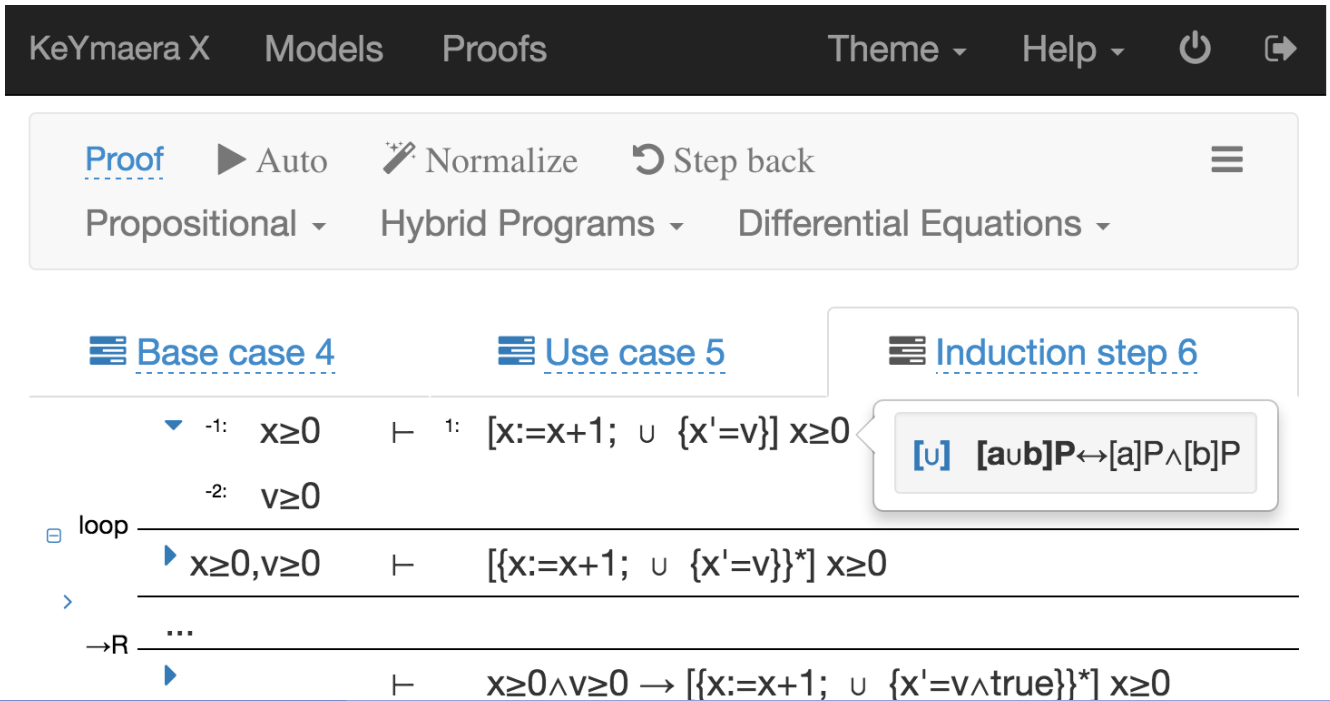
KeYmaera X Tutorial
Tactics and Proofs for Cyber-Physical Systems

Nathan Fulton Stefan Mitsch André Platzer

http://keymaeraX.org/
Computer Science Department
Carnegie Mellon University, Pittsburgh, PA



Outline

Outline

Correctness Questions in Cyber-Physical System Design

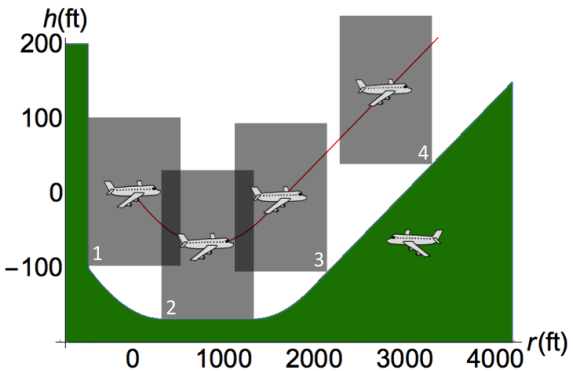
Safety The system must be safe under all circumstances

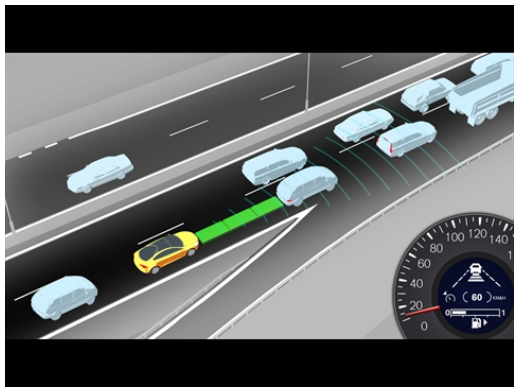
Liveness The system must reach a given goal

How do we make cyber-physical systems safe?

Extensive testing?
Code reviews?

When are we done? How many test cases are enough? Did we cover all relevant tests?





Benefits of Logical Foundations for CPS V & V

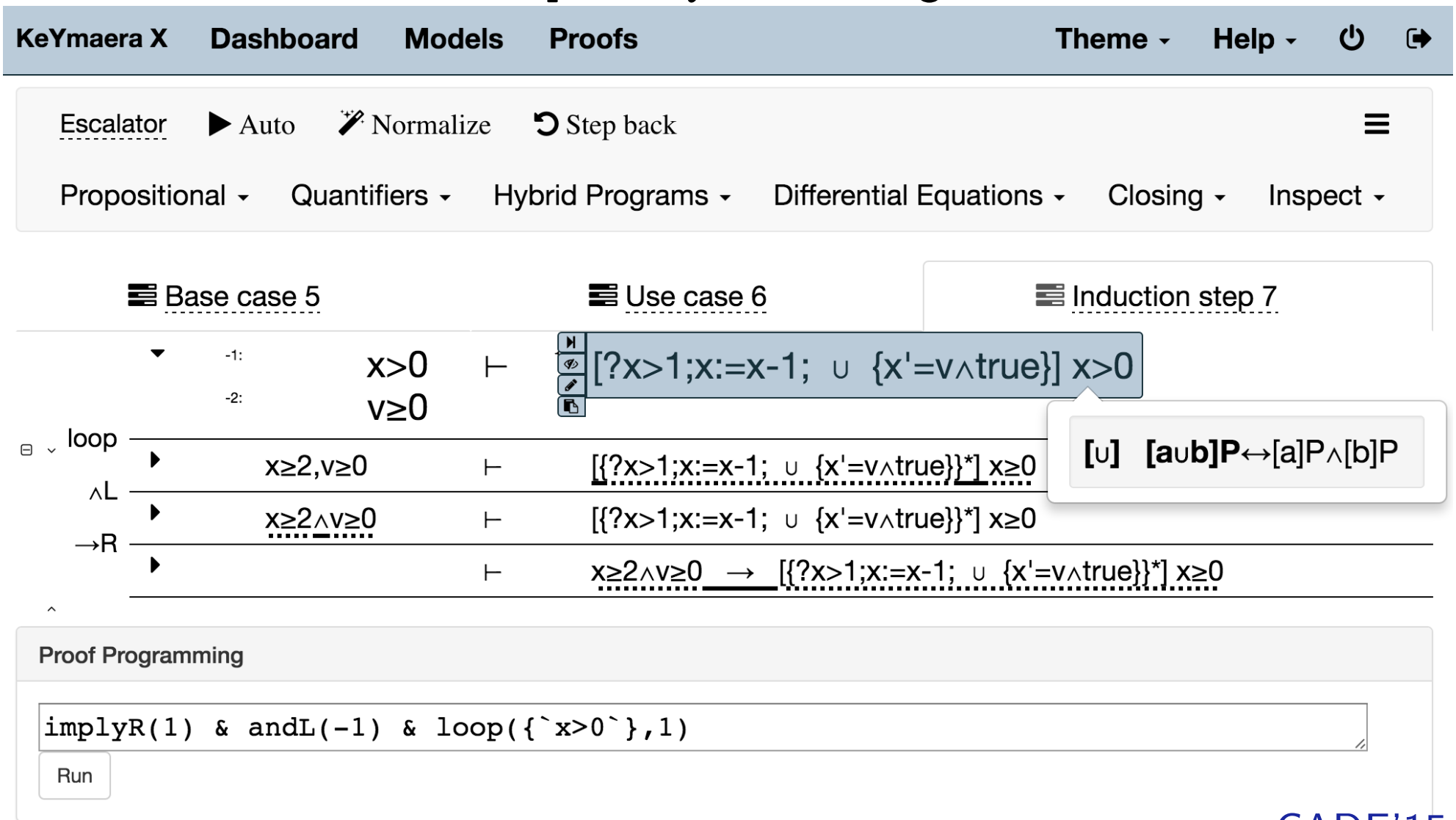
Proofs		LICS'12,JAR'16
Safety	Formalize system properties: What is "Safe"? "Reach goal"?	
Models	Formalize system models, clarify behavior	
Assumptions	Make assumptions explicit rather than silently	
Predictions	Safety analysis predicts behavior for infinitely many cases	
Constraints	Reveal invariants, switching conditions, operating conditions	
Design	Invariants/proofs guide safe controller design	

Byproducts		
Analysis	Determine design trade-offs & feasibility early	arXiv
Synthesis	Turn models into code & safety monitors	ModelPlex
Certificate	Proofs as evidence for certification	CPP'16

Tools		
KeYmaera X	aXiomatic Tactical Theorem Prover for CPS	CADE'15

An aXiomatic Tactical Theorem Prover for CPS

KeYmaera X `http://keymaeraX.org/`



CADE'15

An aXiomatic Tactical Theorem Prover for CPS

KeYmaera X `http://keymaeraX.org/`

Small Core	Increases trust, modularity, enables experimentation	(1652)
Tactics	Bridging between small core and powerful reasoning steps	(Hilbert) (Sequent)
Separation	Tactics can make courageous inferences Core establishes soundness	
Search&Do	Search-based tactics follow proof search strategies Constructive tactics directly build a proof	
Interaction	Interactive proofs mixed with tactical proofs and proof search	
Extensible	Flexible for new algorithms, new tactics, new logics, new proof rules, new axioms, ...	
Customize	Modular user interface, API	

CADE'15

KeYmaera X Microkernel for Soundness

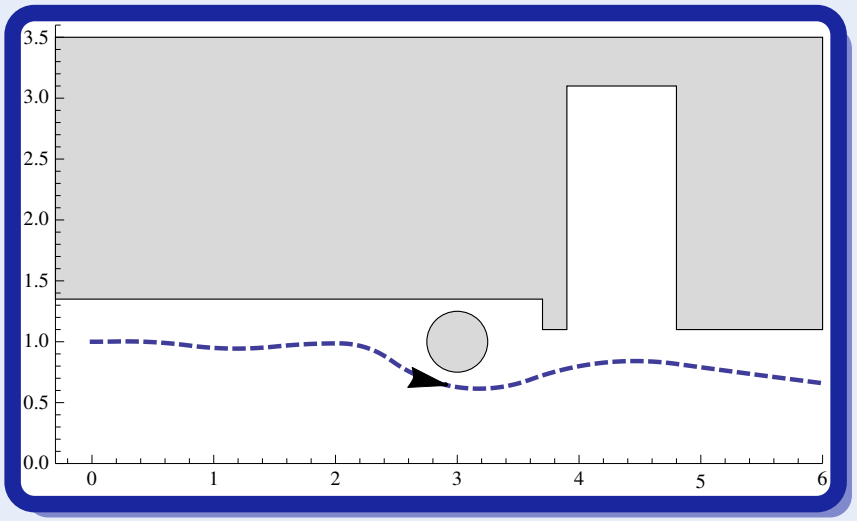
	≈LOC		
KeYmaera X	1 652		hybrid prover
KeYmaera	65 989		
KeY	51 328		Java
Nuprl	15 000	+ 50 000	general math
MetaPRL	8 196		
Isabelle/Pure	8 913		
Coq	16 538		
HOL Light	396		
PHAVer	30 000		hybrid verifier
HSolver	20 000		
SpaceEx	100 000		
Flow*	25 000		
dReal	50 000	+ millions	
HyCreate2	6 081	+ user model analysis	

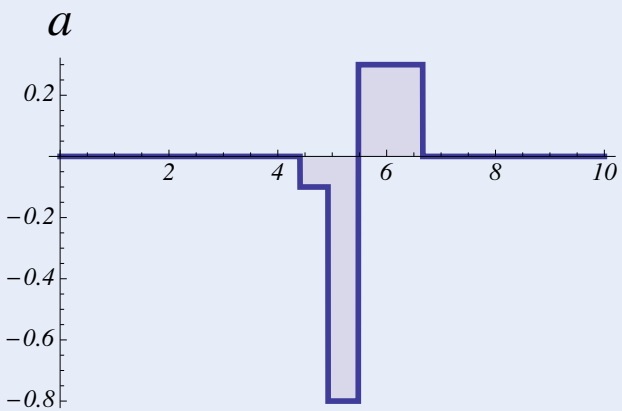
Disclaimer: Self-reported estimates of the soundness-critical lines of code + rules

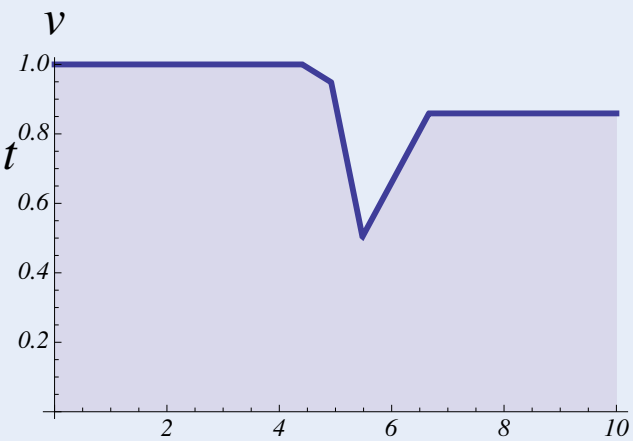
Challenge (Hybrid Systems)

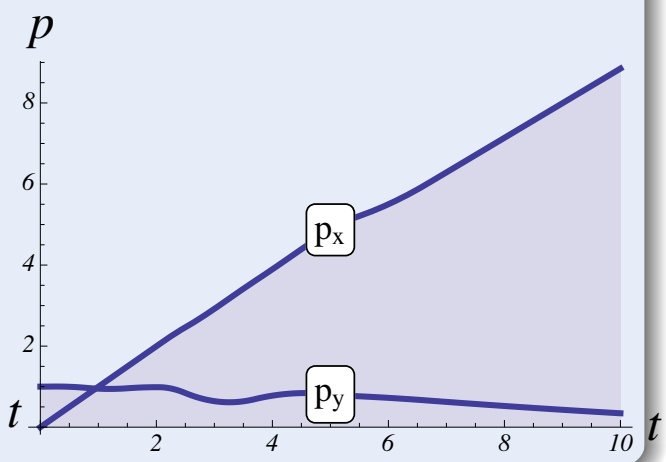
Fixed rule describing state evolution with both

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





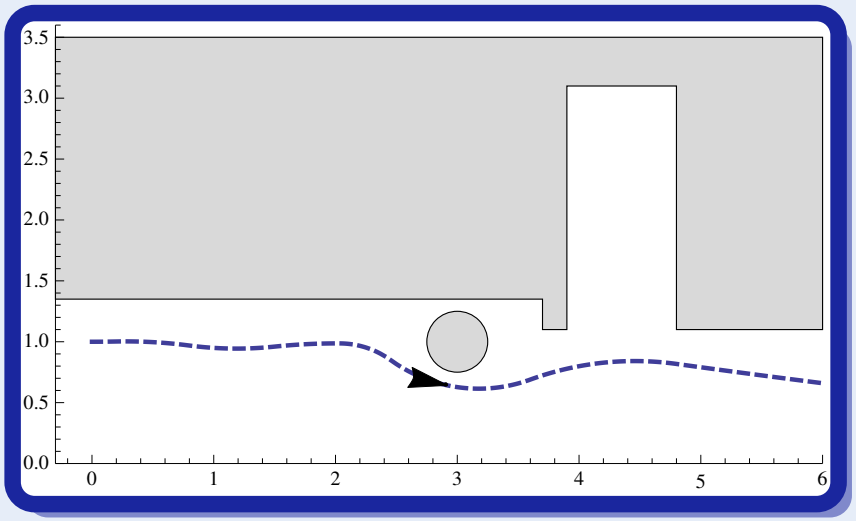


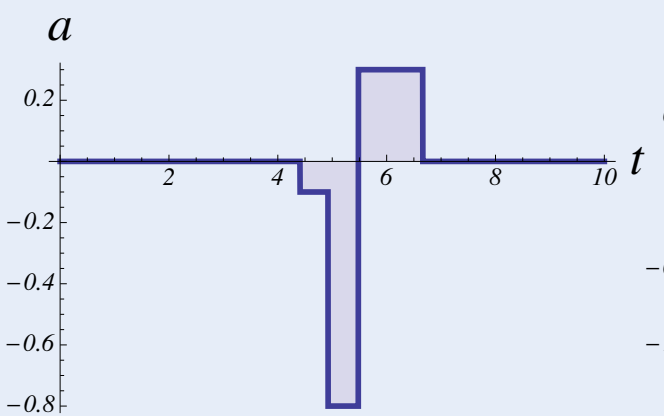


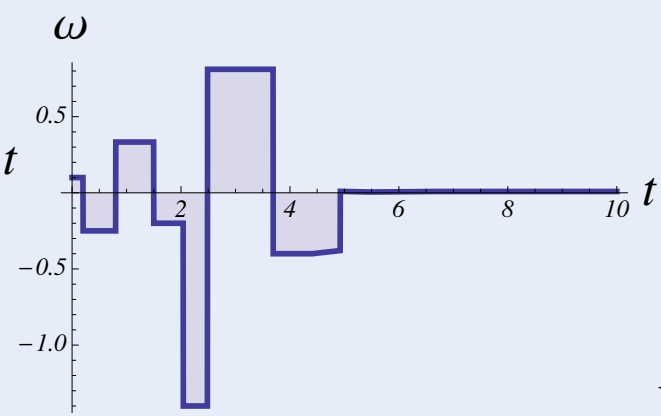
Challenge (Hybrid Systems)

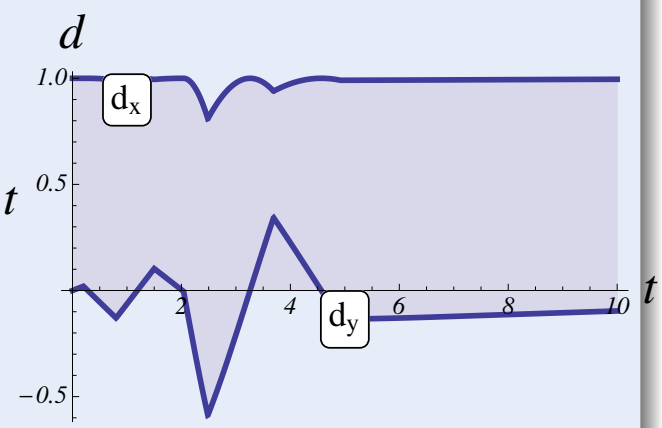
Fixed rule describing state evolution with both

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





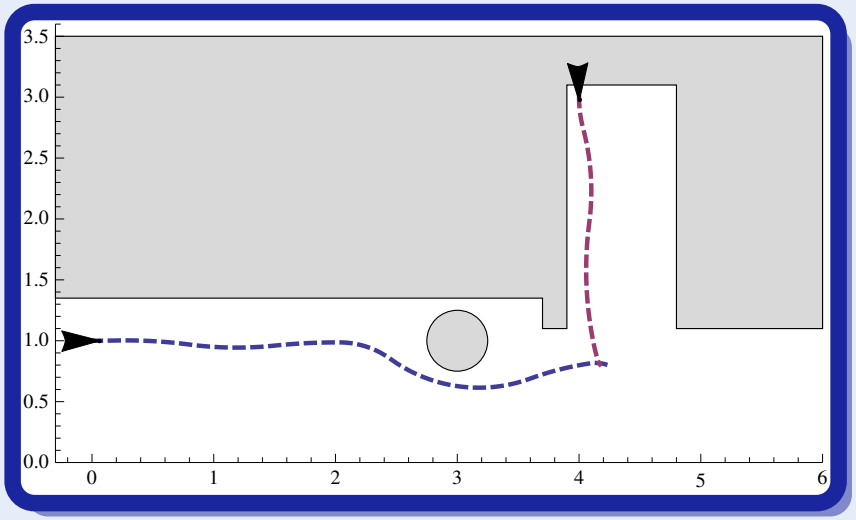


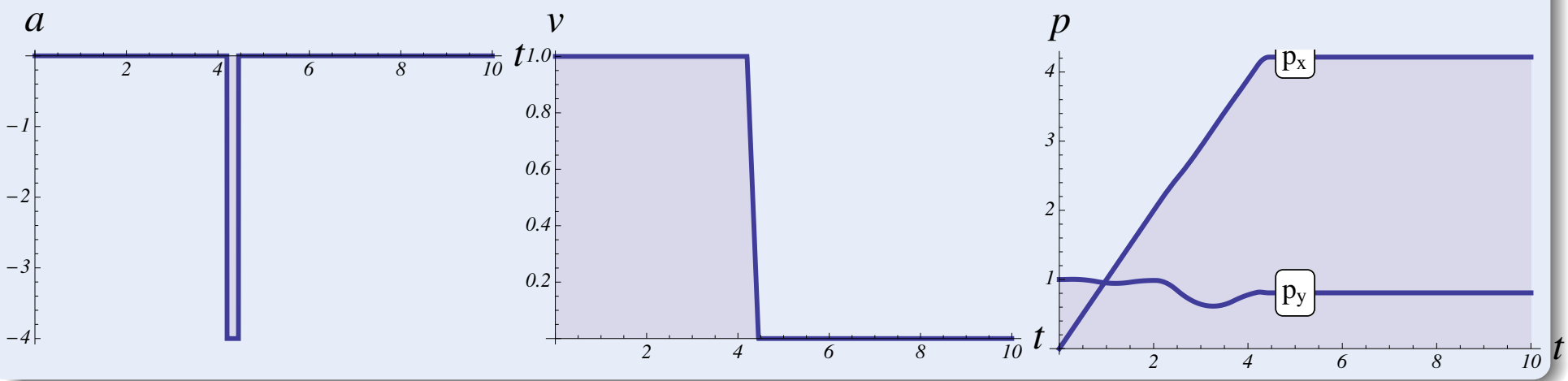


Challenge (Hybrid Systems)

Fixed rule describing state evolution with both

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

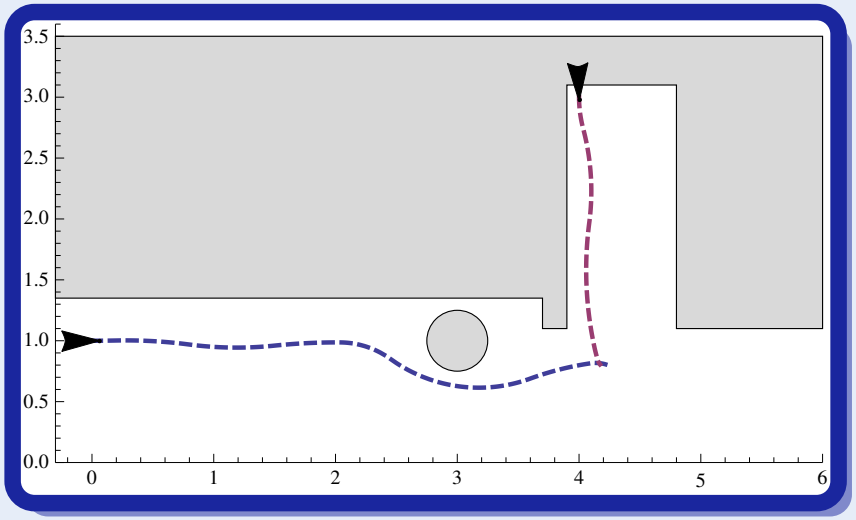


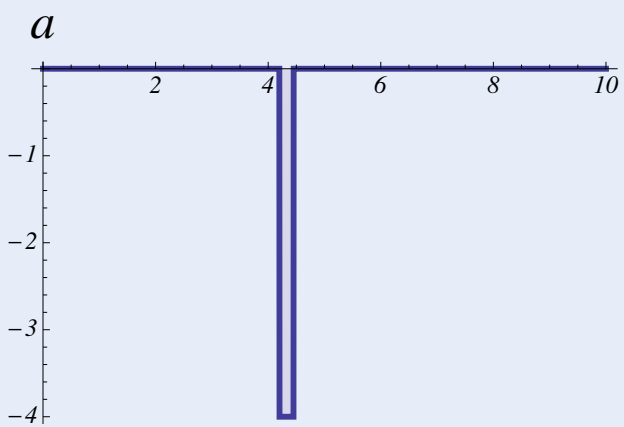


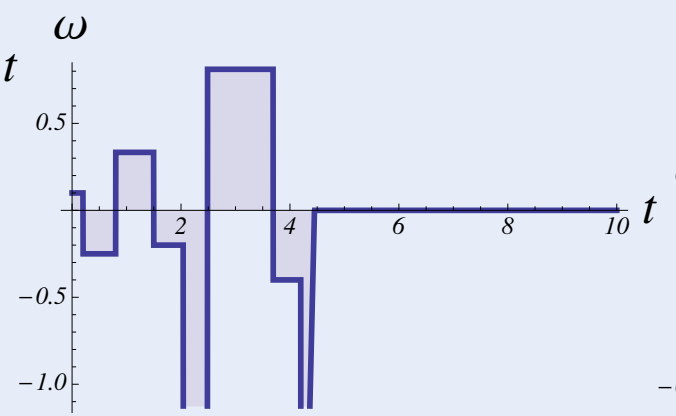
Challenge (Hybrid Systems)

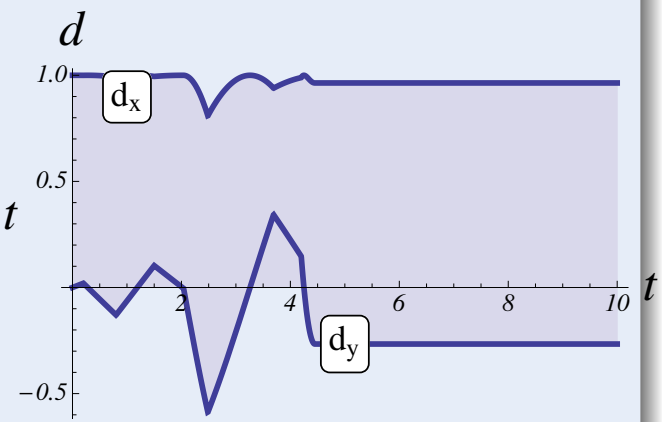
Fixed rule describing state evolution with both

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





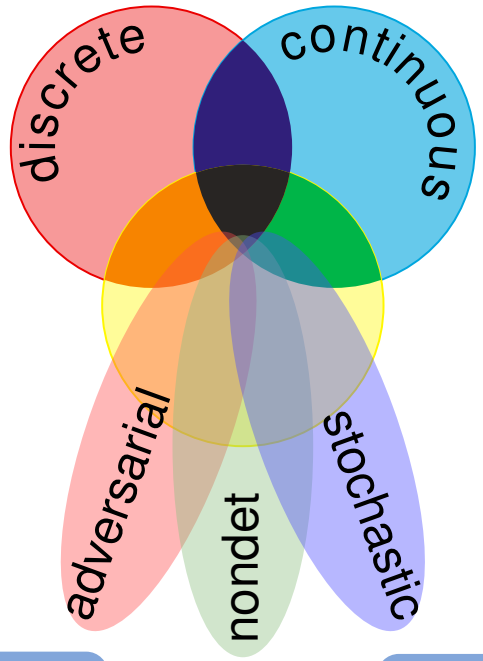




CPSs are Multi-Dynamical Systems

CPS Dynamics

CPS are characterized by multiple facets of dynamical systems.



CPS Compositions

CPS combines multiple simple dynamical effects.

Descriptive simplification

Tame Parts

Exploiting compositionality tames CPS complexity.

Analytic simplification

Learning Objectives

`http://keymaeraX.org/`

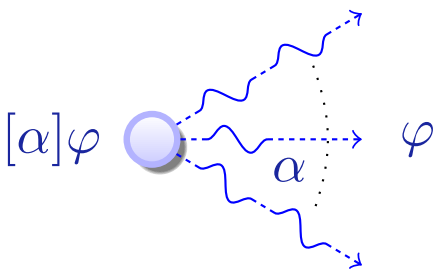
Use KeYmaera X to:

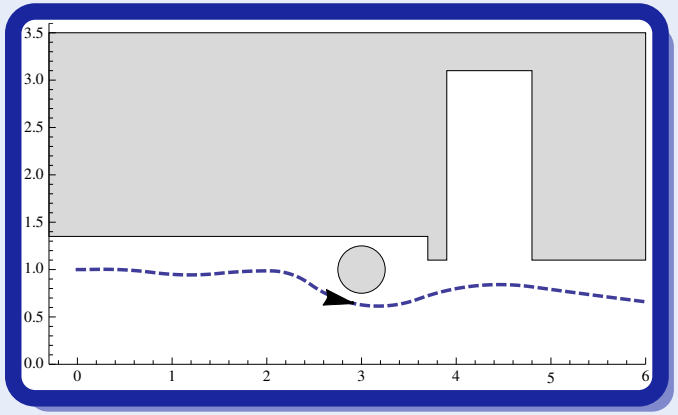
1. Model cyber-physical systems
2. Express safety/correctness properties
3. Find bugs in a system design
4. Identify safety constraints
5. Identify system invariants
6. Verify the final system design
7. Write automated proof tactics
8. Prove differential equations
9. Synthesize correct-by-construction runtime monitors

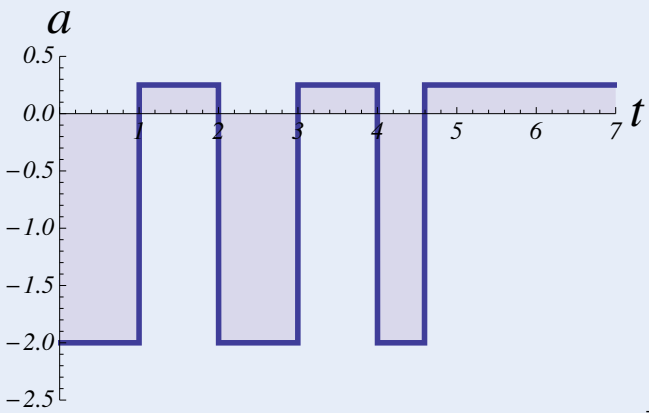
Outline

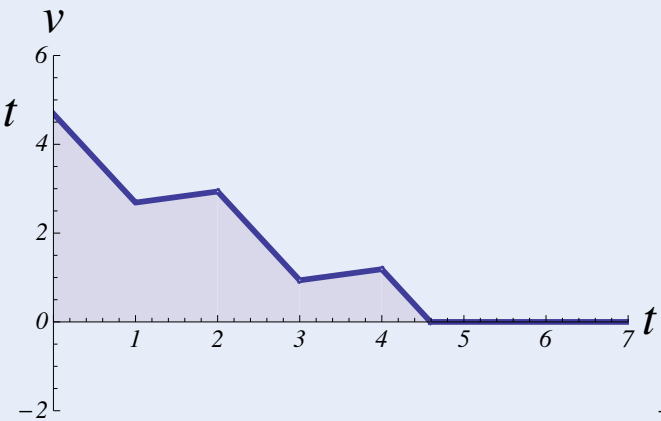
Outline

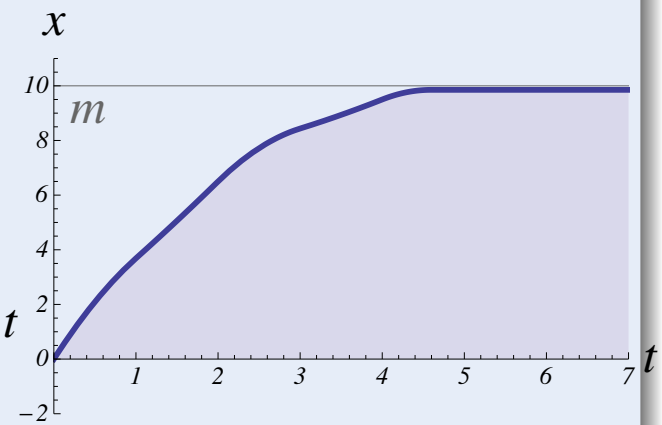
Concept (Differential Dynamic Logic) (JAR'08,LICS'12)



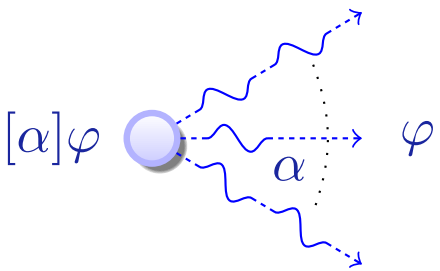


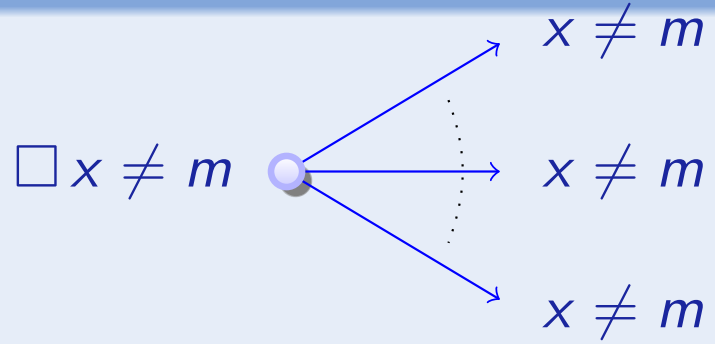


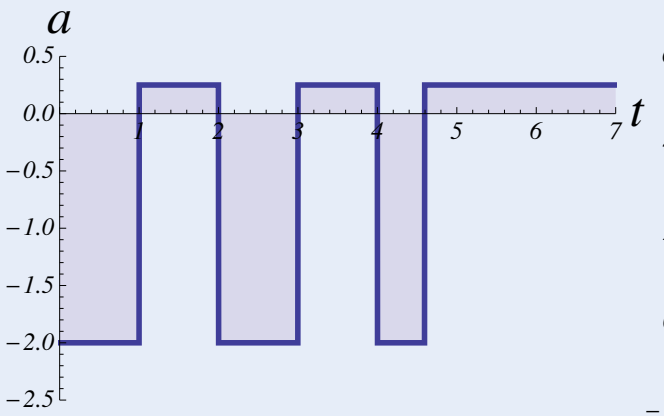


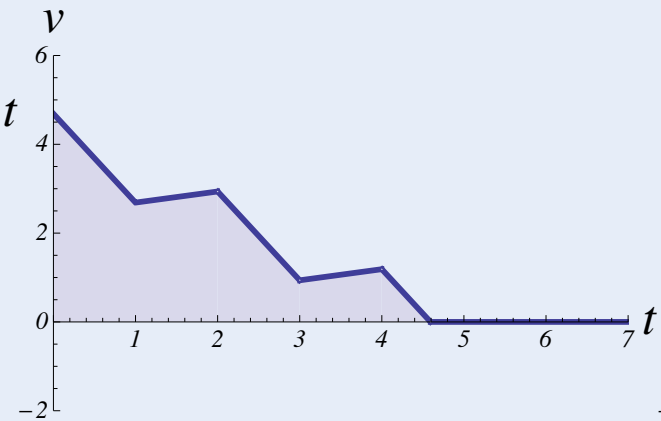


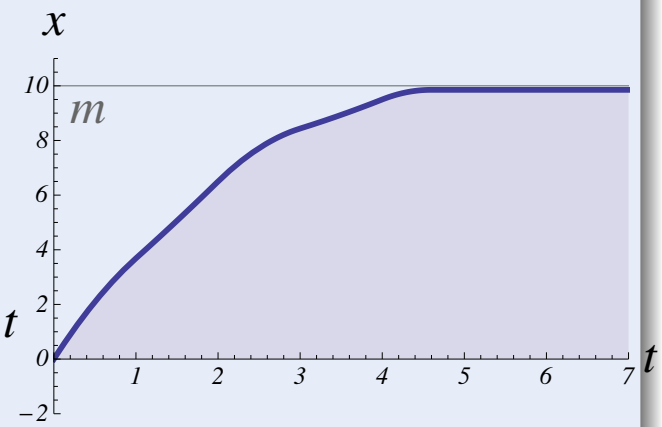
Concept (Differential Dynamic Logic) (JAR'08,LICS'12)



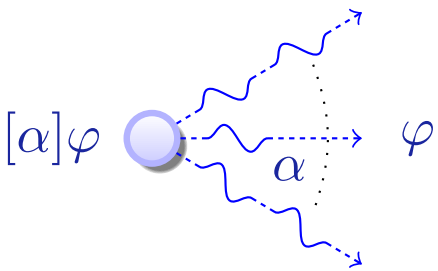


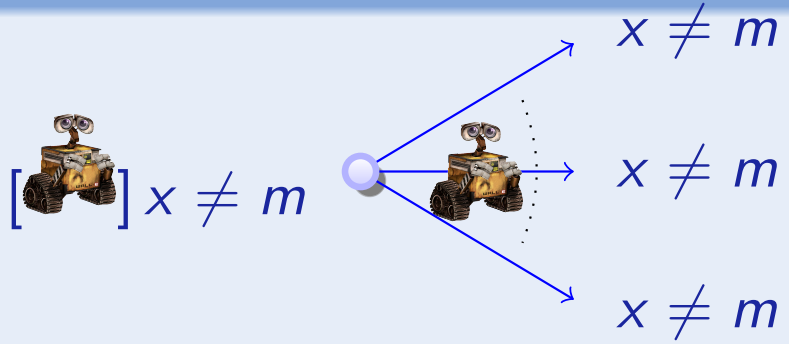


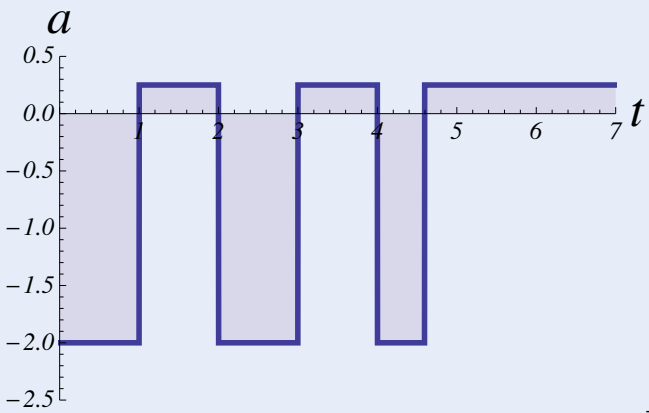


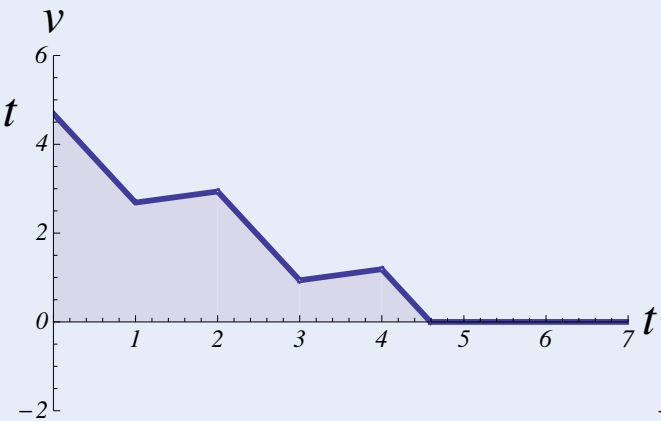


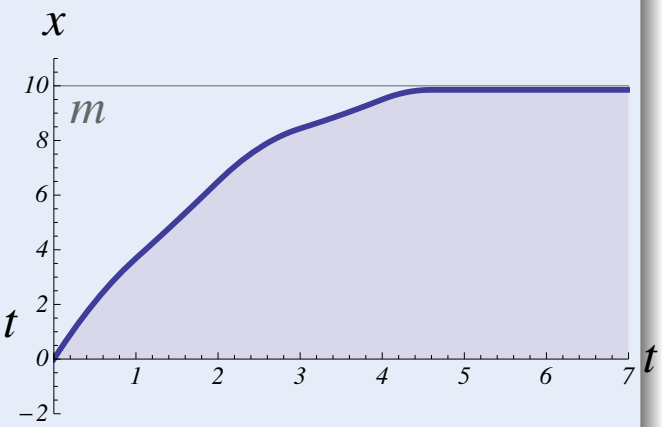
Concept (Differential Dynamic Logic) (JAR'08,LICS'12)



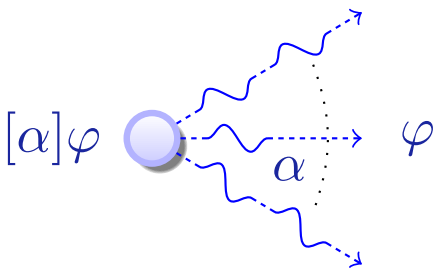


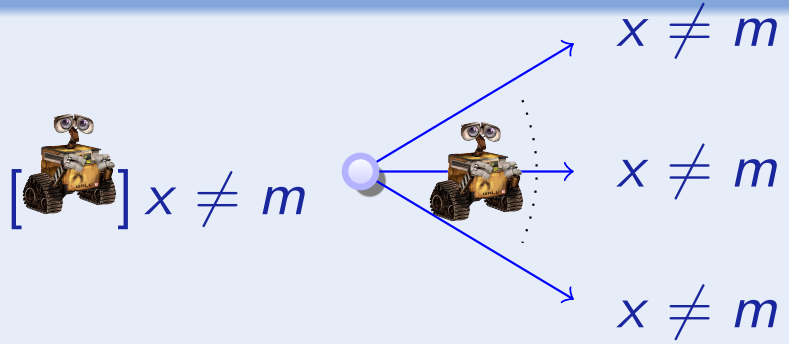


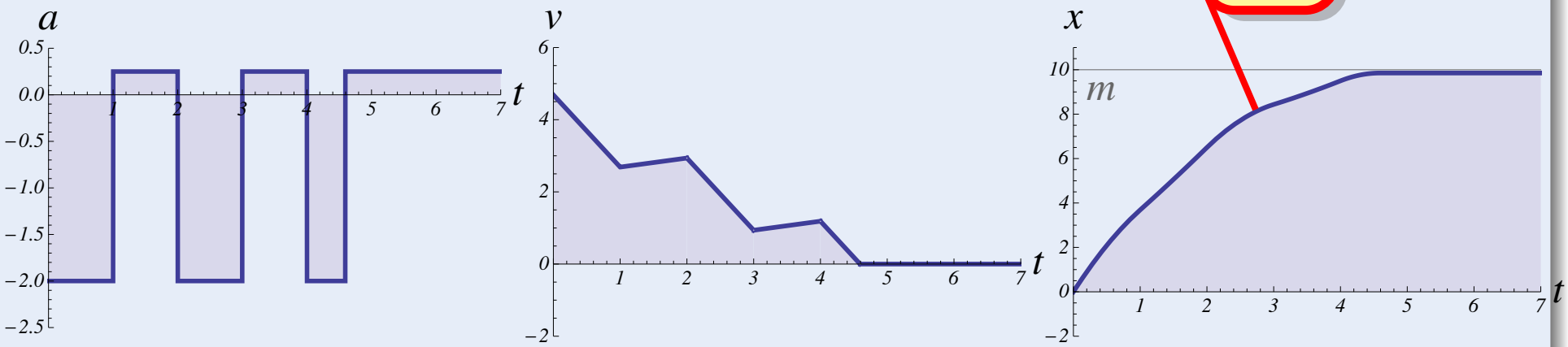




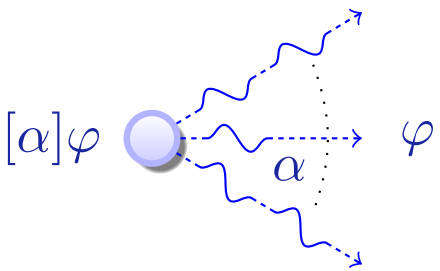
Concept (Differential Dynamic Logic) (JAR'08,LICS'12)

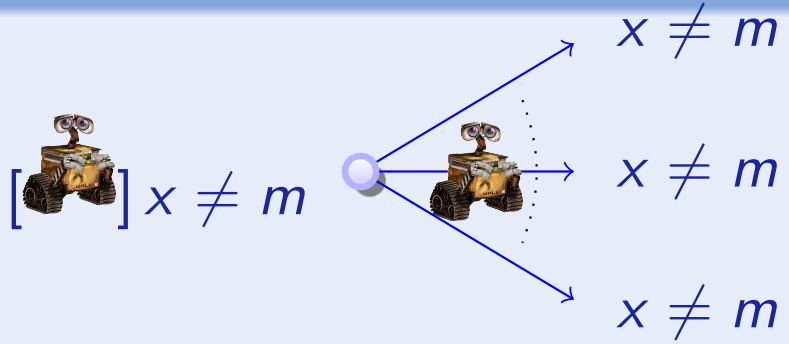


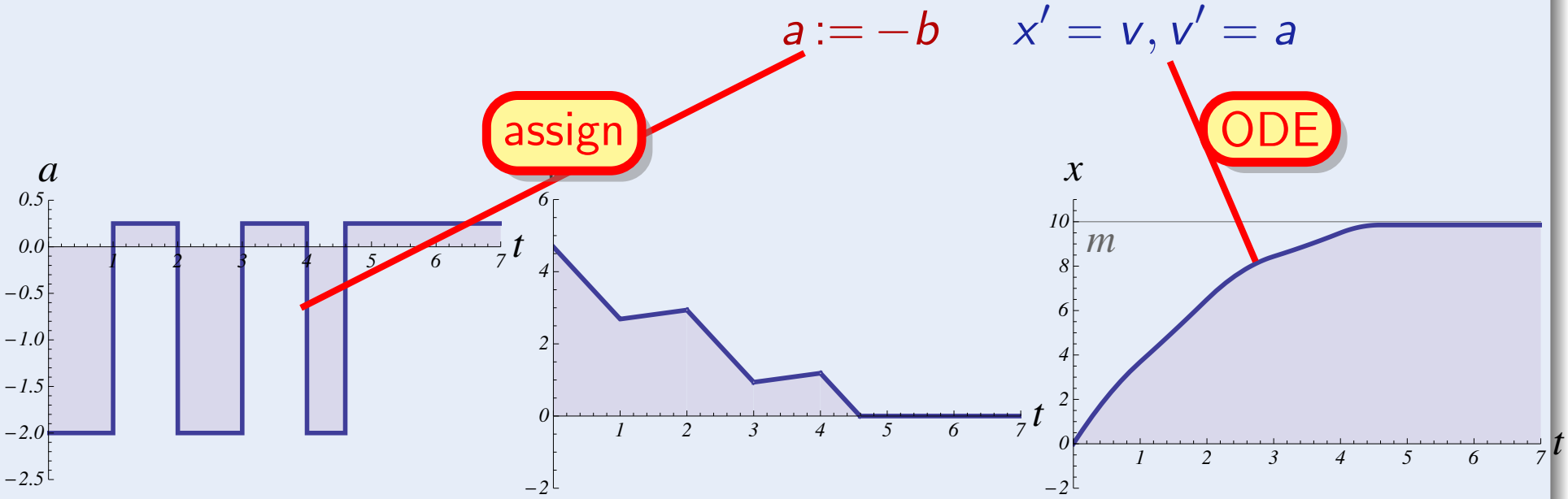




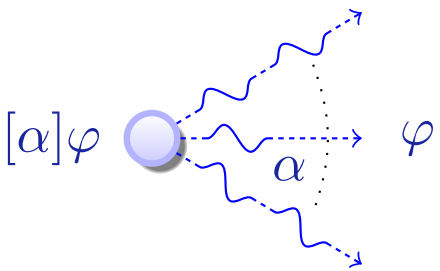
Concept (Differential Dynamic Logic) (JAR'08,LICS'12)

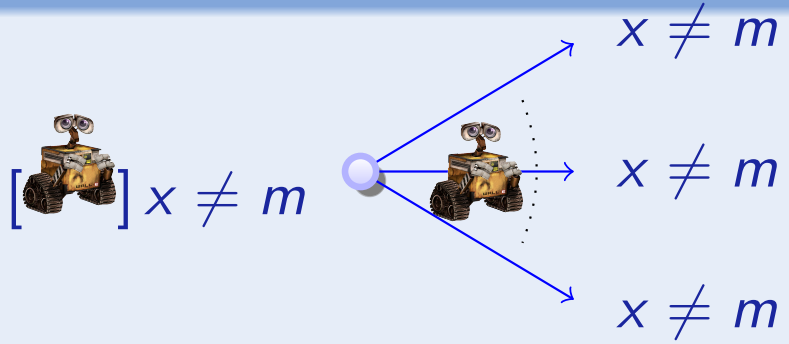






Concept (Differential Dynamic Logic) (JAR'08,LICS'12)



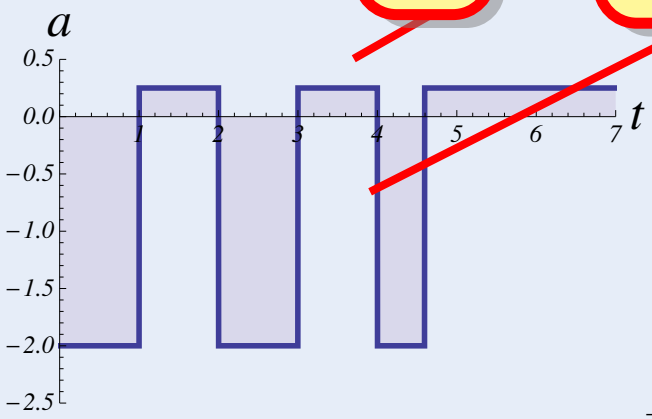


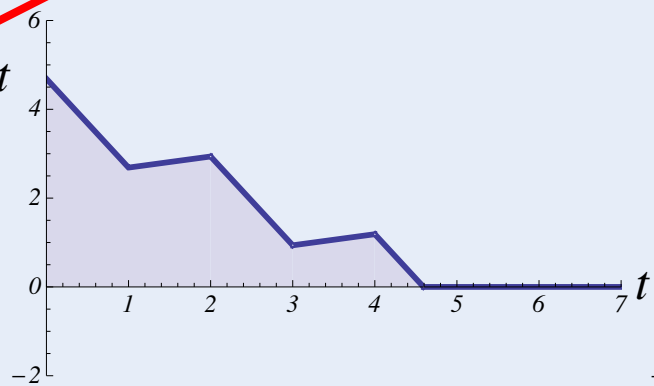
$(\texttt{if}(\mathrm{SB}(x,m))\ a := -b) \quad x' = v, v' = a$

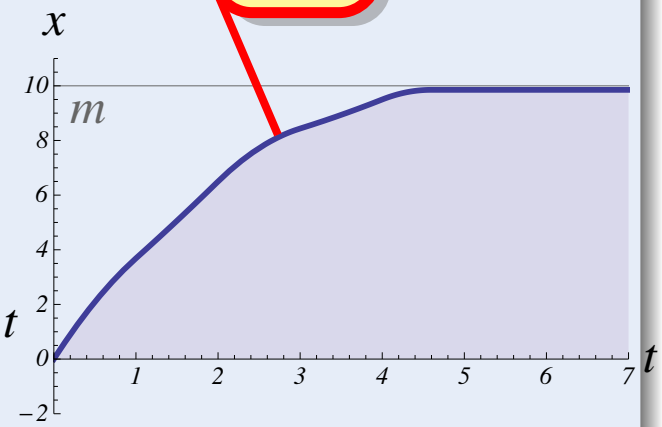
test

assign

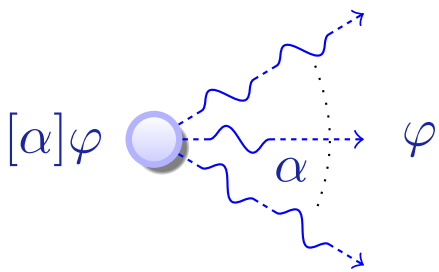
ODE







Concept (Differential Dynamic Logic) (JAR'08,LICS'12)



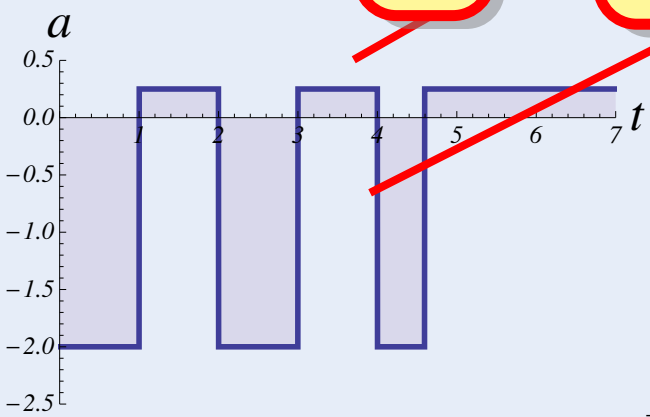
seq. compose

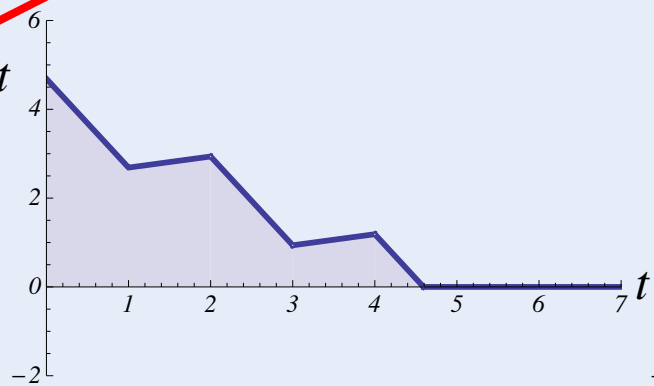
$$(\texttt{if}(\mathrm{SB}(x,m))\ a := -b)\ ;\ x' = v, v' = a$$

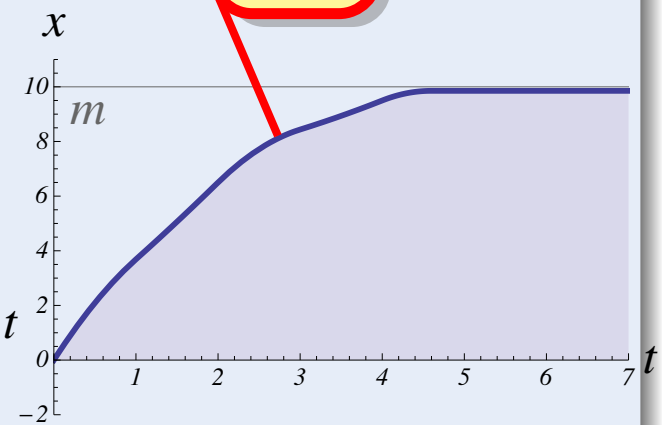
test

assign

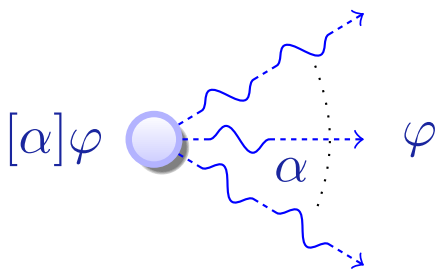
ODE







Concept (Differential Dynamic Logic) (JAR'08,LICS'12)



seq. compose

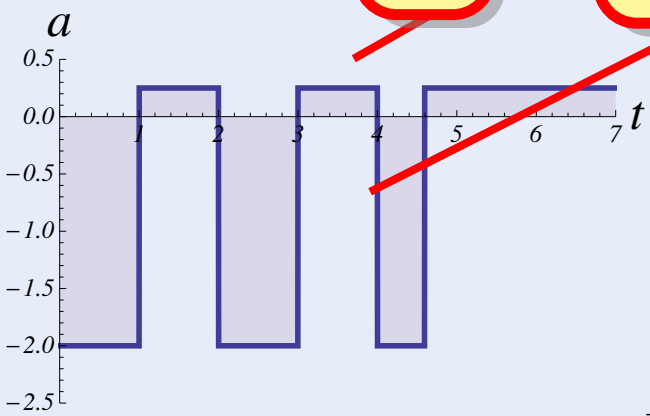
nondet. repeat

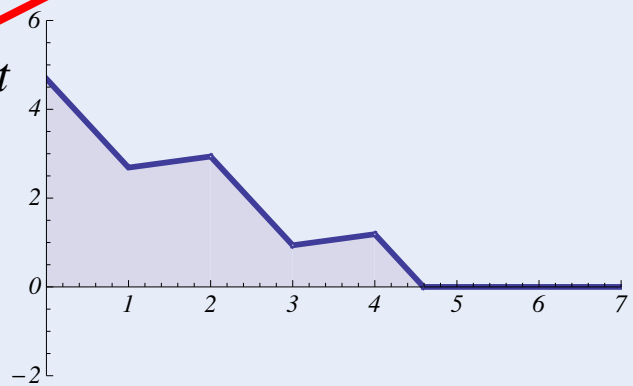
$$\big((\texttt{if}(\mathrm{SB}(x,m))\ a := -b)\ ;\ x' = v, v' = a\big)^*$$

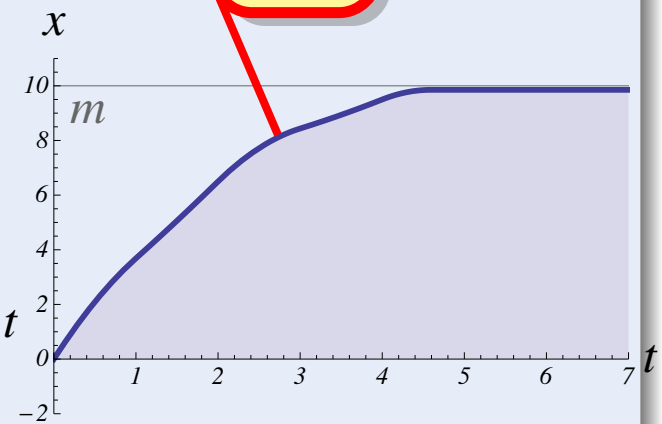
test

assign

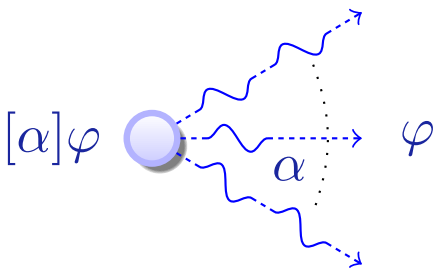
ODE

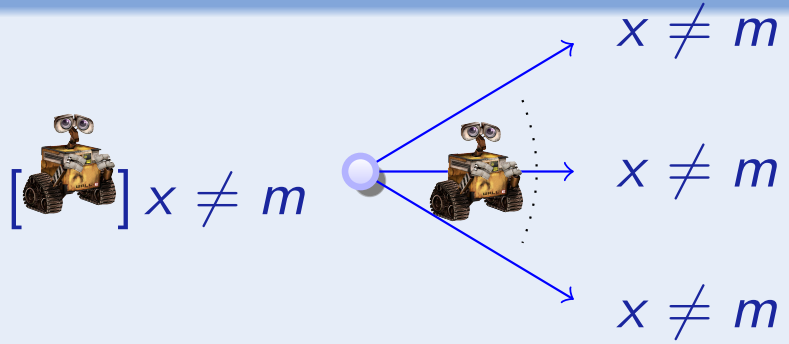






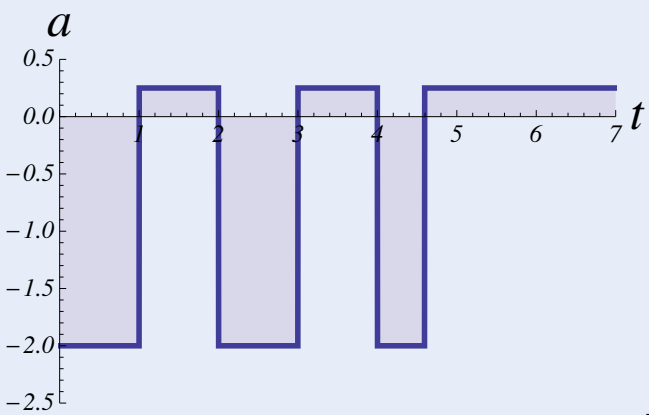
Concept (Differential Dynamic Logic) (JAR'08,LICS'12)

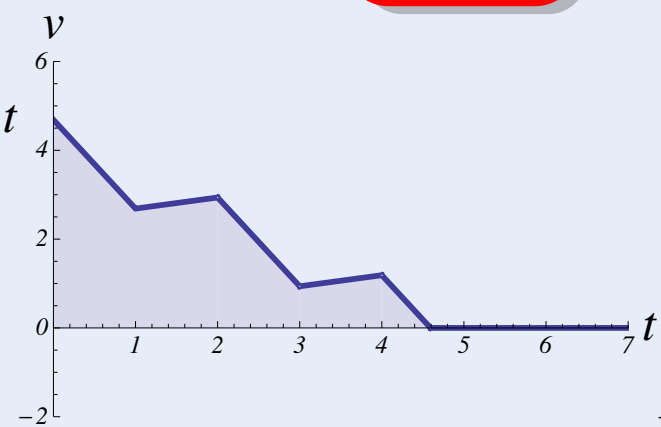


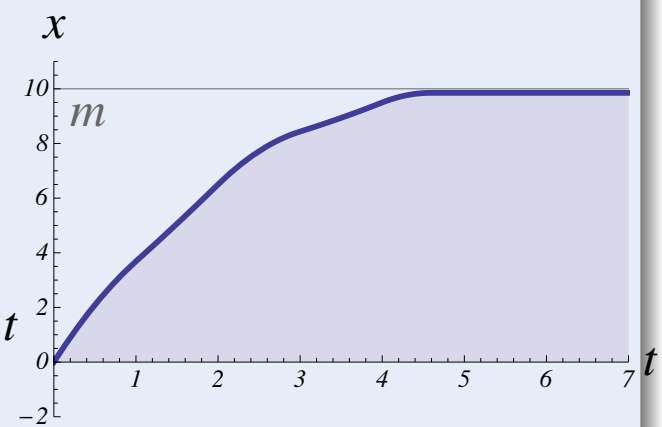


$$\big[\big((\texttt{if}(\mathrm{SB}(x,m))\ a := -b)\ ;\ x' = v, v' = a\big)^*\big]\underbrace{x \neq m}_{\text{post}}$$

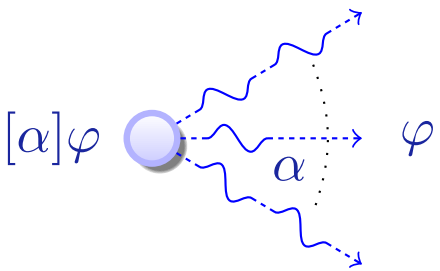
all runs

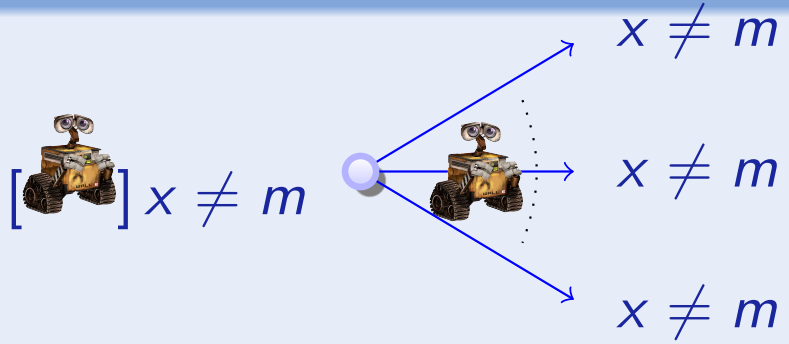




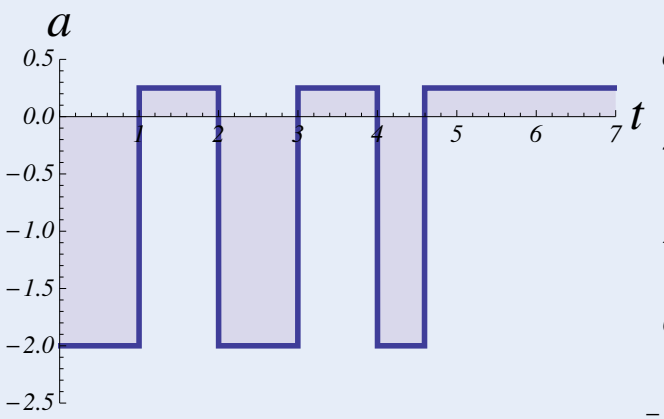


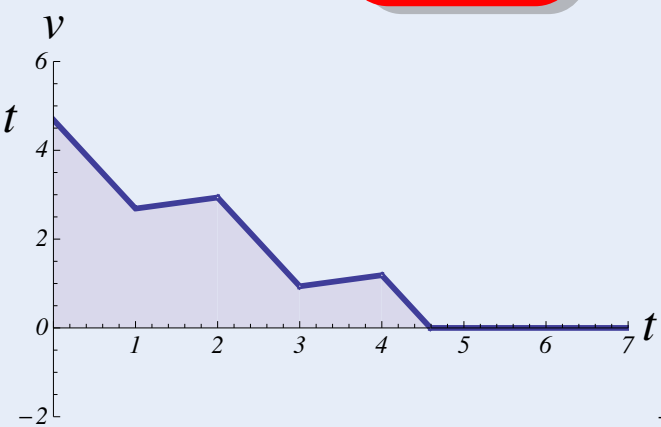
Concept (Differential Dynamic Logic) (JAR'08,LICS'12)

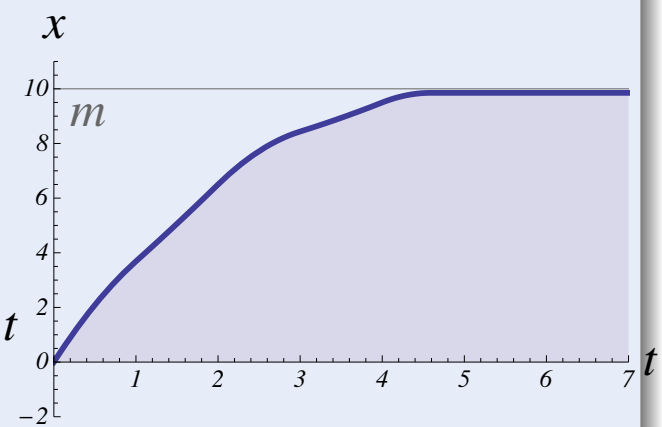




$$\underbrace{x \neq m \wedge b > 0}_{\text{init}} \rightarrow \Big[\underbrace{\big((\texttt{if}(\mathrm{SB}(x,m))\ a := -b)\ ;\ x' = v, v' = a\big)^*}_{\text{all runs}}\Big]\underbrace{x \neq m}_{\text{post}}$$







Differential Dynamic Logic dℒ: Syntax

Definition (Hybrid program α)

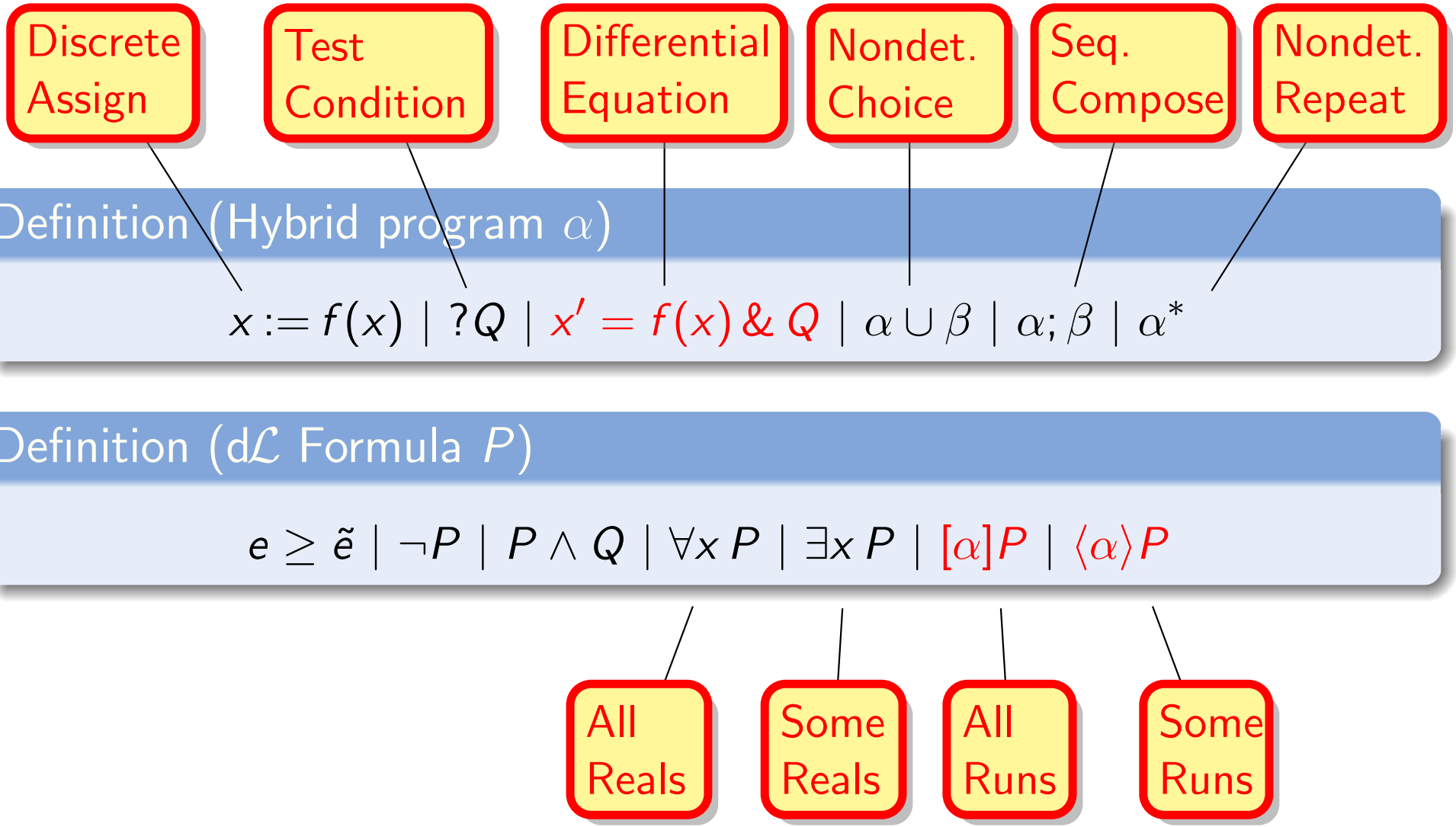
$$x := f(x) \mid ?Q \mid x' = f(x) \,\&\, Q \mid \alpha \cup \beta \mid \alpha;\beta \mid \alpha^*$$

Definition (dℒ Formula P)

$$e \geq \tilde{e} \mid \neg P \mid P \wedge Q \mid \forall x\, P \mid \exists x\, P \mid [\alpha]P \mid \langle\alpha\rangle P$$

Tableaux'07,JAutomReas'08,LICS'12

Differential Dynamic Logic dℒ: Syntax



Tableaux'07,JAutomReas'08,LICS'12

Differential Dynamic Logic dℒ: Semantics

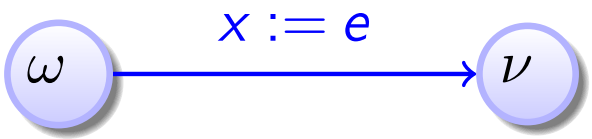
Definition (Hybrid program semantics) $([\![\cdot]\!] : \text{HP} \to \wp(\mathcal{S} \times \mathcal{S}))$

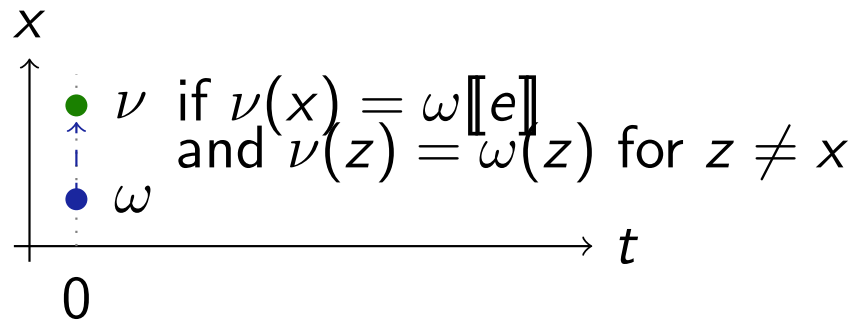
$$
\begin{aligned}
[\![x := e]\!] &= \{(\omega, \nu) \ : \ \nu = \omega \text{ except } \nu[\![x]\!] = \omega[\![e]\!]\} \\
[\![?Q]\!] &= \{(\omega, \omega) \ : \ \omega \in [\![Q]\!]\} \\
[\![x' = f(x)]\!] &= \{(\varphi(0), \varphi(r)) \ : \ \varphi \models x' = f(x) \text{ for some duration } r\} \\
[\![\alpha \cup \beta]\!] &= [\![\alpha]\!] \cup [\![\beta]\!] \\
[\![\alpha; \beta]\!] &= [\![\alpha]\!] \circ [\![\beta]\!] \\
[\![\alpha^*]\!] &= \bigcup_{n \in \mathbb{N}} [\![\alpha^n]\!]
\end{aligned}
$$

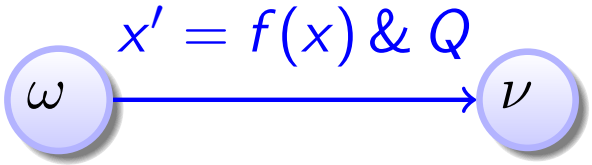
Definition (dℒ semantics) $([\![\cdot]\!] : \text{Fml} \to \wp(\mathcal{S}))$

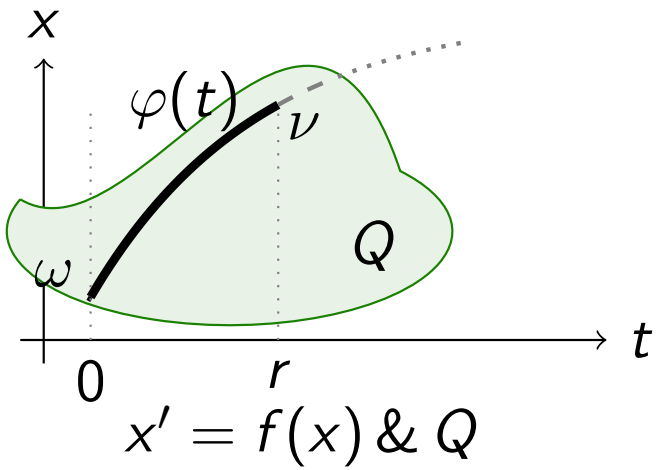
$$
\begin{aligned}
[\![e \geq \tilde{e}]\!] &= \{\omega \ : \ \omega[\![e]\!] \geq \omega[\![\tilde{e}]\!]\} \\
[\![\neg P]\!] &= [\![P]\!]^{\complement} \\
[\![P \wedge Q]\!] &= [\![P]\!] \cap [\![Q]\!] \\
[\![\langle \alpha \rangle P]\!] &= [\![\alpha]\!] \circ [\![P]\!] = \{\omega \ : \ \nu \in [\![P]\!] \text{ for some } \nu : \ (\omega, \nu) \in [\![\alpha]\!]\} \\
[\![[\alpha] P]\!] &= [\![\neg \langle \alpha \rangle \neg P]\!] = \{\omega \ : \ \nu \in [\![P]\!] \text{ for all } \nu : \ (\omega, \nu) \in [\![\alpha]\!]\} \\
[\![\exists x \, P]\!] &= \{\omega \ : \ \omega^r_x \in [\![P]\!] \text{ for some } r \in \mathbb{R}\}
\end{aligned}
$$

Differential Dynamic Logic dℒ: Semantics

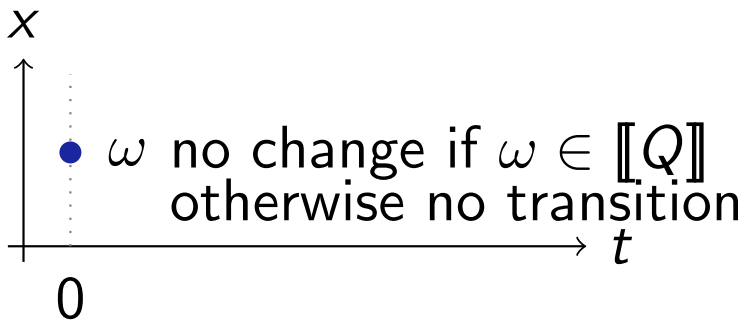




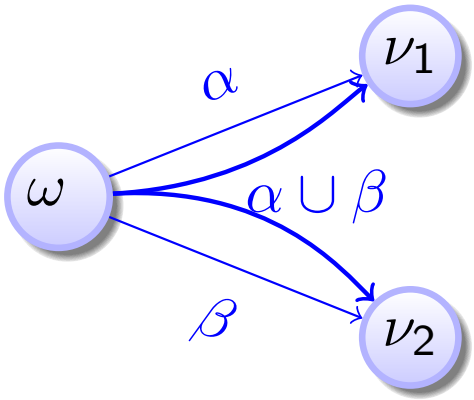


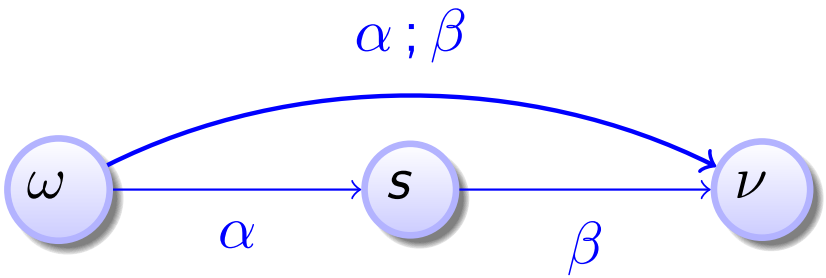


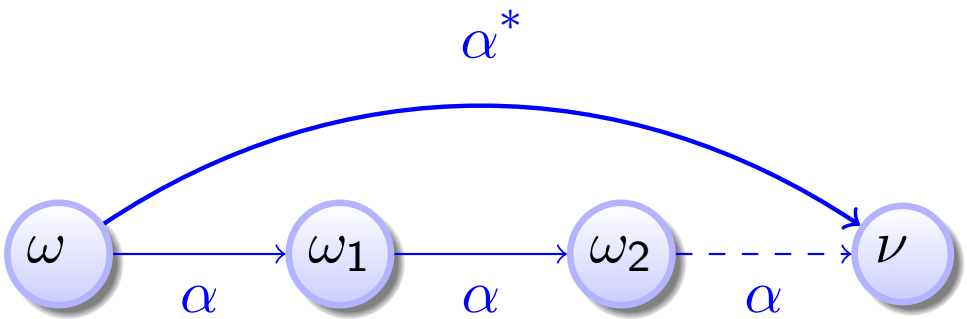
if $\omega \in [\![Q]\!]$

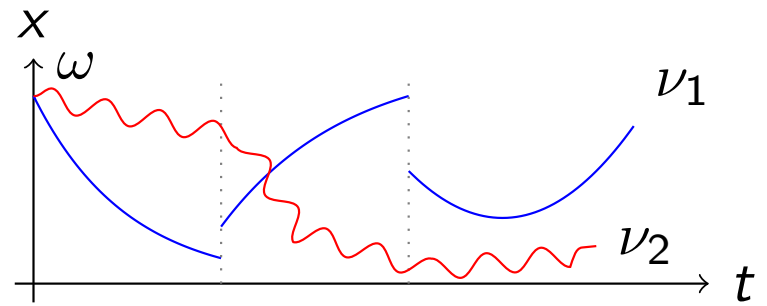


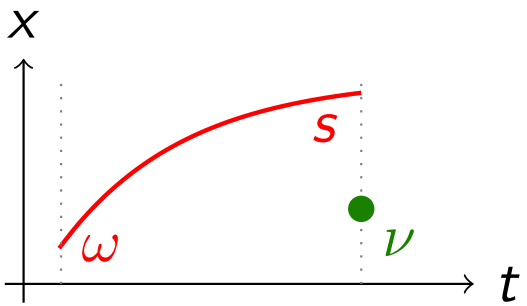
Differential Dynamic Logic dℒ: Semantics

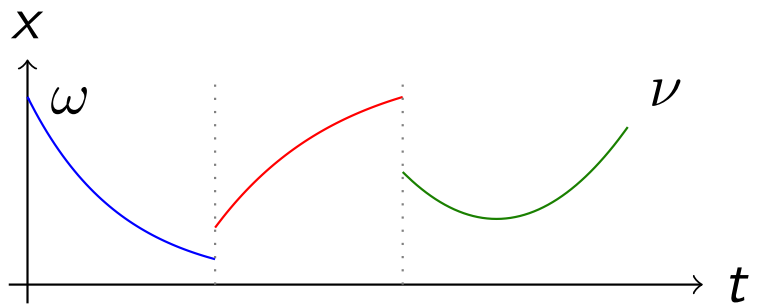




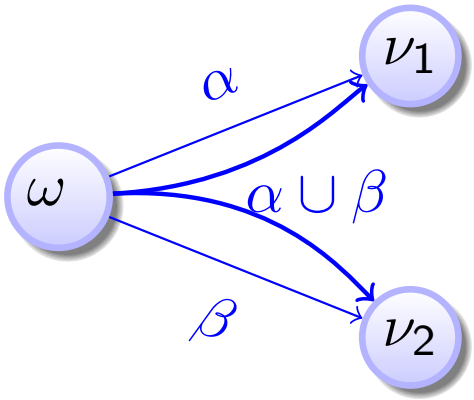


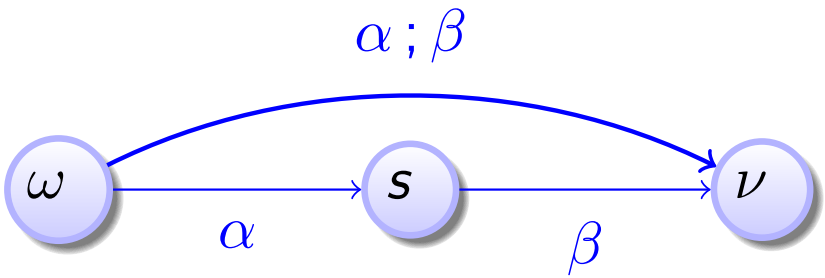


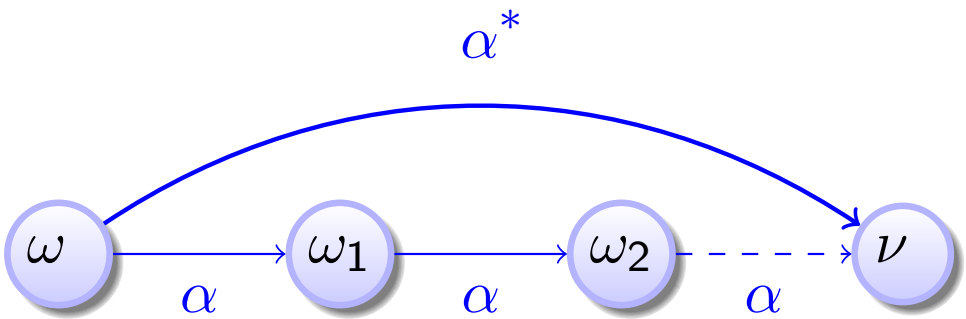


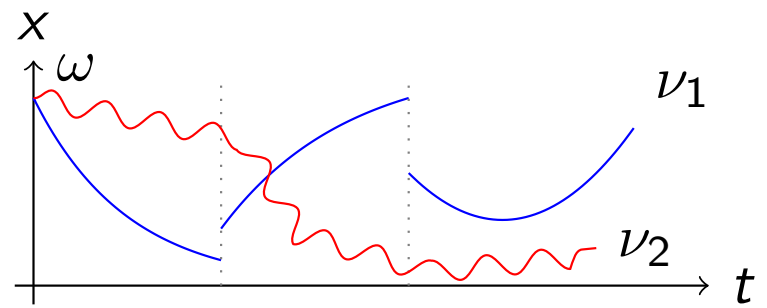


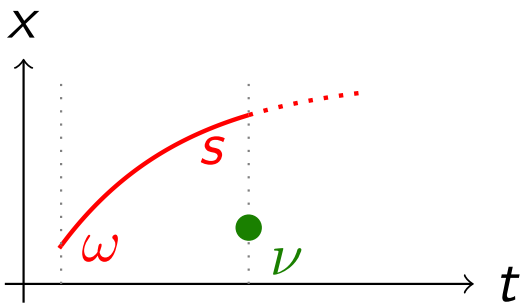
Differential Dynamic Logic dℒ: Semantics

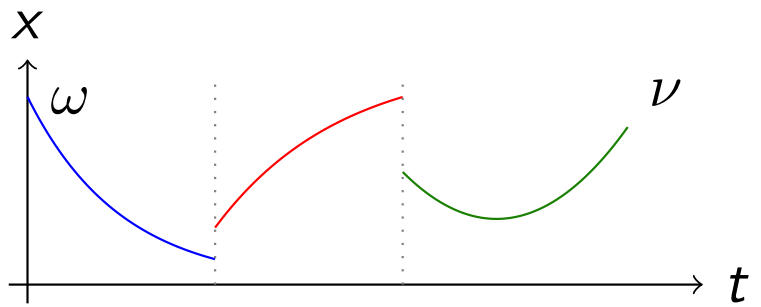




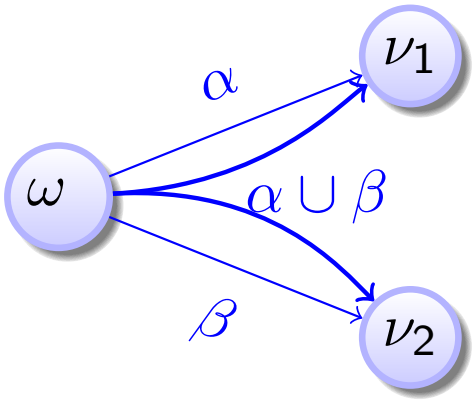


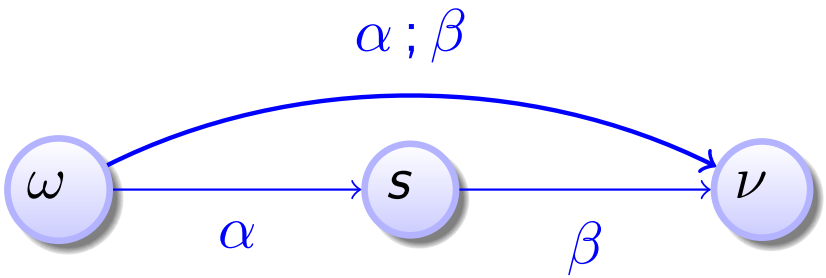


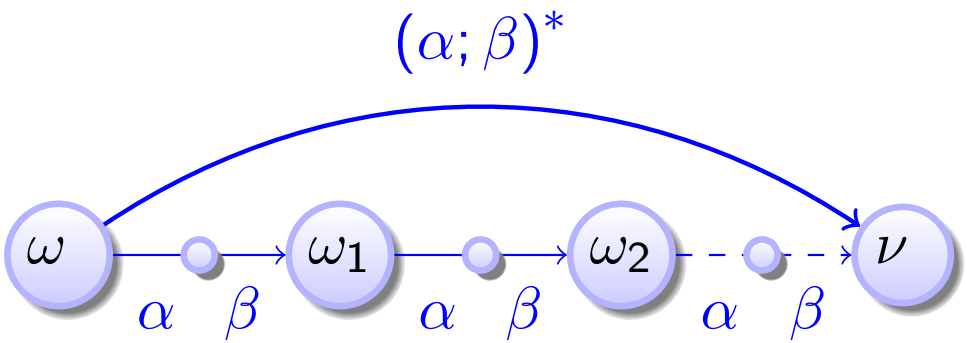


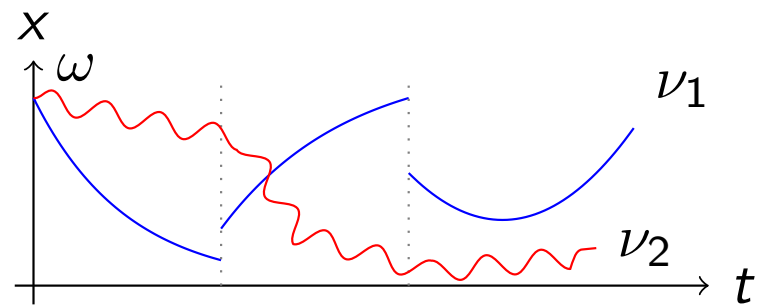


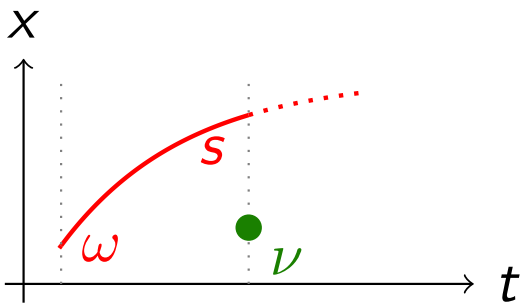
Differential Dynamic Logic dℒ: Semantics

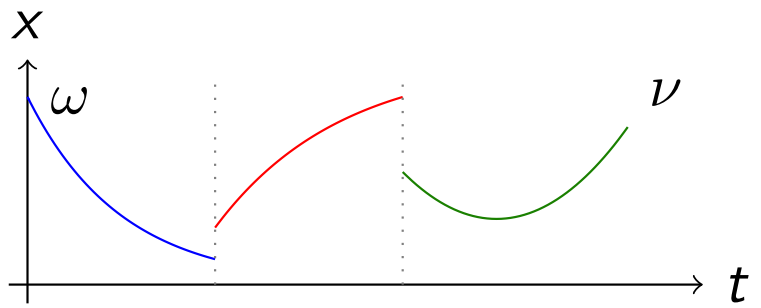




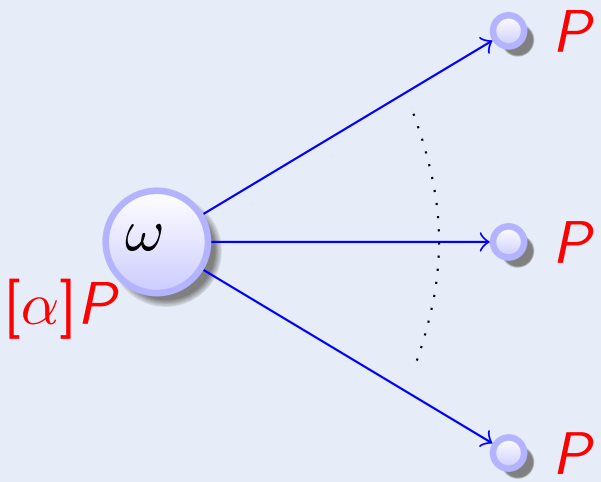




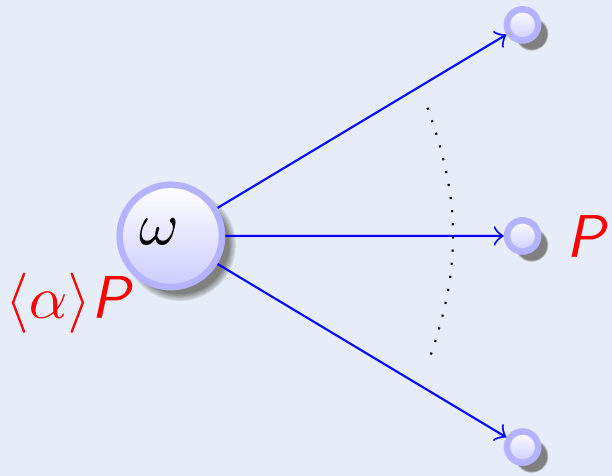




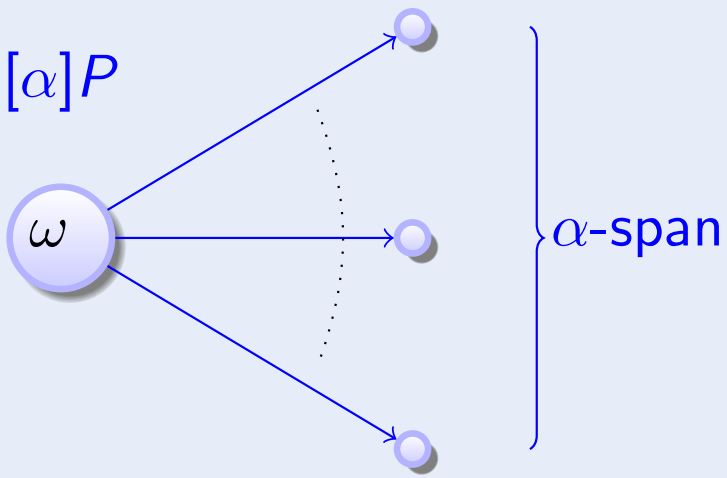
Definition (dℒ Formulas)



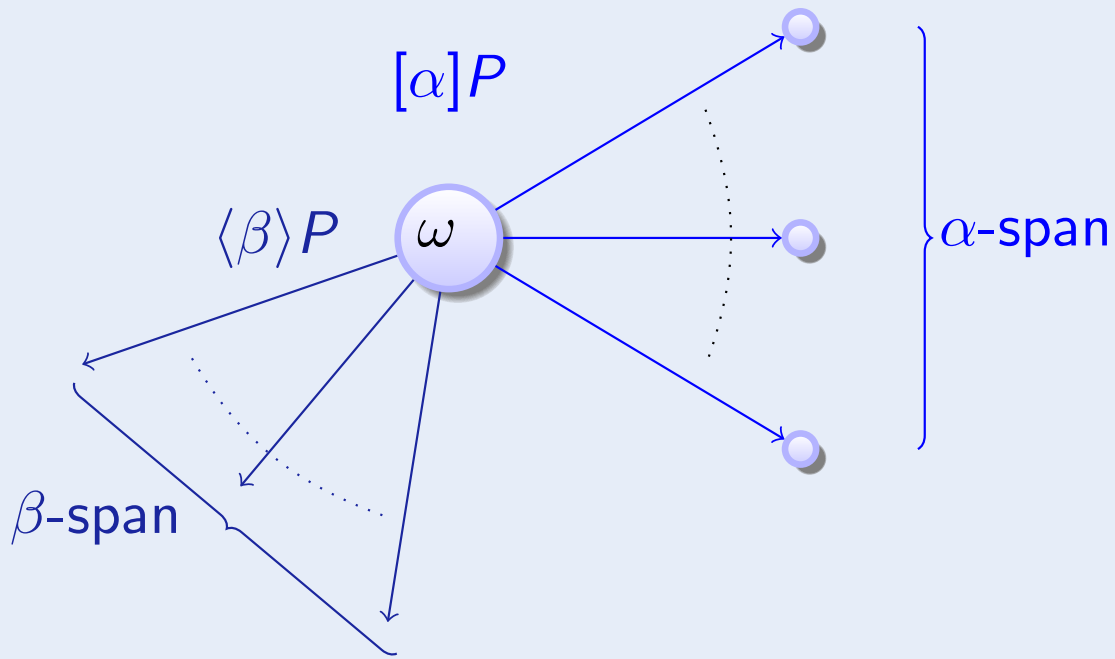
Definition (dℒ Formulas)



Definition (dℒ Formulas)

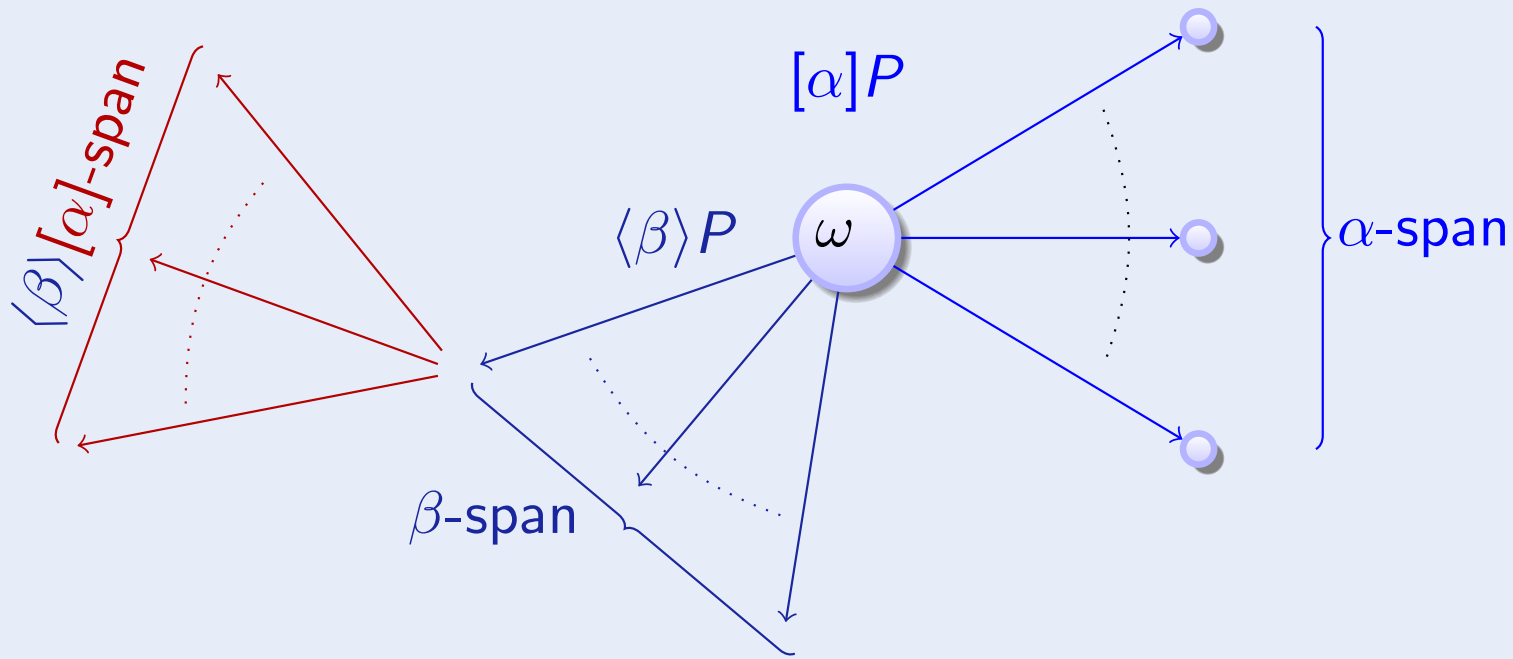


Definition (dℒ Formulas)



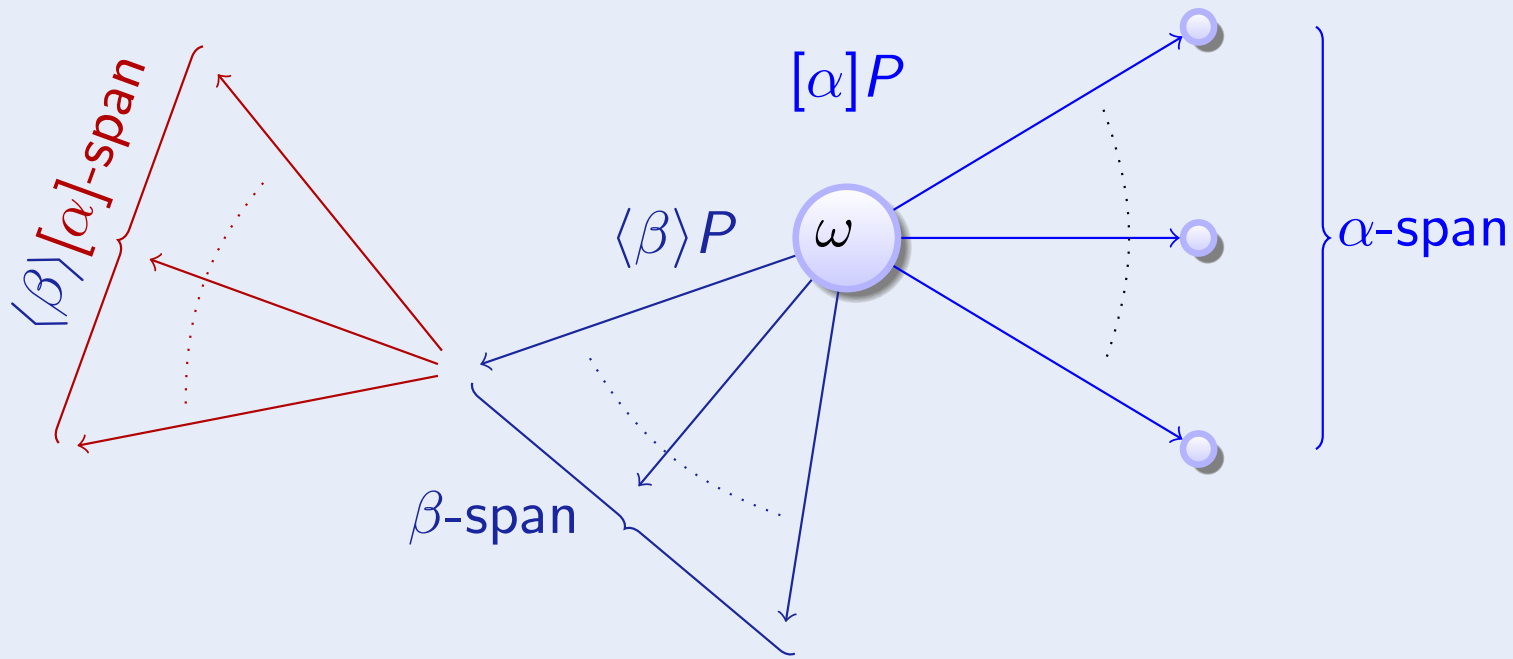
Differential Dynamic Logic dℒ: Semantics

Definition (dℒ Formulas)



Differential Dynamic Logic dℒ: Semantics

Definition (dℒ Formulas)

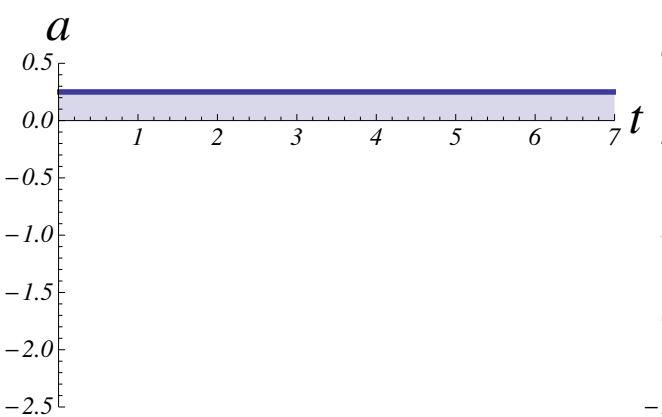


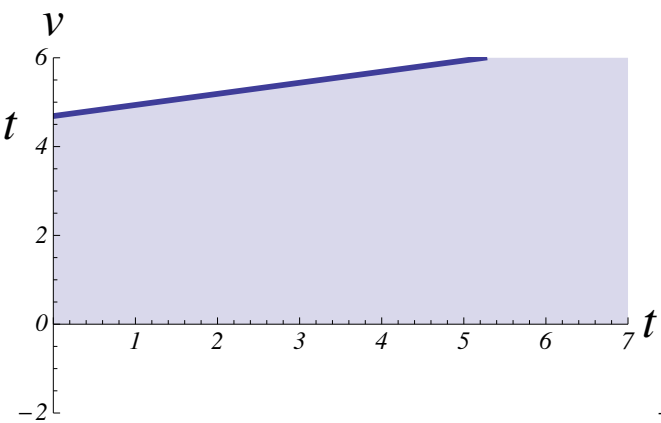
compositional semantics $\Rightarrow$ compositional proofs!

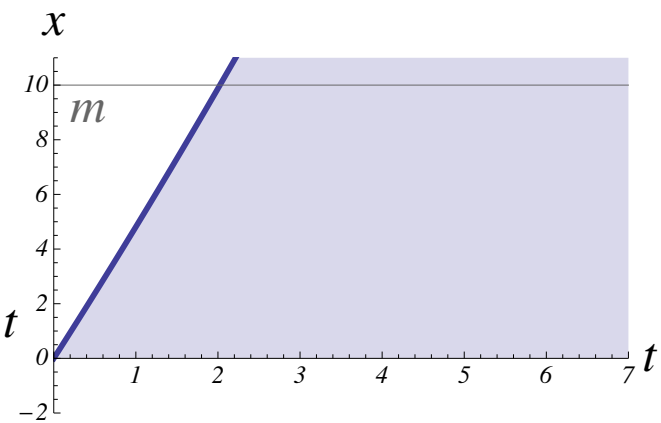
Ex: Car Control

Example (Single car car_s)

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





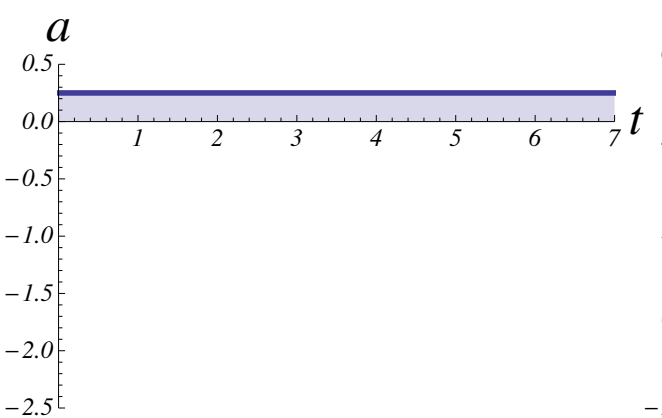


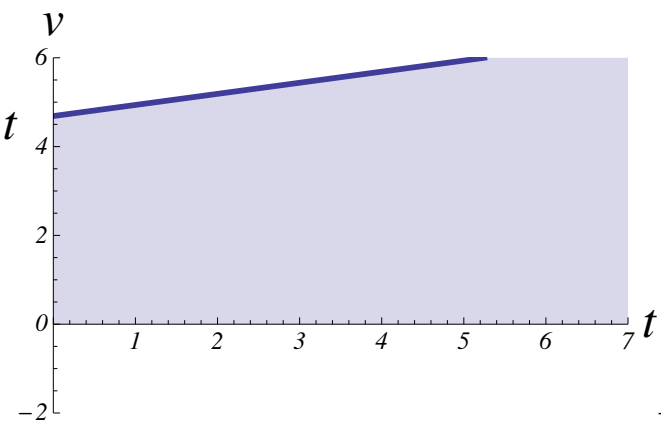
Ex: Car Control

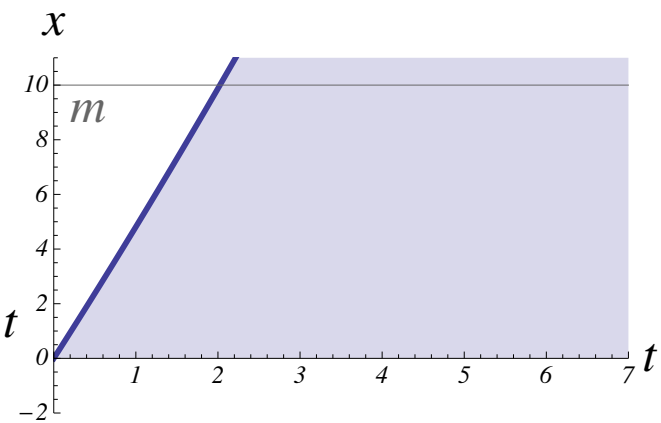
Control decision: accelerate or brake

Example (Single car car_s)

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





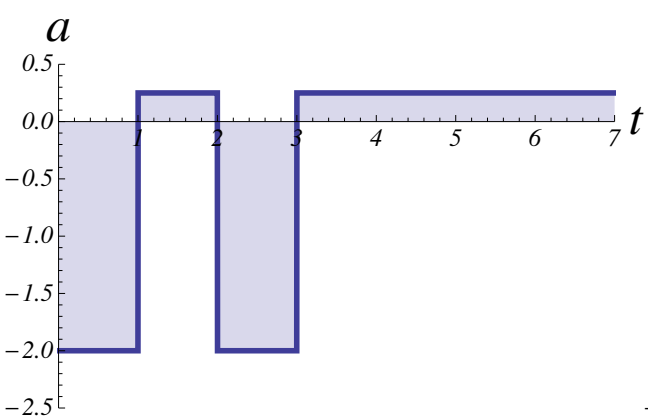


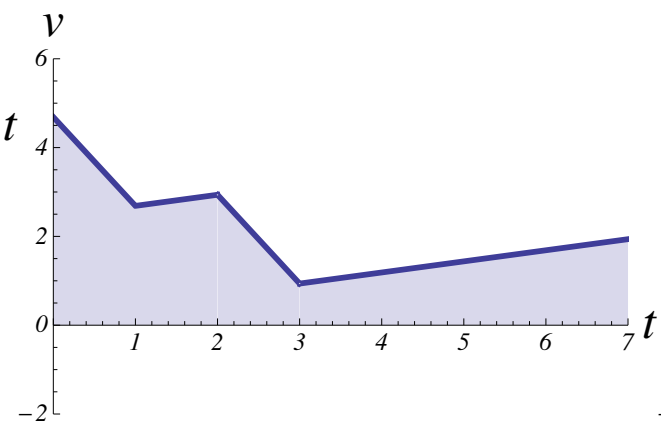
Ex: Car Control

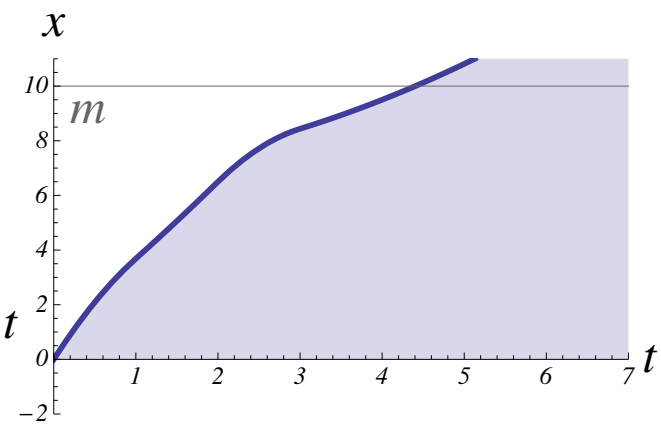
Repeat control decisions

Example (Single car car_s)

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





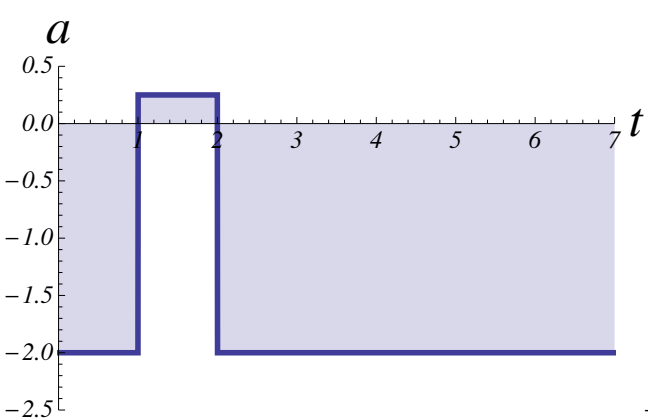


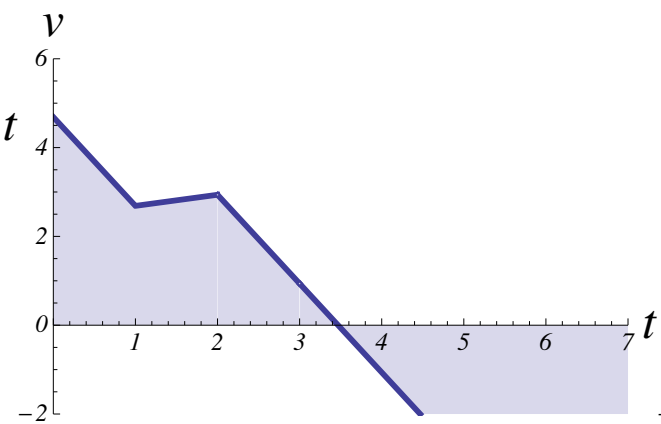
Ex: Car Control

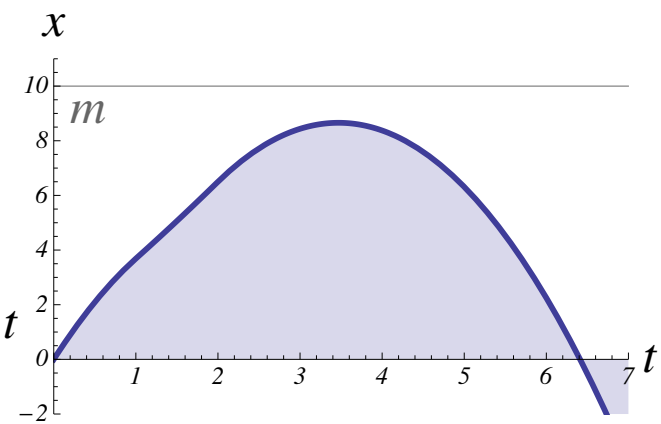
Repeat control decisions

Example (Single car car_s)

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





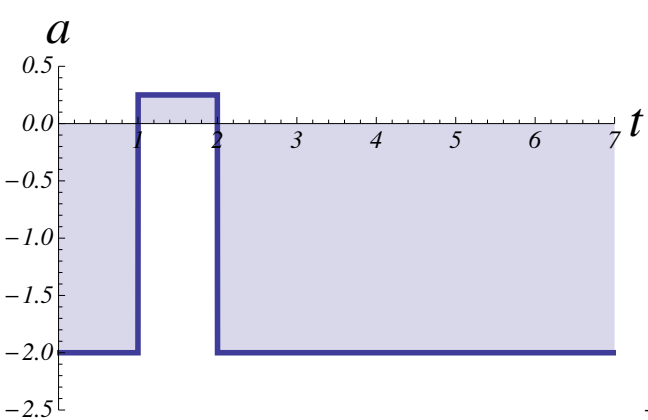


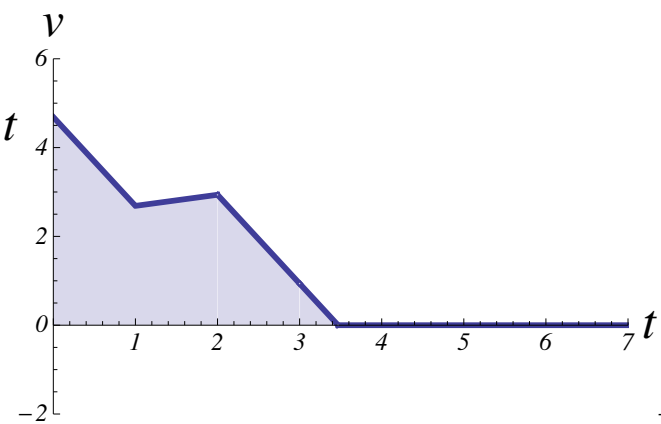
Ex: Car Control

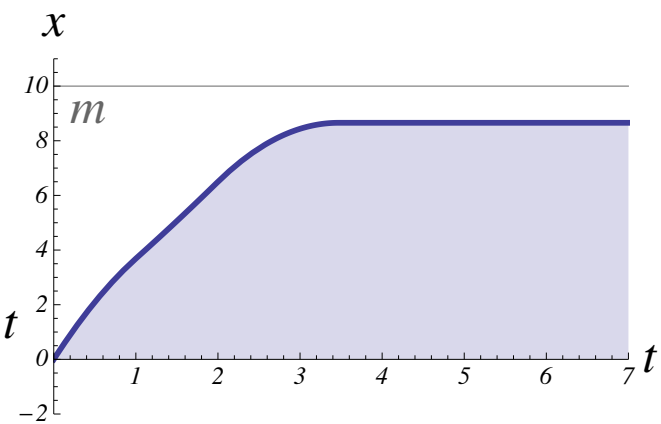
Velocity bound $v \geq 0$

Example (Single car car_s)

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





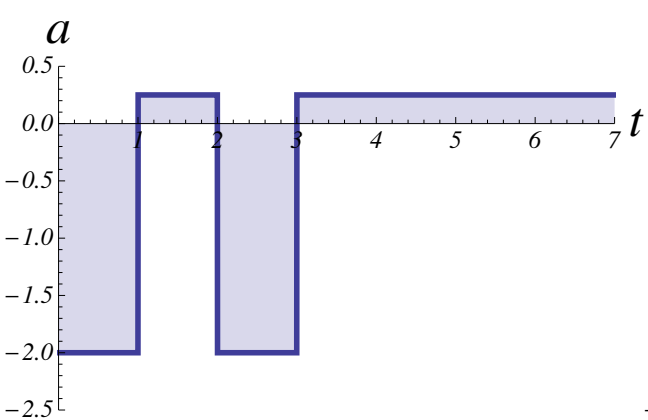


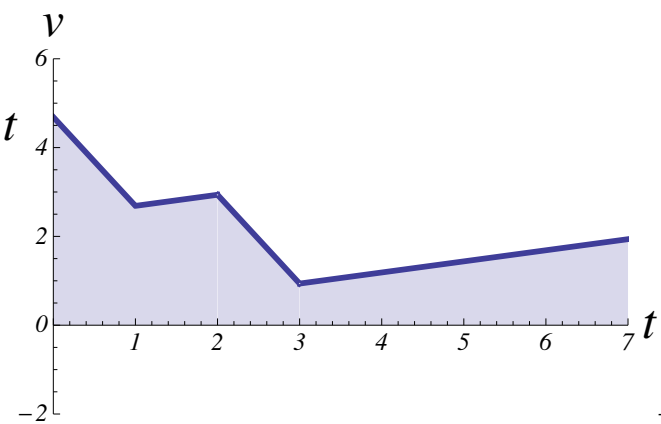
Ex: Car Control

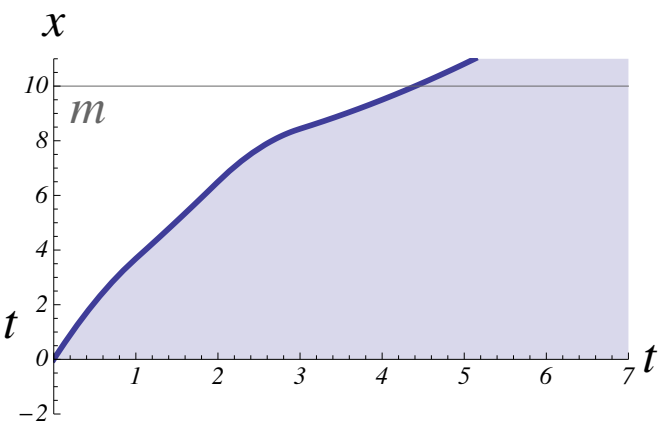
Accelerate not always safe

Example (Single car car_s)

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





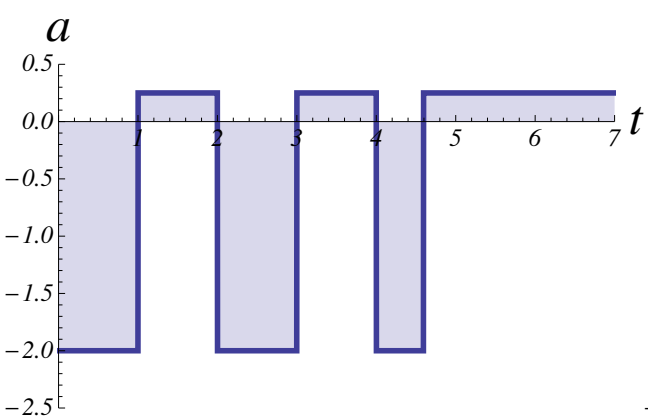


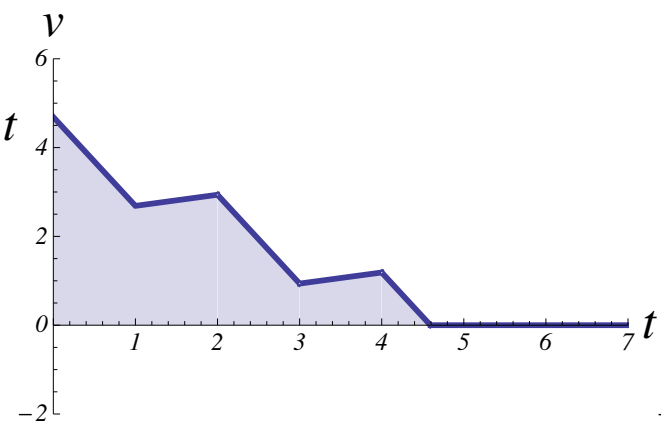
Ex: Car Control

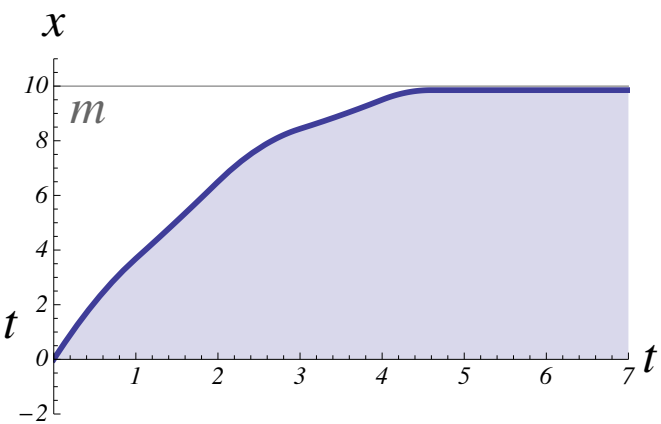
Accelerate condition $?Q$

Example (Single car car_s)

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





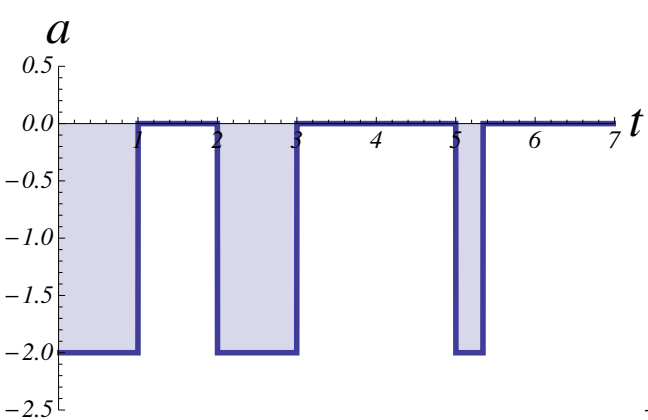


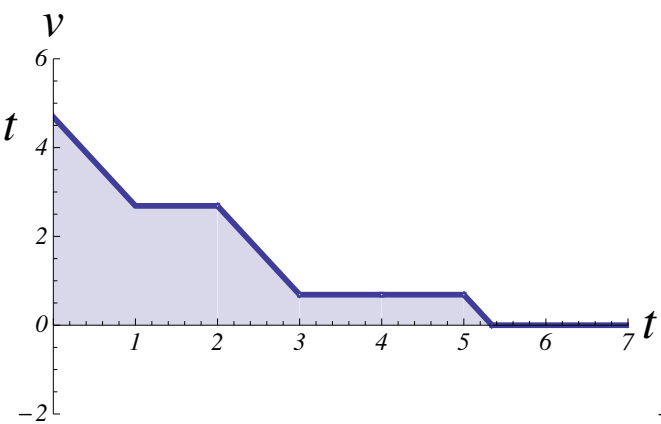
Ex: Car Control

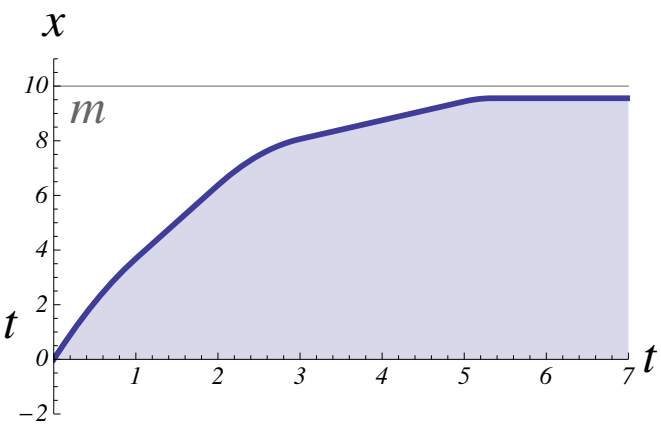
Accelerate condition $?Q$ depends on A

Example (Single car car_s)

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







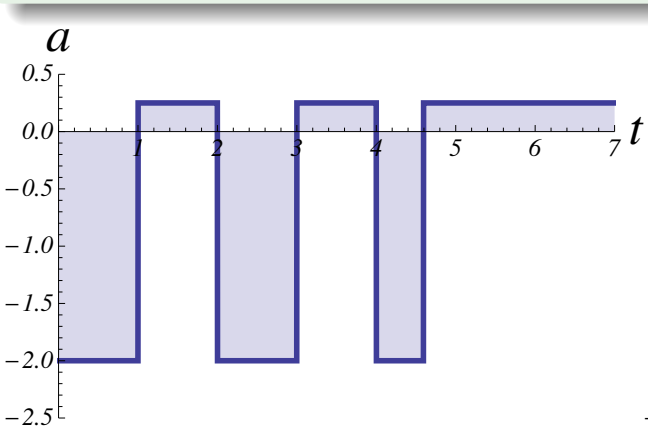
$Q \equiv$

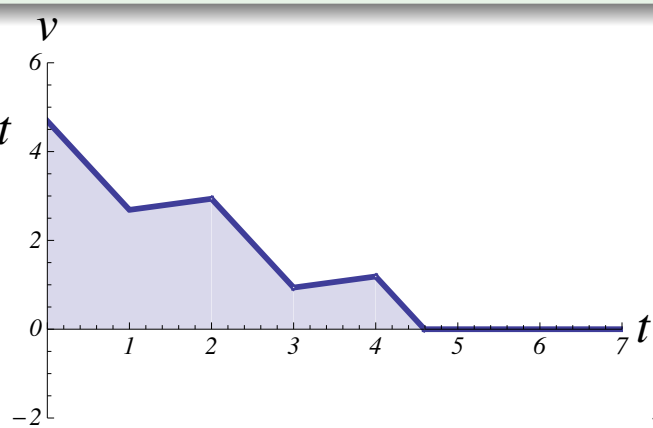
Example (Single car car_ε time-triggered)

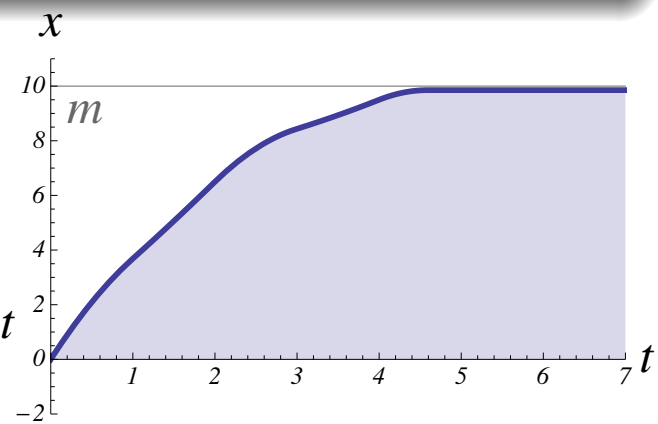
$$\left(((?Q; a := A) \cup a := -b);\ t := 0;\ x' = v, v' = a, t' = 1 \,\&\, v \geq 0 \wedge t \leq \varepsilon\right)^*$$

Example (Safely stays before traffic light m)

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







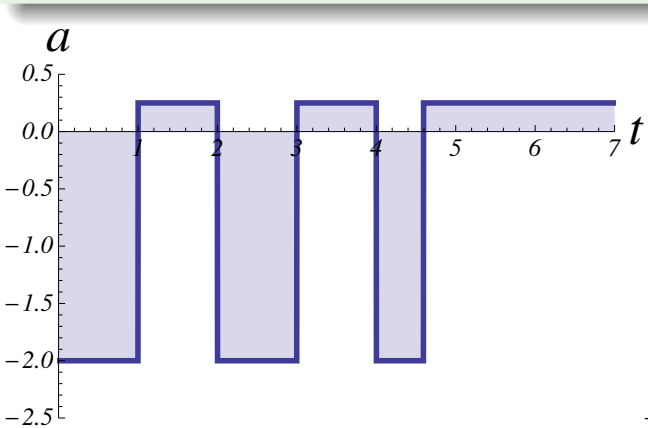
$$Q \equiv 2b(m-x) \geq v^2 + (A+b)(A\varepsilon^2 + 2\varepsilon v)$$

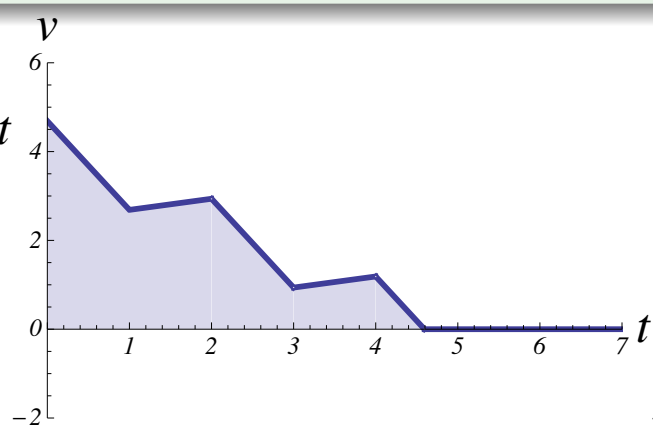
Example (Single car car_ε time-triggered)

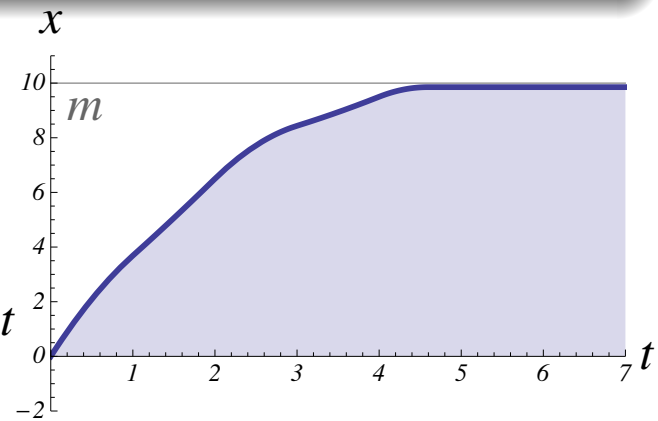
$$\left(((?Q; a := A) \cup a := -b);\ t := 0;\ x' = v, v' = a, t' = 1 \,\&\, v \geq 0 \wedge t \leq \varepsilon\right)^*$$

Example (Safely stays before traffic light m)

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







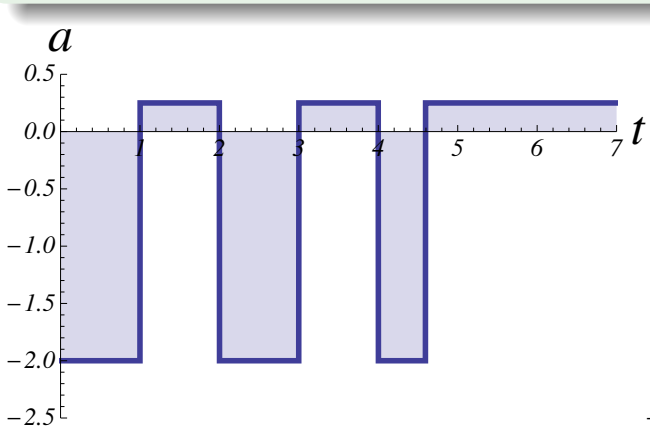
$$Q \equiv 2b(m-x) \geq v^2 + (A+b)(A\varepsilon^2 + 2\varepsilon v)$$

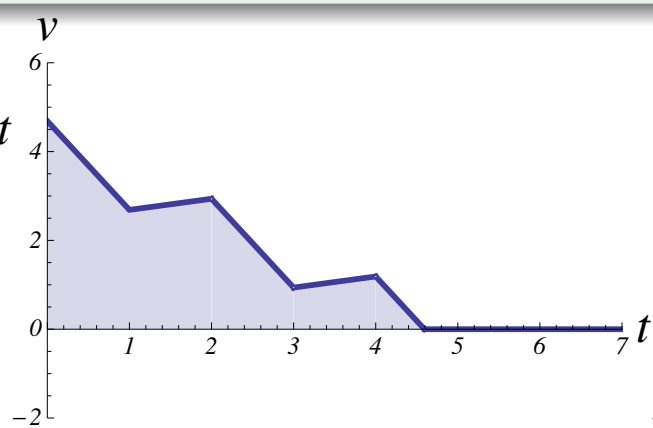
Example (Single car car_ε time-triggered)

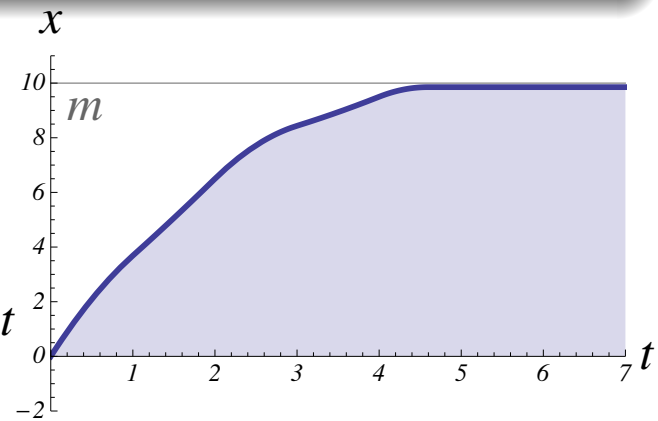
$$\left(((?Q; a := A) \cup a := -b);\ t := 0;\ x' = v, v' = a, t' = 1 \,\&\, v \geq 0 \wedge t \leq \varepsilon\right)^*$$

Example (Live, can move everywhere)

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



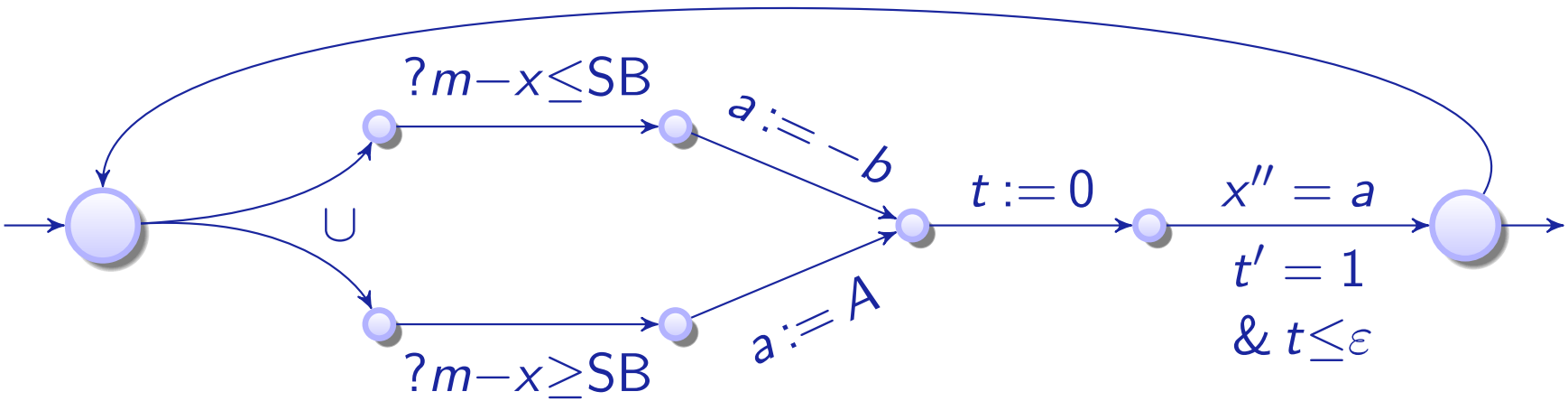




Branching Transitions in Hybrid Programs

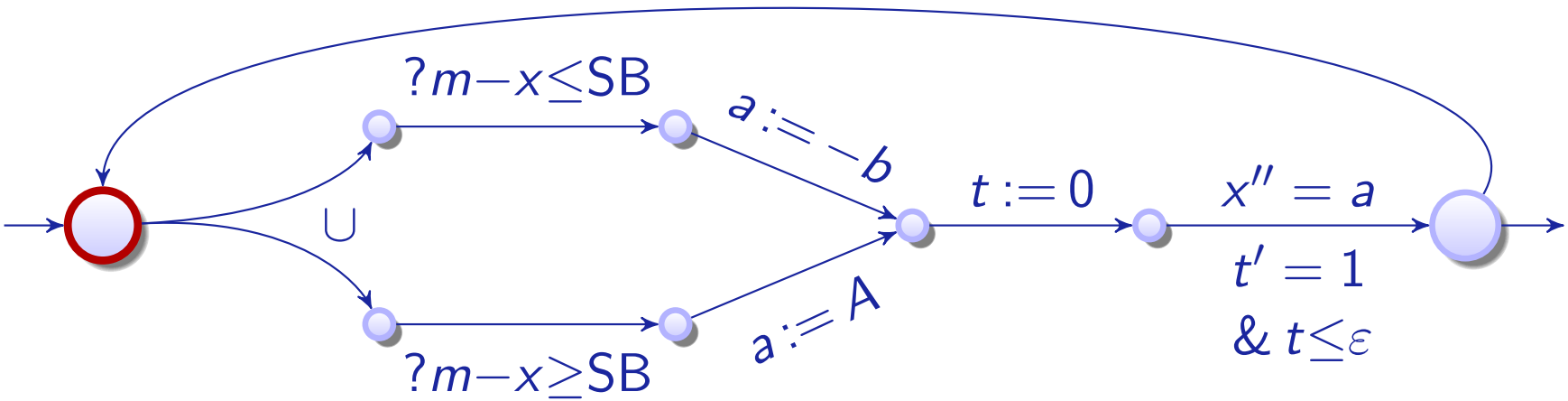
$$
\begin{aligned}
\text{Robot} &\equiv (\text{ctrl}\,;\text{drive})^* \\
\text{ctrl} &\equiv (?m - x \leq \text{SB}; a := -b) \\
&\quad \cup (?m - x \geq \text{SB}; a := A) \\
\text{drive} &\equiv t := 0; (x' = v, v' = a, t' = 1 \\
&\quad \& \, v \geq 0 \wedge t \leq \varepsilon)
\end{aligned}
$$

Branching Transitions in Hybrid Programs



$$\begin{aligned}
\text{Robot} &\equiv (\text{ctrl}\,;\text{drive})^* \\
\text{ctrl} &\equiv (?m - x \leq \text{SB}; a := -b) \\
&\quad \cup (?m - x \geq \text{SB}; a := A) \\
\text{drive} &\equiv t := 0; (x' = v, v' = a, t' = 1 \\
&\quad \& v \geq 0 \wedge t \leq \varepsilon)
\end{aligned}$$

Branching Transitions in Hybrid Programs

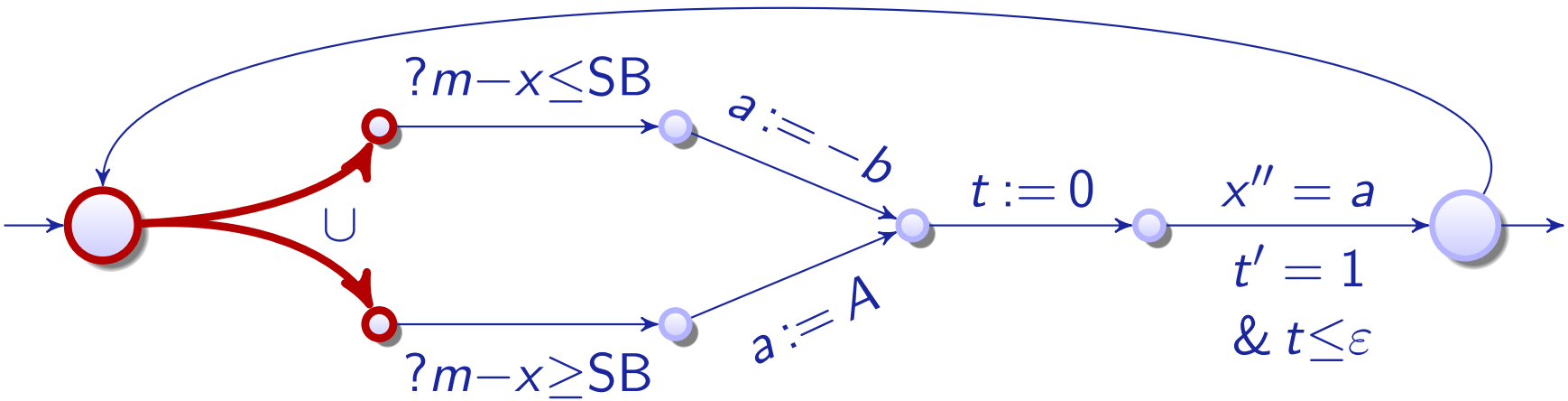


$$\text{Robot} \equiv (\text{ctrl}\,;\text{drive})^*$$

$$\text{ctrl} \equiv (?m - x \leq \text{SB}; a := -b) \cup (?m - x \geq \text{SB}; a := A)$$

$$\text{drive} \equiv t := 0; (x' = v, v' = a, t' = 1 \;\&\; v \geq 0 \wedge t \leq \varepsilon)$$

Branching Transitions in Hybrid Programs

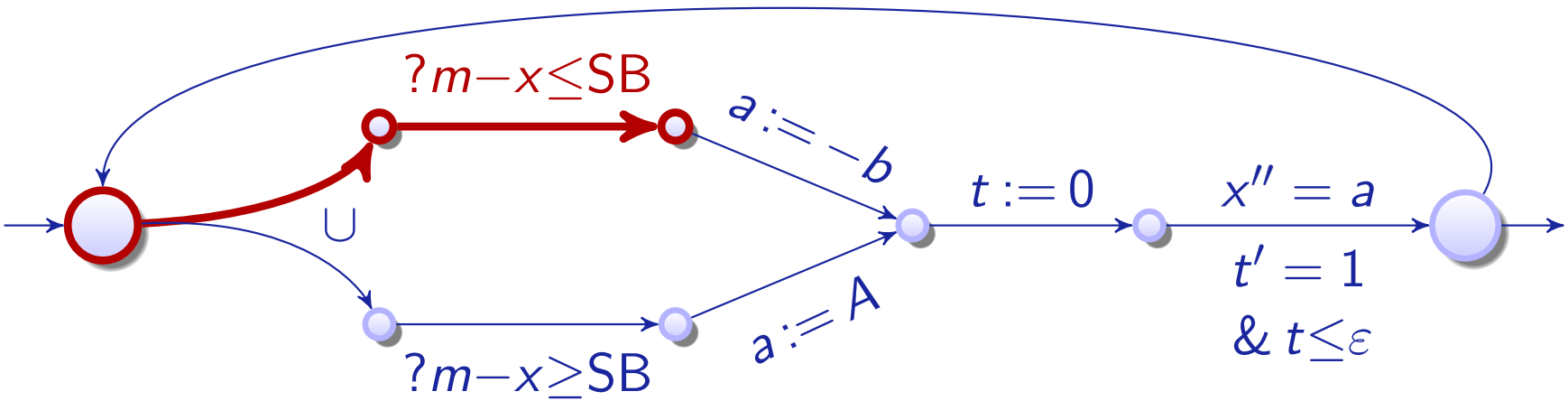


$$\text{Robot} \equiv (\text{ctrl}\,;\text{drive})^*$$

$$\text{ctrl} \equiv (?m - x \leq \text{SB}; a := -b) \cup (?m - x \geq \text{SB}; a := A)$$

$$\text{drive} \equiv t := 0; (x' = v, v' = a, t' = 1 \;\&\; v \geq 0 \wedge t \leq \varepsilon)$$

Branching Transitions in Hybrid Programs

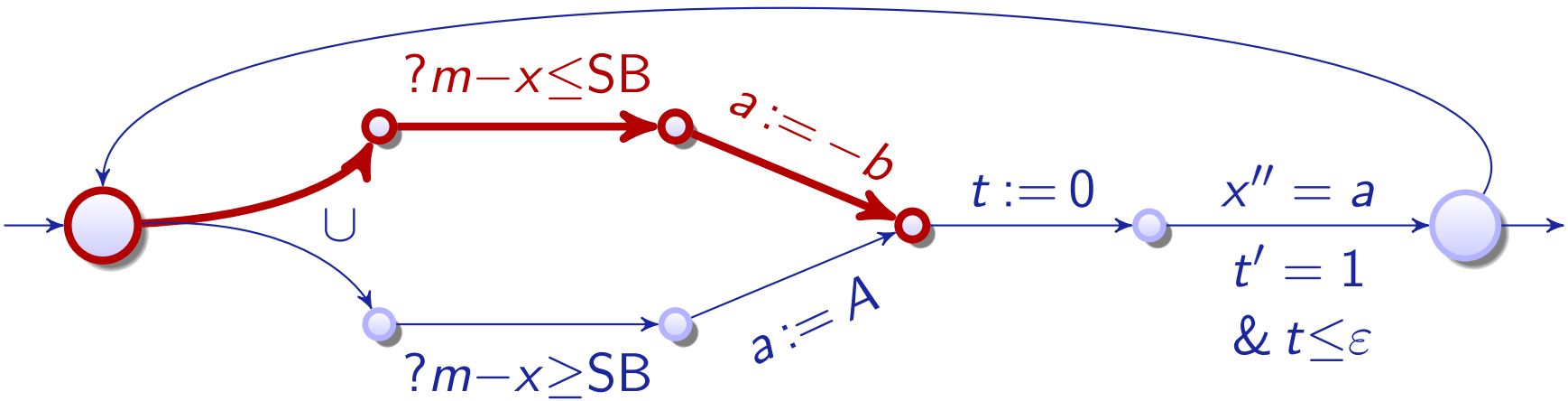


$$\text{Robot} \equiv (\text{ctrl}\,;\text{drive})^*$$

$$\text{ctrl} \equiv (?m - x \leq \text{SB}; a := -b) \cup (?m - x \geq \text{SB}; a := A)$$

$$\text{drive} \equiv t := 0; (x' = v, v' = a, t' = 1 \;\&\; v \geq 0 \wedge t \leq \varepsilon)$$

Branching Transitions in Hybrid Programs

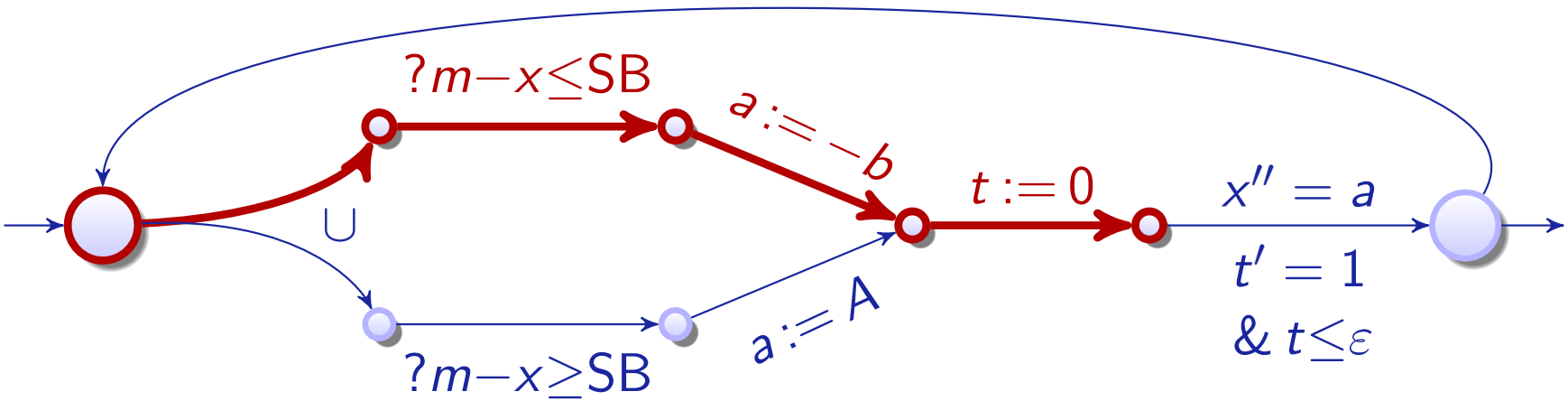


$$\text{Robot} \equiv (\text{ctrl}\,;\text{drive})^*$$

$$\text{ctrl} \equiv (?m - x \leq \text{SB}; a := -b) \cup (?m - x \geq \text{SB}; a := A)$$

$$\text{drive} \equiv t := 0; (x' = v, v' = a, t' = 1 \;\&\; v \geq 0 \wedge t \leq \varepsilon)$$

Branching Transitions in Hybrid Programs

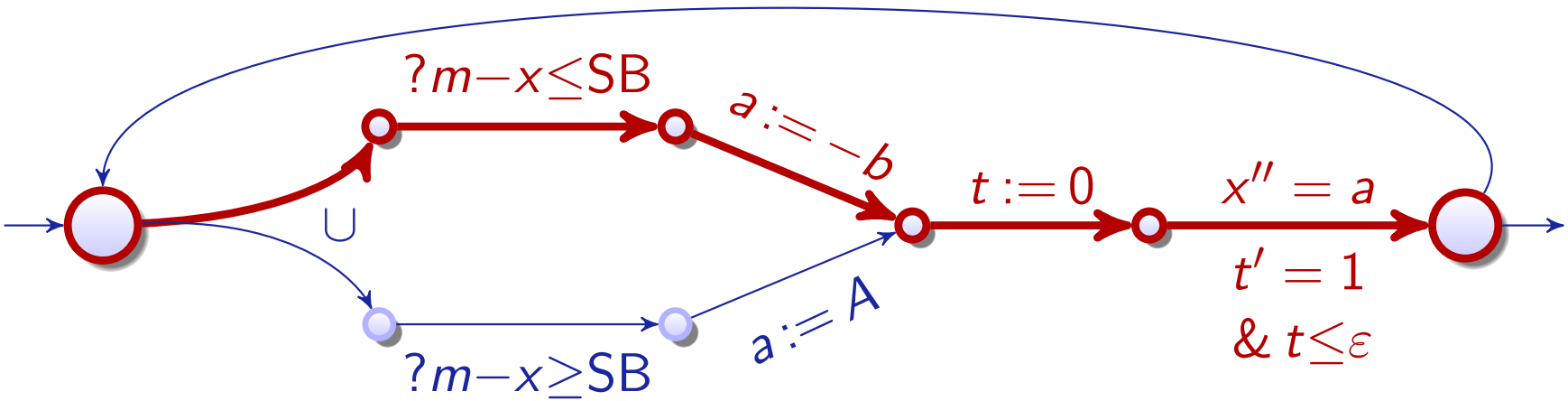


$$\text{Robot} \equiv (\text{ctrl}\,;\text{drive})^*$$

$$\text{ctrl} \equiv (?m - x \leq \text{SB}; a := -b) \cup (?m - x \geq \text{SB}; a := A)$$

$$\text{drive} \equiv t := 0; (x' = v, v' = a, t' = 1 \;\&\; v \geq 0 \wedge t \leq \varepsilon)$$

Branching Transitions in Hybrid Programs

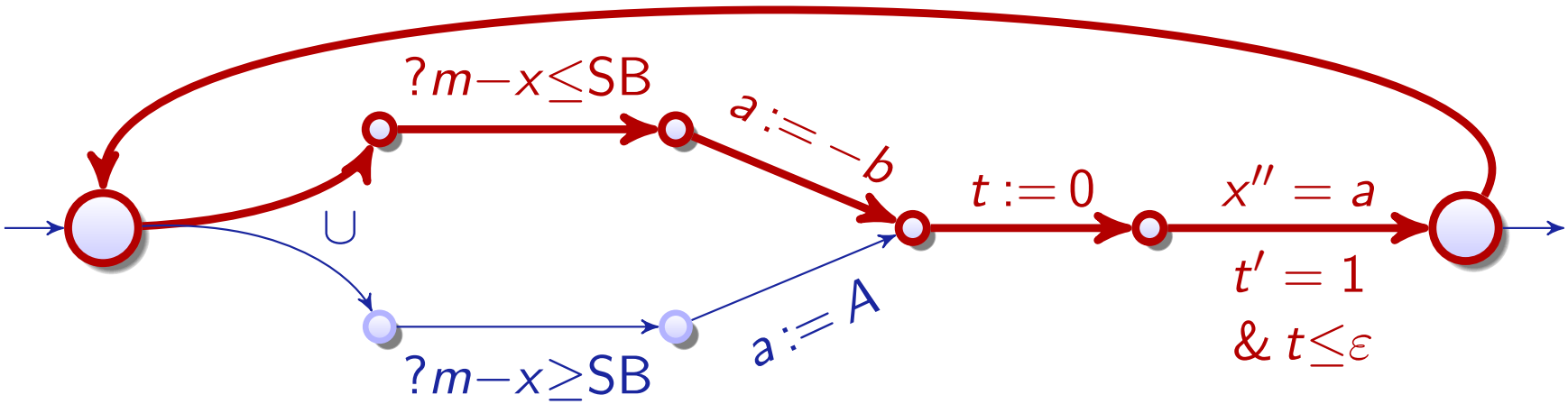


$$\text{Robot} \equiv (\text{ctrl}\,;\text{drive})^*$$

$$\text{ctrl} \equiv (?m - x \leq \text{SB}; a := -b) \cup (?m - x \geq \text{SB}; a := A)$$

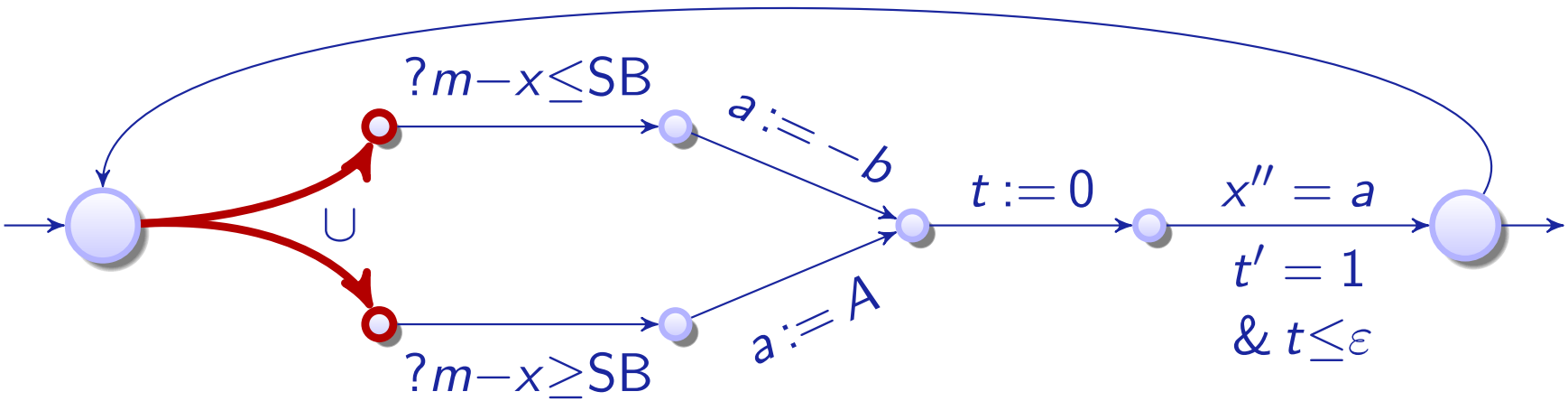
$$\text{drive} \equiv t := 0; (x' = v, v' = a, t' = 1 \;\&\; v \geq 0 \wedge t \leq \varepsilon)$$

Branching Transitions in Hybrid Programs



$$\begin{aligned}
\text{Robot} &\equiv (\text{ctrl}\,;\text{drive})^* \\
\text{ctrl} &\equiv (?m - x \leq \text{SB}; a := -b) \\
&\quad \cup (?m - x \geq \text{SB}; a := A) \\
\text{drive} &\equiv t := 0; (x' = v, v' = a, t' = 1 \\
&\quad \& v \geq 0 \wedge t \leq \varepsilon)
\end{aligned}$$

Branching Transitions in Hybrid Programs

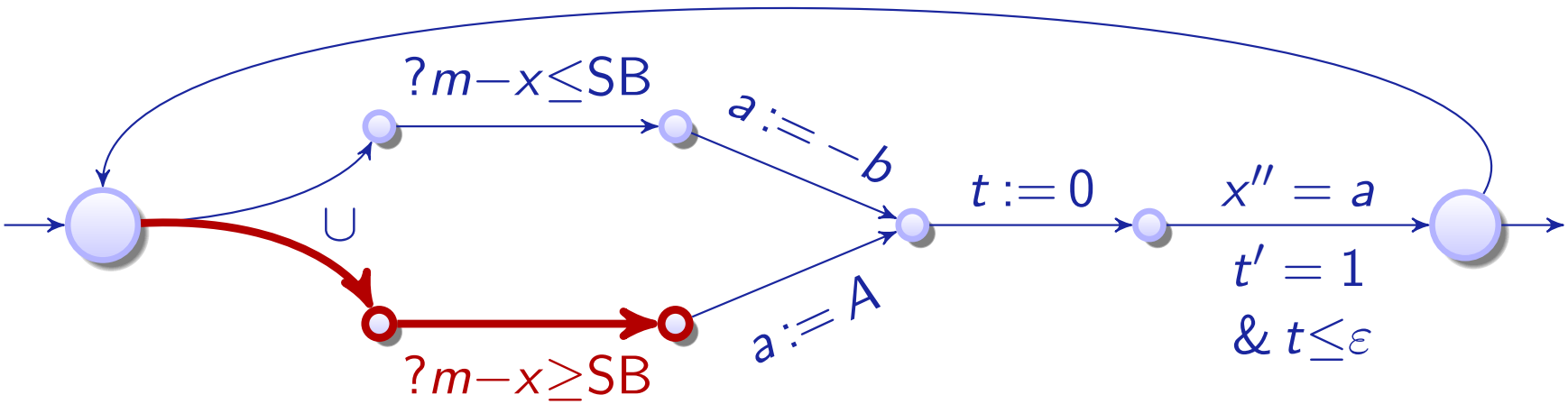


$$\text{Robot} \equiv (\text{ctrl}\,;\text{drive})^*$$

$$\text{ctrl} \equiv (?m - x \leq \text{SB}; a := -b) \cup (?m - x \geq \text{SB}; a := A)$$

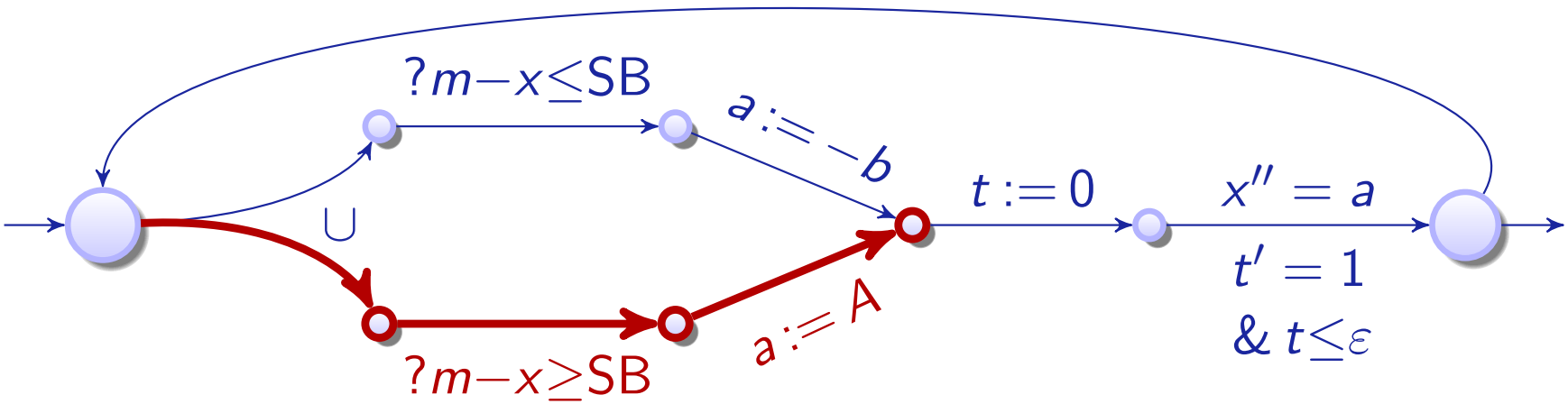
$$\text{drive} \equiv t := 0; (x' = v, v' = a, t' = 1 \;\&\; v \geq 0 \wedge t \leq \varepsilon)$$

Branching Transitions in Hybrid Programs



$$\begin{aligned}
\text{Robot} &\equiv (\text{ctrl}\,;\text{drive})^* \\
\text{ctrl} &\equiv (?m - x \leq \text{SB}; a := -b) \\
&\quad \cup (?m - x \geq \text{SB}; a := A) \\
\text{drive} &\equiv t := 0; (x' = v, v' = a, t' = 1 \\
&\quad \& v \geq 0 \wedge t \leq \varepsilon)
\end{aligned}$$

Branching Transitions in Hybrid Programs

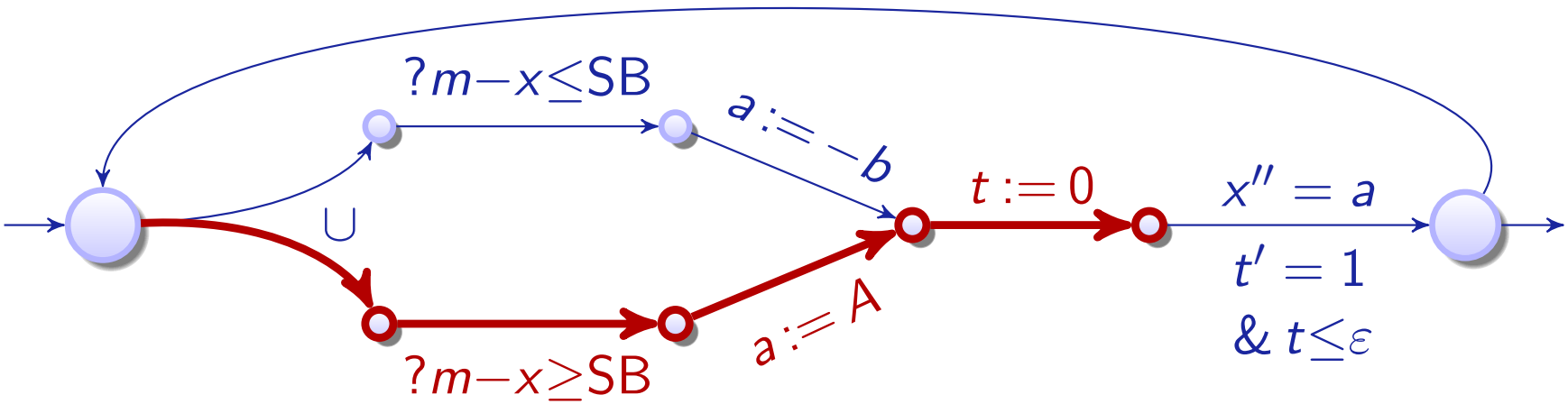


$$\text{Robot} \equiv (\text{ctrl}\,;\text{drive})^*$$

$$\text{ctrl} \equiv (?m - x \leq \text{SB}; a := -b) \cup (?m - x \geq \text{SB}; a := A)$$

$$\text{drive} \equiv t := 0; (x' = v, v' = a, t' = 1 \;\&\; v \geq 0 \wedge t \leq \varepsilon)$$

Branching Transitions in Hybrid Programs

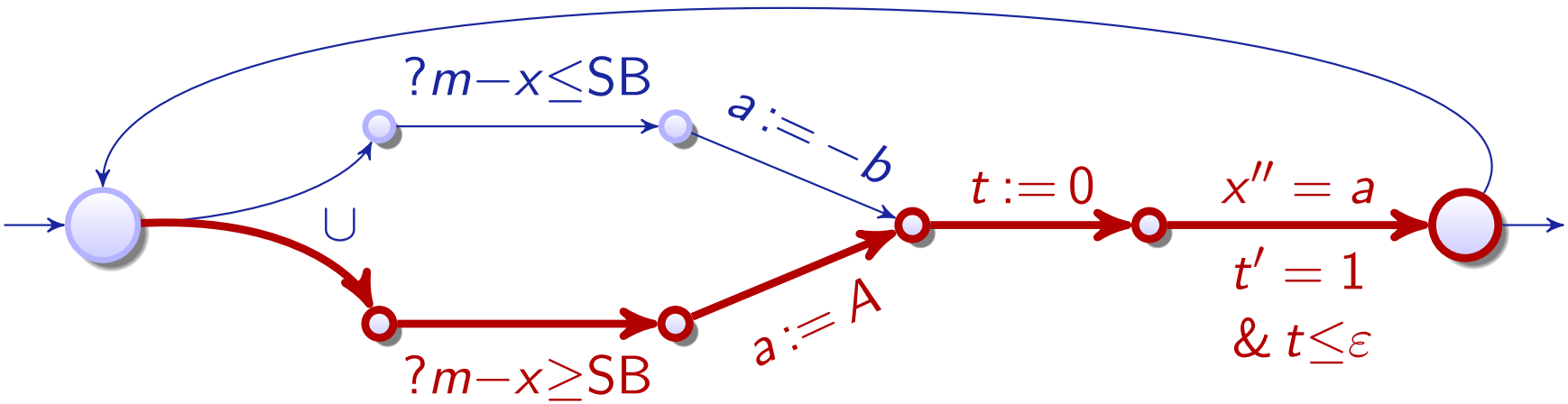


$$\text{Robot} \equiv (\text{ctrl}\,;\text{drive})^*$$

$$\text{ctrl} \equiv (?m - x \leq \text{SB}; a := -b) \cup (?m - x \geq \text{SB}; a := A)$$

$$\text{drive} \equiv t := 0; (x' = v, v' = a, t' = 1 \;\&\; v \geq 0 \wedge t \leq \varepsilon)$$

Branching Transitions in Hybrid Programs

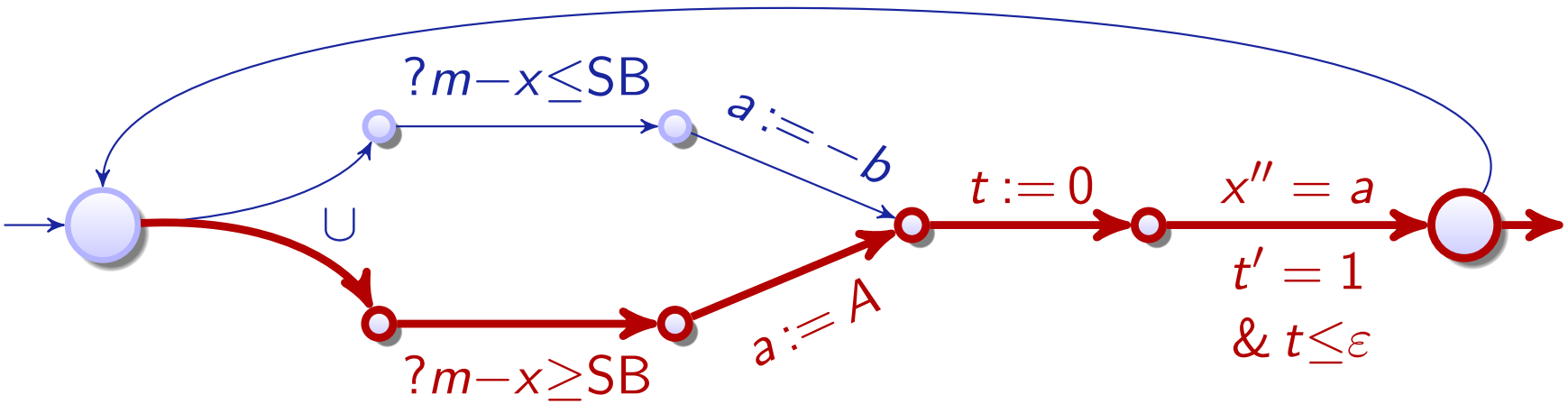


$$\text{Robot} \equiv (\text{ctrl}\,;\text{drive})^*$$

$$\text{ctrl} \equiv (?m - x \leq \text{SB}; a := -b) \cup (?m - x \geq \text{SB}; a := A)$$

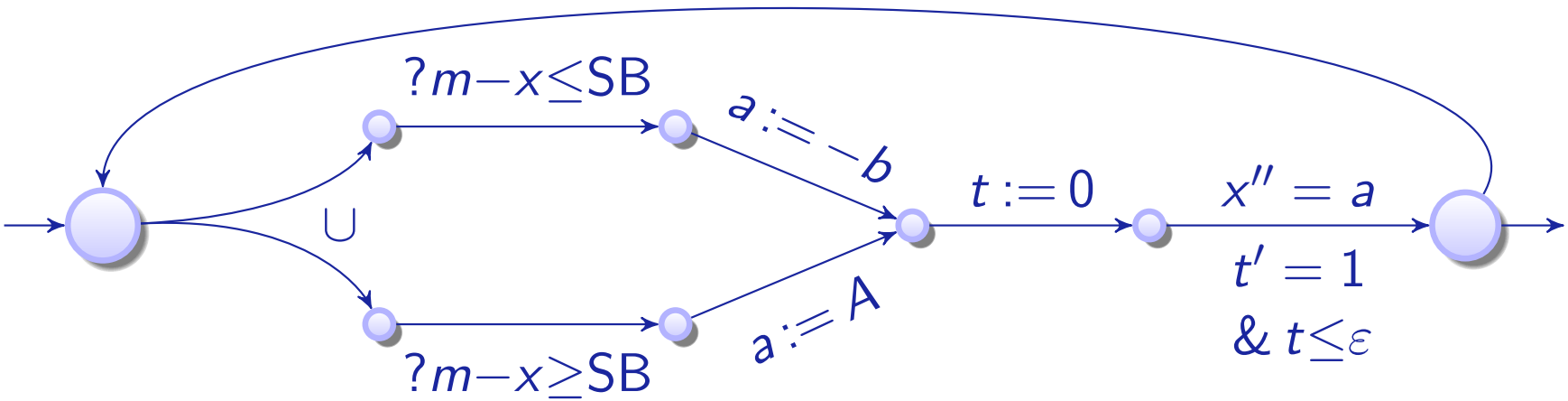
$$\text{drive} \equiv t := 0; (x' = v, v' = a, t' = 1 \;\&\; v \geq 0 \wedge t \leq \varepsilon)$$

Branching Transitions in Hybrid Programs



$$\begin{aligned}
\text{Robot} &\equiv (\text{ctrl}\,;\text{drive})^* \\
\text{ctrl} &\equiv (?m - x \leq \text{SB}; a := -b) \\
&\quad \cup (?m - x \geq \text{SB}; a := A) \\
\text{drive} &\equiv t := 0; (x' = v, v' = a, t' = 1 \\
&\quad \& v \geq 0 \wedge t \leq \varepsilon)
\end{aligned}$$

Branching Transitions in Hybrid Programs



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

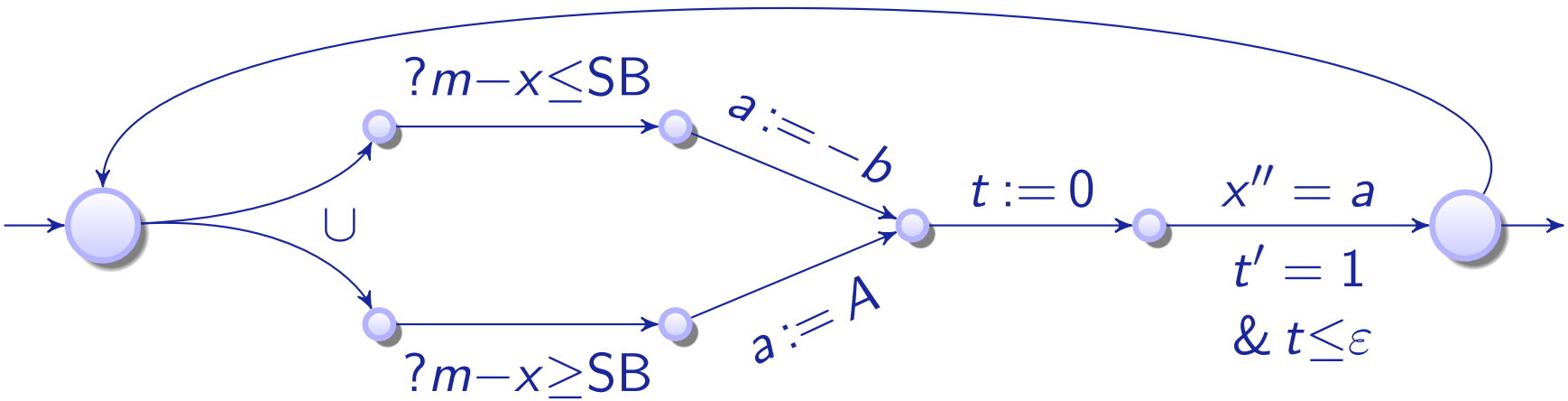
$\texttt{while}(Q)\,\alpha \equiv$

$$\text{Robot} \equiv (\text{ctrl}\,;\text{drive})^*$$

$$\text{ctrl} \equiv (?m - x \leq \text{SB}; a := -b) \cup (?m - x \geq \text{SB}; a := A)$$

$$\text{drive} \equiv t := 0; (x' = v, v' = a, t' = 1 \;\&\; v \geq 0 \wedge t \leq \varepsilon)$$

Branching Transitions in Hybrid Programs

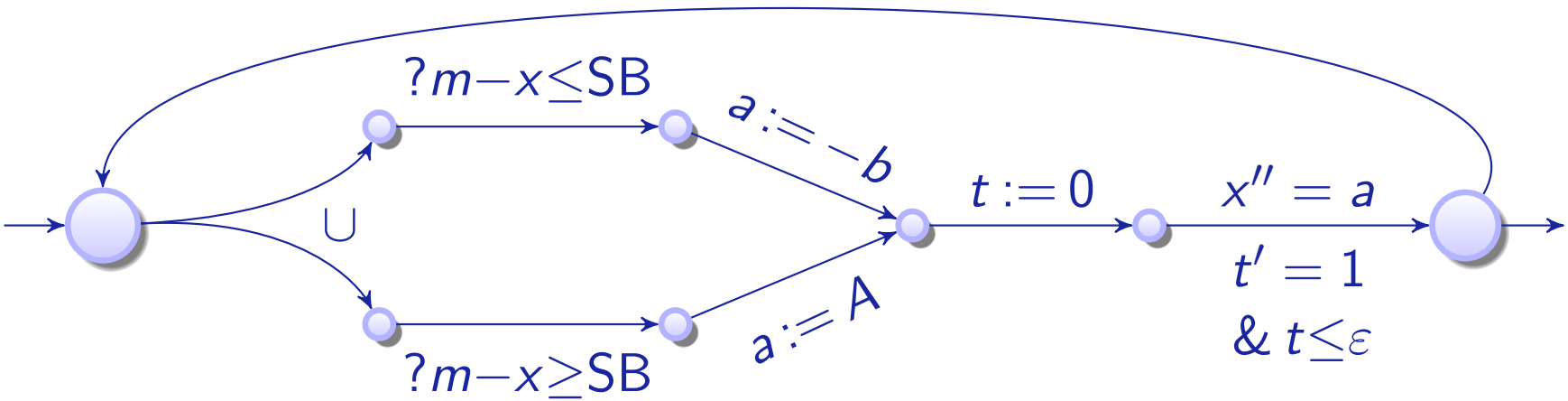


$$\texttt{if}(Q)\,\alpha\,\texttt{else}\,\beta \equiv (?Q;\alpha) \cup (?\neg Q;\beta)$$

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

$$\begin{aligned}
\text{Robot} &\equiv (\text{ctrl}\,;\text{drive})^* \\
\text{ctrl} &\equiv (?m - x \leq \text{SB}; a := -b) \\
&\quad \cup (?m - x \geq \text{SB}; a := A) \\
\text{drive} &\equiv t := 0; (x' = v, v' = a, t' = 1 \\
&\quad \& v \geq 0 \wedge t \leq \varepsilon)
\end{aligned}$$

Branching Transitions in Hybrid Programs



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

$$\begin{aligned}
\text{Robot} &\equiv (\text{ctrl}\,;\text{drive})^* \\
\text{ctrl} &\equiv (?m - x \leq \text{SB}; a := -b) \\
&\quad \cup (?m - x \geq \text{SB}; a := A) \\
\text{drive} &\equiv t := 0; (x' = v, v' = a, t' = 1 \\
&\quad \& v \geq 0 \wedge t \leq \varepsilon)
\end{aligned}$$

Outline

Differential Dynamic Logic: Axiomatization

equations of truth

[:=] $[x := e]P(x) \leftrightarrow P(e)$

[?] $[?Q]P \leftrightarrow (Q \rightarrow P)$

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

[∪] $[\alpha \cup \beta]P \leftrightarrow [\alpha]P \wedge [\beta]P$

[;] $[\alpha; \beta]P \leftrightarrow [\alpha][\beta]P$

[*] $[\alpha^*]P \leftrightarrow P \wedge [\alpha][\alpha^*]P$

K $[\alpha](P \rightarrow Q) \rightarrow ([\alpha]P \rightarrow [\alpha]Q)$

I $[\alpha^*](P \rightarrow [\alpha]P) \rightarrow (P \rightarrow [\alpha^*]P)$

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

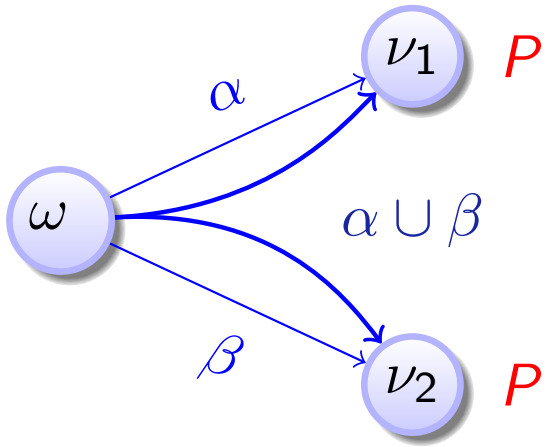
LICS'12,JAR'16

Proofs for Hybrid Systems

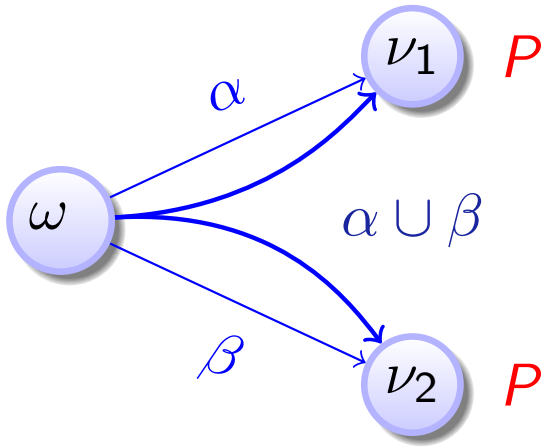
compositional semantics $\Rightarrow$ compositional rules!

Proofs for Hybrid Systems

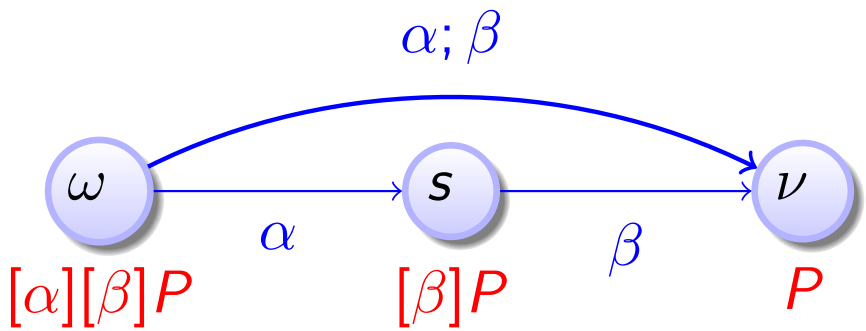
$$[\alpha \cup \beta]P \leftrightarrow [\alpha]P \wedge [\beta]P$$



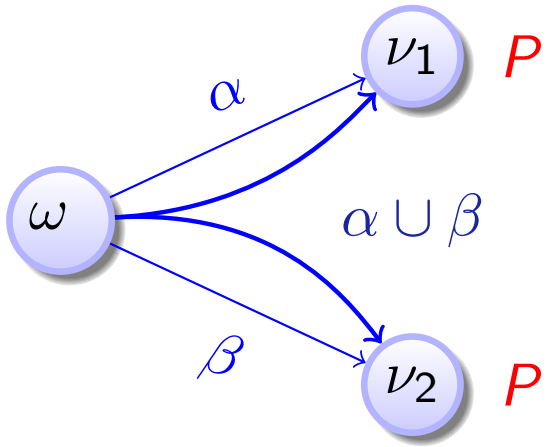
$[\alpha \cup \beta]P \leftrightarrow [\alpha]P \wedge [\beta]P$



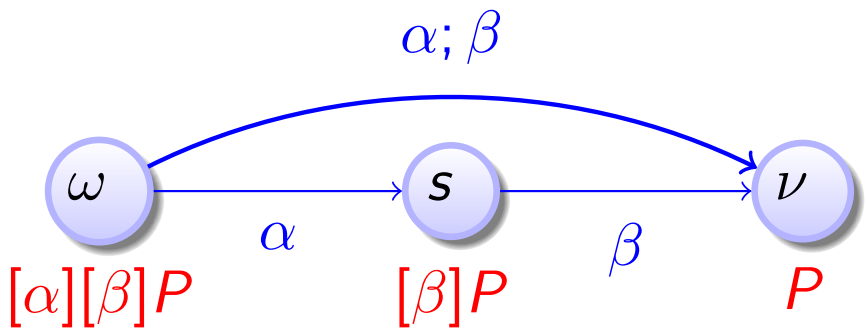
$[\alpha;\beta]P \leftrightarrow [\alpha][\beta]P$



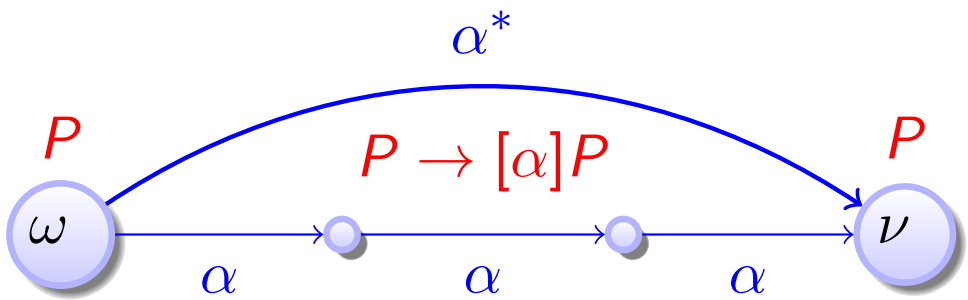
$$[\alpha \cup \beta]P \leftrightarrow [\alpha]P \wedge [\beta]P$$



$$[\alpha;\beta]P \leftrightarrow [\alpha][\beta]P$$



$$\frac{P \quad P \rightarrow [\alpha]P}{[\alpha^*]P}$$



Example Proof: Safe Braking

$J(x, v) \equiv x \leq m$

$$[;]\ \frac{}{J(x,v) \vdash [a := -b; (x' = v, v' = a)]J(x,v)}$$

1. $\Gamma \vdash \Delta$ shape of conjecture to prove — sequent
2. Γ is list of available assumptions — antecedent
3. Δ needs to be proved from assumptions Γ — succedent
4. Proof reduces desired conclusion (at the bottom) to premises with remaining subgoals (top) until no more subgoals ($*$)

Example Proof: Safe Braking

$J(x, v) \equiv x \leq m$

$$\dfrac{\dfrac{}{J(x,v) \vdash [a := -b][x' = v, v' = a]J(x,v)}[:=]}{J(x,v) \vdash [a := -b; (x' = v, v' = a)]J(x,v)}[;]$$

1. $\Gamma \vdash \Delta$ shape of conjecture to prove — sequent
2. Γ is list of available assumptions — antecedent
3. Δ needs to be proved from assumptions Γ — succedent
4. Proof reduces desired conclusion (at the bottom) to premises with remaining subgoals (top) until no more subgoals ($*$)

Example Proof: Safe Braking

$J(x, v) \equiv x \leq m$

$$[']\ \frac{}{J(x,v) \vdash [x' = v, v' = -b]J(x,v)}$$
$$[:=]\ \frac{J(x,v) \vdash [x' = v, v' = -b]J(x,v)}{J(x,v) \vdash [a := -b][x' = v, v' = a]J(x,v)}$$
$$[;]\ \frac{J(x,v) \vdash [a := -b][x' = v, v' = a]J(x,v)}{J(x,v) \vdash [a := -b; (x' = v, v' = a)]J(x,v)}$$

1. $\Gamma \vdash \Delta$ shape of conjecture to prove — sequent
2. Γ is list of available assumptions — antecedent
3. Δ needs to be proved from assumptions Γ — succedent
4. Proof reduces desired conclusion (at the bottom) to premises with remaining subgoals (top) until no more subgoals ($*$)

Example Proof: Safe Braking

$J(x,v) \equiv x \leq m$

$$\dfrac{\dfrac{\dfrac{\dfrac{}{J(x,v) \vdash \forall t{\geq}0\, [x := -\frac{b}{2}t^2 + vt + x]J(x,v)}[:=]}{J(x,v) \vdash [x' = v, v' = -b]J(x,v)}[']}{J(x,v) \vdash [a := -b][x' = v, v' = a]J(x,v)}[:=]}{J(x,v) \vdash [a := -b; (x' = v, v' = a)]J(x,v)}[;]$$

1. $\Gamma \vdash \Delta$ shape of conjecture to prove — sequent
2. Γ is list of available assumptions — antecedent
3. Δ needs to be proved from assumptions Γ — succedent
4. Proof reduces desired conclusion (at the bottom) to premises with remaining subgoals (top) until no more subgoals ($*$)

Example Proof: Safe Braking

$J(x,v) \equiv x \leq m$

$$\text{QE}\,\frac{}{J(x,v) \vdash \forall t{\geq}0\,(-\tfrac{b}{2}t^2 + vt + x \leq m)}$$
$$[:=]\,\frac{J(x,v) \vdash \forall t{\geq}0\,(-\tfrac{b}{2}t^2 + vt + x \leq m)}{J(x,v) \vdash \forall t{\geq}0\,[x := -\tfrac{b}{2}t^2 + vt + x]J(x,v)}$$
$$[']\,\frac{J(x,v) \vdash \forall t{\geq}0\,[x := -\tfrac{b}{2}t^2 + vt + x]J(x,v)}{J(x,v) \vdash [x' = v, v' = -b]J(x,v)}$$
$$[:=]\,\frac{J(x,v) \vdash [x' = v, v' = -b]J(x,v)}{J(x,v) \vdash [a := -b][x' = v, v' = a]J(x,v)}$$
$$[;]\,\frac{J(x,v) \vdash [a := -b][x' = v, v' = a]J(x,v)}{J(x,v) \vdash [a := -b; (x' = v, v' = a)]J(x,v)}$$

1. $\Gamma \vdash \Delta$ shape of conjecture to prove — sequent
2. Γ is list of available assumptions — antecedent
3. Δ needs to be proved from assumptions Γ — succedent
4. Proof reduces desired conclusion (at the bottom) to premises with remaining subgoals (top) until no more subgoals ($*$)

Example Proof: Safe Braking

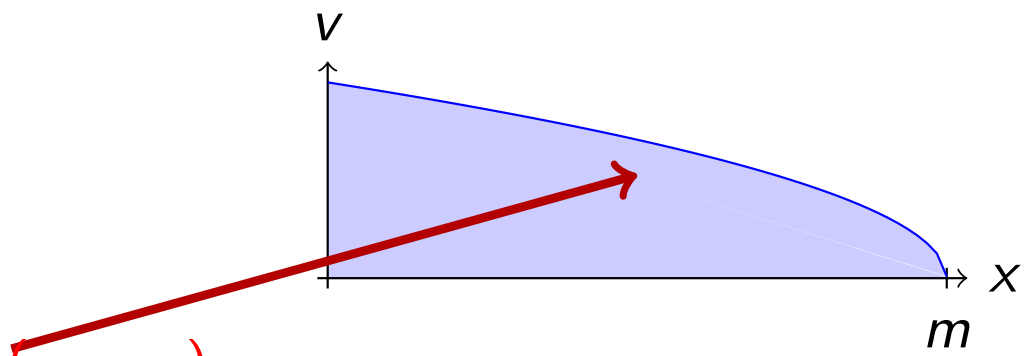
$J(x,v) \equiv x \leq m$

$$\text{QE}\,\frac{J(x,v) \vdash v^2 \leq 2b(m-x)}{J(x,v) \vdash \forall t{\geq}0\,(-\frac{b}{2}t^2 + vt + x \leq m)}$$
$$[:=]\,\frac{J(x,v) \vdash \forall t{\geq}0\,[x := -\frac{b}{2}t^2 + vt + x]J(x,v)}{J(x,v) \vdash [x' = v, v' = -b]J(x,v)}$$
$$[']\,\frac{J(x,v) \vdash [x' = v, v' = -b]J(x,v)}{J(x,v) \vdash [a := -b][x' = v, v' = a]J(x,v)}$$
$$[:=]\,\frac{J(x,v) \vdash [a := -b][x' = v, v' = a]J(x,v)}{J(x,v) \vdash [a := -b; (x' = v, v' = a)]J(x,v)}$$
$$[;]$$

1. $\Gamma \vdash \Delta$ shape of conjecture to prove — sequent
2. Γ is list of available assumptions — antecedent
3. Δ needs to be proved from assumptions Γ — succedent
4. Proof reduces desired conclusion (at the bottom) to premises with remaining subgoals (top) until no more subgoals ($*$)

Example Proof: Safe Braking

$J(x,v) \equiv v^2 \leq 2b(m-x)$



$$\dfrac{J(x,v) \vdash v^2 \leq 2b(m-x)}{J(x,v) \vdash \forall t{\geq}0\,(-\frac{b}{2}t^2 + vt + x \leq m)}\,\text{QE}$$

$$\dfrac{J(x,v) \vdash \forall t{\geq}0\,(-\frac{b}{2}t^2 + vt + x \leq m)}{J(x,v) \vdash \forall t{\geq}0\,[x := -\frac{b}{2}t^2 + vt + x]J(x,v)}\,[:=]$$

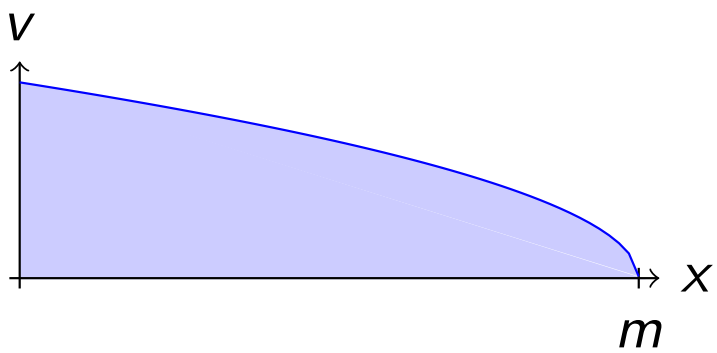
$$\dfrac{J(x,v) \vdash \forall t{\geq}0\,[x := -\frac{b}{2}t^2 + vt + x]J(x,v)}{J(x,v) \vdash [x' = v, v' = -b]J(x,v)}\,[']$$

$$\dfrac{J(x,v) \vdash [x' = v, v' = -b]J(x,v)}{J(x,v) \vdash [a := -b][x' = v, v' = a]J(x,v)}\,[:=]$$

$$\dfrac{J(x,v) \vdash [a := -b][x' = v, v' = a]J(x,v)}{J(x,v) \vdash [a := -b; (x' = v, v' = a)]J(x,v)}\,[;]$$

Example Proof: Safe Accelerating

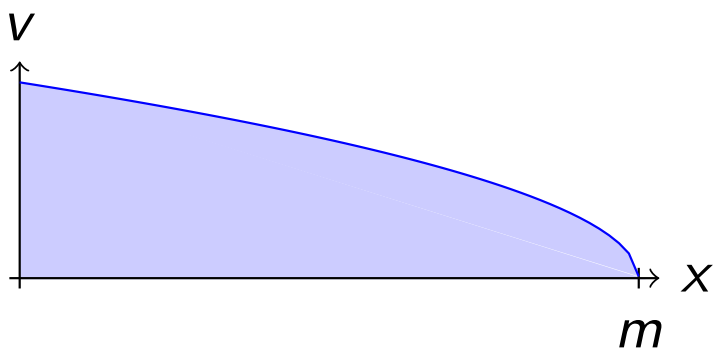
$J(x, v) \equiv v^2 \leq 2b(m - x)$



$$[;]\ \frac{}{J(x,v) \vdash [?\neg \mathsf{SB}; a := A; (x' = v, v' = a, t' = 1 \,\&\, t \leq \varepsilon)]J(x,v)}$$

Example Proof: Safe Accelerating

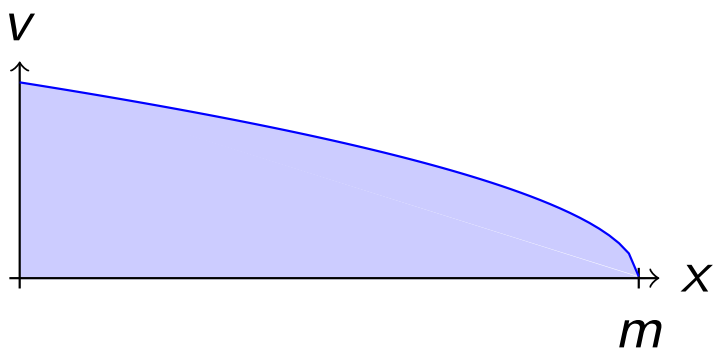
$J(x, v) \equiv v^2 \leq 2b(m - x)$



$$[?]\ \frac{}{J(x,v) \vdash [?\neg \mathsf{SB}][a := A; (x' = v, v' = a, t' = 1 \,\&\, t \leq \varepsilon)]J(x,v)}$$
$$[;]\ \frac{}{J(x,v) \vdash [?\neg \mathsf{SB}; a := A; (x' = v, v' = a, t' = 1 \,\&\, t \leq \varepsilon)]J(x,v)}$$

Example Proof: Safe Accelerating

$J(x, v) \equiv v^2 \leq 2b(m - x)$



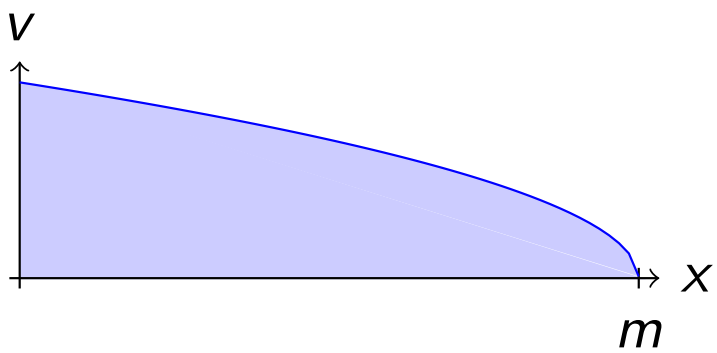
$$[;]\ \frac{J(x,v) \vdash \neg \mathsf{SB} \rightarrow [a := A; (x' = v, v' = a, t' = 1 \,\&\, t \leq \varepsilon)]J(x,v)}{J(x,v) \vdash [?\neg \mathsf{SB}][a := A; (x' = v, v' = a, t' = 1 \,\&\, t \leq \varepsilon)]J(x,v)}$$

$$[?]\ \frac{J(x,v) \vdash [?\neg \mathsf{SB}][a := A; (x' = v, v' = a, t' = 1 \,\&\, t \leq \varepsilon)]J(x,v)}{J(x,v) \vdash [?\neg \mathsf{SB}; a := A; (x' = v, v' = a, t' = 1 \,\&\, t \leq \varepsilon)]J(x,v)}$$

$$[;]\ \frac{J(x,v) \vdash [?\neg \mathsf{SB}; a := A; (x' = v, v' = a, t' = 1 \,\&\, t \leq \varepsilon)]J(x,v)}{}$$

Example Proof: Safe Accelerating

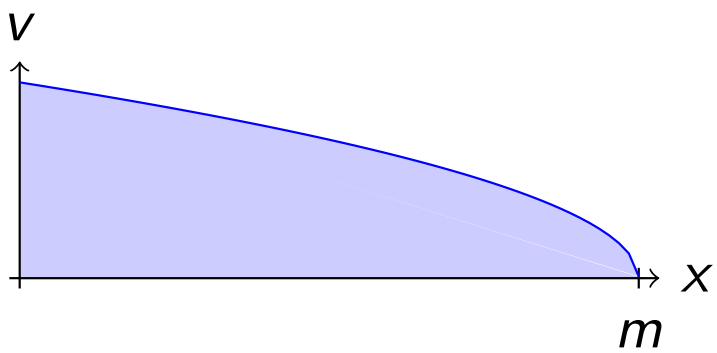
$J(x, v) \equiv v^2 \leq 2b(m - x)$



$$\dfrac{\dfrac{\dfrac{\dfrac{}{J(x,v) \vdash \neg \mathsf{SB} \rightarrow [a := A][x' = v, v' = a, t' = 1 \,\&\, t \leq \varepsilon]J(x,v)}[:=]}{J(x,v) \vdash \neg \mathsf{SB} \rightarrow [a := A; (x' = v, v' = a, t' = 1 \,\&\, t \leq \varepsilon)]J(x,v)}[;]}{J(x,v) \vdash [?\neg \mathsf{SB}][a := A; (x' = v, v' = a, t' = 1 \,\&\, t \leq \varepsilon)]J(x,v)}[?]}{J(x,v) \vdash [?\neg \mathsf{SB}; a := A; (x' = v, v' = a, t' = 1 \,\&\, t \leq \varepsilon)]J(x,v)}[;]$$

Example Proof: Safe Accelerating

$J(x,v) \equiv v^2 \leq 2b(m-x)$



$$[']\ \frac{}{J(x,v) \vdash \neg \mathsf{SB} \rightarrow [x' = v, v' = A, t' = 1 \,\&\, t \leq \varepsilon]J(x,v)}$$

$$[:=]\ \frac{}{J(x,v) \vdash \neg \mathsf{SB} \rightarrow [a := A][x' = v, v' = a, t' = 1 \,\&\, t \leq \varepsilon]J(x,v)}$$

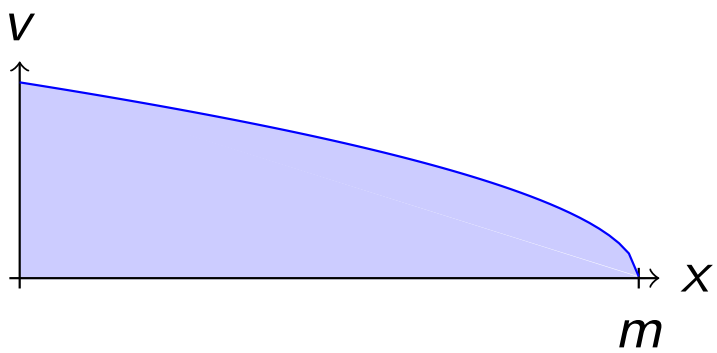
$$[;]\ \frac{}{J(x,v) \vdash \neg \mathsf{SB} \rightarrow [a := A; (x' = v, v' = a, t' = 1 \,\&\, t \leq \varepsilon)]J(x,v)}$$

$$[?]\ \frac{}{J(x,v) \vdash [?\neg \mathsf{SB}][a := A; (x' = v, v' = a, t' = 1 \,\&\, t \leq \varepsilon)]J(x,v)}$$

$$[;]\ \frac{}{J(x,v) \vdash [?\neg \mathsf{SB}; a := A; (x' = v, v' = a, t' = 1 \,\&\, t \leq \varepsilon)]J(x,v)}$$

Example Proof: Safe Accelerating

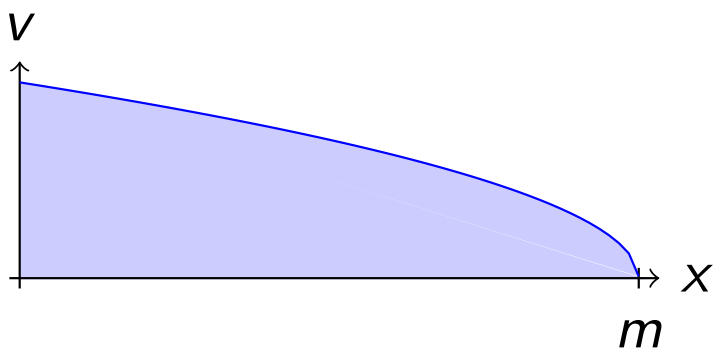
$J(x,v) \equiv v^2 \leq 2b(m-x)$



$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{J(x,v) \vdash \neg \mathrm{SB} \rightarrow \forall t{\geq}0\,(t \leq \varepsilon \rightarrow [x := \frac{A}{2}t^2 + vt + x]J(x,v))}\,[:=]}{J(x,v) \vdash \neg \mathrm{SB} \rightarrow [x' = v, v' = A, t' = 1 \,\&\, t \leq \varepsilon]J(x,v)}\,[']}{J(x,v) \vdash \neg \mathrm{SB} \rightarrow [a := A][x' = v, v' = a, t' = 1 \,\&\, t \leq \varepsilon]J(x,v)}\,[:=]}{J(x,v) \vdash \neg \mathrm{SB} \rightarrow [a := A; (x' = v, v' = a, t' = 1 \,\&\, t \leq \varepsilon)]J(x,v)}\,[;]}{J(x,v) \vdash [?\neg \mathrm{SB}][a := A; (x' = v, v' = a, t' = 1 \,\&\, t \leq \varepsilon)]J(x,v)}\,[?]}{J(x,v) \vdash [?\neg \mathrm{SB}; a := A; (x' = v, v' = a, t' = 1 \,\&\, t \leq \varepsilon)]J(x,v)}\,[;]$$

Example Proof: Safe Accelerating

$J(x,v) \equiv v^2 \leq 2b(m-x)$



$$\frac{}{J(x,v) \vdash \neg \mathrm{SB} \rightarrow \forall t{\geq}0\,(t \leq \varepsilon \rightarrow J(\tfrac{A}{2}t^2 + vt + x, At + v))}$$

$$[:=]\ \frac{J(x,v) \vdash \neg \mathrm{SB} \rightarrow \forall t{\geq}0\,(t \leq \varepsilon \rightarrow J(\tfrac{A}{2}t^2 + vt + x, At + v))}{J(x,v) \vdash \neg \mathrm{SB} \rightarrow \forall t{\geq}0\,(t \leq \varepsilon \rightarrow [x := \tfrac{A}{2}t^2 + vt + x]J(x,v))}$$

$$[']\ \frac{J(x,v) \vdash \neg \mathrm{SB} \rightarrow \forall t{\geq}0\,(t \leq \varepsilon \rightarrow [x := \tfrac{A}{2}t^2 + vt + x]J(x,v))}{J(x,v) \vdash \neg \mathrm{SB} \rightarrow [x' = v, v' = A, t' = 1 \,\&\, t \leq \varepsilon]J(x,v)}$$

$$[:=]\ \frac{J(x,v) \vdash \neg \mathrm{SB} \rightarrow [x' = v, v' = A, t' = 1 \,\&\, t \leq \varepsilon]J(x,v)}{J(x,v) \vdash \neg \mathrm{SB} \rightarrow [a := A][x' = v, v' = a, t' = 1 \,\&\, t \leq \varepsilon]J(x,v)}$$

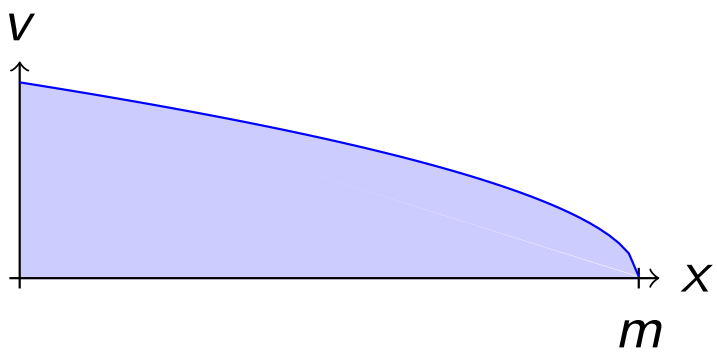
$$[;]\ \frac{J(x,v) \vdash \neg \mathrm{SB} \rightarrow [a := A][x' = v, v' = a, t' = 1 \,\&\, t \leq \varepsilon]J(x,v)}{J(x,v) \vdash \neg \mathrm{SB} \rightarrow [a := A; (x' = v, v' = a, t' = 1 \,\&\, t \leq \varepsilon)]J(x,v)}$$

$$[?]\ \frac{J(x,v) \vdash \neg \mathrm{SB} \rightarrow [a := A; (x' = v, v' = a, t' = 1 \,\&\, t \leq \varepsilon)]J(x,v)}{J(x,v) \vdash [?\neg \mathrm{SB}][a := A; (x' = v, v' = a, t' = 1 \,\&\, t \leq \varepsilon)]J(x,v)}$$

$$[;]\ \frac{J(x,v) \vdash [?\neg \mathrm{SB}][a := A; (x' = v, v' = a, t' = 1 \,\&\, t \leq \varepsilon)]J(x,v)}{J(x,v) \vdash [?\neg \mathrm{SB}; a := A; (x' = v, v' = a, t' = 1 \,\&\, t \leq \varepsilon)]J(x,v)}$$

Example Proof: Safe Accelerating

$J(x,v) \equiv v^2 \leq 2b(m-x)$



$$\text{QE}\ \frac{}{J(x,v) \vdash \neg\text{SB} \rightarrow \forall t{\geq}0\,(t \leq \varepsilon \rightarrow (At+v)^2 \leq 2b(m-\frac{A}{2}t^2-vt-x))}$$

$$\frac{}{J(x,v) \vdash \neg\text{SB} \rightarrow \forall t{\geq}0\,(t \leq \varepsilon \rightarrow J(\frac{A}{2}t^2+vt+x, At+v))}$$

$$[:=]\ \frac{}{J(x,v) \vdash \neg\text{SB} \rightarrow \forall t{\geq}0\,(t \leq \varepsilon \rightarrow [x := \frac{A}{2}t^2+vt+x]J(x,v))}$$

$$[']\ \frac{}{J(x,v) \vdash \neg\text{SB} \rightarrow [x'=v, v'=A, t'=1 \,\&\, t \leq \varepsilon]J(x,v)}$$

$$[:=]\ \frac{}{J(x,v) \vdash \neg\text{SB} \rightarrow [a := A][x'=v, v'=a, t'=1 \,\&\, t \leq \varepsilon]J(x,v)}$$

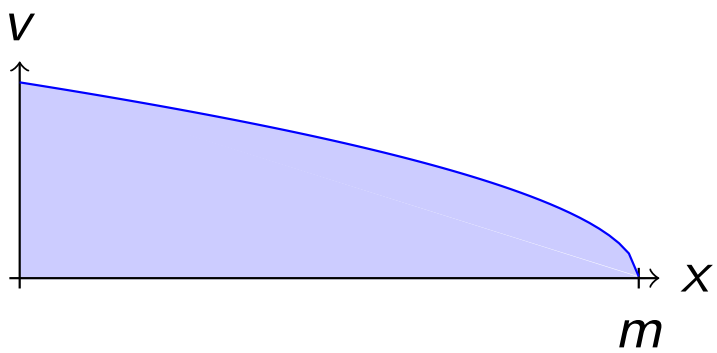
$$[;]\ \frac{}{J(x,v) \vdash \neg\text{SB} \rightarrow [a := A; (x'=v, v'=a, t'=1 \,\&\, t \leq \varepsilon)]J(x,v)}$$

$$[?]\ \frac{}{J(x,v) \vdash [?\neg\text{SB}][a := A; (x'=v, v'=a, t'=1 \,\&\, t \leq \varepsilon)]J(x,v)}$$

$$[;]\ \frac{}{J(x,v) \vdash [?\neg\text{SB}; a := A; (x'=v, v'=a, t'=1 \,\&\, t \leq \varepsilon)]J(x,v)}$$

Example Proof: Safe Accelerating

$J(x,v) \equiv v^2 \leq 2b(m-x)$

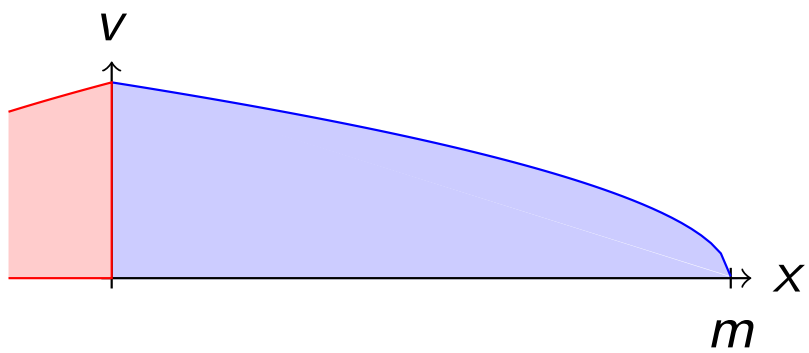


$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{J(x,v) \vdash \neg\mathrm{SB} \rightarrow (A\varepsilon+v)^2 \leq 2b(m-\frac{A}{2}\varepsilon^2-v\varepsilon-x)}\,\mathrm{QE}}{J(x,v) \vdash \neg\mathrm{SB} \rightarrow \forall t{\geq}0\,(t \leq \varepsilon \rightarrow (At+v)^2 \leq 2b(m-\frac{A}{2}t^2-vt-x))}}{J(x,v) \vdash \neg\mathrm{SB} \rightarrow \forall t{\geq}0\,(t \leq \varepsilon \rightarrow J(\frac{A}{2}t^2+vt+x, At+v))}}{J(x,v) \vdash \neg\mathrm{SB} \rightarrow \forall t{\geq}0\,(t \leq \varepsilon \rightarrow [x := \frac{A}{2}t^2+vt+x]J(x,v))}\,[:=]}{J(x,v) \vdash \neg\mathrm{SB} \rightarrow [x'=v, v'=A, t'=1 \,\&\, t \leq \varepsilon]J(x,v)}\,[']}{J(x,v) \vdash \neg\mathrm{SB} \rightarrow [a := A][x'=v, v'=a, t'=1 \,\&\, t \leq \varepsilon]J(x,v)}\,[:=]}{J(x,v) \vdash \neg\mathrm{SB} \rightarrow [a := A; (x'=v, v'=a, t'=1 \,\&\, t \leq \varepsilon)]J(x,v)}\,[;]}{J(x,v) \vdash [?\neg\mathrm{SB}][a := A; (x'=v, v'=a, t'=1 \,\&\, t \leq \varepsilon)]J(x,v)}\,[?]}{J(x,v) \vdash [?\neg\mathrm{SB}; a := A; (x'=v, v'=a, t'=1 \,\&\, t \leq \varepsilon)]J(x,v)}\,[;]$$

Example Proof: Safe Accelerating

$J(x,v) \equiv v^2 \leq 2b(m-x)$

$\text{SB} \equiv 2b(m-x) < v^2 + (A+b)(A\varepsilon^2 + 2\varepsilon v)$



$$\text{QE}\ \frac{J(x,v) \vdash \neg\text{SB} \rightarrow (A\varepsilon + v)^2 \leq 2b(m - \frac{A}{2}\varepsilon^2 - v\varepsilon - x)}{J(x,v) \vdash \neg\text{SB} \rightarrow \forall t{\geq}0\,(t \leq \varepsilon \rightarrow (At+v)^2 \leq 2b(m - \frac{A}{2}t^2 - vt - x))}$$

$$\frac{J(x,v) \vdash \neg\text{SB} \rightarrow \forall t{\geq}0\,(t \leq \varepsilon \rightarrow J(\frac{A}{2}t^2 + vt + x, At + v))}{J(x,v) \vdash \neg\text{SB} \rightarrow \forall t{\geq}0\,(t \leq \varepsilon \rightarrow [x := \frac{A}{2}t^2 + vt + x]J(x,v))}\ [:=]$$

$$\frac{J(x,v) \vdash \neg\text{SB} \rightarrow [x' = v, v' = A, t' = 1\,\&\, t \leq \varepsilon]J(x,v)}{J(x,v) \vdash \neg\text{SB} \rightarrow [x' = v, v' = A, t' = 1\,\&\, t \leq \varepsilon]J(x,v)}\ [']$$

$$\frac{J(x,v) \vdash \neg\text{SB} \rightarrow [a := A][x' = v, v' = a, t' = 1\,\&\, t \leq \varepsilon]J(x,v)}{J(x,v) \vdash \neg\text{SB} \rightarrow [a := A][x' = v, v' = a, t' = 1\,\&\, t \leq \varepsilon]J(x,v)}\ [:=]$$

$$\frac{J(x,v) \vdash \neg\text{SB} \rightarrow [a := A; (x' = v, v' = a, t' = 1\,\&\, t \leq \varepsilon)]J(x,v)}{J(x,v) \vdash \neg\text{SB} \rightarrow [a := A; (x' = v, v' = a, t' = 1\,\&\, t \leq \varepsilon)]J(x,v)}\ [;]$$

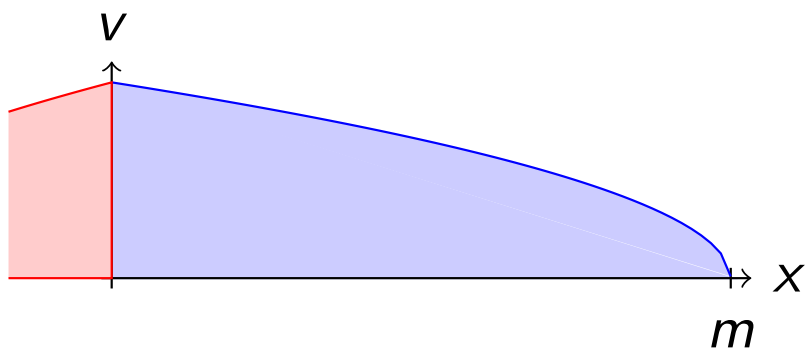
$$\frac{J(x,v) \vdash [?\neg\text{SB}][a := A; (x' = v, v' = a, t' = 1\,\&\, t \leq \varepsilon)]J(x,v)}{J(x,v) \vdash \neg\text{SB} \rightarrow [a := A; (x' = v, v' = a, t' = 1\,\&\, t \leq \varepsilon)]J(x,v)}\ [?]$$

$$\frac{J(x,v) \vdash [?\neg\text{SB}; a := A; (x' = v, v' = a, t' = 1\,\&\, t \leq \varepsilon)]J(x,v)}{J(x,v) \vdash [?\neg\text{SB}][a := A; (x' = v, v' = a, t' = 1\,\&\, t \leq \varepsilon)]J(x,v)}\ [;]$$

Example Proof: Safe Driving

$J(x,v) \equiv v^2 \leq 2b(m-x)$

$\text{SB} \equiv 2b(m-x) < v^2 + (A+b)(A\varepsilon^2 + 2\varepsilon v)$

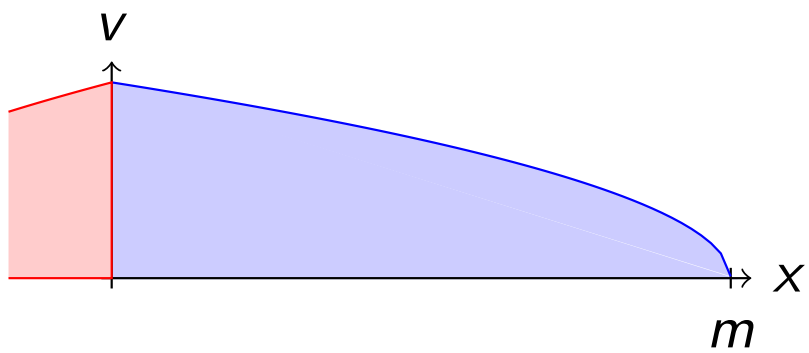


$$\text{ind}\frac{}{J(x,v) \vdash [((a := -b \cup ?\neg\text{SB}; a := A); x'' = a, t' = 1 \,\&\, t \leq \varepsilon)^*]J(x,v)}$$

Example Proof: Safe Driving

$J(x,v) \equiv v^2 \leq 2b(m-x)$

$\text{SB} \equiv 2b(m-x) < v^2 + (A+b)(A\varepsilon^2 + 2\varepsilon v)$

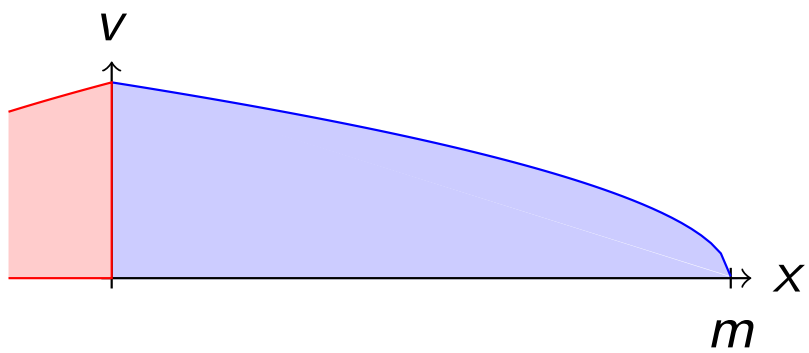


$$\text{ind}\,\frac{\text{[;]}\,\dfrac{J(x,v) \vdash [(a := -b \cup ?\neg\text{SB}; a := A); x'' = a, t' = 1 \,\&\, t \leq \varepsilon]J(x,v)}{J(x,v) \vdash [((a := -b \cup ?\neg\text{SB}; a := A); x'' = a, t' = 1 \,\&\, t \leq \varepsilon)^*]J(x,v)}$$

Example Proof: Safe Driving

$J(x,v) \equiv v^2 \leq 2b(m-x)$

$\text{SB} \equiv 2b(m-x) < v^2 + (A+b)(A\varepsilon^2 + 2\varepsilon v)$

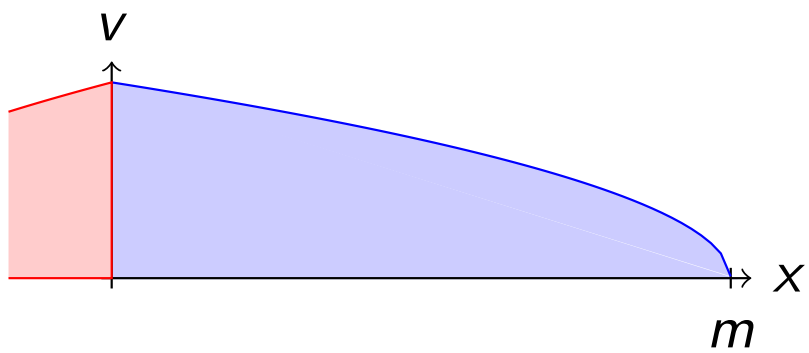


$$\dfrac{\dfrac{\dfrac{}{J(x,v) \vdash [a := -b \cup ?\neg\text{SB}; a := A][x'' = a, t' = 1 \,\&\, t \leq \varepsilon]J(x,v)}\,[\cup]}{J(x,v) \vdash [(a := -b \cup ?\neg\text{SB}; a := A); x'' = a, t' = 1 \,\&\, t \leq \varepsilon]J(x,v)}\,[;]}{J(x,v) \vdash [((a := -b \cup ?\neg\text{SB}; a := A); x'' = a, t' = 1 \,\&\, t \leq \varepsilon)^*]J(x,v)}\,\text{ind}$$

Example Proof: Safe Driving

$J(x,v) \equiv v^2 \leq 2b(m-x)$

$\text{SB} \equiv 2b(m-x) < v^2 + (A+b)(A\varepsilon^2 + 2\varepsilon v)$

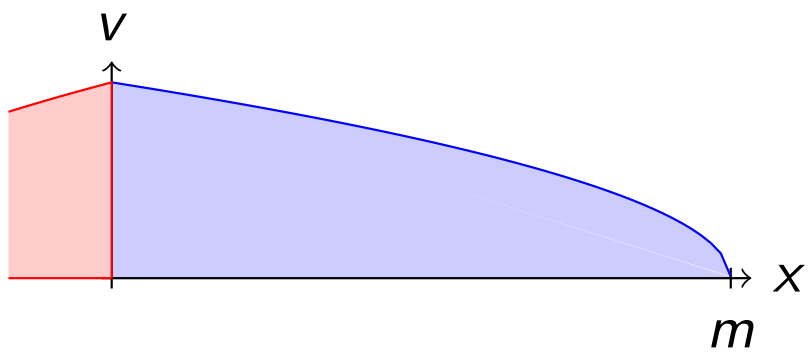


$$\dfrac{\dfrac{\dfrac{\dfrac{}{J(x,v) \vdash [a := -b][x'' = a\,..]J(x,v) \land [?\neg \text{SB}; a := A][x'' = a\,..]J(x,v)}}{J(x,v) \vdash [a := -b \cup ?\neg \text{SB}; a := A][x'' = a, t' = 1 \,\&\, t \leq \varepsilon]J(x,v)}[\cup]}{J(x,v) \vdash [(a := -b \cup ?\neg \text{SB}; a := A); x'' = a, t' = 1 \,\&\, t \leq \varepsilon]J(x,v)}[;]}{J(x,v) \vdash [\big((a := -b \cup ?\neg \text{SB}; a := A); x'' = a, t' = 1 \,\&\, t \leq \varepsilon\big)^*]J(x,v)}\text{ind}$$

Example Proof: Safe Driving

$J(x,v) \equiv v^2 \leq 2b(m-x)$

$\text{SB} \equiv 2b(m-x) < v^2 + (A+b)(A\varepsilon^2 + 2\varepsilon v)$



$$\text{QE}\,\frac{\text{previous proofs for braking and acceleration}}{J(x,v) \vdash [a := -b][x'' = a\,..]J(x,v) \land [?\neg\text{SB}; a := A][x'' = a\,..]J(x,v)}$$

$$[\cup]\,\frac{J(x,v) \vdash [a := -b \cup ?\neg\text{SB}; a := A][x'' = a, t' = 1 \,\&\, t \leq \varepsilon]J(x,v)}{J(x,v) \vdash [(a := -b \cup ?\neg\text{SB}; a := A); x'' = a, t' = 1 \,\&\, t \leq \varepsilon]J(x,v)}$$

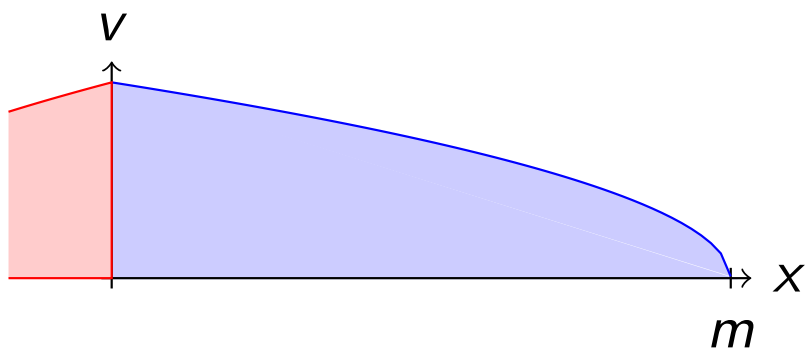
$$[;]\,\frac{J(x,v) \vdash [(a := -b \cup ?\neg\text{SB}; a := A); x'' = a, t' = 1 \,\&\, t \leq \varepsilon]J(x,v)}{J(x,v) \vdash [\big((a := -b \cup ?\neg\text{SB}; a := A); x'' = a, t' = 1 \,\&\, t \leq \varepsilon\big)^*]J(x,v)}$$

$$\text{ind}\,\frac{J(x,v) \vdash [\big((a := -b \cup ?\neg\text{SB}; a := A); x'' = a, t' = 1 \,\&\, t \leq \varepsilon\big)^*]J(x,v)}{}$$

Example Proof: Safe Driving

$J(x,v) \equiv v^2 \leq 2b(m-x)$

$\text{SB} \equiv 2b(m-x) < v^2 + (A+b)(A\varepsilon^2 + 2\varepsilon v)$



$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{*}{\text{previous proofs for braking and acceleration}}\,\text{QE}}{J(x,v) \vdash [a := -b][x'' = a\,..]J(x,v) \wedge [?\neg\text{SB}; a := A][x'' = a\,..]J(x,v)}}{J(x,v) \vdash [a := -b \cup ?\neg\text{SB}; a := A][x'' = a, t' = 1 \,\&\, t \leq \varepsilon]J(x,v)}\,[\cup]}{J(x,v) \vdash [(a := -b \cup ?\neg\text{SB}; a := A); x'' = a, t' = 1 \,\&\, t \leq \varepsilon]J(x,v)}\,[;]}{J(x,v) \vdash [\big((a := -b \cup ?\neg\text{SB}; a := A); x'' = a, t' = 1 \,\&\, t \leq \varepsilon\big)^*]J(x,v)}\,\text{ind}$$

1. Proof is deterministic "follow your nose".
2. Synthesize invariant $J(x,v)$ and parameter constraint SB.
3. $J(x,v)$ is a predicate symbol to prove only once and instantiate later.
4. First looking at proofs of smaller pieces is often effective.

Outline

Differential Dynamic Logic: Axiomatization

equations of truth

[:=] $[x := e]P(x) \leftrightarrow P(e)$

[?] $[?Q]P \leftrightarrow (Q \rightarrow P)$

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

[∪] $[\alpha \cup \beta]P \leftrightarrow [\alpha]P \wedge [\beta]P$

[;] $[\alpha; \beta]P \leftrightarrow [\alpha][\beta]P$

[*] $[\alpha^*]P \leftrightarrow P \wedge [\alpha][\alpha^*]P$

K $[\alpha](P \rightarrow Q) \rightarrow ([\alpha]P \rightarrow [\alpha]Q)$

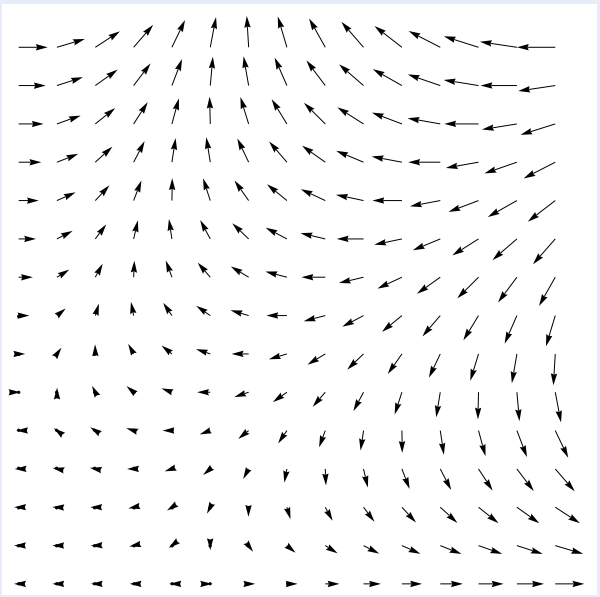
I $[\alpha^*](P \rightarrow [\alpha]P) \rightarrow (P \rightarrow [\alpha^*]P)$

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

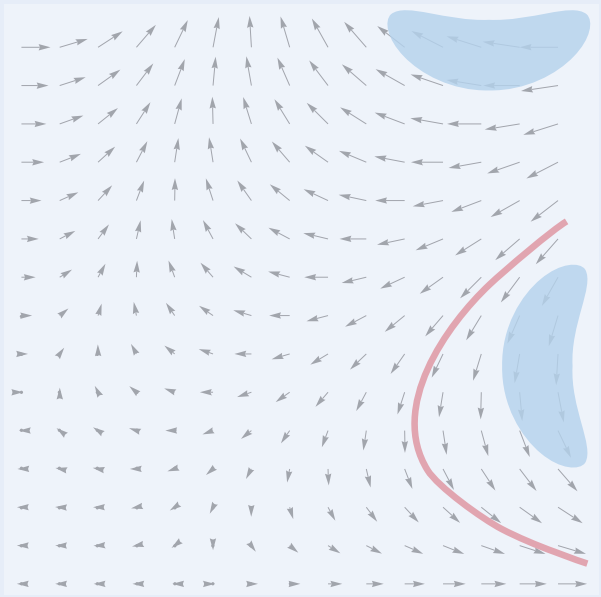
LICS'12,JAR'16

Differential Invariants for Differential Equations

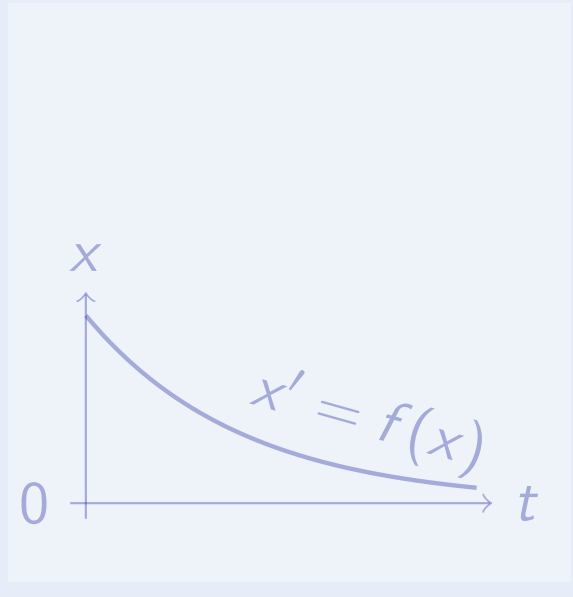
Differential Invariant

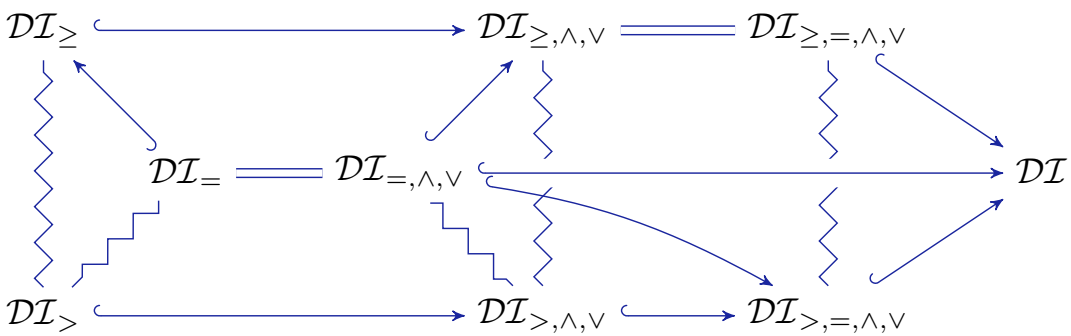


Differential Cut



Differential Ghost





Logic

Provability theory

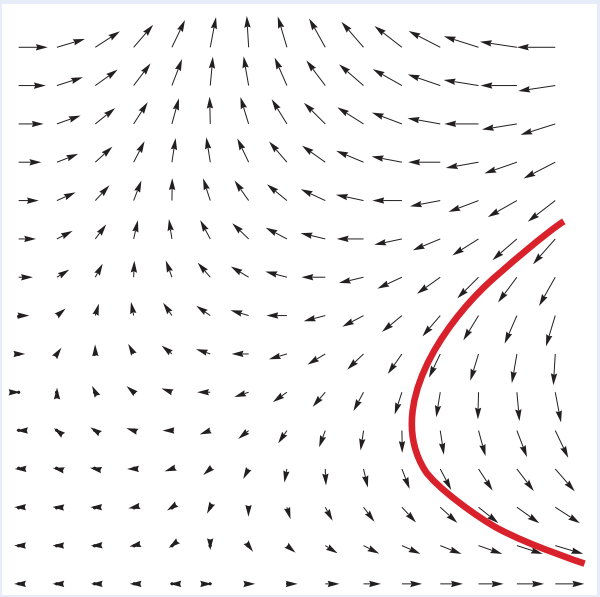
Math

Character-istic PDE

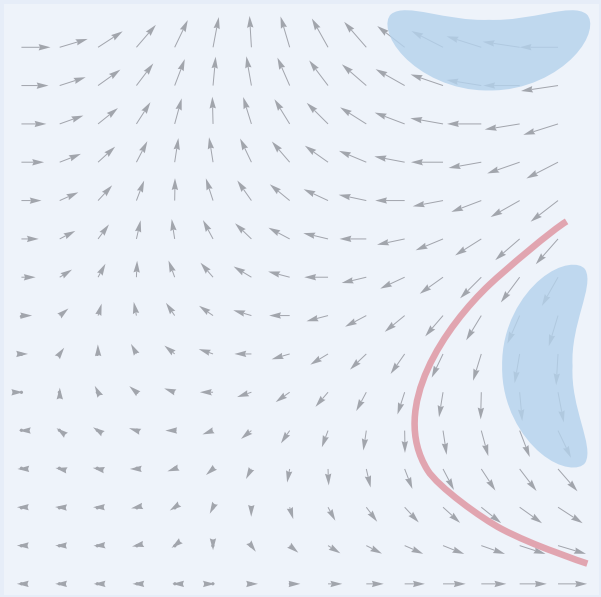
JLogComput'10,CAV'08,FMSD'09,LMCS'12,LICS'12,ITP'12,JAR'16

Differential Invariants for Differential Equations

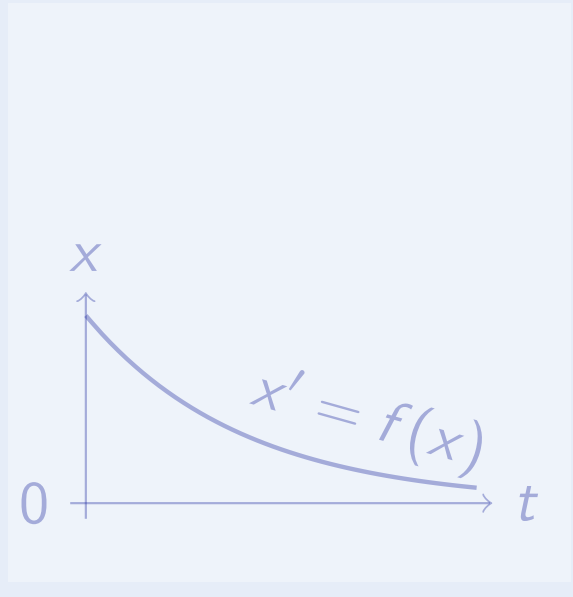
Differential Invariant

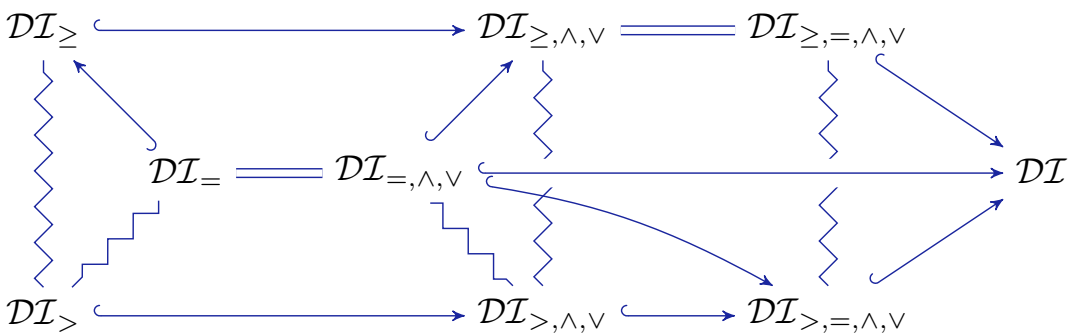


Differential Cut



Differential Ghost





Logic

Provability theory

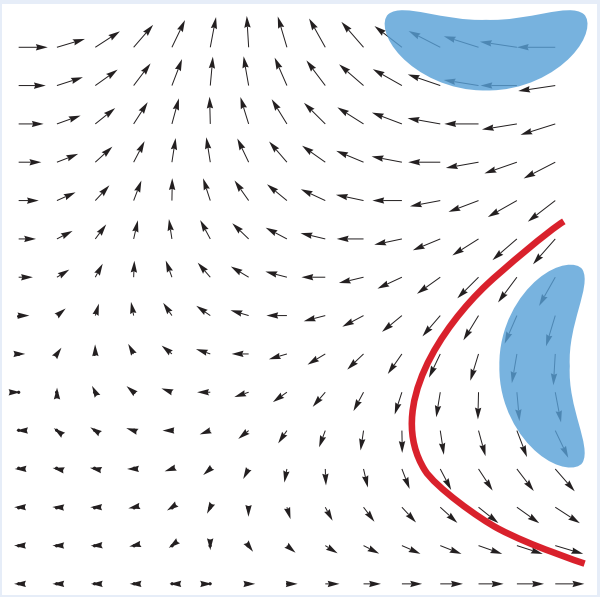
Math

Character-istic PDE

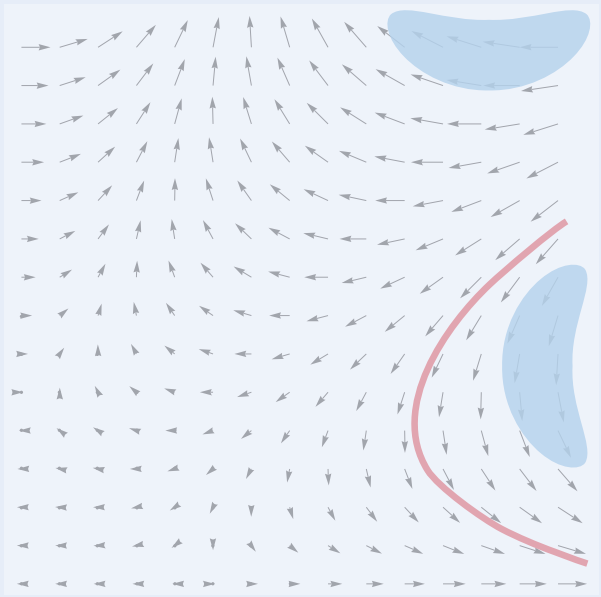
JLogComput'10,CAV'08,FMSD'09,LMCS'12,LICS'12,ITP'12,JAR'16

Differential Invariants for Differential Equations

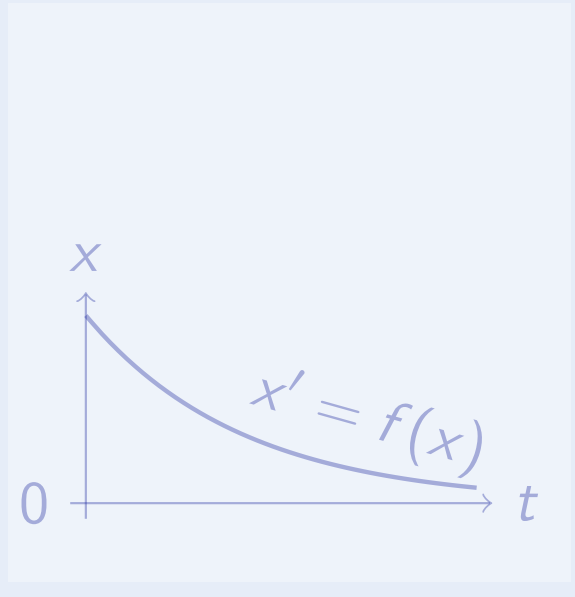
Differential Invariant

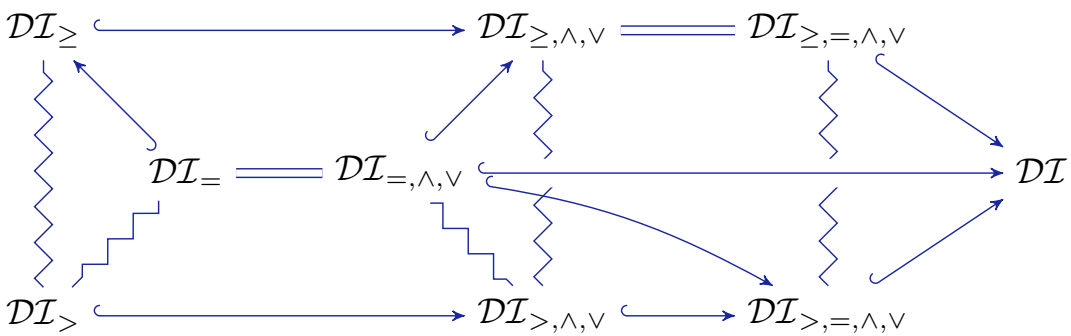


Differential Cut



Differential Ghost





Logic

Provability theory

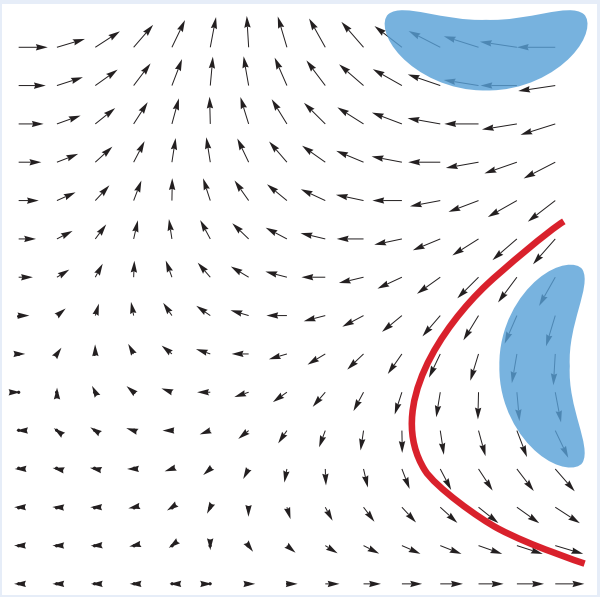
Math

Character-istic PDE

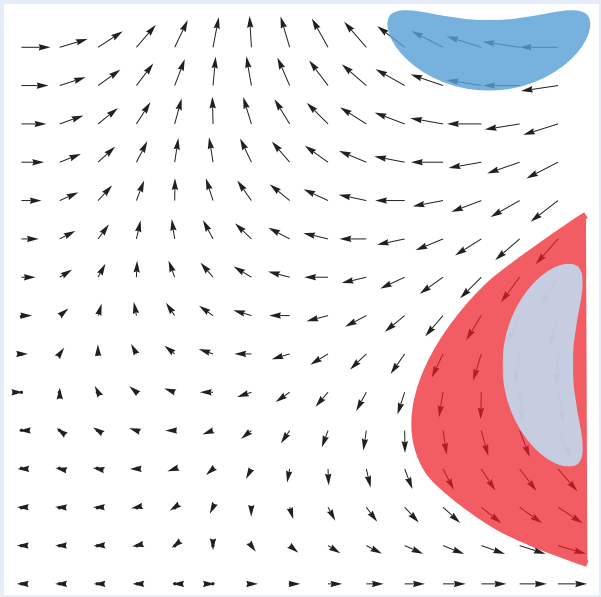
JLogComput'10,CAV'08,FMSD'09,LMCS'12,LICS'12,ITP'12,JAR'16

Differential Invariants for Differential Equations

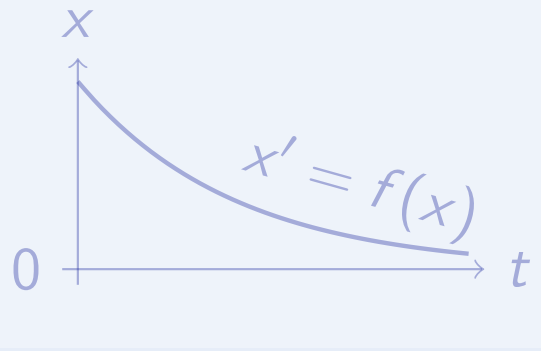
Differential Invariant

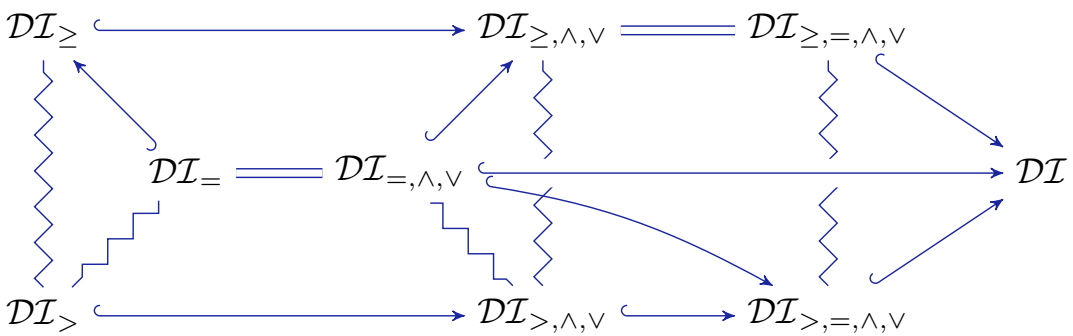


Differential Cut



Differential Ghost





Logic

Provability theory

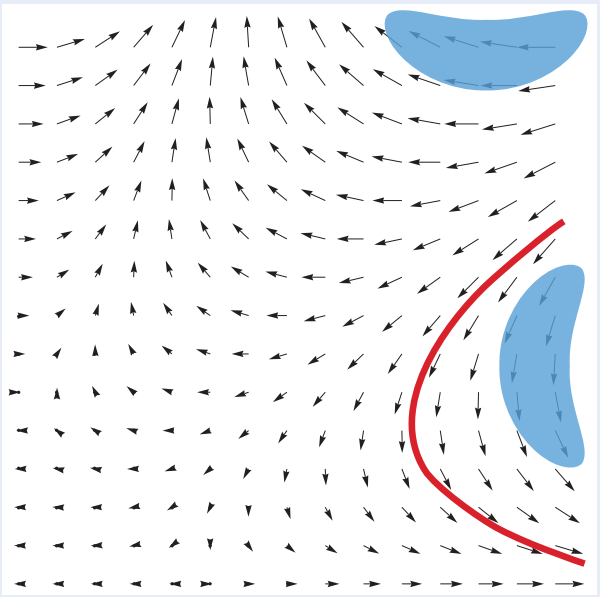
Math

Character-istic PDE

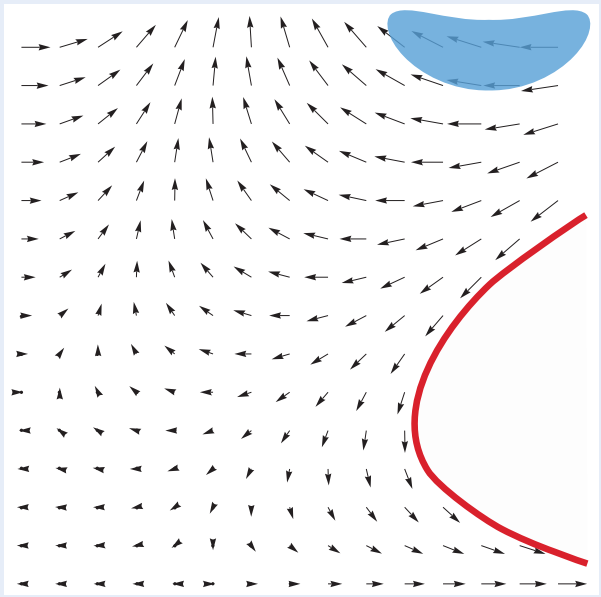
JLogComput'10,CAV'08,FMSD'09,LMCS'12,LICS'12,ITP'12,JAR'16

Differential Invariants for Differential Equations

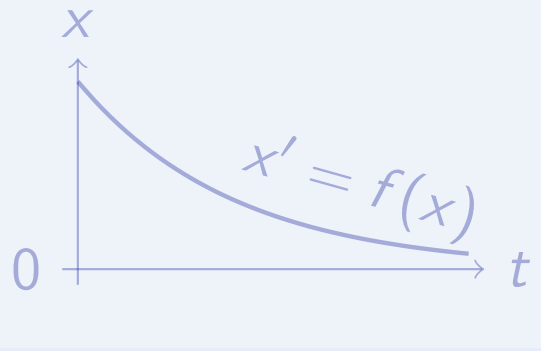
Differential Invariant

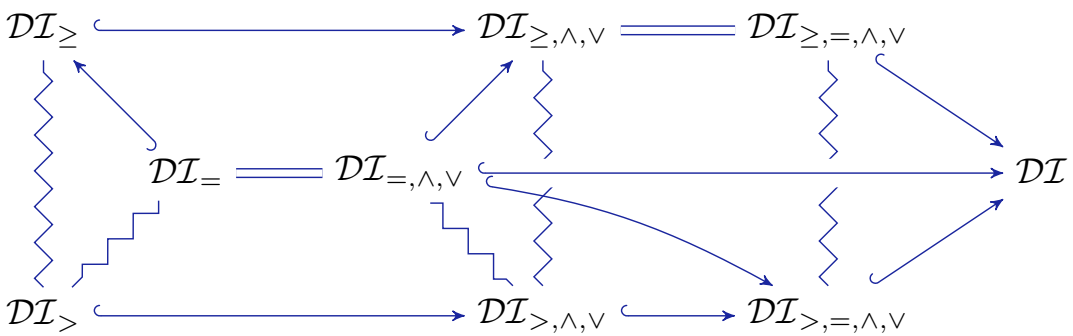


Differential Cut



Differential Ghost





Logic

Provability theory

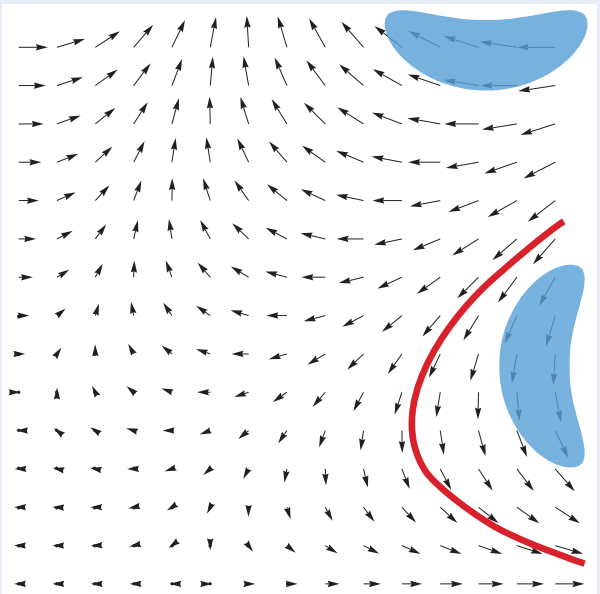
Math

Character-istic PDE

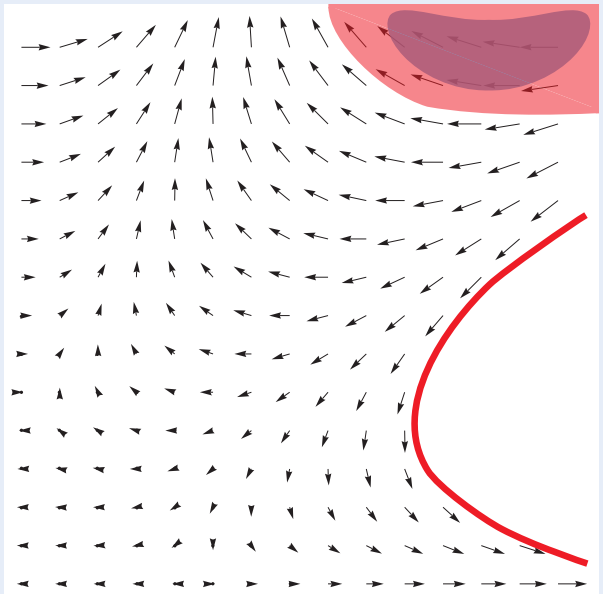
JLogComput'10,CAV'08,FMSD'09,LMCS'12,LICS'12,ITP'12,JAR'16

Differential Invariants for Differential Equations

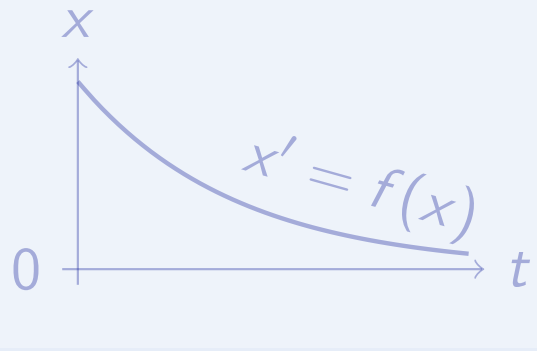
Differential Invariant

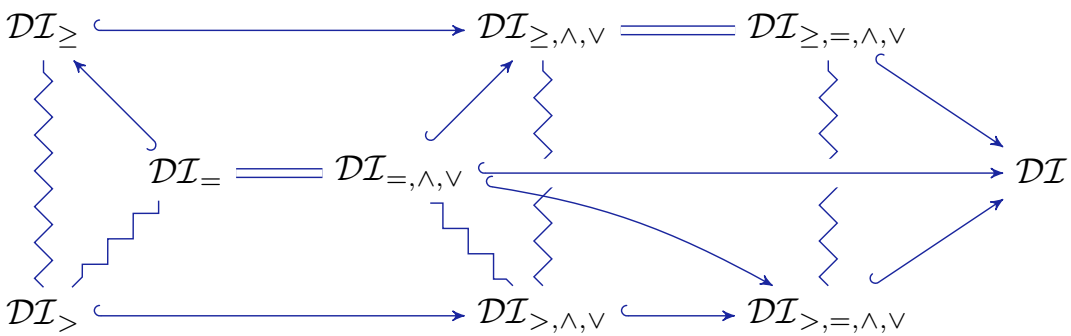


Differential Cut



Differential Ghost





Logic

Provability theory

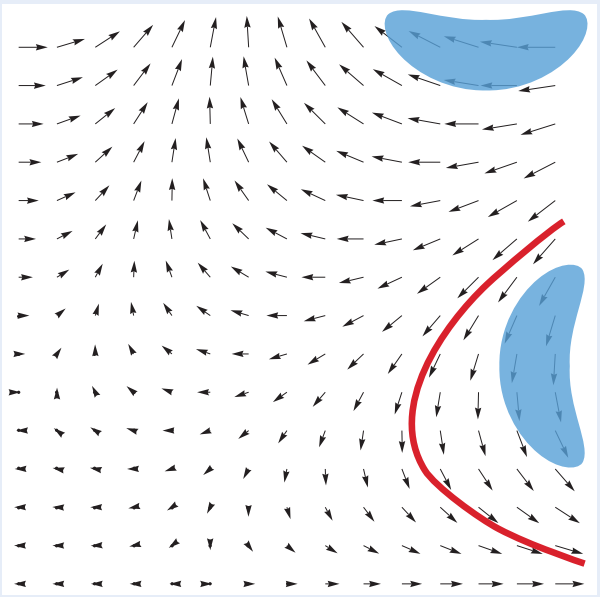
Math

Character-istic PDE

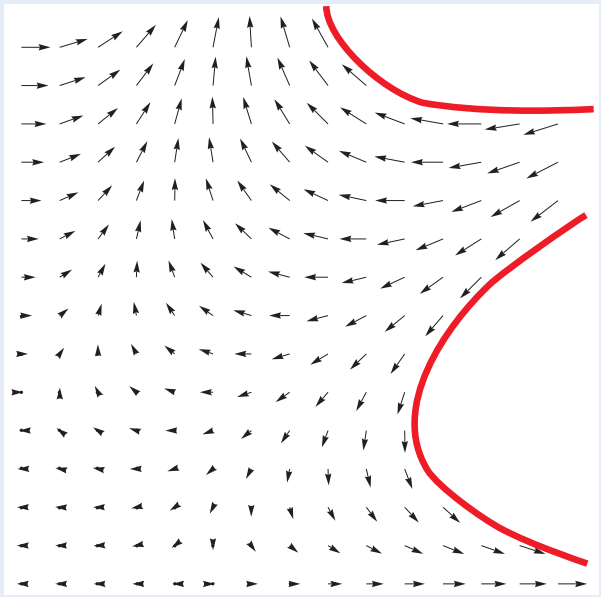
JLogComput'10,CAV'08,FMSD'09,LMCS'12,LICS'12,ITP'12,JAR'16

Differential Invariants for Differential Equations

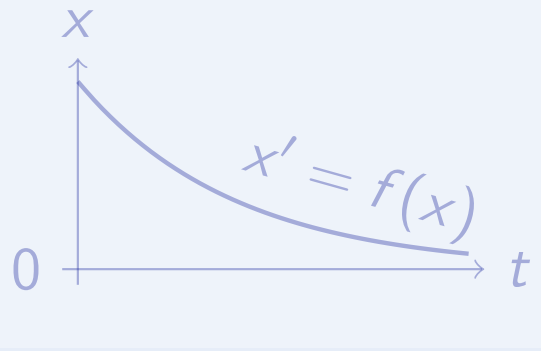
Differential Invariant

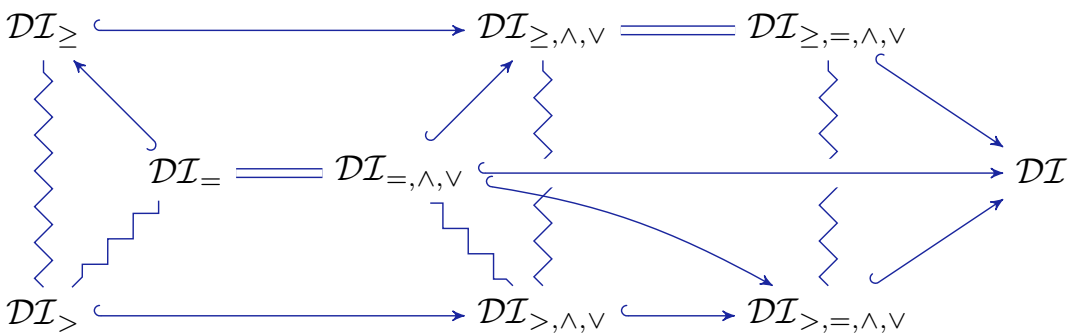


Differential Cut



Differential Ghost





Logic

Provability theory

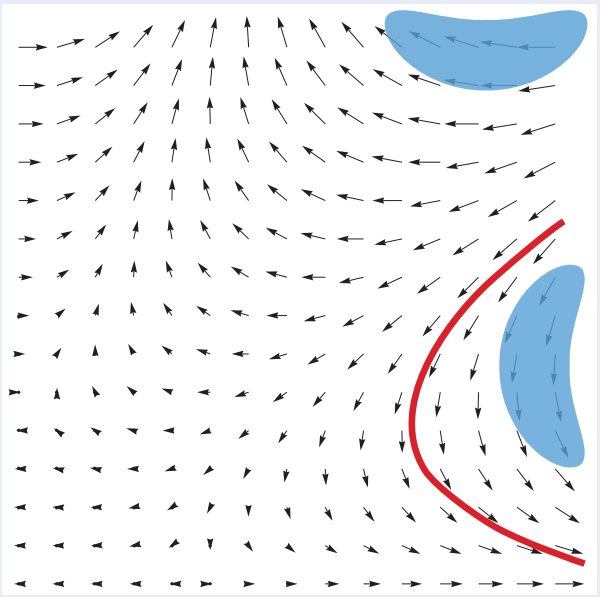
Math

Character-istic PDE

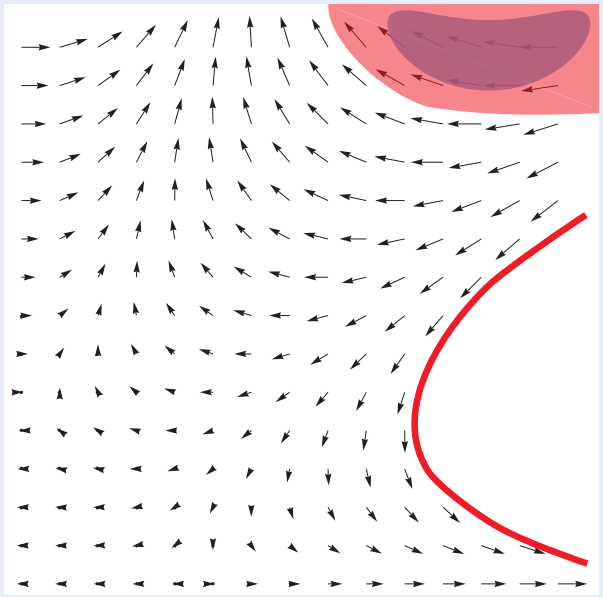
JLogComput'10,CAV'08,FMSD'09,LMCS'12,LICS'12,ITP'12,JAR'16

Differential Invariants for Differential Equations

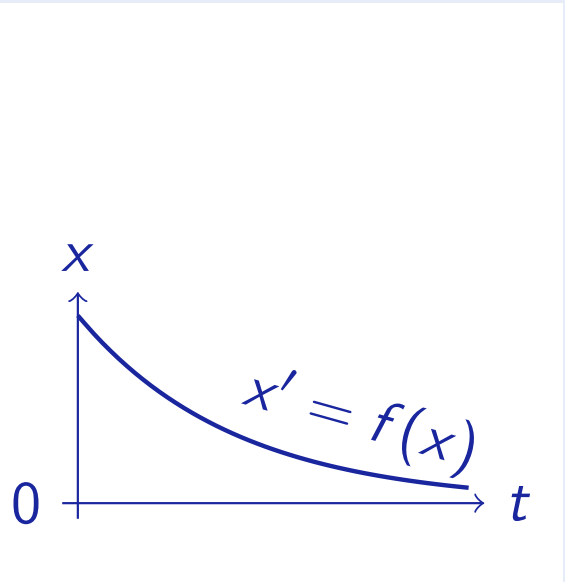
Differential Invariant

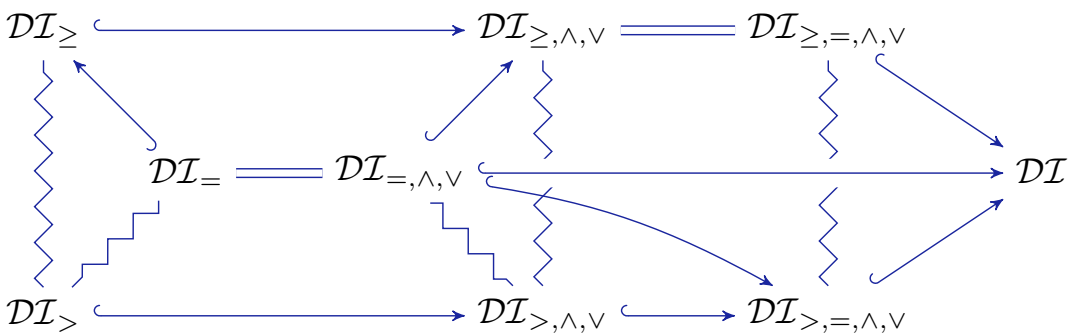


Differential Cut



Differential Ghost





Logic

Provability theory

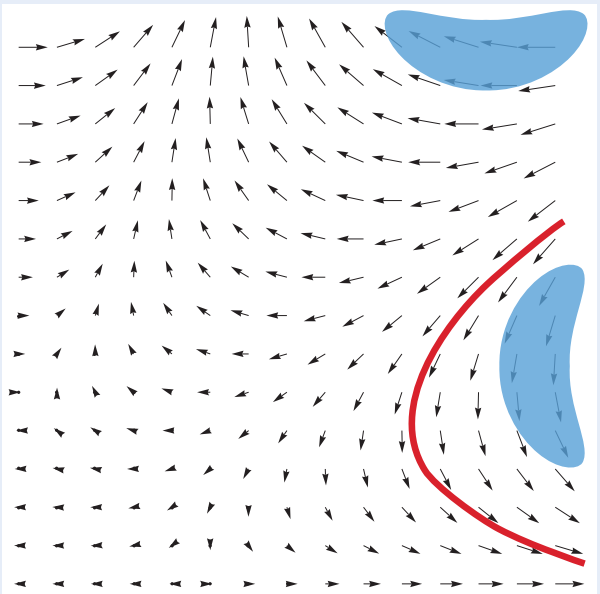
Math

Character-istic PDE

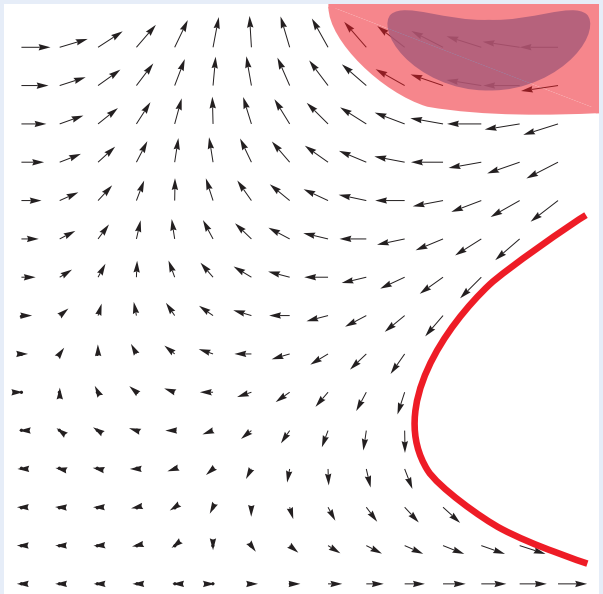
JLogComput'10,CAV'08,FMSD'09,LMCS'12,LICS'12,ITP'12,JAR'16

Differential Invariants for Differential Equations

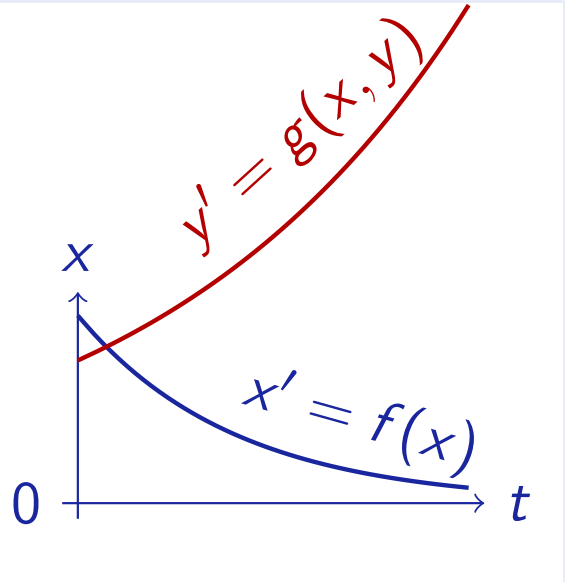
Differential Invariant

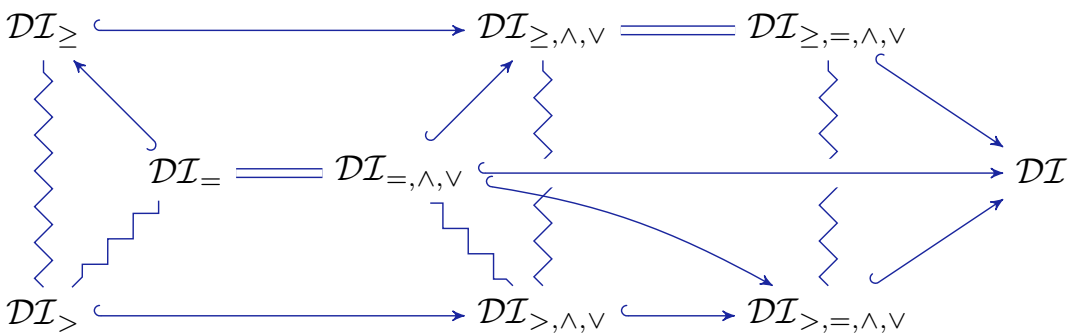


Differential Cut



Differential Ghost





Logic

Provability theory

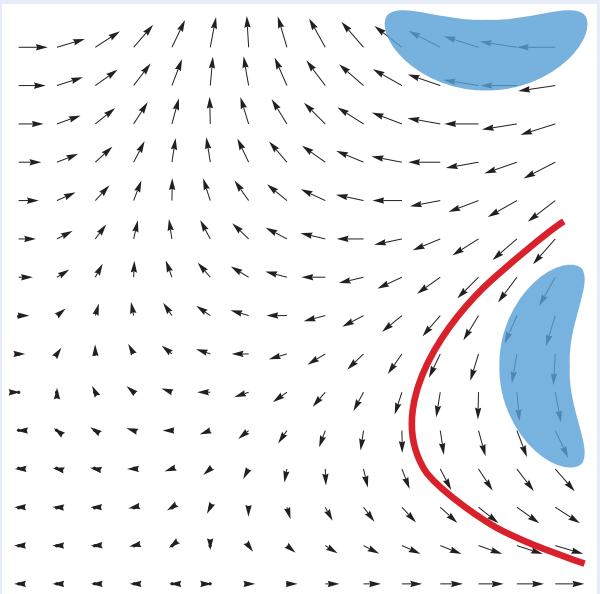
Math

Character-istic PDE

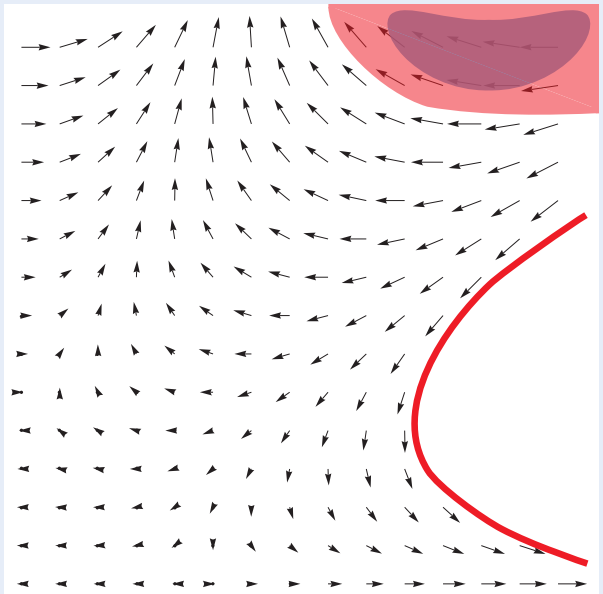
JLogComput'10,CAV'08,FMSD'09,LMCS'12,LICS'12,ITP'12,JAR'16

Differential Invariants for Differential Equations

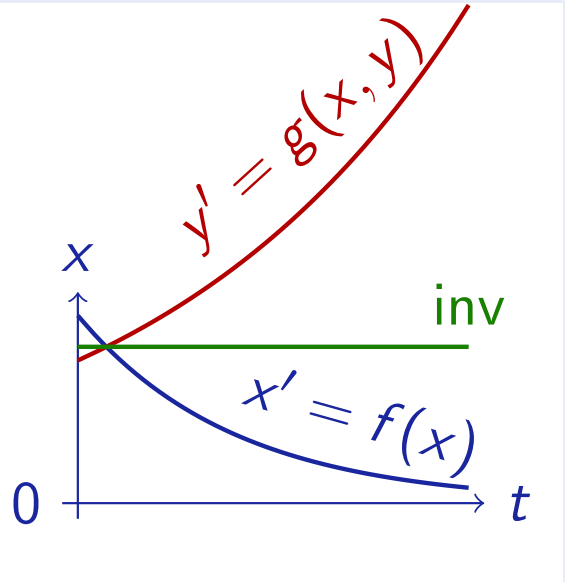
Differential Invariant

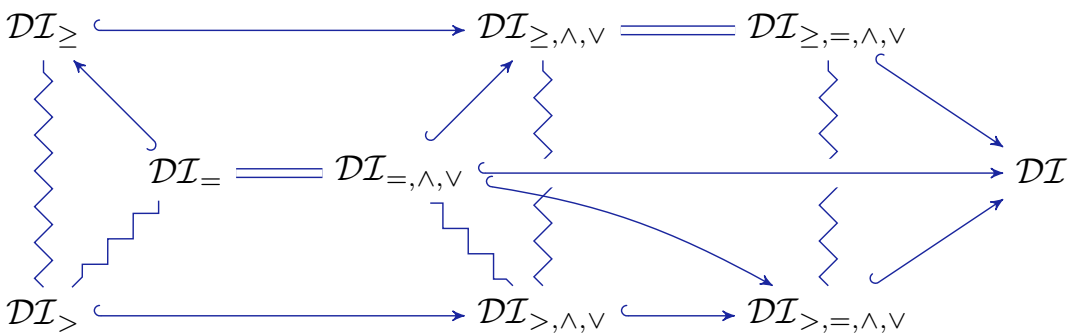


Differential Cut



Differential Ghost





Logic

Provability theory

Math

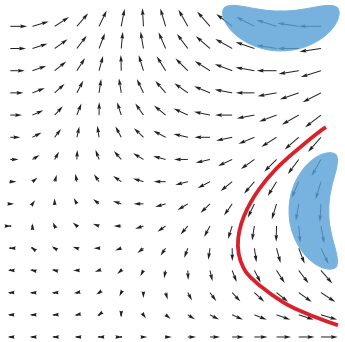
Character-istic PDE

JLogComput'10,CAV'08,FMSD'09,LMCS'12,LICS'12,ITP'12,JAR'16

Differential Invariants for Differential Equations

Differential Invariant

$$\frac{Q \vdash [x' := f(x)](F)'}{F \vdash [x' = f(x) \,\&\, Q]F}$$



JLogComput'10,LMCS'12, LICS'12,JAR'16

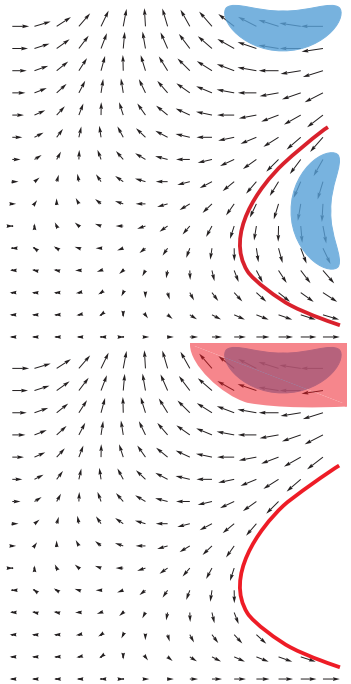
Differential Invariants for Differential Equations

Differential Invariant

$$\frac{Q \vdash [x' := f(x)](F)'}{F \vdash [x' = f(x) \,\&\, Q]F}$$

Differential Cut

$$\frac{F \vdash [x' = f(x) \,\&\, Q]C \quad F \vdash [x' = f(x) \,\&\, Q \wedge C]F}{F \vdash [x' = f(x) \,\&\, Q]F}$$



JLogComput'10,LMCS'12, LICS'12,JAR'16

Differential Invariants for Differential Equations

Differential Invariant

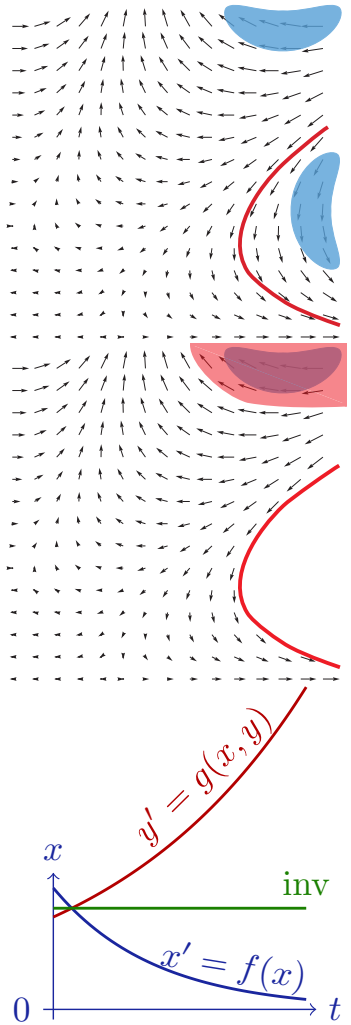
$$\frac{Q \vdash [x' := f(x)](F)'}{F \vdash [x' = f(x) \,\&\, Q]F}$$

Differential Cut

$$\frac{F \vdash [x' = f(x) \,\&\, Q]C \quad F \vdash [x' = f(x) \,\&\, Q \wedge C]F}{F \vdash [x' = f(x) \,\&\, Q]F}$$

Differential Ghost

$$\frac{F \leftrightarrow \exists y\, G \quad G \vdash [x' = f(x), y' = g(x,y) \,\&\, Q]G}{F \vdash [x' = f(x) \,\&\, Q]F}$$



JLogComput'10,LMCS'12, LICS'12,JAR'16

Differential Invariants for Differential Equations

Differential Invariant

$$\frac{Q \vdash [x' := f(x)](F)'}{F \vdash [x' = f(x) \,\&\, Q]F}$$

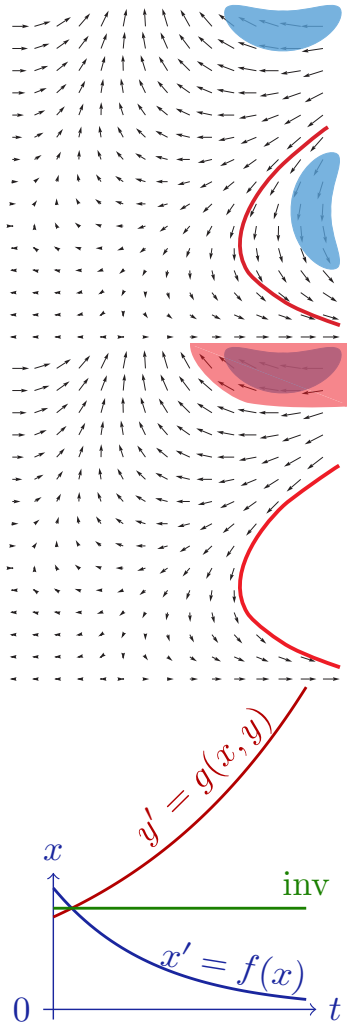
Differential Cut

$$\frac{F \vdash [x' = f(x) \,\&\, Q]C \quad F \vdash [x' = f(x) \,\&\, Q \wedge C]F}{F \vdash [x' = f(x) \,\&\, Q]F}$$

Differential Ghost

$$\frac{F \leftrightarrow \exists y\, G \quad G \vdash [x' = f(x), y' = g(x,y) \,\&\, Q]G}{F \vdash [x' = f(x) \,\&\, Q]F}$$

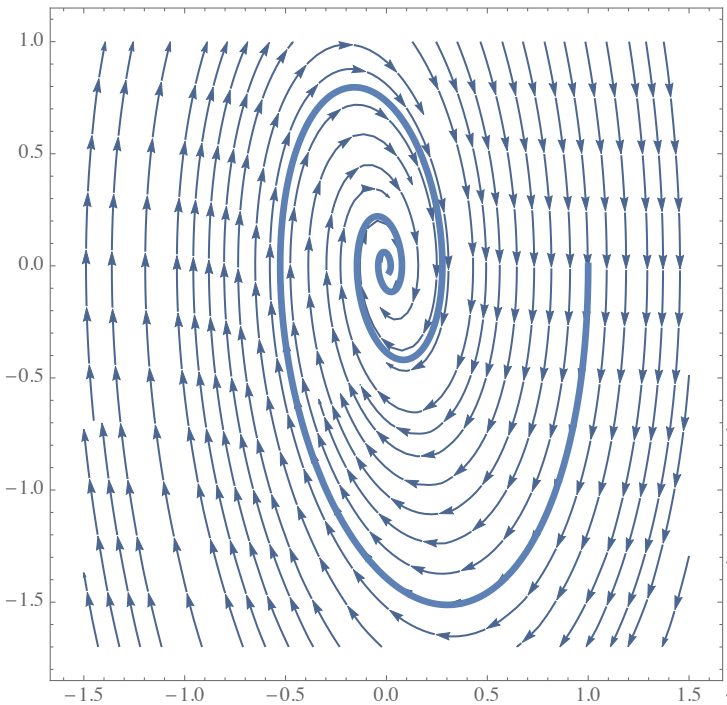
if new $y' = g(x,y)$ has a global solution

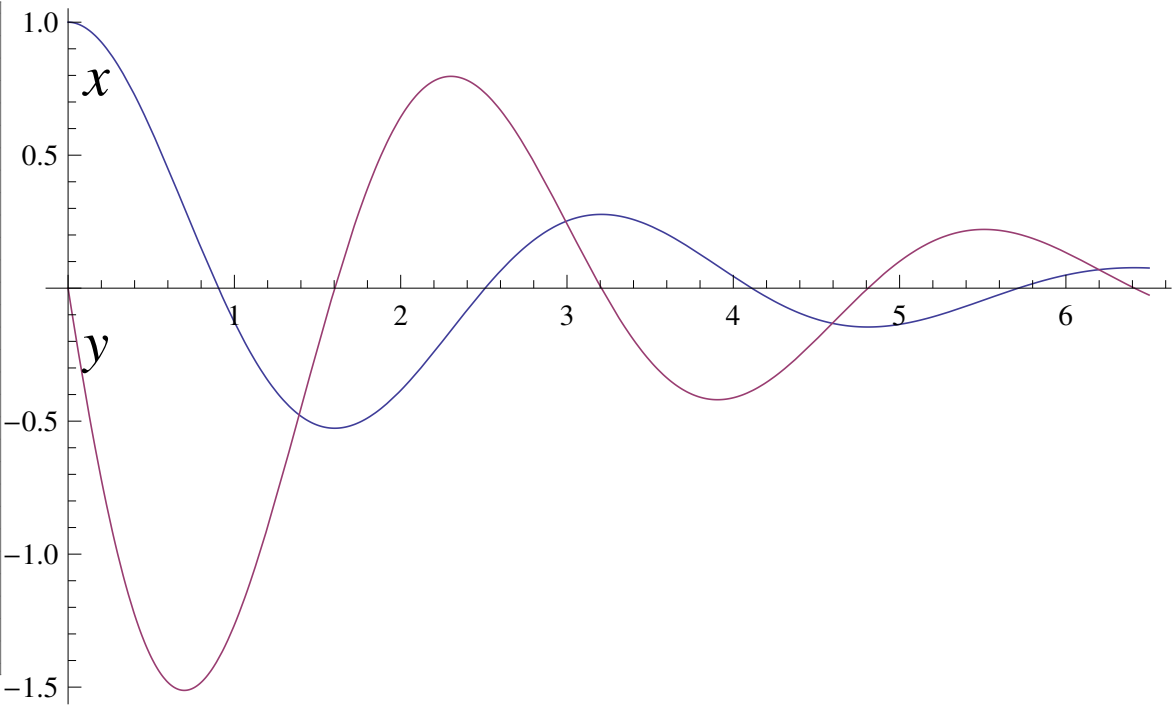


JLogComput'10,LMCS'12, LICS'12,JAR'16

Differential Invariants for Differential Equations

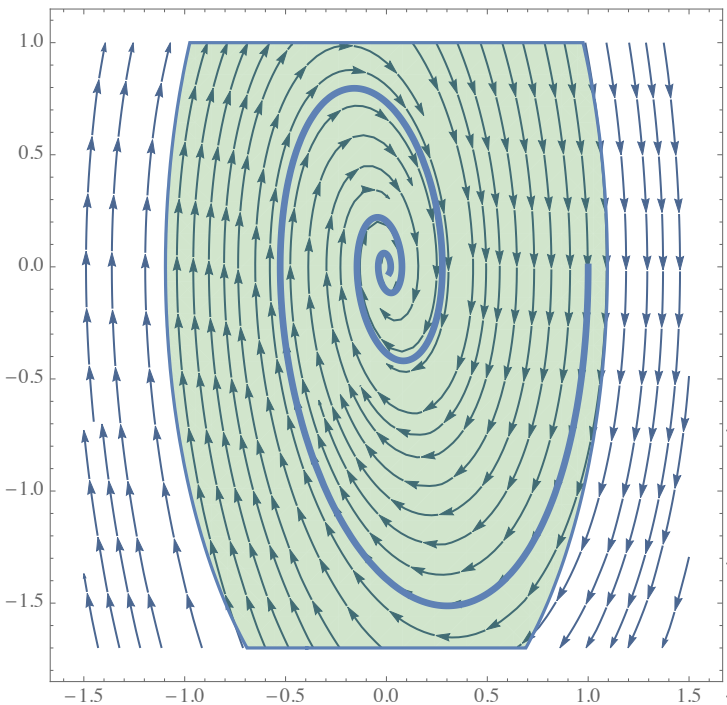
$$\omega^2x^2+y^2\leq c^2 \vdash [x'=y, y'=-\omega^2x-2d\omega y \,\&\, \omega\geq 0 \wedge d\geq 0]\, \omega^2x^2+y^2\leq c^2$$

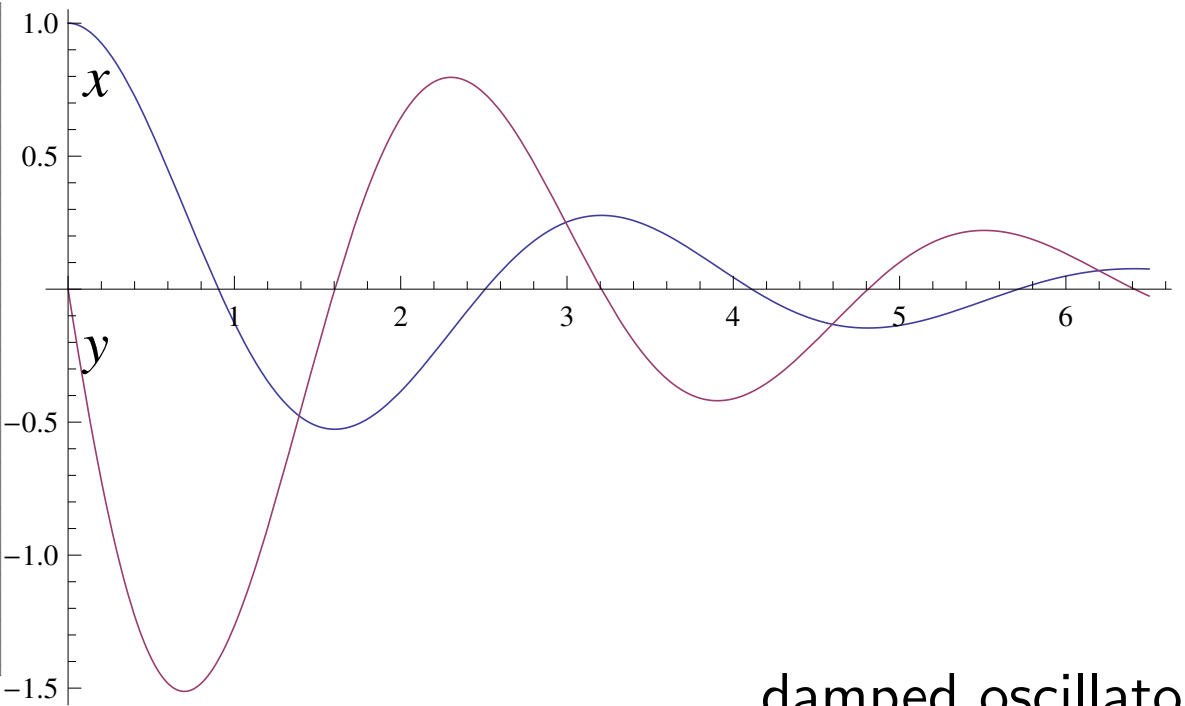




Differential Invariants for Differential Equations

$$\omega^2 x^2 + y^2 \leq c^2 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y \,\&\, \omega \geq 0 \wedge d \geq 0]\, \omega^2 x^2 + y^2 \leq c^2$$

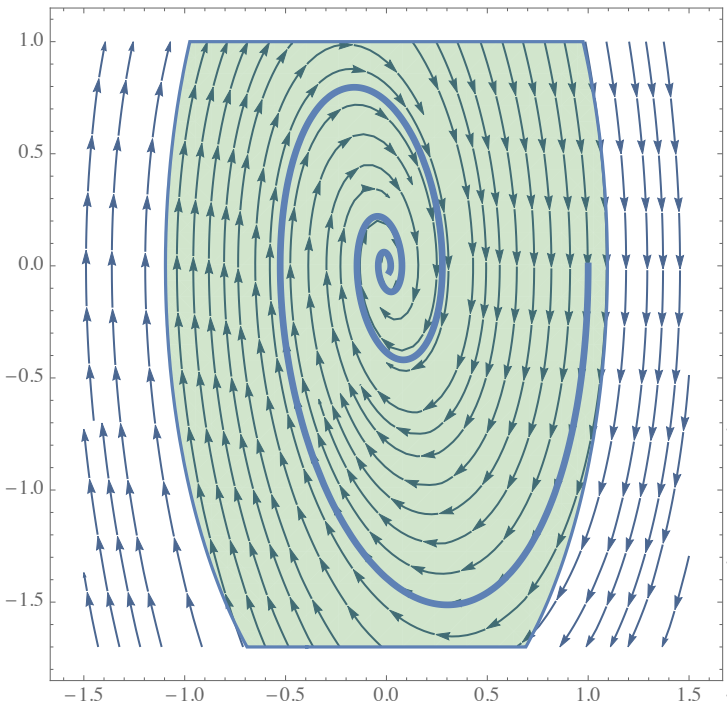


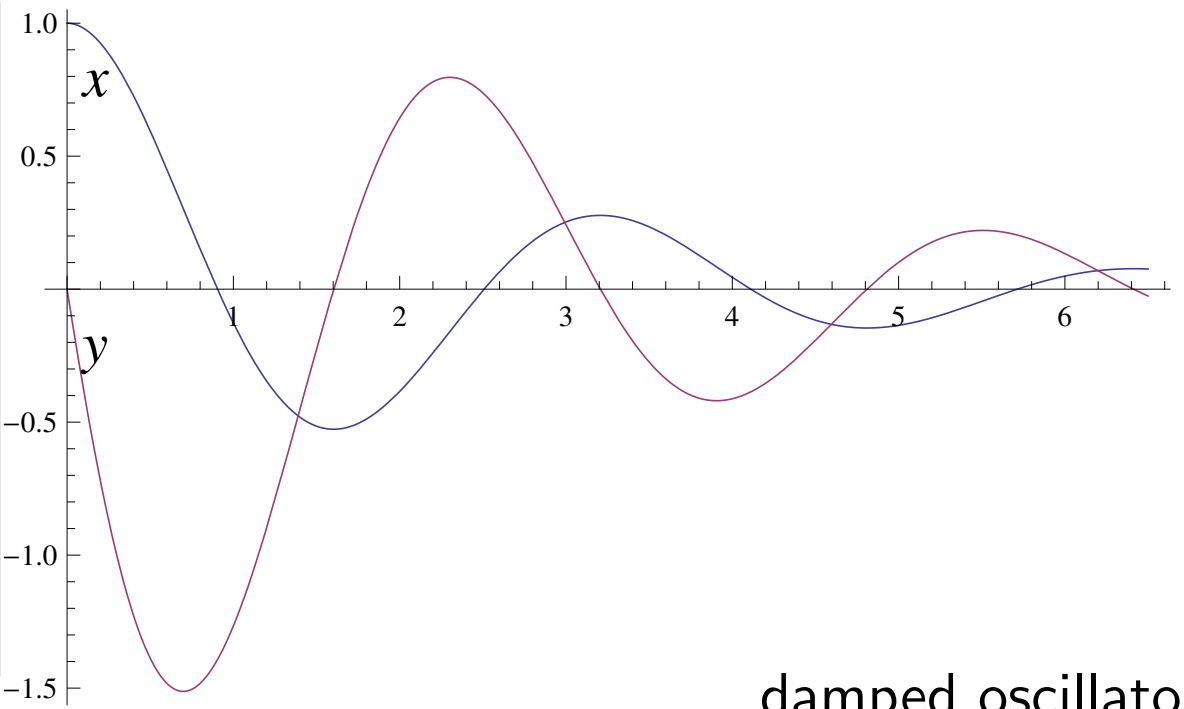


damped oscillator

Differential Invariants for Differential Equations

$$\frac{\omega{\geq}0 \land d{\geq}0 \vdash [x':=y][y':=-\omega^2 x - 2d\omega y]\, 2\omega^2 x x' + 2y y' \leq 0}{\omega^2 x^2 + y^2 \leq c^2 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y \,\&\, \omega{\geq}0 \land d{\geq}0]\, \omega^2 x^2 + y^2 \leq c^2}$$





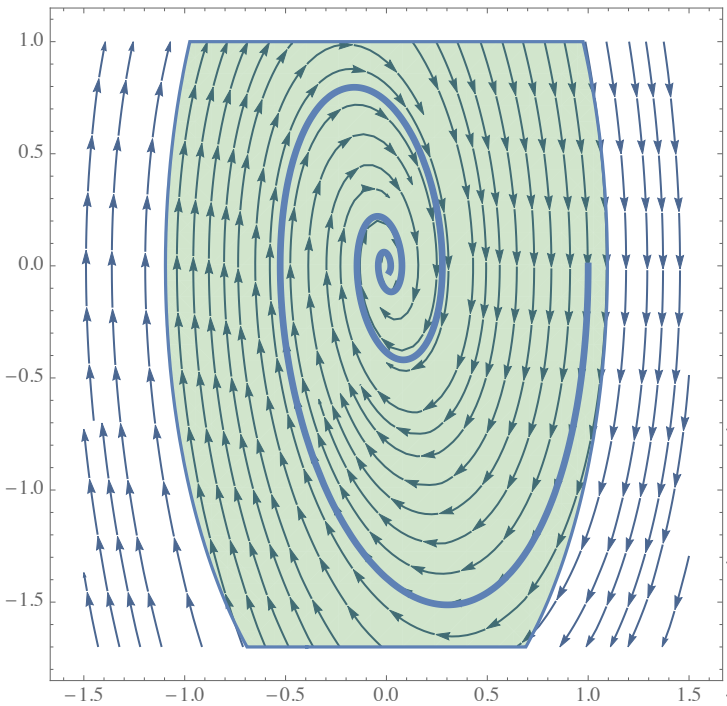
damped oscillator

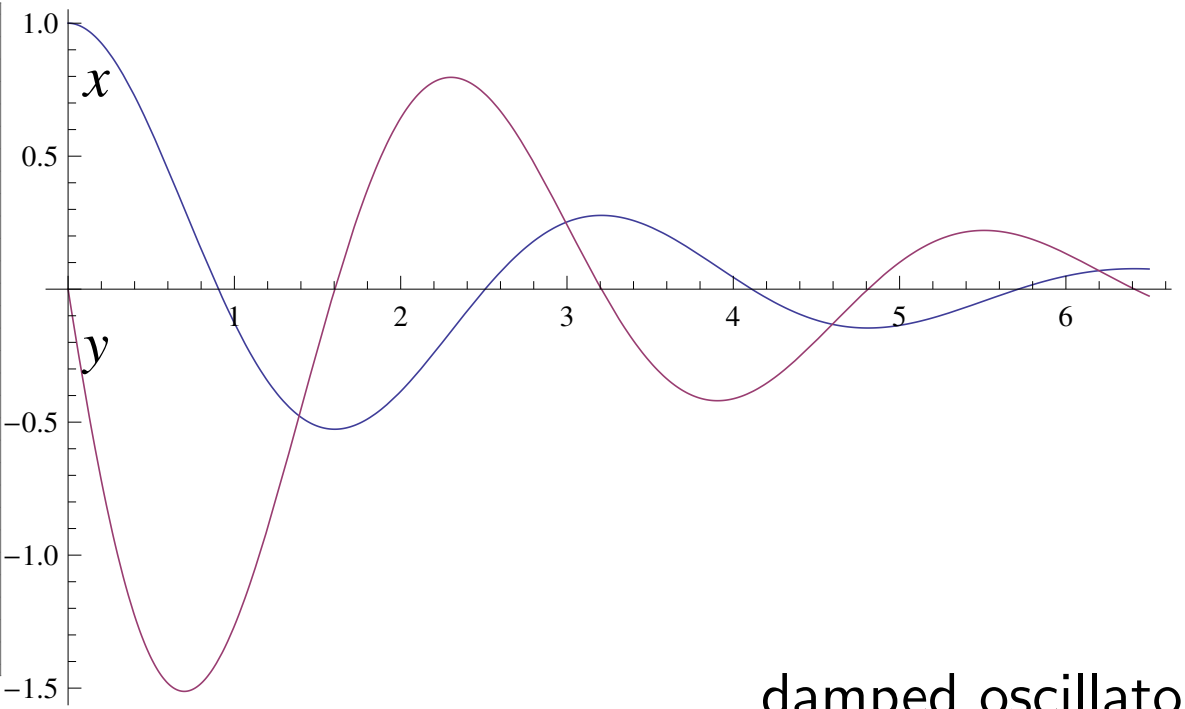
Differential Invariants for Differential Equations

$$\omega{\geq}0 \land d{\geq}0 \vdash 2\omega^2 xy + 2y(-\omega^2 x - 2d\omega y) \leq 0$$

$$\omega{\geq}0 \land d{\geq}0 \vdash [x':=y][y':=-\omega^2 x - 2d\omega y]\, 2\omega^2 xx' + 2yy' \leq 0$$

$$\omega^2x^2{+}y^2{\leq}c^2 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y \,\&\, \omega{\geq}0 \land d{\geq}0]\, \omega^2x^2{+}y^2{\leq}c^2$$

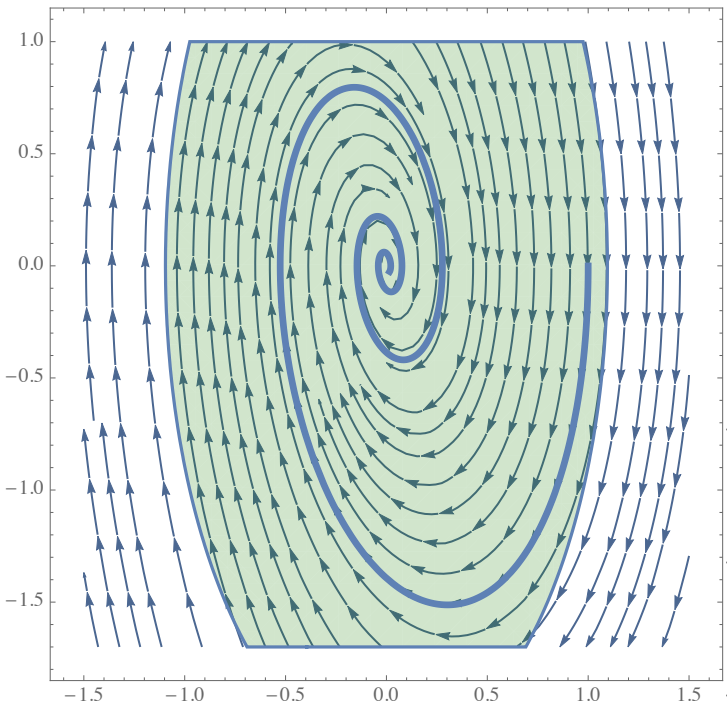


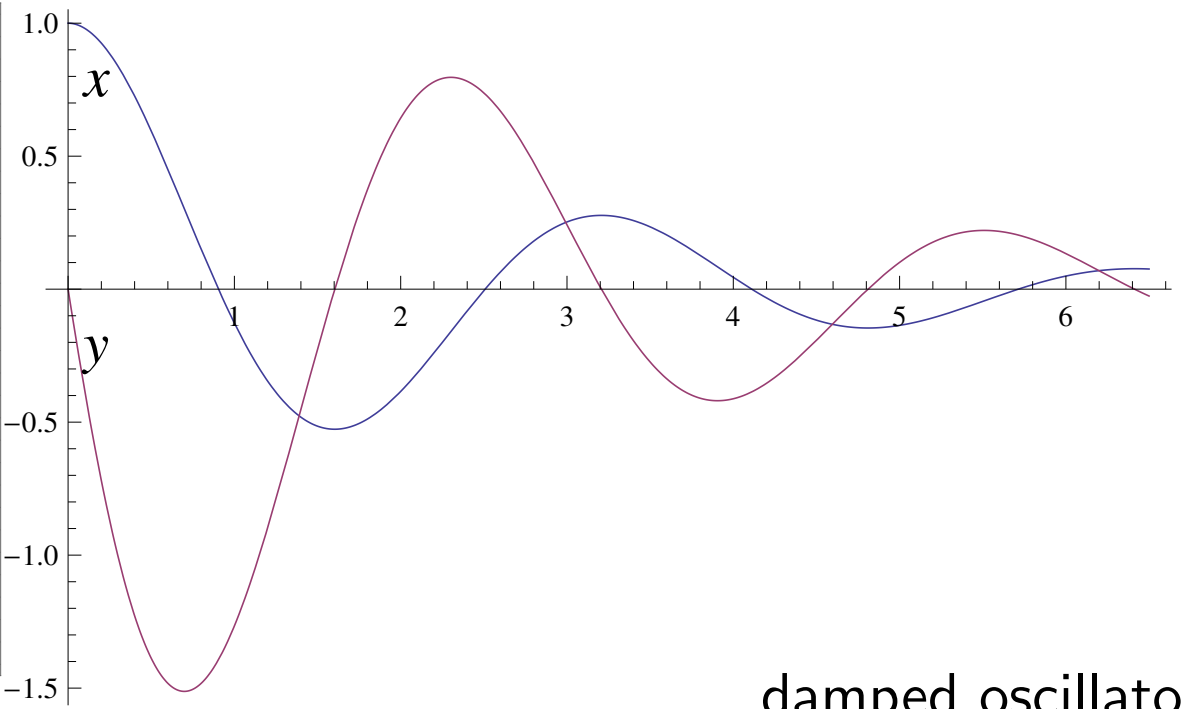


damped oscillator

Differential Invariants for Differential Equations

$$\frac{*}{\omega{\geq}0 \land d{\geq}0 \vdash 2\omega^2 xy + 2y(-\omega^2 x - 2d\omega y) \leq 0}$$
$$\frac{\omega{\geq}0 \land d{\geq}0 \vdash [x':=y][y':=-\omega^2 x - 2d\omega y]\, 2\omega^2 xx' + 2yy' \leq 0}$$
$$\omega^2x^2{+}y^2{\leq}c^2 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y \,\&\, \omega{\geq}0 \land d{\geq}0]\, \omega^2x^2{+}y^2{\leq}c^2$$

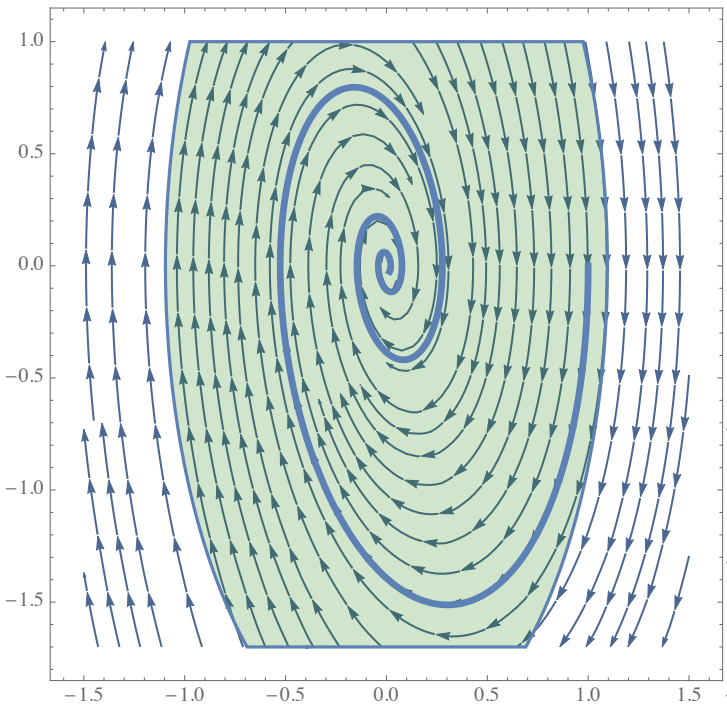


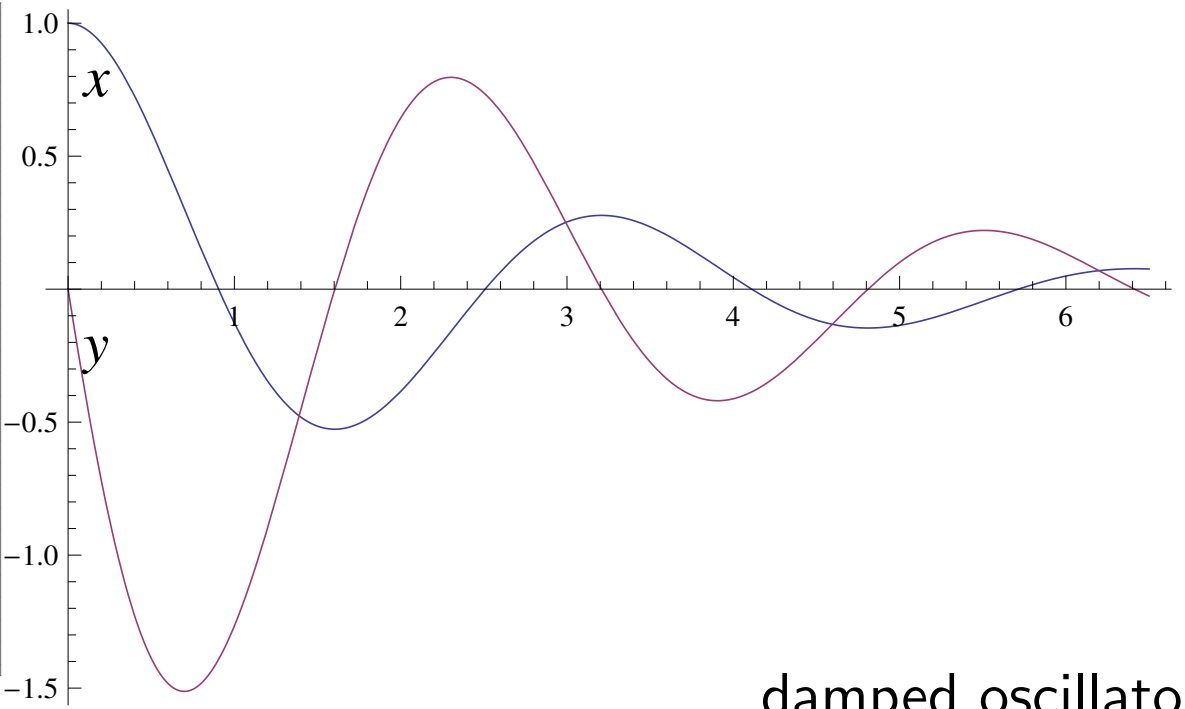


damped oscillator

Differential Invariants for Differential Equations

$$\frac{*}{\omega{\geq}0 \land d{\geq}0 \vdash 2\omega^2 xy + 2y(-\omega^2 x - 2d\omega y) \leq 0}$$
$$\frac{\omega{\geq}0 \land d{\geq}0 \vdash [x':=y][y':=-\omega^2 x - 2d\omega y]\, 2\omega^2 xx' + 2yy' \leq 0}{\omega^2x^2+y^2{\leq}c^2 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y \,\&\, \omega{\geq}0 \land d{\geq}0]\, \omega^2x^2+y^2{\leq}c^2}$$





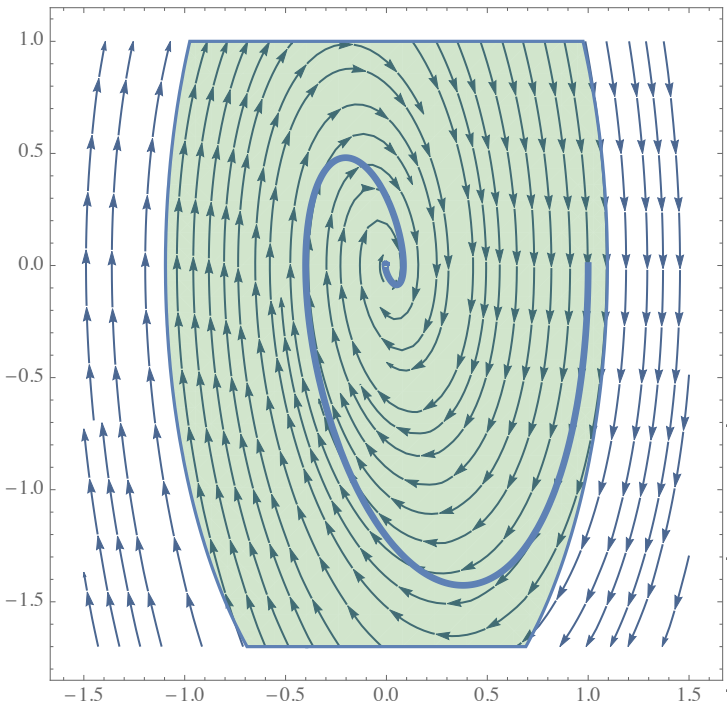
damped oscillator

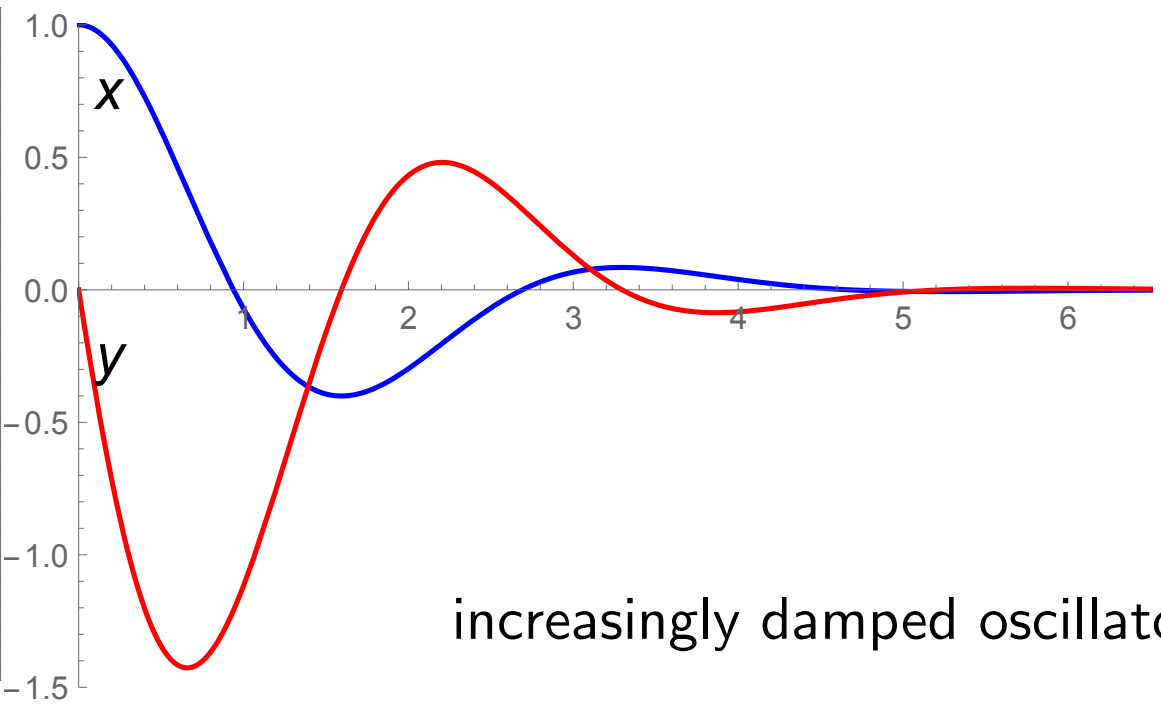
Differential Cuts for Differential Equations

$$\omega^2x^2+y^2\leq c^2 \vdash [x'=y, y'=-\omega^2x-2d\omega y, d'=7 \,\&\, \omega\geq 0]\, \omega^2x^2+y^2\leq c^2$$

Differential Cuts for Differential Equations

$$\omega^2x^2+y^2\leq c^2 \vdash [x'=y, y'=-\omega^2x-2d\omega y, d'=7 \,\&\, \omega\geq 0]\, \omega^2x^2+y^2\leq c^2$$





increasingly damped oscillator

$$\frac{\omega^2x^2+y^2\leq c^2 \vdash [x'=y,y'=-\omega^2x-2d\omega y,d'=7\,\&\,\omega\geq 0\wedge d\geq 0]\,\omega^2x^2+y^2\leq c^2}{\omega^2x^2+y^2\leq c^2 \vdash [x'=y,y'=-\omega^2x-2d\omega y,d'=7\,\&\,\omega\geq 0]\,\omega^2x^2+y^2\leq c^2}$$

increasingly damped oscillator

$$\frac{\omega^2x^2+y^2\leq c^2 \vdash [x'=y,y'=-\omega^2x-2d\omega y,d'=7\,\&\,\omega\geq 0\wedge d\geq 0]\,\omega^2x^2+y^2\leq c^2}{\omega^2x^2+y^2\leq c^2 \vdash [x'=y,y'=-\omega^2x-2d\omega y,d'=7\,\&\,\omega\geq 0]\,\omega^2x^2+y^2\leq c^2}$$

$$\frac{}{d\geq 0 \vdash [x'=y,y'=-\omega^2x-2d\omega y,d'=7\,\&\,\omega\geq 0]\,d\geq 0}$$

increasingly damped oscillator

$$\frac{\omega^2x^2+y^2\leq c^2 \vdash [x'=y,y'=-\omega^2x-2d\omega y,d'=7\,\&\,\omega\geq 0\wedge d\geq 0]\,\omega^2x^2+y^2\leq c^2}{\omega^2x^2+y^2\leq c^2 \vdash [x'=y,y'=-\omega^2x-2d\omega y,d'=7\,\&\,\omega\geq 0]\,\omega^2x^2+y^2\leq c^2}$$

$$\frac{\omega\geq 0 \vdash [d':=7]\,d'\geq 0}{d\geq 0 \vdash [x'=y,y'=-\omega^2x-2d\omega y,d'=7\,\&\,\omega\geq 0]\,d\geq 0}$$

increasingly damped oscillator

$$\frac{\omega^2x^2+y^2\leq c^2 \vdash [x'=y,y'=-\omega^2x-2d\omega y,d'=7\,\&\,\omega\geq 0\wedge d\geq 0]\,\omega^2x^2+y^2\leq c^2}{\omega^2x^2+y^2\leq c^2 \vdash [x'=y,y'=-\omega^2x-2d\omega y,d'=7\,\&\,\omega\geq 0]\,\omega^2x^2+y^2\leq c^2}$$

$$\frac{\frac{\frac{}{\omega\geq 0 \vdash 7\geq 0}}{\omega\geq 0 \vdash [d':=7]\,d'\geq 0}}{d\geq 0 \vdash [x'=y,y'=-\omega^2x-2d\omega y,d'=7\,\&\,\omega\geq 0]\,d\geq 0}$$

increasingly damped oscillator

Differential Cuts for Differential Equations

$$\frac{\omega^2x^2+y^2\leq c^2 \vdash [x'=y,y'=-\omega^2x-2d\omega y,d'=7\,\&\,\omega\geq 0\wedge d\geq 0]\,\omega^2x^2+y^2\leq c^2}{\omega^2x^2+y^2\leq c^2 \vdash [x'=y,y'=-\omega^2x-2d\omega y,d'=7\,\&\,\omega\geq 0]\,\omega^2x^2+y^2\leq c^2}\;\text{DC}$$

$$\frac{*}{\omega\geq 0 \vdash 7\geq 0}$$

$$\frac{\omega\geq 0 \vdash 7\geq 0}{\omega\geq 0 \vdash [d':=7]\,d'\geq 0}$$

$$\frac{\omega\geq 0 \vdash [d':=7]\,d'\geq 0}{d\geq 0 \vdash [x'=y,y'=-\omega^2x-2d\omega y,d'=7\,\&\,\omega\geq 0]\,d\geq 0}$$

increasingly damped oscillator

Differential Cuts for Differential Equations

$$\dfrac{\dfrac{\dfrac{}{\omega\geq 0\wedge d\geq 0 \vdash [x':=y][y':=-\omega^2 x-2d\omega y]\,2\omega^2 xx'+2yy'\leq 0}}{\omega^2x^2+y^2\leq c^2 \vdash [x'=y,y'=-\omega^2x-2d\omega y,d'=7\,\&\,\omega\geq 0\wedge d\geq 0]\,\omega^2x^2+y^2\leq c^2}}{\omega^2x^2+y^2\leq c^2 \vdash [x'=y,y'=-\omega^2x-2d\omega y,d'=7\,\&\,\omega\geq 0]\,\omega^2x^2+y^2\leq c^2}$$

$$\dfrac{\dfrac{\dfrac{*}{\omega\geq 0 \vdash 7\geq 0}}{\omega\geq 0 \vdash [d':=7]\,d'\geq 0}}{d\geq 0 \vdash [x'=y,y'=-\omega^2x-2d\omega y,d'=7\,\&\,\omega\geq 0]\,d\geq 0}$$

increasingly damped oscillator

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\omega\geq 0\wedge d\geq 0 \vdash 2\omega^2 xy + 2y(-\omega^2 x - 2d\omega y) \leq 0}}{\omega\geq 0\wedge d\geq 0 \vdash [x':=y][y':=-\omega^2 x - 2d\omega y]\, 2\omega^2 xx' + 2yy' \leq 0}}{\omega^2x^2+y^2\leq c^2 \vdash [x' = y, y' = -\omega^2 x-2d\omega y, d'=7 \,\&\, \omega\geq 0\wedge d\geq 0]\, \omega^2x^2+y^2\leq c^2}}{\omega^2x^2+y^2\leq c^2 \vdash [x' = y, y' = -\omega^2 x-2d\omega y, d'=7 \,\&\, \omega\geq 0]\, \omega^2x^2+y^2\leq c^2}$$

$$\dfrac{\dfrac{\dfrac{*}{\omega\geq 0 \vdash 7\geq 0}}{\omega\geq 0 \vdash [d':=7]\, d'\geq 0}}{d\geq 0 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y, d'=7 \,\&\, \omega\geq 0]\, d\geq 0}$$

increasingly damped oscillator

Differential Cuts for Differential Equations

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\omega\geq 0\wedge d\geq 0 \vdash 2\omega^2 xy + 2y(-\omega^2 x - 2d\omega y) \leq 0}\,*}{\omega\geq 0\wedge d\geq 0 \vdash [x':=y][y':=-\omega^2 x - 2d\omega y]\, 2\omega^2 xx' + 2yy' \leq 0}}{\omega^2x^2+y^2\leq c^2 \vdash [x' = y, y' = -\omega^2 x-2d\omega y, d'=7 \,\&\, \omega\geq 0\wedge d\geq 0]\, \omega^2x^2+y^2\leq c^2}}{\omega^2x^2+y^2\leq c^2 \vdash [x' = y, y' = -\omega^2 x-2d\omega y, d'=7 \,\&\, \omega\geq 0]\, \omega^2x^2+y^2\leq c^2}$$

$$\dfrac{\dfrac{\dfrac{}{\omega\geq 0 \vdash 7\geq 0}\,*}{\omega\geq 0 \vdash [d':=7]\, d'\geq 0}}{d\geq 0 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y, d'=7 \,\&\, \omega\geq 0]\, d\geq 0}$$

increasingly damped oscillator

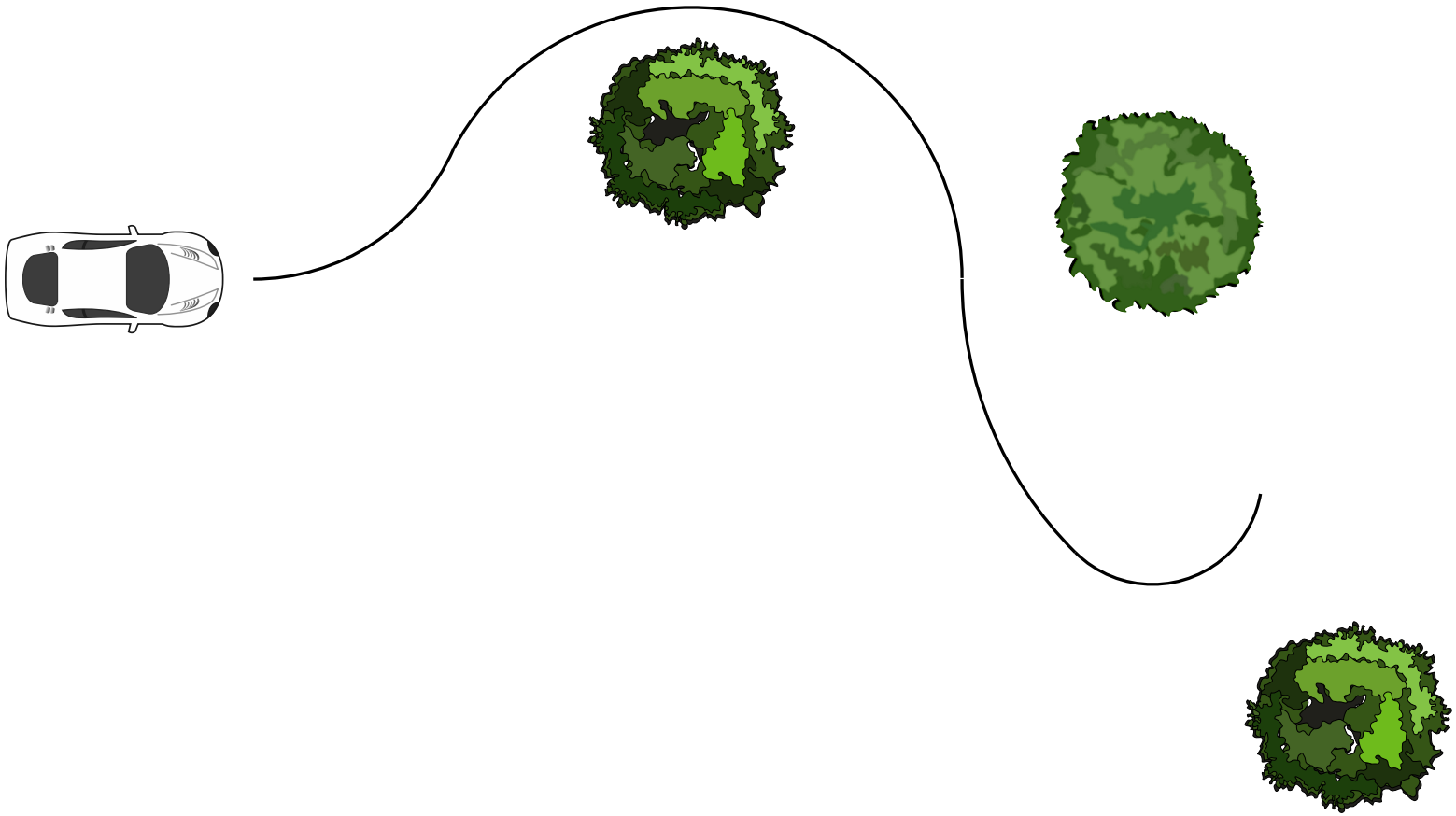
Differential Cuts for Differential Equations

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\omega\geq 0\wedge d\geq 0 \vdash 2\omega^2 xy + 2y(-\omega^2 x - 2d\omega y) \leq 0}\,*}{\omega\geq 0\wedge d\geq 0 \vdash [x':=y][y':=-\omega^2 x - 2d\omega y]\, 2\omega^2 xx' + 2yy' \leq 0}}{\omega^2x^2+y^2\leq c^2 \vdash [x' = y, y' = -\omega^2 x-2d\omega y, d'=7 \,\&\, \omega\geq 0\wedge d\geq 0]\, \omega^2x^2+y^2\leq c^2}}{\omega^2x^2+y^2\leq c^2 \vdash [x' = y, y' = -\omega^2 x-2d\omega y, d'=7 \,\&\, \omega\geq 0]\, \omega^2x^2+y^2\leq c^2}$$

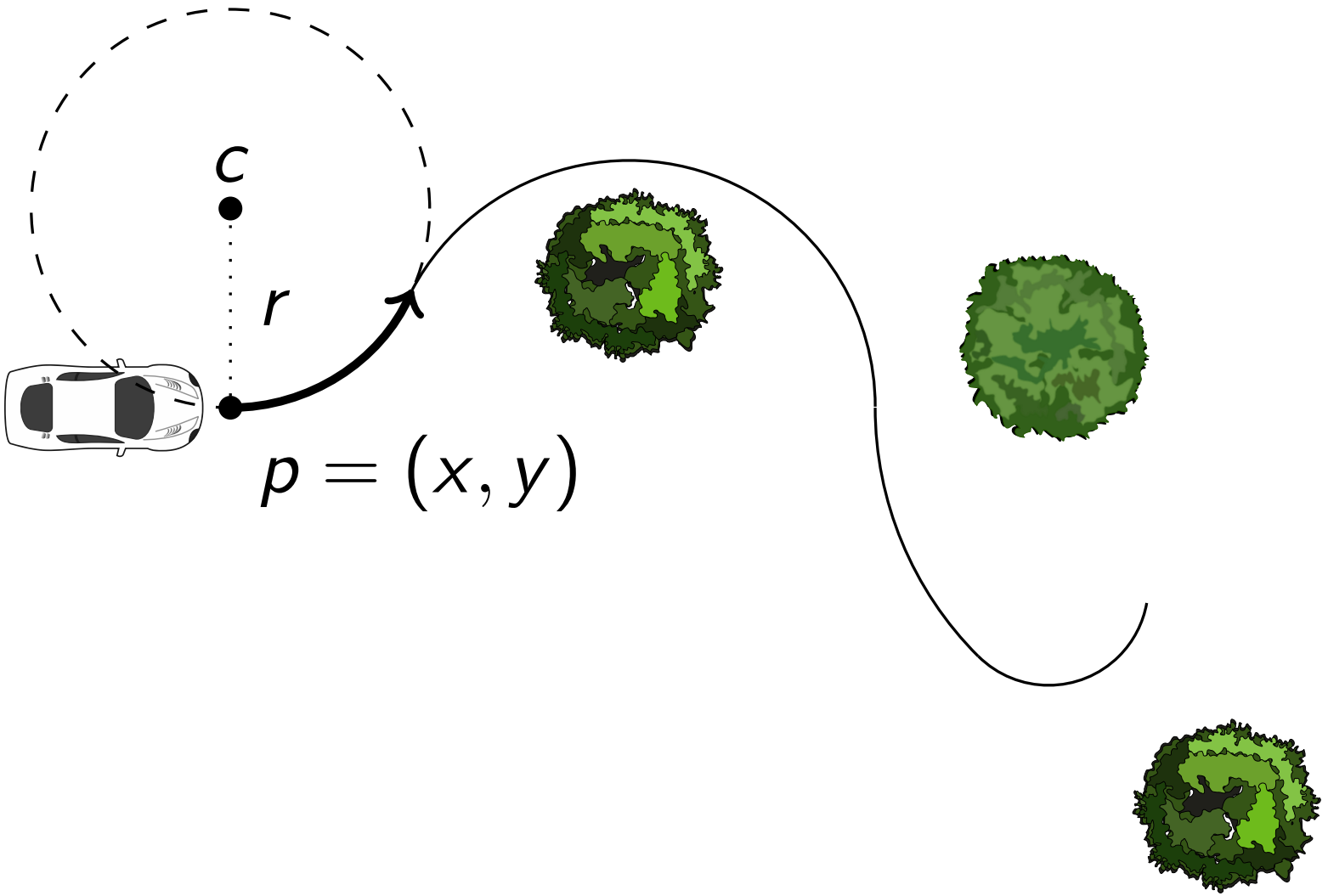
$$\dfrac{\dfrac{\dfrac{\dfrac{}{\omega\geq 0 \vdash 7\geq 0}\,*}{\omega\geq 0 \vdash [d':=7]\, d'\geq 0}}{d\geq 0 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y, d'=7 \,\&\, \omega\geq 0]\, d\geq 0}}{}$$

Could repeatedly diffcut in formulas to help the proof

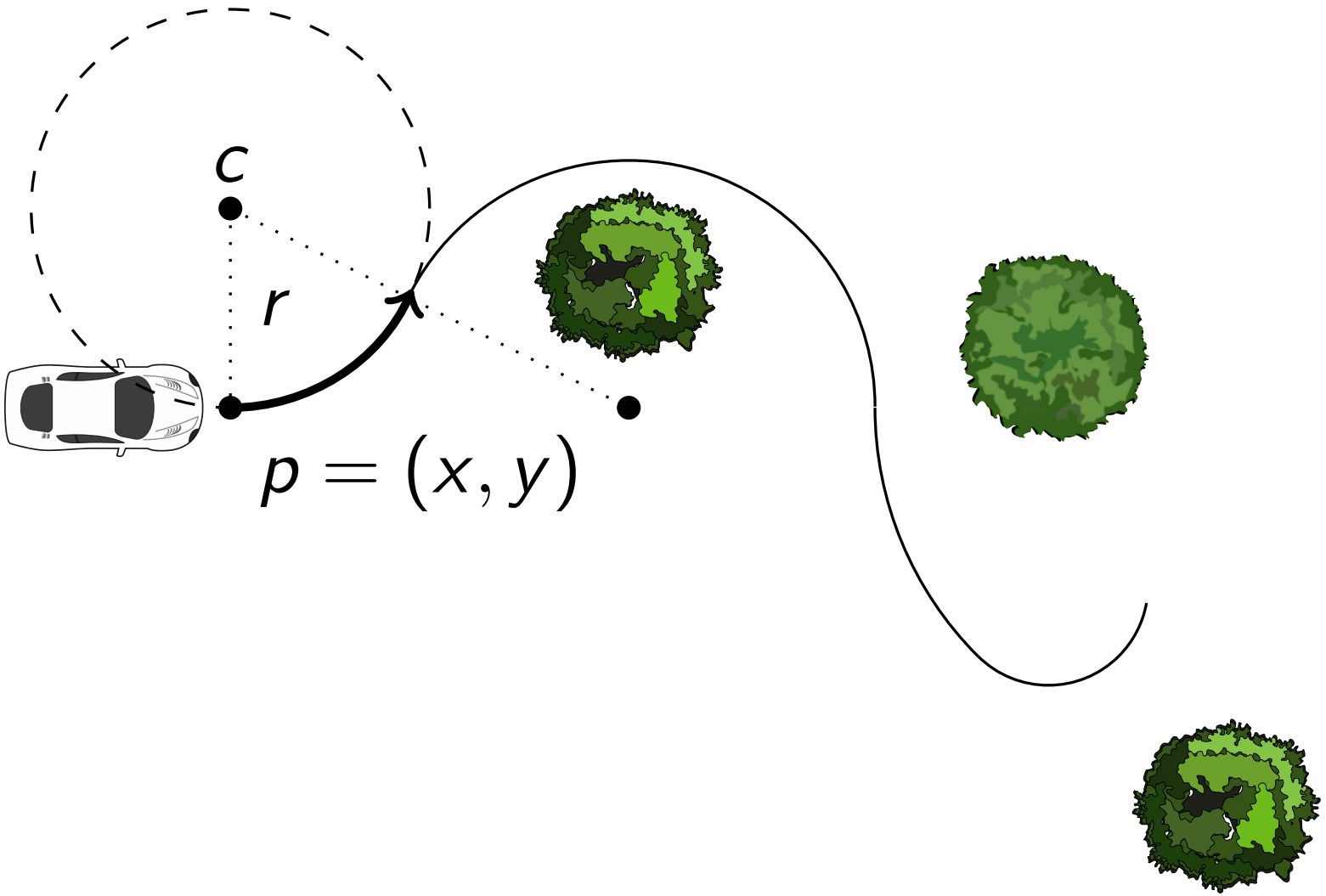
Motion in 2D Plane: Don't Collide with Obstacles



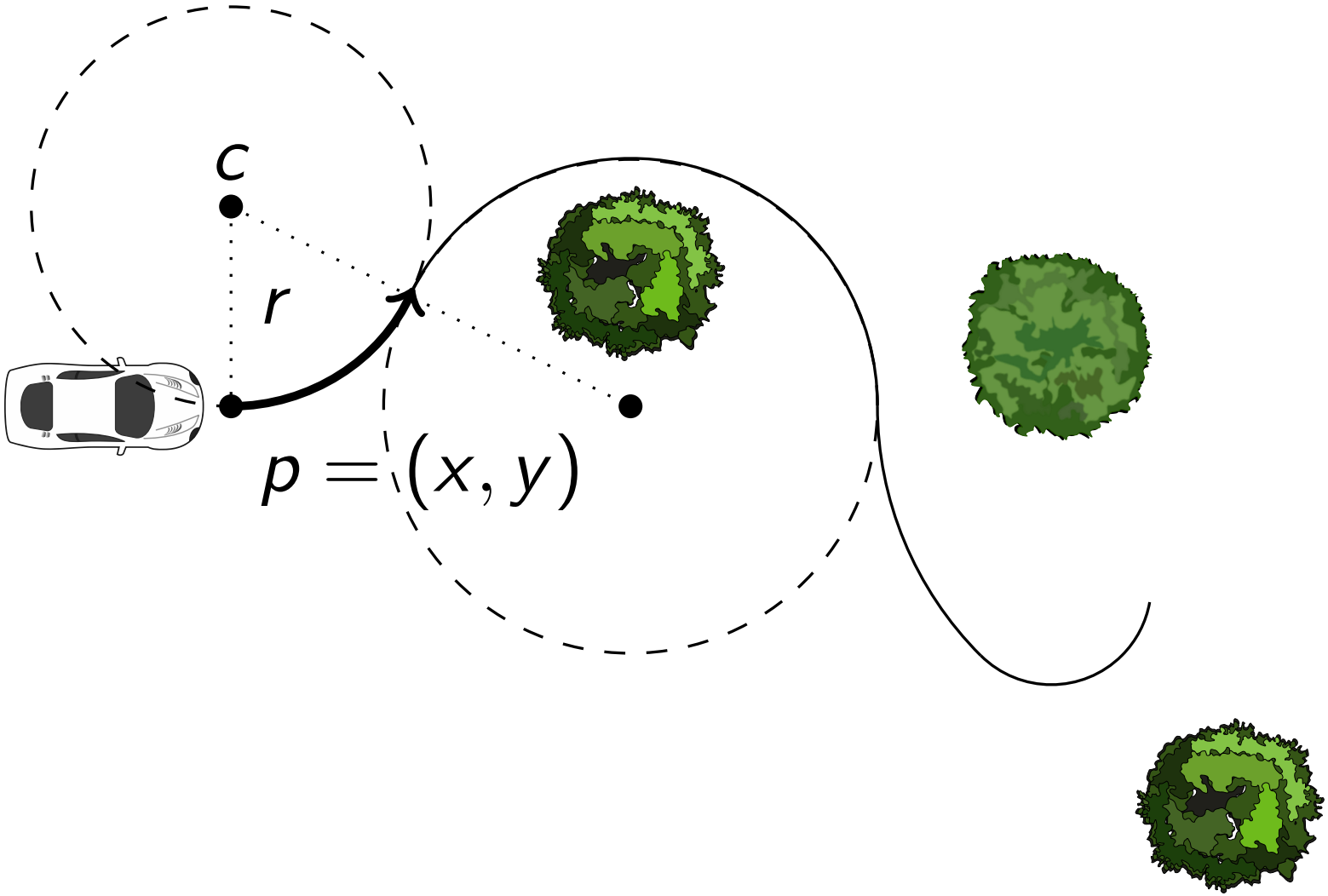
Motion in 2D Plane: Don't Collide with Obstacles



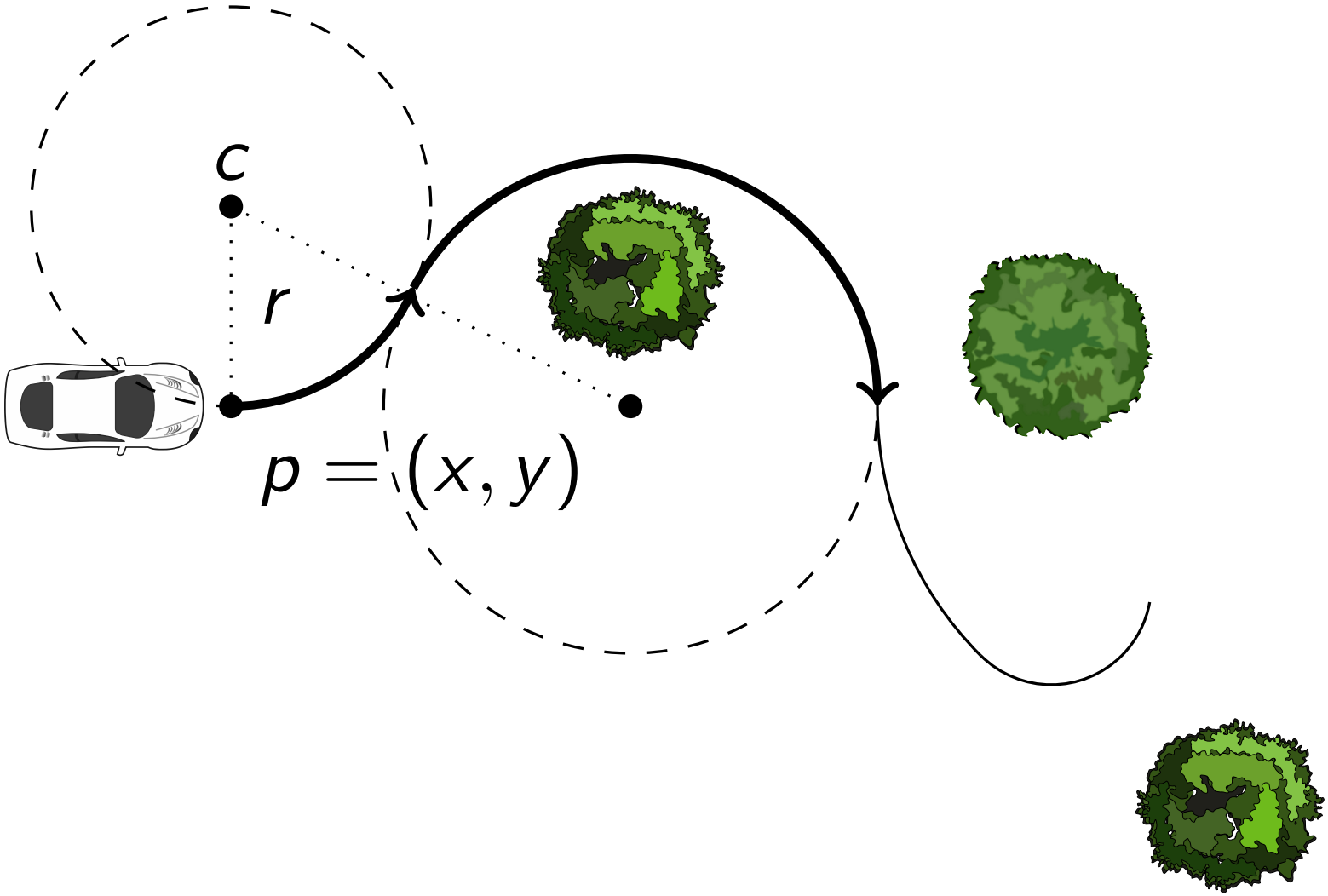
Motion in 2D Plane: Don't Collide with Obstacles



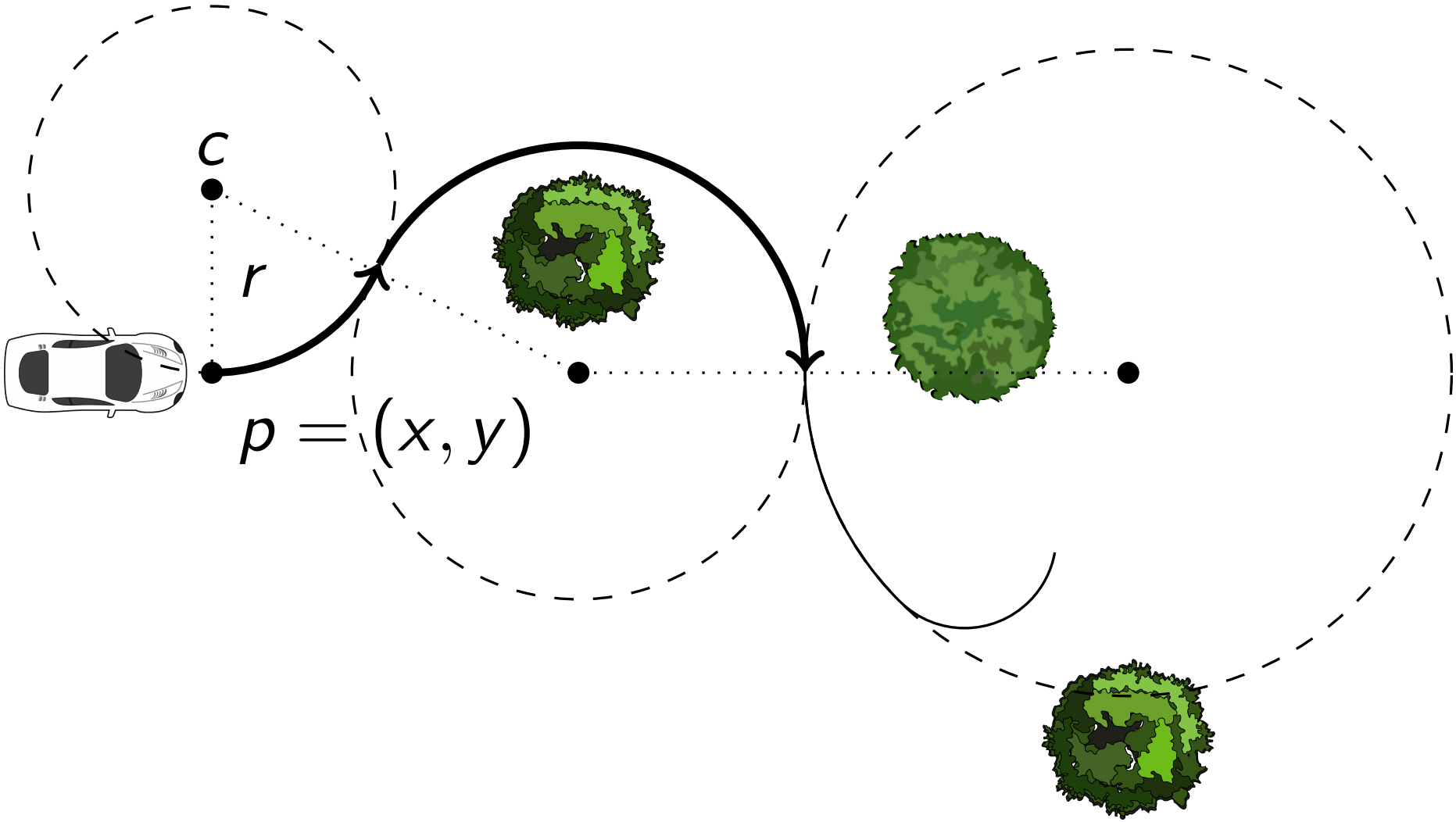
Motion in 2D Plane: Don't Collide with Obstacles



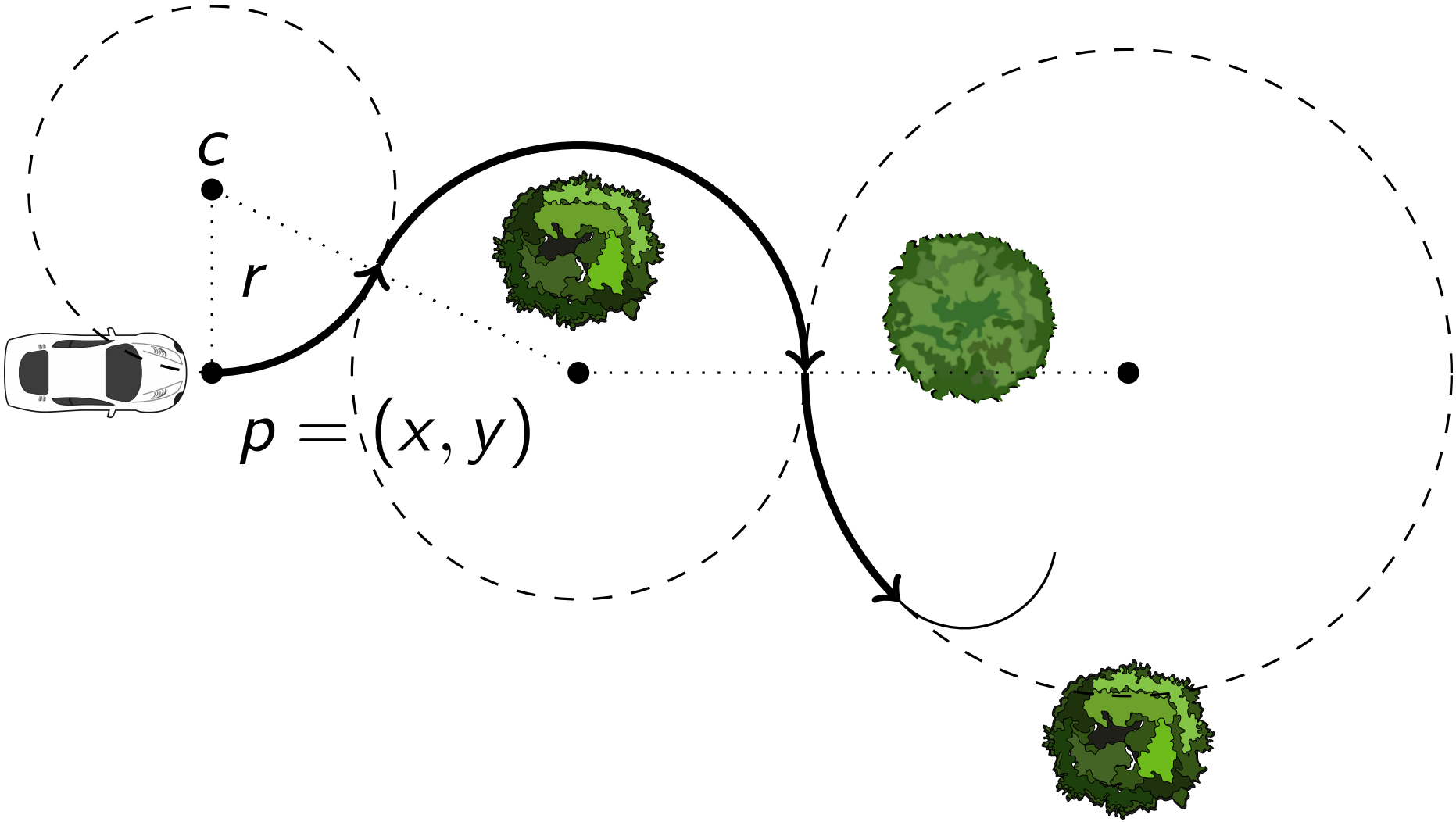
Motion in 2D Plane: Don't Collide with Obstacles



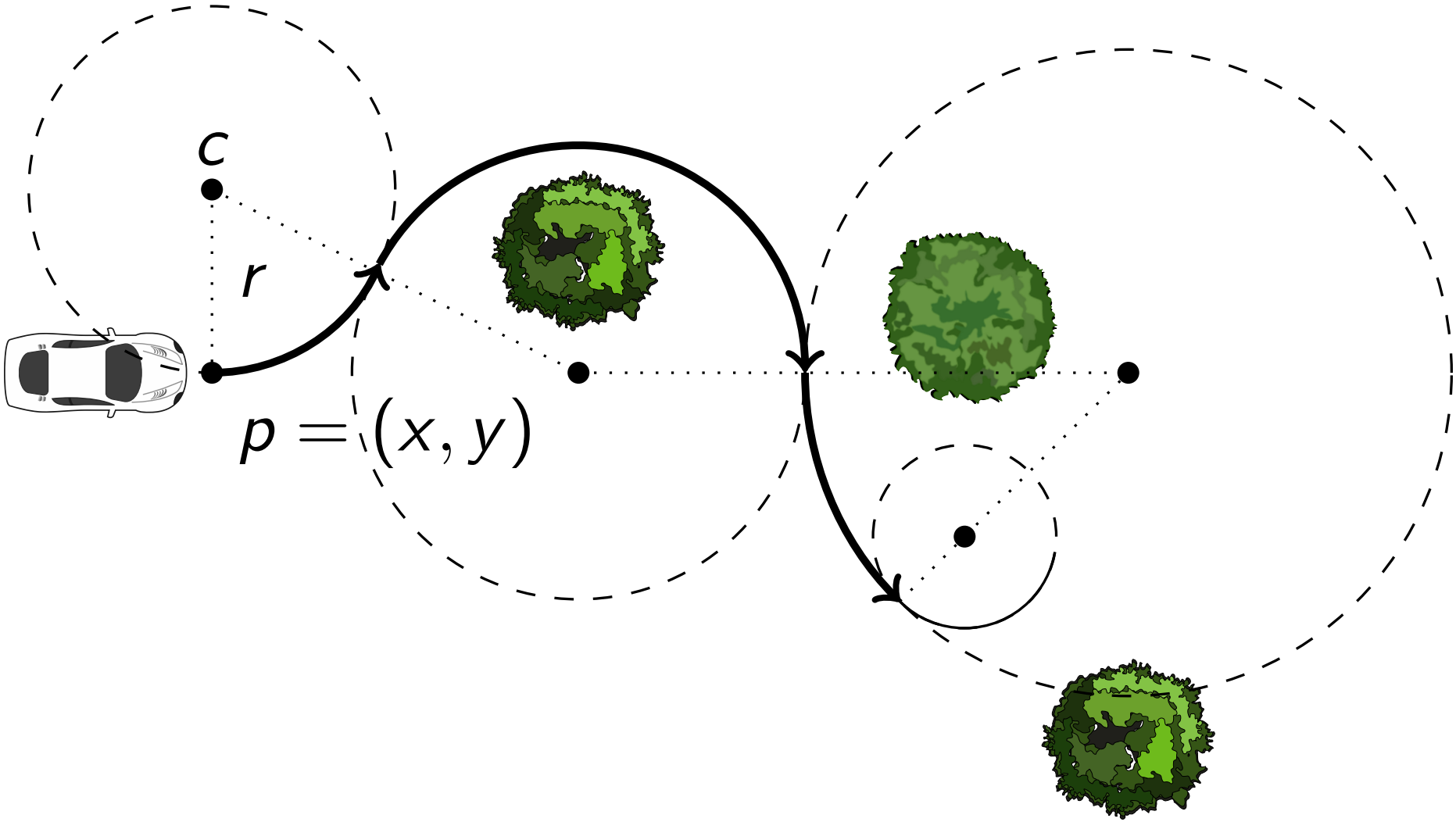
Motion in 2D Plane: Don't Collide with Obstacles



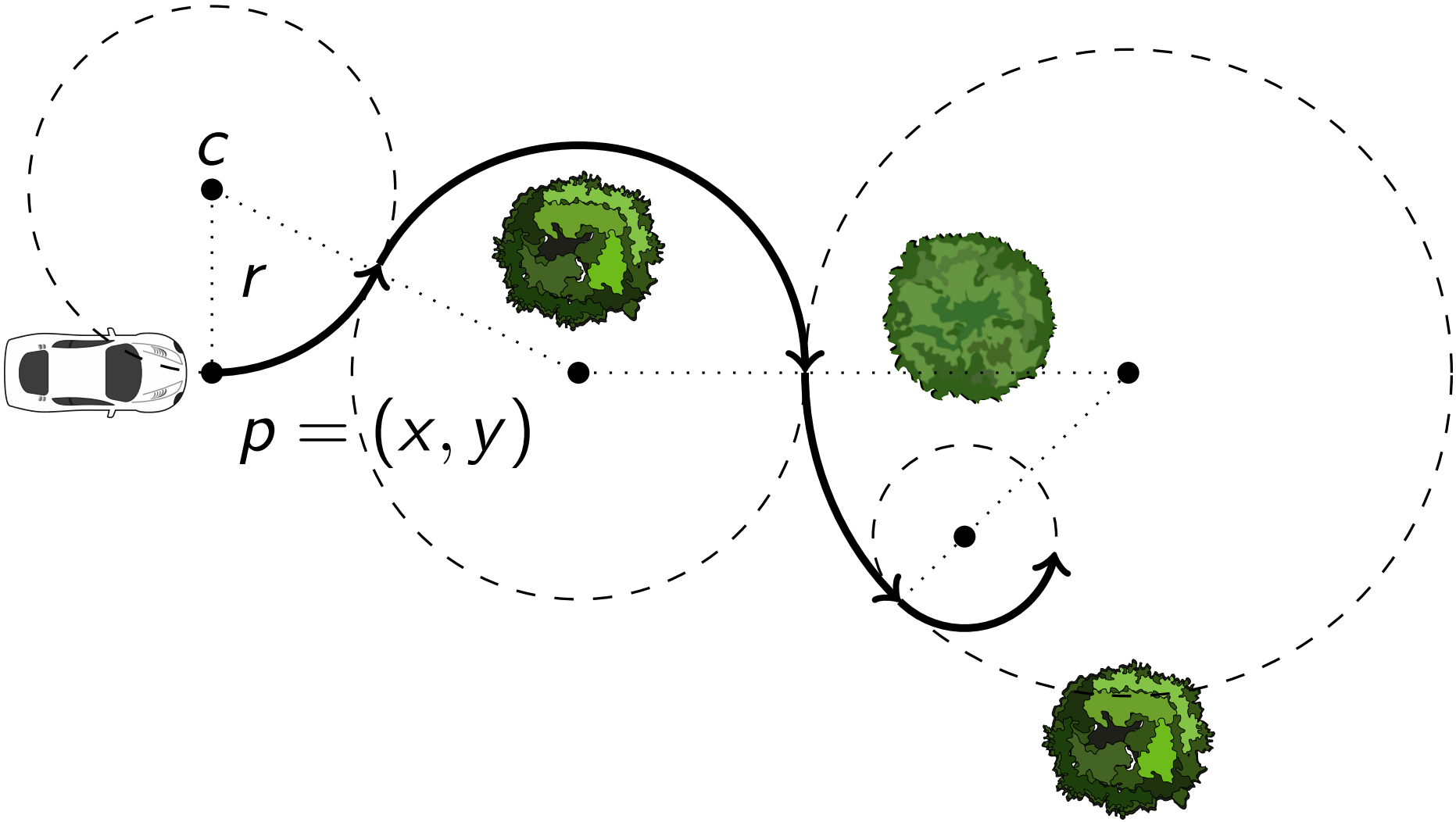
Motion in 2D Plane: Don't Collide with Obstacles



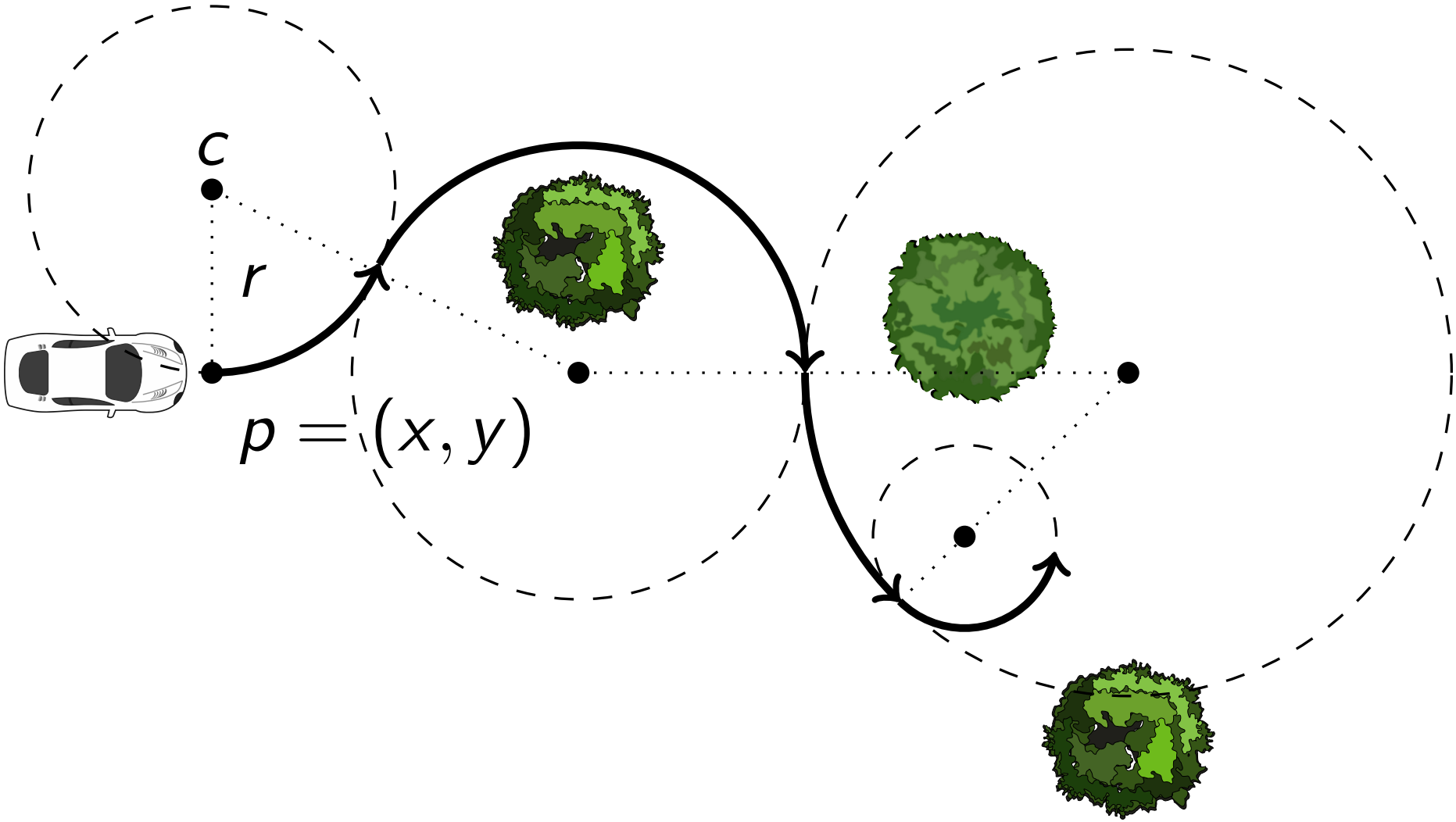
Motion in 2D Plane: Don't Collide with Obstacles



Motion in 2D Plane: Don't Collide with Obstacles



Motion in 2D Plane: Don't Collide with Obstacles

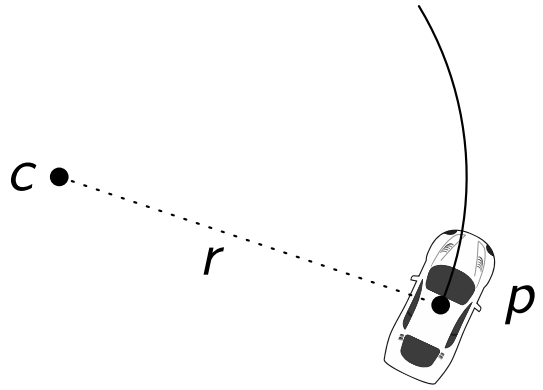


How to always get such motion collision-free?

2D Planar Car Model

State

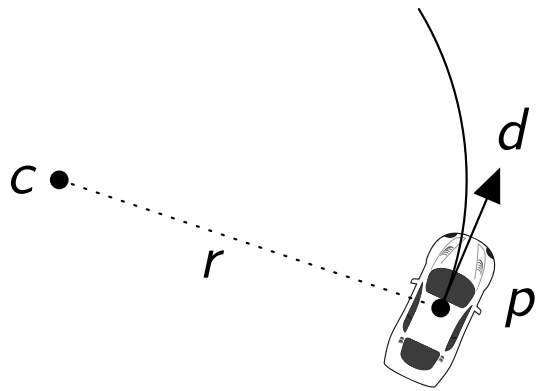
- Position p
- Curve center c
- Curve radius r



2D Planar Car Model

State

- Position p
- Curve center c
- Curve radius r
- Orientation d



2D Planar Car Model

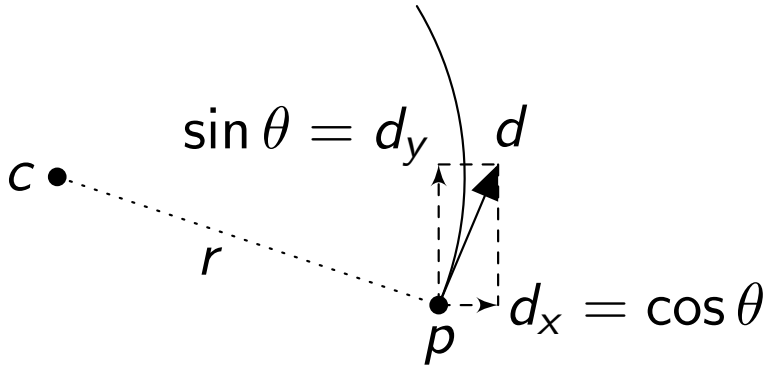
State

- Position p
- Curve center c
- Curve radius r
- Orientation d

Differential Axiomatization

- New variables for (undecidable) transcendental functions

$d_x = \cos(\theta)$, $d_y = \sin(\theta)$



2D Planar Car Model

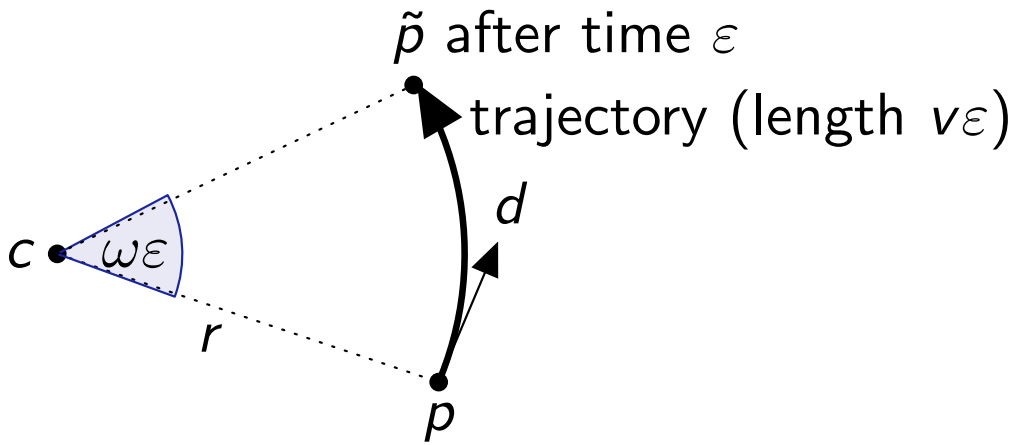
State

- Position p
- Curve center c
- Curve radius r
- Orientation d
- Translational velocity $x' = v$
- Rotational velocity $\theta' = \omega$
- Control cycle duration ε

Differential Axiomatization

- New variables for (undecidable) transcendental functions

$d_x = \cos(\theta)$, $d_y = \sin(\theta)$



2D Planar Car Model

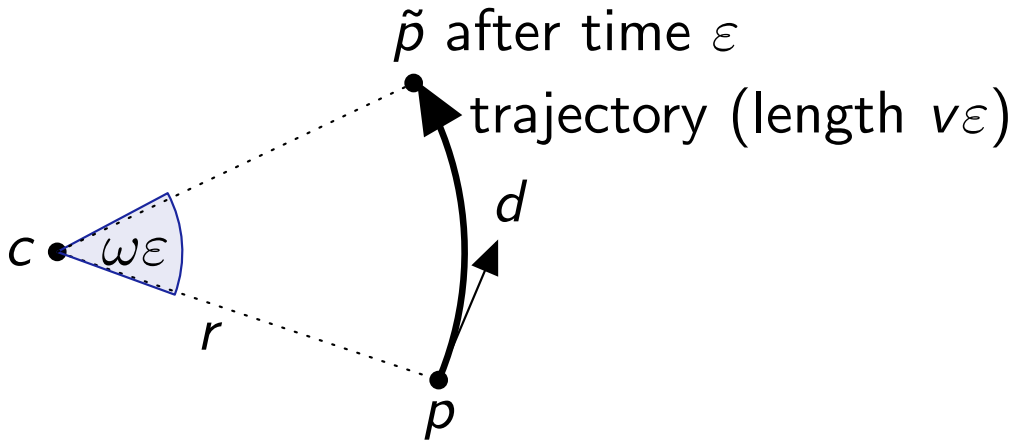
State

- Position p
- Curve center c
- Curve radius r
- Orientation d
- Translational velocity $x' = v$
- Rotational velocity $\theta' = \omega$
- Control cycle duration ε

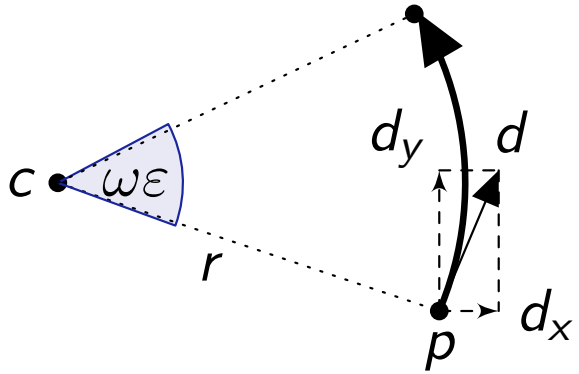
Differential Axiomatization

- New variables for (undecidable) transcendental functions

$d_x = \cos(\theta)$, $d_y = \sin(\theta)$
$d_x' = \cos(\theta)' = -\sin(\theta)\theta' = -\omega d_y$



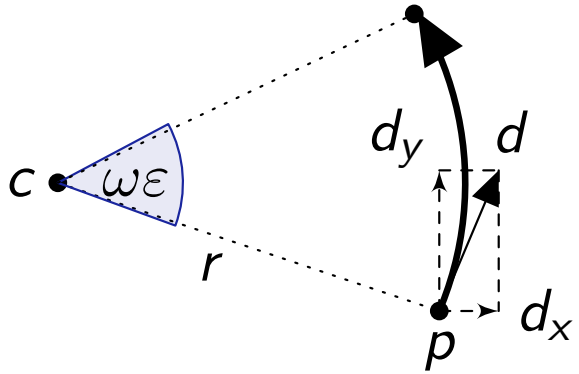
2D Car Model as Hybrid Program



Example: 2D Car with Brake, Accelerate, Steer

$$t := 0;$$
$$\{p' = vd,\ v' = a,\ d' = \omega d^{\perp},\ \omega' = \frac{a}{r},\ t' = 1\ \&\ v \geq 0 \wedge t \leq \varepsilon\}$$

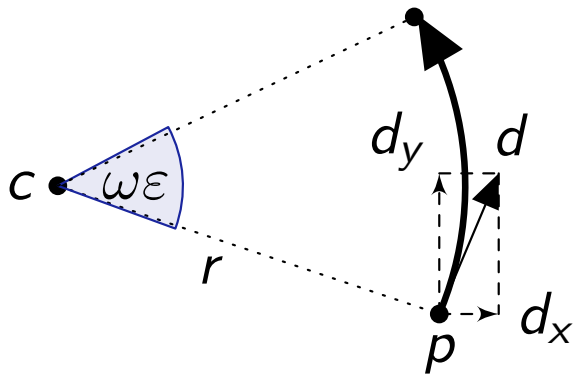
2D Car Model as Hybrid Program



Example: 2D Car with Brake, Accelerate, Steer

$$\Big(\quad a := -b \\ \quad \cup\, a := A;\ \omega := *;\ r := *;\ ?r \neq 0 \wedge r\omega = v;\ m := *;\ ?Q \Big); \\ t := 0; \\ \{p' = vd,\ v' = a,\ d' = \omega d^{\perp},\ \omega' = \frac{a}{r},\ t' = 1\ \&\ v \geq 0 \wedge t \leq \varepsilon\}$$

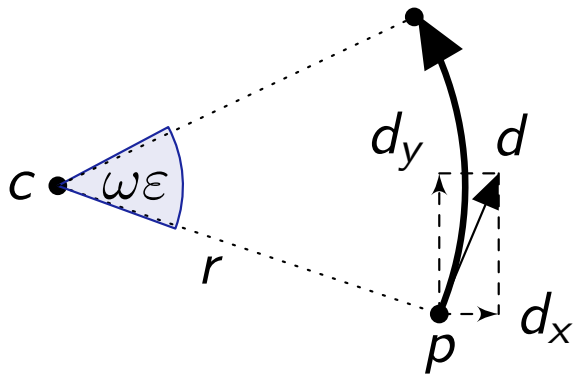
2D Car Model as Hybrid Program



Example: 2D Car with Brake, Accelerate, Steer

$$\Big(\big(a := -b \cup a := A;\ \omega := *;\ r := *;\ ?r \neq 0 \land r\omega = v;\ m := *;\ ?Q \big);\ t := 0;\ \{p' = vd,\ v' = a,\ d' = \omega d^{\perp},\ \omega' = \frac{a}{r},\ t' = 1\ \&\ v \geq 0 \land t \leq \varepsilon\} \Big)^*$$

2D Car Model as Hybrid Program

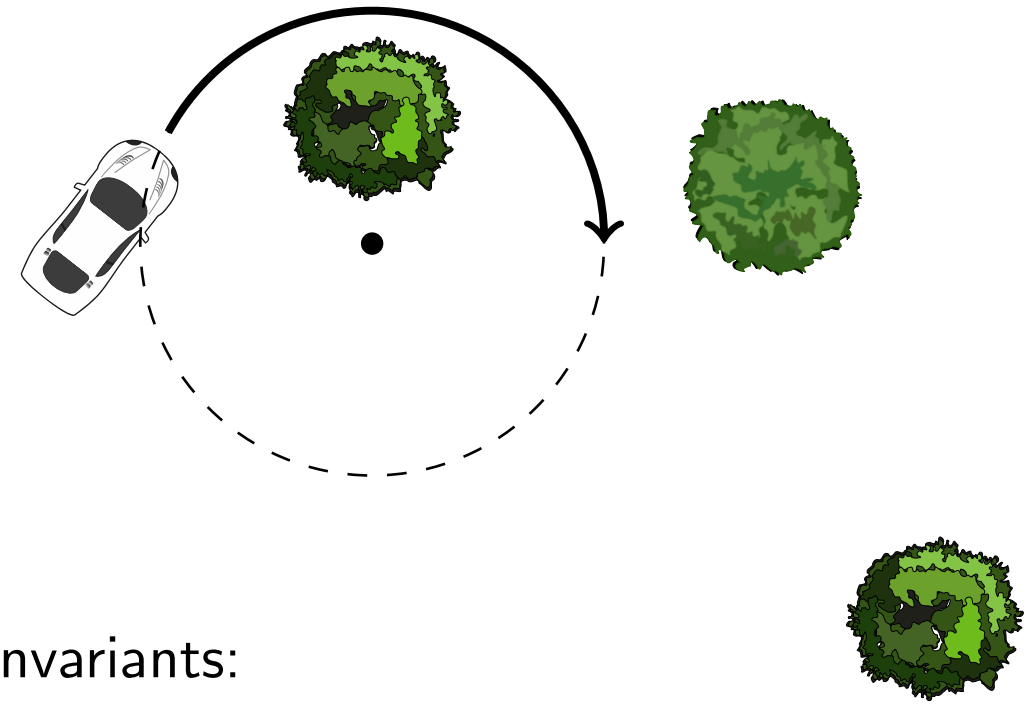


Example: 2D Car with Brake, Accelerate, Steer

$$Q \equiv 2b\|p-m\| \geq v^2 + (A+b)(A\varepsilon^2 + 2\varepsilon v)$$

$$\Big(\big(a := -b \cup a := A;\ \omega := *;\ r := *;\ ?r \neq 0 \wedge r\omega = v;\ m := *;\ ?Q \big);\ t := 0;\ \{p' = vd,\ v' = a,\ d' = \omega d^{\perp},\ \omega' = \frac{a}{r},\ t' = 1\ \&\ v \geq 0 \wedge t \leq \varepsilon\} \Big)^*$$

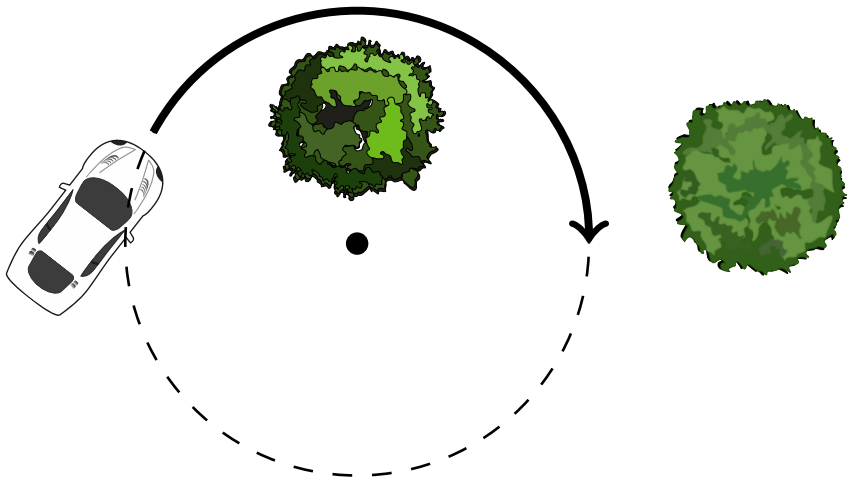
Intuition for Car's Differential Invariants



Differential Invariants:

- Time goes forward: $t \geq 0$

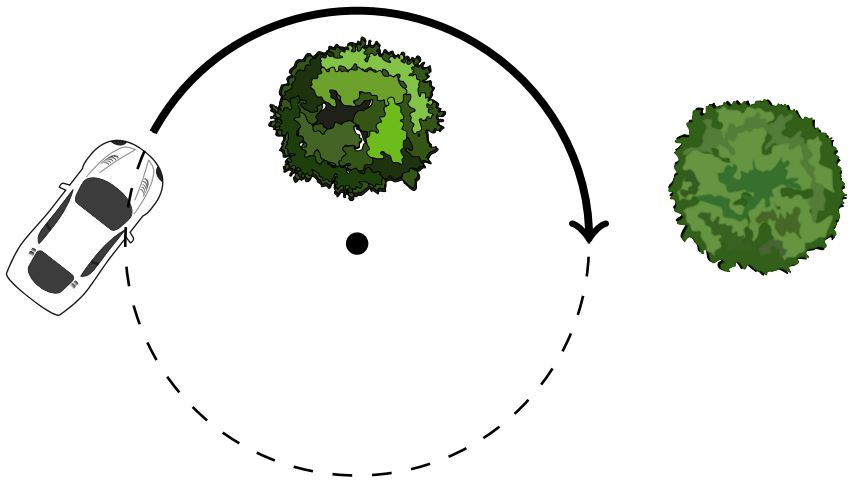
Intuition for Car's Differential Invariants



Differential Invariants:

- Time goes forward: $t \geq 0$
- Velocity follows acceleration: $v = \text{old}(v) + at$

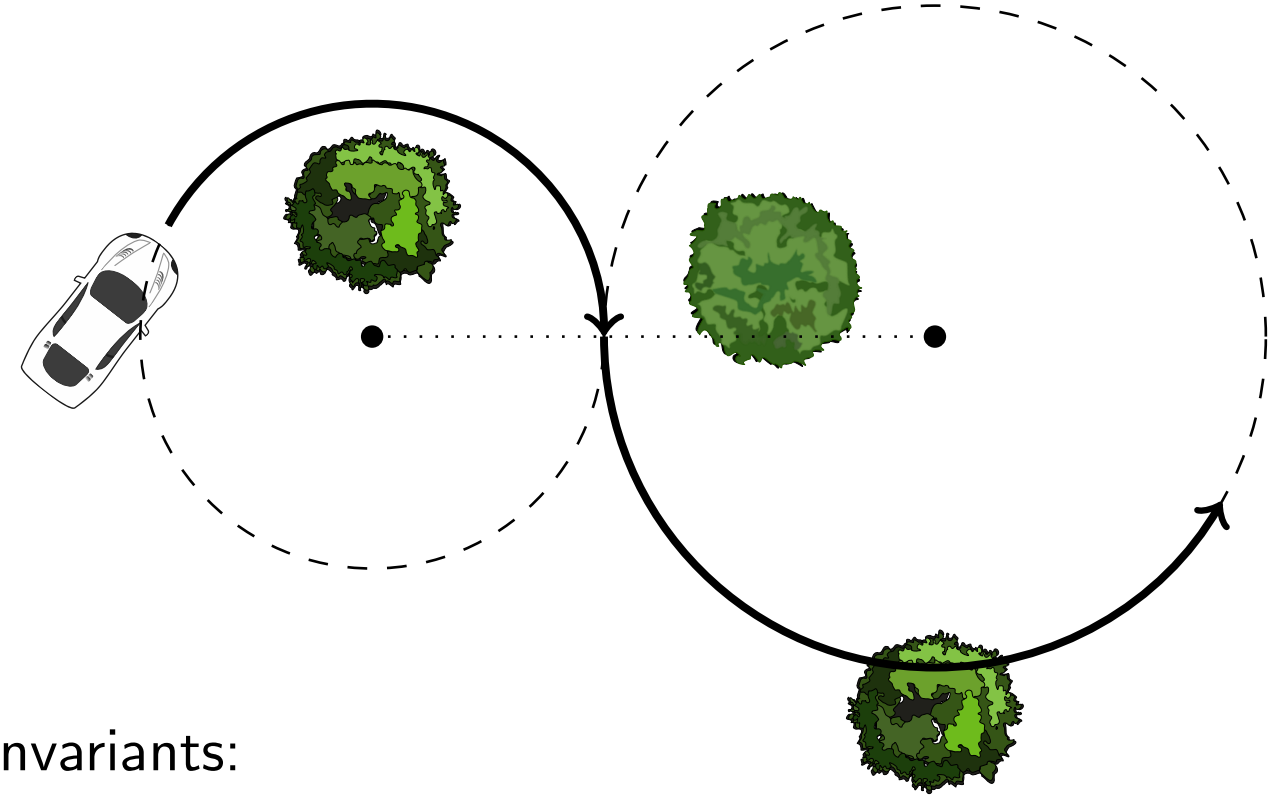
Intuition for Car's Differential Invariants



Differential Invariants:

- Time goes forward: $t \geq 0$
- Velocity follows acceleration: $v = \text{old}(v) + at$
- Stay on the circle: $\|d\| = 1$

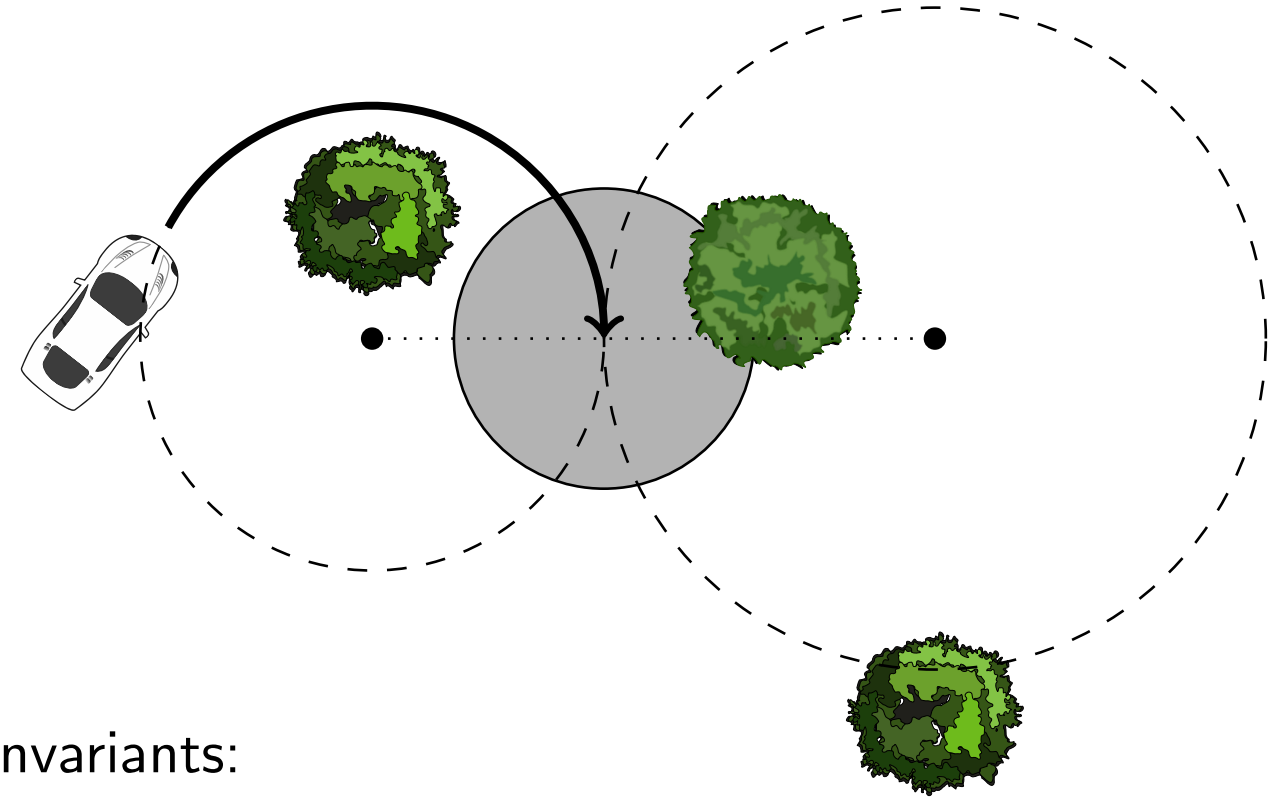
Intuition for Car's Differential Invariants



Differential Invariants:

- Time goes forward: $t \geq 0$
- Velocity follows acceleration: $v = \text{old}(v) + at$
- Stay on the circle: $\|d\| = 1$

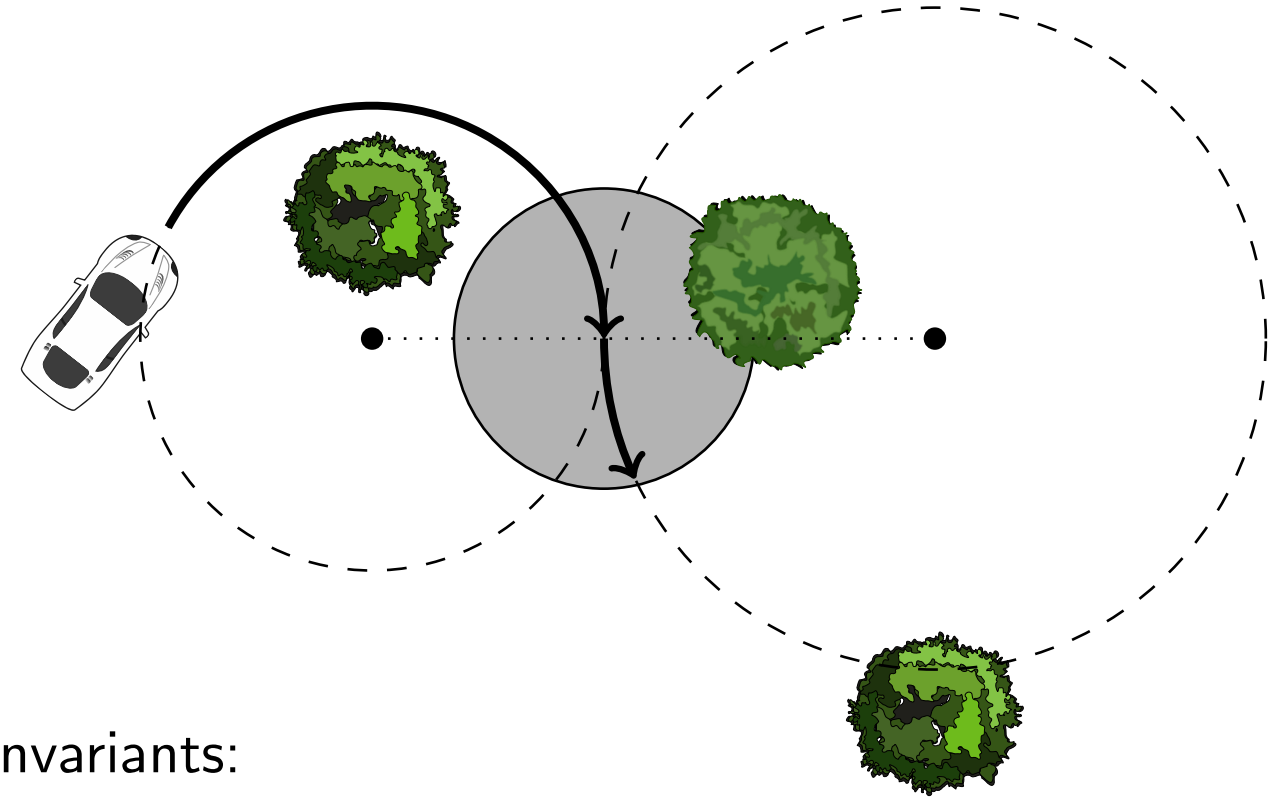
Intuition for Car's Differential Invariants



Differential Invariants:

- Time goes forward: $t \geq 0$
- Velocity follows acceleration: $v = \text{old}(v) + at$
- Stay on the circle: $\|d\| = 1$
- Stay close to position: $\|p - \text{old}(p)\| \leq vt - \frac{a}{2}t^2$

Intuition for Car's Differential Invariants

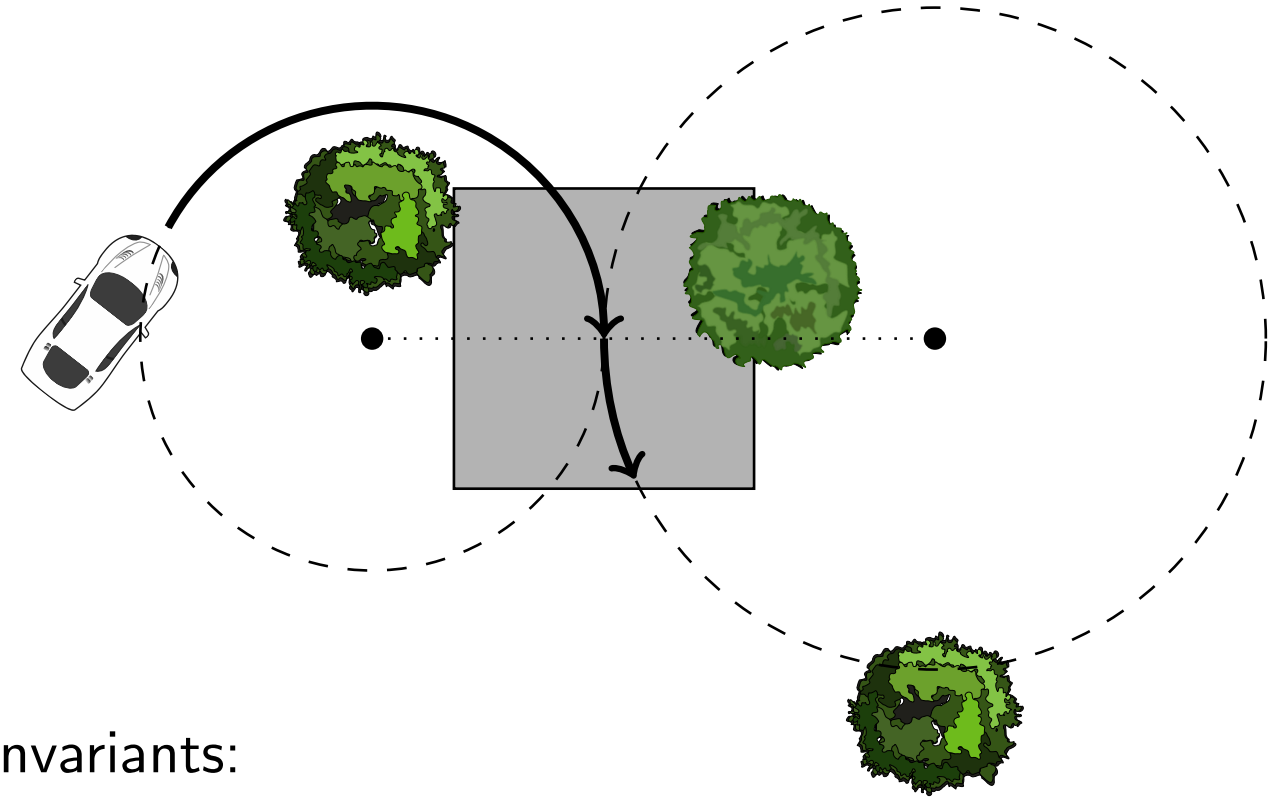


Differential Invariants:

- Time goes forward: $t \geq 0$
- Velocity follows acceleration: $v = \text{old}(v) + at$
- Stay on the circle: $\|d\| = 1$
- Stay close to position: $\|p - \text{old}(p)\| \leq vt - \frac{a}{2}t^2$

} need both

Intuition for Car's Differential Invariants



Differential Invariants:

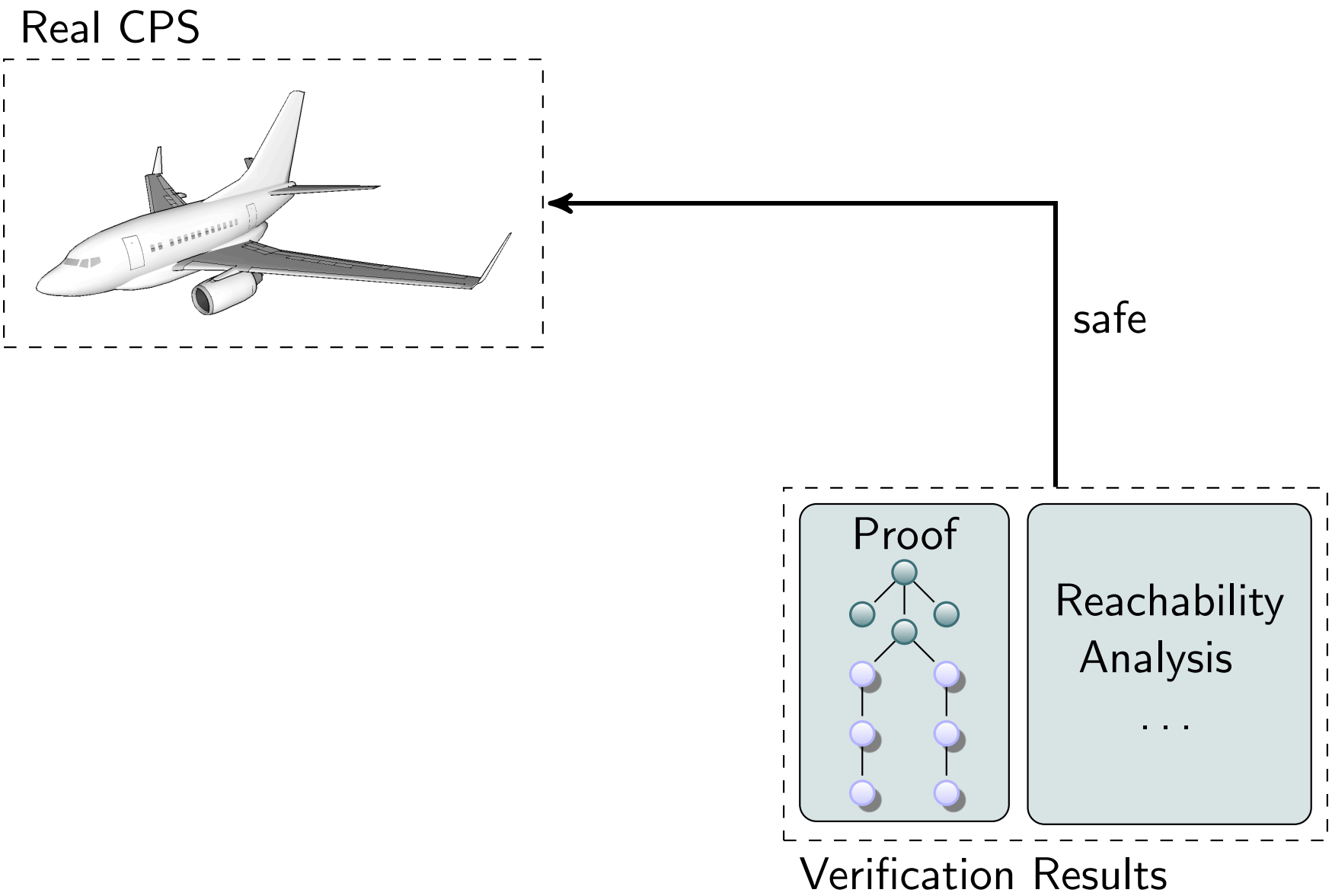
- Time goes forward: $t \geq 0$
- Velocity follows acceleration: $v = \mathrm{old}(v) + at$
- Stay on the circle: $\|d\| = 1$
- Stay close to position: $\|p - \mathrm{old}(p)\|_\infty \leq vt - \frac{a}{2}t^2$

} need both

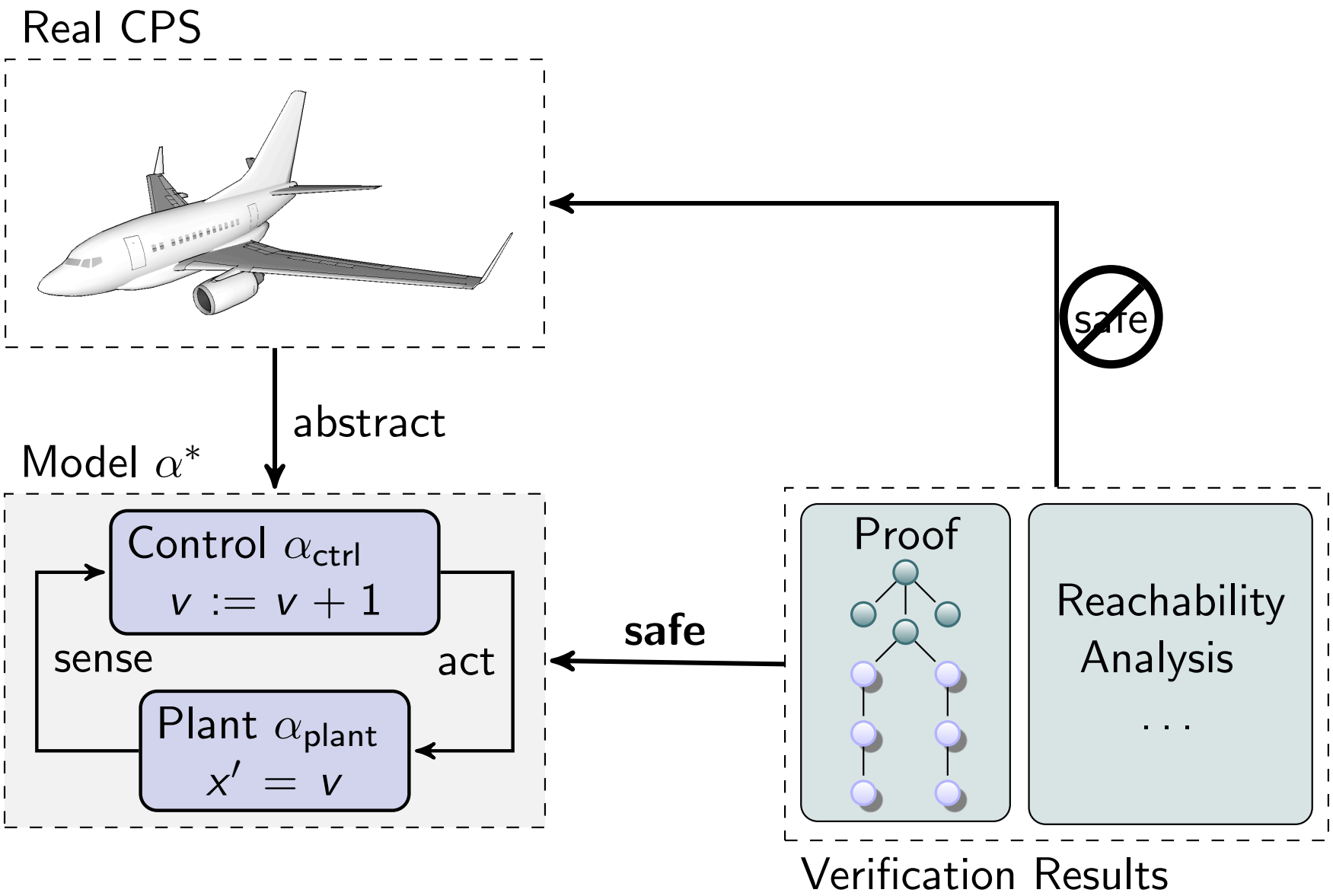
supremum norm overapproximates circle with box (easier arithmetic)

Outline

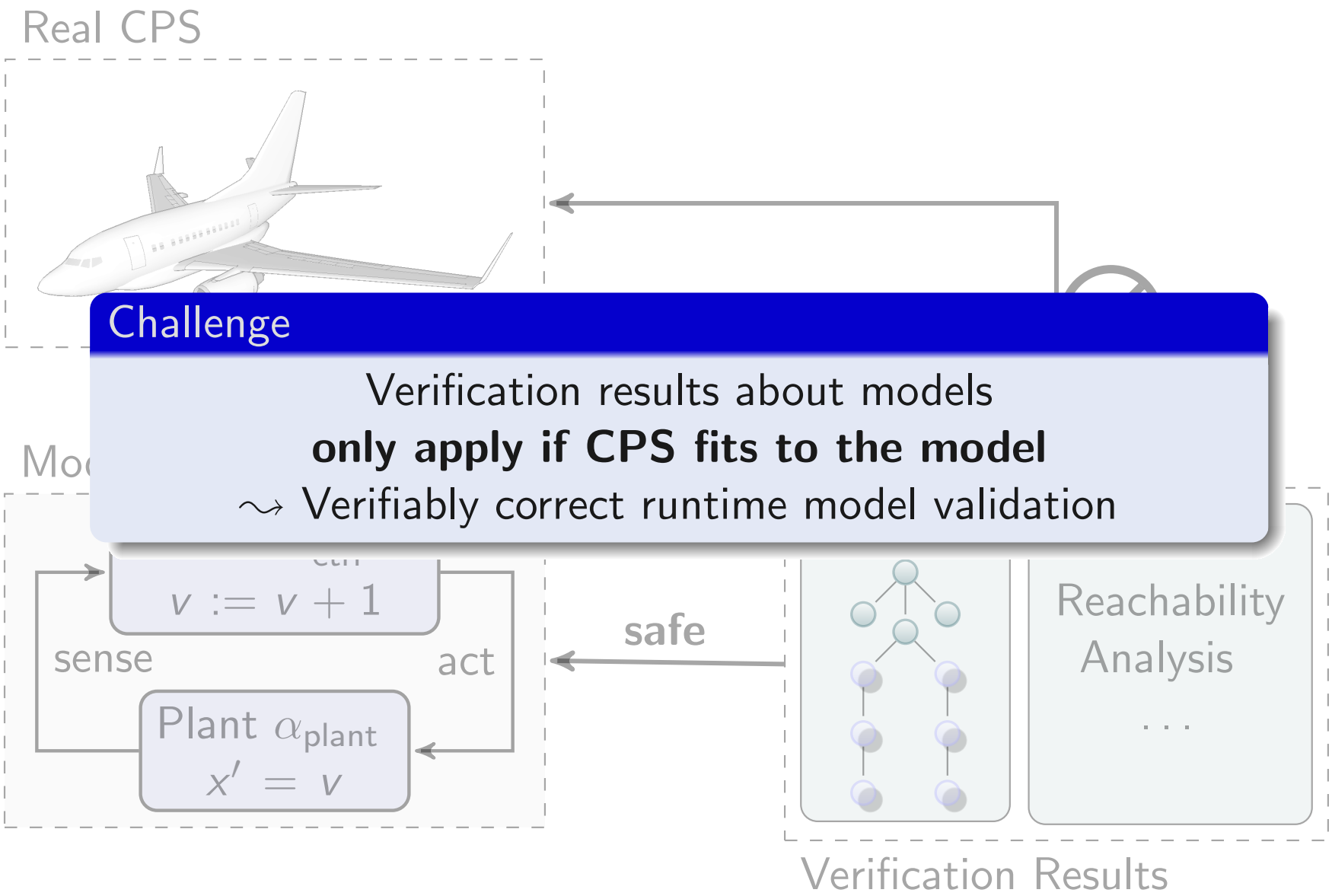
Formal Verification in CPS Development



Formal Verification in CPS Development

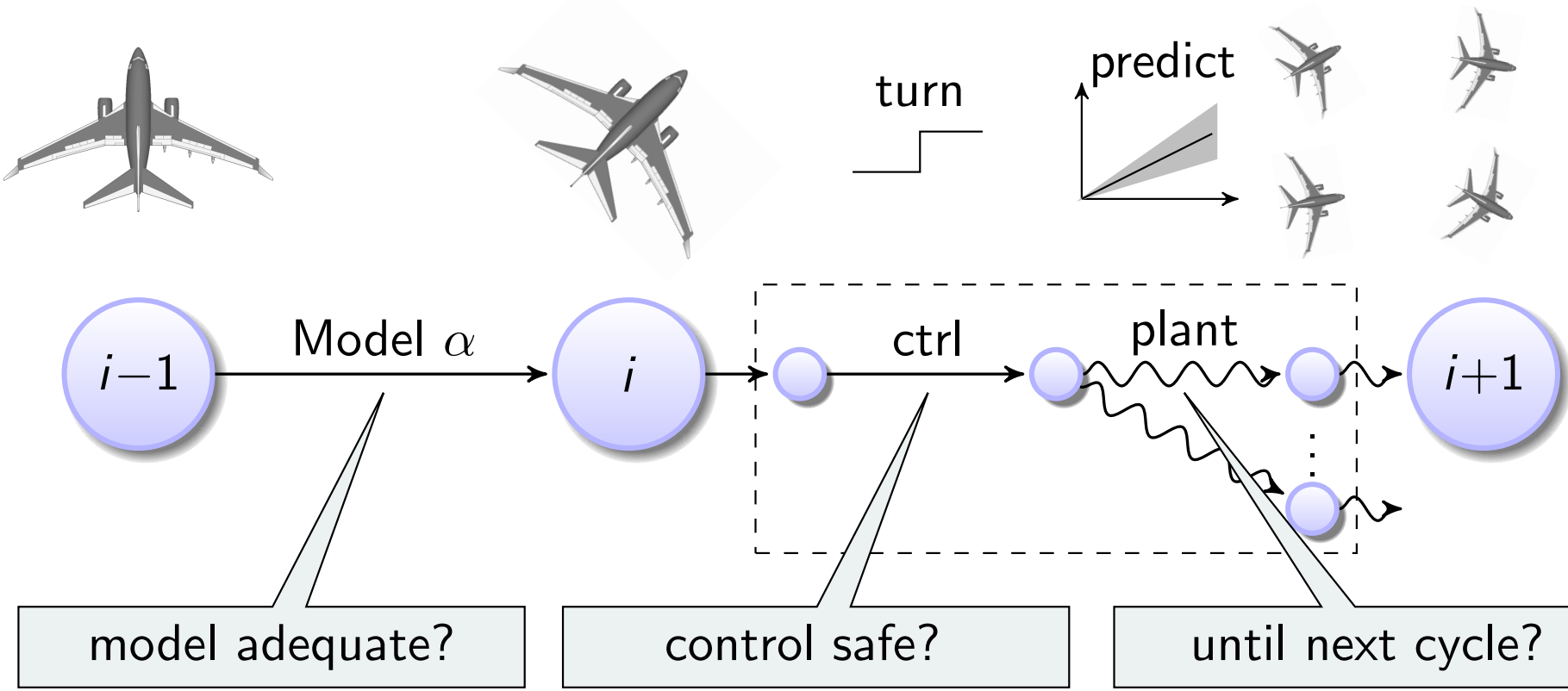


Formal Verification in CPS Development



ModelPlex Runtime Model Validation

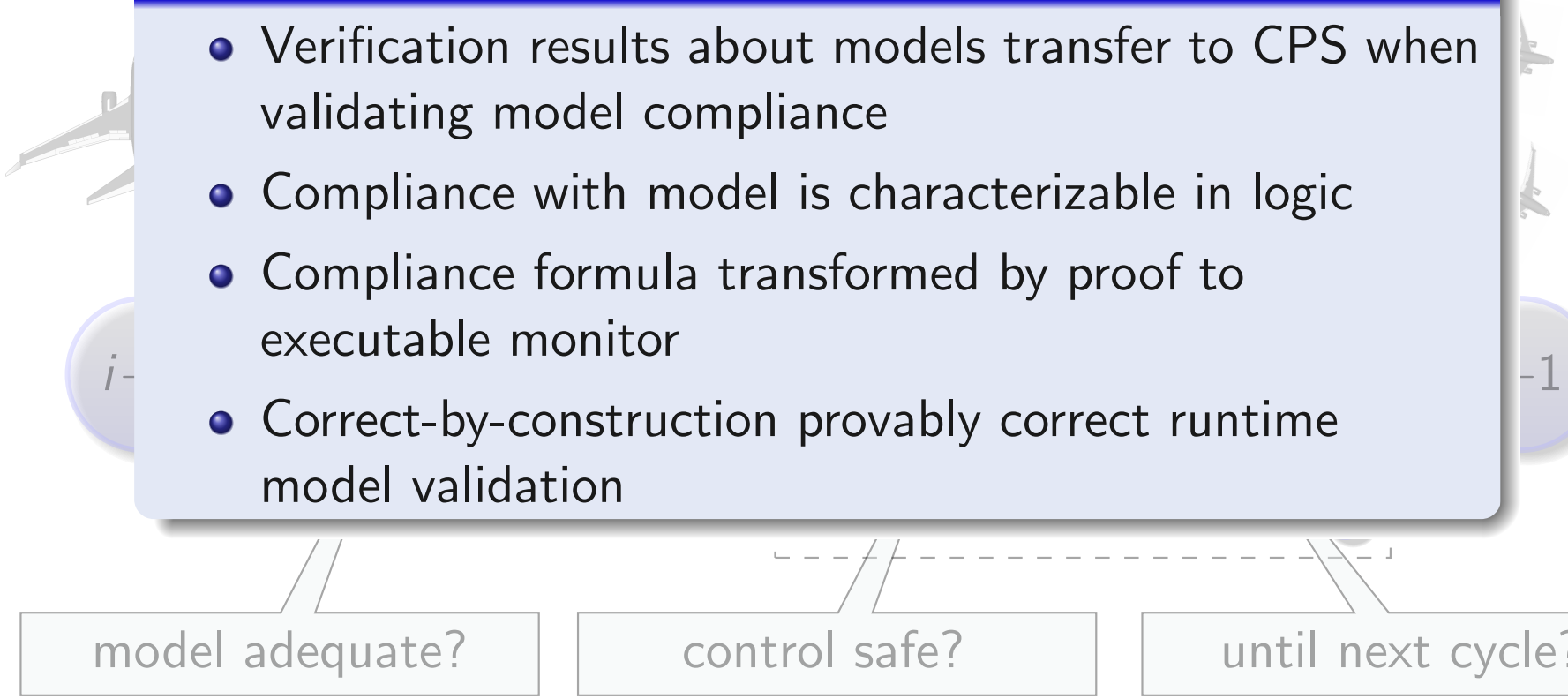
ModelPlex **ensures that verification results** about models **apply to CPS** implementations



RV'14,FMSD'16

ModelPlex Runtime Model Validation

ModelPlex **ensures that verification results** about models **apply to CPS** implementations



Contributions

- Verification results about models transfer to CPS when validating model compliance
- Compliance with model is characterizable in logic
- Compliance formula transformed by proof to executable monitor
- Correct-by-construction provably correct runtime model validation

model adequate?

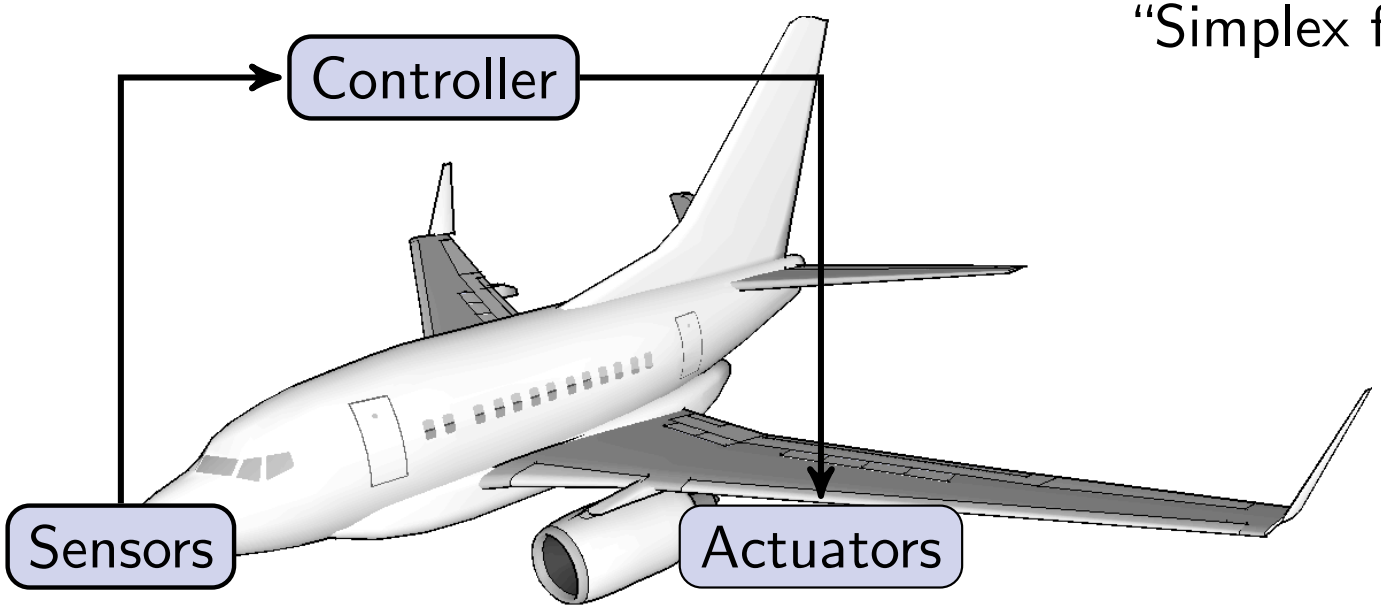
control safe?

until next cycle?

RV'14,FMSD'16

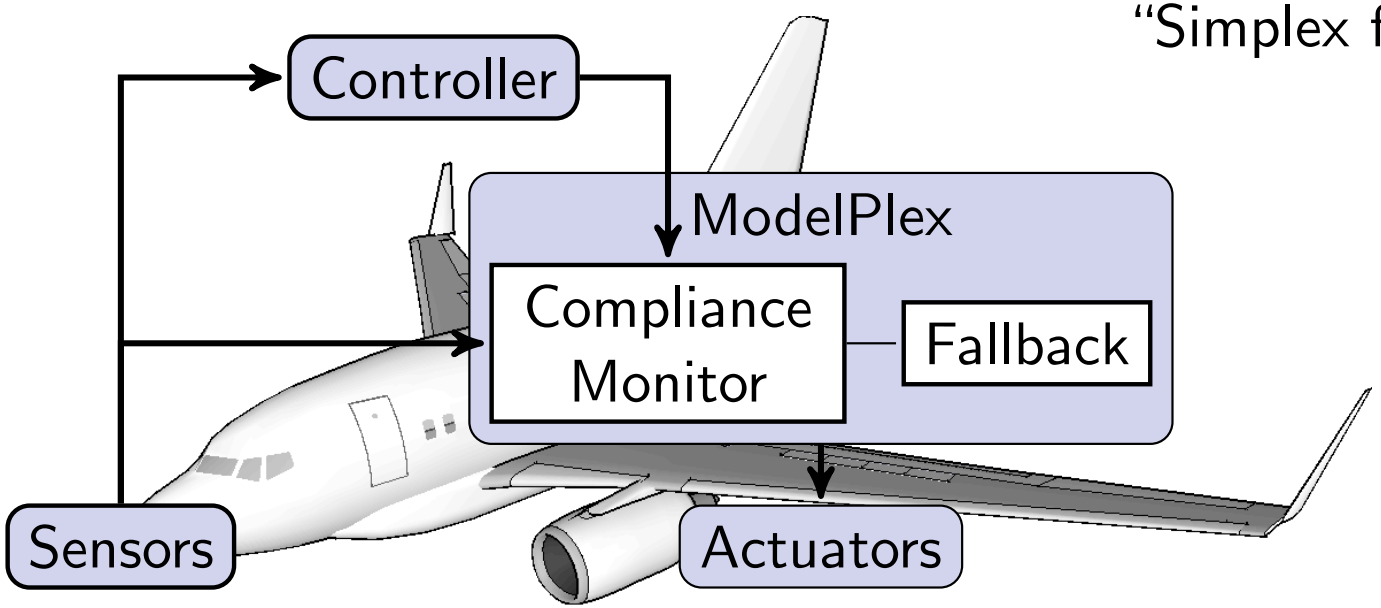
ModelPlex at Runtime

"Simplex for Models"



ModelPlex at Runtime

"Simplex for Models"



Compliance Monitor Checks CPS for compliance with model at runtime

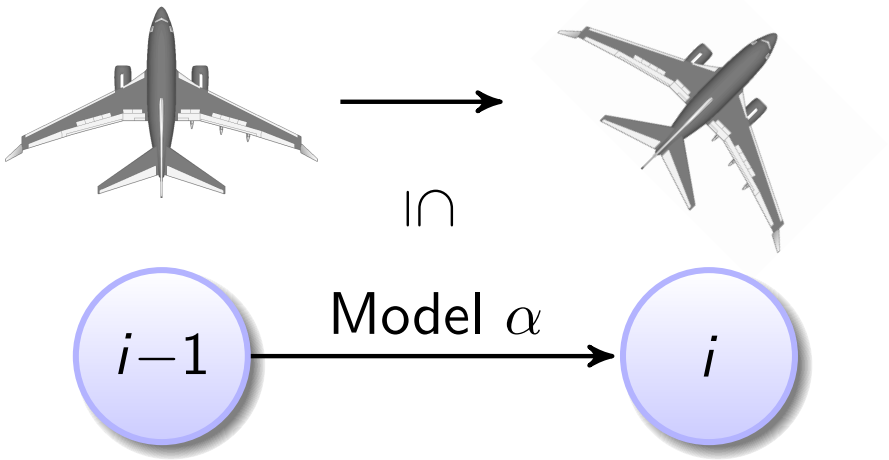
- Model Monitor: model adequate?
- Controller Monitor: control safe?
- Prediction Monitor: until next cycle?

Fallback Safe action, executed when monitor is not satisfied (veto)

Challenge What conditions do the monitors need to check to be safe?

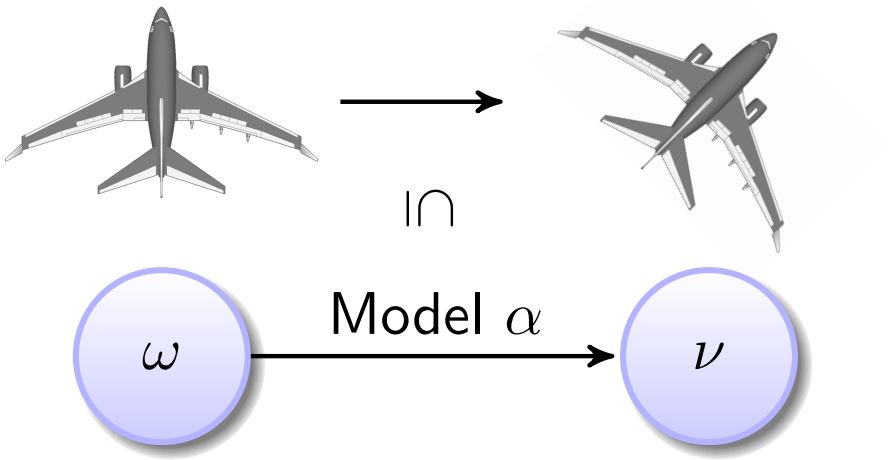
Characterizing State Relations in Logic

When are two states linked through a run of model α?



RV'14,FMSD'16

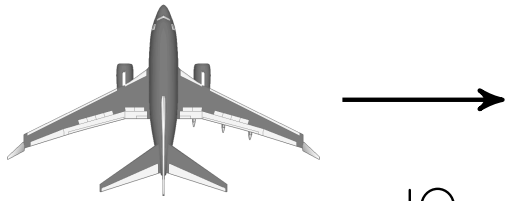
When are two states linked through a run of model α?



RV'14,FMSD'16

Characterizing State Relations in Logic

When are two states linked through a run of model α?



a prior state characterized by x^-

a posterior state characterized by x^+

ω — Model α → ν

Semantical: $(\omega, \nu) \in [\![\alpha]\!]$ ← reachability relation of α

RV'14,FMSD'16

Characterizing State Relations in Logic

When are two states linked through a run of model α?

a prior state characterized by x^-

a posterior state characterized by x^+

ω — Model α → ν

Offline

Semantical: $(\omega, \nu) \in [\![\alpha]\!]$

$\Updownarrow$ Lemma

Logical d$\mathcal{L}$: $(\omega, \nu) \models \langle\alpha\rangle(x = x^+)$

exists a run of α to a state where $x = x^+$

RV'14,FMSD'16

Characterizing State Relations in Logic

When are two states linked through a run of model α?

a prior state characterized by x^-

a posterior state characterized by x^+

ω — Model α → ν

Offline

Semantical: $(\omega, \nu) \in [\![\alpha]\!]$

$\Updownarrow$ Lemma

Logical d$\mathcal{L}$: $(\omega, \nu) \models \langle\alpha\rangle(x = x^+)$

exists a run of α to a state where $x = x^+$

$\Updownarrow$ d$\mathcal{L}$ proof

Arithmetical: $(\omega, \nu) \models F(x^-, x^+)$

check at runtime (efficient)

RV'14,FMSD'16

Characterizing State Relations in Logic

When are two states linked through a run of model α?

a prior state characterized by x^-

a posterior state characterized by x^+

ω — Model α → ν

Offline

Semantical: $(\omega, \nu) \in [\![\alpha]\!]$

$\Updownarrow$ Lemma

Logical d$\mathcal{L}$: $(\omega, \nu) \models \langle\alpha\rangle(x = x^+)$

exists a run of α to a state where $x = x^+$

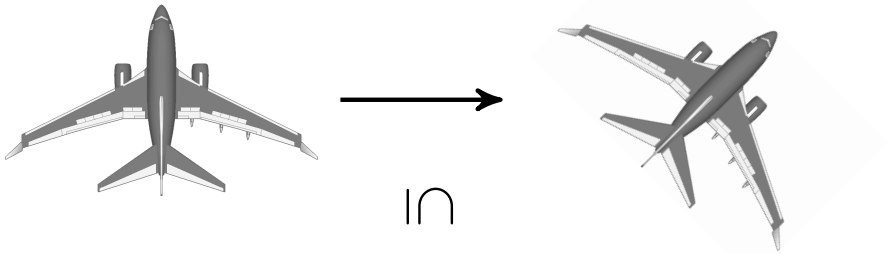
$\Uparrow$ d$\mathcal{L}$ proof

Arithmetical: $(\omega, \nu) \models F(x^-, x^+)$

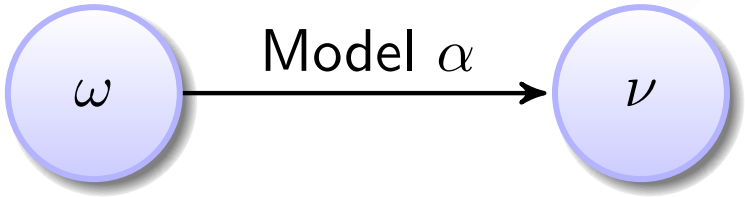
check at runtime (efficient)

RV'14,FMSD'16

Logic reduces CPS safety to runtime monitor with offline proof



d$\mathcal{L}$ proof $A \to [\alpha]S$



Offline

Semantical: $(\omega, \nu) \in [\![\alpha]\!]$

$\Updownarrow$ Lemma

Logical d$\mathcal{L}$: $(\omega, \nu) \models \langle\alpha\rangle(x = x^+)$

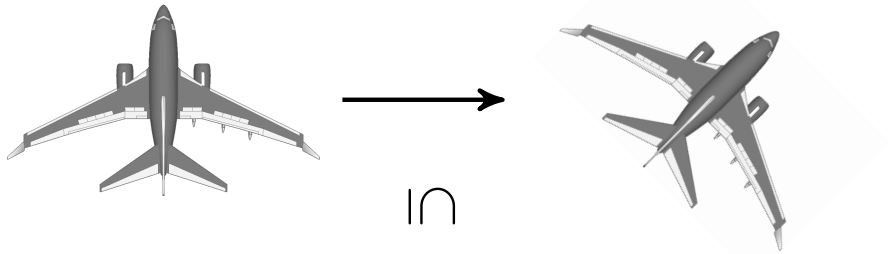
$\Uparrow$ d$\mathcal{L}$ proof

Arithmetical: $(\omega, \nu) \models F(x^-, x^+)$

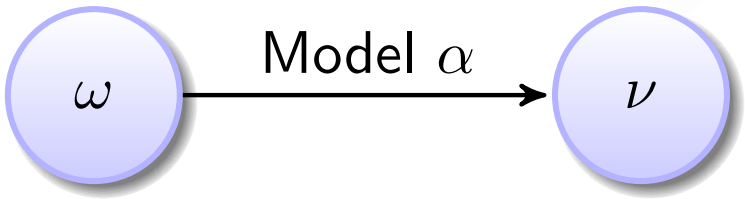
check at runtime (efficient)

RV'14,FMSD'16

Logic reduces CPS safety to runtime monitor with offline proof



d$\mathcal{L}$ proof $A \rightarrow [\alpha]S$



Offline

Init $\omega \in [\![A]\!]$

Semantical: $(\omega, \nu) \in [\![\alpha]\!]$

$\Updownarrow$ Lemma

Logical d$\mathcal{L}$: $(\omega, \nu) \models \langle\alpha\rangle(x = x^+)$

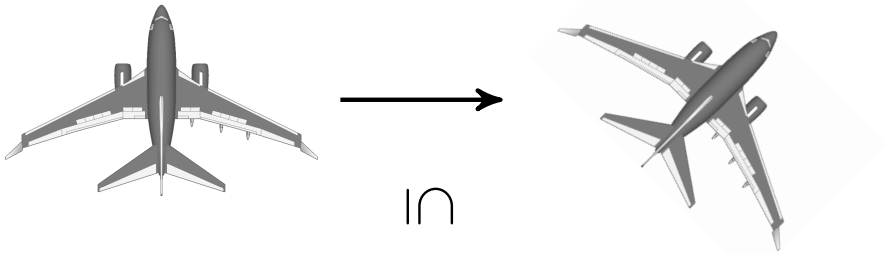
$\Uparrow$ d$\mathcal{L}$ proof

Arithmetical: $(\omega, \nu) \models F(x^-, x^+)$

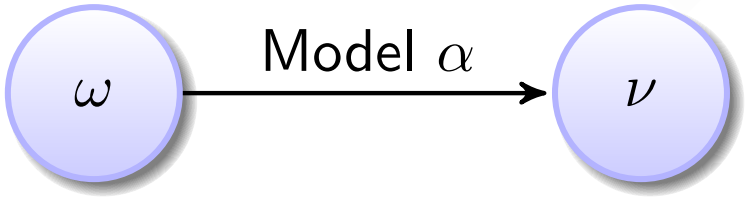
check at runtime (efficient)

RV'14,FMSD'16

Logic reduces CPS safety to runtime monitor with offline proof



d$\mathcal{L}$ proof $A \to [\alpha]S$



Offline

Init $\omega \in [\![A]\!]$ Safe $\nu \in [\![S]\!]$

Semantical: $(\omega, \nu) \in [\![\alpha]\!]$

$\Updownarrow$ Lemma

Logical d$\mathcal{L}$: $(\omega, \nu) \models \langle\alpha\rangle(x = x^+)$

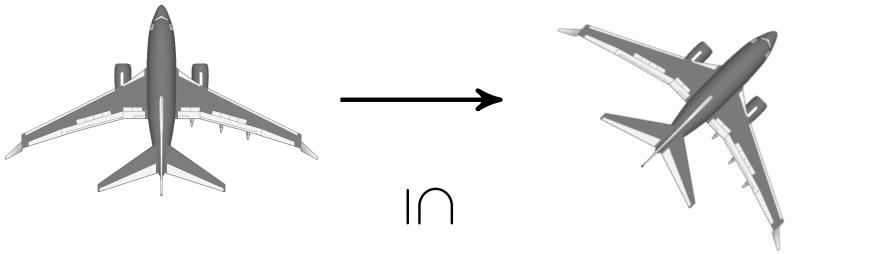
$\Uparrow$ d$\mathcal{L}$ proof

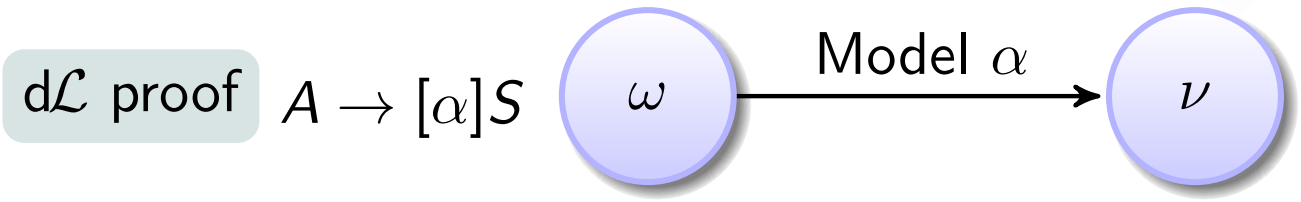
Arithmetical: $(\omega, \nu) \models F(x^-, x^+)$

check at runtime (efficient)

RV'14,FMSD'16

Logic reduces CPS safety to runtime monitor with offline proof





Offline

Init $\omega \in [\![A]\!]$

Safe $\nu \in [\![S]\!]$

Semantical: $(\omega, \nu) \in [\![\alpha]\!]$

$\Updownarrow$ Lemma

Logical dℒ: $(\omega, \nu) \models \langle\alpha\rangle(x = x^+)$

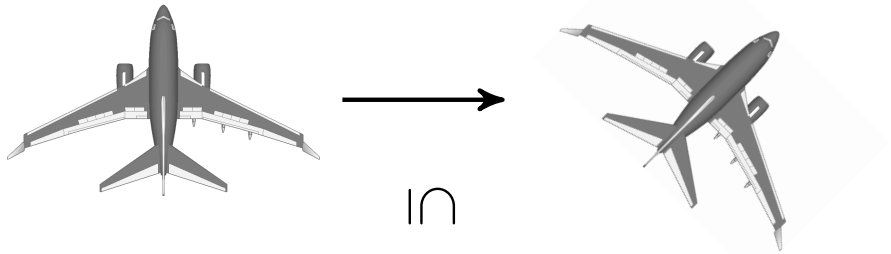
$\Uparrow$ dℒ proof

Arithmetical: $(\omega, \nu) \models F(x^-, x^+)$

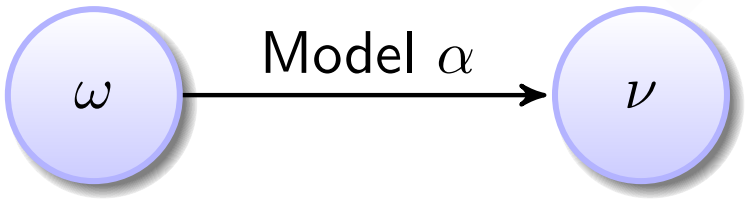
check at runtime (efficient)

RV'14,FMSD'16

Logic reduces CPS safety to runtime monitor with offline proof



d$\mathcal{L}$ proof $A \rightarrow [\alpha]S$



Offline

Init $\omega \in [\![A]\!]$ Safe $\nu \in [\![S]\!]$

Semantical: $(\omega, \nu) \in [\![\alpha]\!]$

$\Updownarrow$ Lemma

Logical d$\mathcal{L}$: $(\omega, \nu) \models \langle\alpha\rangle(x = x^+)$

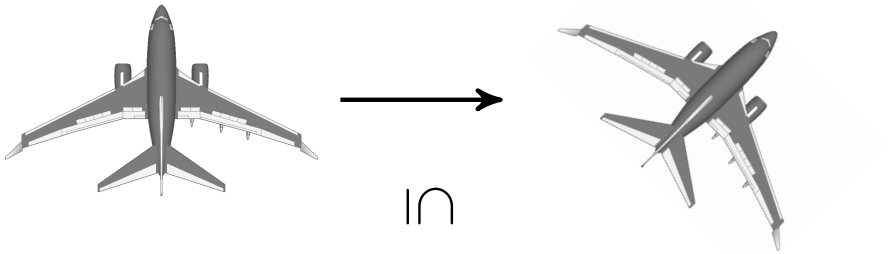
$\Uparrow$ d$\mathcal{L}$ proof

Arithmetical: $(\omega, \nu) \models F(x^-, x^+)$

check at runtime (efficient)

RV'14,FMSD'16

Logic reduces CPS safety to runtime monitor with offline proof



dℒ proof $A \to [\alpha]S$ — ω — Model α — ν

Offline

Init $\omega \in [\![A]\!]$ Safe $\nu \in [\![S]\!]$

Semantical: $(\omega, \nu) \in [\![\alpha]\!]$

$\Updownarrow$ Lemma

Logical dℒ: $(\omega, \nu) \models \langle\alpha\rangle(x = x^+)$

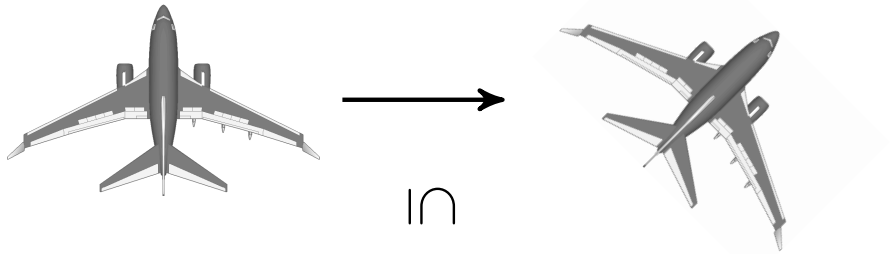
$\Uparrow$ dℒ proof

Arithmetical: $(\omega, \nu) \models F(x^-, x^+)$

check at runtime (efficient)

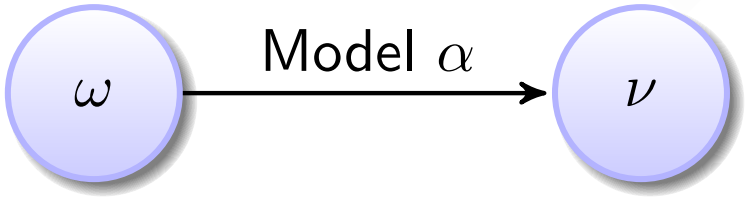
RV'14,FMSD'16

Logic reduces CPS safety to runtime monitor with offline proof



$\models$

dL proof $A \to [\alpha]S$



Offline

Init $\omega \in [\![A]\!]$

Safe $\nu \in [\![S]\!]$

Semantical: $(\omega, \nu) \in [\![\alpha]\!]$

$\Updownarrow$ Lemma

Logical dL: $(\omega, \nu) \models \langle\alpha\rangle(x = x^+)$

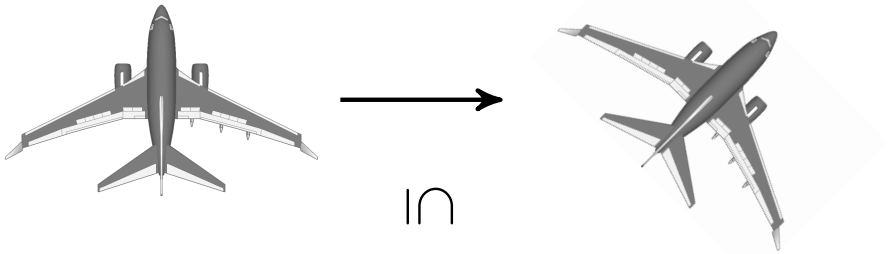
$\Uparrow$ dL proof

Arithmetical: $(\omega, \nu) \models F(x^-, x^+)$

check at runtime (efficient)

RV'14,FMSD'16

Logic reduces CPS safety to runtime monitor with offline proof



dℒ proof $A \to [\alpha]S$ — ω — Model α — ν

Offline

Init $\omega \in [\![A]\!]$ Safe $\nu \in [\![S]\!]$

Semantical: $(\omega, \nu) \in [\![\alpha]\!]$

$\Updownarrow$ Lemma

Logical dℒ: $(\omega, \nu) \models \langle\alpha\rangle(x = x^+)$

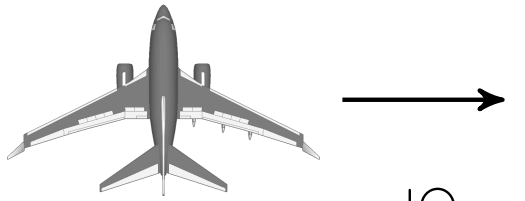
$\Uparrow$ dℒ proof

Arithmetical: $(\omega, \nu) \models F(x^-, x^+)$

check at runtime (efficient)

RV'14,FMSD'16

Logic reduces CPS safety to runtime monitor with offline proof



Not initial state.
Model repeats...

$\in$

dℒ proof $A \to [\alpha]S$

ω — Model α → ν

Offline

Init $\omega \in [\![A]\!]$ Safe $\nu \in [\![S]\!]$

Semantical: $(\omega, \nu) \in [\![\alpha]\!]$

$\Updownarrow$ Lemma

Logical dℒ: $(\omega, \nu) \models \langle\alpha\rangle(x = x^+)$

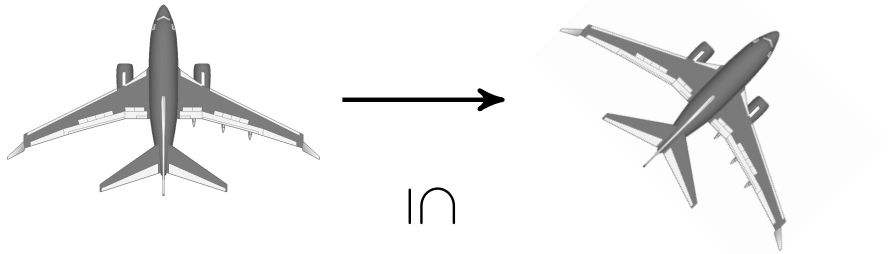
$\Uparrow$ dℒ proof

Arithmetical: $(\omega, \nu) \models F(x^-, x^+)$

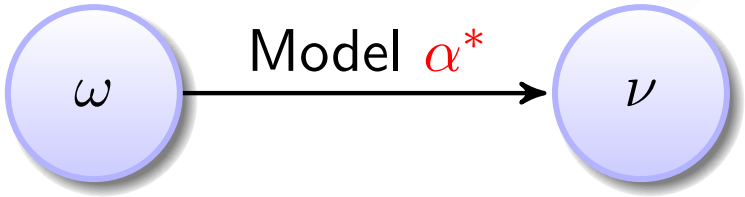
check at runtime (efficient)

RV'14,FMSD'16

Logic reduces CPS safety to runtime monitor with offline proof



d$\mathcal{L}$ proof $A \rightarrow [\alpha^*]S$



Offline

Init $\omega \in [\![A]\!]$ Safe $\nu \in [\![S]\!]$

Semantical: $(\omega, \nu) \in [\![\alpha^*]\!]$

$\Updownarrow$ Lemma

Logical d$\mathcal{L}$: $(\omega, \nu) \models \langle\alpha^*\rangle(x = x^+)$

$\Uparrow$ d$\mathcal{L}$ proof

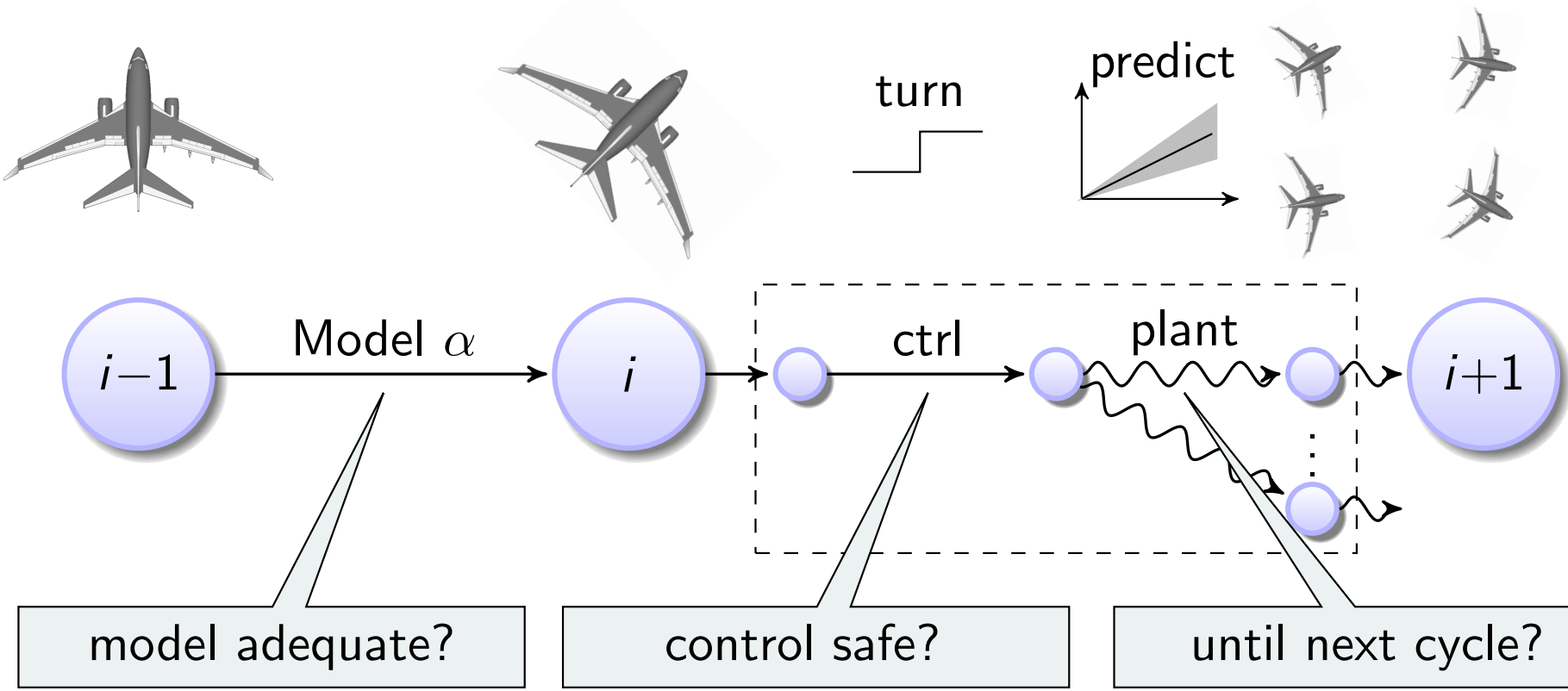
Arithmetical: $(\omega, \nu) \models F(x^-, x^+)$

check at runtime (efficient)

RV'14,FMSD'16

ModelPlex Runtime Model Validation

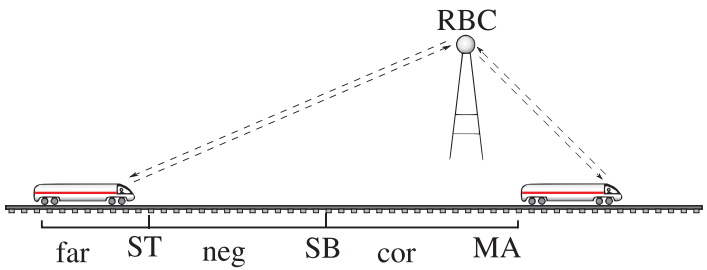
ModelPlex **ensures that verification results** about models **apply to CPS** implementations

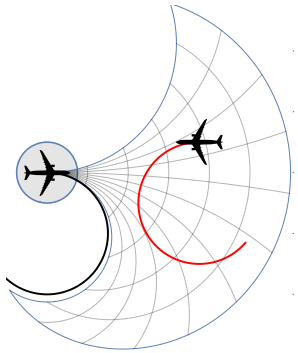


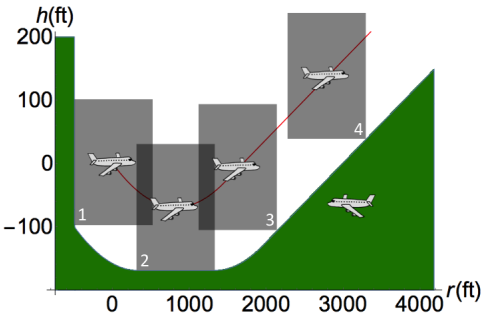
RV'14,FMSD'16

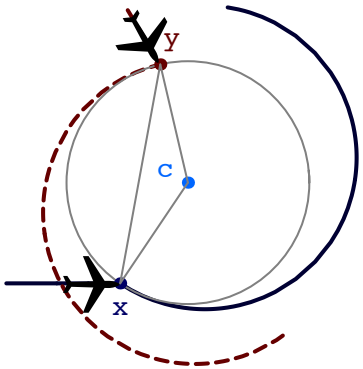
Outline

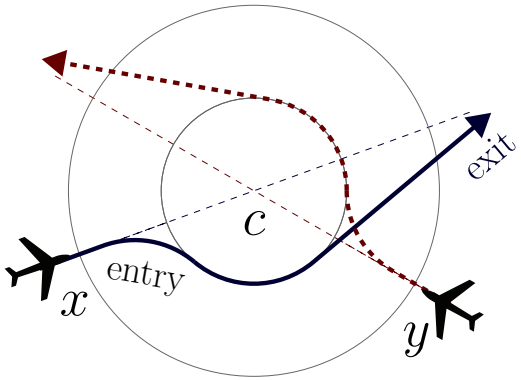
Verified CPS Applications

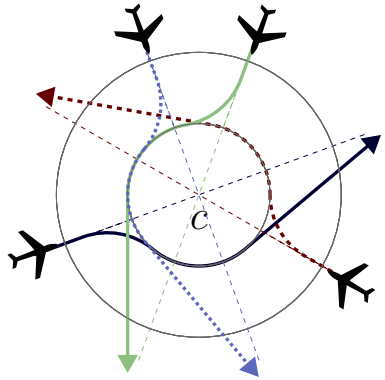


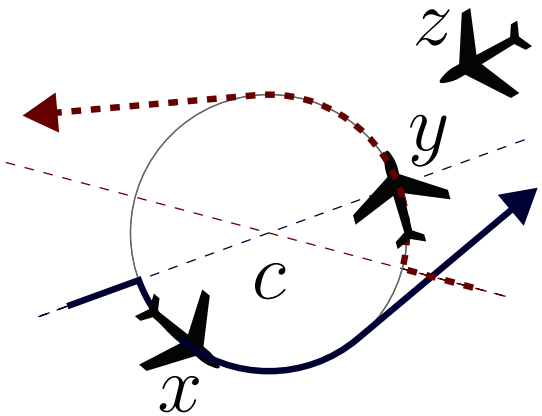


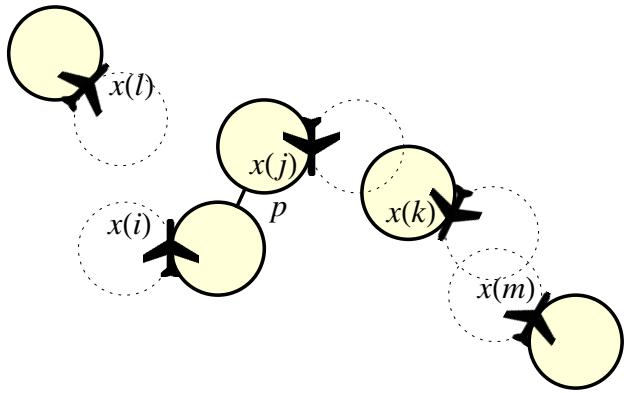


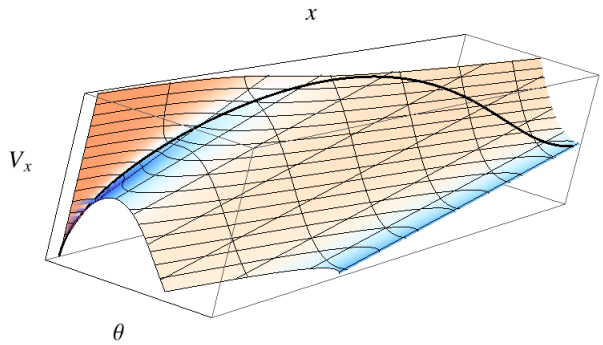






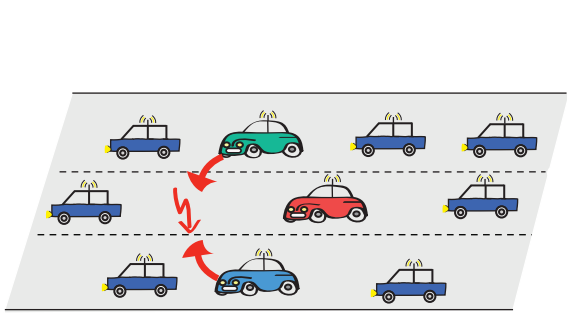


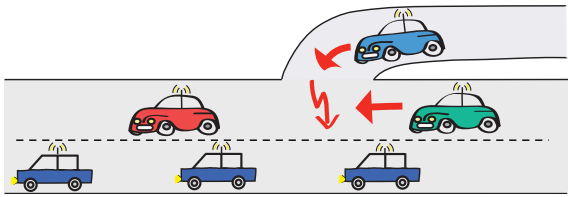


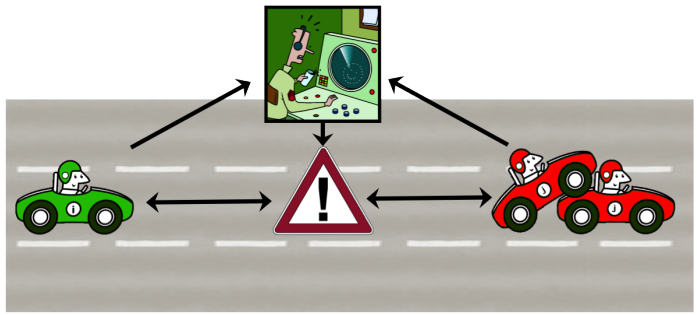


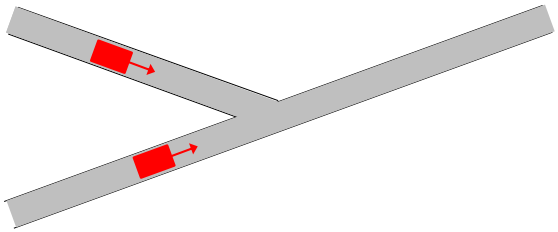
FEM'09,JAIS'14,TACAS'15,EMSOFT'15,CAV'08,FM'09,HSCC'11,HSCC'13,TACAS'14

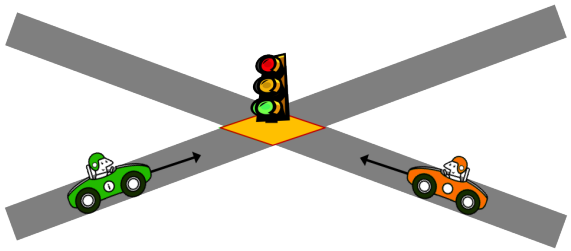
Verified CPS Applications

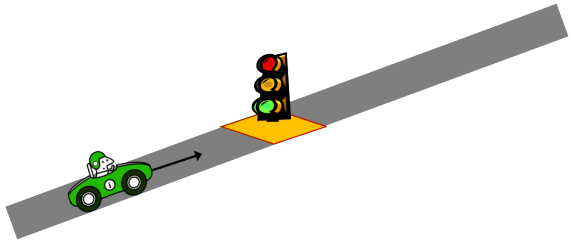


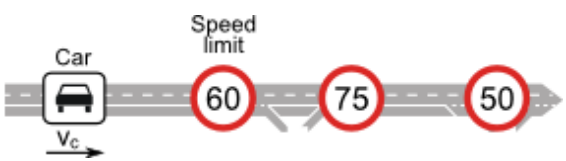


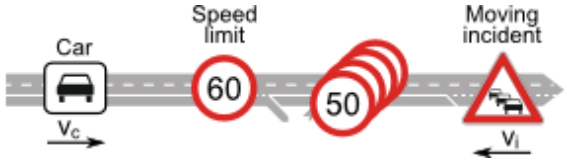


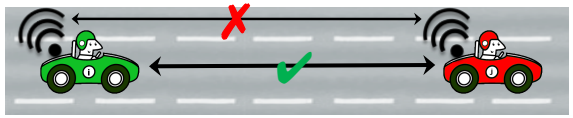


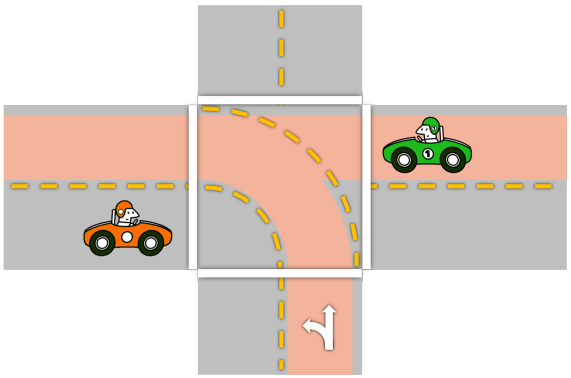


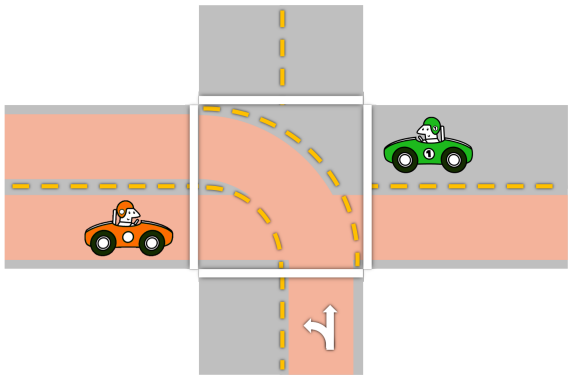


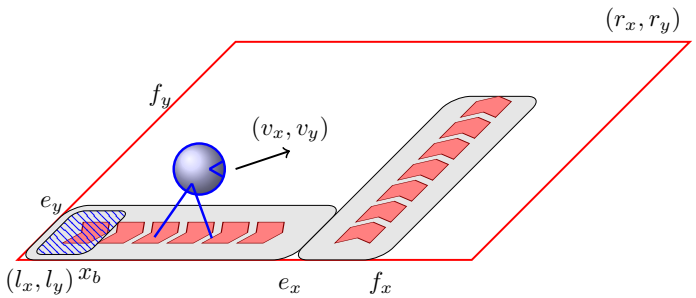






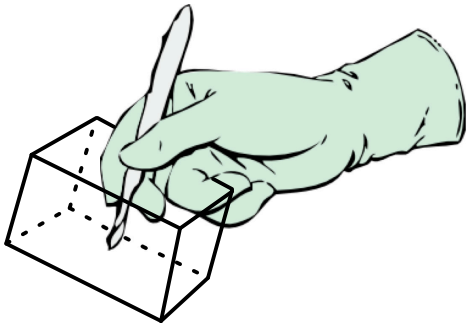


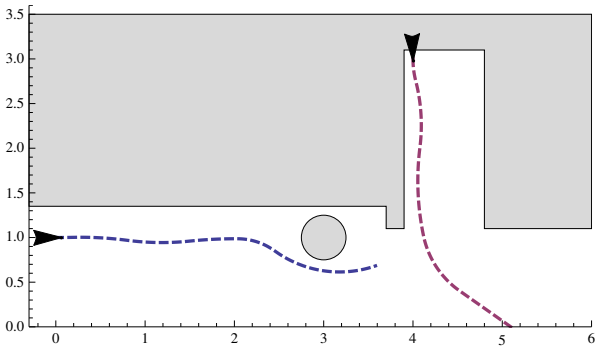


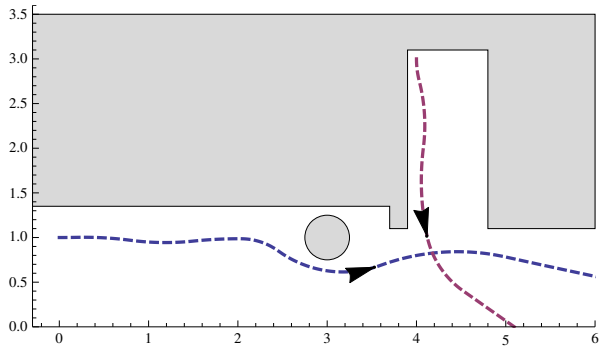


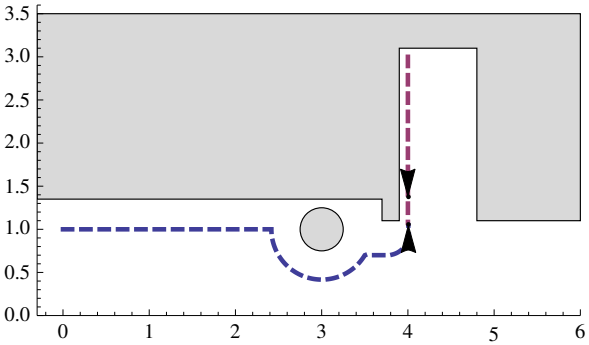
FM'11,LMCS'12,ICCPS'12,ITSC'11,ITSC'13,IJCAR'12

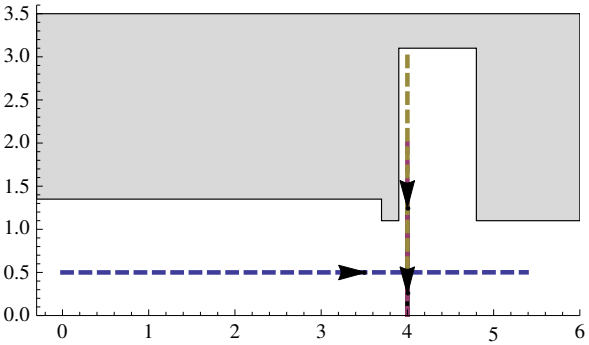
Verified CPS Applications

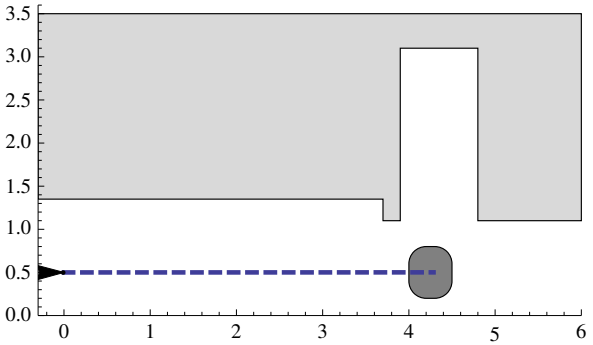


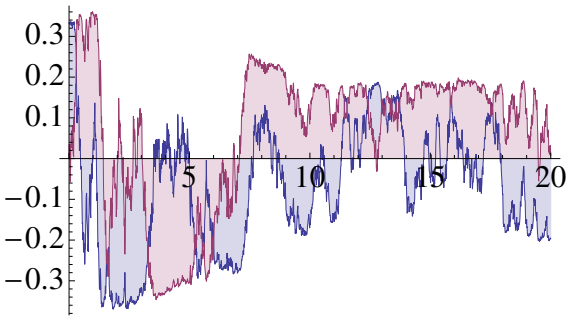


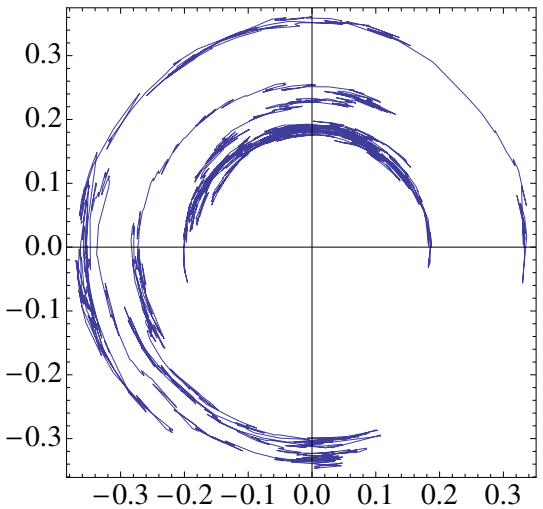


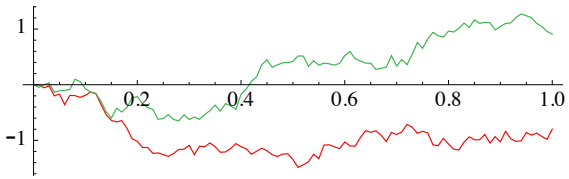




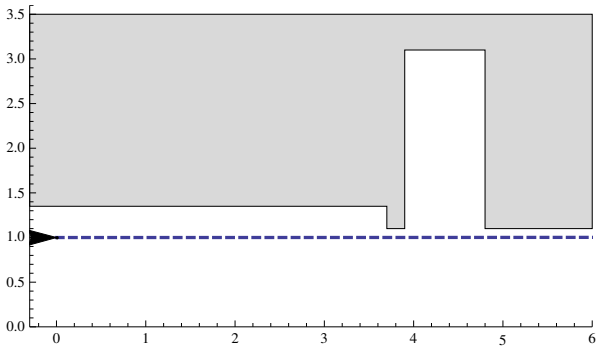
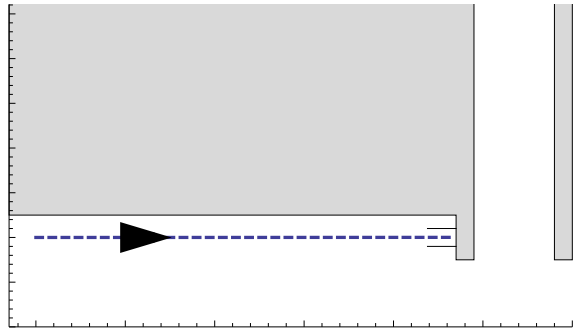
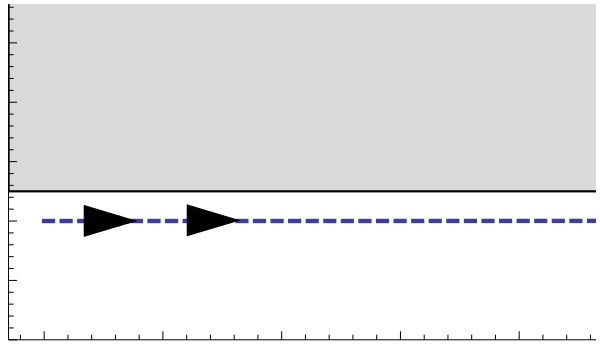
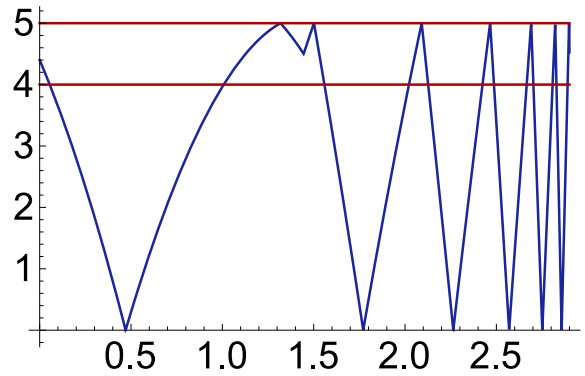
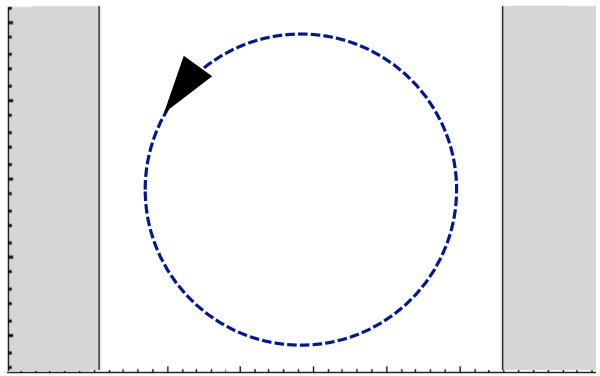
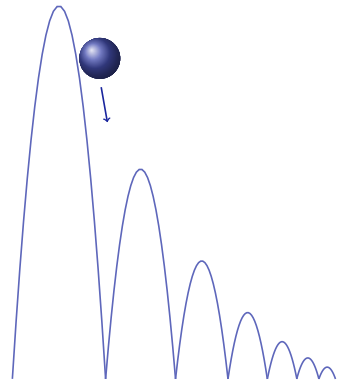
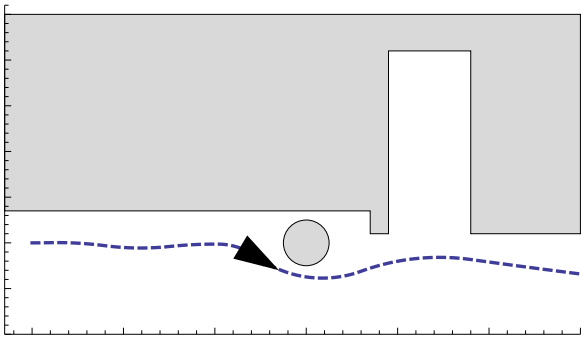
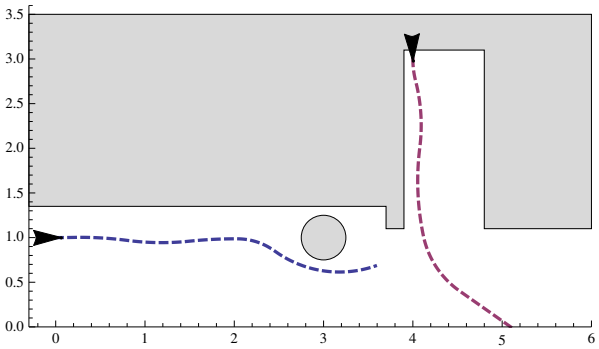
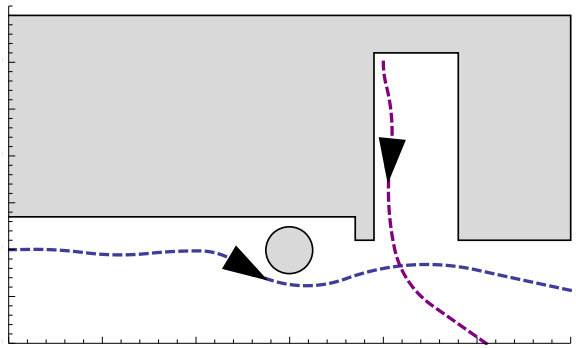








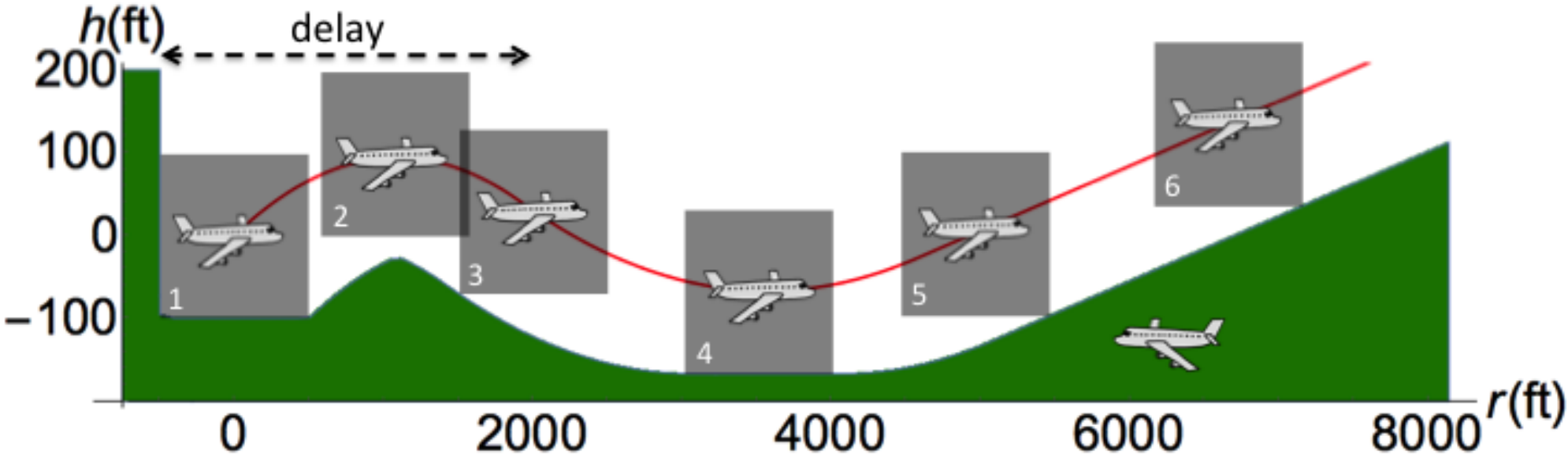
HSCC'13,RSS'13,CADE'12

15-424/624/824 *Foundations of Cyber-Physical Systems* students

Airborne Collision Avoidance System ACAS X: Verify

- Developed by the FAA to replace current TCAS in aircraft
- Approximately optimizes Markov Decision Process on a grid
- Advisory from lookup tables with numerous 5D interpolation regions

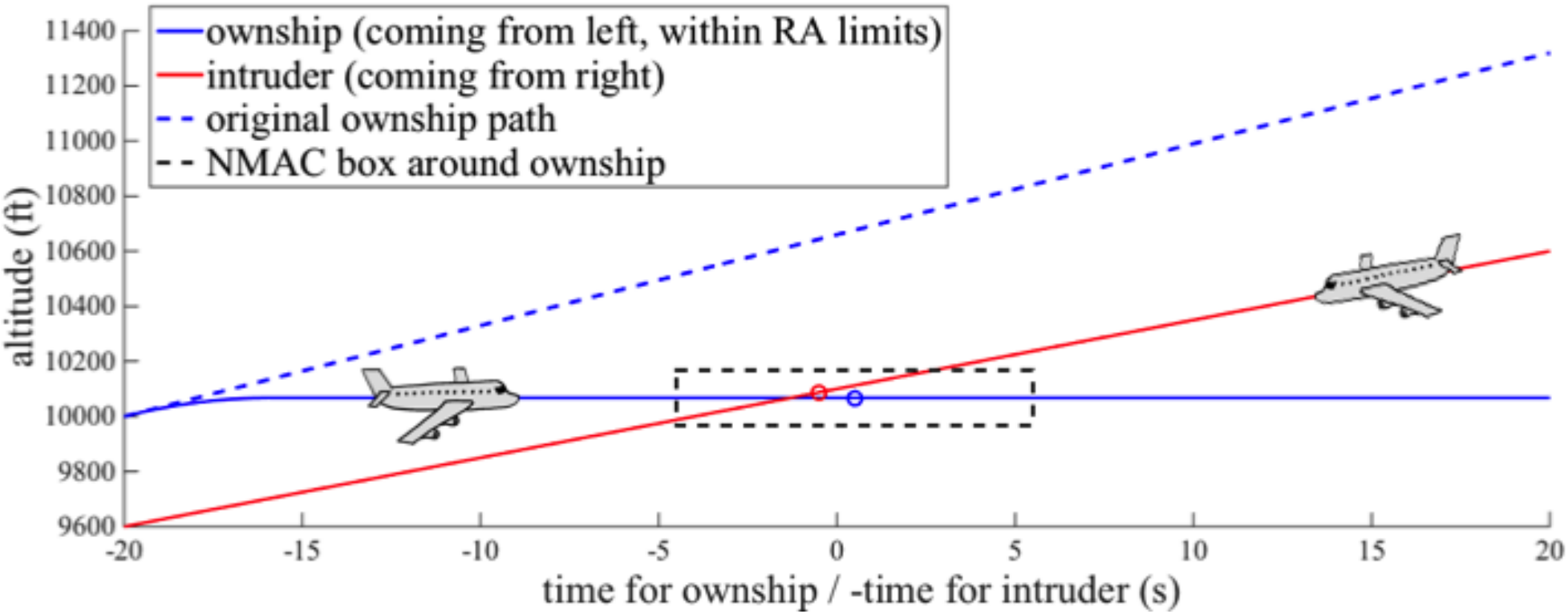


1. Identified safe region for each advisory symbolically
2. Proved safety for hybrid systems flight model in KeYmaera X

TACAS'15,EMSOFT'15,STTT'16

Airborne Collision Avoidance System ACAS X: Compare

ACAS X table comparison shows safe advisory in 97.7% of the 648,591,384,375 states compared (15,160,434,734 counterexamples).

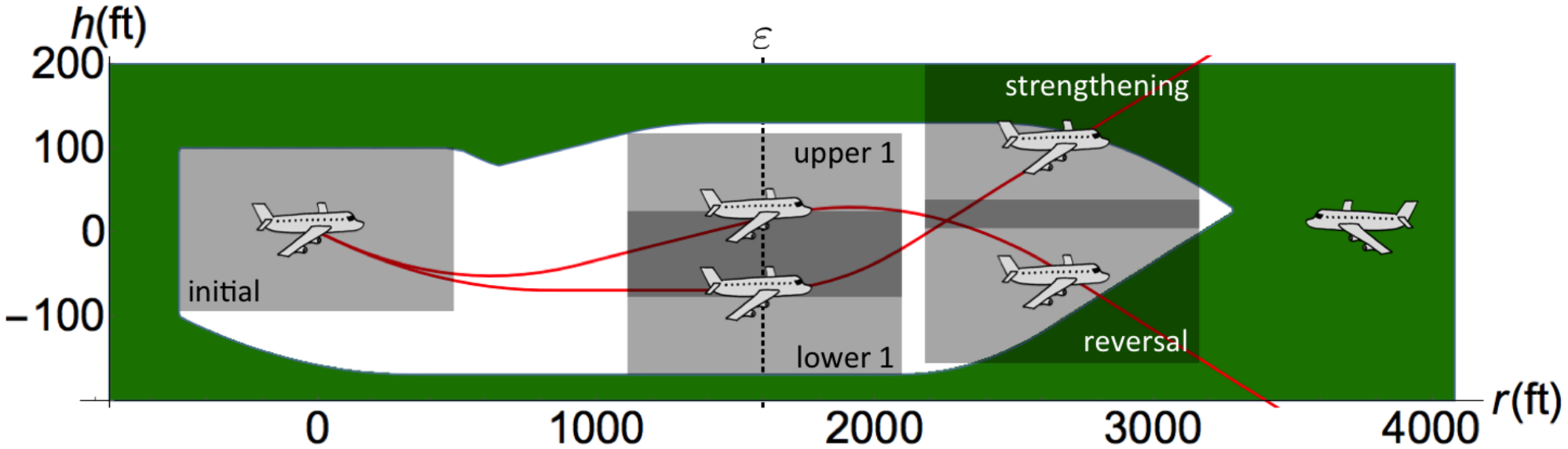


ACAS X issues DNC advisory, which induces collision unless corrected

TACAS'15,EMSOFT'15,STTT'16

Airborne Collision Avoidance System ACAS X: Refine

- Conservative, so too many counterexamples
- Settle for: safe for a little while, with safe future advisory possibility
- Safeable advisory: a subsequent advisory can safely avoid collision

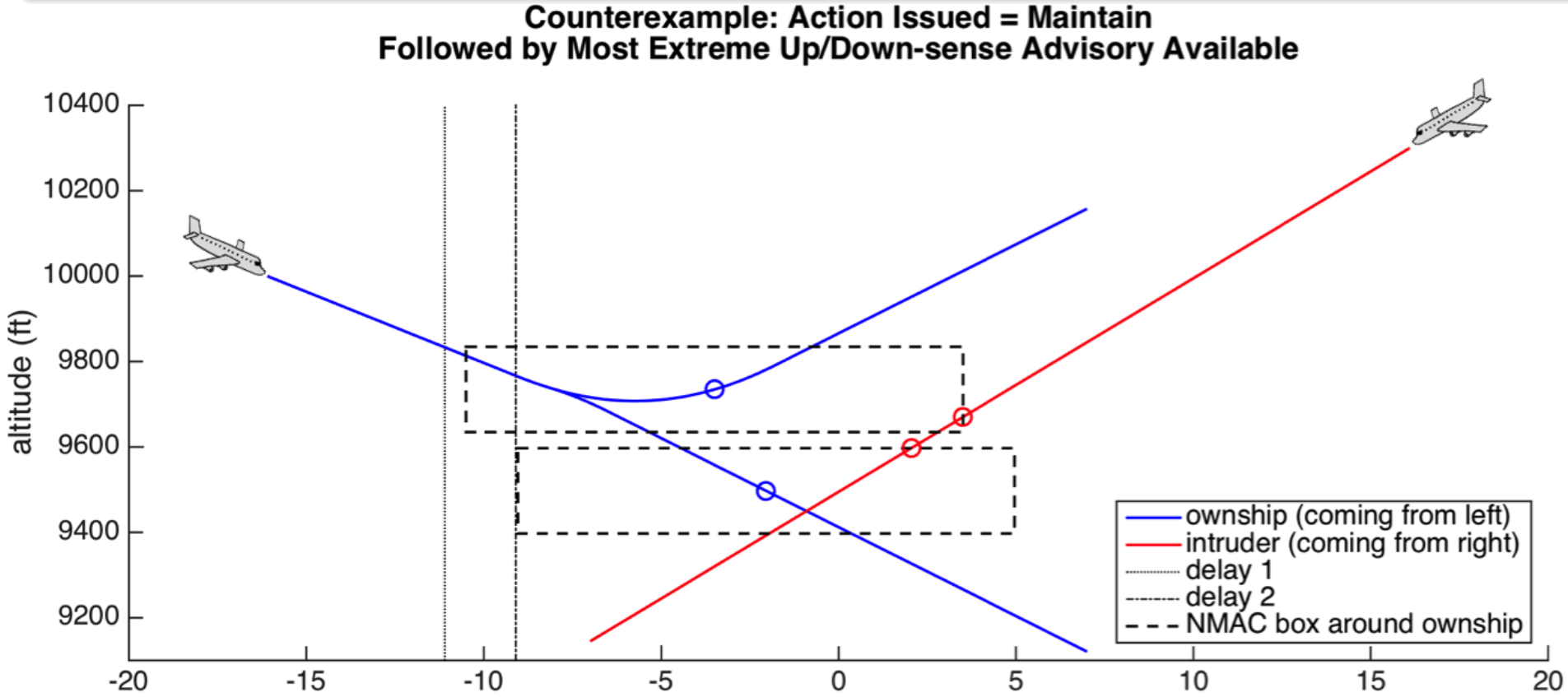


1. Identified safeable region for each advisory symbolically
2. Proved safety for hybrid systems flight model in KeYmaera X

STTT'16

Airborne Collision Avoidance System ACAS X: Compare

ACAS X table comparison shows safeable advisory in more of the 648,591,384,375 states compared (31 to 899 10^6 counterexamples).

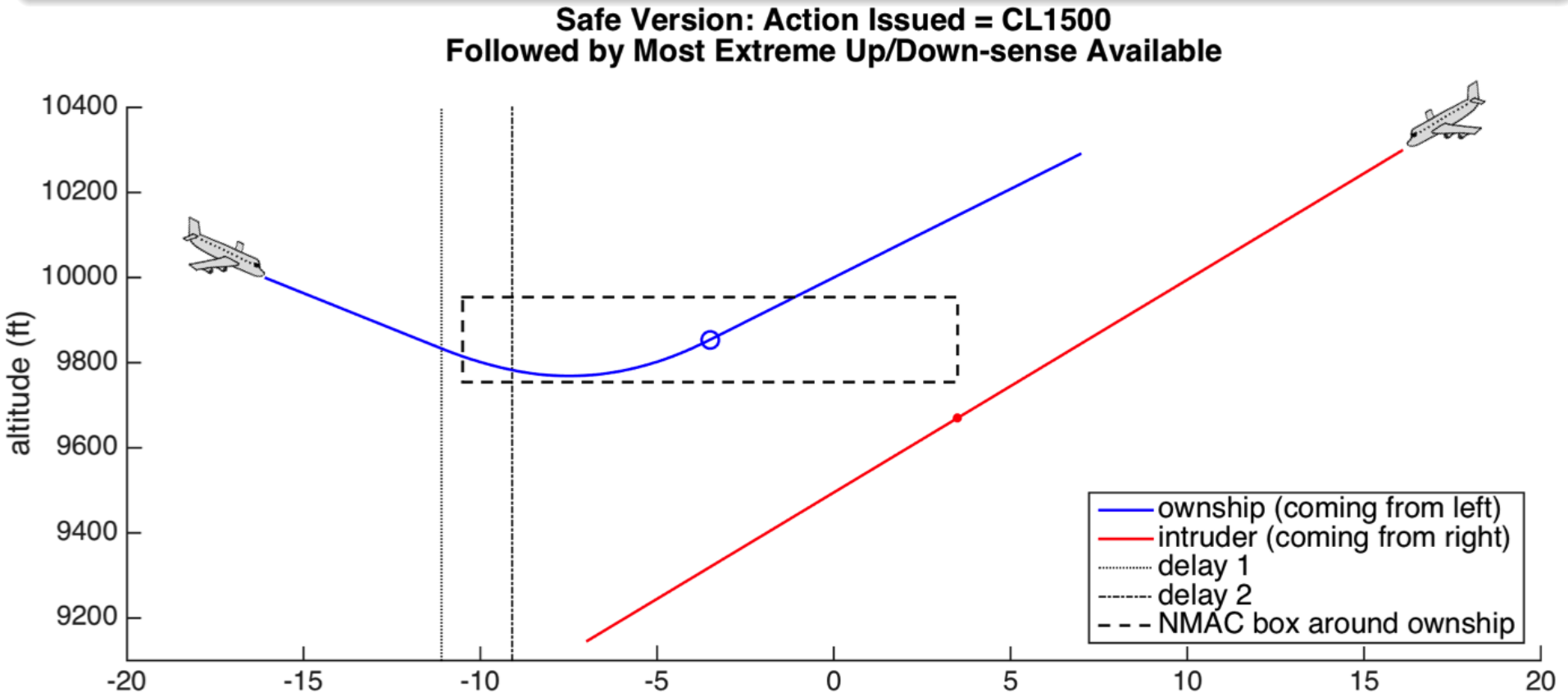


ACAS X issues Maintain advisory instead of CL1500

STTT'16

Airborne Collision Avoidance System ACAS X: Compare

ACAS X table comparison shows safeable advisory in more of the 648,591,384,375 states compared (31 to 899 10^6 counterexamples).

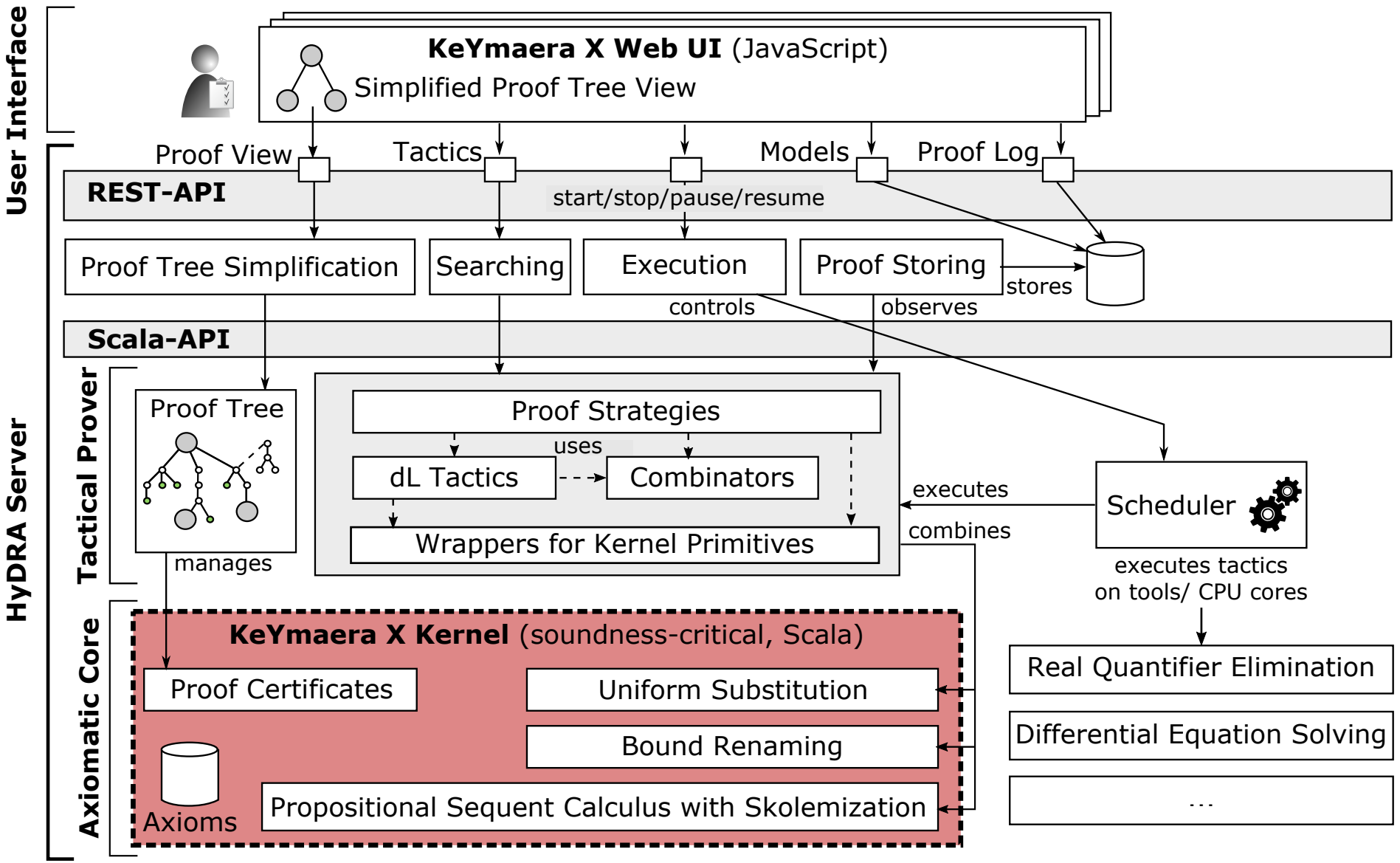


ACAS X issues Maintain advisory instead of CL1500

STTT'16

Outline

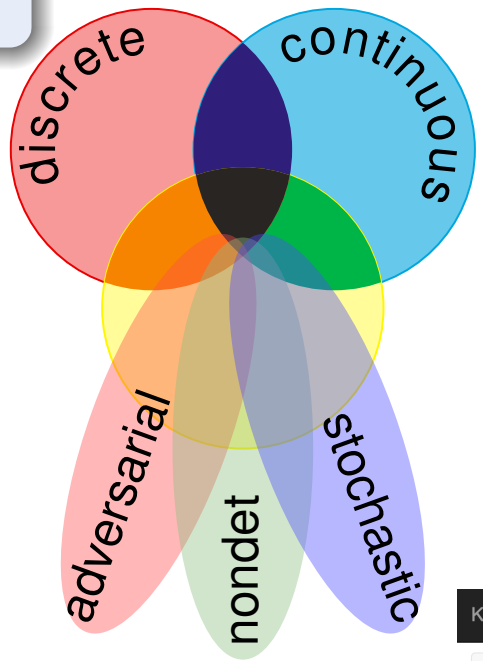
KeYmaera X Tool Architecture



KeYmaera X Theorem Prover for Hybrid Systems

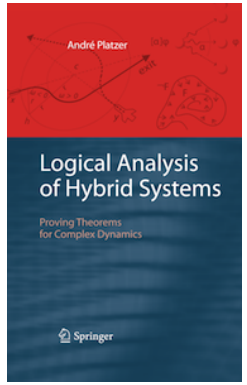
differential dynamic logic

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

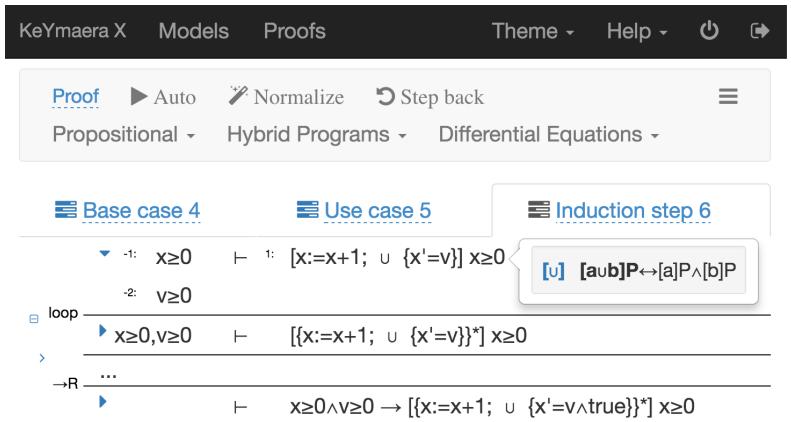


- Multi-dynamical systems
- Compositional
- Logic & proofs for CPS
- Small soundness core
- Proof by pointing
- Interactive proof clicking
- Tactical proof programming
- Proof search tactics
- Flexible + modular API

FCPS course
STTT'16 tutorial
Book



KeYmaera X




André Platzer.
Logics of dynamical systems.
In LICS [17], pages 13–24.
doi:10.1109/LICS.2012.13.


André Platzer.
A complete uniform substitution calculus for differential dynamic logic.

J. Autom. Reas., 2016.
doi:10.1007/s10817-016-9385-1.


Stefan Mitsch and André Platzer.
ModelPlex: Verified runtime validation of verified cyber-physical system models.
Form. Methods Syst. Des., 49(1):33–74, 2016.
Special issue of selected papers from RV'14.
doi:10.1007/s10703-016-0241-z.


Nathan Fulton and André Platzer.

A logic of proofs for differential dynamic logic: Toward independently checkable proof certificates for dynamic logics.
In Jeremy Avigad and Adam Chlipala, editors, *Proceedings of the 2016 Conference on Certified Programs and Proofs, CPP 2016, St. Petersburg, FL, USA, January 18-19, 2016*, pages 110–121. ACM, 2016.
doi:10.1145/2854065.2854078.


André Platzer.
Logic & proofs for cyber-physical systems.
In Nicola Olivetti and Ashish Tiwari, editors, *IJCAR*, volume 9706 of *LNCS*, pages 15–21. Springer, 2016.
doi:10.1007/978-3-319-40229-1_3.


André Platzer.
Differential dynamic logic for hybrid systems.
J. Autom. Reas., 41(2):143–189, 2008.
doi:10.1007/s10817-008-9103-8.


André Platzer.
Differential game logic.

ACM Trans. Comput. Log., 17(1):1:1–1:51, 2015.
doi:10.1145/2817824.

André Platzer.
The complete proof theory of hybrid systems.
In LICS [17], pages 541–550.
doi:10.1109/LICS.2012.64.

André Platzer.
Differential-algebraic dynamic logic for differential-algebraic programs.
J. Log. Comput., 20(1):309–352, 2010.
doi:10.1093/logcom/exn070.

André Platzer and Edmund M. Clarke.
Computing differential invariants of hybrid systems as fixedpoints.
Form. Methods Syst. Des., 35(1):98–120, 2009.
Special issue for selected papers from CAV'08.
doi:10.1007/s10703-009-0079-8.

André Platzer.
The structure of differential invariants and differential cut elimination.

Log. Meth. Comput. Sci., 8(4):1–38, 2012.
doi:10.2168/LMCS-8(4:16)2012.

André Platzer.
A differential operator approach to equational differential invariants.
In Lennart Beringer and Amy Felty, editors, *ITP*, volume 7406 of *LNCS*, pages 28–48. Springer, 2012.
doi:10.1007/978-3-642-32347-8_3.

Jean-Baptiste Jeannin, Khalil Ghorbal, Yanni Kouskoulas, Ryan Gardner, Aurora Schmidt, Erik Zawadzki, and André Platzer.
A formally verified hybrid system for the next-generation airborne collision avoidance system.
In Christel Baier and Cesare Tinelli, editors, *TACAS*, volume 9035 of *LNCS*, pages 21–36. Springer, 2015.
doi:10.1007/978-3-662-46681-0_2.

Jean-Baptiste Jeannin, Khalil Ghorbal, Yanni Kouskoulas, Ryan Gardner, Aurora Schmidt, Erik Zawadzki, and André Platzer.
Formal verification of ACAS X, an industrial airborne collision avoidance system.

In Alain Girault and Nan Guan, editors, *EMSOFT*, pages 127–136. IEEE, 2015.
doi:10.1109/EMSOFT.2015.7318268.

Jean-Baptiste Jeannin, Khalil Ghorbal, Yanni Kouskoulas, Aurora Schmidt, Ryan Gardner, Stefan Mitsch, and André Platzer.
A formally verified hybrid system for safe advisories in the next-generation airborne collision avoidance system.
STTT, 2016.
doi:10.1007/s10009-016-0434-1.

André Platzer.
Logical Analysis of Hybrid Systems: Proving Theorems for Complex Dynamics.
Springer, Heidelberg, 2010.
doi:10.1007/978-3-642-14509-4.

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