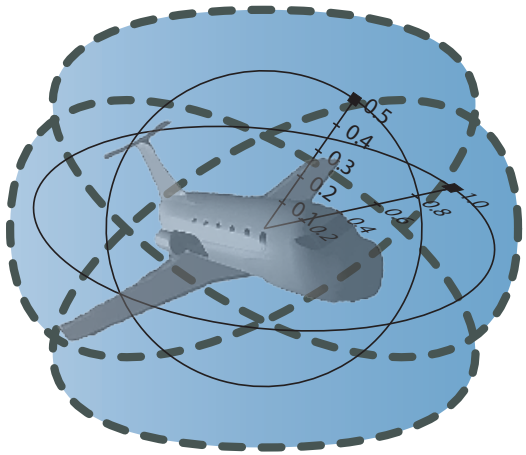
Logical Analysis of Hybrid Systems

Proving Theorems for Complex Dynamics

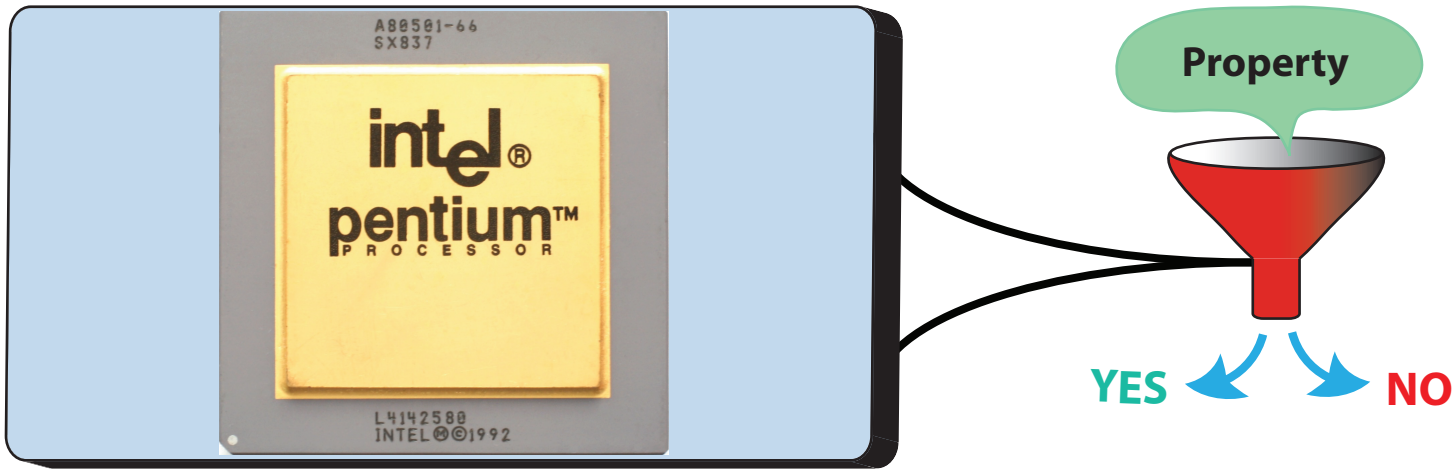
André Platzer

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



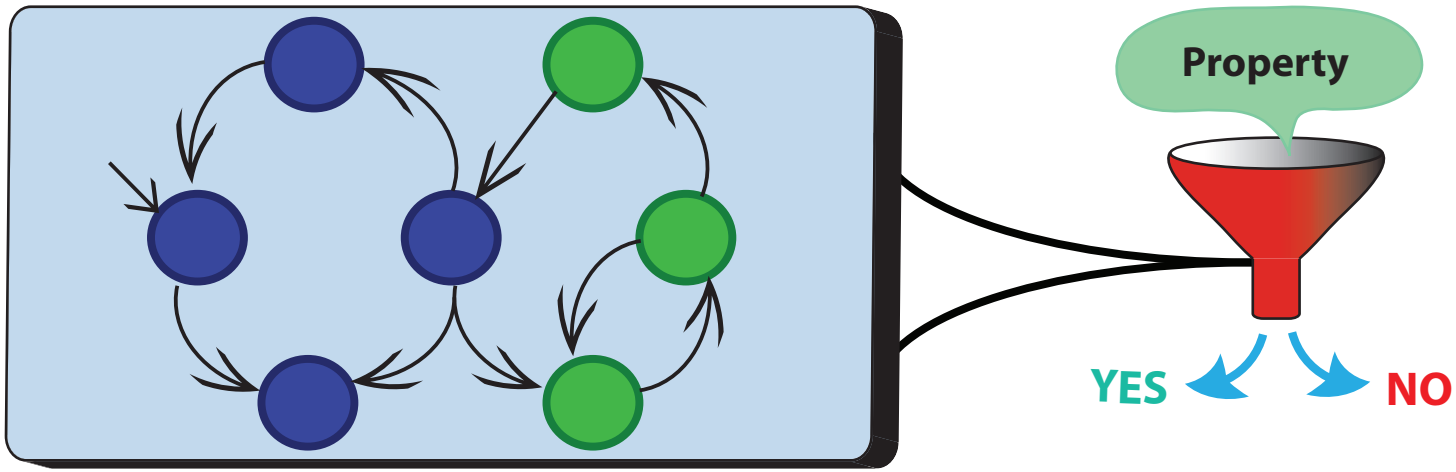
Outline

The Power of Verification



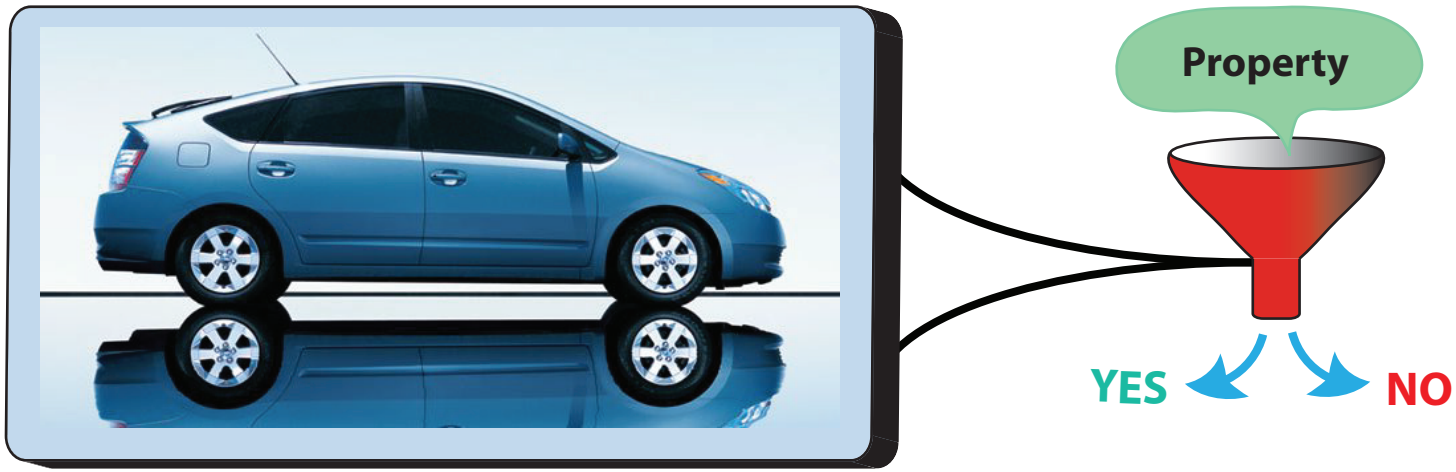
- Formal verification for finite-state digital circuits routine in industry

The Power of Verification



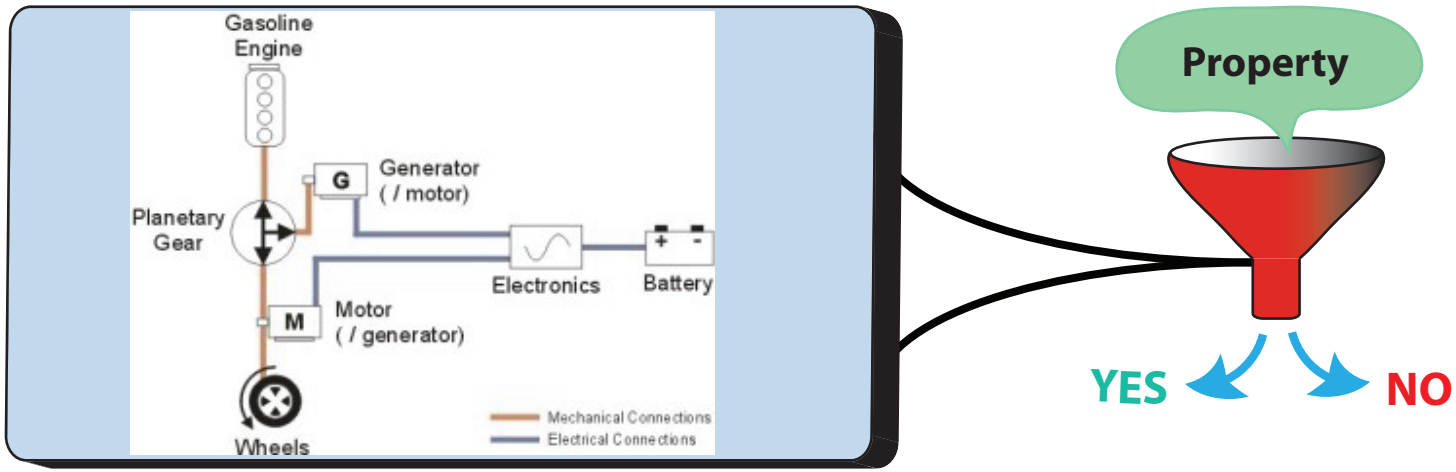
- Formal verification for finite-state digital circuits routine in industry

The Challenge of Verification Today



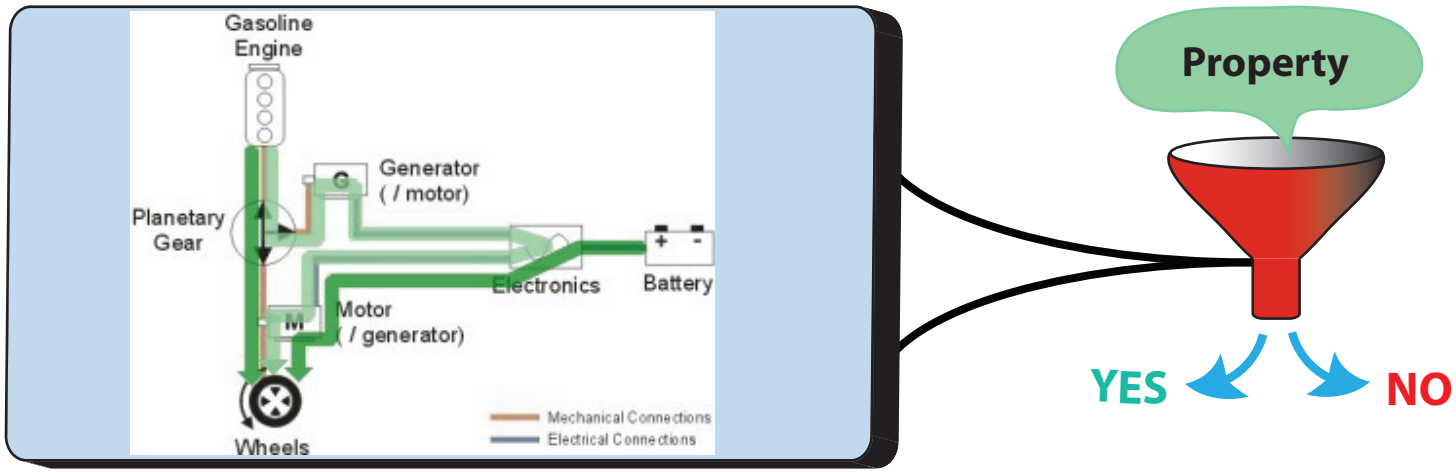
- Safety-critical computers are embedded in cyber-physical systems

The Challenge of Verification Today



- Safety-critical computers are embedded in cyber-physical systems

The Challenge of Verification Today



- Safety-critical computers are embedded in cyber-physical systems

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

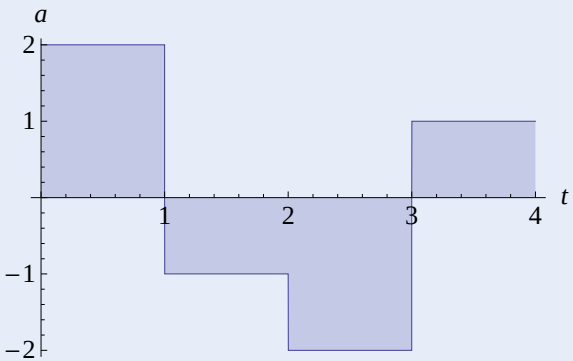
Hybrid Systems Analysis: Train Control

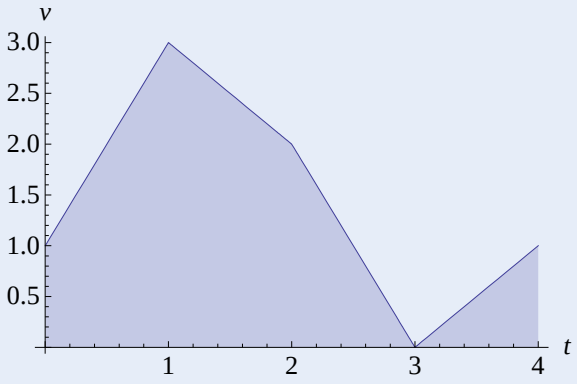
"Time is defined so that motion looks simple" [Henri Poincaré]

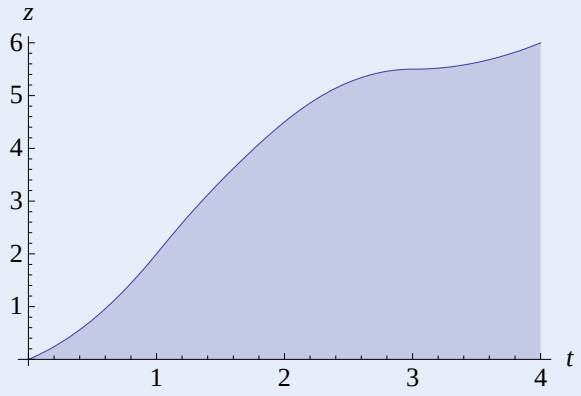
Challenge

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)

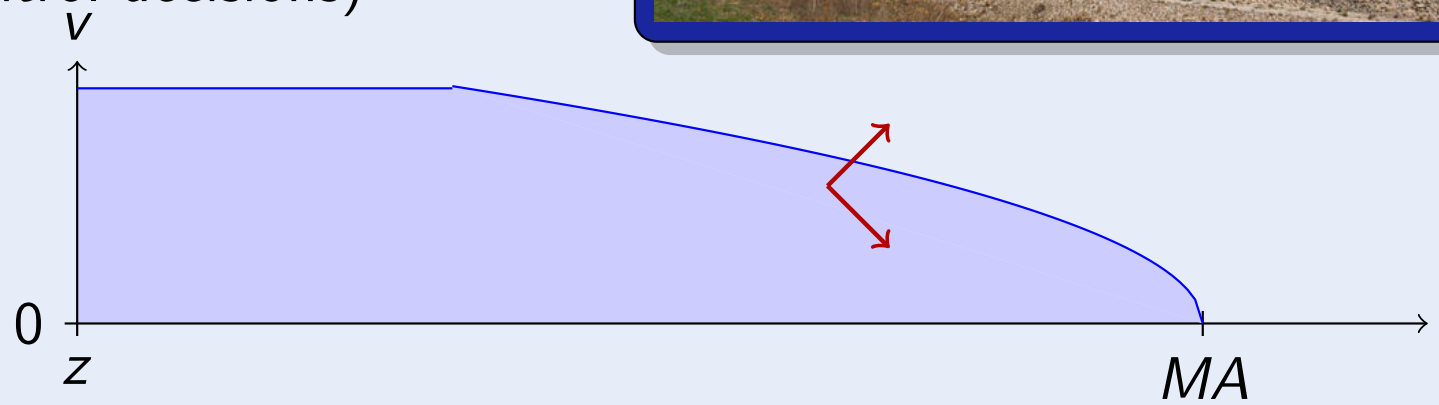
Hybrid Systems Analysis: Train Control

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

Challenge

Hybrid Systems

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



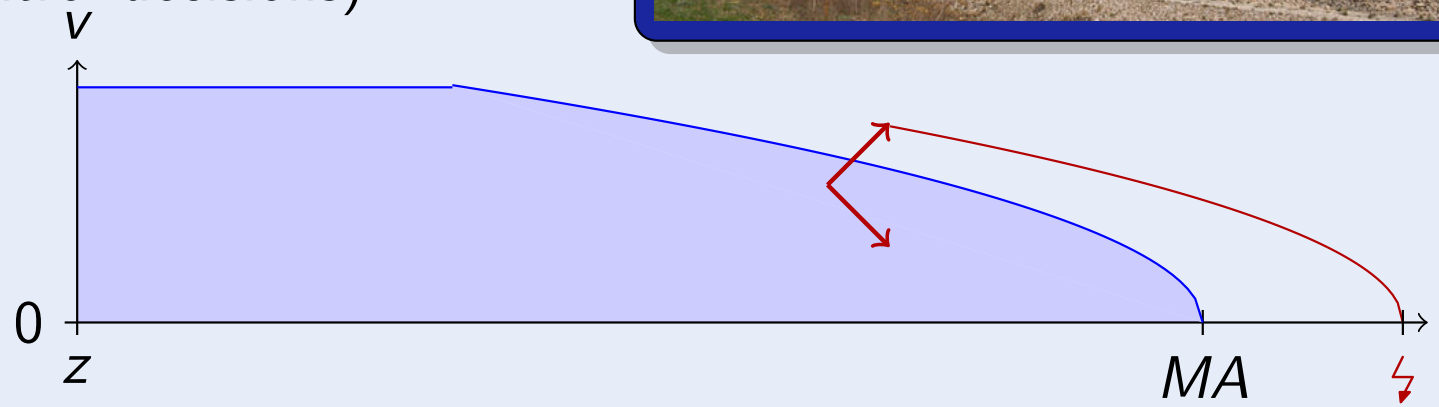
Hybrid Systems Analysis: Train Control

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

Challenge

Hybrid Systems

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



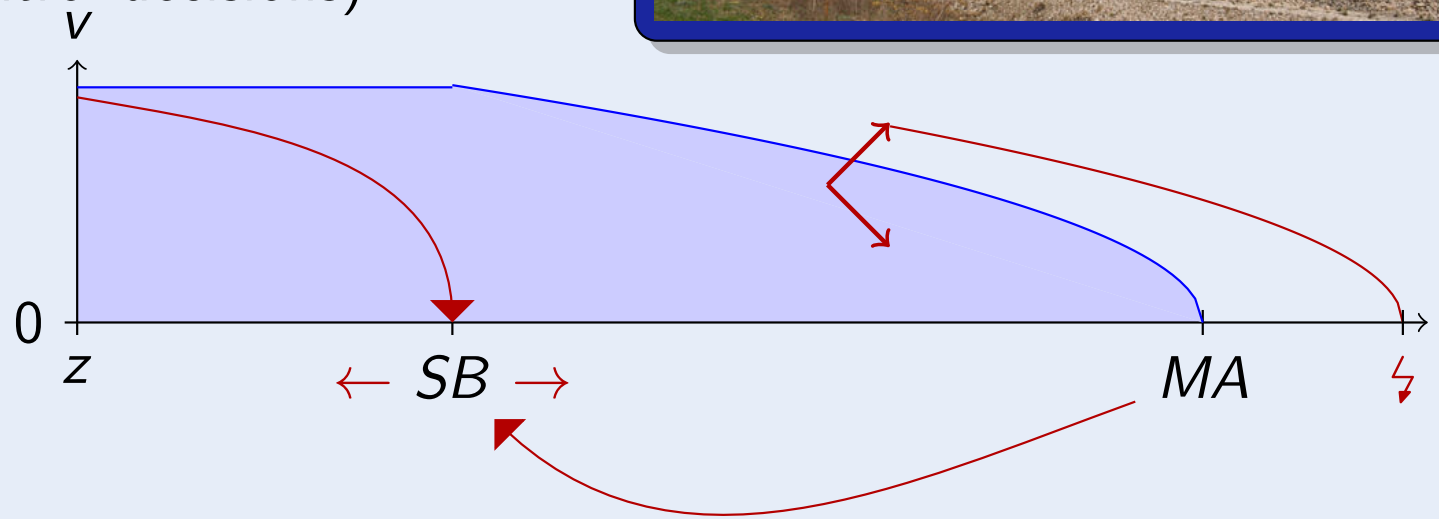
Hybrid Systems Analysis: Train Control

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

Challenge

Hybrid Systems

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



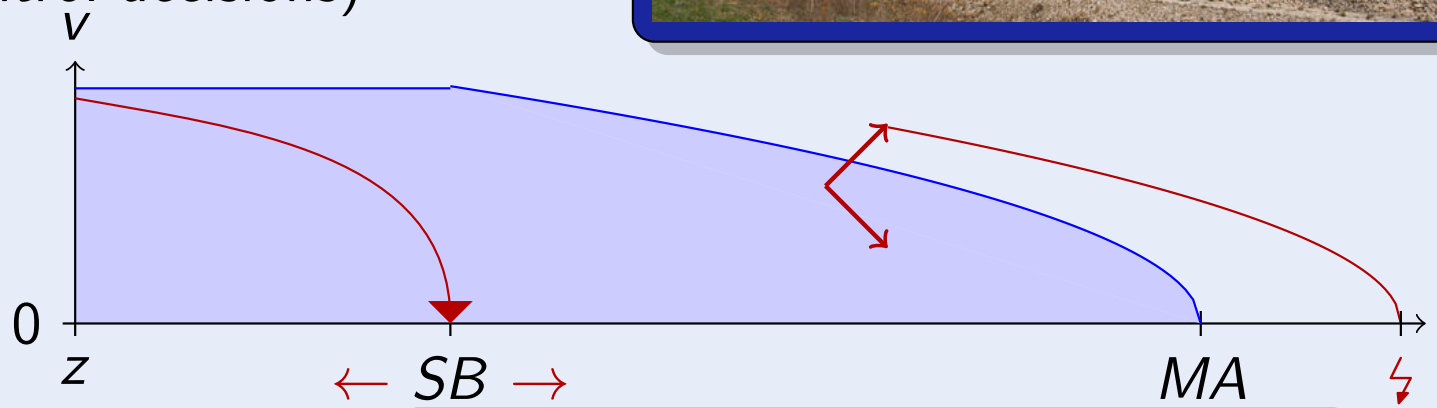
Hybrid Systems Analysis: Train Control

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

Challenge

Hybrid Systems

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



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

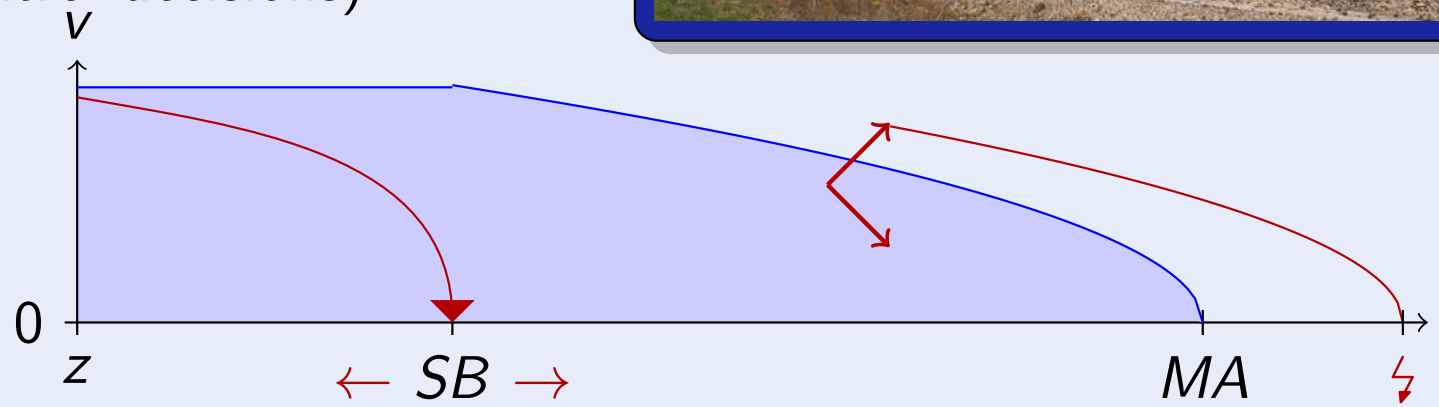
Hybrid Systems Analysis: Train Control

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

Challenge

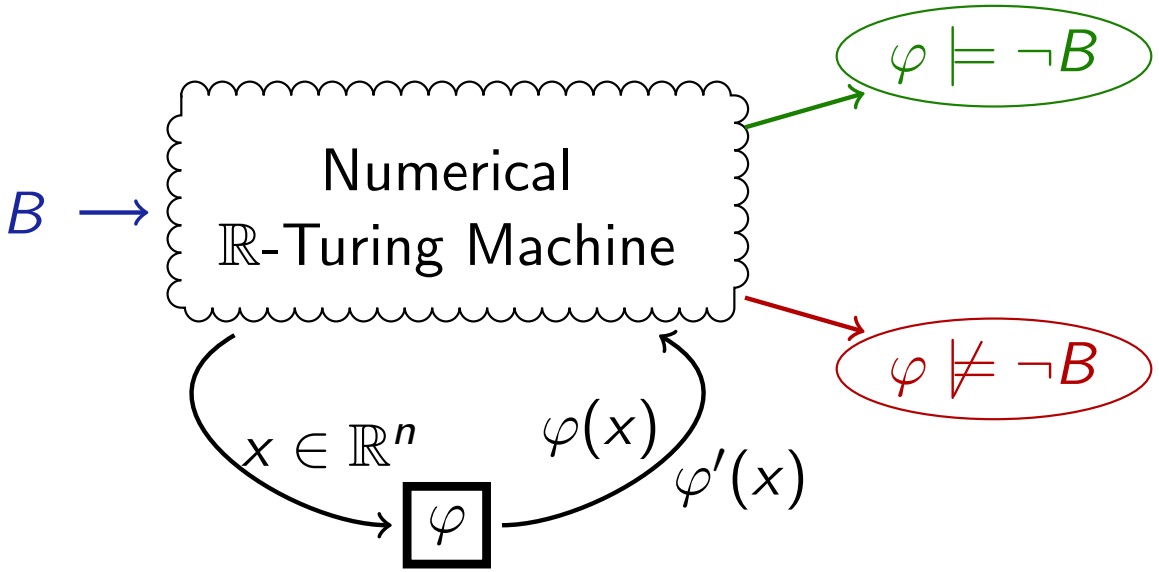
Hybrid Systems

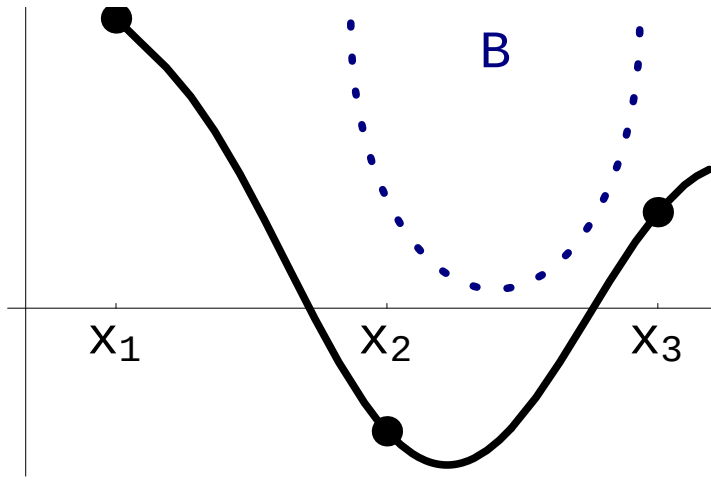
- Continuous dynamics (differential equations)
- Discrete dynamics (control decisions)



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

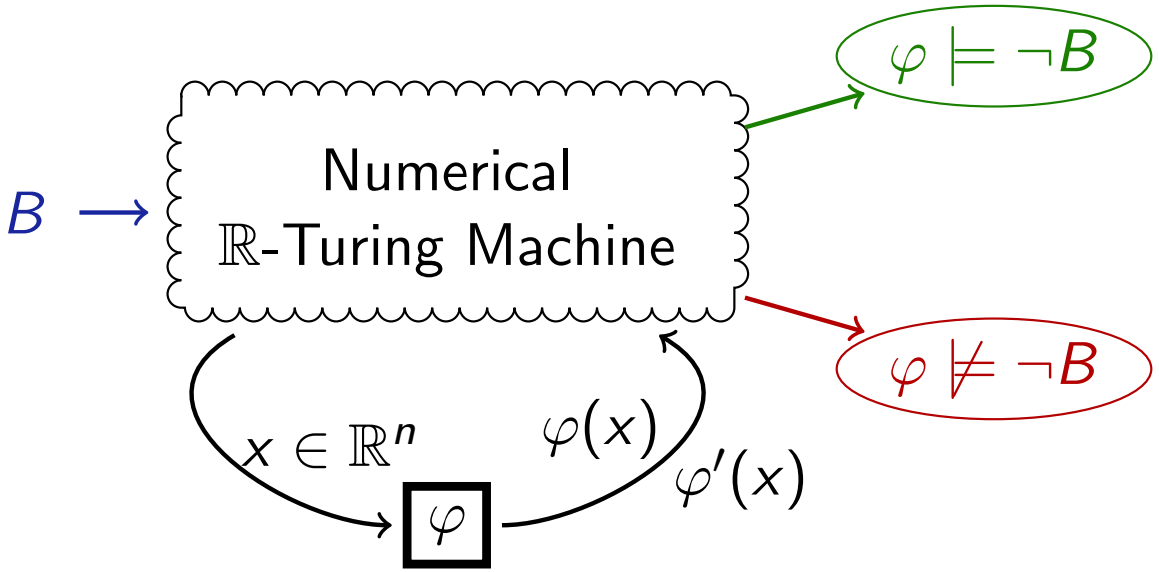
Numerical Image Computation

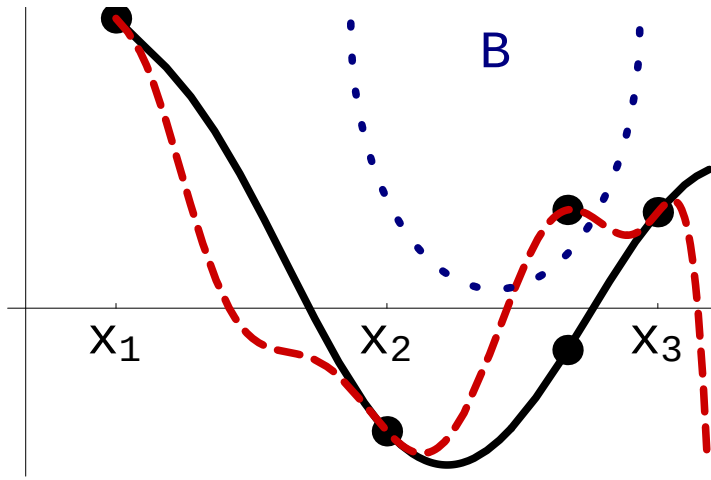




André Platzer and Edmund M. Clarke.
The image computation problem in hybrid systems model checking.
HSCC, vol. 4416 of *LNCS*, 473–486. Springer, 2007.

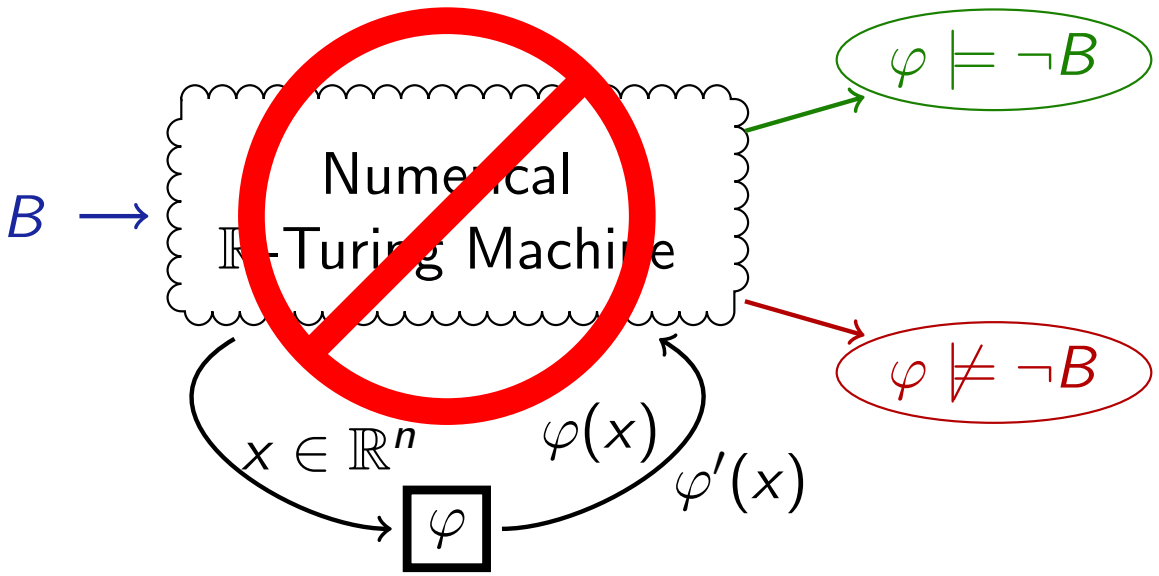
Numerical Image Computation

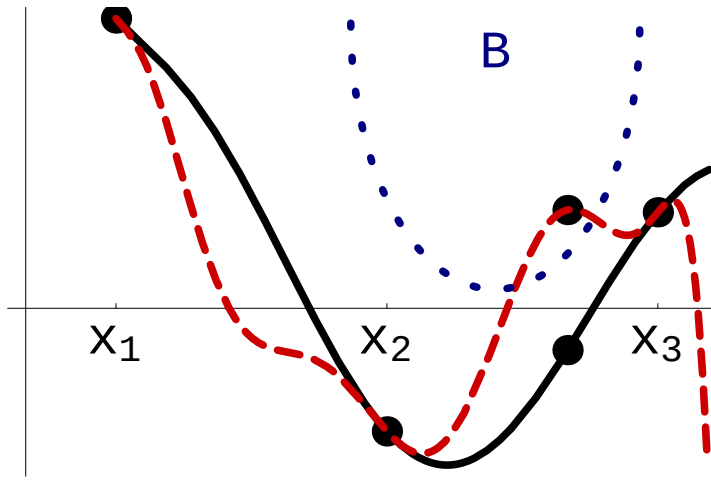




André Platzer and Edmund M. Clarke.
The image computation problem in hybrid systems model checking.
HSCC, vol. 4416 of *LNCS*, 473–486. Springer, 2007.

Numerical Image Computation



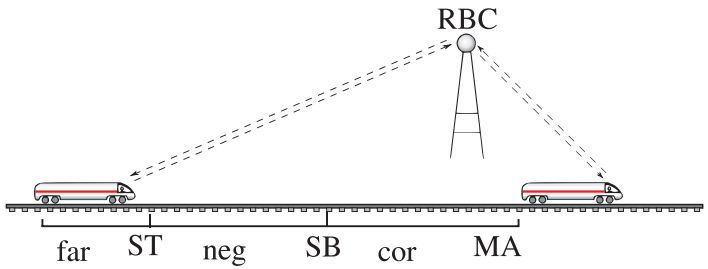


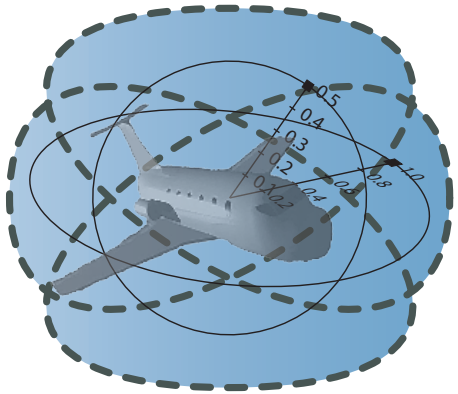
Proposition (Image computation undecidable numerically for...)

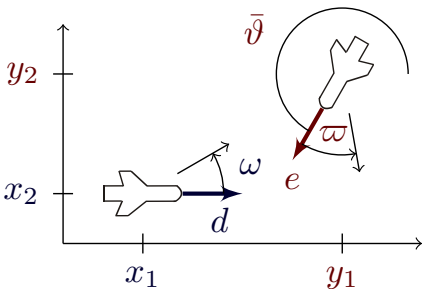
- *arbitrarily effective flow* $\varphi \in C^k(D \subseteq \mathbb{R}^n, \mathbb{R}^m)$*; D, B effective*
- *tolerate error* $\epsilon > 0$ *in decisions*

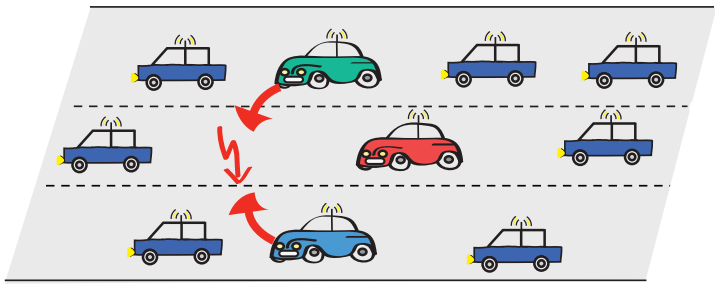
André Platzer and Edmund M. Clarke.
The image computation problem in hybrid systems model checking.
HSCC, vol. 4416 of *LNCS*, 473–486. Springer, 2007.

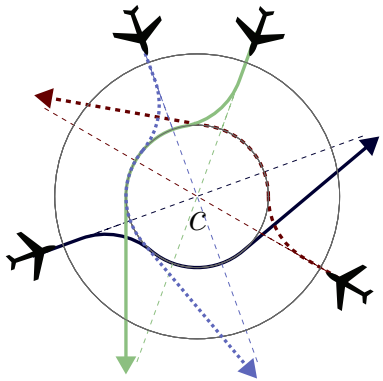
Hybrid Systems Analysis is Important for . . .

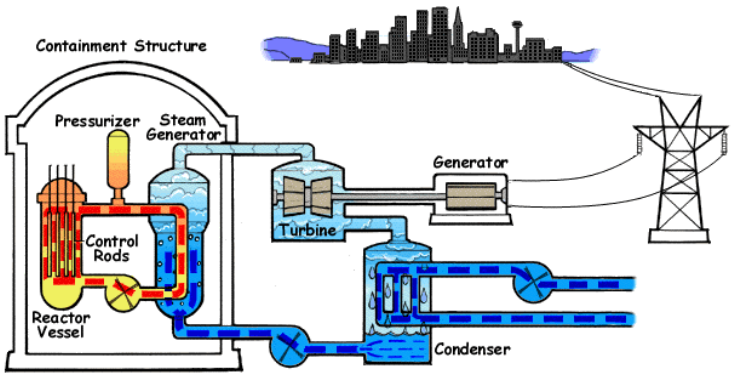


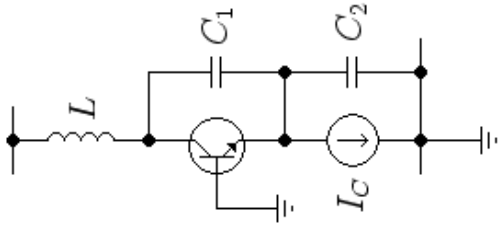












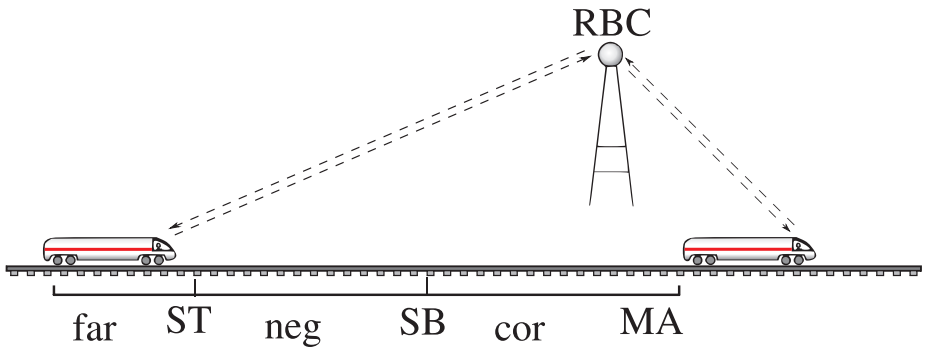
Outline

Outline (Conceptual Approach)

dℒ Design: The Logic of Hybrid Systems

differential dynamic logic

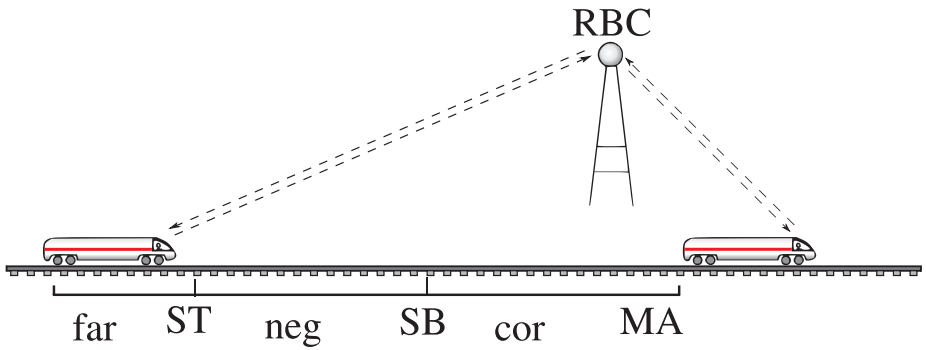
$$\mathsf{d}\mathcal{L} = \mathsf{DL} + \mathsf{HP}$$

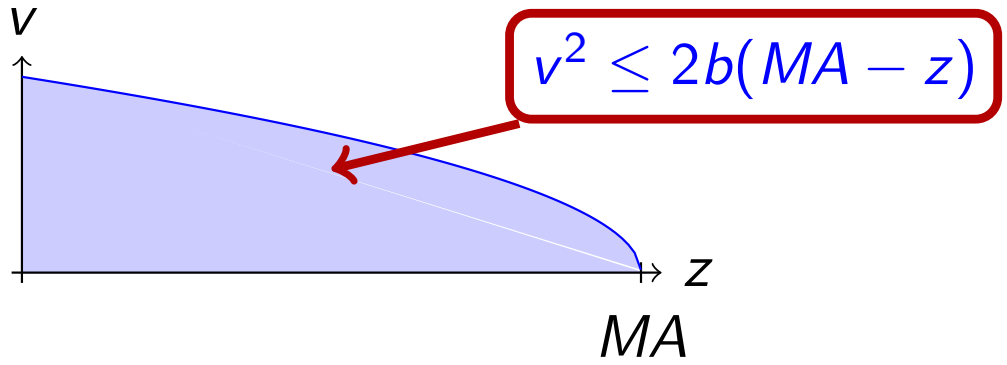


dℒ Design: Regions in First-Order Logic

differential dynamic logic

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

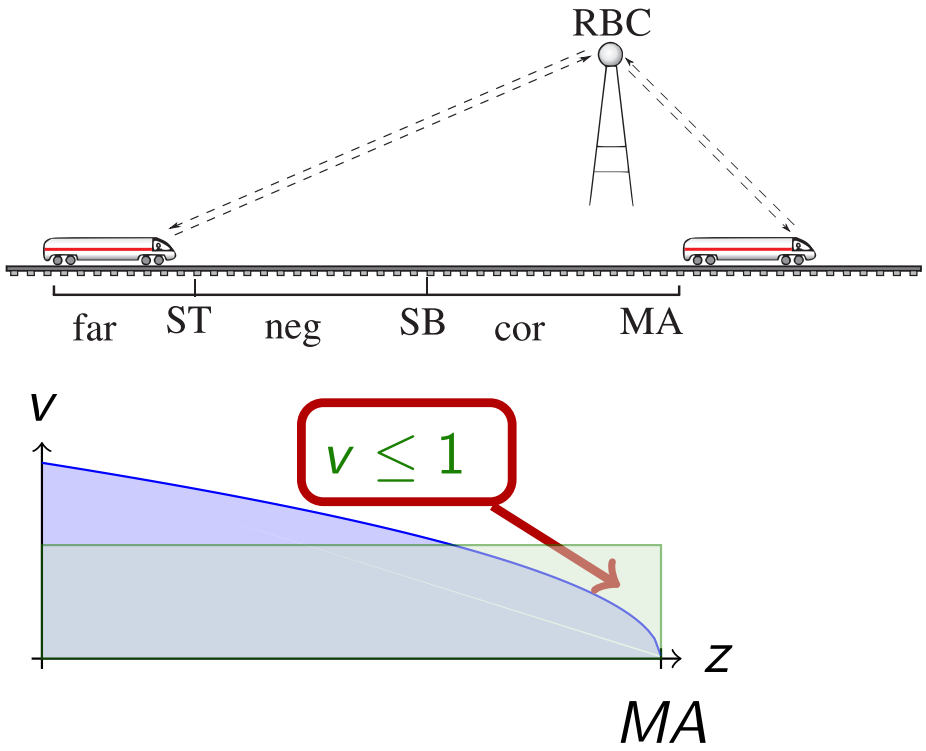




dℒ Design: Regions in First-Order Logic

differential dynamic logic

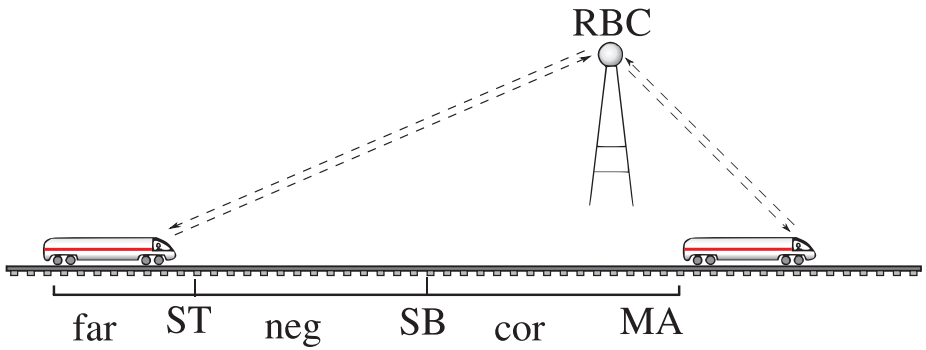
$d\mathcal{L} = \text{FOL}_{\mathbb{R}}$

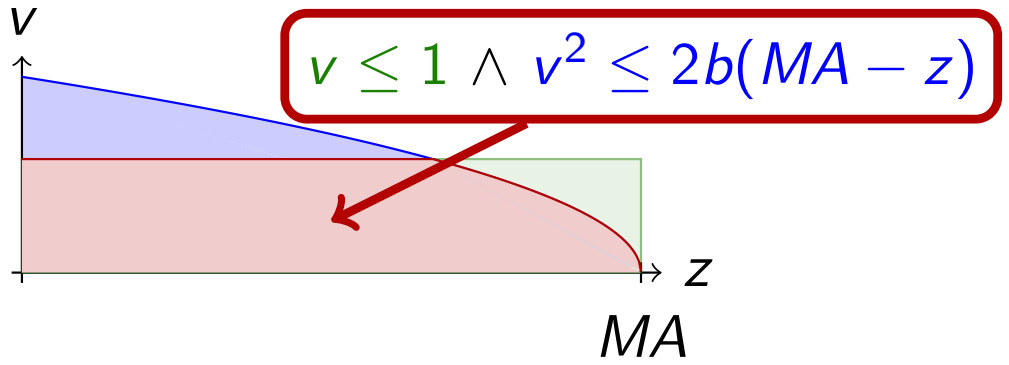


dℒ Design: Regions in First-Order Logic

differential dynamic logic

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

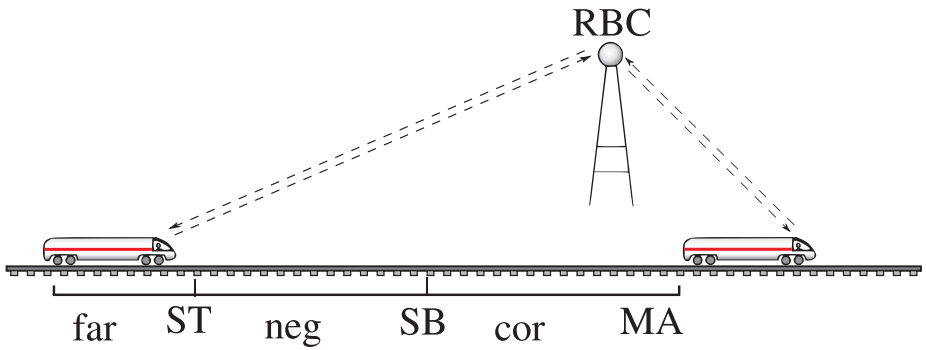


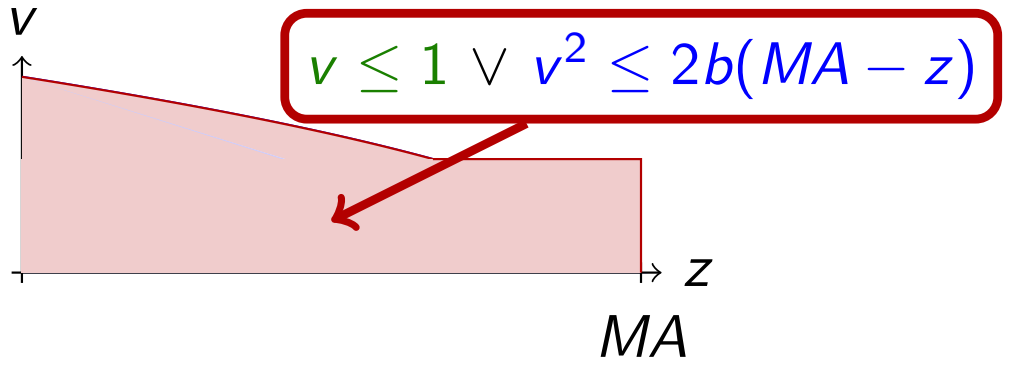


dℒ Design: Regions in First-Order Logic

differential dynamic logic

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





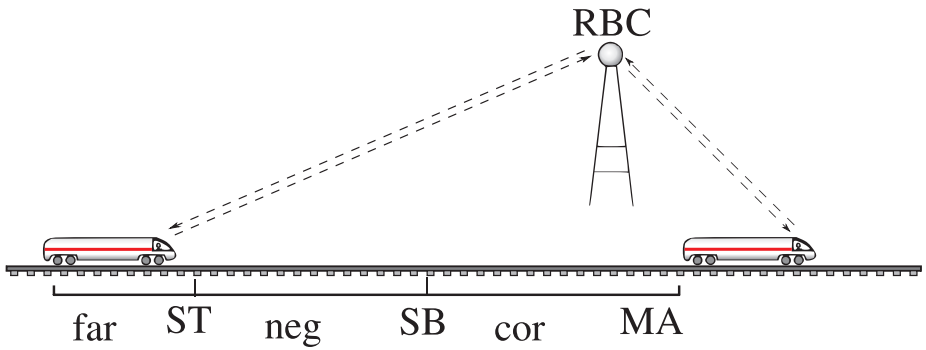
dℒ Design: Regions in First-Order Logic

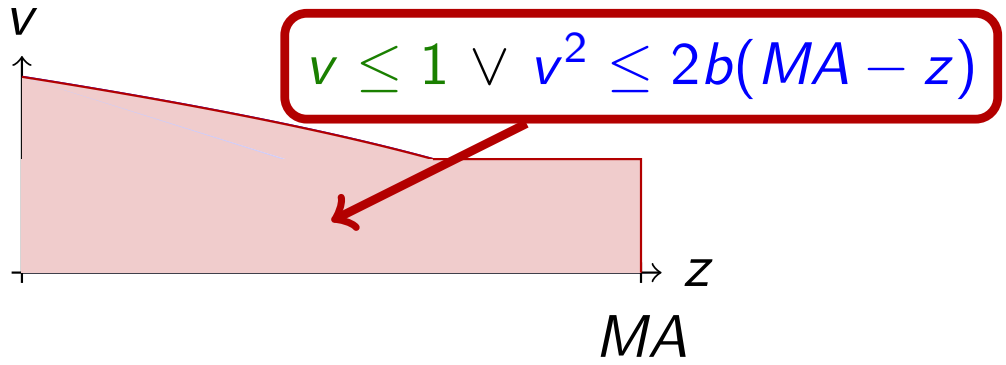
differential dynamic logic

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

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

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

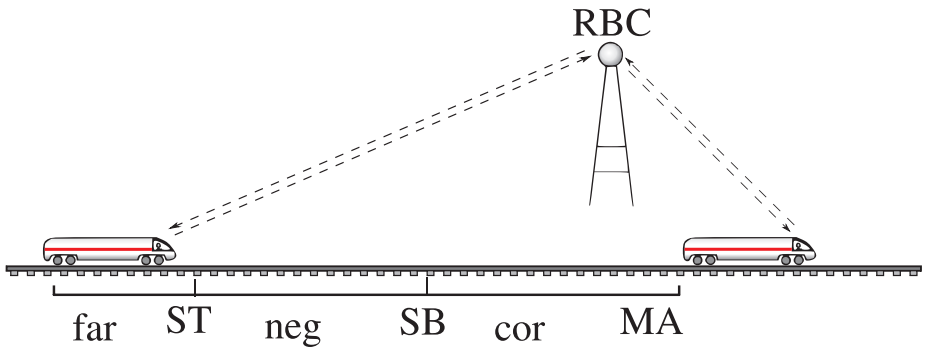


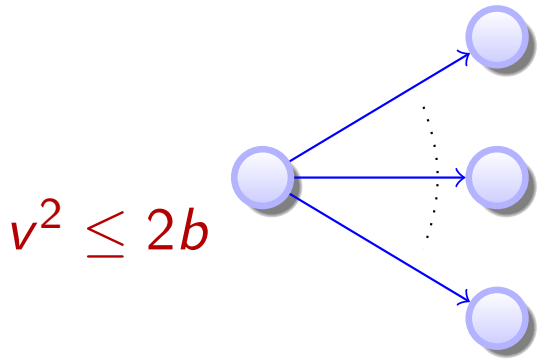


d$\mathcal{L}$ Design: State Transitions in Dynamic Logic

differential dynamic logic

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

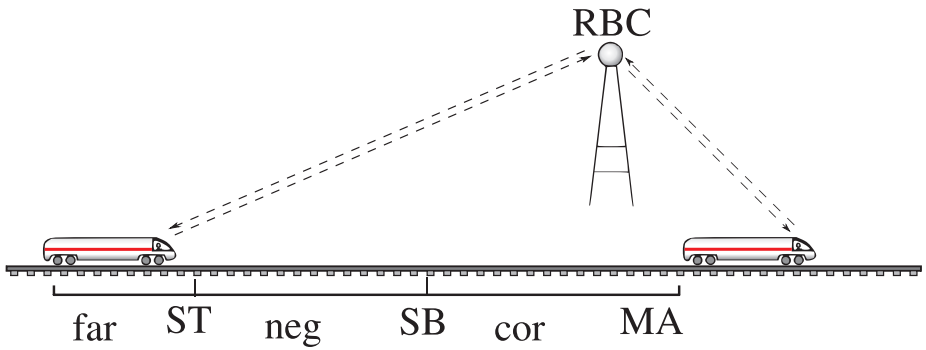


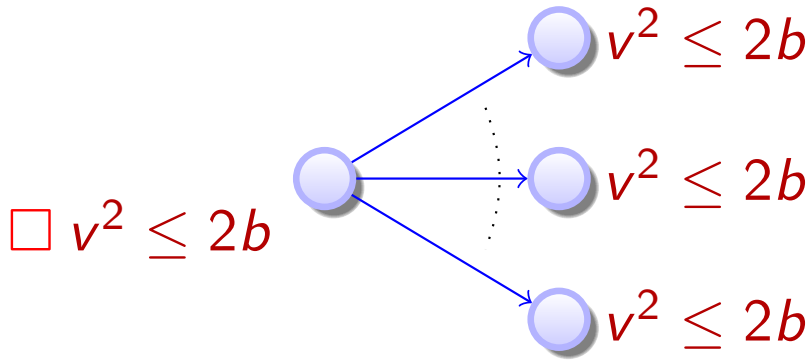


dℒ Design: State Transitions in Dynamic Logic

differential dynamic logic

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

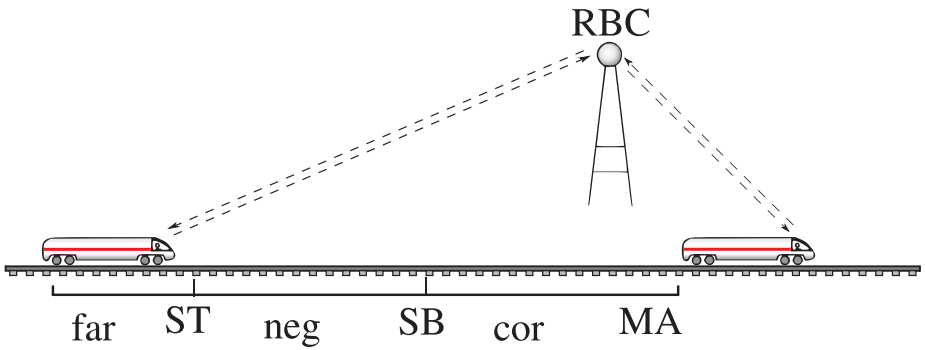


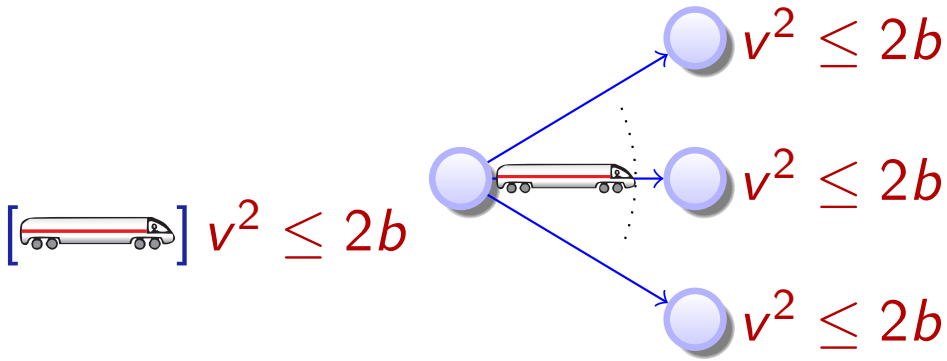


dℒ Design: State Transitions in Dynamic Logic

differential dynamic logic

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

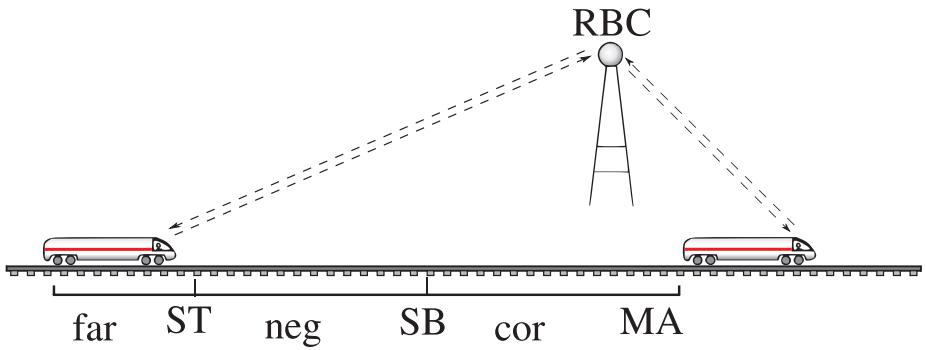




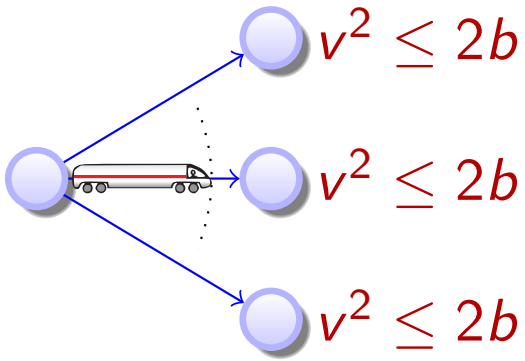
dℒ Design: Hybrid Programs as Uniform Model

differential dynamic logic

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



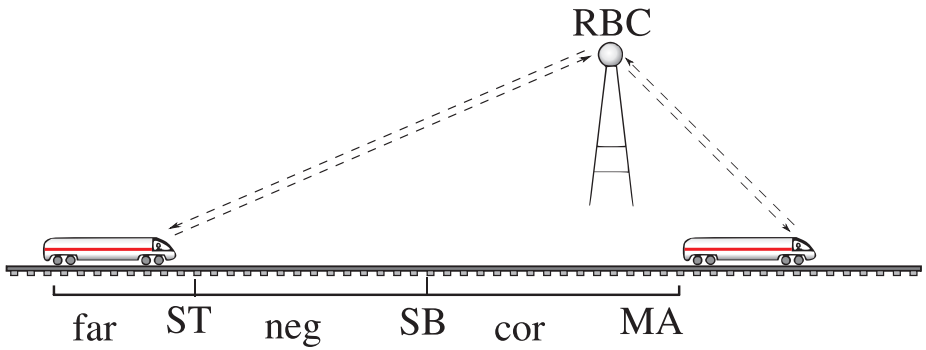
$$[z'' = a]\, v^2 \leq 2b$$



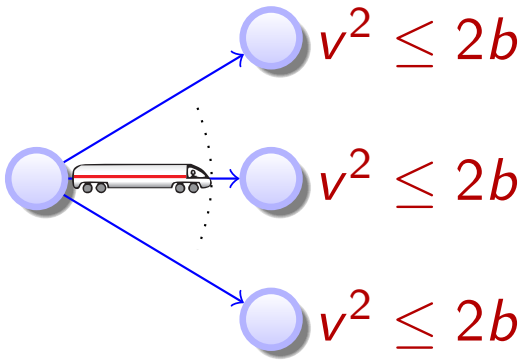
dℒ Design: Hybrid Programs as Uniform Model

differential dynamic logic

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



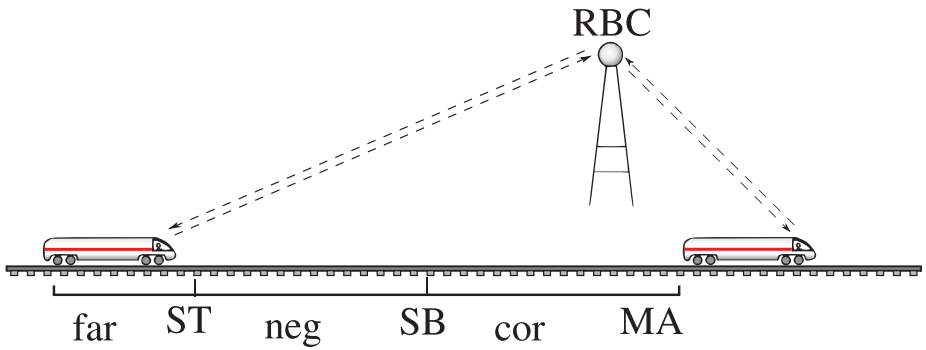
$$[\texttt{if}(z > SB)\, a := -b;\ z'' = a]\, v^2 \leq 2b$$



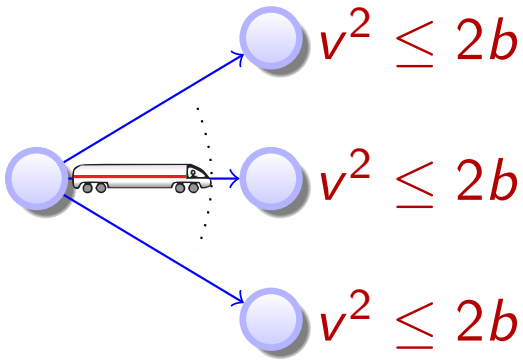
dℒ Design: Hybrid Programs as Uniform Model

differential dynamic logic

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



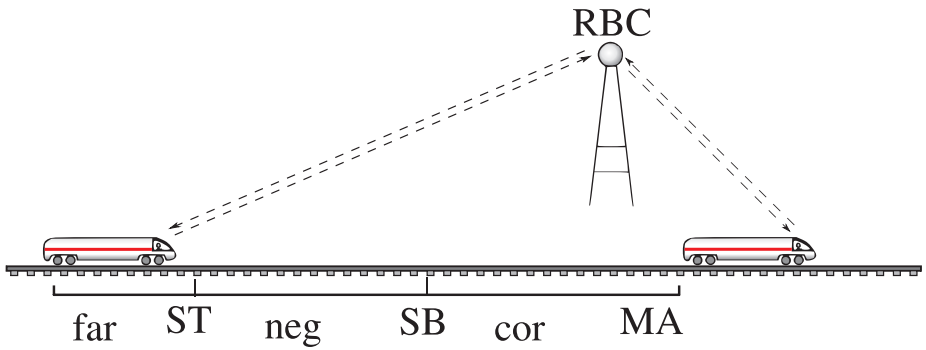
$$[\underbrace{\texttt{if}(z > SB)\, a := -b;\ z'' = a}_{\text{hybrid program}}]\, v^2 \leq 2b$$



dℒ Design: Hybrid Programs as Uniform Model

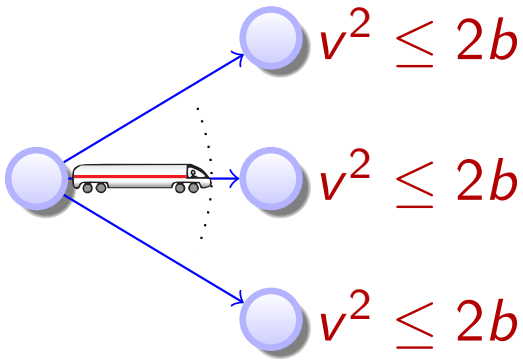
differential dynamic logic

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



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

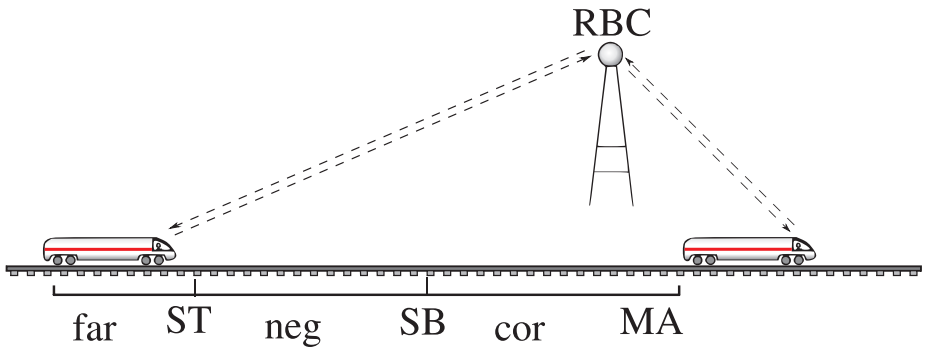
Initial condition



dℒ Design: Hybrid Programs as Uniform Model

differential dynamic logic

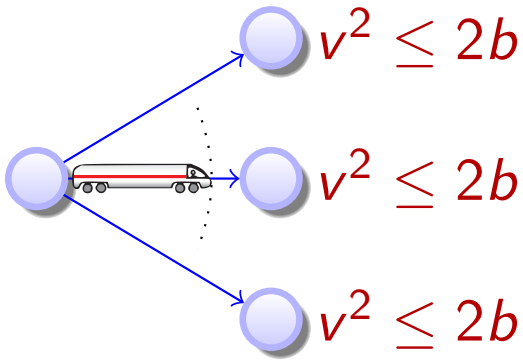
$$\mathrm{d}\mathcal{L} = \mathrm{FOL}_{\mathbb{R}} + \mathrm{DL} + \mathrm{HP}$$



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

Initial condition

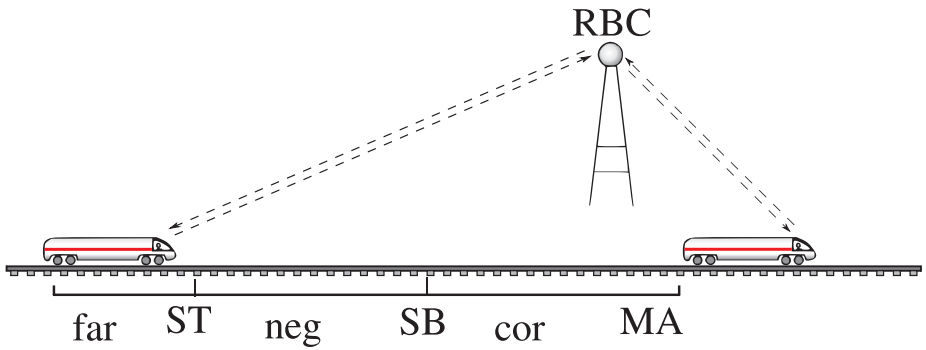
System dynamics

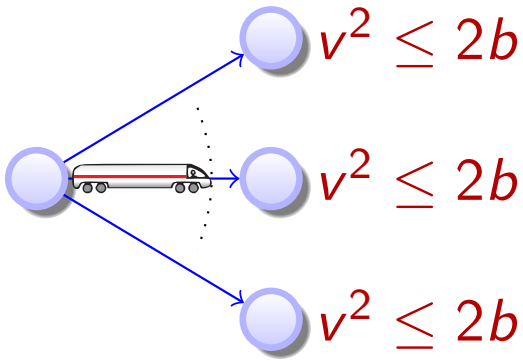


dℒ Design: Hybrid Programs as Uniform Model

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

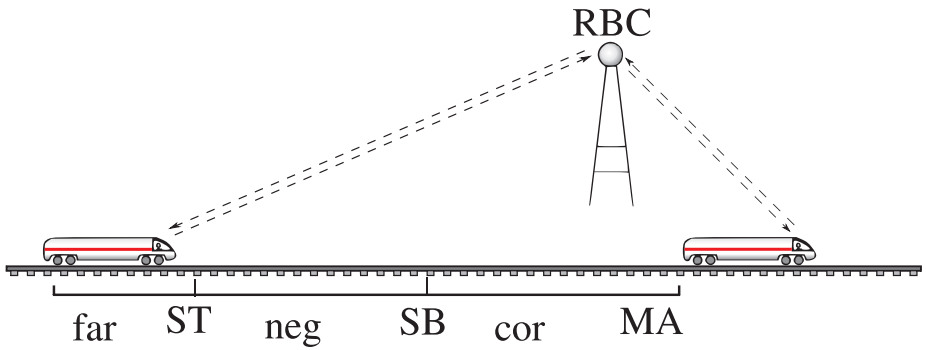
System dynamics

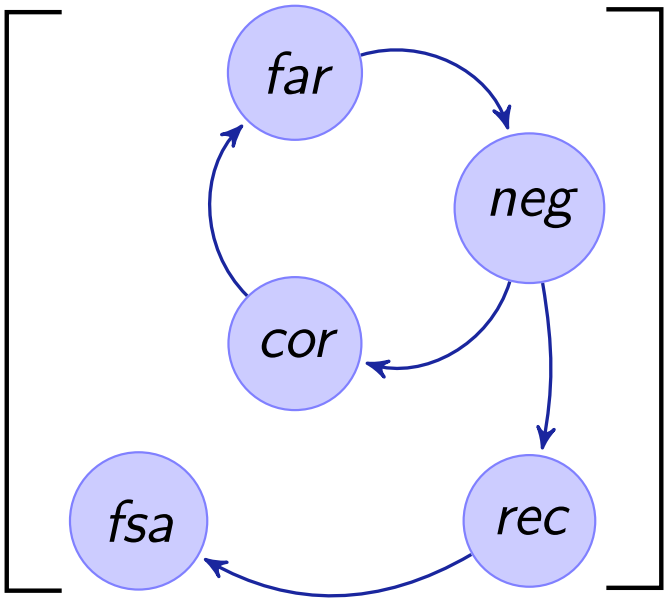
Post condition

dℒ Design: What about Hybrid Automata?

differential dynamic logic

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



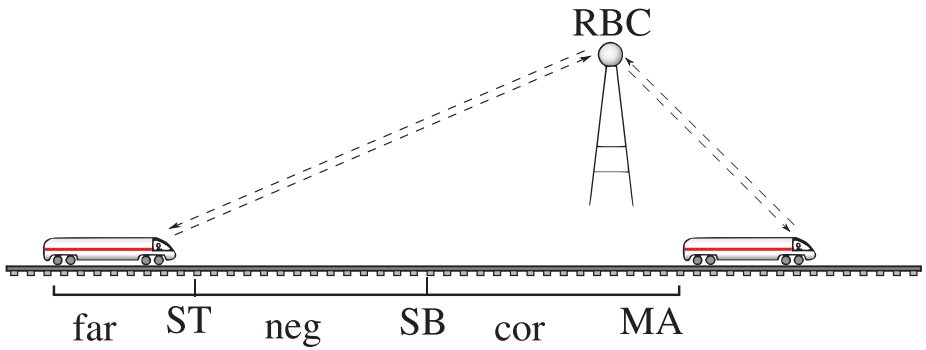


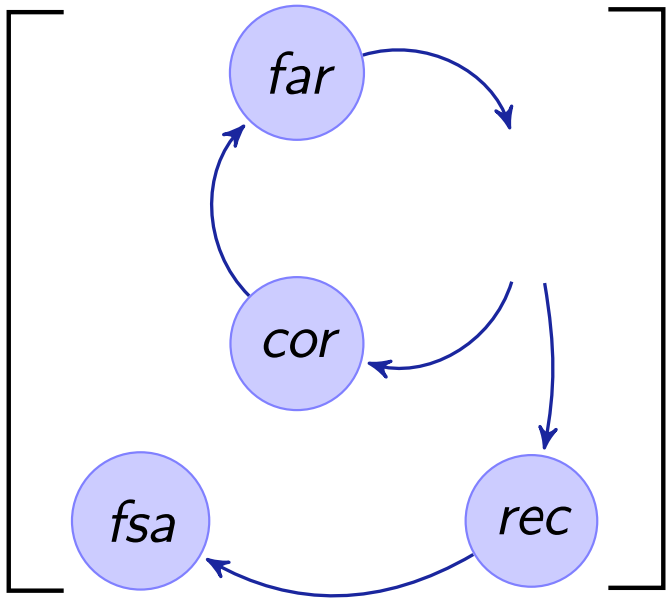
$v^2 \leq 2b\,..$

dℒ Design: What about Hybrid Automata?

differential dynamic logic

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



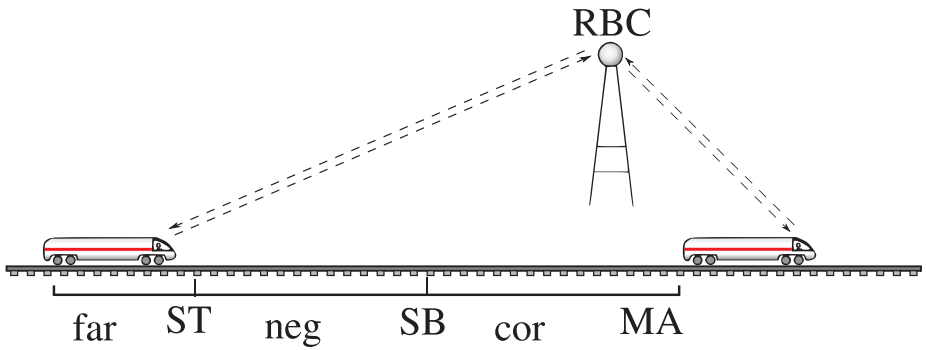


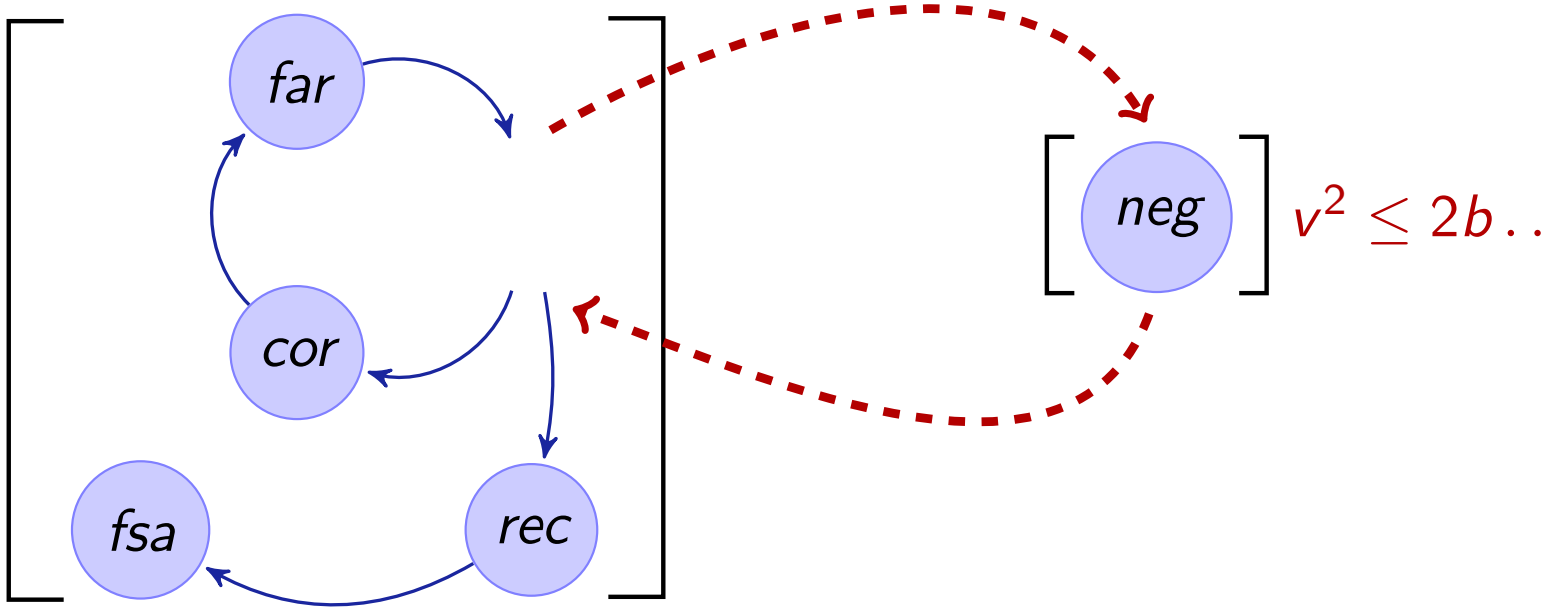
$$[\,\textit{neg}\,]\; v^2 \leq 2b\,..$$

d$\mathcal{L}$ Design: What about Hybrid Automata?

differential dynamic logic

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

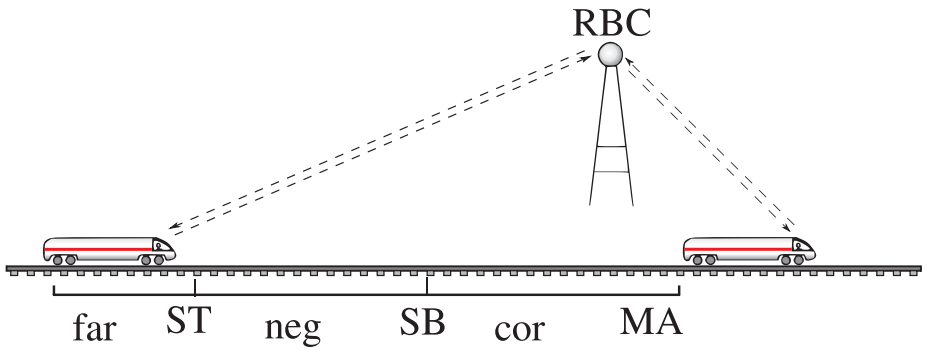


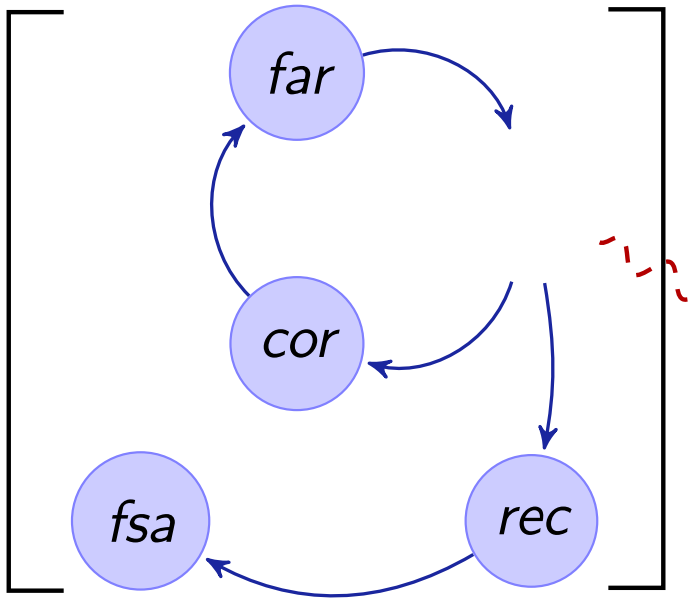


dℒ Design: What about Hybrid Automata?

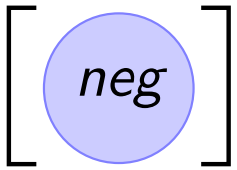
differential dynamic logic

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





not compositional



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

Differential Dynamic Logic dℒ: Syntax

Definition (Hybrid program α)

$x' = f(x)$	(continuous evolution)	
$x := f(x)$	(discrete jump)	jump & test
$\texttt{if}(\chi)\,\alpha\,\texttt{else}\,\beta$	(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

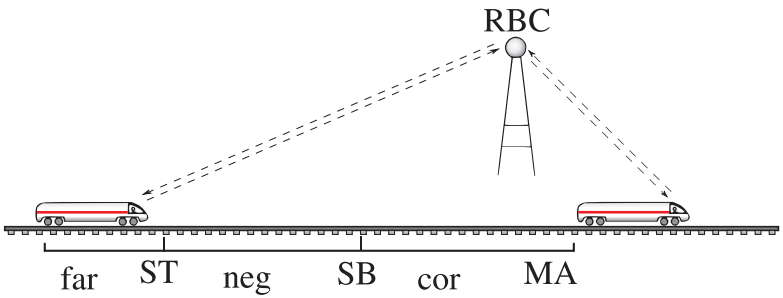
Definition (Hybrid program α)

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

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

$$ctrl \equiv \texttt{if}\ (MA - z < SB)\ \texttt{then}\ a := -b\ \texttt{else}\ a := \ldots$$

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



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

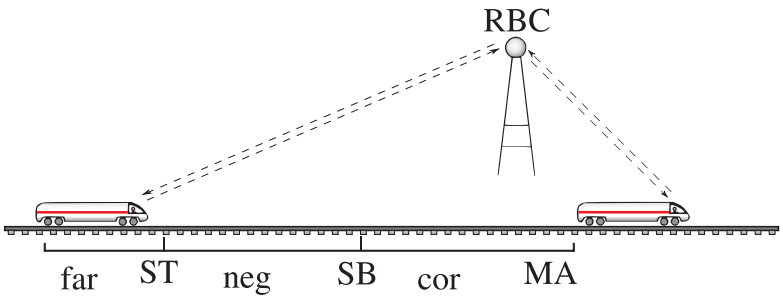
Definition (Hybrid program α)

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

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

$$ctrl \equiv \texttt{if}\ (MA - z < SB)\ \texttt{then}\ a := -b$$
$$\texttt{else}\ a := \ldots$$

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



Differential Dynamic Logic dℒ: Syntax

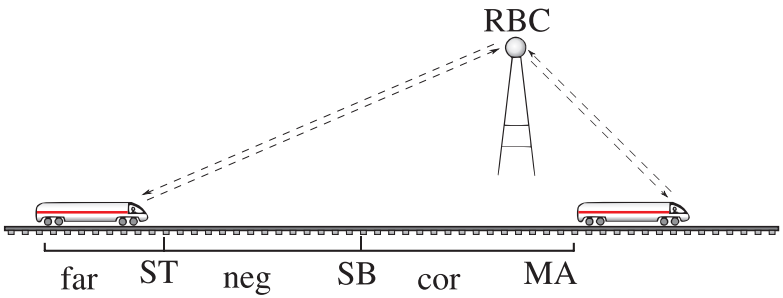
Definition (Hybrid program α)

$x' = f(x) \wedge \chi$	(continuous evolution)	
$x := f(x)$	(discrete jump)	jump & test
$\texttt{if}(\chi)\,\alpha\,\texttt{else}\,\beta$	(conditional execution)	jump & test
$\alpha;\beta$	(seq. composition)	Kleene algebra
$\alpha \cup \beta$	(nondet. choice)	Kleene algebra
α^*	(nondet. repetition)	Kleene algebra

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

$$ctrl \equiv \texttt{if}\ (MA - z < SB)\ \texttt{then}\ a := -b$$
$$\texttt{else}\ a := \ldots$$

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

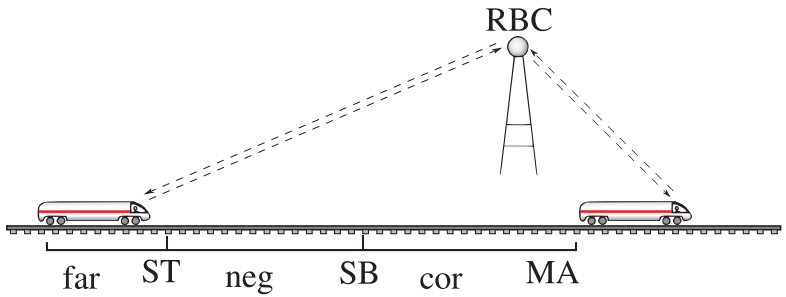


Differential Dynamic Logic d$\mathcal{L}$: Syntax

Definition (Formulas ϕ)

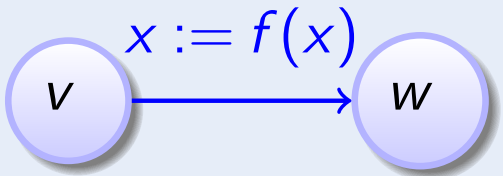
$\neg, \wedge, \vee, \rightarrow,\ \forall x\,, \exists x\,,\ \ =, \leq,\ +, \cdot$ ($\mathbb{R}$-first-order part)
$[\alpha]\phi,\quad \langle\alpha\rangle\phi$ (dynamic part)

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

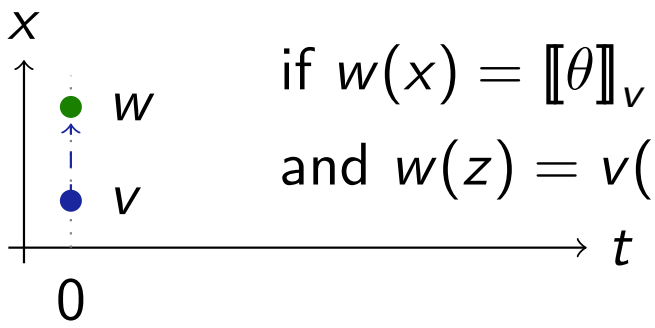


All trains respect *MA*
RBC partitions *MA*
$\Rightarrow$ system collision free

Definition (Hybrid programs α: transition semantics)

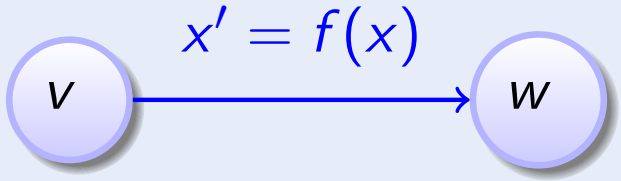


▸ Details ▸▸

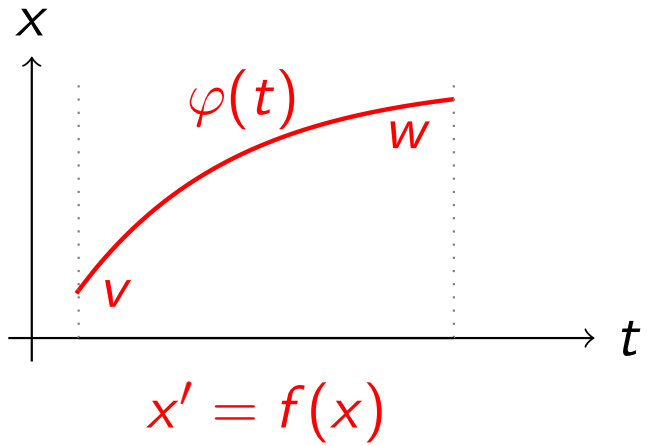


if $w(x) = [\![\theta]\!]_v$
and $w(z) = v(z)$ for $z \neq x$

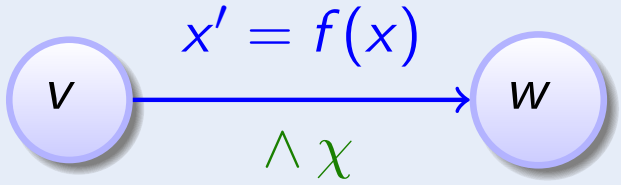
Definition (Hybrid programs α: transition semantics)



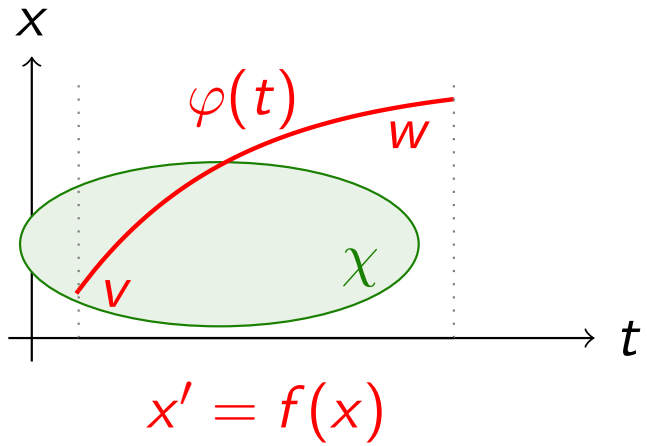
▸ Details ▸▸



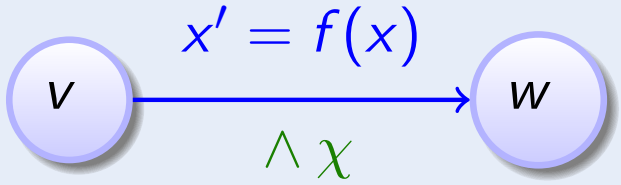
Definition (Hybrid programs α: transition semantics)



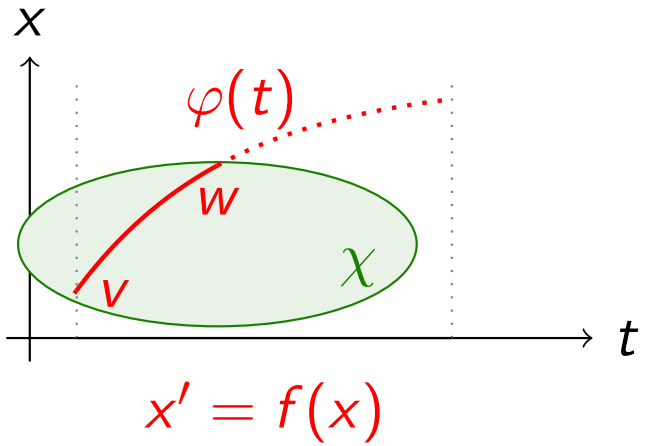
▸ Details ▸▸



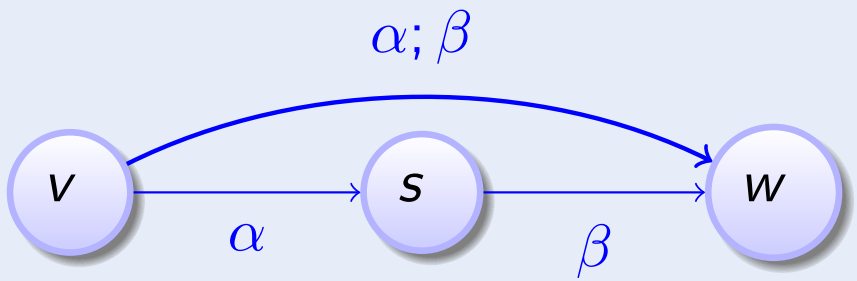
Definition (Hybrid programs α: transition semantics)



▸ Details ▸▸

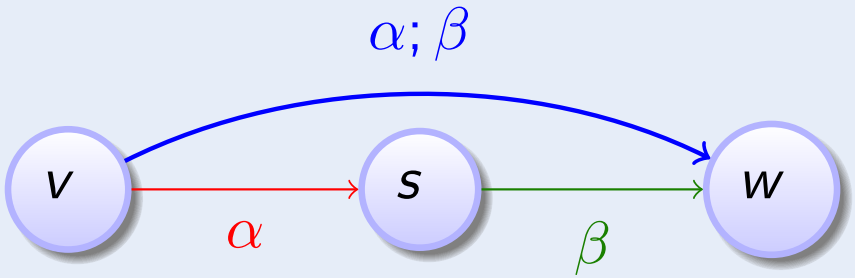


Definition (Hybrid programs α: transition semantics)

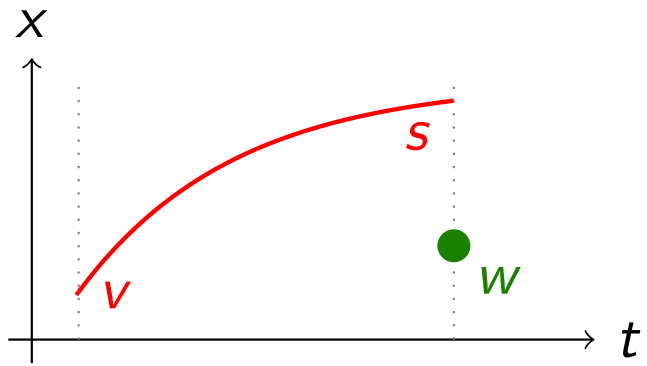


▸ Details ⏩

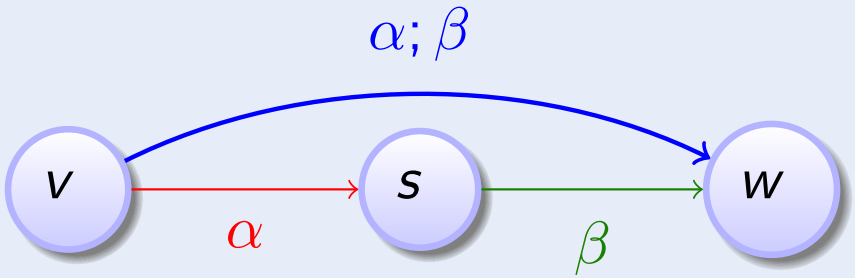
Definition (Hybrid programs α: transition semantics)



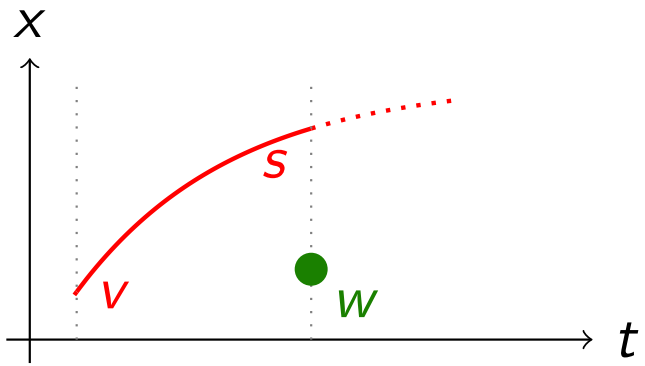
▸ Details ▸▸



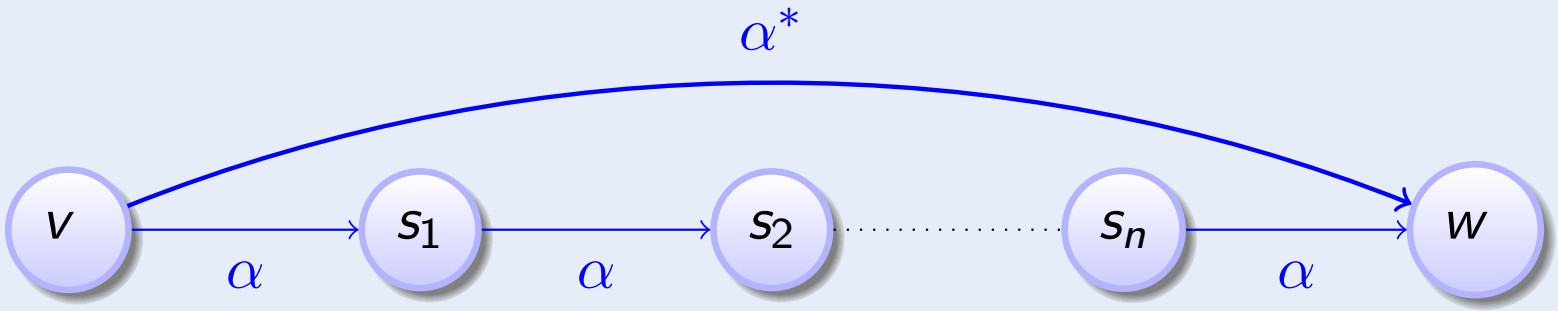
Definition (Hybrid programs α: transition semantics)



▸ Details ▸▸

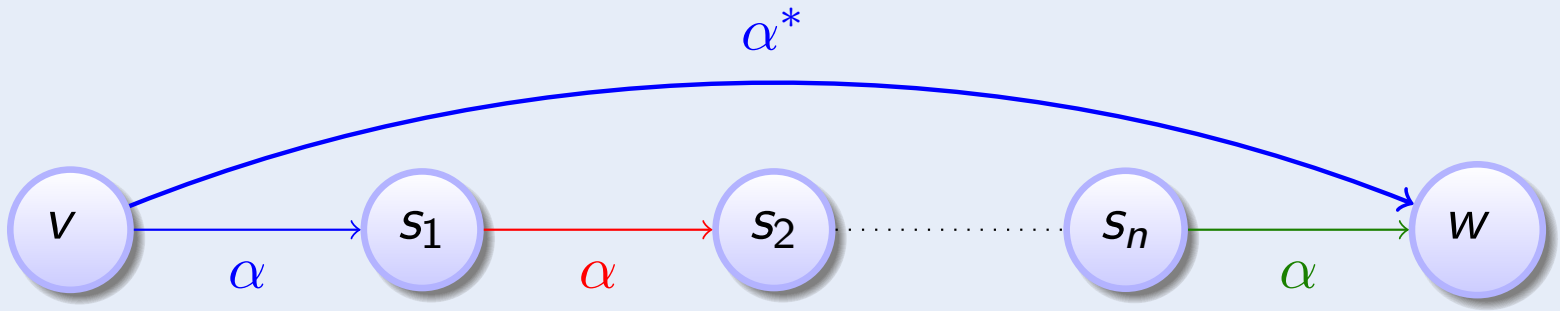


Definition (Hybrid programs α: transition semantics)

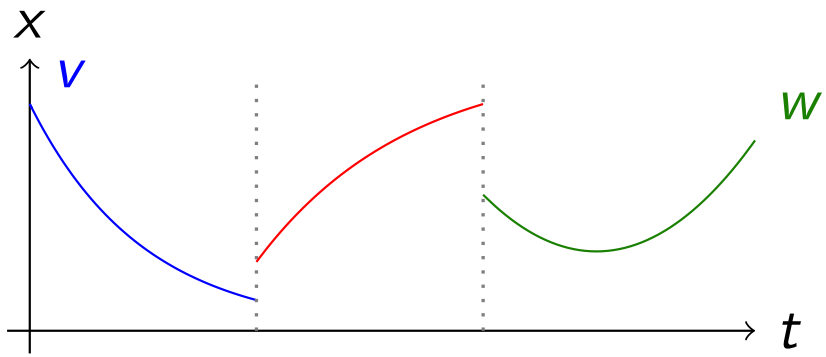


Details

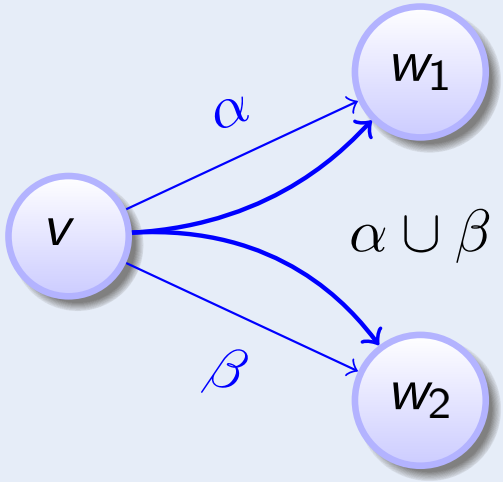
Definition (Hybrid programs α: transition semantics)



▸ Details ▸▸

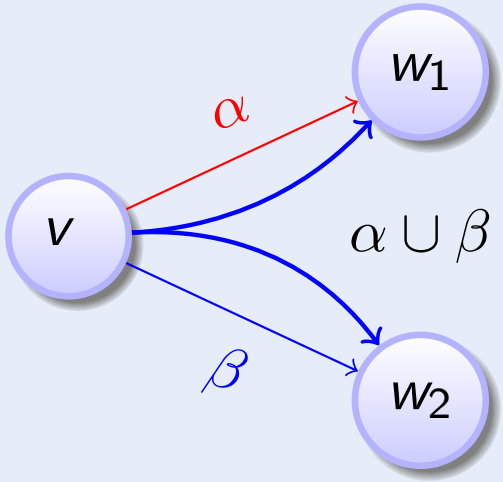


Definition (Hybrid programs α: transition semantics)

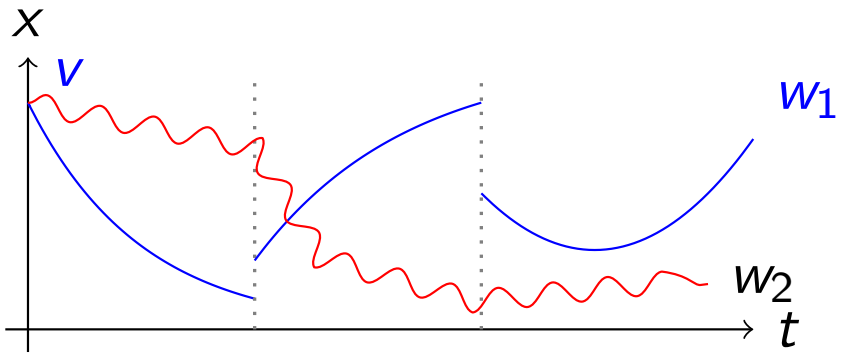


▸ Details ▸▸

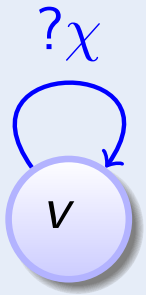
Definition (Hybrid programs α: transition semantics)



▸ Details ▸▸

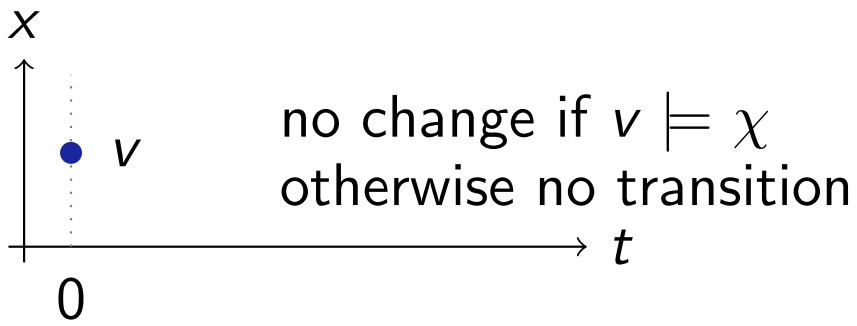


Definition (Hybrid programs α: transition semantics)



if $v \models \chi$

Details



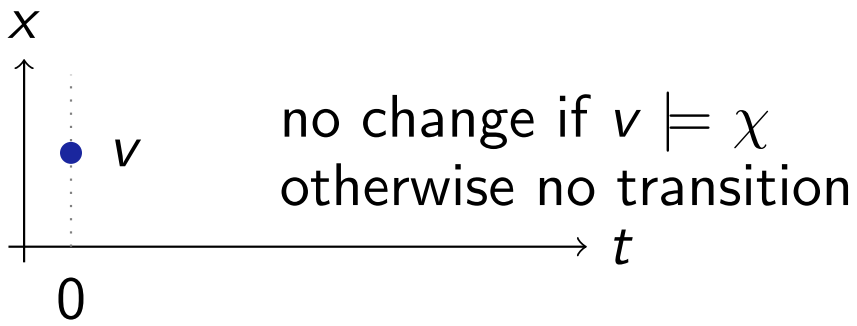
no change if $v \models \chi$
otherwise no transition

Definition (Hybrid programs α: transition semantics)

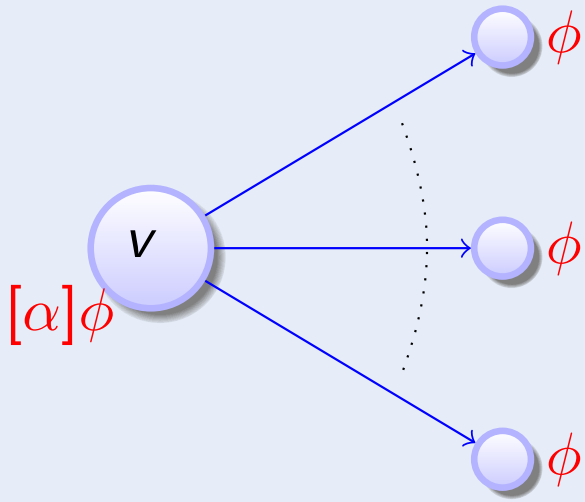
v

if $v \not\models \chi$

Details

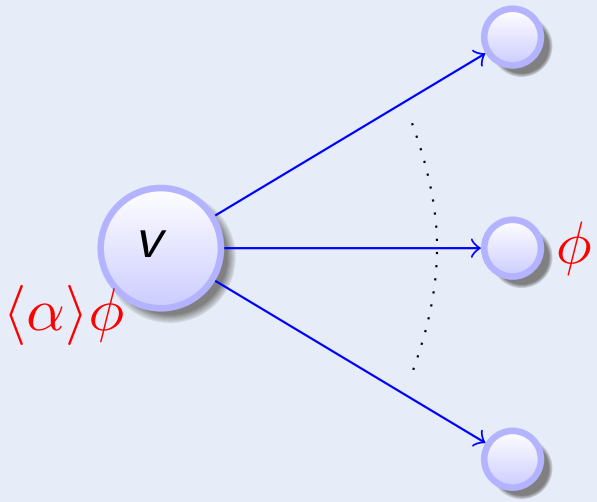


Definition (Formulas ϕ)



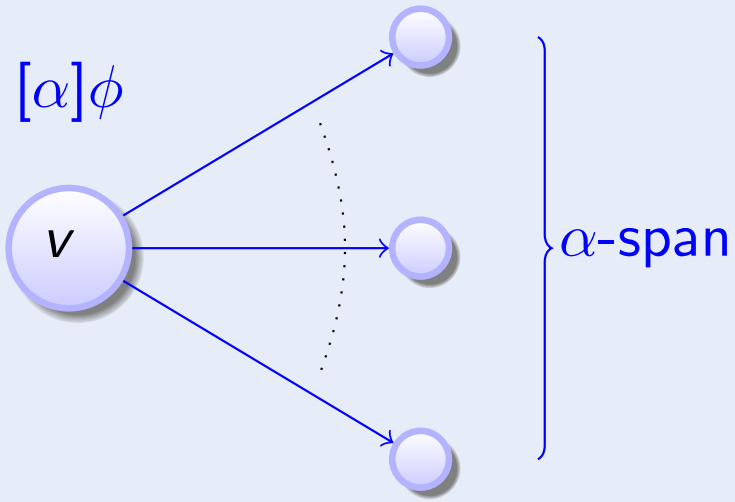
▸ Details ▸▸

Definition (Formulas ϕ)



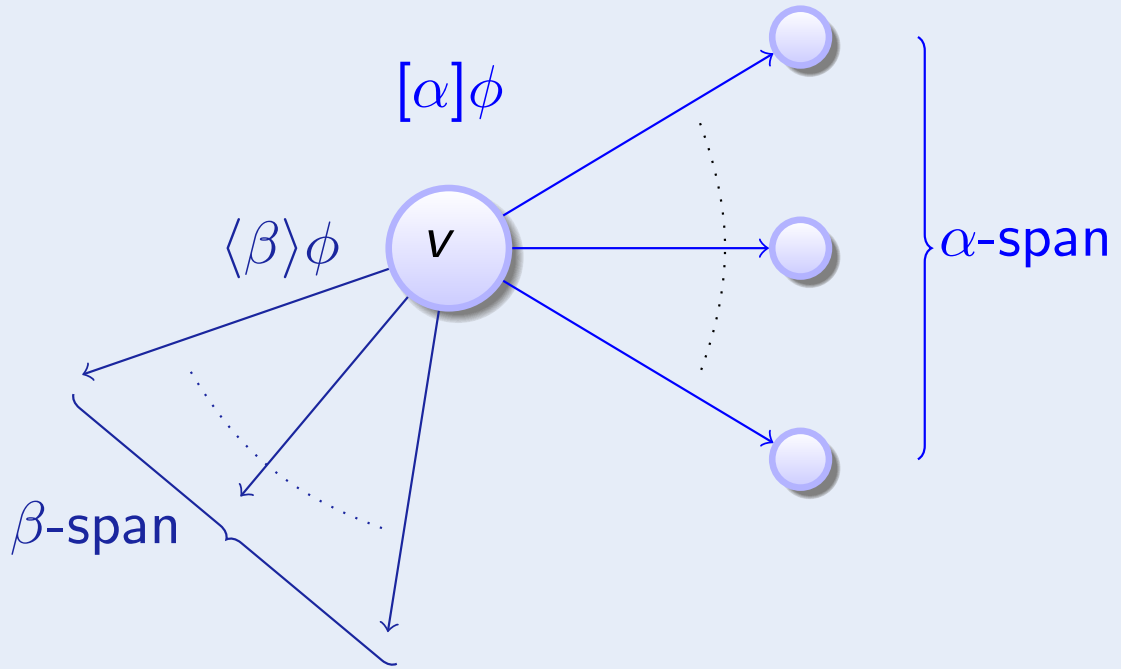
▸ Details ▸▸

Definition (Formulas ϕ)



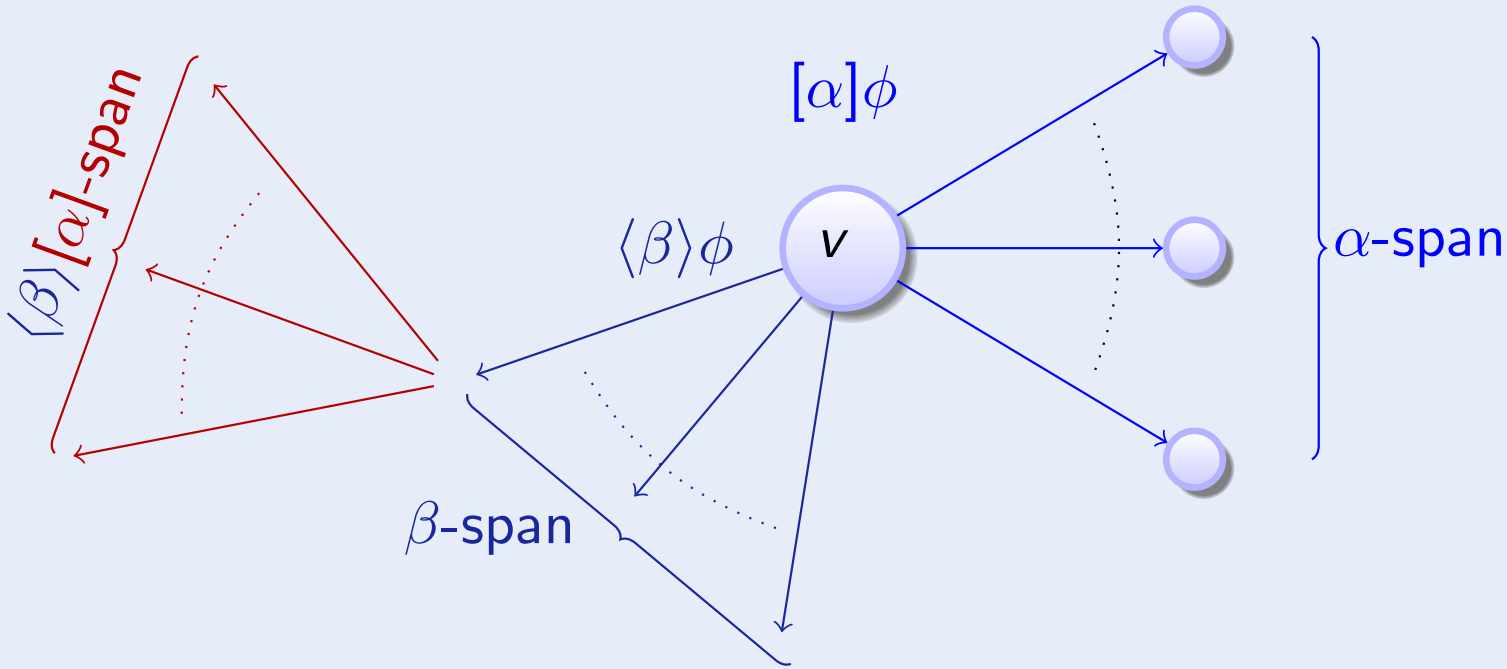
▸ Details ▸▸

Definition (Formulas ϕ)



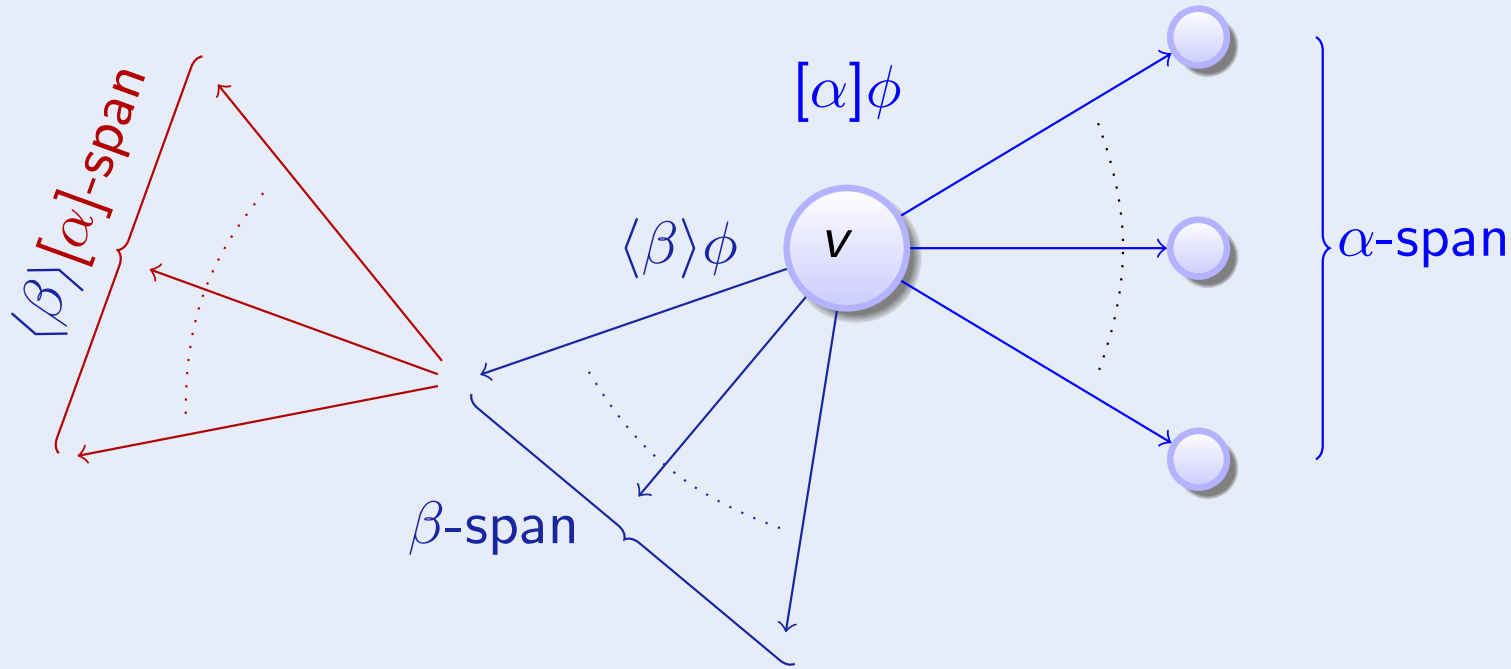
▸ Details ▸▸

Definition (Formulas ϕ)



▸ Details ▸▸

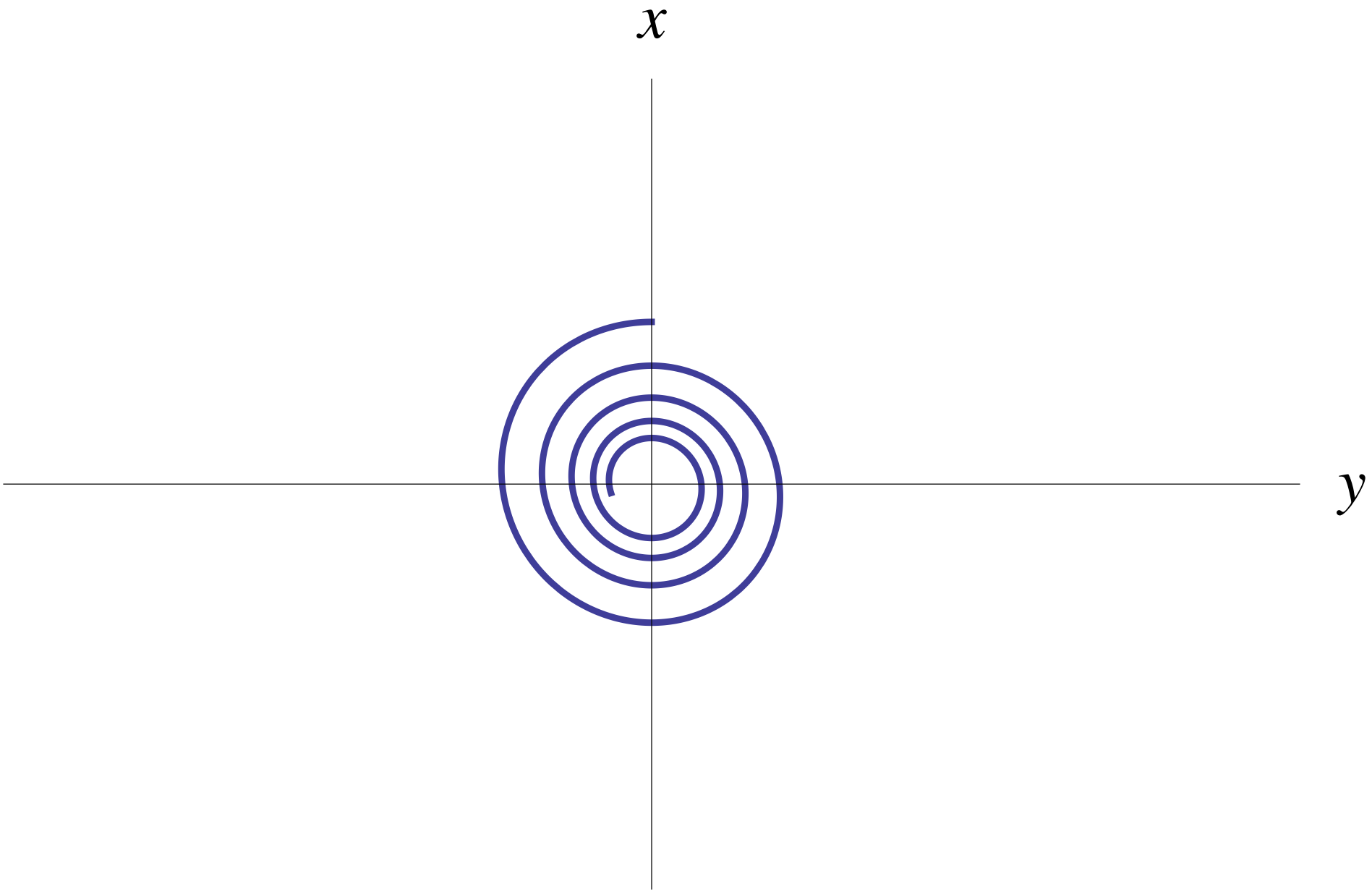
Definition (Formulas ϕ)

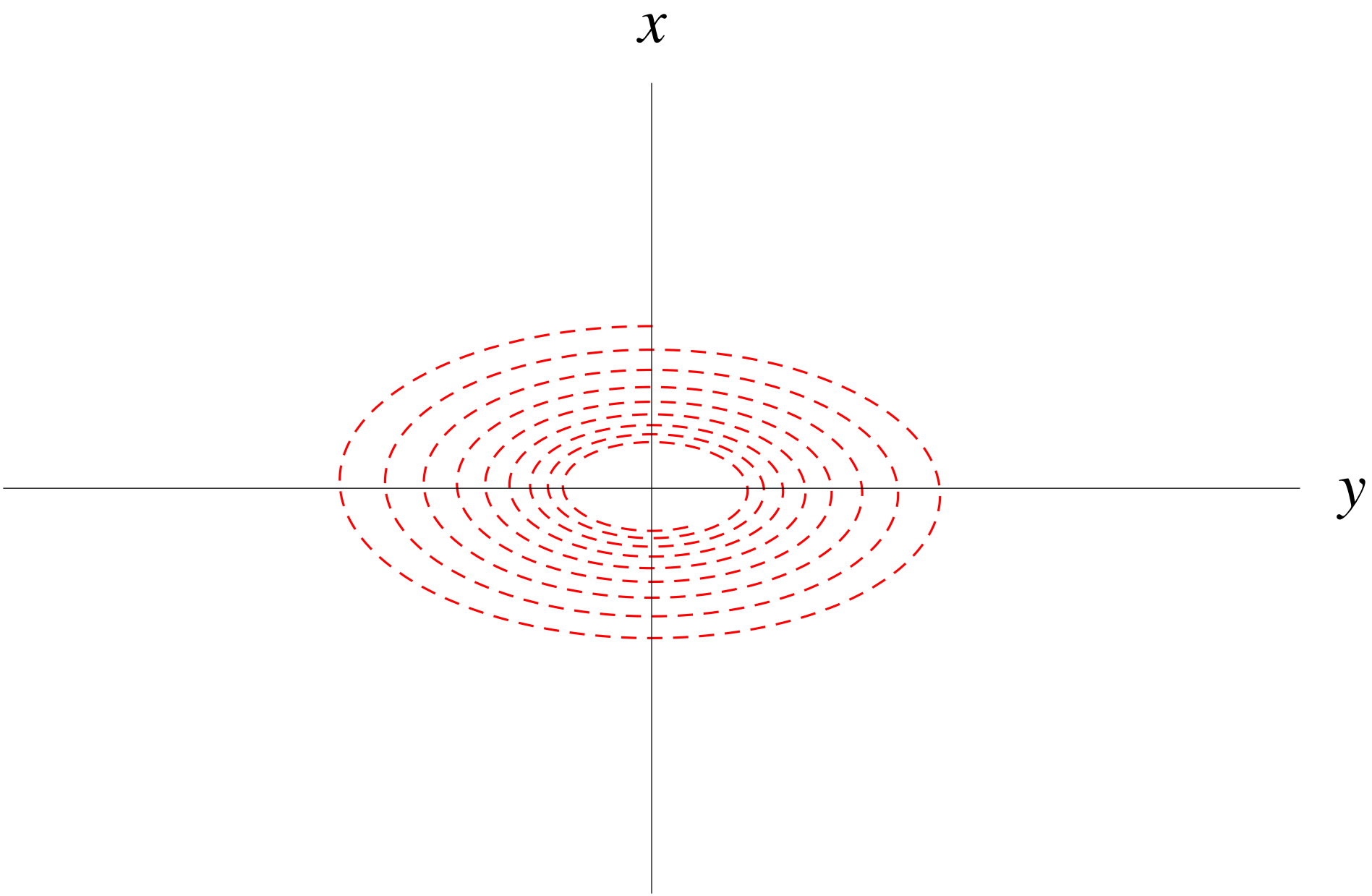


▸ Details ▸▸

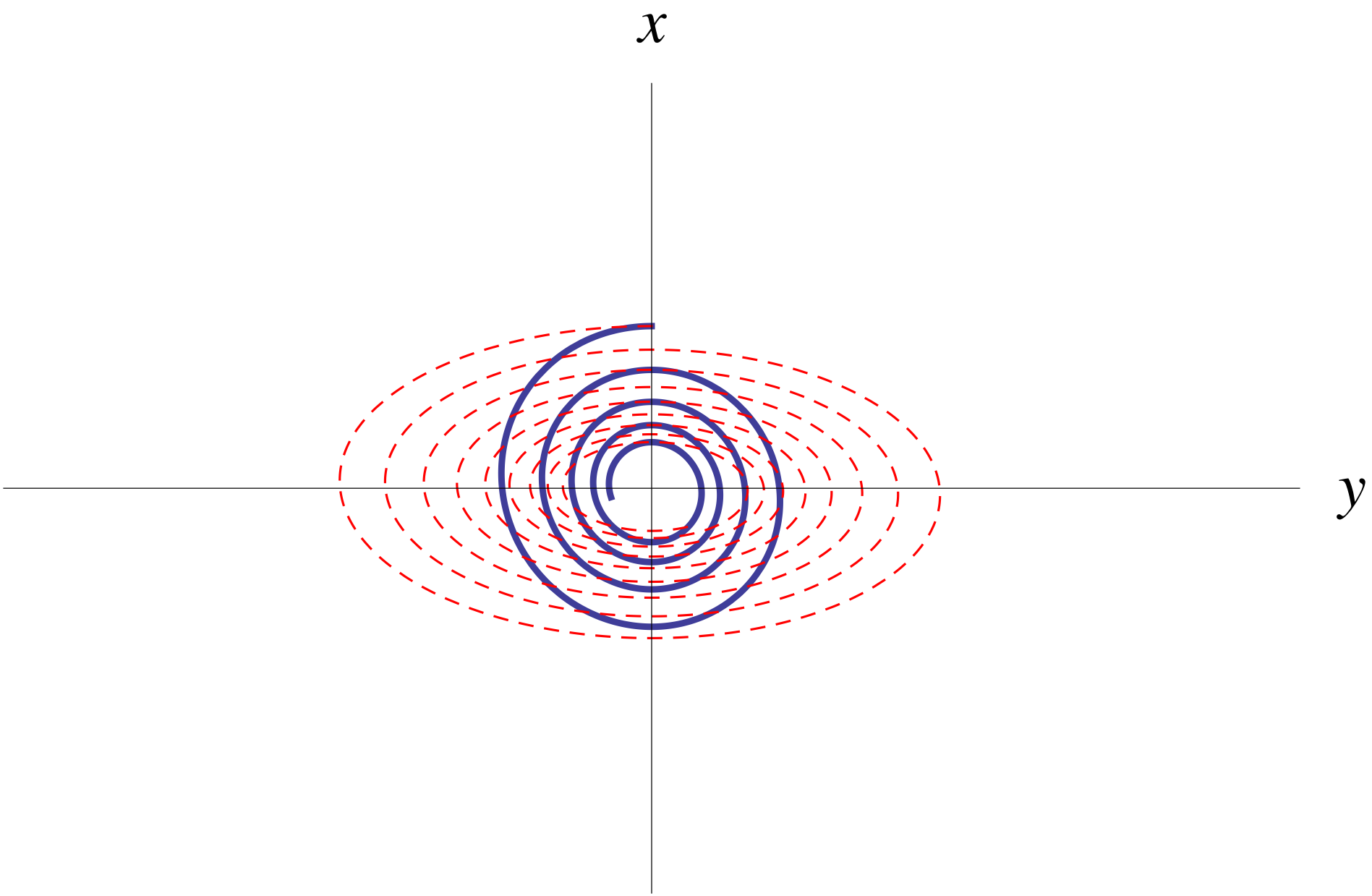
compositional semantics $\Rightarrow$ compositional proofs!

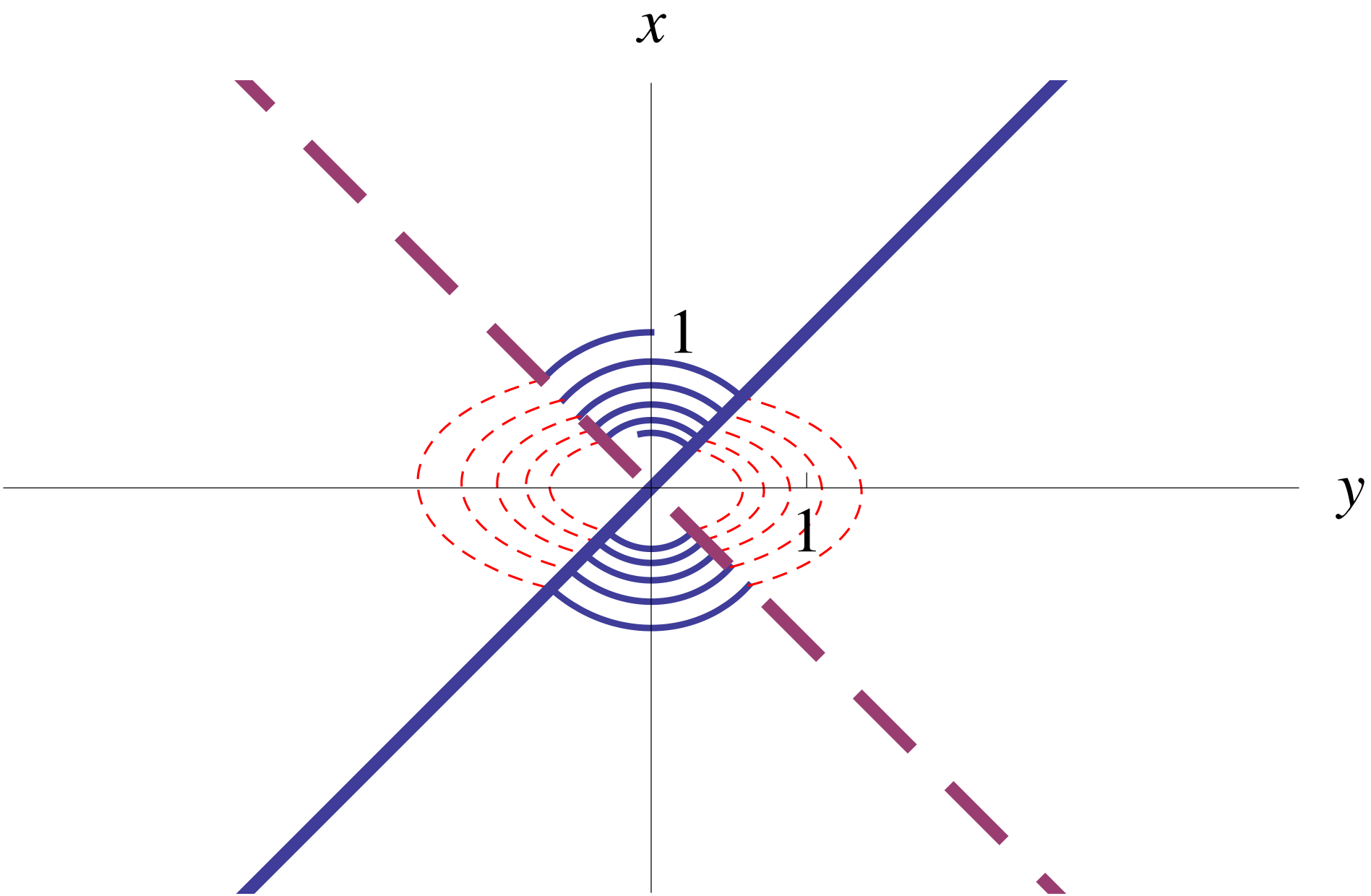
Safe Switching in Hybrid Systems



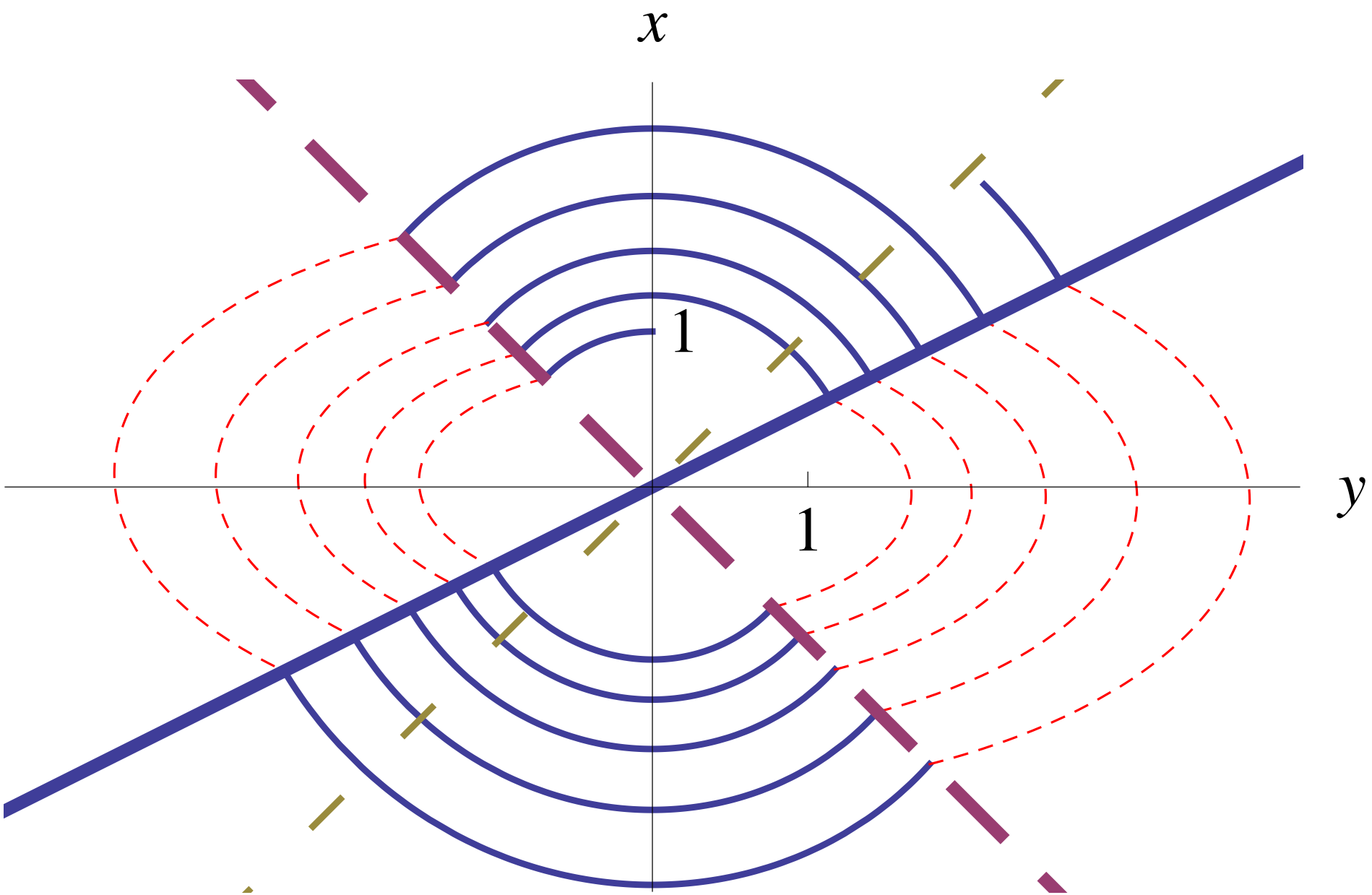

x
y

Safe Switching in Hybrid Systems



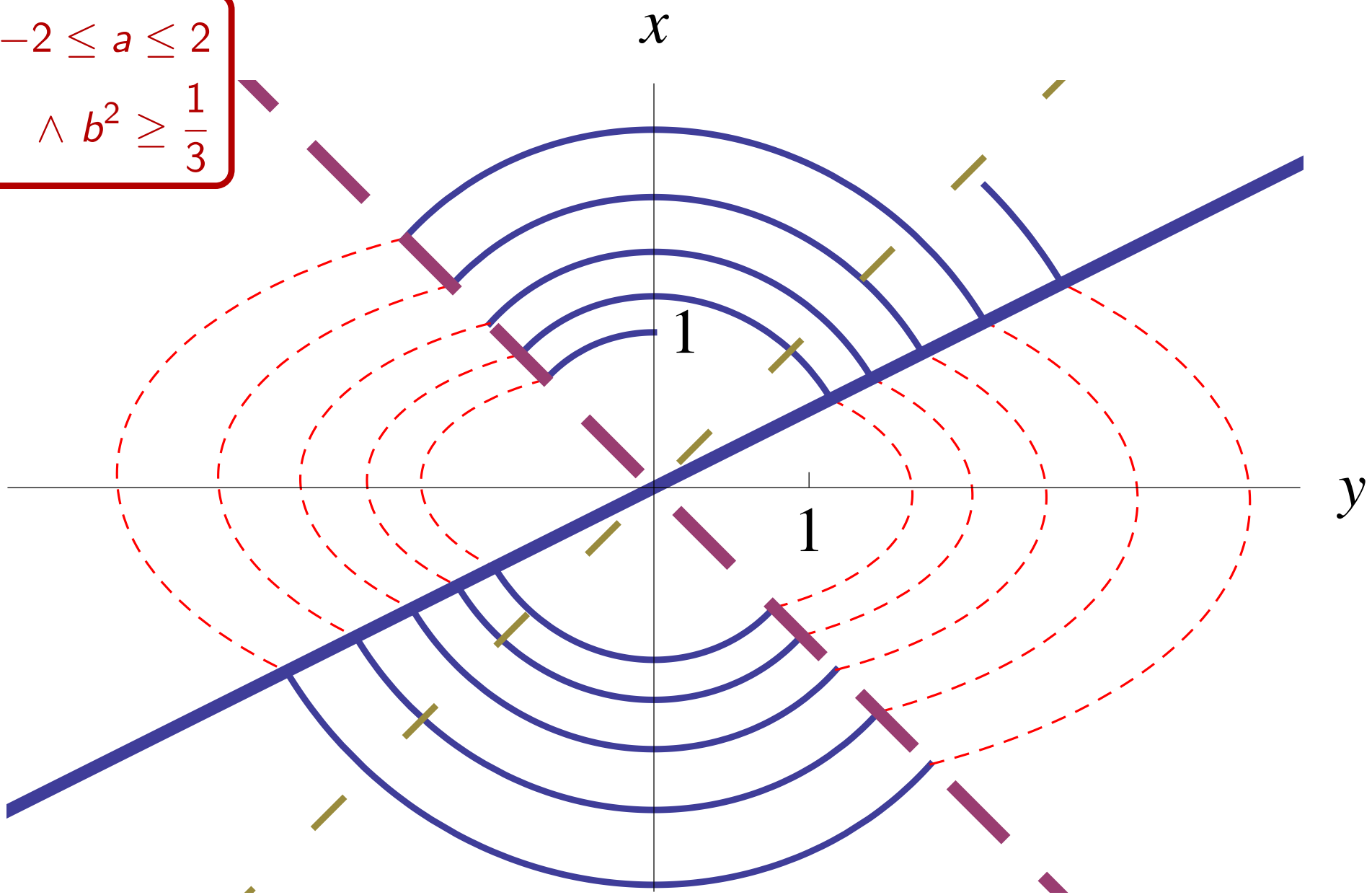

x
y
1
1

Safe Switching in Hybrid Systems



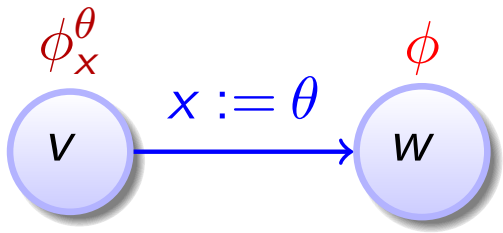
$-2 \leq a \leq 2$

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



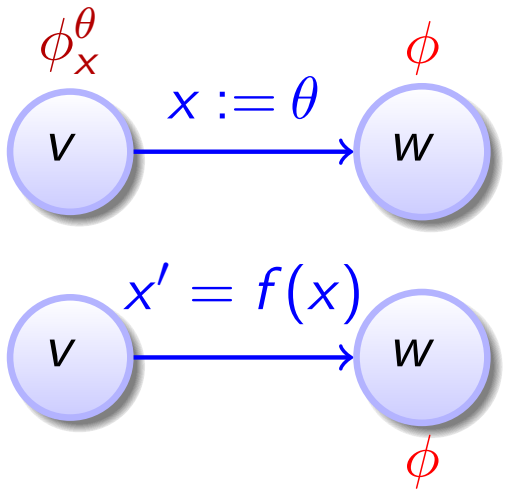
Outline (Verification Approach)

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



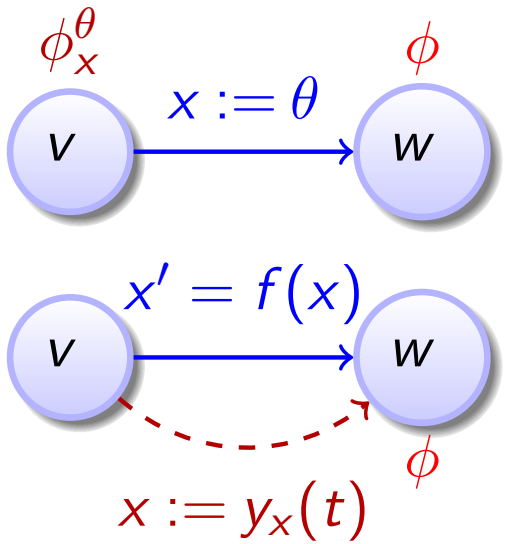
$$\frac{\phi_x^\theta}{\langle x := \theta \rangle \phi}$$

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



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

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

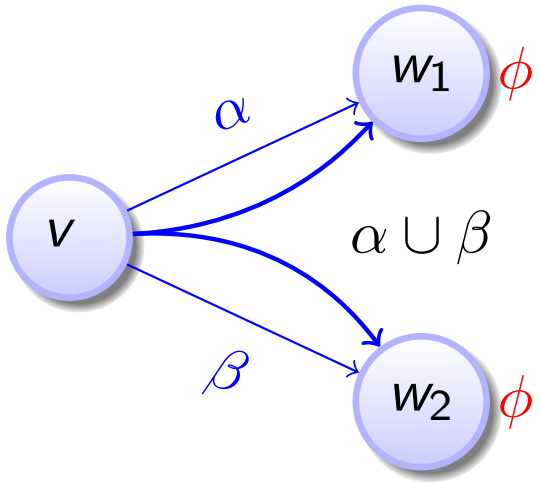


Proof Calculus for Differential Dynamic Logic

compositional semantics $\Rightarrow$ compositional rules!

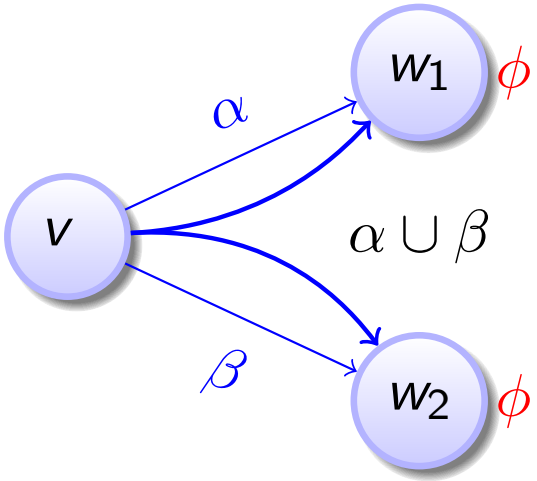
Proof Calculus for Differential Dynamic Logic

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

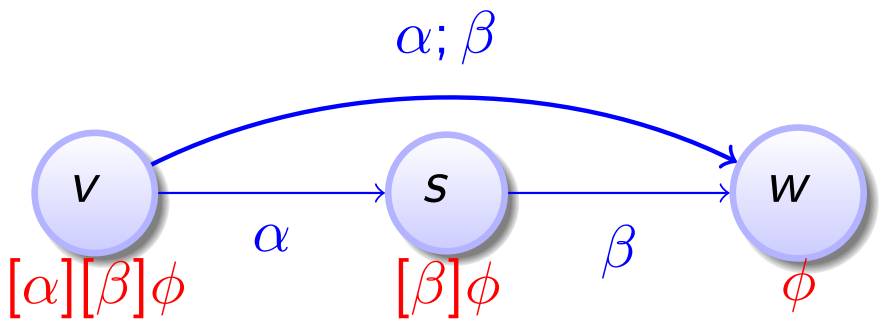


Proof Calculus for Differential Dynamic Logic

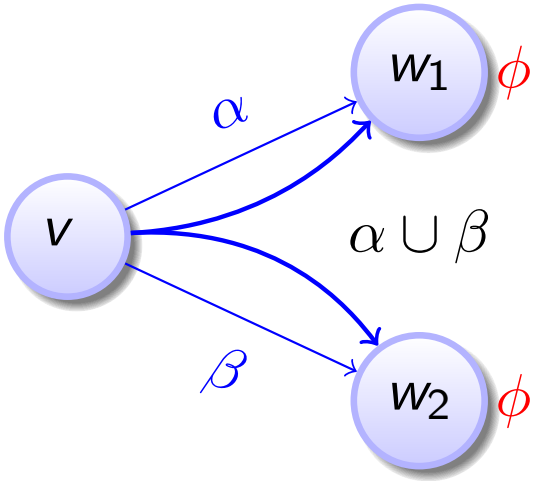
$$\frac{[\alpha]\phi \wedge [\beta]\phi}{[\alpha \cup \beta]\phi}$$



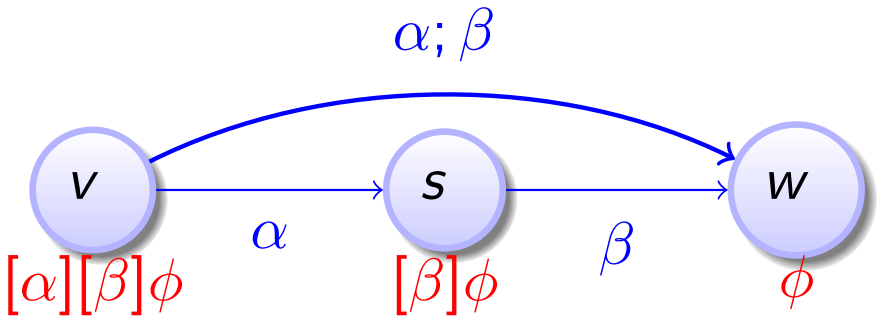
$$\frac{[\alpha][\beta]\phi}{[\alpha;\beta]\phi}$$



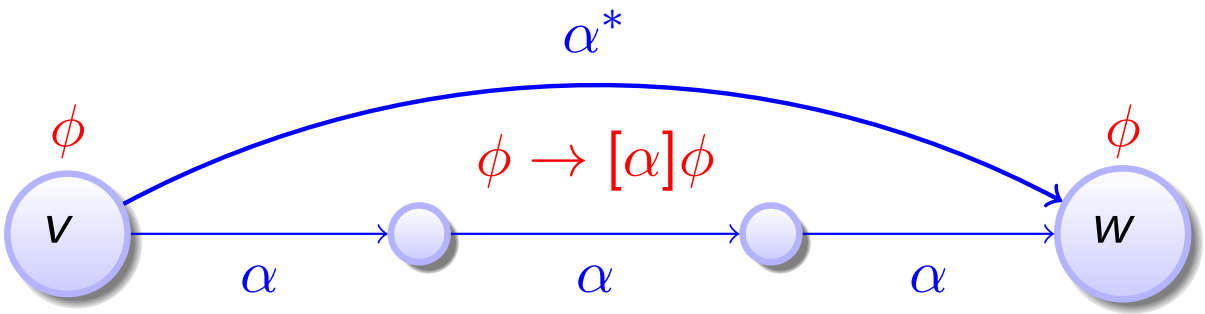
$$\frac{[\alpha]\phi \wedge [\beta]\phi}{[\alpha \cup \beta]\phi}$$



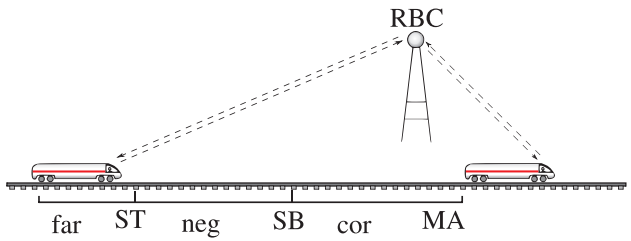
$$\frac{[\alpha][\beta]\phi}{[\alpha;\beta]\phi}$$



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

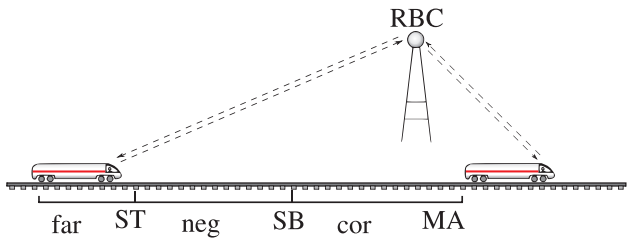


Deduction Modulo Real Arithmetic

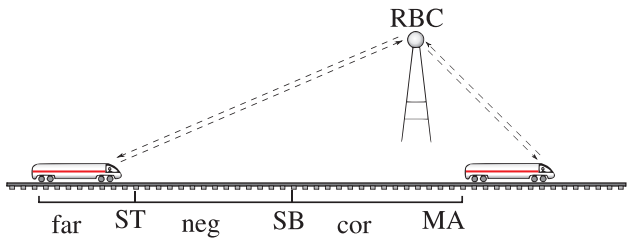


$$\vdash v \geq 0 \land z < MA \rightarrow \langle z' = v, v' = -b \rangle\ z > MA$$

Deduction Modulo Real Arithmetic



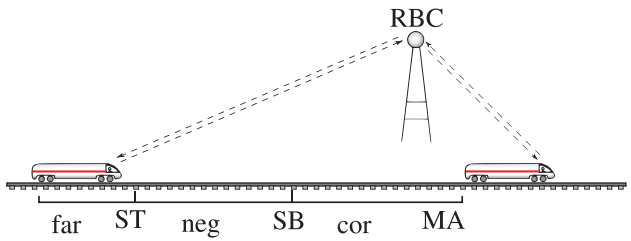
$$\frac{\dfrac{}{v \geq 0, z < MA \vdash \exists t{\geq}0\, \langle z := -\frac{b}{2}t^2 + vt + z\rangle z > MA}}{v \geq 0, z < MA \vdash \langle z' = v, v' = -b\rangle z > MA}$$
$$\vdash v \geq 0 \land z < MA \rightarrow \langle z' = v, v' = -b\rangle\ z > MA$$



Collins/Tarski QE not applicable!

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

Deduction Modulo (Side Deduction)

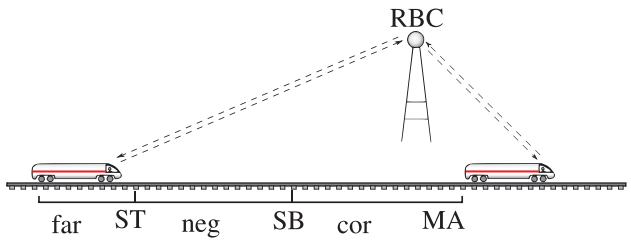


$$\frac{}{v \geq 0, z < MA \vdash t \geq 0 \wedge \langle z := -\frac{b}{2}t^2 + vt + z \rangle z > MA}$$

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)

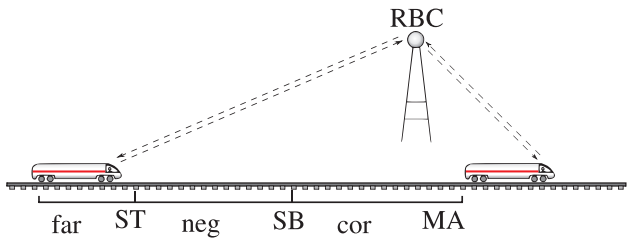


$$\dfrac{v \geq 0, z < MA \vdash t \geq 0 \qquad \dfrac{v \geq 0, z < MA \vdash -\frac{b}{2}t^2 + vt + z > MA}{v \geq 0, z < MA \vdash \langle z := -\frac{b}{2}t^2 + vt + z \rangle z > MA}}{v \geq 0, z < MA \vdash t \geq 0 \wedge \langle z := -\frac{b}{2}t^2 + vt + z \rangle z > MA}$$

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)

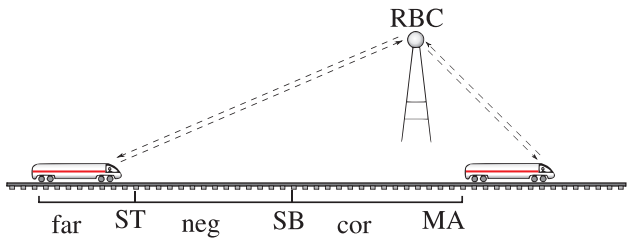


$$\dfrac{\dfrac{v \geq 0, z < MA \vdash t \geq 0 \qquad v \geq 0, z < MA \vdash -\frac{b}{2}t^2 + vt + z > MA}{v \geq 0, z < MA \vdash \langle z := -\frac{b}{2}t^2 + vt + z \rangle z > MA}\,\text{QE}}{v \geq 0, z < MA \vdash t \geq 0 \wedge \langle z := -\frac{b}{2}t^2 + vt + z \rangle z > MA}$$

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

start side

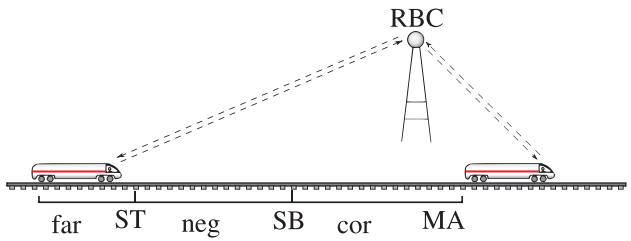
Deduction Modulo (Side Deduction)



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

start side

Deduction Modulo (Free Variables for Automation)

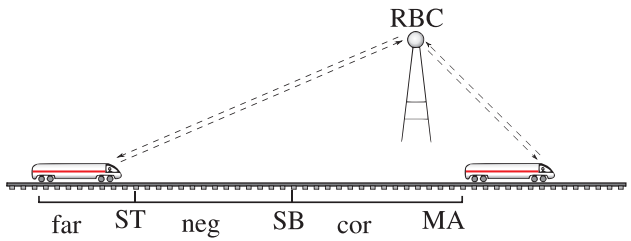


$$v \geq 0, z < MA \vdash \exists t{\geq}0\, \langle z := -\tfrac{b}{2}t^2 + vt + z \rangle z > MA$$

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

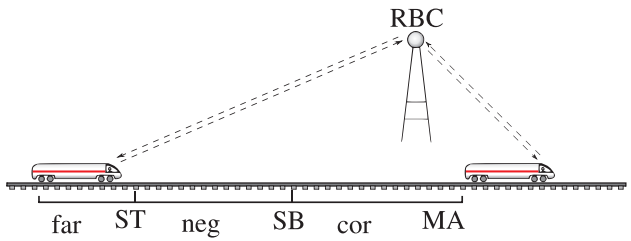
$$\vdash v \geq 0 \wedge z < MA \rightarrow \langle z' = v, v' = -b \rangle z > MA$$

Deduction Modulo (Free Variables for Automation)



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

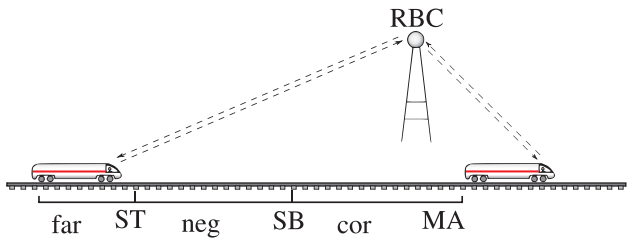
Deduction Modulo (Free Variables for Automation)



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

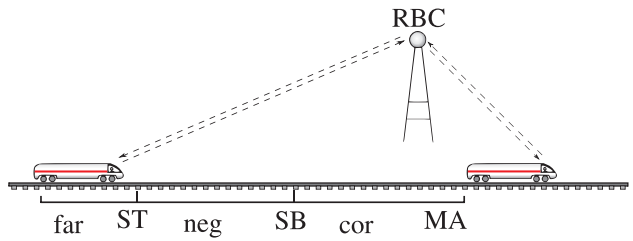
$$v \geq 0, z < MA \vdash \exists T \, (\ldots T \geq 0 \wedge -\tfrac{b}{2}T^2 + vT + z > MA)$$

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)

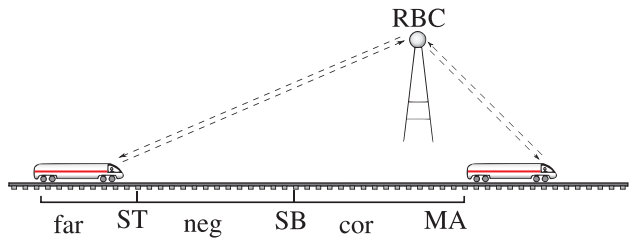


$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{v \geq 0, z < MA \vdash v^2 > 2b(MA - z)}{v \geq 0, z < MA \vdash -\frac{b}{2}T^2 + vT + z > MA}}{v \geq 0, z < MA \vdash \langle z := -\frac{b}{2}T^2 + vT + z \rangle z > MA}}{v \geq 0, z < MA \vdash T \geq 0 \wedge \langle z := -\frac{b}{2}T^2 + vT + z \rangle z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

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

Deduction Modulo (Free Variables for Automation)

- 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)

$$\frac{}{\vdash (X < S)}$$

$$\frac{\vdash (X < S)}{\vdash \forall s\,(X < s)}$$

$$\frac{\vdash \forall s\,(X < s)}{\vdash \exists x\,\forall s\,(x < s)}$$

Deduction Modulo (Free Variables for Automation)

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

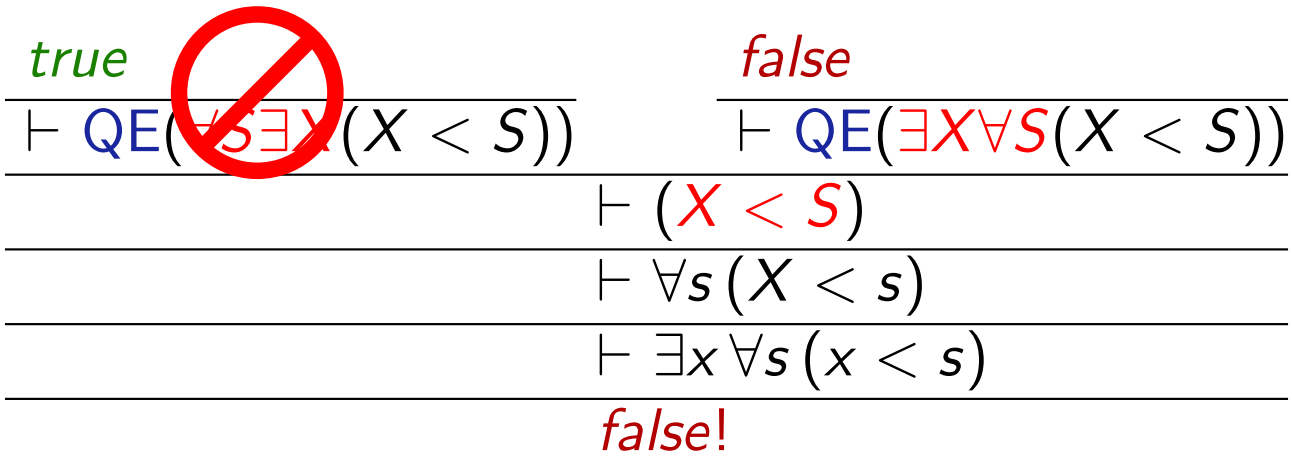
Deduction Modulo (Free Variables for Automation)

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

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 $S(X)$

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

$\textit{false}!$

Soundness and Completeness

Theorem (Relative Completeness)

dℒ 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!

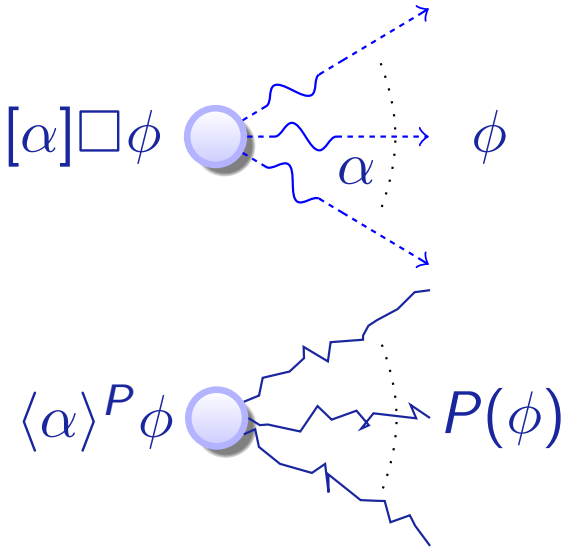
Corollary (Compositionality)

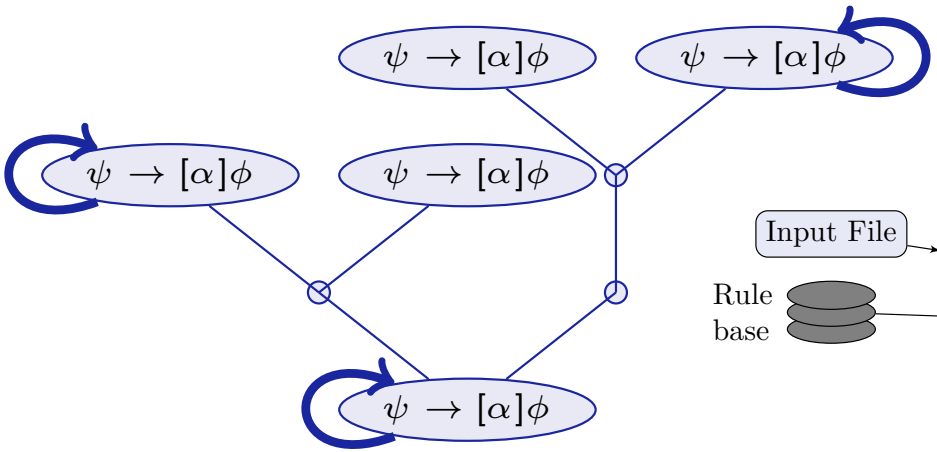
hybrid systems can be verified by recursive decomposition

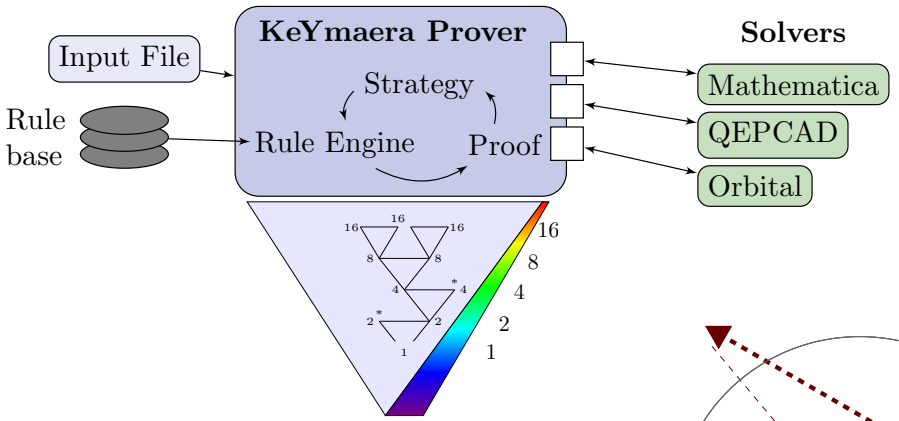

André Platzer.
Differential dynamic logic for hybrid systems.
J. Autom. Reas., 41(2):143–189, 2008.

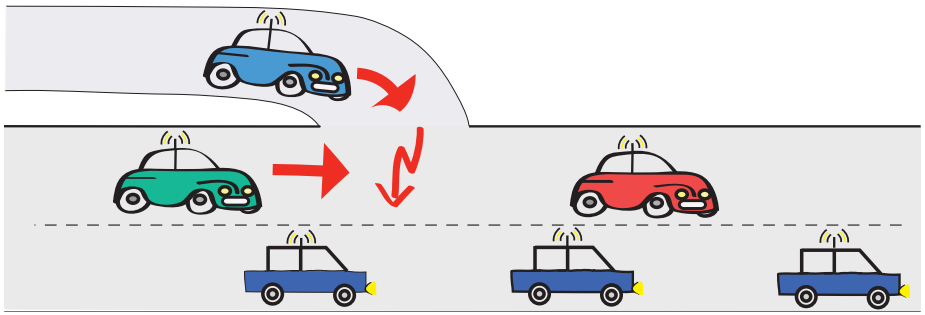
Outline

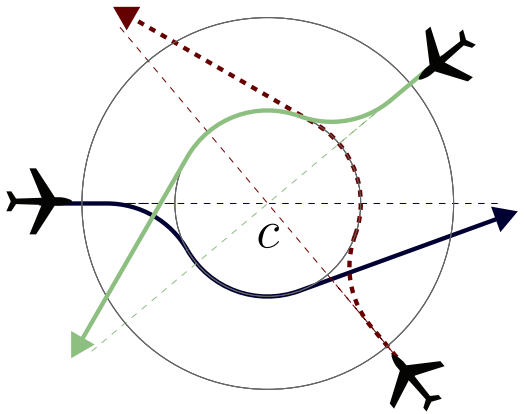

¬F
F
F


[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



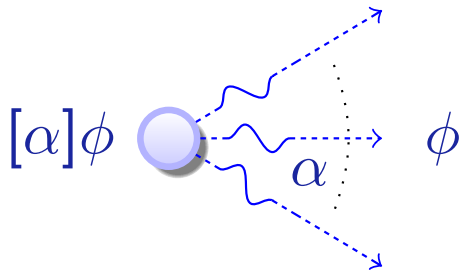

c

Outline

Conclusions

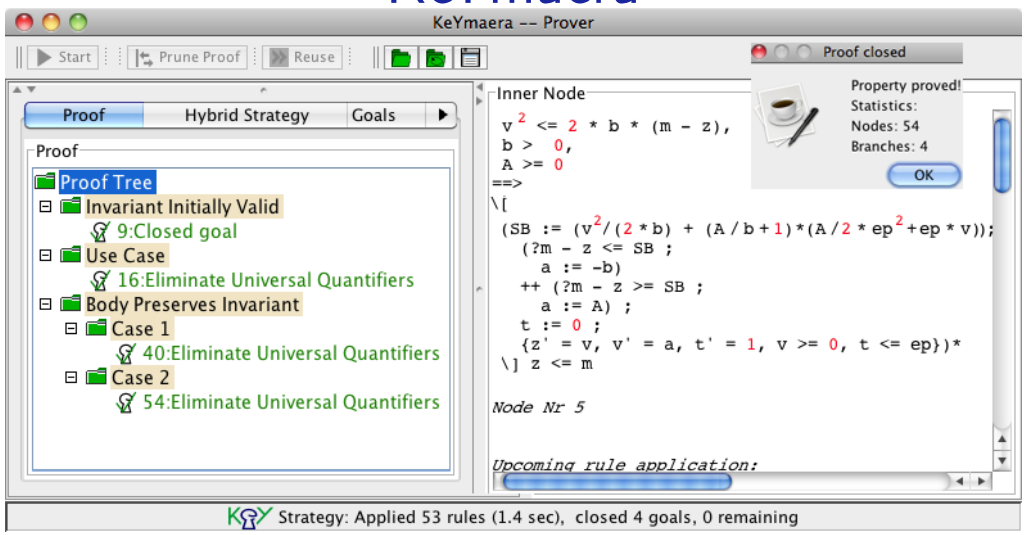
differential dynamic logic

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



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

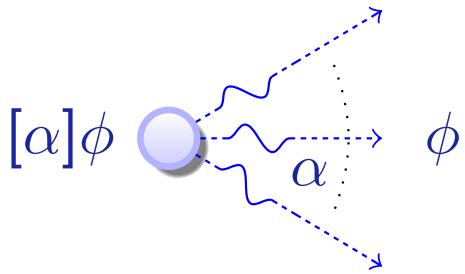
KeYmaera



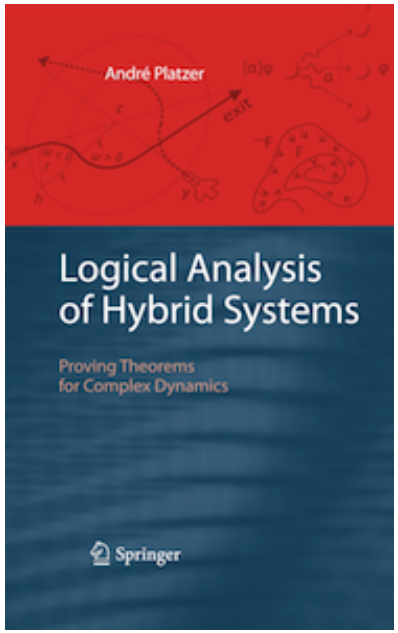
Conclusions

differential dynamic logic

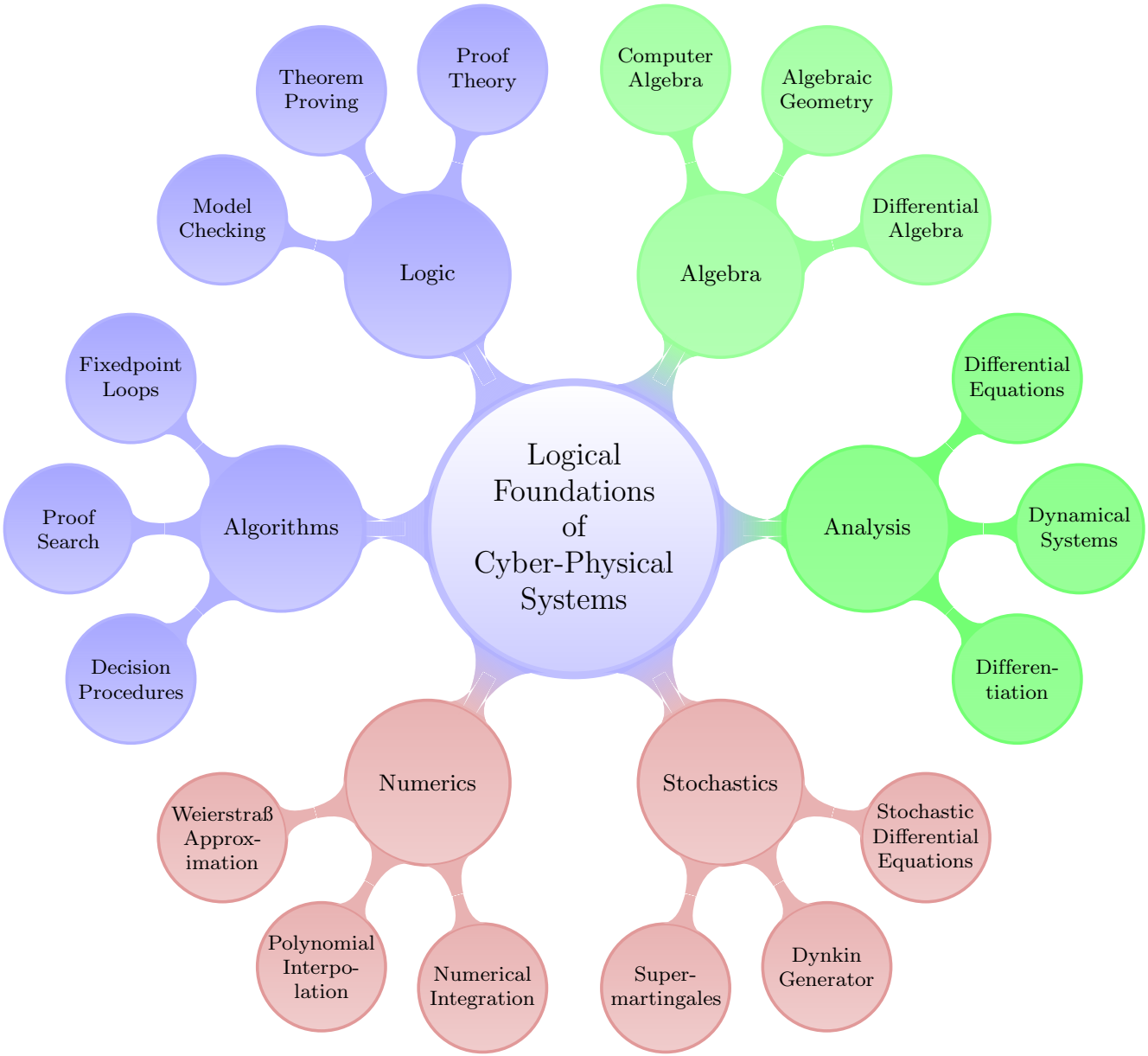
$$\mathrm{d}\mathcal{L} = \mathrm{DL} + \mathrm{HP}$$



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



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

Definition (Kripke state)

$\nu : V \to \mathbb{R}$ with set of variables V

Return

Definition (Formulas ϕ)

$$
\begin{array}{lll}
v \models [\alpha]\phi & :\Longleftrightarrow & w \models \phi \quad \text{for all } w \text{ with } (v,w) \in \rho(\alpha) \\
v \models \langle\alpha\rangle\phi & :\Longleftrightarrow & w \models \phi \quad \text{for some } w \text{ with } (v,w) \in \rho(\alpha)
\end{array}
$$

Definition (Hybrid programs α)

$$
\begin{array}{rcl}
\rho(x' = f(x)) & = & \{(\varphi(0), \varphi(r)) \,:\, \varphi \models x' = f(x) \text{ for duration } r\} \\
(v,w) \in \rho(x := \theta) & :\Longleftrightarrow & w = v[x \mapsto [\![\theta]\!]_v] \\
\rho(?\chi) & = & \{(v,v) \,:\, v \models \chi\} \\
\rho(\alpha \cup \gamma) & = & \rho(\alpha) \cup \rho(\gamma) \\
\rho(\alpha;\gamma) & = & \rho(\alpha) \circ \rho(\gamma)
\end{array}
$$

$$
(v,w) \in \rho(\alpha^*) :\Longleftrightarrow \text{there is } v \xrightarrow{\rho(\alpha)} v_1 \xrightarrow{\rho(\alpha)} v_2 \cdots\cdots \xrightarrow{\rho(\alpha)} w
$$

Return

Definition (Formulas ϕ)

$$
\begin{array}{lll}
v \models [\alpha]\phi & :\Longleftrightarrow & w \models \phi \quad \text{for all } w \text{ with } (v,w) \in \rho(\alpha) \\
v \models \langle\alpha\rangle\phi & :\Longleftrightarrow & w \models \phi \quad \text{for some } w \text{ with } (v,w) \in \rho(\alpha)
\end{array}
$$

Definition (Hybrid programs α)

$$
\begin{array}{rcl}
\rho(x' = f(x)) & = & \{(\varphi(0), \varphi(r)) \,:\, \varphi \models x' = f(x) \text{ for duration } r\} \\
(v,w) \in \rho(x := \theta) & :\Longleftrightarrow & w = v[x \mapsto [\![\theta]\!]_v] \\
\rho(?\chi) & = & \{(v,v) \,:\, v \models \chi\} \\
\rho(\alpha \cup \gamma) & = & \rho(\alpha) \cup \rho(\gamma) \\
\rho(\alpha;\gamma) & = & \rho(\alpha) \circ \rho(\gamma)
\end{array}
$$

$$
(v,w) \in \rho(\alpha^*) :\Longleftrightarrow \text{there is } v \xrightarrow{\rho(\alpha)} v_1 \xrightarrow{\rho(\alpha)} v_2 \cdots\cdots \xrightarrow{\rho(\alpha)} w
$$

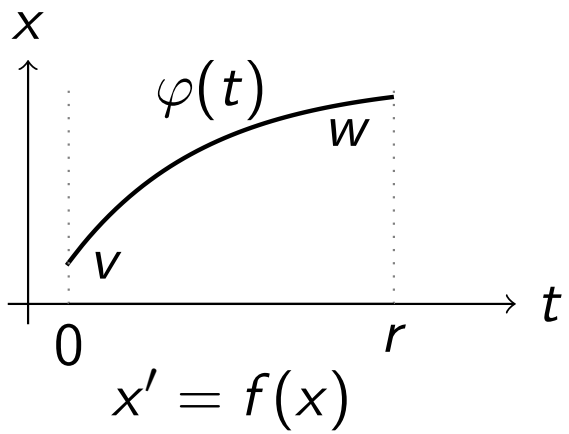
Return

Definition (Hybrid programs α)

$$\rho(x' = f(x)) \quad = \quad \{(\varphi(0), \varphi(r)) \ : \ \varphi \models x' = f(x) \text{ for duration } r\}$$

$$\text{with } [\![x']\!]_{\varphi(\zeta)} = \frac{\mathrm{d}\varphi(t)(x)}{\mathrm{d}t}(\zeta)$$

- there is $\varphi : [0, r] \to \text{States}$ with $\varphi(0) = v, \varphi(r) = w$
- $[\![x]\!]_{\varphi(\zeta)}$ is continuous in ζ on $[0, r]$
- $\frac{\mathrm{d}\,[\![x]\!]_{\varphi(t)}}{\mathrm{d}t}(\zeta) = [\![f(x)]\!]_{\varphi(\zeta)}$ for $\zeta \in (0, r)$
- $[\![y]\!]_{\varphi(\zeta)} = [\![y]\!]_v$ otherwise



Return

Proof (Soundness).

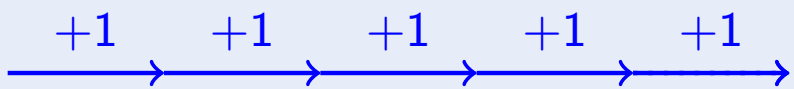
- $x' = f(x)$
- Side deductions
- Free variables & Skolemisation

Return

Proof (Incompleteness).

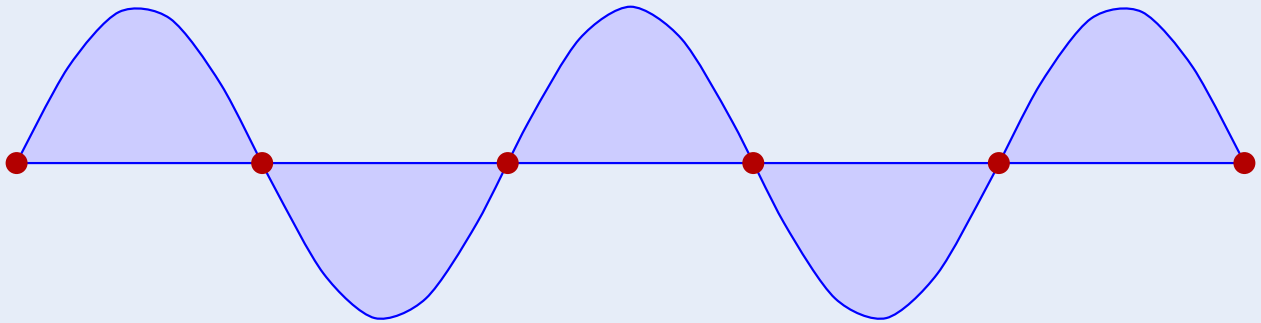
Discrete fragment:

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



Continuous fragment:

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



Return

Incomplete! But are we missing proof rules?

Incomplete! But are we missing proof rules?

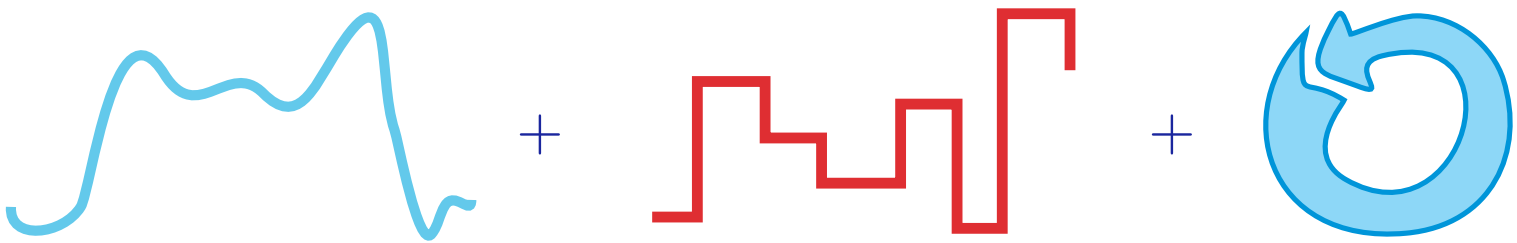
Relativity

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

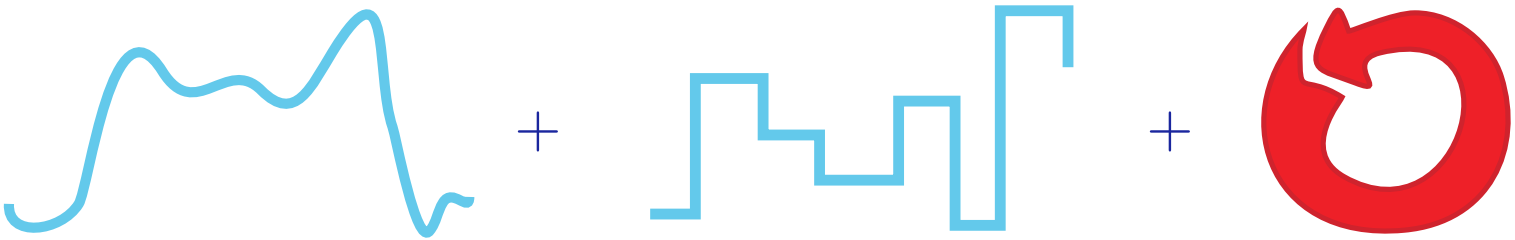
Sources of Incompleteness


continuous

Sources of Incompleteness

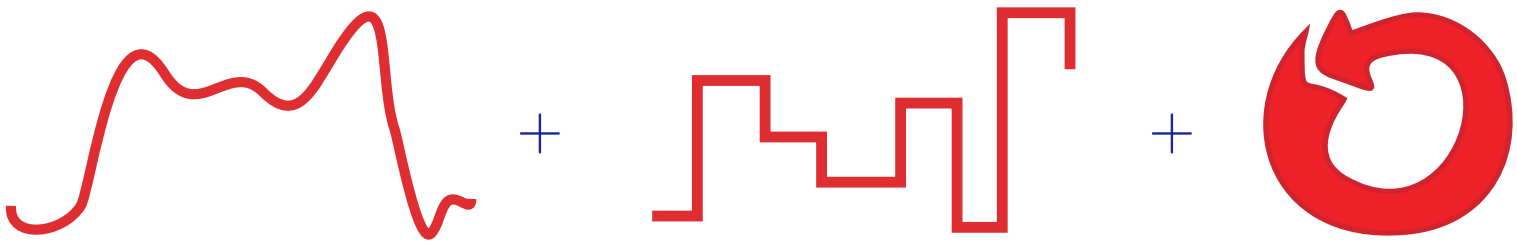


Sources of Incompleteness

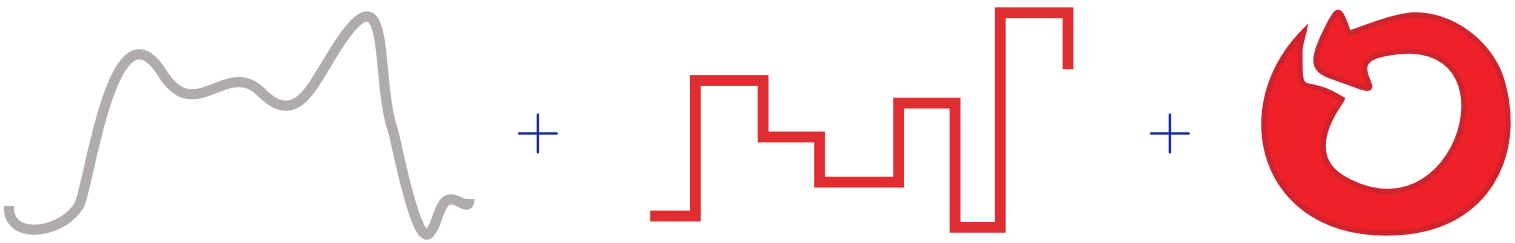


continuous + discrete + repeat

Sources of Incompleteness



Relative Completeness



Relative Completeness

Theorem (Relative Completeness)

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

$$\models \phi \quad \text{iff} \quad \mathit{Taut}_{\mathsf{FOD}} \vdash \phi$$

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

▸ Proof Outline 15p

continuous + discrete + repeat

$\Downarrow$

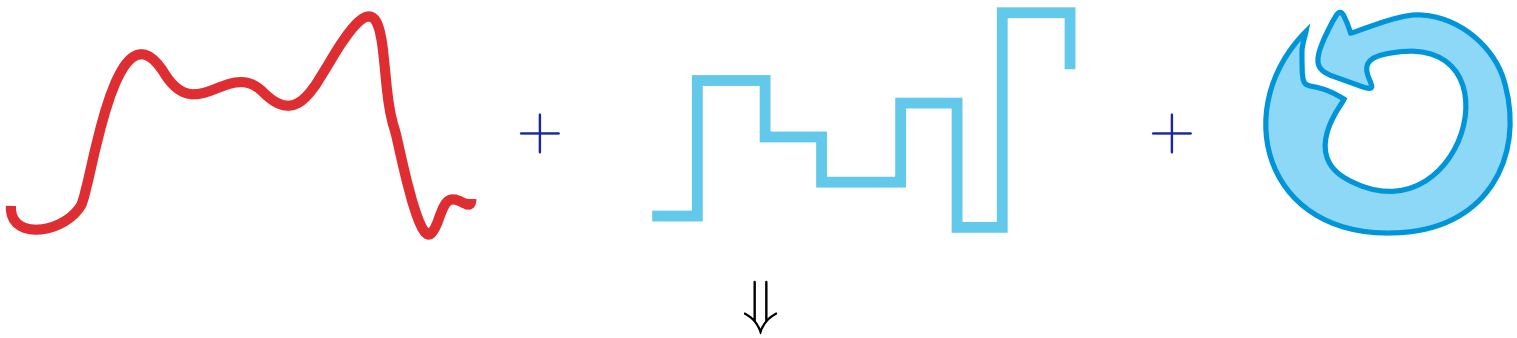
Theorem (Relative Completeness)

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

$$\models \phi \quad \text{iff} \quad \mathit{Taut}_{\mathsf{FOD}} \vdash \phi$$

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

▸ Proof Outline 15p



Relativity

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

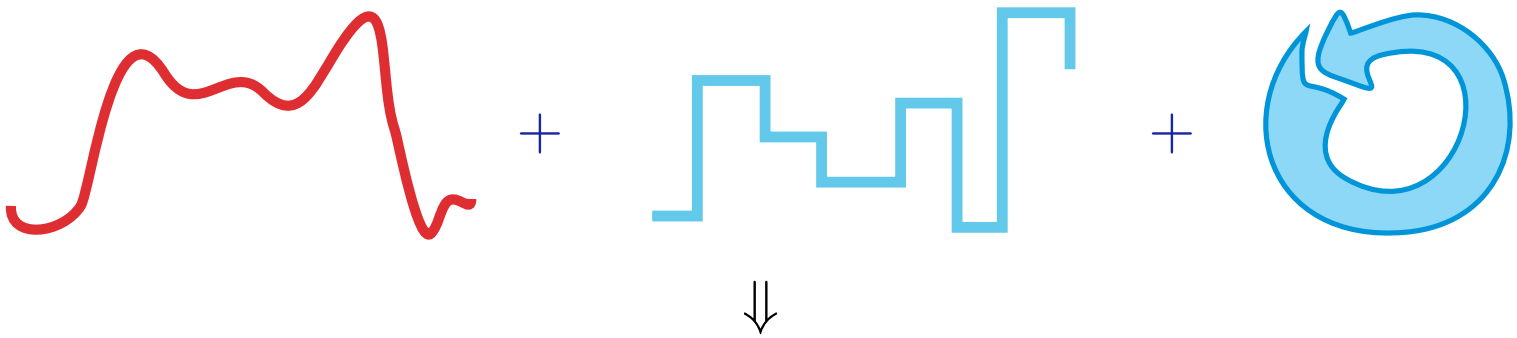
Theorem (Relative Completeness)

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

$$\models \phi \quad \text{iff} \quad \mathit{Taut}_{\mathsf{FOD}} \vdash \phi$$

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

▸ Proof Outline 15p



Corollary (Proof-theoretical Alignment)

verification of hybrid systems = verification of dynamical systems!

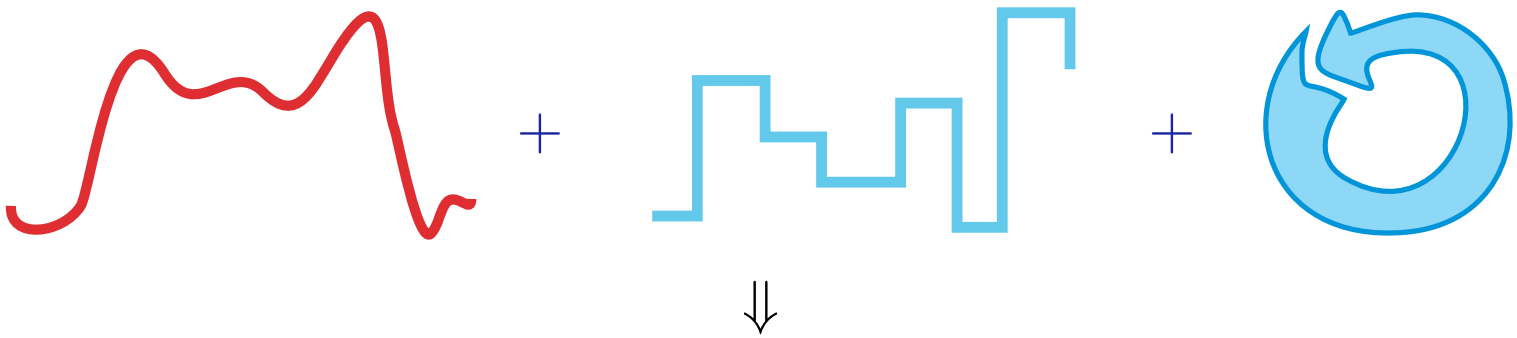
Theorem (Relative Completeness)

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

$$\models \phi \quad \text{iff} \quad \mathit{Taut}_{\mathsf{FOD}} \vdash \phi$$

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

▸ Proof Outline 15p



Corollary (Deductive Power)

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

Relative Completeness Proof

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

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

Return

Proof (Relative Completeness, 10 pages).

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

Relative Completeness Proof

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

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

Return

Proof (Relative Completeness, 10 pages).

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

Relative Completeness Proof

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

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

Return

Proof (Relative Completeness, 10 pages).

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

Relative Completeness Proof

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

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

Return

Proof (Relative Completeness, 10 pages).

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

Relative Completeness Proof

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

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

Return

Proof (Relative Completeness, 10 pages).

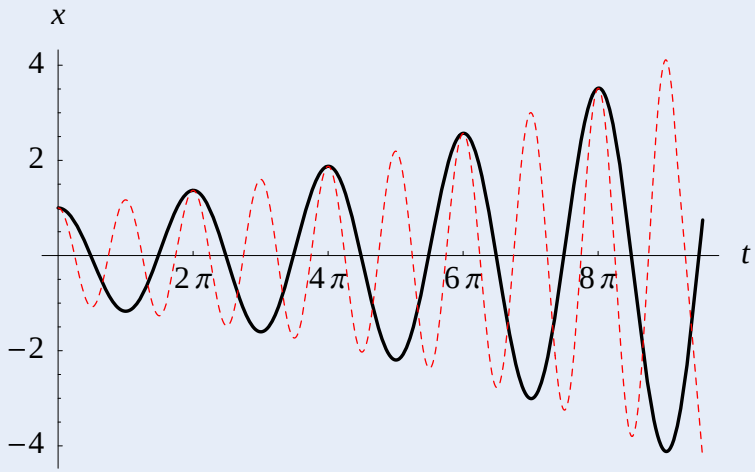
1. Strong invariants and variants expressible in d$\mathcal{L}$
2. d$\mathcal{L}$ expressible in FOD
3. valid d$\mathcal{L}$ formulas d$\mathcal{L}$-derivable from corresponding FOD axioms
4. finite FOD formula characterising unbounded hybrid repetition
5. FOD characterises $\mathbb{R}$-Gödel encoding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

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

Return

Proof ($\mathbb{R}$-Gödel encoding).

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

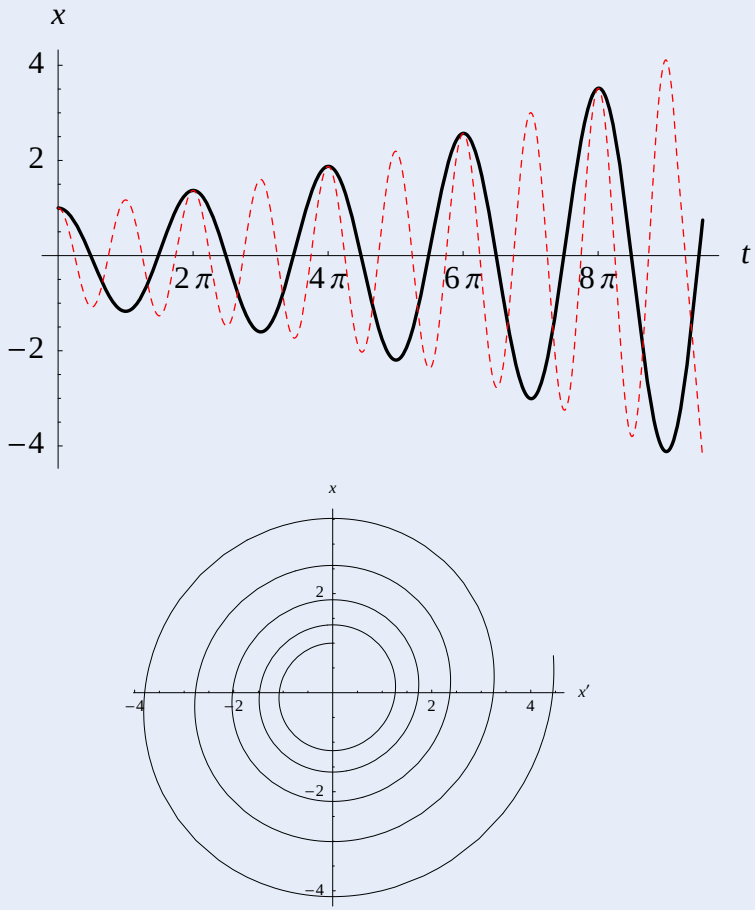


where $\mathsf{FOD} = \mathsf{FOL}_{\mathbb{R}} + [x_1' = \theta_1, \ldots, x_n' = \theta_n]F$

Return

Proof ($\mathbb{R}$-Gödel encoding).

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

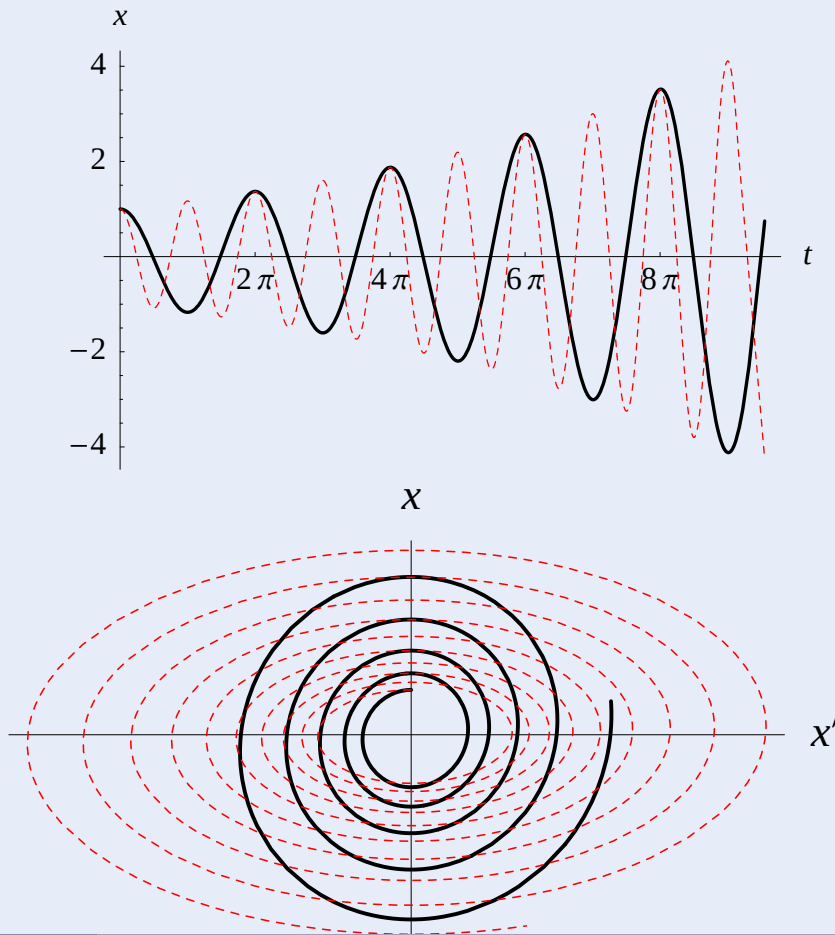


where $\mathsf{FOD} = \mathsf{FOL}_{\mathbb{R}} + [x_1' = \theta_1, \ldots, x_n' = \theta_n]F$

Return

Proof ($\mathbb{R}$-Gödel encoding).

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

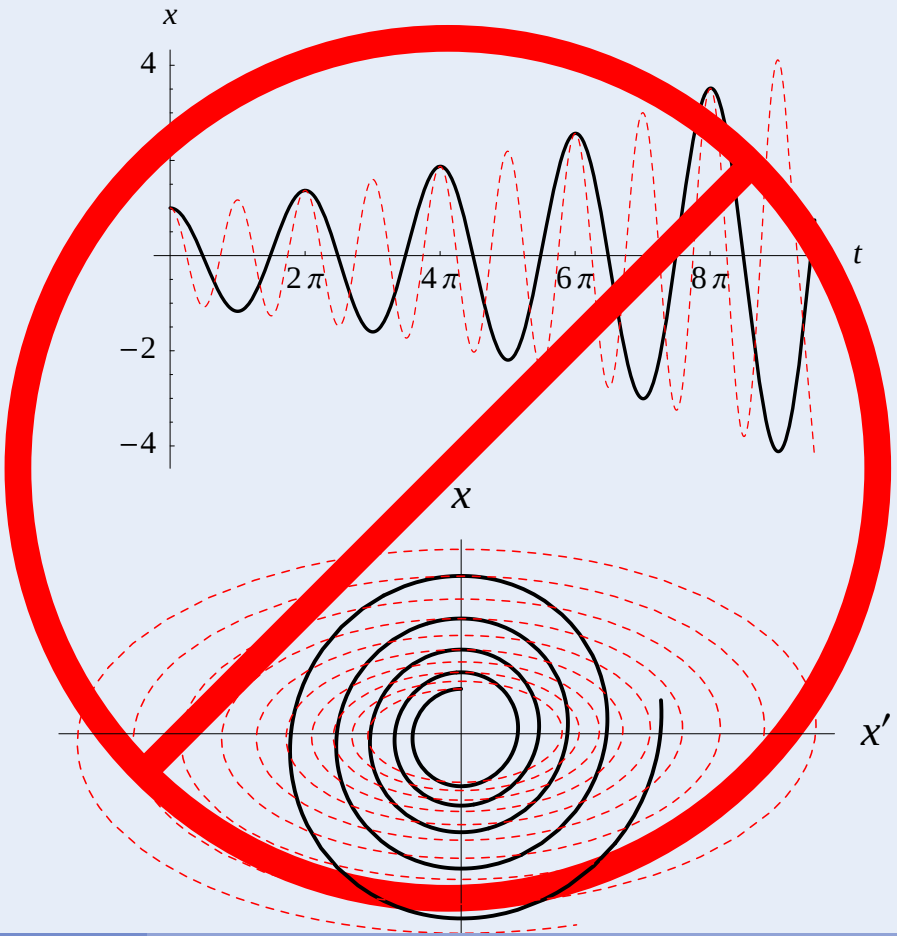


where $\text{FOD} = \text{FOL}_{\mathbb{R}} + [x_1' = \theta_1, \ldots, x_n' = \theta_n]F$

Return

Proof ($\mathbb{R}$-Gödel encoding).

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



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

Return

Proof ($\mathbb{R}$-Gödel encoding).

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

$$\sum_{i=1}^{\infty} \frac{a_i}{2^i} = 0.a_1 a_2 \ldots \qquad \sum_{i=1}^{\infty} \frac{b_i}{2^i} = 0.b_1 b_2 \ldots$$

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

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

Return

Proof ($\mathbb{R}$-Gödel encoding).

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

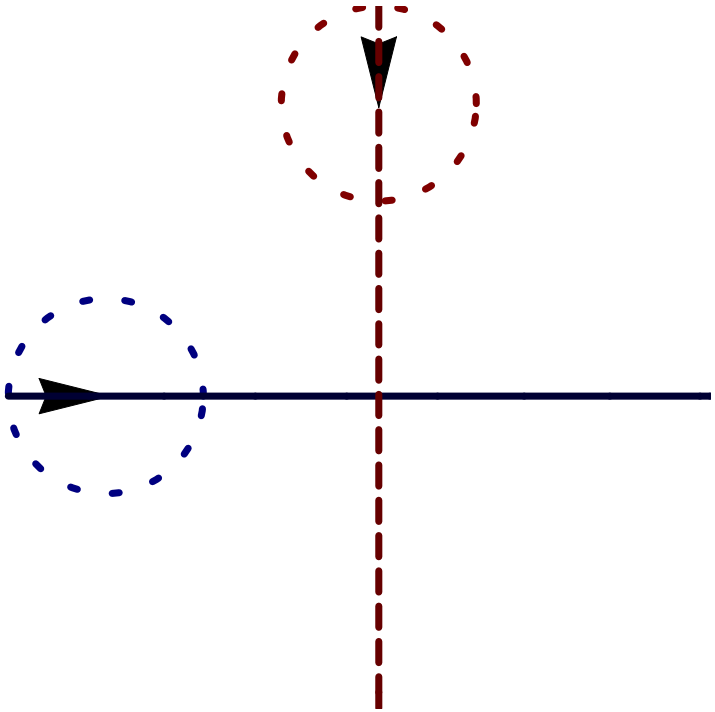
$$\sum_{i=1}^{\infty} \frac{a_i}{2^i} = 0.a_1 a_2 \ldots \qquad \sum_{i=1}^{\infty} \frac{b_i}{2^i} = 0.b_1 b_2 \ldots \quad \rightsquigarrow \quad \sum_{i=0}^{\infty} \left(\frac{a_i}{2^{2i+1}} + \frac{b_i}{2^{2i+2}} \right) = 0.a_1 b_1 a_2 b_2 \ldots$$

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

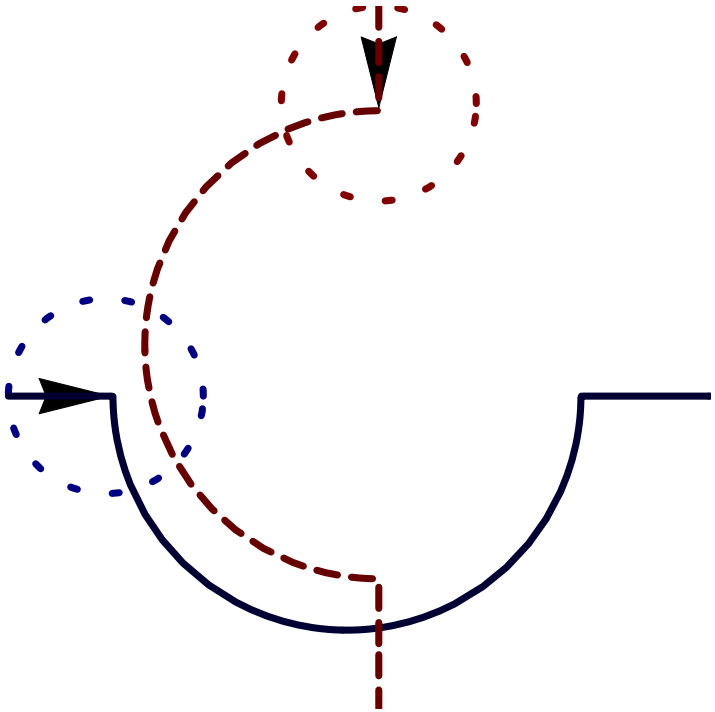
$$\ln 2 = z \quad \leftrightarrow \quad \langle x := 1; \tau := 0; x' = x \wedge \tau' = 1 \rangle (x = 2 \wedge \tau = z)$$

Outline

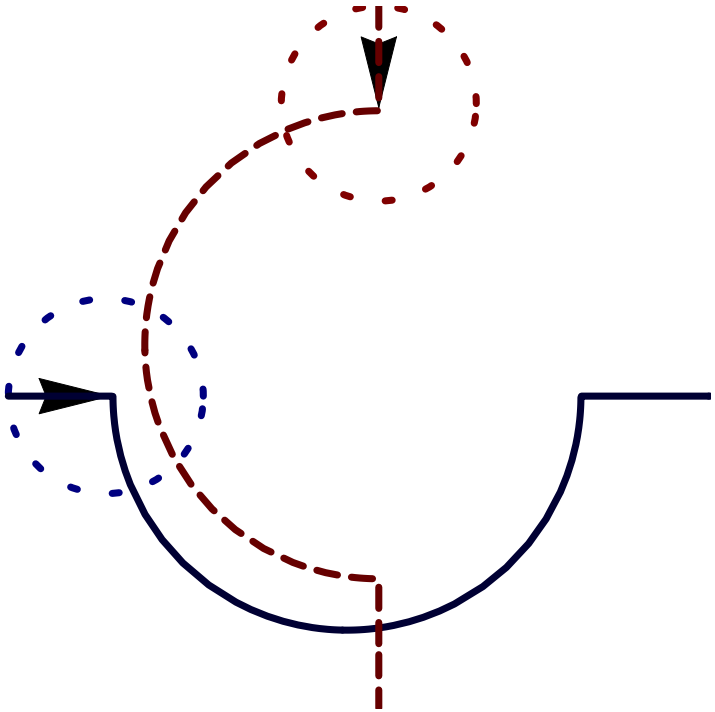
Air Traffic Control



Air Traffic Control



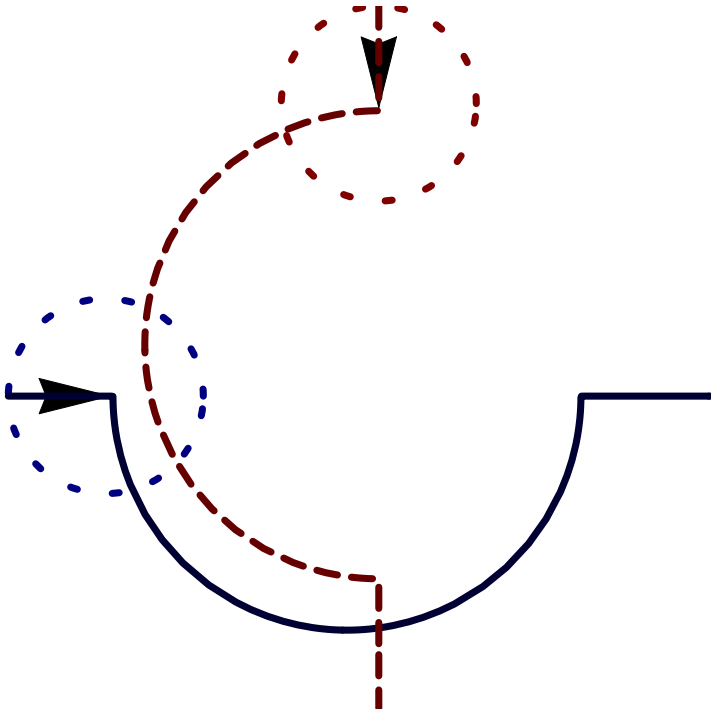
Air Traffic Control

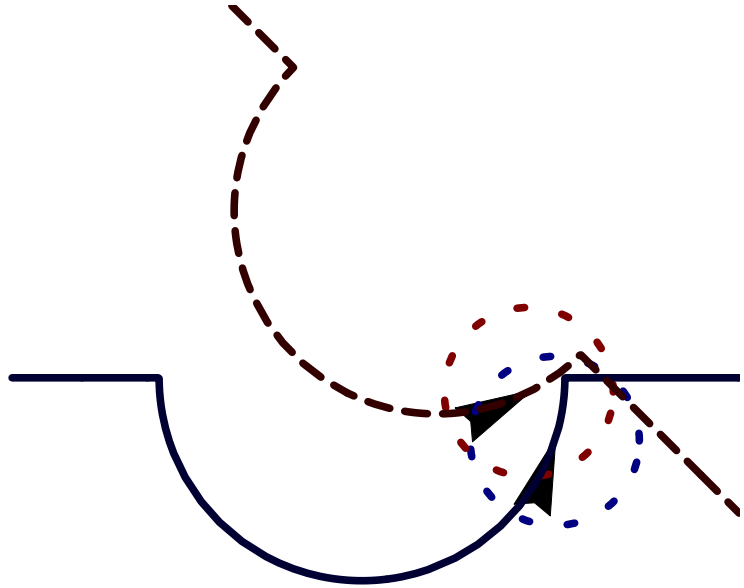


Verification?

looks correct

Air Traffic Control

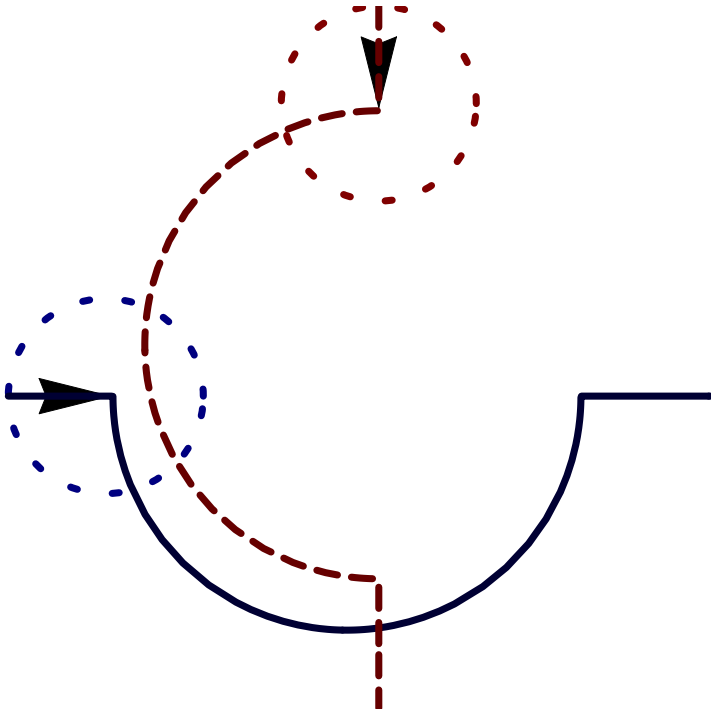


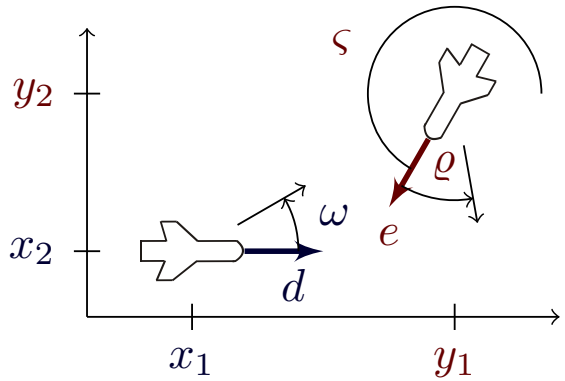


Verification?

looks correct NO!

Air Traffic Control



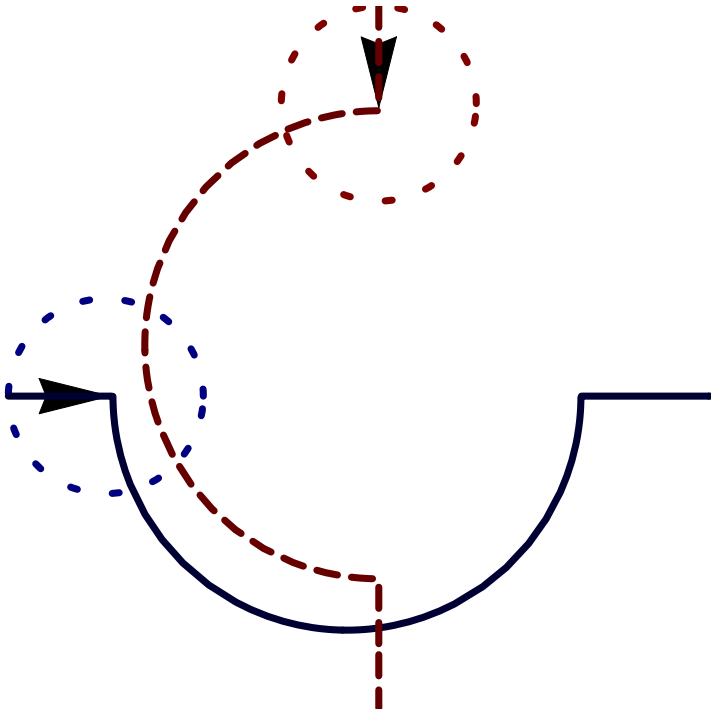


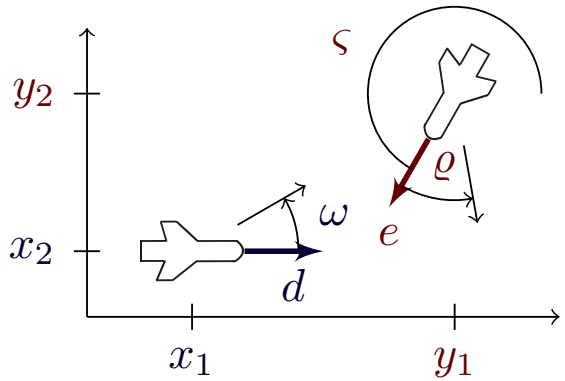
$$\begin{bmatrix} x_1' = -v_1 + v_2 \cos\vartheta + \omega x_2 \\ x_2' = v_2 \sin\vartheta - \omega x_1 \\ \vartheta' = \varpi - \omega \end{bmatrix}$$

Verification?

looks correct NO!

Air Traffic Control



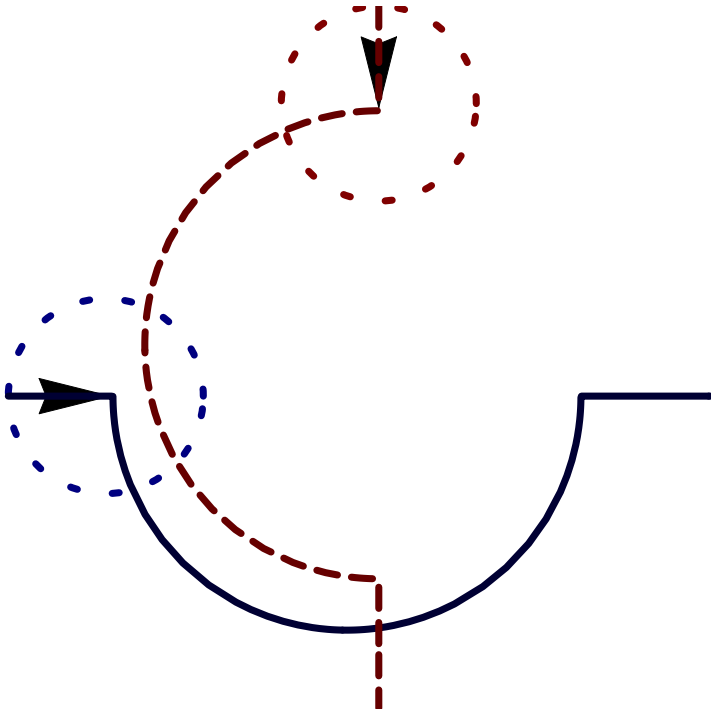


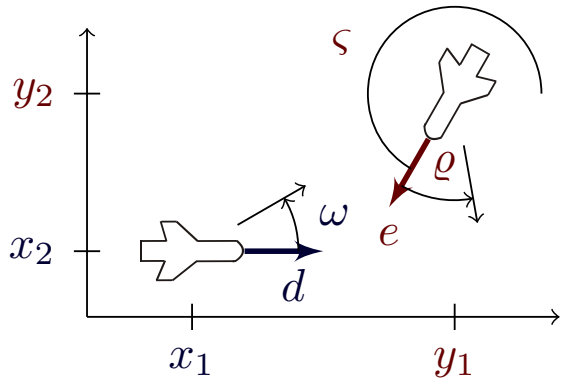
$$\begin{bmatrix} x_1' = -v_1 + v_2 \cos\vartheta + \omega x_2 \\ x_2' = v_2 \sin\vartheta - \omega x_1 \\ \vartheta' = \varpi - \omega \end{bmatrix}$$

Example ("Solving" differential equations)

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

Air Traffic Control





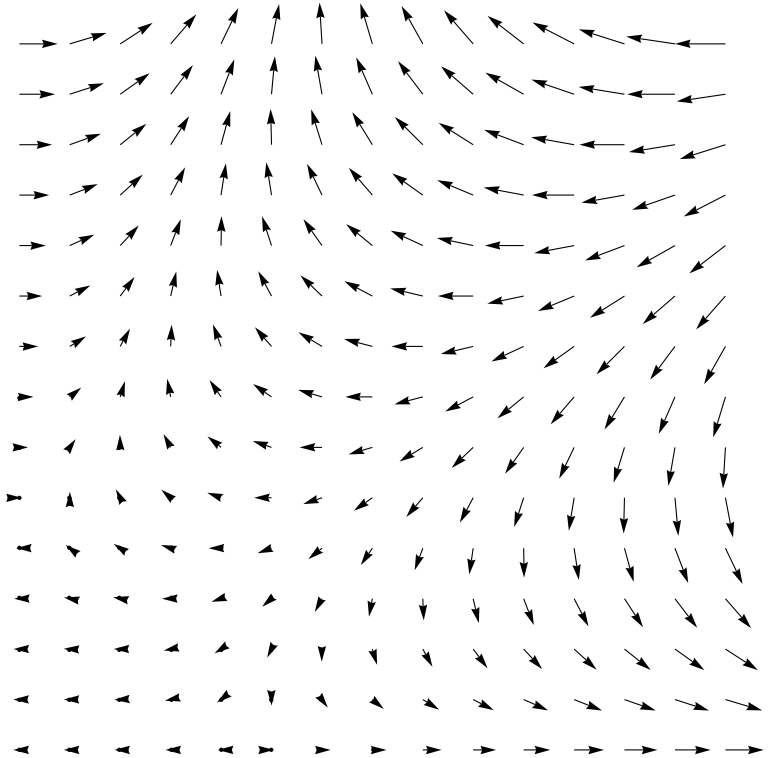
$$\begin{bmatrix} x_1' = -v_1 + v_2 \cos\vartheta + \omega x_2 \\ x_2' = v_2 \sin\vartheta - \omega x_1 \\ \vartheta' = \varpi - \omega \end{bmatrix}$$

Example ("Solving" differential equations)

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

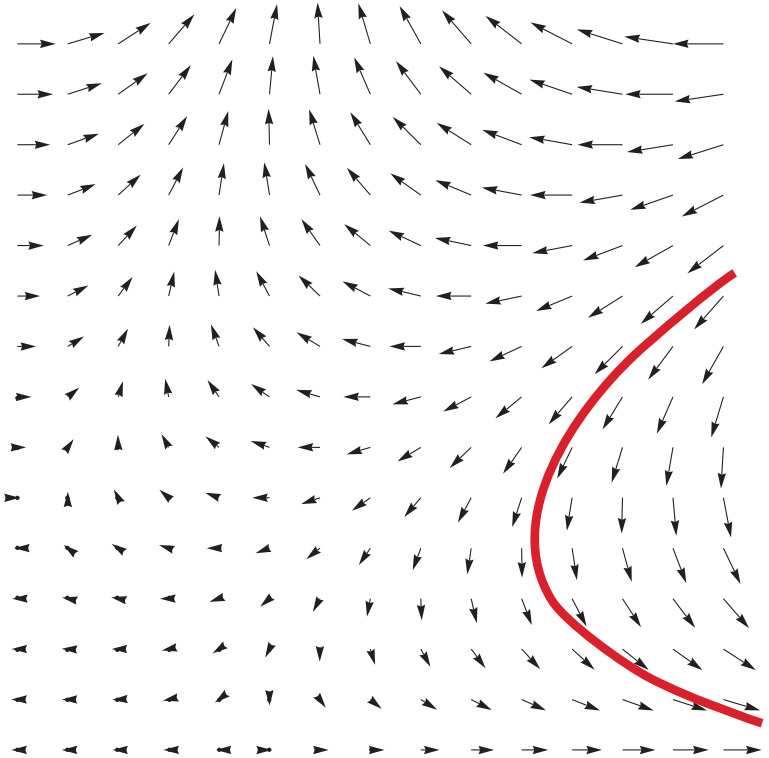
"Definition" (Differential Invariant)

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



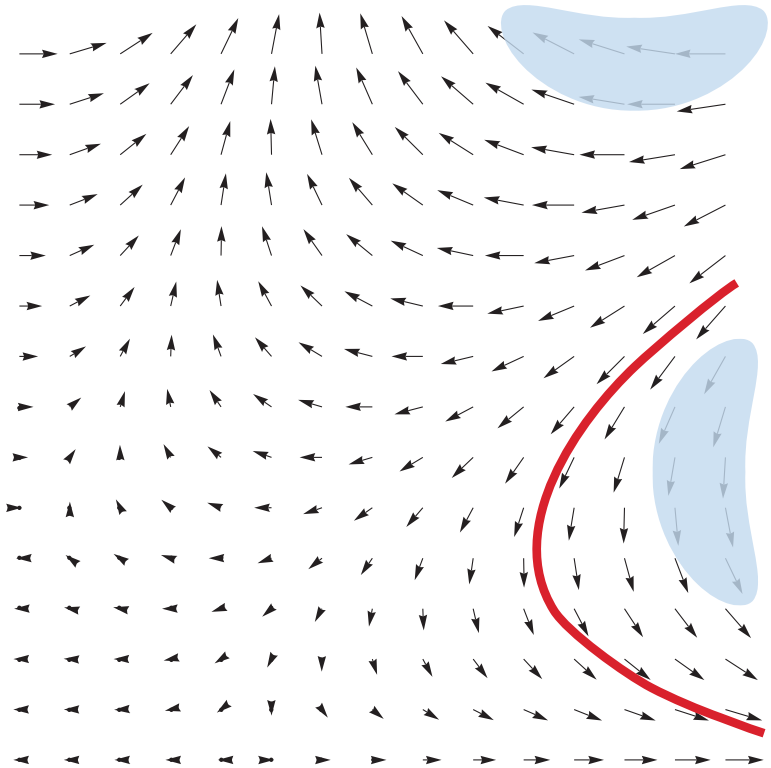
"Definition" (Differential Invariant)

"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)

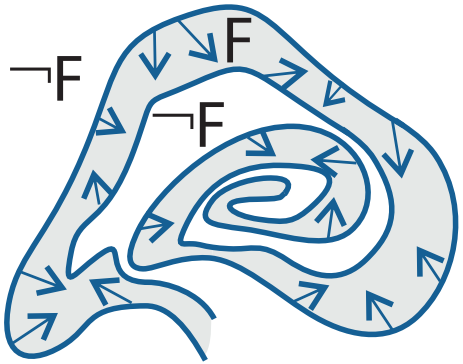
F closed under total differentiation with respect to differential constraints

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

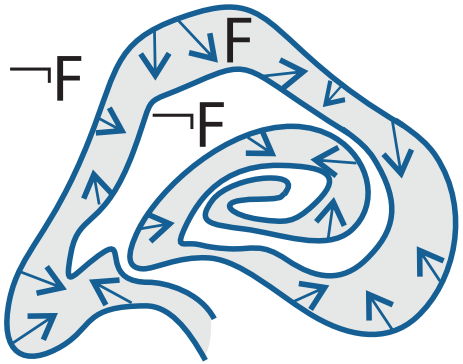


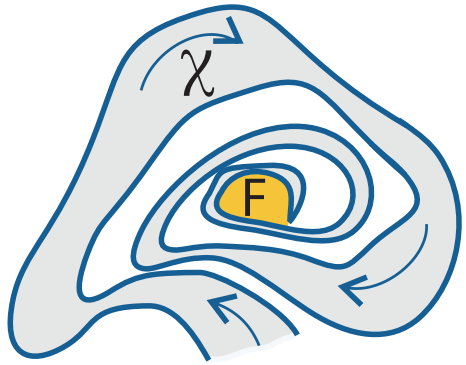
Details

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

Definition (Differential Invariant)

F closed under total differentiation with respect to differential constraints





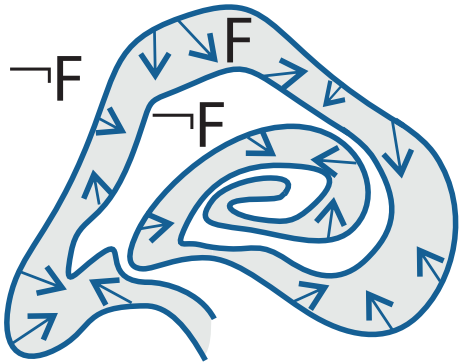
Details

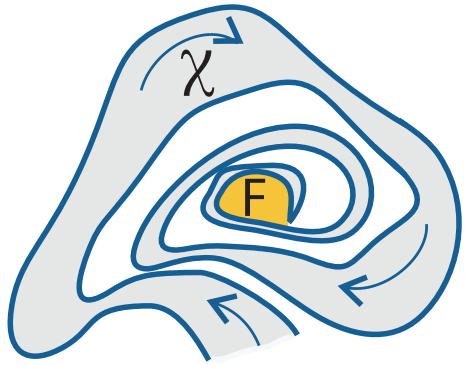
$$\frac{\vdash (\chi \to F')}{\chi \to F \vdash [x' = \theta \wedge \chi]F}$$

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

Definition (Differential Invariant)

F closed under total differentiation with respect to differential constraints





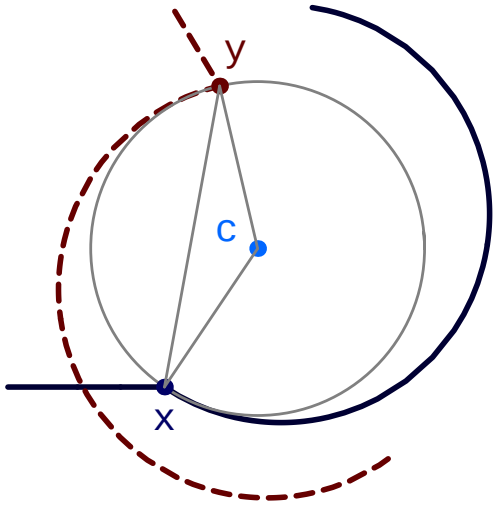
Details

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

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

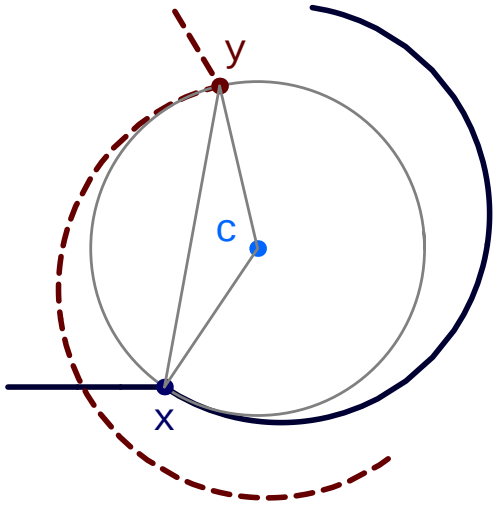
Total differential F' of *formulas*?

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



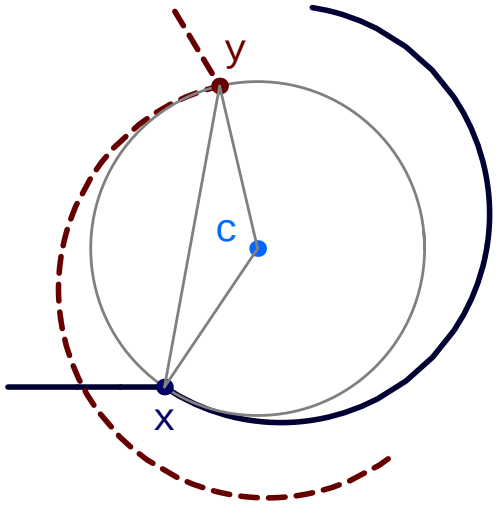
$$\vdash \frac{\partial \|x-y\|^2}{\partial x_1} x_1' + \frac{\partial \|x-y\|^2}{\partial y_1} y_1' + \frac{\partial \|x-y\|^2}{\partial x_2} x_2' + \frac{\partial \|x-y\|^2}{\partial y_2} y_2' \geq \frac{\partial p^2}{\partial x_1} x_1' \dots$$

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



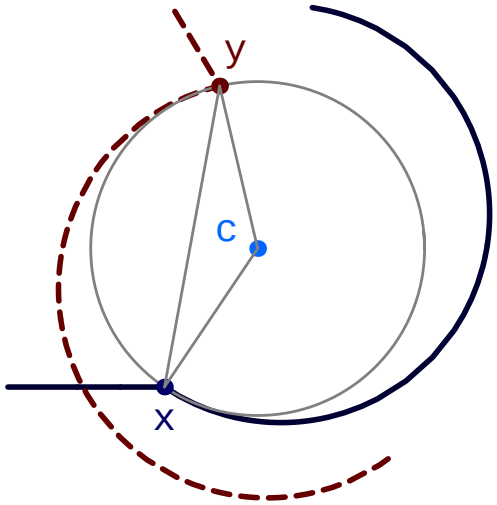
$$\vdash \frac{\partial \|x-y\|^2}{\partial x_1} x_1' + \frac{\partial \|x-y\|^2}{\partial y_1} y_1' + \frac{\partial \|x-y\|^2}{\partial x_2} x_2' + \frac{\partial \|x-y\|^2}{\partial y_2} y_2' \geq \frac{\partial p^2}{\partial x_1} x_1' \dots$$

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



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

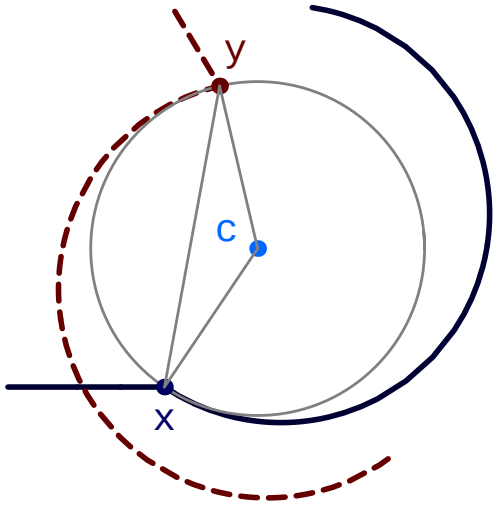
$$\vdash [x_1' = d_1, d_1' = -\omega d_2, x_2' = d_2, d_2' = \omega d_1, ..](x_1 - y_1)^2 + (x_2 - y_2)^2 \geq p^2$$

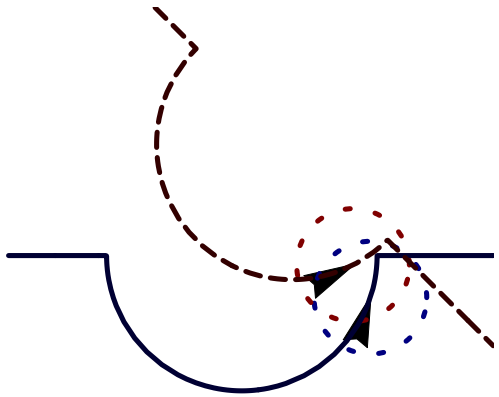


$$\vdash 2(x_1 - y_1)(d_1 - e_1) + 2(x_2 - y_2)(d_2 - e_2) \geq 0$$

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

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

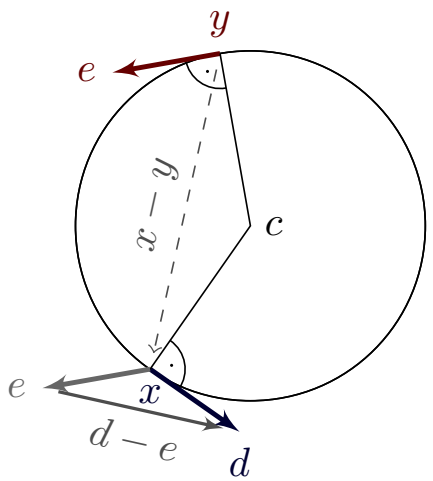


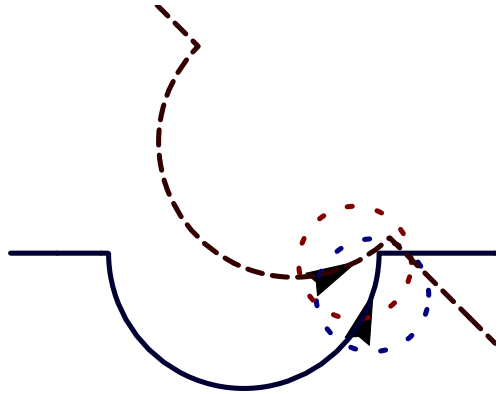


$$\vdash 2(x_1 - y_1)(d_1 - e_1) + 2(x_2 - y_2)(d_2 - e_2) \geq 0$$

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

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

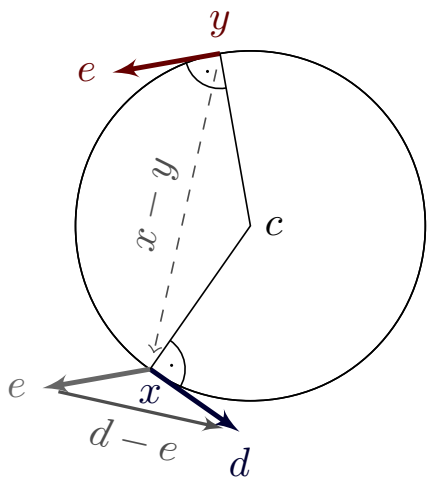


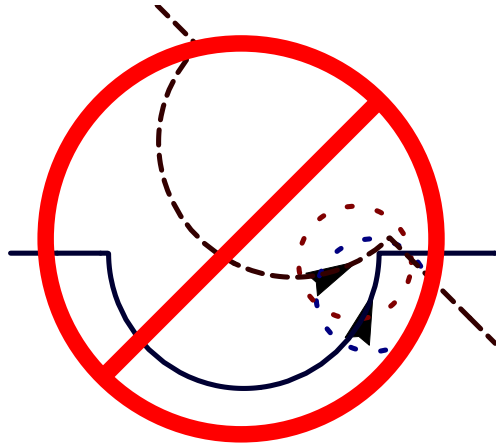


$$\vdash 2(x_1 - y_1)(d_1 - e_1) + 2(x_2 - y_2)(d_2 - e_2) \geq 0$$

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

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





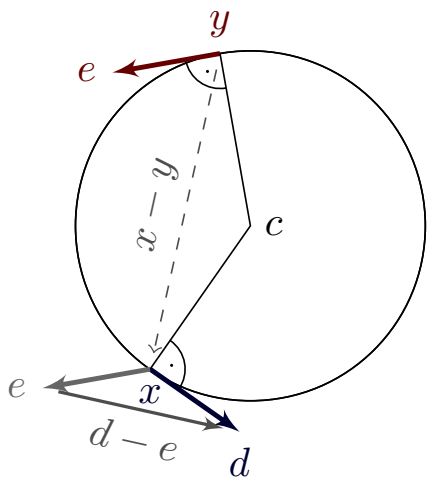
$$.. \vdash [d_1' = -\omega d_2, e_1' = -\omega e_2, x_2' = d_2, d_2' = \omega d_1, ..] d_1 - e_1 = -\omega(x_2 - y_2)$$

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

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

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

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



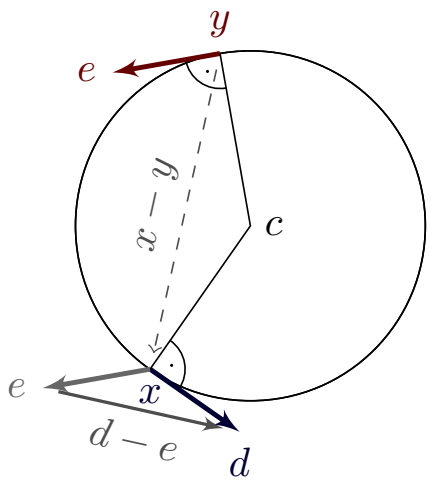
$$.. \vdash [d_1'=-\omega d_2,e_1'=-\omega e_2,x_2'=d_2,d_2'=\omega d_1,..]d_1-e_1=-\omega(x_2-y_2)$$

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

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

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

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



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

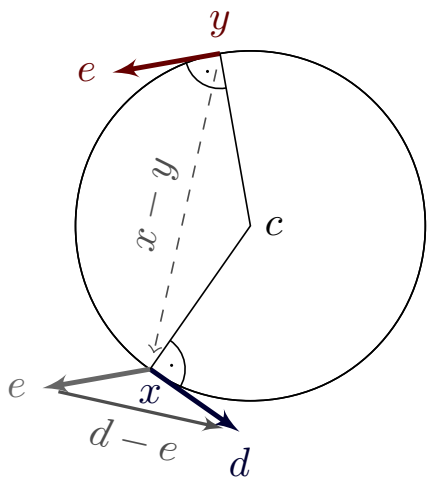
$$..\vdash [d_1'=-\omega d_2,e_1'=-\omega e_2,x_2'=d_2,d_2'=\omega d_1,..]d_1-e_1=-\omega(x_2-y_2)$$

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

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

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

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



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

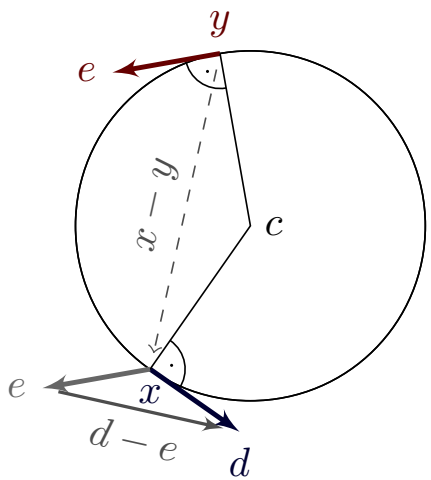
$$..\vdash [d_1'=-\omega d_2,e_1'=-\omega e_2,x_2'=d_2,d_2'=\omega d_1,..]d_1-e_1=-\omega(x_2-y_2)$$

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

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

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

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



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

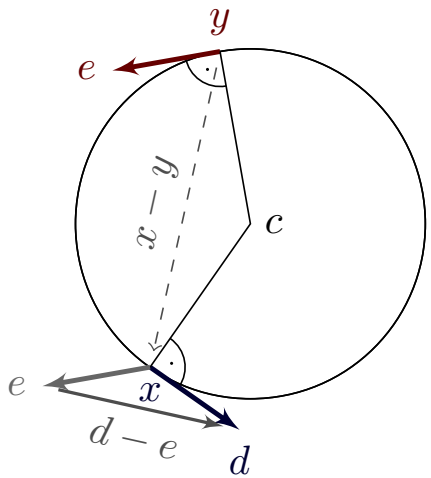
$$..\vdash [d_1'=-\omega d_2,e_1'=-\omega e_2,x_2'=d_2,d_2'=\omega d_1,..]d_1-e_1=-\omega(x_2-y_2)$$

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

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

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

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



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

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

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

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

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

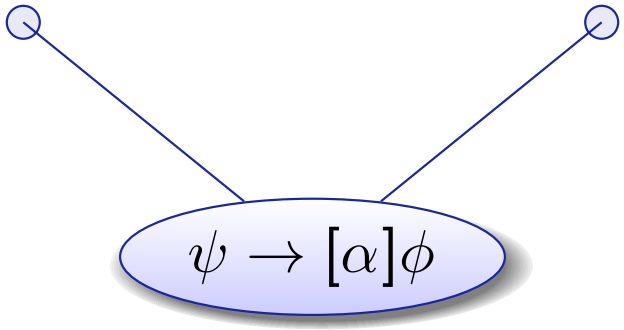
Differential Invariants as Fixedpoints

$\psi_3 \rightarrow [\alpha_3]\phi_3$

$\psi_4 \rightarrow [\alpha_4]\phi_4$

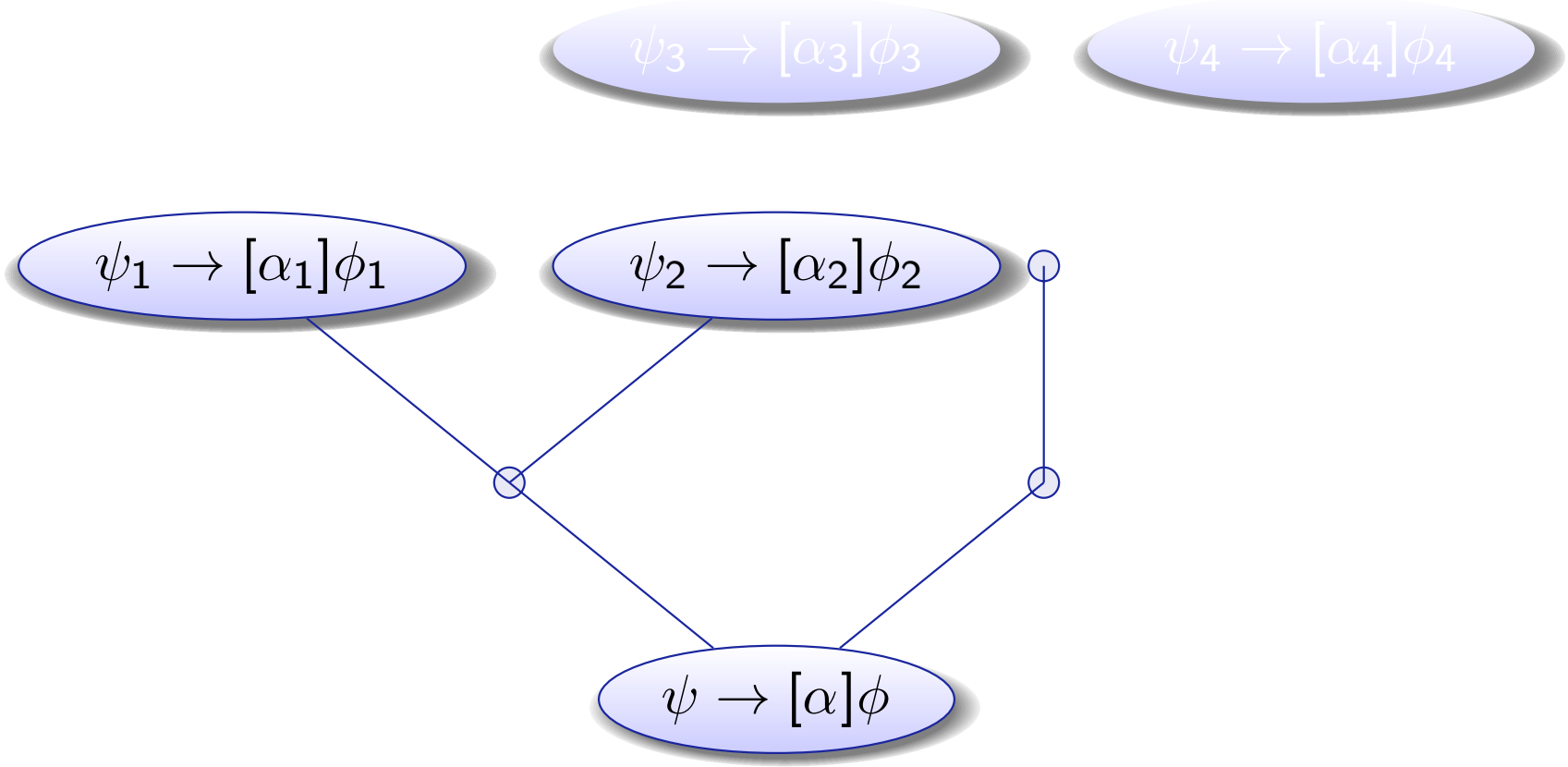
$\psi_1 \rightarrow [\alpha_1]\phi_1$

$\psi_2 \rightarrow [\alpha_2]\phi_2$



Details

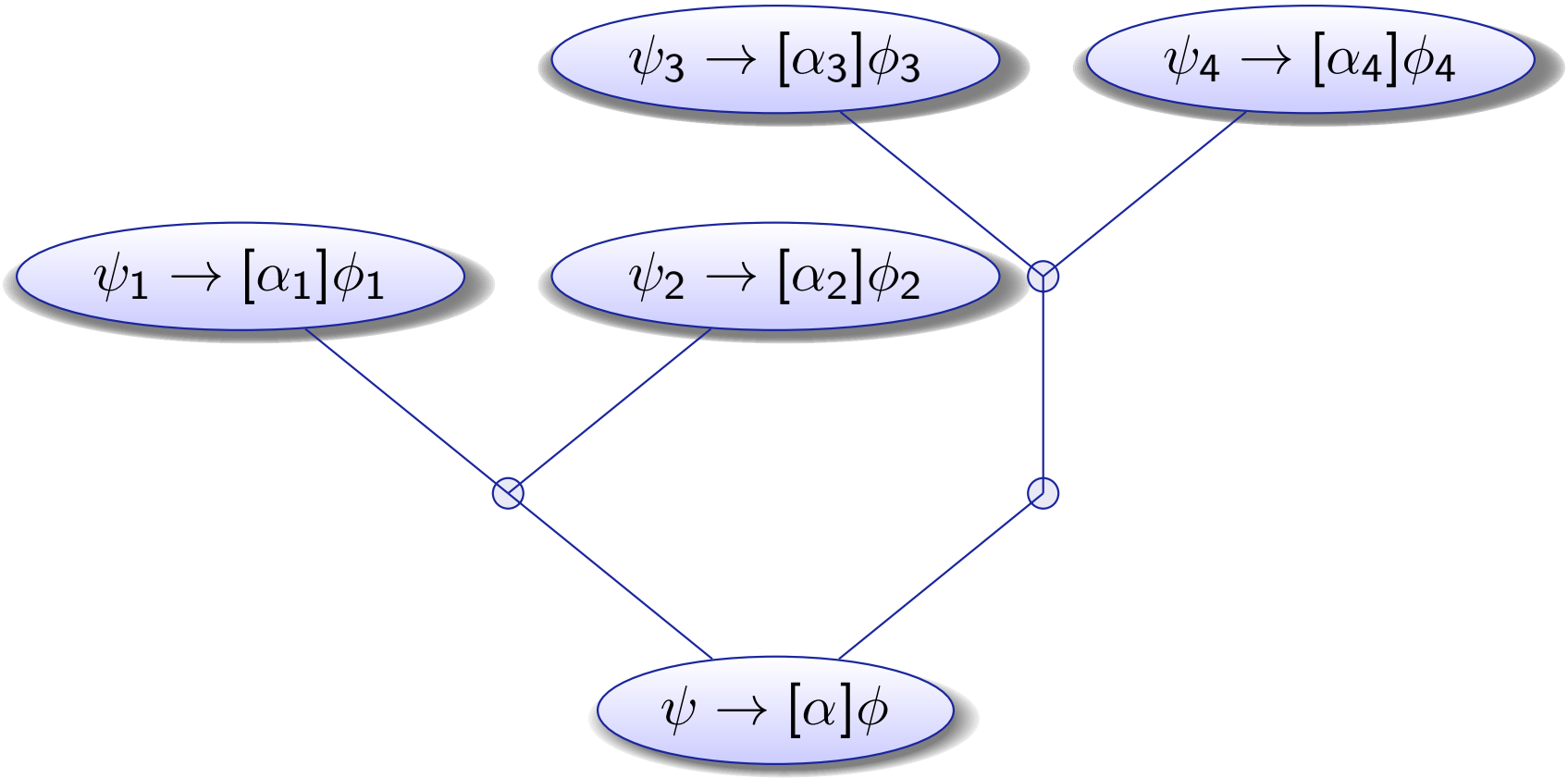
Differential Invariants as Fixedpoints



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

Details

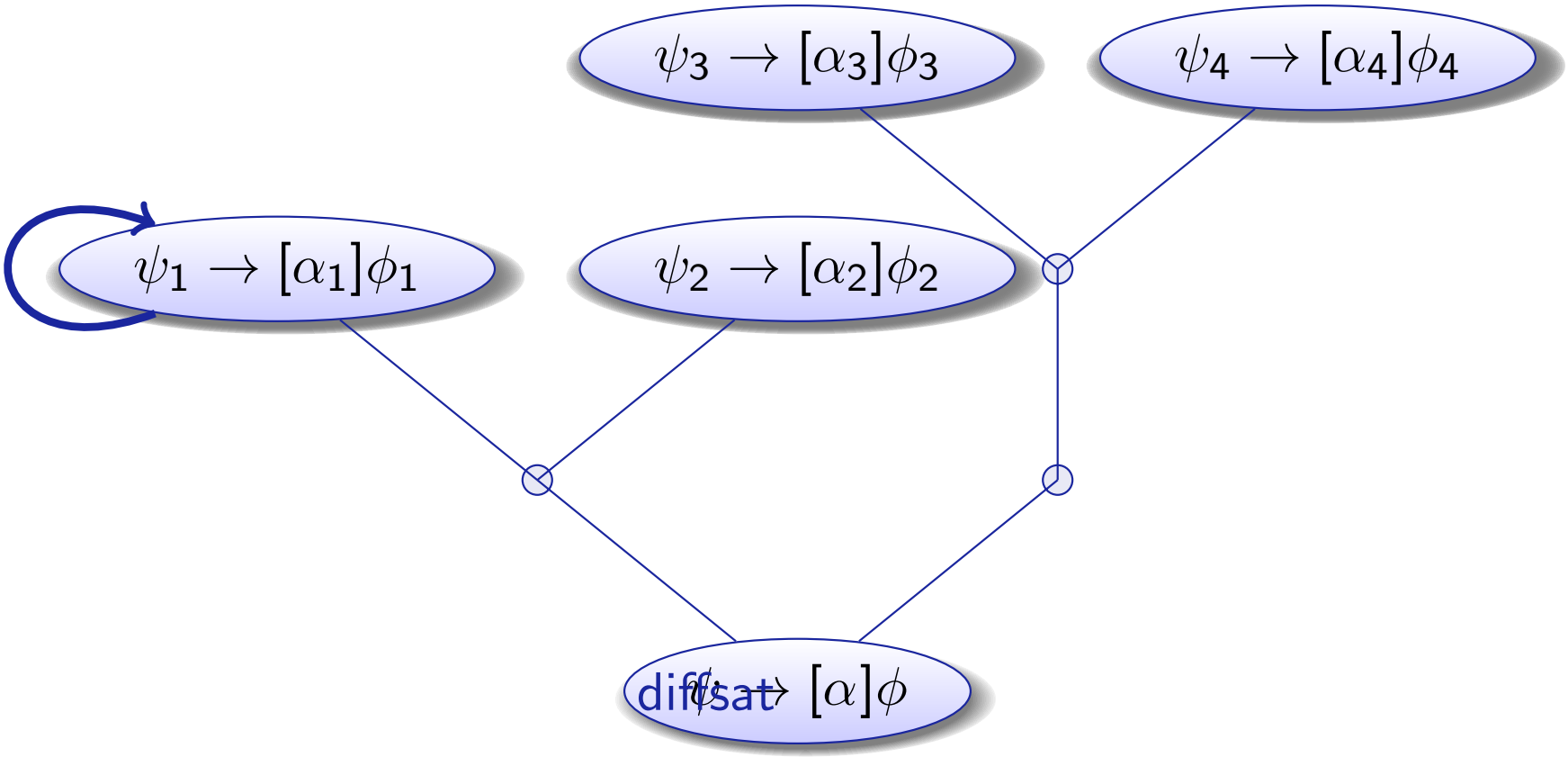
Differential Invariants as Fixedpoints



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

Details

Differential Invariants as Fixedpoints

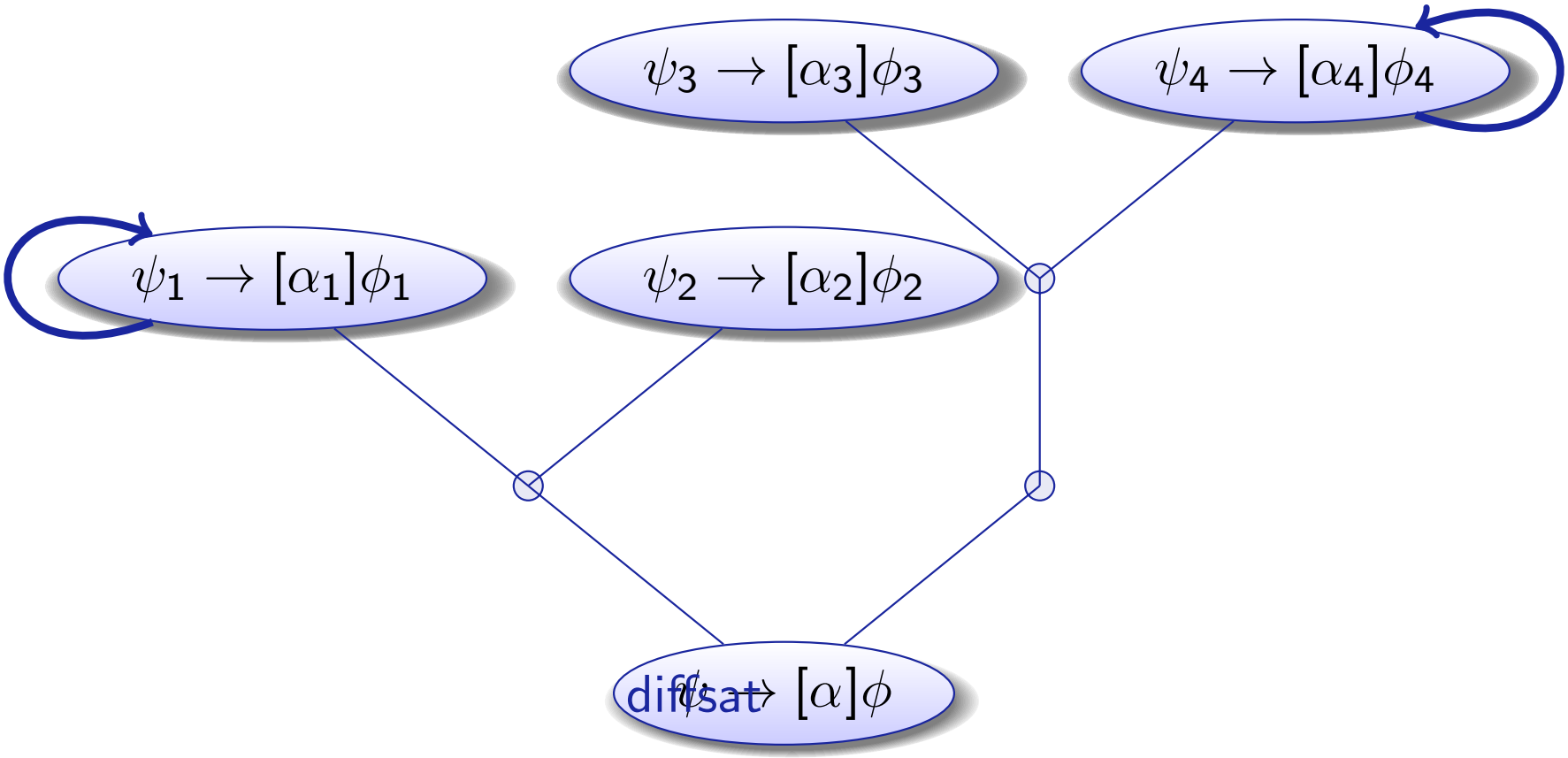


for $\cup, ;, :=$ do decompose

for $x' = \ldots$ do diffsat

▸ Details

Differential Invariants as Fixedpoints

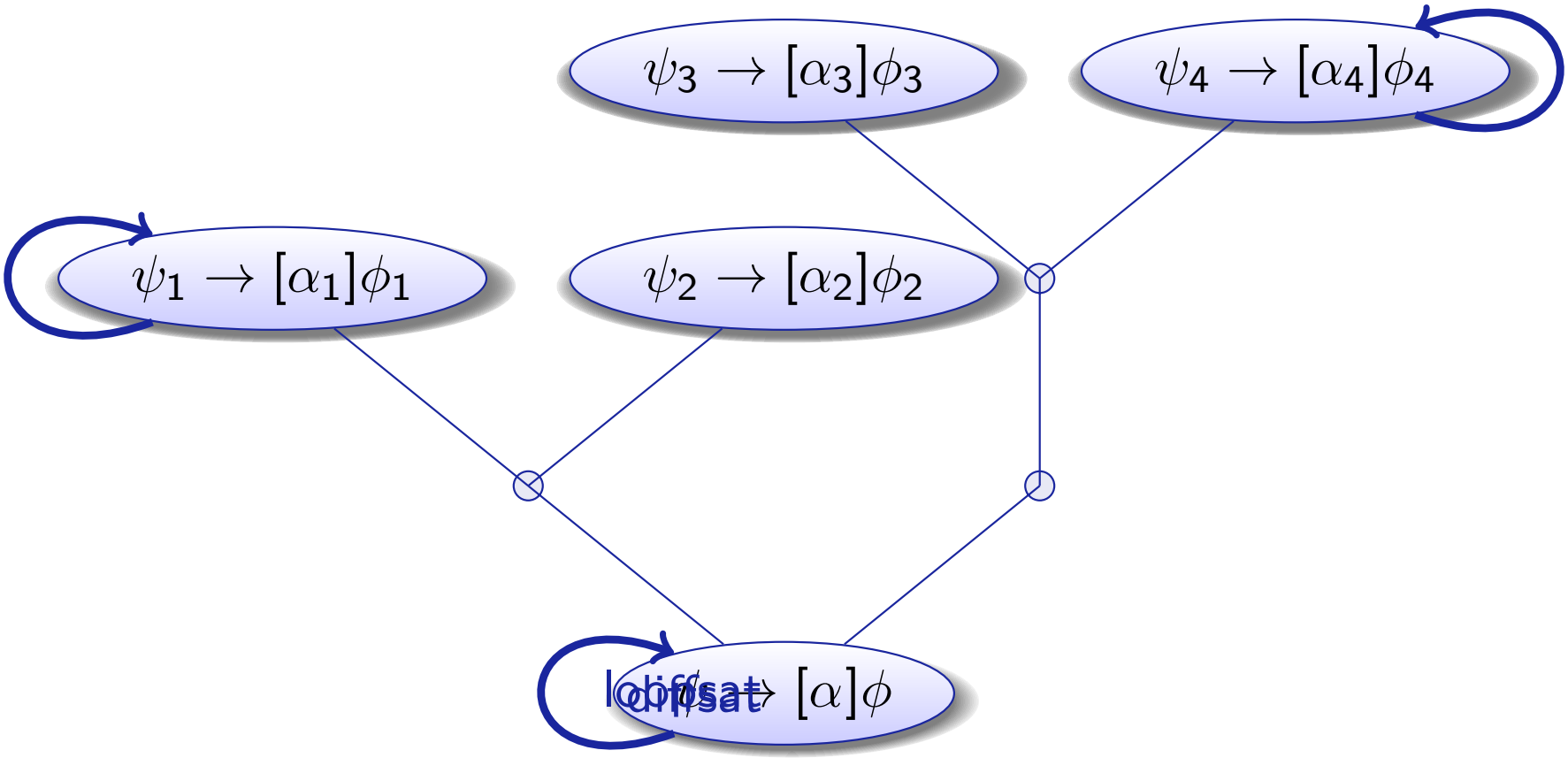


for $\cup, ;, :=$ do decompose

for $x' = \ldots$ do diffsat

Details

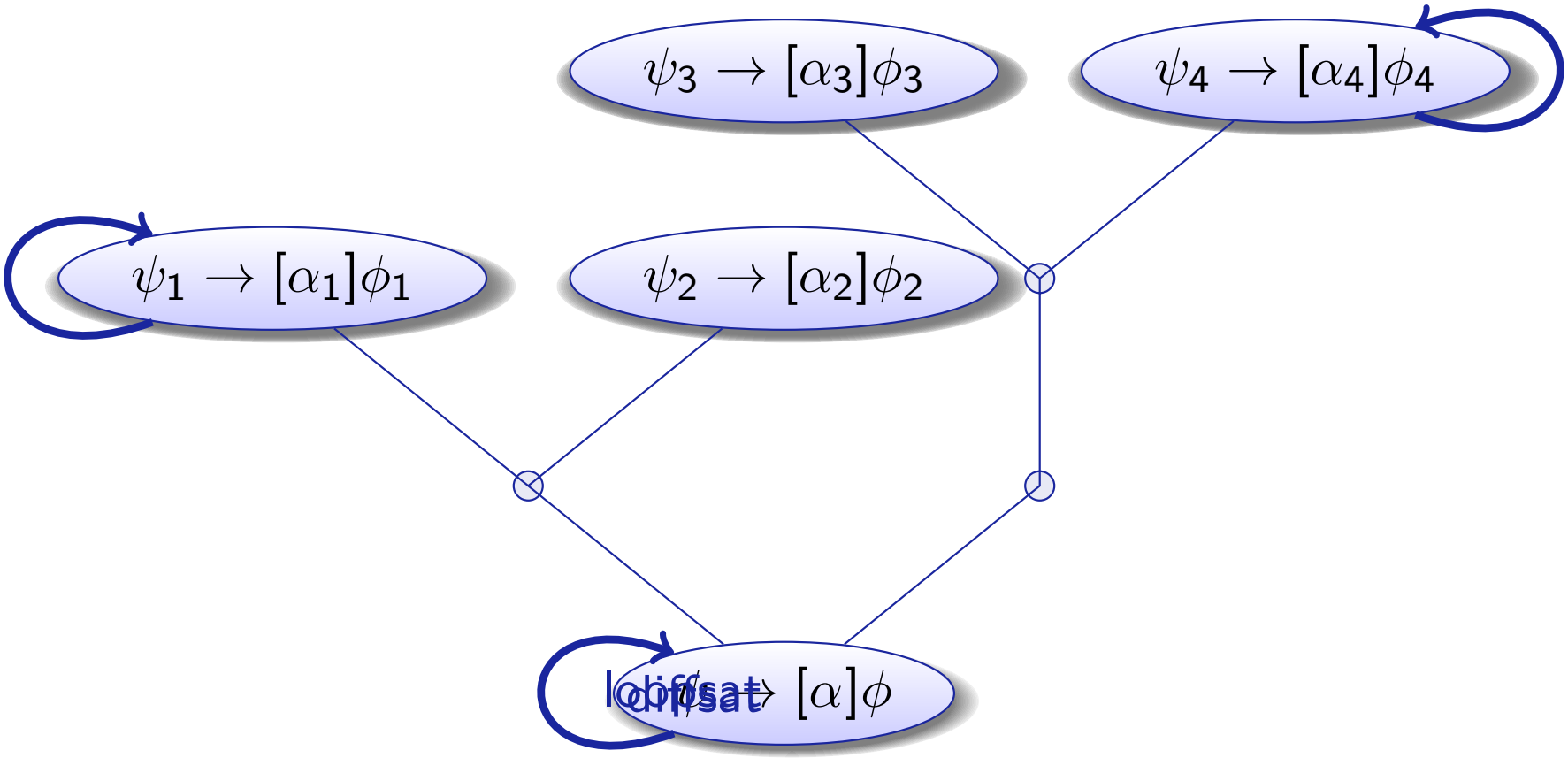
Differential Invariants as Fixedpoints



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

Details

Differential Invariants as Fixedpoints



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

} repeat until fixedpoint

Details

Differential Induction Principle

$$\sigma_1 \quad \mapsto \quad [\![F]\!]_{\sigma_1}$$

Return

Differential Induction Principle

$$\begin{array}{lcl} \sigma_1 & \mapsto & [\![F]\!]_{\sigma_1} \\ \sigma_2 & \mapsto & [\![F]\!]_{\sigma_2} \end{array}$$

Return

Differential Induction Principle

$$\begin{array}{lcl} \sigma_1 & \mapsto & [\![F]\!]_{\sigma_1} \\ \sigma_2 & \mapsto & [\![F]\!]_{\sigma_2} \end{array}$$

In the limit:

$$\frac{\mathrm{d}\,[\![F]\!]_{\sigma}}{\mathrm{d}\sigma}$$

Return

$$\begin{array}{lcl} \sigma_1 & \mapsto & [\![F]\!]_{\sigma_1} \\ \sigma_2 & \mapsto & [\![F]\!]_{\sigma_2} \end{array}$$

In the limit:

$$\frac{\mathrm{d}\,[\![F]\!]_{\sigma(t)}}{\mathrm{d}t}$$

where $\frac{\mathrm{d}\sigma(t)}{\mathrm{d}t}$ according to ODE

Return

Differential Induction Principle

$$\begin{array}{lcl} \sigma_1 & \mapsto & [\![F]\!]_{\sigma_1} \\ \sigma_2 & \mapsto & [\![F]\!]_{\sigma_2} \end{array}$$

In the limit:

$$\frac{\mathrm{d}\,[\![F]\!]_{\sigma(t)}}{\mathrm{d}t}(\zeta) = [\![F']\!]_{\bar{\sigma}(\zeta)}$$

where $\frac{\mathrm{d}\sigma(t)}{\mathrm{d}t}$ according to ODE

Return

$$\begin{array}{lcl} \sigma_1 & \mapsto & [\![F]\!]_{\sigma_1} \\ \sigma_2 & \mapsto & [\![F]\!]_{\sigma_2} \end{array}$$

In the limit:

$$\frac{\mathrm{d}\,[\![F]\!]_{\sigma(t)}}{\mathrm{d}t}(\zeta) = [\![F']\!]_{\bar{\sigma}(\zeta)}$$

where $\frac{\mathrm{d}\sigma(t)}{\mathrm{d}t}$ according to ODE

Lemma (Derivation lemma)

Valuation is a differential homomorphism

Return

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

$$D(r) = 0 \quad \text{if } r \text{ is a (rigid) number symbol}$$

$$D(x^{(n)}) = x^{(n+1)} \quad \text{if } x \in \Sigma \text{ is non-rigid}, n \geq 0$$

$$D(a + b) = D(a) + D(b)$$

$$D(a \cdot b) = D(a) \cdot b + a \cdot D(b)$$

$$D(a/b) = (D(a) \cdot b - a \cdot D(b))/b^2$$

$$D(F) \equiv \bigwedge_{i=1}^{m} D(F_i) \quad \{F_1, \ldots, F_m\} \text{ all literals of } F$$

$$D(a \geq b) \equiv D(a) \geq D(b) \quad \text{accordingly for } <, >, \leq, =$$

Return

Lemma (Derivation lemma)

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

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

Lemma (Differential substitution principle)

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

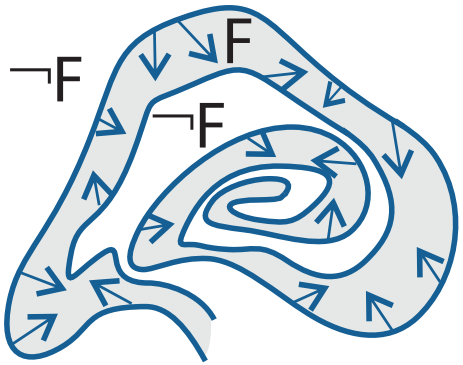
Definition (Differential Invariant)

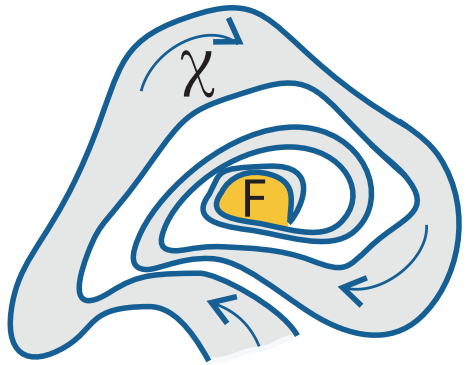
$$(\chi \rightarrow F') \;\equiv\; \chi \rightarrow D(F)^{\theta_i}_{x_i'} \qquad \text{for } [x_i' = \theta_i \wedge \chi]F$$

Return

Definition (Differential Invariant)

F closed under total differentiation with respect to differential constraints





Details

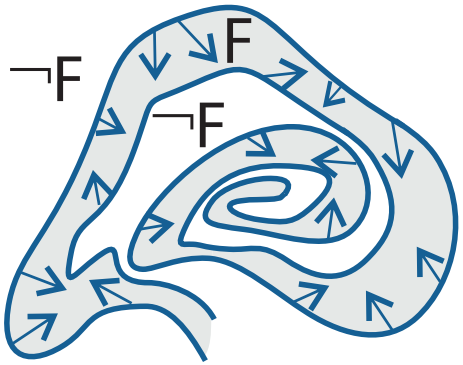
$$\frac{\vdash (\chi \to F')}{\chi \to F \vdash [x' = \theta \wedge \chi]F}$$

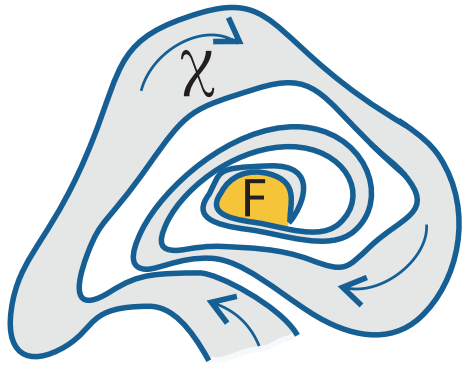
$$\frac{\vdash (\neg F \wedge \chi \to F'_{\gg})}{[x' = \theta \wedge \neg F]\chi \vdash \langle x' = \theta \wedge \chi \rangle F}$$

$$\begin{aligned}(d_1^2 + d_2^2 \geq a^2)' \quad &\equiv \quad \frac{\partial(d_1^2 + d_2^2)}{\partial d_1} d_1' + \frac{\partial(d_1^2 + d_2^2)}{\partial d_2} d_2' \geq \frac{\partial a^2}{\partial d_1} d_1' + \frac{\partial a^2}{\partial d_2} d_2' \\ &\equiv \quad 2d_1(-\omega d_2) + 2d_2(\omega d_1) \geq 0 \\ \text{for} \quad &d_1' = -\omega d_2 \quad d_2' = \omega d_1\end{aligned}$$

Definition (Differential Invariant)

F closed under total differentiation with respect to differential constraints



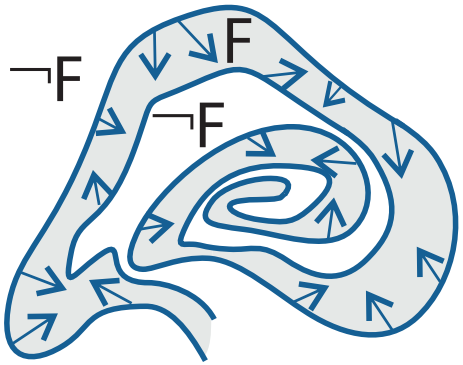


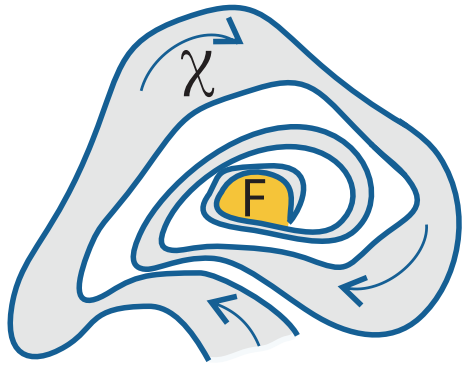
Details

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

Definition (Differential Invariant)

F closed under total differentiation with respect to differential constraints



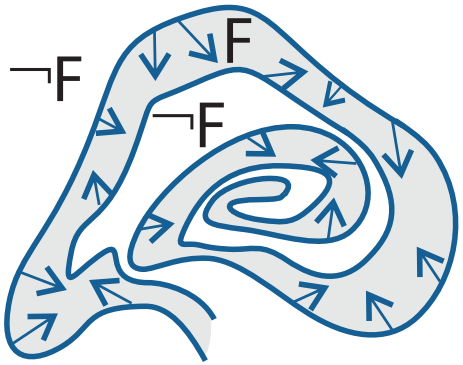


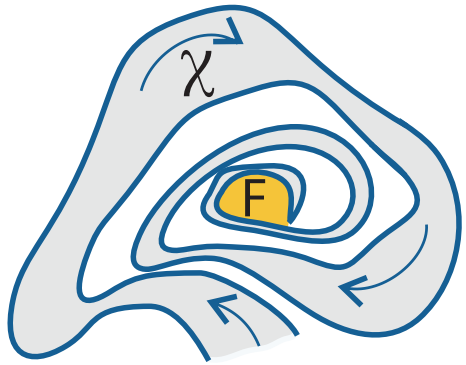
Details

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

Definition (Differential Invariant)

F closed under total differentiation with respect to differential constraints





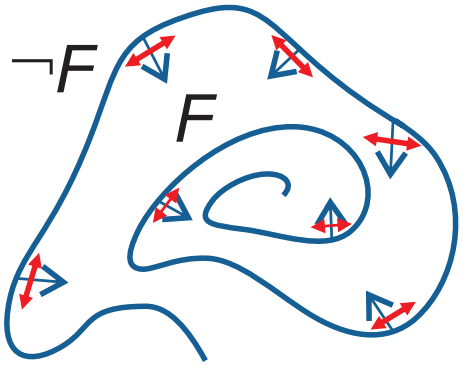
Details

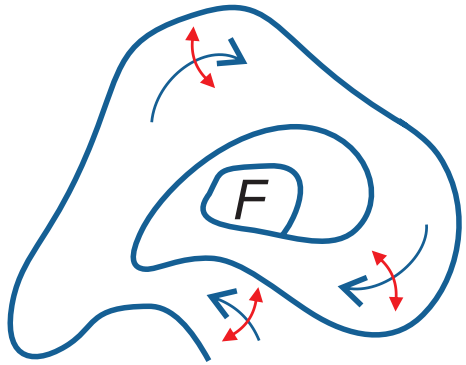
$$d_1 \geq d_2 \rightarrow [x := a^2 + 1;$$
$$\exists \omega \, (\omega \leq 1 \wedge d_1' = -\omega d_2 \wedge d_2' = \omega d_1) \vee (d_1' \leq 2 d_1)$$
$$] \, d_1 \geq d_2$$

Differential Induction: Local Dynamics w/o Solutions

Definition (Differential Invariant)

F closed under total differentiation with respect to differential constraints





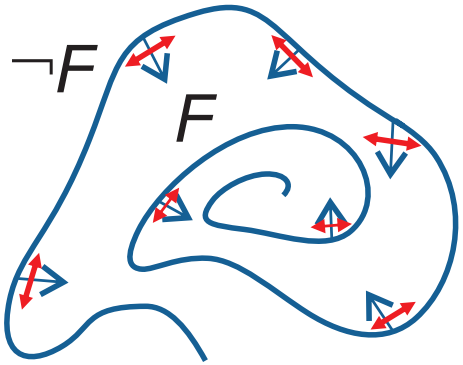
▸ Details

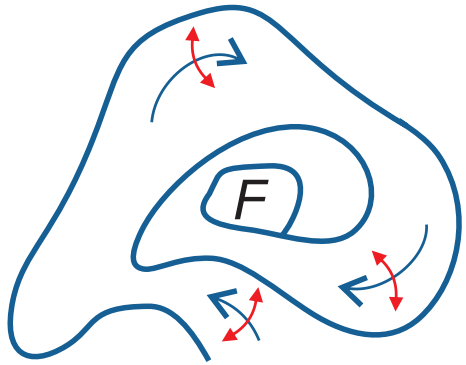
$$d_1 \geq d_2 \rightarrow [x := a^2 + 1;$$
$$\exists \omega \, (\omega \leq 1 \wedge d_1' = -\omega d_2 \wedge d_2' = \omega d_1) \vee (d_1' \leq 2 d_1)$$
$$] \, d_1 \geq d_2$$

- quantified nondeterminism/disturbance

Definition (Differential Invariant)

F closed under total differentiation with respect to differential constraints





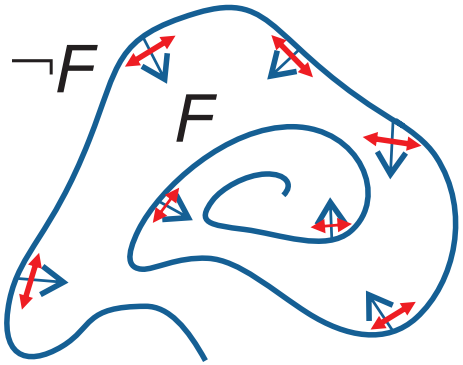
Details

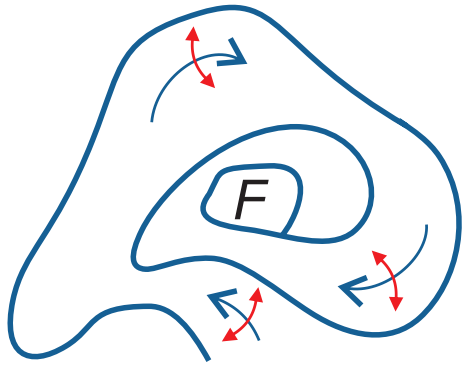
$$d_1 \geq d_2 \rightarrow [x := a^2 + 1;$$
$$\exists \omega \, (\omega \leq 1 \wedge d_1' = -\omega d_2 \wedge d_2' = \omega d_1) \vee (d_1' \leq 2d_1)$$
$$] \, d_1 \geq d_2$$

- quantified nondeterminism/disturbance

Definition (Differential Invariant)

F closed under total differentiation with respect to differential constraints





Details

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

- discrete quantified nondeterminism/disturbance

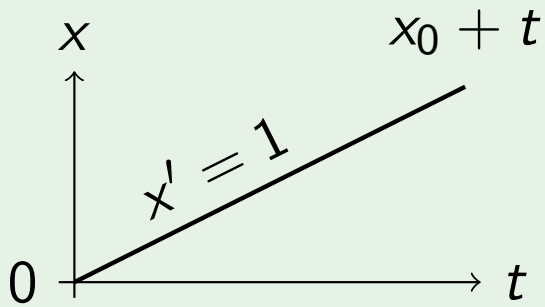
Counterexample

$$\frac{\vdash \forall x\,(x^2 \leq 0 \rightarrow 2x \cdot 1 \leq 0)}{x^2 \leq 0 \vdash [x' = 1]x^2 \leq 0}$$

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

Counterexample

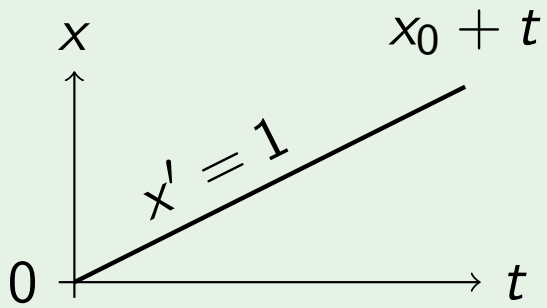
$$\frac{\vdash \forall x\,(x^2 \leq 0 \rightarrow 2x \cdot 1 \leq 0)}{x^2 \leq 0 \vdash [x' = 1]x^2 \leq 0}$$



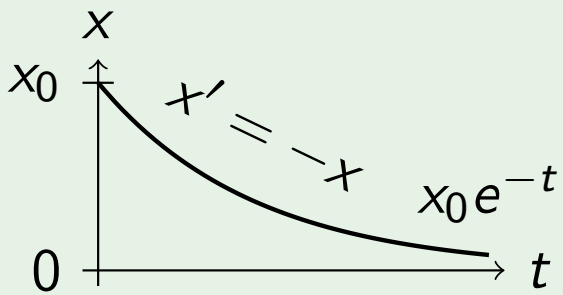
$$\frac{\vdash \forall x\,(x > 0 \rightarrow -x < 0)}{\vdash \langle x' = -x\rangle x \leq 0}$$

Counterexample

$$\frac{\vdash \forall x\,(x^2 \leq 0 \rightarrow 2x \cdot 1 \leq 0)}{x^2 \leq 0 \vdash [x' = 1]x^2 \leq 0}$$



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



Differential Saturation Procedure

refine d$\mathcal{L}$ verification calculus to automatic verification fixedpoint algorithm

$\wr$

```
function prove(ψ ⊢ [D ∧ H]φ):
2: if prove((H → φ)) then
     return true /* property proven */
   for each F ∈Candidates(ψ ⊢ [D ∧ H]φ, H) do
     if prove(ψ ∧ H ⊢ F) and prove((H → F′)) then
       H := H ∧ F /* refine by differential invariant*/
       goto 2;    /* repeat fixedpoint loop */
   end for
   return "not provable using candidates"
```

Return

Outline

Temporal Modalities + Dynamic Modalities

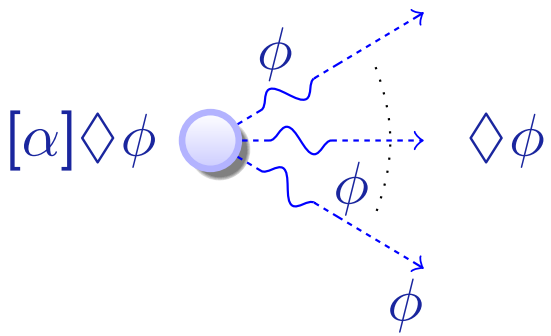
problem	technique	Op	Par	T	closed
$ETCS \models z < MA$	TL-MC	✓	×	✓	×
$\models (\mathrm{Ax}(ETCS) \rightarrow z < MA)$	TL-calculus	×	...	✓	...
$\models [ETCS]\, z < MA$	DL-calculus	✓	✓	×	✓
$\models [ETCS]\Box\, z < MA$	dTL-calculus	✓	✓	✓	✓

Temporal Modalities + Dynamic Modalities

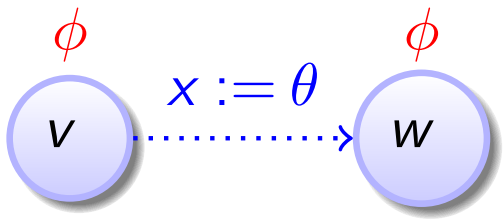
problem	technique	Op	Par	T	closed
$ETCS \models z < MA$	TL-MC	✓	×	✓	×
$\models (\mathrm{Ax}(ETCS) \rightarrow z < MA)$	TL-calculus	×	...	✓	...
$\models [ETCS]\, z < MA$	DL-calculus	✓	✓	×	✓
$\models [ETCS]\Box\, z < MA$	dTL-calculus	✓	✓	✓	✓

differential temporal dynamic logic

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

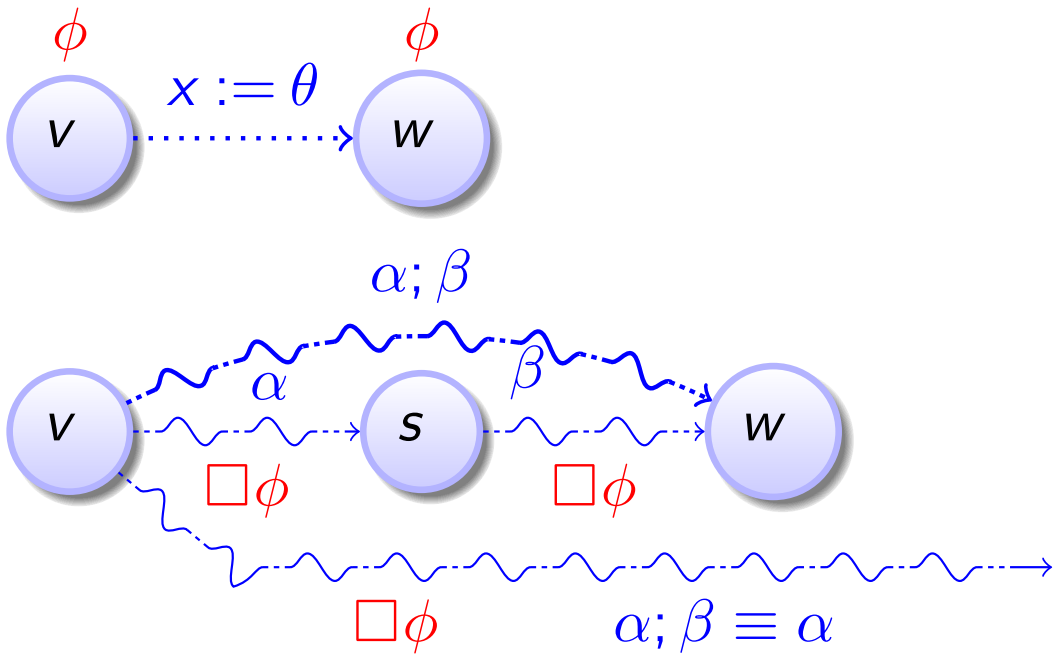


$$\frac{\phi \wedge [x := \theta]\phi}{[x := \theta]\Box\phi}$$

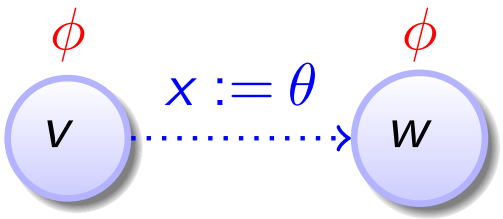


$$\frac{\phi \wedge [x := \theta]\phi}{[x := \theta]\Box\phi}$$

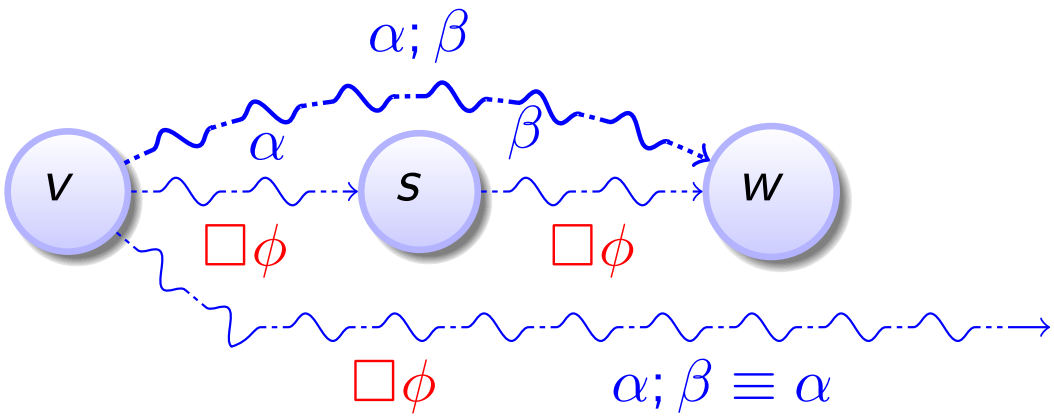
$$\frac{[\alpha]\Box\phi \wedge [\alpha][\beta]\Box\phi}{[\alpha;\beta]\Box\phi}$$



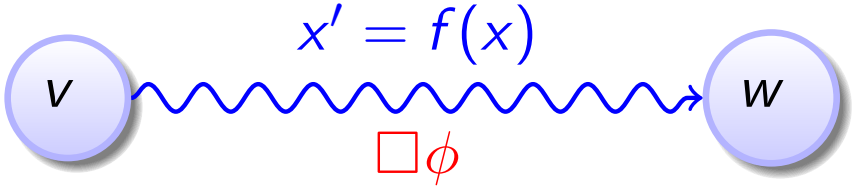
$$\frac{\phi \wedge [x := \theta]\phi}{[x := \theta]\Box\phi}$$



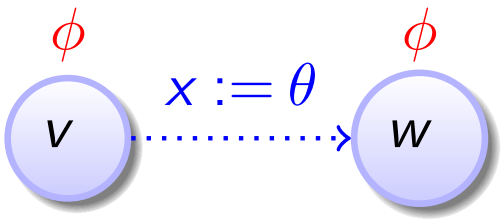
$$\frac{[\alpha]\Box\phi \wedge [\alpha][\beta]\Box\phi}{[\alpha;\beta]\Box\phi}$$



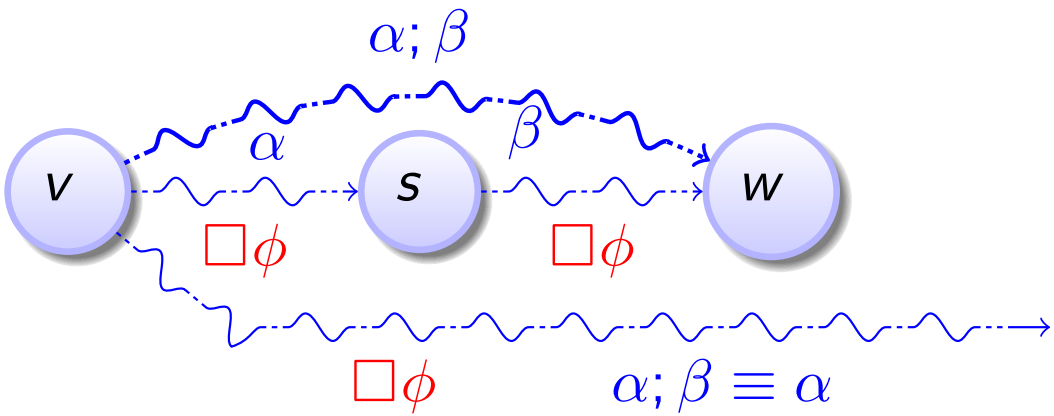
$$\frac{[x' = \theta]\phi}{[x' = \theta]\Box\phi}$$



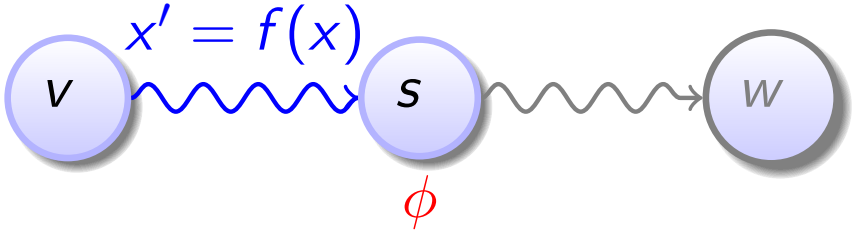
$$\frac{\phi \wedge [x := \theta]\phi}{[x := \theta]\Box\phi}$$



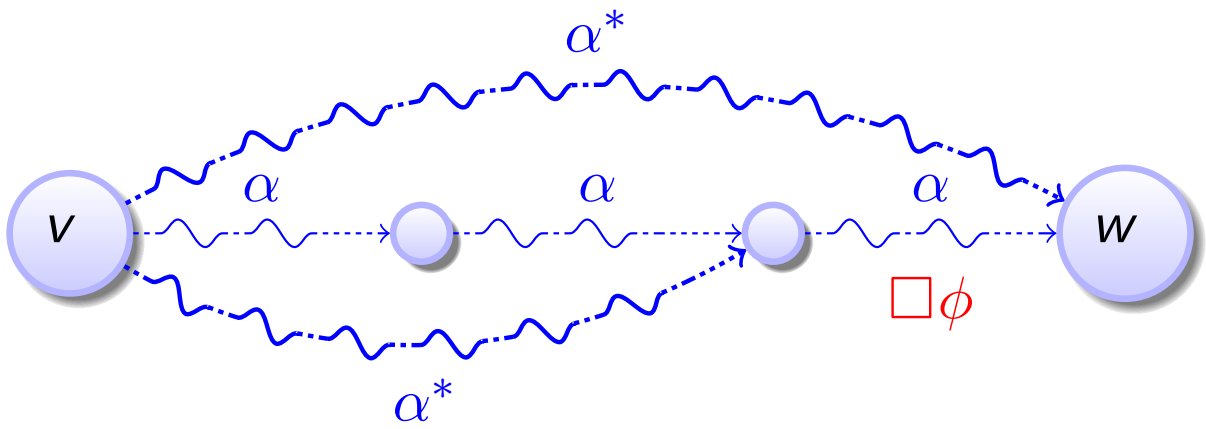
$$\frac{[\alpha]\Box\phi \wedge [\alpha][\beta]\Box\phi}{[\alpha;\beta]\Box\phi}$$



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

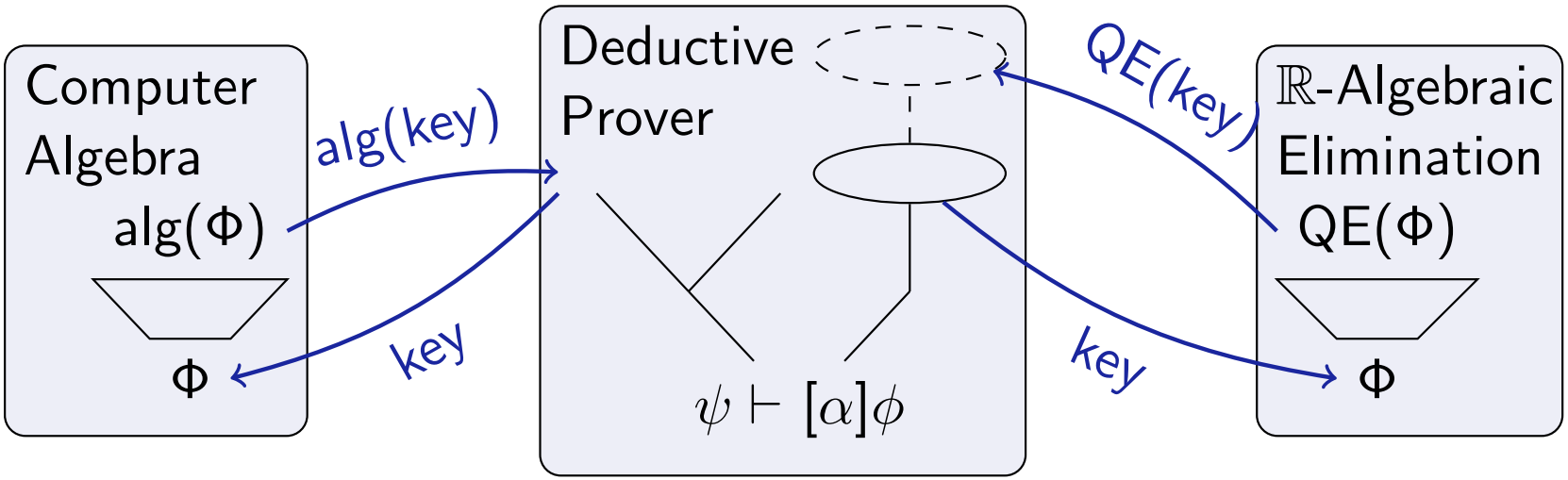


$$\frac{[\alpha^*][\alpha]\Box\phi}{[\alpha^*]\Box\phi}$$

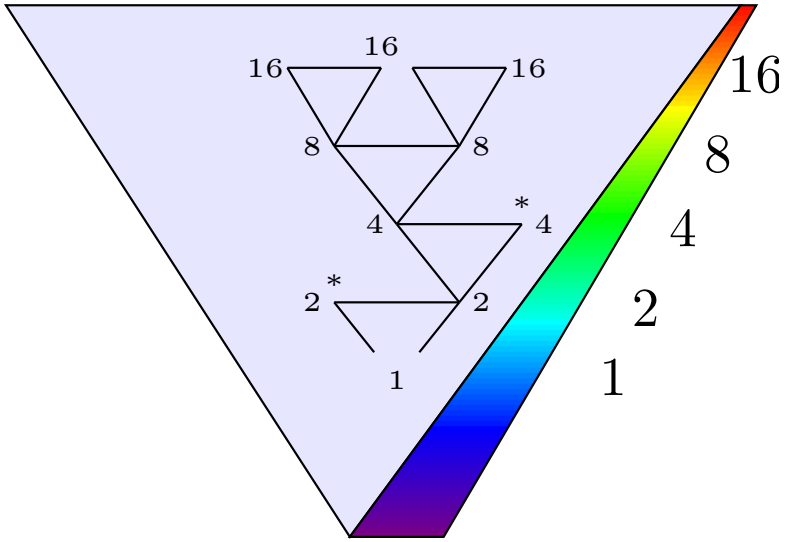


Outline

KeYmaera Verification Architecture

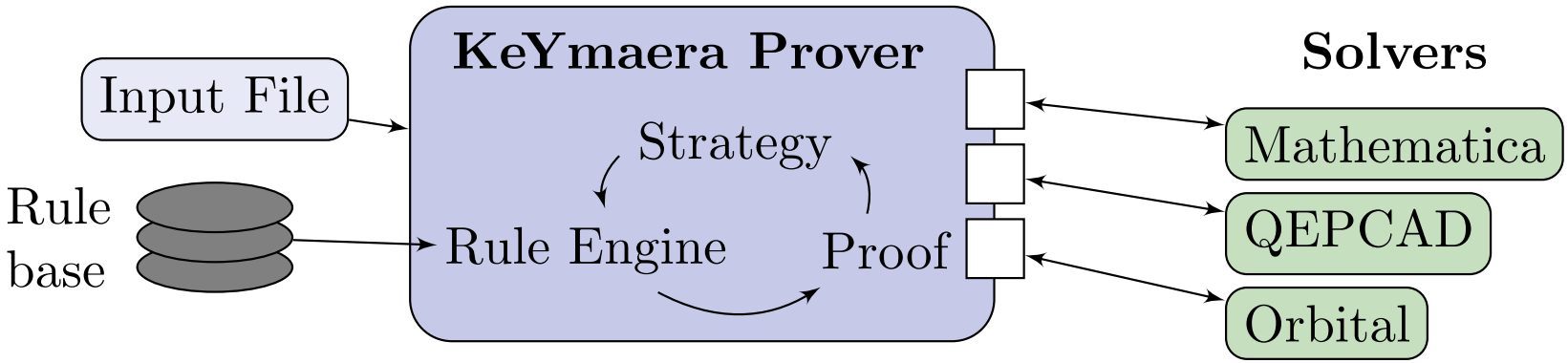


56 interactions?



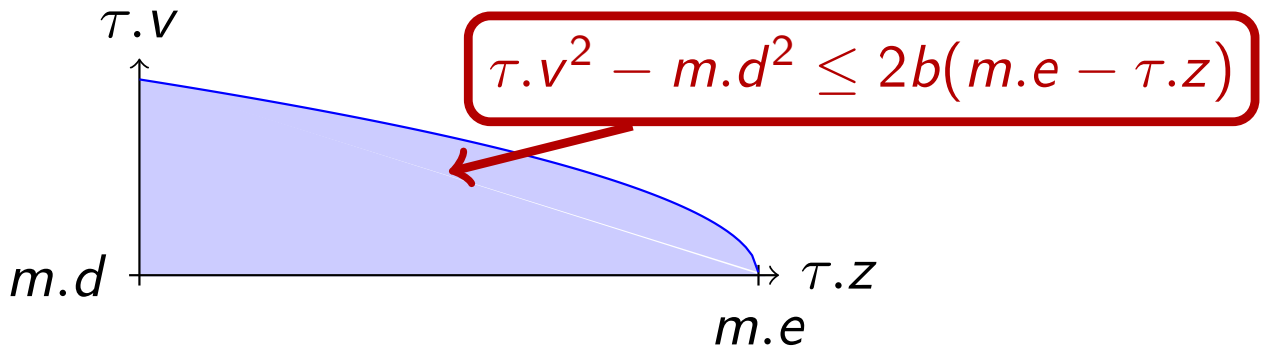
0–1 interactions!

KeYmaera Prover Architecture



Outline

ETCS Controllability

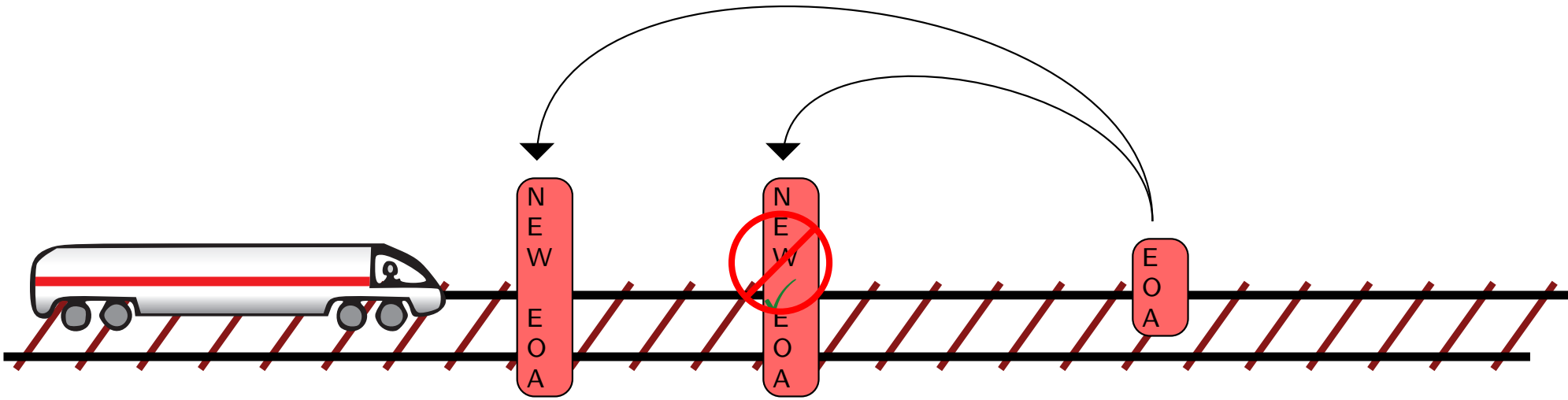


Proposition (Controllability)

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

$$\equiv \tau.v^2 - m.d^2 \leq 2b(m.e - \tau.z)$$

ETCS RBC Controllability



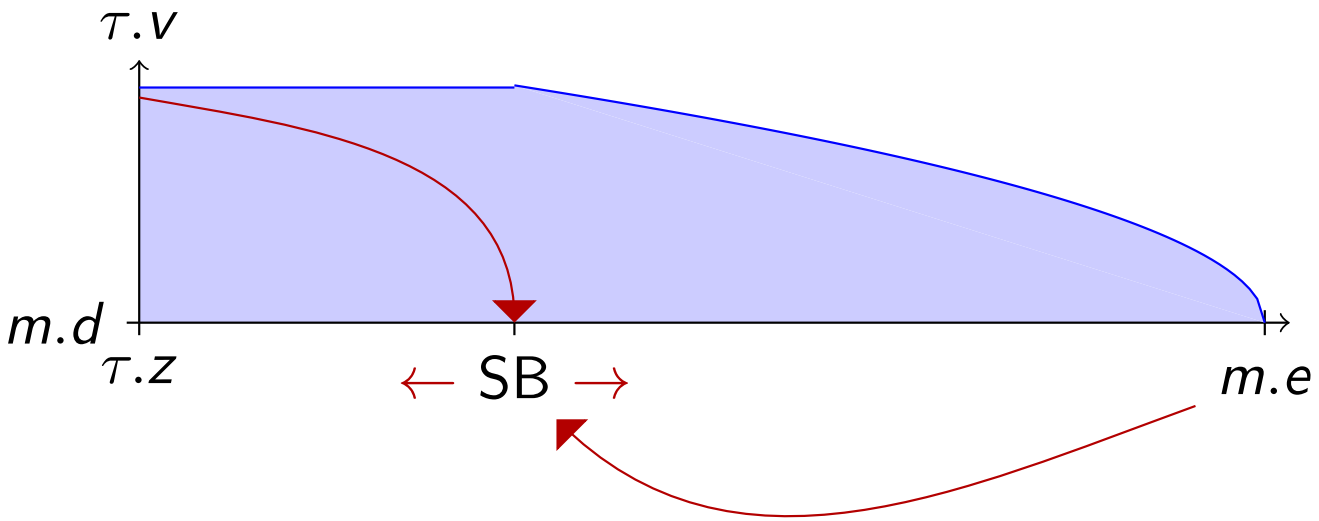
Proposition (RBC Controllability)

$$m.d \geq 0 \wedge b > 0 \rightarrow [m_0 := m;\ RBC] \Big($$

$$m_0.d^2 - m.d^2 \leq 2b(m.e - m_0.e) \wedge m_0.d \geq 0 \wedge m.d \geq 0 \leftrightarrow$$

$$\forall \tau \left((\langle m := m_0 \rangle \tau.v^2 - m.d^2 \leq 2b(m.e - \tau.z)) \rightarrow \tau.v^2 - m.d^2 \leq 2b(m.e - \tau.\right.$$

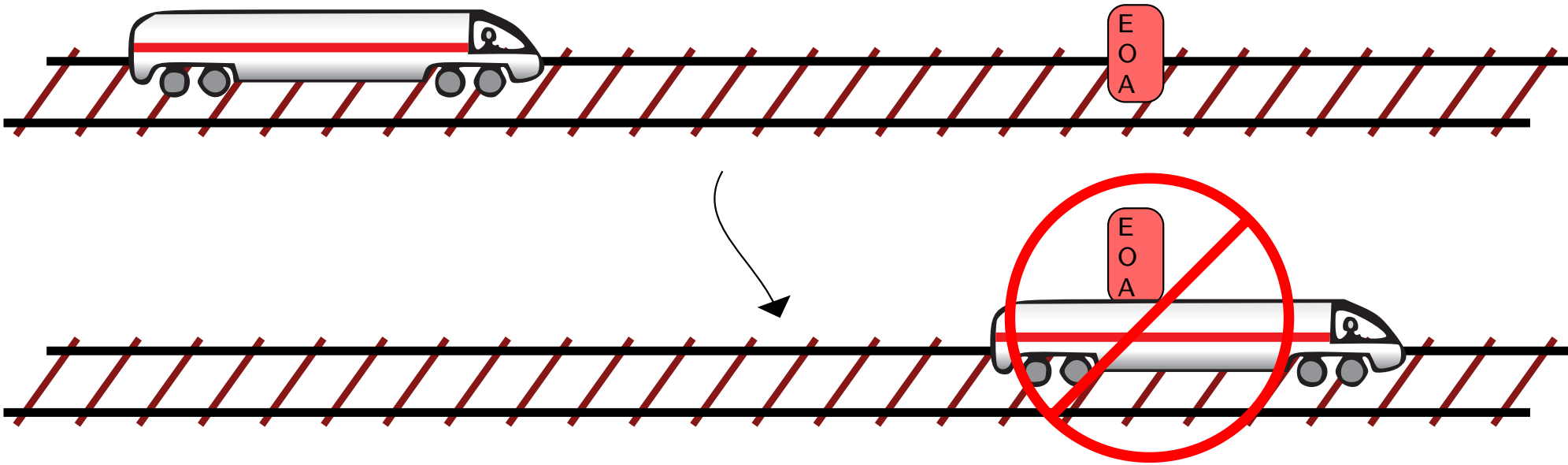
ETCS Reactivity



Proposition (Reactivity)

$$\Big(\forall m.e\, \forall \tau.z \left(m.e - \tau.z \geq SB \wedge \tau.v^2 - m.d^2 \leq 2b(m.e - \tau.z) \rightarrow \right.$$
$$\left. [\tau.a := A;\ drive]\, \tau.v^2 - m.d^2 \leq 2b(m.e - \tau.z)\right)\Big)$$
$$\equiv SB \geq \frac{\tau.v^2 - m.d^2}{2b} + \left(\frac{A}{b} + 1\right)\left(\frac{A}{2}\varepsilon^2 + \varepsilon\, \tau.v\right)$$

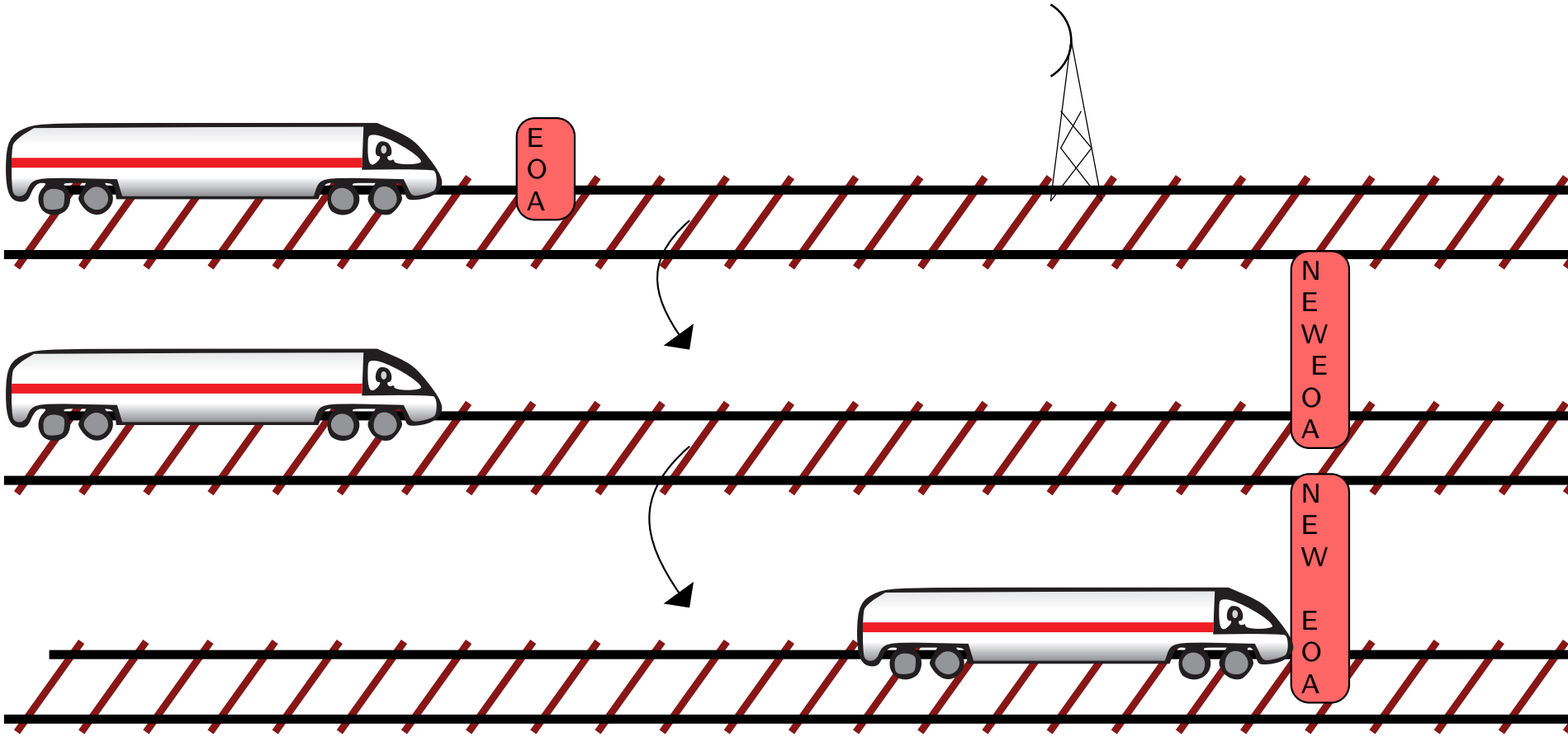
ETCS Safety



Proposition (Safety)

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

ETCS Liveness



Proposition (Liveness)

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

Full European Train Control System (ETCS)

provable automatically!

spec : $\tau.v^2 - \mathbf{m}.d^2 \leq 2b(\mathbf{m}.e - \tau.p) \wedge \tau.v \geq 0 \wedge \mathbf{m}.d \geq 0 \wedge b > 0$
$\rightarrow$ [ETCS]$(\tau.p \geq \mathbf{m}.e \rightarrow \tau.v \leq \mathbf{m}.d)$

ETCS: $(\text{train} \cup \text{rbc})^*$

train : spd; atp; move

spd : $(?\tau.v \leq \mathbf{m}.r;\ \tau.a := *;\ ? - b \leq \tau.a \leq A)$
$\cup (?\tau.v \geq \mathbf{m}.r;\ \tau.a := *;\ ?0 > \tau.a \geq -b)$

atp : $SB := \frac{\tau.v^2 - \mathbf{m}.d^2}{2b} + \left(\frac{A}{b} + 1\right)\left(\frac{A}{2}\varepsilon^2 + \varepsilon\,\tau.v\right);$
$(?(\mathbf{m}.e - \tau.p \leq SB \vee rbc.message = emergency);\ \tau.a := -b)$
$\cup (?\mathbf{m}.e - \tau.p \geq SB \wedge rbc.message \neq emergency)$

move : $t := 0;\ (\tau.p' = \tau.v, \tau.v' = \tau.a, t' = 1 \wedge \tau.v \geq 0 \wedge t \leq \varepsilon)$

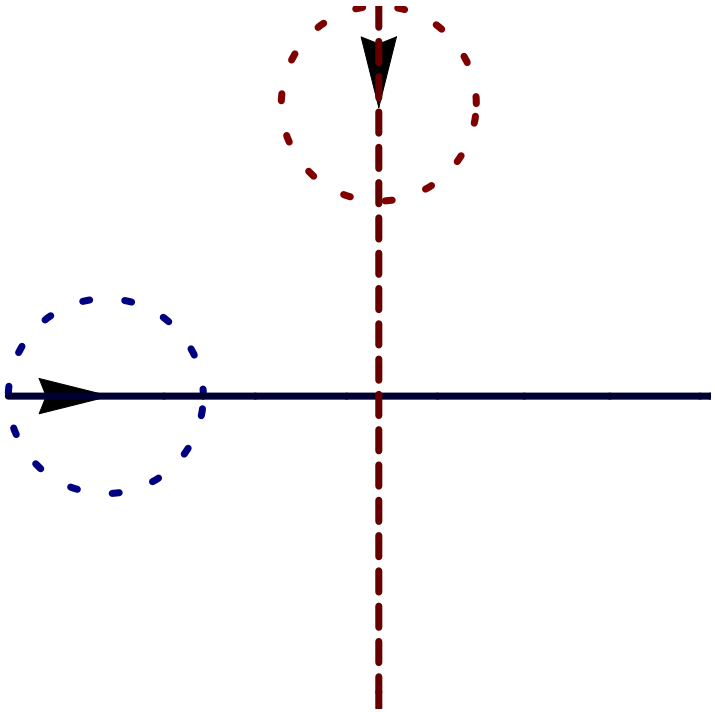
rbc : $(rbc.message := emergency)$
$\cup\ (\mathbf{m}_0 := \mathbf{m}; \mathbf{m} := *;$
$?\mathbf{m}.r \geq 0 \wedge \mathbf{m}.d \geq 0 \wedge \mathbf{m}_0.d^2 - \mathbf{m}.d^2 \leq 2b(\mathbf{m}.e - \mathbf{m}_0.e))$

Full European Train Control System (ETCS)

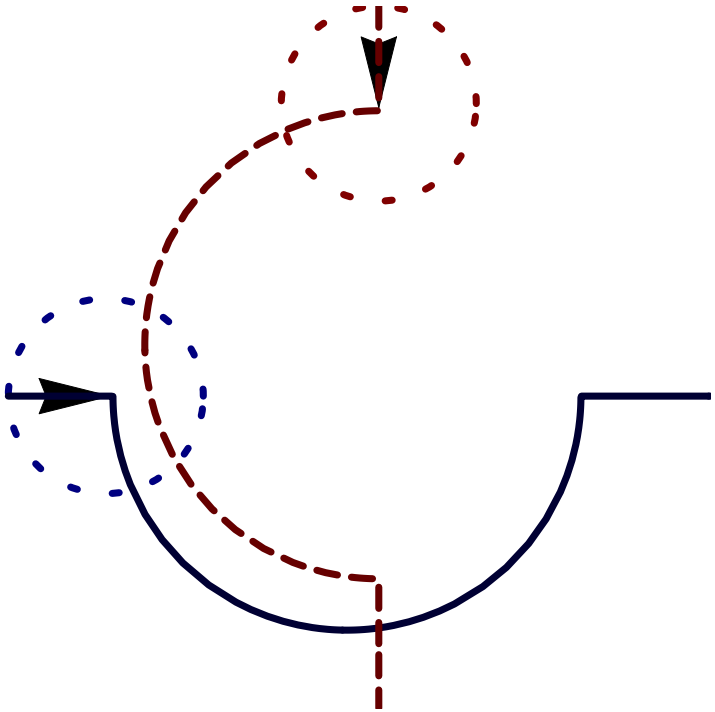
```
 state = 0,
 2 * b * (m - z) >= v ^ 2 - d ^ 2,
 v >= 0, d >= 0, v >= 0, ep >  0, b >  0, amax >  0, d >= 0
==>
    v <= vdes
 -> \forall R a_3;
      (   a_3 >= 0 & a_3 <= amax
       ->   (      m - z
              <= (amax / b + 1) * ep * v
               + (v ^ 2 - d ^ 2) / (2 * b)
          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

Outline

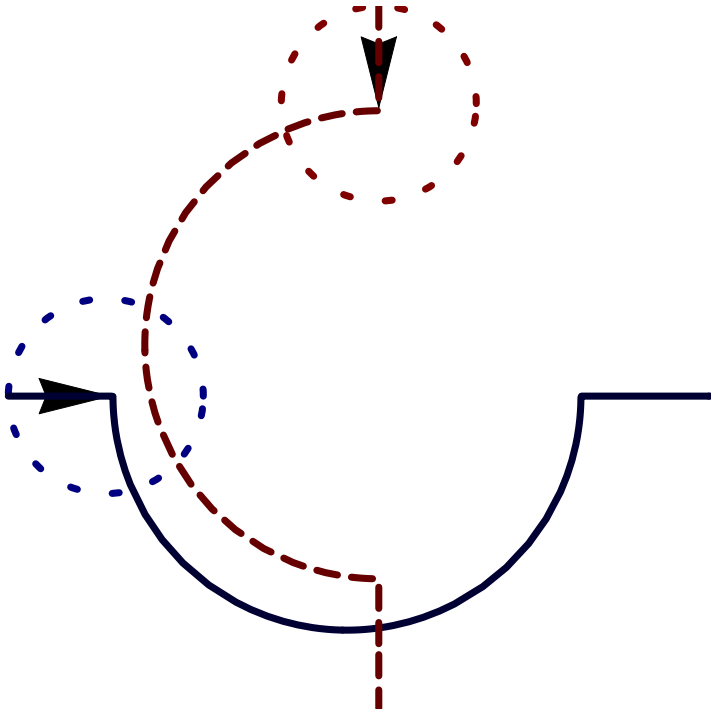
Air Traffic Control



Air Traffic Control



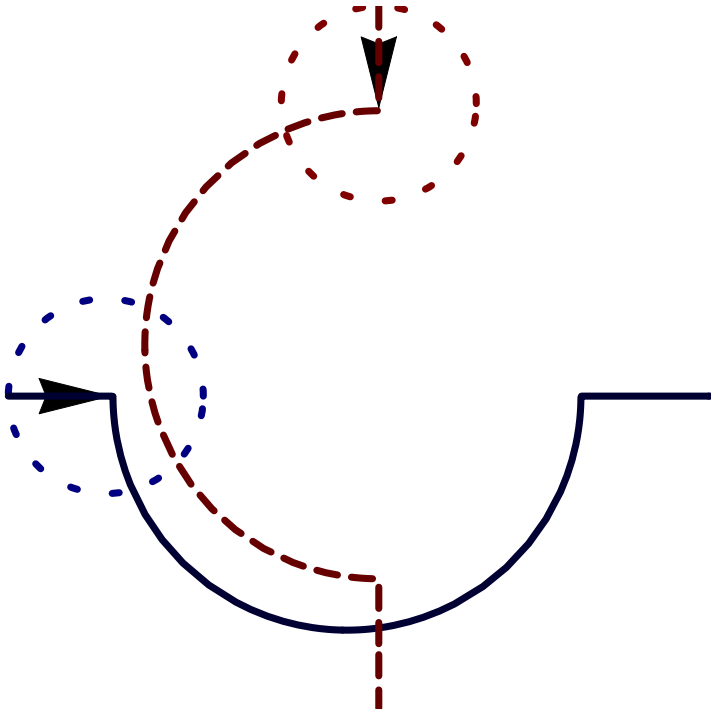
Air Traffic Control

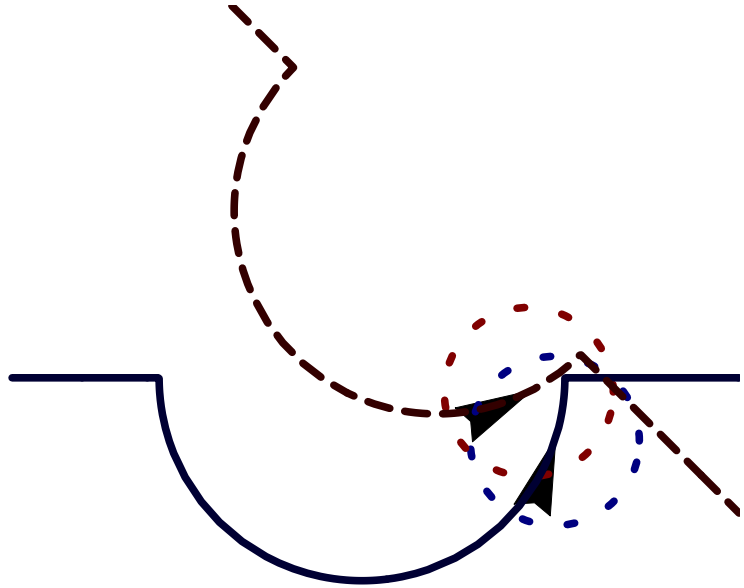


Verification?

looks correct

Air Traffic Control

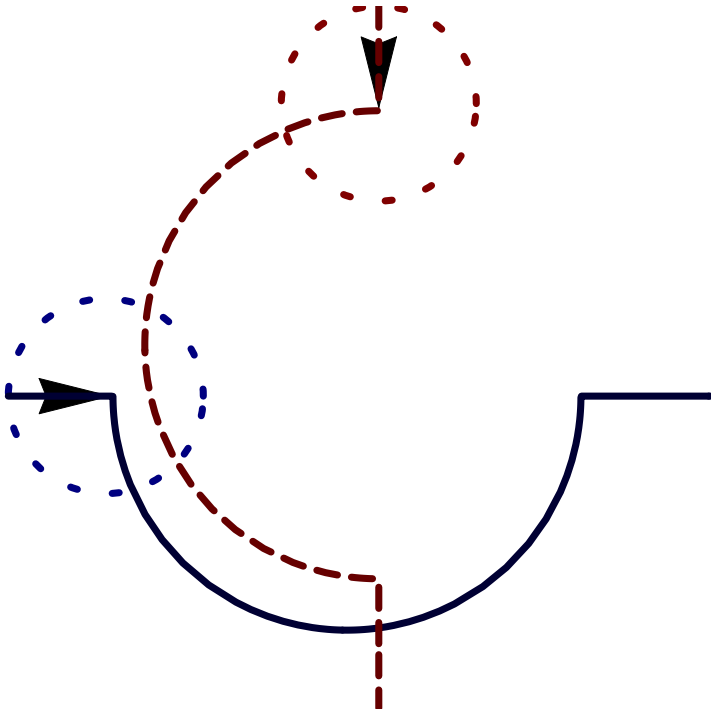


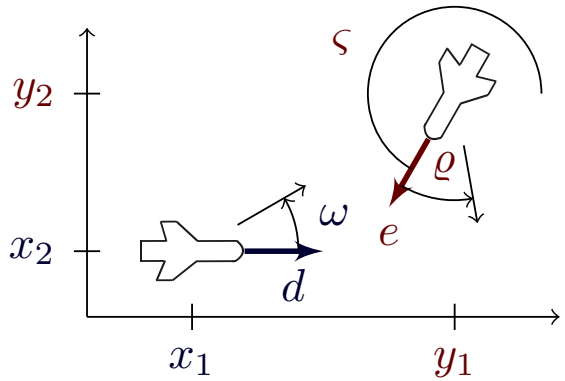


Verification?

looks correct NO!

Air Traffic Control



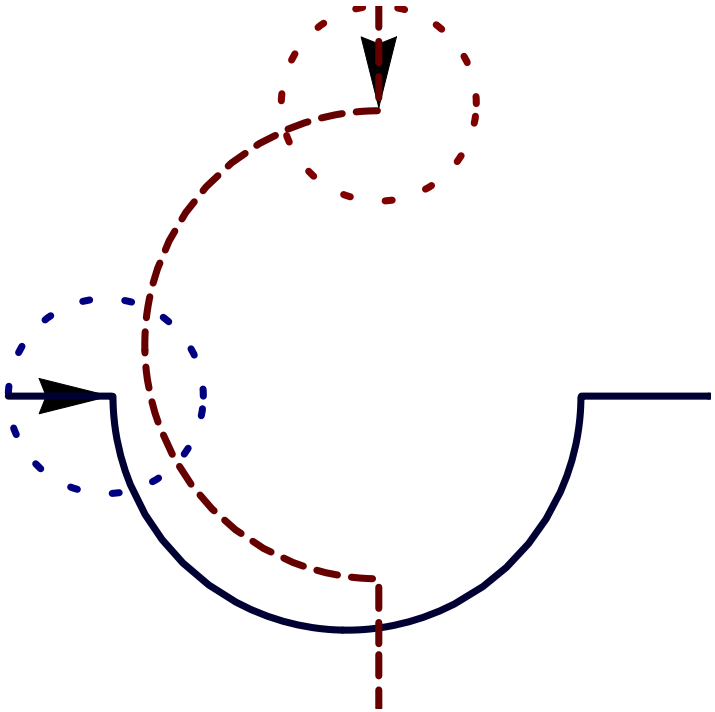


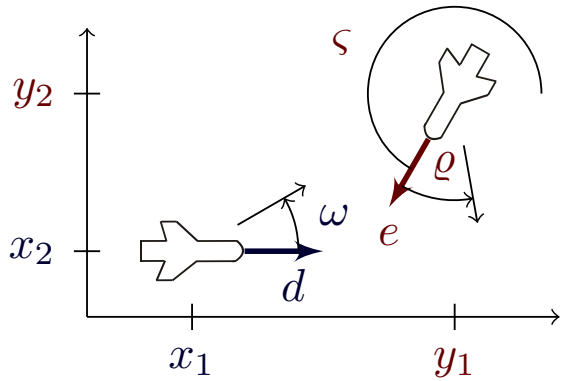
$$\begin{bmatrix} x_1' = -v_1 + v_2 \cos\vartheta + \omega x_2 \\ x_2' = v_2 \sin\vartheta - \omega x_1 \\ \vartheta' = \varpi - \omega \end{bmatrix}$$

Verification?

looks correct NO!

Air Traffic Control



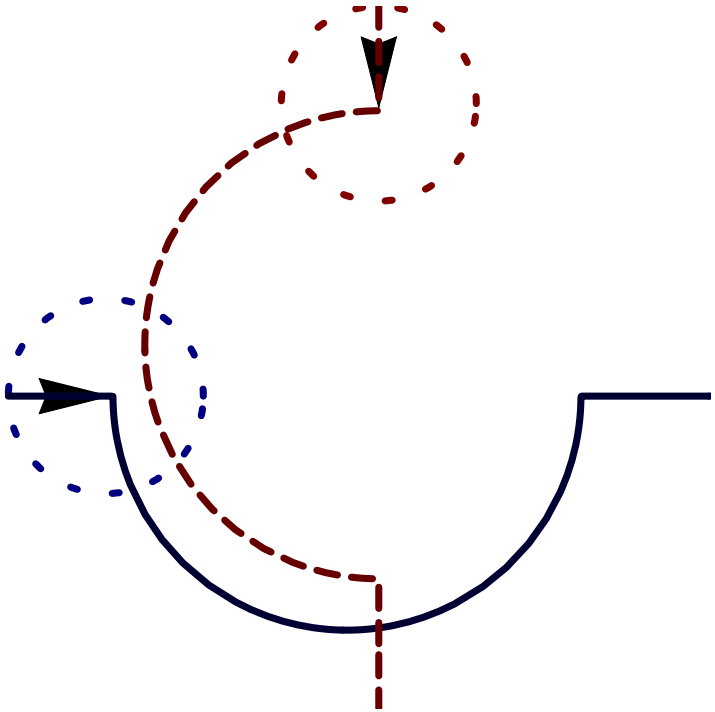


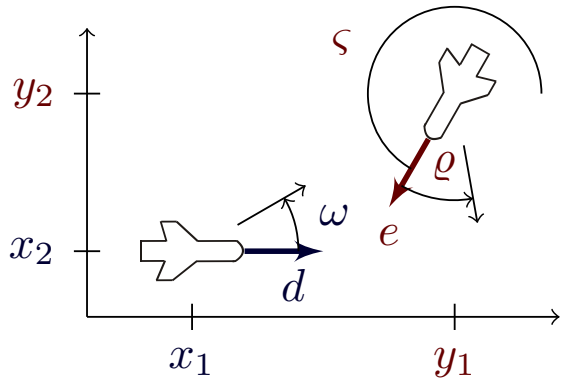
$$\begin{bmatrix} x_1' = -v_1 + v_2 \cos\vartheta + \omega x_2 \\ x_2' = v_2 \sin\vartheta - \omega x_1 \\ \vartheta' = \varpi - \omega \end{bmatrix}$$

Example ("Solving" differential equations)

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

Air Traffic Control



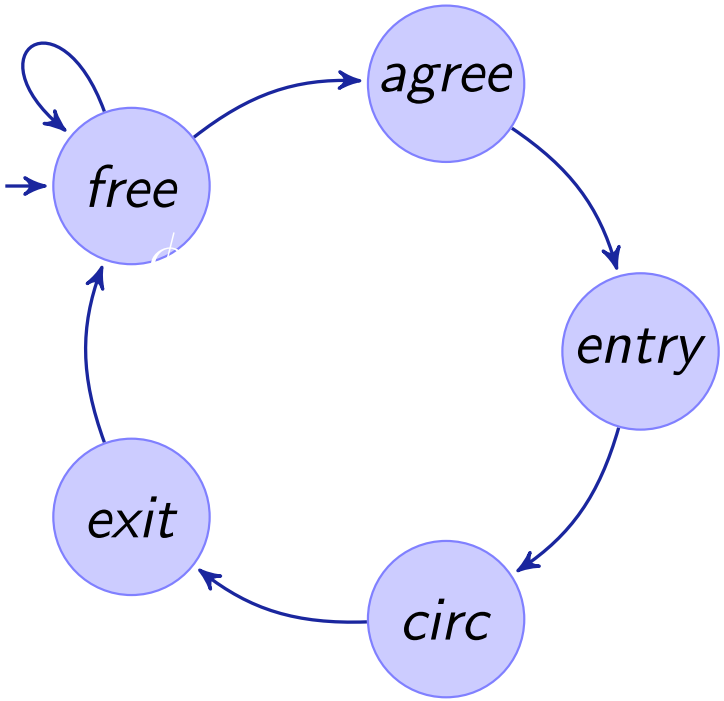


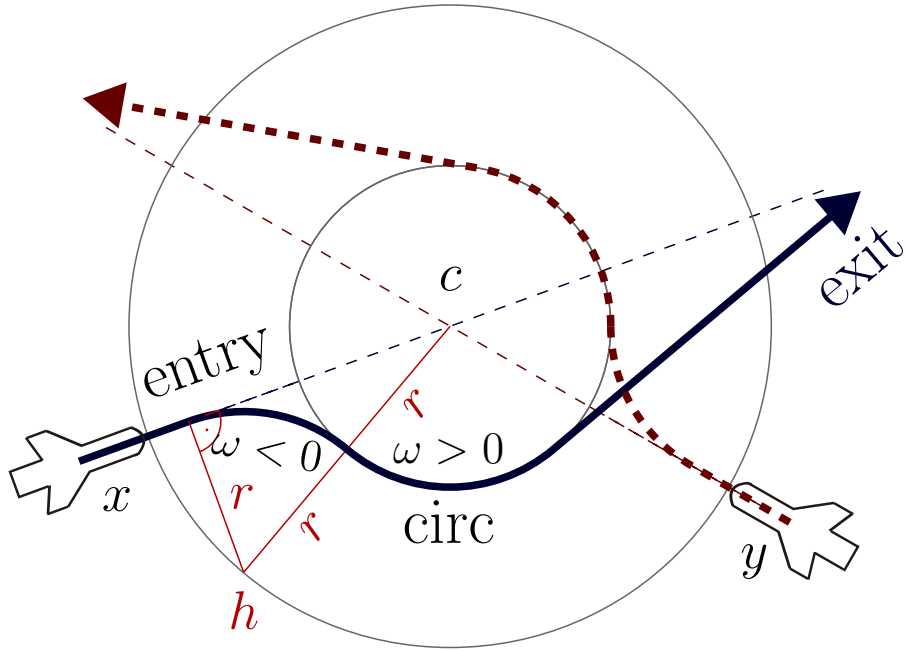
$$\begin{bmatrix} x_1' = -v_1 + v_2 \cos\vartheta + \omega x_2 \\ x_2' = v_2 \sin\vartheta - \omega x_1 \\ \vartheta' = \varpi - \omega \end{bmatrix}$$

Example ("Solving" differential equations)

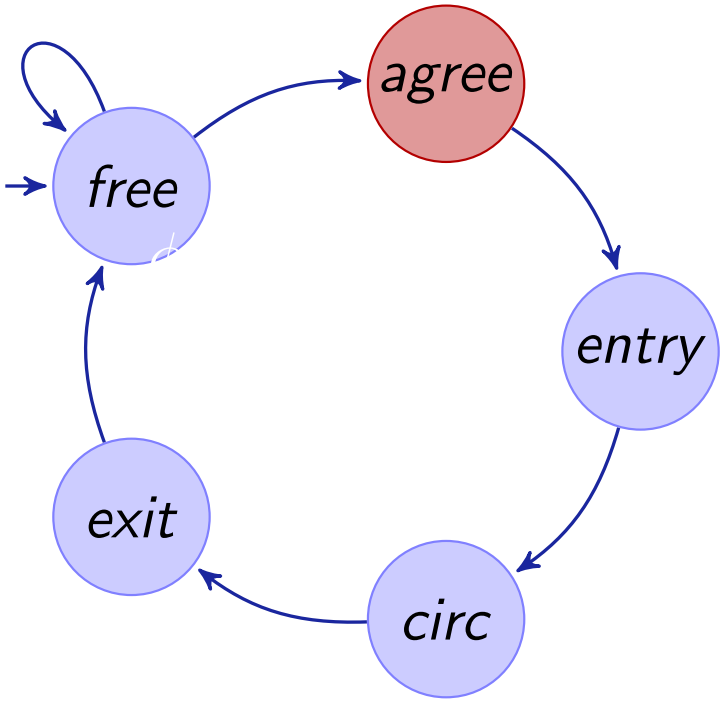
$$\forall t \geq 0 \quad \frac{1}{\omega\varpi}\big(x_1\omega\varpi\cos t\omega - v_2\omega\cos t\omega\sin\vartheta + v_2\omega\cos t\omega\cos t\varpi\sin\vartheta - v_1\varpi\sin t\omega$$
$$+ x_2\omega\varpi\sin t\omega - v_2\omega\cos\vartheta\cos t\varpi\sin t\omega - v_2\omega\sqrt{1-\sin\vartheta^2}\sin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

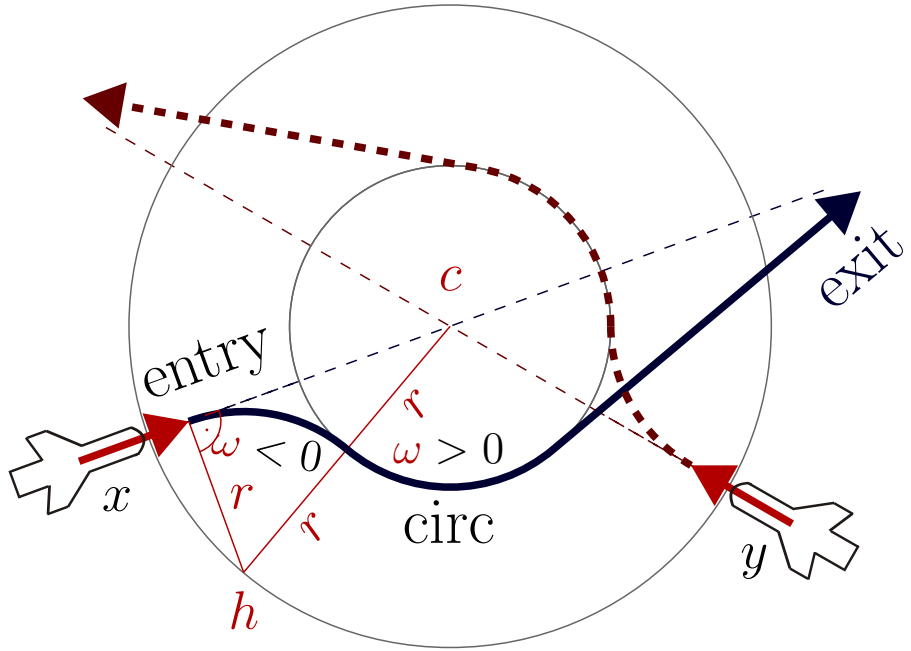
Flyable Roundabout Maneuver: Overview



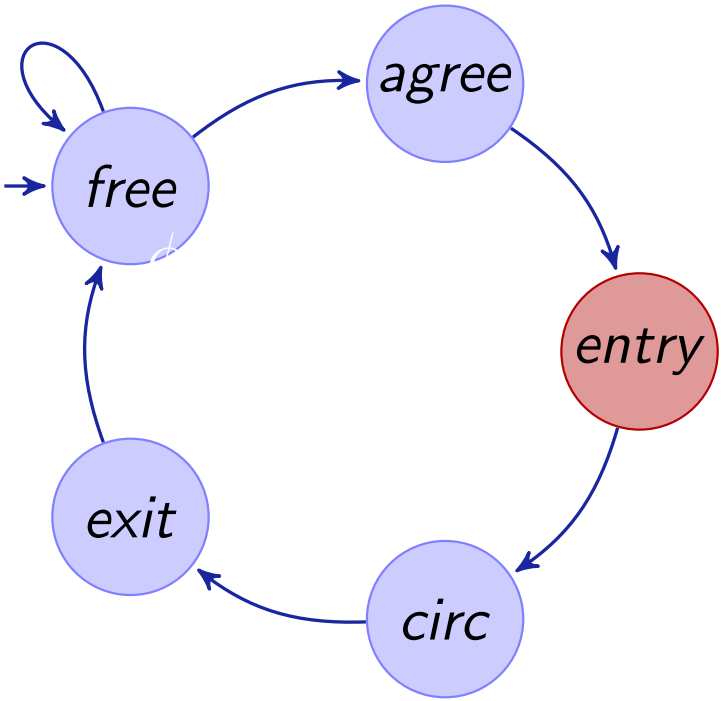


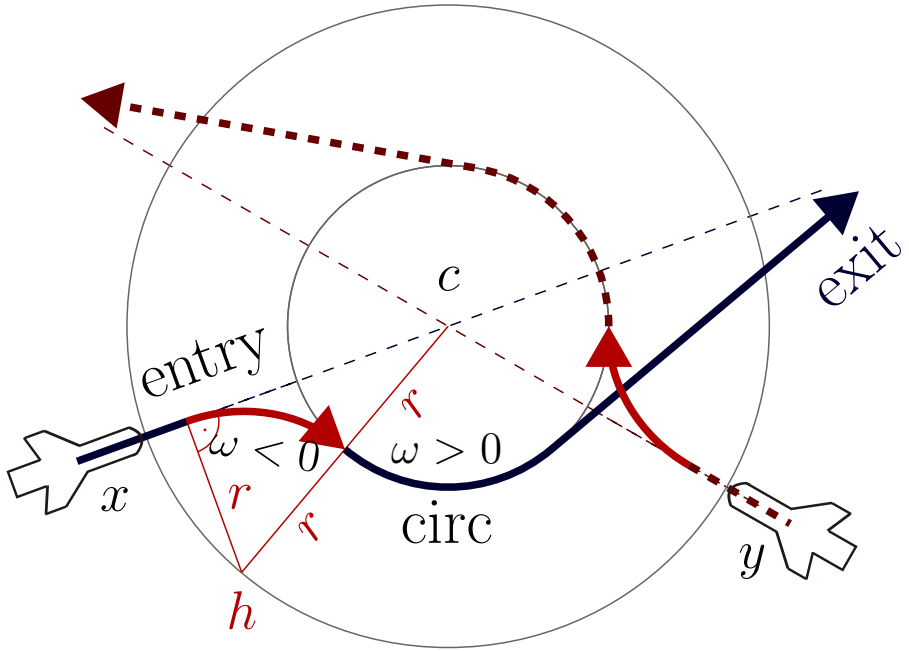
Flyable Roundabout Maneuver: Overview



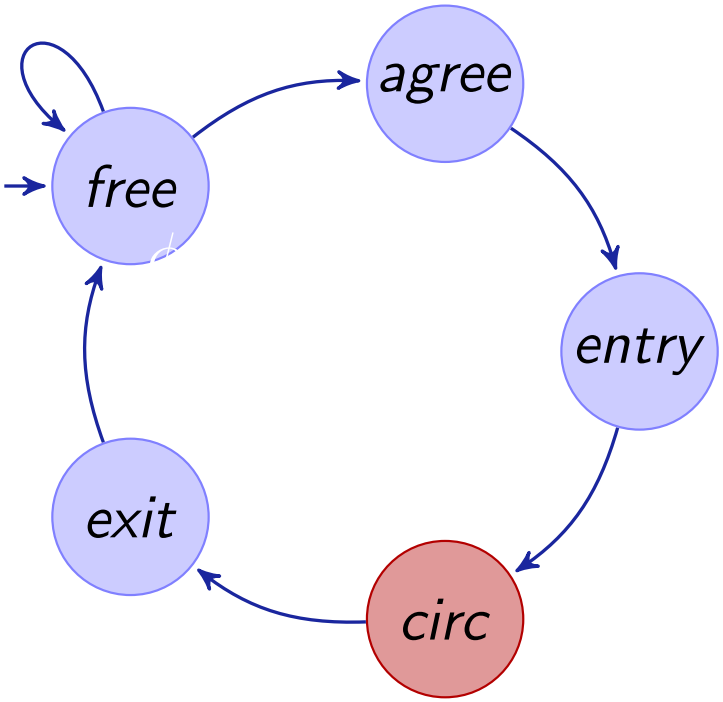


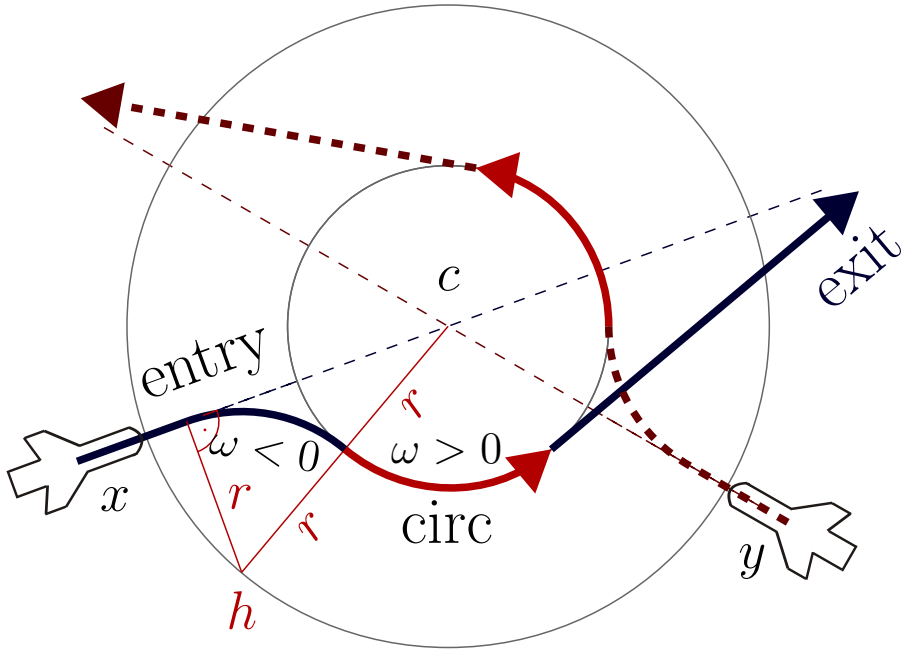
Flyable Roundabout Maneuver: Overview



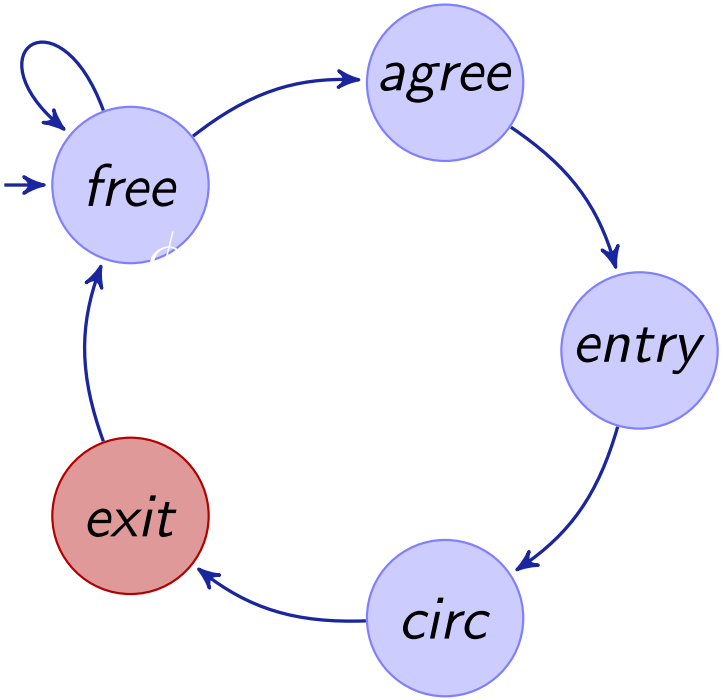


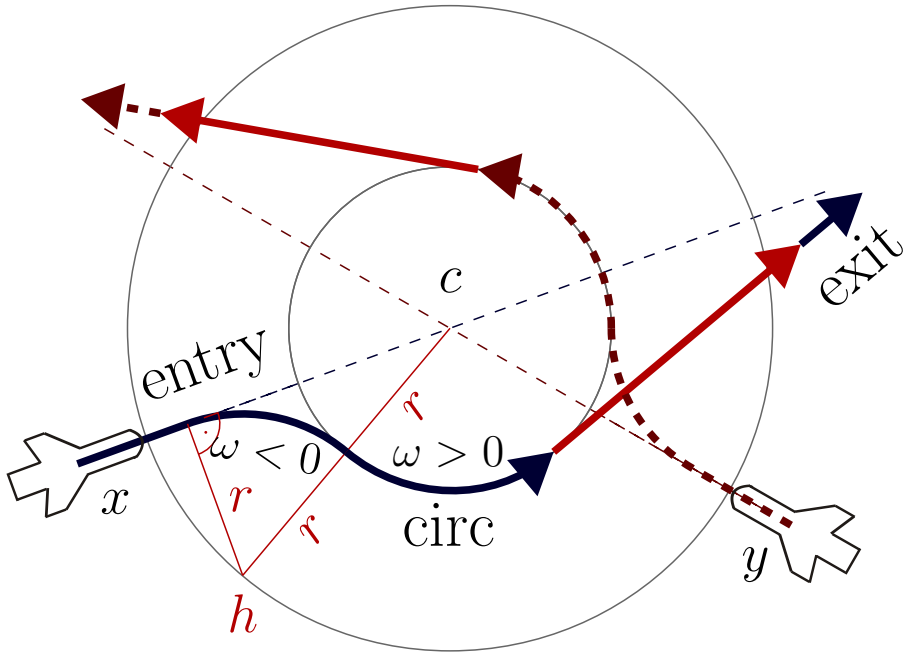
Flyable Roundabout Maneuver: Overview



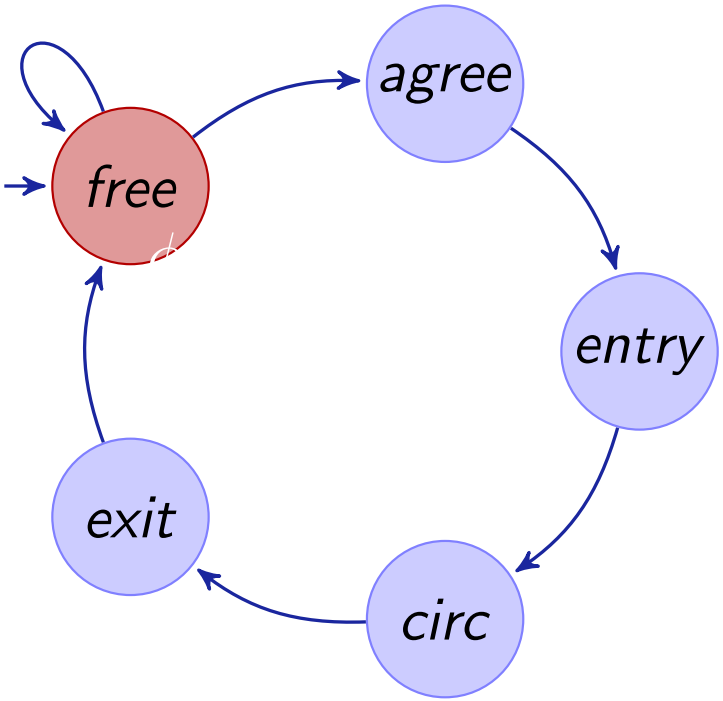


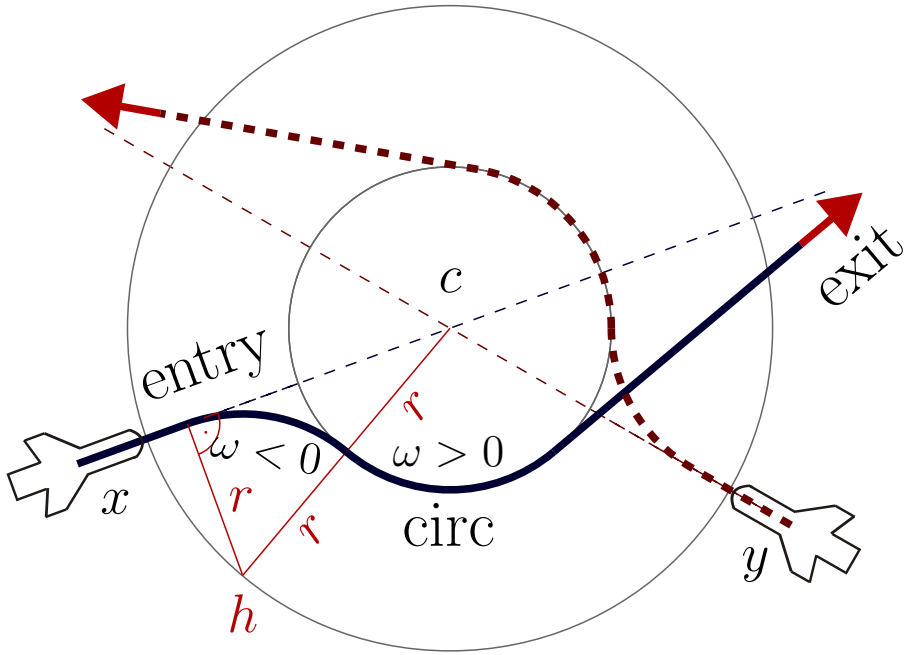
Flyable Roundabout Maneuver: Overview

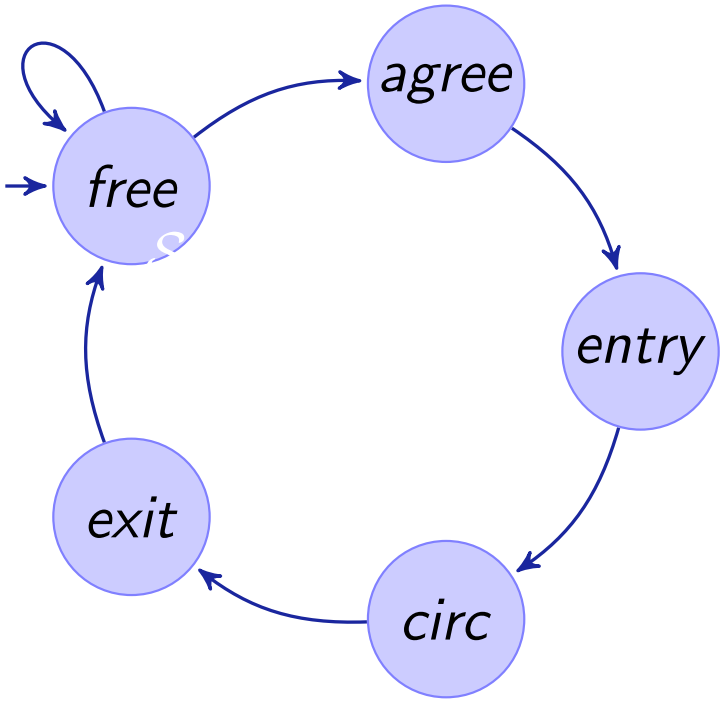


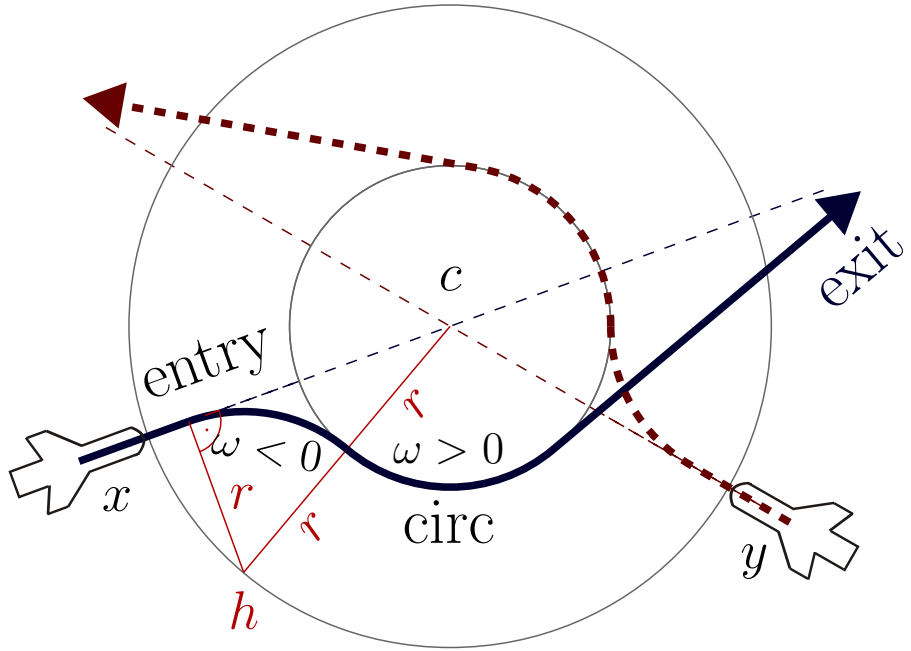


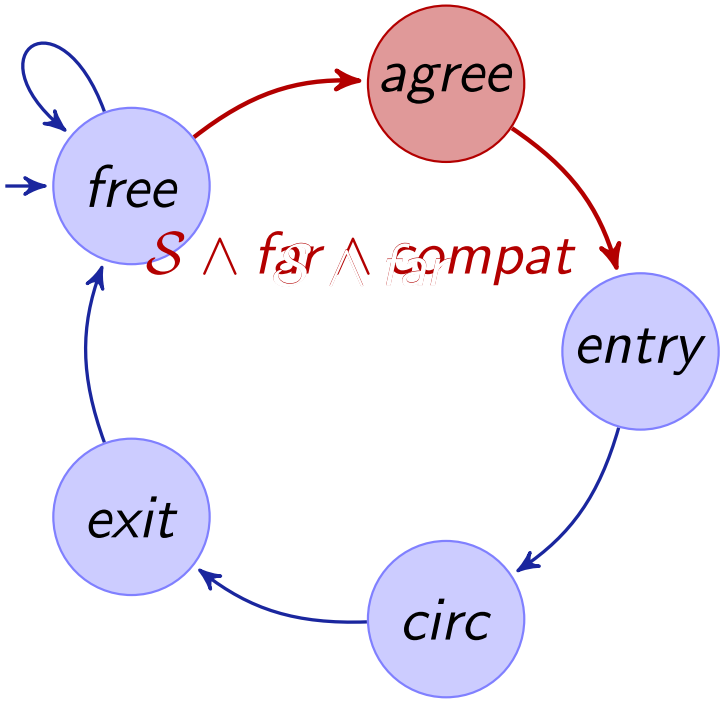
Flyable Roundabout Maneuver: Overview

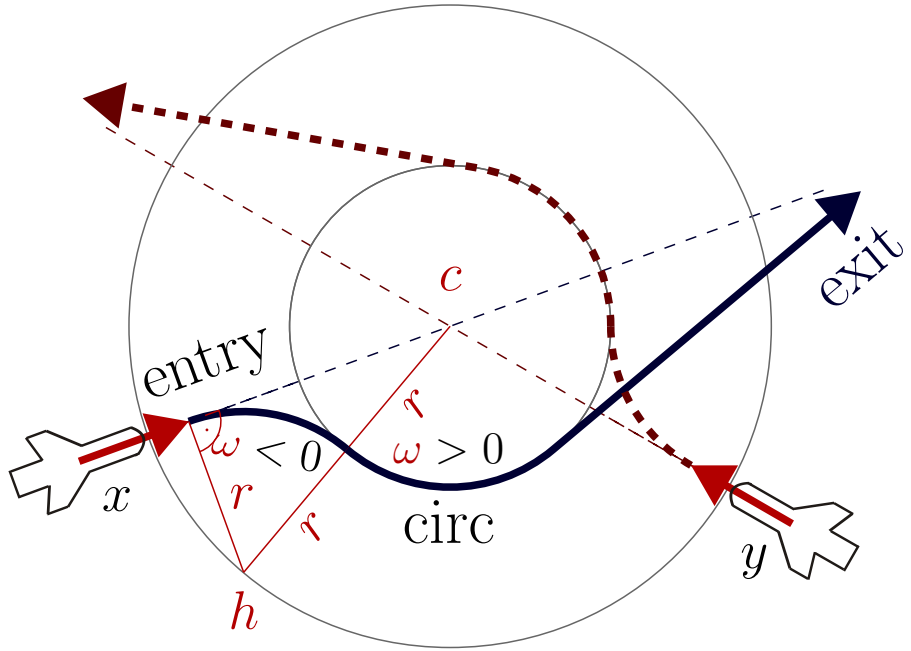





free
agree
entry
circ
exit

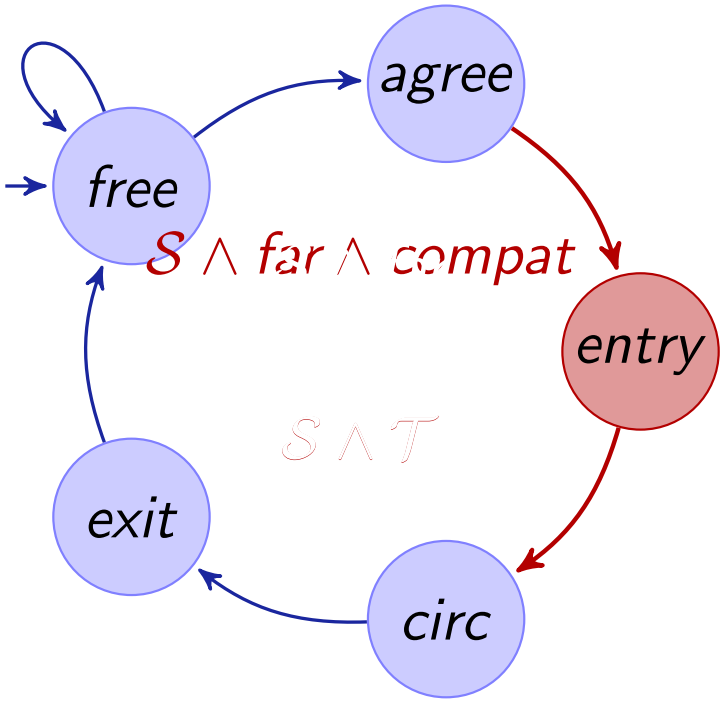

c
entry
exit
x
y
h
r
r
r
ω < 0
ω > 0
circ

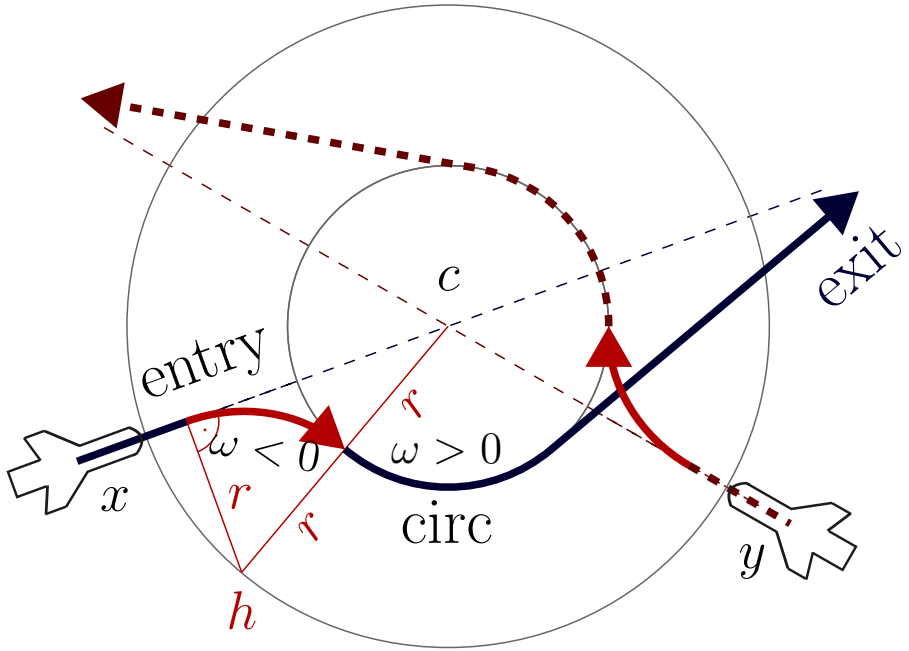




Example (d$\mathcal{L}$ formula of verification subgoal)

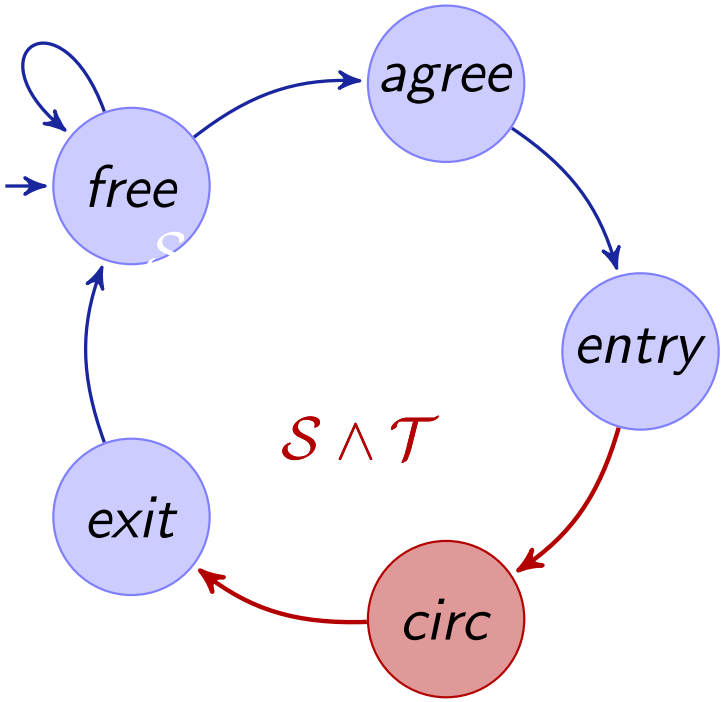
$$safe \wedge far \rightarrow [agree](safe \wedge far \wedge compatible)$$

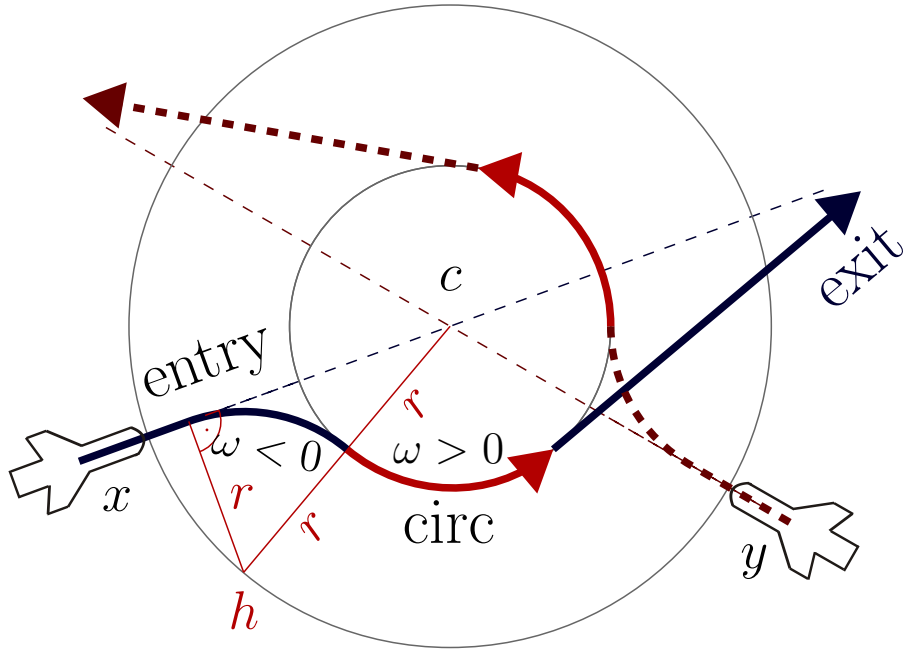




Example (d$\mathcal{L}$ formula of verification subgoal)

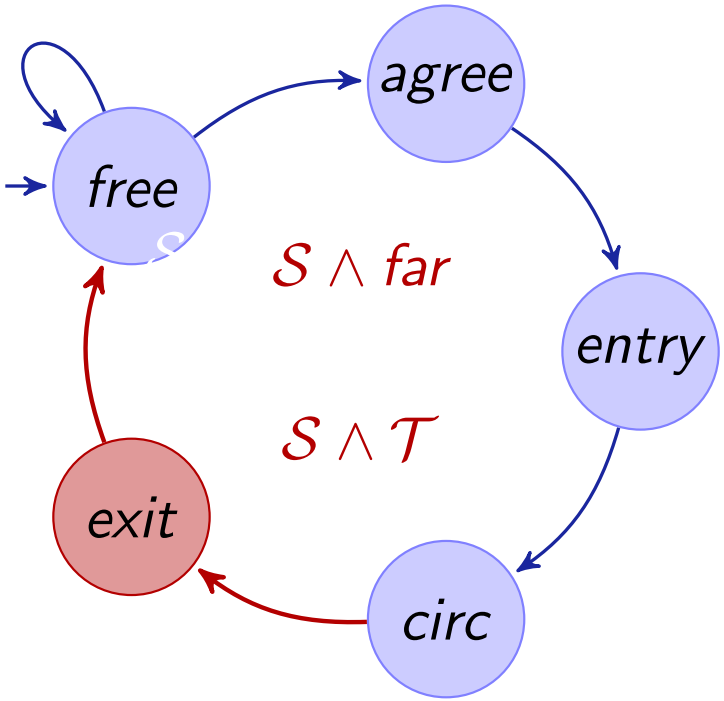
$$safe \wedge far \wedge compatible \rightarrow [entry](safe \wedge tangential)$$

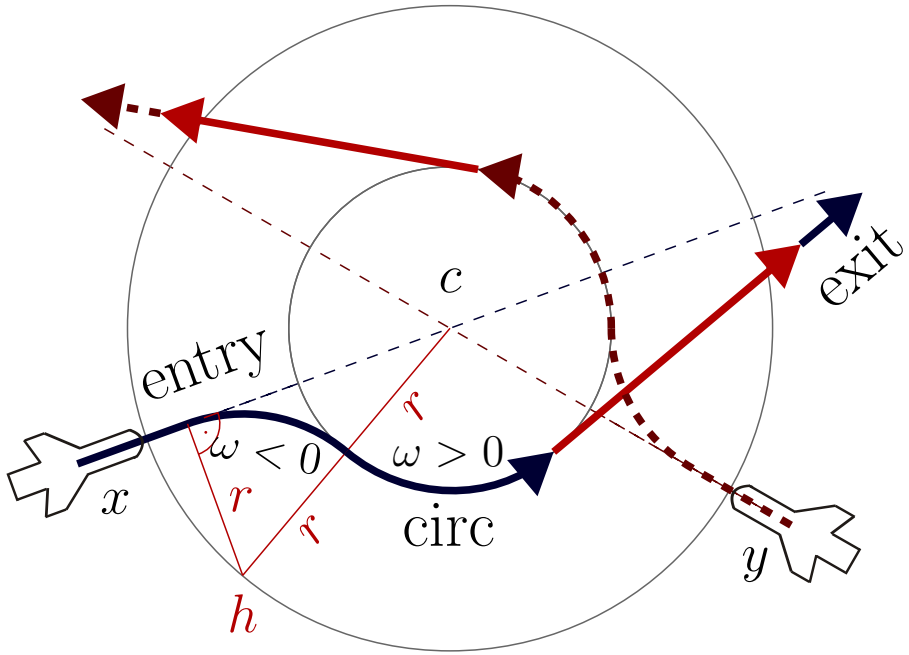




Example (d$\mathcal{L}$ formula of verification subgoal)

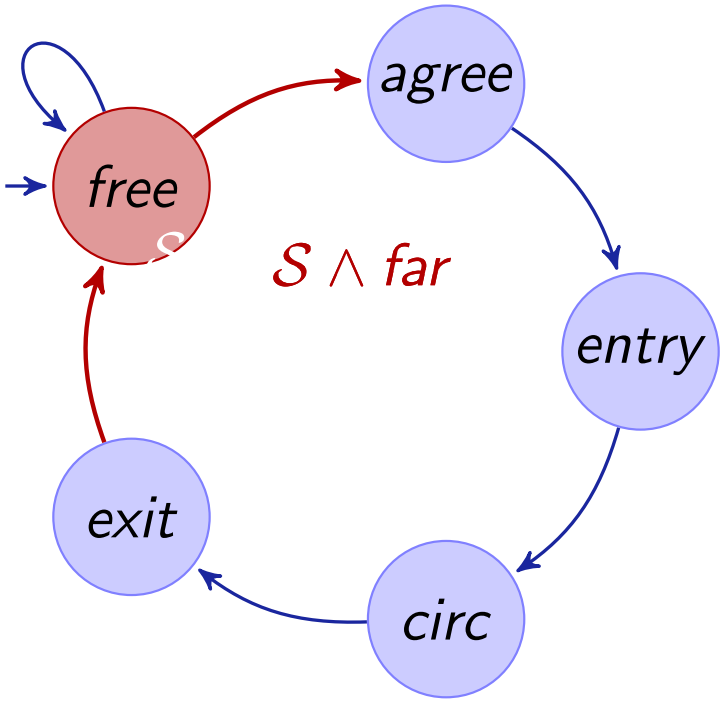
$$safe \wedge tangential \rightarrow [circ](safe \wedge tangential)$$

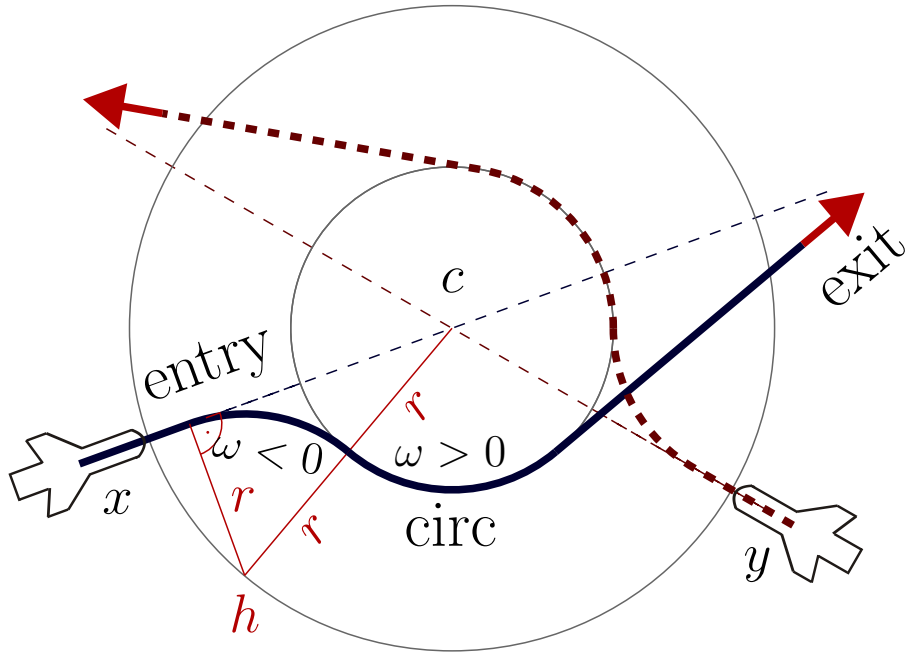




Example (d$\mathcal{L}$ formula of verification subgoal)

$$safe \wedge tangential \rightarrow [exit](safe \wedge far)$$

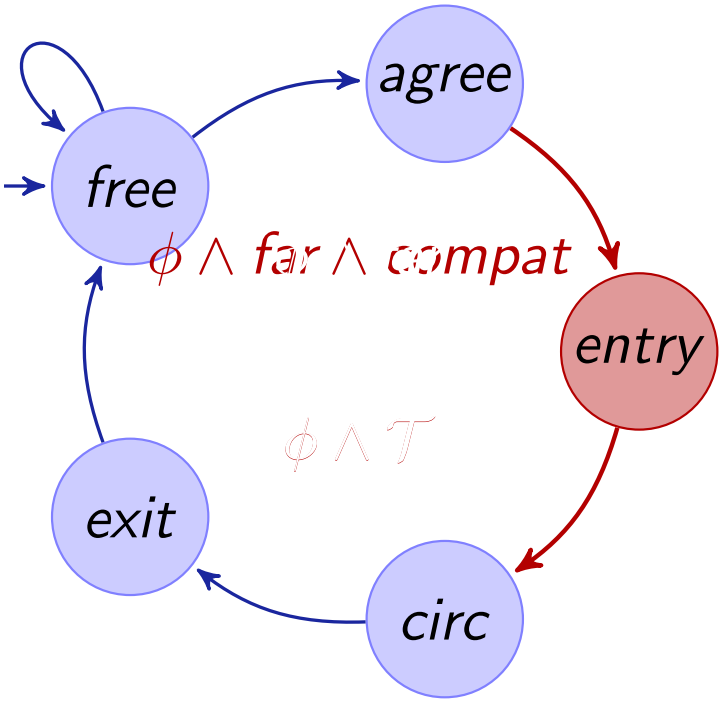


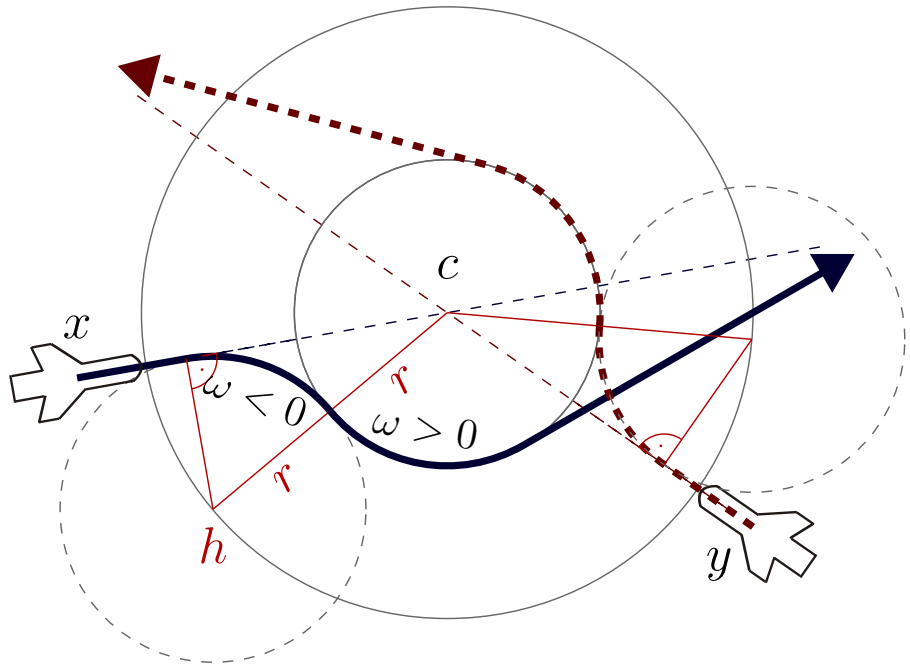


Example (d$\mathcal{L}$ formula of verification subgoal)

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

Flyable Roundabout Maneuver: Entry

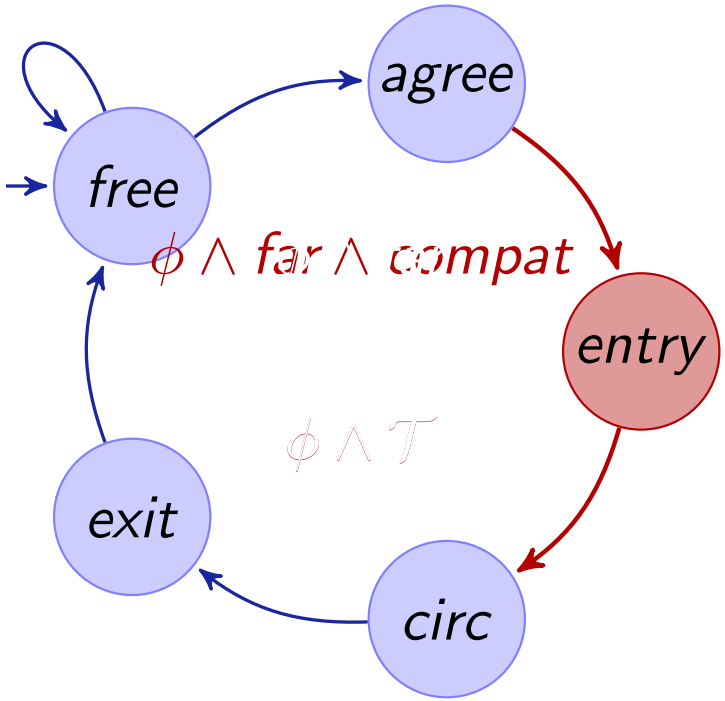


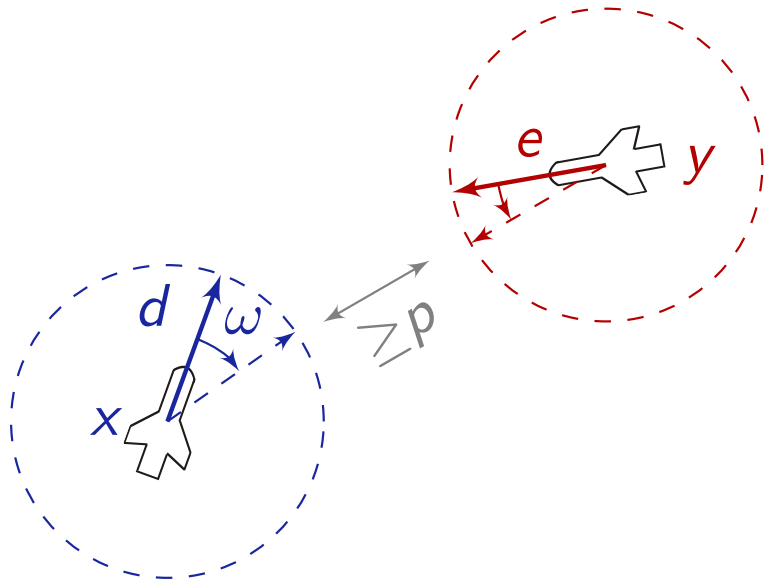


Example (d$\mathcal{L}$ formula of verification subgoal)

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

Flyable Roundabout Maneuver: Entry

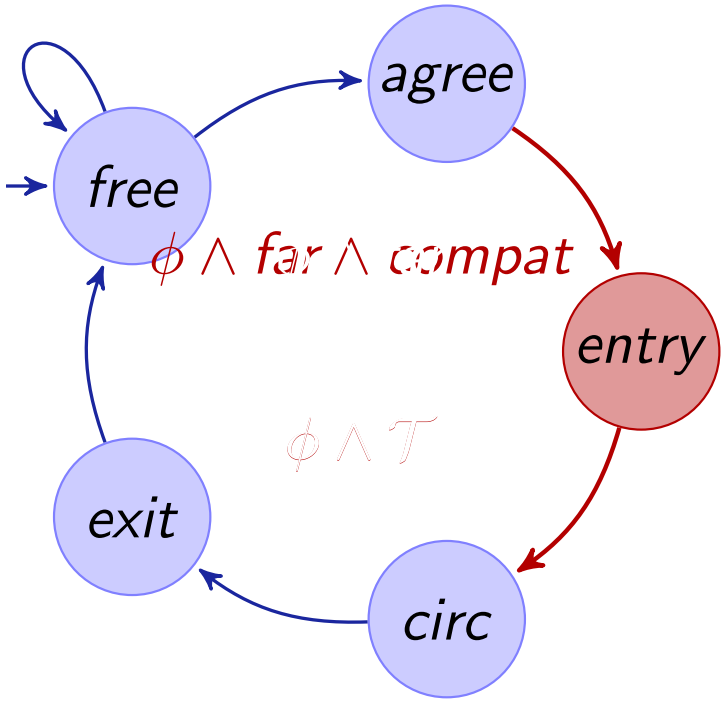


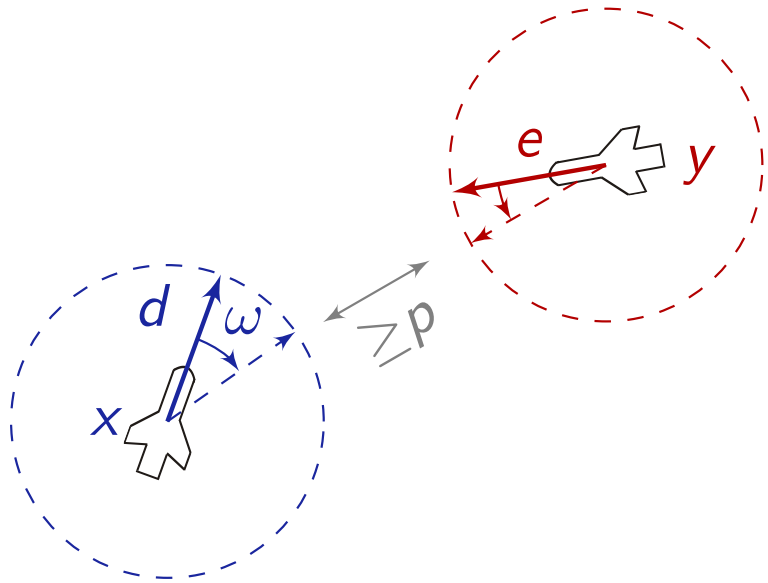


Example (d$\mathcal{L}$ formula of verification subgoal)

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

Flyable Roundabout Maneuver: Entry

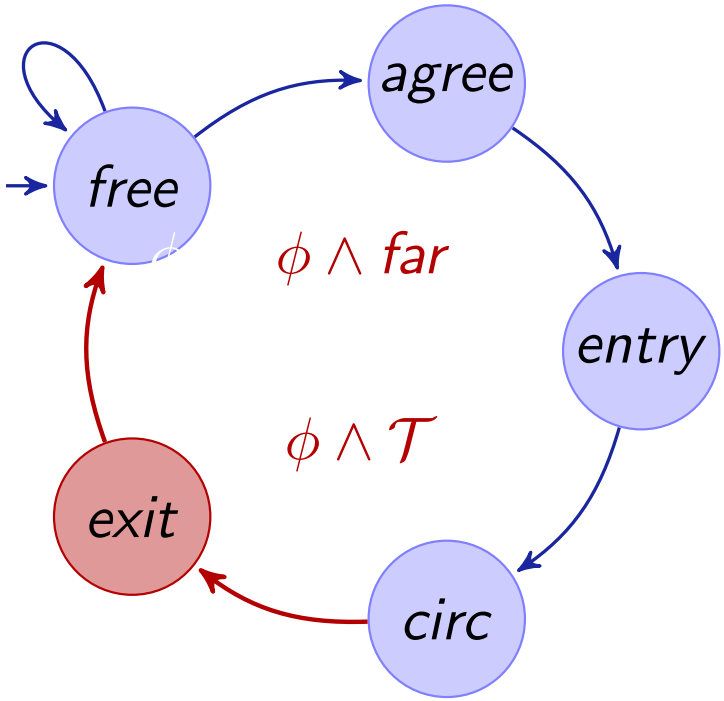


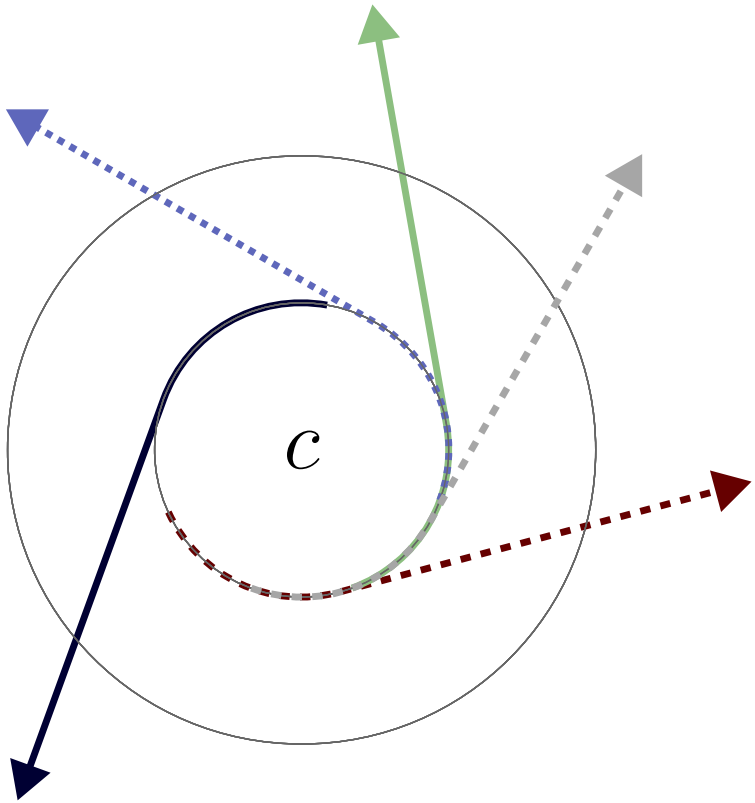


Example (d$\mathcal{L}$ formula of verification subgoal)

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

Flyable Roundabout Maneuver: Exit

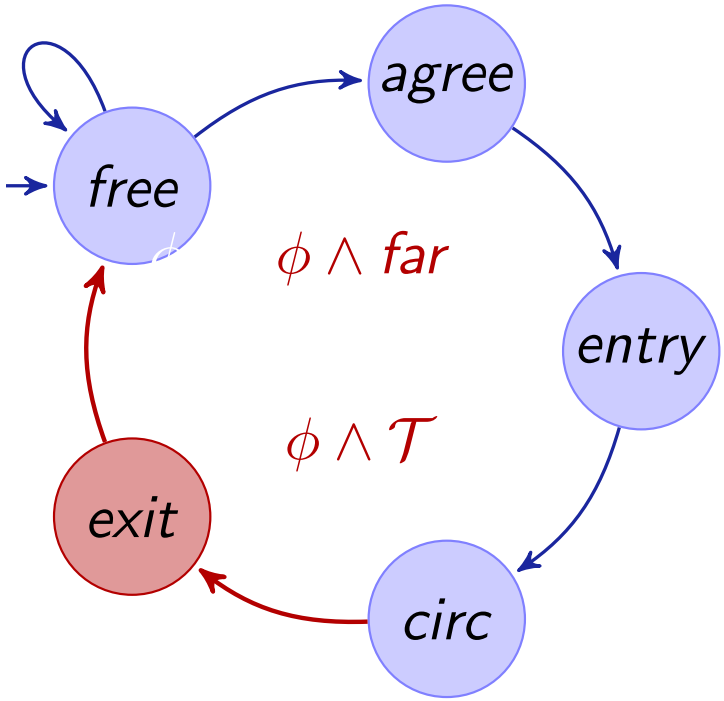


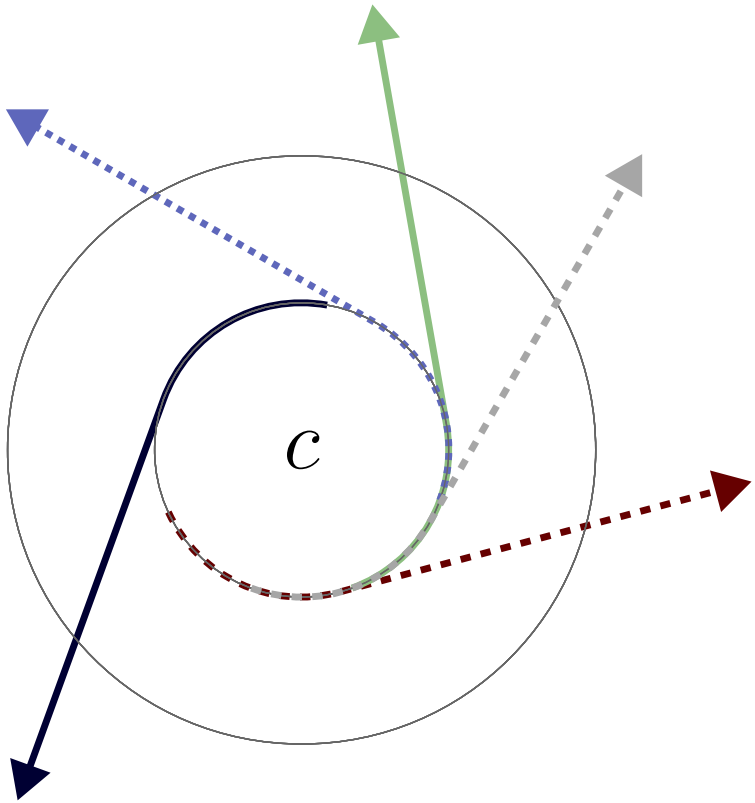


Example (d$\mathcal{L}$ formula of verification subgoal)

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

Flyable Roundabout Maneuver: Exit

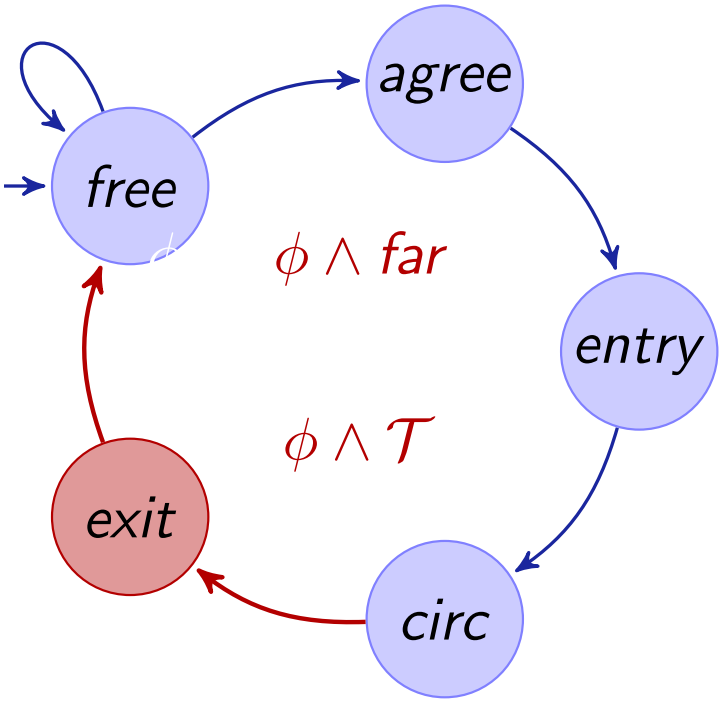


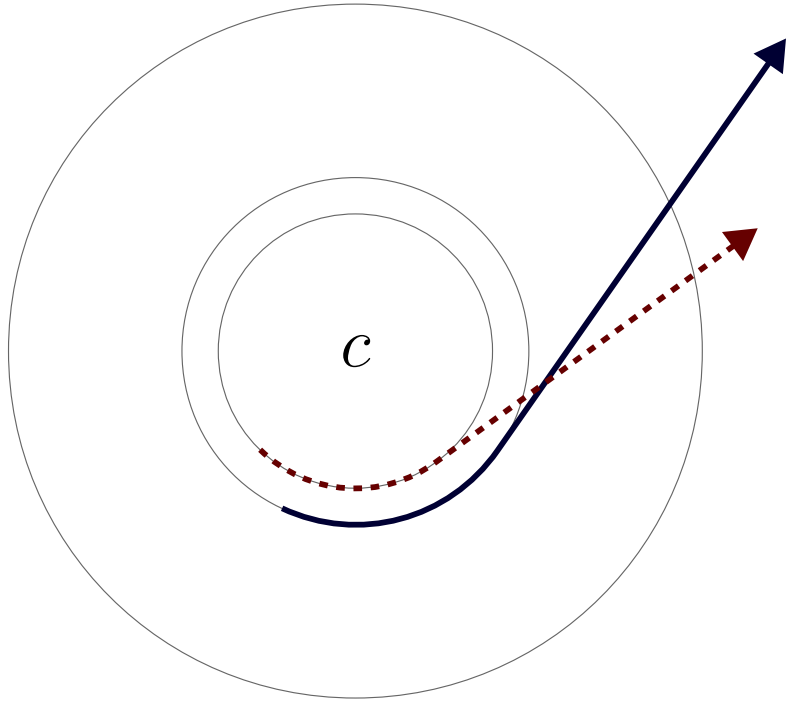


Example (d$\mathcal{L}$ formula of verification subgoal)

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

Flyable Roundabout Maneuver: Exit

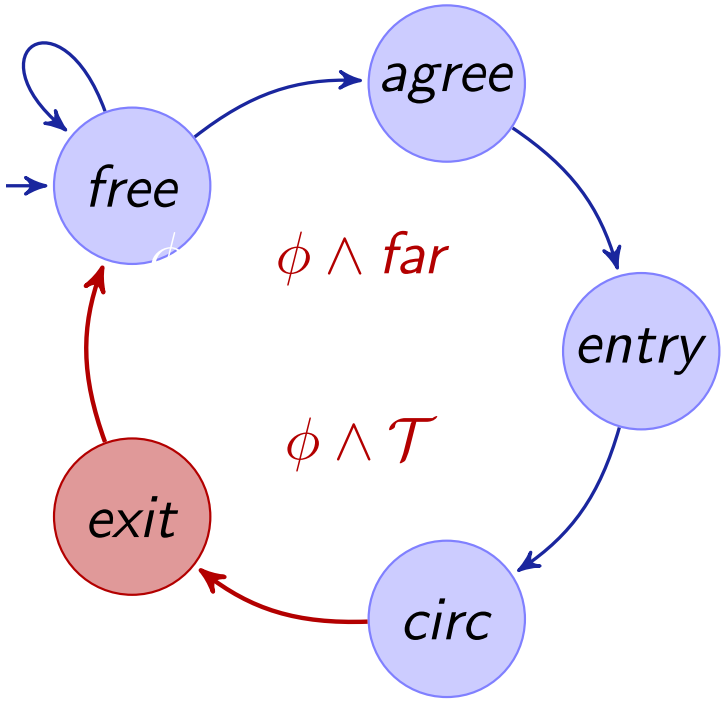


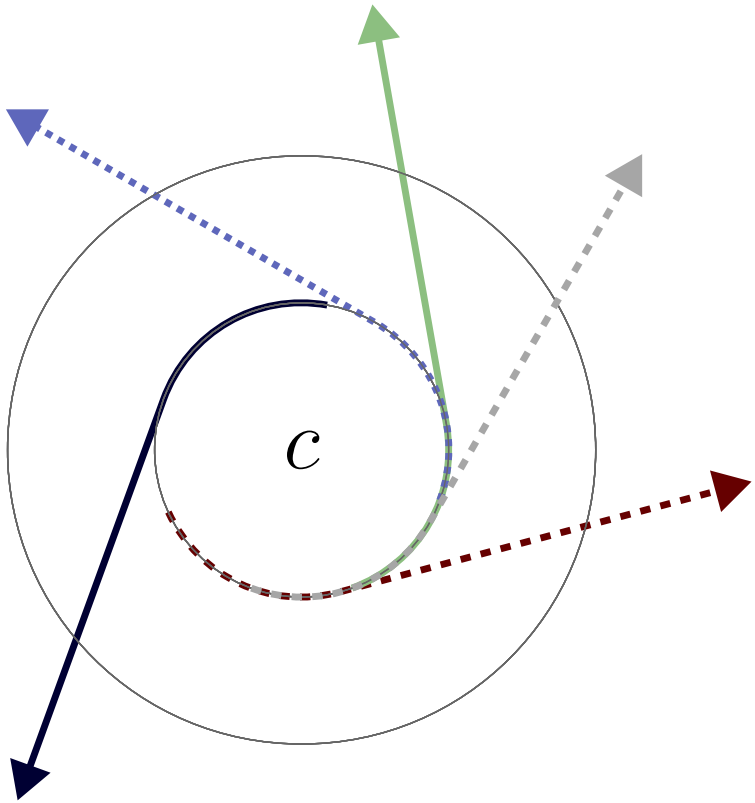


Example (d$\mathcal{L}$ formula of verification subgoal)

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

Flyable Roundabout Maneuver: Exit

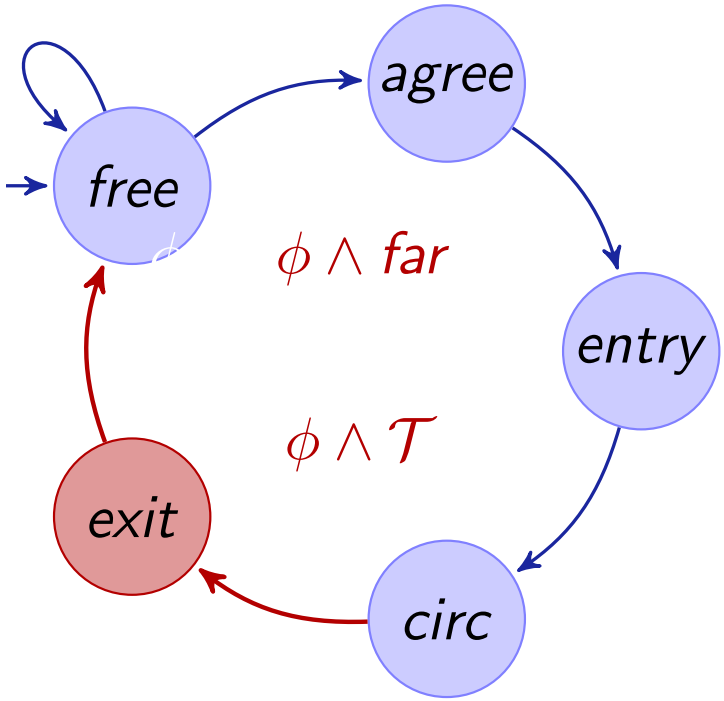


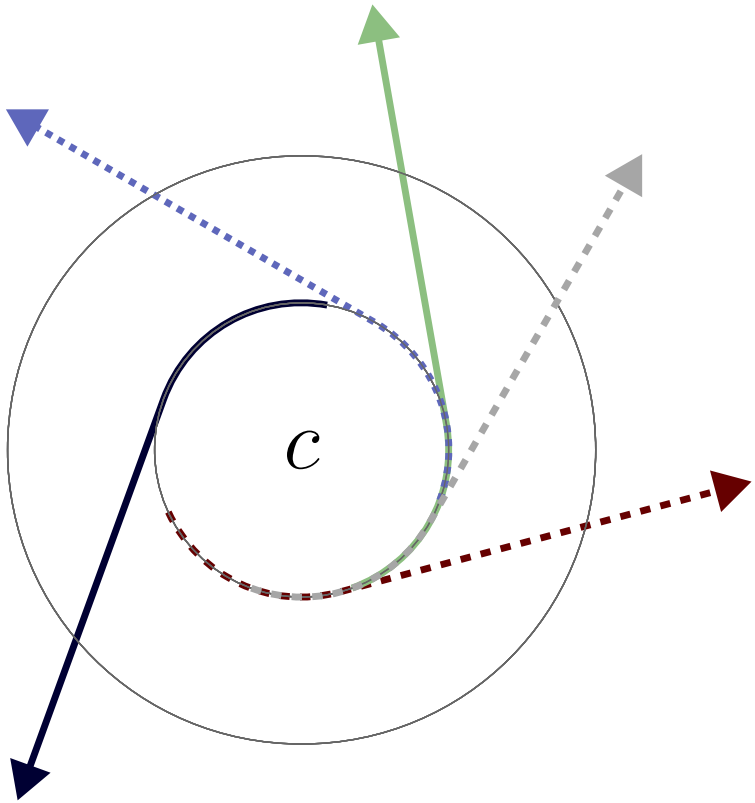


Example (d$\mathcal{L}$ formula of verification subgoal)

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

Flyable Roundabout Maneuver: Exit





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

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

provable automatically!

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

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

provable automatically!

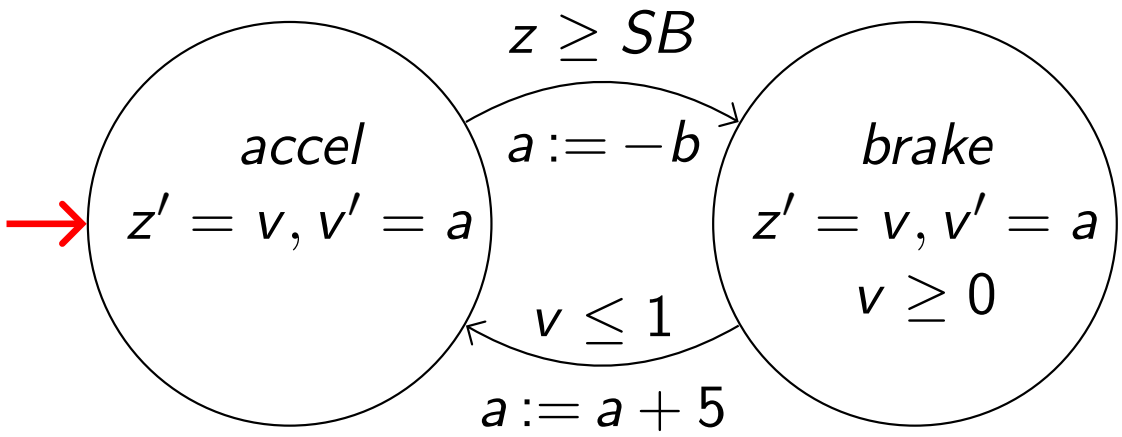
$$
\begin{aligned}
\psi \quad &\equiv \quad \phi \rightarrow [trm^*]\phi \\
\phi \quad &\equiv \quad (x_1-y_1)^2+(x_2-y_2)^2 \geq p^2 \wedge (y_1-z_1)^2+(y_2-z_2)^2 \geq p^2 \\
&\quad \wedge (x_1-z_1)^2+(x_2-z_2)^2 \geq p^2 \wedge (x_1-u_1)^2+(x_2-u_2)^2 \geq p^2 \\
&\quad \wedge (y_1-u_1)^2+(y_2-u_2)^2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

Experimental Results (Air Traffic Control)

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

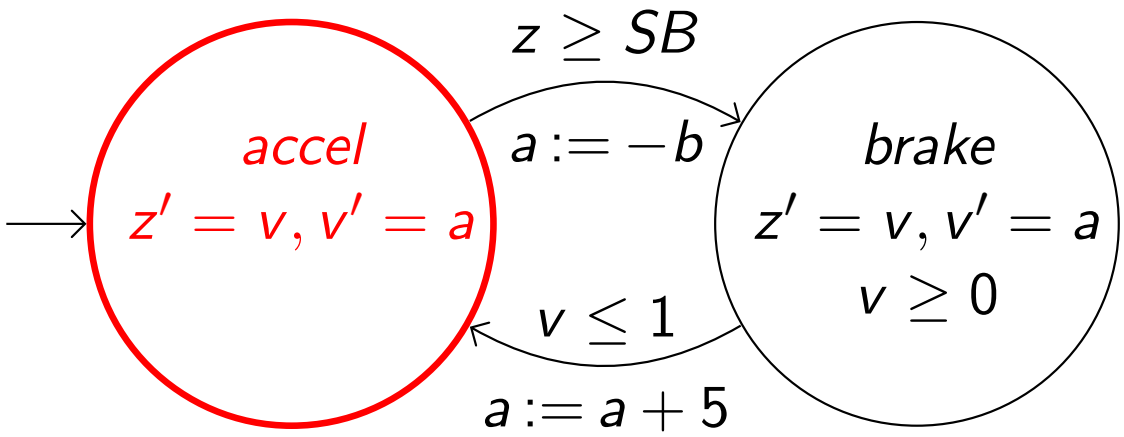
Outline

Embedding Hybrid Automata as Hybrid Programs



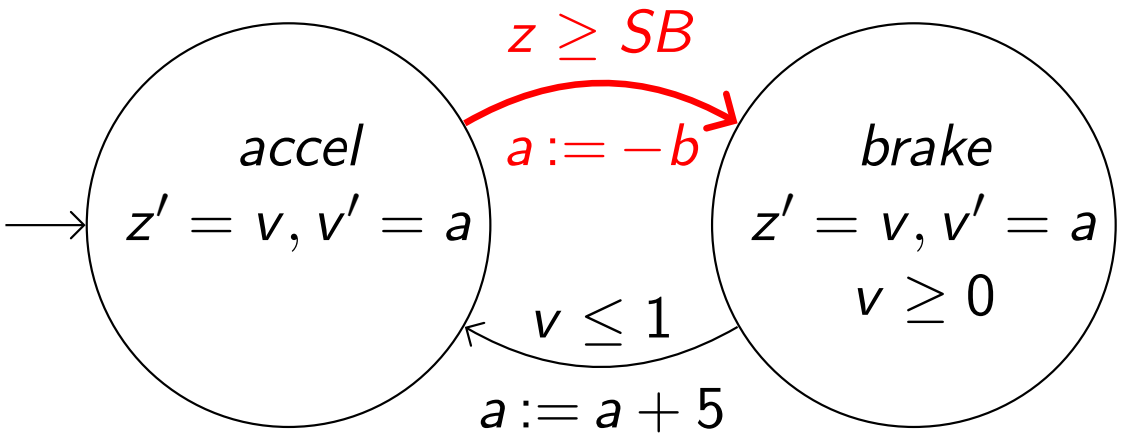
$$
\begin{aligned}
& q := accel; \\
& \big(\quad (?q = accel; \quad z' = v, v' = a) \\
& \cup \ (?q = accel \wedge z \geq SB; \quad a := -b; \quad q := brake; \quad ?v \geq 0) \\
& \cup \ (?q = brake; \quad z' = v, v' = a \wedge v \geq 0) \\
& \cup \ (?q = brake \wedge v \leq 1; \quad a := a + 5; \quad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

Embedding Hybrid Automata as Hybrid Programs



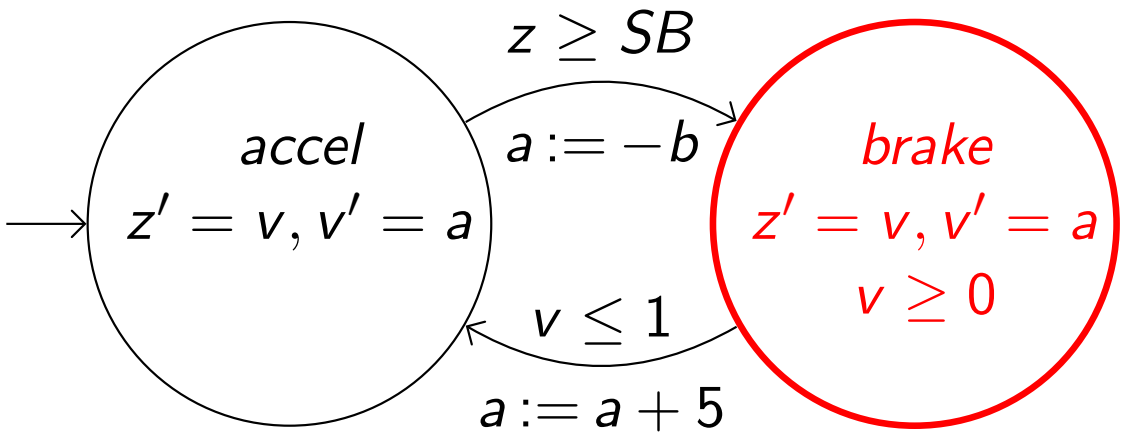
$$
\begin{aligned}
& q := accel; \\
& \big(\quad (?q = accel; \quad z' = v, v' = a) \\
& \cup \ (?q = accel \wedge z \geq SB; \quad a := -b; \quad q := brake; \quad ?v \geq 0) \\
& \cup \ (?q = brake; \quad z' = v, v' = a \wedge v \geq 0) \\
& \cup \ (?q = brake \wedge v \leq 1; \quad a := a + 5; \quad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

Embedding Hybrid Automata as Hybrid Programs



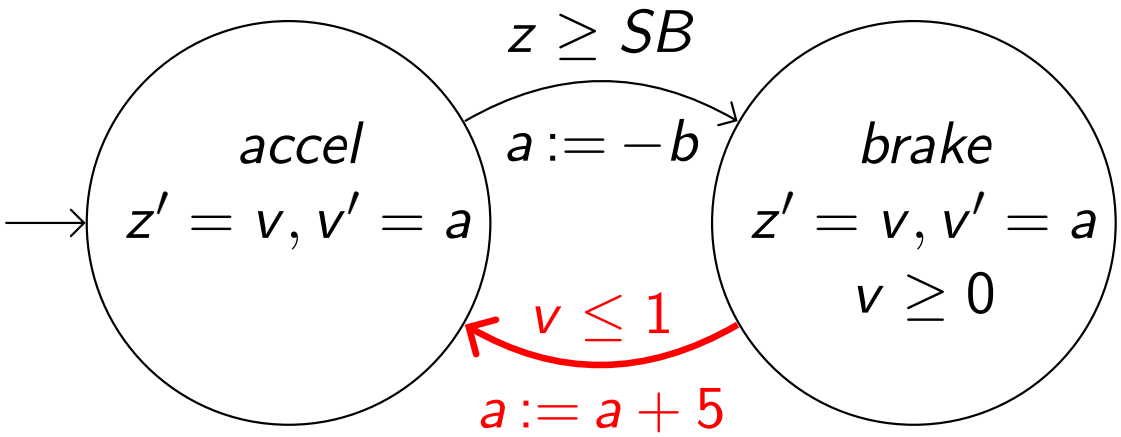
$$
\begin{aligned}
& q := accel; \\
& \big(\quad (?q = accel; \quad z' = v, v' = a) \\
& \cup \ (?q = accel \wedge z \geq SB; \quad a := -b; \quad q := brake; \quad ?v \geq 0) \\
& \cup \ (?q = brake; \quad z' = v, v' = a \wedge v \geq 0) \\
& \cup \ (?q = brake \wedge v \leq 1; \quad a := a + 5; \quad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

Embedding Hybrid Automata as Hybrid Programs



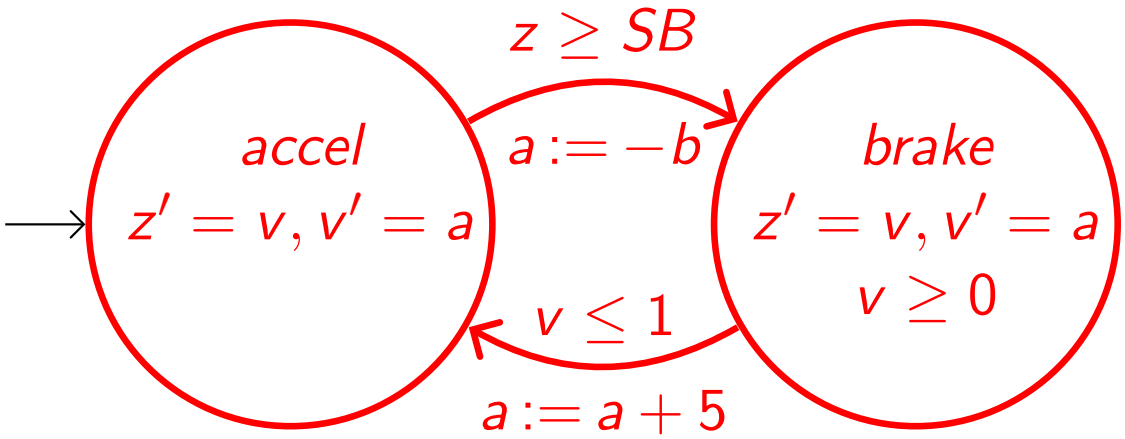
$$
\begin{aligned}
&q := accel; \\
&\big(\quad (?q = accel; \quad z' = v, v' = a) \\
&\cup \ (?q = accel \wedge z \geq SB; \quad a := -b; \quad q := brake; \quad ?v \geq 0) \\
&\cup \ (?q = brake; \quad z' = v, v' = a \wedge v \geq 0) \\
&\cup \ (?q = brake \wedge v \leq 1; \quad a := a + 5; \quad q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

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



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

Outline

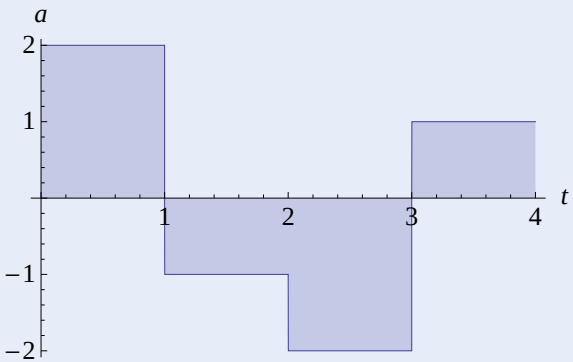
Q: I want to verify my car

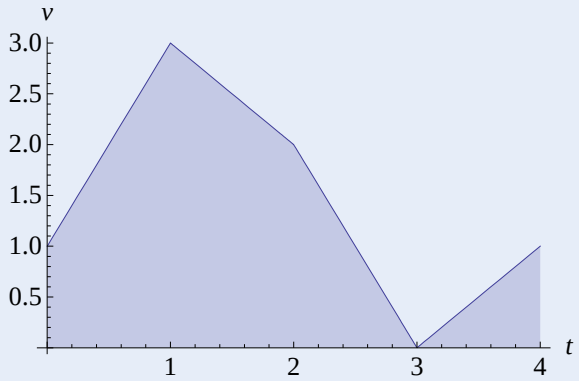
Challenge

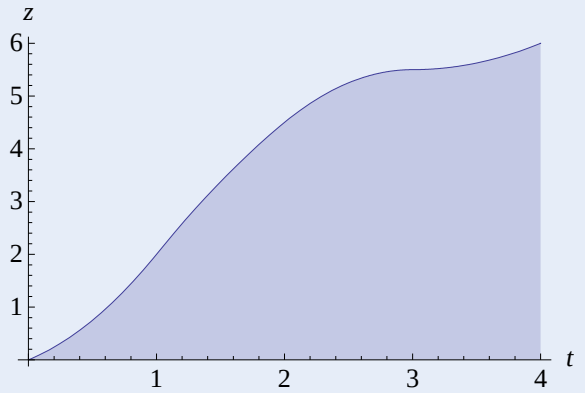
Q: I want to verify my car A: Hybrid systems

Challenge (Hybrid Systems)

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



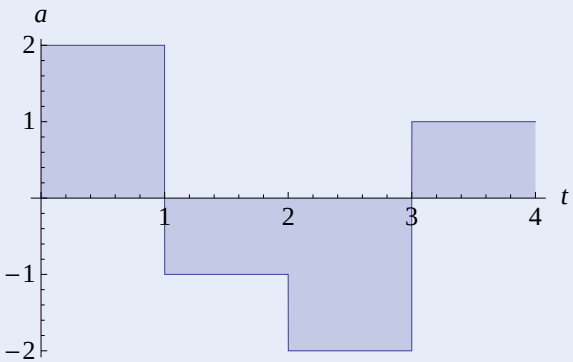


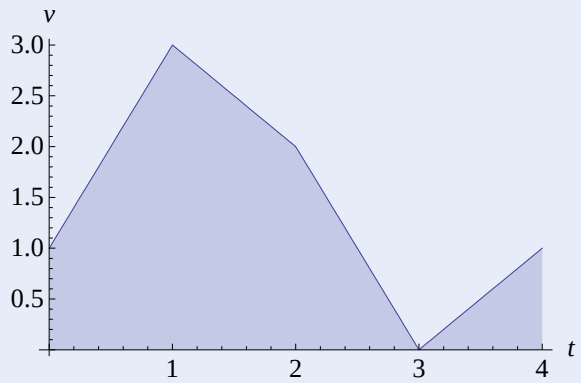


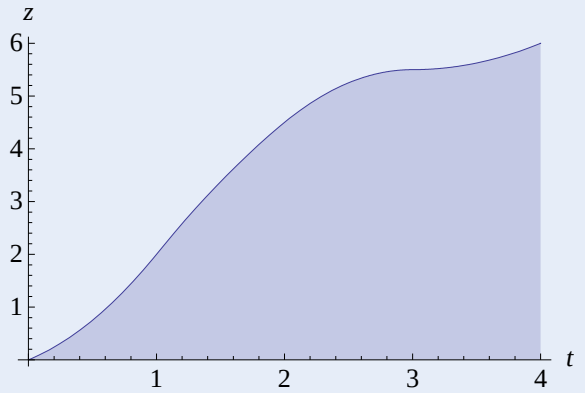
Q: I want to verify my car A: Hybrid systems Q: But there's a lot of cars!

Challenge (Hybrid Systems)

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

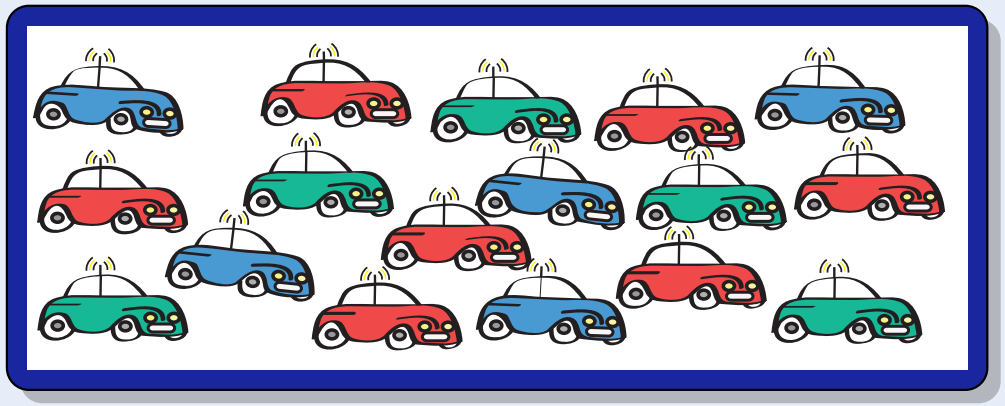






Q: I want to verify a lot of cars

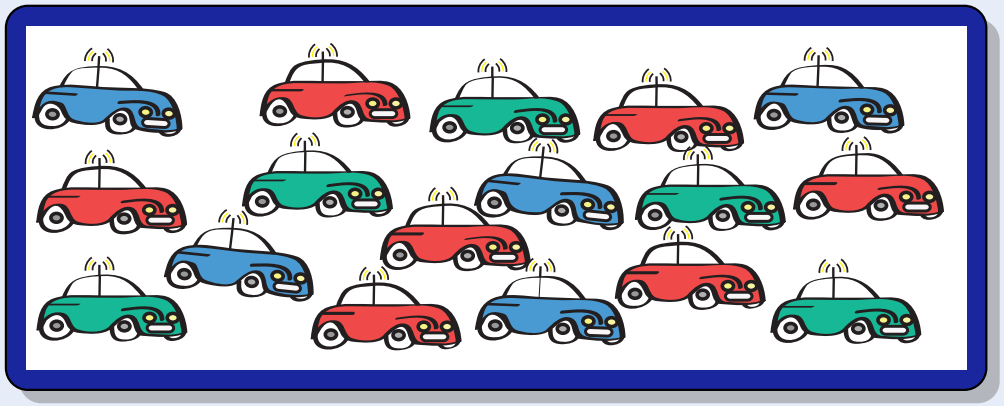
Challenge

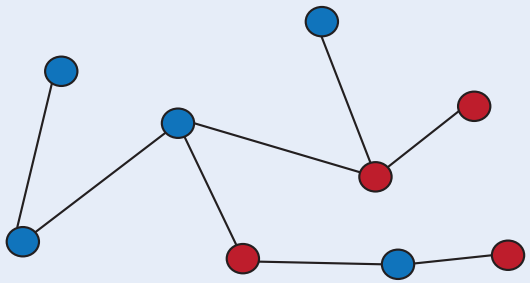


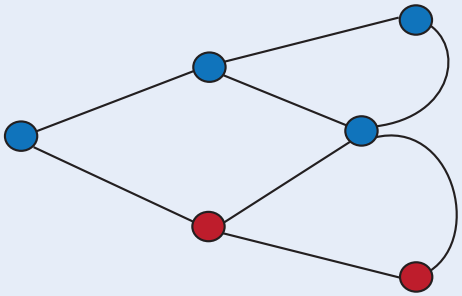
Q: I want to verify a lot of cars A: Distributed systems

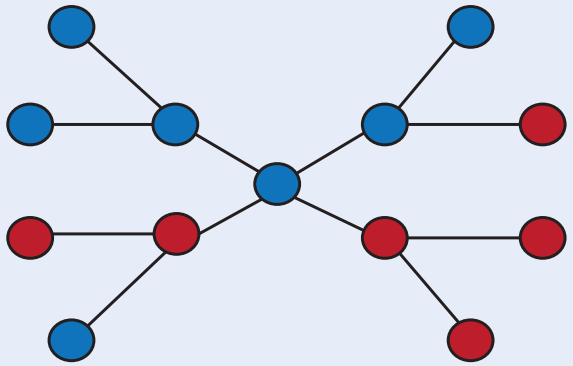
Challenge (Distributed Systems)

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





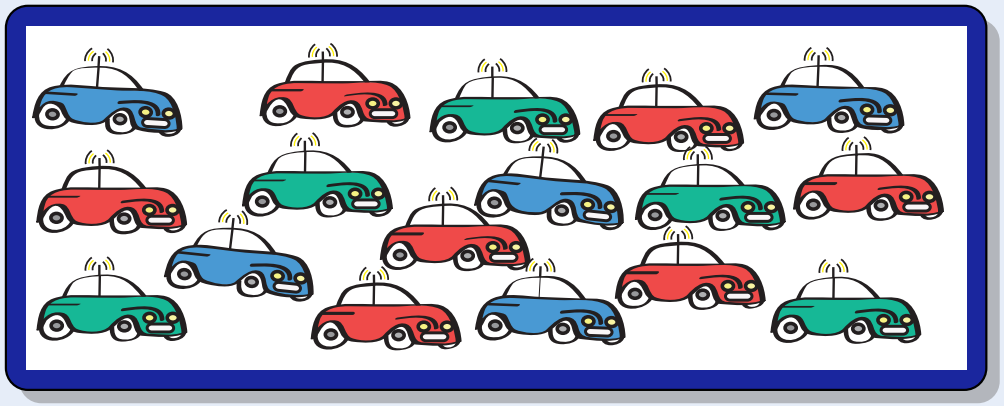


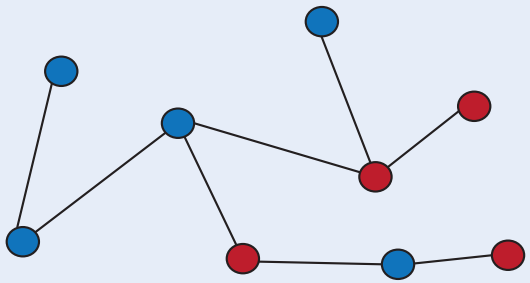


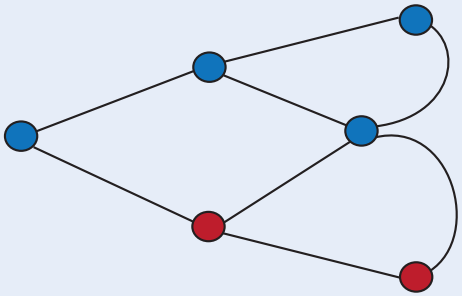
Q: I want to verify a lot of cars A: Distributed systems Q: But they move!

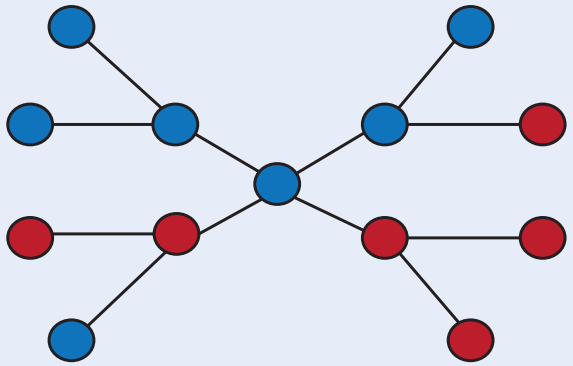
Challenge (Distributed Systems)

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



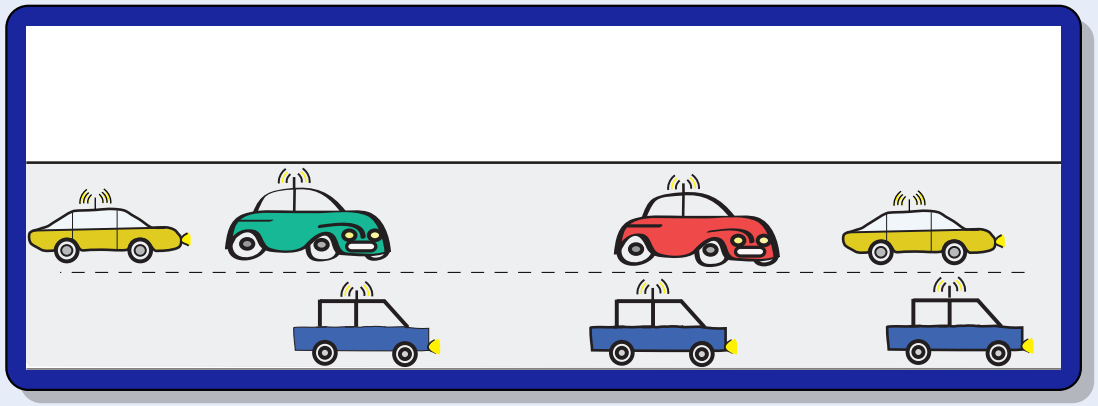






Q: I want to verify lots of moving cars

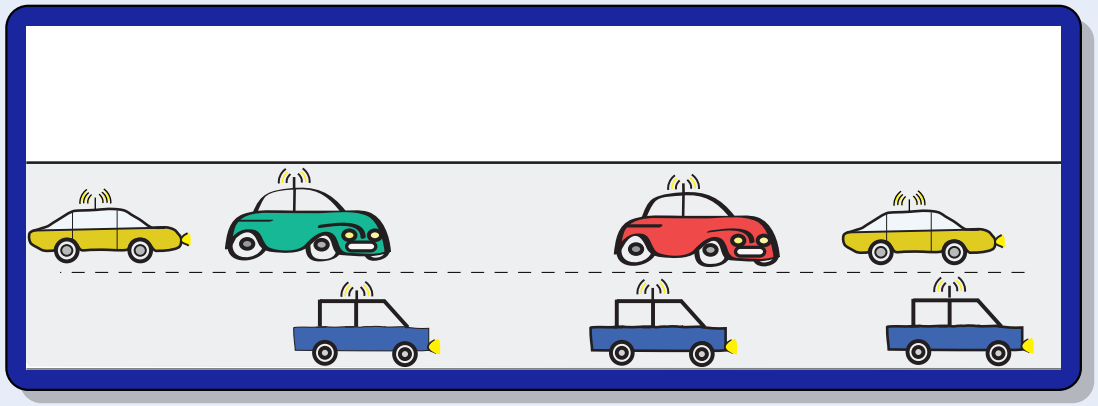
Challenge



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

Challenge (Distributed Hybrid Systems)

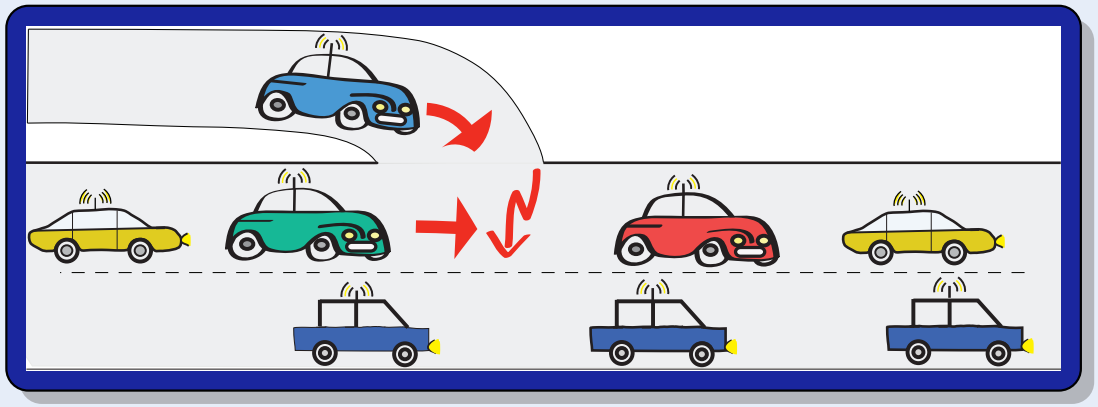
- Continuous dynamics (differential equations)
- Discrete dynamics (control decisions)
- Structural dynamics (remote communication)



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

Challenge (Distributed Hybrid Systems)

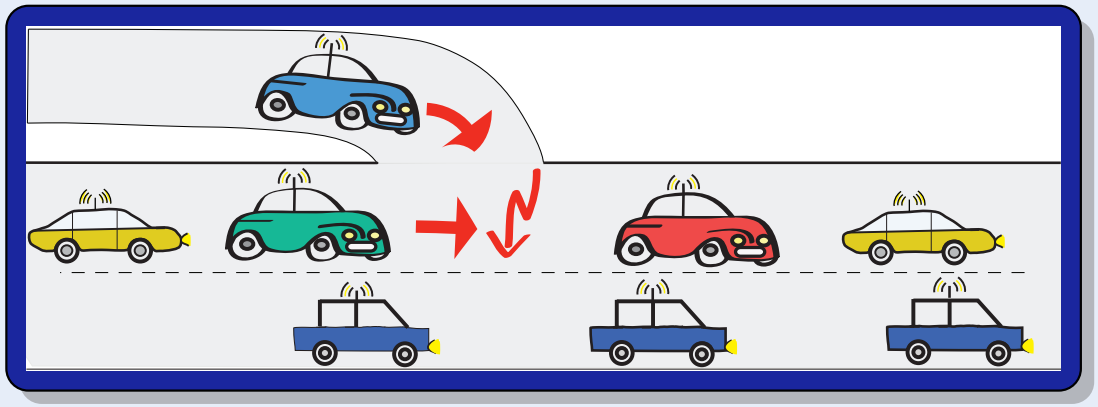
- Continuous dynamics (differential equations)
- Discrete dynamics (control decisions)
- Structural dynamics (remote communication)
- Dimensional dynamics (appearance)



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)

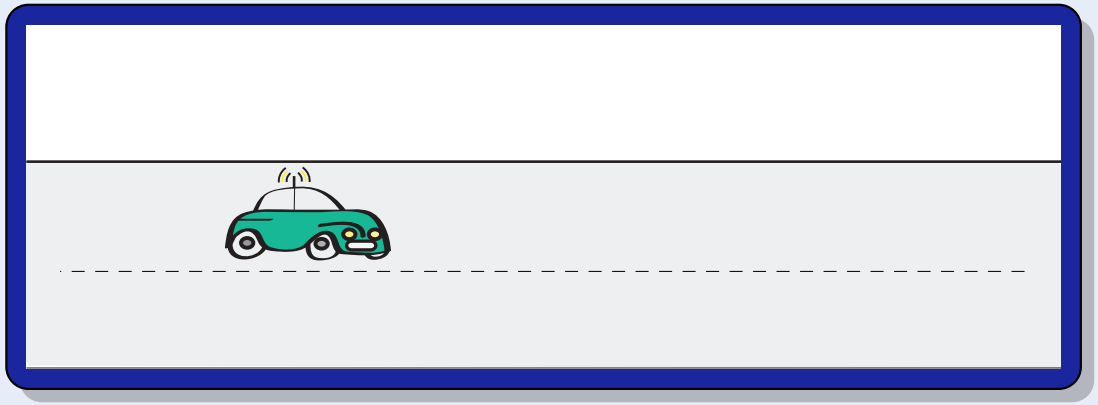


Model for Distributed Hybrid Systems

Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

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

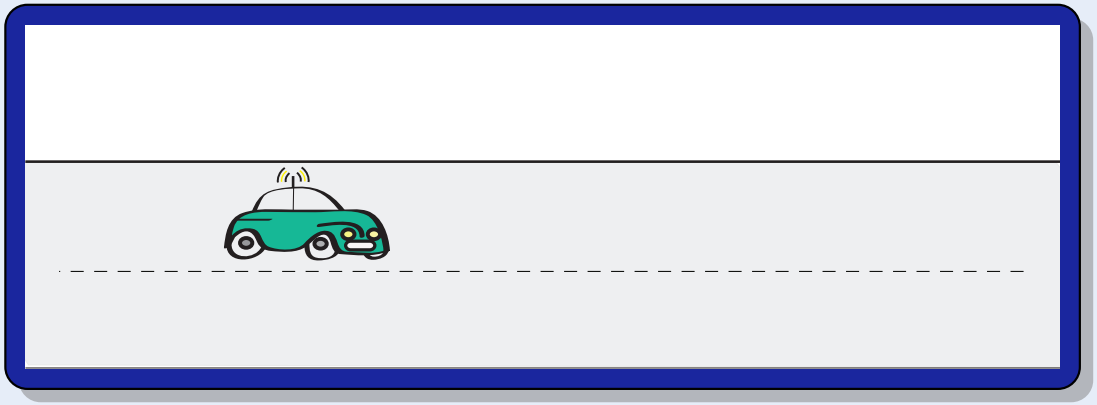


Model for Distributed Hybrid Systems

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)



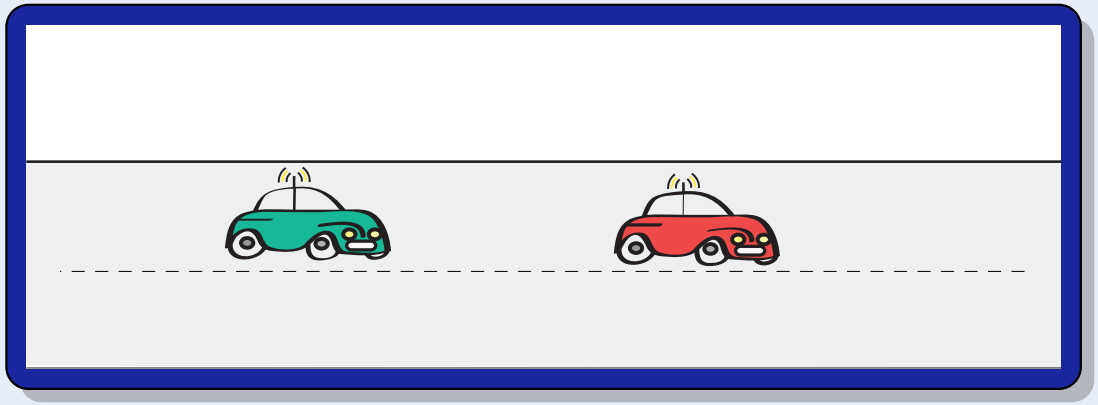
Q: How to model distributed hybrid systems

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)



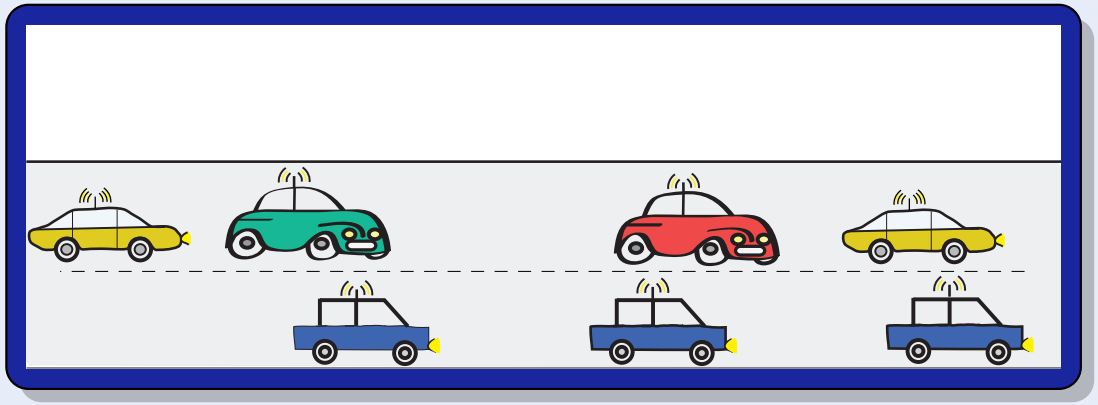
Q: How to model distributed hybrid systems

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)

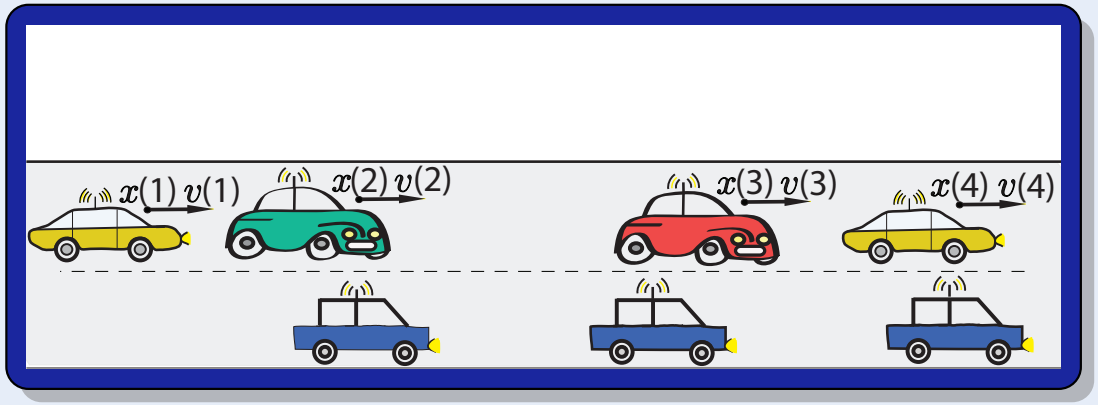
- Continuous dynamics (differential equations)

$$x'' = a$$

- Discrete dynamics (control decisions)

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

- Structural dynamics (remote communication)



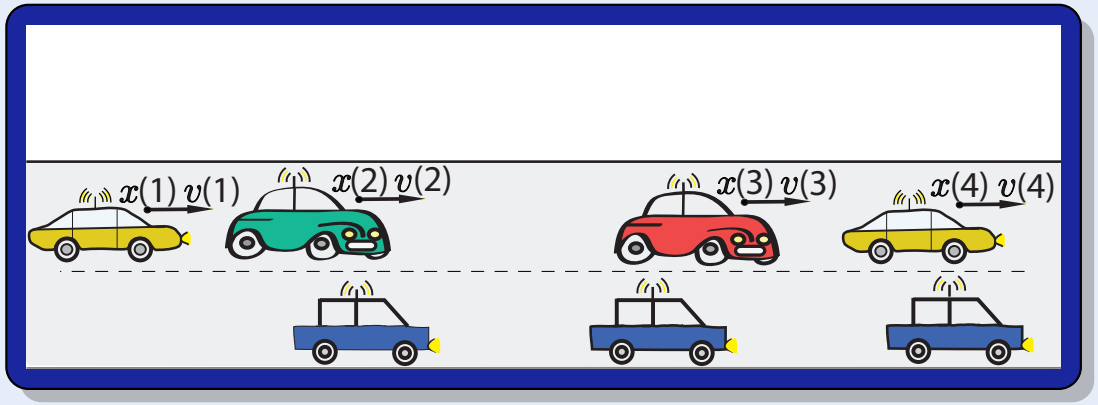
Q: How to model distributed hybrid systems

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)



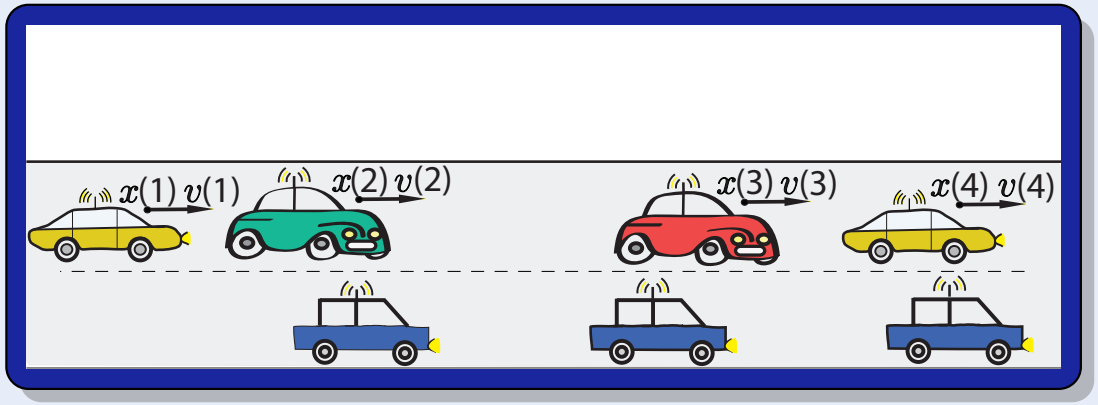
Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

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

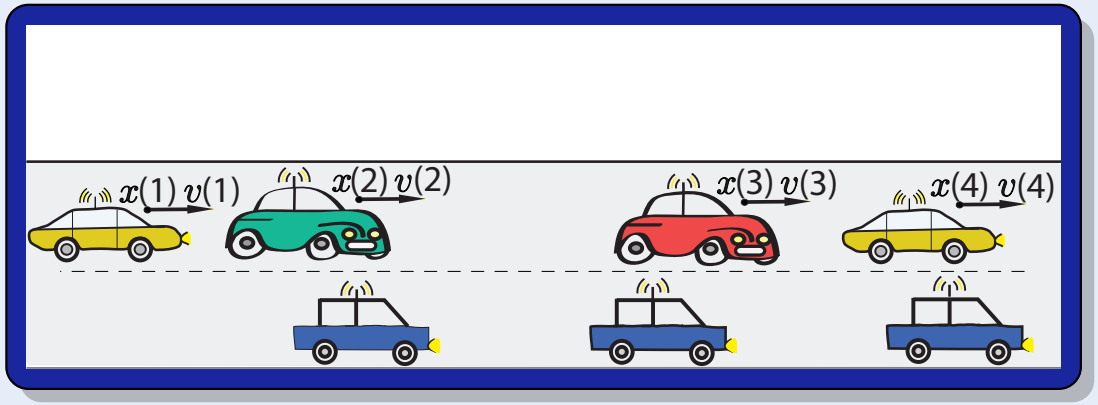
$\forall i\, a(i) := \texttt{if}\, .. \,\texttt{then}\, a \,\texttt{else}\, -b \,\texttt{fi}$

- Structural dynamics (remote communication)



Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)



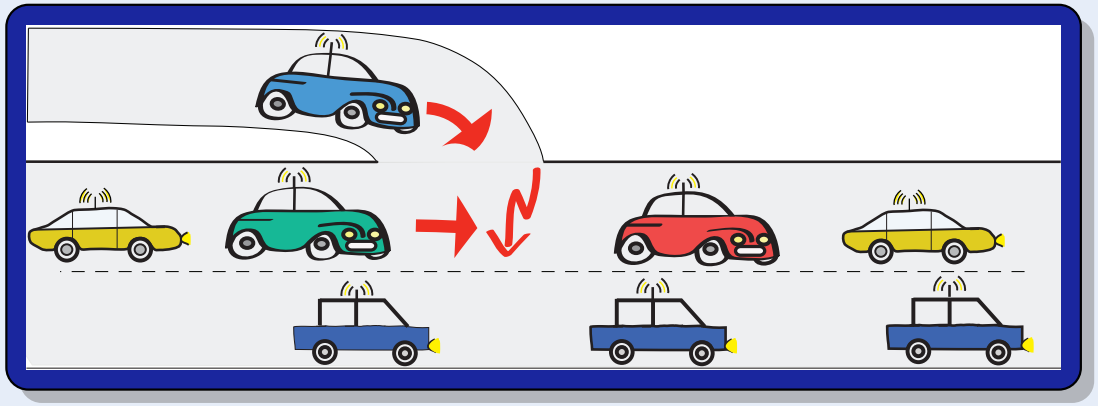
- Continuous dynamics
  (differential equations)
  $\forall i\, x(i)'' = a(i)$
- Discrete dynamics
  (control decisions)

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

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

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

Model (Distributed Hybrid Systems)



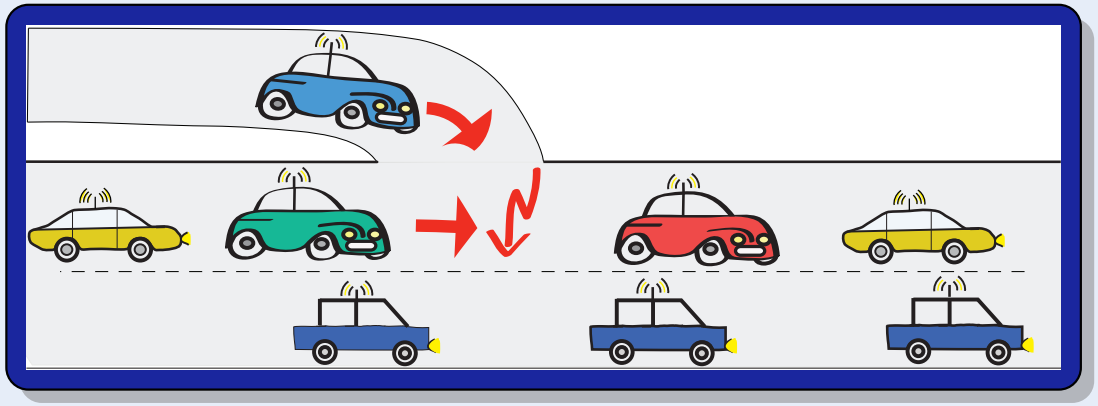
- Continuous dynamics (differential equations)
  $\forall i\, x(i)'' = a(i)$
- Discrete dynamics (control decisions)

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

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

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.

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!)

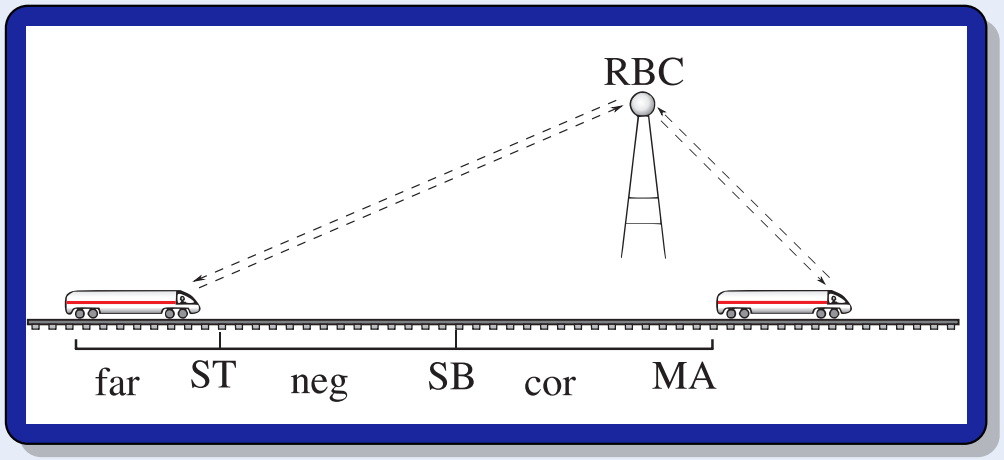
distributed hybrid systems can be verified by recursive decomposition

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

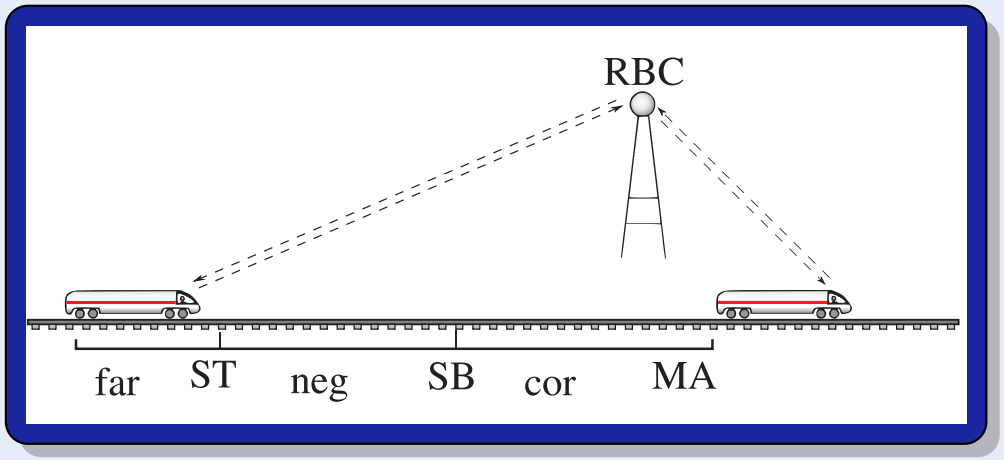
Challenge



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)

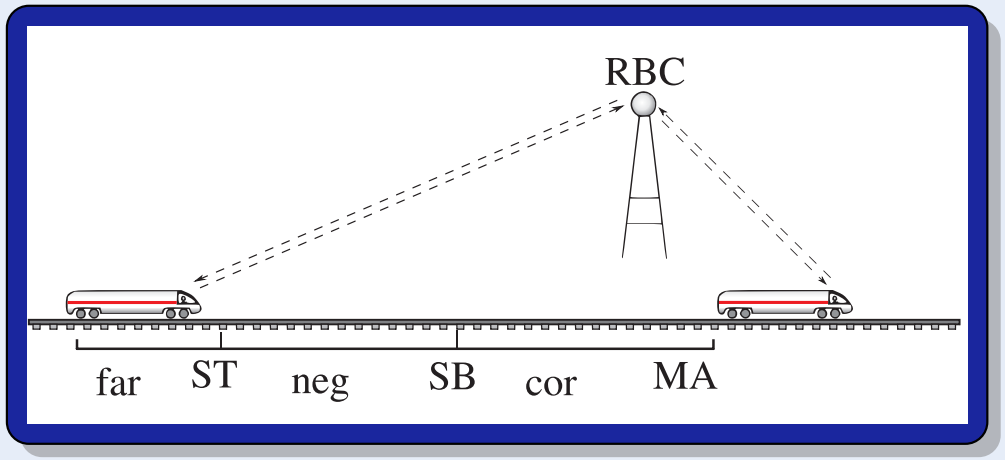


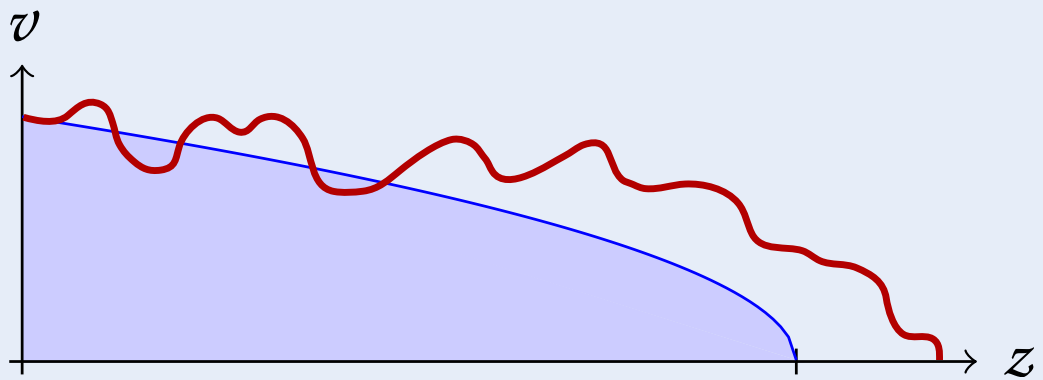
Q: I want to verify uncertain trains A: Stochastic hybrid systems

Challenge (Stochastic Hybrid Systems)

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

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

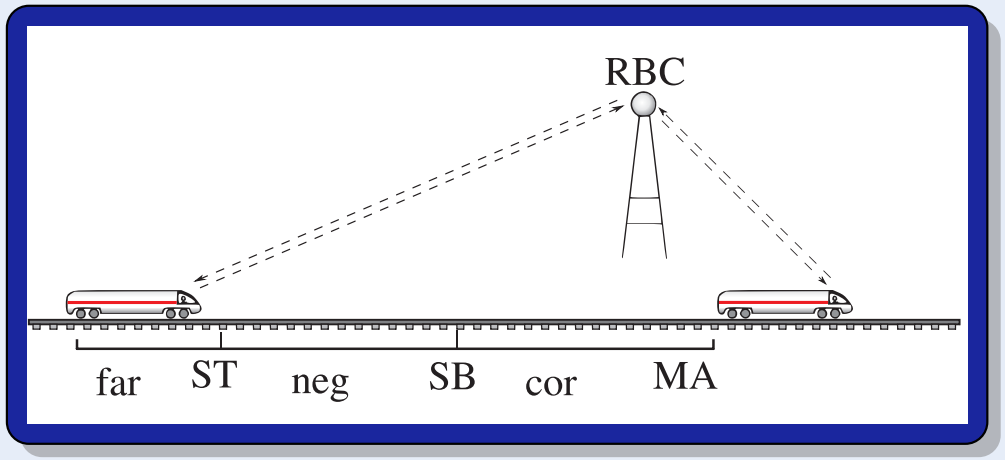


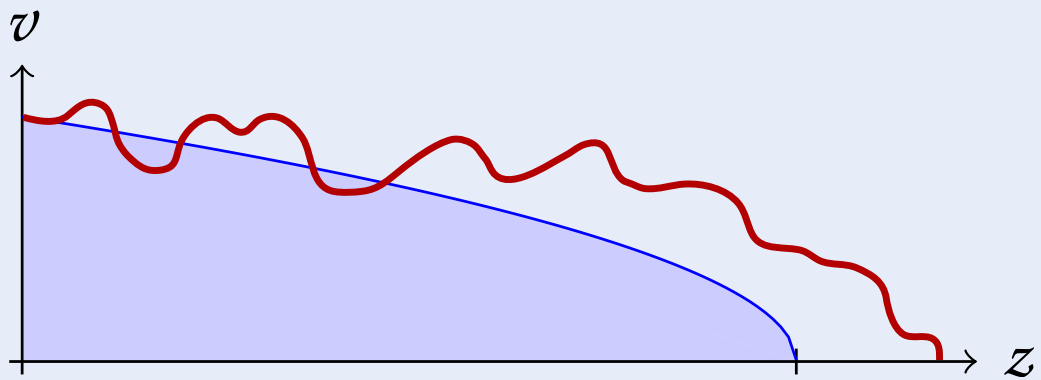


Q: I want to verify uncertain trains A: Stochastic hybrid systems Q: How?

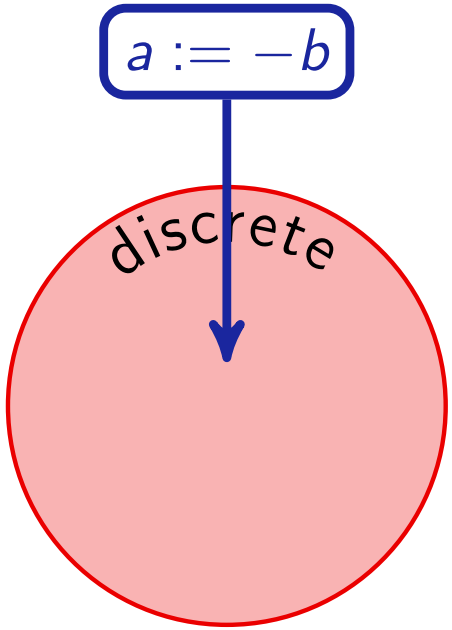
Challenge (Stochastic Hybrid Systems)

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

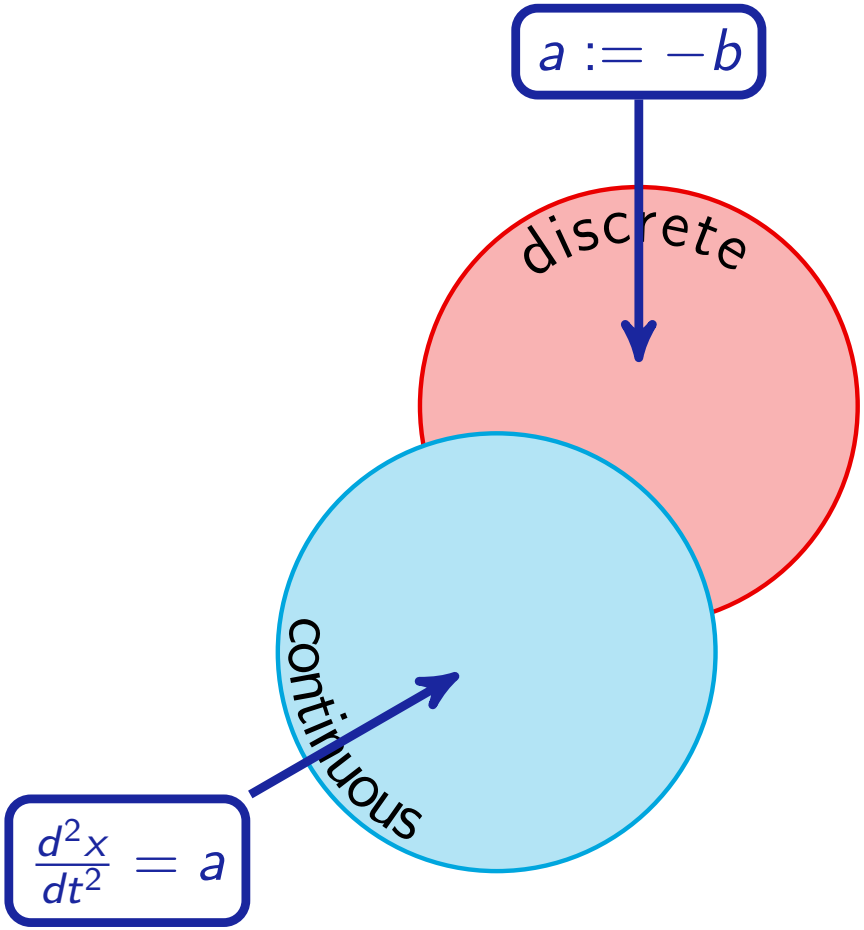




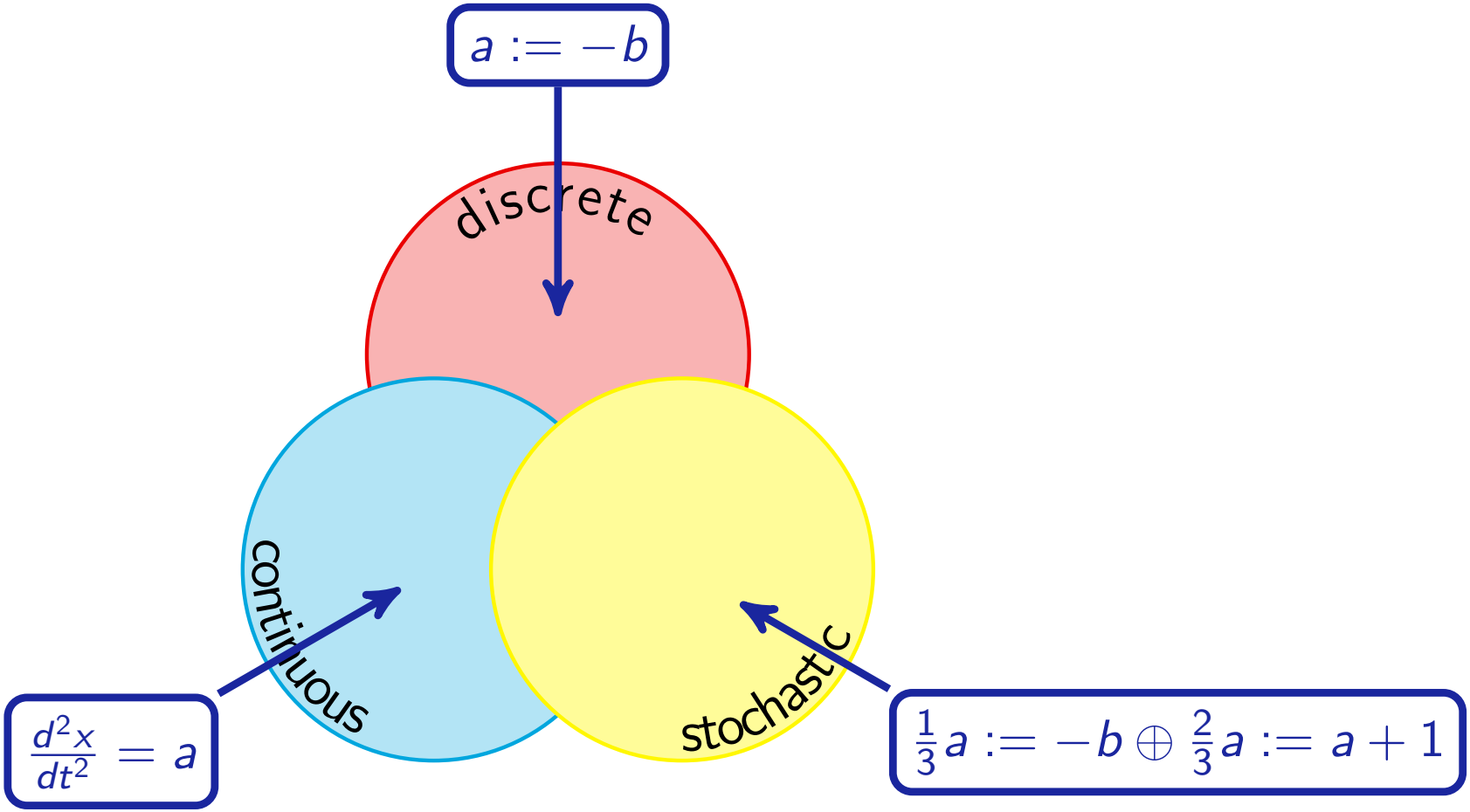
Model for Stochastic Hybrid Systems



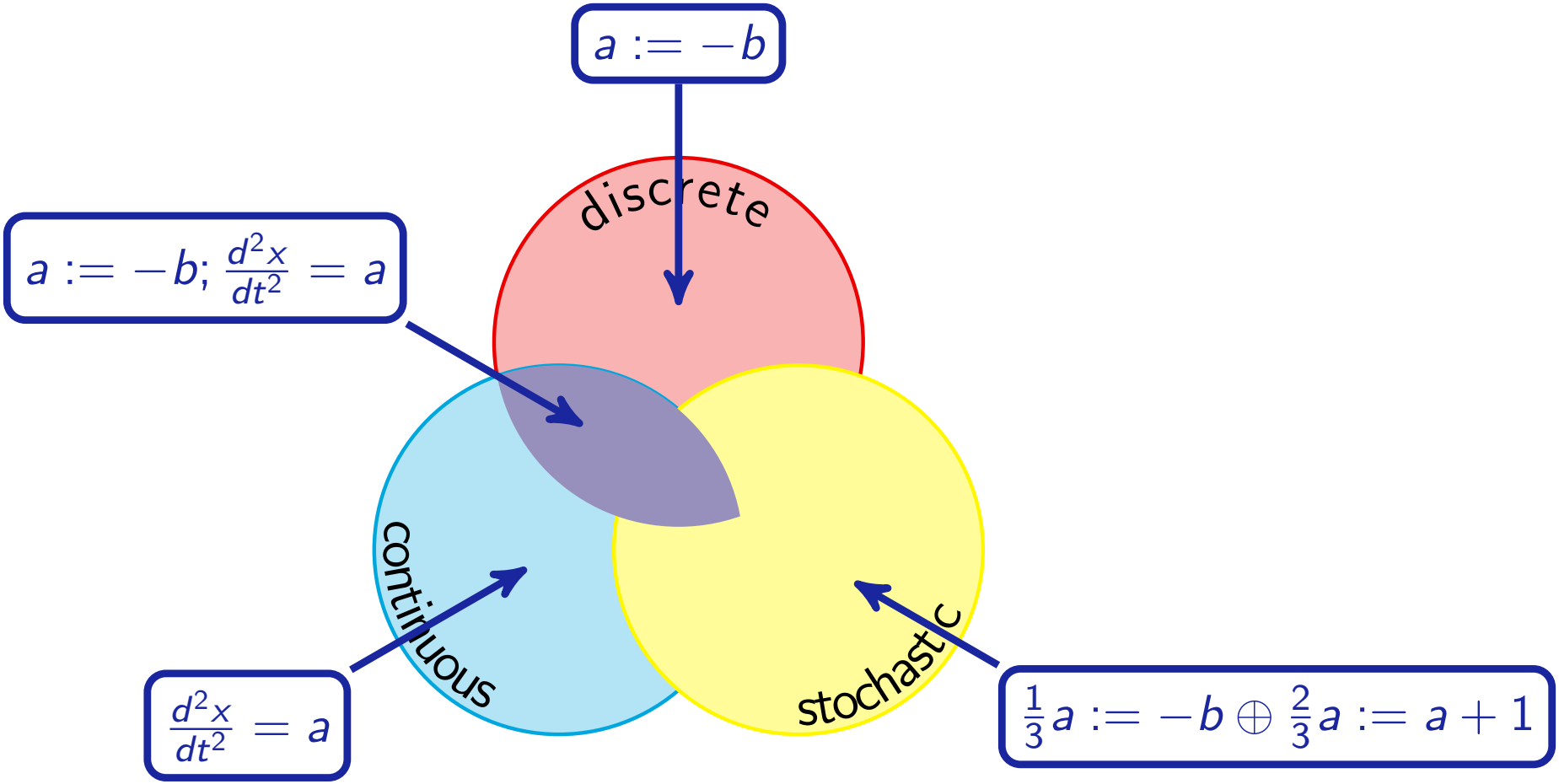
Model for Stochastic Hybrid Systems



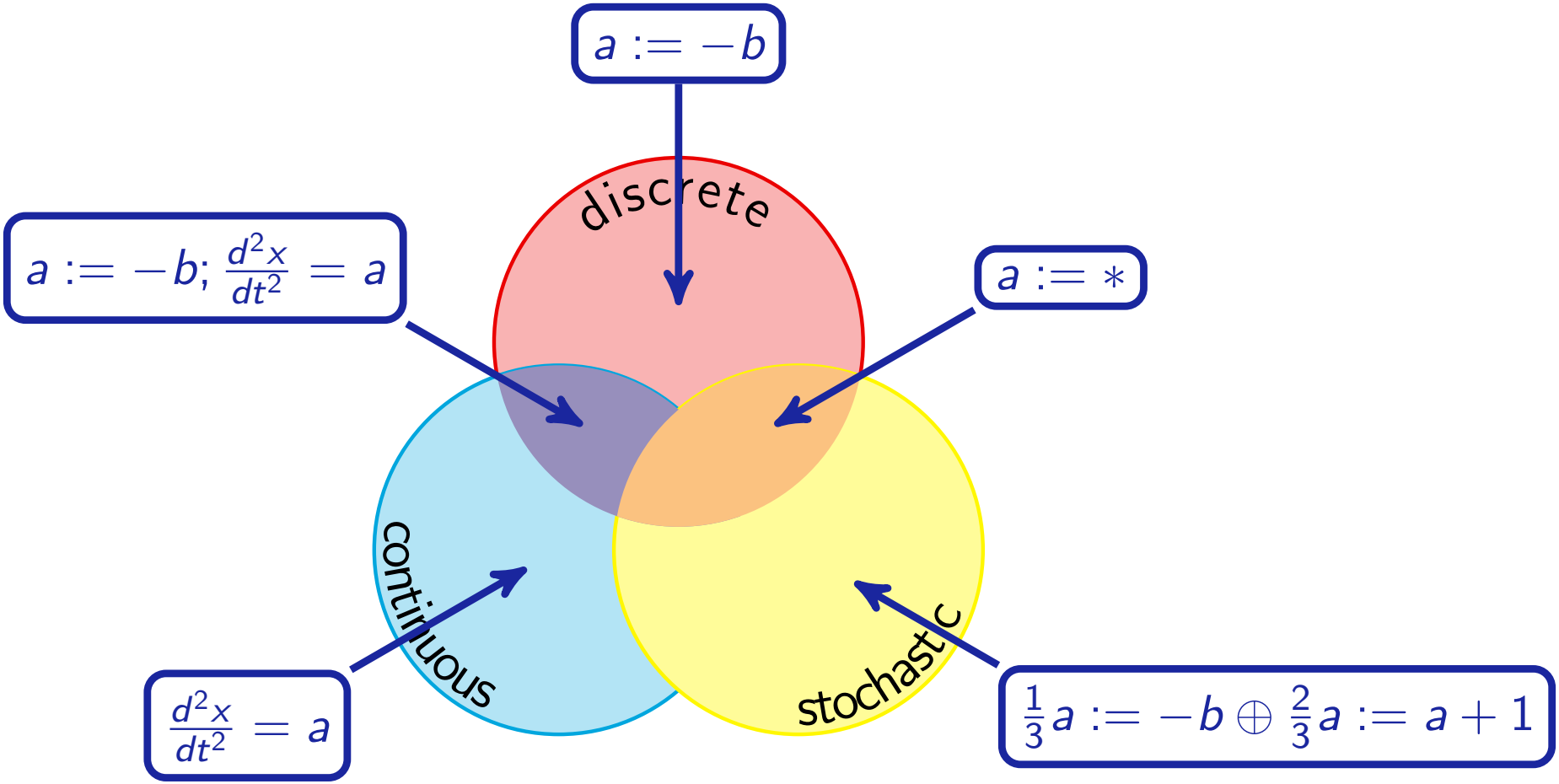
Model for Stochastic Hybrid Systems



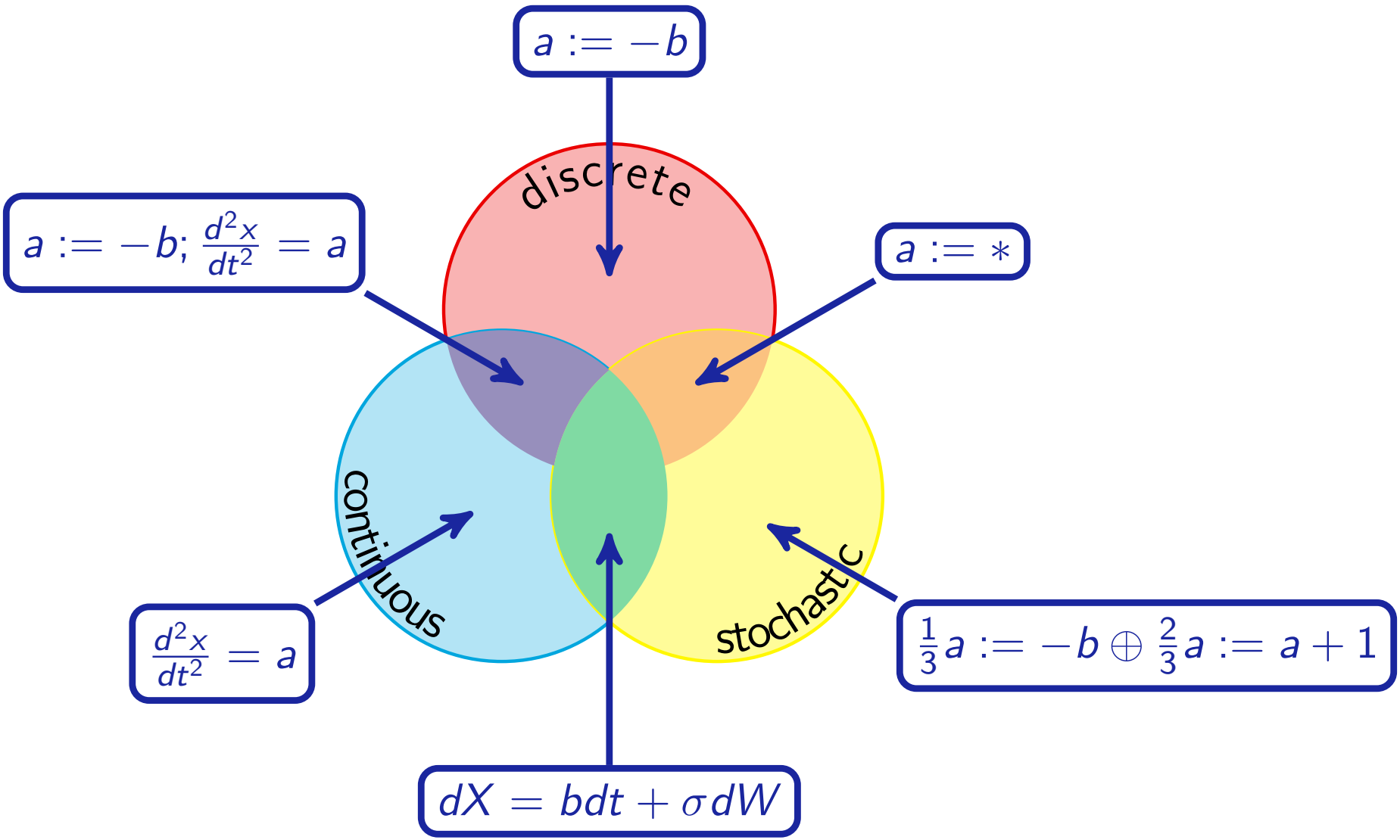
Model for Stochastic Hybrid Systems



Model for Stochastic Hybrid Systems



Model for Stochastic Hybrid Systems



Model for Stochastic Hybrid Systems