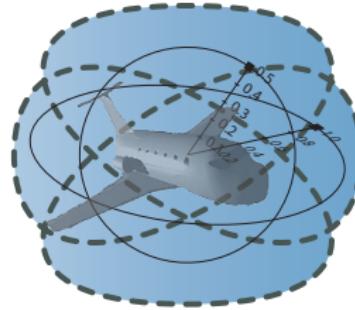


# Logical Foundations & Cyber-Physical Systems

André Platzer

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

<http://symbolaris.com/>



# A Outline

## 1 CPS are Multi-Dynamical Systems

- Hybrid Systems
- Hybrid Games

## 2 Dynamic Logic for Multi-Dynamical Systems

- Syntax
- Semantics

## 3 Proofs for CPS

## 4 Theory of CPS

- Soundness and Completeness
- Differential Invariants
- Examples
- Differential Radical Invariants

## 5 Applications

## 6 Summary

## 1 CPS are Multi-Dynamical Systems

- Hybrid Systems
- Hybrid Games

## 2 Dynamic Logic for Multi-Dynamical Systems

- Syntax
- Semantics

## 3 Proofs for CPS

## 4 Theory of CPS

- Soundness and Completeness
- Differential Invariants
- Examples
- Differential Radical Invariants

## 5 Applications

## 6 Summary

# Can you trust a computer to control physics?

# Can you trust a computer to control physics?

## Rationale

- ① Safety guarantees require analytic foundations.
- ② Foundations revolutionized digital computer science & our society.
- ③ Need even stronger foundations when software reaches out into our physical world.

How can we provide people with cyber-physical systems they can bet their lives on?  
— Jeannette Wing

## Cyber-physical Systems

CPS combine cyber capabilities with physical capabilities to solve problems that neither part could solve alone.

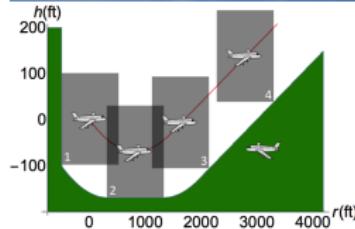
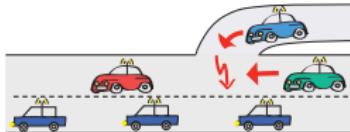
# CPSs Promise Transformative Impact!

## Prospects: Safe & Efficient

Driver assistance  
Autonomous cars

Pilot decision support  
Autopilots / UAVs

Train protection  
Robots help people



Prerequisite: CPS need to be safe

How do we make sure CPS make the world a better place?

# Benefits of Logical Foundations for CPS V & V

## Proofs

- Safety** Formalize system properties: What is “Safe”? “Reach goal”?
- Models** Formalize system models, clarify behavior
- Assumptions** Make assumptions explicit rather than silently
- Constraints** Reveal invariants, switching conditions, operating conditions
- Design** Invariants guide safe controller design
- Constructive** Construct system models along with their proofs

## Byproducts

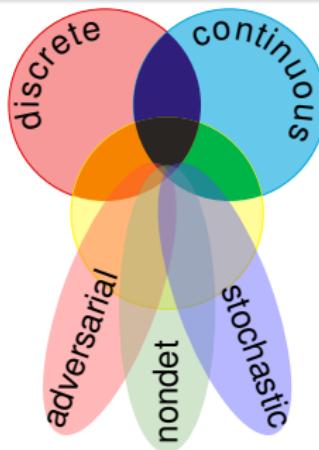
- Analysis** Determine design trade-offs & feasibility early
- Synthesis** Turn high-level models into code & correctness monitors
- Certificate** Proofs as artifacts for certification

## Tools

**KeYmaera X** aXiomatic Tactical Theorem Prover for CPS

### CPS Dynamics

CPS are characterized by multiple facets of dynamical systems.



### CPS Compositions

CPS combine multiple simple dynamical effects.

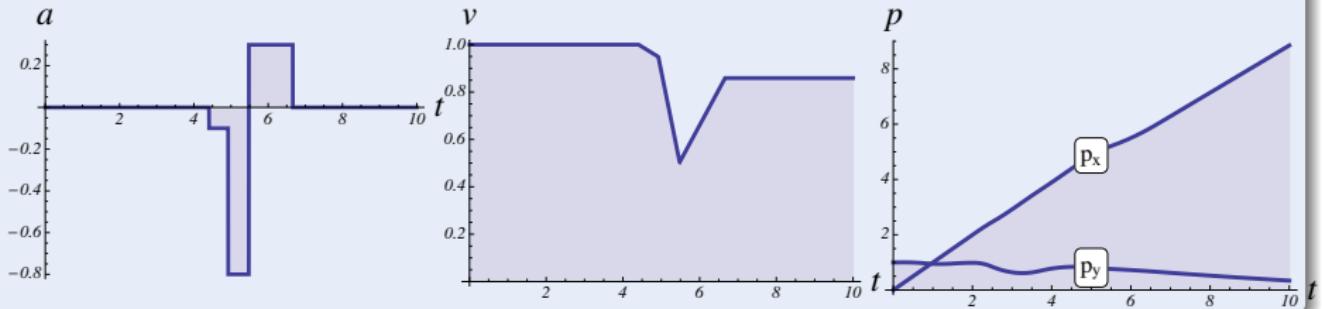
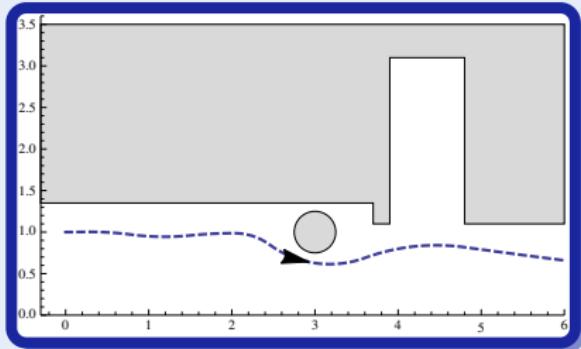
### Tame Parts

Exploiting compositionality tames CPS complexity.

## Challenge (Hybrid Systems)

Fixed rule describing state evolution with both

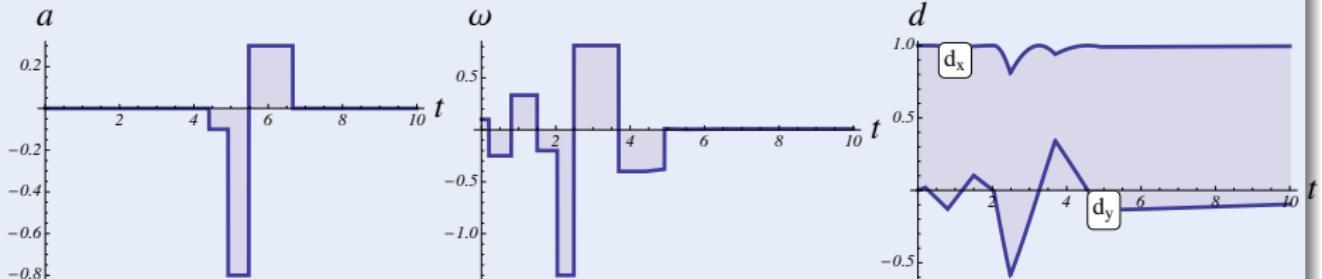
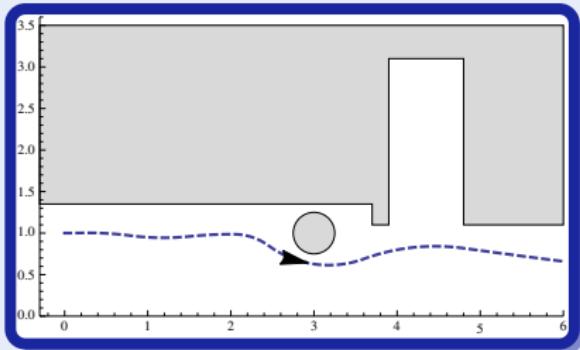
- Discrete dynamics (control decisions)
- Continuous dynamics (differential equations)



## Challenge (Hybrid Systems)

Fixed rule describing state evolution with both

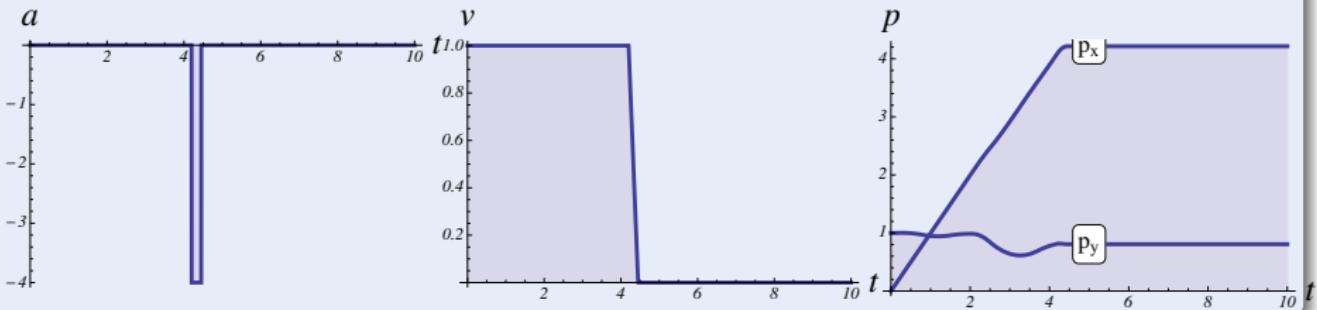
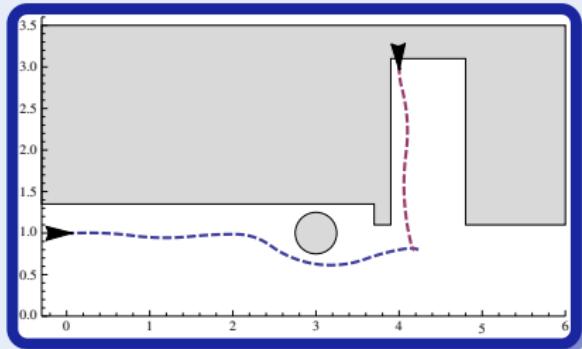
- Discrete dynamics (control decisions)
- Continuous dynamics (differential equations)



## Challenge (Hybrid Systems)

Fixed rule describing state evolution with both

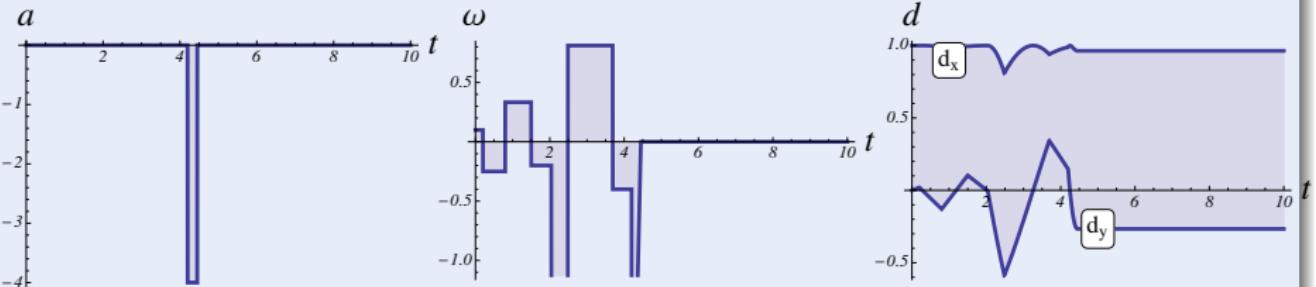
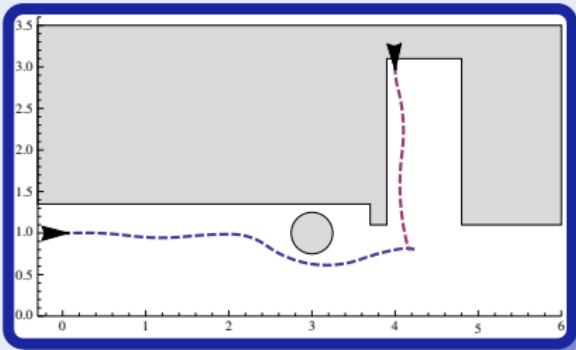
- Discrete dynamics (control decisions)
- Continuous dynamics (differential equations)



## Challenge (Hybrid Systems)

Fixed rule describing state evolution with both

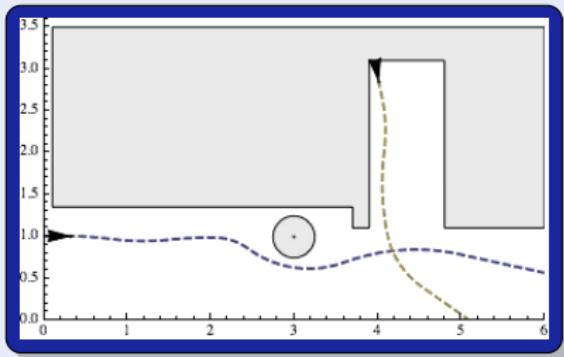
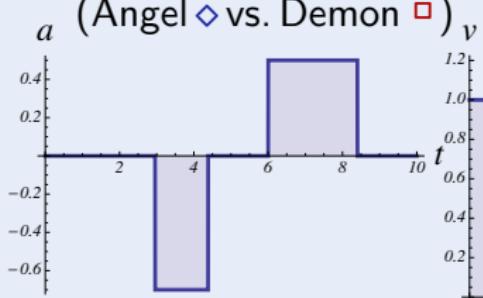
- Discrete dynamics (control decisions)
- Continuous dynamics (differential equations)



## Challenge (Hybrid Games)

Game rules describing play evolution with

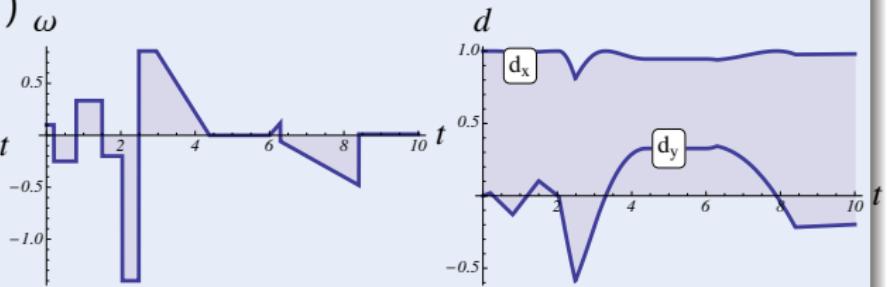
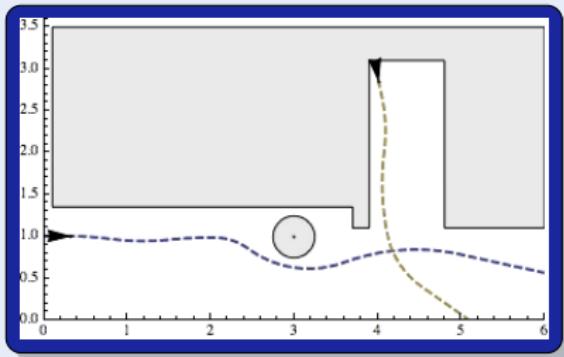
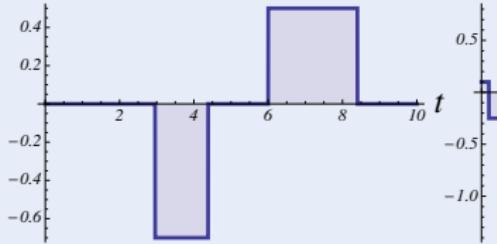
- Discrete dynamics (control decisions)
- Continuous dynamics (differential equations)
- Adversarial dynamics (Angel  $\diamond$  vs. Demon  $\square$ )



## Challenge (Hybrid Games)

Game rules describing play evolution with

- Discrete dynamics (control decisions)
  - Continuous dynamics (differential equations)
  - Adversarial dynamics (Angel  $\diamond$  vs. Demon  $\square$ )
- $a$  ( $\omega$ )



## 1 CPS are Multi-Dynamical Systems

- Hybrid Systems
- Hybrid Games

## 2 Dynamic Logic for Multi-Dynamical Systems

- Syntax
- Semantics

## 3 Proofs for CPS

## 4 Theory of CPS

- Soundness and Completeness
- Differential Invariants
- Examples
- Differential Radical Invariants

## 5 Applications

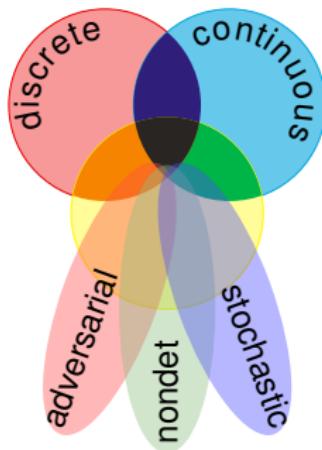
## 6 Summary

hybrid systems

$$\text{HS} = \text{discrete} + \text{ODE}$$

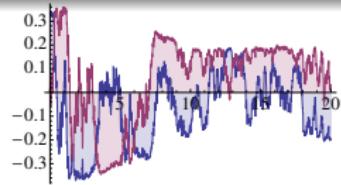
hybrid games

$$\text{HG} = \text{HS} + \text{adversary}$$



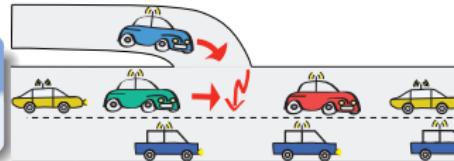
stochastic hybrid sys.

$$\text{SHS} = \text{HS} + \text{stochastics}$$



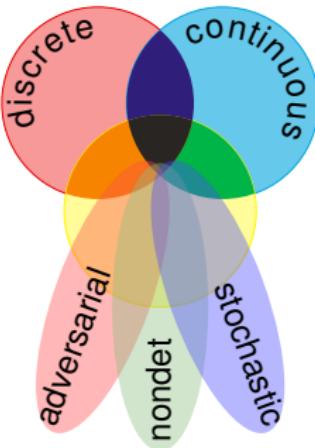
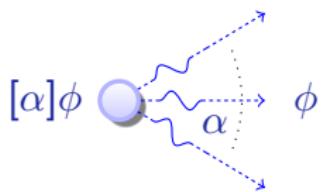
distributed hybrid sys.

$$\text{DHS} = \text{HS} + \text{distributed}$$



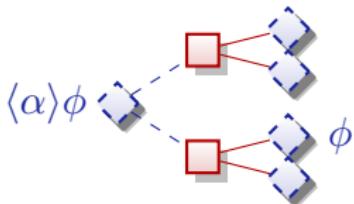
differential dynamic logic

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



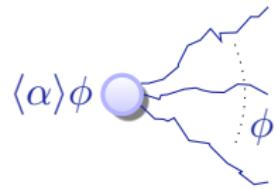
differential game logic

$$dG\mathcal{L} = GL + HG$$



stochastic differential DL

$$Sd\mathcal{L} = DL + SHP$$



quantified differential DL

$$Qd\mathcal{L} = FOL + DL + QHP$$

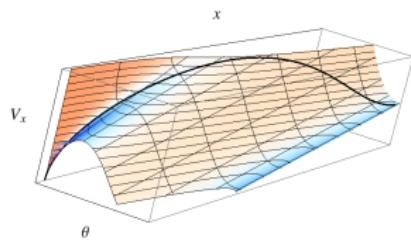
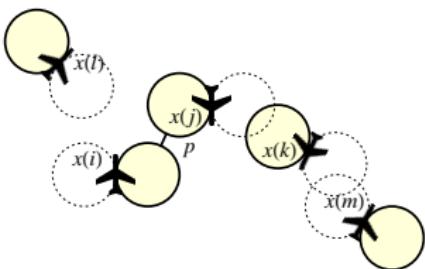
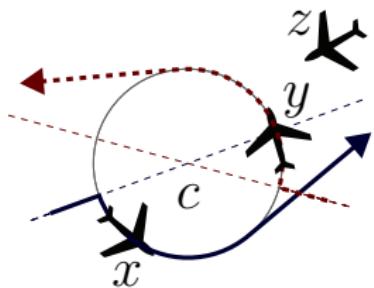
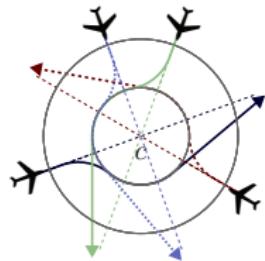
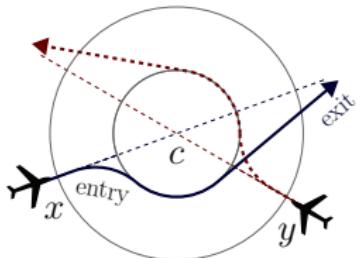
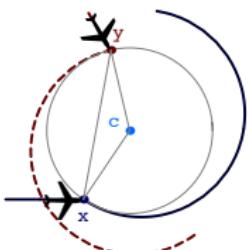
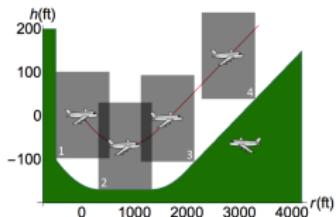
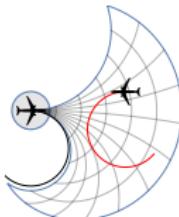
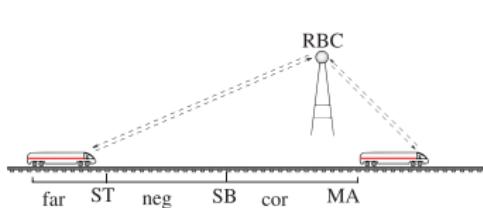
## Logical foundations of cyber-physical systems

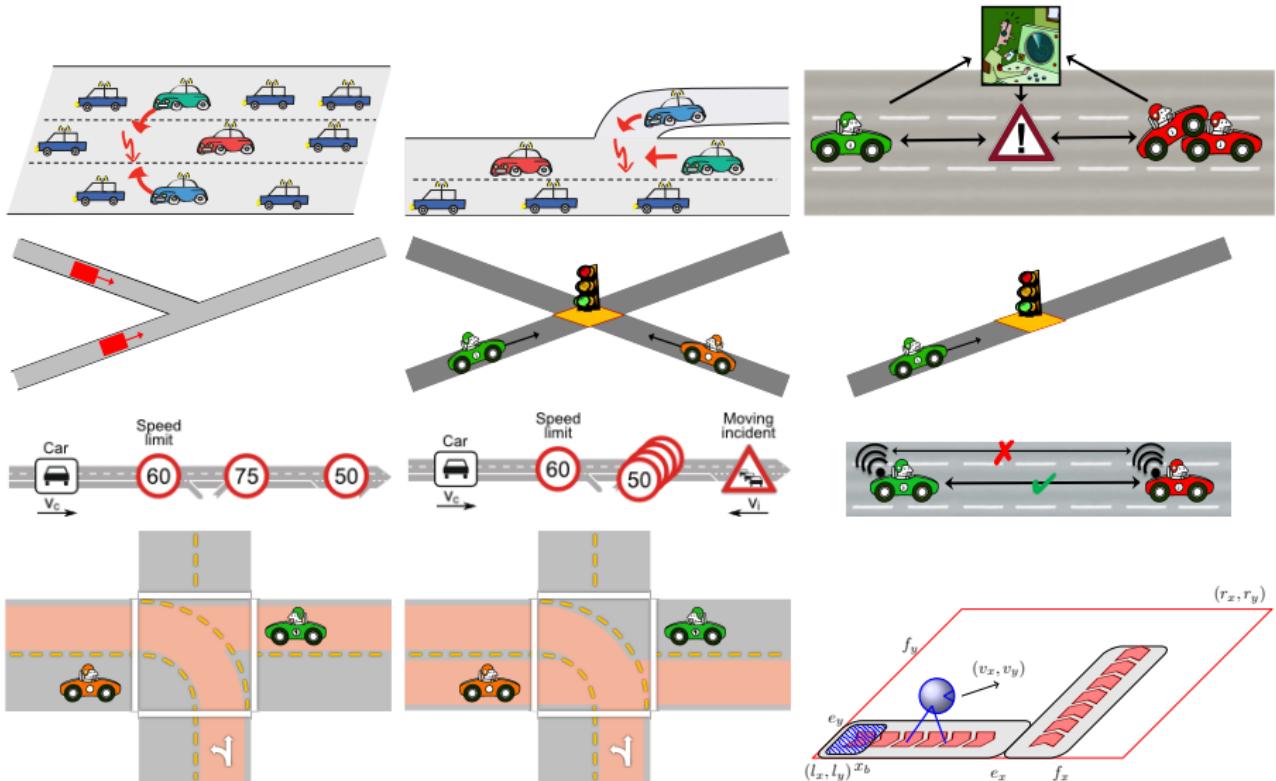
LICS'12

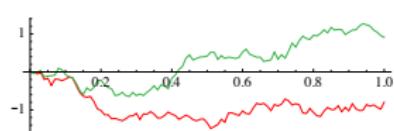
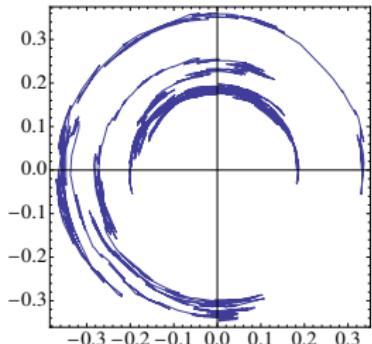
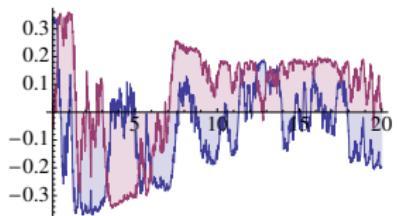
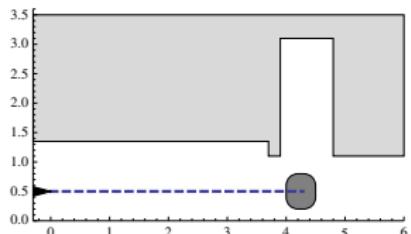
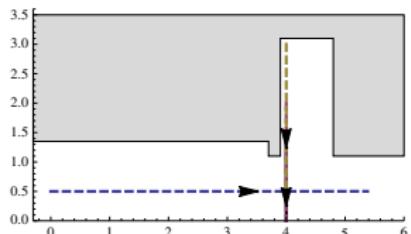
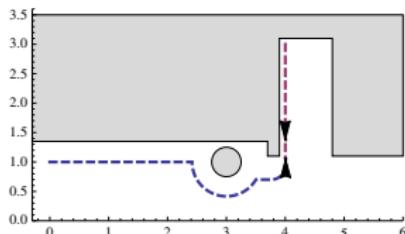
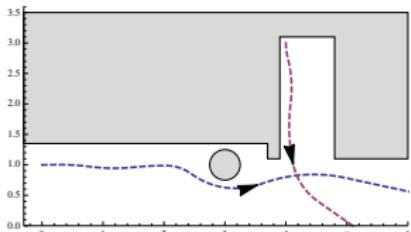
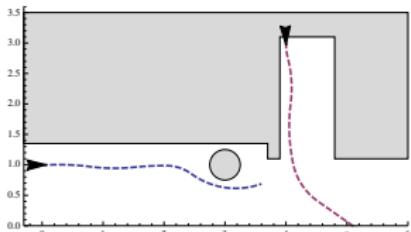
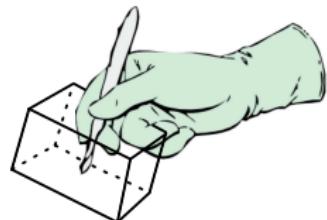
- ① Multi-dynamical systems
- ② Tame complexity by combinations of simple dynamics
- ③ Compositional programming language for CPS
- ④ Compositional logics and proof calculi
- ⑤ Differentialize logic & Logicalize differentials
- ⑥ Proofs for differential equations
- ⑦ Solid foundation for theory
- ⑧ Many useful applications
- ⑨ Education: Foundations of CPS course

## Basis for other technology

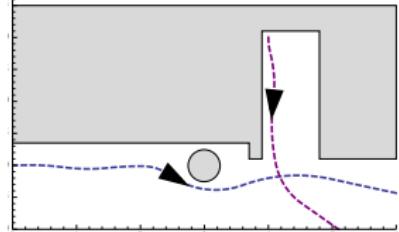
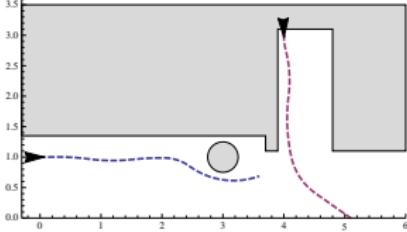
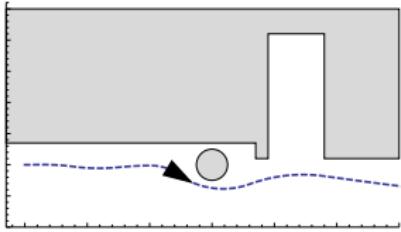
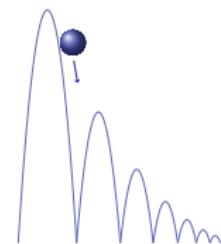
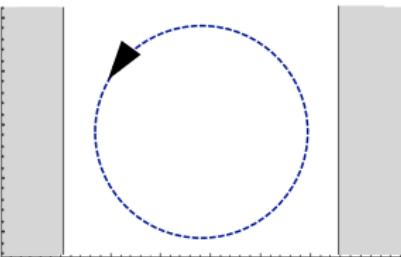
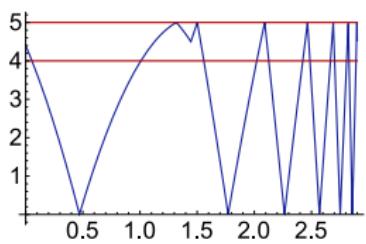
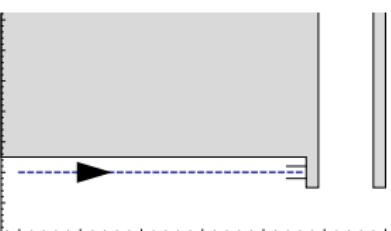
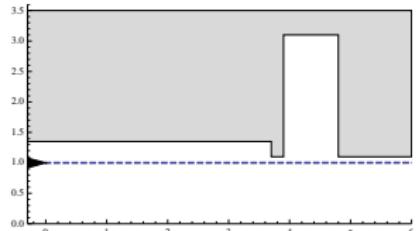
- ① ModelPlex transfers CPS model  $\leadsto$  implementation safety FMSD'16
- ② Proof-aware refactoring to co-evolve model + proof FM'14
- ③ Control envelope design ACC'12





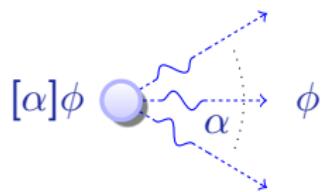


HSCC'13, RSS'13, CADE'12



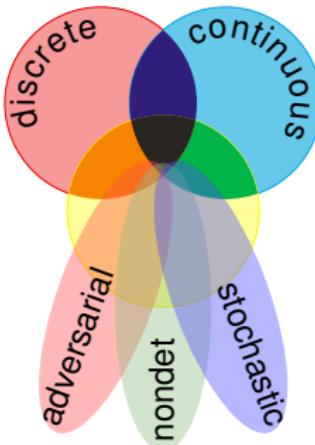
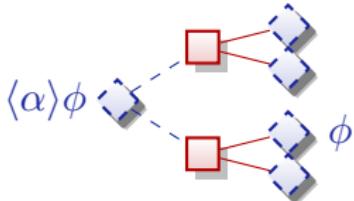
differential dynamic logic

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



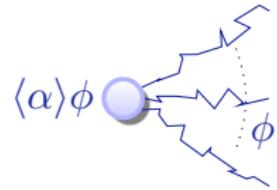
differential game logic

$$dGL = GL + HG$$



stochastic differential DL

$$Sd\mathcal{L} = DL + SHP$$

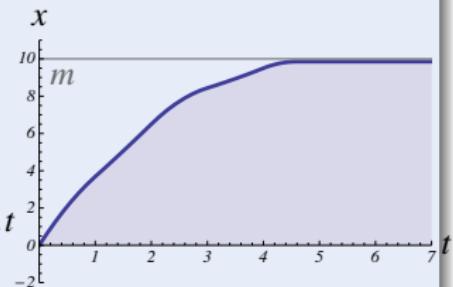
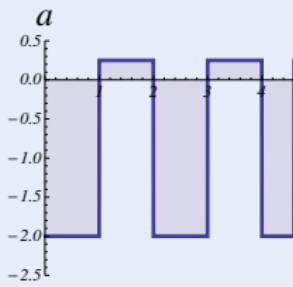
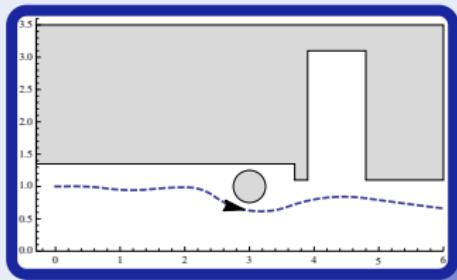
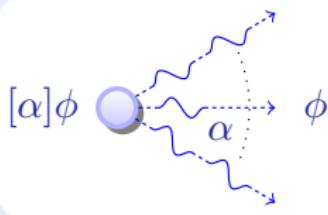


quantified differential DL

$$Qd\mathcal{L} = FOL + DL + QHP$$

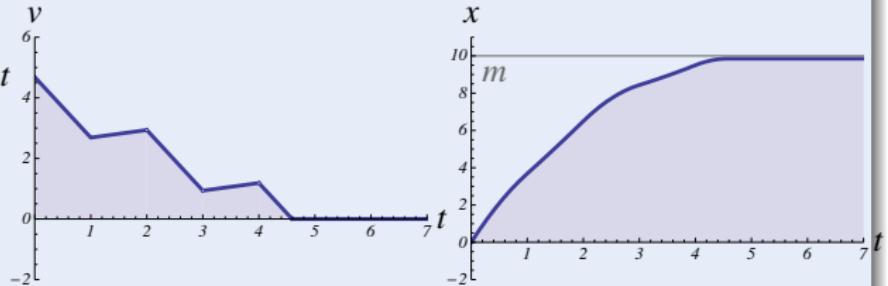
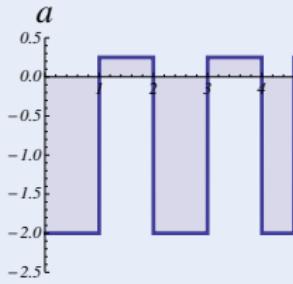
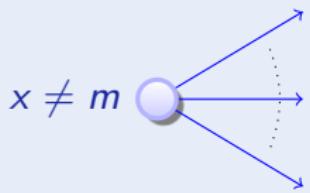
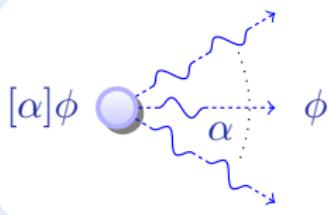
## Concept (Differential Dynamic Logic)

(JAR'08,LICS'12)



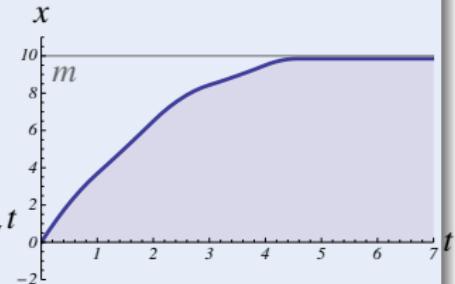
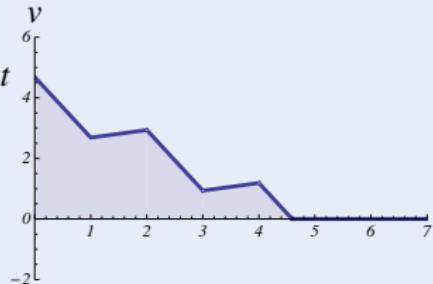
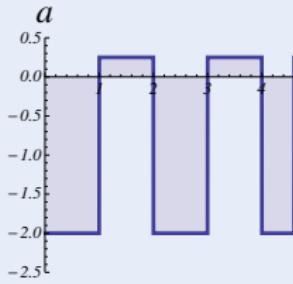
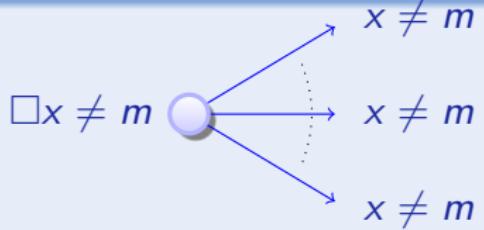
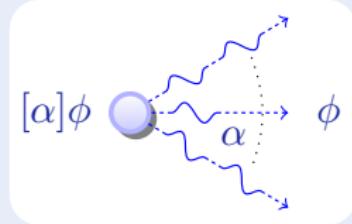
## Concept (Differential Dynamic Logic)

(JAR'08,LICS'12)

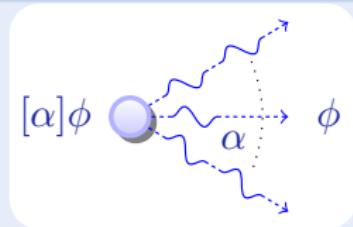


## Concept (Differential Dynamic Logic)

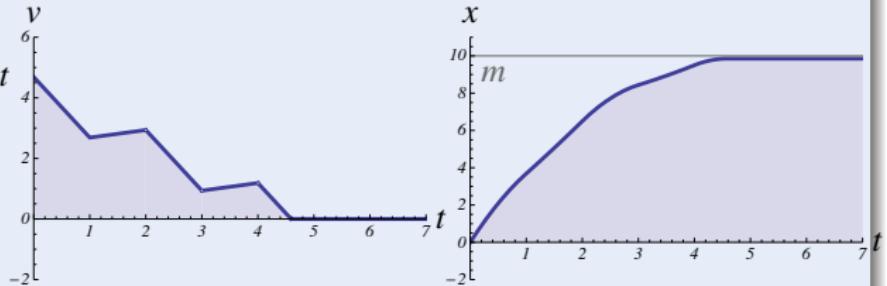
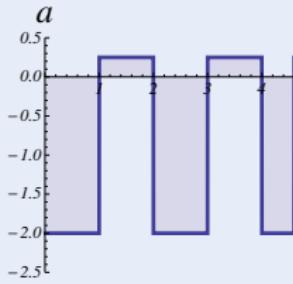
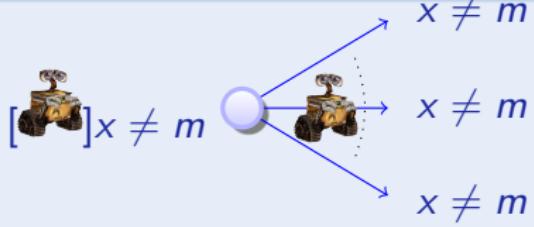
(JAR'08,LICS'12)



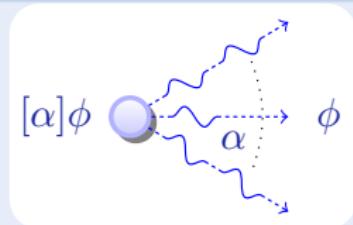
## Concept (Differential Dynamic Logic)



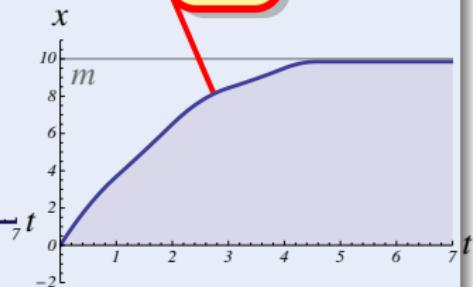
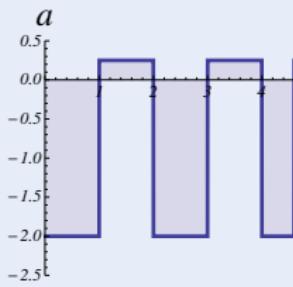
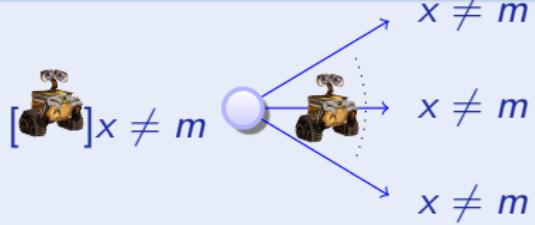
(JAR'08,LICS'12)



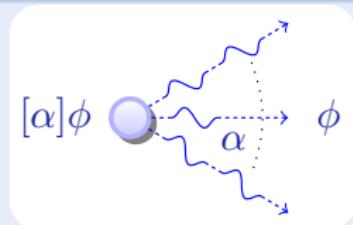
## Concept (Differential Dynamic Logic)



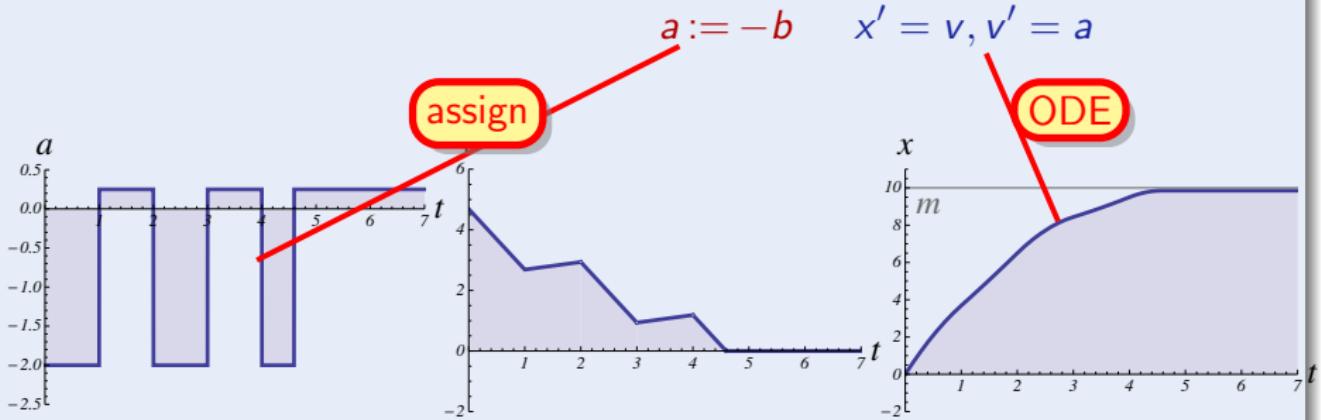
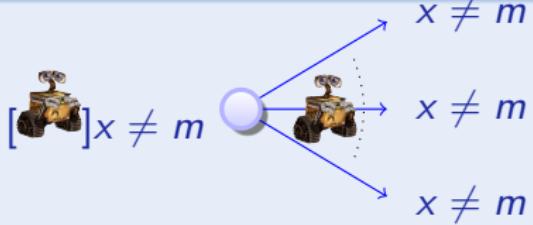
(JAR'08,LICS'12)



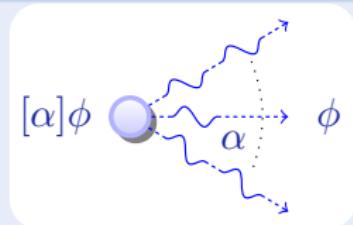
## Concept (Differential Dynamic Logic)



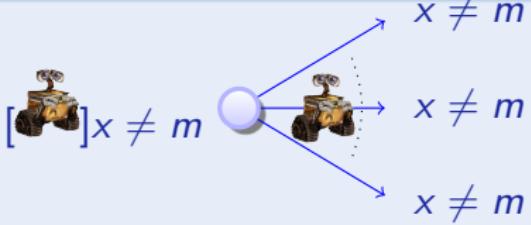
(JAR'08,LICS'12)



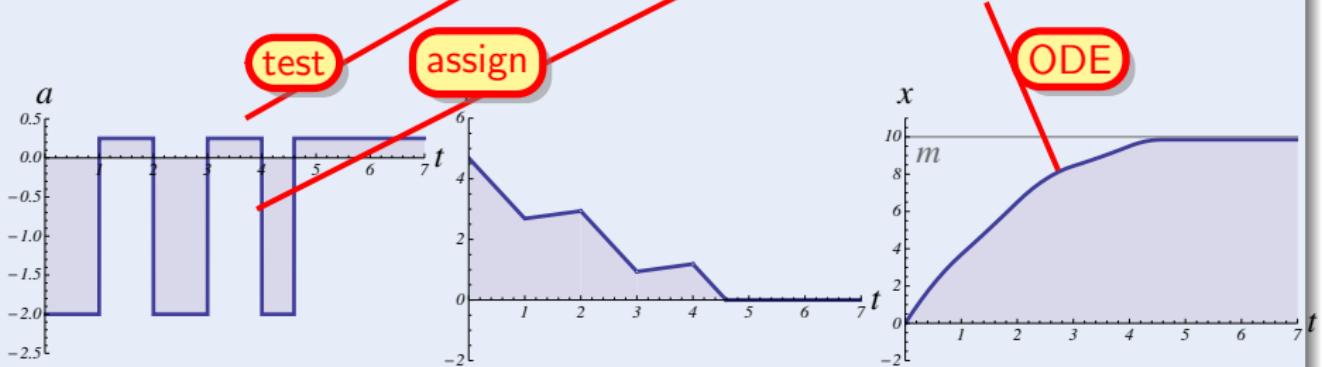
## Concept (Differential Dynamic Logic)



(JAR'08,LICS'12)

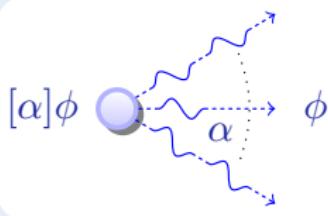


$(\text{if}(\text{SB}(x, m)) a := -b) \quad x' = v, v' = a$



## Concept (Differential Dynamic Logic)

(JAR'08,LICS'12)

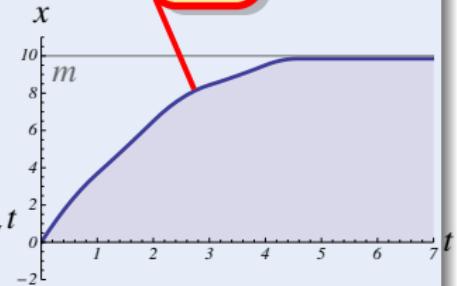
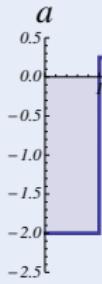


seq.  
compose

(if(SB( $x, m$ ))  $a := -b$ ) ;  $x' = v, v' = a$

test

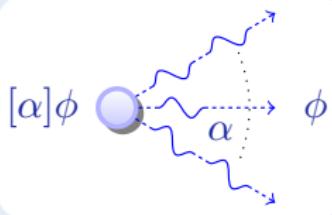
assign



ODE

## Concept (Differential Dynamic Logic)

(JAR'08,LICS'12)



seq.  
compose

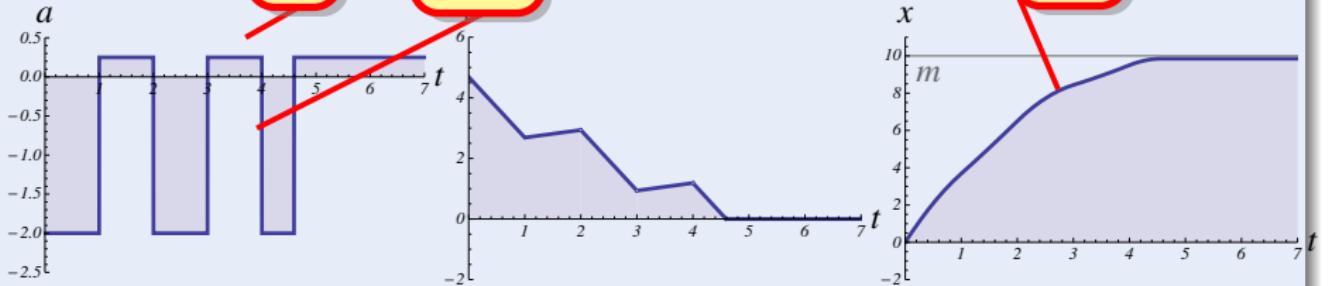
nondet.  
repeat

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

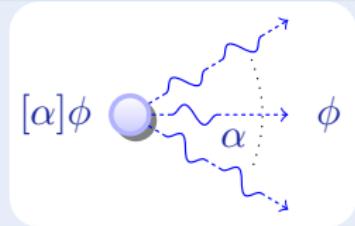
test

assign

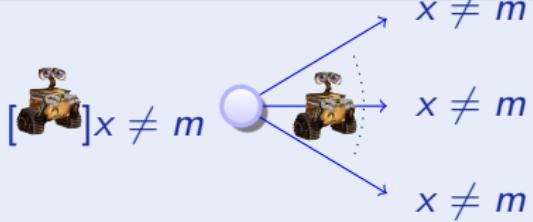
ODE



## Concept (Differential Dynamic Logic)



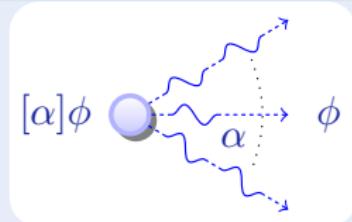
(JAR'08,LICS'12)



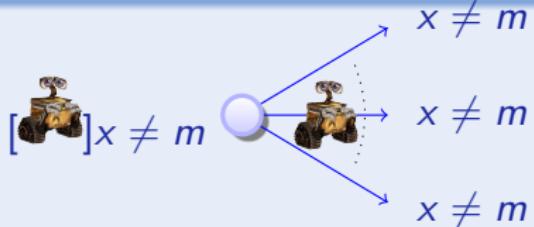
$$[((\text{if}(\text{SB}(x, m)) \ a := -b) ; \ x' = v, v' = a)^*]_{x \neq m} \text{ post}$$



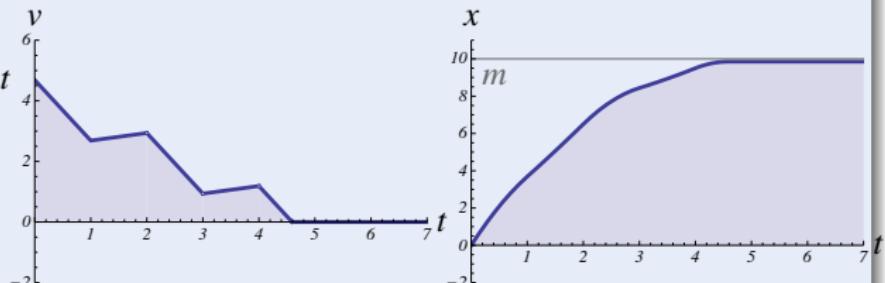
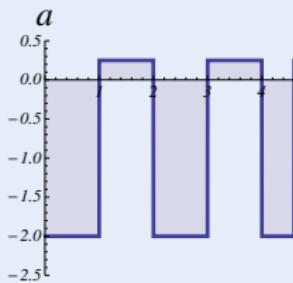
## Concept (Differential Dynamic Logic)



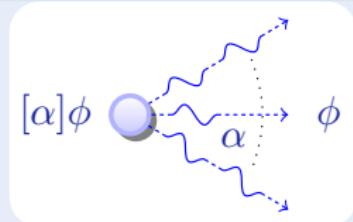
(JAR'08,LICS'12)



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

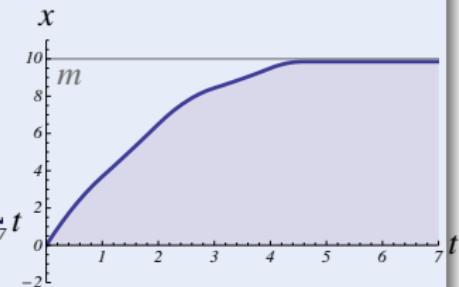
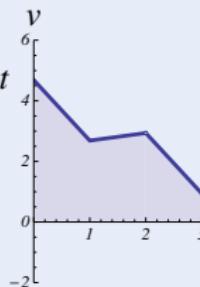
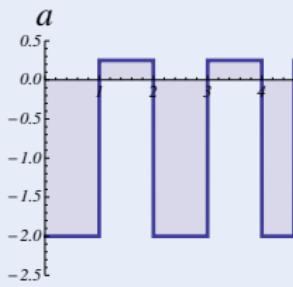
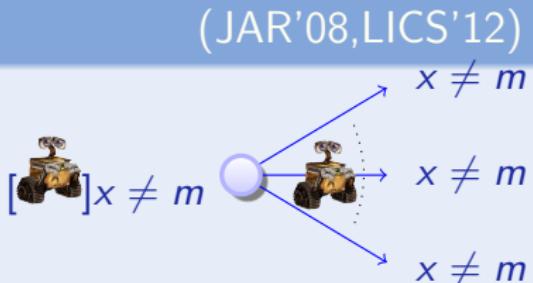


## Concept (Differential Dynamic Logic)

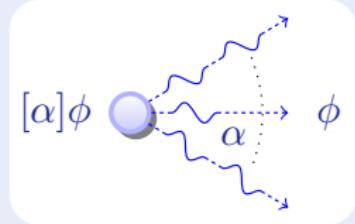


$$\underbrace{x \neq m \wedge b > 0}_{\text{init}} \rightarrow [((? \neg \text{SB}(x, m) \cup a := -b) ; x' = v, v' = a)^*] \underbrace{x \neq m}_{\text{post}}$$

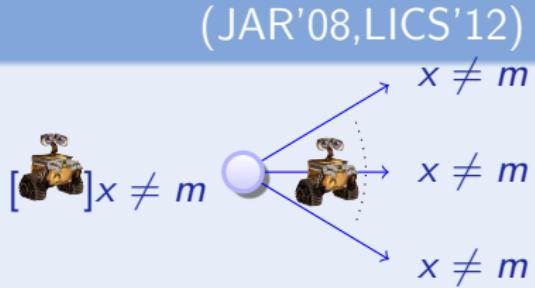
nondet.  
choice



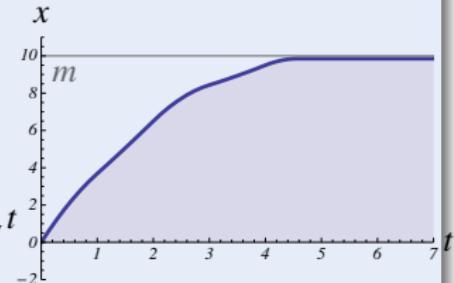
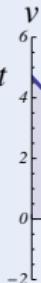
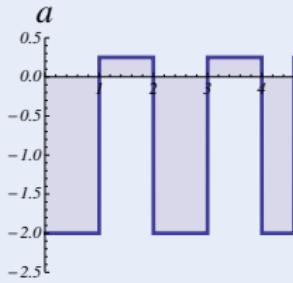
## Concept (Differential Dynamic Logic)



test  
nondet.  
choice



$$\underbrace{x \neq m \wedge b > 0}_{\text{init}} \rightarrow [((? \neg \text{SB}(x, m) \cup a := -b) ; x' = v, v' = a)^*] \underbrace{x \neq m}_{\text{post}}$$



Definition (Hybrid program  $\alpha$ )

$$x := f(x) \mid ?Q \mid \textcolor{red}{x' = f(x) \& Q} \mid \alpha \cup \beta \mid \alpha; \beta \mid \alpha^*$$

Definition (dL Formula  $P$ )

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

Discrete Assign

Test Condition

Differential Equation

Nondet. Choice

Seq. Compose

Nondet. Repeat

Definition (Hybrid program  $\alpha$ ) $x := f(x) \mid ?Q \mid x' = f(x) \& Q \mid \alpha \cup \beta \mid \alpha; \beta \mid \alpha^*$ Definition (dL Formula  $P$ ) $e \geq \tilde{e} \mid \neg P \mid P \wedge Q \mid \forall x P \mid \exists x P \mid [\alpha]P \mid \langle \alpha \rangle P$ 

All Reals

Some Reals

All Runs

Some Runs

# $\mathcal{R}$ Differential Dynamic Logic dL: Semantics

Definition (Hybrid program semantics)

$([\![\cdot]\!]: \text{HP} \rightarrow \wp(\mathcal{S} \times \mathcal{S}))$

$$[\![x := f(x)]\!] = \{(\omega, \nu) : \nu = \omega \text{ except } [\![x]\!] \nu = [\![f(x)]\!] \omega\}$$

$$[\![?Q]\!] = \{(\omega, \omega) : \omega \models Q\}$$

$$[\![x' = f(x)]\!] = \{(\varphi(0), \varphi(r)) : \varphi \models x' = f(x) \text{ for some duration } r\}$$

$$[\![\alpha \cup \beta]\!] = [\![\alpha]\!] \cup [\![\beta]\!]$$

$$[\![\alpha; \beta]\!] = [\![\alpha]\!] \circ [\![\beta]\!]$$

$$[\![\alpha^*]\!] = \bigcup_{n \in \mathbb{N}} [\![\alpha^n]\!]$$

Definition (dL semantics)

$([\![\cdot]\!]: \text{Fml} \rightarrow \wp(\mathcal{S}))$

$$[\![e \geq \tilde{e}]\!] = \{\omega : [\![e]\!] \omega \geq [\![\tilde{e}]\!] \omega\}$$

$$[\![\neg P]\!] = ([\![P]\!])^C$$

$$[\![P \wedge Q]\!] = [\![P]\!] \cap [\![Q]\!]$$

$$[\![\langle \alpha \rangle P]\!] = [\![\alpha]\!] \circ [\![P]\!] = \{\omega : \nu \models P \text{ for some } \nu : (\omega, \nu) \in [\![\alpha]\!]\}$$

$$[\![\exists \alpha P]\!] = [\![\neg \langle \alpha \rangle \neg P]\!] = \{\omega : \nu \models P \text{ for all } \nu : (\omega, \nu) \in [\![\alpha]\!]\}$$

$$[\![\exists x P]\!] = \{\omega : \omega_x^r \in [\![P]\!] \text{ for some } r \in \mathbb{R}\}$$

# R Outline

## 1 CPS are Multi-Dynamical Systems

- Hybrid Systems
- Hybrid Games

## 2 Dynamic Logic for Multi-Dynamical Systems

- Syntax
- Semantics

## 3 Proofs for CPS

## 4 Theory of CPS

- Soundness and Completeness
- Differential Invariants
- Examples
- Differential Radical Invariants

## 5 Applications

## 6 Summary

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

equations of truth

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

$$['] \quad [x' = f(x)]P \leftrightarrow \forall t \geq 0 [x := y(t)]P \quad (y'(t) = f(y))$$

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

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

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

$$\mathsf{K} \quad [\alpha](P \rightarrow Q) \rightarrow ([\alpha]P \rightarrow [\alpha]Q)$$

$$\mathsf{I} \quad [\alpha^*](P \rightarrow [\alpha]P) \rightarrow (P \rightarrow [\alpha^*]P)$$

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

LICS'12, CADE'15

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

equations of truth

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

$$['] \quad [x' = f(x)]P \leftrightarrow \forall t \geq 0 [x := y(t)]P \quad (y'(t) = f(y))$$

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

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

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

$$\mathsf{K} \quad [\alpha](P \rightarrow Q) \rightarrow ([\alpha]P \rightarrow [\alpha]Q)$$

$$\mathsf{I} \quad [\alpha^*](P \rightarrow [\alpha]P) \rightarrow (P \rightarrow [\alpha^*]P)$$

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

LICS'12, CADE'15

rules of truth

$$G \quad \frac{P}{[\alpha]P}$$

$$\forall \quad \frac{P}{\forall x P}$$

$$MP \quad \frac{P \rightarrow Q \quad P}{Q}$$

rules of truth

$$\text{G} \quad \frac{P}{[\alpha]P}$$

$$\forall \quad \frac{P}{\forall x P}$$

$$\text{MP} \quad \frac{P \rightarrow Q \quad P}{Q}$$

$$\vee \quad p \rightarrow [\alpha]p \quad (FV(p) \cap BV(\alpha) = \emptyset)$$

$$\text{CT} \quad \frac{f(x) = g(x)}{c(f(x)) = c(g(x))}$$

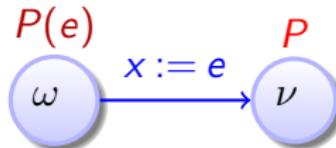
$$\text{CQ} \quad \frac{f(x) = g(x)}{p(f(x)) \leftrightarrow p(g(x))}$$

$$\text{CE} \quad \frac{P \leftrightarrow Q}{C(P) \leftrightarrow C(Q)}$$

LICS'12, CADE'15

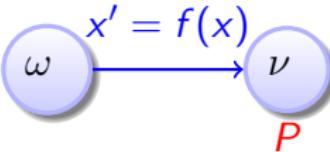
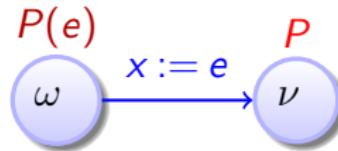
# $\mathcal{P}$ Proofs for Hybrid Systems

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



# $\mathcal{P}$ Proofs for Hybrid Systems

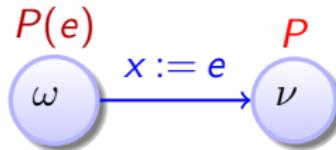
$$[x := e]P \leftrightarrow P(e)$$



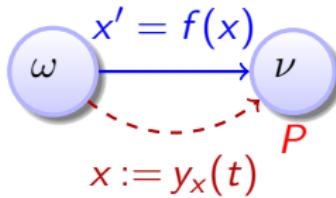
$$\begin{aligned} [x' = f(x)]P \\ \leftrightarrow \forall t \geq 0 [x := y_x(t)]P \end{aligned}$$

# $\mathcal{P}$ Proofs for Hybrid Systems

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



$$\begin{aligned} [x' = f(x)]P \\ \leftrightarrow \forall t \geq 0 [x := y_x(t)]P \end{aligned}$$

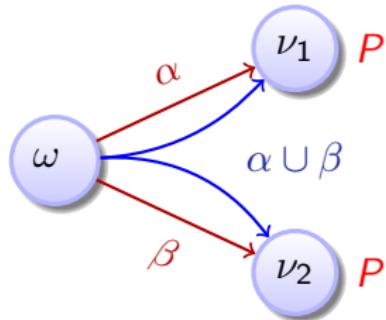


# Proofs for Hybrid Systems

compositional semantics  $\Rightarrow$  compositional rules!

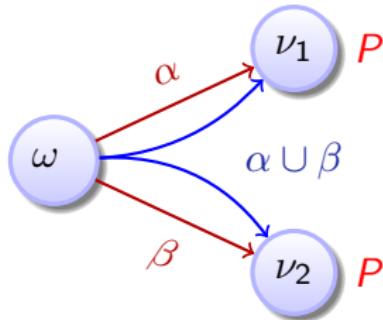
# $\mathcal{P}$ Proofs for Hybrid Systems

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

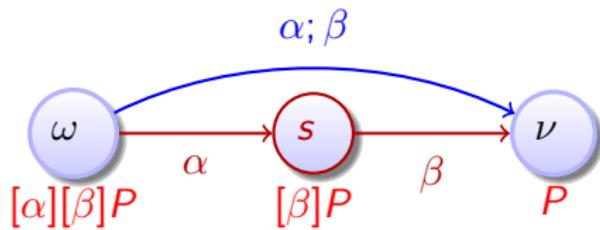


# $\mathcal{P}$ Proofs for Hybrid Systems

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

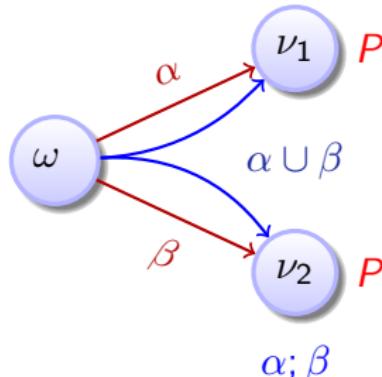


$$[\alpha; \beta]P \leftrightarrow [\alpha][\beta]P$$

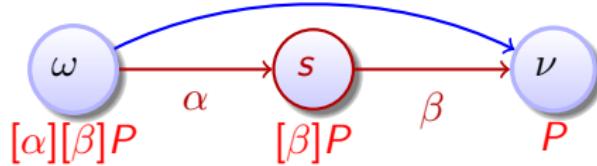


# $\mathcal{P}$ Proofs for Hybrid Systems

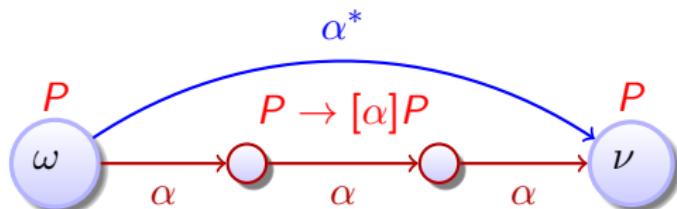
$$[\alpha \cup \beta]P \leftrightarrow [\alpha]P \wedge [\beta]P$$



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



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



## 1 CPS are Multi-Dynamical Systems

- Hybrid Systems
- Hybrid Games

## 2 Dynamic Logic for Multi-Dynamical Systems

- Syntax
- Semantics

## 3 Proofs for CPS

## 4 Theory of CPS

- Soundness and Completeness
- Differential Invariants
- Examples
- Differential Radical Invariants

## 5 Applications

## 6 Summary

Theorem (Sound & Complete) (J.Autom.Reas. 2008, LICS'12)

*dL calculus is a sound & complete axiomatization of hybrid systems relative to either differential equations **or** discrete dynamics.*

▶ Proof 25pp

Corollary (Complete Proof-theoretical Alignment & Bridging)  
proving continuous = proving hybrid = proving discrete

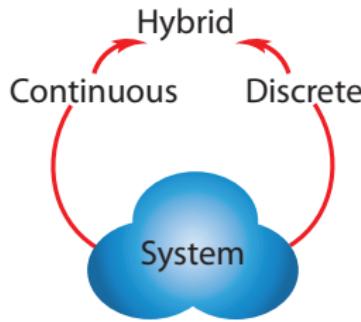
Theorem (Sound &amp; Complete)

(J.Autom.Reas. 2008, LICS'12)

*dL calculus is a sound & complete axiomatization of hybrid systems relative to either differential equations or discrete dynamics.*

▶ Proof 25pp

Corollary (Complete Proof-theoretical Alignment & Bridging)  
proving continuous = proving hybrid = proving discrete



JAutomReas'08, LICS'12

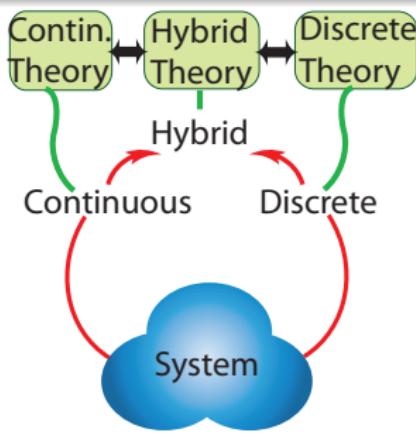
Theorem (Sound & Complete)

(J.Autom.Reas. 2008, LICS'12)

*dL calculus is a sound & complete axiomatization of hybrid systems relative to either differential equations or discrete dynamics.*

▶ Proof 25pp

Corollary (Complete Proof-theoretical Alignment & Bridging)  
proving continuous = proving hybrid = proving discrete



JAutomReas'08, LICS'12

# $\mathcal{R}$ Differential Equation Axioms & Differential Axioms

DW  $[x' = f(x) \& Q]Q$

DC  $([x' = f(x) \& Q]P \leftrightarrow [x' = f(x) \& Q \wedge r(x)]P)$   
 $\quad \leftarrow [x' = f(x) \& Q]r(x)$

DE  $[x' = f(x) \& Q]P \leftrightarrow [x' = f(x) \& Q][x' := f(x)]P$

DI  $[x' = f(x) \& Q]P \leftarrow (Q \rightarrow P \wedge [x' = f(x) \& Q](P)')$

DG  $[x' = f(x) \& Q]P \leftrightarrow \exists y [x' = f(x), y' = a(x)y + b(x) \& Q]P$

DS  $[x' = c() \& Q]P \leftrightarrow \forall t \geq 0 ((\forall 0 \leq s \leq t q(x + c()(s))) \rightarrow [x := x + c()t]P)$

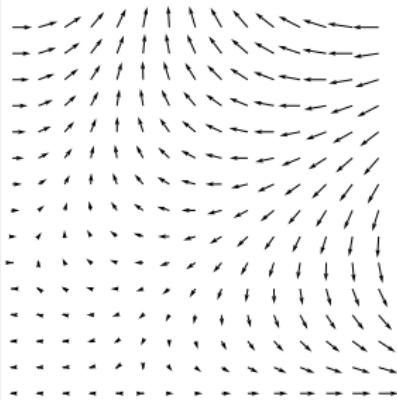
$[':=]$   $[x' := e]p(x') \leftrightarrow p(e)$

$$+' (e + k)' = (e)' + (k)'$$

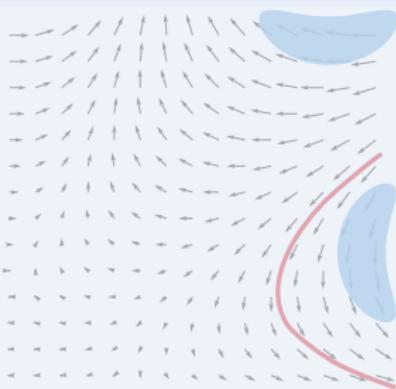
$$\cdot' (e \cdot k)' = (e)' \cdot k + e \cdot (k)'$$

$$\circ' [y := g(x)][y' := 1]((f(g(x))))' = (f(y))' \cdot (g(x))'$$

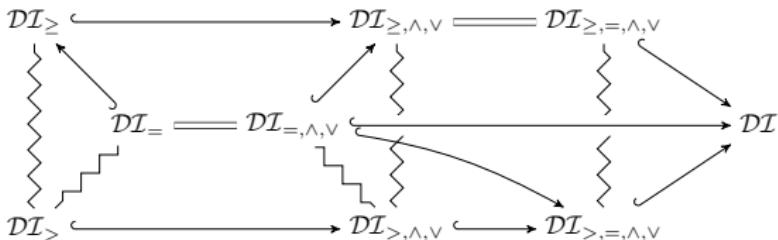
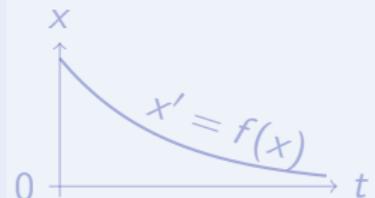
## Differential Invariant



## Differential Cut



## Differential Ghost

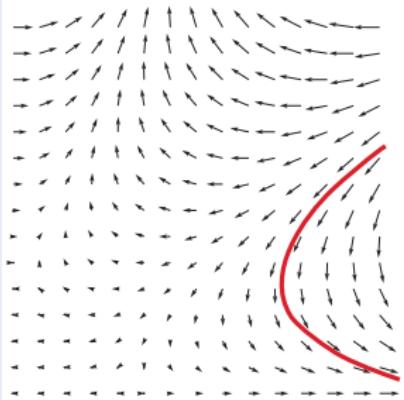


Logic  
Probability theory

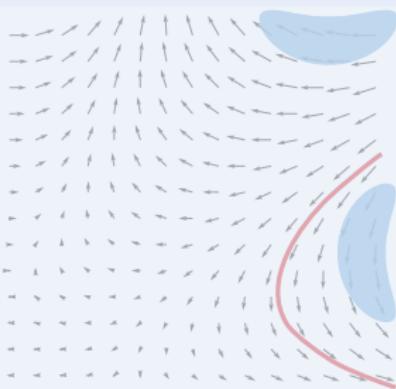
Math  
Characteristic PDE

JLogComput'10, CAV'08, FMSD'09, LMCS'12, LICS'12, CADE'15

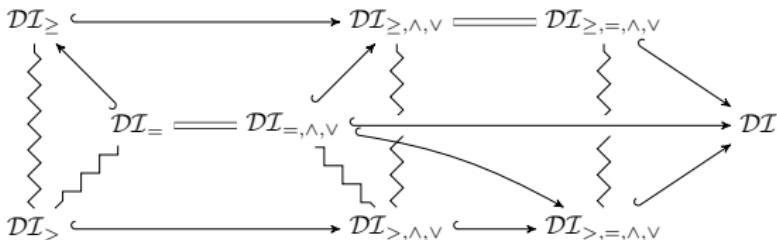
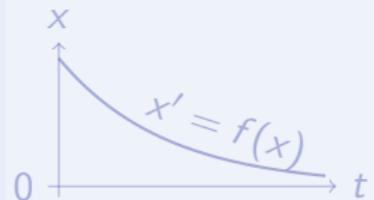
## Differential Invariant



## Differential Cut



## Differential Ghost

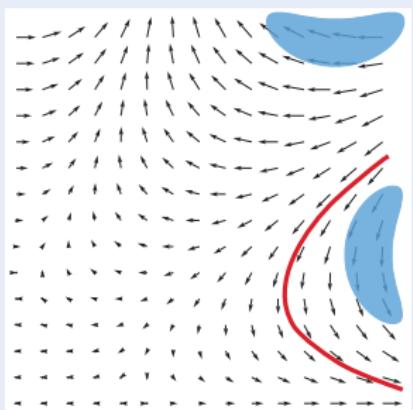


Logic  
Probability theory

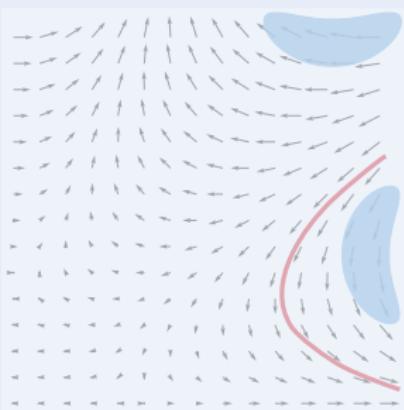
Math  
Characteristic PDE

JLogComput'10, CAV'08, FMSD'09, LMCS'12, LICS'12, CADE'15

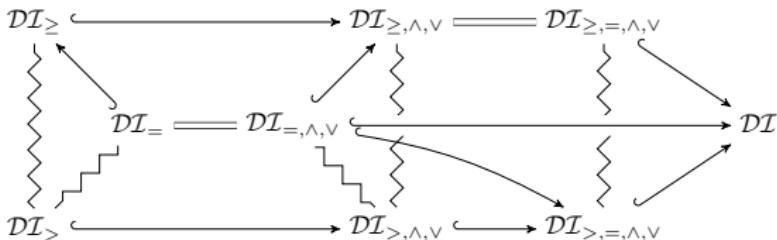
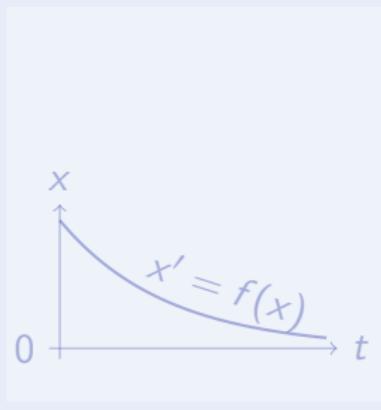
## Differential Invariant



## Differential Cut



## Differential Ghost

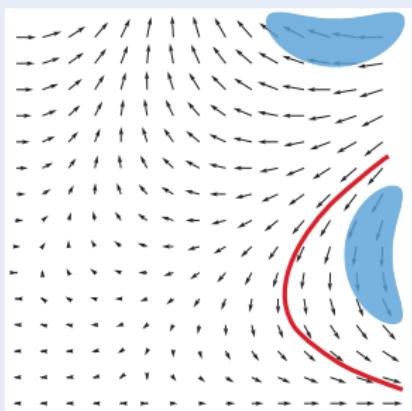


Logic  
Probability theory

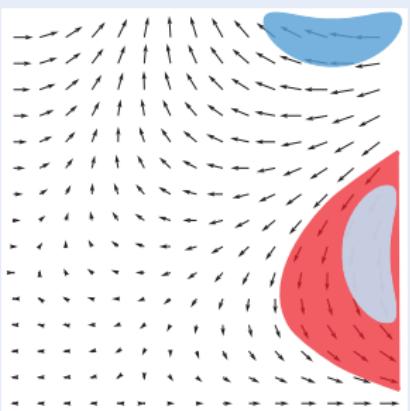
Math  
Characteristic PDE

JLogComput'10, CAV'08, FMSD'09, LMCS'12, LICS'12, CADE'15

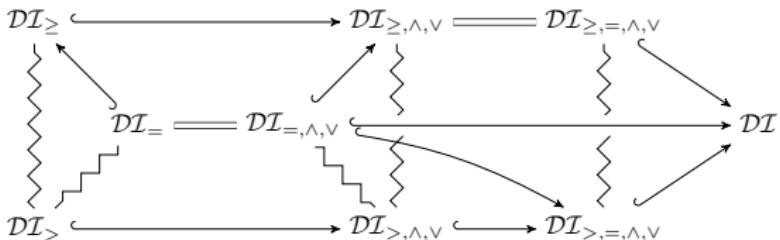
## Differential Invariant



## Differential Cut



## Differential Ghost

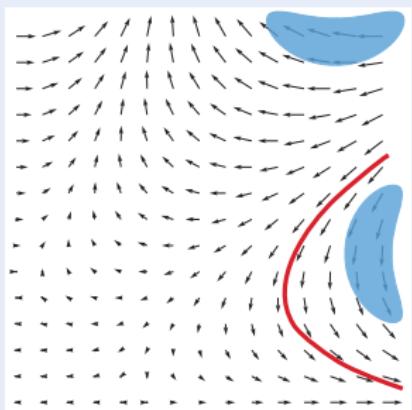


Logic  
Probability  
theory

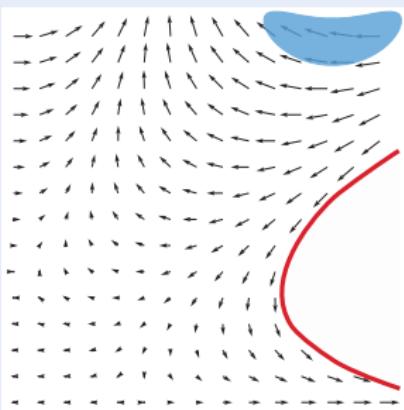
Math  
Characteristic PDE

JLogComput'10, CAV'08, FMSD'09, LMCS'12, LICS'12, CADE'15

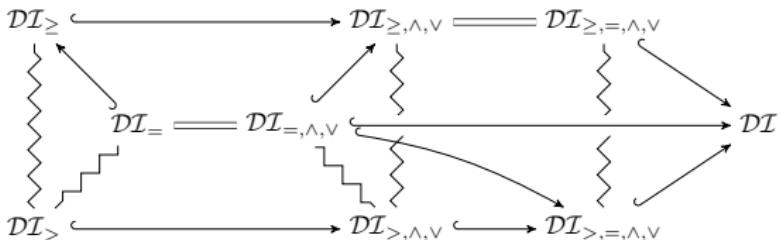
## Differential Invariant



## Differential Cut



## Differential Ghost

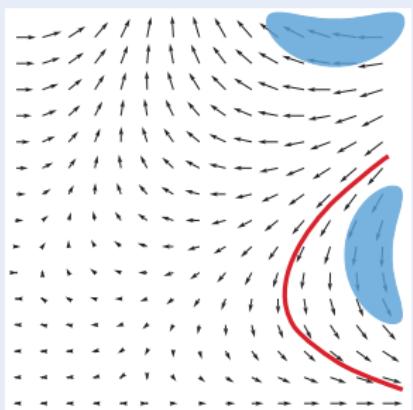


Logic  
Probability theory

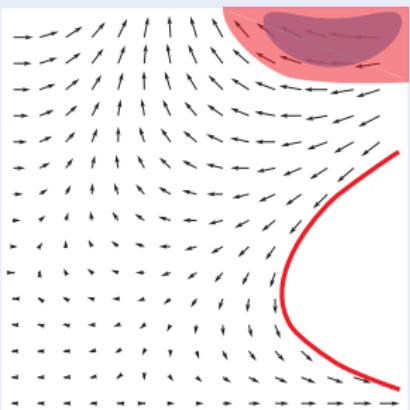
Math  
Characteristic PDE

JLogComput'10, CAV'08, FMSD'09, LMCS'12, LICS'12, CADE'15

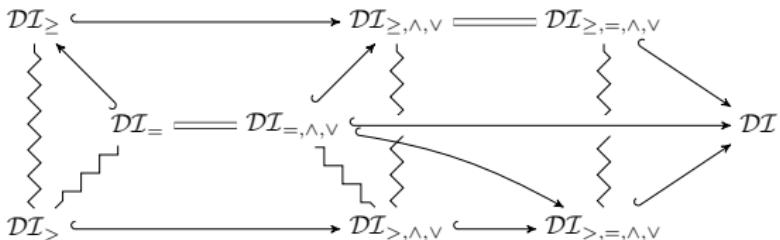
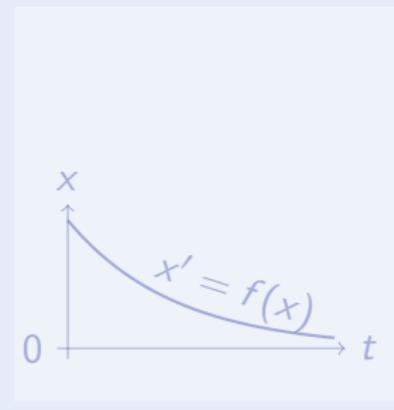
## Differential Invariant



## Differential Cut



## Differential Ghost

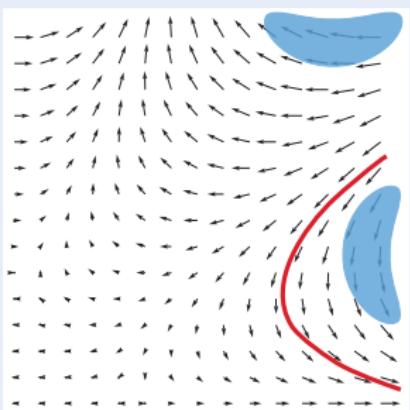


Logic  
Probability theory

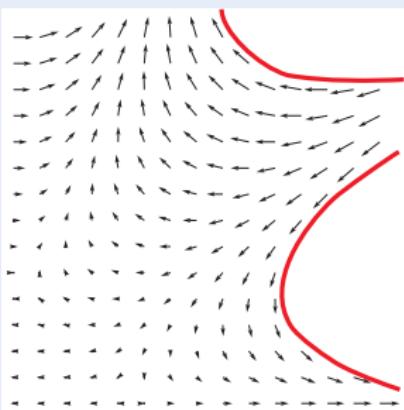
Math  
Characteristic PDE

JLogComput'10, CAV'08, FMSD'09, LMCS'12, LICS'12, CADE'15

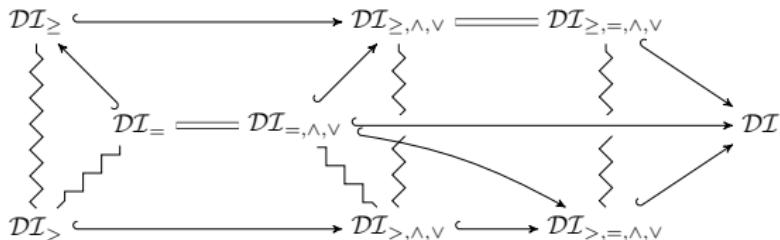
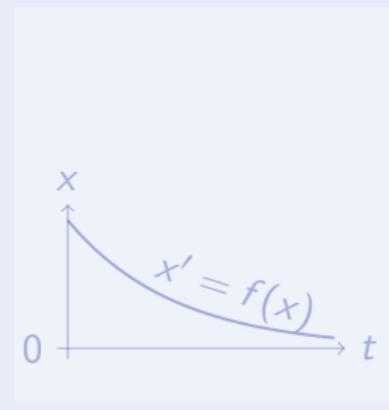
## Differential Invariant



## Differential Cut



## Differential Ghost

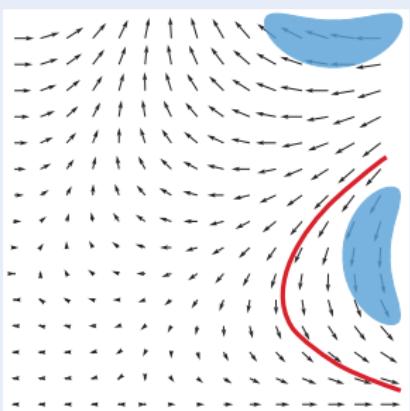


Logic  
Probability theory

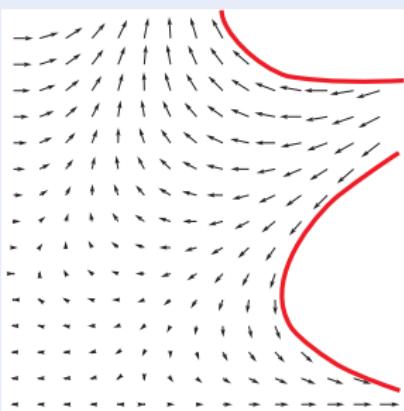
Math  
Characteristic PDE

JLogComput'10, CAV'08, FMSD'09, LMCS'12, LICS'12, CADE'15

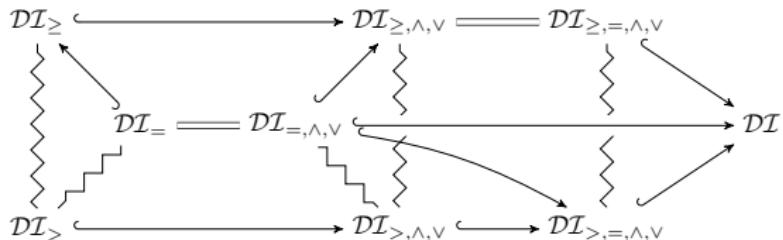
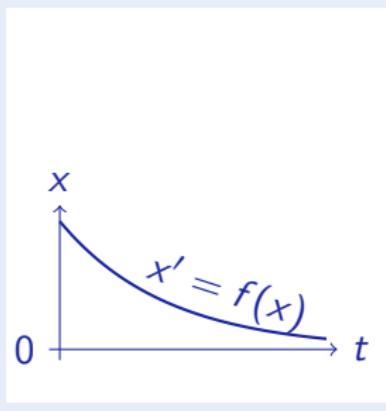
## Differential Invariant



## Differential Cut



## Differential Ghost

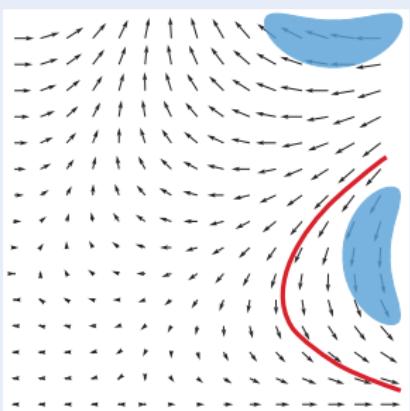


Logic  
Probability  
theory

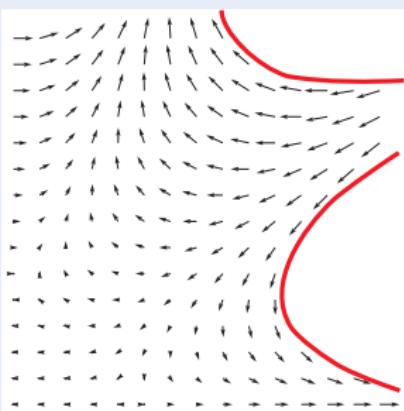
Math  
Characteristic PDE

JLogComput'10, CAV'08, FMSD'09, LMCS'12, LICS'12, CADE'15

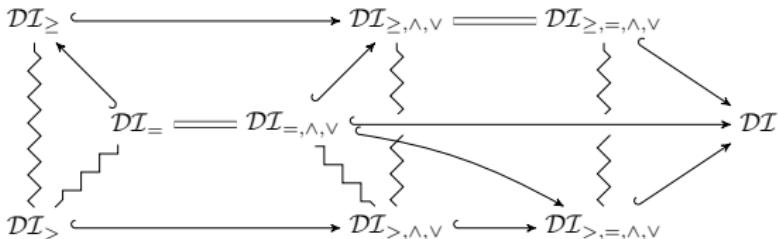
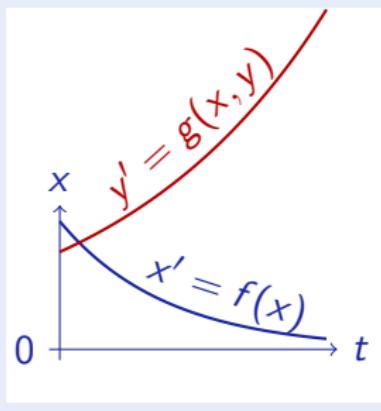
## Differential Invariant



## Differential Cut



## Differential Ghost

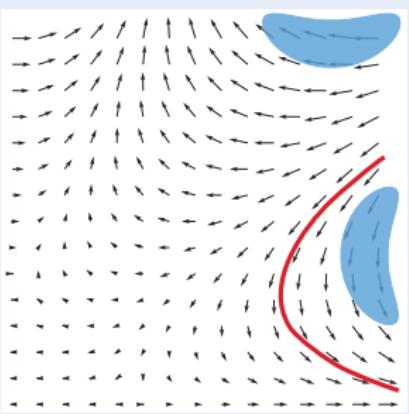


Logic  
Probability theory

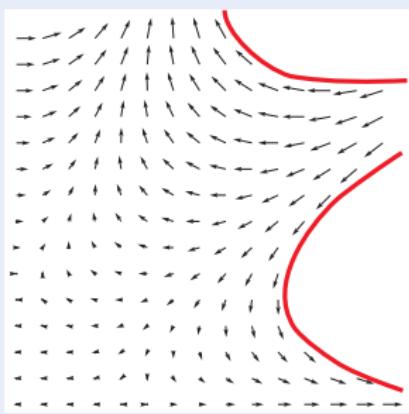
Math  
Characteristic PDE

JLogComput'10, CAV'08, FMSD'09, LMCS'12, LICS'12, CADE'15

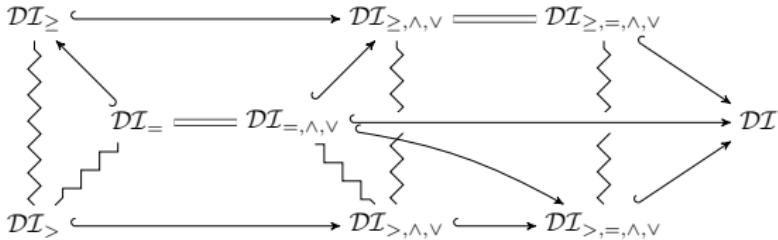
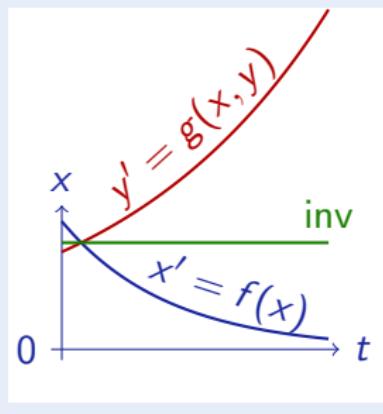
## Differential Invariant



## Differential Cut



## Differential Ghost



Logic  
Probability theory

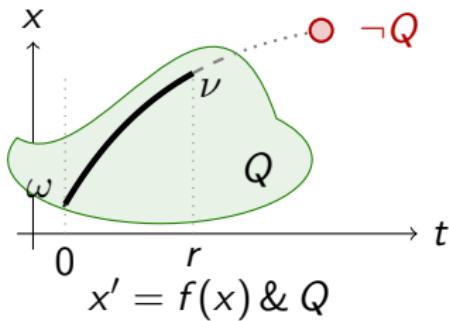
Math  
Characteristic PDE

JLogComput'10, CAV'08, FMSD'09, LMCS'12, LICS'12, CADE'15

## Axiom (Differential Weakening)

(CADE'15)

$$\text{DW } [x' = f(x) \& Q]Q$$



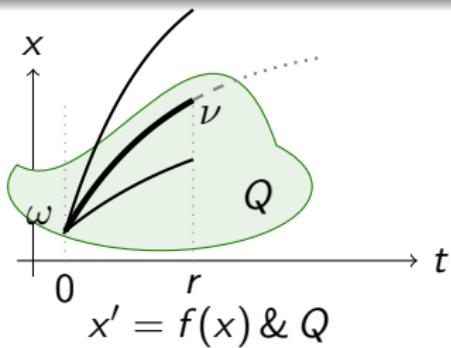
Differential equations cannot leave their evolution domains. Implies:

$$[x' = f(x) \& Q]P \leftrightarrow [x' = f(x) \& Q](Q \rightarrow P)$$

## Axiom (Differential Cut)

(CADE'15)

$$\text{DC} \quad ([x' = f(x) \& Q]P \leftrightarrow [x' = f(x) \& Q \wedge r(x)]P) \\ \qquad \qquad \qquad \leftarrow [x' = f(x) \& Q]r(x)$$



DC is a cut for differential equations.

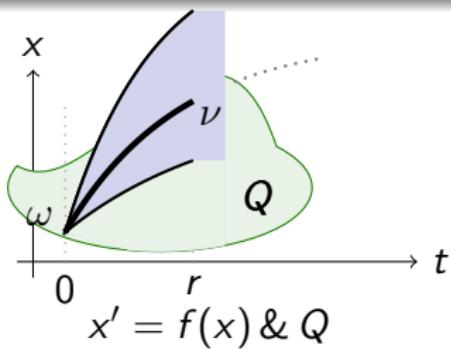
DC is a differential modal modus ponens K.

Can't leave  $r(x)$ , then might as well restrict state space to  $r(x)$ .

## Axiom (Differential Cut)

(CADE'15)

$$\text{DC} \quad ([x' = f(x) \& Q]P \leftrightarrow [x' = f(x) \& Q \wedge r(x)]P) \\ \qquad \qquad \qquad \leftarrow [x' = f(x) \& Q]r(x)$$



DC is a cut for differential equations.

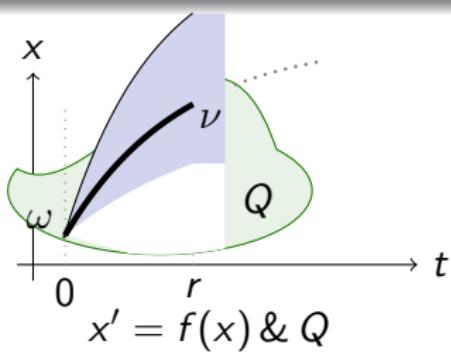
DC is a differential modal modus ponens K.

Can't leave  $r(x)$ , then might as well restrict state space to  $r(x)$ .

## Axiom (Differential Cut)

(CADE'15)

$$\text{DC} \quad ([x' = f(x) \& Q]P \leftrightarrow [x' = f(x) \& Q \wedge r(x)]P) \\ \qquad \qquad \qquad \leftarrow [x' = f(x) \& Q]r(x)$$



DC is a cut for differential equations.

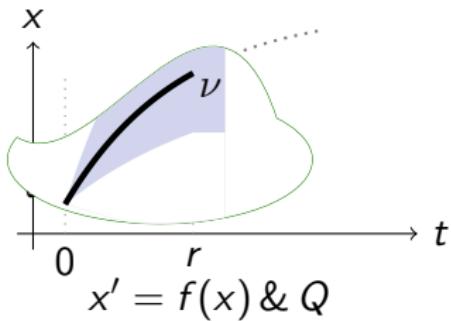
DC is a differential modal modus ponens K.

Can't leave  $r(x)$ , then might as well restrict state space to  $r(x)$ .

## Axiom (Differential Cut)

(CADE'15)

$$\text{DC} \quad ([x' = f(x) \& Q]P \leftrightarrow [x' = f(x) \& Q \wedge r(x)]P) \\ \qquad \qquad \qquad \leftarrow [x' = f(x) \& Q]r(x)$$



DC is a cut for differential equations.

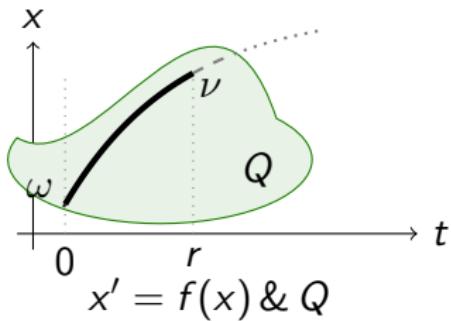
DC is a differential modal modus ponens K.

Can't leave  $r(x)$ , then might as well restrict state space to  $r(x)$ .

Axiom (Differential Cut)

(CADE'15)

$$\text{DC} \quad ([x' = f(x) \& Q]P \leftrightarrow [x' = f(x) \& Q \wedge r(x)]P) \\ \leftarrow [x' = f(x) \& Q]r(x)$$



DC is a cut for differential equations.

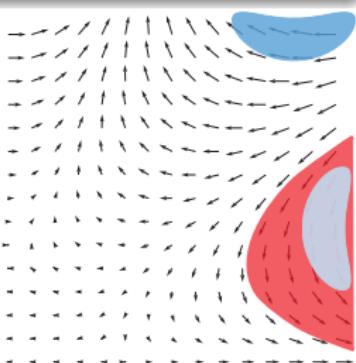
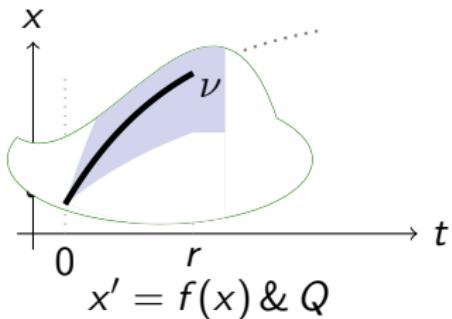
DC is a differential modal modus ponens K.

Can't leave  $r(x)$ , then might as well restrict state space to  $r(x)$ .

Axiom (Differential Cut)

(CADE'15)

$$\text{DC} \quad ([x' = f(x) \& Q]P \leftrightarrow [x' = f(x) \& Q \wedge r(x)]P) \\ \leftarrow [x' = f(x) \& Q]r(x)$$



DC is a cut for differential equations.

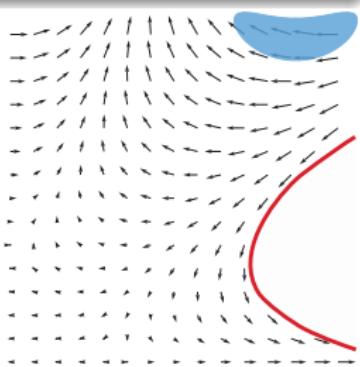
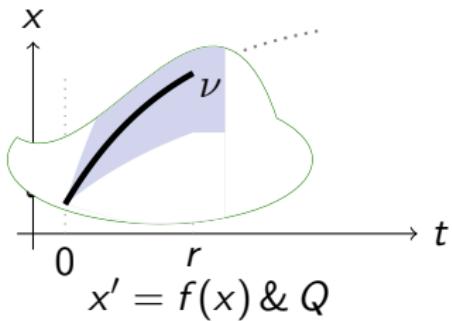
DC is a differential modal modus ponens K.

Can't leave  $r(x)$ , then might as well restrict state space to  $r(x)$ .

Axiom (Differential Cut)

(CADE'15)

$$\text{DC} \quad ([x' = f(x) \& Q]P \leftrightarrow [x' = f(x) \& Q \wedge r(x)]P) \\ \leftarrow [x' = f(x) \& Q]r(x)$$



DC is a cut for differential equations.

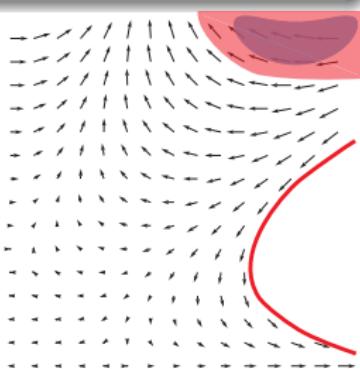
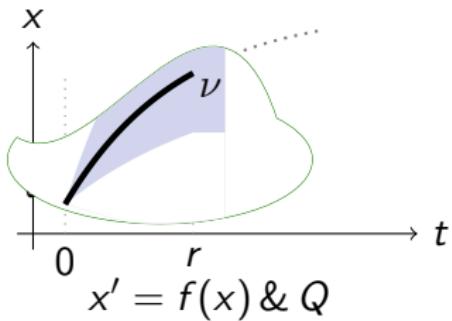
DC is a differential modal modus ponens K.

Can't leave  $r(x)$ , then might as well restrict state space to  $r(x)$ .

Axiom (Differential Cut)

(CADE'15)

$$\text{DC} \quad ([x' = f(x) \& Q]P \leftrightarrow [x' = f(x) \& Q \wedge r(x)]P) \\ \leftarrow [x' = f(x) \& Q]r(x)$$



DC is a cut for differential equations.

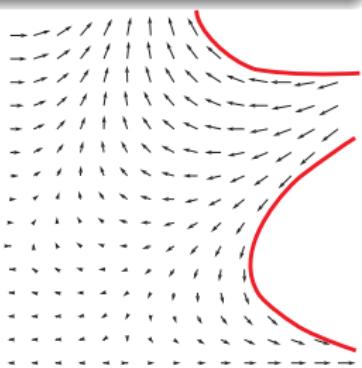
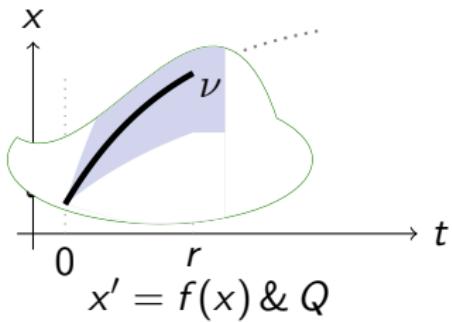
DC is a differential modal modus ponens K.

Can't leave  $r(x)$ , then might as well restrict state space to  $r(x)$ .

Axiom (Differential Cut)

(CADE'15)

$$\text{DC} \quad ([x' = f(x) \& Q]P \leftrightarrow [x' = f(x) \& Q \wedge r(x)]P) \\ \leftarrow [x' = f(x) \& Q]r(x)$$



DC is a cut for differential equations.

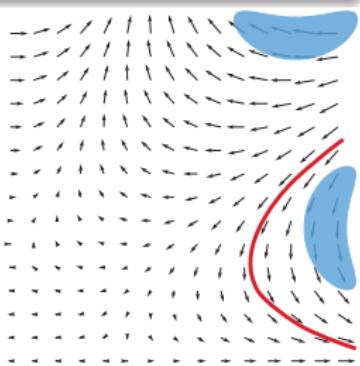
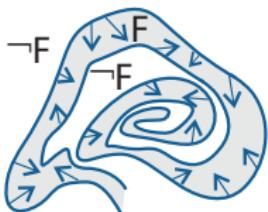
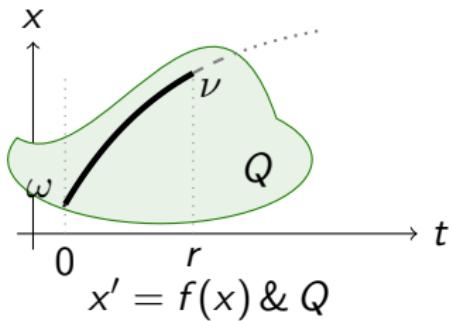
DC is a differential modal modus ponens K.

Can't leave  $r(x)$ , then might as well restrict state space to  $r(x)$ .

Axiom (Differential Invariant)

(CADE'15)

$$\text{DI } [x' = f(x) \& Q]P \leftarrow (Q \rightarrow P \wedge [x' = f(x) \& Q](P)')$$



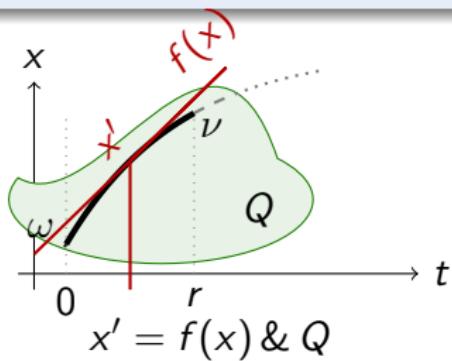
Differential invariant:  $p(x)$  true now and its differential  $(p(x))'$  true always  
 What's the differential of a formula???

What's the meaning of a differential term ... in a state???

## Axiom (Differential Effect)

(CADE'15)

$$\text{DE } [x' = f(x) \& Q]P \leftrightarrow [x' = f(x) \& Q][x' := f(x)]P$$



Effect of differential equation on differential symbol  $x'$

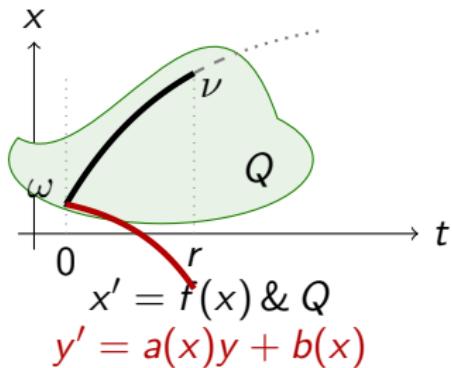
$[x' := f(x)]$  instantly mimics continuous effect  $[x' = f(x)]$  on  $x'$

$[x' := f(x)]$  selects vector field  $x' = f(x)$  for subsequent differentials

Axiom (Differential Ghost)

(CADE'15)

$$\text{DG } [x' = f(x) \& Q]P \leftrightarrow \exists y [x' = f(x), y' = a(x)y + b(x) \& Q]P$$

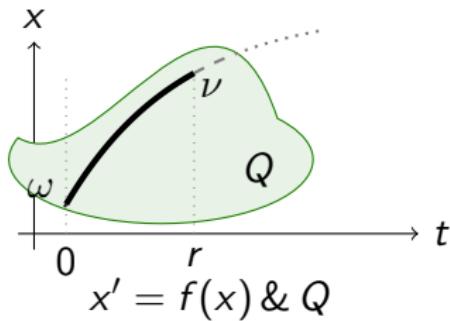


Differential ghost/auxiliaries: extra differential equations that exist  
Can cause new invariants  
“Dark matter” counterweight to balance conserved quantities

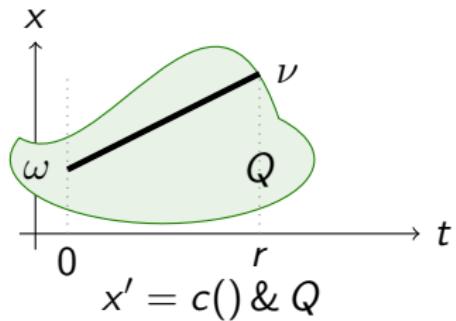
## Axiom (Differential Solution)

(CADE'15)

$$\text{DS } [x' = c() \& Q]P \leftrightarrow \forall t \geq 0 ((\forall 0 \leq s \leq t q(x + c()s)) \rightarrow [x := x + c()t]P)$$



Differential solutions: solve differential equations  
with DG,DC and inverse companions



- ① DI proves a property of an ODE inductively by its differentials
- ② DE exports vector field, possibly after DW exports evolution domain
- ③ CE+CQ reason efficiently in Equivalence or eQuational context
- ④ G isolates postcondition
- ⑤ [':=] differential substitution uses vector field
- ⑥ .' differential computations are axiomatic (US)

$$\begin{array}{c}
 * \\
 \frac{\text{QE} \quad \frac{*}{\vdash x^3 \cdot x + x \cdot x^3 \geq 0}}{\vdash [x' := x^3] x' \cdot x + x \cdot x' \geq 0} \quad \frac{\text{US} \quad \frac{'}{(f(\bar{x}) \cdot g(\bar{x}))' = (f(\bar{x}))' \cdot g(\bar{x}) + f(\bar{x}) \cdot (g(\bar{x}))'} \\ \quad \frac{}{(x \cdot x)' = (x)' \cdot x + x \cdot (x)'} \\ \quad \frac{}{(x \cdot x)' = x' \cdot x + x \cdot x'}}{(x \cdot x)' \geq 0 \leftrightarrow x' \cdot x + x \cdot x' \geq 0} \\
 \quad \frac{\text{CQ} \quad \frac{(x \cdot x)' \geq 0 \leftrightarrow x' \cdot x + x \cdot x' \geq 0}{(x \cdot x \geq 1)' \leftrightarrow x' \cdot x + x \cdot x' \geq 0}}{\vdash [x' = x^3][x' := x^3](x \cdot x \geq 1)'}
 \end{array}$$

QE      \*      \*      \*      \*      \*

':=      |      |      |      |      |

G      |      |      |      |      |

CE      |      |      |      |      |

DE      |      |      |      |      |

DI      |      |      |      |      |

$x \cdot x \geq 1 \vdash [x' = x^3] x \cdot x \geq 1$

# Differential Substitution Lemmas

## Lemma (Differential lemma)

If  $\varphi \models x' = f(x) \wedge Q$  for duration  $r > 0$ , then for all  $0 \leq \zeta \leq r$ :

$$\text{Syntactic} \quad \llbracket (e)' \rrbracket \varphi(\zeta) = \frac{d \llbracket e \rrbracket \varphi(t)}{dt}(\zeta) \quad \text{Analytic}$$

## Lemma (Differential assignment)

If  $\varphi \models x' = f(x) \wedge Q$  then  $\varphi \models P \leftrightarrow [x' := f(x)]P$

## Lemma (Derivations)

$$(e + k)' = (e)' + (k)'$$

$$(e \cdot k)' = (e)' \cdot k + e \cdot (k)'$$

$$[y := e][y' := 1]((f(e))' = (f(y))' \cdot (e)') \quad \text{for } y, y' \notin e$$

$$(c())' = 0 \quad \text{for arity 0 functions/numbers } c()$$

## Theorem (Differential radical invariant characterization)

$$\frac{h = 0 \rightarrow \bigwedge_{i=0}^{N-1} h_p^{(i)} = 0}{h = 0 \rightarrow [x' = p]h = 0}$$

characterizes **all algebraic invariants**, where  $N = \text{ord } \sqrt[N]{(h)}$ , i.e.

$$h_p^{(N)} = \sum_{i=0}^{N-1} g_i h_p^{(i)} \quad (g_i \in \mathbb{R}[x]) \quad h_p^{(i+1)} = [x' := p](h_p^{(i)})'$$

## Corollary (Algebraic Invariants Decidable)

*Algebraic invariants of algebraic differential equations are decidable.*

with Khalil Ghorbal TACAS'14

# Case Study: Longitudinal Dynamics of an Airplane

## Study (6th Order Longitudinal Flight Equations)

$$u' = \frac{X}{m} - g \sin(\theta) - qw \quad \text{axial velocity}$$

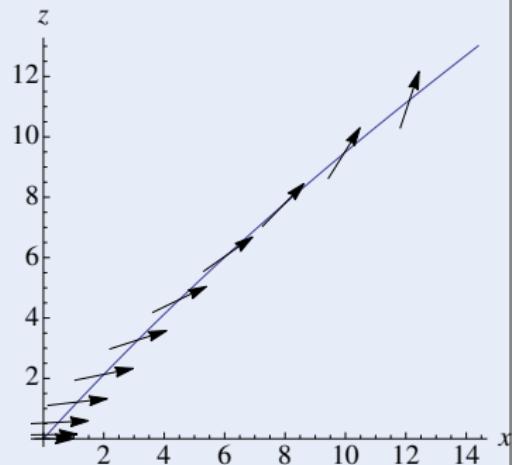
$$w' = \frac{Z}{m} + g \cos(\theta) + qu \quad \text{vertical velocity}$$

$$x' = \cos(\theta)u + \sin(\theta)w \quad \text{range}$$

$$z' = -\sin(\theta)u + \cos(\theta)w \quad \text{altitude}$$

$$\theta' = q \quad \text{pitch angle}$$

$$q' = \frac{M}{I_{yy}} \quad \text{pitch rate}$$



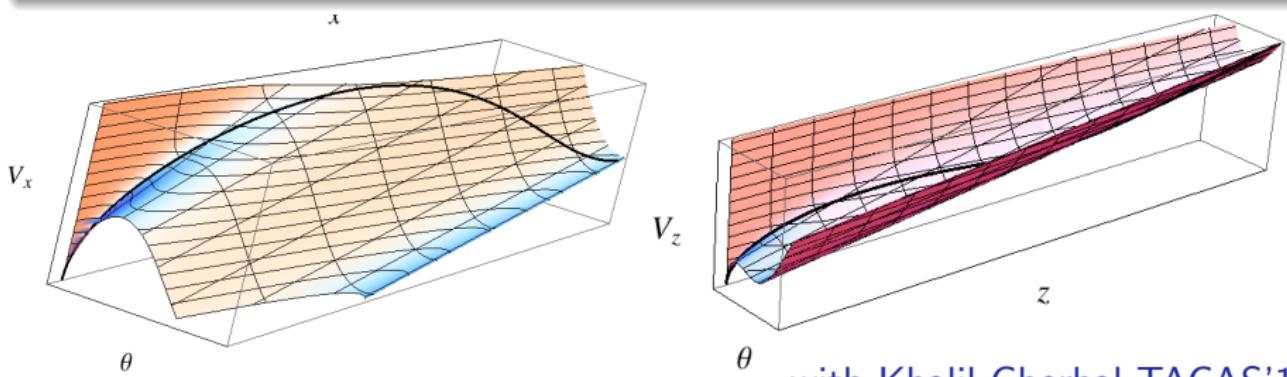
$X$  : thrust along  $u$      $Z$  : thrust along  $w$      $M$  : thrust moment for  $w$   
 $g$  : gravity                   $m$  : mass                   $I_{yy}$  : inertia second diagonal

with Khalil Ghorbal TACAS'14

# $\mathcal{R}$ Case Study: Longitudinal Dynamics of an Airplane

Result (DRI Automatically Generates Invariant Functions)

$$\begin{aligned} \frac{Mz}{I_{yy}} + g\theta + \left( \frac{X}{m} - qw \right) \cos(\theta) + \left( \frac{Z}{m} + qu \right) \sin(\theta) \\ \frac{Mx}{I_{yy}} - \left( \frac{Z}{m} + qu \right) \cos(\theta) + \left( \frac{X}{m} - qw \right) \sin(\theta) \\ - q^2 + \frac{2M\theta}{I_{yy}} \end{aligned}$$

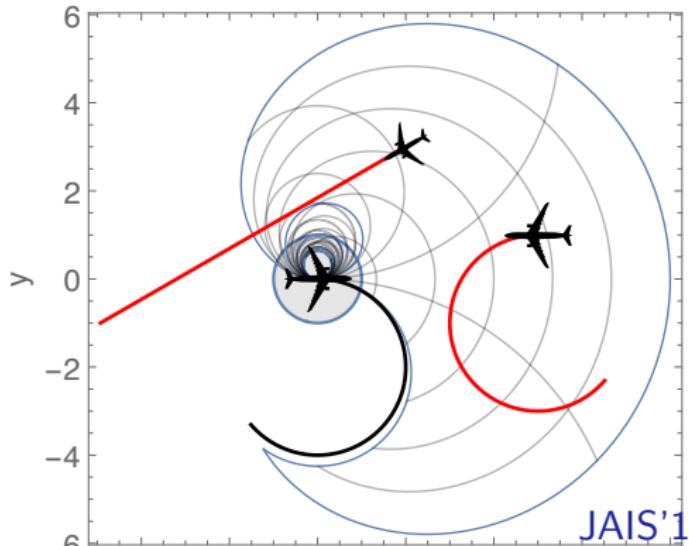
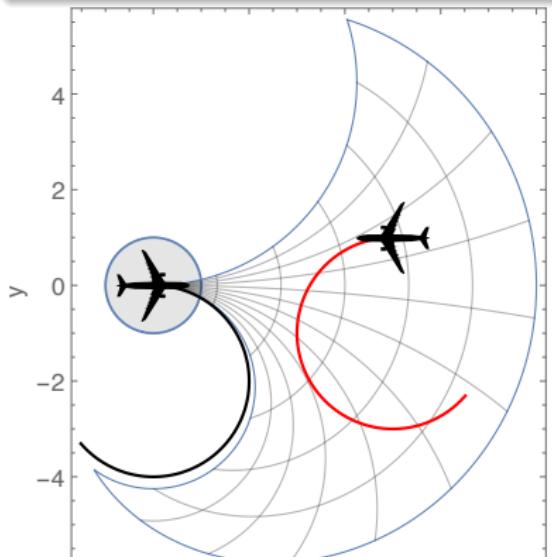


with Khalil Ghorbal TACAS'14

Result (DRI Automatically Generates Invariants)

$$\omega_1 = 0 \wedge \omega_2 = 0 \rightarrow v_2 \sin \vartheta x = (v_2 \cos \vartheta - v_1)y > p(v_1 + v_2)$$

$$\begin{aligned} \omega_1 \neq 0 \vee \omega_2 \neq 0 \rightarrow -\omega_1 \omega_2 (x^2 + y^2) + 2v_2 \omega_1 \sin \vartheta x + 2(v_1 \omega_2 - v_2 \omega_1 \cos \vartheta)y \\ + 2v_1 v_2 \cos \vartheta > 2v_1 v_2 + 2p(v_2 |\omega_1| + v_1 |\omega_2|) + p^2 |\omega_1 \omega_2| \end{aligned}$$



## 1 CPS are Multi-Dynamical Systems

- Hybrid Systems
- Hybrid Games

## 2 Dynamic Logic for Multi-Dynamical Systems

- Syntax
- Semantics

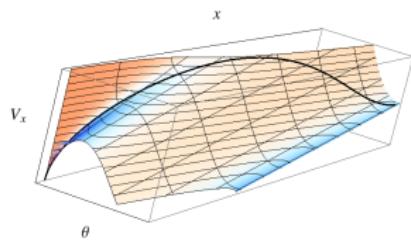
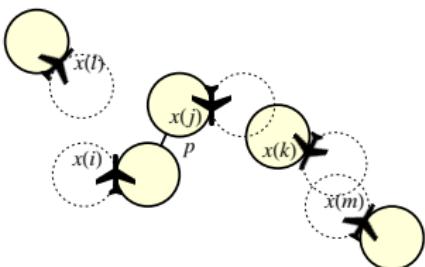
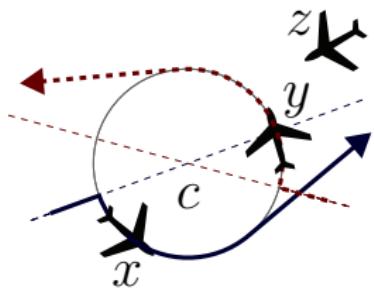
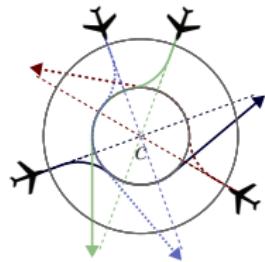
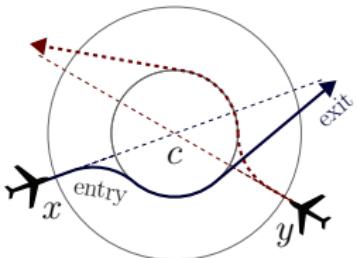
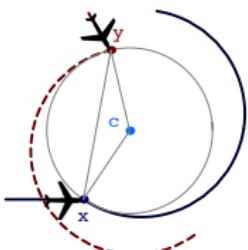
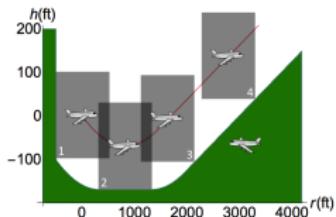
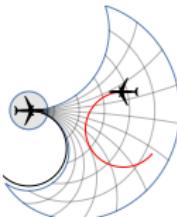
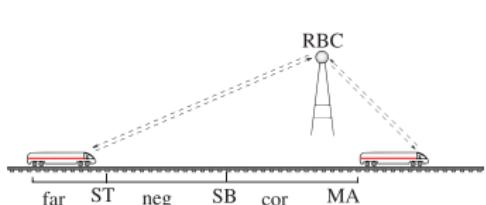
## 3 Proofs for CPS

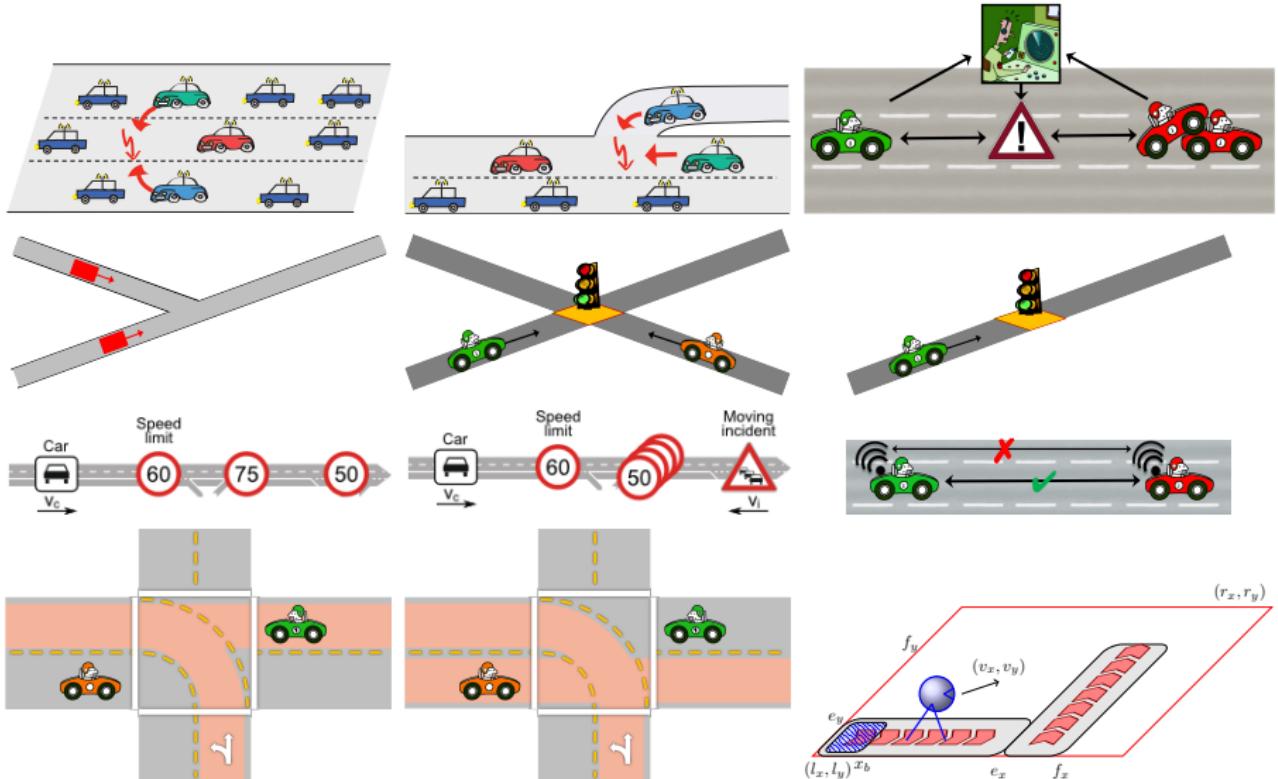
## 4 Theory of CPS

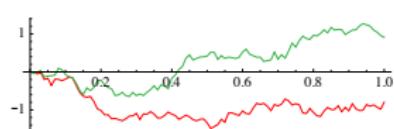
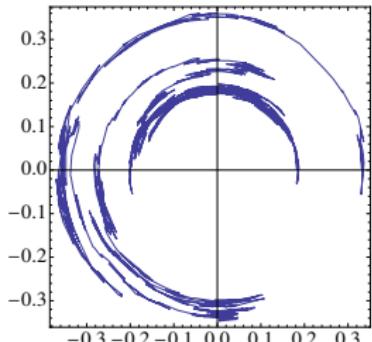
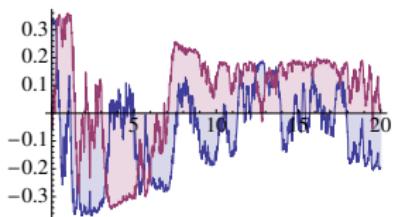
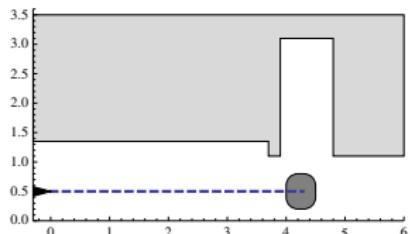
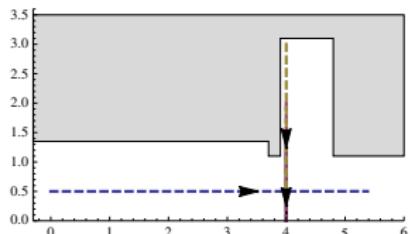
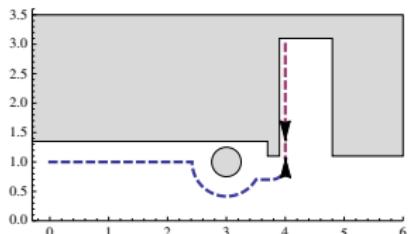
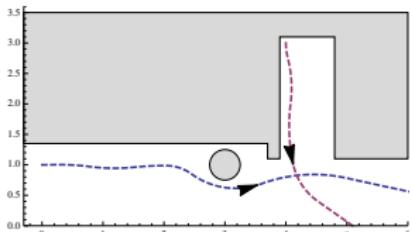
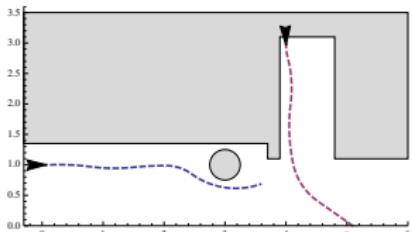
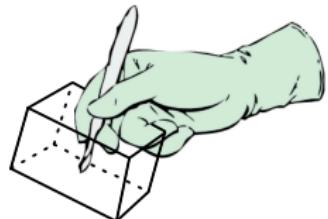
- Soundness and Completeness
- Differential Invariants
- Examples
- Differential Radical Invariants

## 5 Applications

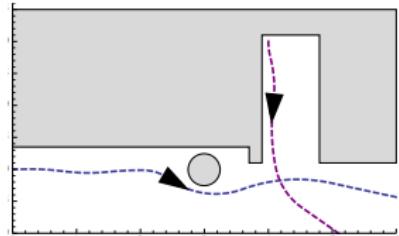
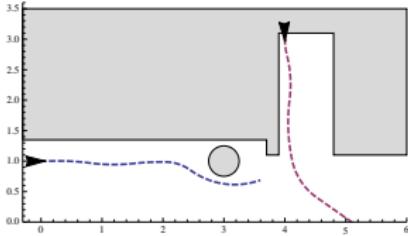
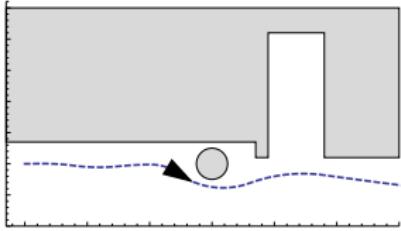
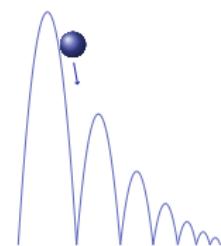
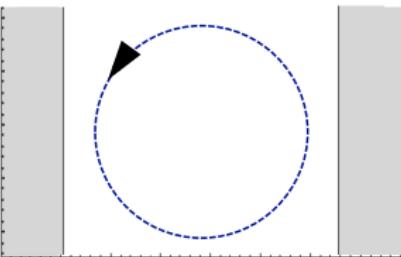
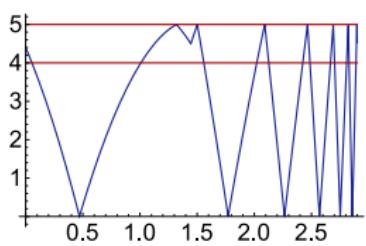
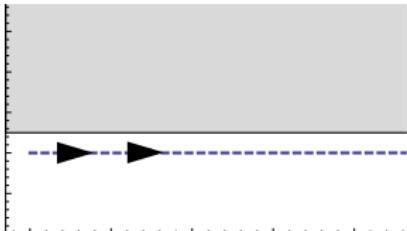
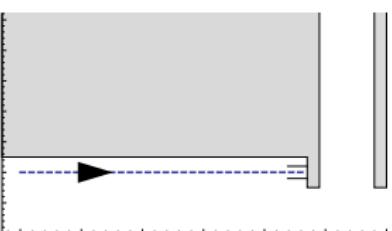
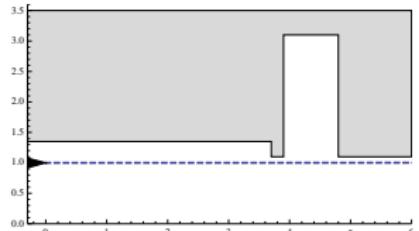
## 6 Summary

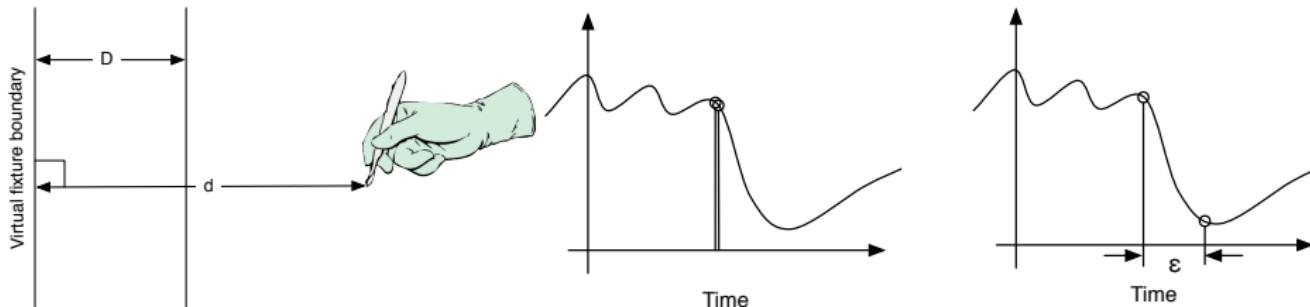




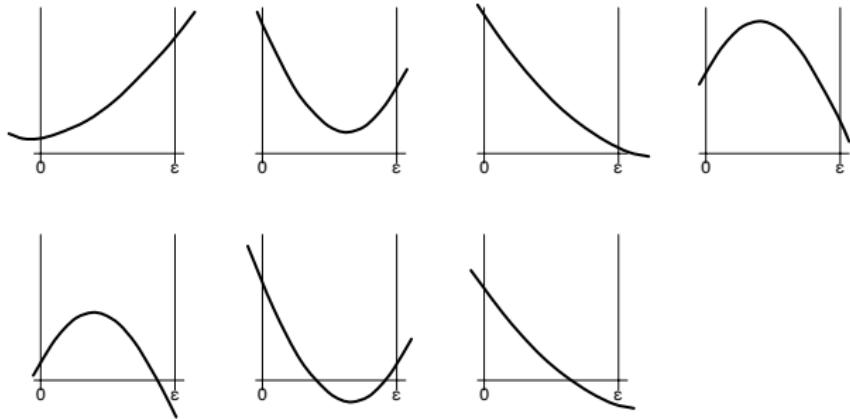


HSCC'13, RSS'13, CADE'12

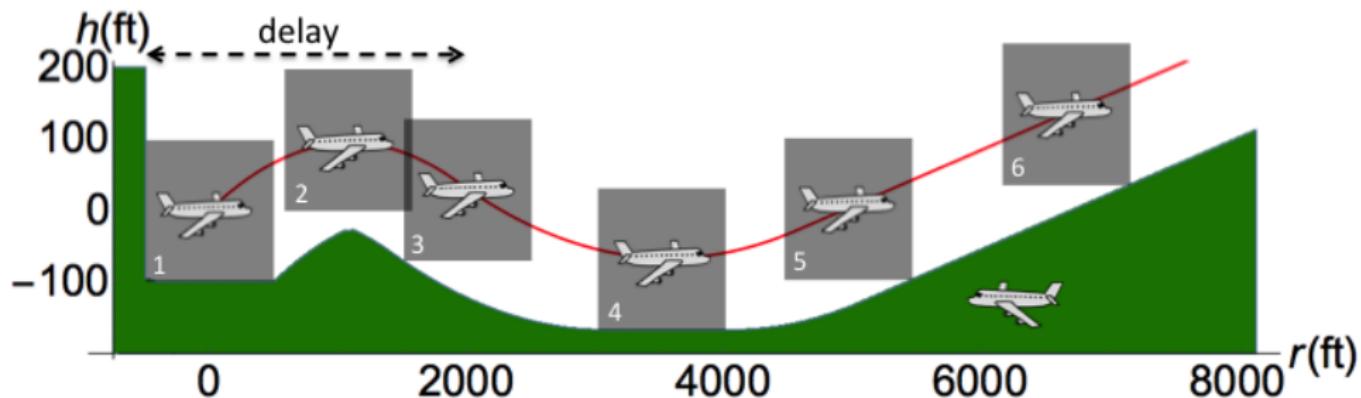




Redesign to predictive control

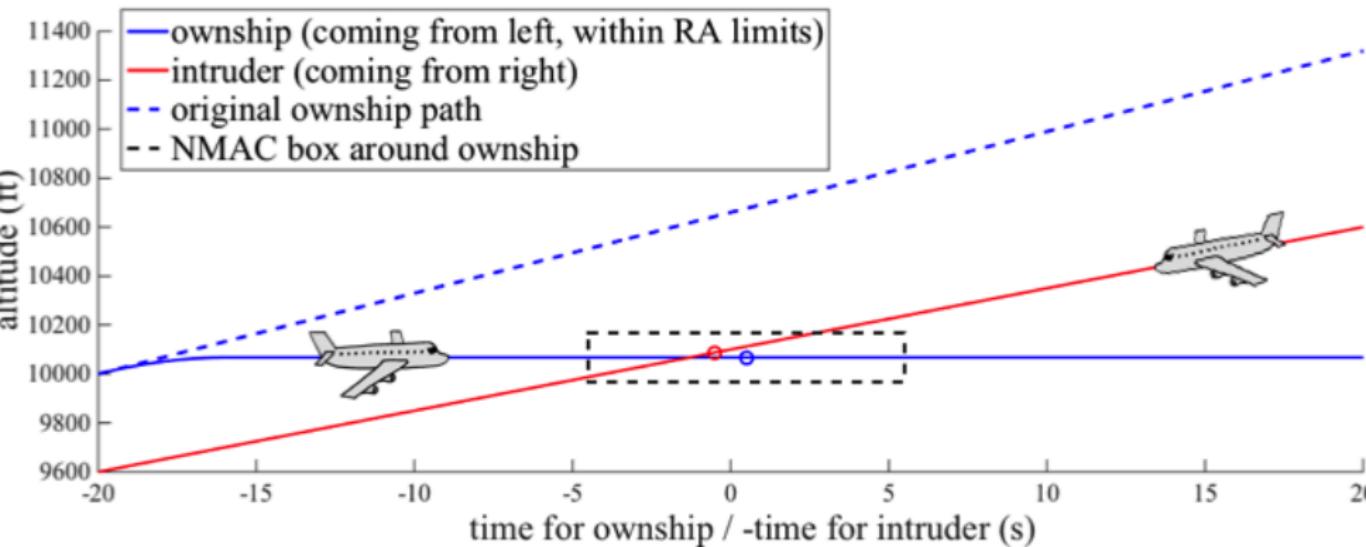


- Developed by the FAA to replace current TCAS in aircraft
- Approximately optimizes Markov Decision Process on a grid
- Advisory from lookup tables with numerous 5D interpolation regions



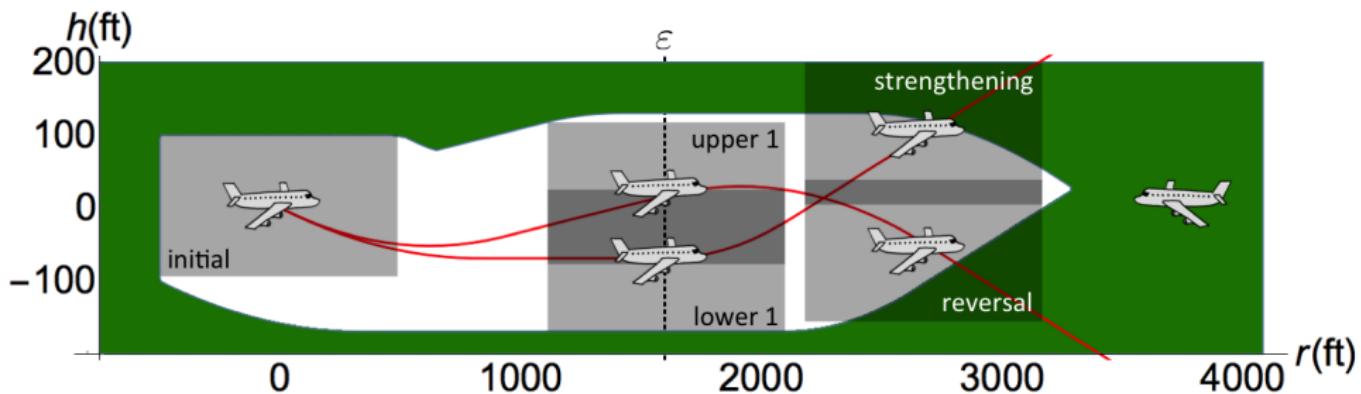
- ① Identified safe region for each advisory symbolically
- ② Proved safety for hybrid systems flight model in KeYmaera X

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

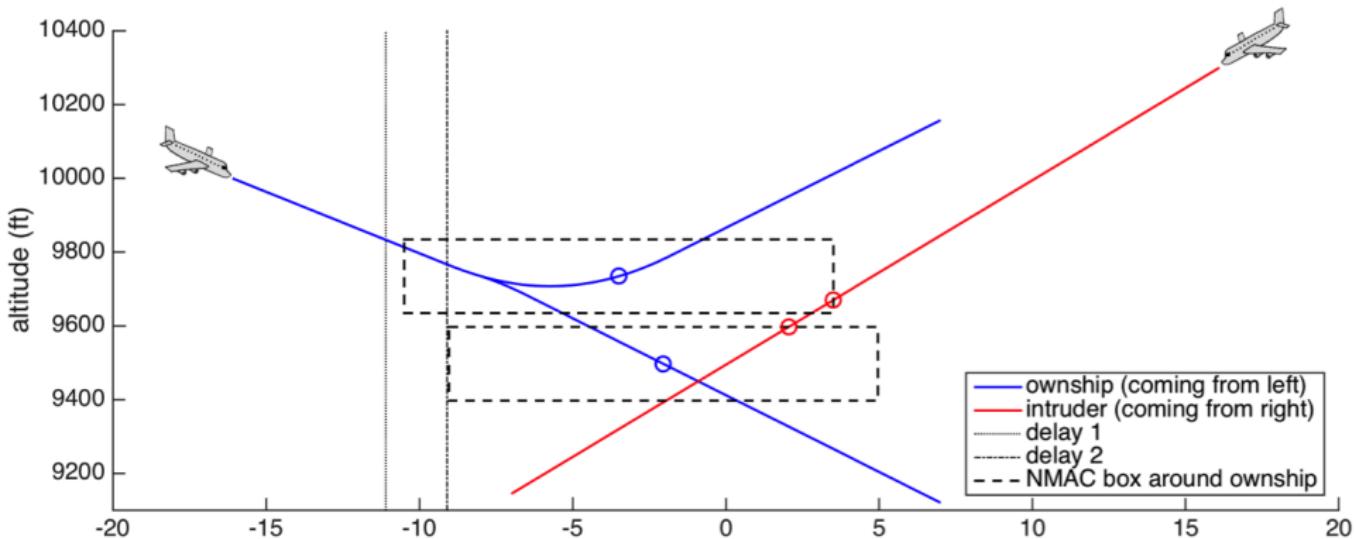
- Conservative, so too many counterexamples
- Settle for safe for a little while with safe possible future
- Safeable advisory: a subsequent advisory can safely avoid NMAC



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

ACAS X table comparison shows safeable advisory in more of the 648,591,384,375 states compared ( $\approx 31.6 \text{ to } 898.7 \cdot 10^6$  counterexamples).

**Counterexample: Action Issued = Maintain  
Followed by Most Extreme Up/Down-sense Advisory Available**

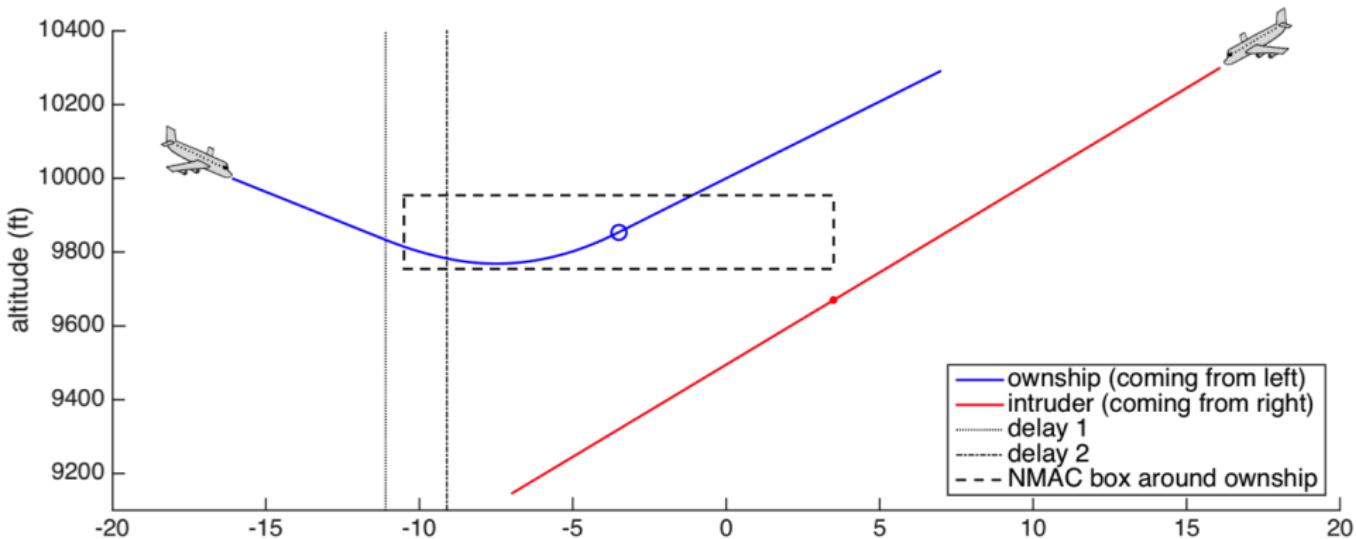


ACAS X issues Maintain advisory instead of CLI1500

STTT

ACAS X table comparison shows safeable advisory in more of the 648,591,384,375 states compared ( $\approx 31.6 \text{ to } 898.7 \cdot 10^6$  counterexamples).

**Safe Version: Action Issued = CL1500  
Followed by Most Extreme Up/Down-sense Available**



ACAS X issues Maintain advisory instead of CLI1500

## 1 CPS are Multi-Dynamical Systems

- Hybrid Systems
- Hybrid Games

## 2 Dynamic Logic for Multi-Dynamical Systems

- Syntax
- Semantics

## 3 Proofs for CPS

## 4 Theory of CPS

- Soundness and Completeness
- Differential Invariants
- Examples
- Differential Radical Invariants

## 5 Applications

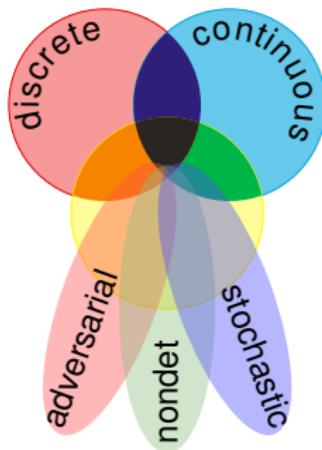
## 6 Summary

hybrid systems

$$\text{HS} = \text{discrete} + \text{ODE}$$

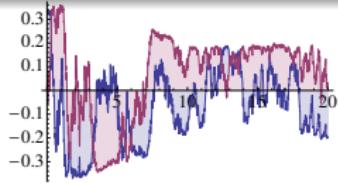
hybrid games

$$\text{HG} = \text{HS} + \text{adversary}$$



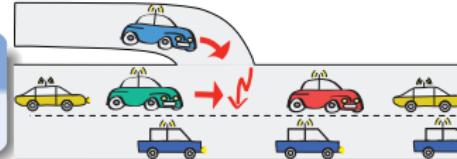
stochastic hybrid sys.

$$\text{SHS} = \text{HS} + \text{stochastics}$$



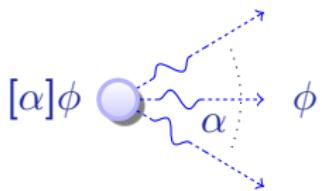
distributed hybrid sys.

$$\text{DHS} = \text{HS} + \text{distributed}$$



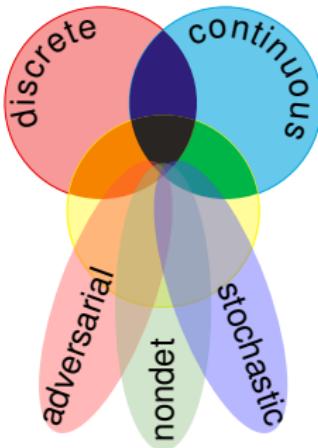
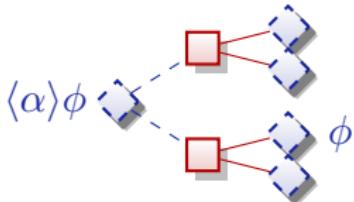
differential dynamic logic

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



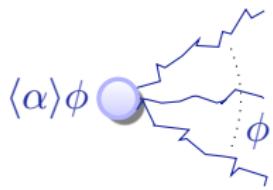
differential game logic

$$dG\mathcal{L} = GL + HG$$



stochastic differential DL

$$Sd\mathcal{L} = DL + SHP$$



quantified differential DL

$$Qd\mathcal{L} = FOL + DL + QHP$$

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

equations of truth

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

$$['] \quad [x' = f(x)]P \leftrightarrow \forall t \geq 0 [x := y(t)]P \quad (y'(t) = f(y))$$

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

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

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

$$\mathsf{K} \quad [\alpha](P \rightarrow Q) \rightarrow ([\alpha]P \rightarrow [\alpha]Q)$$

$$\mathsf{I} \quad [\alpha^*](P \rightarrow [\alpha]P) \rightarrow (P \rightarrow [\alpha^*]P)$$

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

LICS'12, CADE'15

# $\mathcal{R}$ Differential Equation Axioms & Differential Axioms

DW  $[x' = f(x) \& Q]Q$

$$\begin{aligned} \text{DC } ([x' = f(x) \& Q]P &\leftrightarrow [x' = f(x) \& Q \wedge r(x)]P) \\ &\leftarrow [x' = f(x) \& Q]r(x) \end{aligned}$$

DE  $[x' = f(x) \& Q]P \leftrightarrow [x' = f(x) \& Q][x' := f(x)]P$

DI  $[x' = f(x) \& Q]P \leftarrow (Q \rightarrow P \wedge [x' = f(x) \& Q](P)')$

DG  $[x' = f(x) \& Q]P \leftrightarrow \exists y [x' = f(x), y' = a(x)y + b(x) \& Q]P$

DS  $[x' = c() \& Q]P \leftrightarrow \forall t \geq 0 ((\forall 0 \leq s \leq t q(x + c()(s))) \rightarrow [x := x + c()t]P)$

$[':=]$   $[x' := e]p(x') \leftrightarrow p(e)$

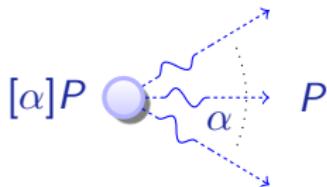
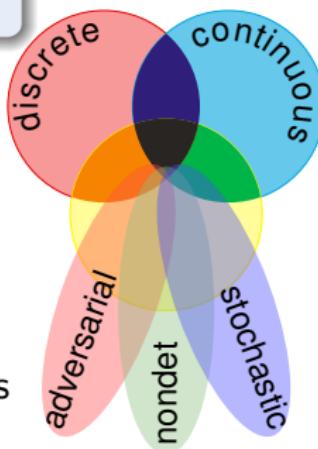
$$+' (e + k)' = (e)' + (k)'$$

$$\cdot' (e \cdot k)' = (e)' \cdot k + e \cdot (k)'$$

$$\circ' [y := g(x)][y' := 1]((f(g(x))))' = (f(y))' \cdot (g(x))'$$

differential dynamic logic

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



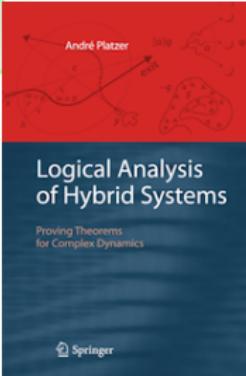
- Multi-dynamical systems
- Combine simple dynamics
- Tame complexity
- Logic & proofs for CPS
- Theory for CPS
- Applications
- Course: Foundations of CPS

KeYmaera X

The screenshot shows the KeYmaera X interface with several tabs: Agenda, Overview, Induction Step, and Rule Application. The Induction Step tab displays an invariant  $v \geq 0 \wedge A > 0 \wedge B > 0 \wedge v \geq 0 \wedge B > 0 \wedge A > 0$  and an induction step involving a choice between two cases based on  $v$  and  $A$ . The Rule Application tab shows a proof step involving a choice rule and a vector equation  $(v, u) \xrightarrow{\text{choice}} (v', u')$ .



# Logical Foundations of Cyber-Physical Systems





André Platzer.

Logics of dynamical systems.

In LICS [23], pages 13–24.

doi:10.1109/LICS.2012.13.



André Platzer.

Foundations of cyber-physical systems.

Lecture Notes 15-424/624, Carnegie Mellon University, 2014.

URL: <http://www.cs.cmu.edu/~aplatzer/course/fcps14/fcps14.pdf>.



André Platzer.

*Logical Analysis of Hybrid Systems: Proving Theorems for Complex Dynamics.*

Springer, Heidelberg, 2010.

doi:10.1007/978-3-642-14509-4.



André Platzer.

Differential dynamic logic for hybrid systems.

*J. Autom. Reas.*, 41(2):143–189, 2008.

[doi:10.1007/s10817-008-9103-8](https://doi.org/10.1007/s10817-008-9103-8).



André Platzer.

A uniform substitution calculus for differential dynamic logic.

In Amy Felty and Aart Middeldorp, editors, *CADE*, volume 9195 of *LNCS*, pages 467–481. Springer, 2015.

[doi:10.1007/978-3-319-21401-6\\_32](https://doi.org/10.1007/978-3-319-21401-6_32).



André Platzer.

Differential game logic.

*ACM Trans. Comput. Log.*, 17(1):1:1–1:51, 2015.

[doi:10.1145/2817824](https://doi.org/10.1145/2817824).



André Platzer.

The complete proof theory of hybrid systems.

In LICS [23], pages 541–550.

[doi:10.1109/LICS.2012.64](https://doi.org/10.1109/LICS.2012.64).



André Platzer.

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



André Platzer.

Stochastic differential dynamic logic for stochastic hybrid programs.

In Nikolaj Bjørner and Viorica Sofronie-Stokkermans, editors, *CADE*, volume 6803 of *LNCS*, pages 431–445. Springer, 2011.

doi:10.1007/978-3-642-22438-6\_34.



Stefan Mitsch and André Platzer.

ModelPlex: Verified runtime validation of verified cyber-physical system models.

*Form. Methods Syst. Des.*, 2016.

Special issue of selected papers from RV'14.

doi:10.1007/s10703-016-0241-z.



Stefan Mitsch, Jan-David Quesel, and André Platzer.

Refactoring, refinement, and reasoning: A logical characterization for hybrid systems.

In Cliff B. Jones, Pekka Pihlajasaari, and Jun Sun, editors, *FM*, volume 8442 of *LNCS*, pages 481–496. Springer, 2014.  
[doi:10.1007/978-3-319-06410-9\\_33](https://doi.org/10.1007/978-3-319-06410-9_33).

 Nikos Aréchiga, Sarah M. Loos, André Platzer, and Bruce H. Krogh. Using theorem provers to guarantee closed-loop system properties. In Dawn Tilbury, editor, *ACC*, pages 3573–3580, 2012.  
[doi:10.1109/ACC.2012.6315388](https://doi.org/10.1109/ACC.2012.6315388).

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

 André Platzer and Edmund M. Clarke. Computing differential invariants of hybrid systems as fixedpoints. In Aarti Gupta and Sharad Malik, editors, *CAV*, volume 5123 of *LNCS*, pages 176–189. Springer, 2008.  
[doi:10.1007/978-3-540-70545-1\\_17](https://doi.org/10.1007/978-3-540-70545-1_17).

 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](https://doi.org/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](https://doi.org/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](https://doi.org/10.1007/978-3-642-32347-8_3).



Khalil Ghorbal and André Platzer.

Characterizing algebraic invariants by differential radical invariants.

In Erika Ábrahám and Klaus Havelund, editors, *TACAS*, volume 8413 of *LNCS*, pages 279–294. Springer, 2014.

[doi:10.1007/978-3-642-54862-8\\_19](https://doi.org/10.1007/978-3-642-54862-8_19).



Yanni Kouskoulas, David W. Renshaw, André Platzer, and Peter Kazanzides.

Certifying the safe design of a virtual fixture control algorithm for a surgical robot.

In Calin Belta and Franjo Ivancic, editors, *HSCC*, pages 263–272. ACM, 2013.

[doi:10.1145/2461328.2461369](https://doi.org/10.1145/2461328.2461369).



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](https://doi.org/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 Press, 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, 2015.  
<http://www.cs.cmu.edu/~aplatzer/pub/acasx-long.pdf>.

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