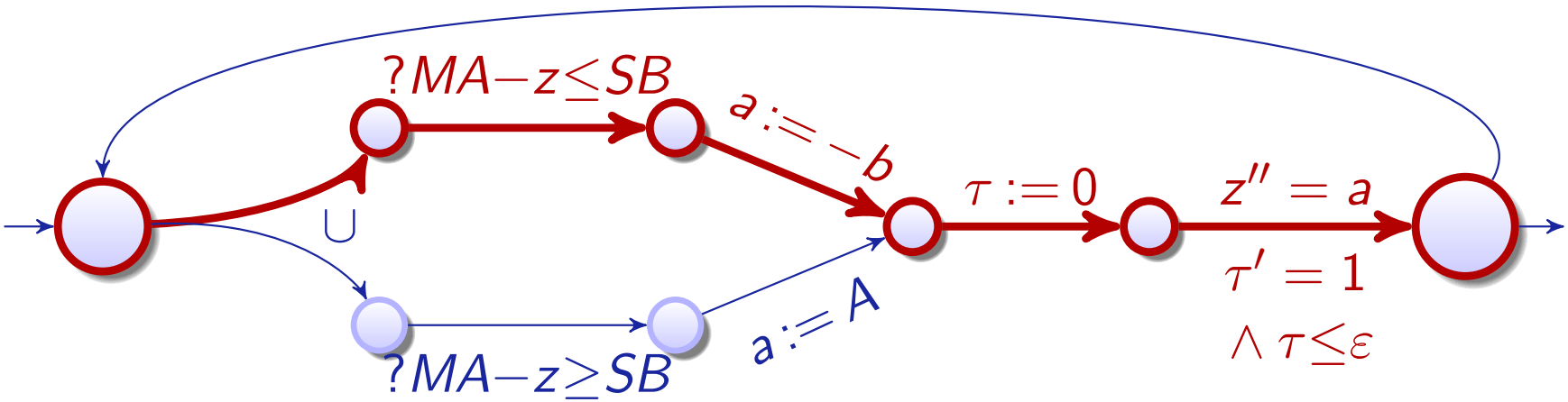
$$ETCS \equiv (ctrl; drive)^*$$

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

$$drive \equiv \tau := 0; (z' = v, v' = a, \tau' = 1 \wedge v \geq 0 \wedge \tau \leq \varepsilon)$$



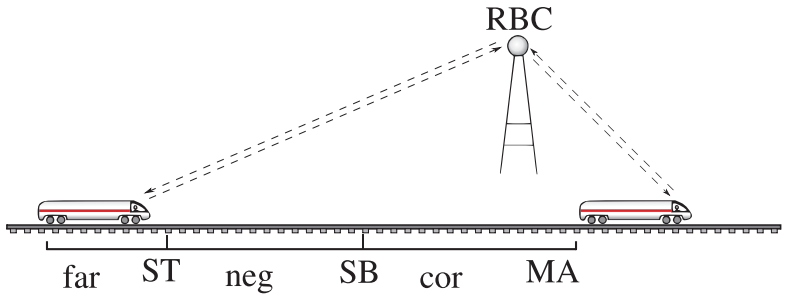
Branching Transitions in Hybrid Programs



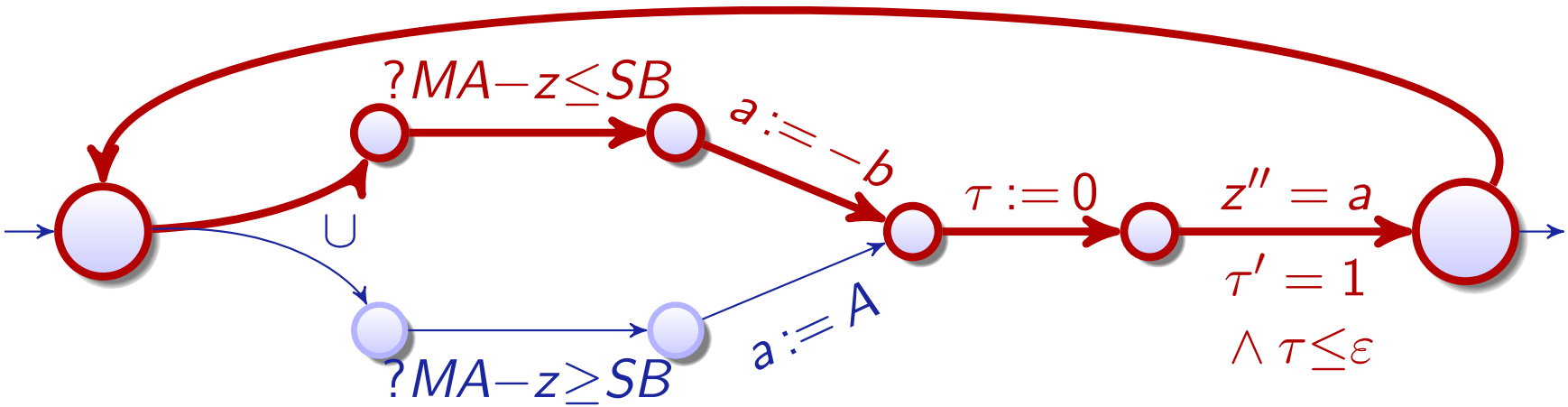
$$ETCS \equiv (ctrl\,;\, drive)^*$$

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

$$drive \equiv \tau := 0;\, (z' = v, v' = a, \tau' = 1 \wedge v \geq 0 \wedge \tau \leq \varepsilon)$$



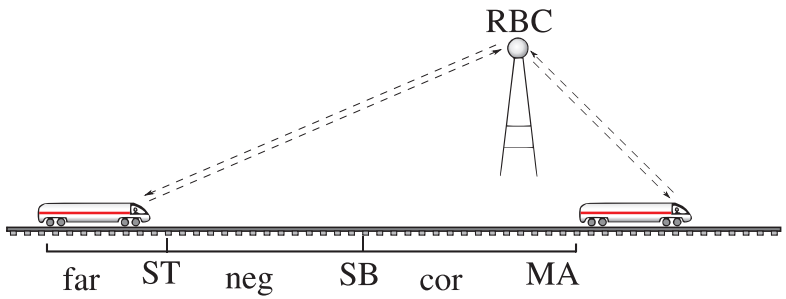
Branching Transitions in Hybrid Programs



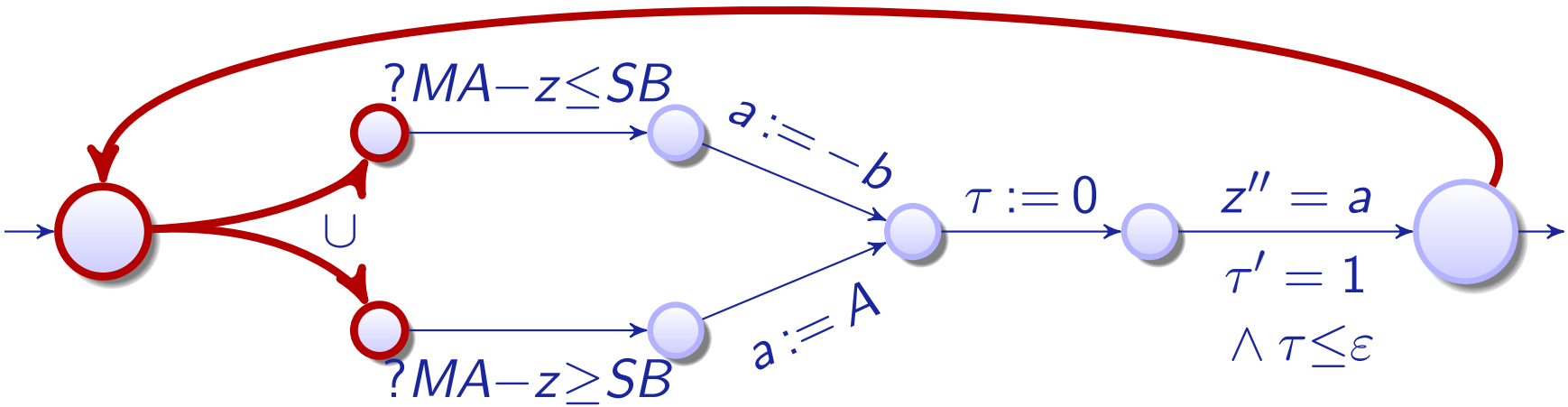
$$ETCS \equiv (ctrl\,;drive)^*$$

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

$$drive \equiv \tau := 0; (z' = v, v' = a, \tau' = 1 \wedge v \geq 0 \wedge \tau \leq \varepsilon)$$



Branching Transitions in Hybrid Programs



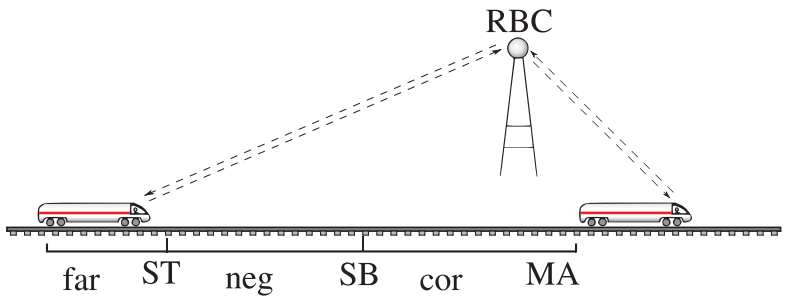
$$ETCS \equiv (ctrl\,;drive)^*$$

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

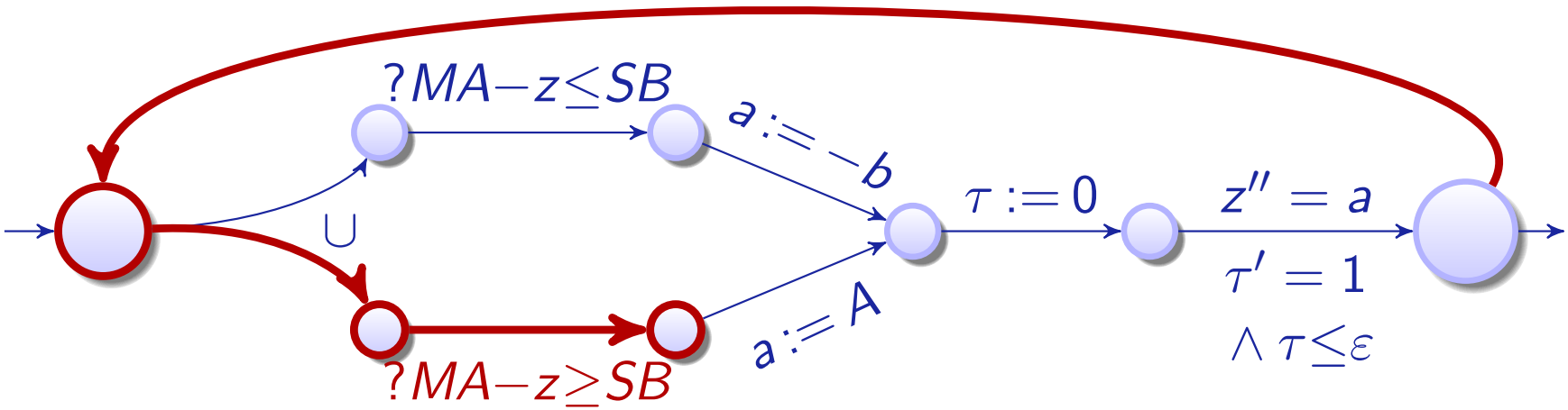
$$\cup\,(?MA - z \geq SB; a := A)$$

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

$$\wedge\, v \geq 0 \wedge \tau \leq \varepsilon)$$



Branching Transitions in Hybrid Programs



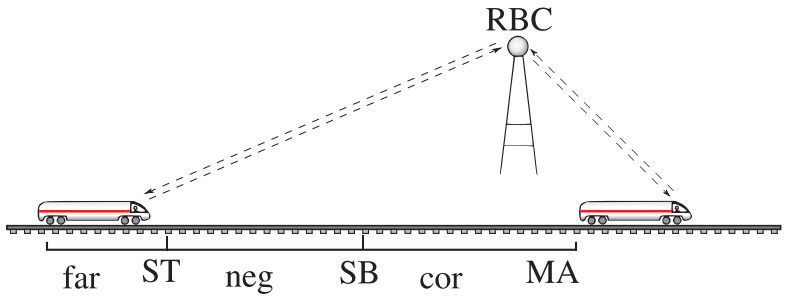
$$ETCS \equiv (ctrl\,;drive)^*$$

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

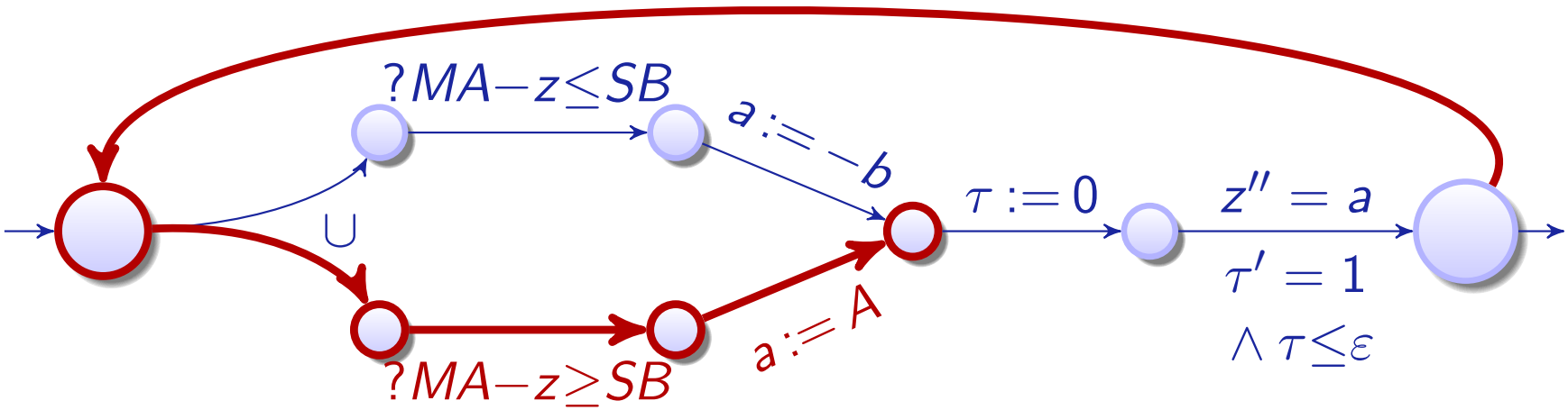
$$\cup\,(?MA - z \geq SB; a := A)$$

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

$$\wedge\, v \geq 0 \wedge \tau \leq \varepsilon)$$



Branching Transitions in Hybrid Programs



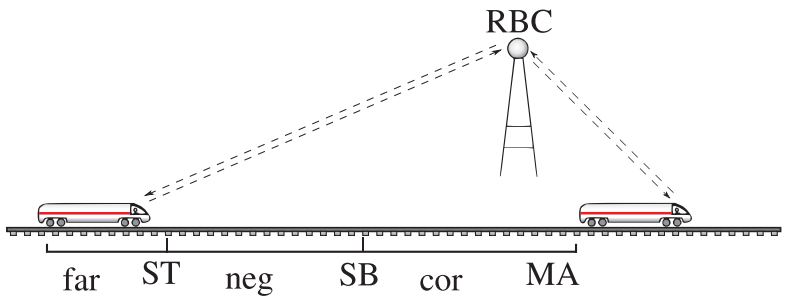
$$ETCS \equiv (ctrl\,;drive)^*$$

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

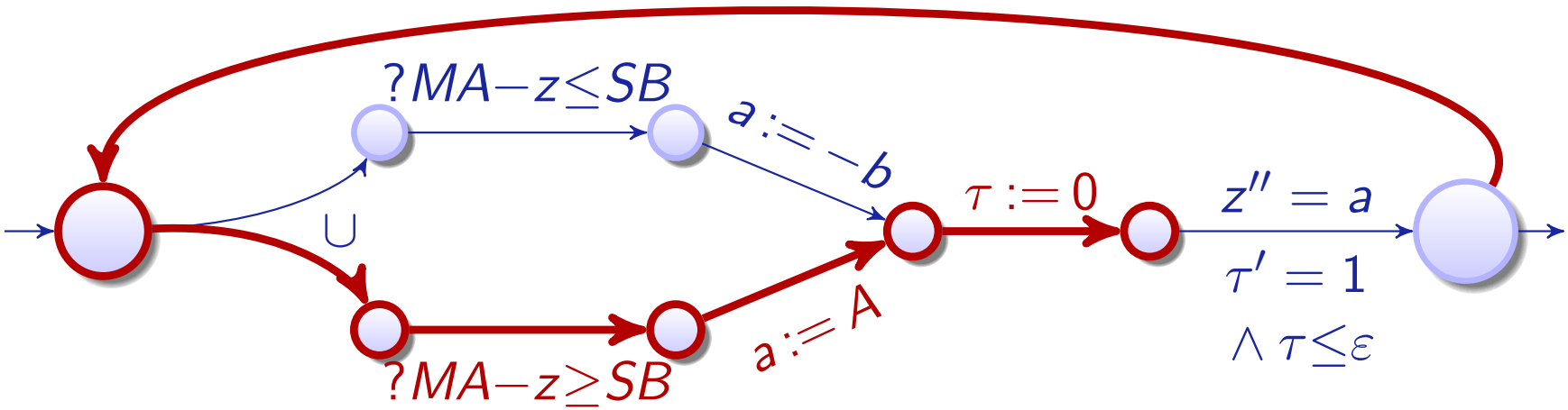
$$\cup\,(?MA - z \geq SB; a := A)$$

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

$$\wedge\, v \geq 0 \wedge \tau \leq \varepsilon)$$



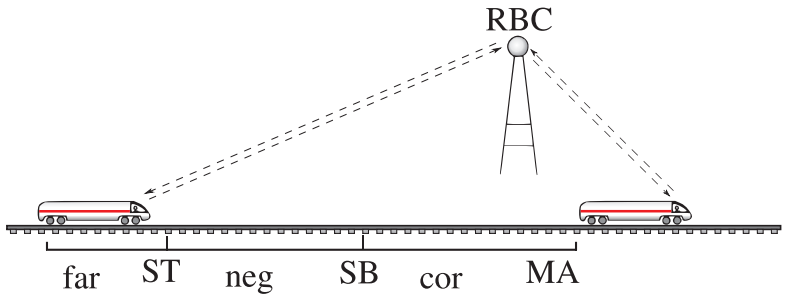
Branching Transitions in Hybrid Programs



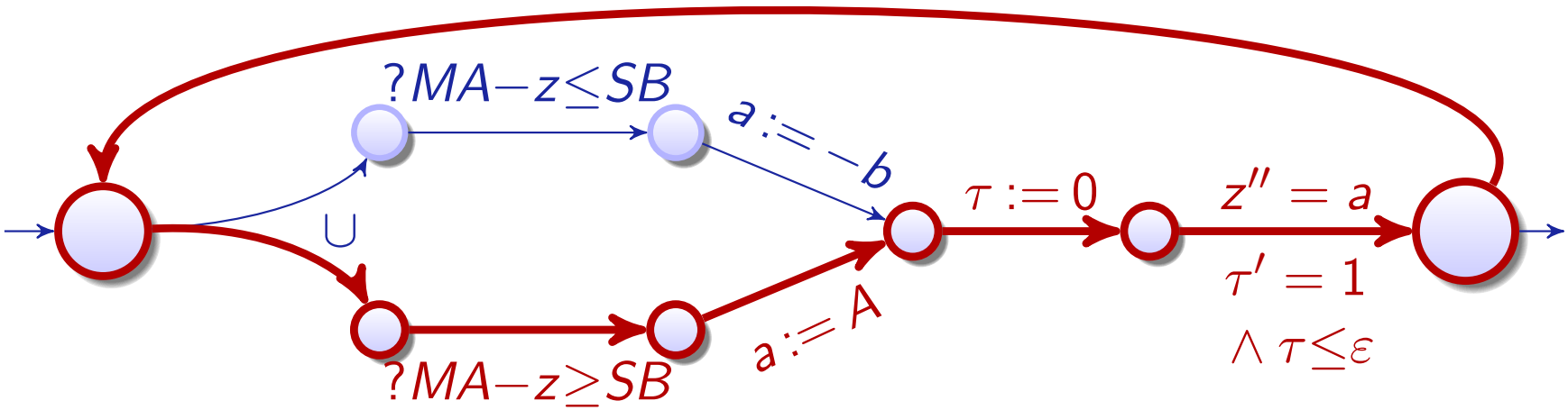
$$ETCS \equiv (ctrl\,;\, drive)^*$$

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

$$drive \equiv \tau := 0; (z' = v, v' = a, \tau' = 1 \wedge v \geq 0 \wedge \tau \leq \varepsilon)$$



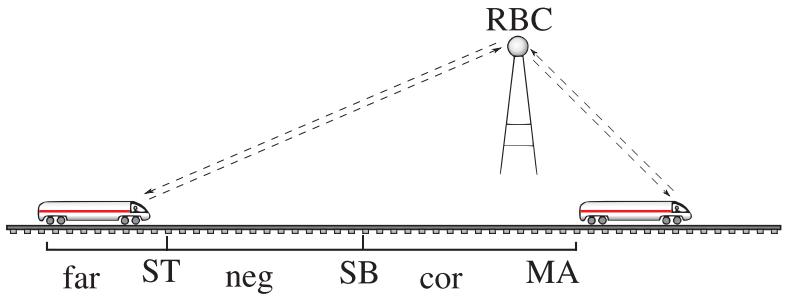
Branching Transitions in Hybrid Programs



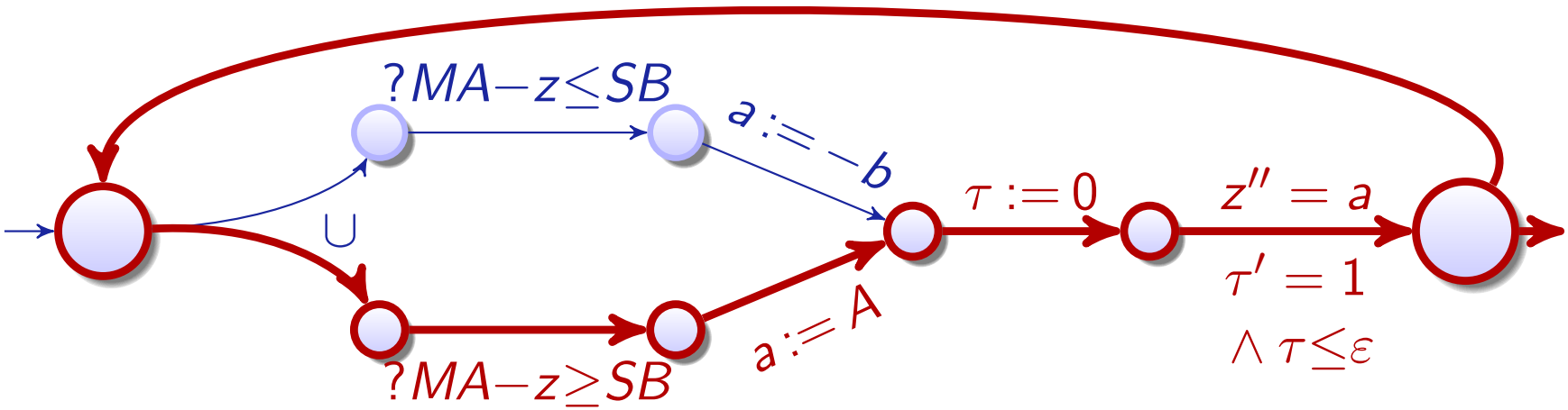
$$ETCS \equiv (ctrl\,;drive)^*$$

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

$$drive \equiv \tau := 0; (z' = v, v' = a, \tau' = 1 \wedge v \geq 0 \wedge \tau \leq \varepsilon)$$



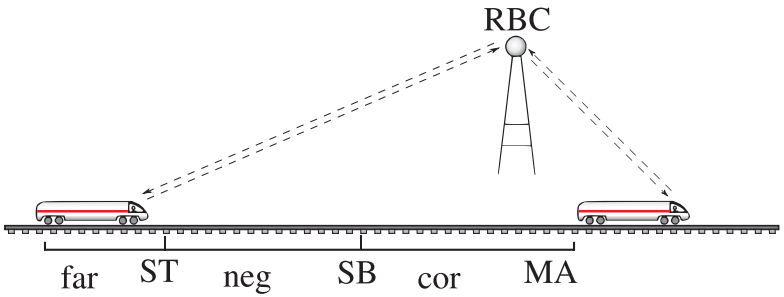
Branching Transitions in Hybrid Programs



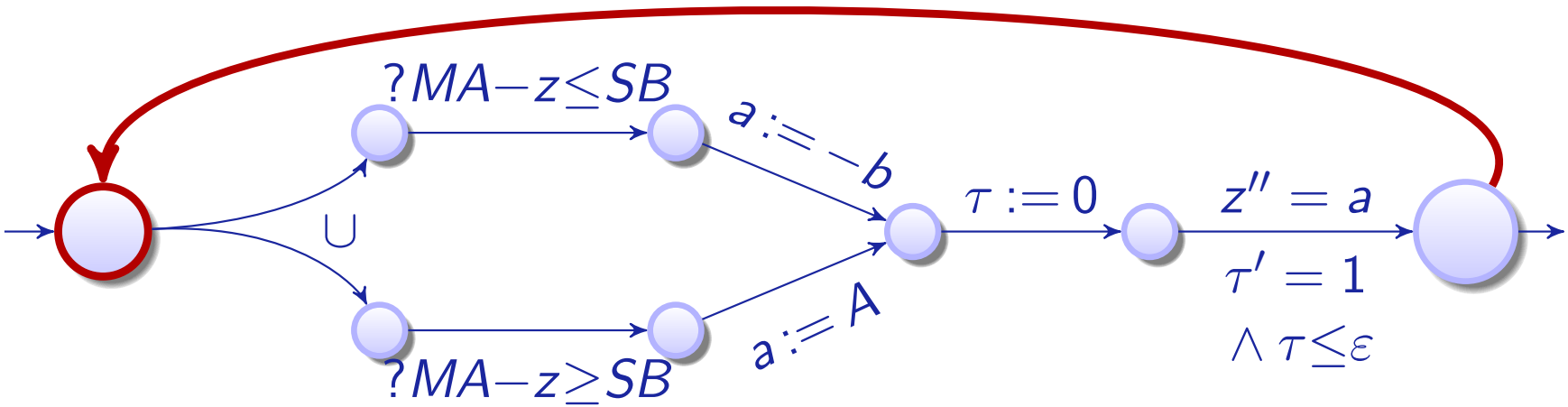
$$ETCS \equiv (ctrl\,;drive)^*$$

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

$$drive \equiv \tau := 0; (z' = v, v' = a, \tau' = 1 \wedge v \geq 0 \wedge \tau \leq \varepsilon)$$



Branching Transitions in Hybrid Programs



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

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

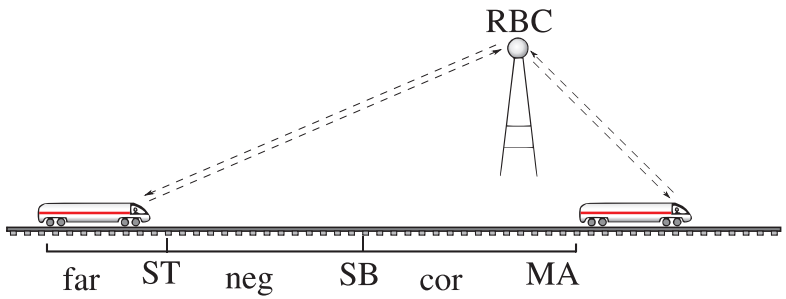
$$ETCS \equiv (ctrl\,;drive)^*$$

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

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

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

$$\wedge\, v \geq 0 \wedge \tau \leq \varepsilon)$$

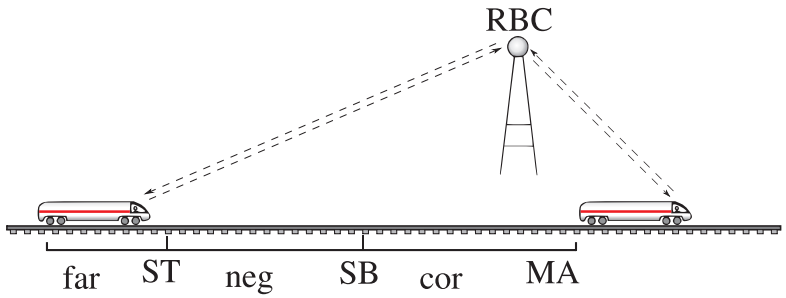


Differential Dynamic Logic d$\mathcal{L}$: Syntax

Definition (Formulas ϕ)

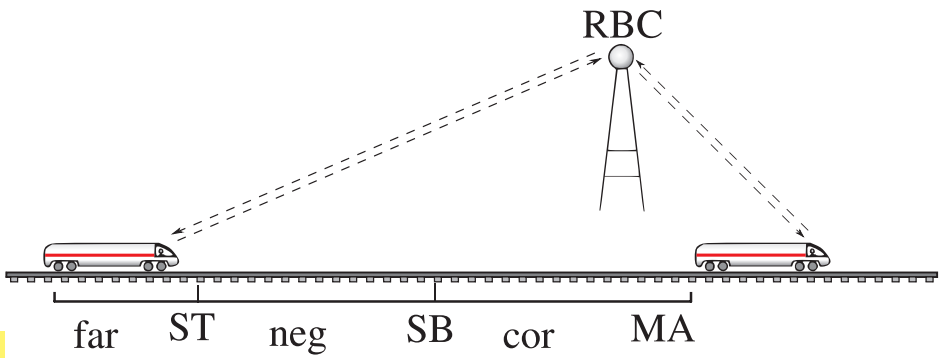
$\neg, \wedge, \vee, \rightarrow,\ \forall x\,, \exists x\,,\ \ =, \leq,\ +, \cdot$ ($\mathbb{R}$-first-order part)
$[\alpha]\phi,\quad \langle\alpha\rangle\phi$ (dynamic part)

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



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

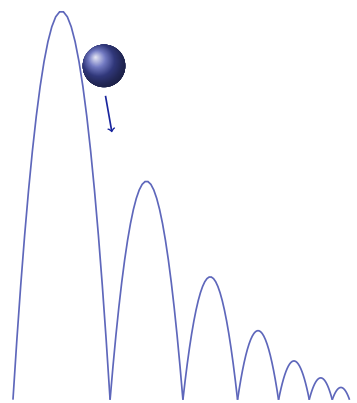
```
/* initial state characterization */
(v^2 <= 2*b*(m-z) & b>0 & A>=0) ->
\[(
   SB := (v^2)/(2*b) + ((A/b)+1) * ((A/2)*ep^2+ep*v);
   ((?m-z <= SB; a:= -b) ++ (?m-z >= SB; a:=A));
   t:=0;
   {z'=v, v'=a, t'=1, (v >= 0&t <= ep)}   /* drive */
  )*                                      /* repeat these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



ETCS Train Control [simple nondet]

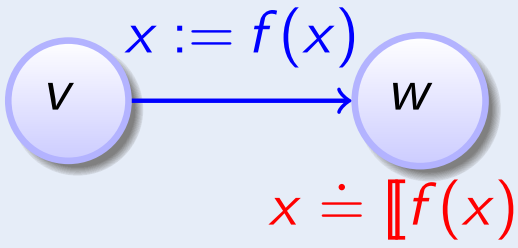
Differential Dynamic Logic dℒ: Bouncing Ball

```
/* initial state characterization */
g>0 & h>=0&t>=0 & v^2<=2*g*(H-h) & H>=0 ->
\[(
    {h'=v,v'=-g,t'=1, h>=0}; /* falling/jumping */
    if (t>0 & h=0) then      /* if on ground */
        c := *; ?0<=c & c<1; /* choose damping */
        v := -c*v;           /* bounce back */
        t := 0               /* reset clock */
    fi
)*                           /* repeat these transitio
\] (0<=h & h<=H)             /* safety/postcondition *
```

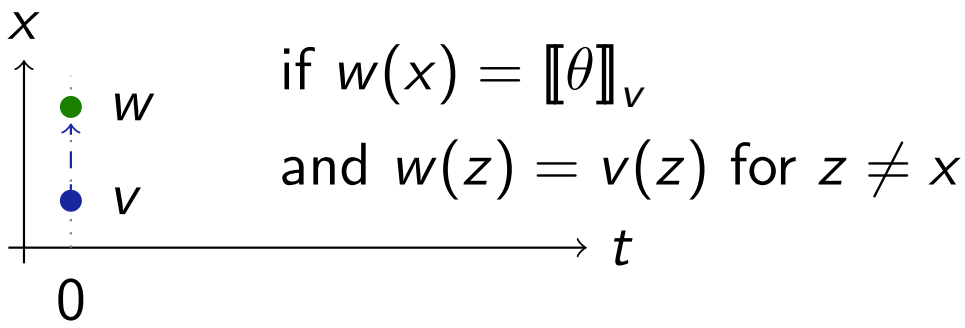


Bouncing Ball [dynamic]

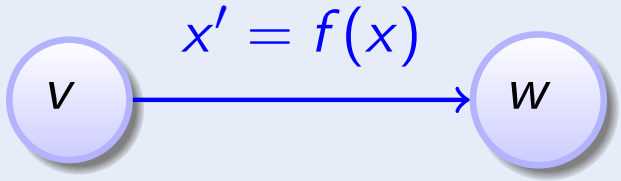
Definition (Hybrid programs α: transition semantics)



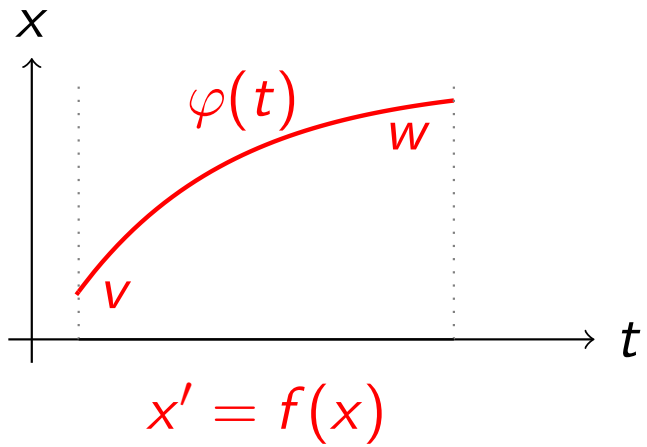
▸ Details ▸▸



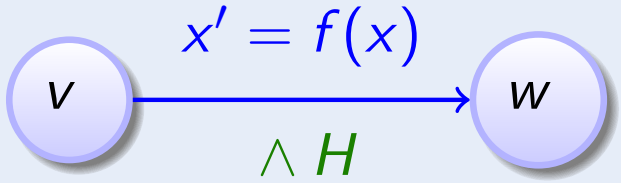
Definition (Hybrid programs α: transition semantics)



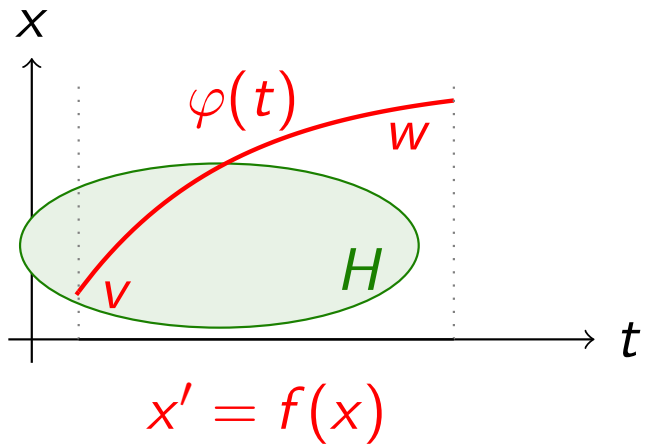
▸ Details ▸▸



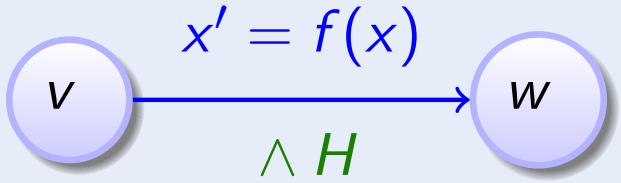
Definition (Hybrid programs α: transition semantics)



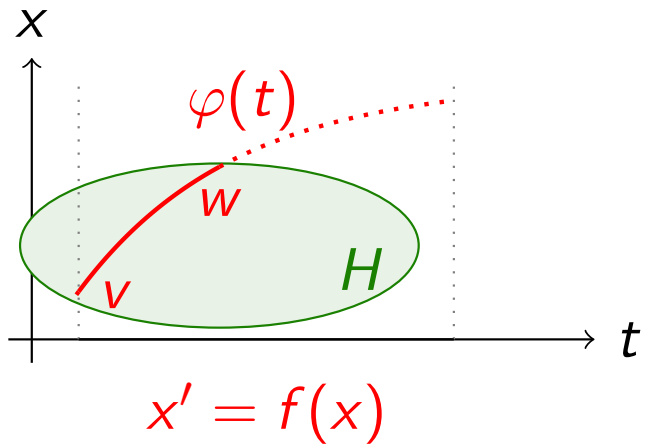
Details



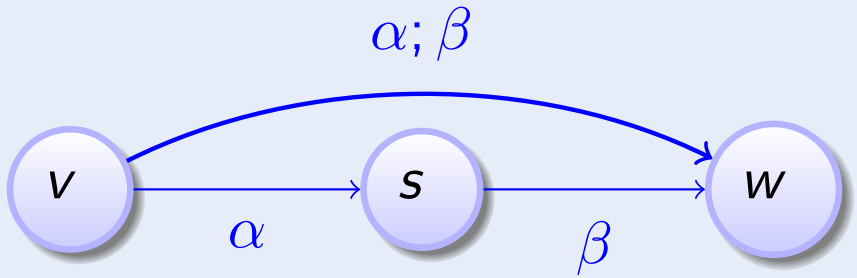
Definition (Hybrid programs α: transition semantics)



▸ Details ▸▸

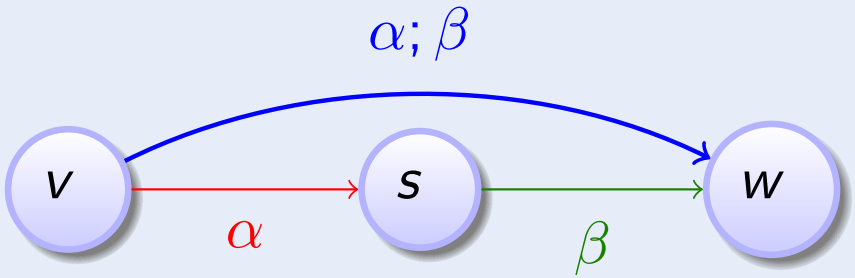


Definition (Hybrid programs α: transition semantics)

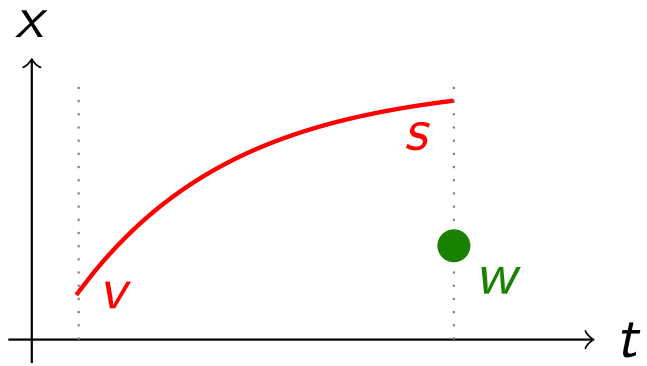


▸ Details ▸▸

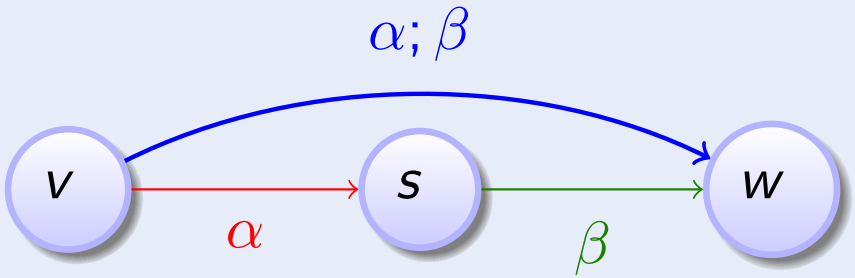
Definition (Hybrid programs α: transition semantics)



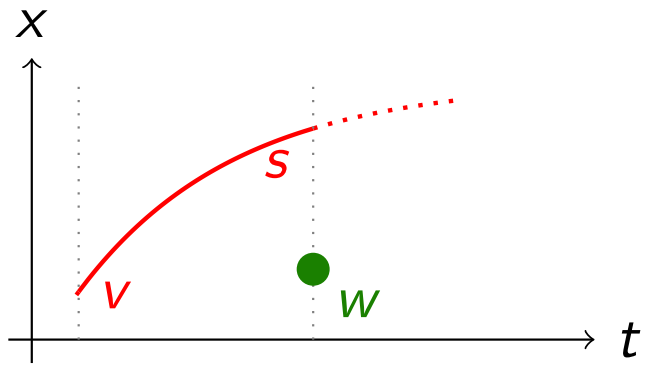
▸ Details ▸▸



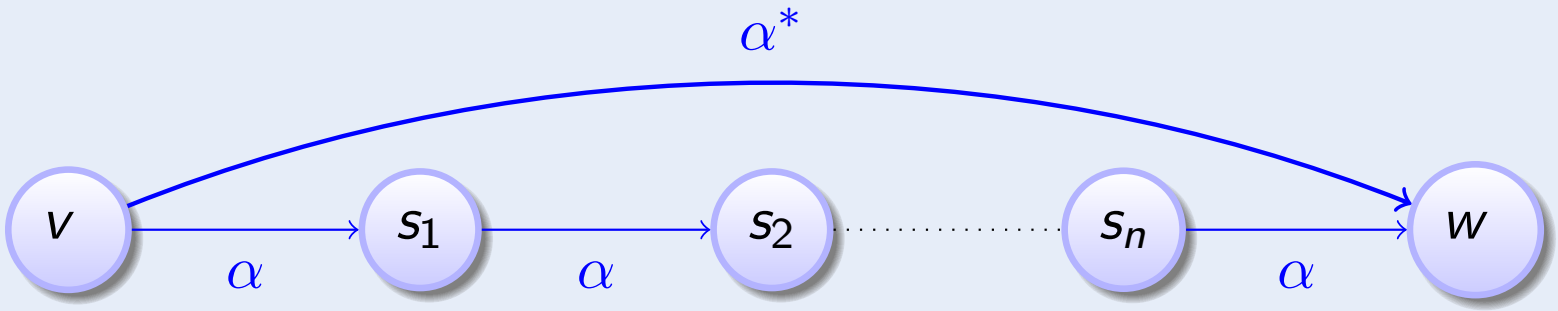
Definition (Hybrid programs α: transition semantics)



▸ Details ▸▸

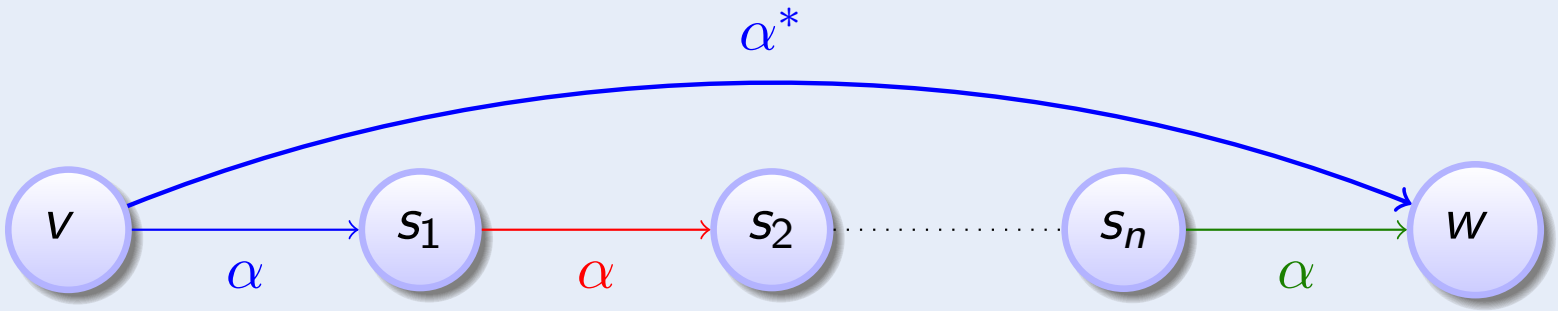


Definition (Hybrid programs α: transition semantics)

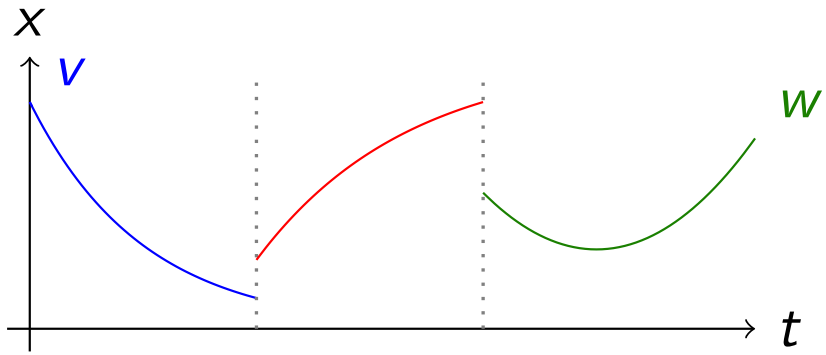


Details

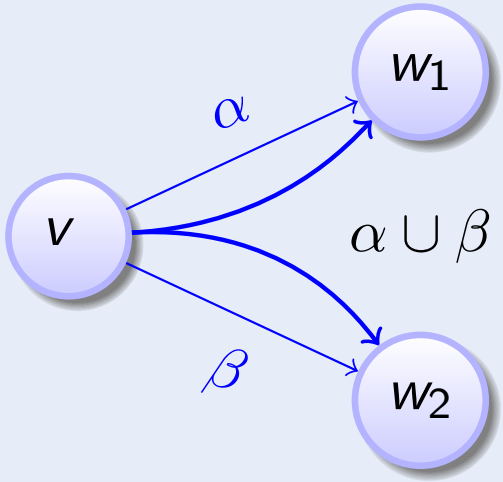
Definition (Hybrid programs α: transition semantics)



▸ Details ▸▸

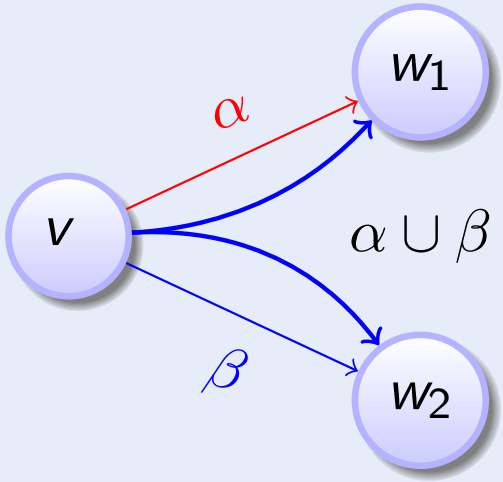


Definition (Hybrid programs α: transition semantics)

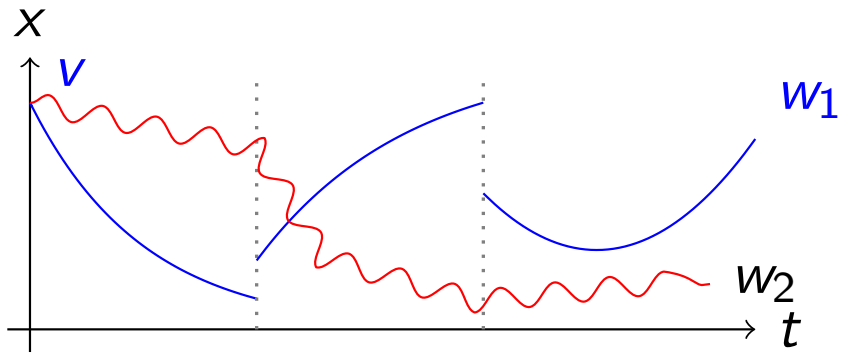


Details

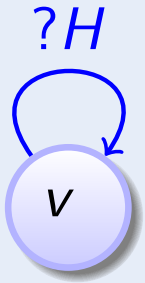
Definition (Hybrid programs α: transition semantics)



▸ Details ▸▸

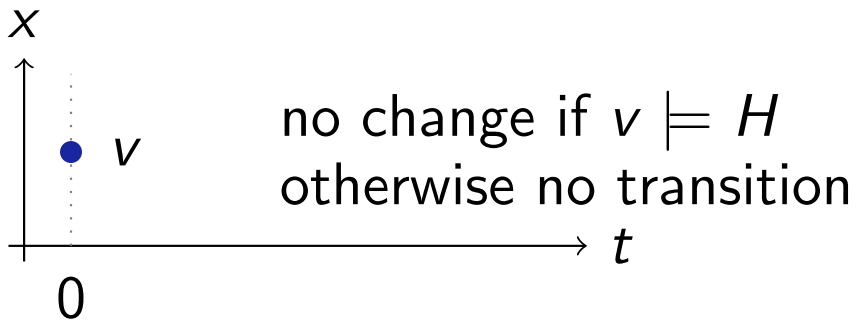


Definition (Hybrid programs α: transition semantics)



if $v \models H$

Details

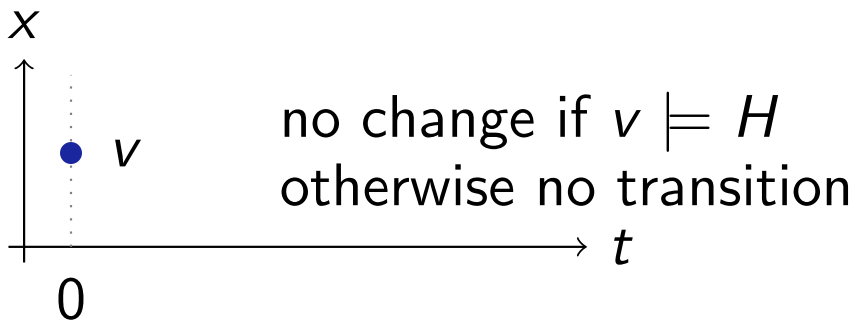


Definition (Hybrid programs α: transition semantics)

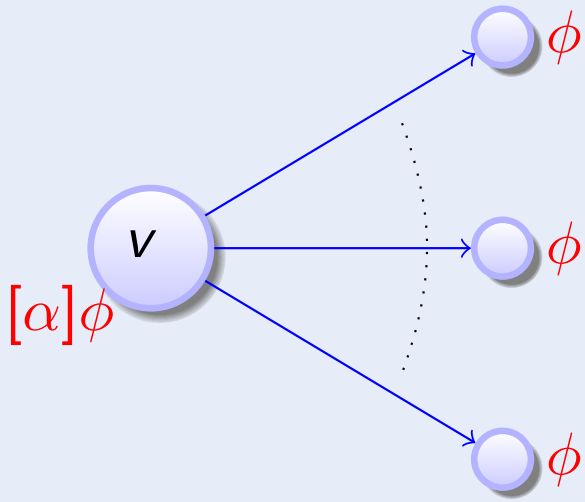
v

if $v \not\models H$

Details

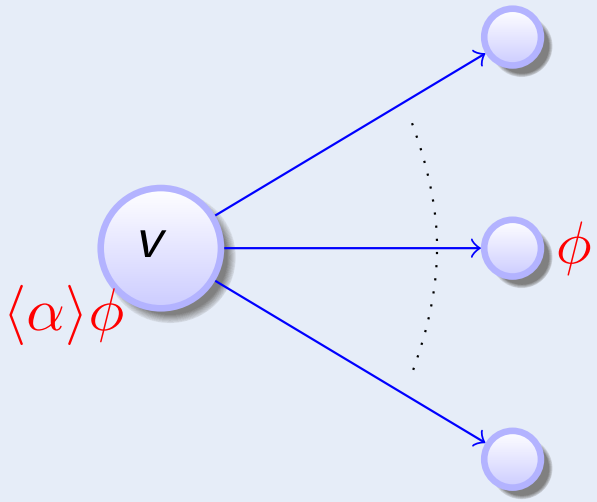


Definition (Formulas ϕ)



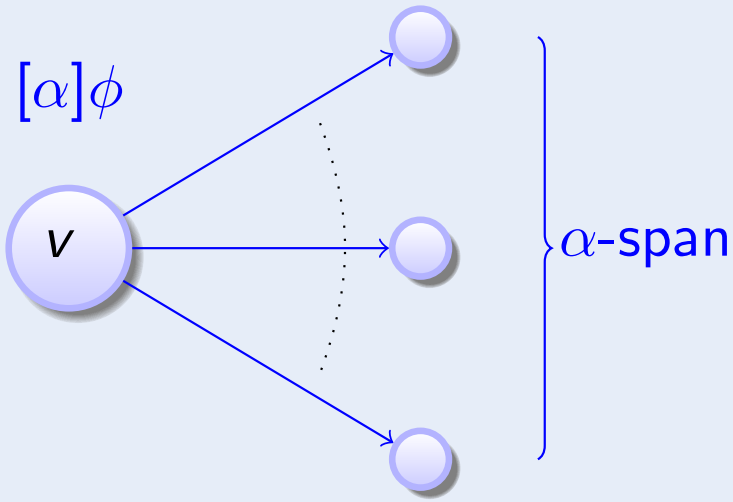
▸ Details ▸▸

Definition (Formulas ϕ)



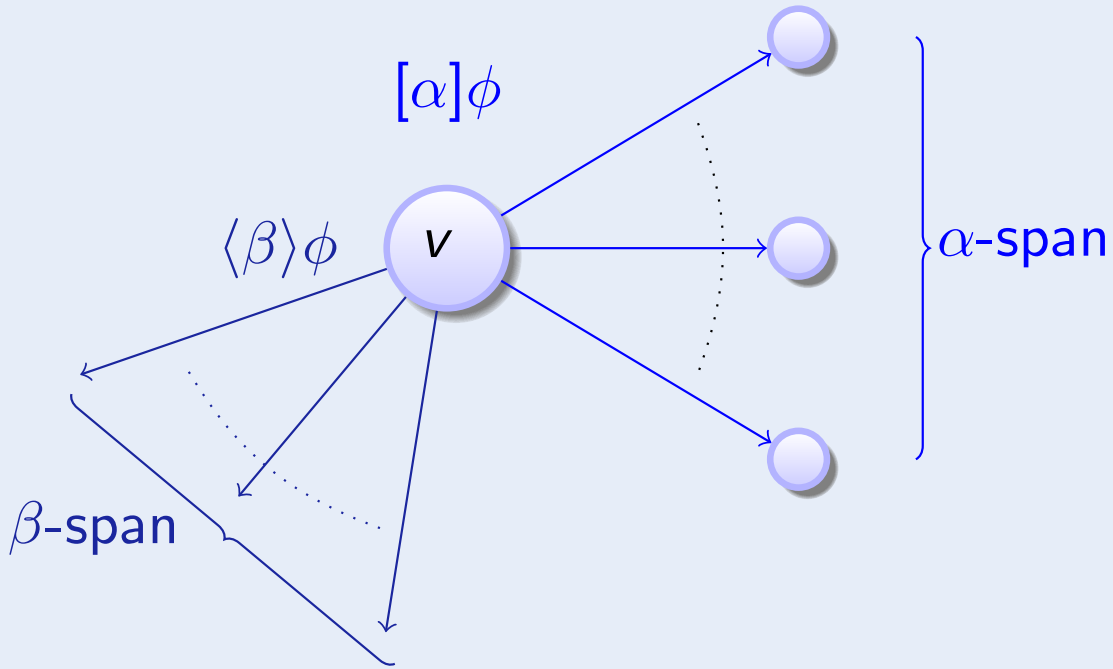
Details

Definition (Formulas ϕ)



▸ Details ▸▸

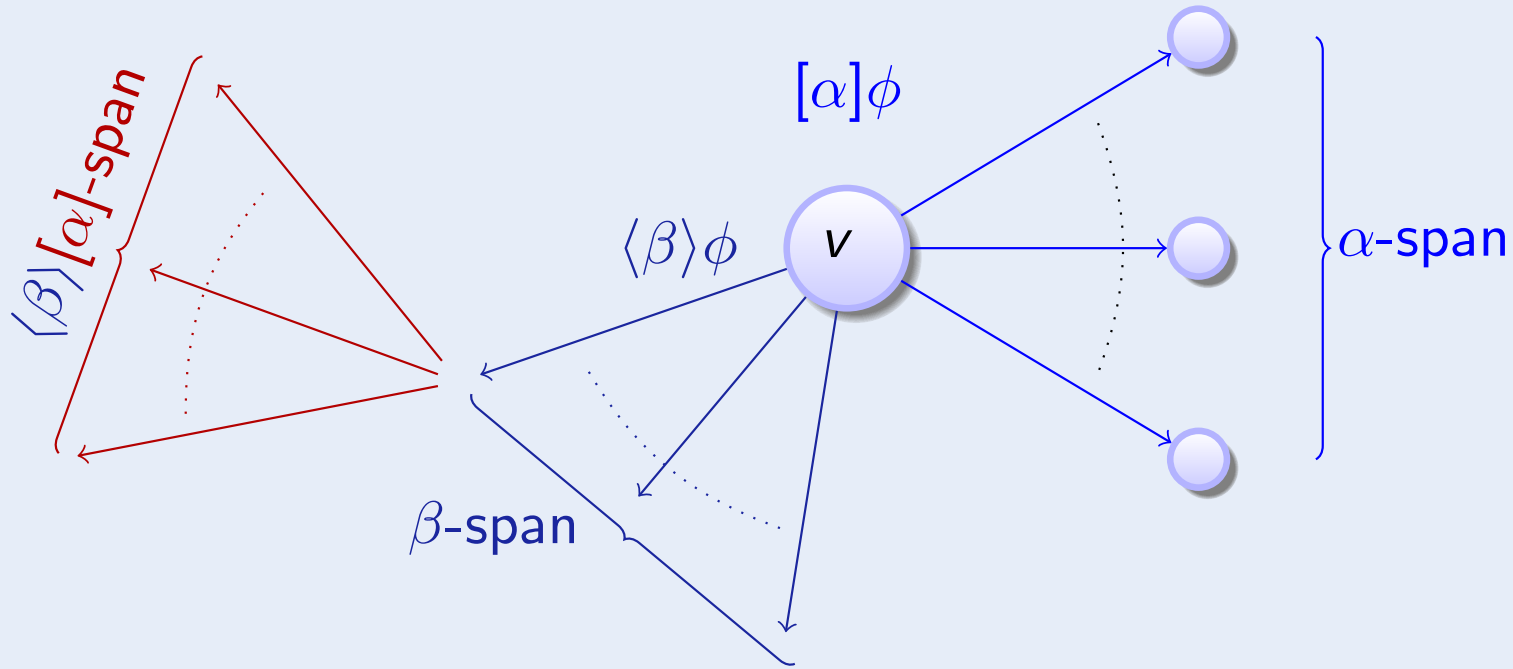
Definition (Formulas ϕ)



▸ Details ▸▸

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

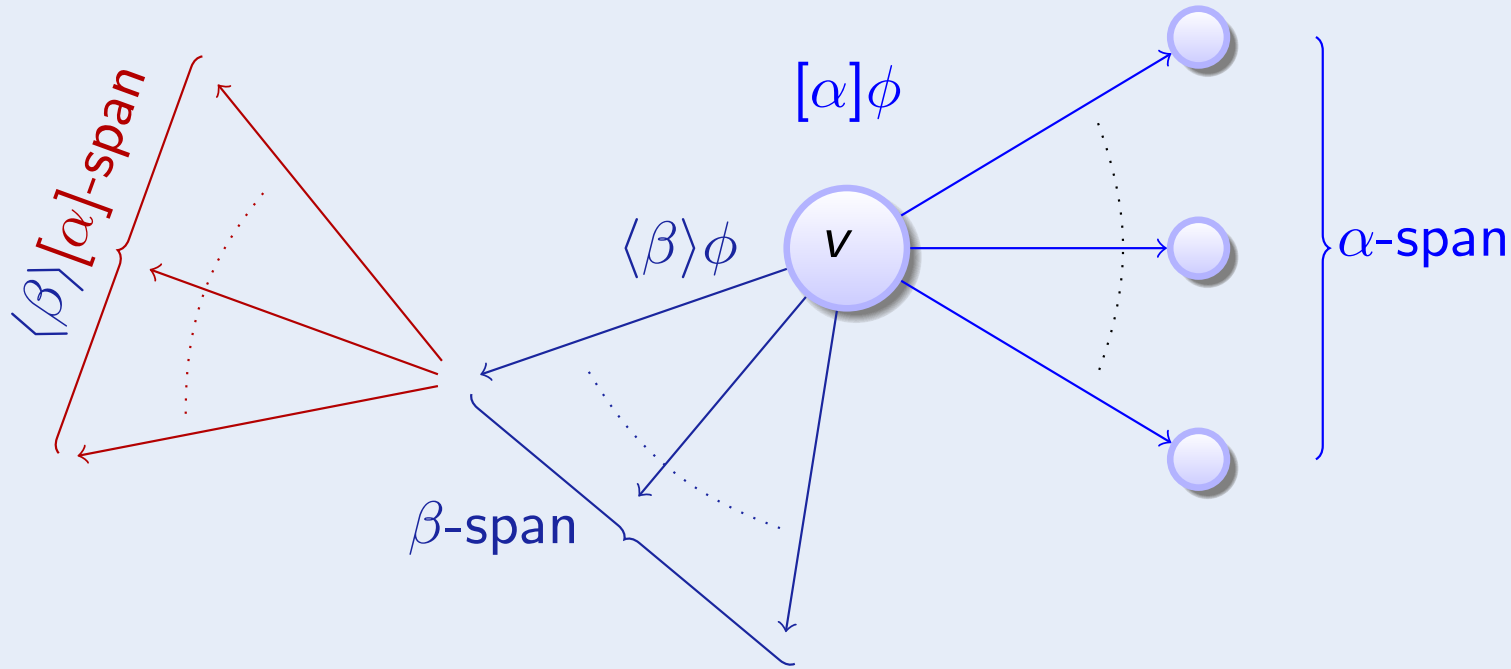
Definition (Formulas ϕ)



▸ Details ▸▸

Differential Dynamic Logic dℒ: Semantics

Definition (Formulas ϕ)



▸ Details ▸▸

compositional semantics $\Rightarrow$ compositional proofs!

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

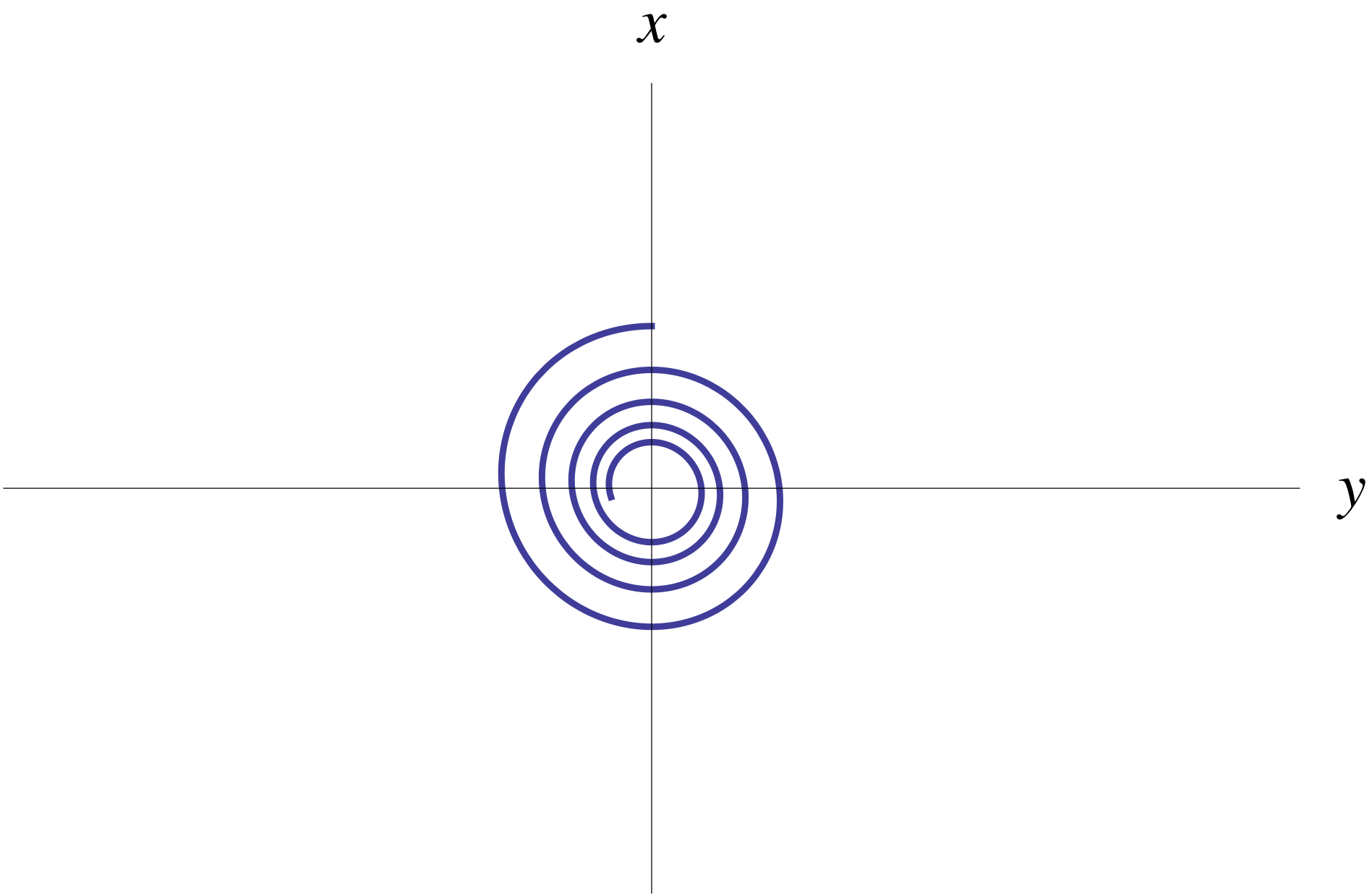
Definition (Formulas ϕ)

$$
\begin{array}{lcl}
v \models \theta_1 \geq \theta_2 & :\Longleftrightarrow & [\![\theta_1]\!]_v \geq [\![\theta_2]\!]_v \\
v \models \phi \wedge \psi & :\Longleftrightarrow & v \models \phi \text{ and } v \models \psi \\
v \models \neg\phi & :\Longleftrightarrow & v \models \phi \text{ does not hold} \\
v \models \forall x\, \phi & :\Longleftrightarrow & w \models \phi \text{ for all } w \text{ that agree with } v \\
 & & \quad \text{except for the value of } x \\
v \models \exists x\, \phi & :\Longleftrightarrow & w \models \phi \text{ for some } w \text{ that agrees with } v \\
 & & \quad \text{except for the value of } x \\
v \models [\alpha]\phi & :\Longleftrightarrow & w \models \phi \quad \text{for all } w \text{ with } (v,w) \in \rho(\alpha) \\
v \models \langle\alpha\rangle\phi & :\Longleftrightarrow & w \models \phi \quad \text{for some } w \text{ with } (v,w) \in \rho(\alpha)
\end{array}
$$

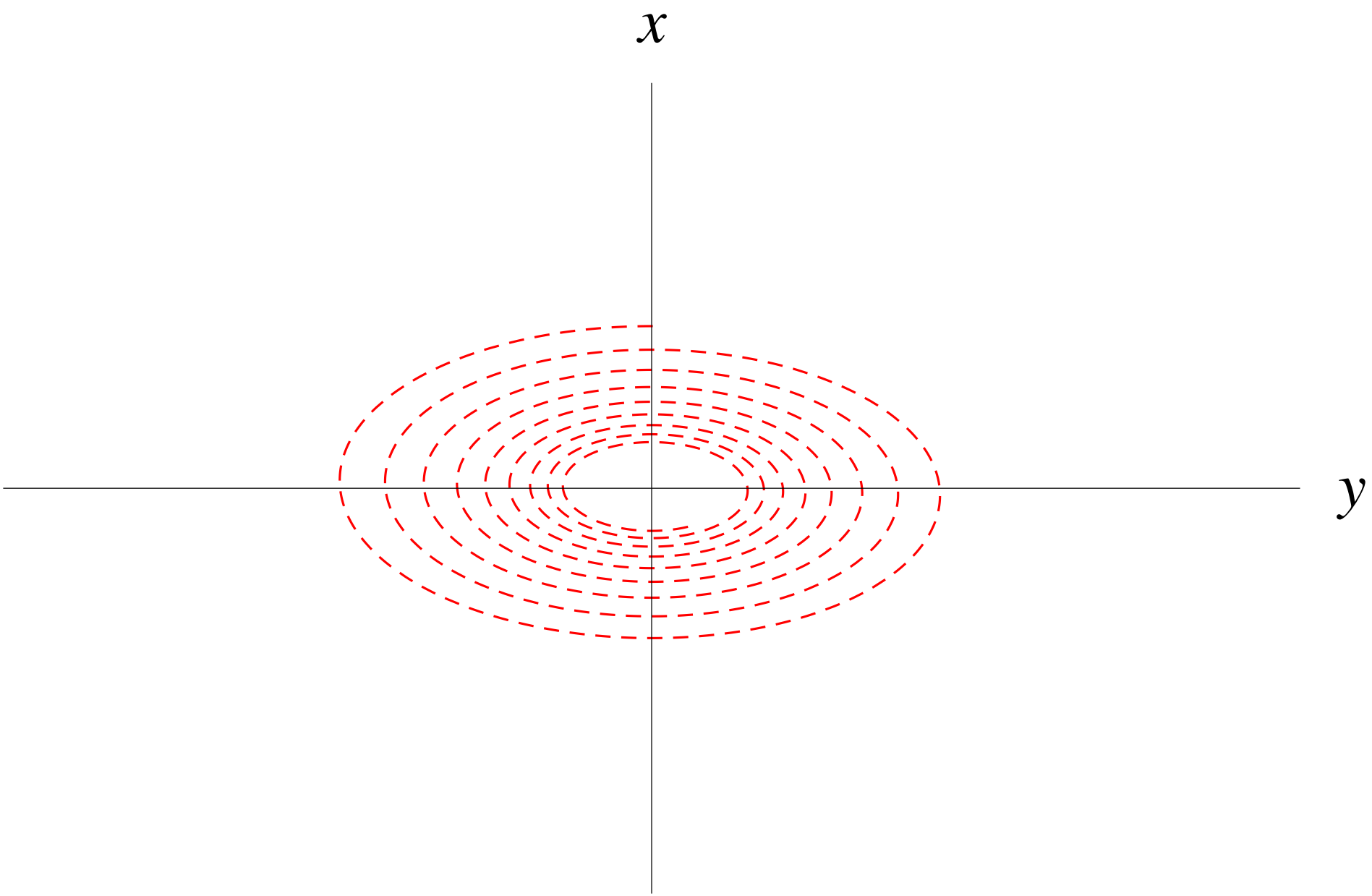
Nested Modalities and Nested Quantifiers

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

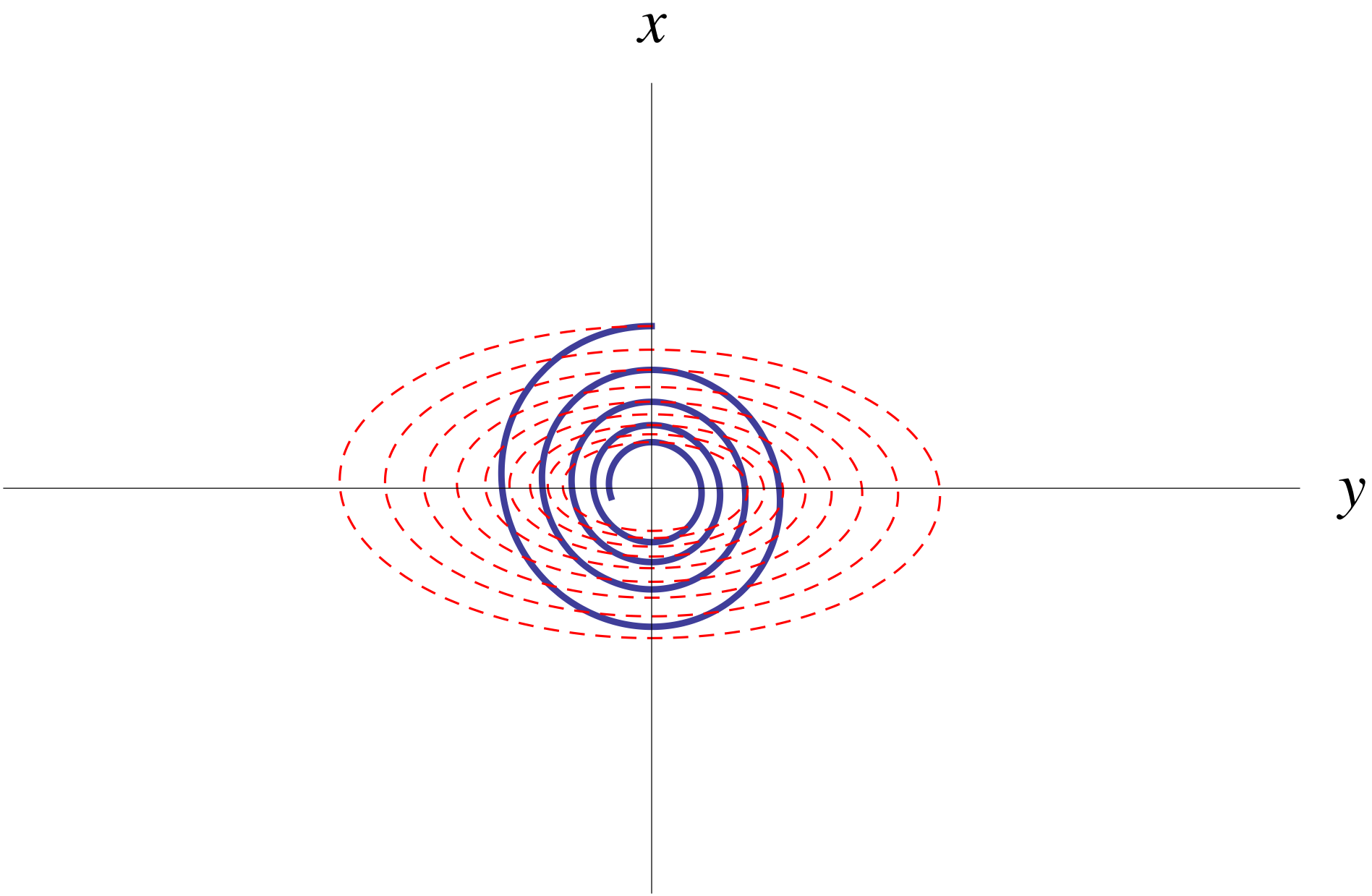
Safe Switching in Hybrid Systems



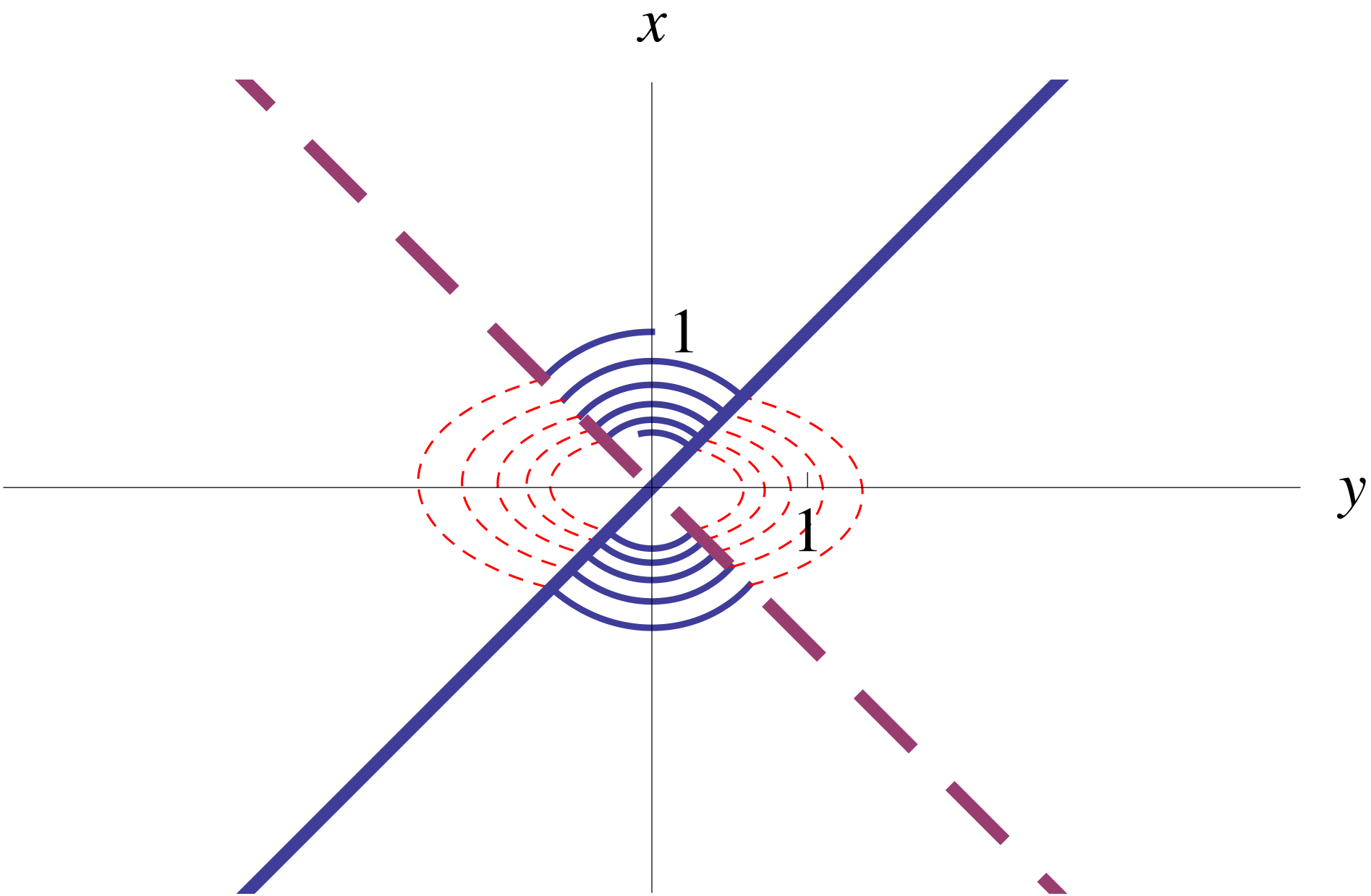
Safe Switching in Hybrid Systems



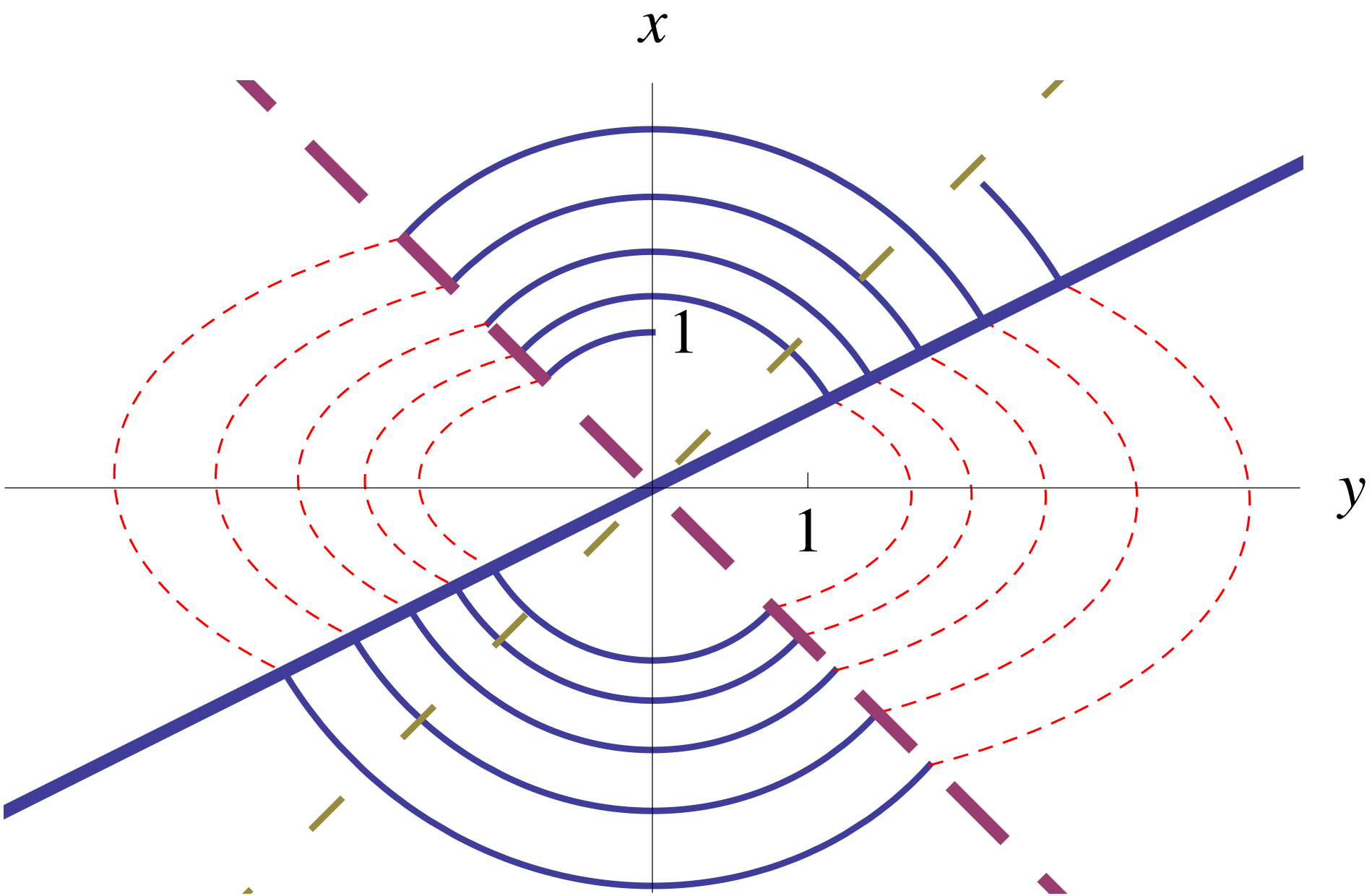
Safe Switching in Hybrid Systems



Safe Switching in Hybrid Systems

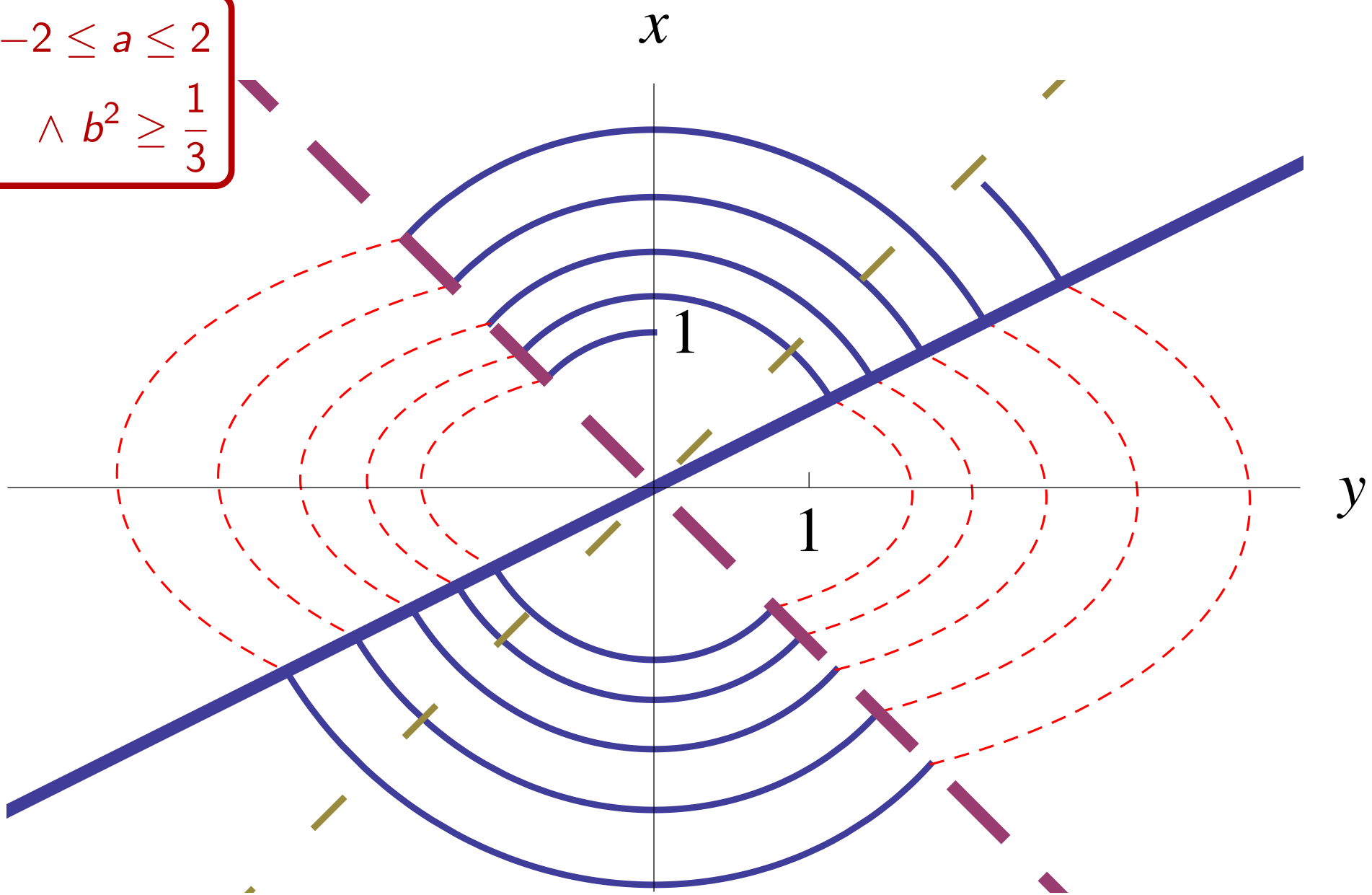


Safe Switching in Hybrid Systems



Safe Switching in Hybrid Systems

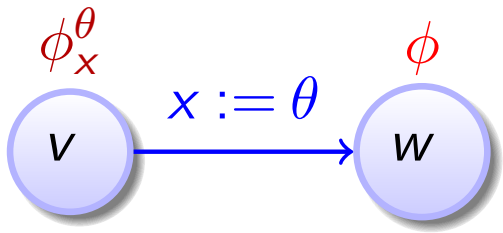
$$-2 \leq a \leq 2 \wedge b^2 \geq \frac{1}{3}$$



Outline (Verification Approach)

Proof Calculus for Differential Dynamic Logic

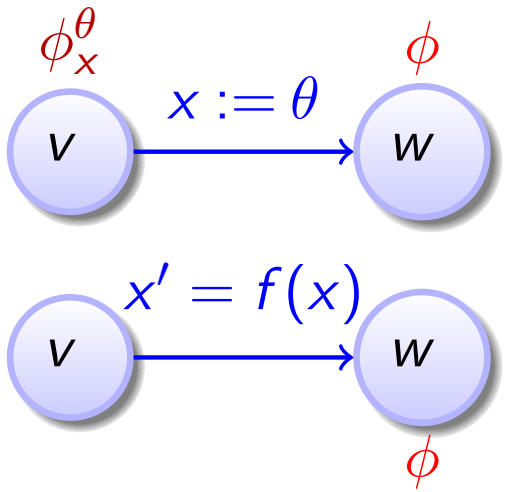
$$\frac{\phi_x^\theta}{\langle x := \theta \rangle \phi}$$



Proof Calculus for Differential Dynamic Logic

$$\frac{\phi_x^\theta}{\langle x := \theta \rangle \phi}$$

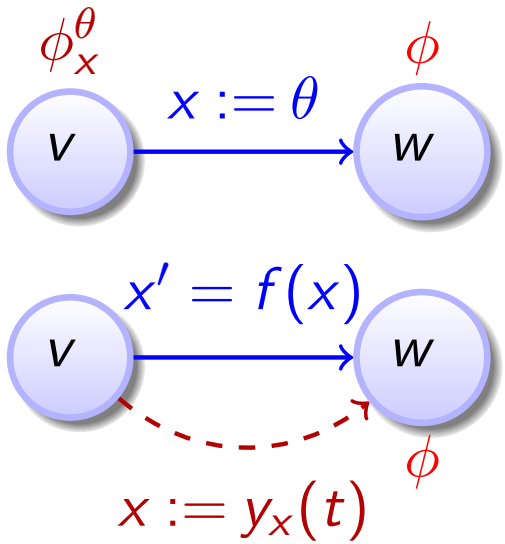
$$\frac{\exists t \geq 0 \, \langle x := y_x(t) \rangle \phi}{\langle x' = f(x) \rangle \phi}$$



Proof Calculus for Differential Dynamic Logic

$$\frac{\phi_x^\theta}{\langle x := \theta \rangle \phi}$$

$$\frac{\exists t \geq 0 \, \langle x := y_x(t) \rangle \phi}{\langle x' = f(x) \rangle \phi}$$

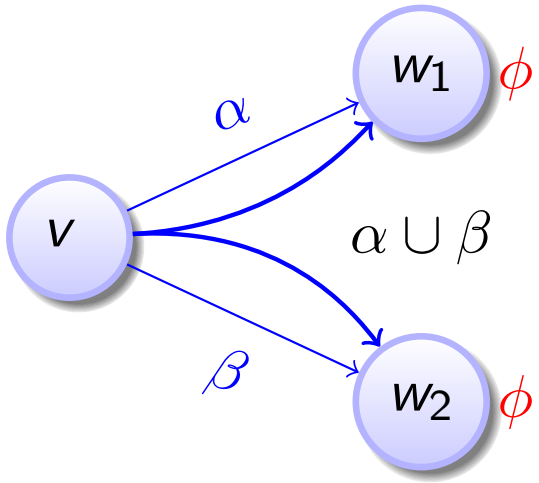


Proof Calculus for Differential Dynamic Logic

compositional semantics $\Rightarrow$ compositional rules!

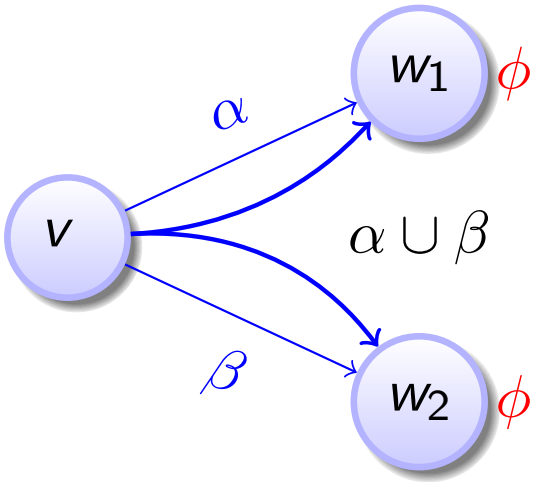
Proof Calculus for Differential Dynamic Logic

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

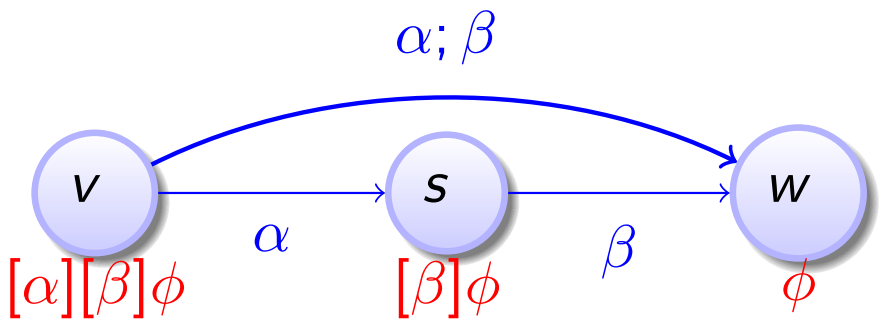


Proof Calculus for Differential Dynamic Logic

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

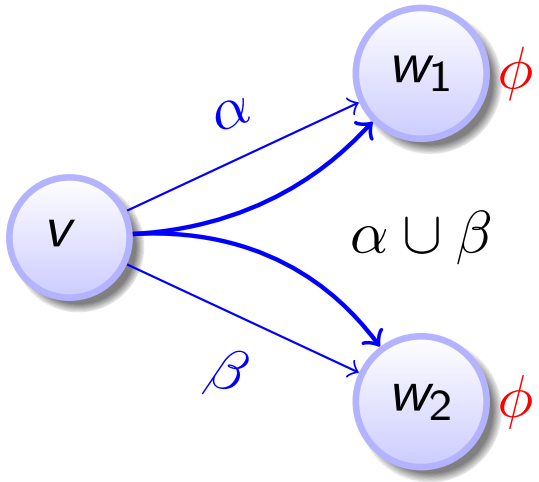


$$\frac{[\alpha][\beta]\phi}{[\alpha;\beta]\phi}$$

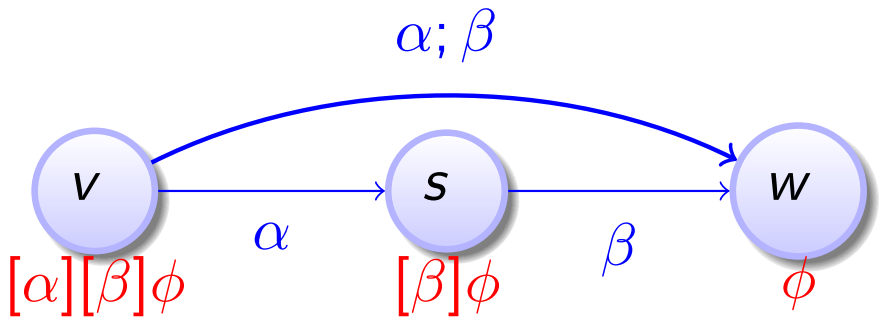


Proof Calculus for Differential Dynamic Logic

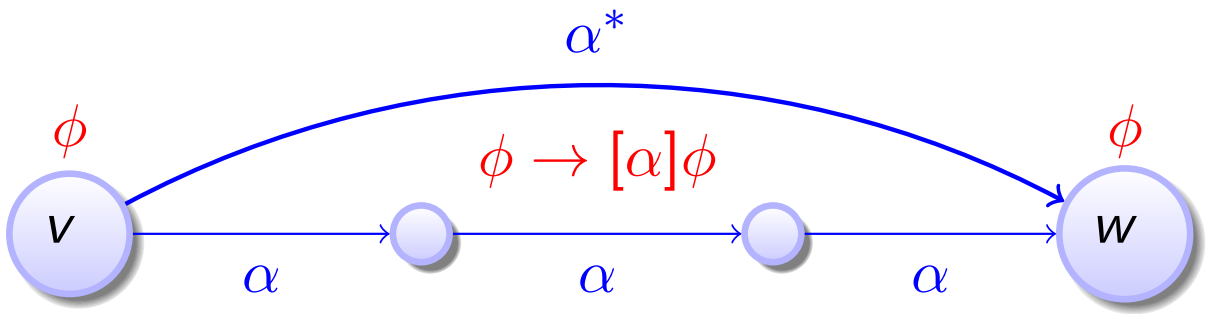
$$\frac{[\alpha]\phi \wedge [\beta]\phi}{[\alpha \cup \beta]\phi}$$



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

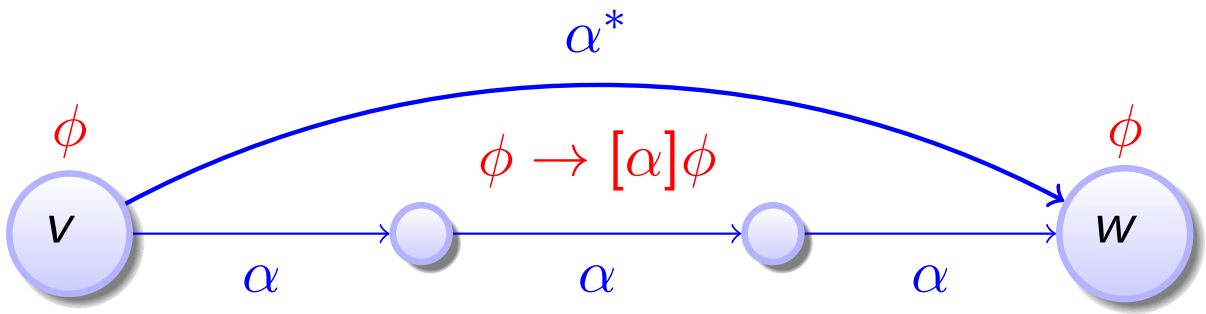


$$\frac{\vdash \phi \quad \vdash (\phi \rightarrow [\alpha]\phi)}{\vdash [\alpha^*]\phi}$$



Proof Calculus: Differential Invariants in a Nutshell

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

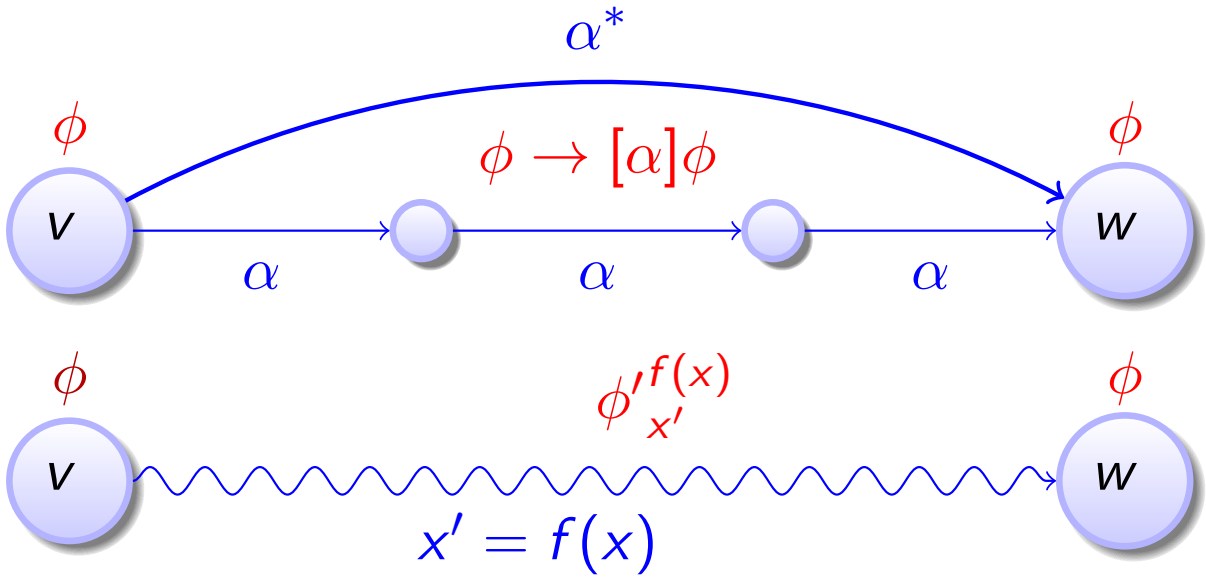


André Platzer.
Differential-algebraic dynamic logic for differential-algebraic programs.
J. Log. Comput., 35(1): 309–352, 2010.

Proof Calculus: Differential Invariants in a Nutshell

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

$$\frac{\vdash \phi \quad \vdash (H \to \phi'^{f(x)}_{x'})}{\vdash [x' = f(x) \wedge H]\phi}$$

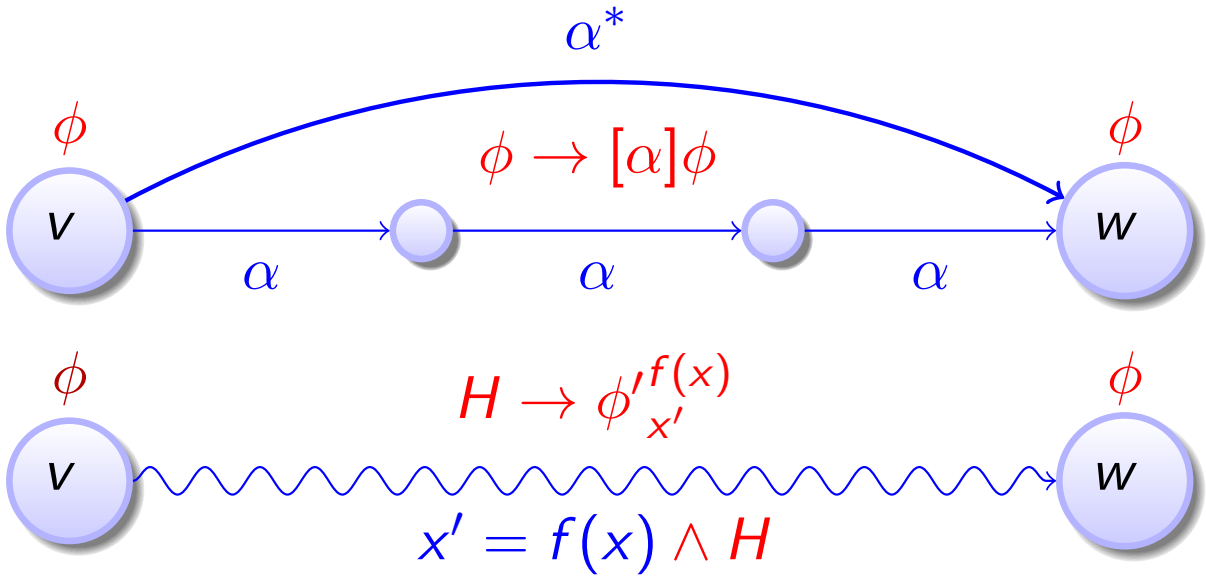


André Platzer.
Differential-algebraic dynamic logic for differential-algebraic programs.
J. Log. Comput., 35(1): 309–352, 2010.

Proof Calculus: Differential Invariants in a Nutshell

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

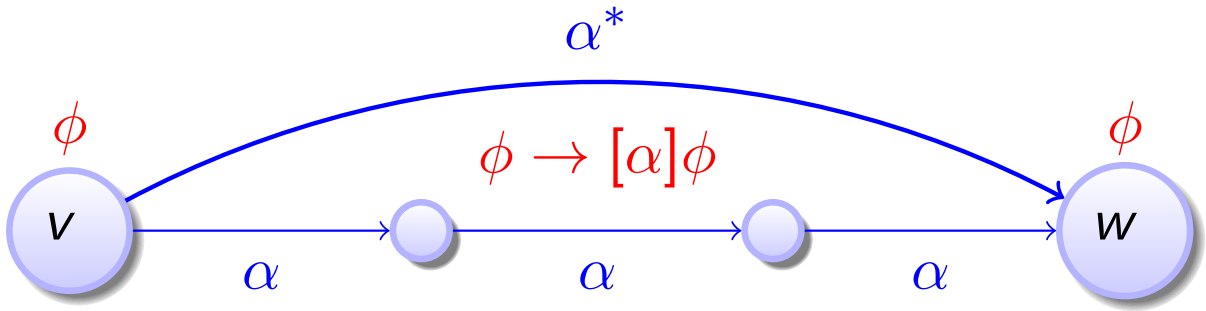
$$\frac{\vdash \phi \quad \vdash (H \to \phi'^{f(x)}_{x'})}{\vdash [x' = f(x) \wedge H]\phi}$$



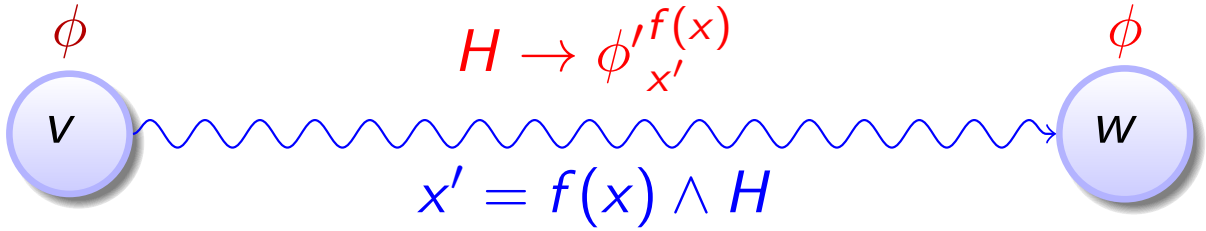
André Platzer.
Differential-algebraic dynamic logic for differential-algebraic programs.
J. Log. Comput., 35(1): 309–352, 2010.

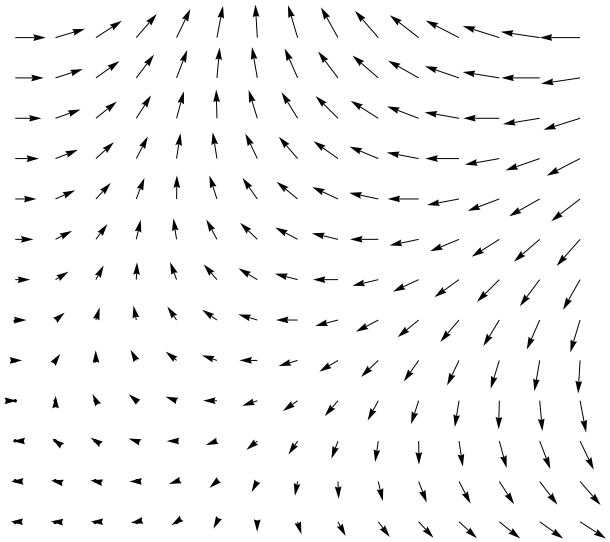
Proof Calculus: Differential Invariants in a Nutshell

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

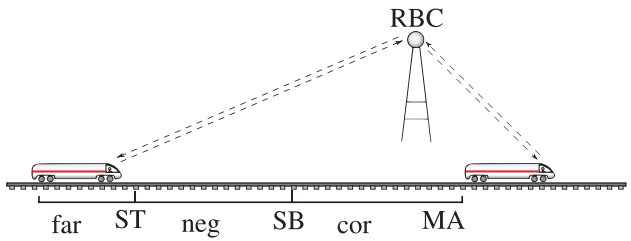


$$\frac{\vdash \phi \quad \vdash (H \to \phi'^{f(x)}_{x'})}{\vdash [x' = f(x) \wedge H]\phi}$$



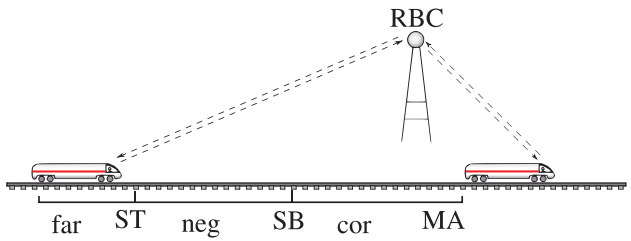


Deduction Modulo Real Arithmetic

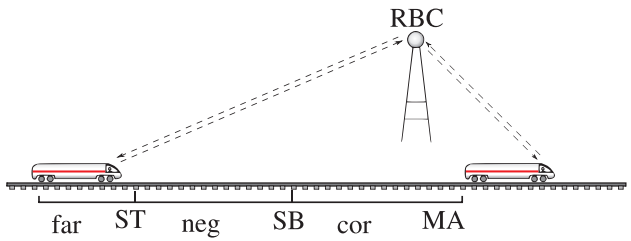


$$\vdash v \geq 0 \wedge z < MA \rightarrow \langle z' = v, v' = -b \rangle\ z > MA$$

Deduction Modulo Real Arithmetic



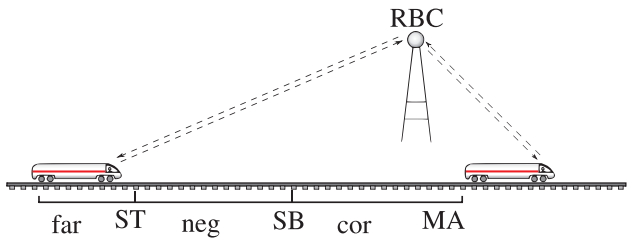
$$\frac{\dfrac{v \geq 0, z < MA \vdash \exists t{\geq}0\, \langle z := -\frac{b}{2}t^2 + vt + z\rangle z > MA}{v \geq 0, z < MA \vdash \langle z' = v, v' = -b\rangle z > MA}}{\vdash v \geq 0 \wedge z < MA \rightarrow \langle z' = v, v' = -b\rangle\ z > MA}$$



Collins/Tarski QE not applicable!

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

Deduction Modulo (Side Deduction)

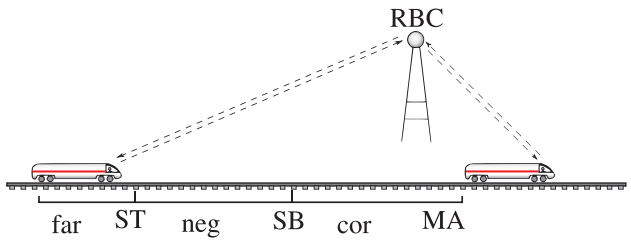


$$\frac{}{v \geq 0, z < MA \vdash t \geq 0 \wedge \langle z := -\frac{b}{2}t^2 + vt + z \rangle z > MA}$$

start side

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

Deduction Modulo (Side Deduction)

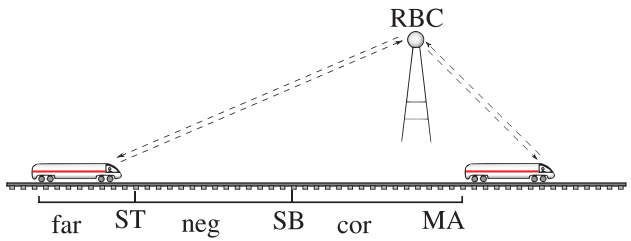


$$\dfrac{v \geq 0, z < MA \vdash t \geq 0 \qquad \dfrac{v \geq 0, z < MA \vdash -\frac{b}{2}t^2 + vt + z > MA}{v \geq 0, z < MA \vdash \langle z := -\frac{b}{2}t^2 + vt + z \rangle z > MA}}{v \geq 0, z < MA \vdash t \geq 0 \wedge \langle z := -\frac{b}{2}t^2 + vt + z \rangle z > MA}$$

start side

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

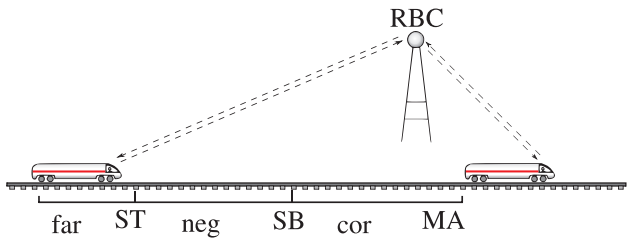
Deduction Modulo (Side Deduction)



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

start side

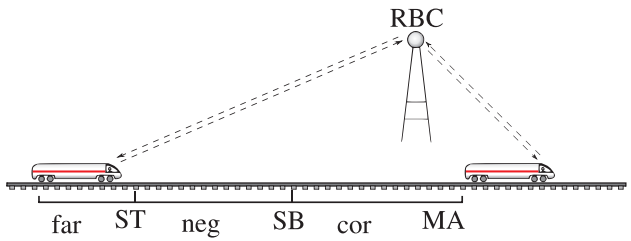
Deduction Modulo (Side Deduction)



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

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

Deduction Modulo (Free Variables for Automation)

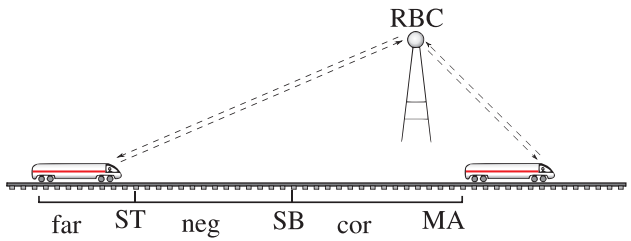


$$\frac{}{v \geq 0, z < MA \vdash \exists t{\geq}0\, \langle z := -\frac{b}{2}t^2 + vt + z\rangle z{>}MA}$$

$$\frac{v \geq 0, z < MA \vdash \langle z' = v, v' = -b\rangle z > MA}{}$$

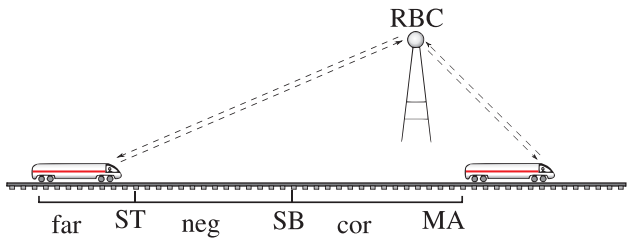
$$\vdash v \geq 0 \wedge z < MA \rightarrow \langle z' = v, v' = -b\rangle z > MA$$

Deduction Modulo (Free Variables for Automation)



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

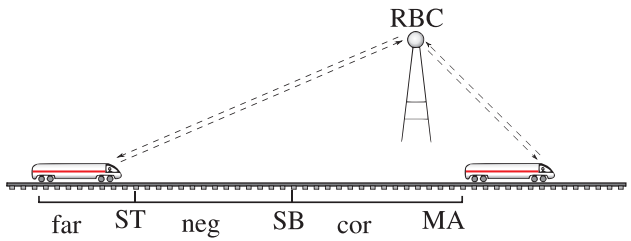
Deduction Modulo (Free Variables for Automation)



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

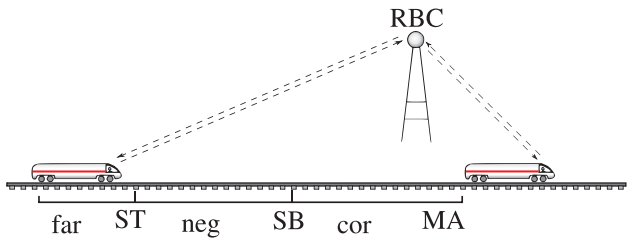
$$v \geq 0, z < MA \vdash \exists T \, (\ldots T \geq 0 \wedge -\tfrac{b}{2}T^2 + vT + z > MA)$$

Deduction Modulo (Free Variables for Automation)



$$\dfrac{\dfrac{v \geq 0, z < MA \vdash \mathrm{QE}\big(\exists T\,(\ldots T \geq 0 \wedge -\frac{b}{2}T^2 + vT + z > MA)\big)}{\dfrac{v \geq 0, z < MA \vdash -\frac{b}{2}T^2 + vT + z > MA}{v \geq 0, z < MA \vdash \langle z := -\frac{b}{2}T^2 + vT + z\rangle z > MA}}}{\dfrac{v \geq 0, z < MA \vdash T \geq 0 \wedge \langle z := -\frac{b}{2}T^2 + vT + z\rangle z > MA}{\dfrac{v \geq 0, z < MA \vdash \exists t \geq 0\, \langle z := -\frac{b}{2}t^2 + vt + z\rangle z > MA}{\dfrac{v \geq 0, z < MA \vdash \langle z' = v, v' = -b\rangle z > MA}{\vdash v \geq 0 \wedge z < MA \rightarrow \langle z' = v, v' = -b\rangle z > MA}}}}$$

Deduction Modulo (Free Variables for Automation)



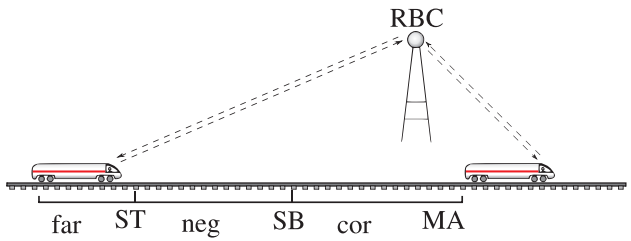
$$\dfrac{\dfrac{v \geq 0, z < MA \vdash v^2 > 2b(MA - z)}{\dfrac{v \geq 0, z < MA \vdash -\frac{b}{2}T^2 + vT + z > MA}{v \geq 0, z < MA \vdash \langle z := -\frac{b}{2}T^2 + vT + z \rangle z > MA}} \quad v \geq 0, z < MA \vdash T \geq 0}{v \geq 0, z < MA \vdash T \geq 0 \wedge \langle z := -\frac{b}{2}T^2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

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

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

Deduction Modulo (Free Variables for Automation)

- For requantification, not for unification



$$\dfrac{\dfrac{v \geq 0, z < MA \vdash \mathrm{QE}\big(\exists T\,(\ldots T \geq 0 \wedge -\frac{b}{2}T^2 + vT + z > MA)\big)}{\dfrac{v \geq 0, z < MA \vdash -\frac{b}{2}T^2 + vT + z > MA}{v \geq 0, z < MA \vdash \langle z := -\frac{b}{2}T^2 + vT + z\rangle z > MA}}}{\dfrac{v \geq 0, z < MA \vdash T \geq 0 \wedge \langle z := -\frac{b}{2}T^2 + vT + z\rangle z > MA}{\dfrac{v \geq 0, z < MA \vdash \exists t \geq 0\, \langle z := -\frac{b}{2}t^2 + vt + z\rangle z > MA}{\dfrac{v \geq 0, z < MA \vdash \langle z' = v, v' = -b\rangle z > MA}{\vdash v \geq 0 \wedge z < MA \rightarrow \langle z' = v, v' = -b\rangle z > MA}}}}$$

Deduction Modulo (Free Variables for Automation)

$$\vdash (X < S)$$

$$\vdash \forall s\,(X < s)$$

$$\vdash \exists x\,\forall s\,(x < s)$$

Deduction Modulo (Free Variables for Automation)

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

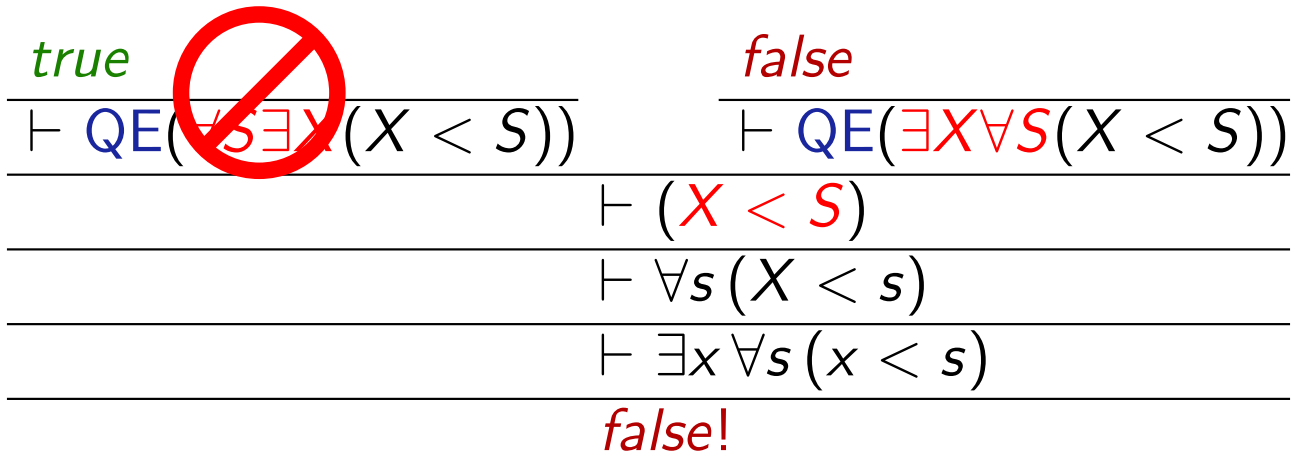
Deduction Modulo (Free Variables for Automation)

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

Deduction Modulo (Free Variables for Automation)

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

Deduction Modulo (Free Variables for Automation)



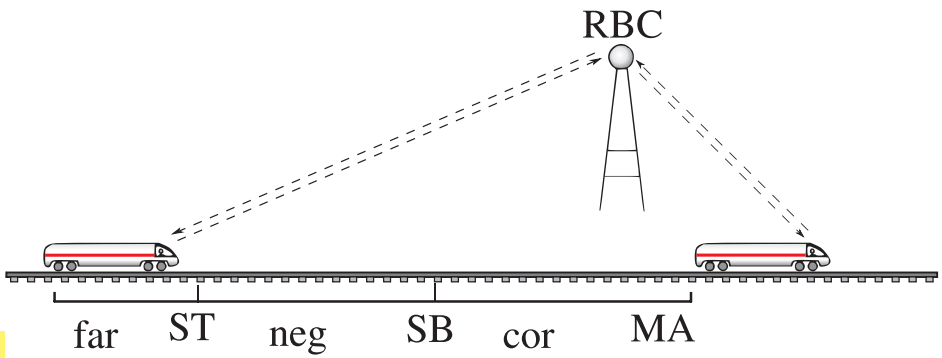
Skolemisation $S(X)$

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

$\textit{false}!$

Differential Dynamic Logic dℒ: Simple ETCS

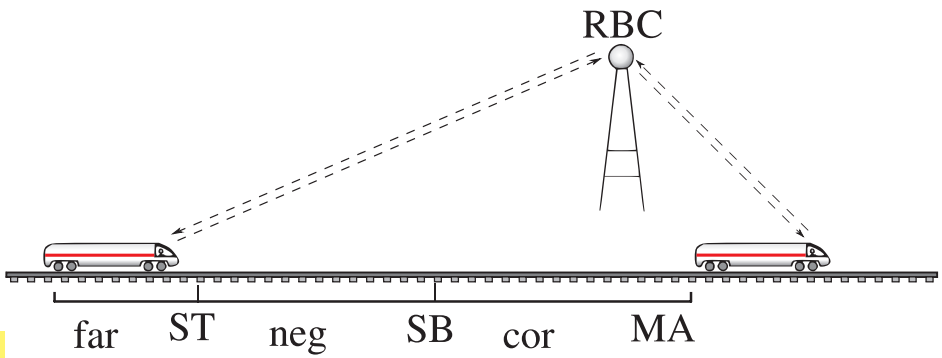
```
/* initial state characterization */
(v^2 <= 2*b*(m-z) & b>0 & A>=0) ->
\[(
   SB := (v^2)/(2*b) + ((A/b)+1) * ((A/2)*ep^2+ep*v);
   ((?m-z <= SB; a:= -b) ++ (?m-z >= SB; a:=A));
   t:=0;
   {z'=v, v'=a, t'=1, (v >= 0&t <= ep)}   /* drive */
  )*                                      /* repeat these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



ETCS Train Control [simple nondet]

Differential Dynamic Logic dℒ: Buggy Simple ETCS

```
/* 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]

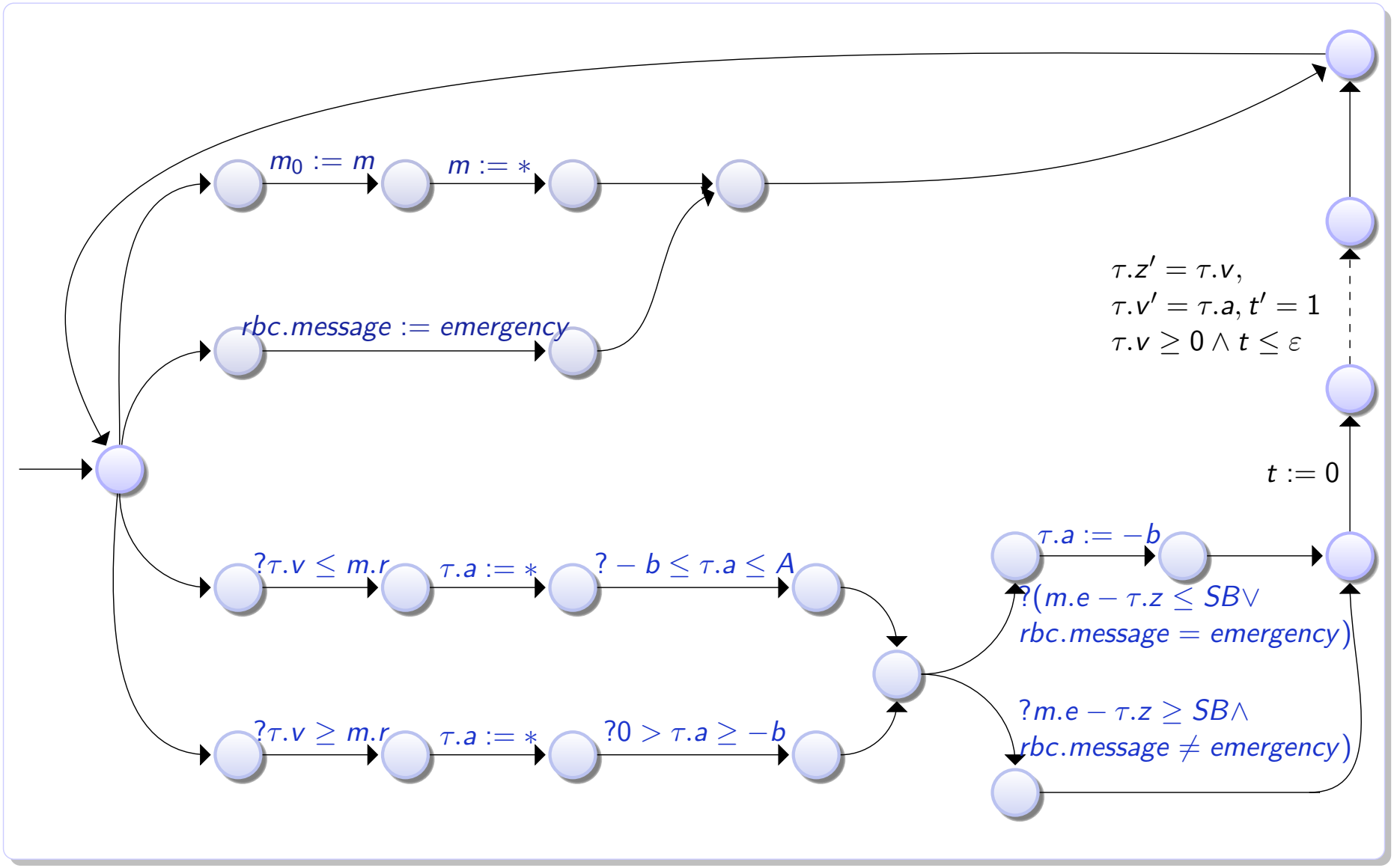
Parametric Skeleton of ETCS

Read from the informal specification...

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

As transition system...

ETCS Train Control [safety]



Soundness and Completeness

Theorem (Relative Completeness)

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

▸ Proof Outline 15p


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

Soundness and Completeness

Theorem (Relative Completeness)

dℒ calculus is a sound & complete axiomatisation of hybrid systems relative to differential equations.

▸ Proof Outline 15p

Corollary (Proof-theoretical Alignment)

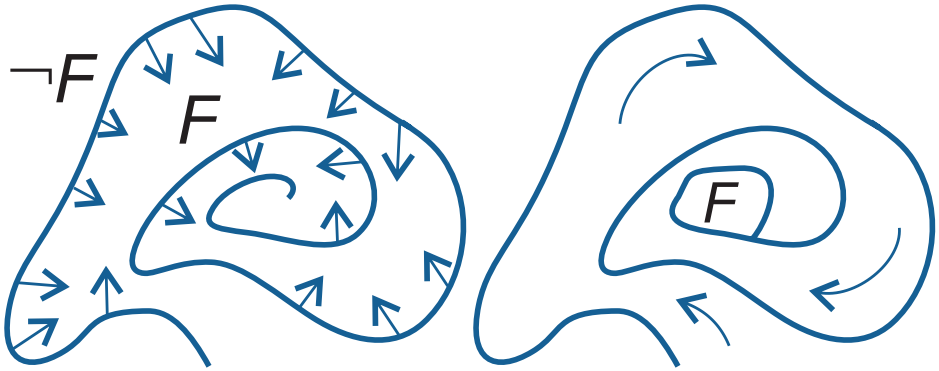
verification of hybrid systems = verification of dynamical systems!

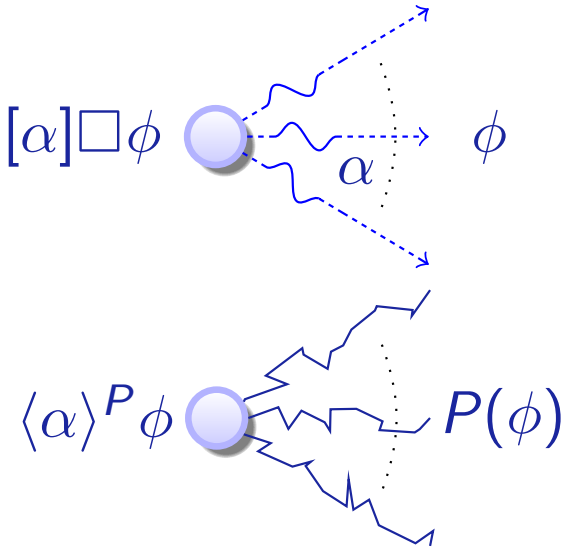
Corollary (Compositionality)

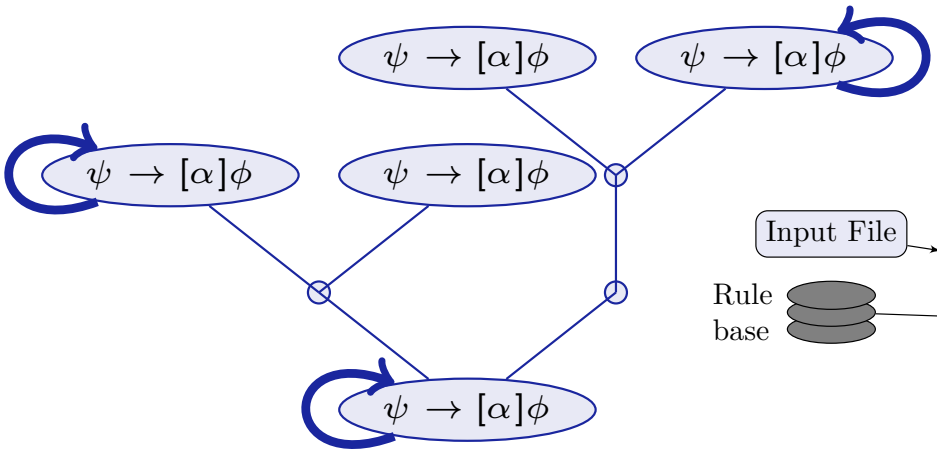
hybrid systems can be verified by recursive decomposition

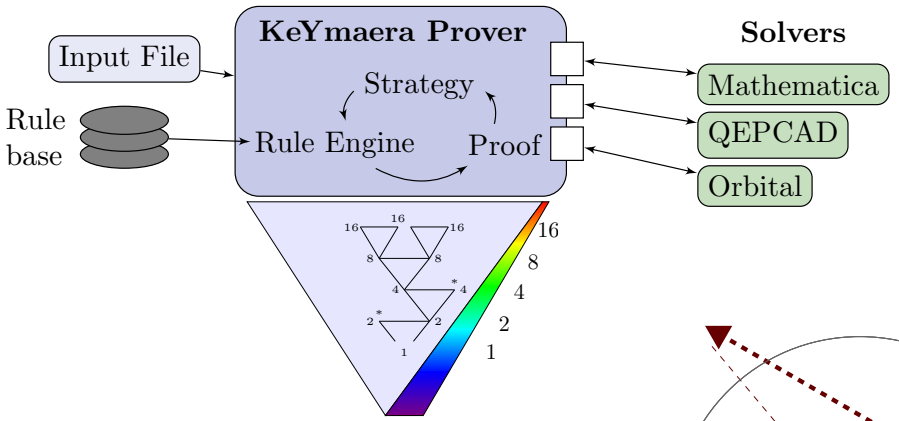

André Platzer.
Differential dynamic logic for hybrid systems.
J. Autom. Reas., 41(2):143–189, 2008.

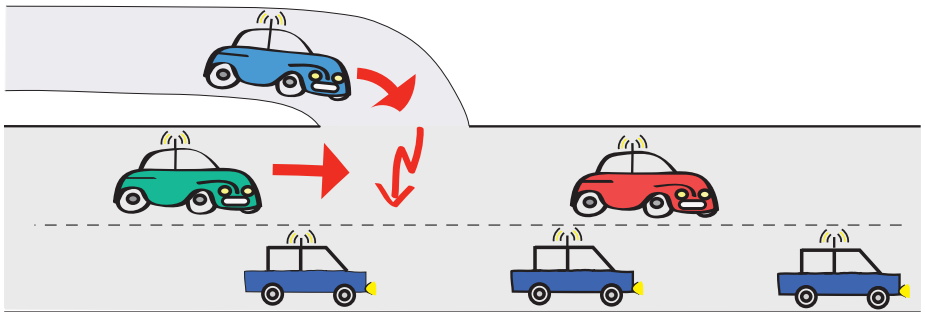
Outline

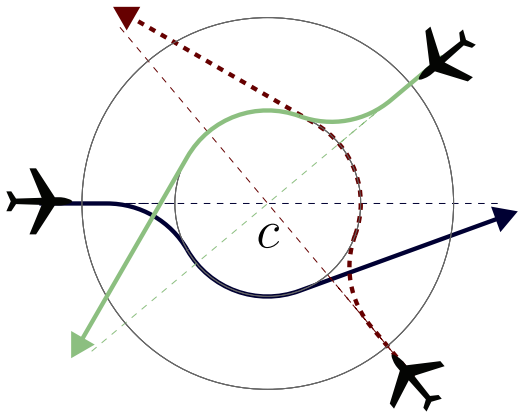

¬F
F
F


[α]□φ
α
φ
⟨α⟩^P φ
P(φ)


ψ → [α]φ
ψ → [α]φ
ψ → [α]φ
ψ → [α]φ
ψ → [α]φ


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



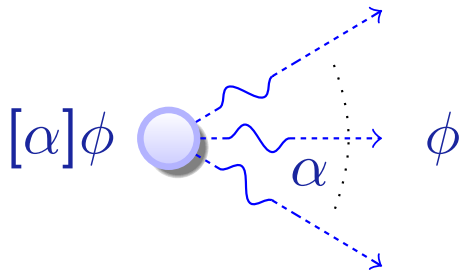

c

Outline

Summary

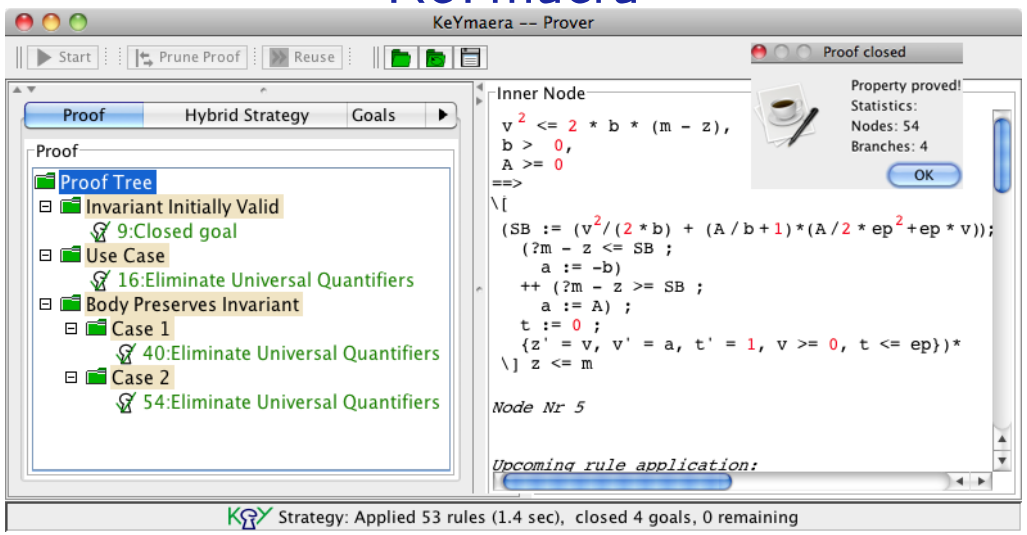
differential dynamic logic

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



- Logics for hybrid systems
- Compositional proofs
- Sound & complete / ODE
- Differential invariants
- Theory+practice+apps
- Distributed hybrid systems
- Stochastic hybrid systems

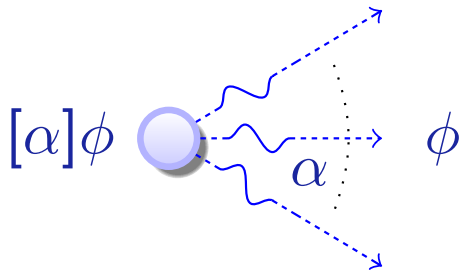
KeYmaera



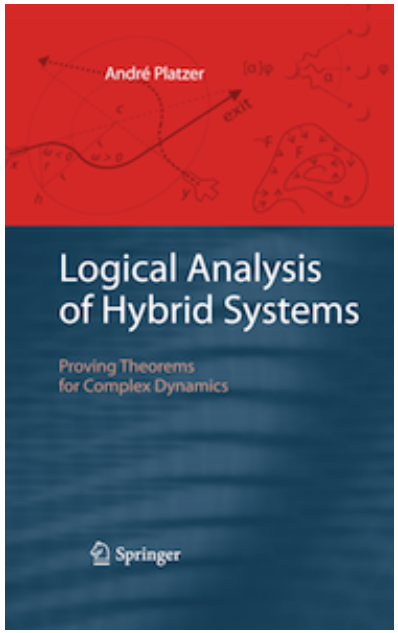
Summary

differential dynamic logic

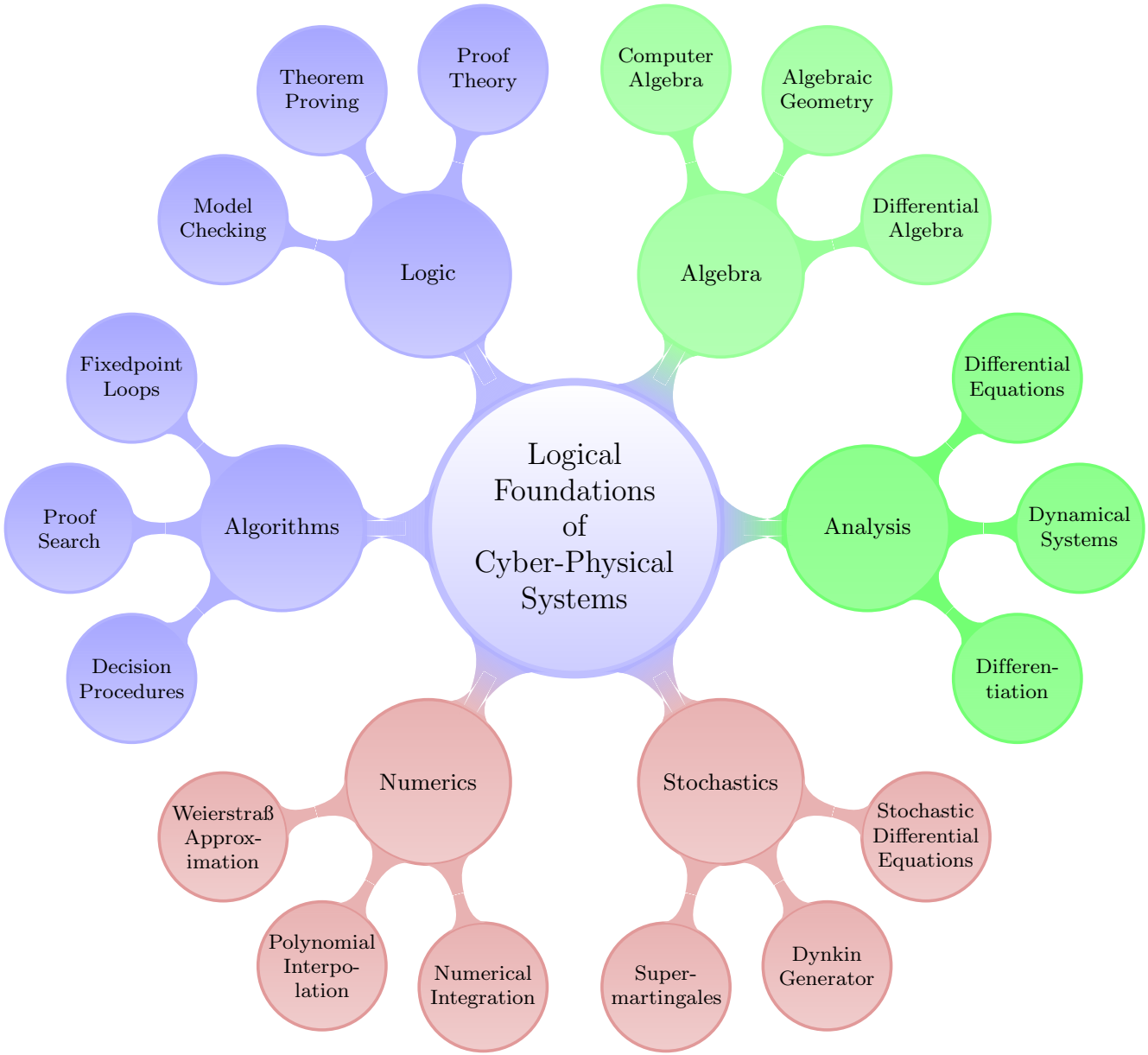
$$\mathsf{d}\mathcal{L} = \mathsf{DL} + \mathsf{HP}$$



- Logics for hybrid systems
- Compositional proofs
- Sound & complete / ODE
- Differential invariants
- Theory+practice+apps
- Distributed hybrid systems
- Stochastic hybrid systems



Logical Analysis of Hybrid Systems




André Platzer.
Logical Analysis of Hybrid Systems:
Proving Theorems for Complex Dynamics.
Springer, 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., 35(1): 309–352, 2010.


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


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.

Outline

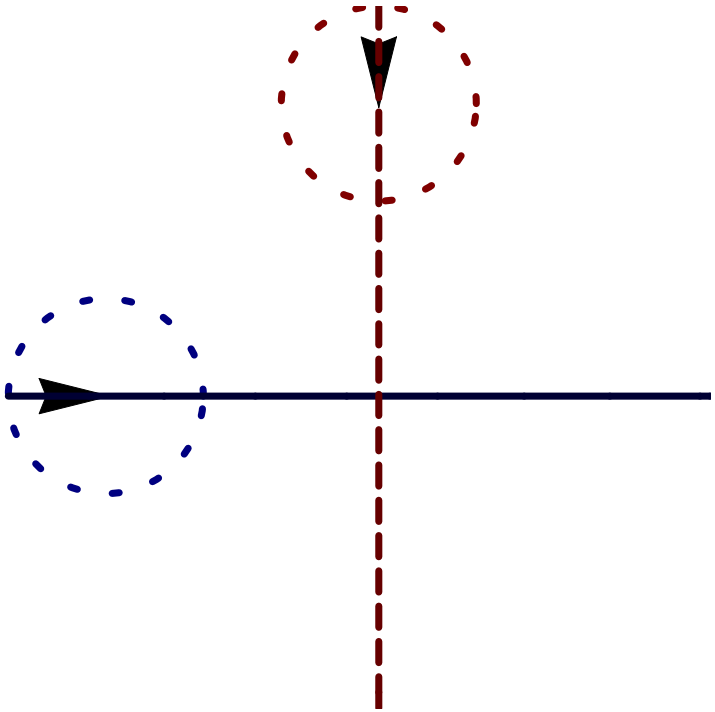
Outline

	Op	Par	T	CI	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.

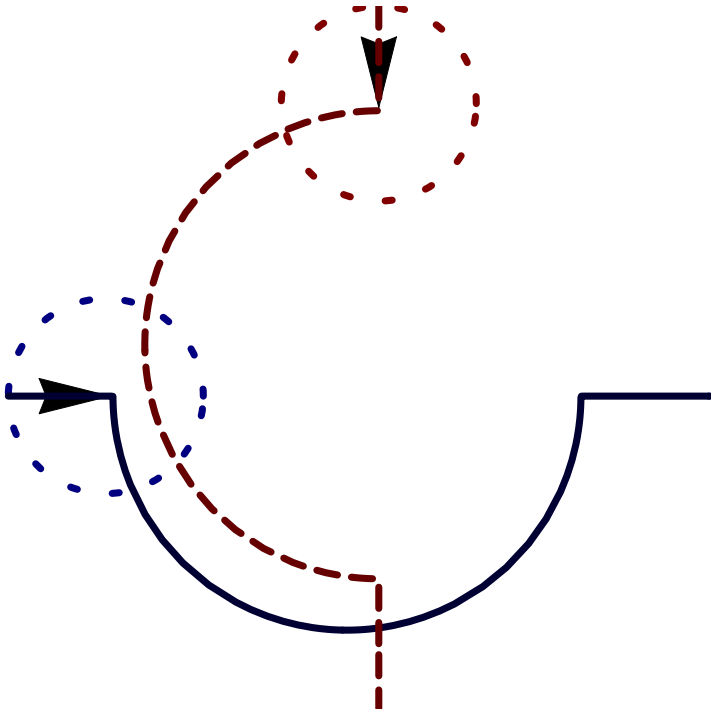
	Dom	Op	Base	Modal	Quant	Cmpl	Aut
DL	$\mathbb{N}$		$\text{FOL}_{(\mathbb{N})}$		FV+unify	$/\mathbb{N}$	
d$\mathcal{L}$	$\mathbb{R}$	x'	$\text{FOL}_{\mathbb{R}}$	ODE	FV+requant+QE	/ODE	IBC

Outline

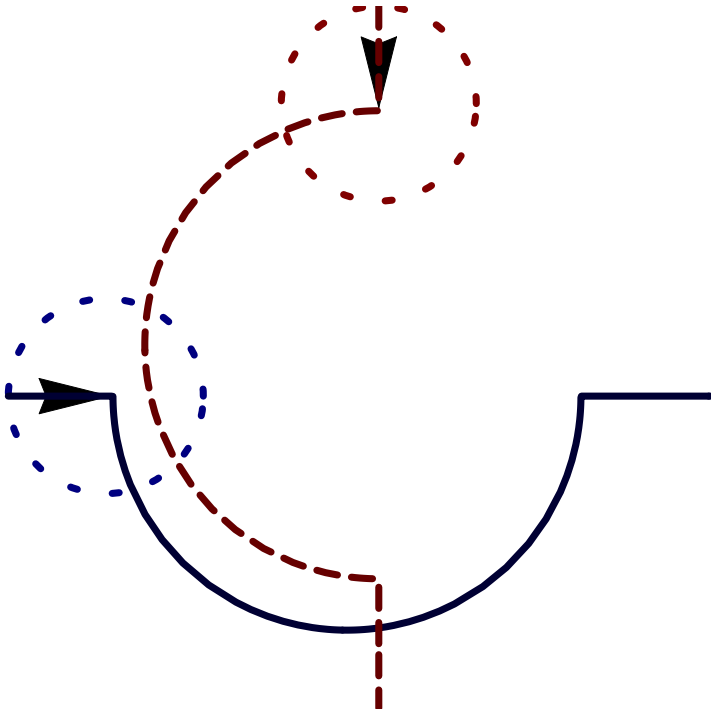
Air Traffic Control



Air Traffic Control



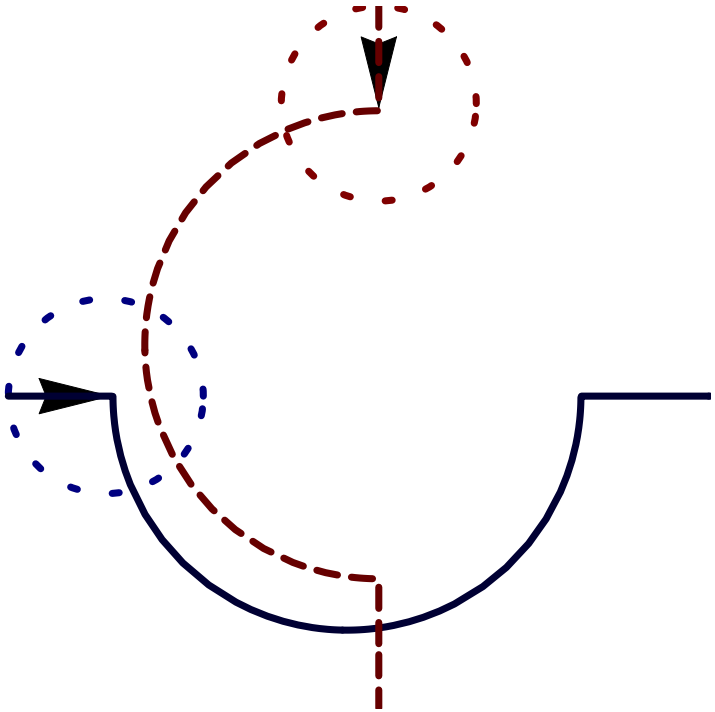
Air Traffic Control

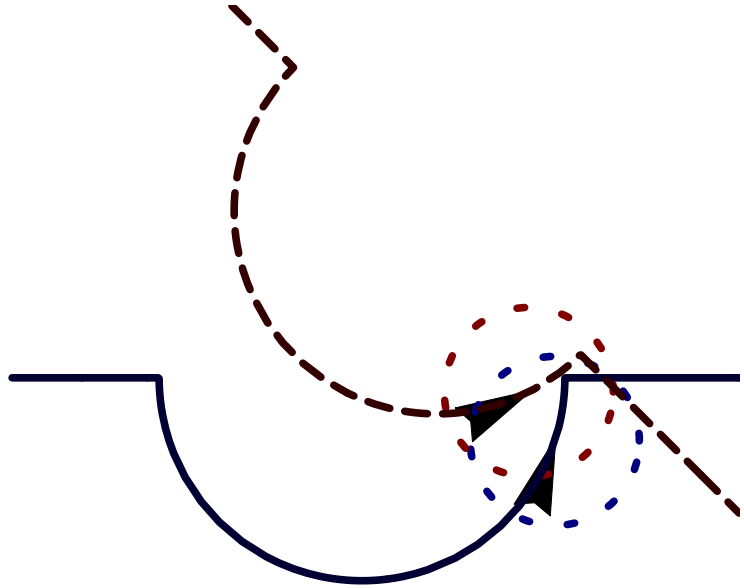


Verification?

looks correct

Air Traffic Control

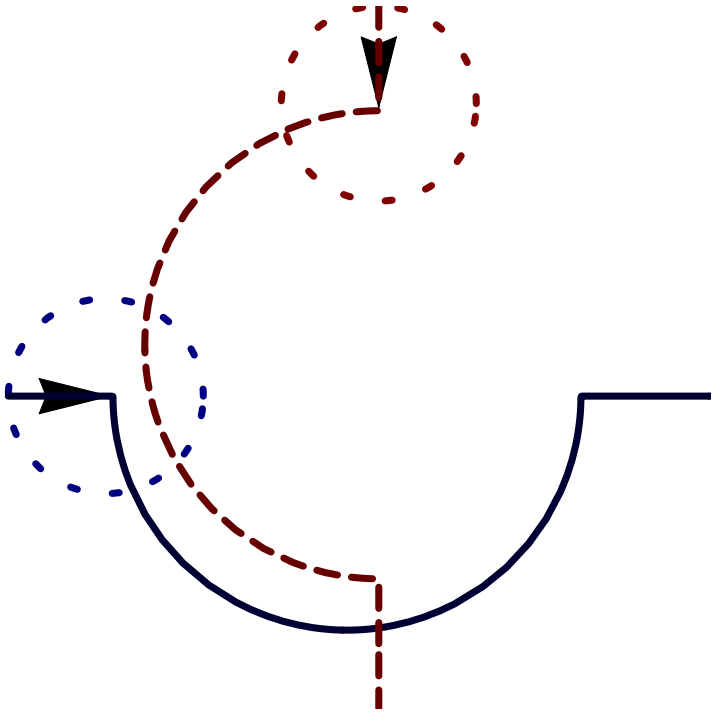


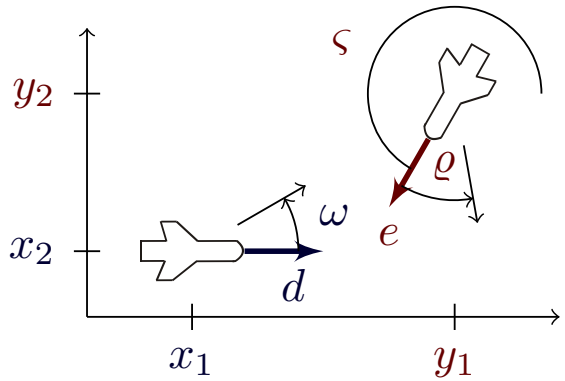


Verification?

looks correct NO!

Air Traffic Control

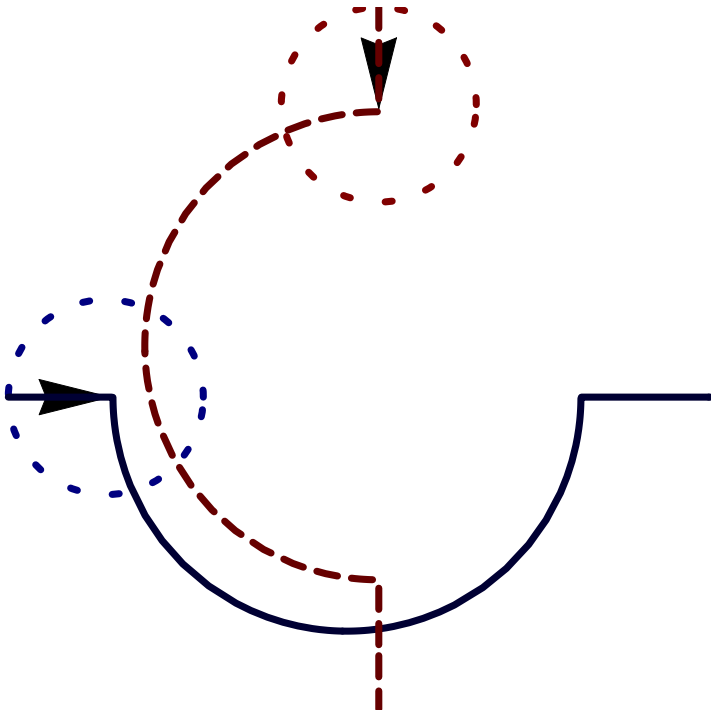


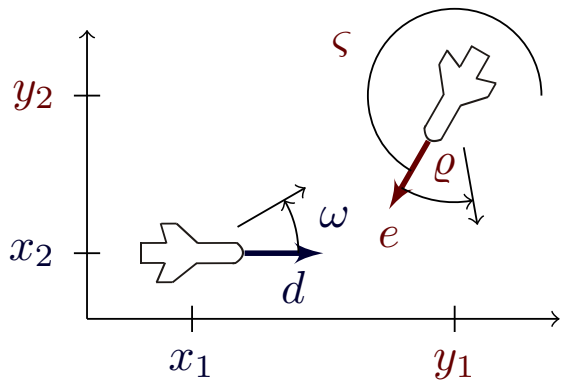


$$\begin{bmatrix} x_1' = -v_1 + v_2 \cos\vartheta + \omega x_2 \\ x_2' = v_2 \sin\vartheta - \omega x_1 \\ \vartheta' = \varpi - \omega \end{bmatrix}$$

Verification?

looks correct NO!

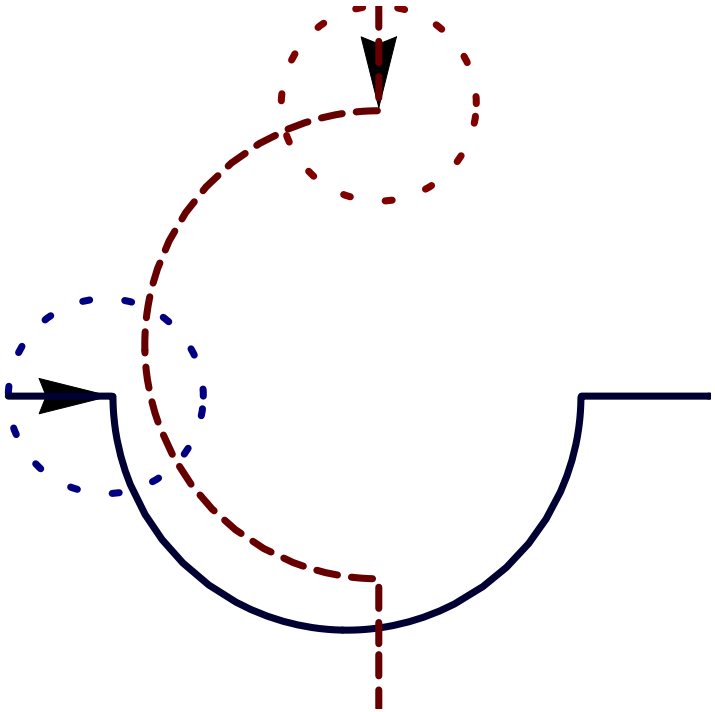


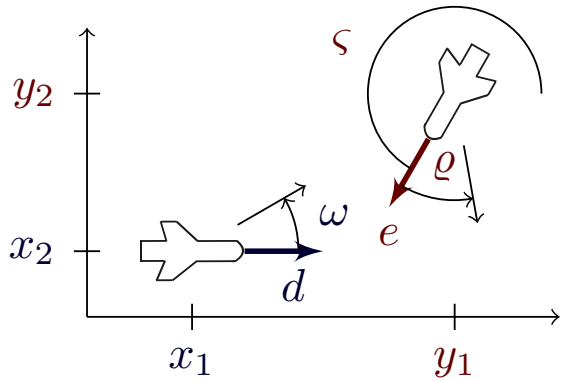


$$\begin{bmatrix} x_1' = -v_1 + v_2 \cos\vartheta + \omega x_2 \\ x_2' = v_2 \sin\vartheta - \omega x_1 \\ \vartheta' = \varpi - \omega \end{bmatrix}$$

Example ("Solving" differential equations)

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





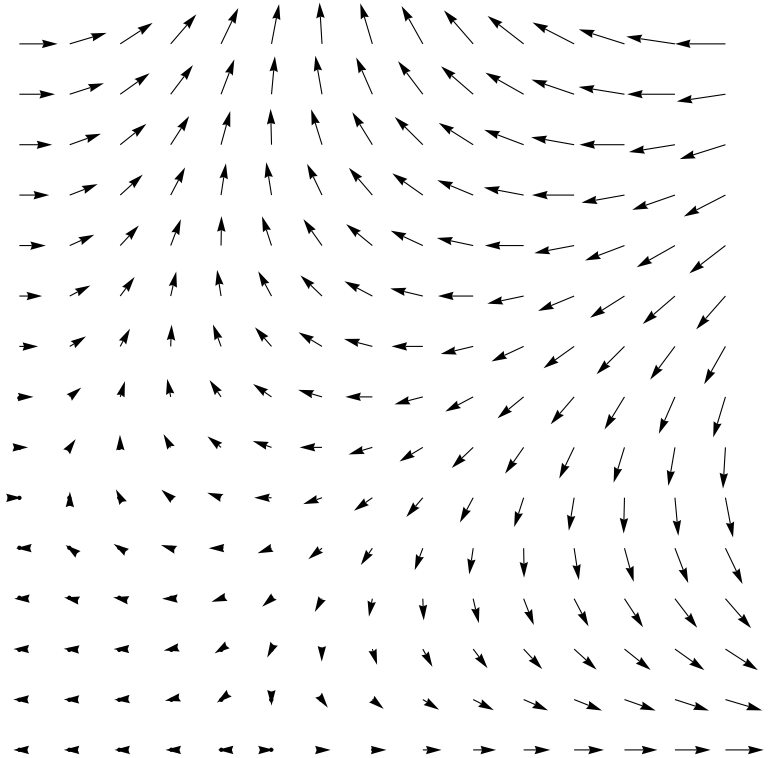
$$\begin{bmatrix} x_1' = -v_1 + v_2 \cos\vartheta + \omega x_2 \\ x_2' = v_2 \sin\vartheta - \omega x_1 \\ \vartheta' = \varpi - \omega \end{bmatrix}$$

Example ("Solving" differential equations)

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

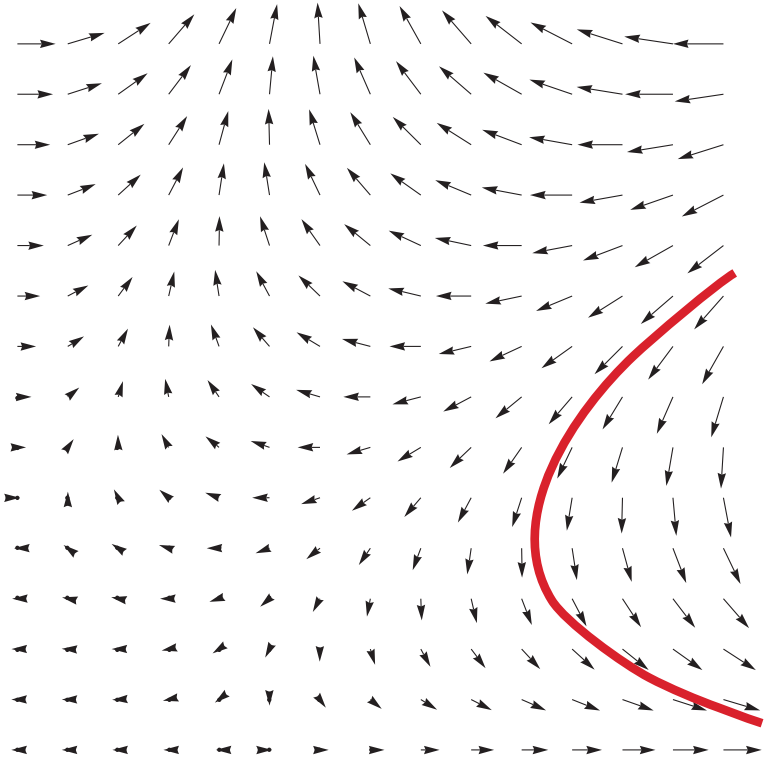
"Definition" (Differential Invariant)

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



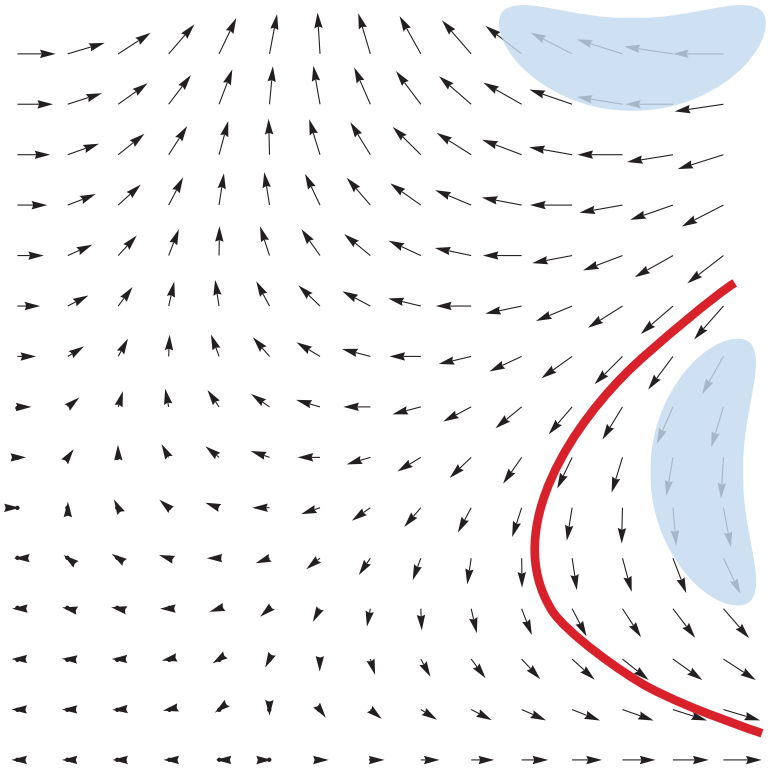
"Definition" (Differential Invariant)

"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)

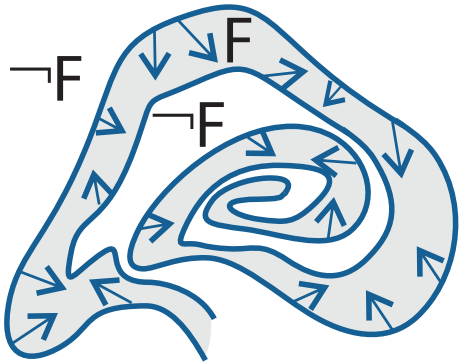
F closed under total differentiation with respect to differential constraints

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

▸ Details

Definition (Differential Invariant)

F closed under total differentiation with respect to differential constraints

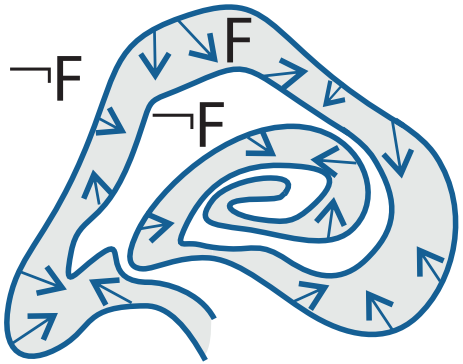


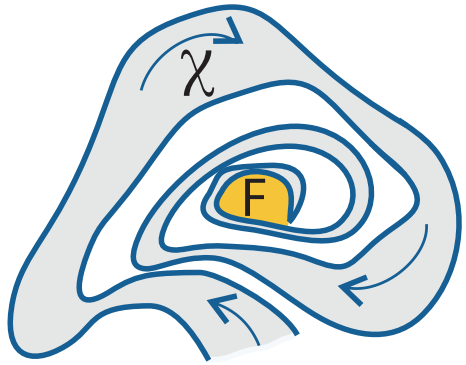
Details

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

Definition (Differential Invariant)

F closed under total differentiation with respect to differential constraints





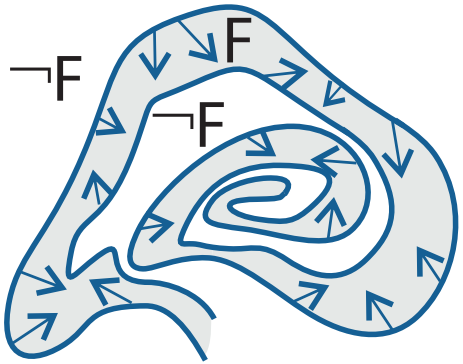
Details

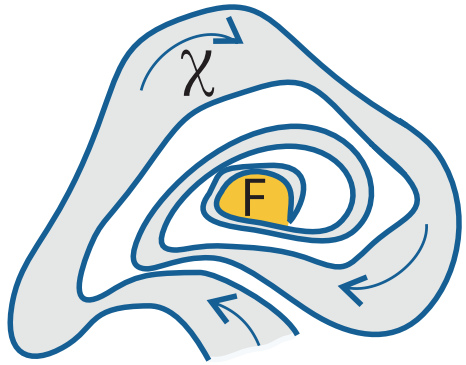
$$\frac{\vdash (\chi \rightarrow F')}{\chi \rightarrow F \vdash [x' = \theta \wedge \chi]F}$$

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

Definition (Differential Invariant)

F closed under total differentiation with respect to differential constraints





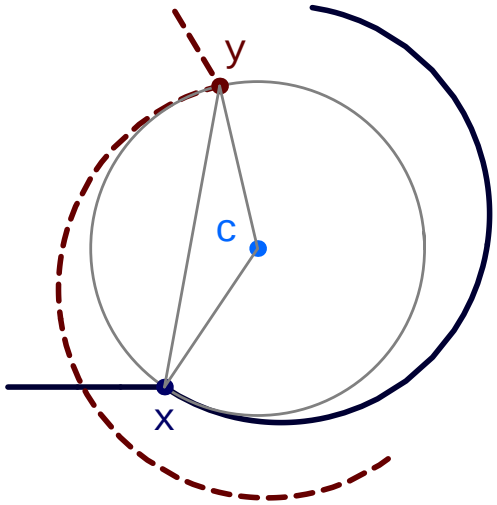
Details

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

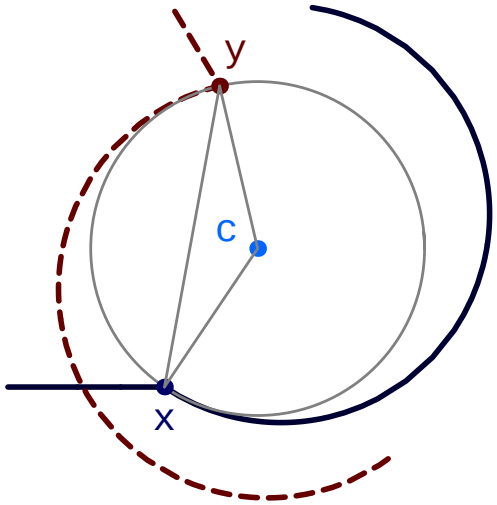
$$\frac{\vdash (\neg F \wedge \chi \rightarrow F'_{\gg})}{[x' = \theta \wedge \neg F]\chi \vdash \langle x' = \theta \wedge \chi \rangle F}$$

Total differential F' of *formulas*?

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

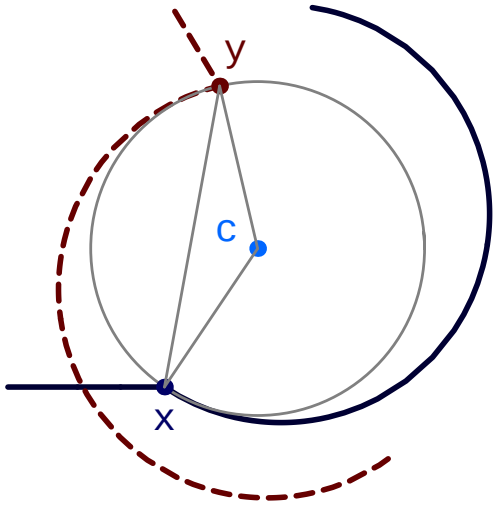


$$\frac{\vdash \frac{\partial\|x-y\|^2}{\partial x_1}x_1' + \frac{\partial\|x-y\|^2}{\partial y_1}y_1' + \frac{\partial\|x-y\|^2}{\partial x_2}x_2' + \frac{\partial\|x-y\|^2}{\partial y_2}y_2' \geq \frac{\partial p^2}{\partial x_1}x_1' \ldots}{\vdash [x_1' = d_1, d_1' = -\omega d_2, x_2' = d_2, d_2'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



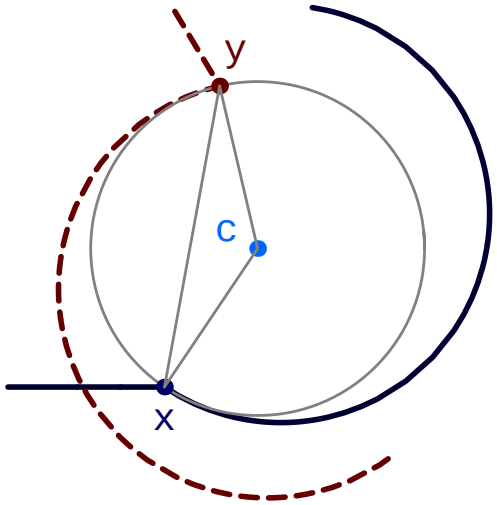
$$\vdash \frac{\partial \|x-y\|^2}{\partial x_1} x_1' + \frac{\partial \|x-y\|^2}{\partial y_1} y_1' + \frac{\partial \|x-y\|^2}{\partial x_2} x_2' + \frac{\partial \|x-y\|^2}{\partial y_2} y_2' \geq \frac{\partial p^2}{\partial x_1} x_1' \dots$$

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



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

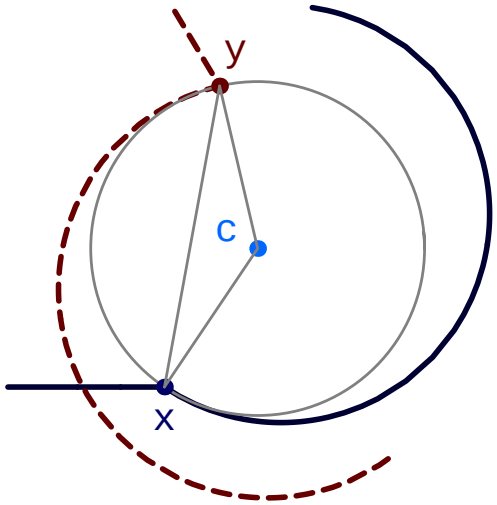
$$\vdash [x_1' = d_1, d_1' = -\omega d_2, x_2' = d_2, d_2' = \omega d_1, ..](x_1 - y_1)^2 + (x_2 - y_2)^2 \geq p^2$$

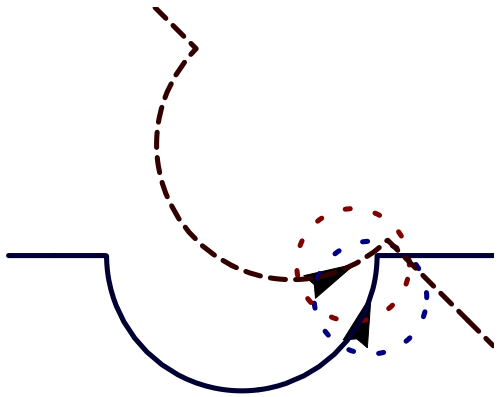


$$\vdash 2(x_1 - y_1)(d_1 - e_1) + 2(x_2 - y_2)(d_2 - e_2) \geq 0$$

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

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

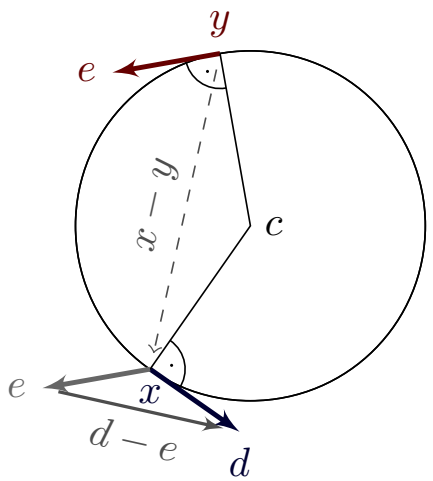


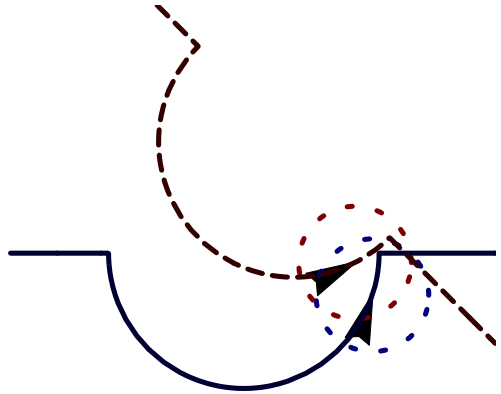


$$\vdash 2(x_1 - y_1)(d_1 - e_1) + 2(x_2 - y_2)(d_2 - e_2) \geq 0$$

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

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

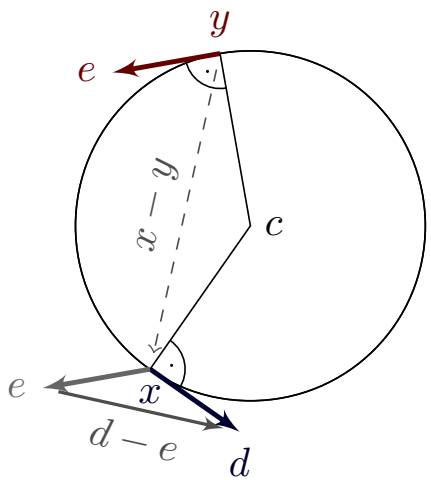


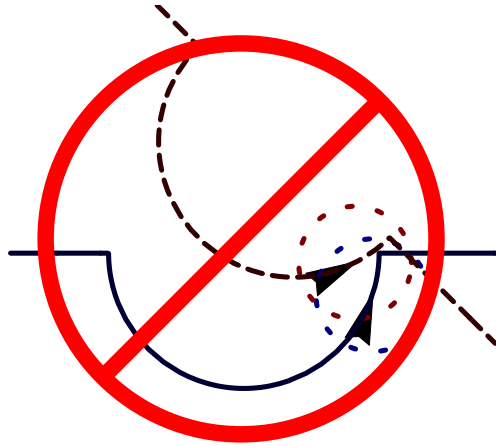


$$\vdash 2(x_1 - y_1)(d_1 - e_1) + 2(x_2 - y_2)(d_2 - e_2) \geq 0$$

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

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





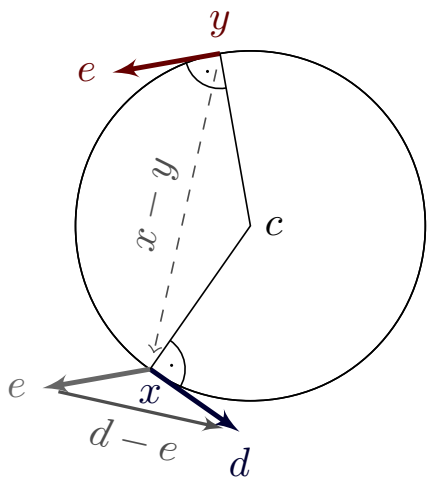
$$.. \vdash [d_1' = -\omega d_2, e_1' = -\omega e_2, x_2' = d_2, d_2' = \omega d_1, ..] d_1 - e_1 = -\omega(x_2 - y_2)$$

$$\vdash 2(x_1 - y_1)(-\omega(x_2 - y_2)) + 2(x_2 - y_2)\omega(x_1 - y_1) \geq 0$$

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

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

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



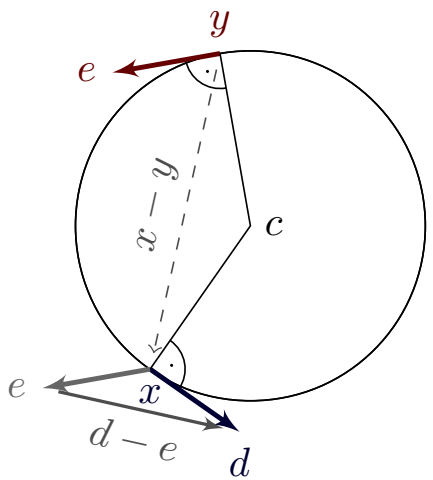
$$.. \vdash [d_1' = -\omega d_2, e_1' = -\omega e_2, x_2' = d_2, d_2' = \omega d_1, ..] d_1 - e_1 = -\omega(x_2 - y_2)$$

$$\vdash 2(x_1-y_1)(-\omega(x_2-y_2))+2(x_2-y_2)\omega(x_1-y_1)\geq 0$$

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

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

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



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

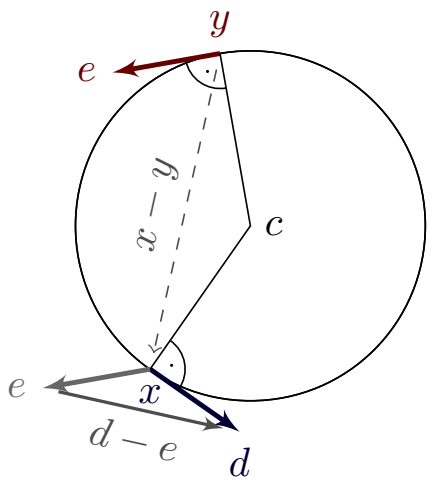
$$.. \vdash [d_1'=-\omega d_2,e_1'=-\omega e_2,x_2'=d_2,d_2'=\omega d_1,..]d_1-e_1=-\omega(x_2-y_2)$$

$$\vdash 2(x_1-y_1)(-\omega(x_2-y_2))+2(x_2-y_2)\omega(x_1-y_1)\geq 0$$

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

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

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



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

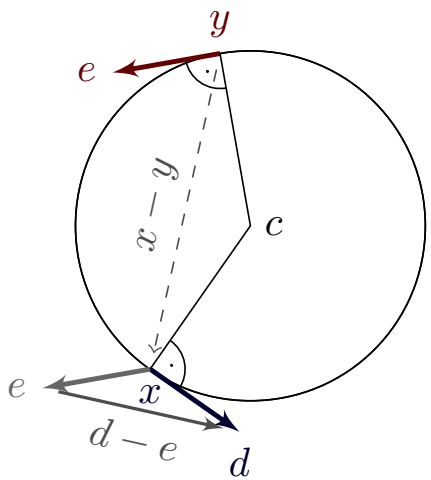
$$.. \vdash [d_1'=-\omega d_2,e_1'=-\omega e_2,x_2'=d_2,d_2'=\omega d_1,..]d_1-e_1=-\omega(x_2-y_2)$$

$$\vdash 2(x_1-y_1)(-\omega(x_2-y_2))+2(x_2-y_2)\omega(x_1-y_1)\geq 0$$

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

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

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



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

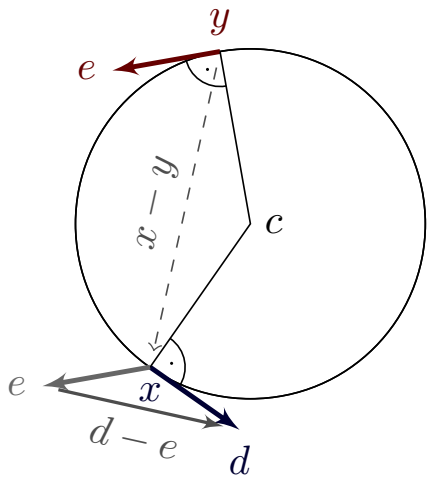
$$..\vdash [d_1'=-\omega d_2,e_1'=-\omega e_2,x_2'=d_2,d_2'=\omega d_1,..]d_1-e_1=-\omega(x_2-y_2)$$

$$\vdash 2(x_1-y_1)(-\omega(x_2-y_2))+2(x_2-y_2)\omega(x_1-y_1)\geq 0$$

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

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

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



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

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

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

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

Proposition (Differential cut saturation)

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

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

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

$$\vdash 2(x_1-y_1)(-\omega(x_2-y_2))+2(x_2-y_2)\omega(x_1-y_1)\geq 0$$

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

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

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

refine dynamics

by differential cut

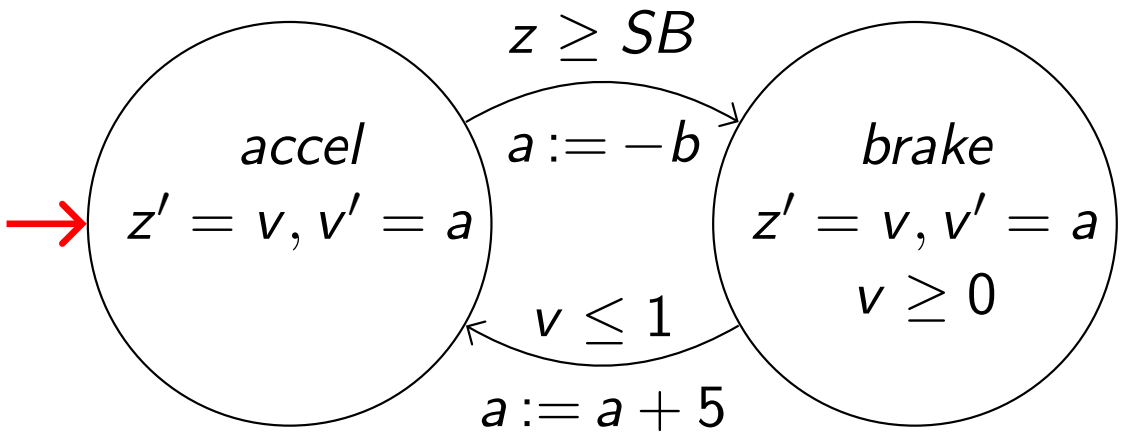
$$\vdash -\omega d_2+\omega e_2=-\omega(d_2-e_2)$$

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

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

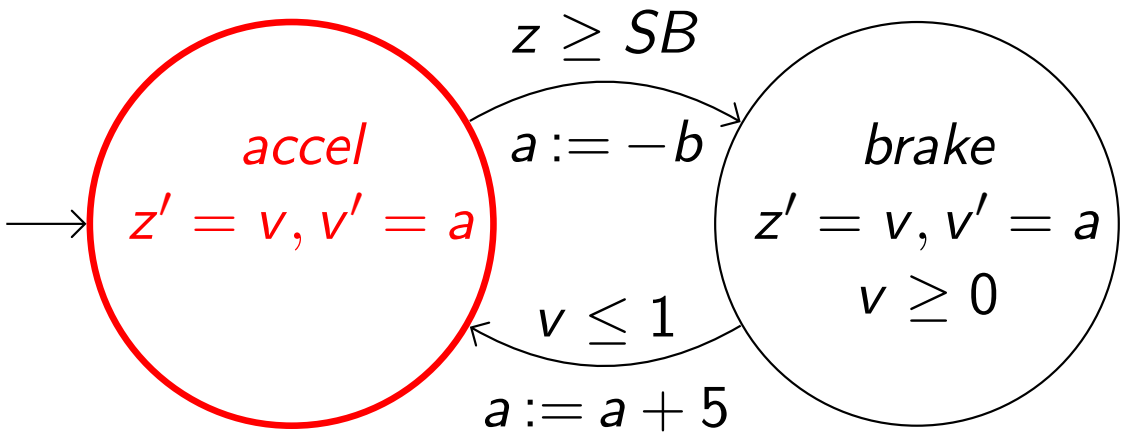
Outline

Embedding Hybrid Automata as Hybrid Programs



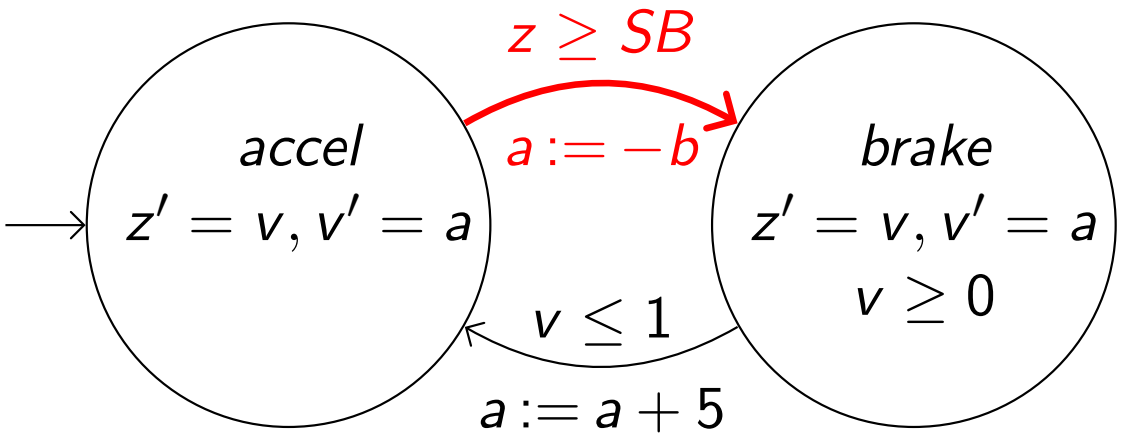
$$
\begin{aligned}
& q := accel; \\
& \big(\quad (?q = accel; \quad z' = v, v' = a) \\
& \cup \ (?q = accel \wedge z \geq SB; \quad a := -b; \quad q := brake; \quad ?v \geq 0) \\
& \cup \ (?q = brake; \quad z' = v, v' = a \wedge v \geq 0) \\
& \cup \ (?q = brake \wedge v \leq 1; \quad a := a + 5; \quad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

Embedding Hybrid Automata as Hybrid Programs



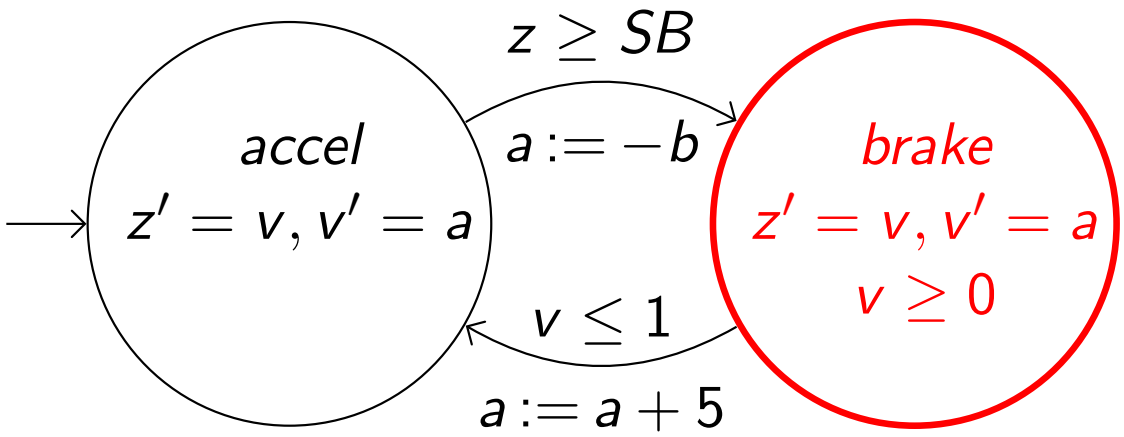
$$
\begin{aligned}
& q := \mathit{accel}; \\
& \big(\quad (?q = \mathit{accel}; \quad z' = v, v' = a) \\
& \cup \ (?q = \mathit{accel} \wedge z \geq SB; \quad a := -b; \quad q := \mathit{brake}; \quad ?v \geq 0) \\
& \cup \ (?q = \mathit{brake}; \quad z' = v, v' = a \wedge v \geq 0) \\
& \cup \ (?q =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

Embedding Hybrid Automata as Hybrid Programs



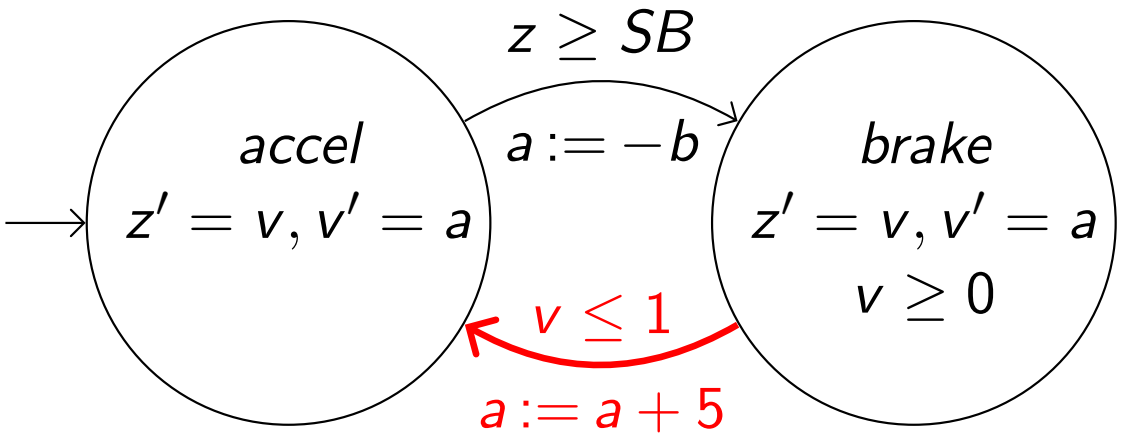
$$
\begin{aligned}
&q := accel; \\
&\big(\quad (?q = accel; \quad z' = v, v' = a) \\
&\cup \ (?q = accel \wedge z \geq SB; \quad a := -b; \quad q := brake; \quad ?v \geq 0) \\
&\cup \ (?q = brake; \quad z' = v, v' = a \wedge v \geq 0) \\
&\cup \ (?q = brake \wedge v \leq 1; \quad a := a + 5; \quad q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

Embedding Hybrid Automata as Hybrid Programs



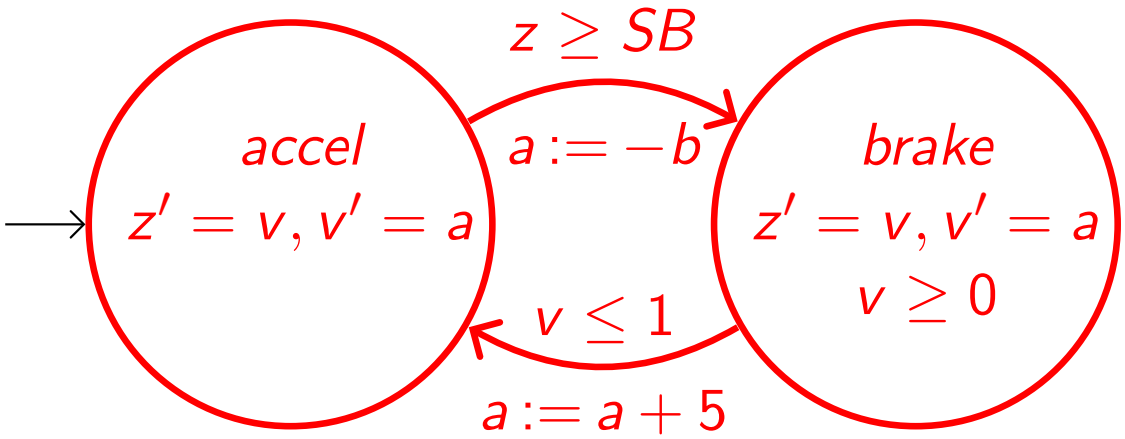
$$
\begin{aligned}
&q := accel; \\
&\big(\quad (?q = accel; \quad z' = v, v' = a) \\
&\cup \ (?q = accel \wedge z \geq SB; \quad a := -b; \quad q := brake; \quad ?v \geq 0) \\
&\cup \ (?q = brake; \quad z' = v, v' = a \wedge v \geq 0) \\
&\cup \ (?q = brake \wedge v \leq 1; \quad a := a + 5; \quad q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

Embedding Hybrid Automata as Hybrid Programs



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

Embedding Hybrid Automata as Hybrid Programs



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

Outline

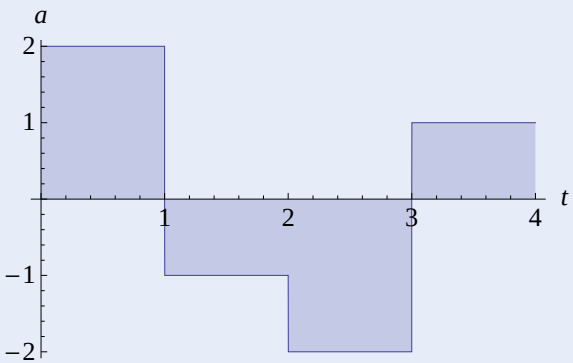
Q: I want to verify my car

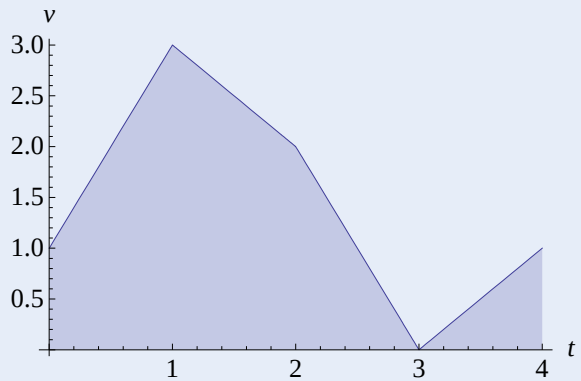
Challenge

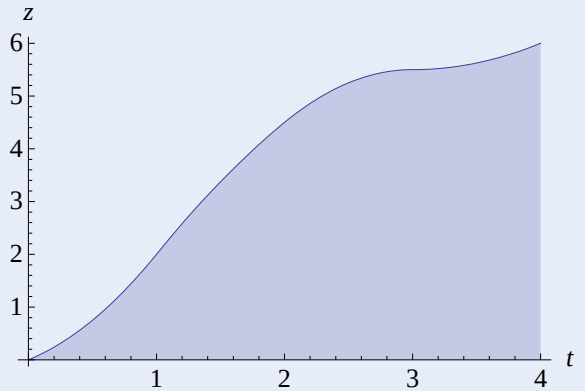
Q: I want to verify my car A: Hybrid systems

Challenge (Hybrid Systems)

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



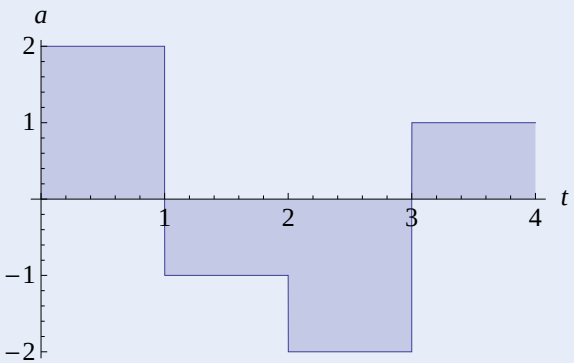


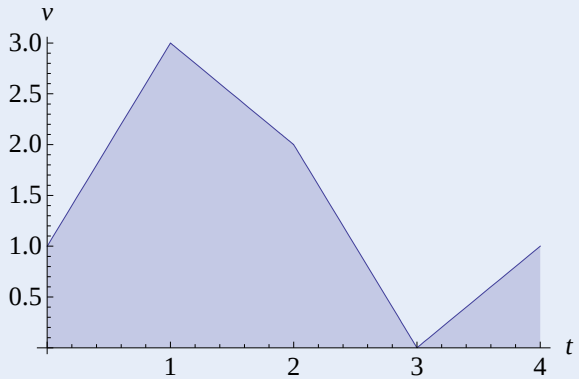


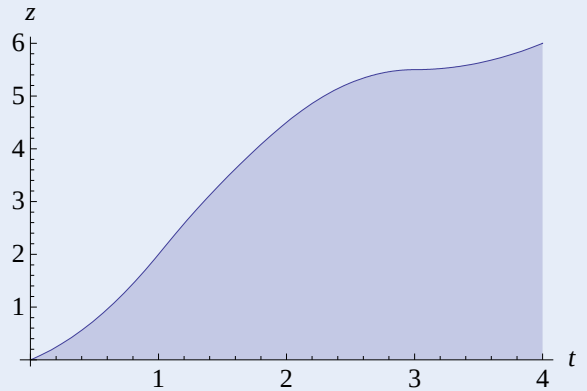
Q: I want to verify my car A: Hybrid systems Q: But there's a lot of cars!

Challenge (Hybrid Systems)

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

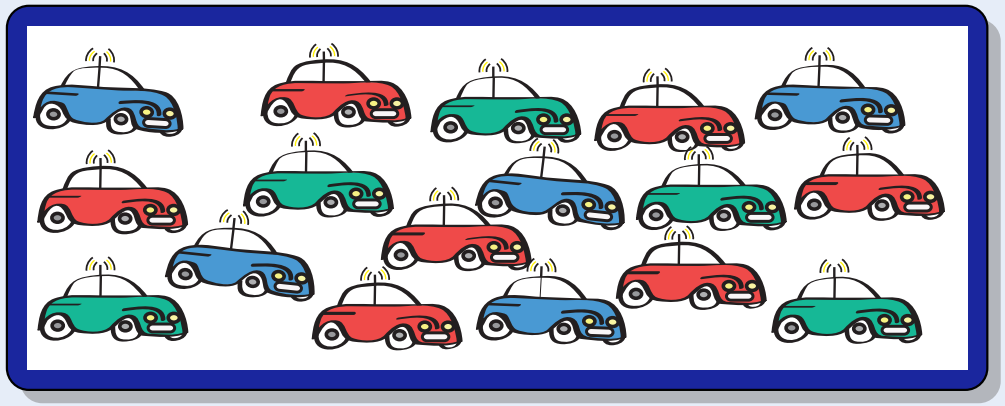






Q: I want to verify a lot of cars

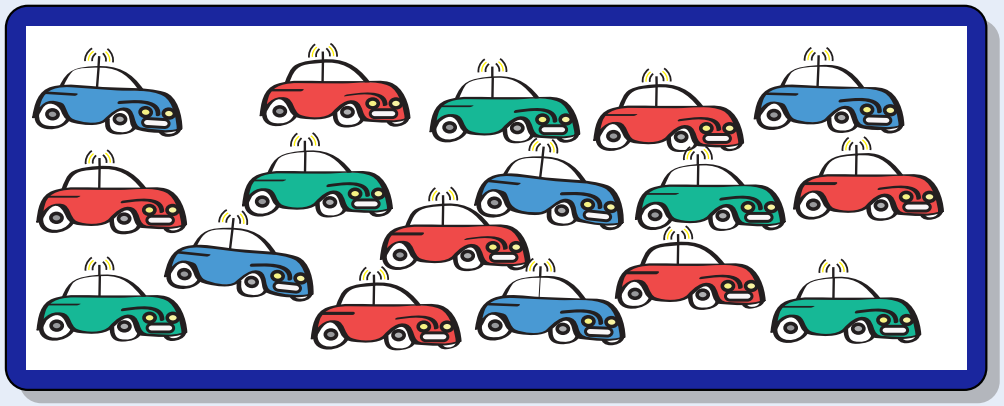
Challenge

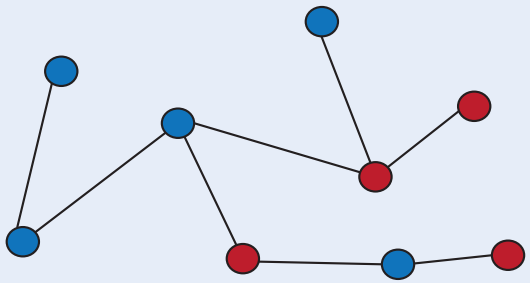


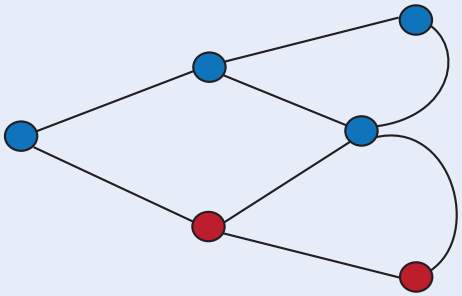
Q: I want to verify a lot of cars A: Distributed systems

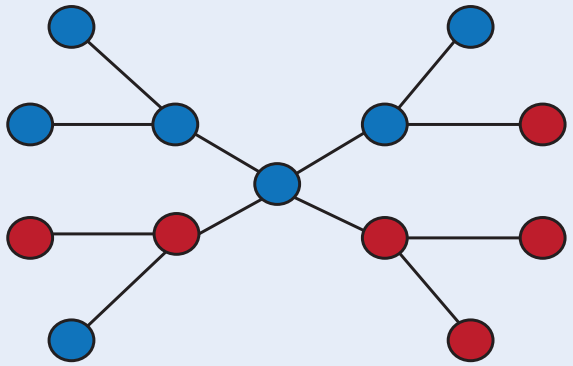
Challenge (Distributed Systems)

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





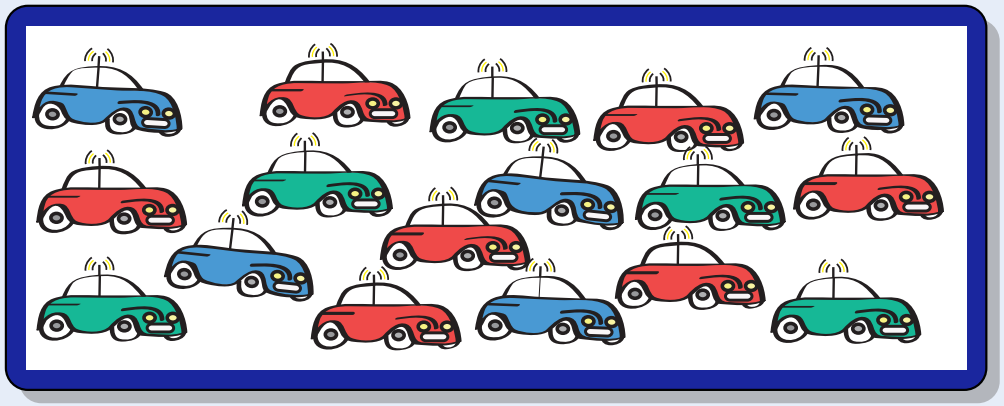


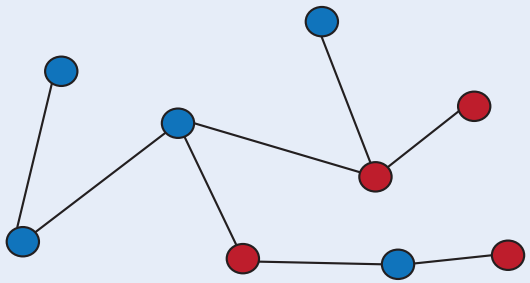


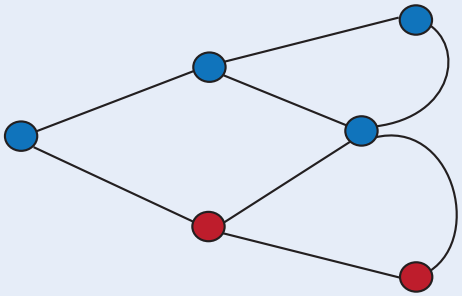
Q: I want to verify a lot of cars A: Distributed systems Q: But they move!

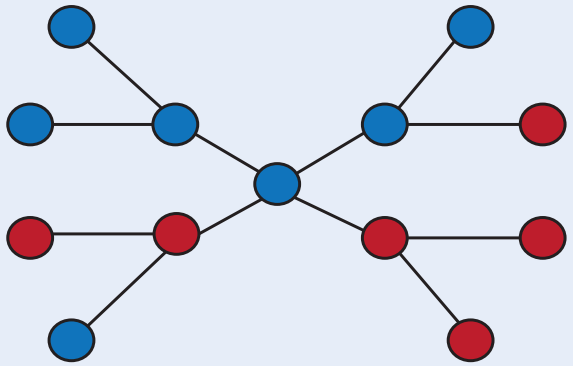
Challenge (Distributed Systems)

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



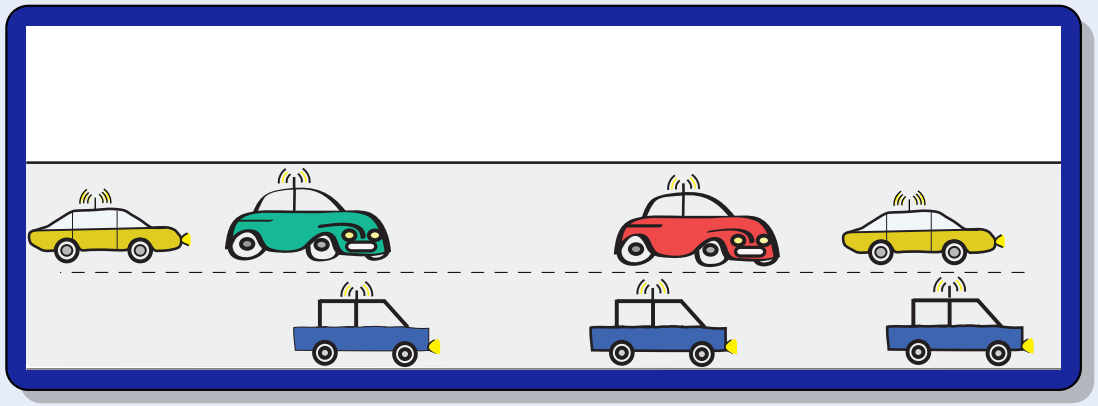






Q: I want to verify lots of moving cars

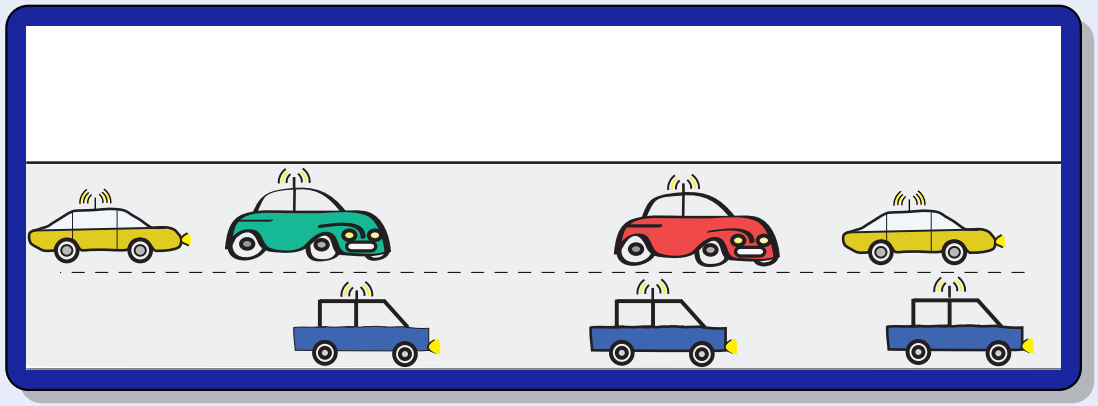
Challenge



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

Challenge (Distributed Hybrid Systems)

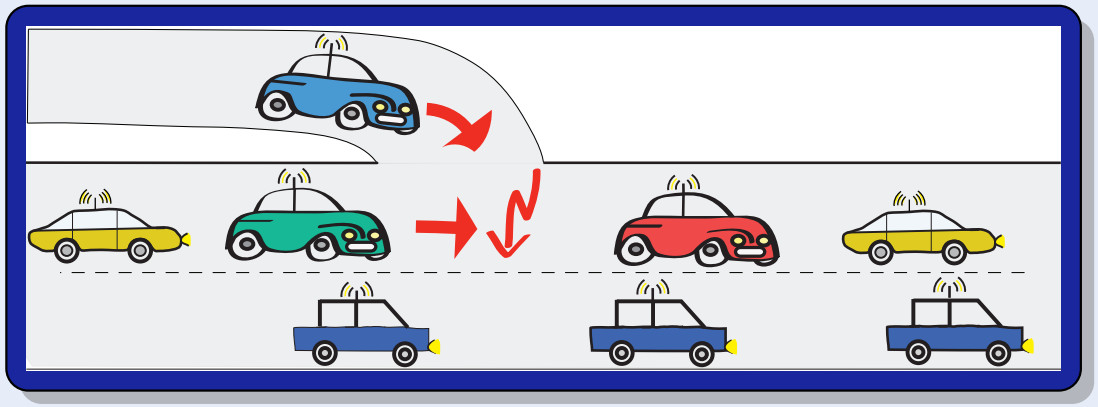
- Continuous dynamics (differential equations)
- Discrete dynamics (control decisions)
- Structural dynamics (remote communication)



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

Challenge (Distributed Hybrid Systems)

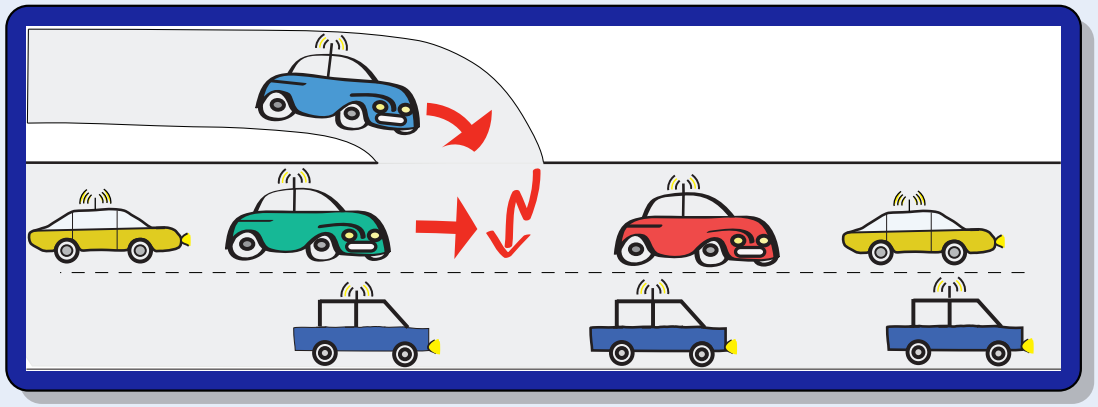
- Continuous dynamics (differential equations)
- Discrete dynamics (control decisions)
- Structural dynamics (remote communication)
- Dimensional dynamics (appearance)



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 (remote communication)
- Dimensional dynamics (appearance)

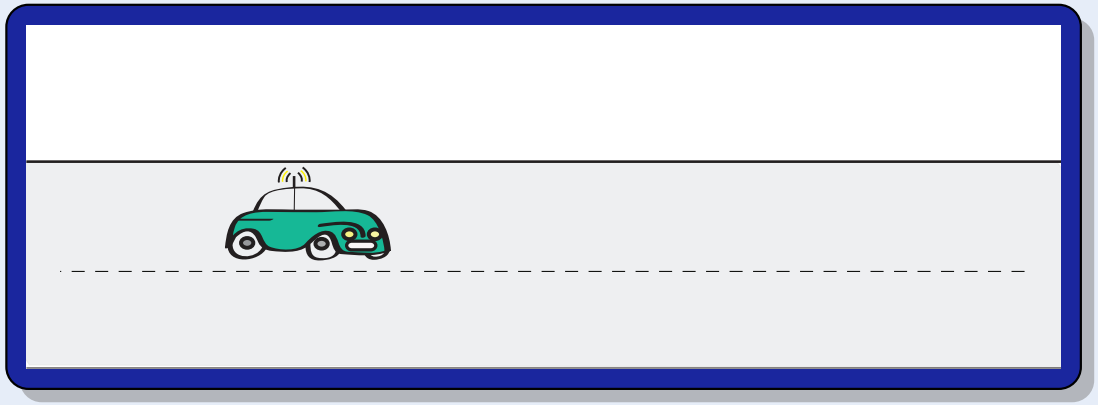


Model for Distributed Hybrid Systems

Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

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

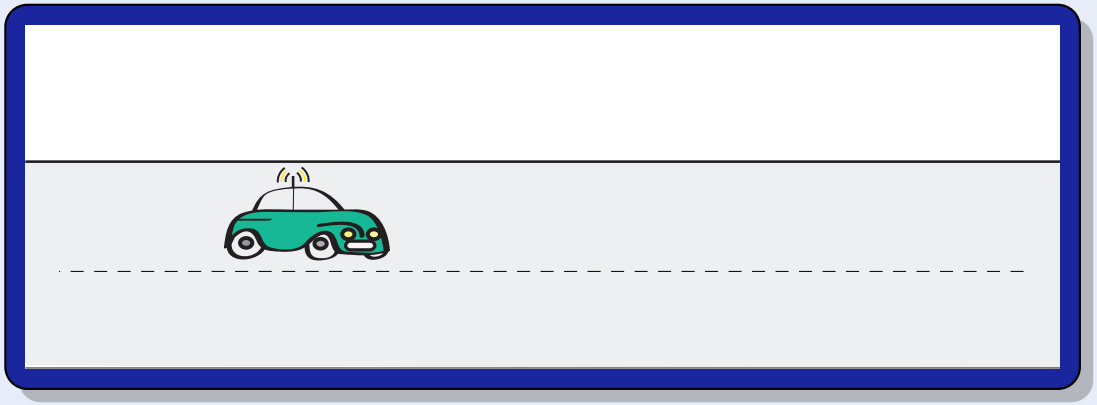


Model for Distributed Hybrid Systems

Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

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



Model for Distributed Hybrid Systems

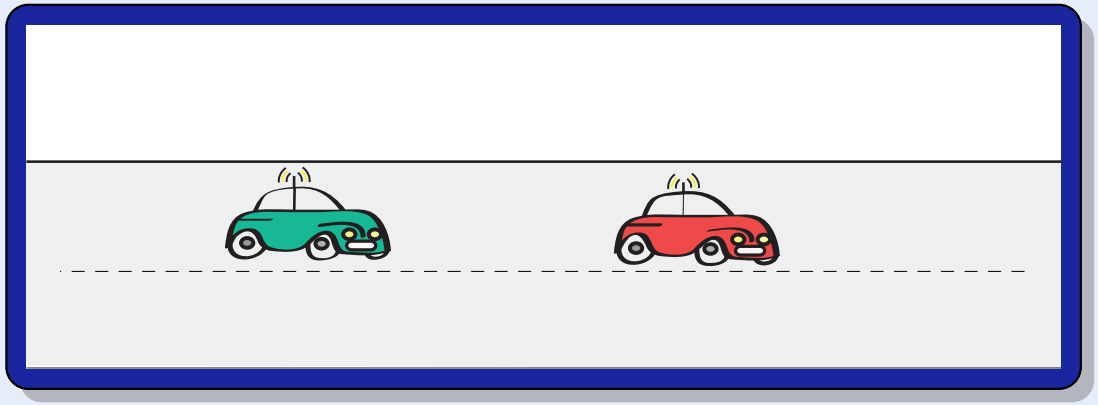
Q: How to model distributed hybrid systems

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 (remote communication)



Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

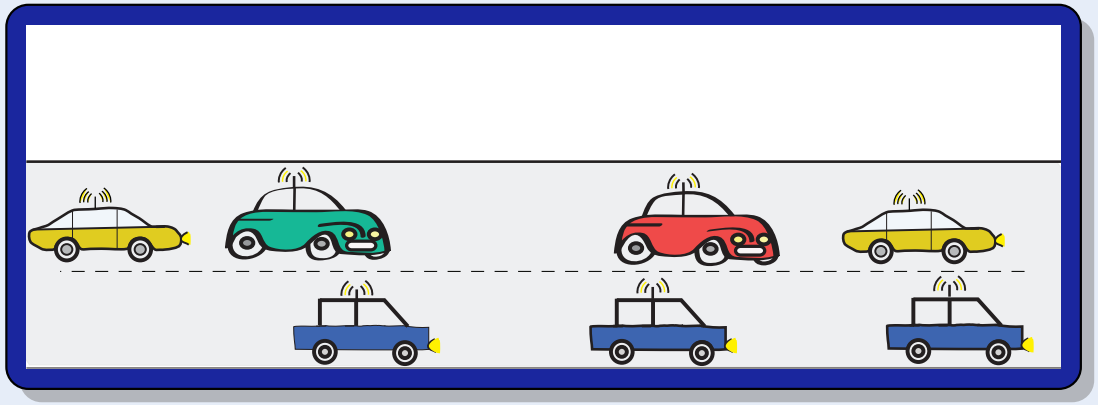
- Continuous dynamics (differential equations)

$$x'' = a$$

- Discrete dynamics (control decisions)

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

- Structural dynamics (remote communication)



Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

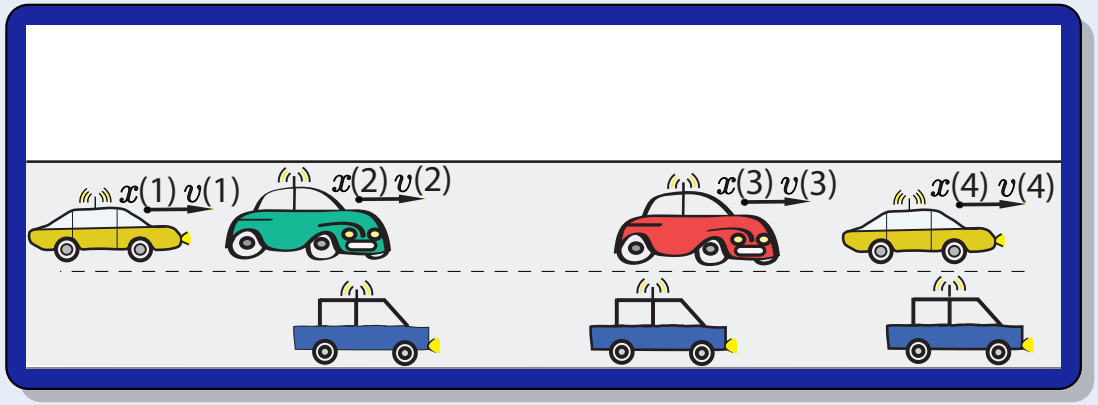
- Continuous dynamics (differential equations)

$$x'' = a$$

- Discrete dynamics (control decisions)

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

- Structural dynamics (remote communication)

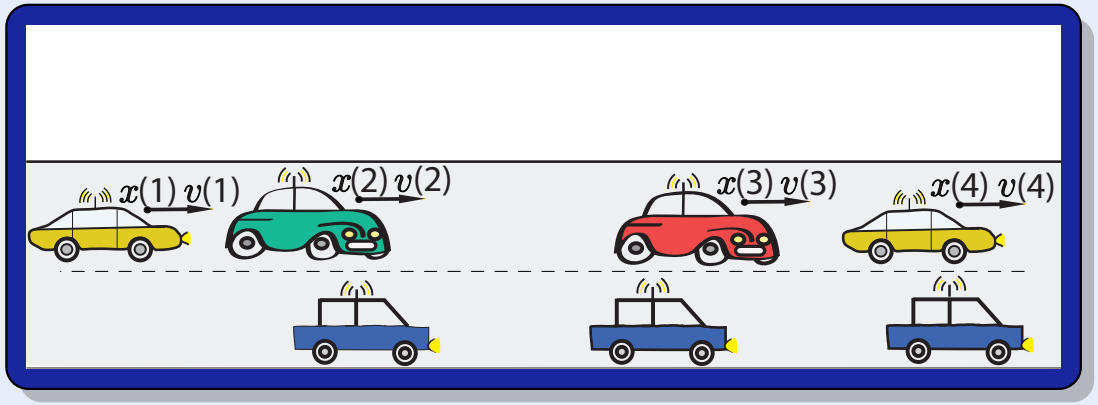


Model for Distributed Hybrid Systems

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 (remote communication)



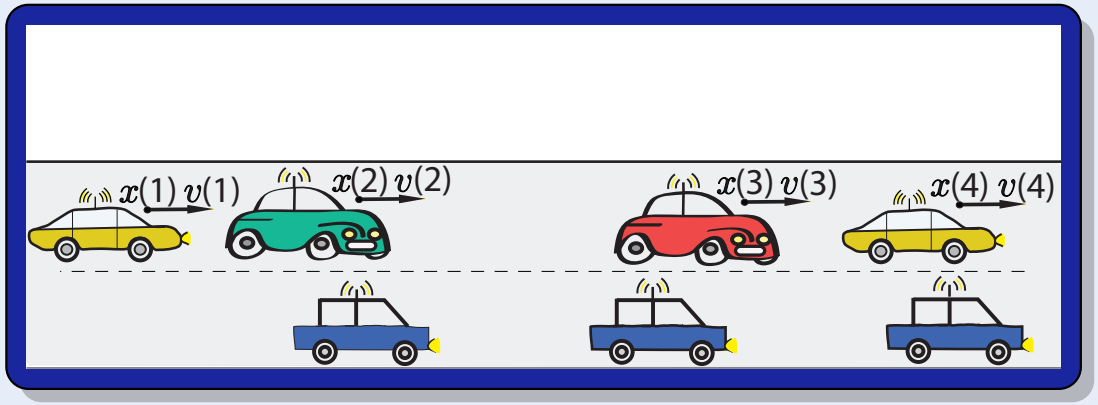
Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

- Continuous dynamics (differential equations)
  $\forall i\, x(i)'' = a(i)$
- Discrete dynamics (control decisions)

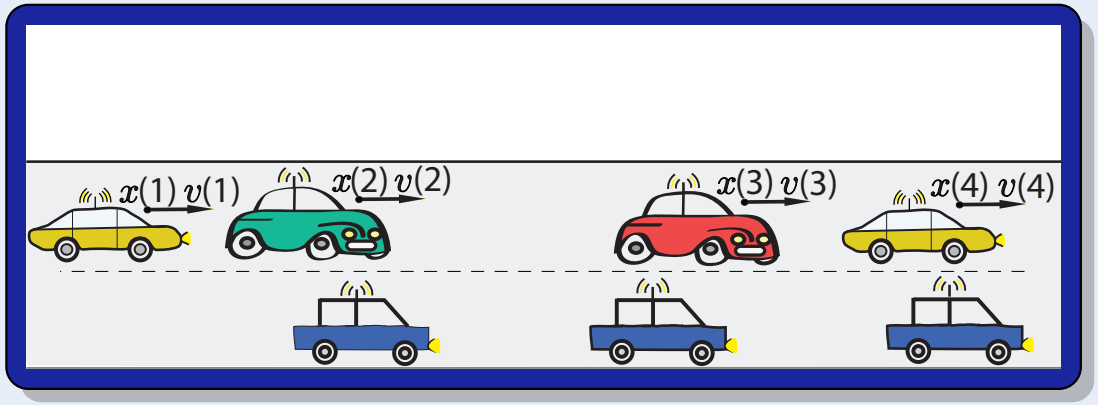
$\forall i\, a(i) := \texttt{if}\, .. \,\texttt{then}\, a \,\texttt{else}\, -b \,\texttt{fi}$

- Structural dynamics (remote communication)



Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)



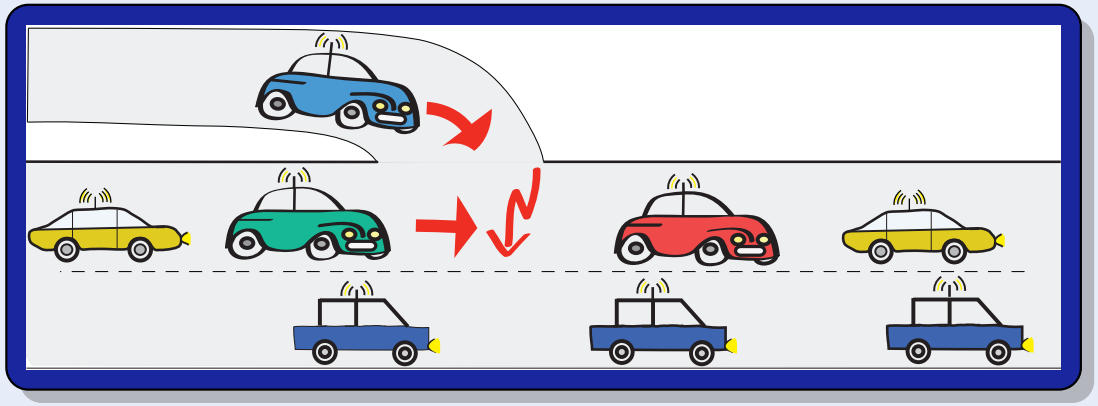
- Continuous dynamics (differential equations)
  $\forall i\, x(i)'' = a(i)$
- Discrete dynamics (control decisions)

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

- Structural dynamics (remote communication)
  $\ell(i) := \mathit{carInFrontOf(i)}$

Q: How to model distributed hybrid systems A: Quantified Hybrid Programs

Model (Distributed Hybrid Systems)



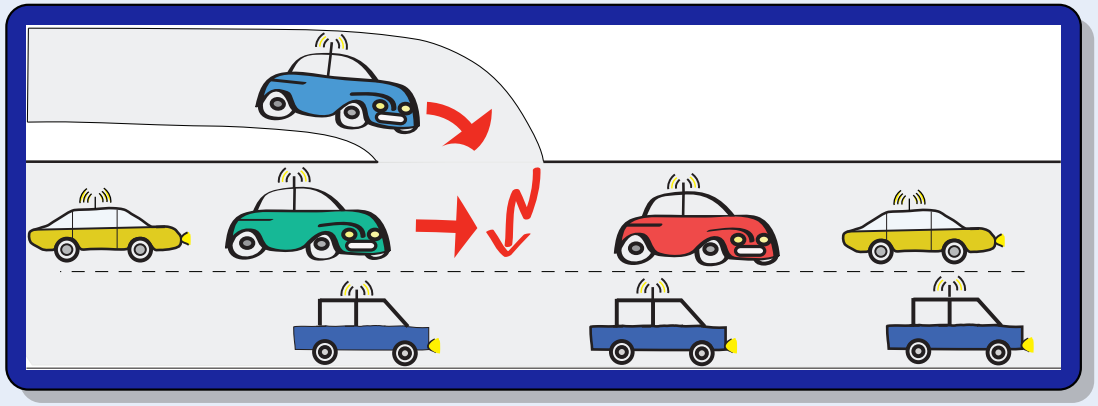
- Continuous dynamics (differential equations)
  $\forall i\, x(i)'' = a(i)$
- Discrete dynamics (control decisions)

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

- Structural dynamics (remote communication)
  $\ell(i) := \mathit{carInFrontOf}(i)$
- Dimensional dynamics (appearance)

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 (remote communication)
  $$\ell(i) := \mathit{carInFrontOf}(i)$$
- Dimensional dynamics (appearance)
  $$n := \texttt{new}\ \mathit{Car}$$

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!)

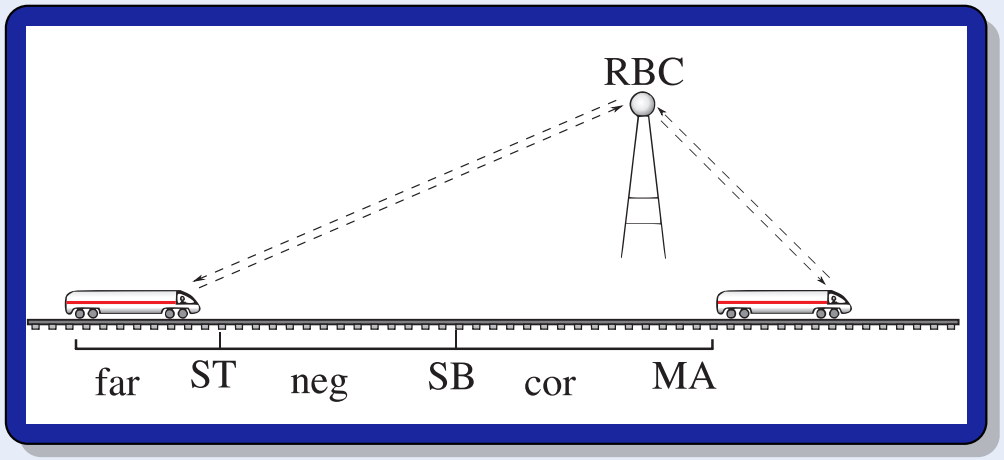
distributed hybrid systems can be verified by recursive decomposition

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.

Outline

Q: I want to verify uncertain trains

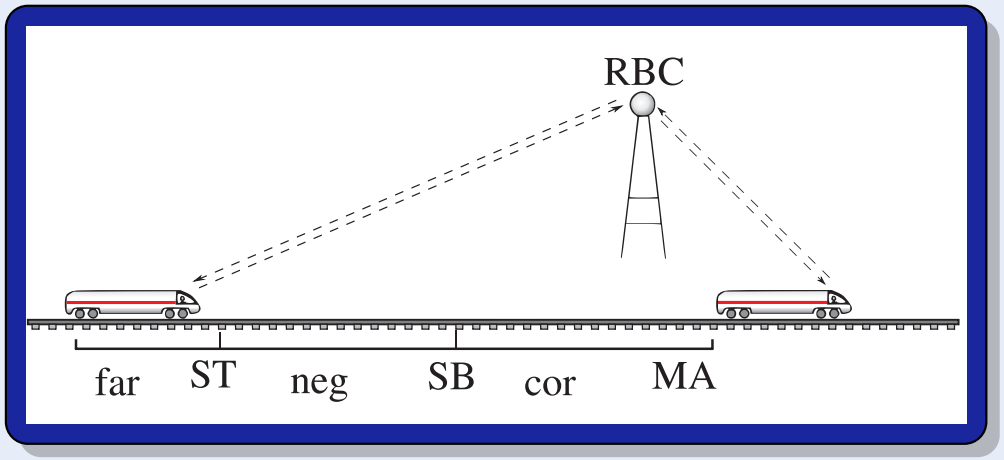
Challenge



Q: I want to verify uncertain trains A: Stochastic hybrid systems

Challenge (Stochastic Hybrid Systems)

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

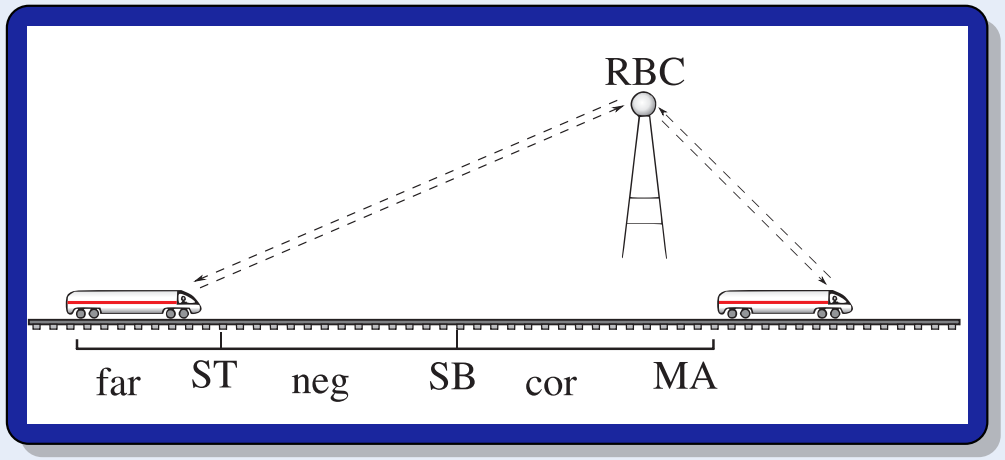


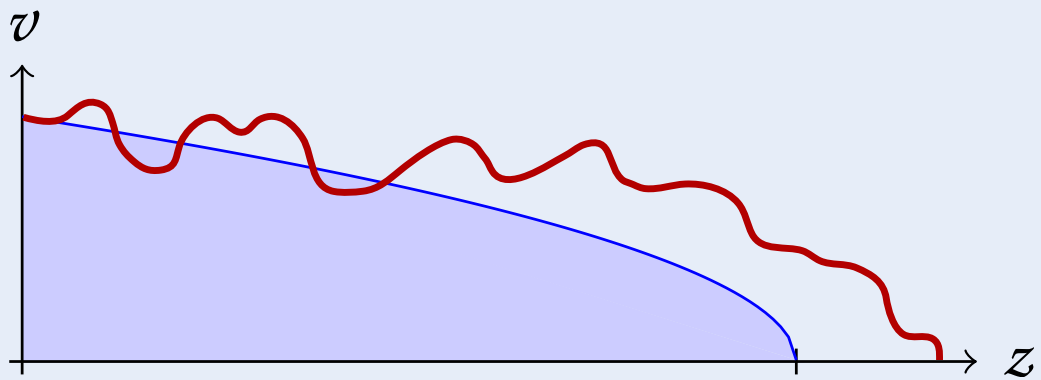
Q: I want to verify uncertain trains A: Stochastic hybrid systems

Challenge (Stochastic Hybrid Systems)

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

- Discrete stochastic (lossy communication)
- Continuous stochastic (wind, track)

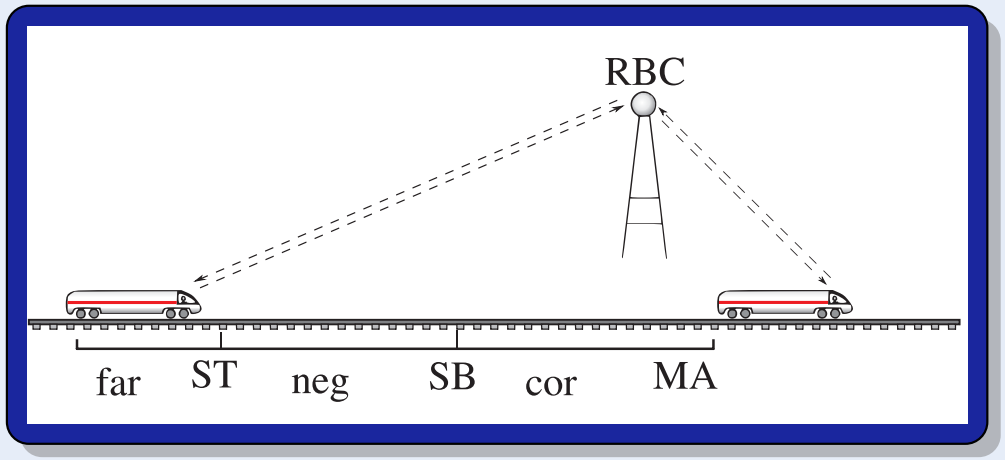


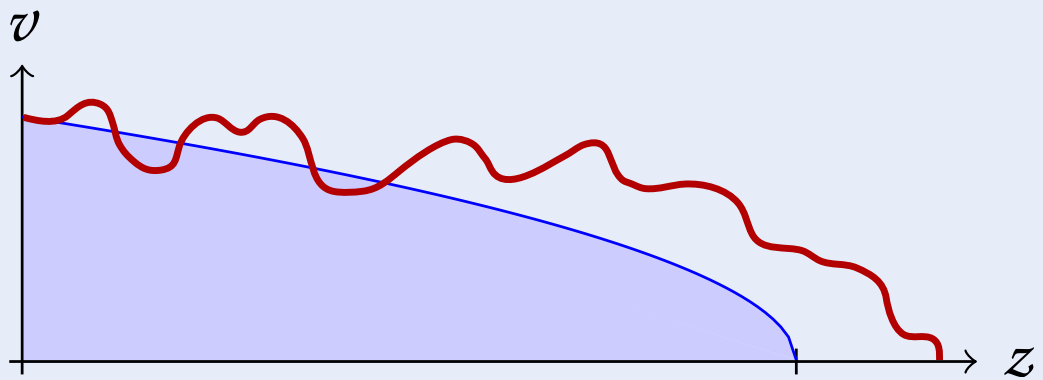


Q: I want to verify uncertain trains A: Stochastic hybrid systems Q: How?

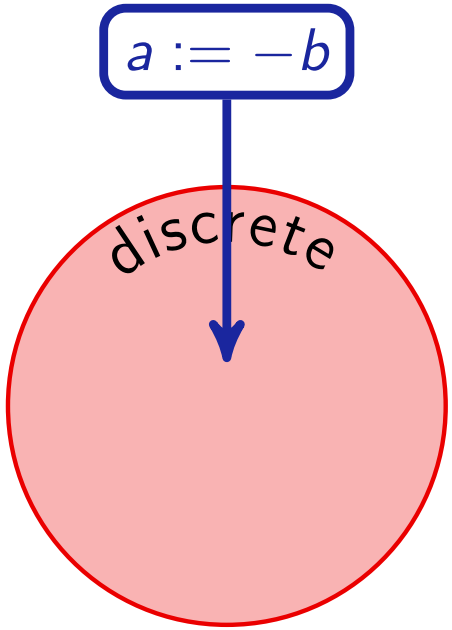
Challenge (Stochastic Hybrid Systems)

- Continuous dynamics (differential equations)
- Discrete dynamics (control decisions)
- Stochastic dynamics (uncertainty)
- Discrete stochastic (lossy communication)
- Continuous stochastic (wind, track)

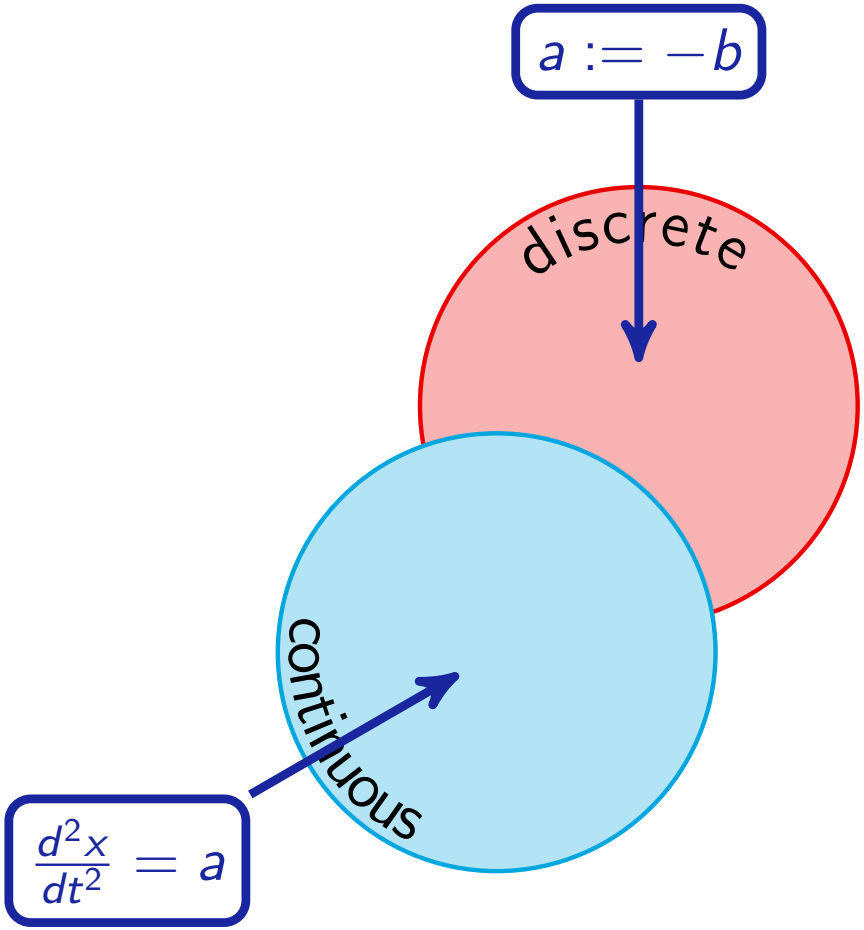




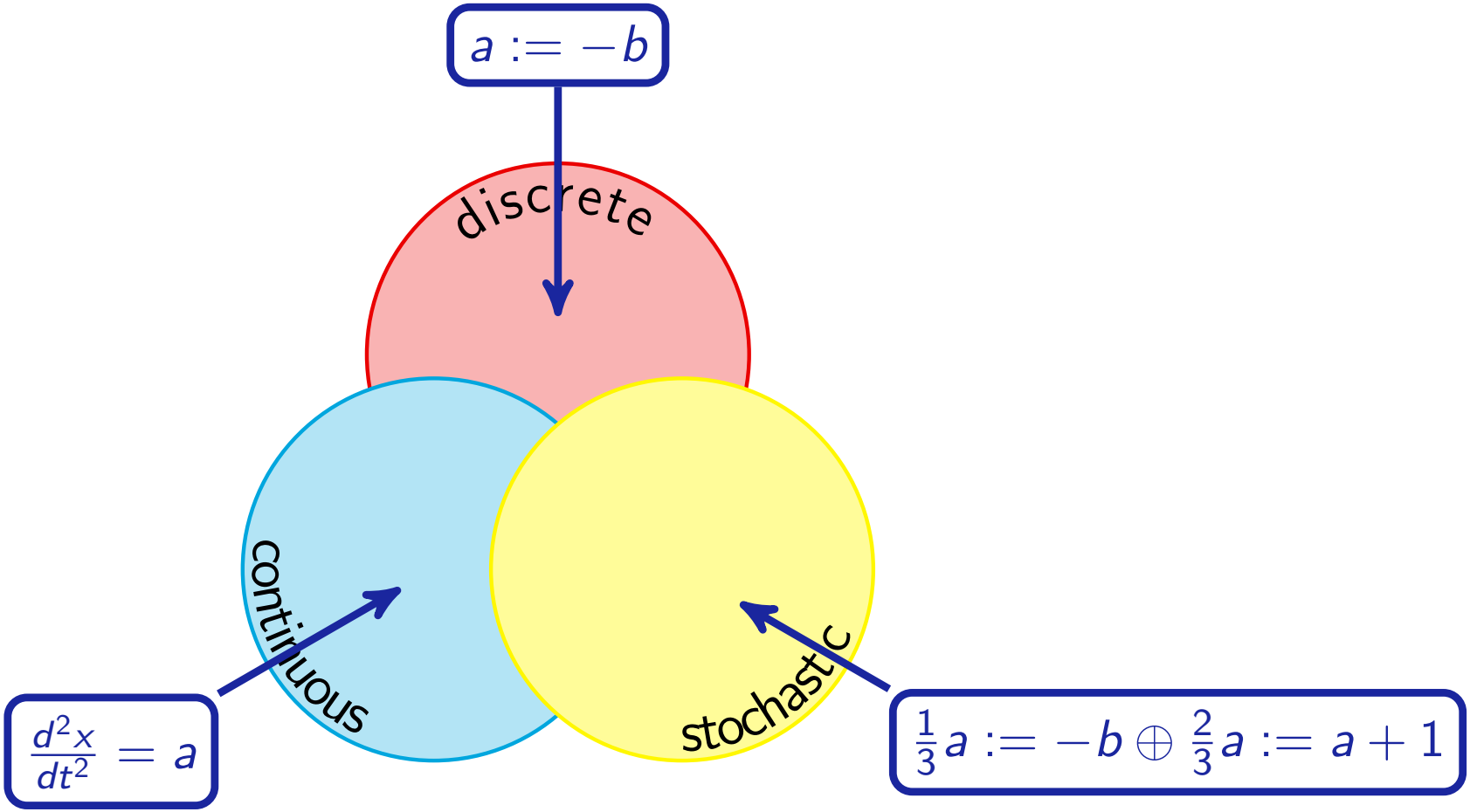
Model for Stochastic Hybrid Systems



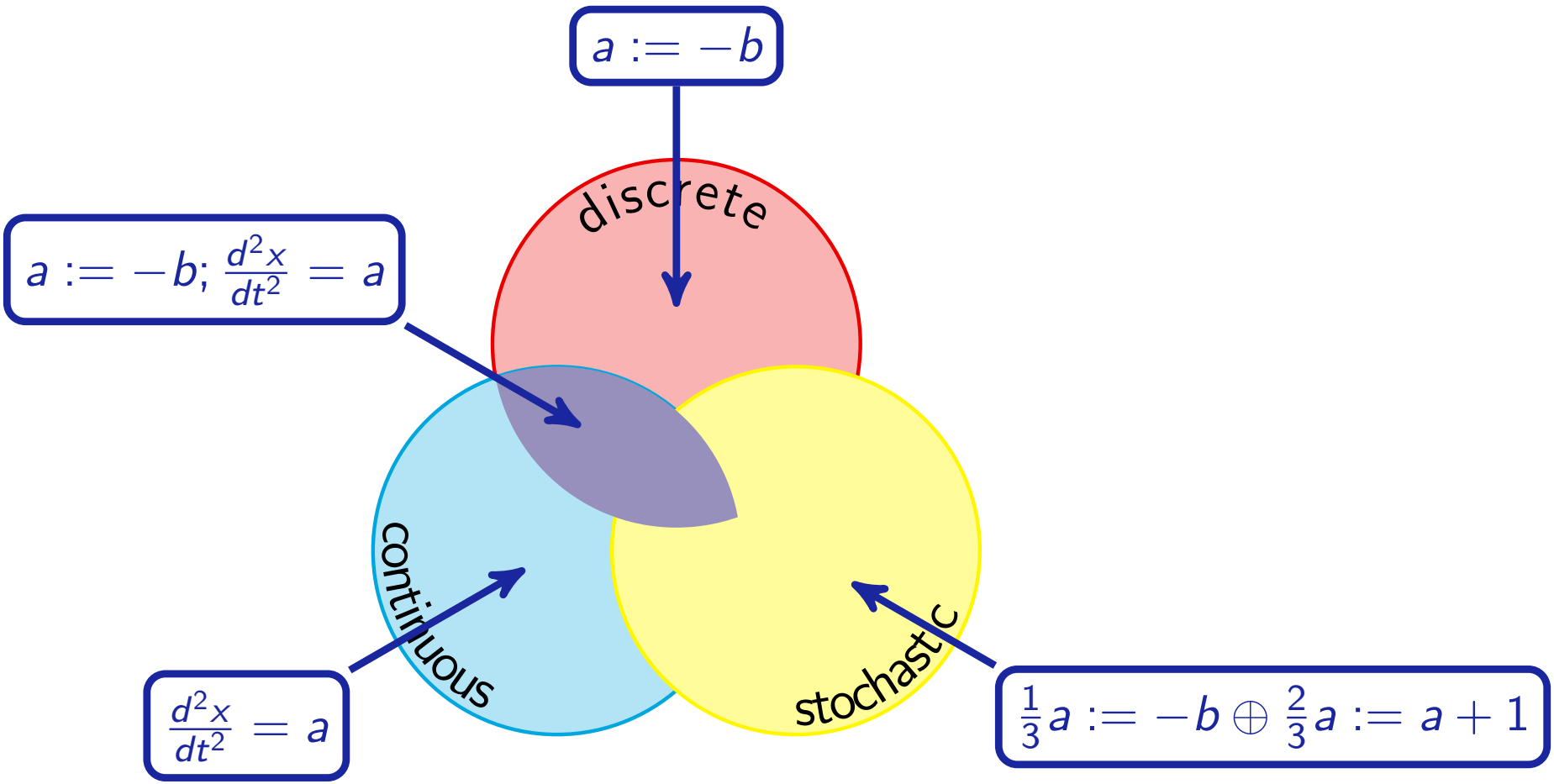
Model for Stochastic Hybrid Systems



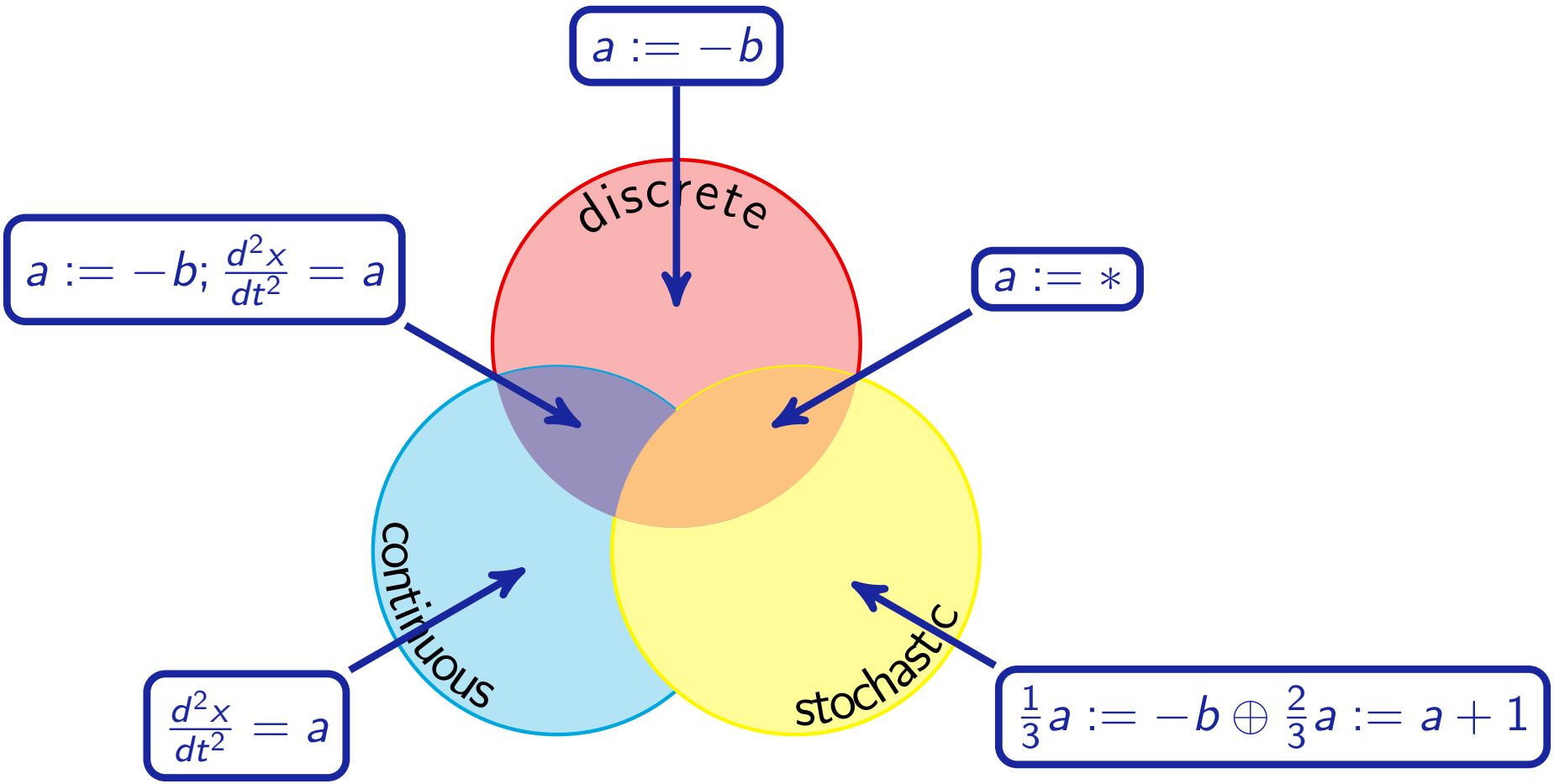
Model for Stochastic Hybrid Systems



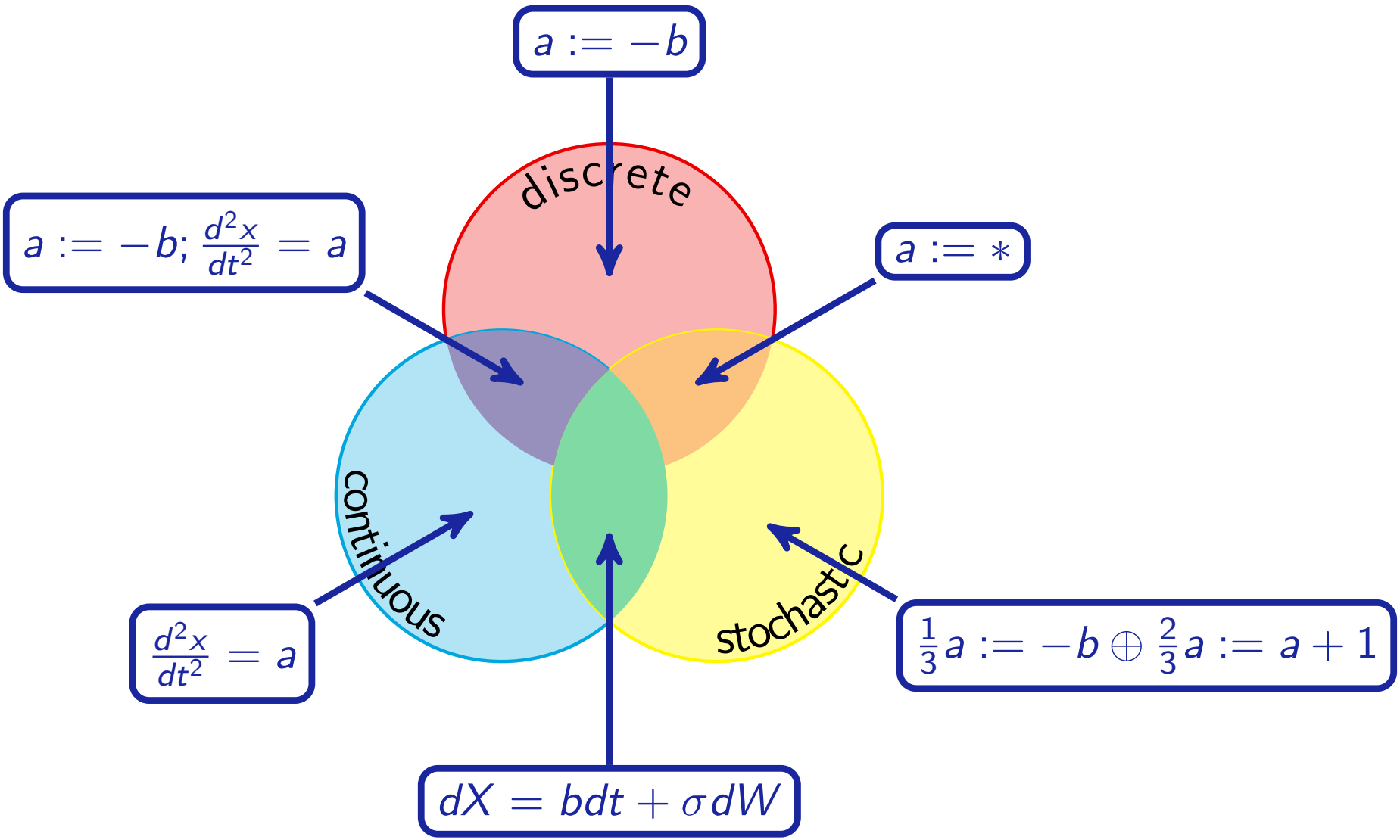
Model for Stochastic Hybrid Systems



Model for Stochastic Hybrid Systems



Model for Stochastic Hybrid Systems



Model for Stochastic Hybrid Systems