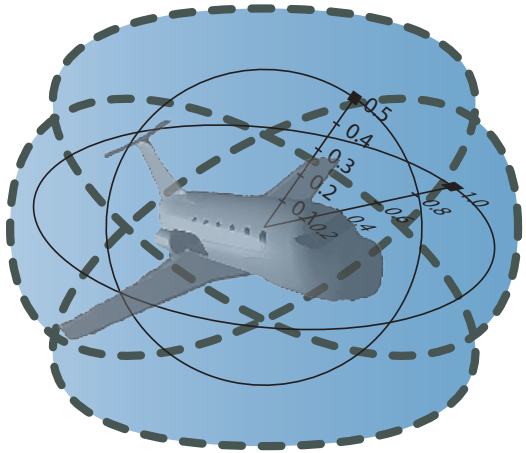
Logical Foundations & Cyber-Physical Systems

André Platzer

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

http://symbolaris.com/



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
   - Differential Radical Invariants
5. Applications
6. Summary

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
   - 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 & society
3. Need even stronger foundations when software reaches out into our physical world

Cyber-physical Systems

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

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

Benefits of Logical Foundations for V & V

Proving

Safety	Formalize system properties: What is "Safe"? "Reach goal"?
Models	Formalize system models, clarify behavior
Assumptions	Make assumptions explicit rather than silent
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

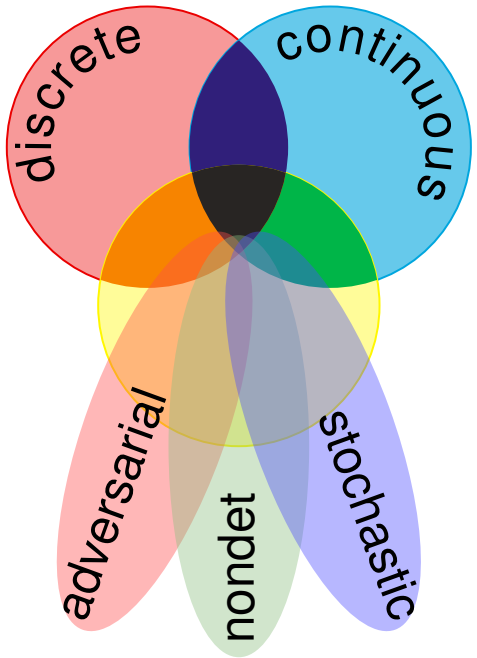
Tools

KeYmaera	Theorem prover for CPS

CPS are Multi-Dynamical Systems

CPS Dynamics

CPS are characterized by multiple facets of dynamical systems.



CPS Compositions

CPS combine multiple simple dynamical effects.

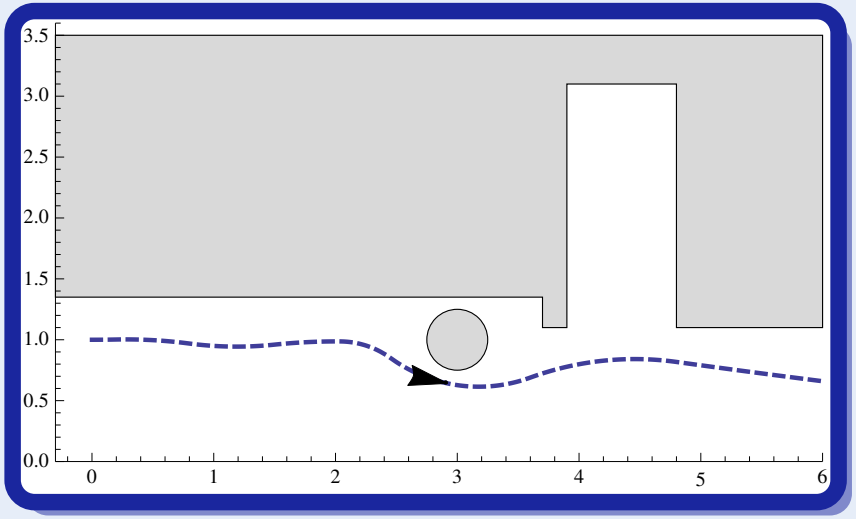
Tame Parts

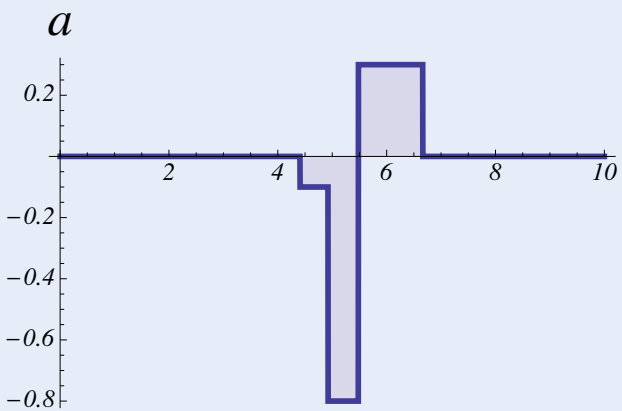
Exploiting compositionality tames CPS complexity.

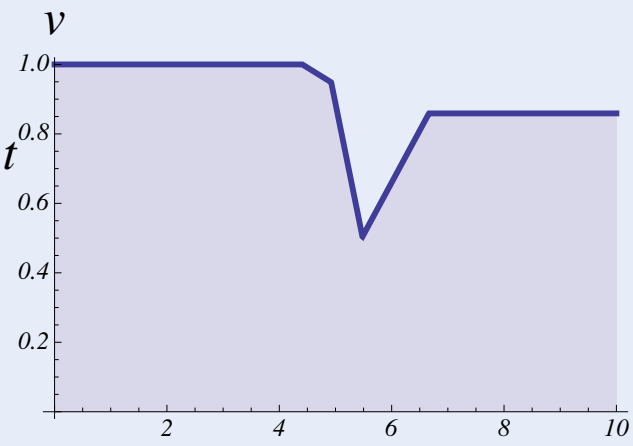
Challenge (Hybrid Systems)

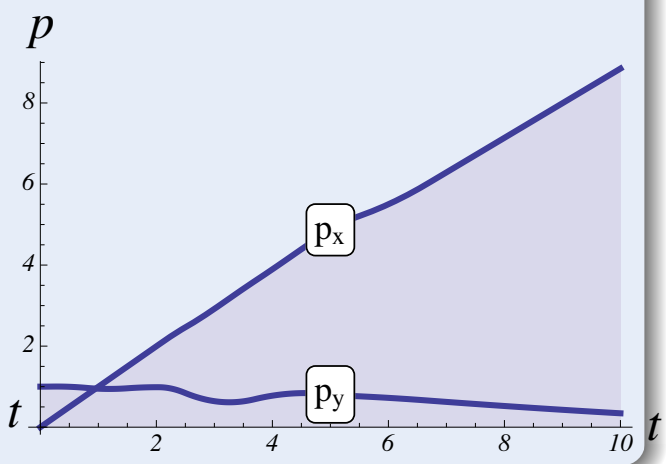
Fixed rule describing state evolution with both

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





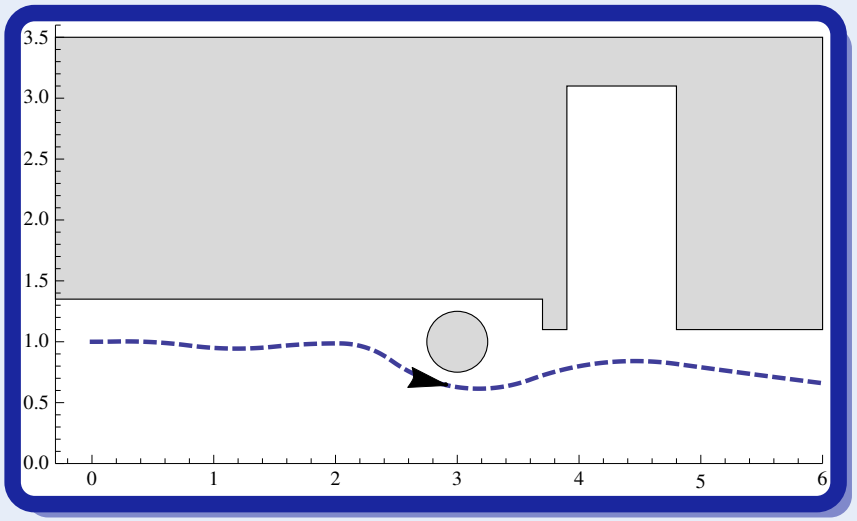


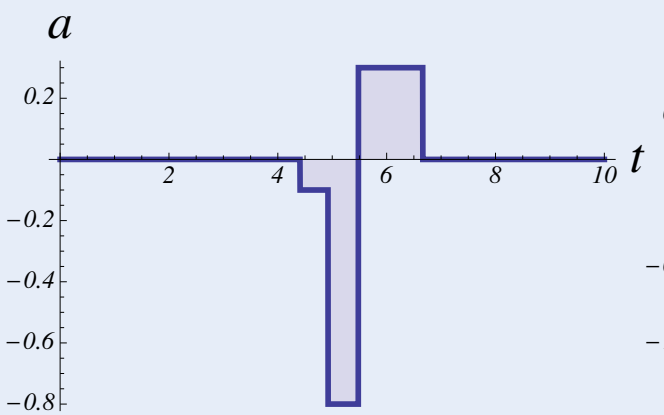


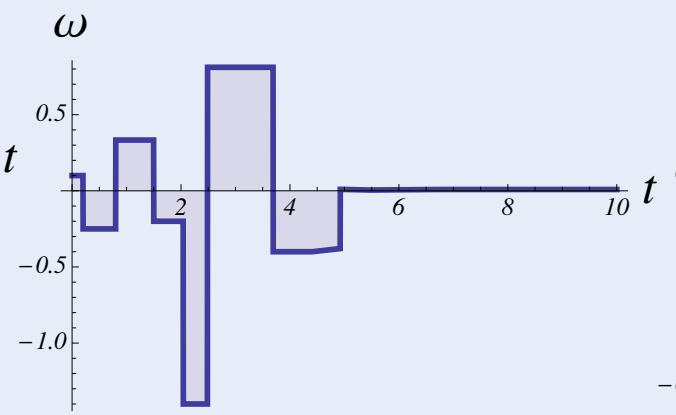
Challenge (Hybrid Systems)

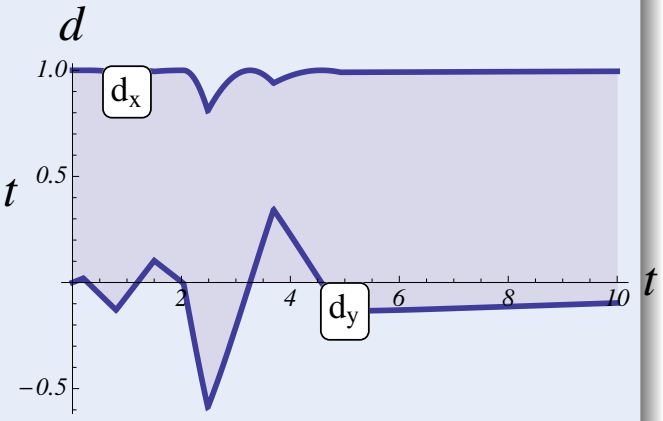
Fixed rule describing state evolution with both

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





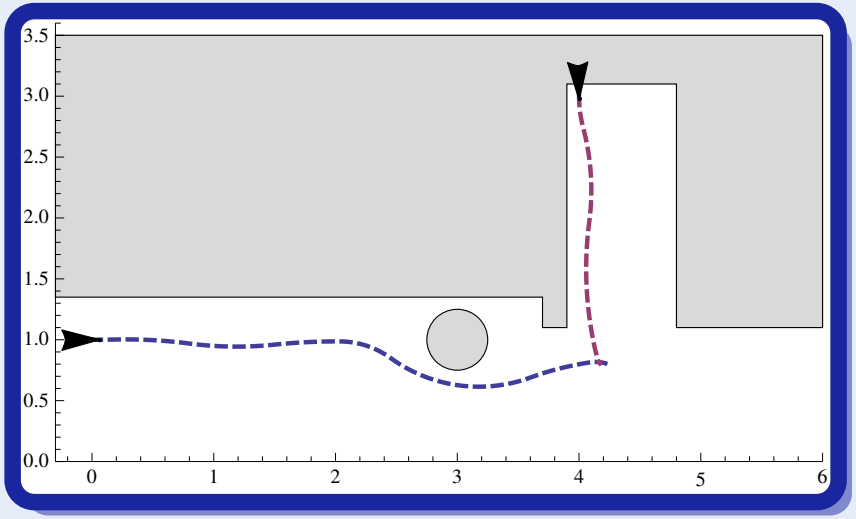


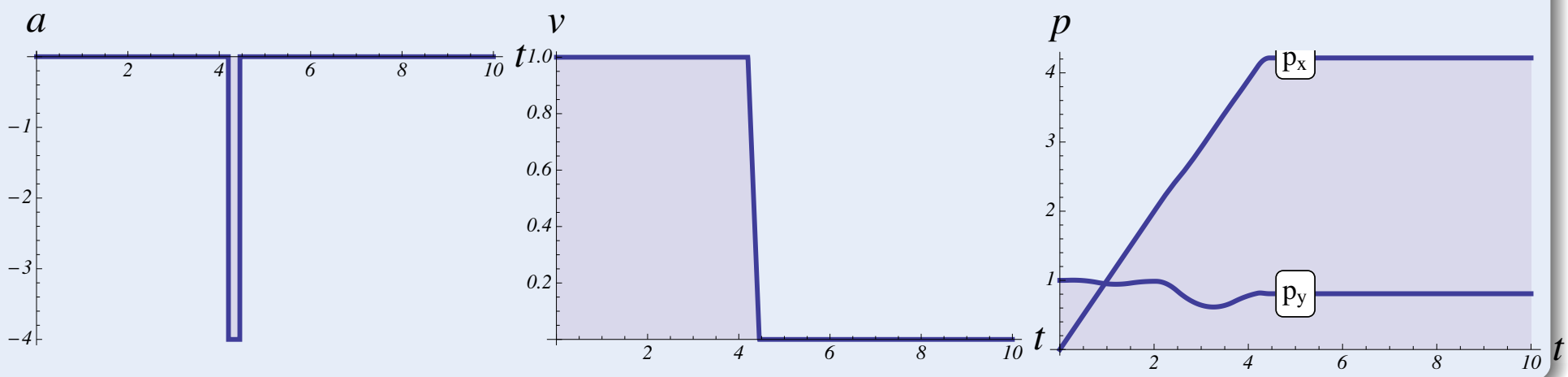


Challenge (Hybrid Systems)

Fixed rule describing state evolution with both

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

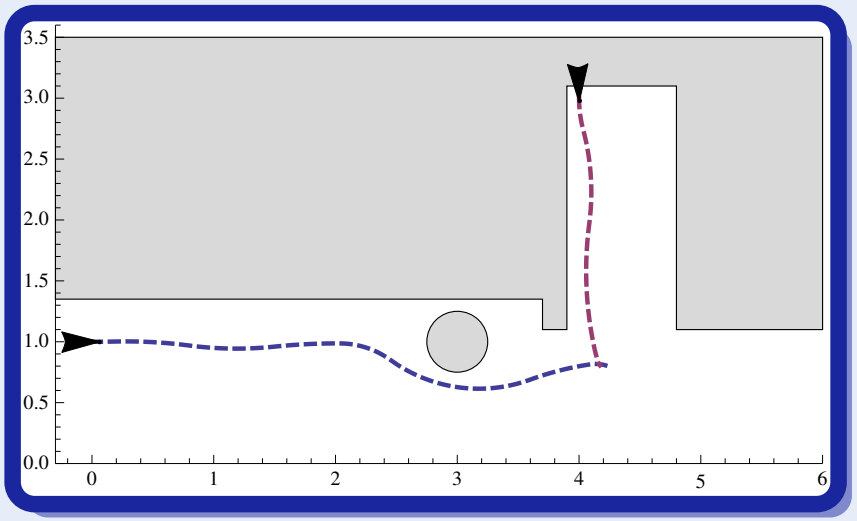


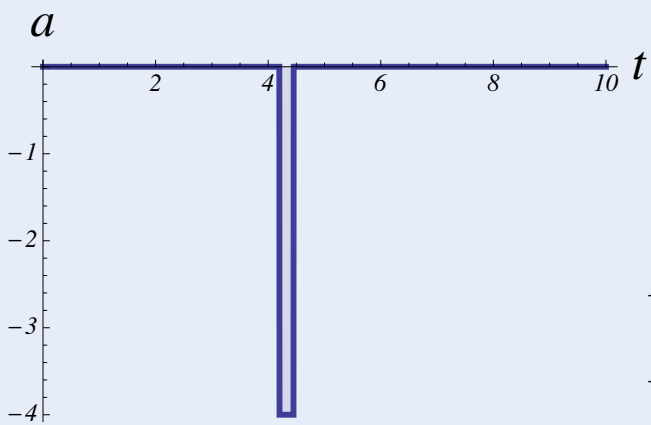


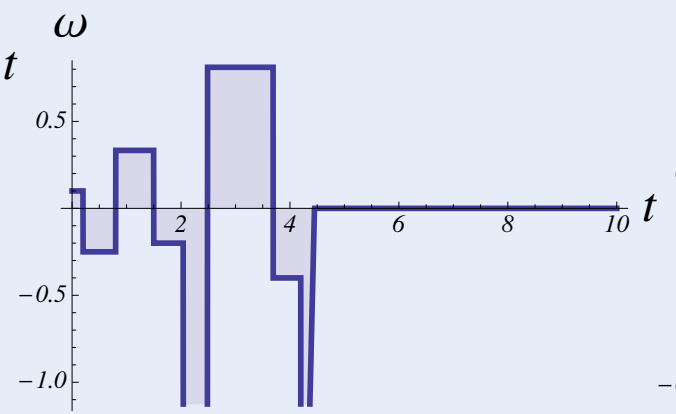
Challenge (Hybrid Systems)

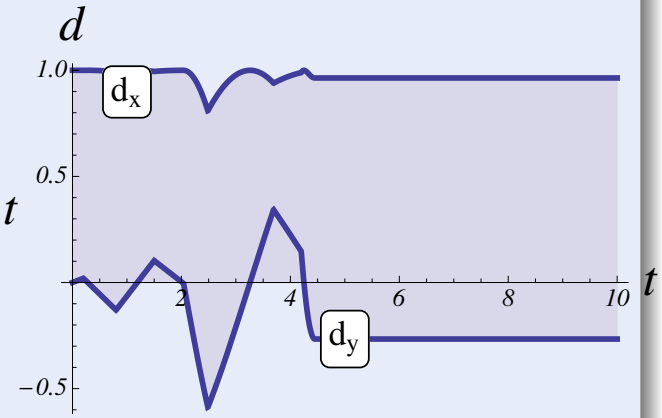
Fixed rule describing state evolution with both

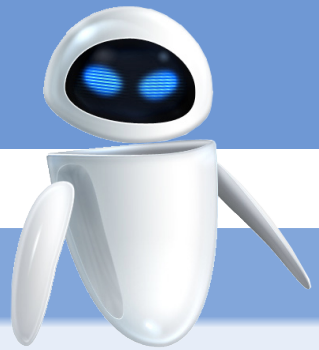
- Discrete dynamics (control decisions)
- Continuous dynamics (differential equations)







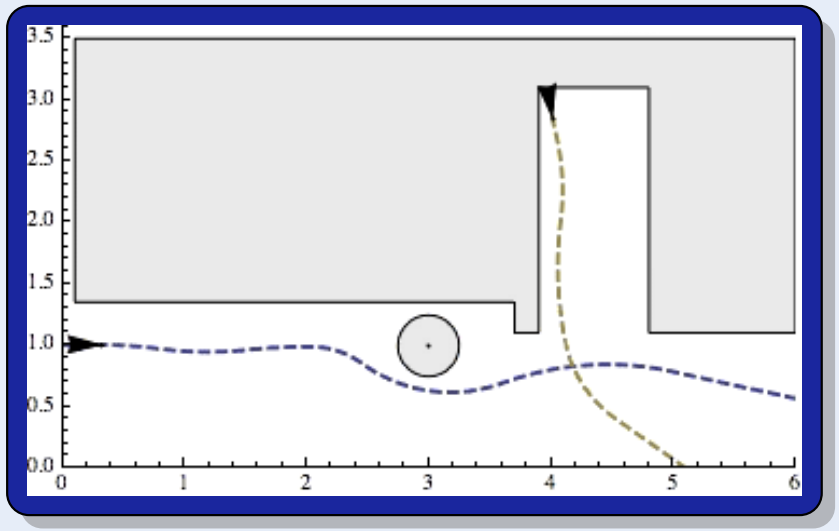


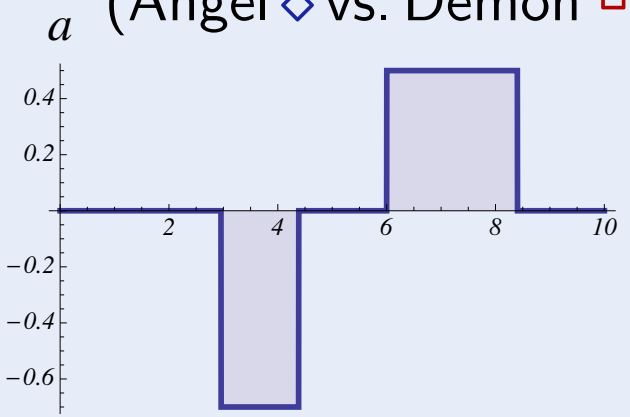


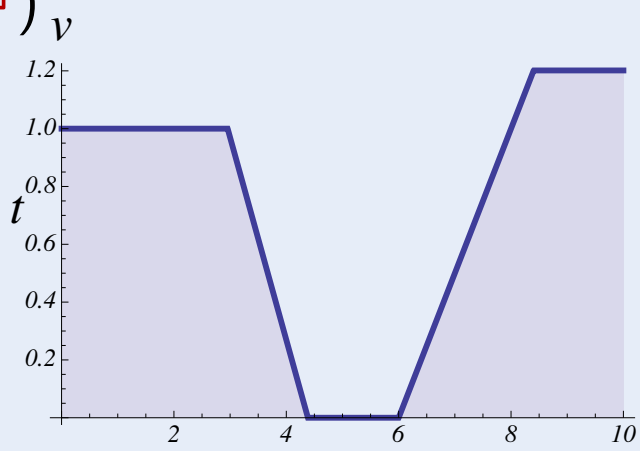
Challenge (Hybrid Games)

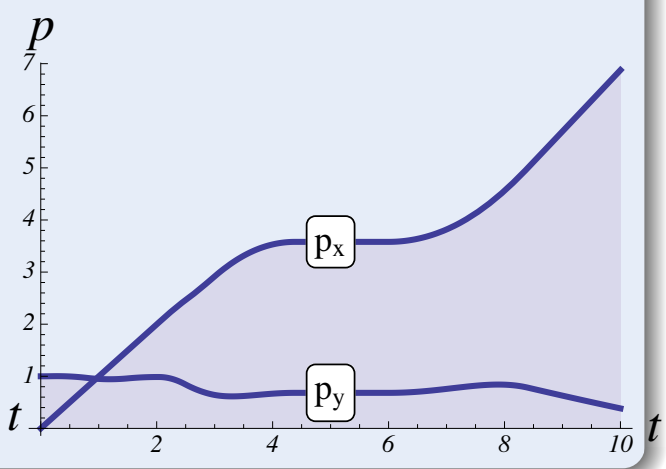
Game rules describing play evolution with

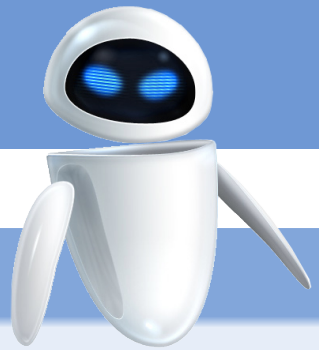
- Discrete dynamics (control decisions)
- Continuous dynamics (differential equations)
- Adversarial dynamics (Angel ◇ vs. Demon □)







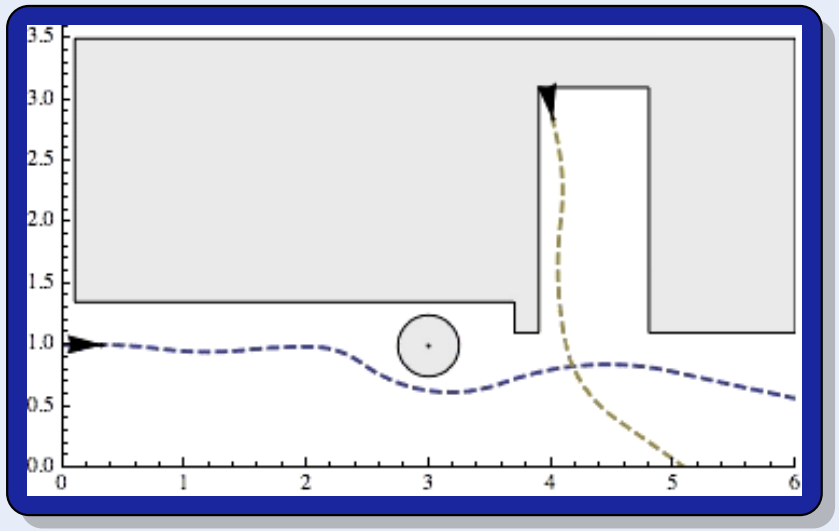


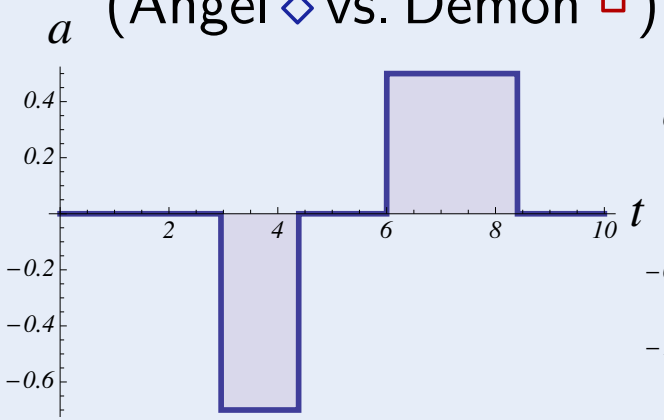


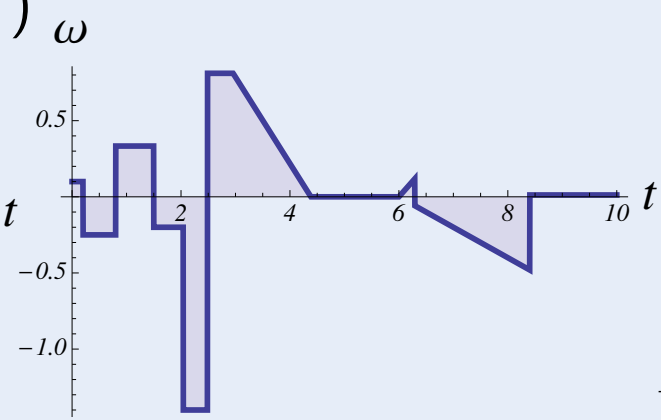
Challenge (Hybrid Games)

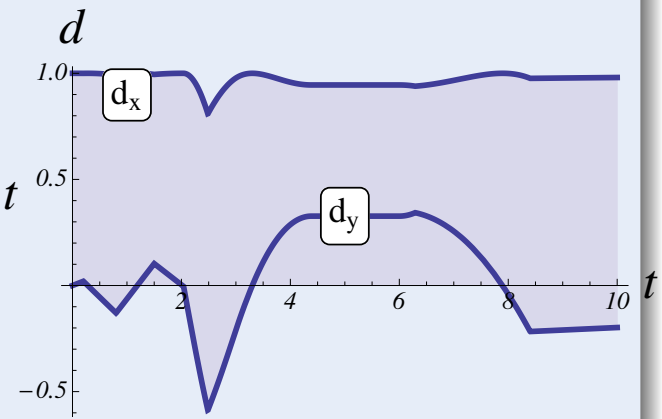
Game rules describing play evolution with

- Discrete dynamics (control decisions)
- Continuous dynamics (differential equations)
- Adversarial dynamics (Angel ◇ vs. Demon □)









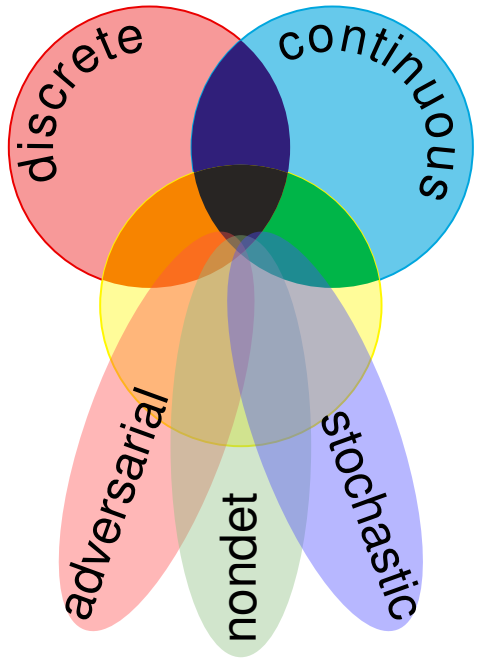
Outline

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
   - Differential Radical Invariants
5. Applications
6. Summary

CPS are Multi-Dynamical Systems

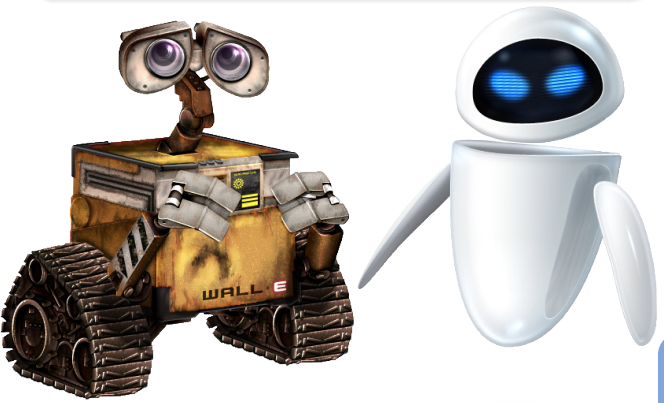
hybrid systems

HS = discrete + ODE



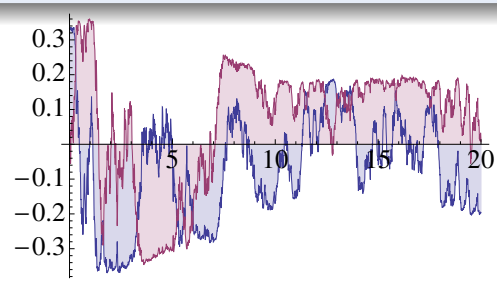
hybrid games

HG = HS + adversary



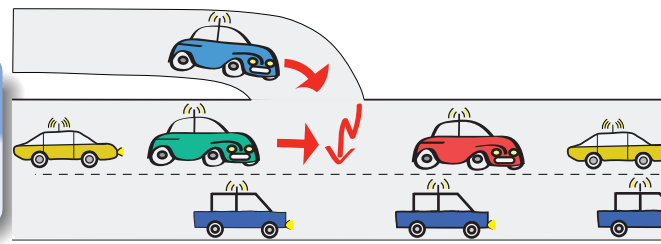
stochastic hybrid sys.

SHS = HS + stochastics

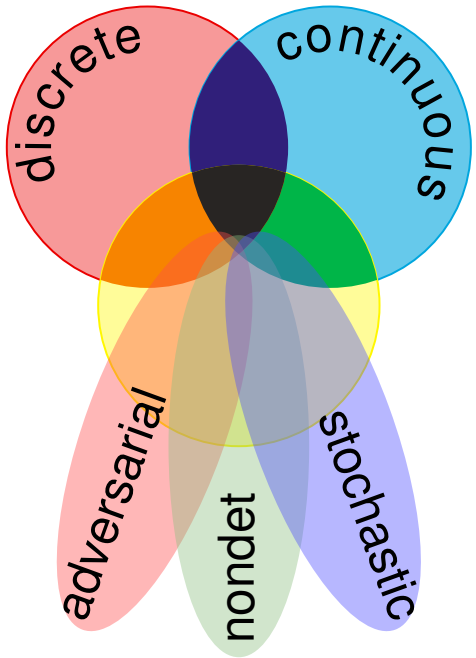


distributed hybrid sys.

DHS = HS + distributed



Family of Differential Dynamic Logics

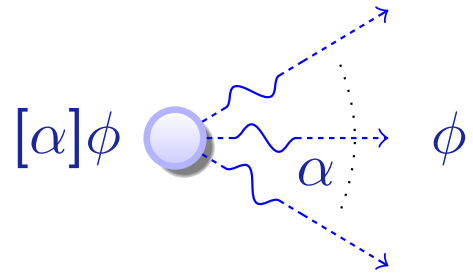


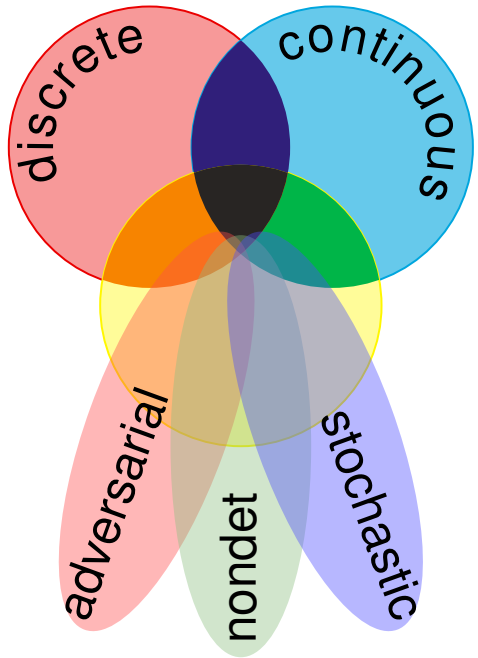
JAR'08,CADE'11,LMCS'12,LICS'12,LICS'12

Family of Differential Dynamic Logics

differential dynamic logic

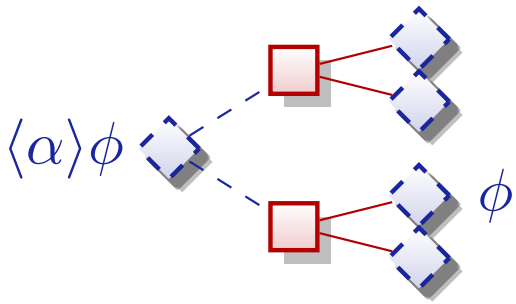
$$\mathsf{d}\mathcal{L} = \mathsf{DL} + \mathsf{HP}$$





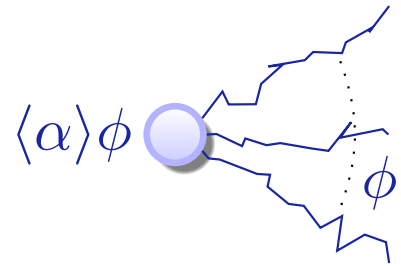
differential game logic

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



stochastic differential DL

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



quantified differential DL

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

JAR'08,CADE'11,LMCS'12,LICS'12,LICS'12

Benefits

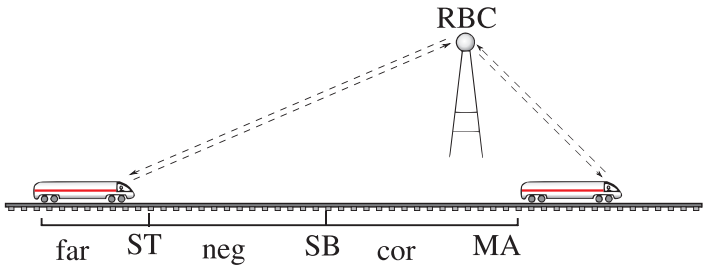
Logical foundations of cyber-physical systems

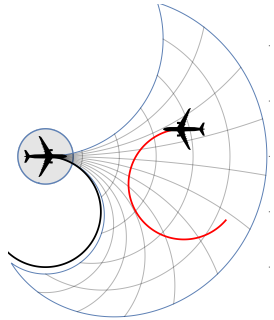
1. Multi-dynamical systems
2. Tame complexity by combinations of simple dynamics
3. Compositional programming language for CPS
4. Compositional logics and proof calculi
5. Proofs for differential equations
6. Elegant theory
7. Many useful applications
8. Education: undergrad course Foundations of CPS

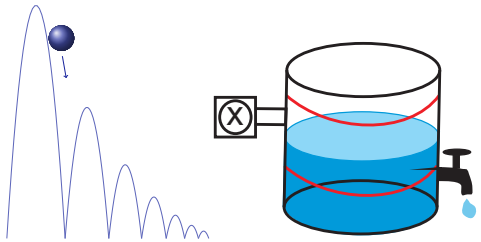
Basis for other technology

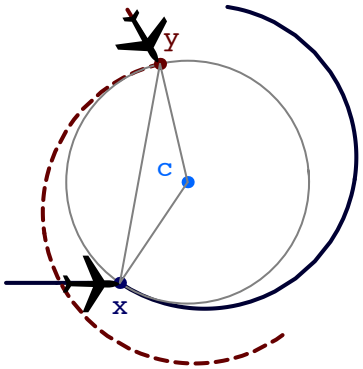
1. ModelPlex transfers CPS model $\rightsquigarrow$ CPS implementation safety RV'14
2. Proof-aware refactoring to co-evolve model + proof FM'14
3. Control envelope design ACC'12

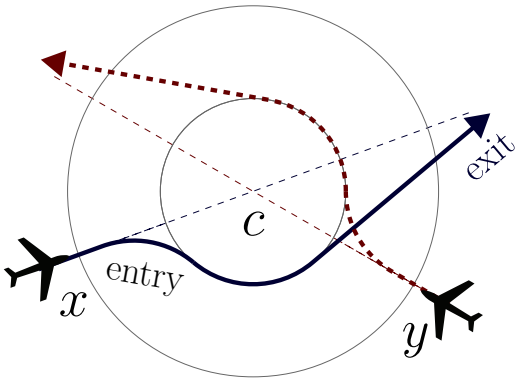
Successful CPS Proofs

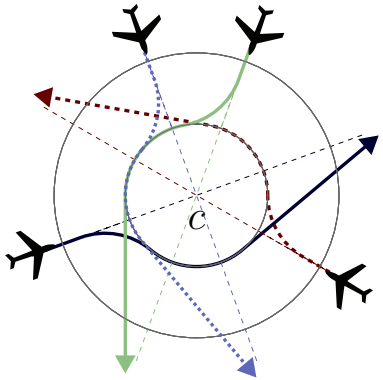


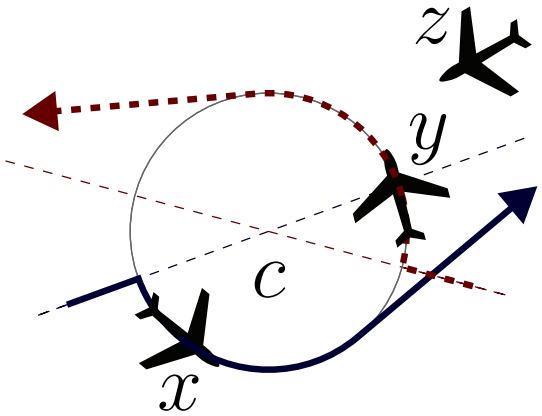


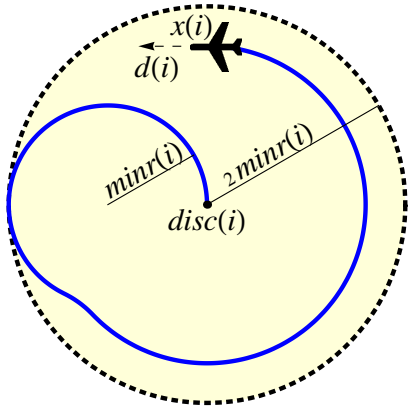


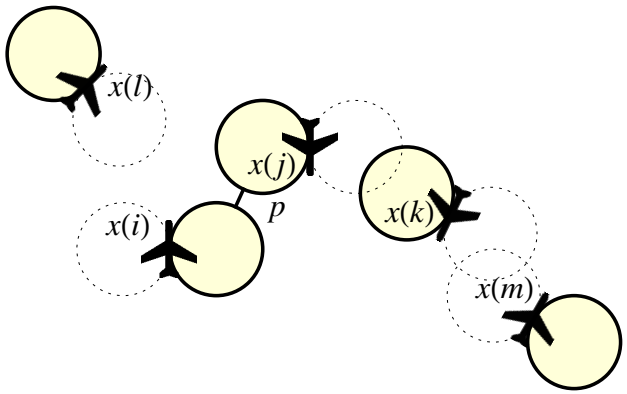






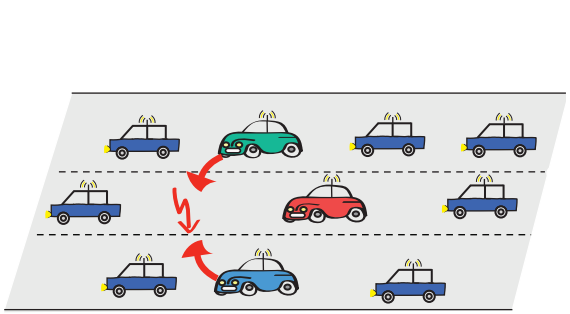


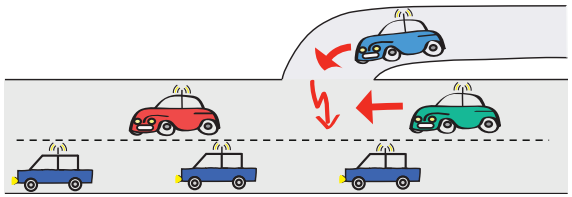


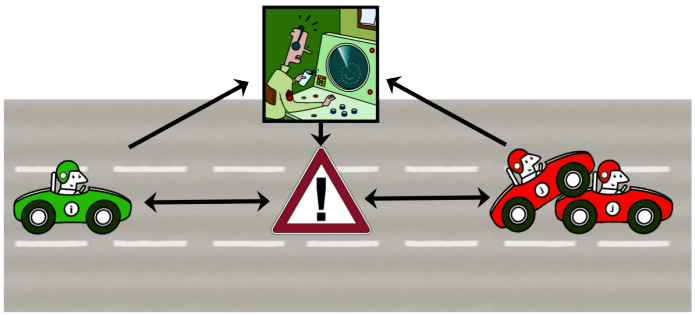


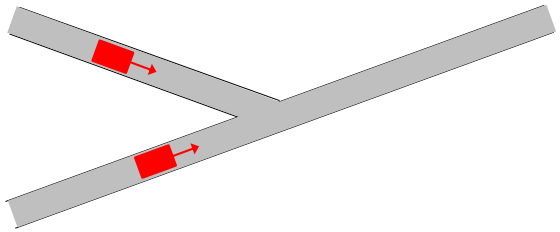
ICFEM'09,JAIS'14,CAV'08,FM'09,HSCC'11,HSCC'13

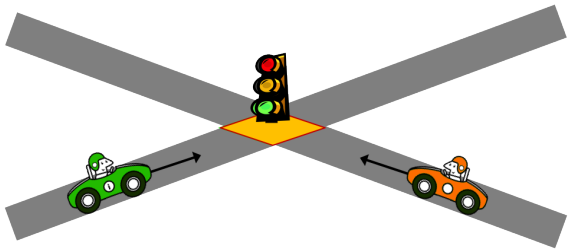
Successful CPS Proofs

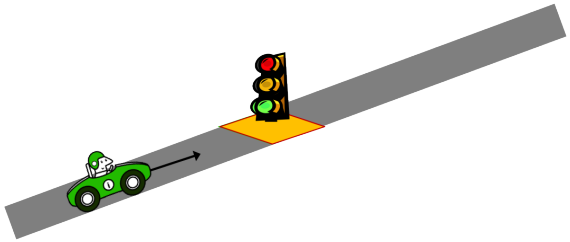


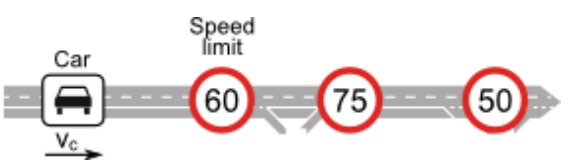


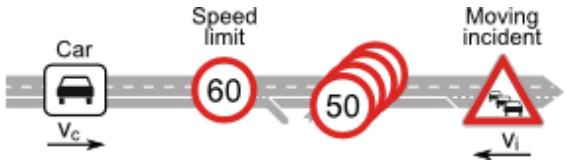


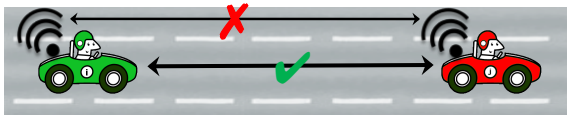


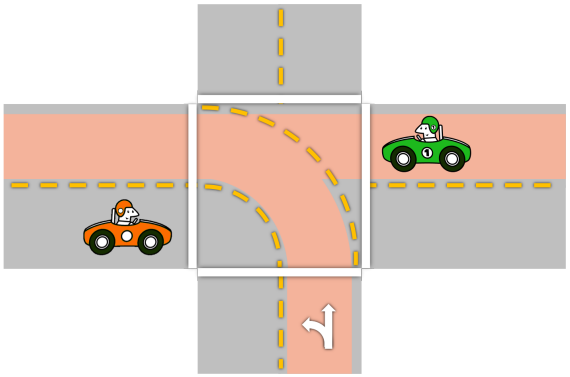


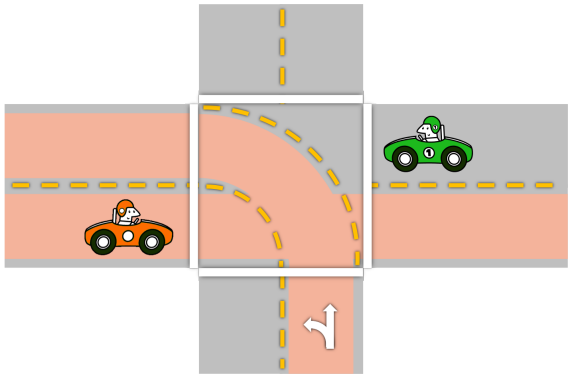


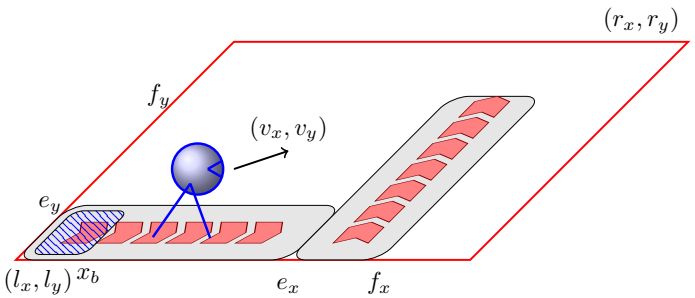






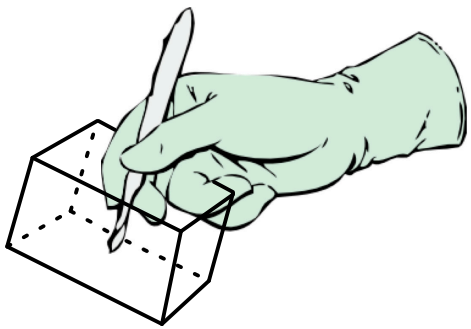


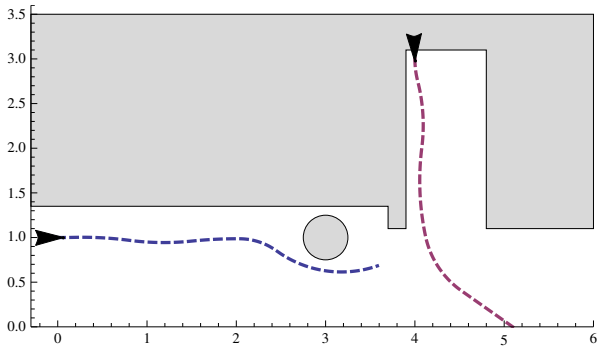


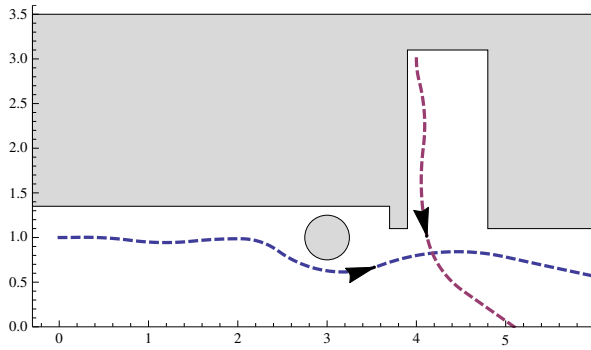


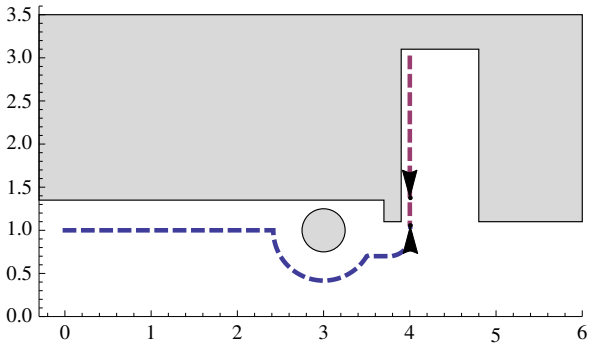
FM'11,LMCS'12,ICCPS'12,ITSC'11,ITSC'13,IJCAR'12

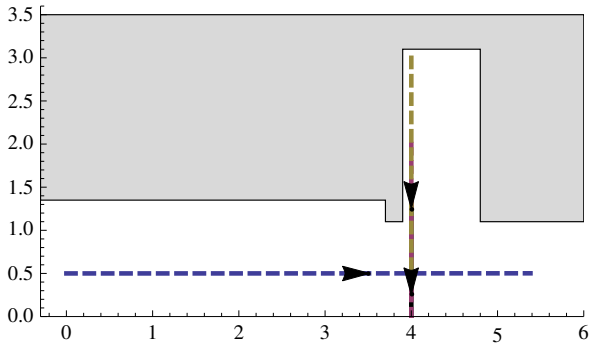
Successful CPS Proofs

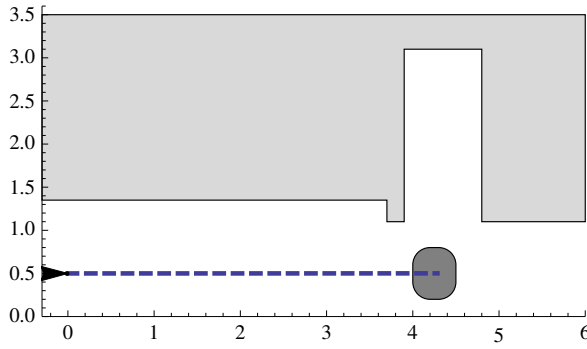


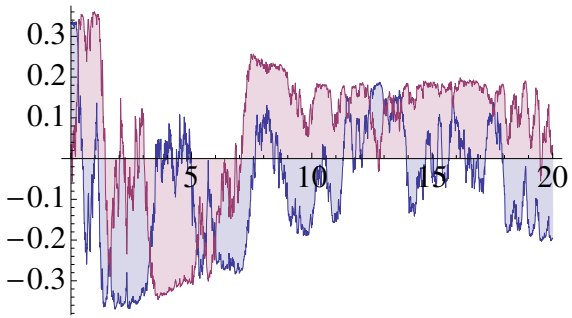


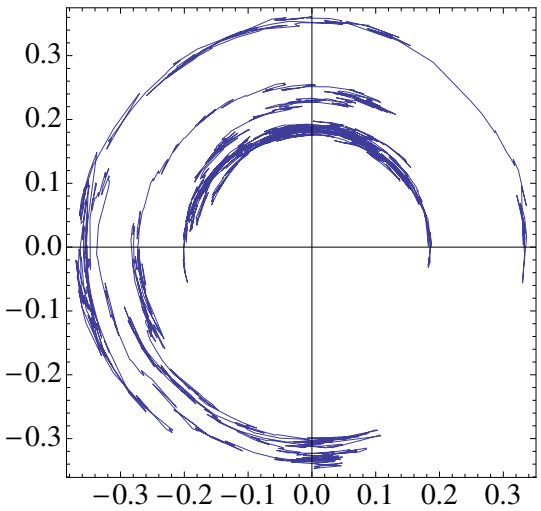


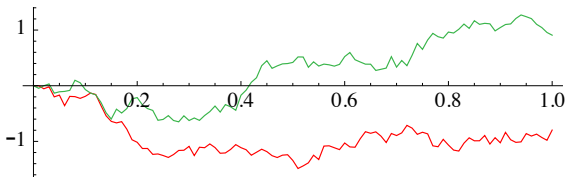






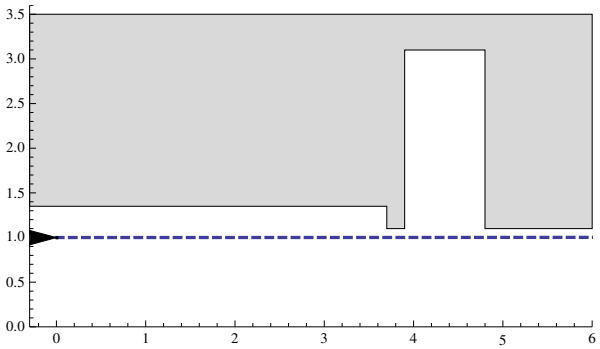
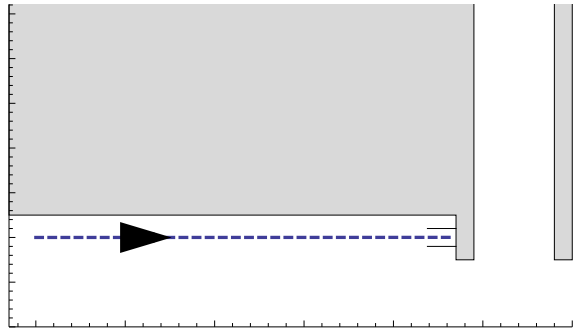
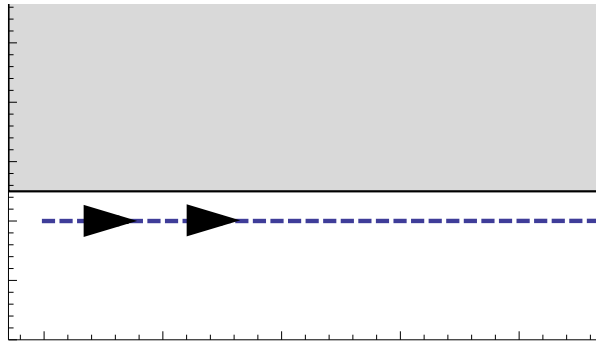
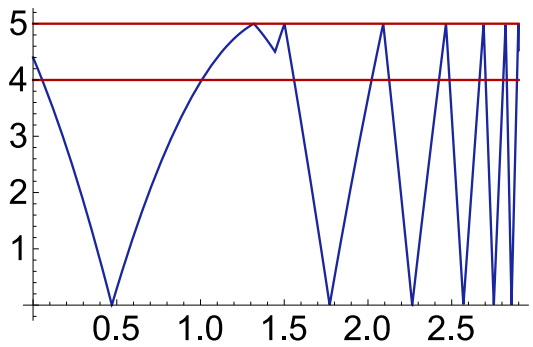
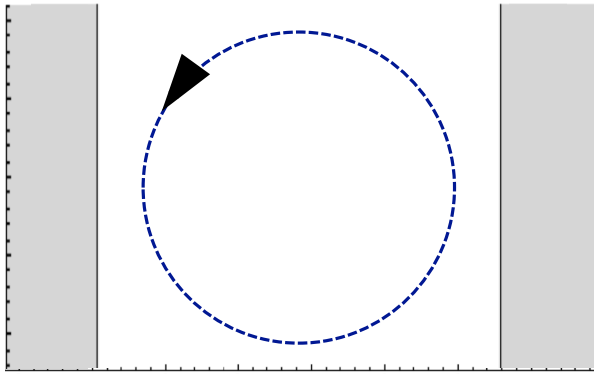
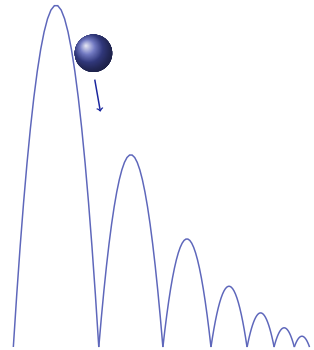
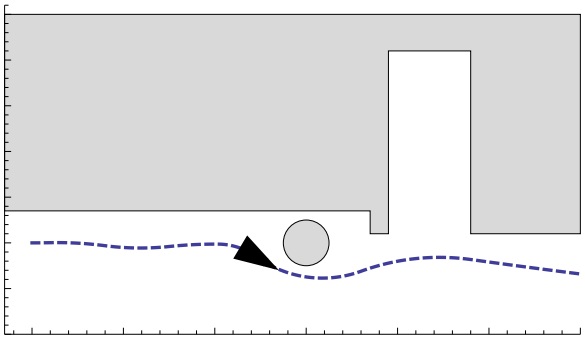
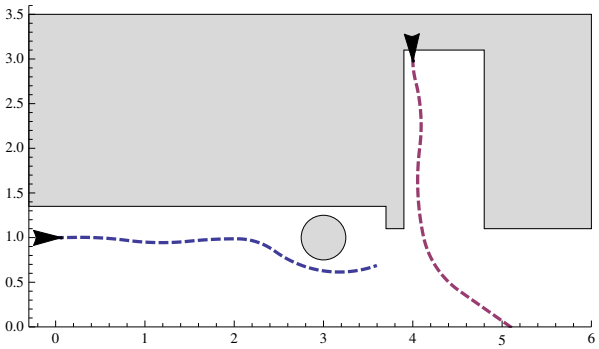
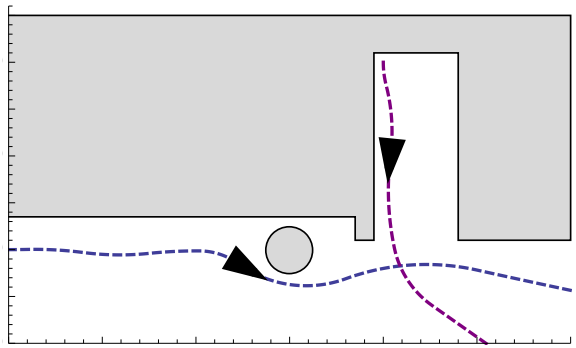






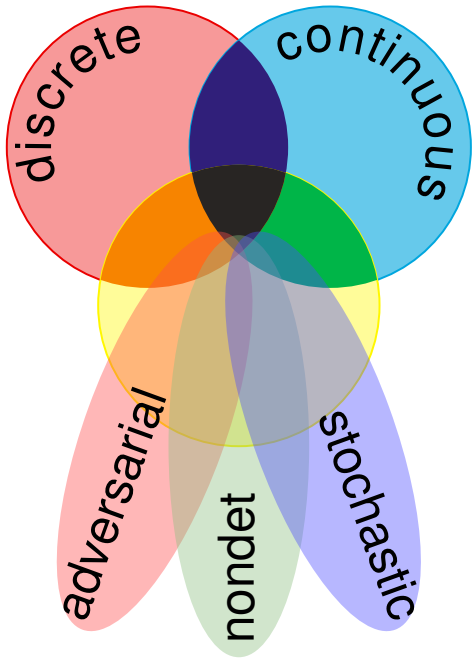
HSCC'13,RSS'13,CADE'12

Successful CPS Proofs By Undergrads

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

Family of Differential Dynamic Logics

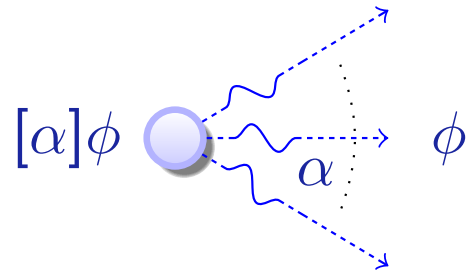


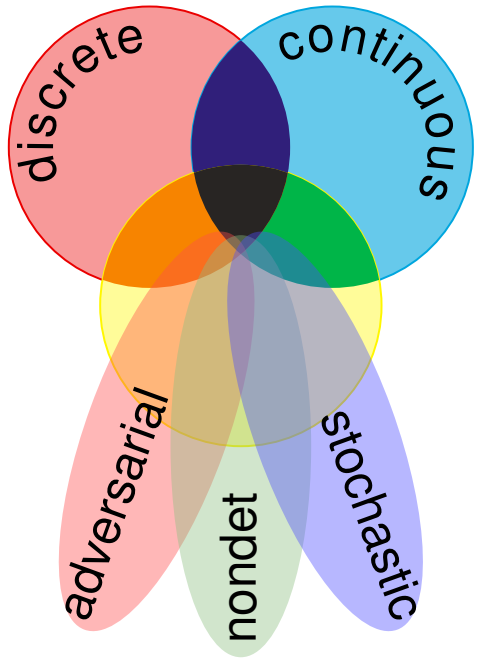
JAR'08,CADE'11,LMCS'12,LICS'12,LICS'12

Family of Differential Dynamic Logics

differential dynamic logic

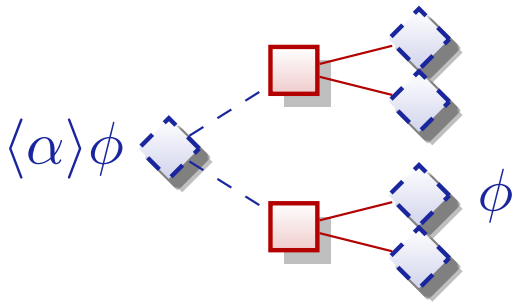
$$\mathsf{d}\mathcal{L} = \mathsf{DL} + \mathsf{HP}$$





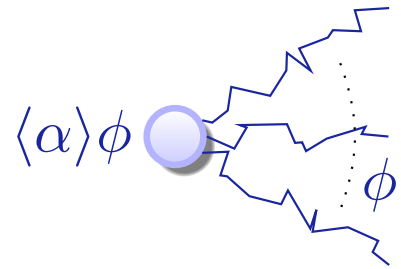
differential game logic

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



stochastic differential DL

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



quantified differential DL

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

JAR'08,CADE'11,LMCS'12,LICS'12,LICS'12

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

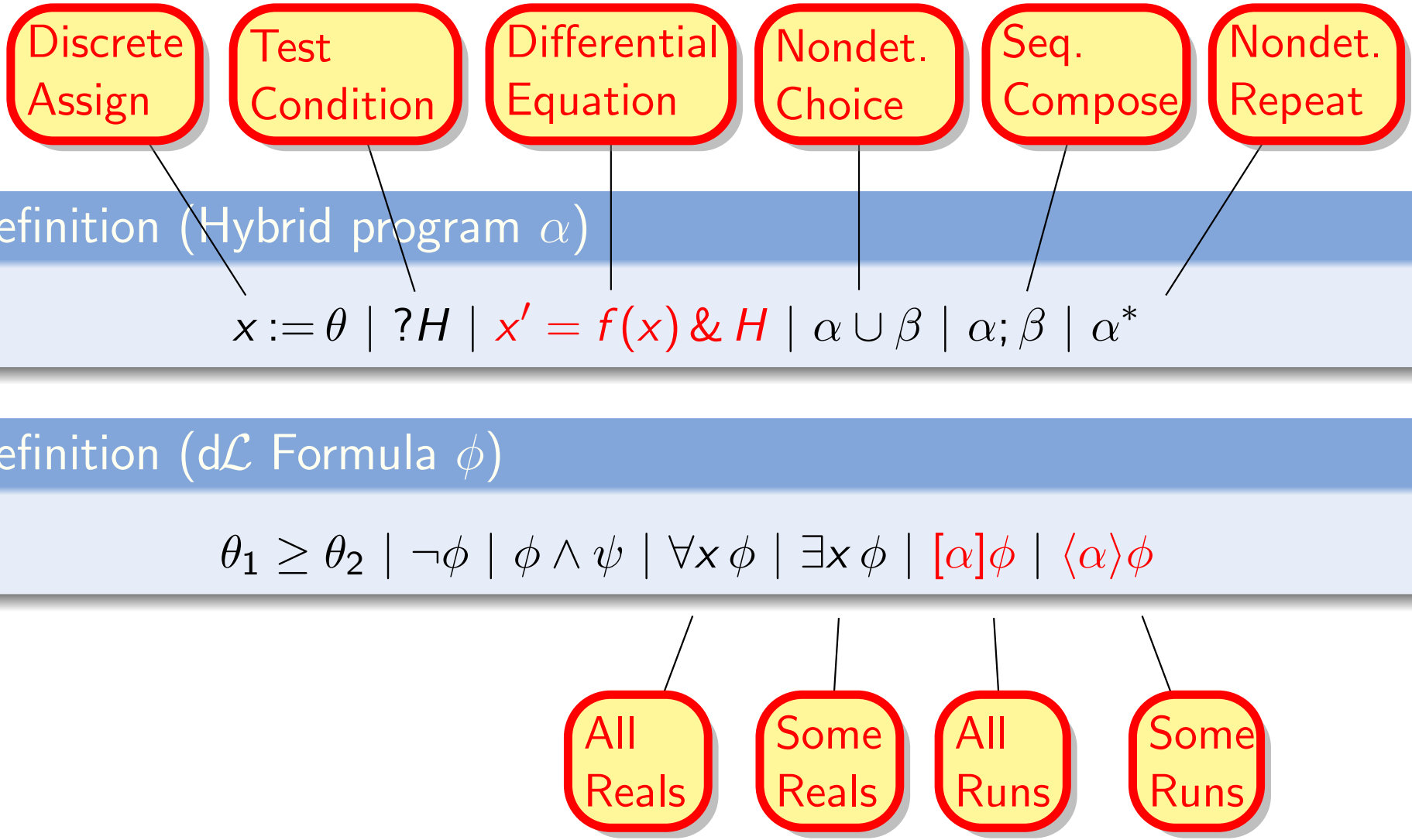
Definition (Hybrid program α)

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

Definition (d$\mathcal{L}$ Formula ϕ)

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

Differential Dynamic Logic dℒ: Syntax



Differential Dynamic Logic dℒ: Semantics

Definition (Hybrid program α) (Reachability Semantics)

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

Definition (dℒ Formula ϕ) (Modal Semantics)

$v \models \phi \wedge \psi$ iff $v \models \phi$ and $v \models \psi$

$v \models [\alpha]\phi$ iff $w \models \phi$ for all w with $(v,w) \in \rho(\alpha)$

$v \models \langle\alpha\rangle\phi$ iff $w \models \phi$ for some w with $(v,w) \in \rho(\alpha)$

$v \models \forall x\, \phi$ iff $w \models \phi$ for all w that agree with v except for x

$v \models \exists x\, \phi$ iff $w \models \phi$ for some w that agrees with v except for x

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
   - Differential Radical Invariants
5. Applications
6. Summary

Differential Dynamic Logic: Axiomatization

equations of truth

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

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

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

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

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

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

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

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

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

LICS'12

Differential Dynamic Logic: Axiomatization

equations of truth

$$\mathsf{G}\ \frac{\phi}{[\alpha]\phi}$$

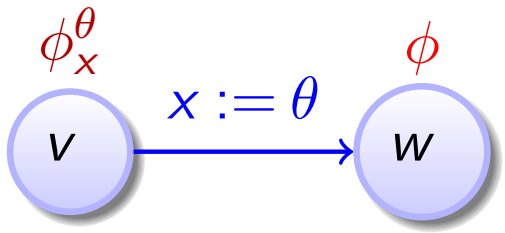
$$\mathsf{MP}\ \frac{\phi \to \psi \quad \phi}{\psi}$$

$$\forall\ \frac{\phi}{\forall x\, \phi}$$

LICS'12

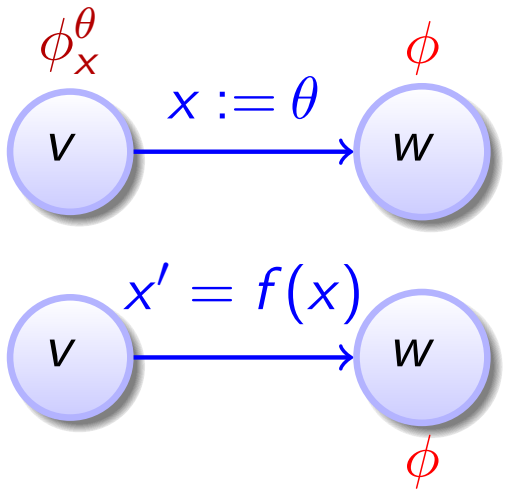
Proofs for Hybrid Systems

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



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

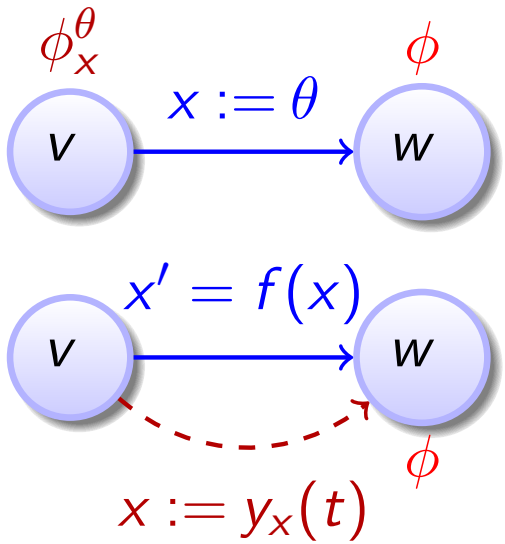
$$\frac{\forall t{\geq}0\,[x := y_x(t)]\phi}{[x' = f(x)]\phi}$$



Proofs for Hybrid Systems

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

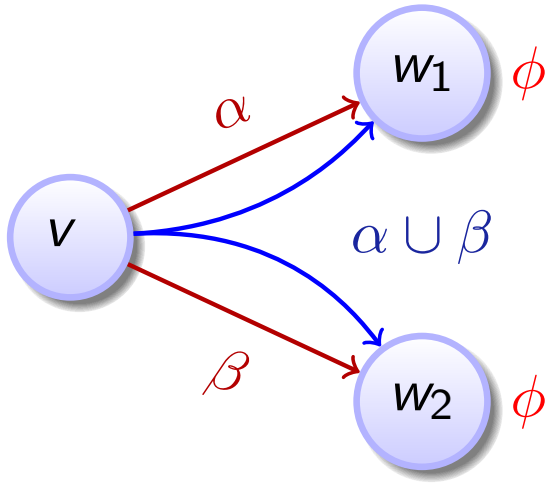
$$\frac{\forall t{\geq}0\,[x := y_x(t)]\phi}{[x' = f(x)]\phi}$$



Proofs for Hybrid Systems

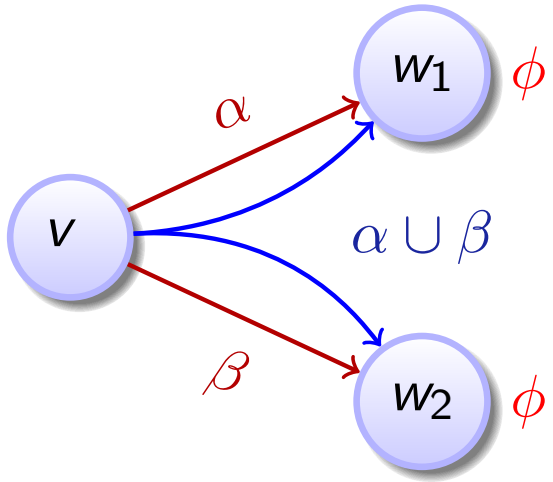
compositional semantics $\Rightarrow$ compositional rules!

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

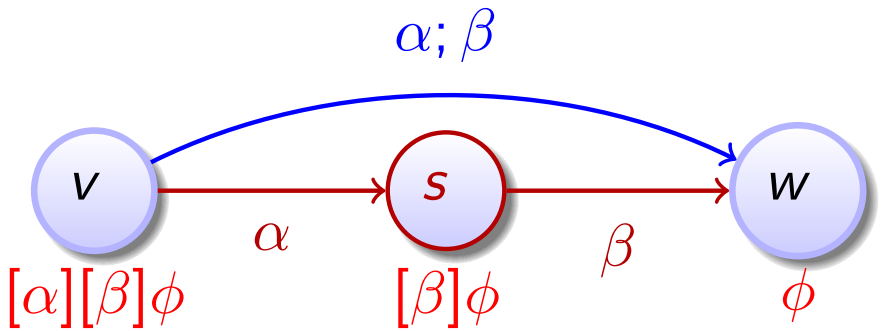


Proofs for Hybrid Systems

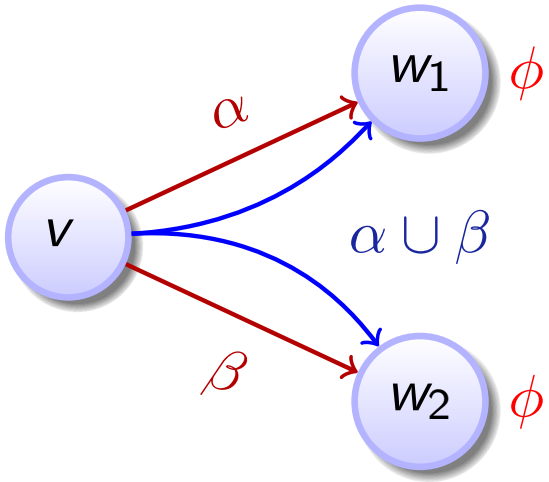
$$\frac{[\alpha]\phi \wedge [\beta]\phi}{[\alpha \cup \beta]\phi}$$



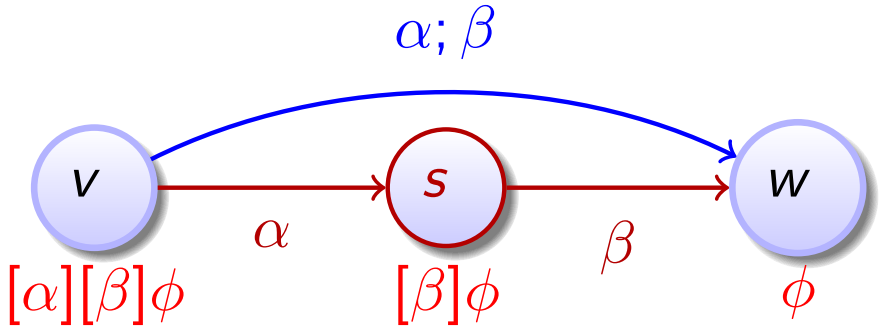
$$\frac{[\alpha][\beta]\phi}{[\alpha;\beta]\phi}$$



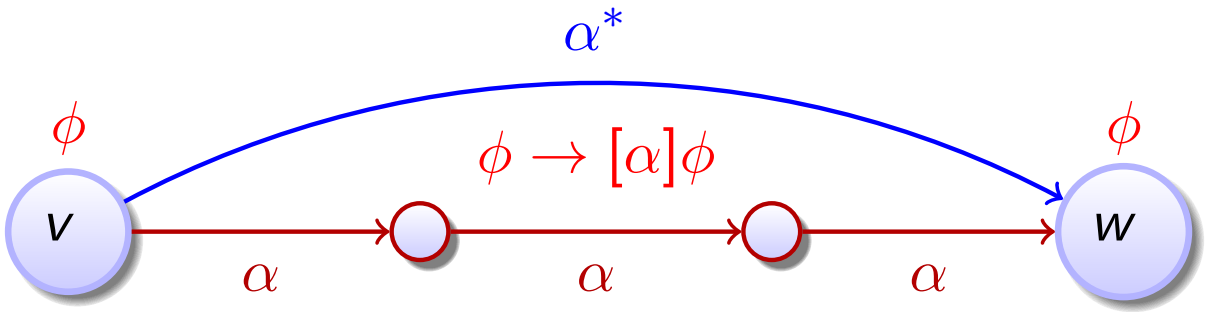
$$\frac{[\alpha]\phi \wedge [\beta]\phi}{[\alpha \cup \beta]\phi}$$



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



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



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
   - Differential Radical Invariants
5. Applications
6. Summary

Complete Proof Theory of Hybrid Systems

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

$d\mathcal{L}$ 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

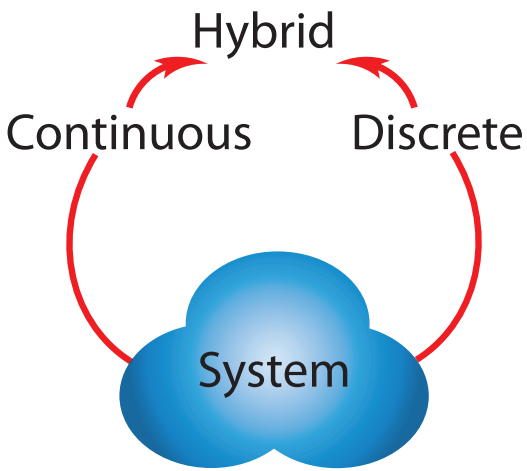
Complete Proof Theory of Hybrid Systems

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

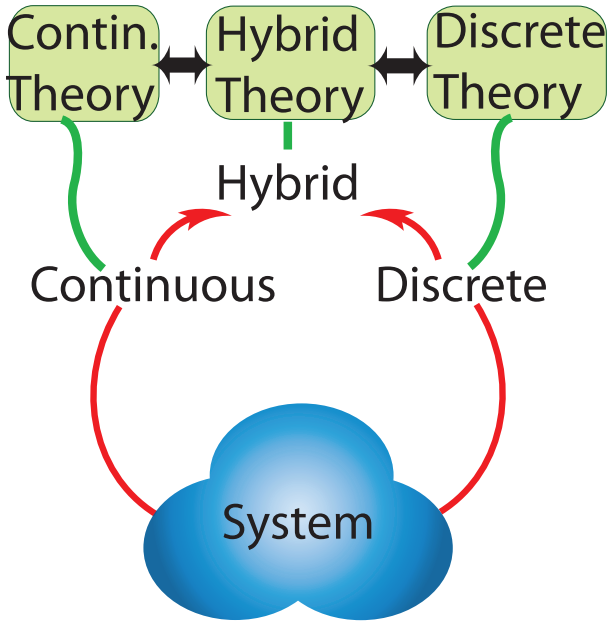
Complete Proof Theory of Hybrid Systems

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

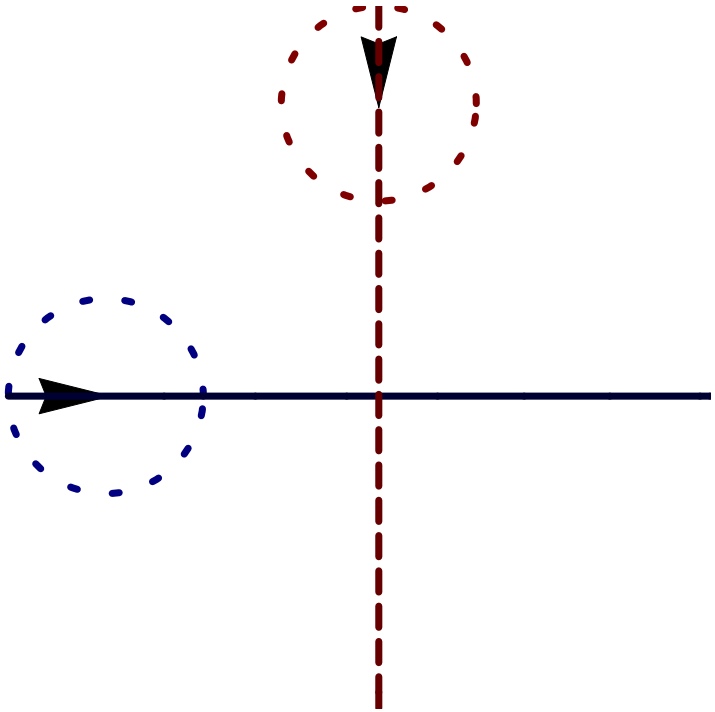
Corollary (Complete Proof-theoretical Alignment & Bridging)

proving continuous = proving hybrid = proving discrete

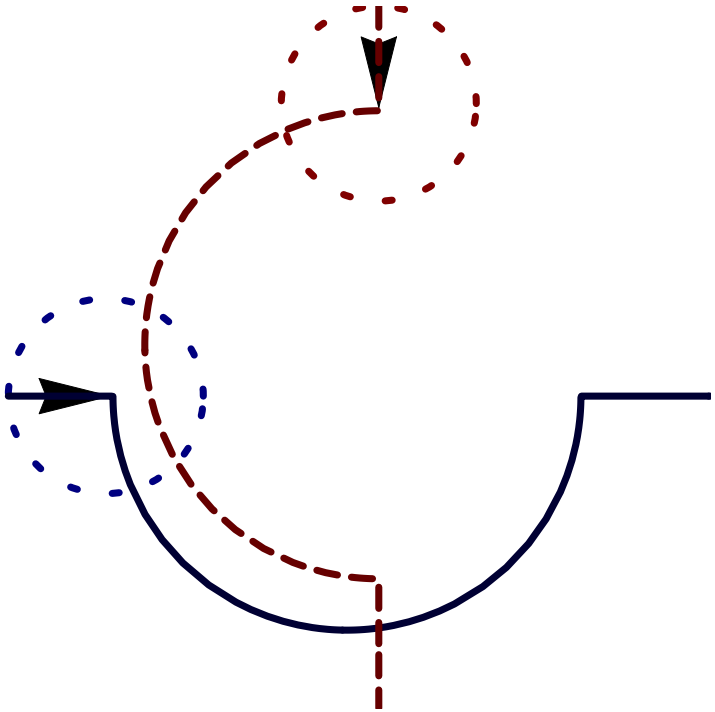


JAutomReas'08,LICS'12

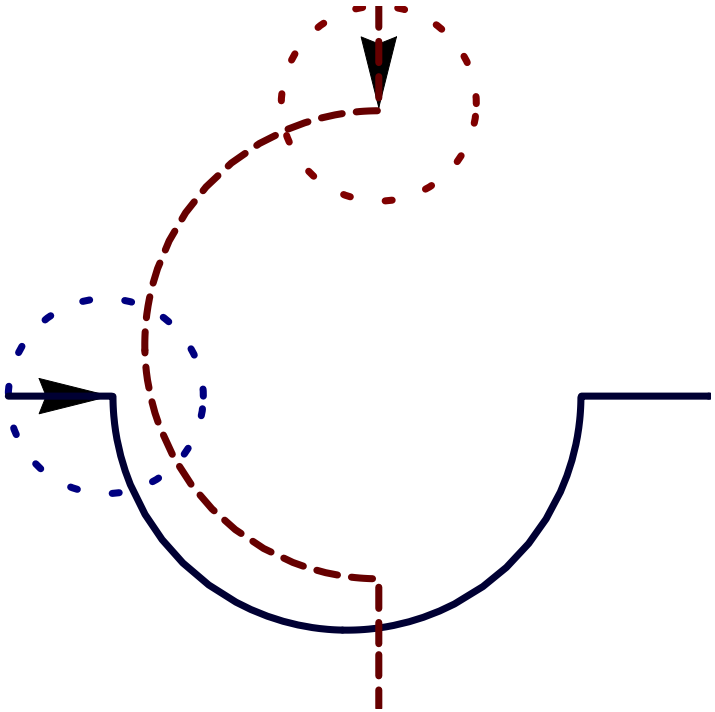
Air Traffic Control



Air Traffic Control



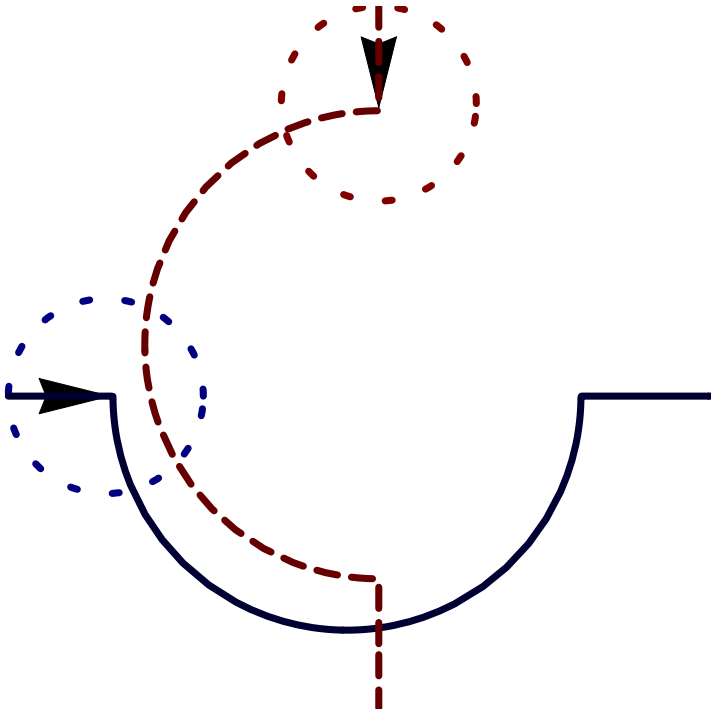
Air Traffic Control

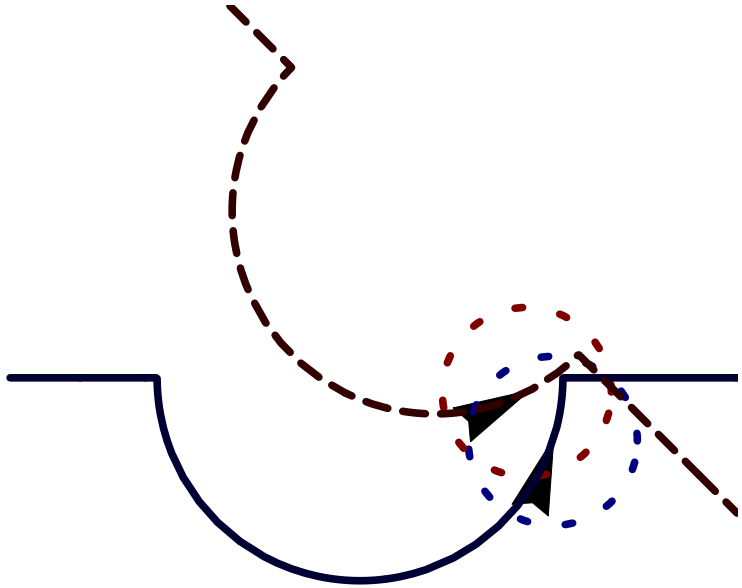


Verification?

looks correct

Air Traffic Control

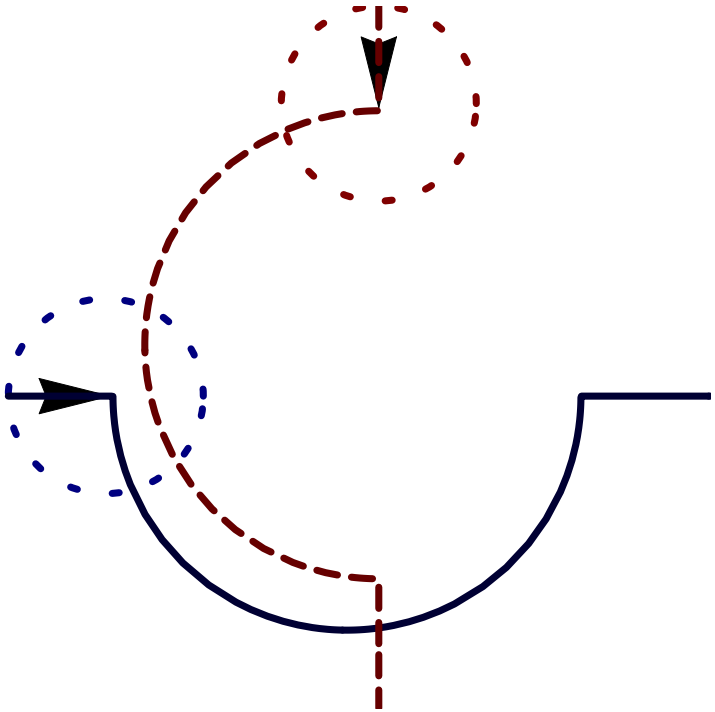


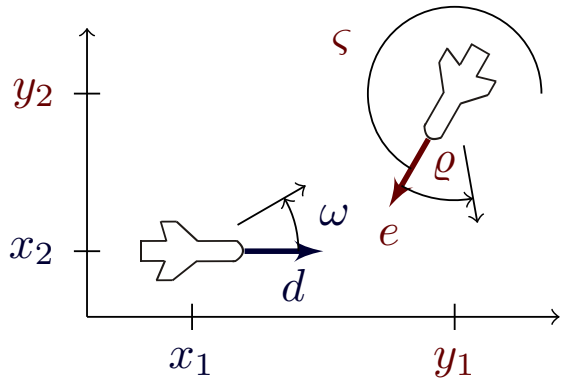


Verification?

looks correct NO!

Air Traffic Control



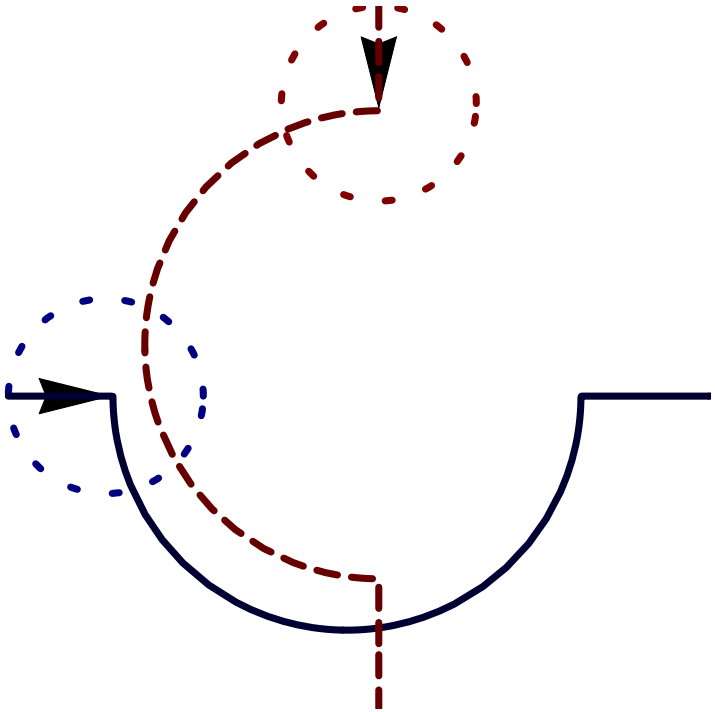


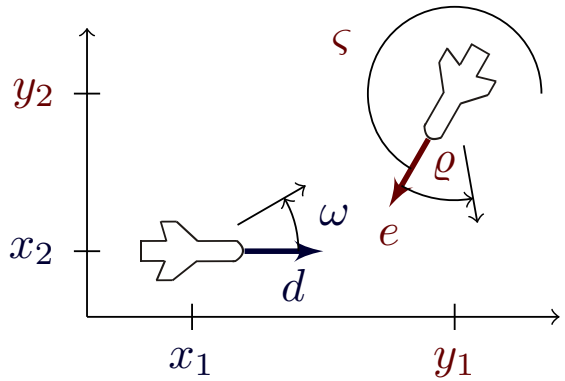
$$\begin{bmatrix} x_1' = -v_1 + v_2 \cos\vartheta + \omega x_2 \\ x_2' = v_2 \sin\vartheta - \omega x_1 \\ \vartheta' = \varpi - \omega \end{bmatrix}$$

Verification?

looks correct NO!

Air Traffic Control



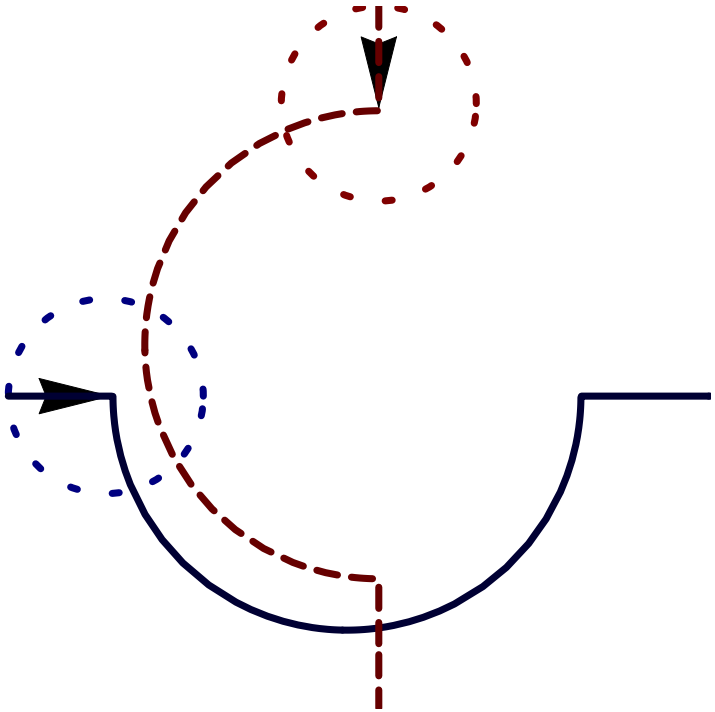


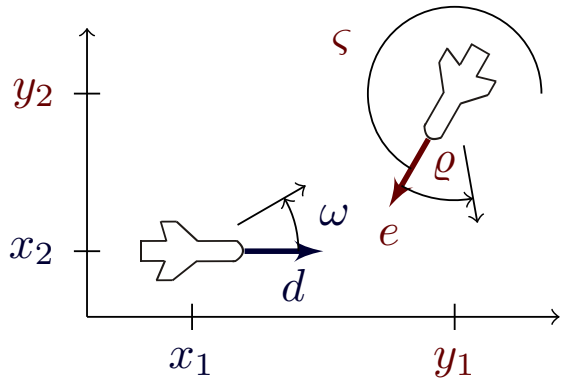
$$\begin{bmatrix} x_1' = -v_1 + v_2 \cos\vartheta + \omega x_2 \\ x_2' = v_2 \sin\vartheta - \omega x_1 \\ \vartheta' = \varpi - \omega \end{bmatrix}$$

Example ("Solving" differential equations)

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

Air Traffic Control





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

Example ("Solving" differential equations)

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

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

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

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

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

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

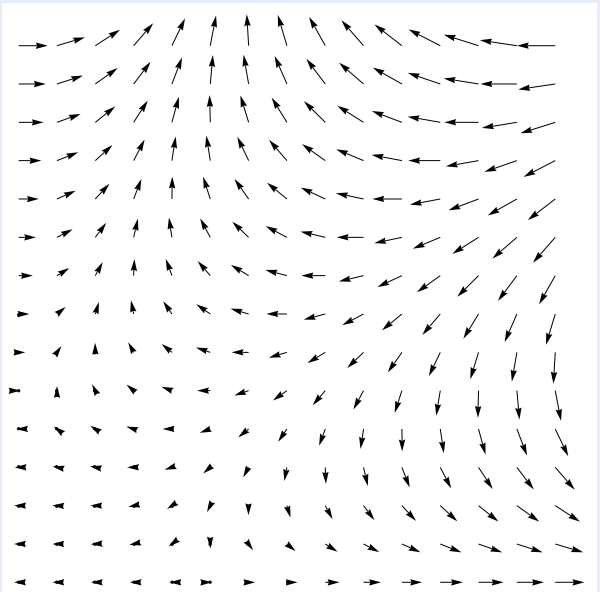
```
+   -1
  * om_3
  * (  (om_1)^-1
     * (omb_1)^-1
     * (  om_1 * omb_1 * x1 * Cos(om_1 * t2_0)
        +   om_1
          * v2
          * Cos(om_1 * t2_0)
          * (1 + -1 * (Cos(u))^2)^(1 / 2)
        + -1 * omb_1 * v1 * Sin(om_1 * t2_0)
        + om_1 * omb_1 * x2 * Sin(om_1 * t2_0)
   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

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

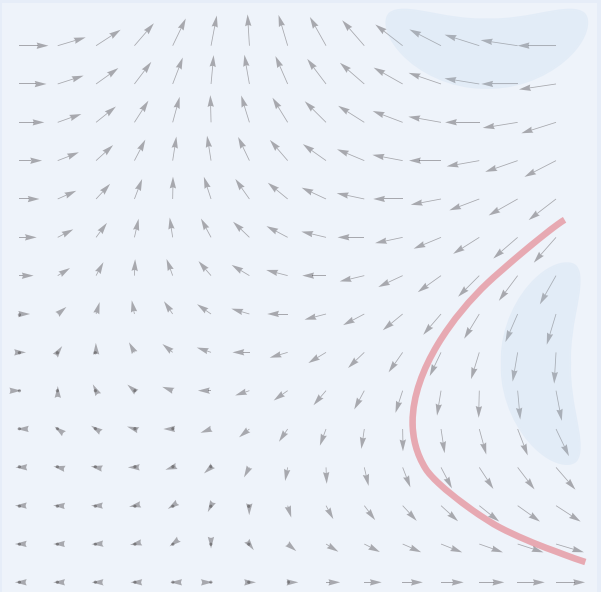
This is just one branch to prove for aircraft . . .

Differential Invariants for Differential Equations

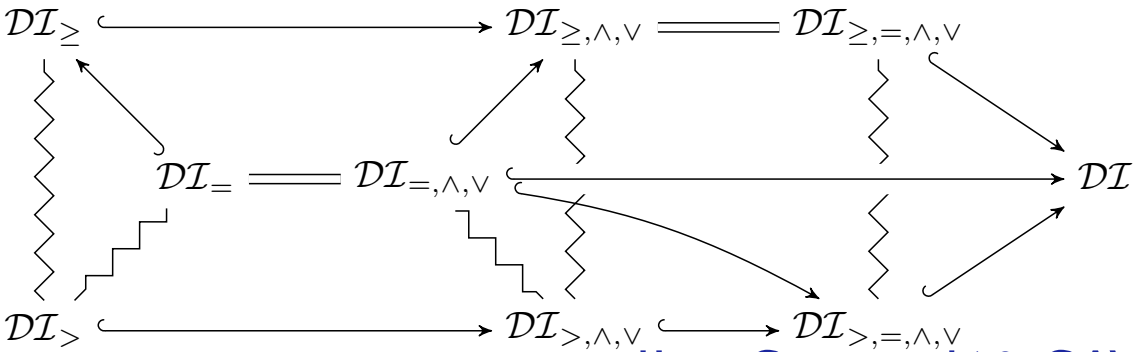
Differential Invariant



Differential Cut



Differential Ghost



Logic

Provability theory

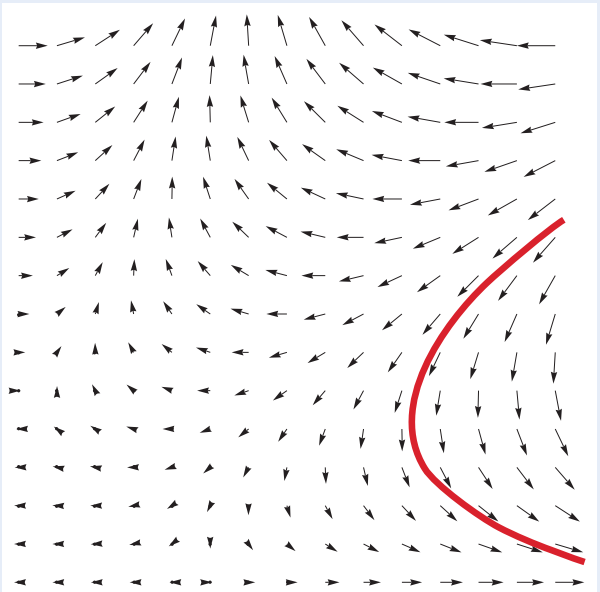
Math

Characteristic PDE

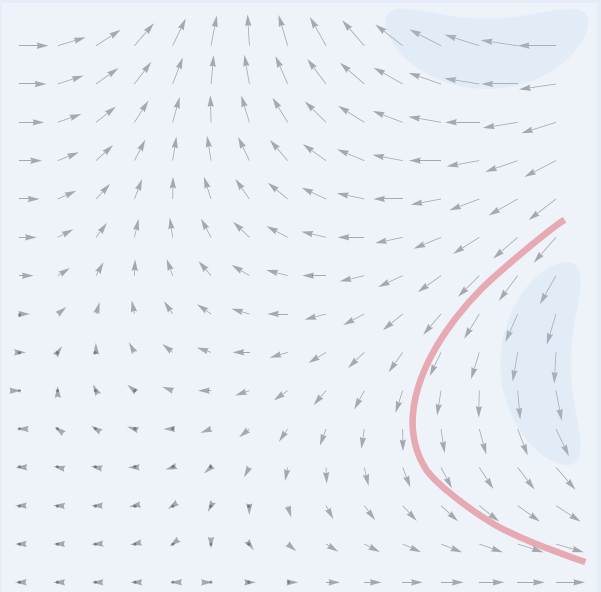
JLogComput'10,CAV'08,FMSD'09,LMCS'12,ITP'12

Differential Invariants for Differential Equations

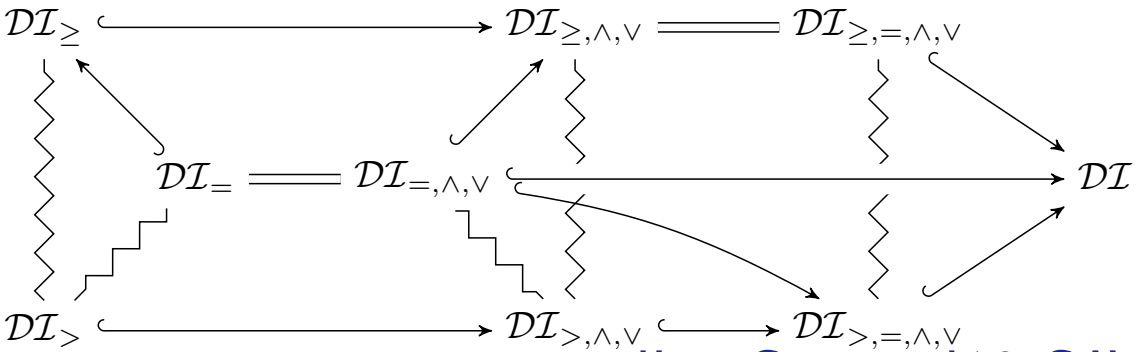
Differential Invariant



Differential Cut



Differential Ghost



Logic

Provability theory

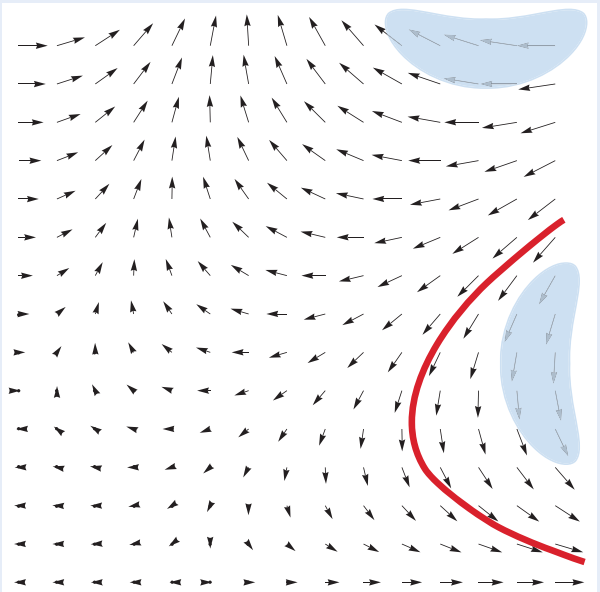
Math

Characteristic PDE

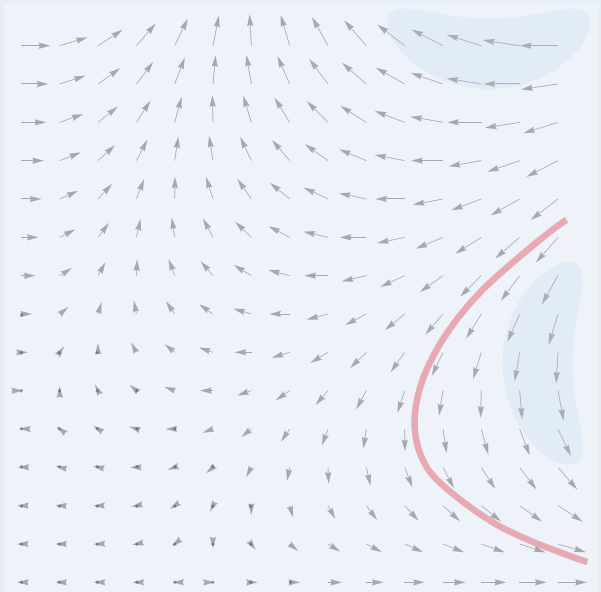
JLogComput'10,CAV'08,FMSD'09,LMCS'12,ITP'12

Differential Invariants for Differential Equations

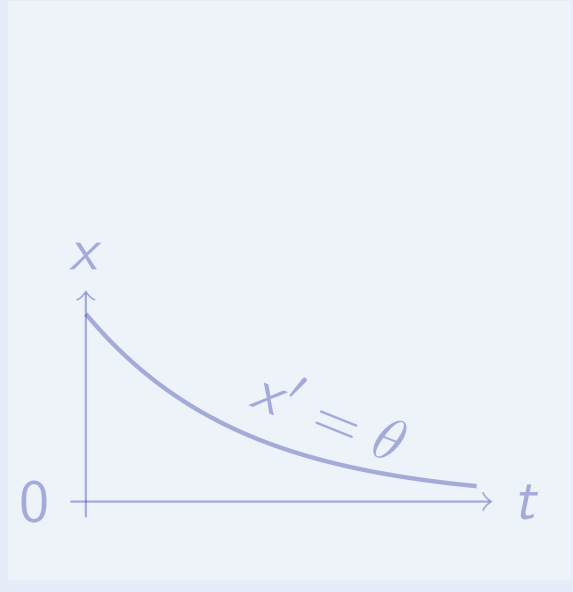
Differential Invariant

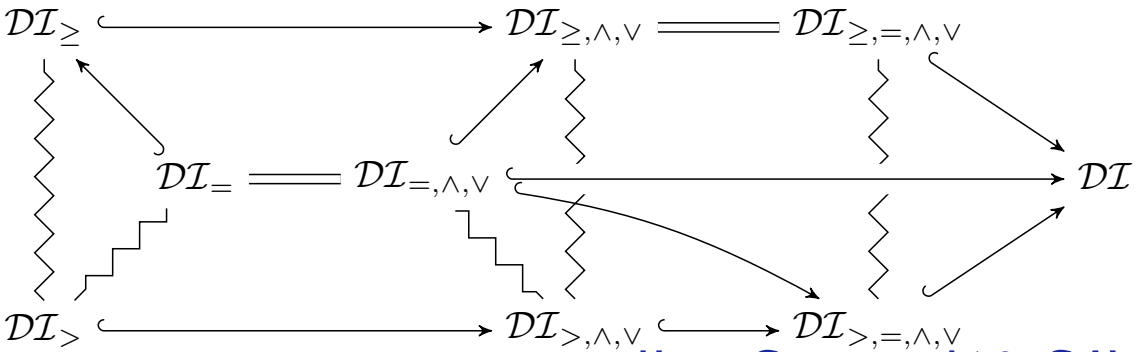


Differential Cut



Differential Ghost





Logic

Provability theory

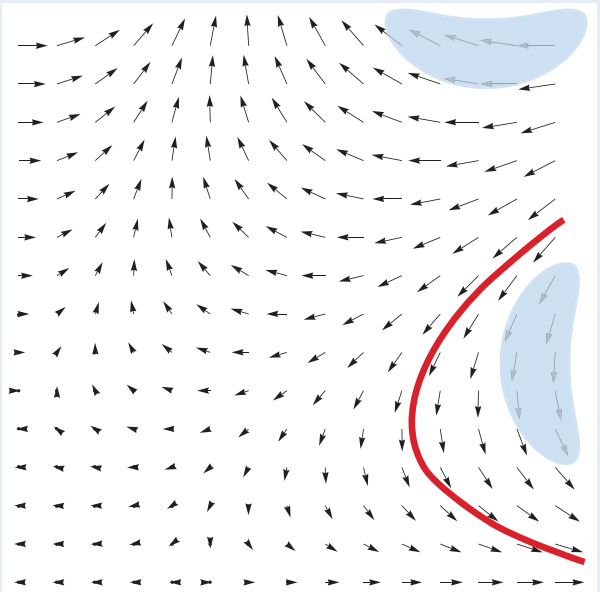
Math

Characteristic PDE

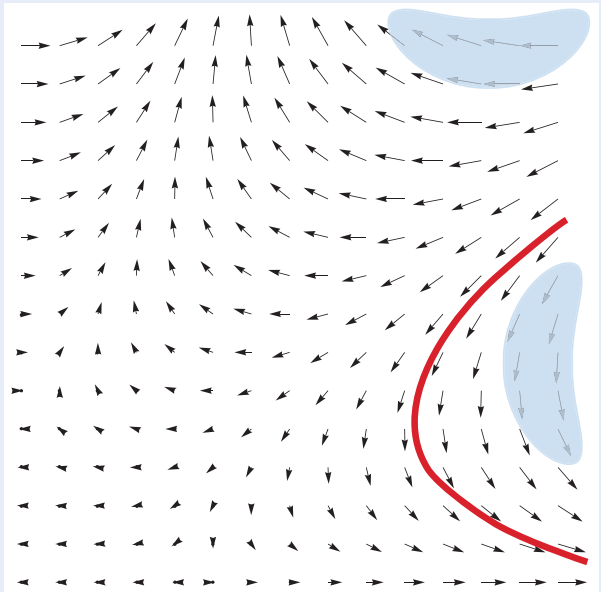
JLogComput'10,CAV'08,FMSD'09,LMCS'12,ITP'12

Differential Invariants for Differential Equations

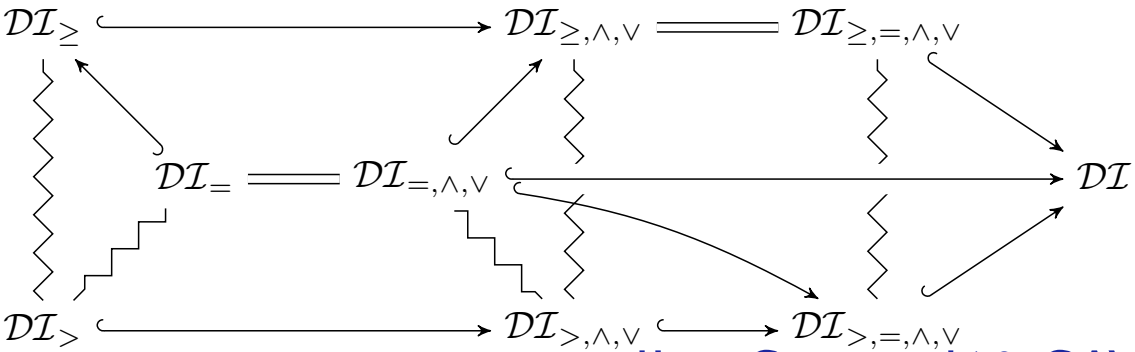
Differential Invariant



Differential Cut



Differential Ghost



Logic

Provability theory

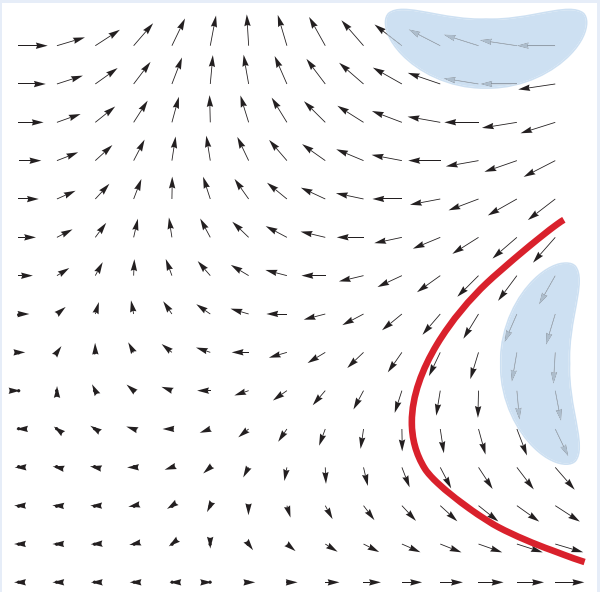
Math

Characteristic PDE

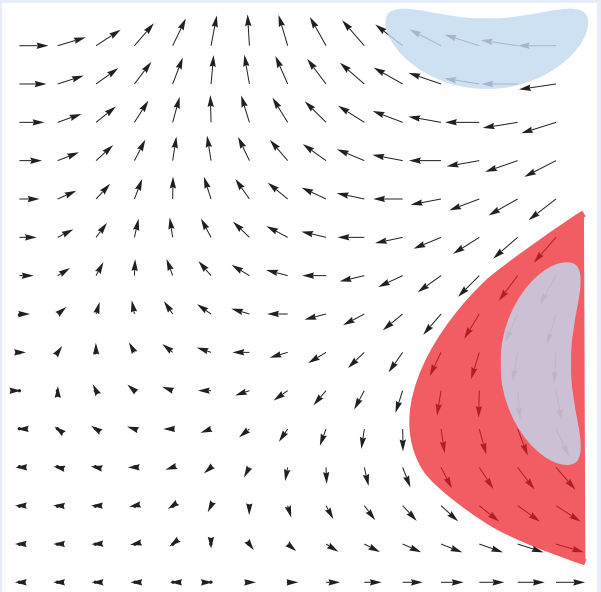
JLogComput'10,CAV'08,FMSD'09,LMCS'12,ITP'12

Differential Invariants for Differential Equations

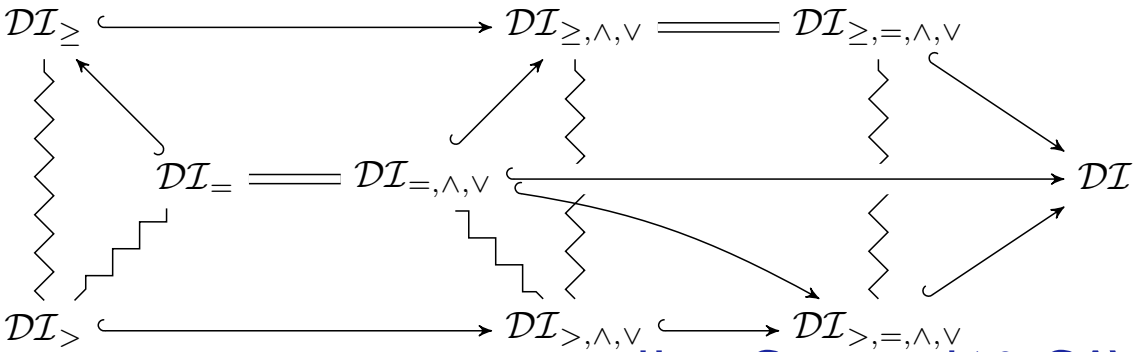
Differential Invariant



Differential Cut



Differential Ghost



Logic

Provability theory

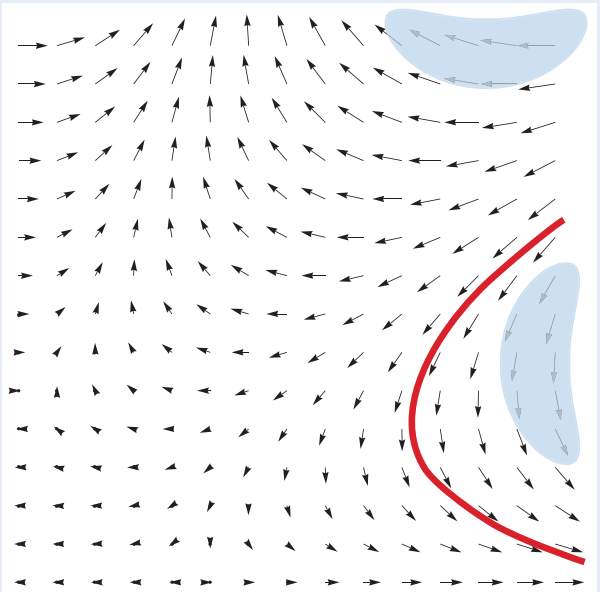
Math

Characteristic PDE

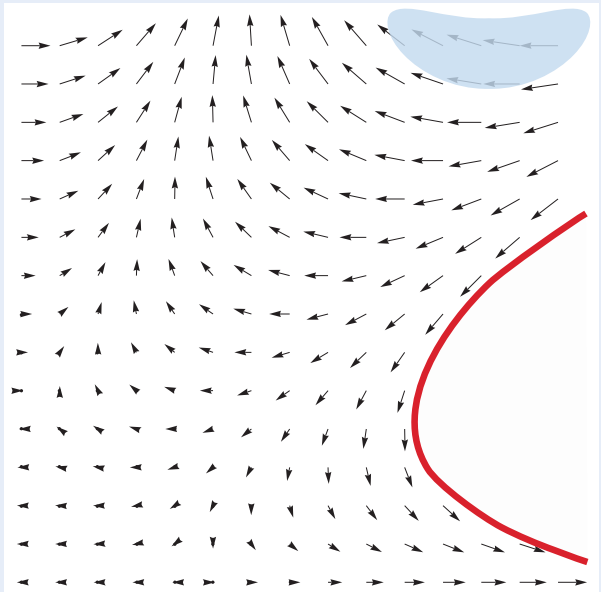
JLogComput'10,CAV'08,FMSD'09,LMCS'12,ITP'12

Differential Invariants for Differential Equations

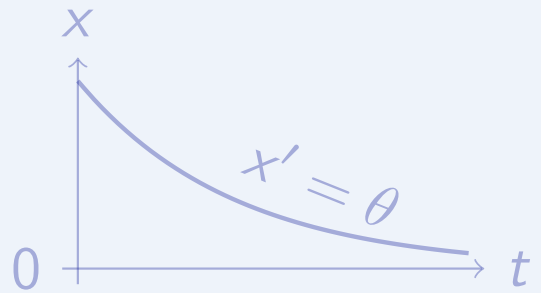
Differential Invariant

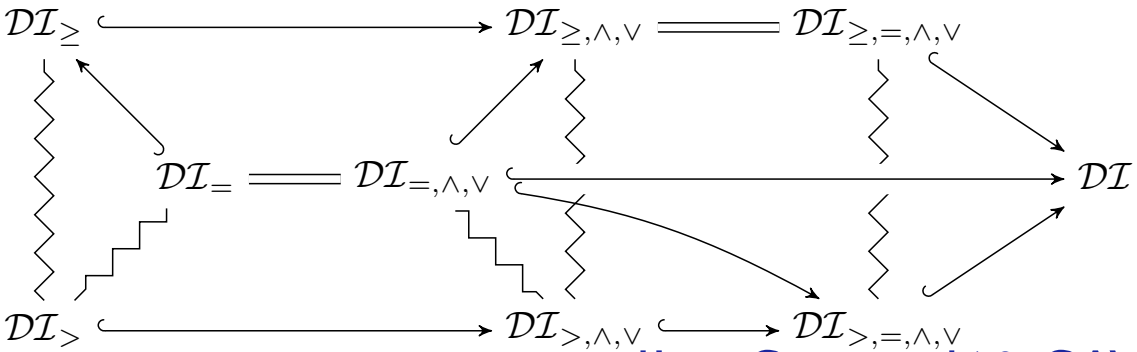


Differential Cut



Differential Ghost





Logic

Provability theory

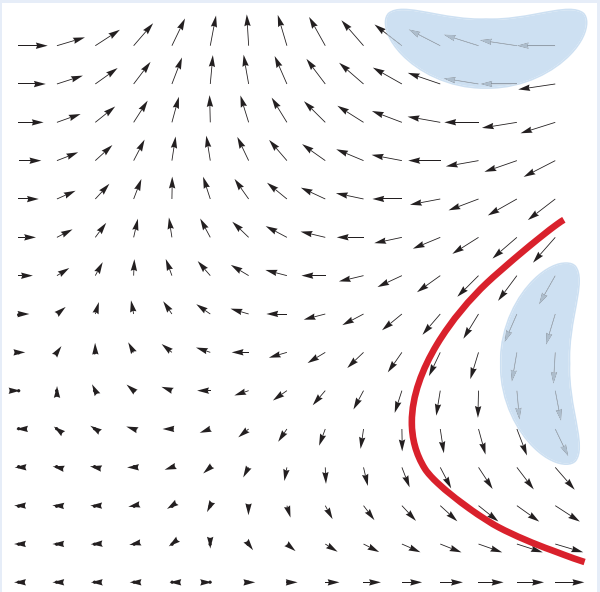
Math

Characteristic PDE

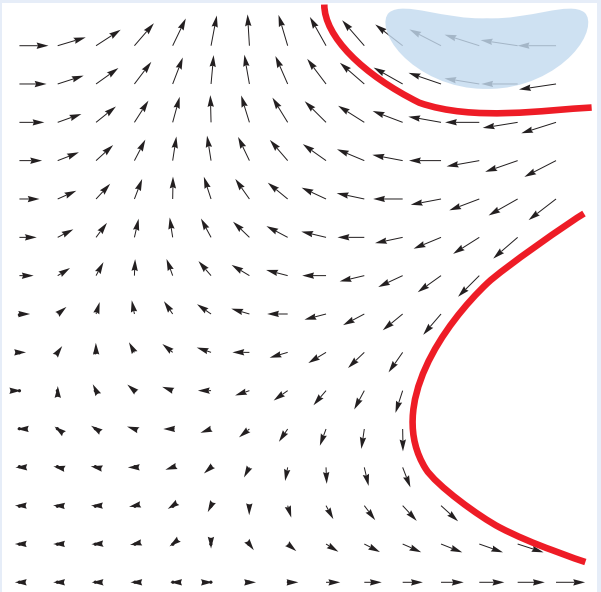
JLogComput'10,CAV'08,FMSD'09,LMCS'12,ITP'12

Differential Invariants for Differential Equations

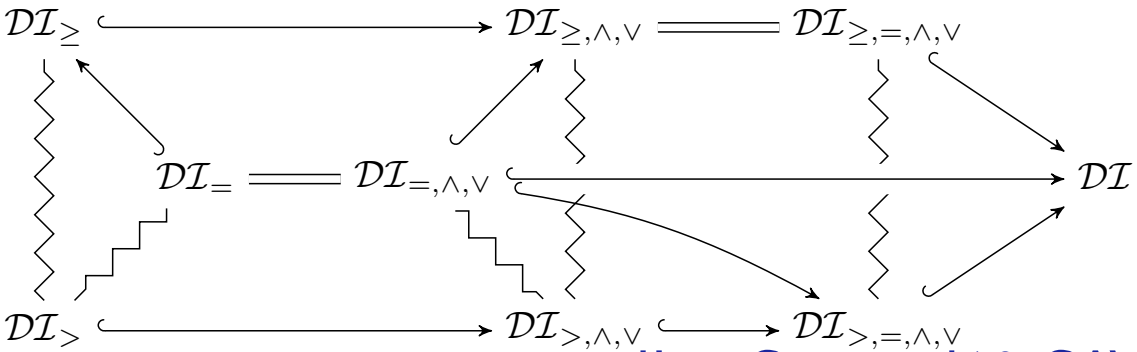
Differential Invariant



Differential Cut



Differential Ghost



Logic

Provability theory

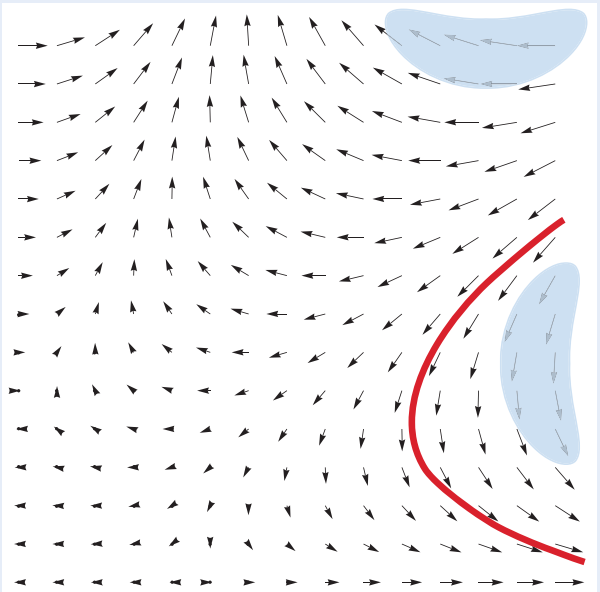
Math

Characteristic PDE

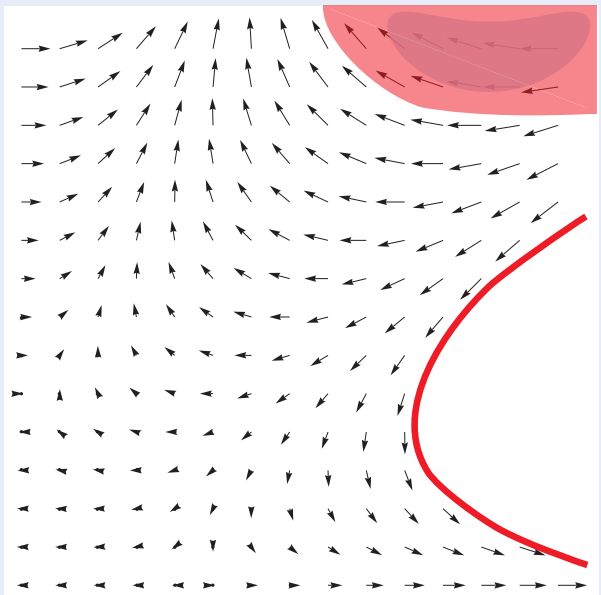
JLogComput'10,CAV'08,FMSD'09,LMCS'12,ITP'12

Differential Invariants for Differential Equations

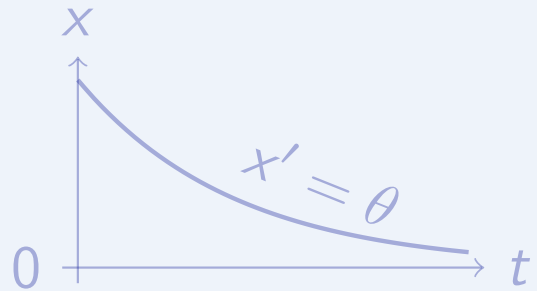
Differential Invariant

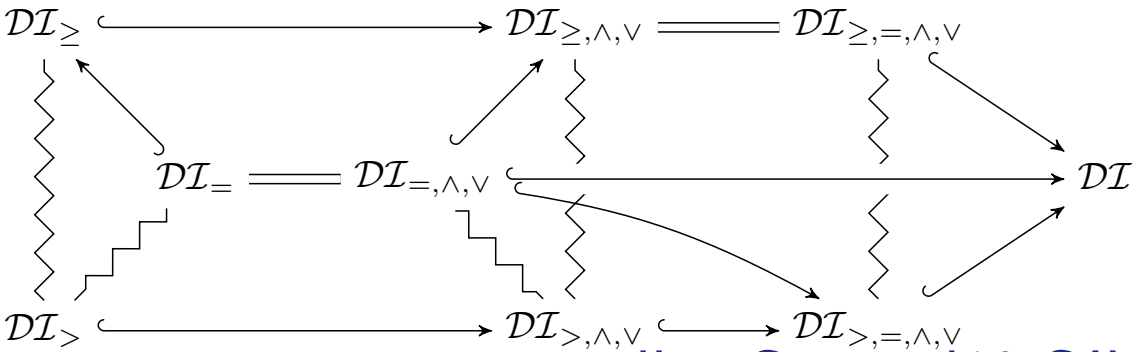


Differential Cut



Differential Ghost





Logic

Provability theory

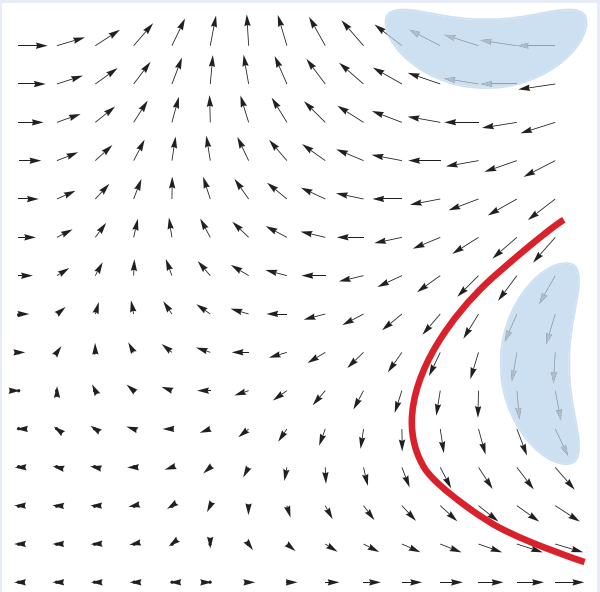
Math

Characteristic PDE

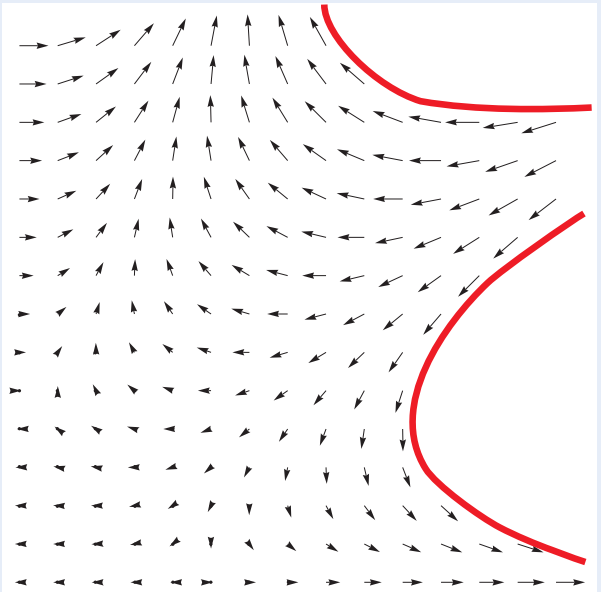
JLogComput'10,CAV'08,FMSD'09,LMCS'12,ITP'12

Differential Invariants for Differential Equations

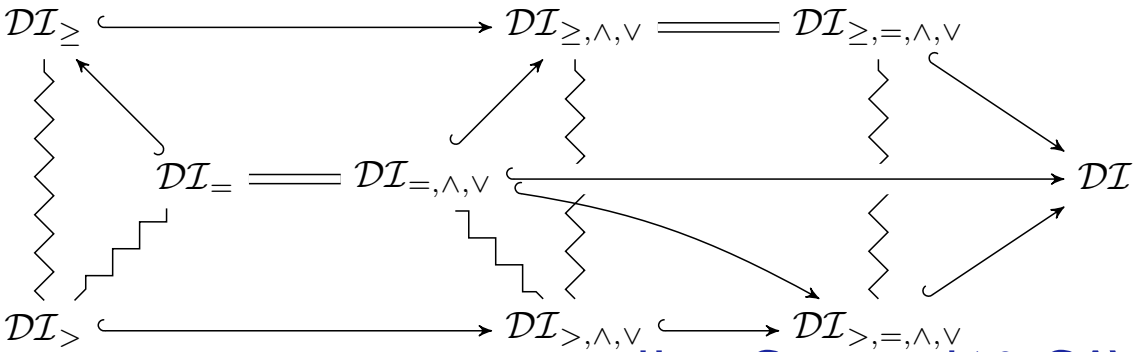
Differential Invariant



Differential Cut



Differential Ghost



Logic

Provability theory

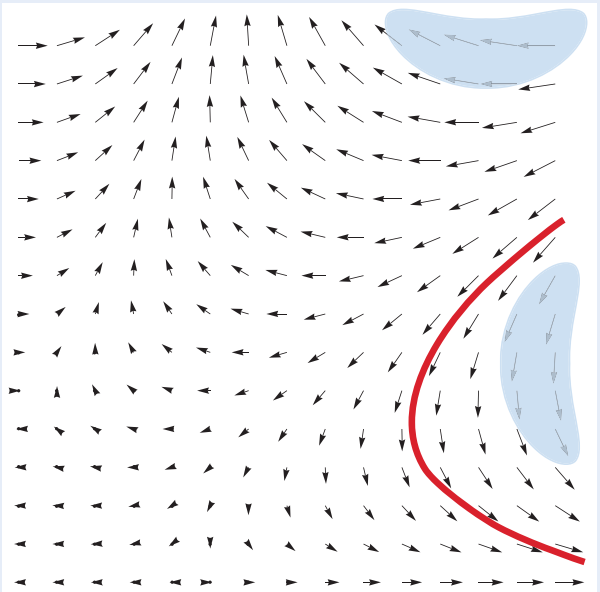
Math

Characteristic PDE

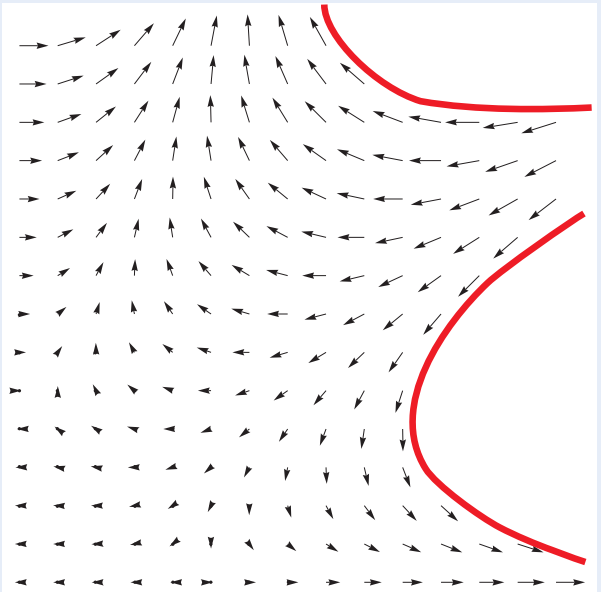
JLogComput'10,CAV'08,FMSD'09,LMCS'12,ITP'12

Differential Invariants for Differential Equations

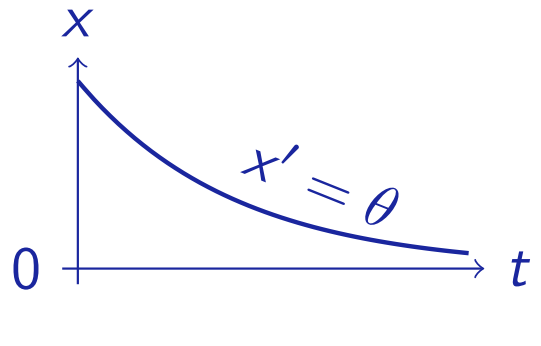
Differential Invariant

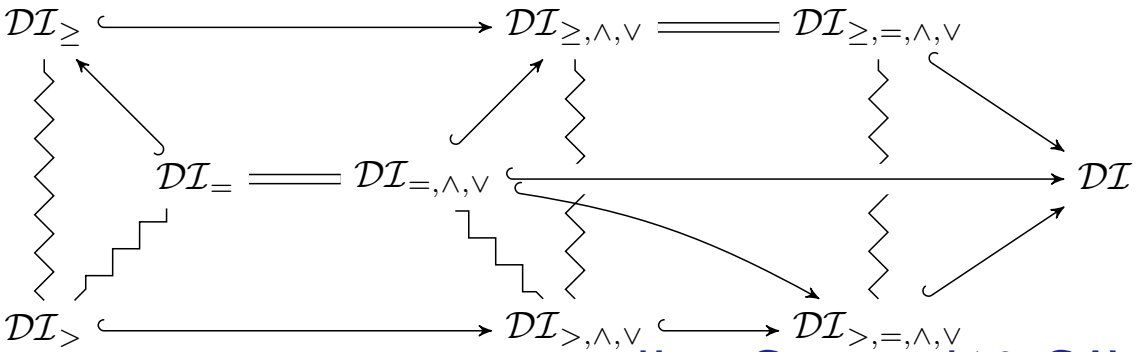


Differential Cut



Differential Ghost





Logic

Provability theory

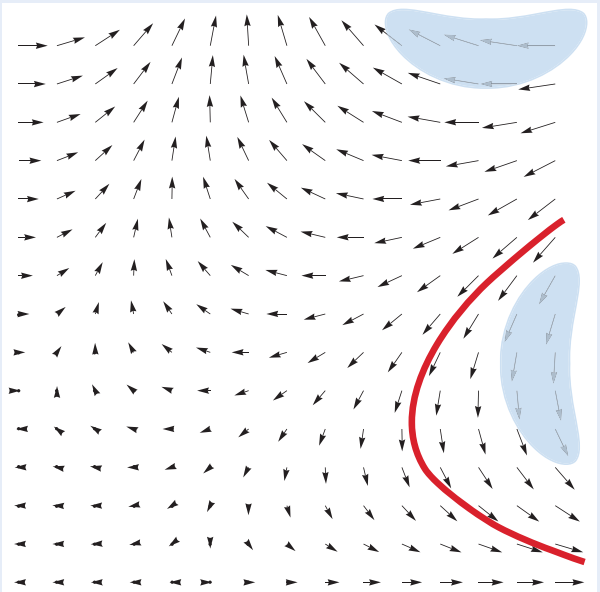
Math

Characteristic PDE

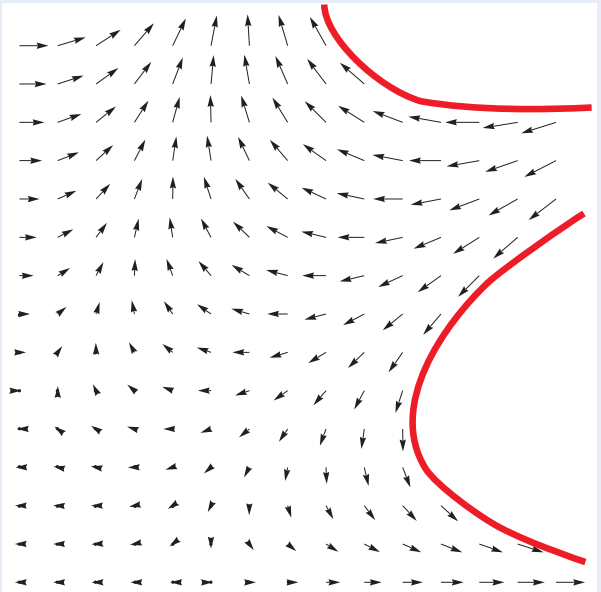
JLogComput'10,CAV'08,FMSD'09,LMCS'12,ITP'12

Differential Invariants for Differential Equations

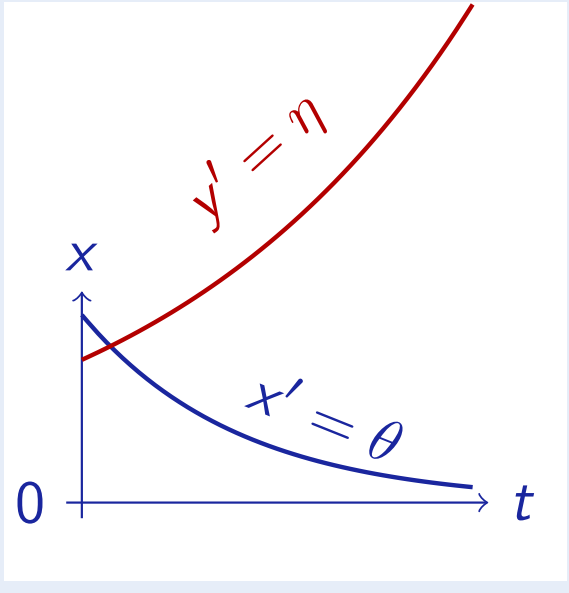
Differential Invariant

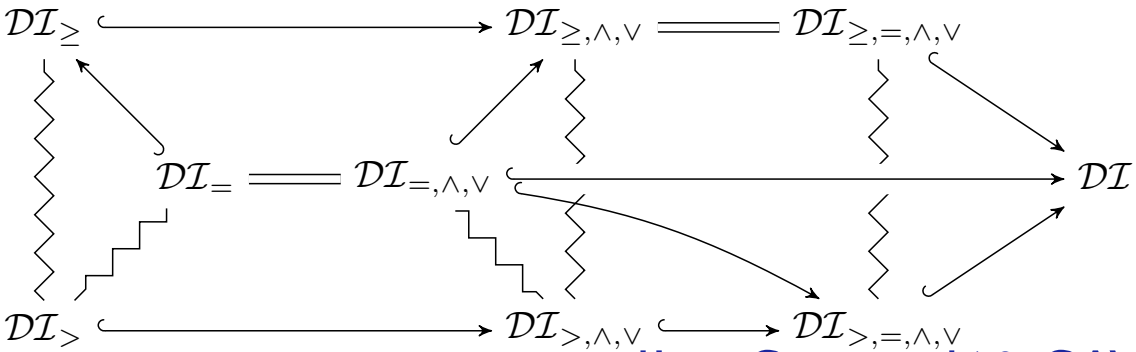


Differential Cut



Differential Ghost





Logic

Provability theory

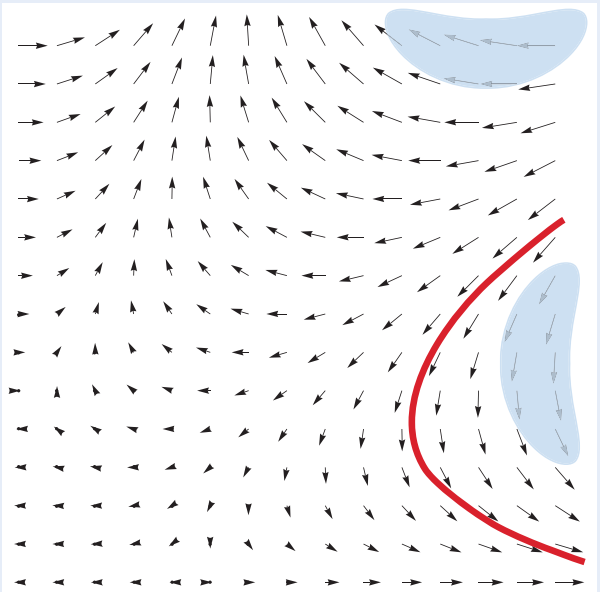
Math

Characteristic PDE

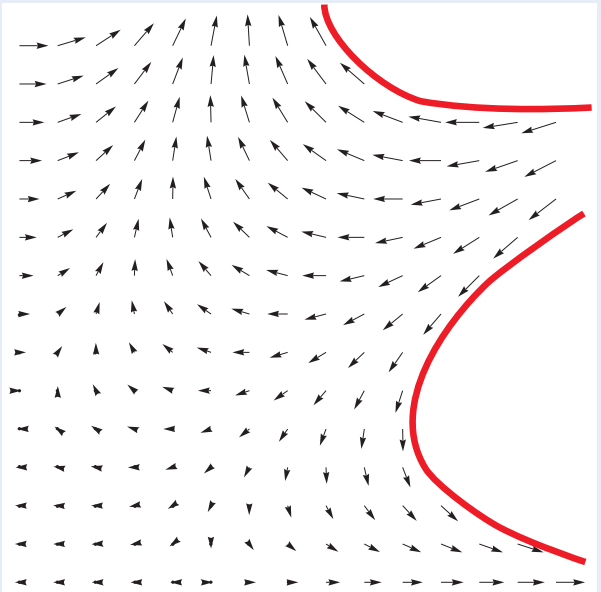
JLogComput'10,CAV'08,FMSD'09,LMCS'12,ITP'12

Differential Invariants for Differential Equations

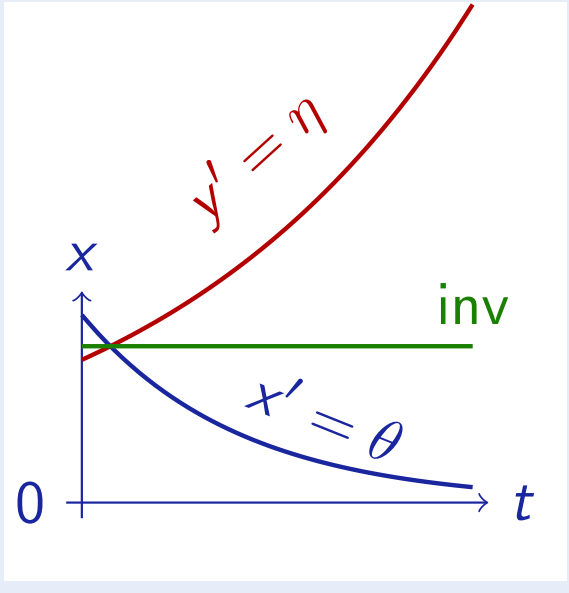
Differential Invariant

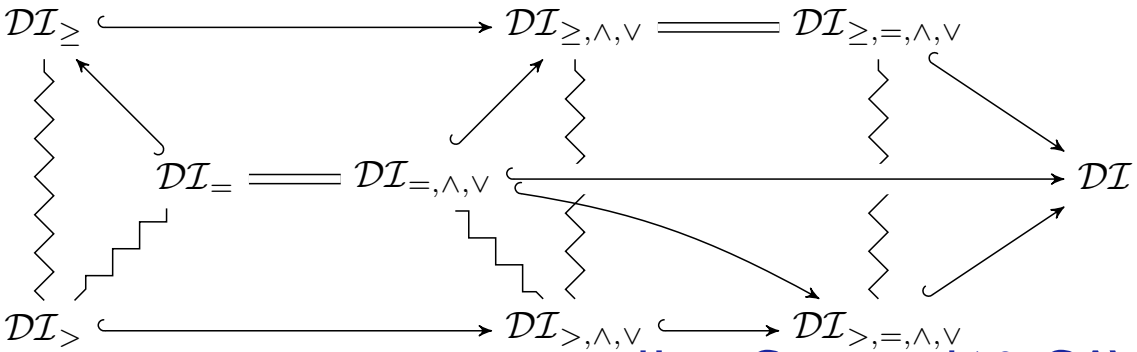


Differential Cut



Differential Ghost





Logic

Provability theory

Math

Characteristic PDE

JLogComput'10,CAV'08,FMSD'09,LMCS'12,ITP'12

Differential Invariant

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

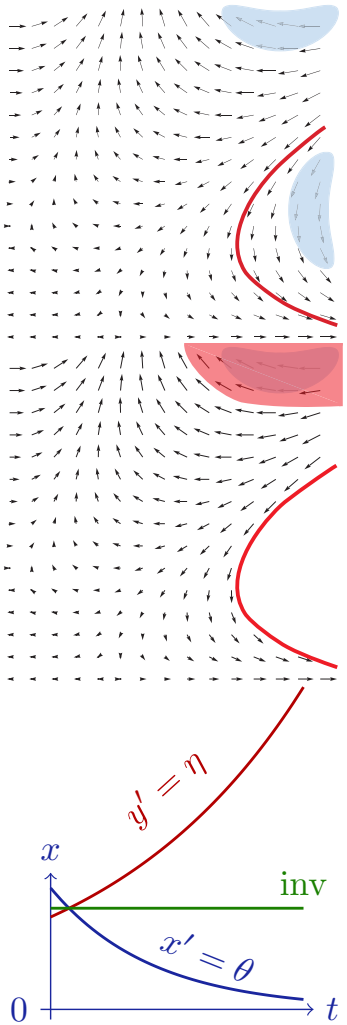
Differential Cut

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

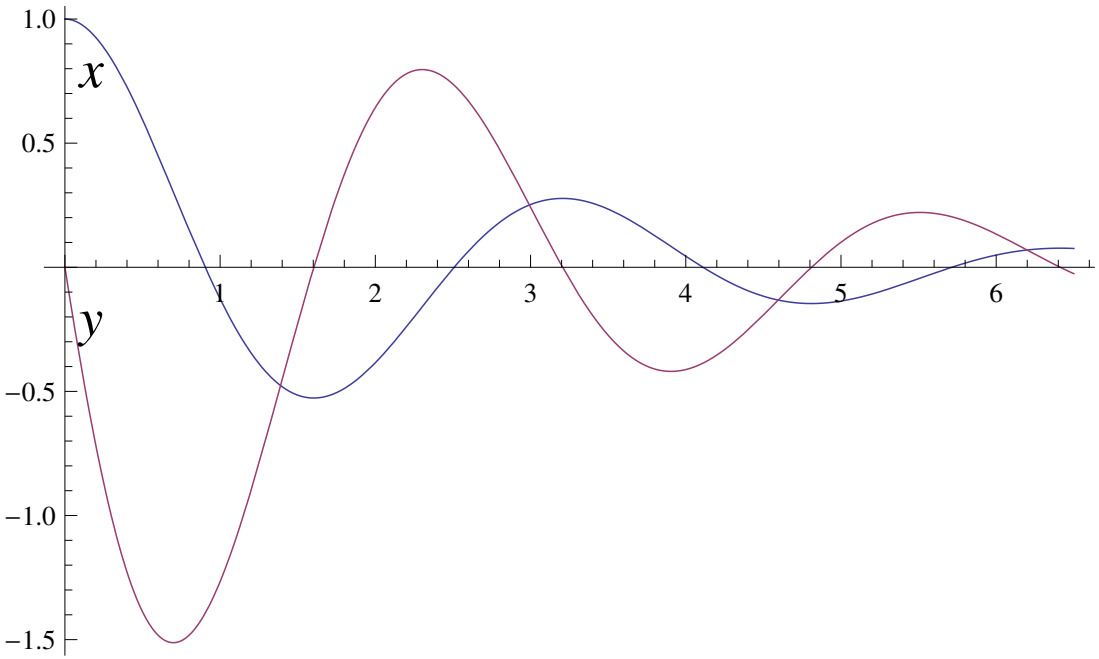
Differential Ghost

$$\frac{F \leftrightarrow \exists y\, G \quad G \rightarrow [x' = \theta, y' = \eta \,\&\, H]G}{F \rightarrow [x' = \theta \,\&\, H]F}$$

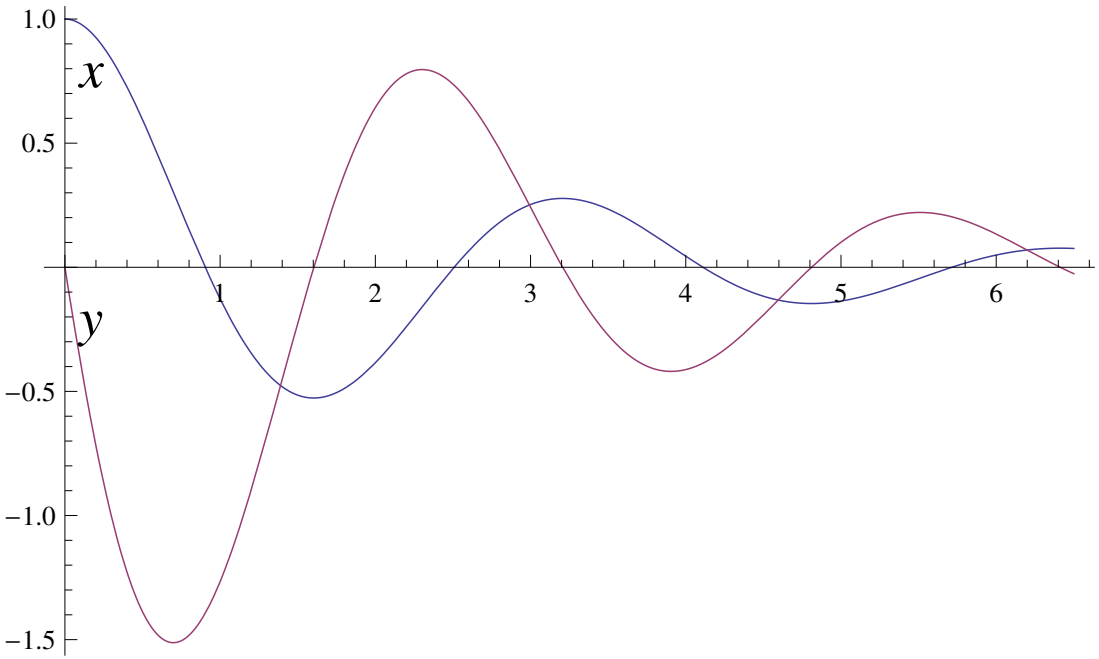
if new $y' = \eta$ has a global solution



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



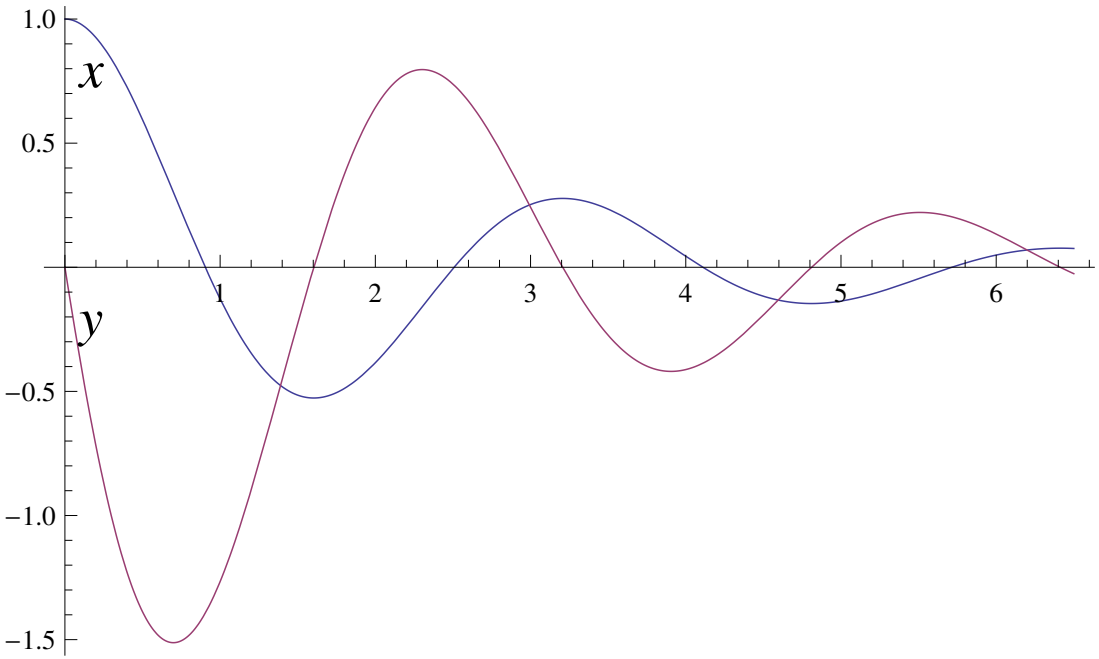
$$\frac{\omega\geq 0 \wedge d\geq 0 \rightarrow (2\omega^2 x x' + 2y y' \leq 0)^{y,\ -\omega^2 x - 2d\omega y}_{x',\ y'}}{\omega^2 x^2 + y^2 \leq c^2 \rightarrow [x' = y, y' = -\omega^2 x - 2d\omega y \,\&\, (\omega\geq 0 \wedge d\geq 0)]\, \omega^2 x^2 + y^2 \leq c^2}$$



$$\omega \geq 0 \wedge d \geq 0 \rightarrow 2\omega^2 xy + 2y(-\omega^2 x - 2d\omega y) \leq 0$$

$$\omega \geq 0 \wedge d \geq 0 \rightarrow (2\omega^2 xx' + 2yy' \leq 0)^{y,\ -\omega^2 x - 2d\omega y}_{x',\ y'}$$

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

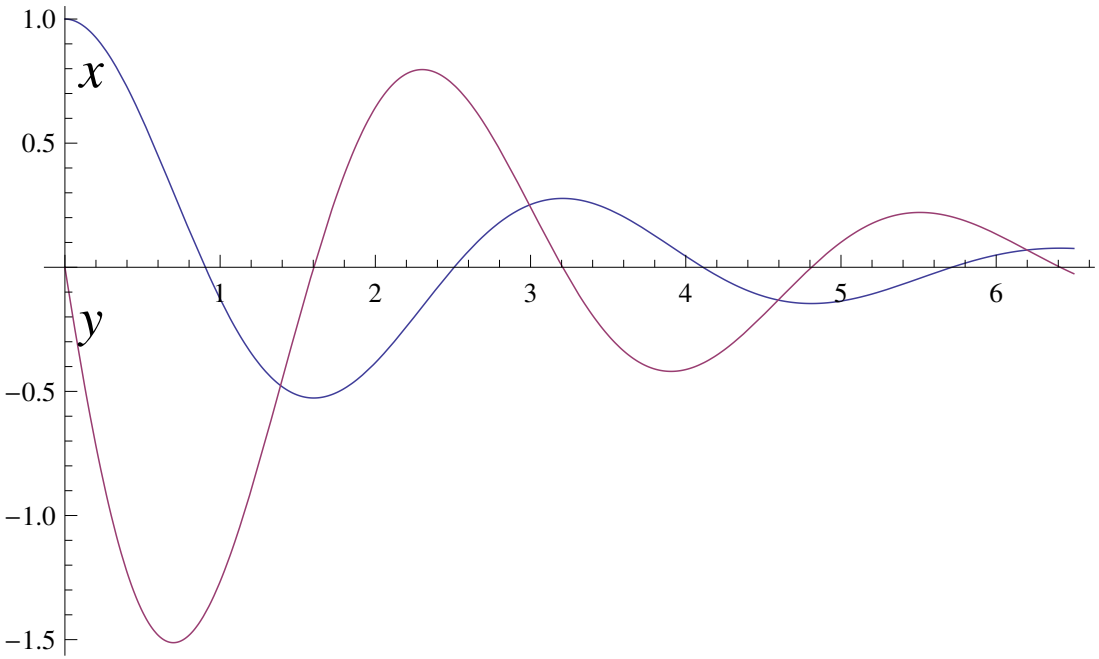


$$*$$

$$\omega\geq 0 \wedge d\geq 0 \rightarrow 2\omega^2 xy + 2y(-\omega^2 x - 2d\omega y) \leq 0$$

$$\omega\geq 0 \wedge d\geq 0 \rightarrow (2\omega^2 xx' + 2yy' \leq 0)^{y,\ -\omega^2 x-2d\omega y}_{x',\ y'}$$

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

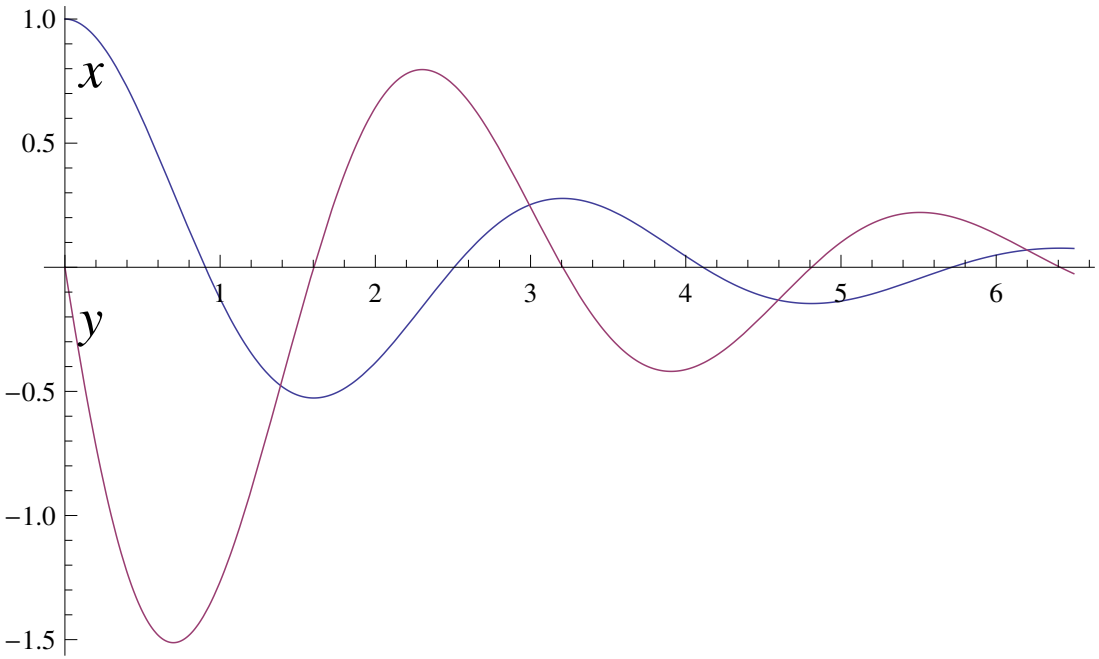


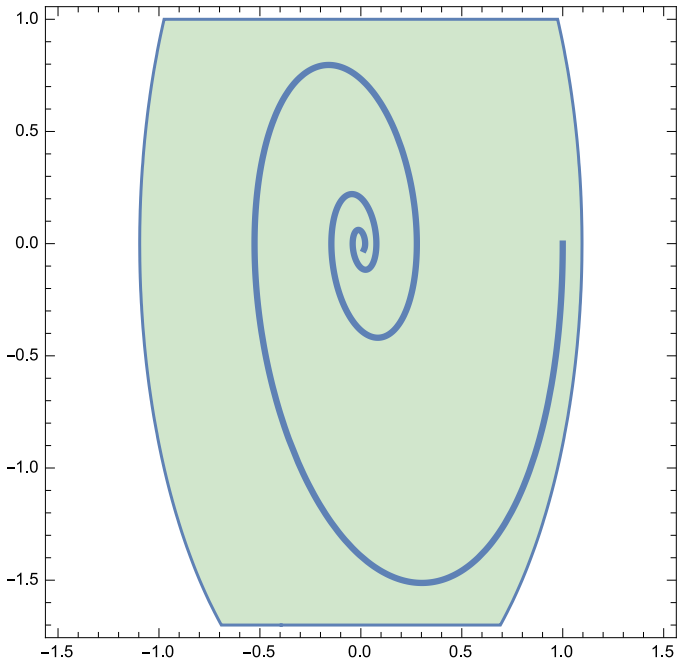
$$*$$

$$\omega \geq 0 \wedge d \geq 0 \rightarrow 2\omega^2 xy + 2y(-\omega^2 x - 2d\omega y) \leq 0$$

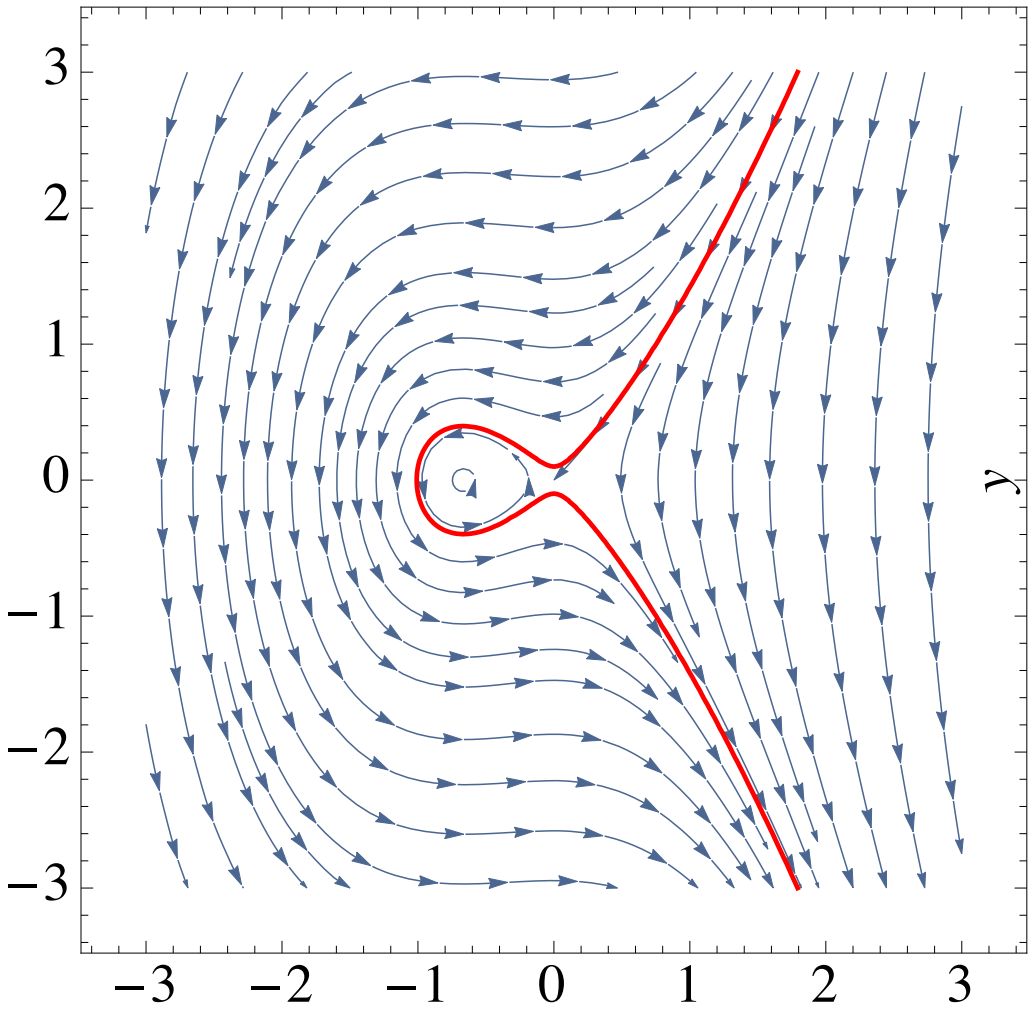
$$\omega \geq 0 \wedge d \geq 0 \rightarrow (2\omega^2 x x' + 2y y' \leq 0)^{y,\ -\omega^2 x - 2d\omega y}_{x',\ y'}$$

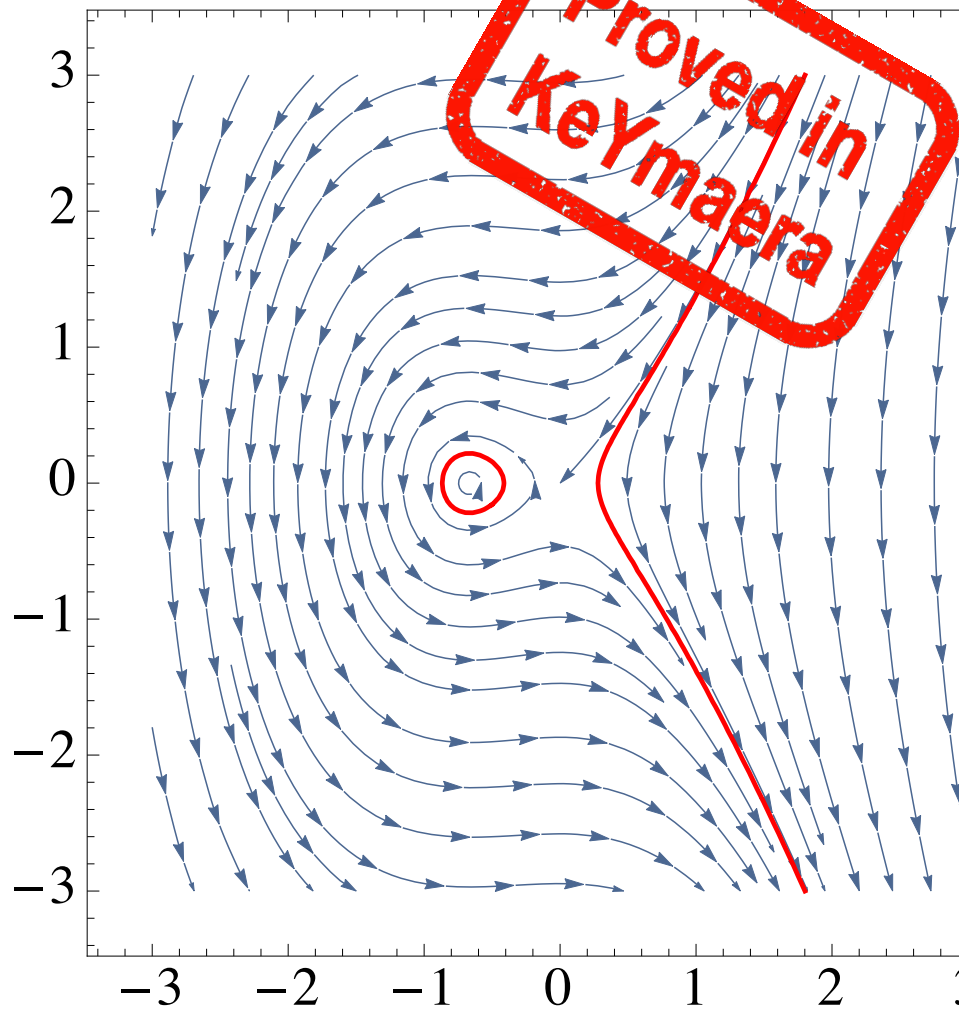
$$\omega^2 x^2 + y^2 \leq c^2 \rightarrow [x' = y, y' = -\omega^2 x - 2d\omega y \,\&\, (\omega \geq 0 \wedge d \geq 0)]\, \omega^2 x^2 + y^2 \leq c^2$$





Differential Invariants for Differential Equations



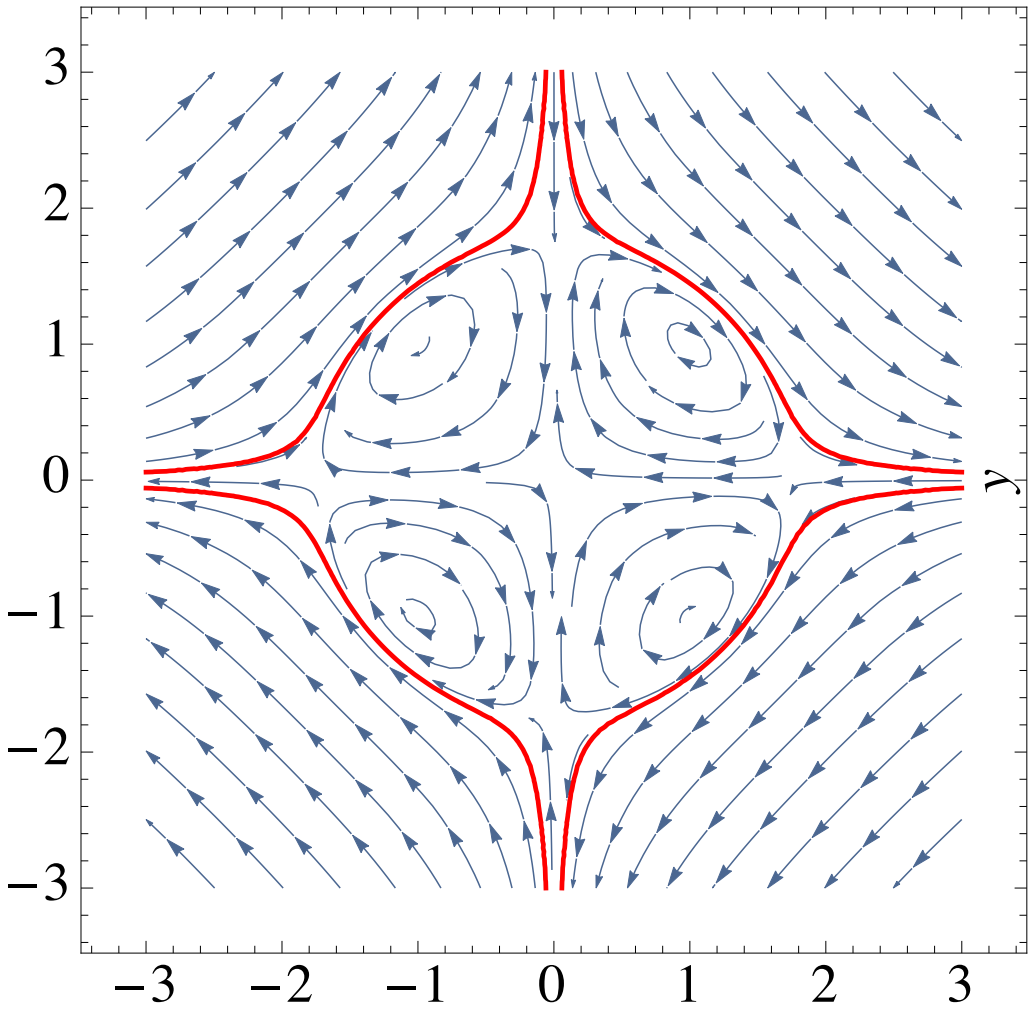


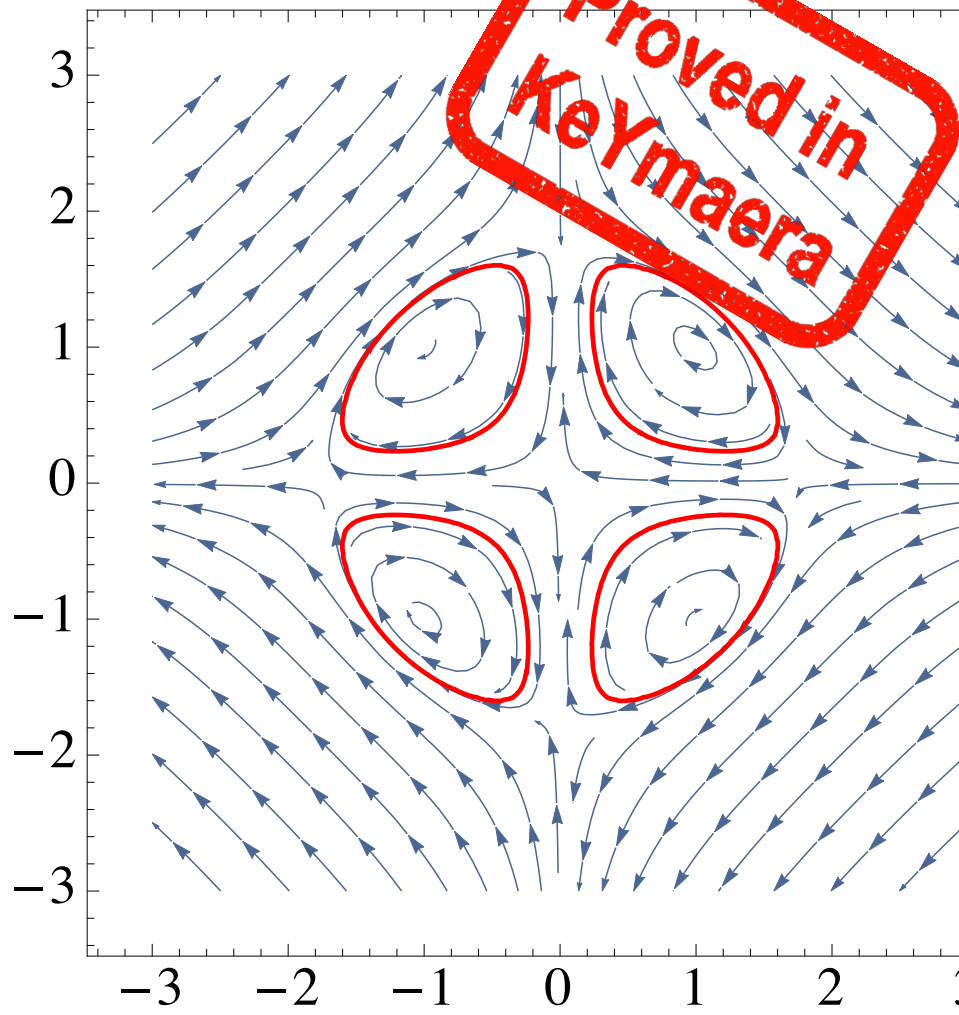
Proved in KeYmaera

$$x^2 + x^3 - y^2 - c = 0 \rightarrow [x' = -2y, y' = -2x - 3x^2]\, x^2 + x^3 - y^2 - c = 0$$

SAS'14

Differential Invariants for Differential Equations



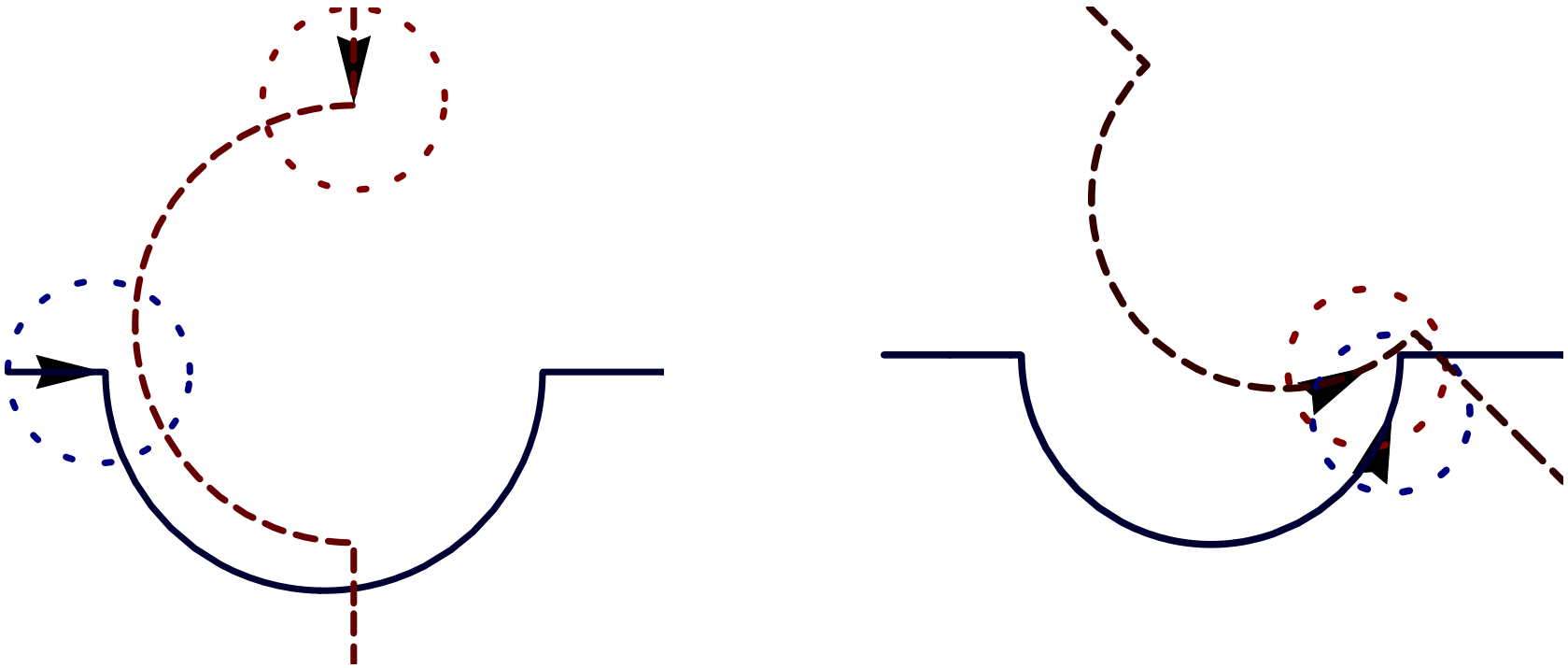


Proved in KeYmaera

$$[x' = 2x^4y+4x^2y^3-6x^2y, y' = -4x^3y^2-2xy^4+6xy^2]x^4y^2+x^2y^4-3x^2y^2 \leq c$$

SAS'14

Air Traffic Control

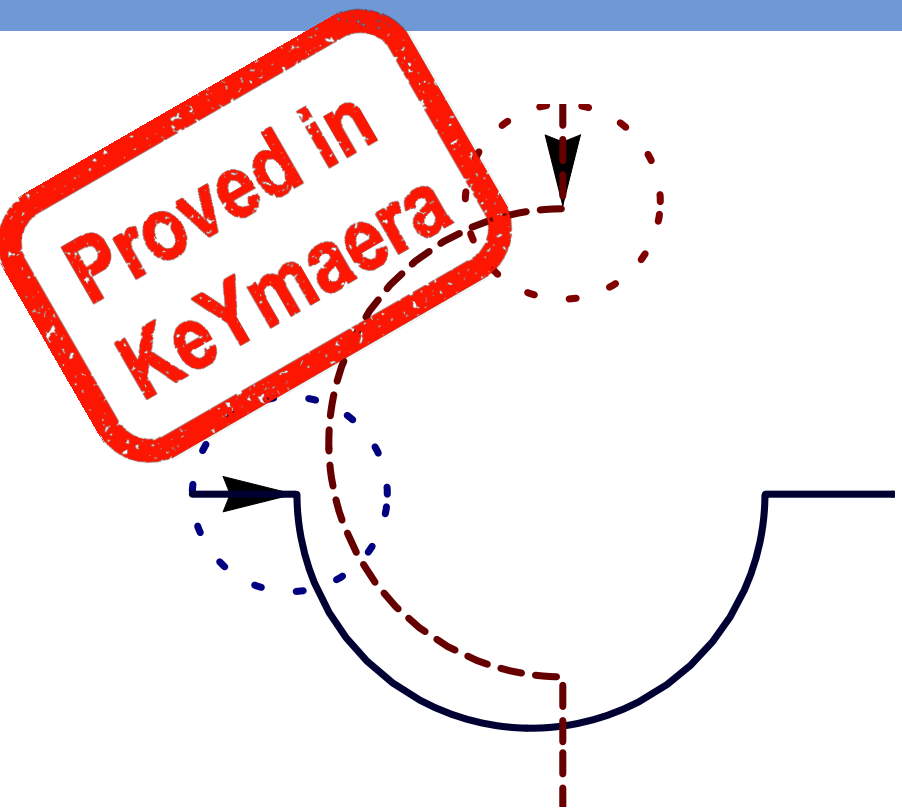


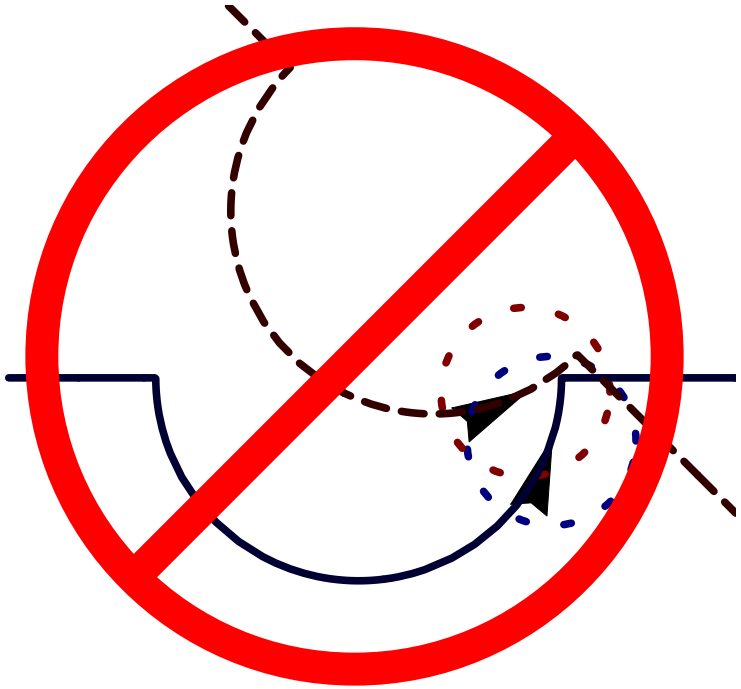
Verification

Simple proof distinguishes safe from unsafe flight maneuver

JLogComput'10

Air Traffic Control





Verification

Simple proof distinguishes safe from unsafe flight maneuver

JLogComput'10

Differential Radical Invariants

Theorem (Differential radical invariant characterization)

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

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

$$(h^{(N)})^p_{x'} = \sum_{i=0}^{N-1} g_i (h^{(i)})^p_{x'} \quad (g_i \in \mathbb{R}[x])$$

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)

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

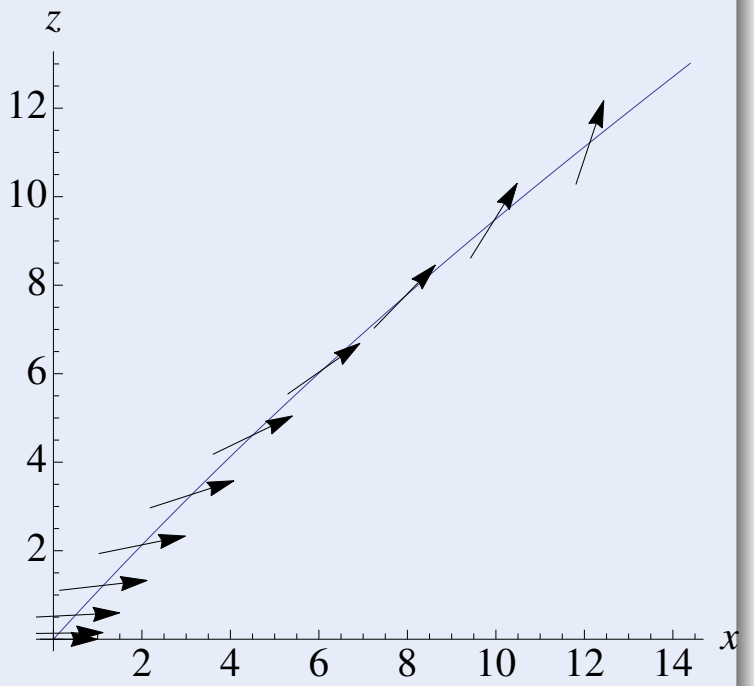
$$w' = \frac{Z}{m} + g\cos(\theta) + qu \quad \text{vertical velocity}$$

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

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

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

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



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

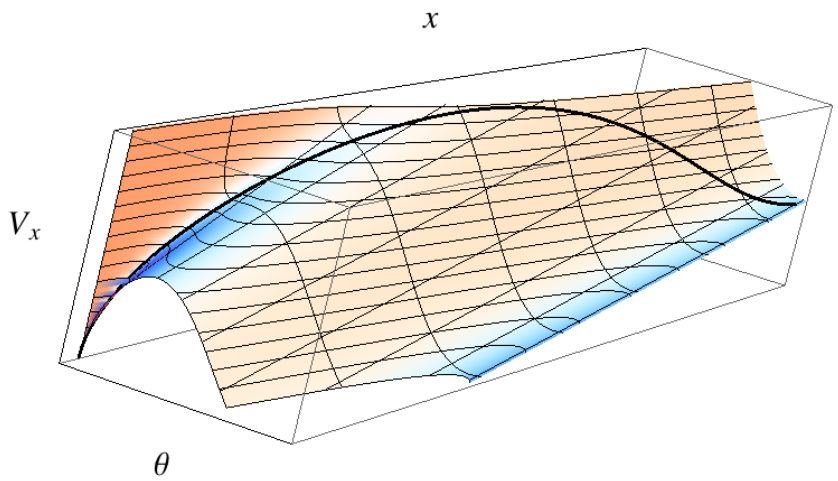
Case Study: Longitudinal Dynamics of an Airplane

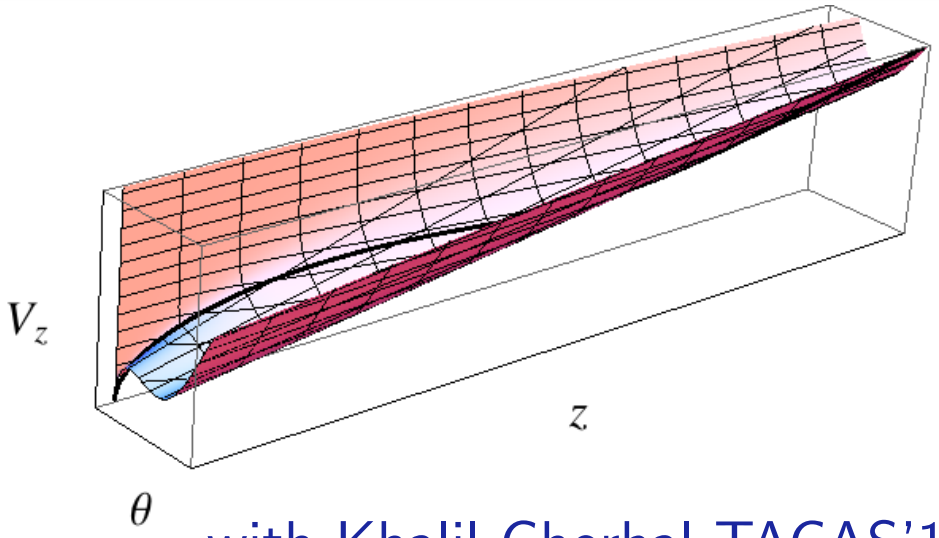
Result (DRI Automatically Generates Invariant Functions)

$$\frac{Mz}{I_{yy}} + g\theta + \left(\frac{X}{m} - qw\right)\cos(\theta) + \left(\frac{Z}{m} + qu\right)\sin(\theta)$$

$$\frac{Mx}{I_{yy}} - \left(\frac{Z}{m} + qu\right)\cos(\theta) + \left(\frac{X}{m} - qw\right)\sin(\theta)$$

$$-q^2 + \frac{2M\theta}{I_{yy}}$$





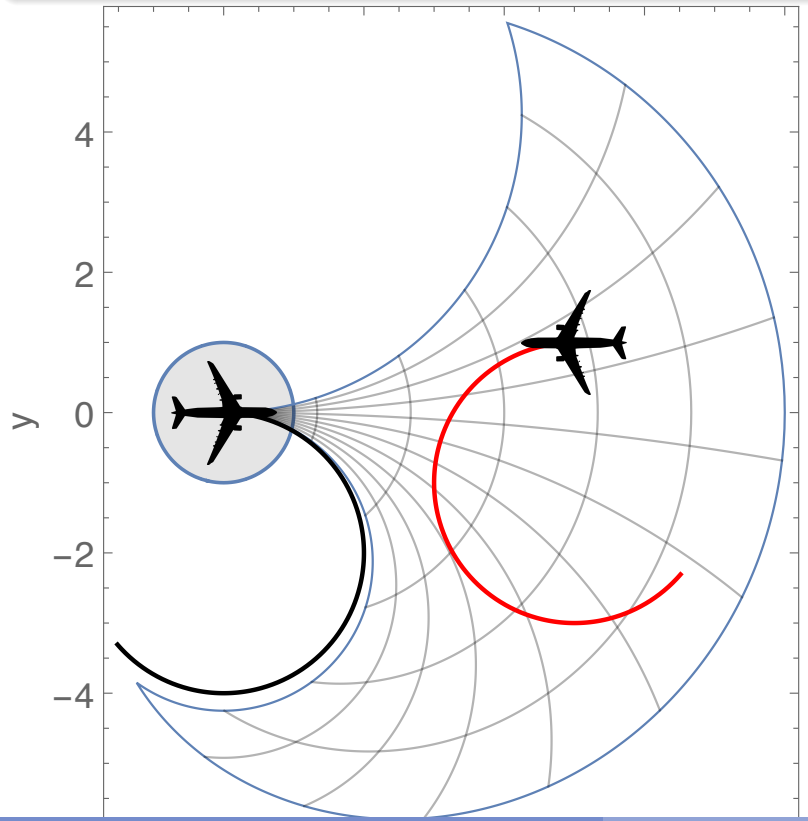
with Khalil Ghorbal TACAS'14

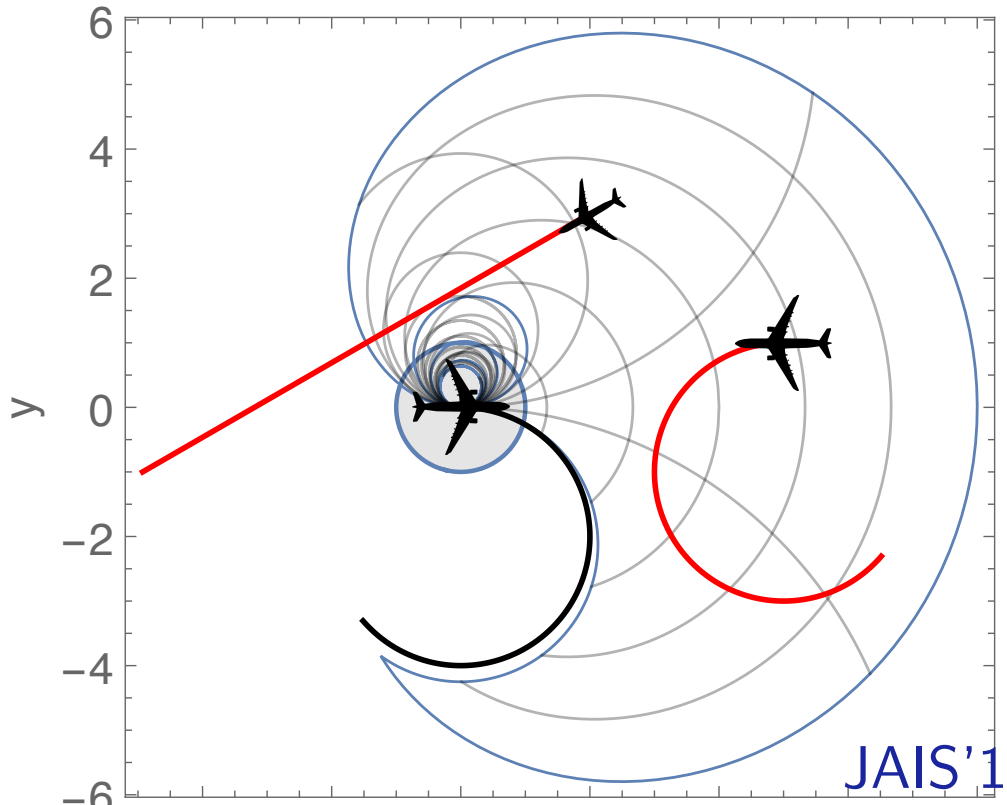
Case Study: Dubins Dynamics of 2 Airplanes

Result (DRI Automatically Generates Invariants)

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

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

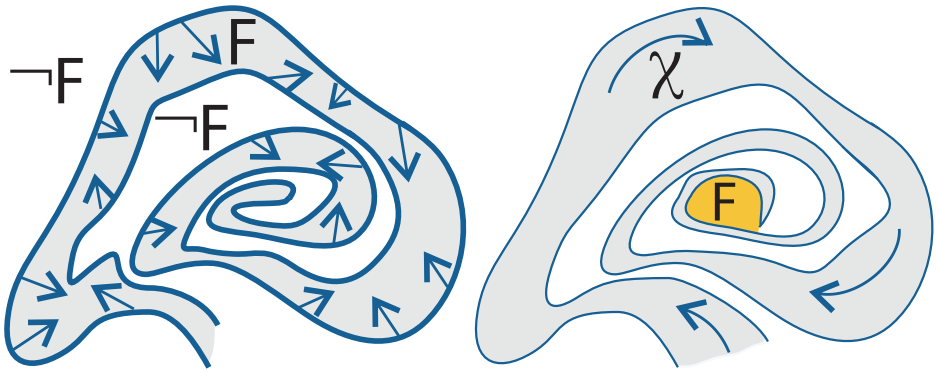


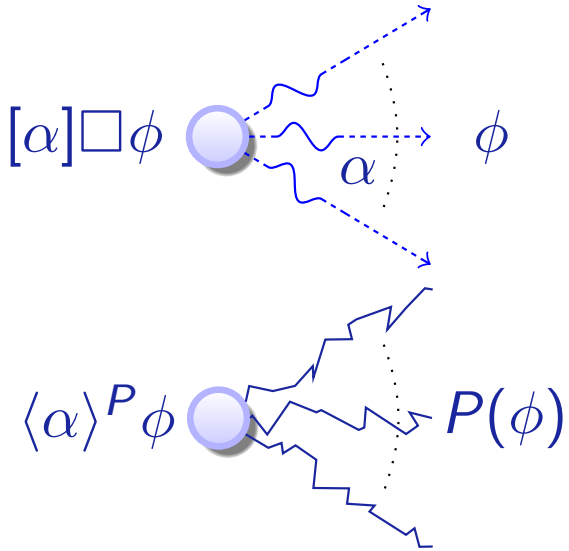


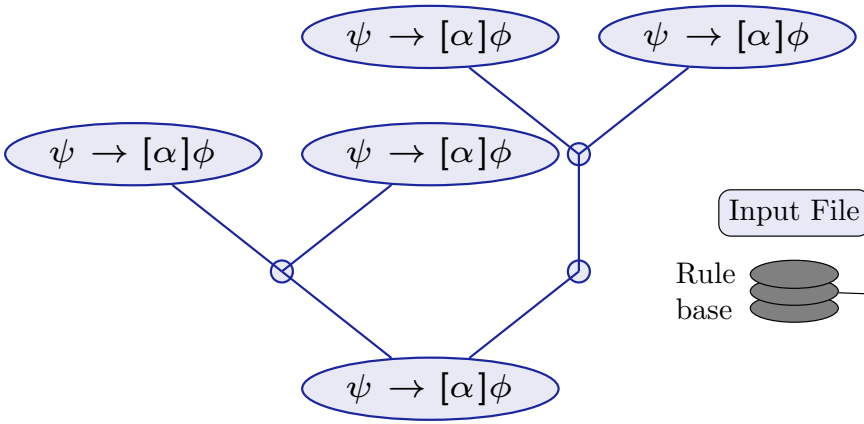
JAIS'14

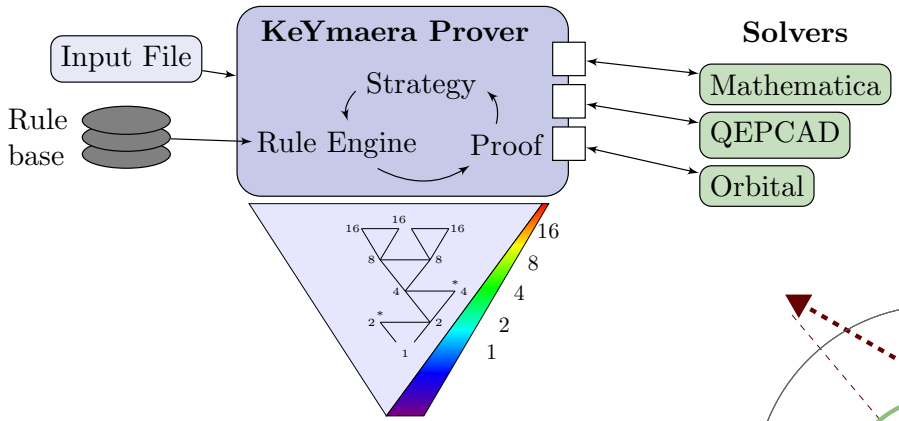
Outline

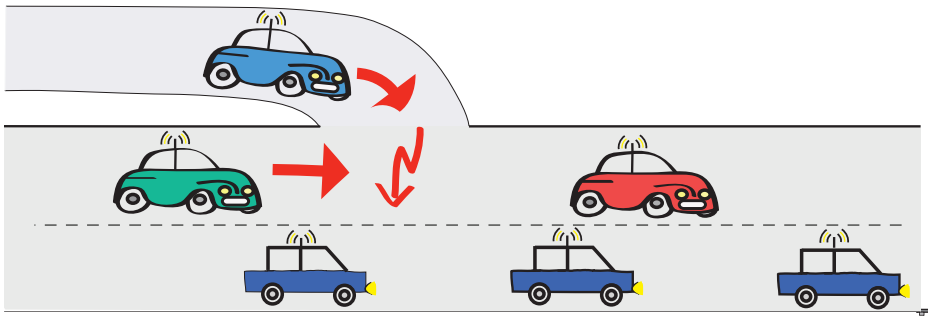
1. CPS are Multi-Dynamical Systems
   - Hybrid Systems
   - Hybrid Games
2. Dynamic Logic for Multi-Dynamical Systems
   - Syntax
   - Semantics
3. Proofs for CPS
4. Theory of CPS
   - Soundness and Completeness
   - Differential Invariants
   - Differential Radical Invariants
5. Applications
6. Summary

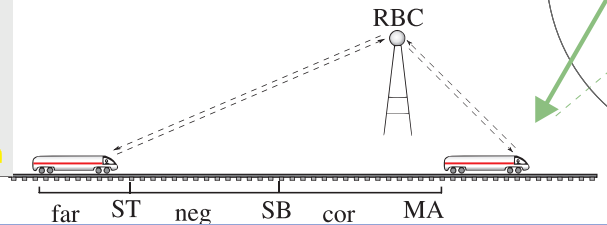

¬F
F
¬F
χ
F

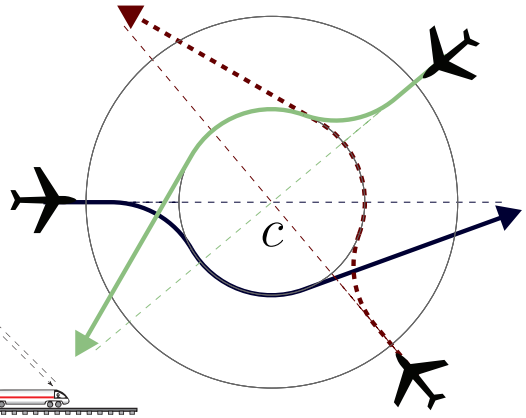

[α]□φ
α
φ
⟨α⟩^P φ
P(φ)


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

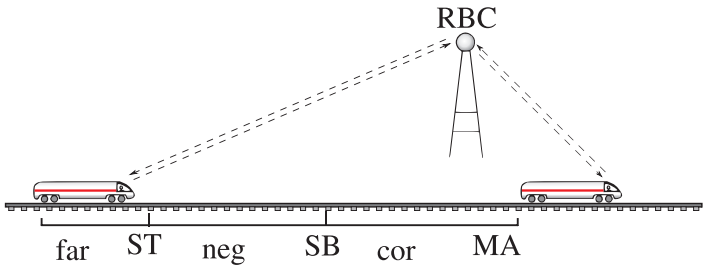

KeYmaera Prover
Input File
Strategy
Rule base
Rule Engine
Proof
Solvers
Mathematica
QEPCAD
Orbital
16
8
4
2
1

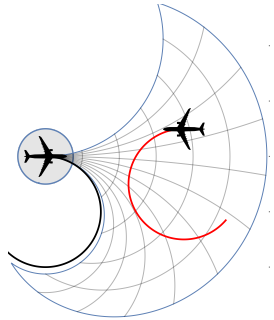


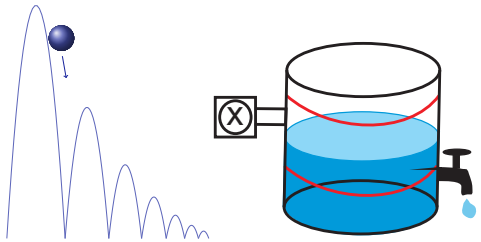

RBC
far
ST
neg
SB
cor
MA

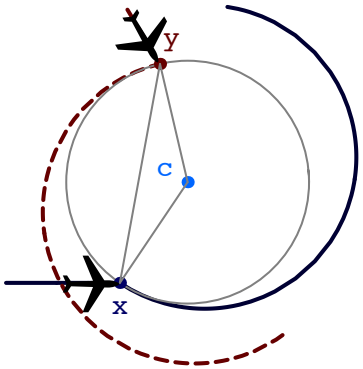

c

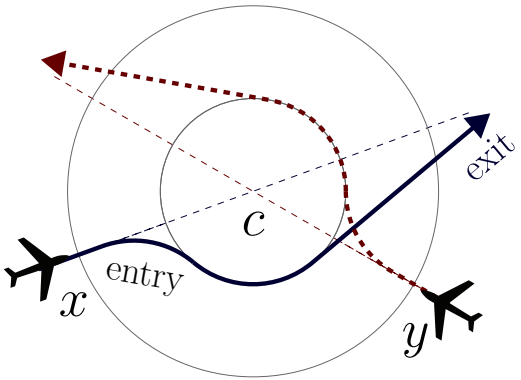
Successful CPS Proofs

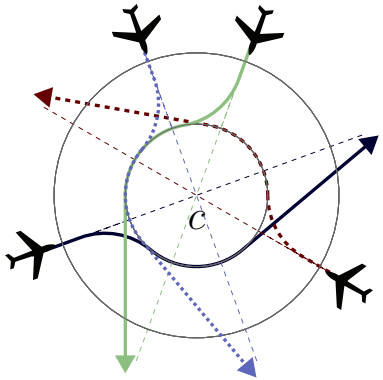


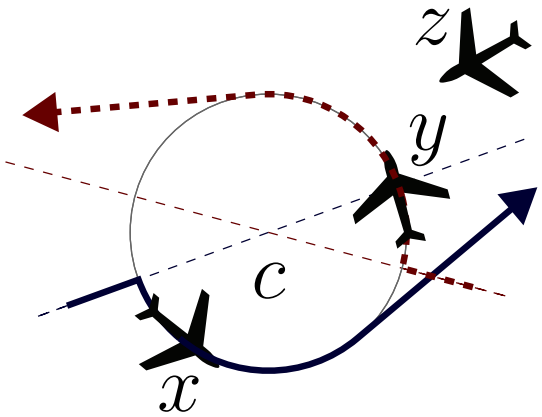


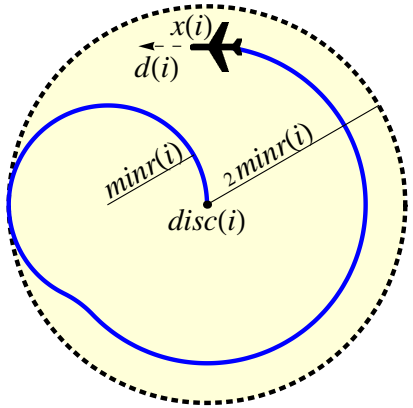


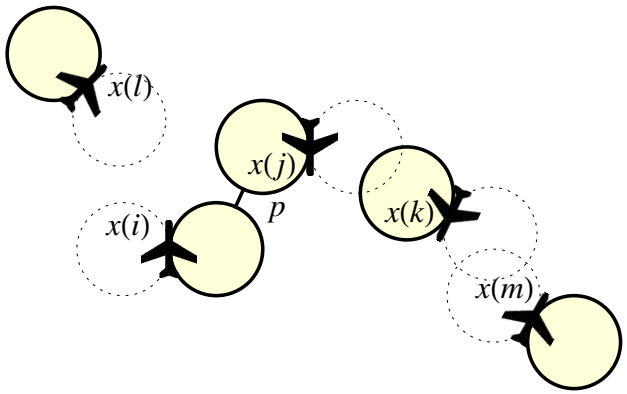






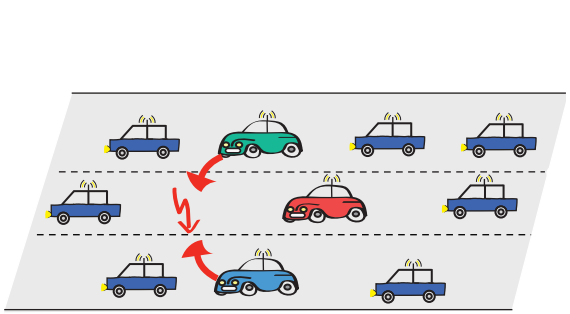


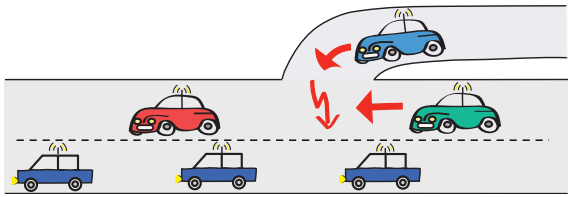


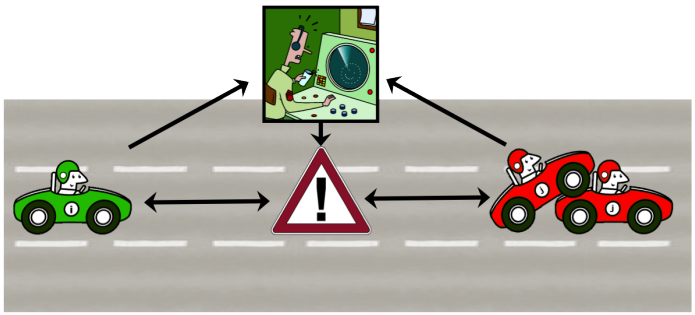


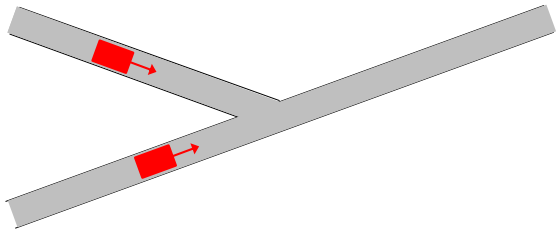
ICFEM'09,JAIS'14,CAV'08,FM'09,HSCC'11,HSCC'13

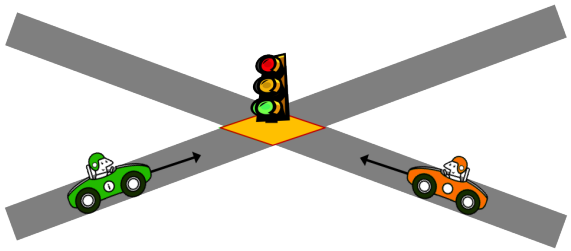
Successful CPS Proofs

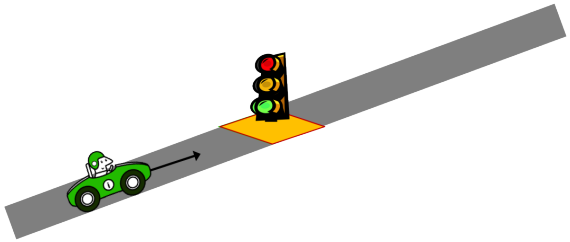


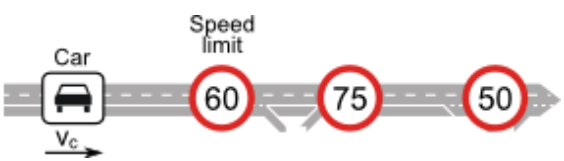


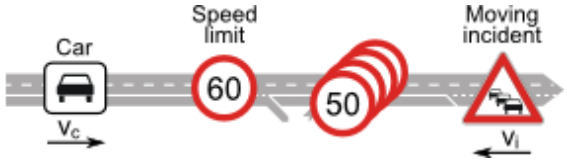


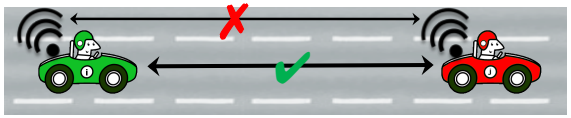


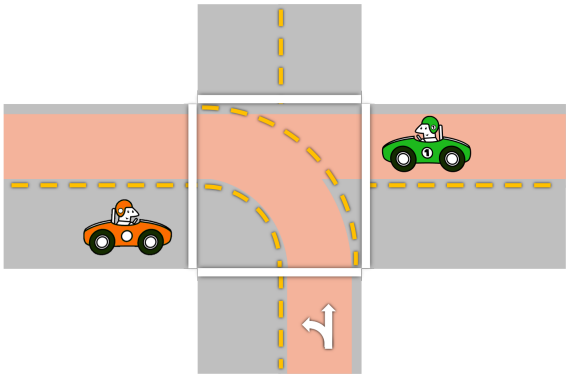


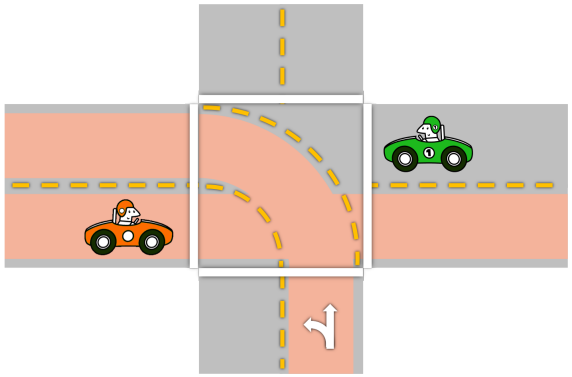


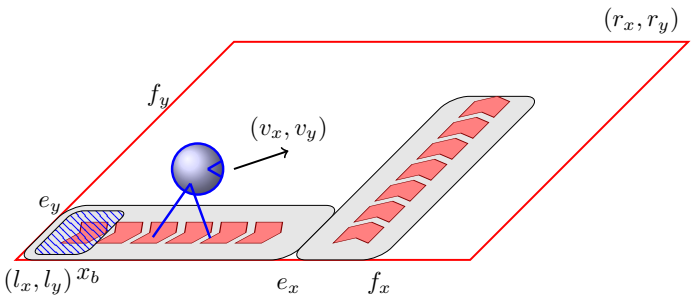






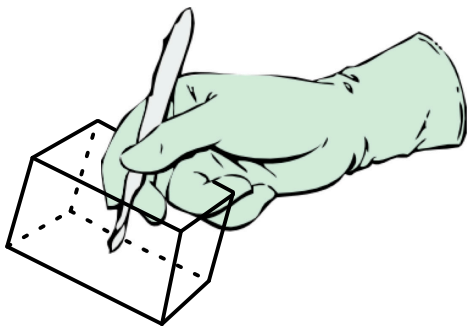


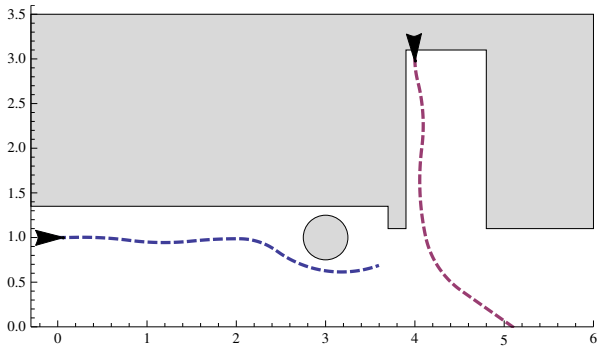


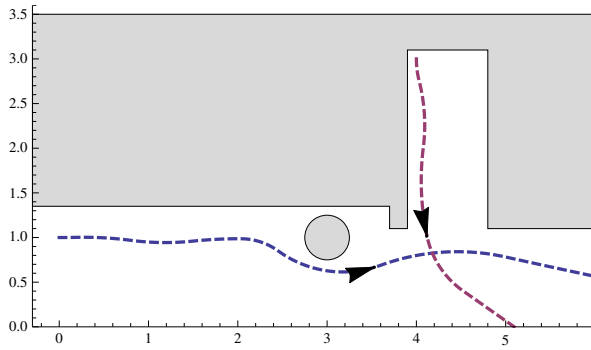


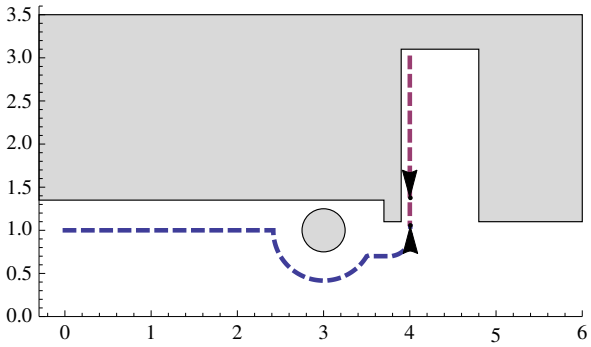
FM'11,LMCS'12,ICCPS'12,ITSC'11,ITSC'13,IJCAR'12

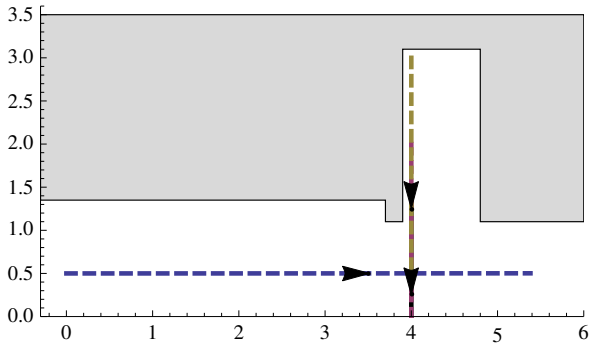
Successful CPS Proofs

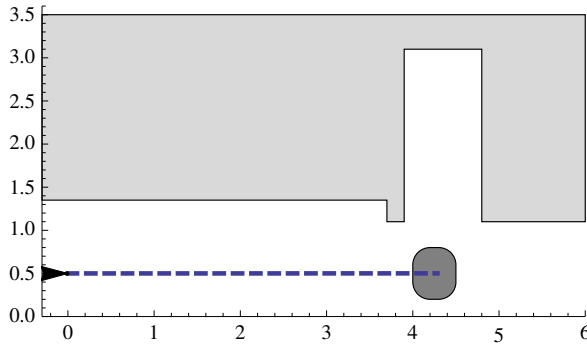


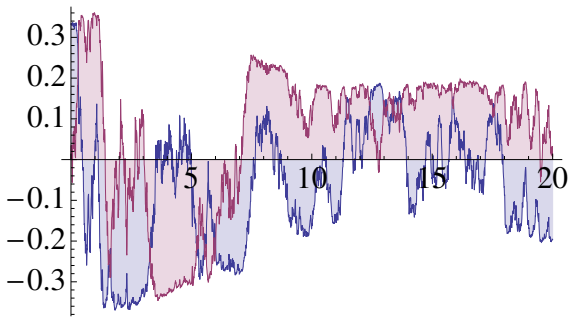


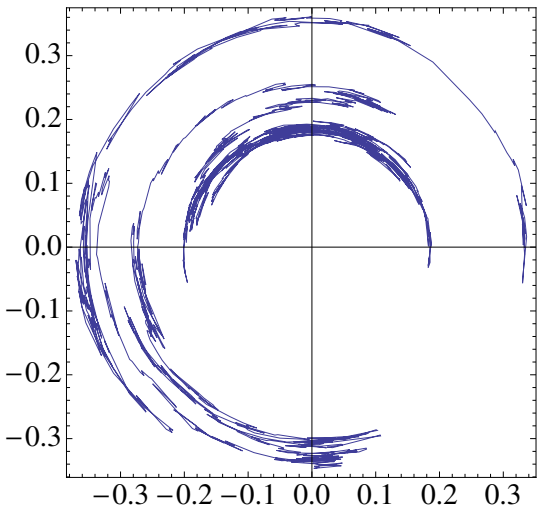


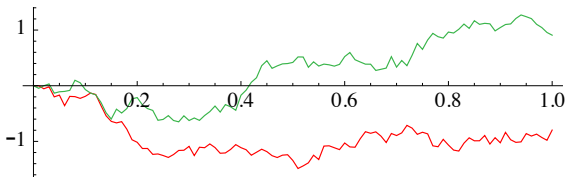






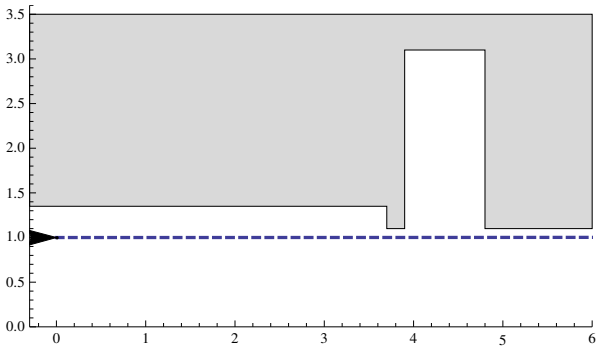
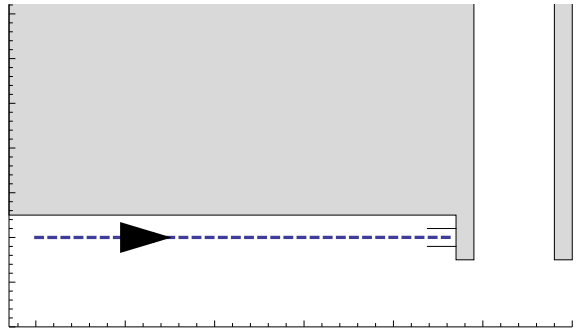
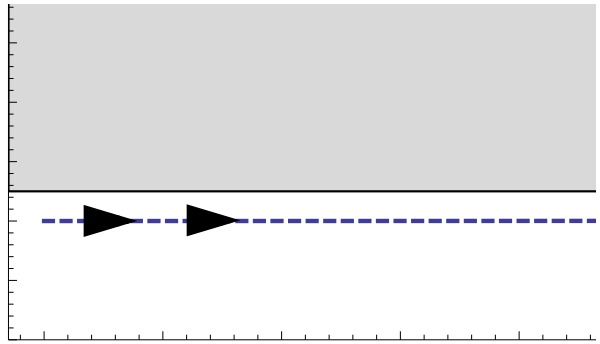
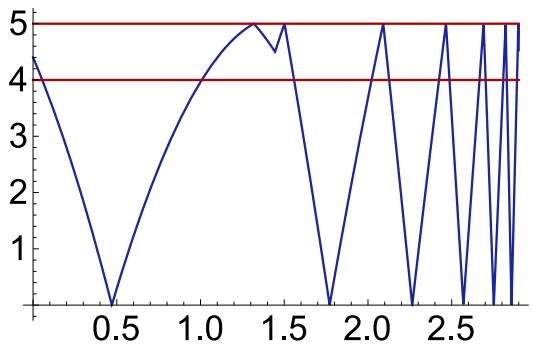
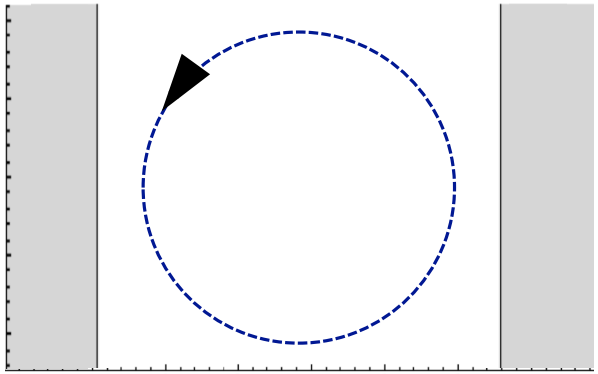
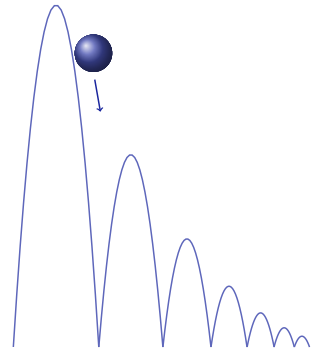
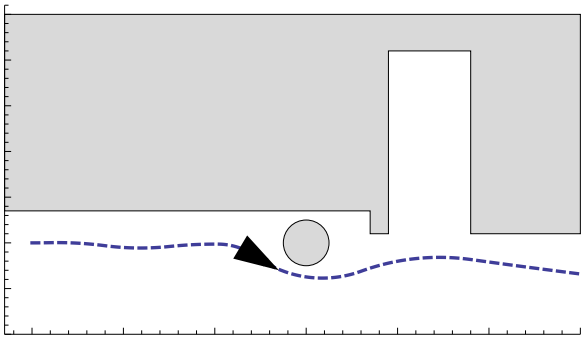
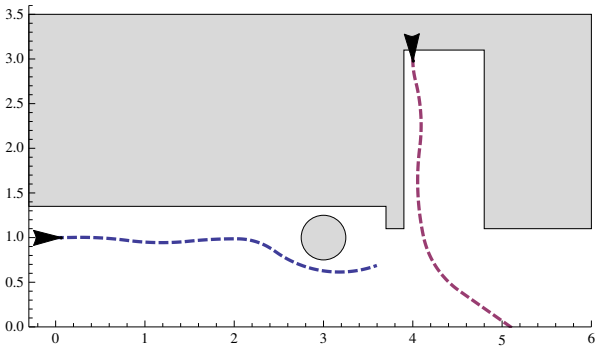
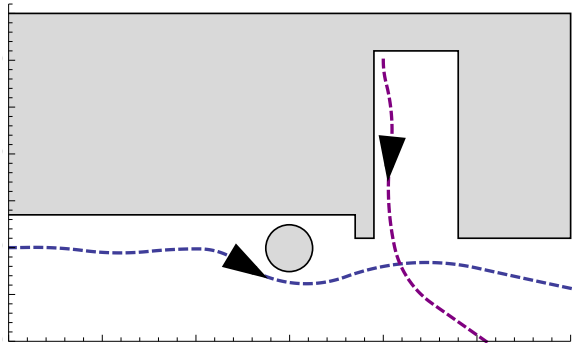






HSCC'13,RSS'13,CADE'12

Successful CPS Proofs By Undergrads

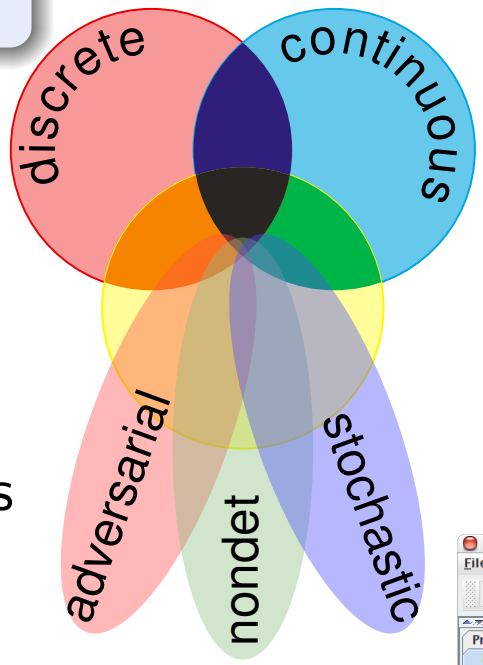
15-424/624 *Foundations of Cyber-Physical Systems* students

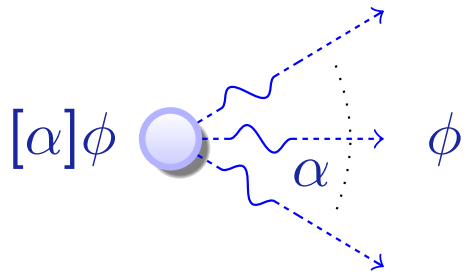
Outline

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
   - Differential Radical Invariants
5. Applications
6. Summary

differential dynamic logic

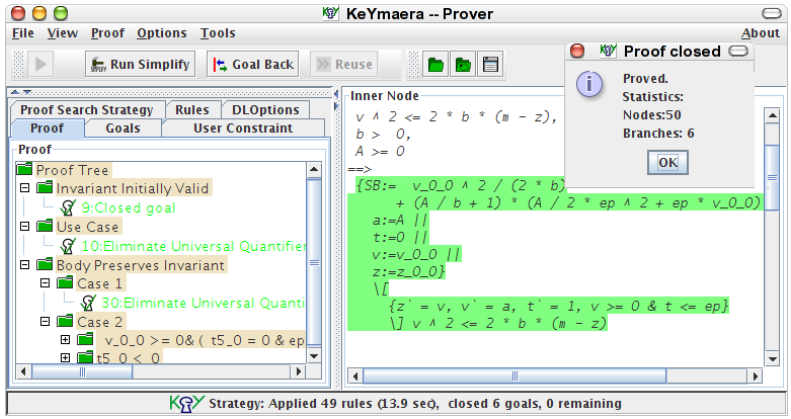
$$\mathrm{d}\mathcal{L} = \mathrm{DL} + \mathrm{HP}$$

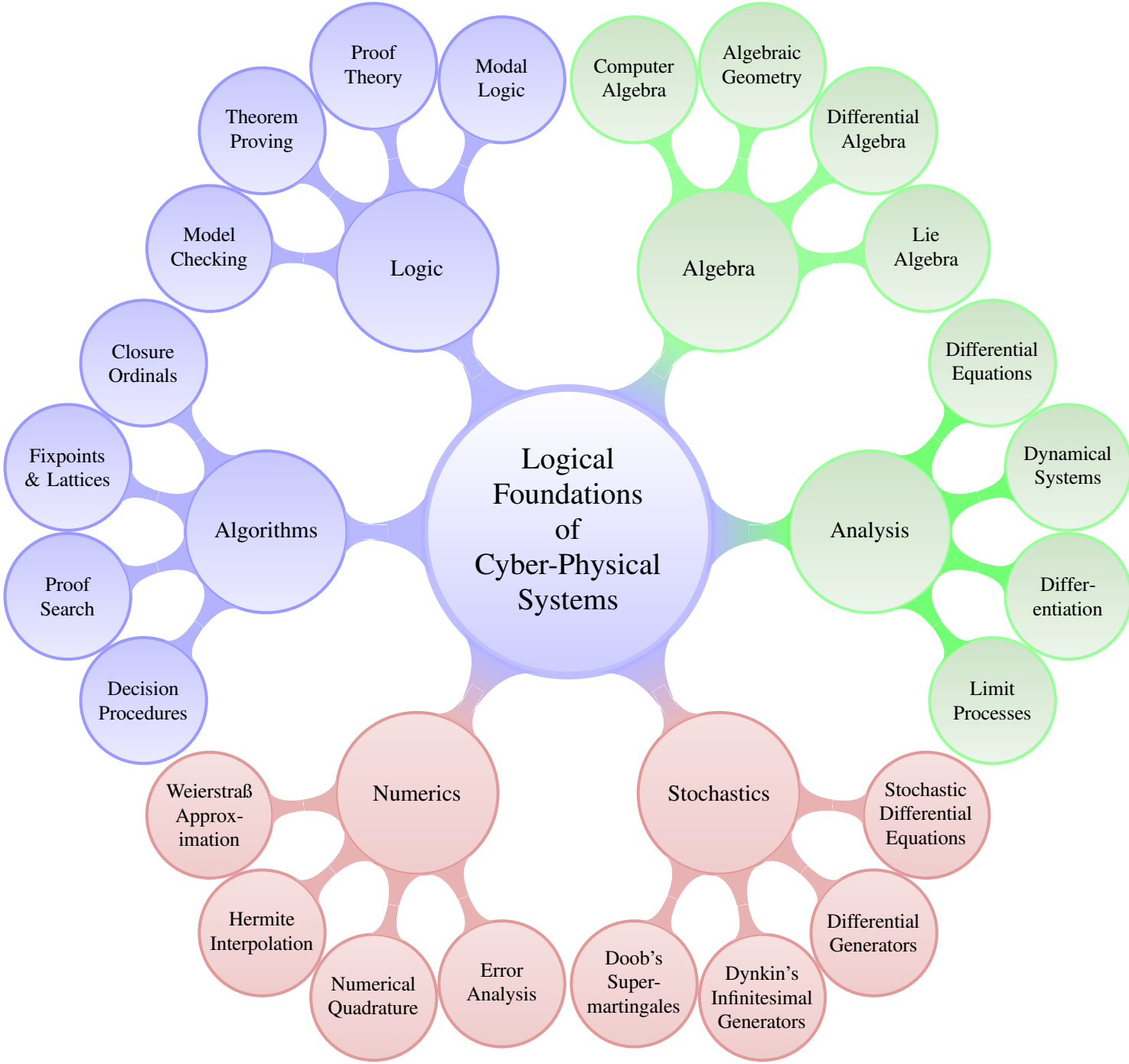


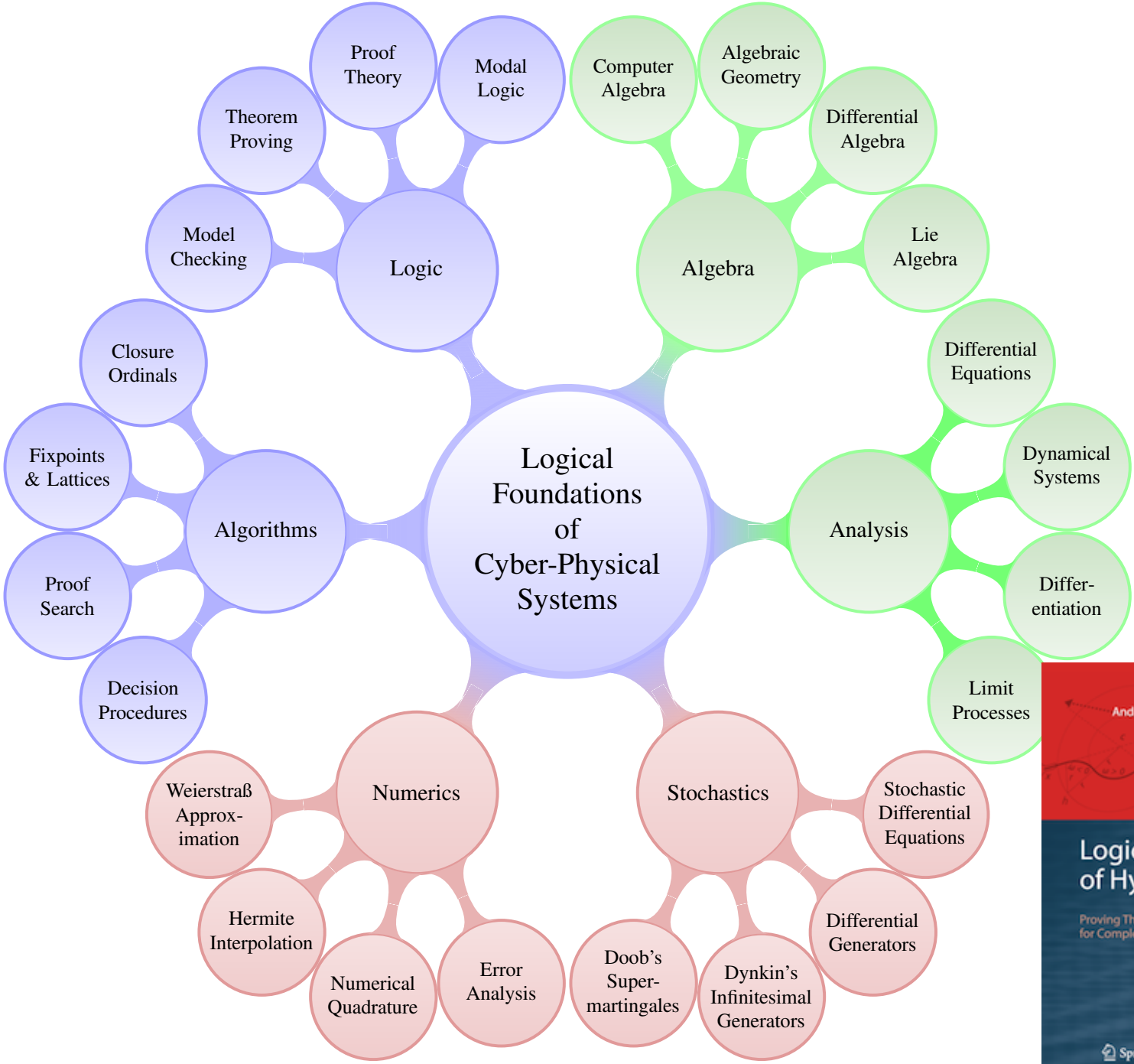


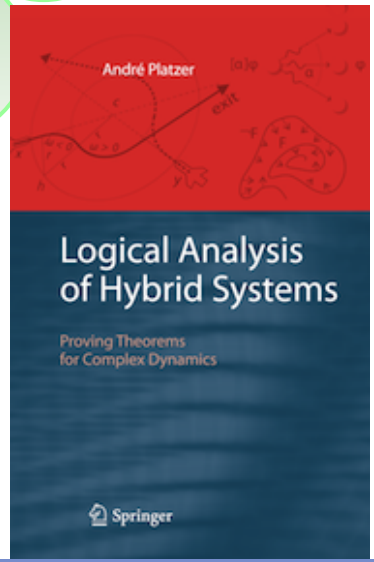
- Multi-dynamical systems
- Combine simple dynamics
- Tame complexity
- Logic & proofs for CPS
- Theory of CPS
- Applications
- Undergrad course 15-424

KeYmaera




Logical Foundations of Cyber-Physical Systems
Logic
Proof Theory
Modal Logic
Theorem Proving
Model Checking
Algebra
Computer Algebra
Algebraic Geometry
Differential Algebra
Lie Algebra
Analysis
Differential Equations
Dynamical Systems
Differ-entiation
Limit Processes
Algorithms
Closure Ordinals
Fixpoints & Lattices
Proof Search
Decision Procedures
Numerics
Weierstraß Approx-imation
Hermite Interpolation
Numerical Quadrature
Error Analysis
Stochastics
Stochastic Differential Equations
Differential Generators
Doob's Super-martingales
Dynkin's Infinitesimal Generators


Logical Foundations of Cyber-Physical Systems
Logic
Proof Theory
Modal Logic
Theorem Proving
Model Checking
Algebra
Computer Algebra
Algebraic Geometry
Differential Algebra
Lie Algebra
Analysis
Differential Equations
Dynamical Systems
Differ-entiation
Limit Processes
Algorithms
Closure Ordinals
Fixpoints & Lattices
Proof Search
Decision Procedures
Numerics
Weierstraß Approx-imation
Hermite Interpolation
Numerical Quadrature
Error Analysis
Stochastics
Stochastic Differential Equations
Differential Generators
Doob's Super-martingales
Dynkin's Infinitesimal Generators


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

André Platzer.
Logics of dynamical systems.
In LICS [15], pages 13–24.
doi:10.1109/LICS.2012.13.

André Platzer.
Foundations of cyber-physical systems.
Lecture Notes 15-424/624, Carnegie Mellon University, 2013.
URL: http://www.cs.cmu.edu/~aplatzer/course/fcps13/fcps13.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.
A complete axiomatization of quantified differential dynamic logic for distributed hybrid systems.

Logical Methods in Computer Science, 8(4):1–44, 2012.
Special issue for selected papers from CSL'10.
doi:10.2168/LMCS-8(4:17)2012.

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.

André Platzer.
A complete axiomatization of differential game logic for hybrid games.
Technical Report CMU-CS-13-100R, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, January, Revised and extended in July 2013.

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

André Platzer.
The complete proof theory of hybrid systems.
In LICS [15], pages 541–550.
doi:10.1109/LICS.2012.64.

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

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

André Platzer.
The structure of differential invariants and differential cut elimination.
Logical Methods in Computer Science, 8(4):1–38, 2012.
doi:10.2168/LMCS-8(4:16)2012.

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

Khalil Ghorbal, Andrew Sogokon, and André Platzer.
Invariance of conjunctions of polynomial equalities for algebraic differential equations.
In Markus Müller-Olm and Helmut Seidl, editors, *SAS*, volume 8723 of *LNCS*, pages 151–167. Springer, 2014.
doi:10.1007/978-3-319-10936-7_10.

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.

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

André Platzer.
Differential-algebraic dynamic logic for differential-algebraic programs.
J. Log. Comput., 20(1):309–352, 2010.
Advance Access published on November 18, 2008.
doi:10.1093/logcom/exn070.

André Platzer and Jan-David Quesel.
KeYmaera: A hybrid theorem prover for hybrid systems.
In Alessandro Armando, Peter Baumgartner, and Gilles Dowek, editors, *IJCAR*, volume 5195 of *LNCS*, pages 171–178. Springer, 2008.
doi:10.1007/978-3-540-71070-7_15.

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

doi:10.1007/978-3-540-73099-6_17.

André Platzer.
Quantified differential dynamic logic for distributed hybrid systems.
In Anuj Dawar and Helmut Veith, editors, *CSL*, volume 6247 of *LNCS*, pages 469–483. Springer, 2010.
doi:10.1007/978-3-642-15205-4_36.

André Platzer.
Quantified differential invariants.
In Emilio Frazzoli and Radu Grosu, editors, *HSCC*, pages 63–72. ACM, 2011.
doi:10.1145/1967701.1967713.

André Platzer.
Logics of dynamical systems.
In LICS [15], pages 13–24.
doi:10.1109/LICS.2012.13.

André Platzer.
Foundations of cyber-physical systems.

Lecture Notes 15-424/624, Carnegie Mellon University, 2013.
URL: http://www.cs.cmu.edu/~aplatzer/course/fcps13/fcps13.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.
A complete axiomatization of quantified differential dynamic logic for distributed hybrid systems.
Logical Methods in Computer Science, 8(4):1–44, 2012.
Special issue for selected papers from CSL'10.
doi:10.2168/LMCS-8(4:17)2012.

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.

André Platzer.
A complete axiomatization of differential game logic for hybrid games.
Technical Report CMU-CS-13-100R, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, January, Revised and extended in July 2013.

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

André Platzer.
The complete proof theory of hybrid systems.
In LICS [15], pages 541–550.
doi:10.1109/LICS.2012.64.

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

doi:10.1093/logcom/exn070.

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

André Platzer.
The structure of differential invariants and differential cut elimination.
Logical Methods in Computer Science, 8(4):1–38, 2012.
doi:10.2168/LMCS-8(4:16)2012.

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

Khalil Ghorbal, Andrew Sogokon, and André Platzer.

Invariance of conjunctions of polynomial equalities for algebraic differential equations.
In Markus Müller-Olm and Helmut Seidl, editors, *SAS*, volume 8723 of *LNCS*, pages 151–167. Springer, 2014.
doi:10.1007/978-3-319-10936-7_10.

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.

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

André Platzer.
Differential-algebraic dynamic logic for differential-algebraic programs.
J. Log. Comput., 20(1):309–352, 2010.
Advance Access published on November 18, 2008.
doi:10.1093/logcom/exn070.

André Platzer and Jan-David Quesel.
KeYmaera: A hybrid theorem prover for hybrid systems.
In Alessandro Armando, Peter Baumgartner, and Gilles Dowek, editors, *IJCAR*, volume 5195 of *LNCS*, pages 171–178. Springer, 2008.
doi:10.1007/978-3-540-71070-7_15.

André Platzer.
Differential dynamic logic for verifying parametric hybrid systems.
In Nicola Olivetti, editor, *TABLEAUX*, volume 4548 of *LNCS*, pages 216–232. Springer, 2007.
doi:10.1007/978-3-540-73099-6_17.

André Platzer.
Quantified differential dynamic logic for distributed hybrid systems.
In Anuj Dawar and Helmut Veith, editors, *CSL*, volume 6247 of *LNCS*, pages 469–483. Springer, 2010.
doi:10.1007/978-3-642-15205-4_36.

André Platzer.
Quantified differential invariants.

In Emilio Frazzoli and Radu Grosu, editors, *HSCC*, pages 63–72. ACM, 2011.
doi:10.1145/1967701.1967713.

7. Formal Details
   - Soundness Proof
   - Completeness Proof
8. Differential Algebraic Dynamic Logic DAL (Excerpt)
   - Differential Invariants
9. Differential Temporal Dynamic Logic dTL (Excerpt)
10. Deduction Modulo Real Algebraic and Computer Algebraic Constraints
11. European Train Control System
12. Collision Avoidance Maneuvers in Air Traffic Control
13. Hybrid Automata Embedding
14. Distributed Hybrid Systems
15. Car Control Verification
16. Stochastic Hybrid Systems

7 Formal Details
- Soundness Proof
- Completeness Proof

8 Differential Algebraic Dynamic Logic DAL (Excerpt)
- Differential Invariants

9 Differential Temporal Dynamic Logic dTL (Excerpt)

10 Deduction Modulo Real Algebraic and Computer Algebraic Constraints

11 European Train Control System

12 Collision Avoidance Maneuvers in Air Traffic Control

13 Hybrid Automata Embedding

14 Distributed Hybrid Systems

15 Car Control Verification

16 Stochastic Hybrid Systems

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

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

7 Formal Details
- Soundness Proof
- Completeness Proof

8 Differential Algebraic Dynamic Logic DAL (Excerpt)
- Differential Invariants

9 Differential Temporal Dynamic Logic dTL (Excerpt)

10 Deduction Modulo Real Algebraic and Computer Algebraic Constraints

11 European Train Control System

12 Collision Avoidance Maneuvers in Air Traffic Control

13 Hybrid Automata Embedding

14 Distributed Hybrid Systems

15 Car Control Verification

16 Stochastic Hybrid Systems

Proof (Soundness).

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

Return

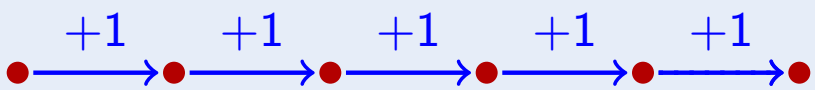
Theorem

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

Proof.

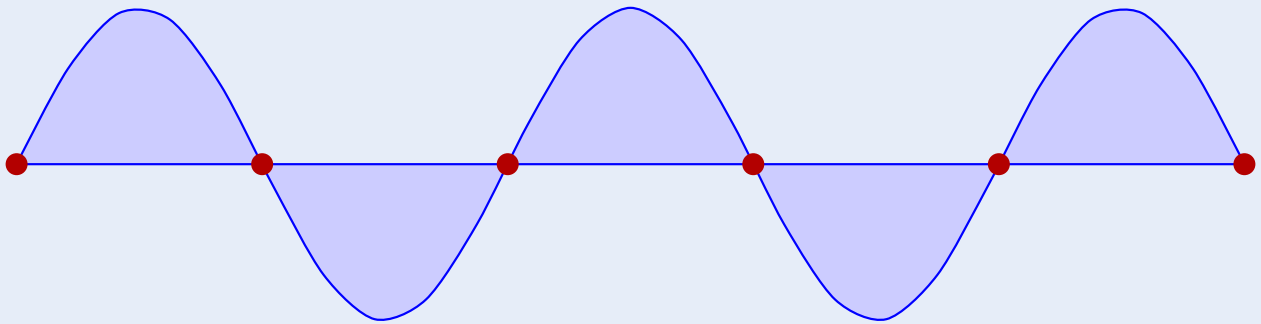
Discrete fragment:

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



Continuous fragment:

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



7 Formal Details
- Soundness Proof
- Completeness Proof

8 Differential Algebraic Dynamic Logic DAL (Excerpt)
- Differential Invariants

9 Differential Temporal Dynamic Logic dTL (Excerpt)

10 Deduction Modulo Real Algebraic and Computer Algebraic Constraints

11 European Train Control System

12 Collision Avoidance Maneuvers in Air Traffic Control

13 Hybrid Automata Embedding

14 Distributed Hybrid Systems

15 Car Control Verification

16 Stochastic Hybrid Systems

Incomplete! But are we missing proof rules?

Incomplete! But are we missing proof rules?

Relativity

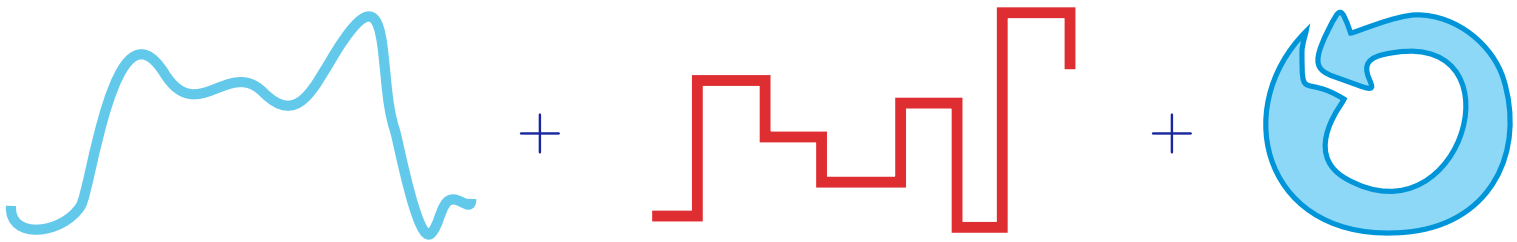
Cook,Harel: discrete-DL/data$_{\mathbb{N}}$ hybrid-d$\mathcal{L}$/data$_{\mathbb{R}}$??

Sources of Incompleteness

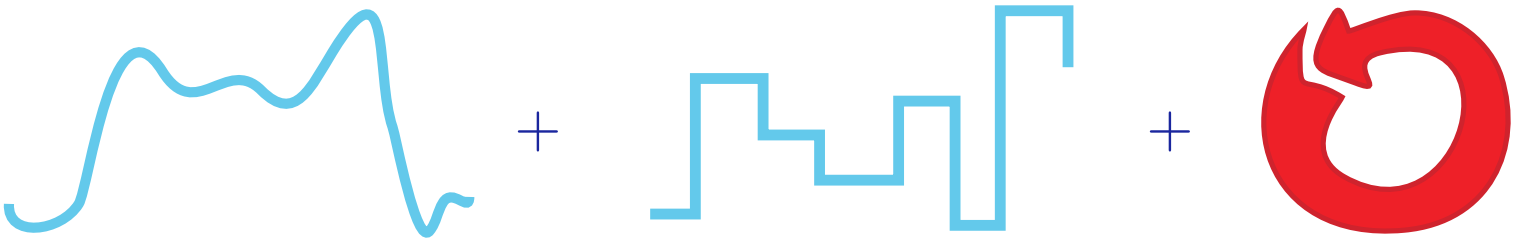
Sources of Incompleteness

continuous

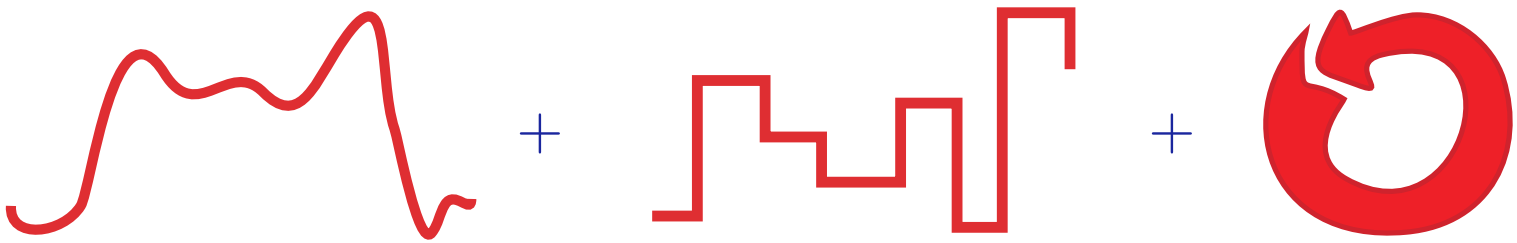
Sources of Incompleteness

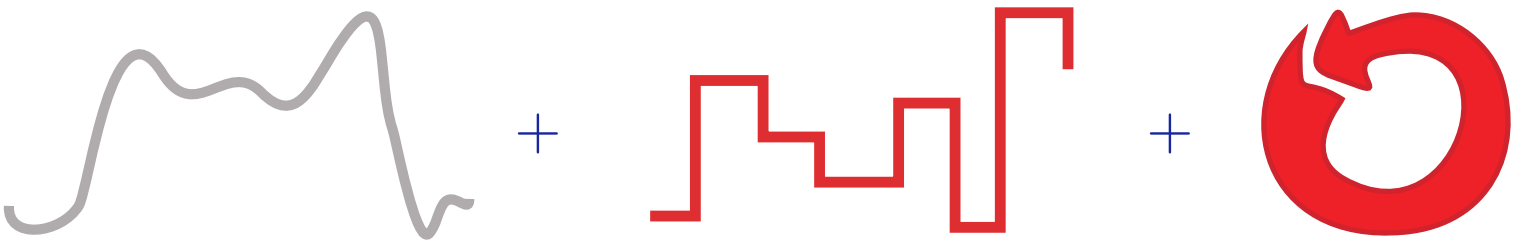


Sources of Incompleteness



Sources of Incompleteness




continuous
discrete
repeat

Relative Completeness

Theorem (Relative Completeness)

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

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

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

▸ Proof Outline 15p

continuous + discrete + repeat

$\Downarrow$

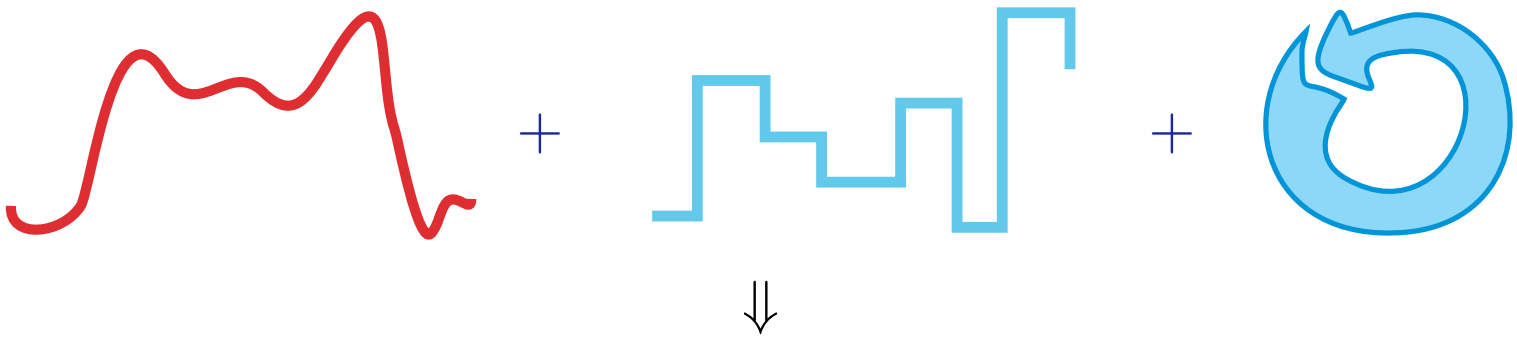
Theorem (Relative Completeness)

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

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

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

▸ Proof Outline 15p



Relativity

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

Relative Completeness

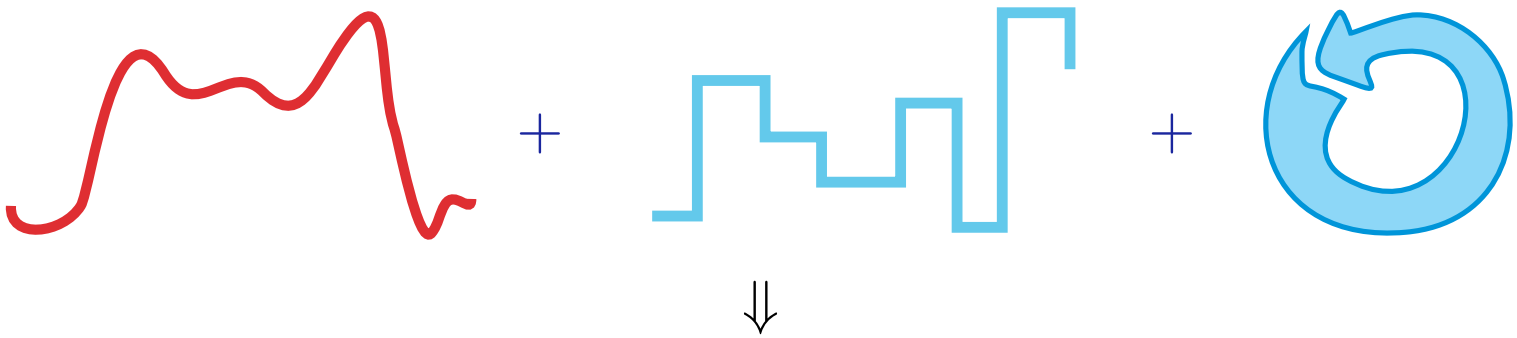
Theorem (Relative Completeness)

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

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

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

▸ Proof Outline 15p



Corollary (Proof-theoretical Alignment)

verification of hybrid systems = verification of dynamical systems!

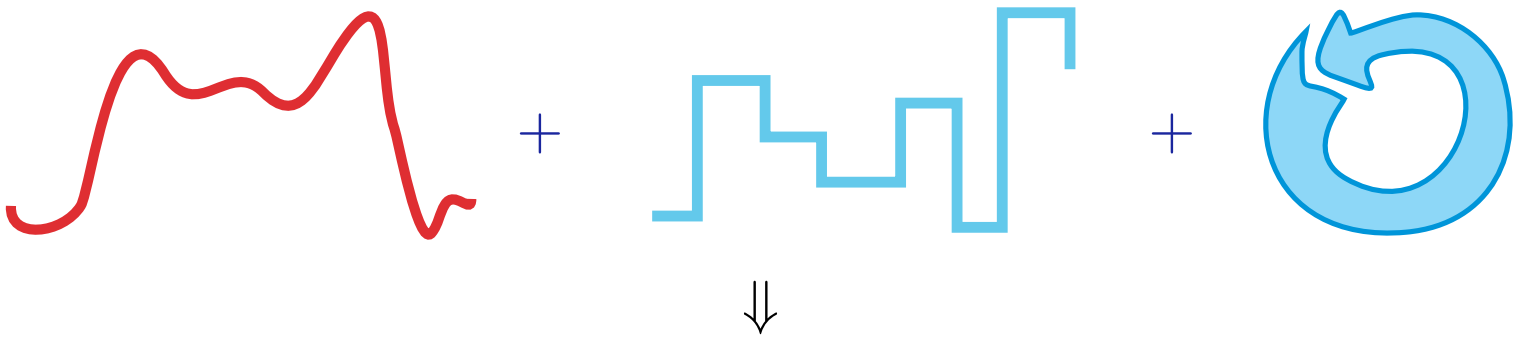
Theorem (Relative Completeness)

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

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

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

▸ Proof Outline 15p



Corollary (Deductive Power)

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

Relative Completeness Proof

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

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

Proof (Relative Completeness, 10 pages)

Return

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

Relative Completeness Proof

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

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

Proof (Relative Completeness, 10 pages)

Return

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

Relative Completeness Proof

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

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

Proof (Relative Completeness, 10 pages)

Return

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

Relative Completeness Proof

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

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

Proof (Relative Completeness, 10 pages)

Return

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

Relative Completeness Proof

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

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

Proof (Relative Completeness, 10 pages)

Return

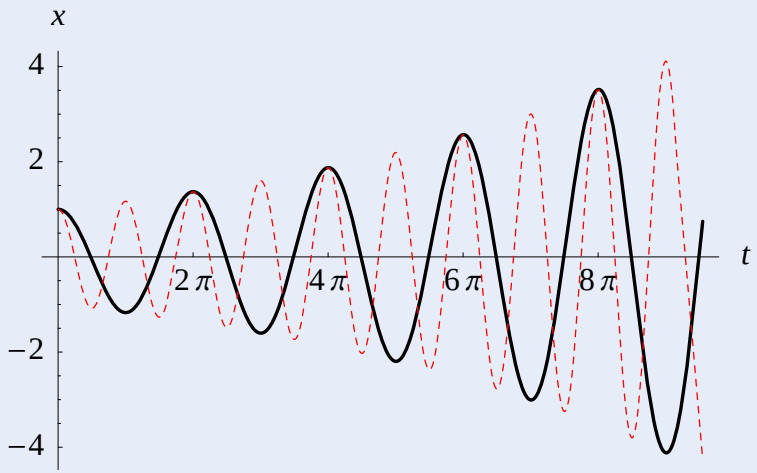
1. Strong invariants and variants expressible in d$\mathcal{L}$
2. d$\mathcal{L}$ expressible in FOD
3. valid d$\mathcal{L}$ formulas d$\mathcal{L}$-derivable from corresponding FOD axioms
4. finite FOD formula characterising unbounded hybrid repetition
5. FOD characterises $\mathbb{R}$-Gödel encoding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

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

Proof ($\mathbb{R}$-Gödel encoding)

Return

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

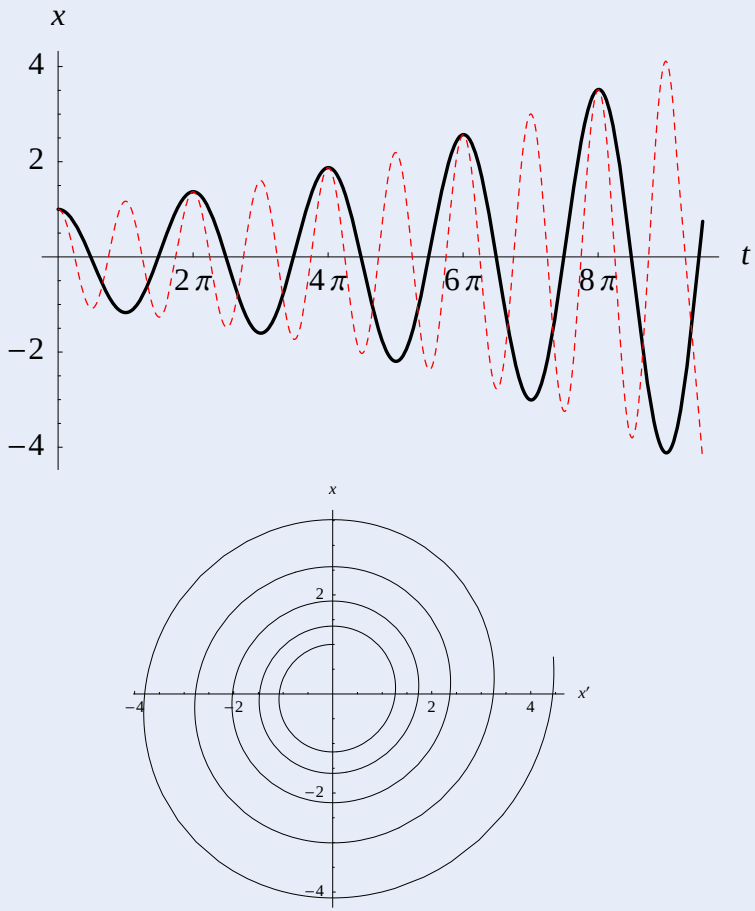


where $\mathsf{FOD} = \mathsf{FOL}_{\mathbb{R}} + [x_1' = \theta_1, \ldots, x_n' = \theta_n]F$

Proof ($\mathbb{R}$-Gödel encoding)

Return

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

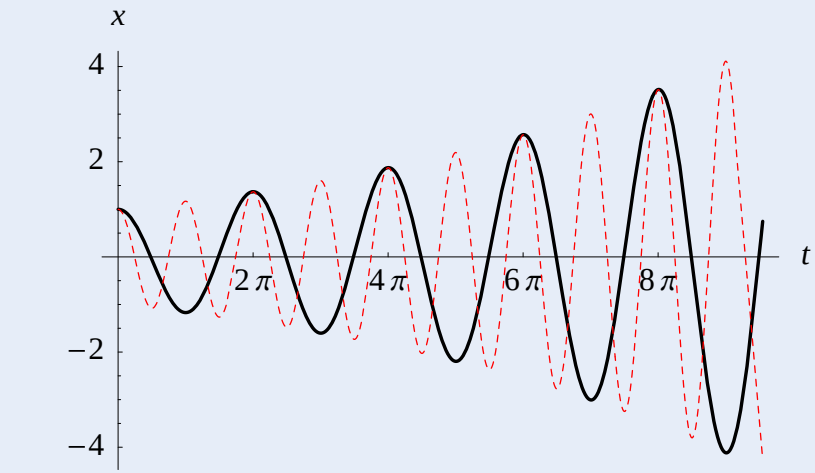


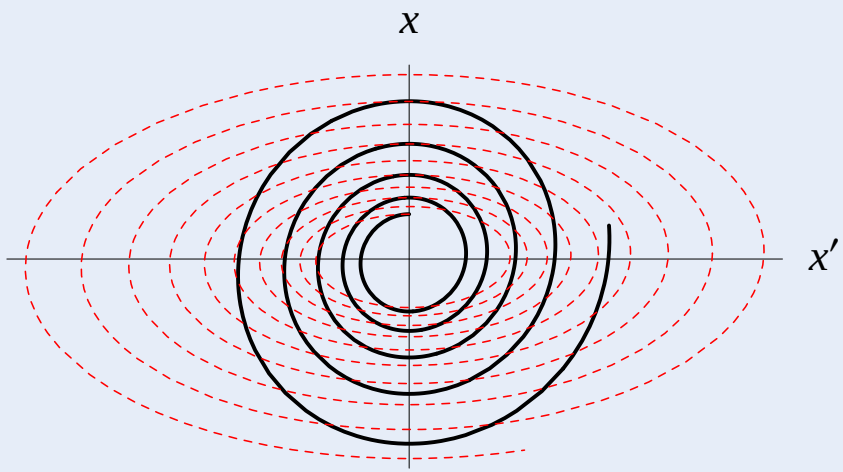
where $\text{FOD} = \text{FOL}_{\mathbb{R}} + [x_1' = \theta_1, \ldots, x_n' = \theta_n]F$

Proof ($\mathbb{R}$-Gödel encoding)

Return

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



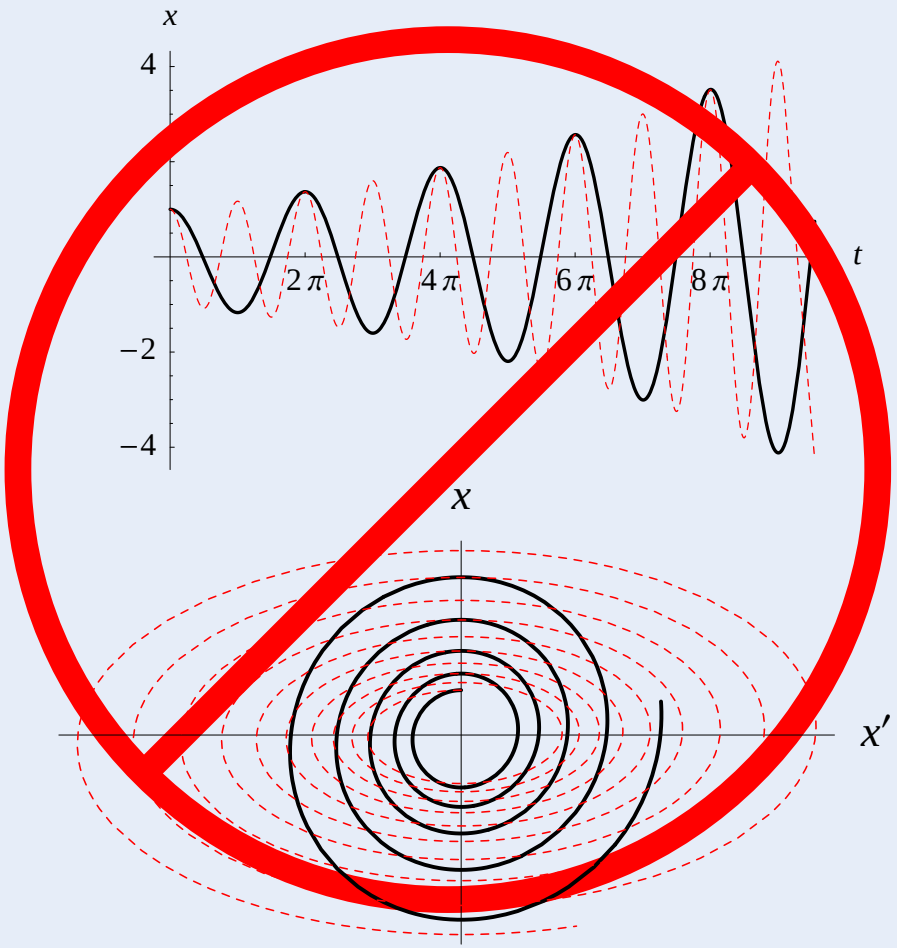


where $\text{FOD} = \text{FOL}_{\mathbb{R}} + [x_1' = \theta_1, \ldots, x_n' = \theta_n]F$

Proof ($\mathbb{R}$-Gödel encoding)

Return

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



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

Proof ($\mathbb{R}$-Gödel encoding)

Return

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

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

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

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

Proof ($\mathbb{R}$-Gödel encoding)

Return

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

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

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

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

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

7 Formal Details
- Soundness Proof
- Completeness Proof

8 Differential Algebraic Dynamic Logic DAL (Excerpt)
- Differential Invariants

9 Differential Temporal Dynamic Logic dTL (Excerpt)

10 Deduction Modulo Real Algebraic and Computer Algebraic Constraints

11 European Train Control System

12 Collision Avoidance Maneuvers in Air Traffic Control

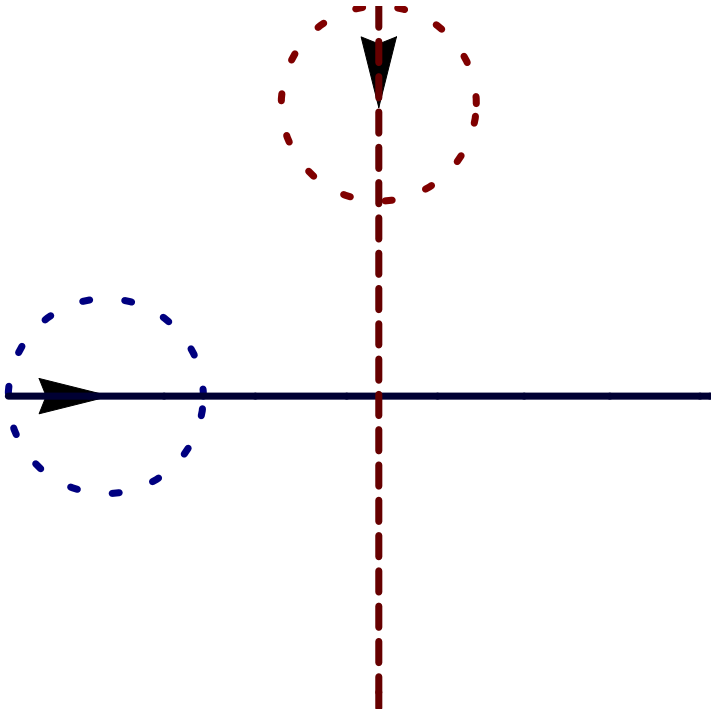
13 Hybrid Automata Embedding

14 Distributed Hybrid Systems

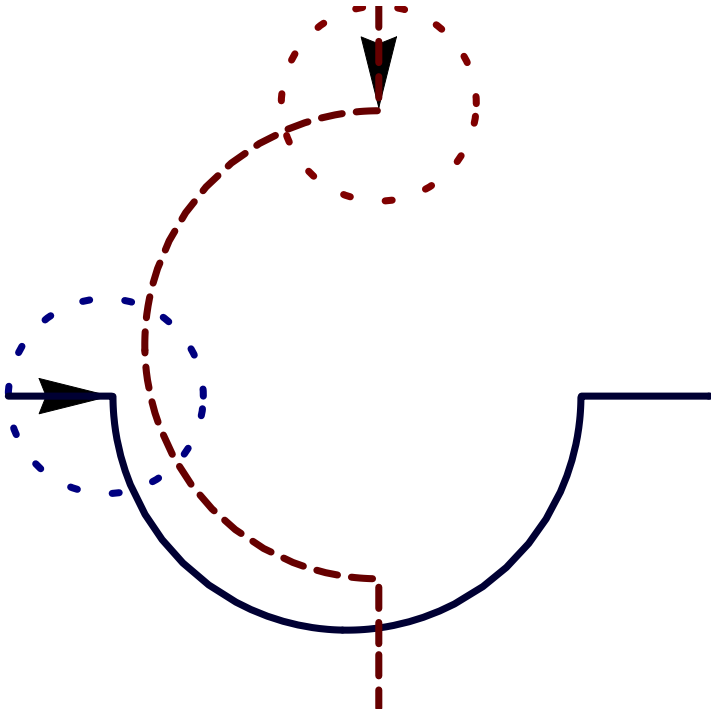
15 Car Control Verification

16 Stochastic Hybrid Systems

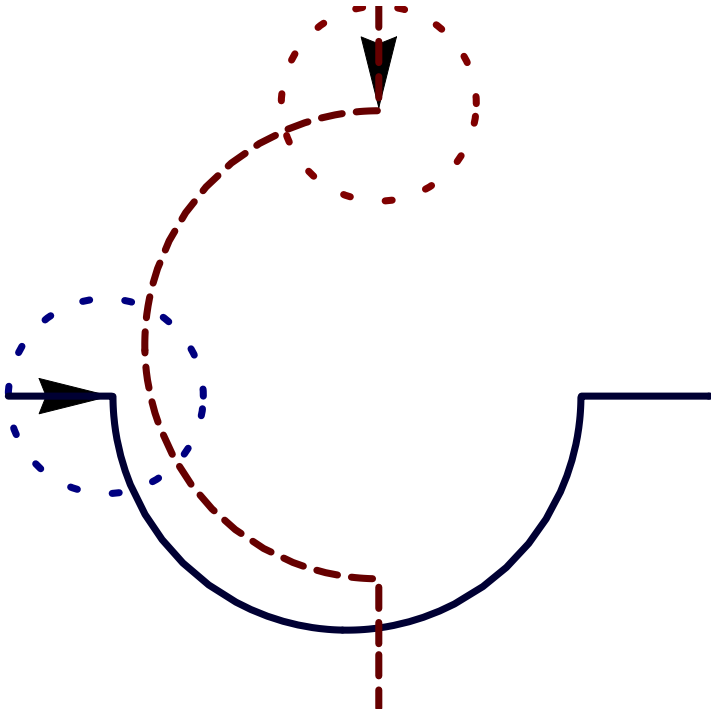
Air Traffic Control



Air Traffic Control



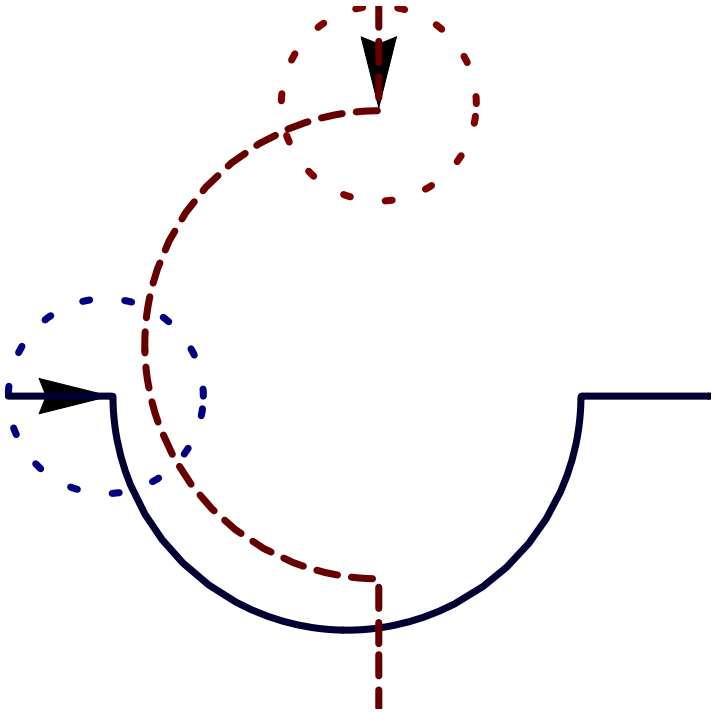
Air Traffic Control

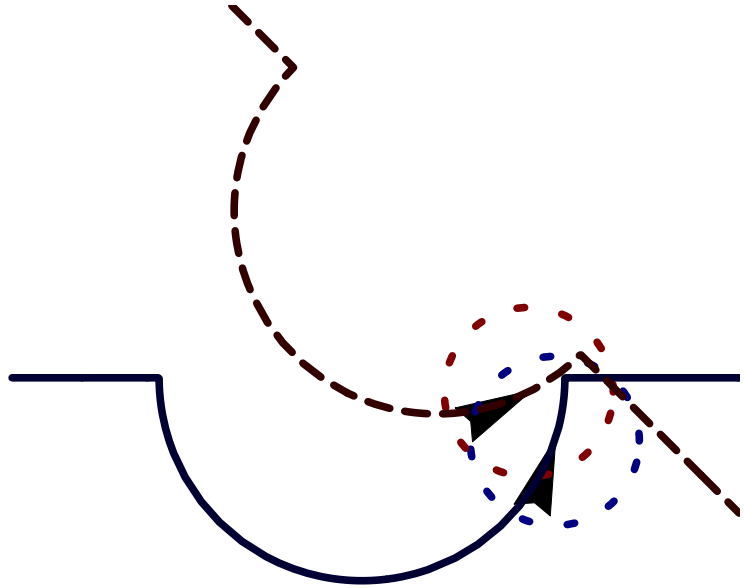


Verification?

looks correct

Air Traffic Control

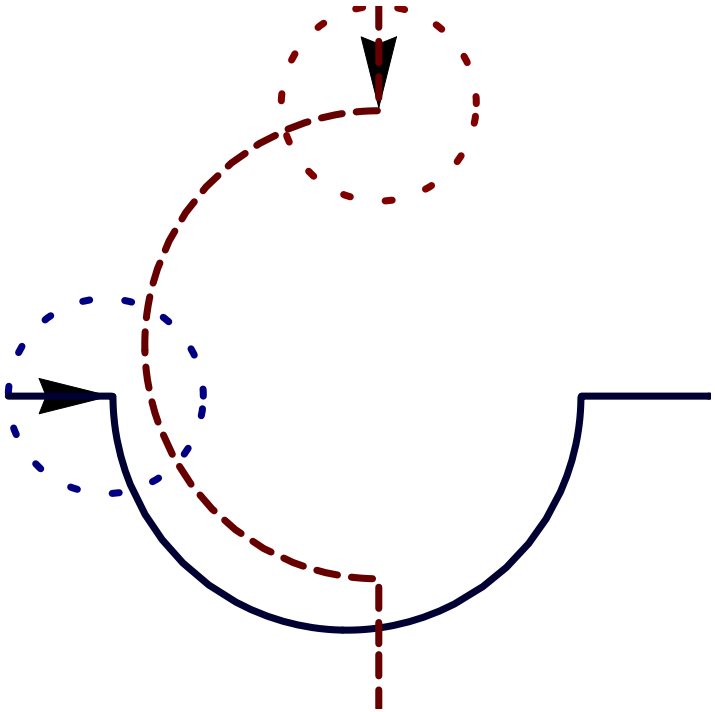


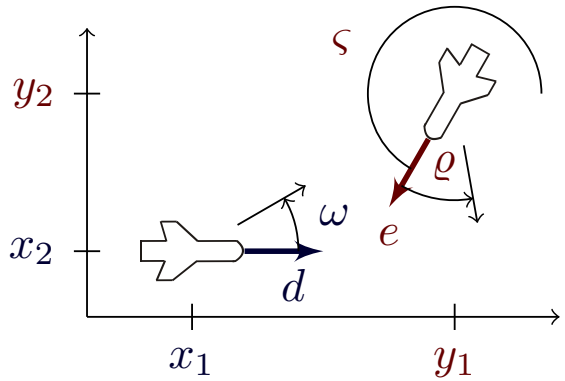


Verification?

looks correct NO!

Air Traffic Control

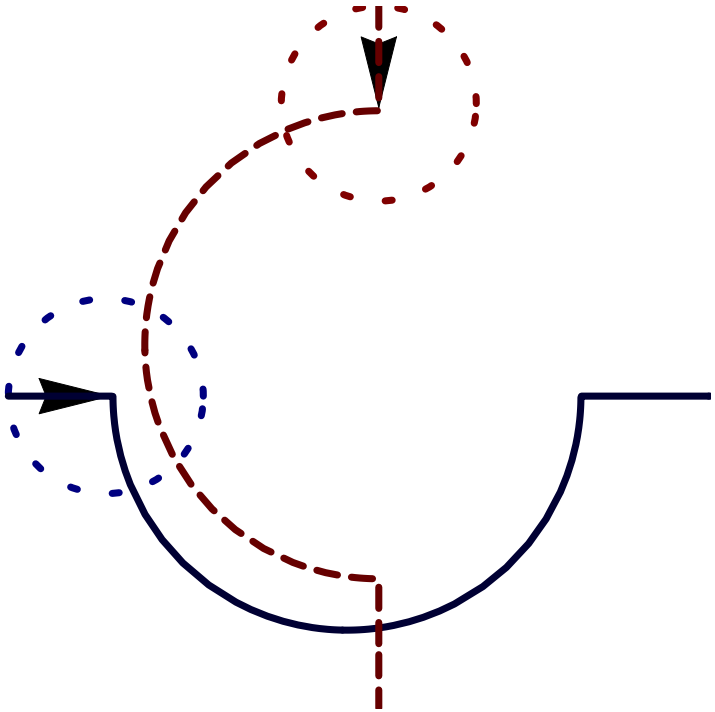


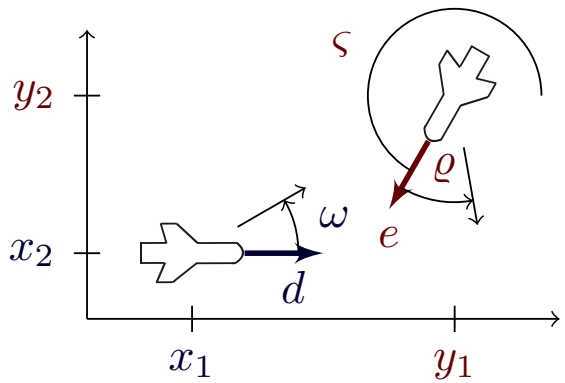


$$\begin{bmatrix} x_1' = -v_1 + v_2 \cos\vartheta + \omega x_2 \\ x_2' = v_2 \sin\vartheta - \omega x_1 \\ \vartheta' = \varpi - \omega \end{bmatrix}$$

Verification?

looks correct NO!



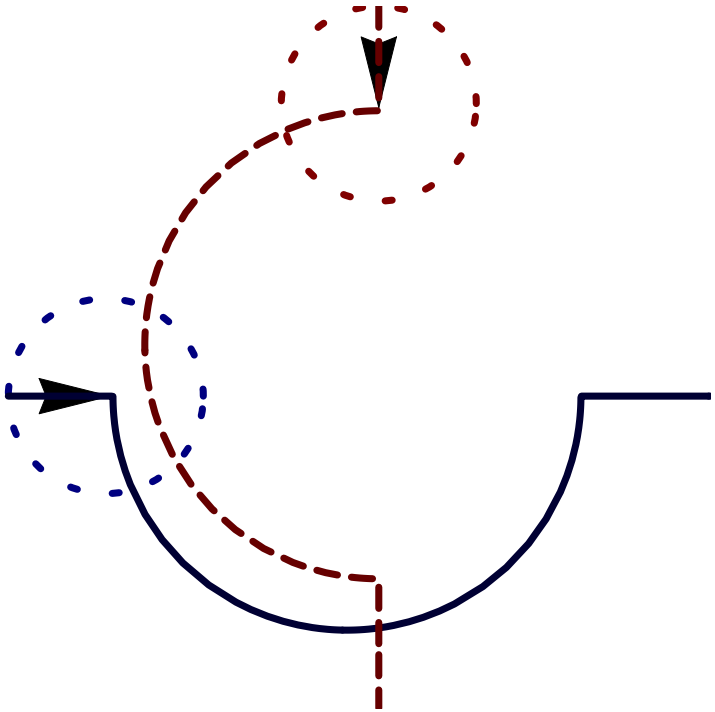


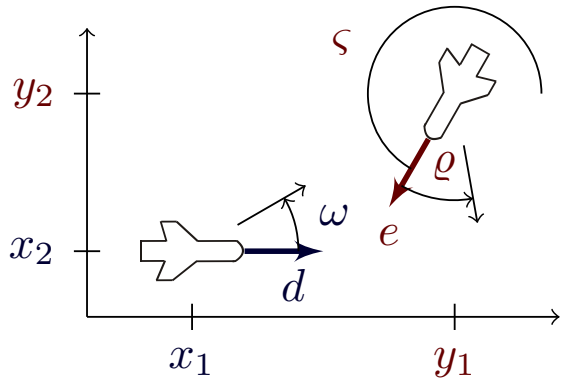
$$\begin{bmatrix} x_1' = -v_1 + v_2 \cos\vartheta + \omega x_2 \\ x_2' = v_2 \sin\vartheta - \omega x_1 \\ \vartheta' = \varpi - \omega \end{bmatrix}$$

Example ("Solving" differential equations)

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

Air Traffic Control





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

Example ("Solving" differential equations)

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

7 Formal Details
- Soundness Proof
- Completeness Proof

8 Differential Algebraic Dynamic Logic DAL (Excerpt)
- Differential Invariants

9 Differential Temporal Dynamic Logic dTL (Excerpt)

10 Deduction Modulo Real Algebraic and Computer Algebraic Constraints

11 European Train Control System

12 Collision Avoidance Maneuvers in Air Traffic Control

13 Hybrid Automata Embedding

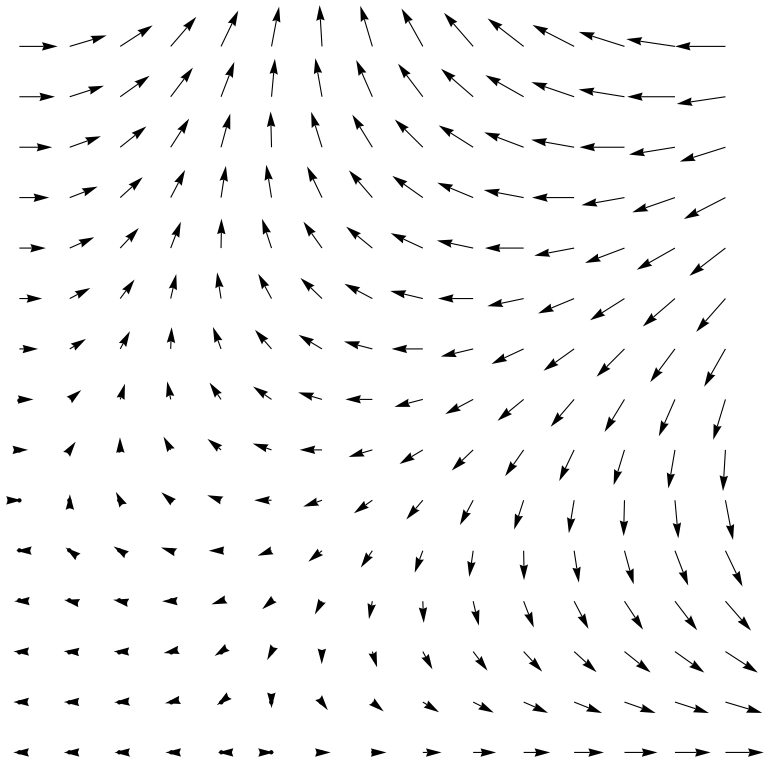
14 Distributed Hybrid Systems

15 Car Control Verification

16 Stochastic Hybrid Systems

"Definition" (Differential Invariant)

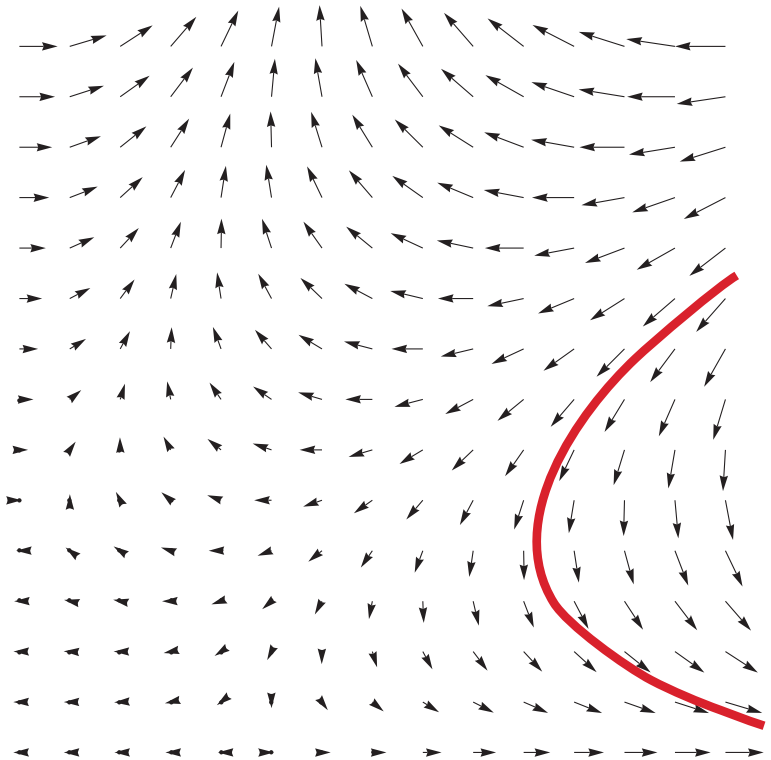
"Formula that remains true in the direction of the dynamics"



JLogComput'10,CAV'08,FMSD'09,LMCS'12

"Definition" (Differential Invariant)

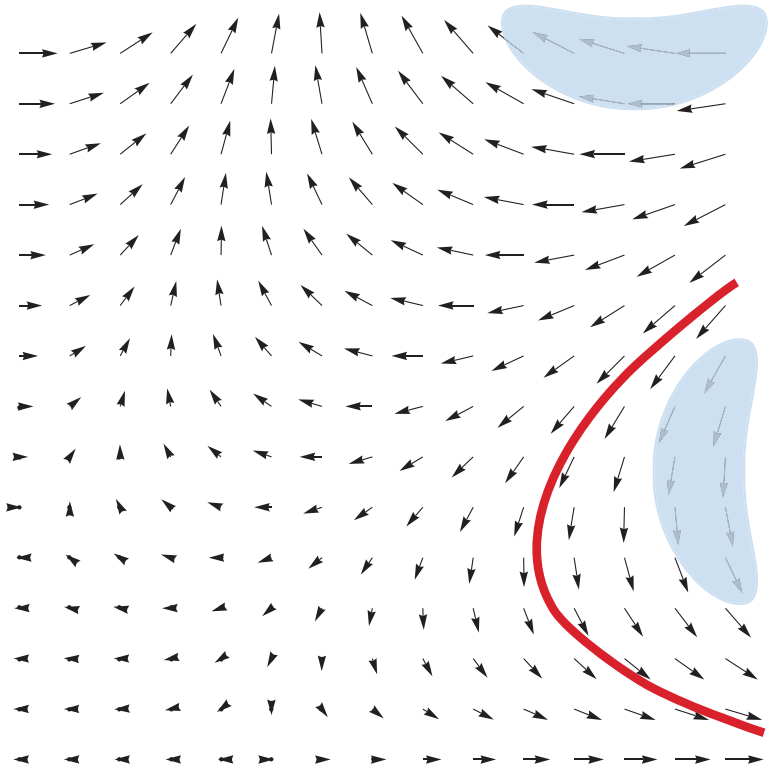
"Formula that remains true in the direction of the dynamics"



JLogComput'10,CAV'08,FMSD'09,LMCS'12

"Definition" (Differential Invariant)

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



JLogComput'10,CAV'08,FMSD'09,LMCS'12

7 Formal Details
- Soundness Proof
- Completeness Proof

8 Differential Algebraic Dynamic Logic DAL (Excerpt)
- Differential Invariants

9 Differential Temporal Dynamic Logic dTL (Excerpt)

10 Deduction Modulo Real Algebraic and Computer Algebraic Constraints

11 European Train Control System

12 Collision Avoidance Maneuvers in Air Traffic Control

13 Hybrid Automata Embedding

14 Distributed Hybrid Systems

15 Car Control Verification

16 Stochastic Hybrid Systems

Temporal Modalities + Dynamic Modalities

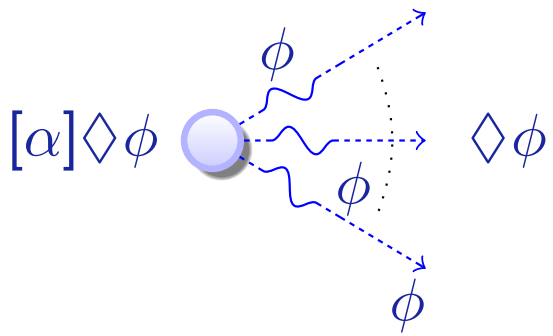
problem	technique	Op	Par	T	closed
$train \models z < M$	TL-MC	✓	×	✓	×
$\models (\mathrm{Ax}(train) \rightarrow z < M)$	TL-calculus	×	...	✓	...
$\models [train]\, z < M$	DL-calculus	✓	✓	×	✓
$\models [train]\Box\, z < M$	dTL-calculus	✓	✓	✓	✓

Temporal Modalities + Dynamic Modalities

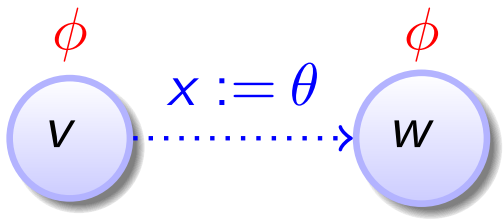
problem	technique	Op	Par	T	closed
$train \models z < M$	TL-MC	✓	×	✓	×
$\models (\mathrm{Ax}(train) \rightarrow z < M)$	TL-calculus	×	...	✓	...
$\models [train]\, z < M$	DL-calculus	✓	✓	×	✓
$\models [train]\Box\, z < M$	dTL-calculus	✓	✓	✓	✓

differential temporal dynamic logic

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

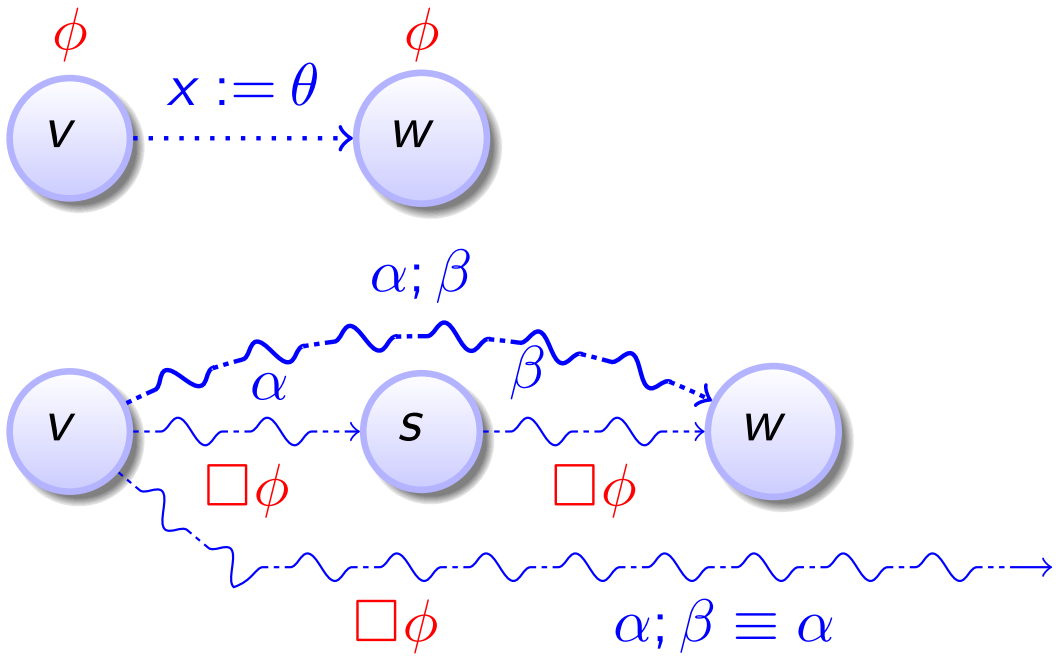


$$\frac{\phi \wedge [x := \theta]\phi}{[x := \theta]\Box\phi}$$



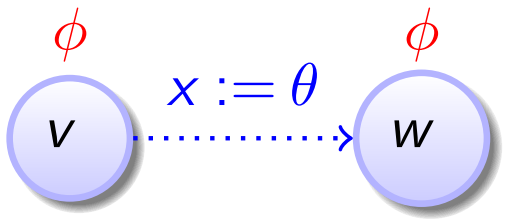
$$\frac{\phi \wedge [x := \theta]\phi}{[x := \theta]\Box\phi}$$

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

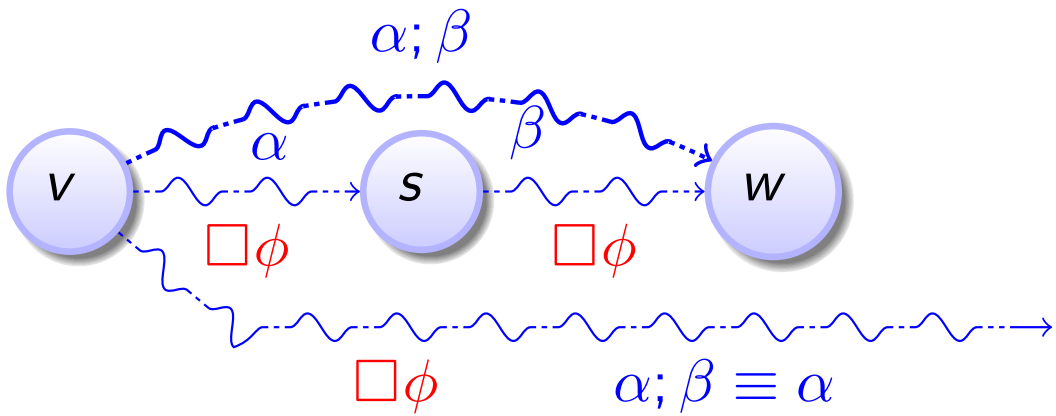


Proof Calculus for Temporal dTL

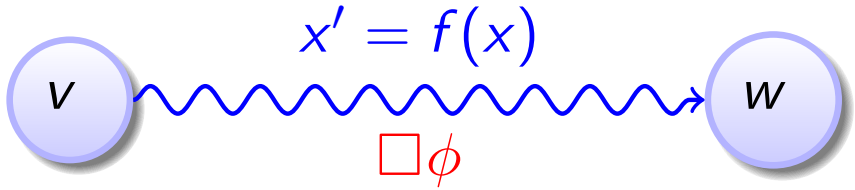
$$\frac{\phi \wedge [x := \theta]\phi}{[x := \theta]\Box\phi}$$



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

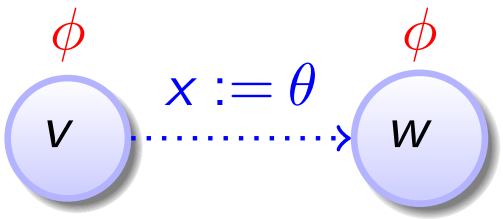


$$\frac{[x' = \theta]\phi}{[x' = \theta]\Box\phi}$$

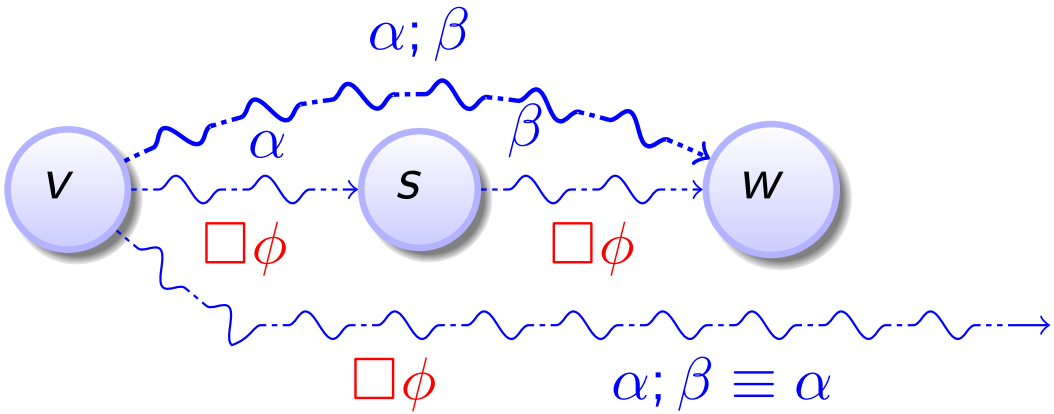


Proof Calculus for Temporal dTL

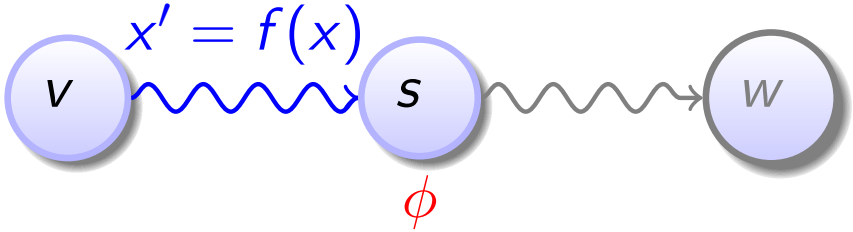
$$\frac{\phi \wedge [x := \theta]\phi}{[x := \theta]\Box\phi}$$



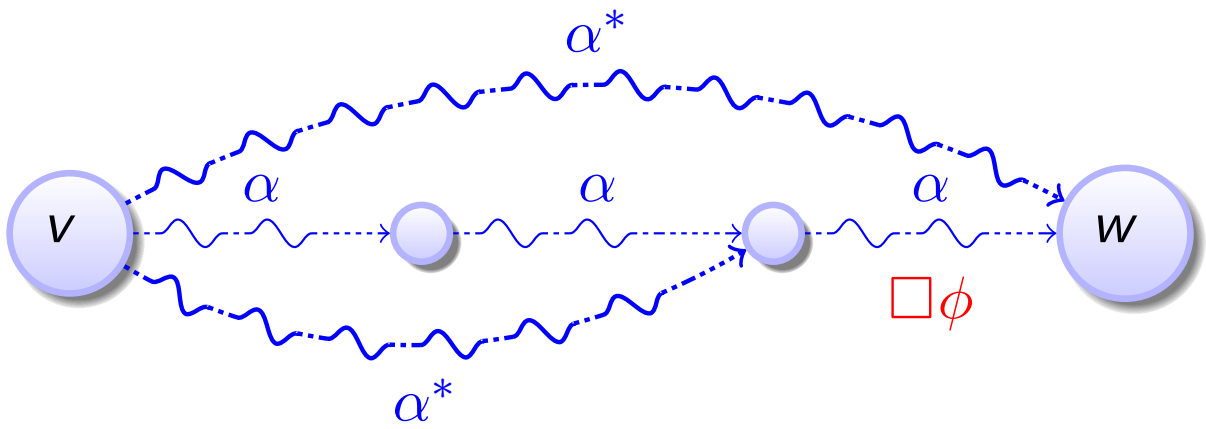
$$\frac{[\alpha]\Box\phi \wedge [\alpha][\beta]\Box\phi}{[\alpha;\beta]\Box\phi}$$



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



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



Theorem (Relative Completeness) (P. 2008)

dTL *calculus is a sound & complete axiomatization relative to* d$\mathcal{L}$.

Corollary (Continuous Relative Completeness)

dTL *calculus is a sound & complete axiomatization relative to differential equations.*

Corollary (Discrete Relative Completeness)

dTL *calculus is a sound & complete axiomatization relative to discrete systems.*

7 Formal Details
- Soundness Proof
- Completeness Proof

8 Differential Algebraic Dynamic Logic DAL (Excerpt)
- Differential Invariants

9 Differential Temporal Dynamic Logic dTL (Excerpt)

10 Deduction Modulo Real Algebraic and Computer Algebraic Constraints

11 European Train Control System

12 Collision Avoidance Maneuvers in Air Traffic Control

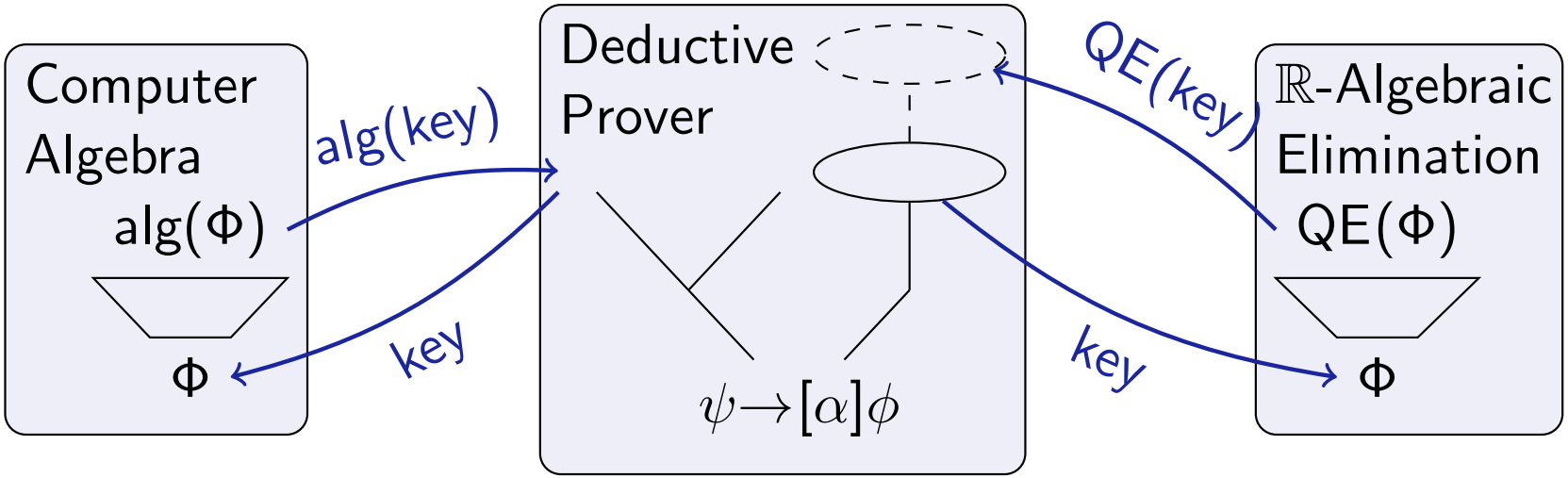
13 Hybrid Automata Embedding

14 Distributed Hybrid Systems

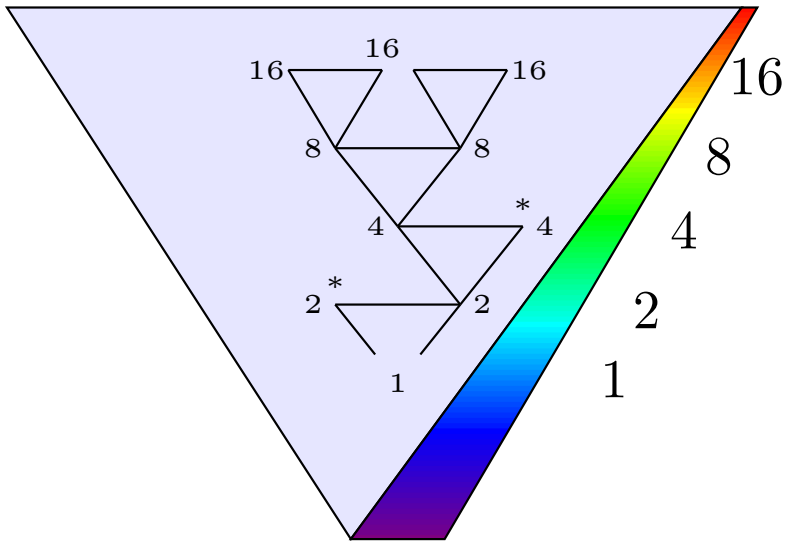
15 Car Control Verification

16 Stochastic Hybrid Systems

KeYmaera Verification Architecture

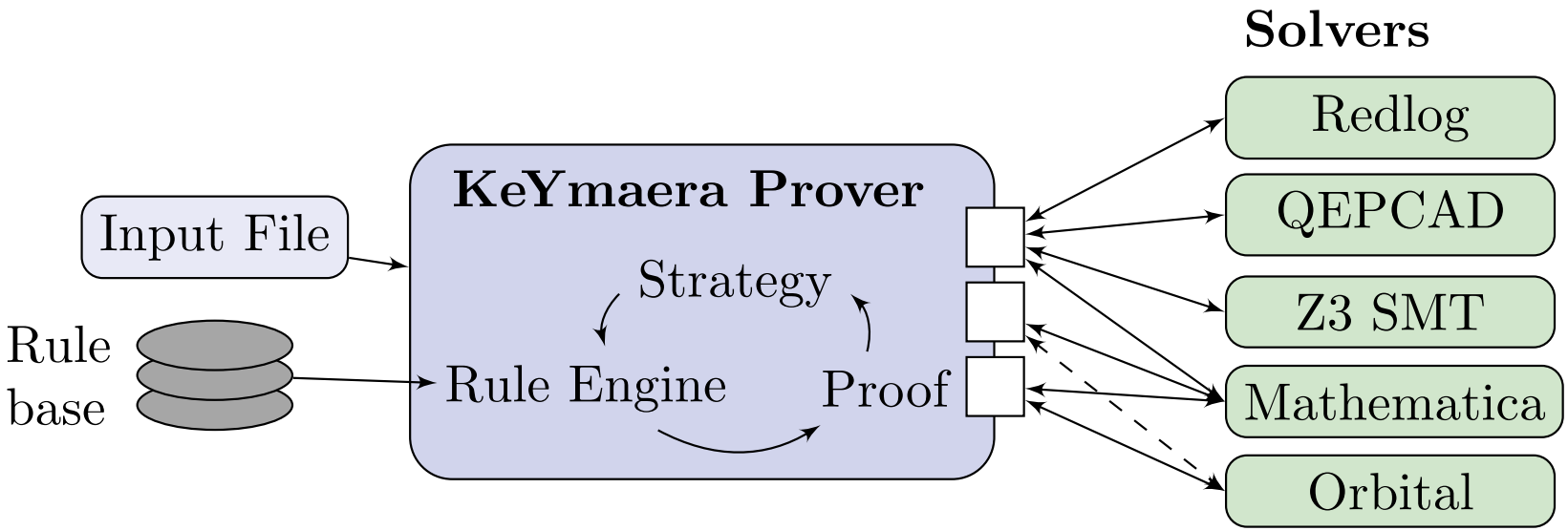


56 interactions?



0–1 interactions!

KeYmaera Prover Architecture



7 Formal Details
- Soundness Proof
- Completeness Proof

8 Differential Algebraic Dynamic Logic DAL (Excerpt)
- Differential Invariants

9 Differential Temporal Dynamic Logic dTL (Excerpt)

10 Deduction Modulo Real Algebraic and Computer Algebraic Constraints

11 European Train Control System

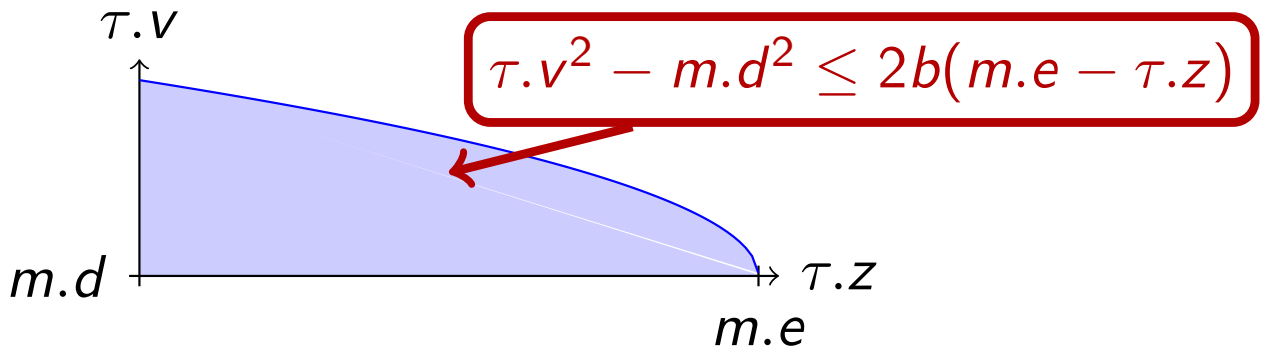
12 Collision Avoidance Maneuvers in Air Traffic Control

13 Hybrid Automata Embedding

14 Distributed Hybrid Systems

15 Car Control Verification

16 Stochastic Hybrid Systems

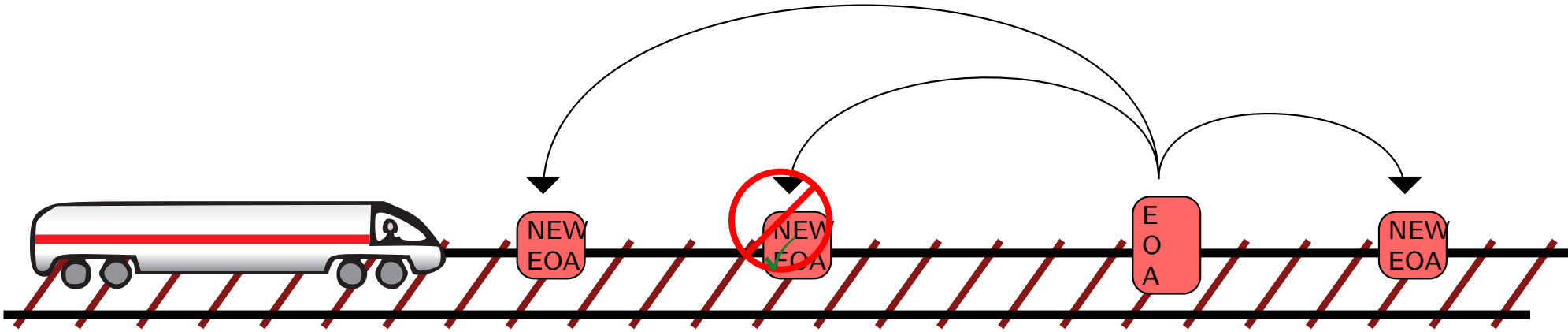


Proposition (Controllability)

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

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

ETCS RBC Controllability

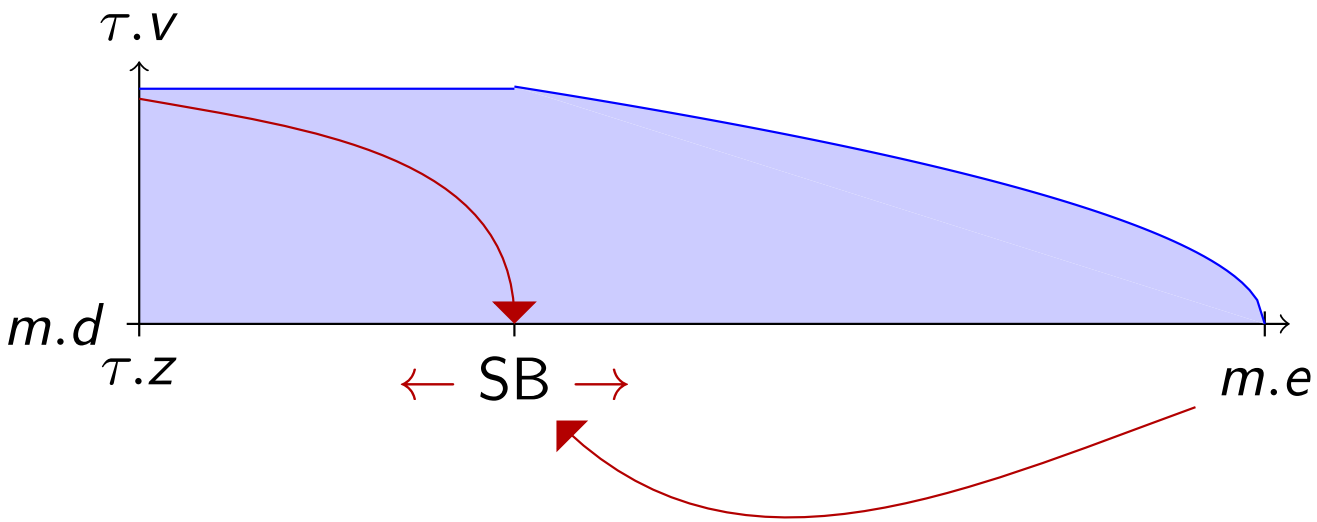


Proposition (RBC Controllability)

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

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

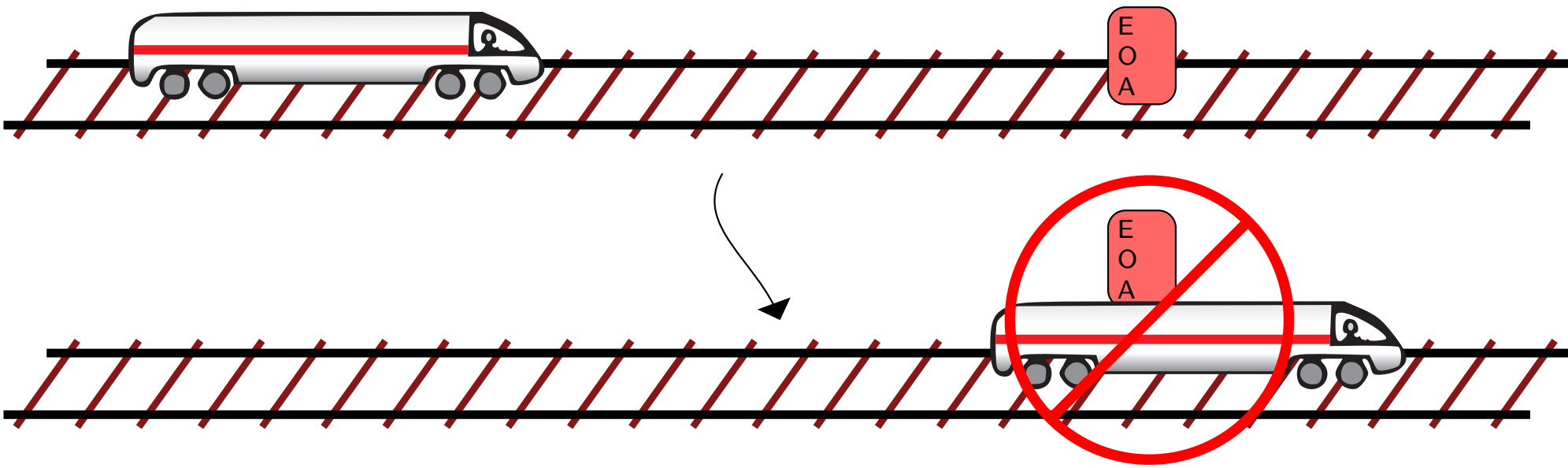
$$\big((\langle m := m_0 \rangle \tau.v^2 - m.d^2 \leq 2b(m.e - \tau.z)) \rightarrow \tau.v^2 - m.d^2 \leq 2b(m.e - \tau.z)\big)$$



Proposition (Reactivity)

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

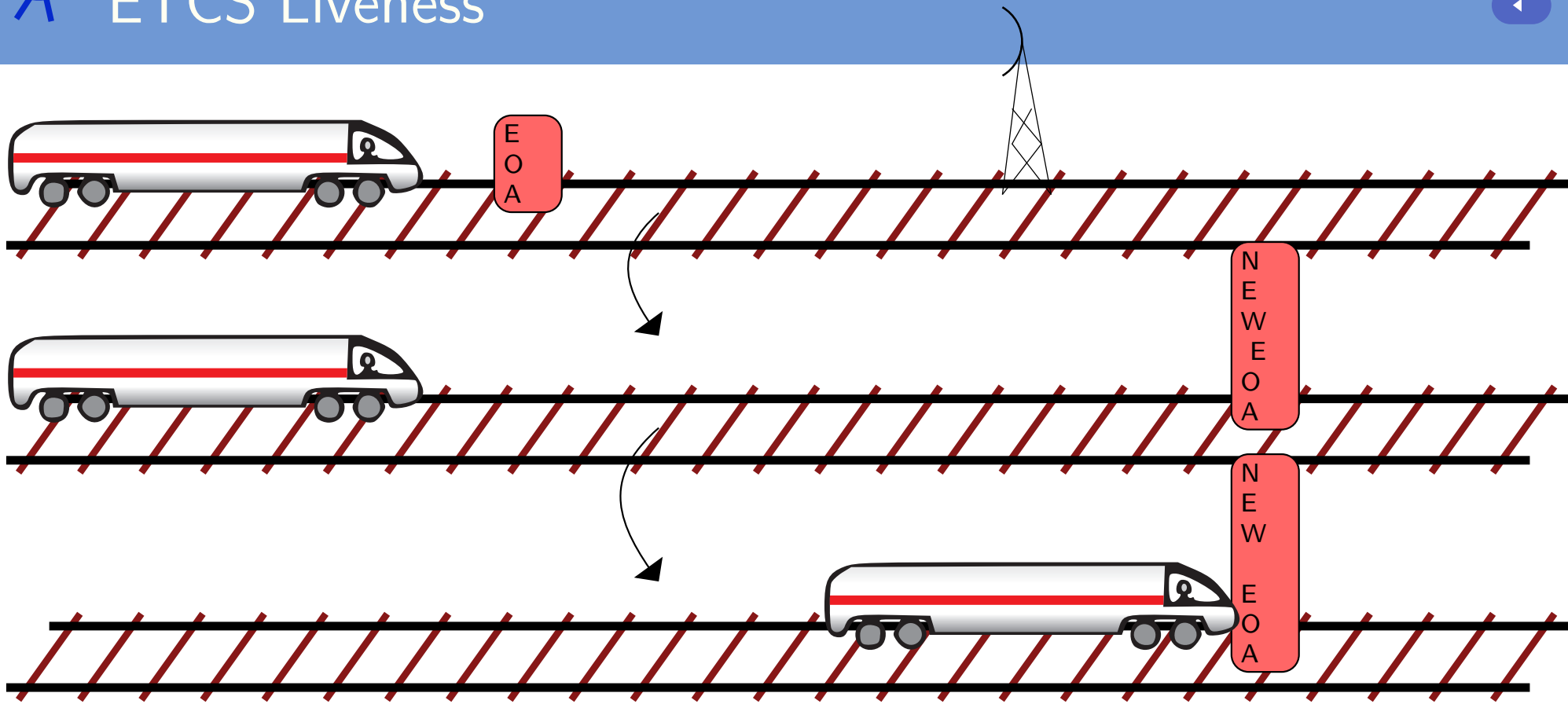
ETCS Safety



Proposition (Safety)

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

ETCS Liveness



Proposition (Liveness)

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

Safety Despite Disturbances

So far: no wind, friction, etc.

Direct control of the acceleration

Safety Despite Disturbances

So far: no wind, friction, etc.

Direct control of the acceleration

Issue

This is unrealistic!

Safety Despite Disturbances

So far: no wind, friction, etc.

Direct control of the acceleration

Issue

This is unrealistic!

Solution Take disturbances into account.

Theorem

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

Safety Despite Disturbances

So far: no wind, friction, etc.

Direct control of the acceleration

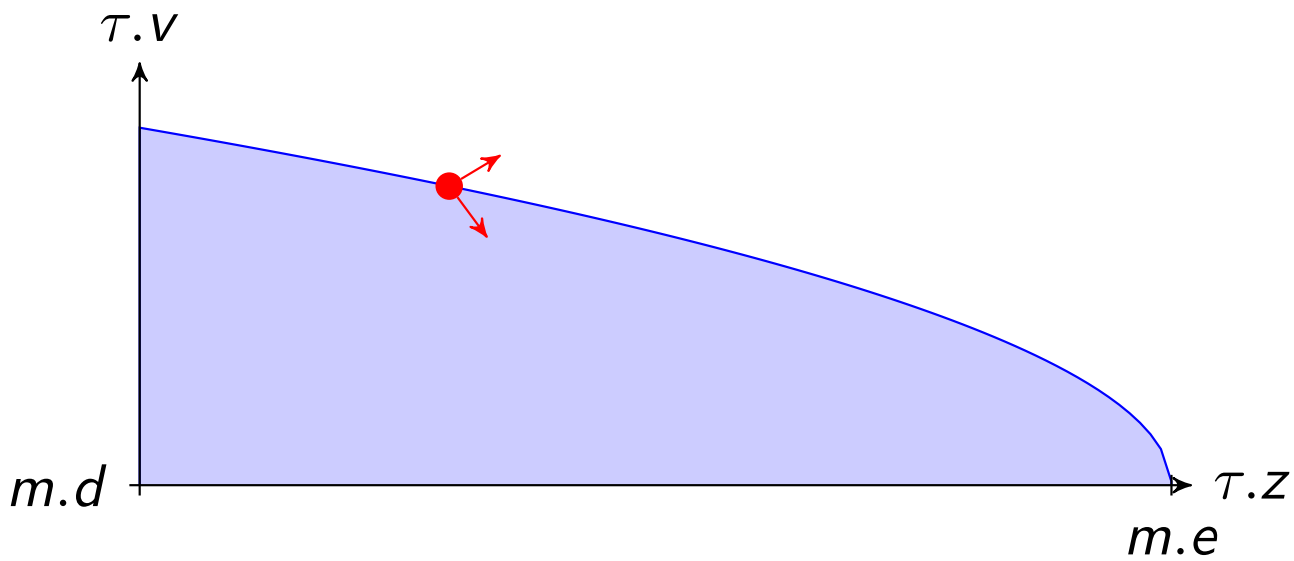
Issue

This is unrealistic!

Solution Take disturbances into account.

Theorem

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



Safety Despite Disturbances

So far: no wind, friction, etc.

Direct control of the acceleration

Issue

This is unrealistic!

Solution Take disturbances into account.

Theorem

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

Proof sketch

The system now contains $\tau.a - l \leq \tau.v' \leq \tau.a + u$ instead of $\tau.v' = \tau.a$.

$\rightsquigarrow$ We cannot solve the differential equations anymore.

$\rightsquigarrow$ Use differential invariants for approximation. For details see paper.

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

So far

Almost completely non-deterministic control.

Smooth Speed Control

So far

Almost completely non-deterministic control.

Issue

This is unrealistic!

Smooth Speed Control

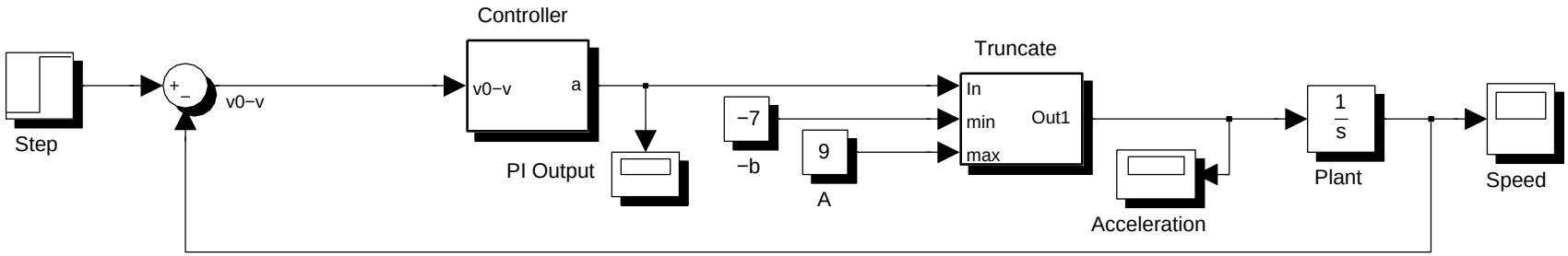
So far

Almost completely non-deterministic control.

Issue

This is unrealistic!

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



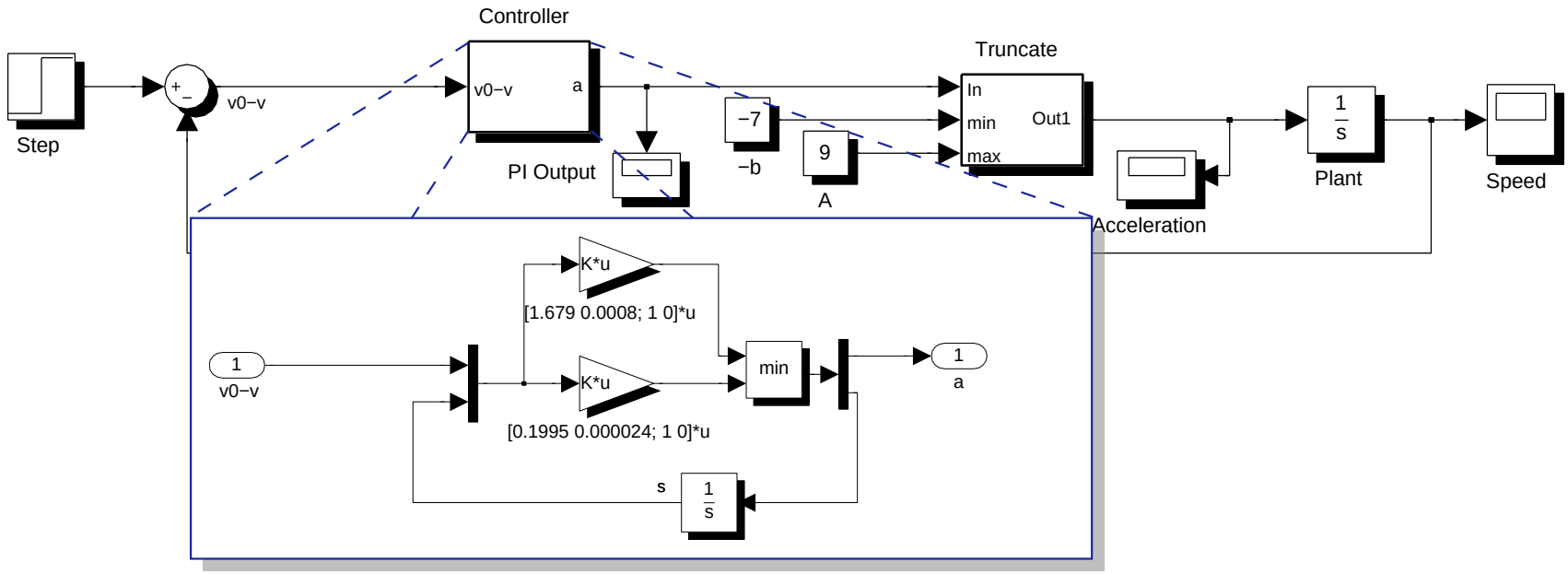
So far

Almost completely non-deterministic control.

Issue

This is unrealistic!

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



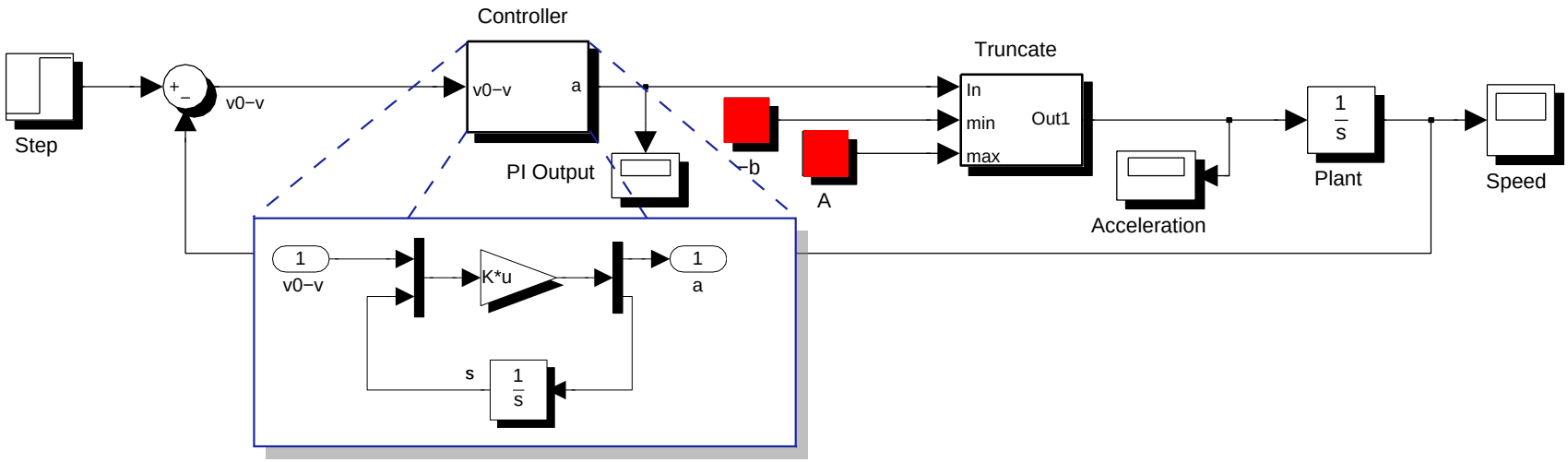
So far

Almost completely non-deterministic control.

Issue

This is unrealistic!

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



Differential equation system

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

Smooth Speed Control

So far

Almost completely non-deterministic control.

Issue

This is unrealistic!

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

Theorem

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

Proof sketch

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

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

Experimental Results (ETCS)

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

European Train Control System (ETCS)

provable automatically!

$$
\begin{array}{lll}
\text{spec} & : & \tau.v^2 - \mathbf{m}.d^2 \leq 2b(\mathbf{m}.e - \tau.p) \wedge \tau.v \geq 0 \wedge \mathbf{m}.d \geq 0 \wedge b > 0 \\
& & \rightarrow\ [\text{ETCS}](\tau.p \geq \mathbf{m}.e \rightarrow \tau.v \leq \mathbf{m}.d) \\
\text{ETCS:} & & (\text{train} \cup \text{rbc})^* \\
\text{train} & : & \text{spd};\ \text{atp};\ \text{move} \\
\text{spd} & : & (?\tau.v \leq \mathbf{m}.r;\ \tau.a := *;\ ? - b \leq \tau.a \leq A) \\
& & \cup (?\tau.v \geq \mathbf{m}.r;\ \tau.a := *;\ ?0 > \tau.a \geq -b) \\
\text{atp} & : & SB := \frac{\tau.v^2 - \mathbf{m}.d^2}{2b} + \left(\frac{A}{b} + 1\right)\left(\frac{A}{2}\varepsilon^2 + \varepsilon\,\tau.v\right); \\
& & (?(\mathbf{m}.e - \tau.p \leq SB \vee rbc.message = emergency);\ \tau.a := -b) \\
& & \cup (?\mathbf{m}.e - \tau.p \geq SB \wedge rbc.message \neq emergency) \\
\text{move} : & & t := 0;\ (\tau.p' = \tau.v, \tau.v' = \tau.a, t' = 1\,\&\,\tau.v \geq 0 \wedge t \leq \varepsilon) \\
\text{rbc} & : & (rbc.message := emergency) \\
& & \cup\ \big(\mathbf{m}_0 := \mathbf{m}; \mathbf{m} := *; \\
& & \quad ?\mathbf{m}.r \geq 0 \wedge \mathbf{m}.d \geq 0 \wedge \mathbf{m}_0.d^2 - \mathbf{m}.d^2 \leq 2b(\mathbf{m}.e - \mathbf{m}_0.e)\big)
\end{array}
$$

1 Branch of ETCS Proof

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

7 Formal Details
- Soundness Proof
- Completeness Proof

8 Differential Algebraic Dynamic Logic DAL (Excerpt)
- Differential Invariants

9 Differential Temporal Dynamic Logic dTL (Excerpt)

10 Deduction Modulo Real Algebraic and Computer Algebraic Constraints

11 European Train Control System

12 Collision Avoidance Maneuvers in Air Traffic Control

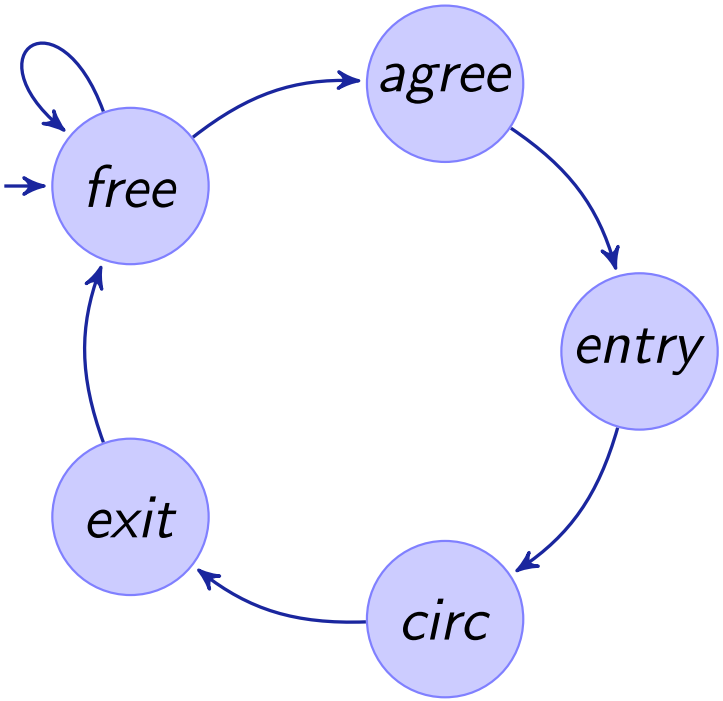
13 Hybrid Automata Embedding

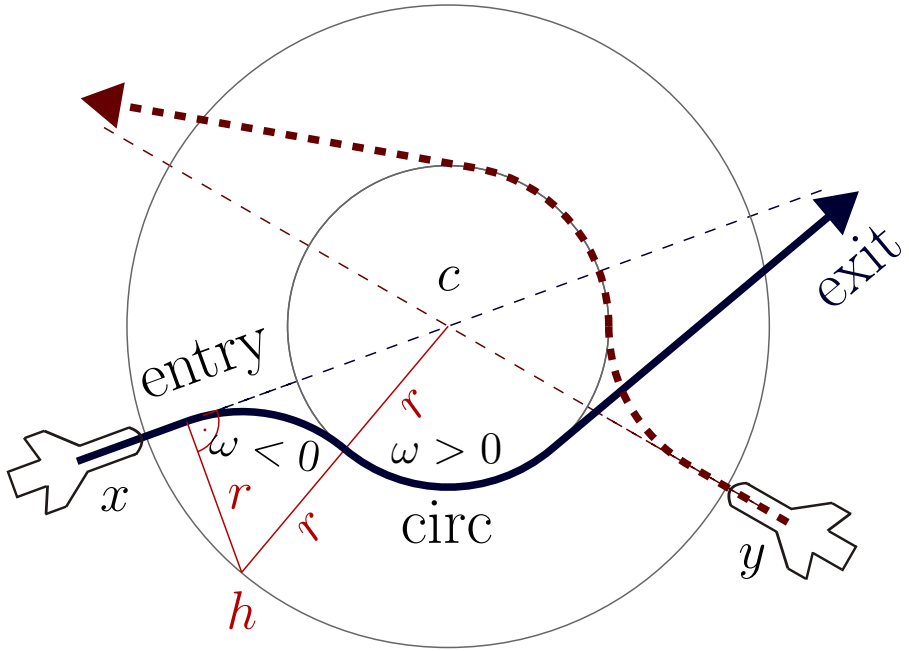
14 Distributed Hybrid Systems

15 Car Control Verification

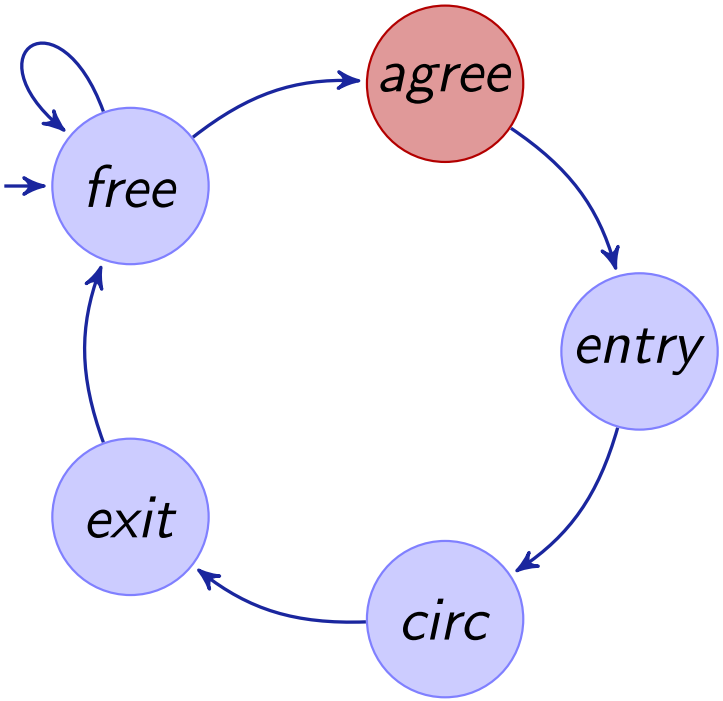
16 Stochastic Hybrid Systems

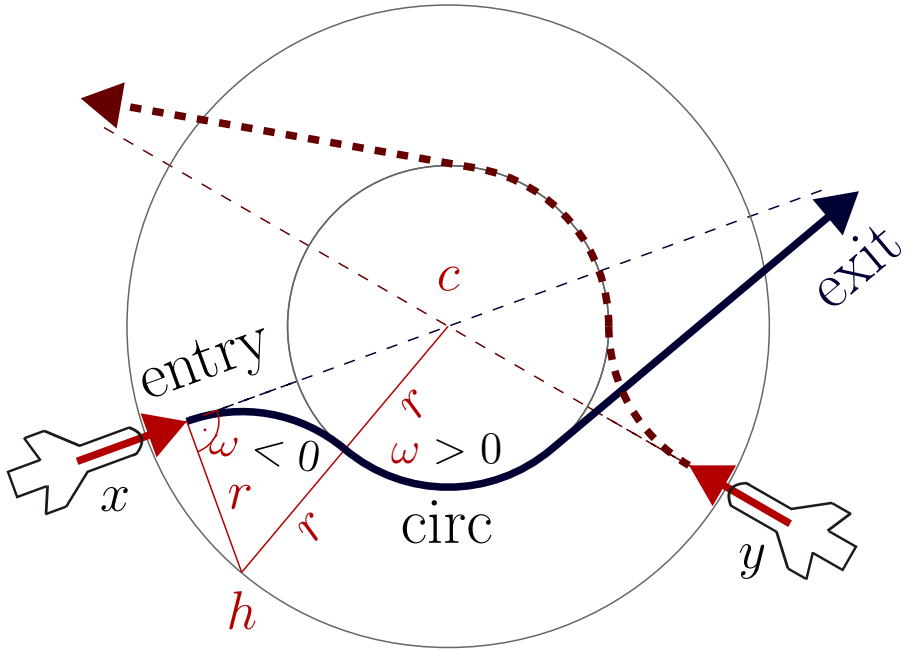
Flyable Roundabout Maneuver: Overview



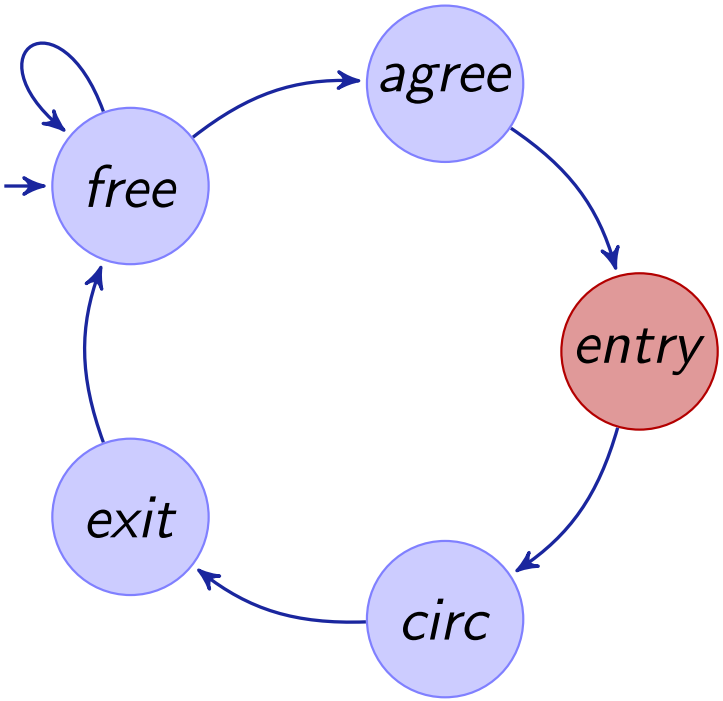


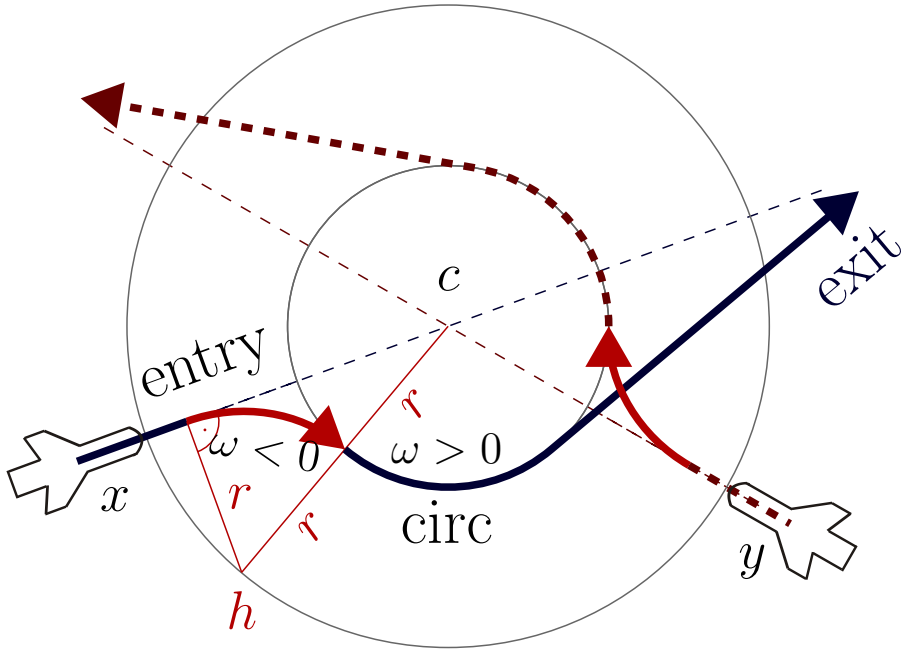
Flyable Roundabout Maneuver: Overview



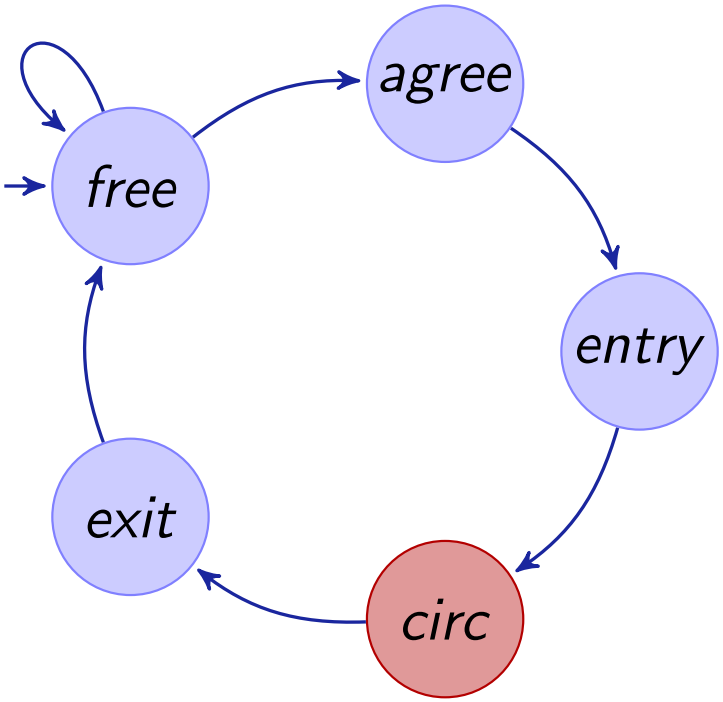


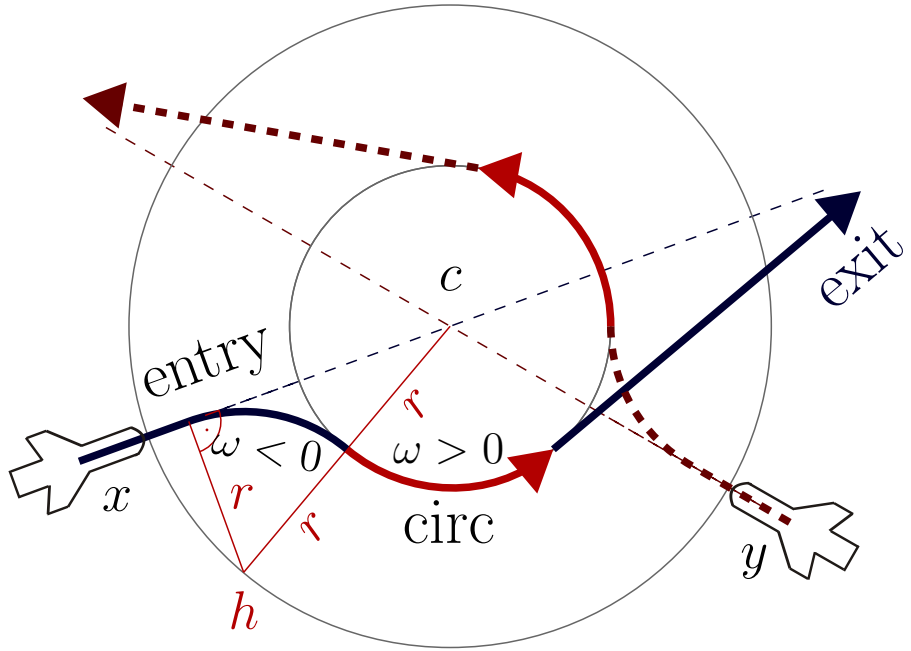
Flyable Roundabout Maneuver: Overview



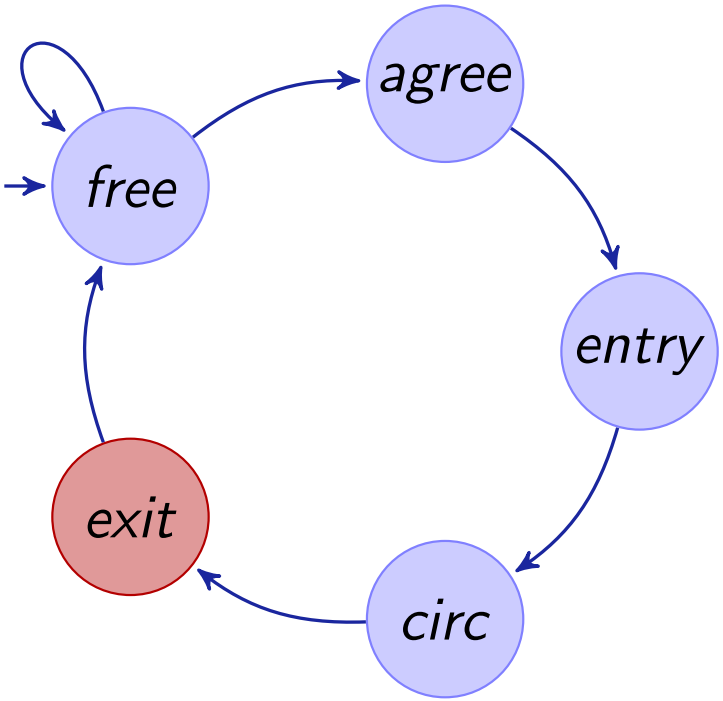


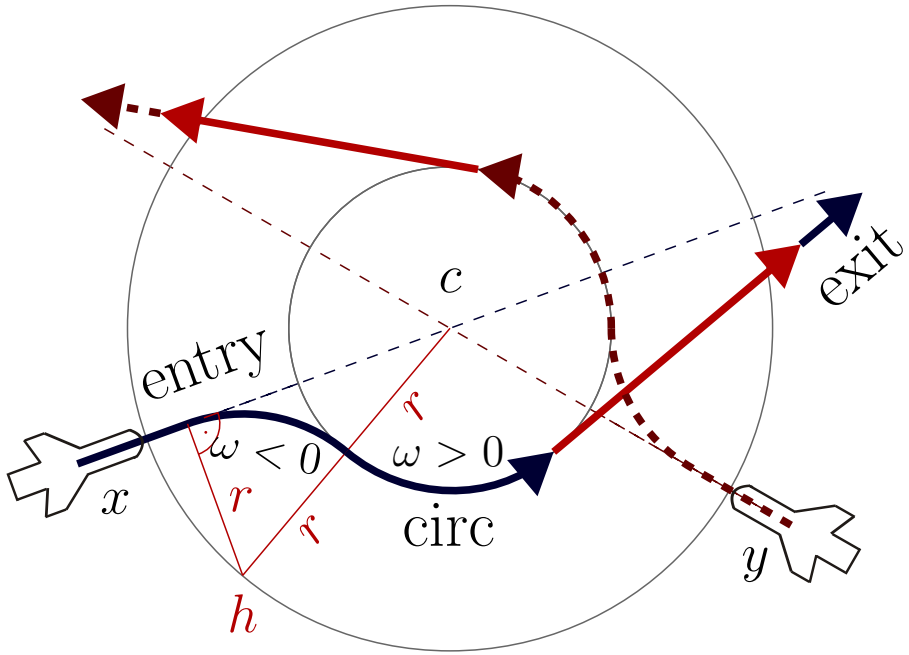
Flyable Roundabout Maneuver: Overview



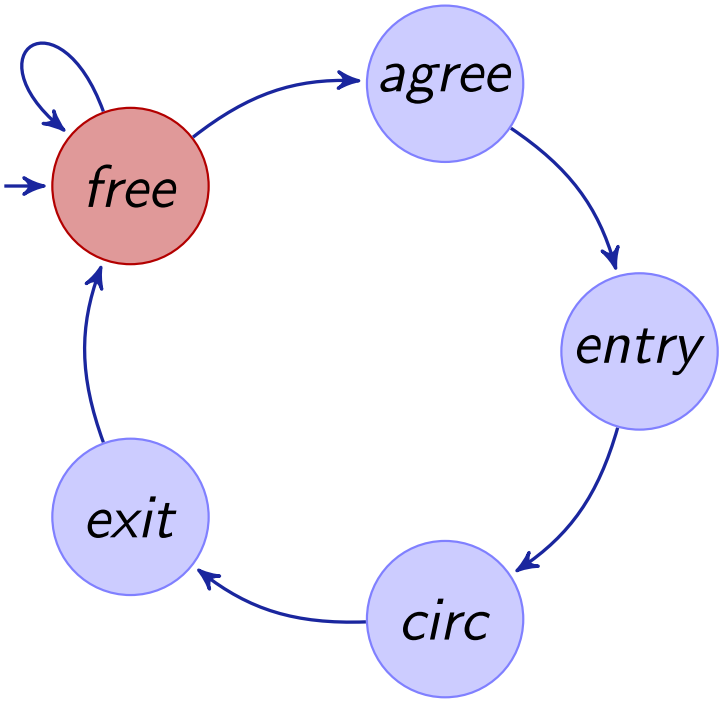


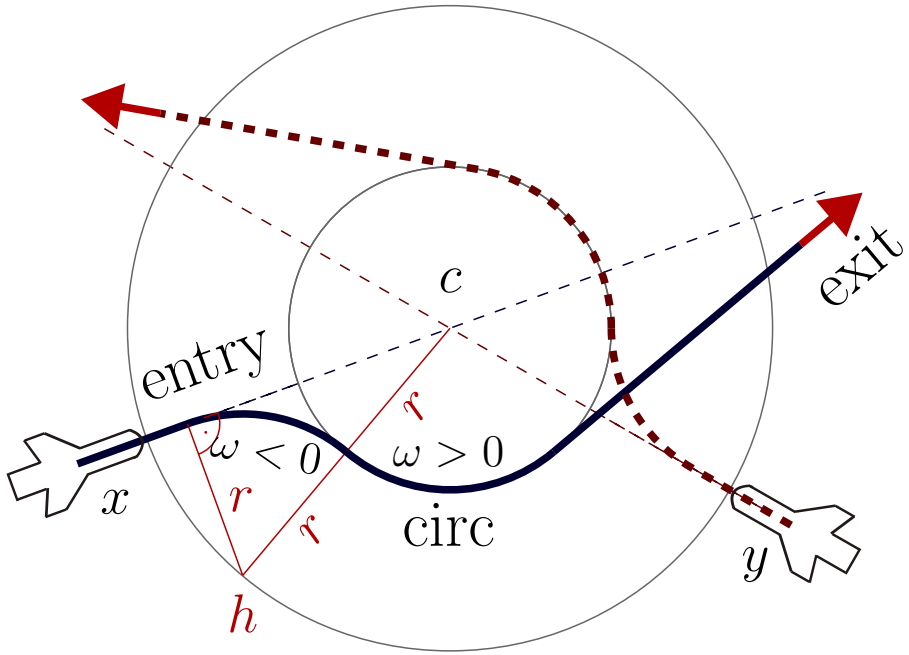
Flyable Roundabout Maneuver: Overview





Flyable Roundabout Maneuver: Overview





7 Formal Details
- Soundness Proof
- Completeness Proof

8 Differential Algebraic Dynamic Logic DAL (Excerpt)
- Differential Invariants

9 Differential Temporal Dynamic Logic dTL (Excerpt)

10 Deduction Modulo Real Algebraic and Computer Algebraic Constraints

11 European Train Control System

12 Collision Avoidance Maneuvers in Air Traffic Control

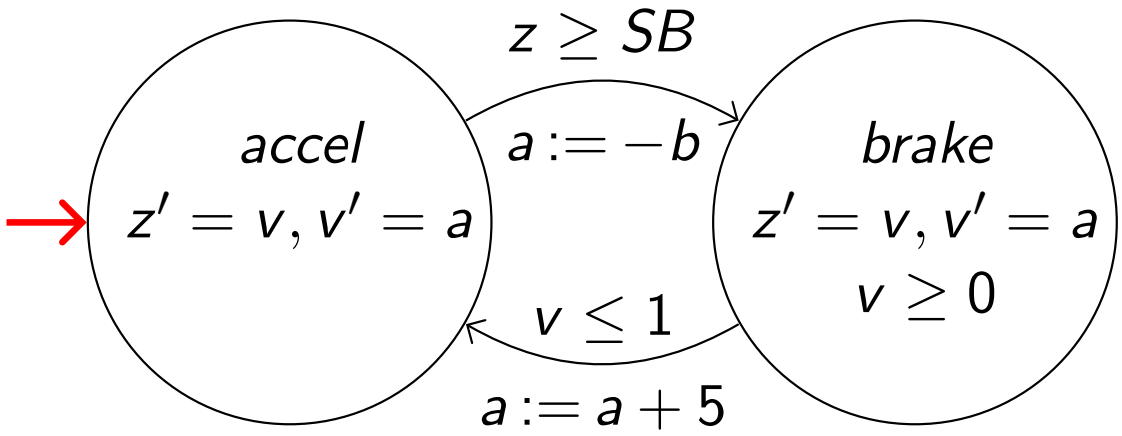
13 Hybrid Automata Embedding

14 Distributed Hybrid Systems

15 Car Control Verification

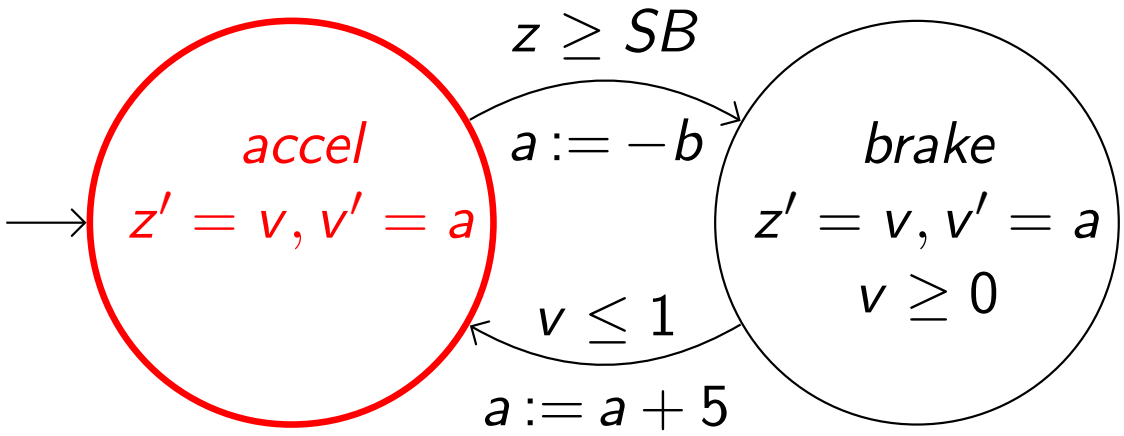
16 Stochastic Hybrid Systems

Embedding Hybrid Automata as Hybrid Programs



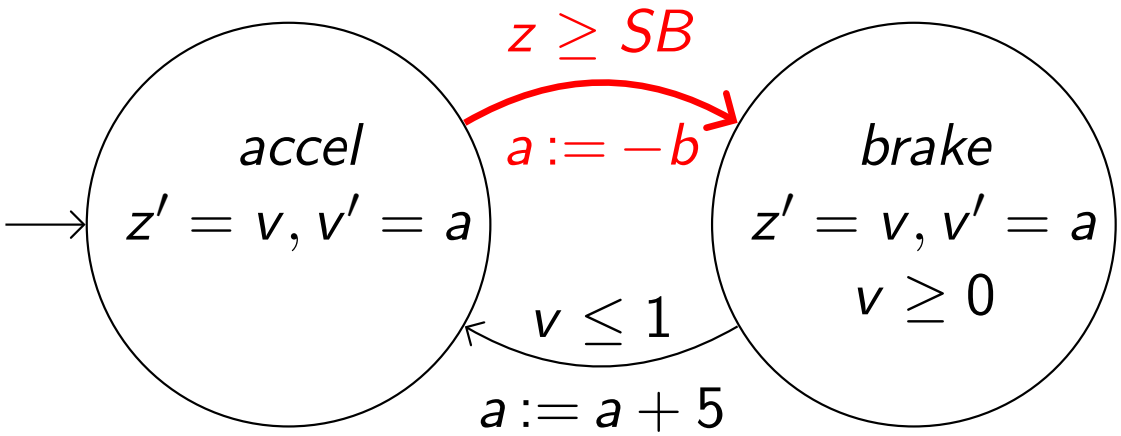
$$
\begin{array}{l}
q := accel; \\
\big(\quad (?q = accel; \quad z' = v, v' = a) \\
\cup \ (?q = accel \wedge z \geq SB; \quad a := -b; \quad q := brake; \quad ?v \geq 0) \\
\cup \ (?q = brake; \quad z' = v, v' = a \,\&\, v \geq 0) \\
\cup \ (?q = brake \wedge v \leq 1; \quad a := a + 5; \quad q := accel)\big)^*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

Embedding Hybrid Automata as Hybrid Programs



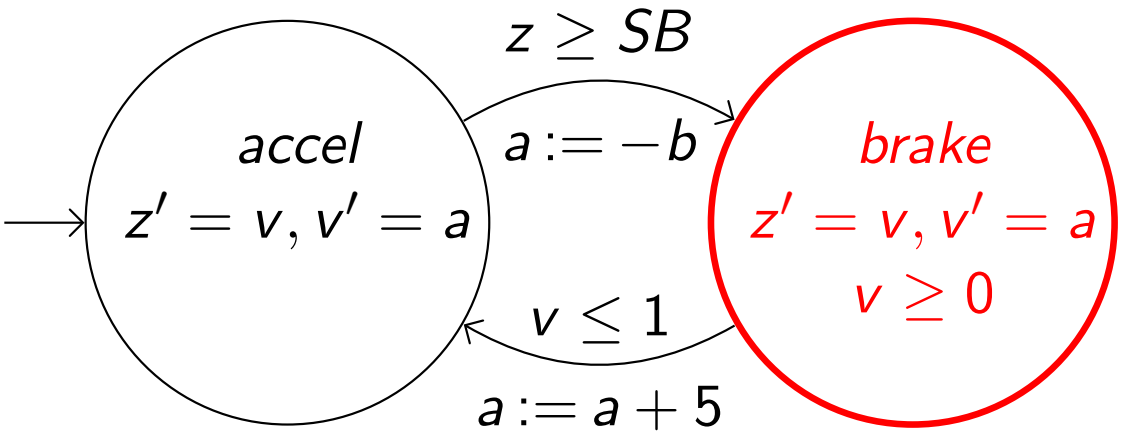
$$
\begin{array}{l}
q := \mathit{accel}; \\
\big(\quad (?q = \mathit{accel}; \quad z' = v, v' = a) \\
\cup \ (?q = \mathit{accel} \wedge z \geq SB; \quad a := -b; \quad q := \mathit{brake}; \quad ?v \geq 0) \\
\cup \ (?q = \mathit{brake}; \quad z' = v, v' = a \,\&\, v \geq 0) \\
\cup \ (?q = \mathit{brake}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

Embedding Hybrid Automata as Hybrid Programs



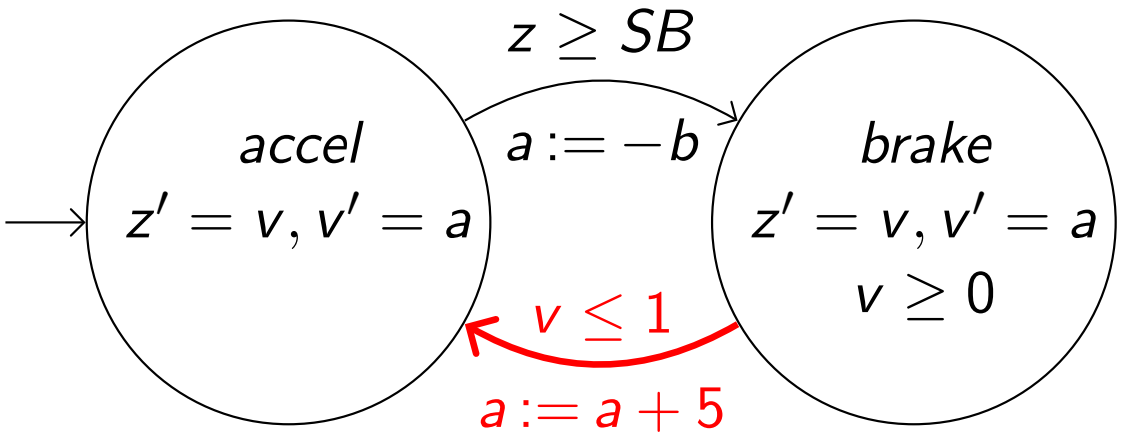
$$
\begin{array}{ll}
& q := accel; \\
(& (?q = accel; \quad z' = v, v' = a) \\
\cup & (?q = accel \wedge z \geq SB; \quad a := -b; \quad q := brake; \quad ?v \geq 0) \\
\cup & (?q = brake; \quad z' = v, v' = a \,\&\, v \geq 0) \\
\cup & (?q = brake \wedge v \leq 1; \quad a := a + 5; \quad q := accel))^*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

Embedding Hybrid Automata as Hybrid Programs



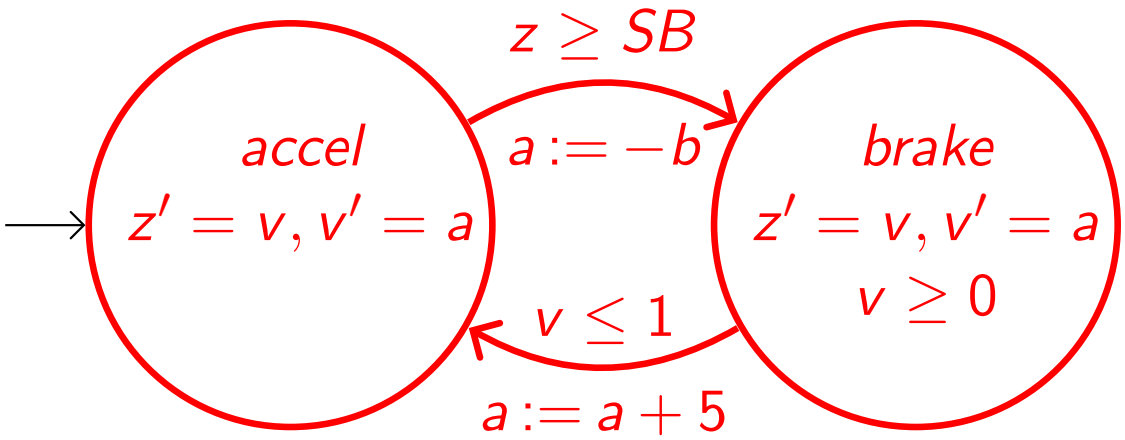
$$
\begin{array}{l}
q := \mathit{accel}; \\
\big(\quad (?q = \mathit{accel}; \quad z' = v, v' = a) \\
\cup \ (?q = \mathit{accel} \wedge z \geq SB; \quad a := -b; \quad q := \mathit{brake}; \quad ?v \geq 0) \\
\cup \ (?q = \mathit{brake}; \quad z' = v, v' = a \,\&\, v \geq 0) \\
\cup \ (?q = \mathit{brake}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

Embedding Hybrid Automata as Hybrid Programs



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

Embedding Hybrid Automata as Hybrid Programs



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

7 Formal Details
- Soundness Proof
- Completeness Proof

8 Differential Algebraic Dynamic Logic DAL (Excerpt)
- Differential Invariants

9 Differential Temporal Dynamic Logic dTL (Excerpt)

10 Deduction Modulo Real Algebraic and Computer Algebraic Constraints

11 European Train Control System

12 Collision Avoidance Maneuvers in Air Traffic Control

13 Hybrid Automata Embedding

14 Distributed Hybrid Systems

15 Car Control Verification

16 Stochastic Hybrid Systems

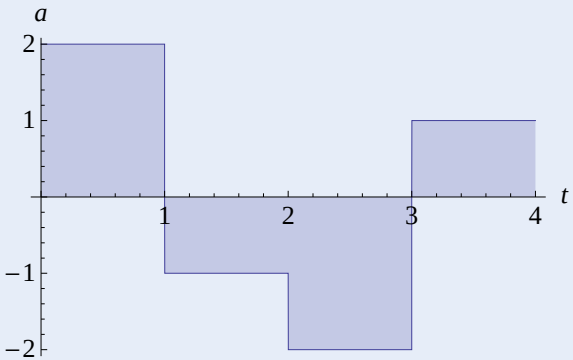
Q: I want to verify my car

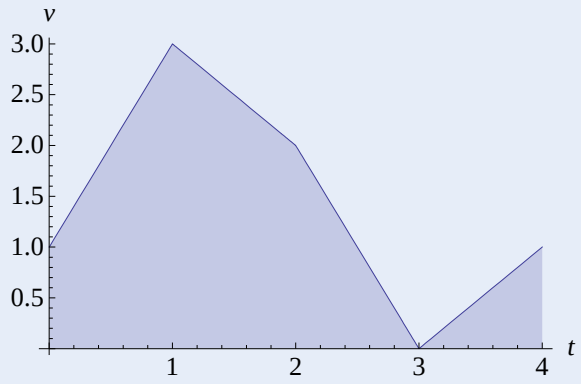
Challenge

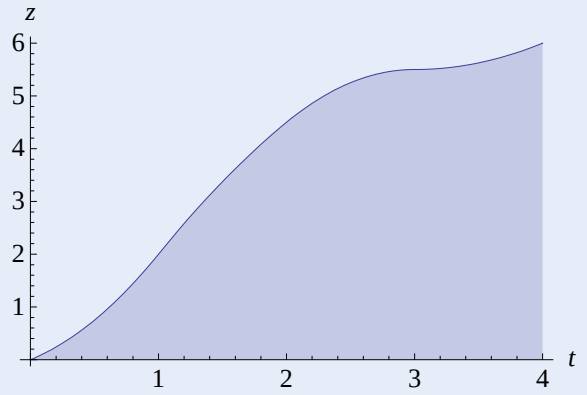
Q: I want to verify my car A: Hybrid systems

Challenge (Hybrid Systems)

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



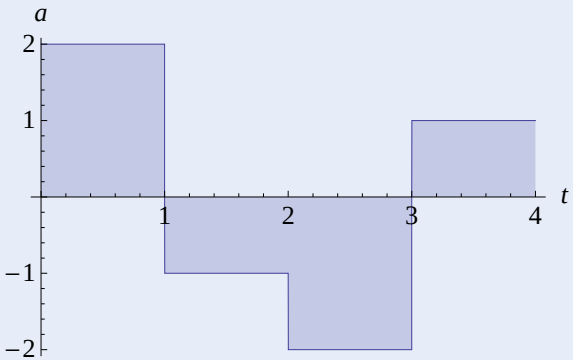


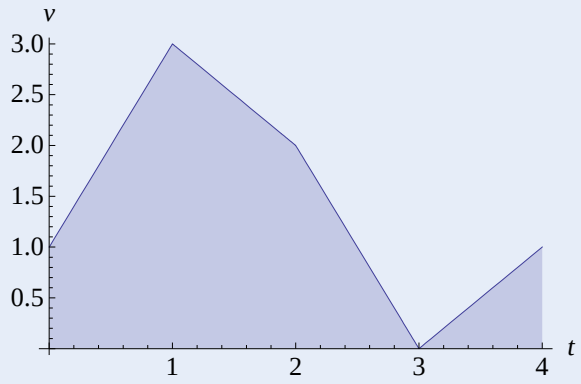


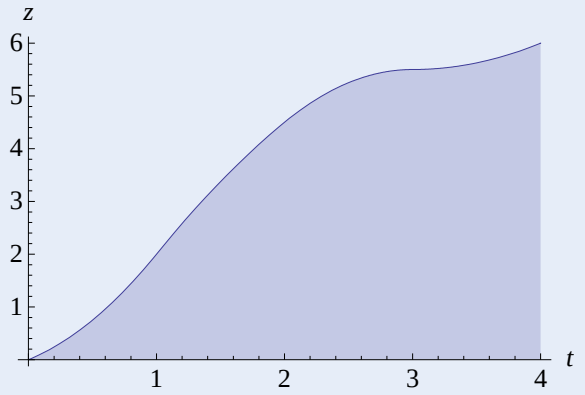
Q: I want to verify my car A: Hybrid systems Q: But there's a lot of cars!

Challenge (Hybrid Systems)

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

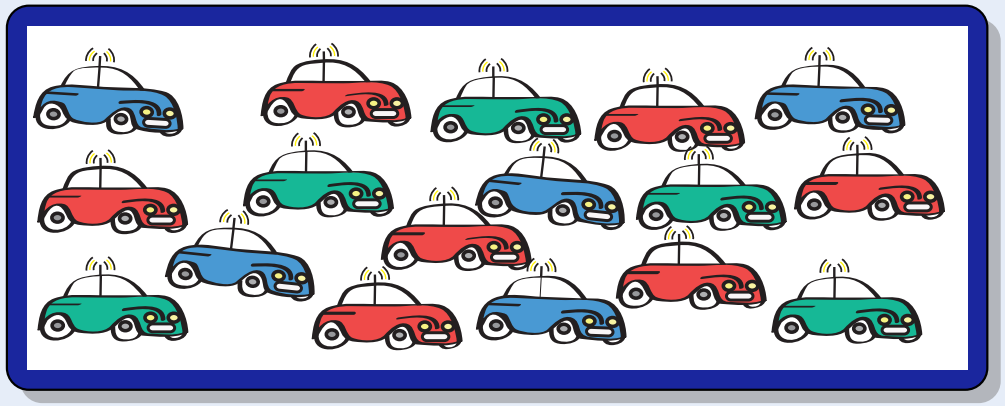






Q: I want to verify a lot of cars

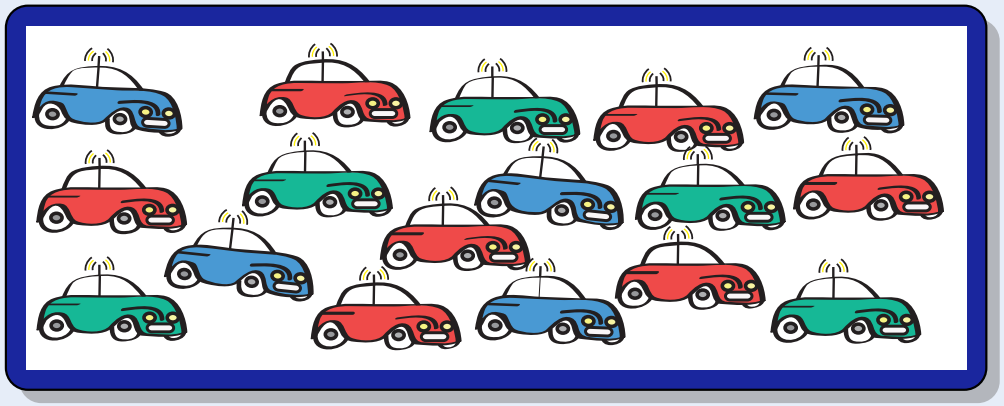
Challenge

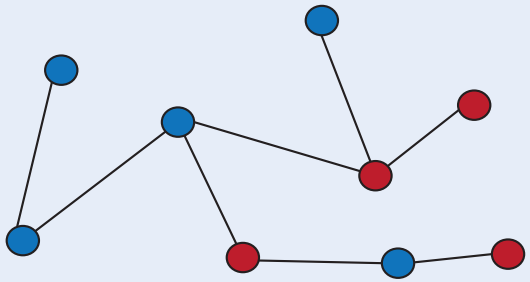


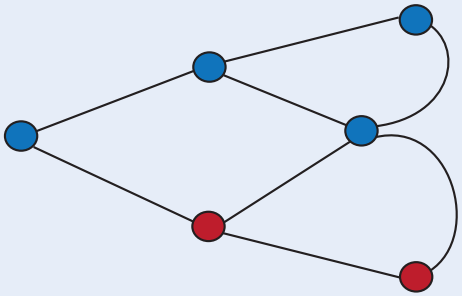
Q: I want to verify a lot of cars A: Distributed systems

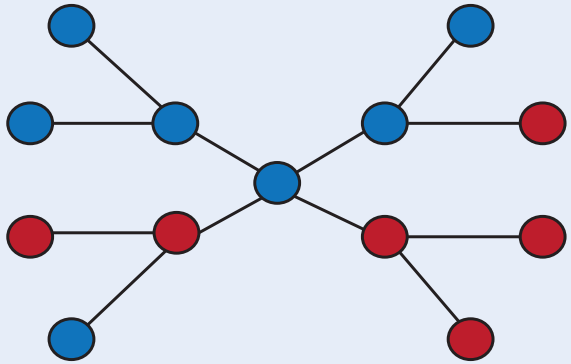
Challenge (Distributed Systems)

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





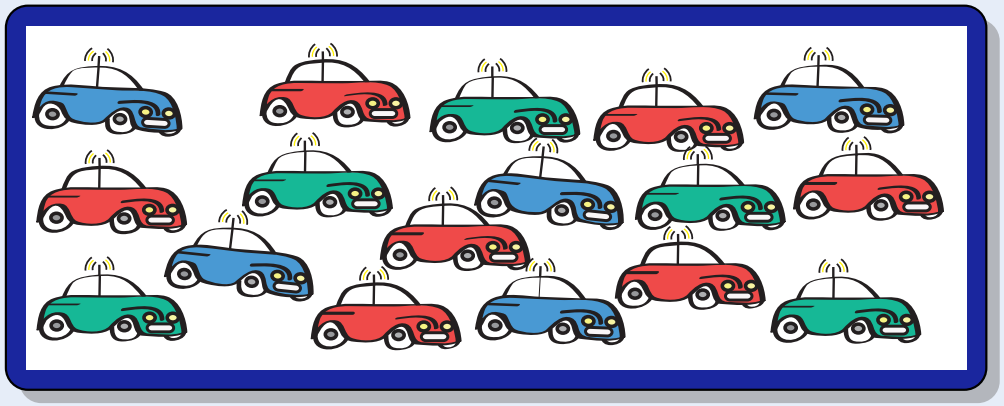


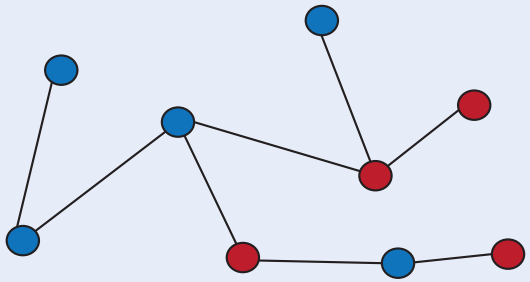


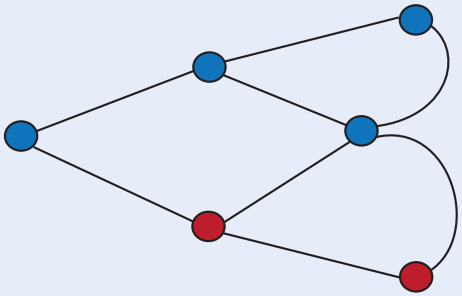
Q: I want to verify a lot of cars A: Distributed systems Q: But they move!

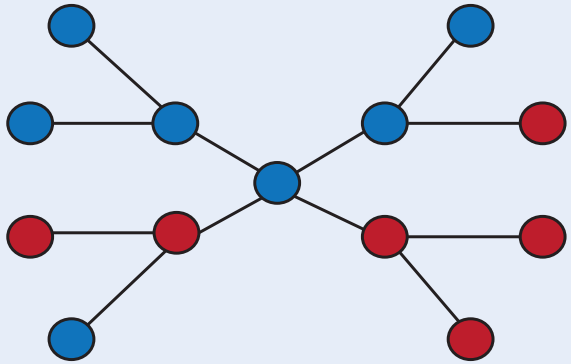
Challenge (Distributed Systems)

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



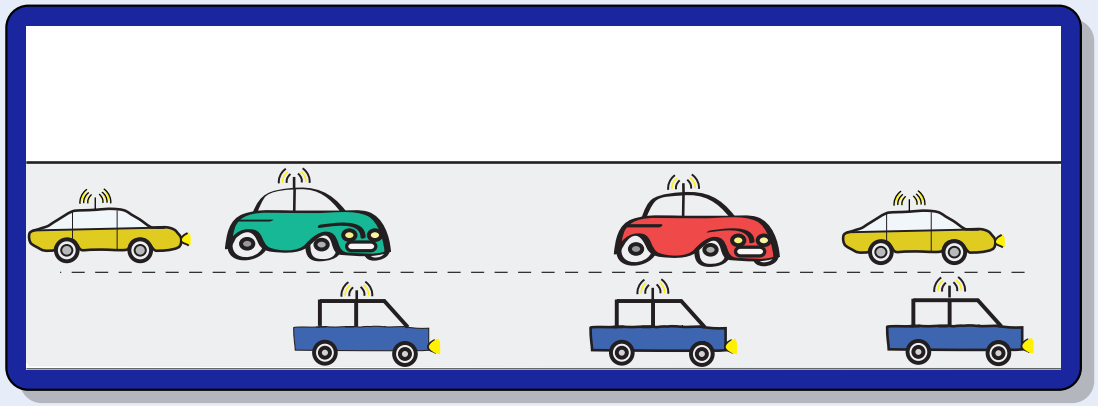






Q: I want to verify lots of moving cars

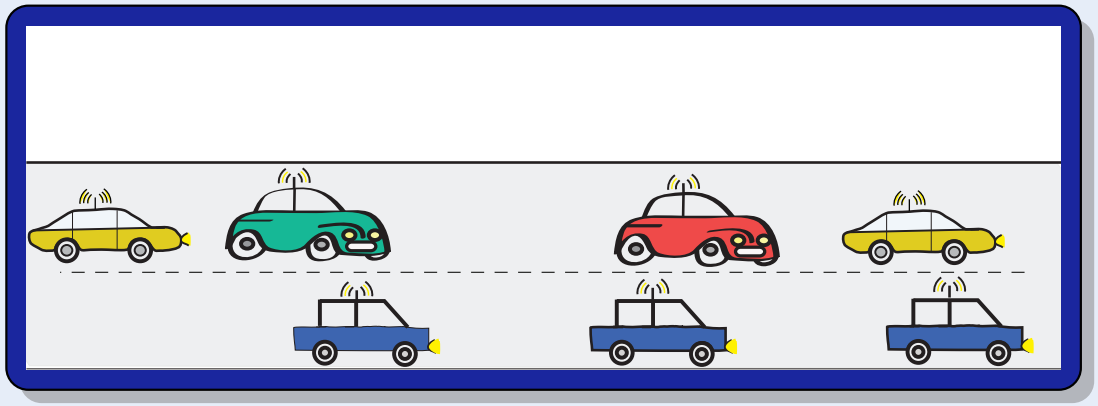
Challenge



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

Challenge (Distributed Hybrid Systems)

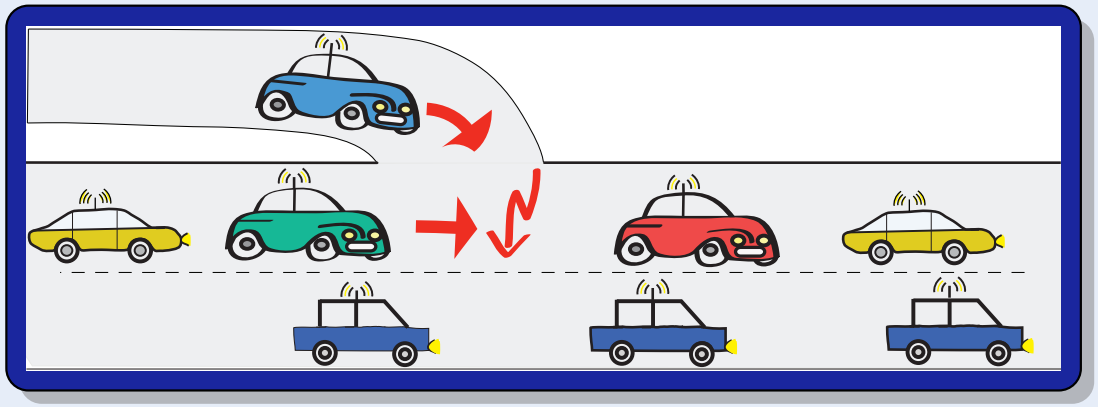
- Continuous dynamics (differential equations)
- Discrete dynamics (control decisions)
- Structural dynamics (communication/coupling)



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

Challenge (Distributed Hybrid Systems)

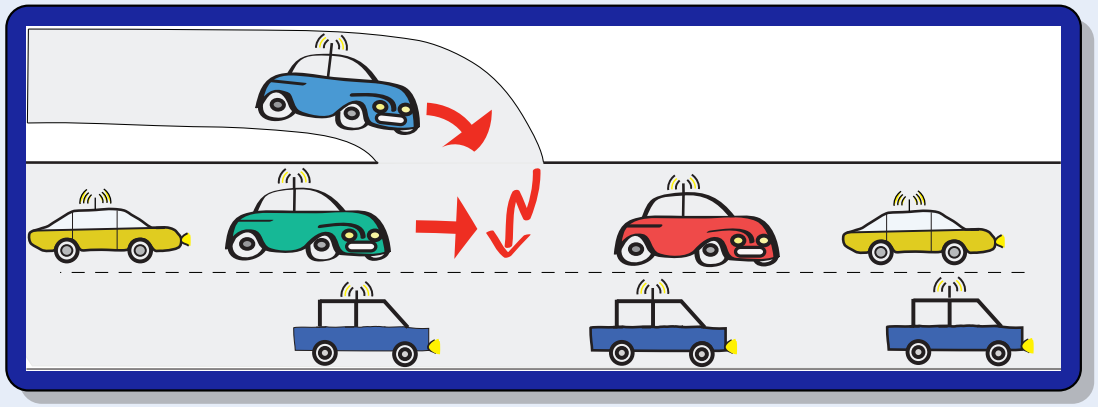
- Continuous dynamics (differential equations)
- Discrete dynamics (control decisions)
- Structural dynamics (communication/coupling)
- Dimensional dynamics (appearance)



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

Challenge (Distributed Hybrid Systems)

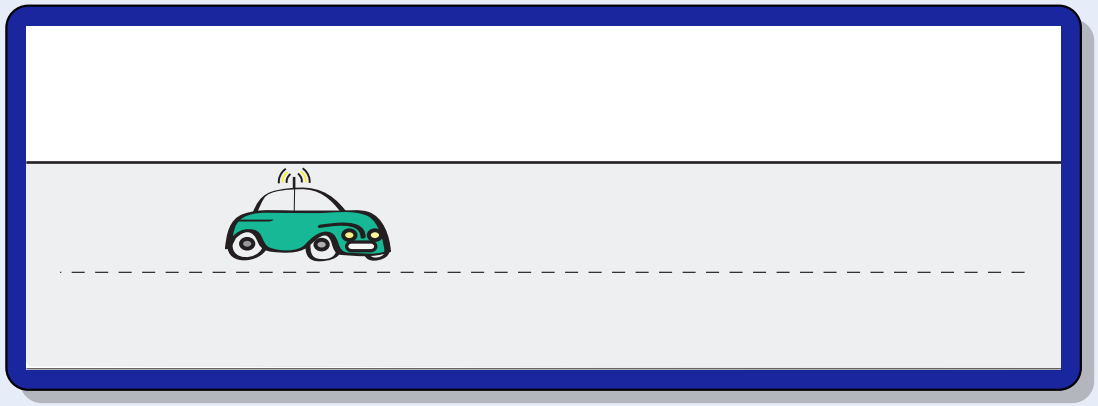
- Continuous dynamics (differential equations)
- Discrete dynamics (control decisions)
- Structural dynamics (communication/coupling)
- Dimensional dynamics (appearance)



Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

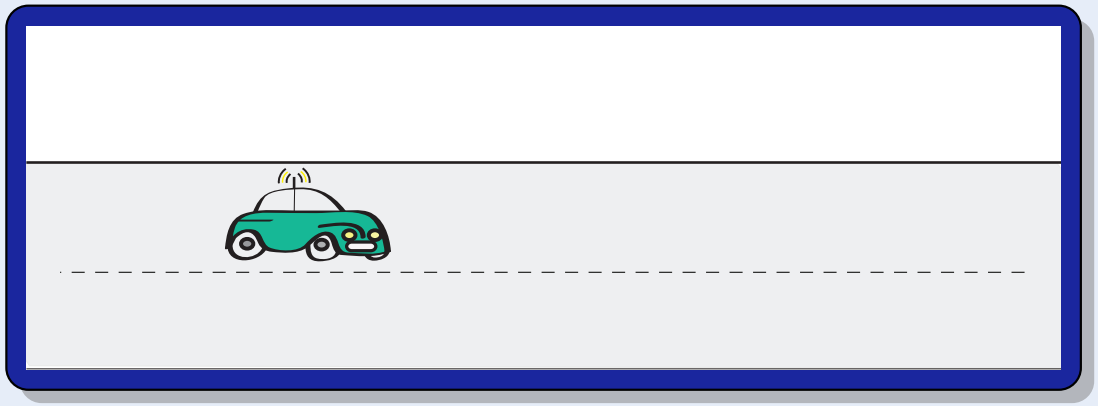
- Continuous dynamics (differential equations)
- Discrete dynamics (control decisions)
- Structural dynamics (communication/coupling)



Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

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



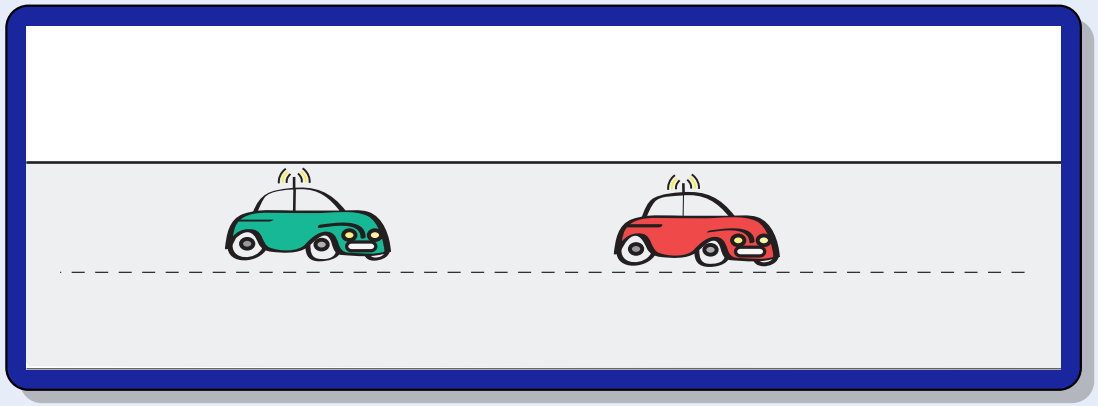
Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

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

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

- Structural dynamics (communication/coupling)



Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

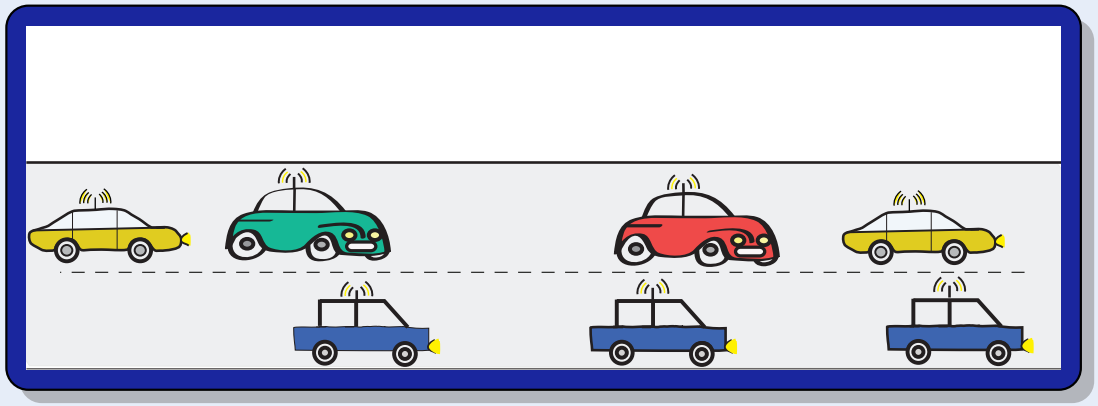
- Continuous dynamics (differential equations)

$$x'' = a$$

- Discrete dynamics (control decisions)

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

- Structural dynamics (communication/coupling)



Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

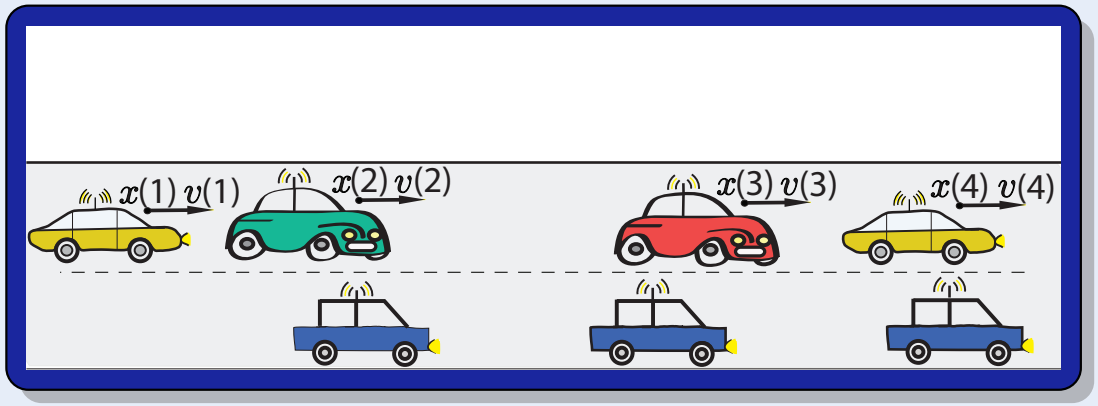
- Continuous dynamics (differential equations)

$$x'' = a$$

- Discrete dynamics (control decisions)

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

- Structural dynamics (communication/coupling)



Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

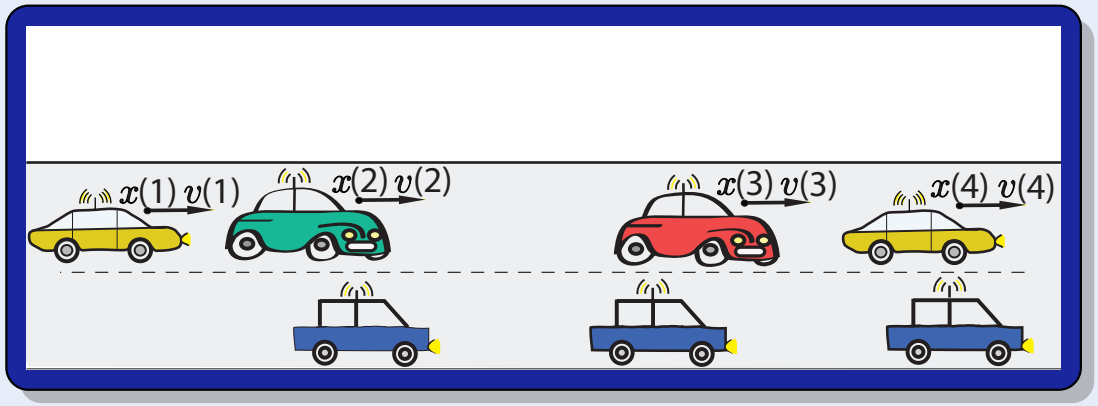
- Continuous dynamics (differential equations)

$$x(i)'' = a(i)$$

- Discrete dynamics (control decisions)

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

- Structural dynamics (communication/coupling)



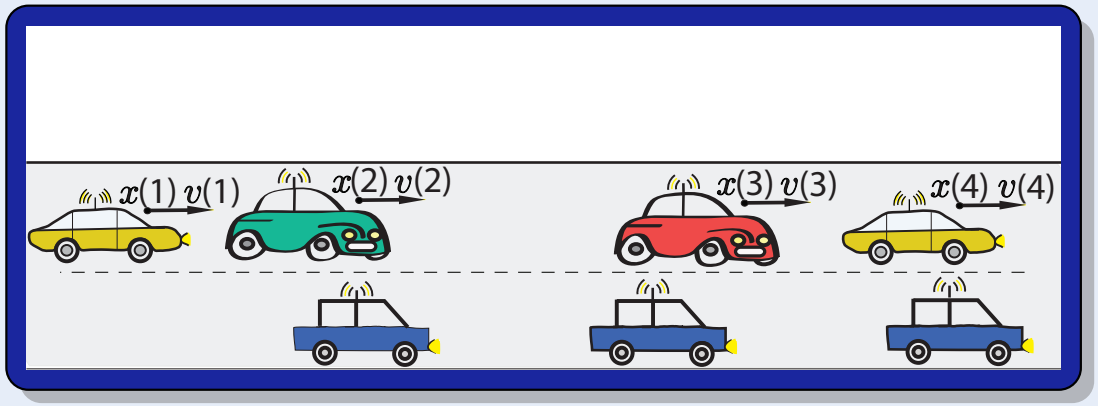
Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

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

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

- Structural dynamics (communication/coupling)



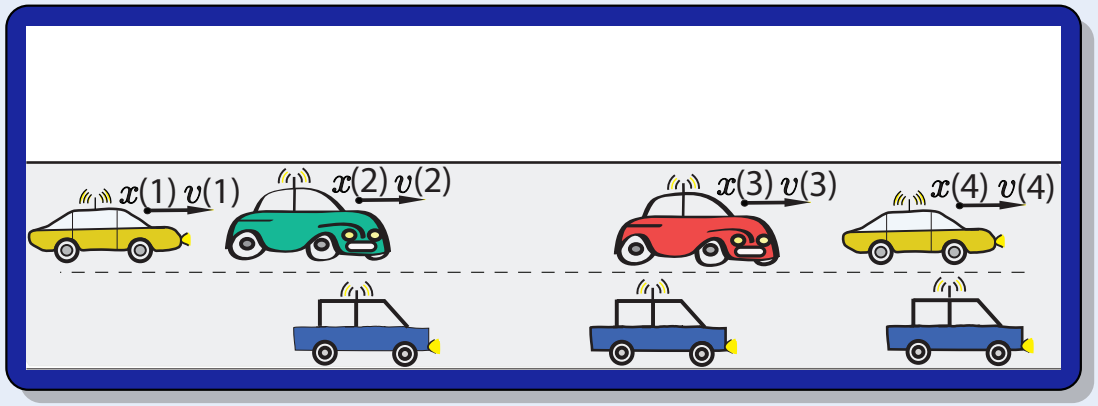
Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

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

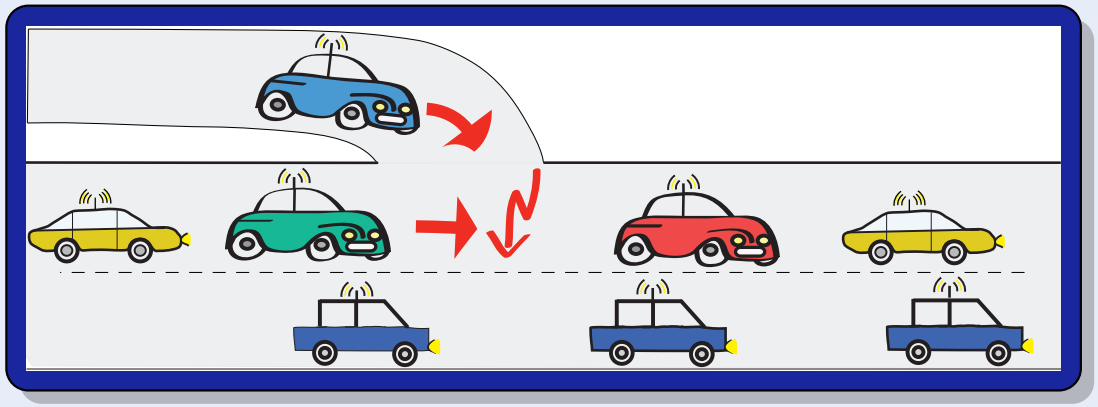
$\forall i\, a(i) := \texttt{if}\,..\,\texttt{then}\, a\, \texttt{else}\, -b\, \texttt{fi}$

- Structural dynamics (communication/coupling)
  $\ell(i) := carInFrontOf(i)$



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

Model (Distributed Hybrid Systems)



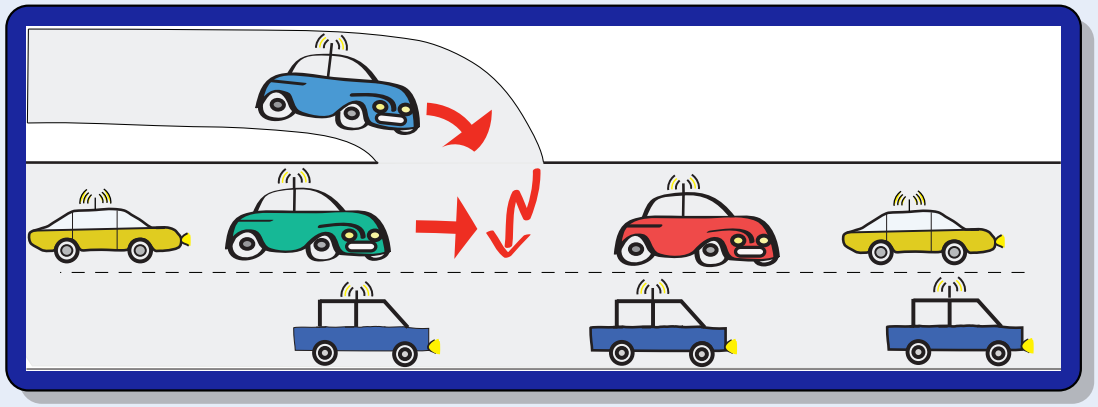
- Continuous dynamics (differential equations)
  $\forall i\, x(i)'' = a(i)$
- Discrete dynamics (control decisions)

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

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

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

Model (Distributed Hybrid Systems)



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

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

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

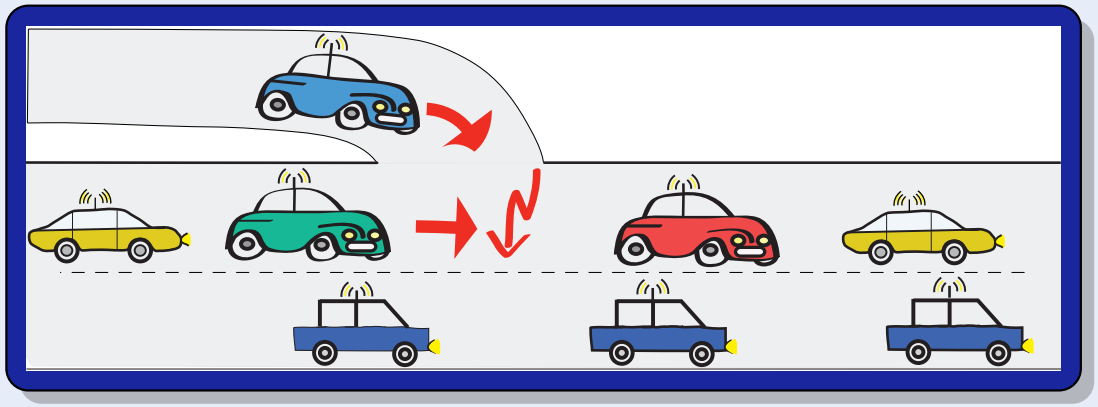
Q: How to model distributed hybrid systems A: Quantified Hybrid Programs

Model (Distributed Hybrid Systems)

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

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

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

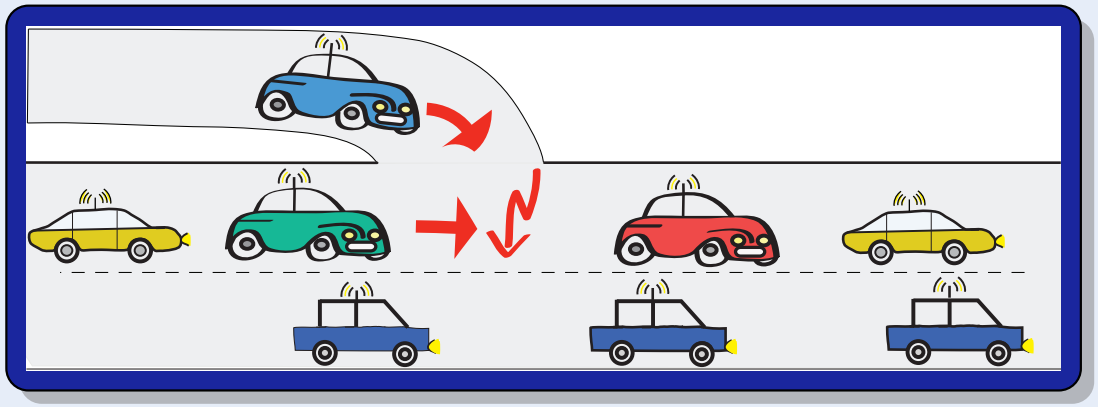


$\Rightarrow$ Communication

$$d(i,\ell(i)) := d(i,\ell(i)) + 10$$

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

Model (Distributed Hybrid Systems)



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

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

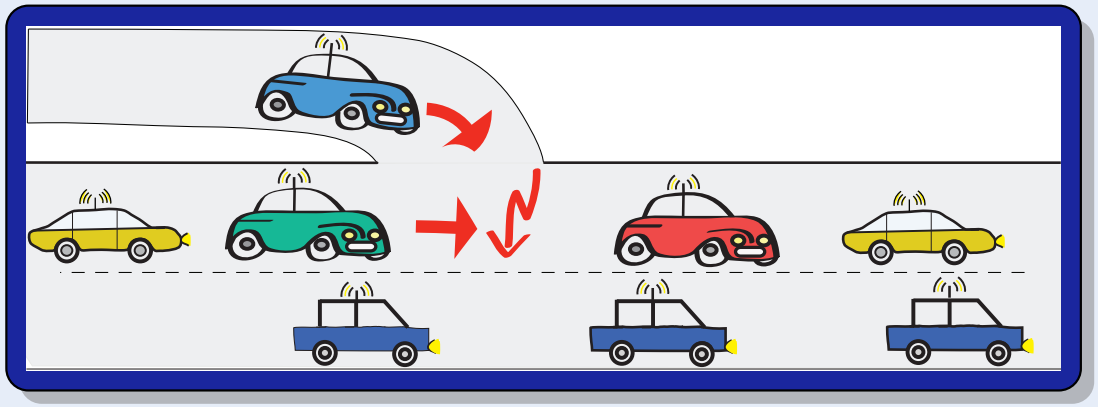
- Structural dynamics (communication/coupling)
  $$\ell(i) := \mathit{carInFrontOf}(i)$$
- Dimensional dynamics (appearance)
  $$n := \texttt{new}\ \mathit{Car}$$

$\Rightarrow$ Communication

$$\forall i\ d(i,\ell(i)) := d(i,\ell(i)) + 10$$

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

Model (Distributed Hybrid Systems)



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

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

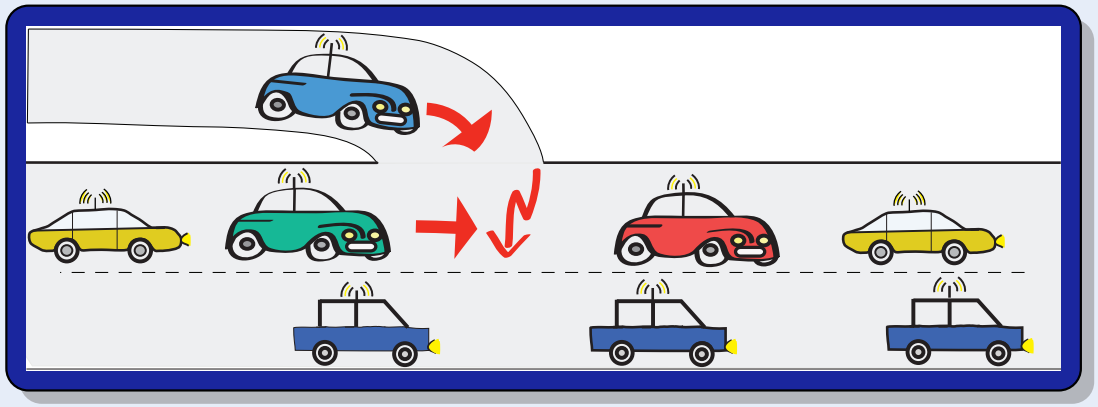
- Structural dynamics (communication/coupling)
  $\ell(i) := \mathit{carInFrontOf}(i)$
- Dimensional dynamics (appearance)
  $n := \texttt{new}\ \mathit{Car}$

$\Rightarrow$ Communication
$\forall i\, d(i, \ell(i)) := d(i, \ell(i)) + 10$

$\Rightarrow$ Discrete structural dynamics
$\ell(i) := \ell(\ell(i))$

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

Model (Distributed Hybrid Systems)



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

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

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

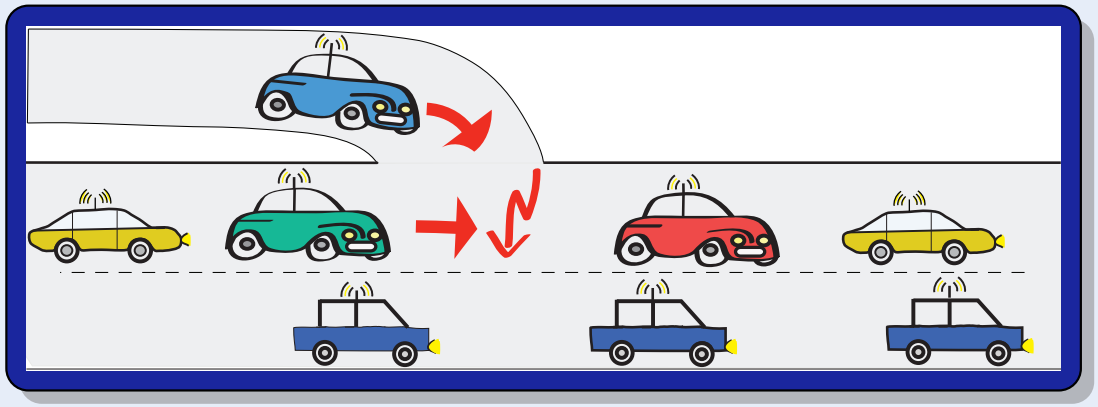
$\Rightarrow$ Communication
$$\forall i\, d(i,\ell(i)) := d(i,\ell(i)) + 10$$

$\Rightarrow$ Discrete structural dynamics
$$\ell(i) := \ell(\ell(i))$$

$\Rightarrow$ Continuous structural dynamics
$$x(i)'' = a(i) + c(i,\ell(i))a(\ell(i))$$

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

Model (Distributed Hybrid Systems)



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

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

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

$\Rightarrow$ Communication
$$\forall i\, d(i,\ell(i)) := d(i,\ell(i)) + 10$$

$\Rightarrow$ Discrete structural dynamics
$$\ell(i) := \ell(\ell(i))$$

$\Rightarrow$ Continuous structural dynamics
$$\forall i\, x(i)'' = a(i) + c(i,\ell(i))a(\ell(i))$$

7 Formal Details
- Soundness Proof
- Completeness Proof

8 Differential Algebraic Dynamic Logic DAL (Excerpt)
- Differential Invariants

9 Differential Temporal Dynamic Logic dTL (Excerpt)

10 Deduction Modulo Real Algebraic and Computer Algebraic Constraints

11 European Train Control System

12 Collision Avoidance Maneuvers in Air Traffic Control

13 Hybrid Automata Embedding

14 Distributed Hybrid Systems

15 Car Control Verification

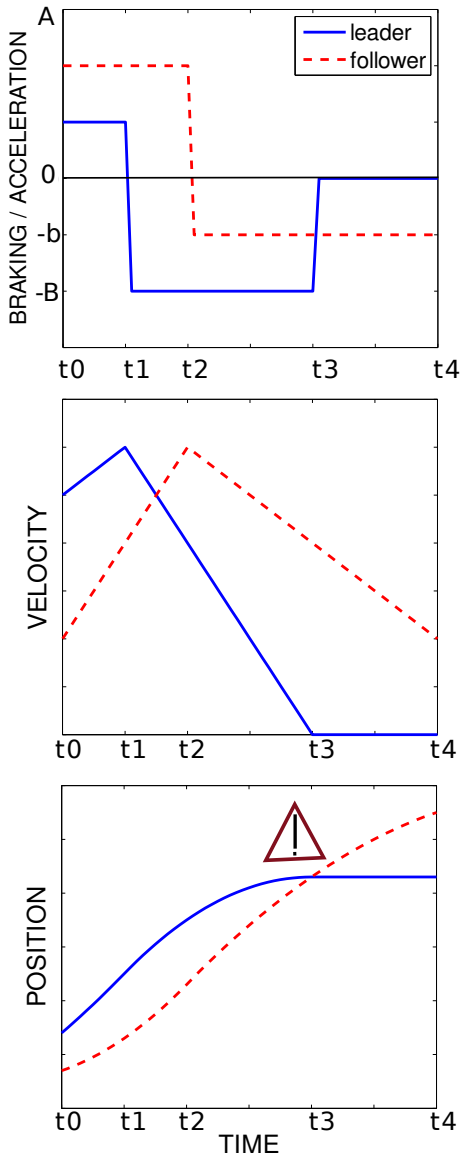
16 Stochastic Hybrid Systems

Challenge: Local lane dynamics

- A car controller for a differential equation respects separation of local lane.

Challenge: Local lane dynamics

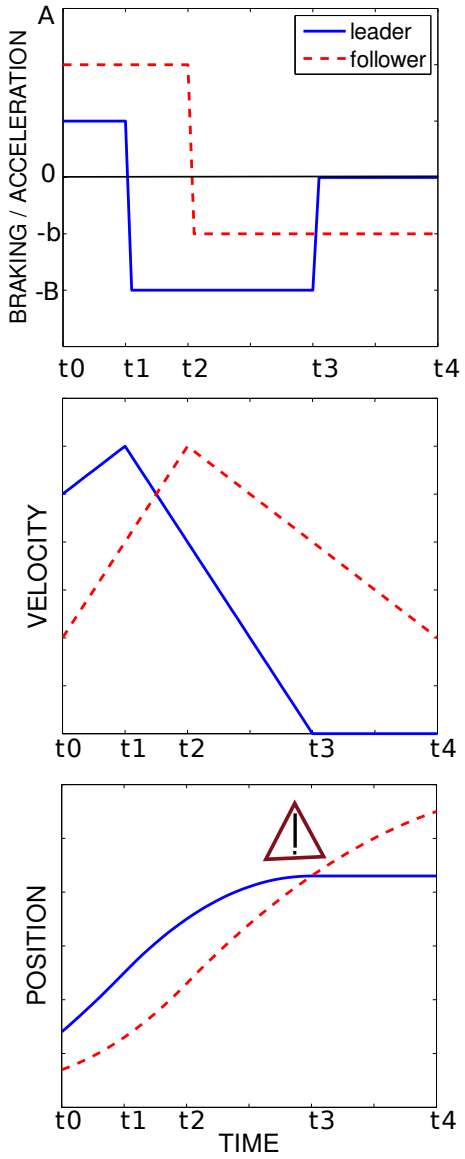
- A car controller for a differential equation respects separation of local lane.
- Follower car maintains safe distance to leader:



Challenge: Local lane dynamics

- A car controller for a differential equation respects separation of local lane.
- Follower car maintains safe distance to leader:

$$f \ll \ell \rightarrow [(a_i := ctrl;\ x_i'' = a_i)^*]\, f \ll \ell$$

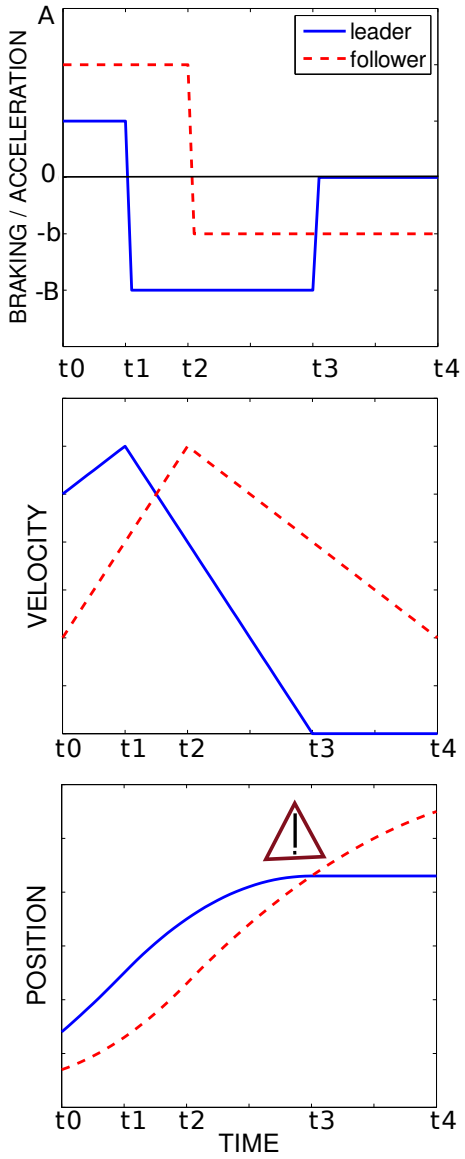


Challenge: Local lane dynamics

- A car controller for a differential equation respects separation of local lane.
- Follower car maintains safe distance to leader:

$$f \ll \ell \rightarrow [(a_i := ctrl;\ x_i'' = a_i)^*]\, f \ll \ell$$

$$f \ll \ell \equiv (x_f \leq x_\ell) \wedge (f \neq \ell) \rightarrow (x_\ell > x_f + \frac{v_f^2}{2b} - \frac{v_\ell^2}{2B} \wedge x_\ell > x_f \wedge v_f \geq 0 \wedge v_\ell \geq 0)$$



Challenge: Global lane dynamics

- All controllers for arbitrarily many differential equations respect separation globally on lane.

Challenge: Global lane dynamics

- All controllers for arbitrarily many differential equations respect separation globally on lane.
- Each car safe behind all others

Challenge: Global lane dynamics

- All controllers for arbitrarily many differential equations respect separation globally on lane.
- Each car safe behind all others

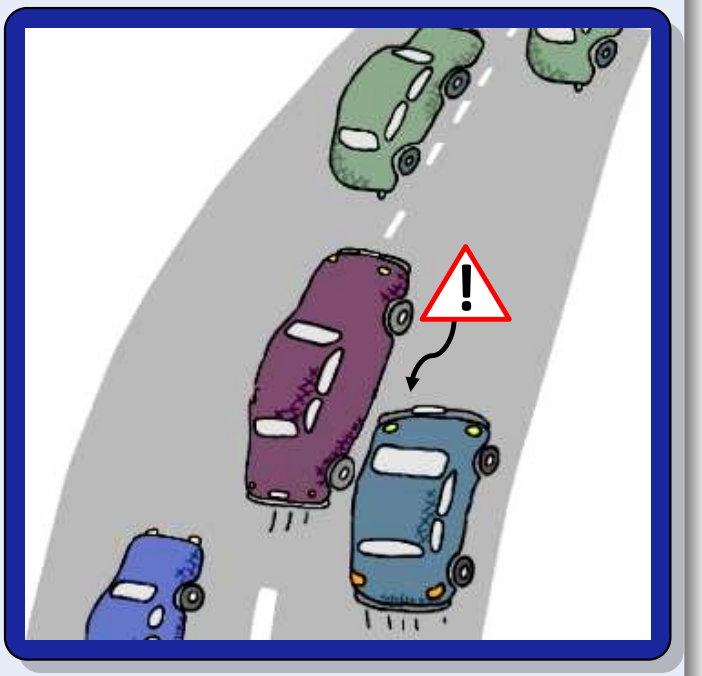
$$[(\forall i\, a(i) := ctrl;\ \forall i\, x(i)'' = a(i))^*]\forall i,j\, i \ll j$$

Challenge: Local highway dynamics

- All controllers for arbitrarily many differential equations respect separation locally on highway.

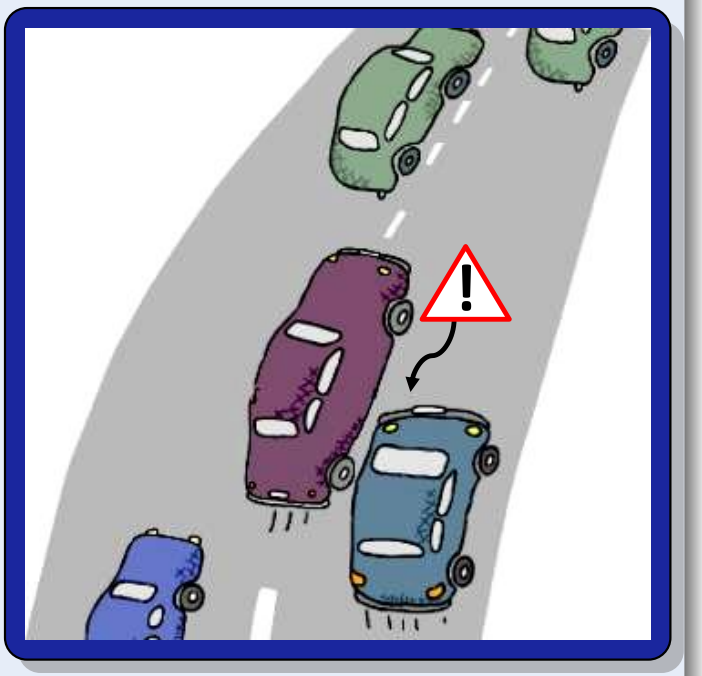
Challenge: Local highway dynamics

- All controllers for arbitrarily many differential equations respect separation locally on highway.
- For each lane: all controllers for the differential equations respect separation even if cars appear or disappear.



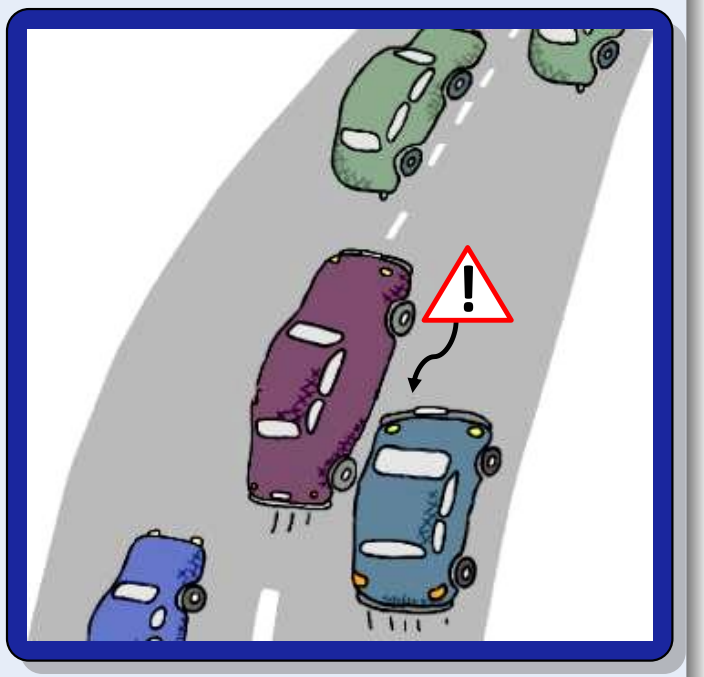
Challenge: Local highway dynamics

- All controllers for arbitrarily many differential equations respect separation locally on highway.
- For each lane: all controllers for the differential equations respect separation even if cars appear or disappear.
- Each car safe behind all others, even if new cars appear or disappear.



Challenge: Local highway dynamics

- All controllers for arbitrarily many differential equations respect separation locally on highway.
- For each lane: all controllers for the differential equations respect separation even if cars appear or disappear.
- Each car safe behind all others, even if new cars appear or disappear.



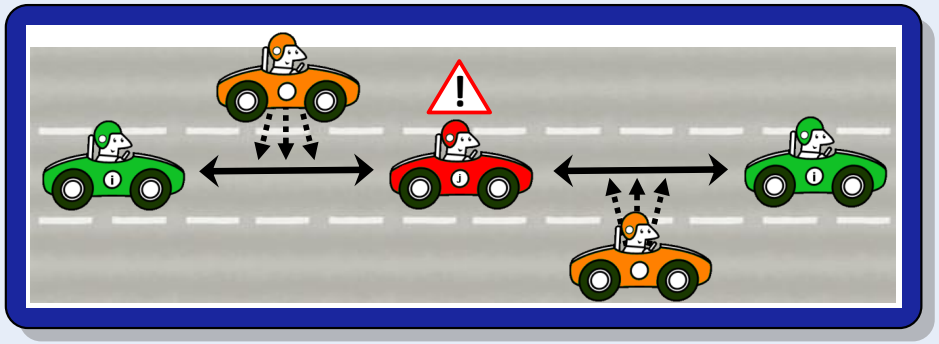
$$[(n := \texttt{new}\ C;\ \forall i\ a(i) := ctrl;\ \forall i\ x(i)'' = a(i))^*]\forall i,j\ i \ll j$$

Challenge: Global highway dynamics

- All controllers for arbitrarily many differential equations respect separation globally on highway.

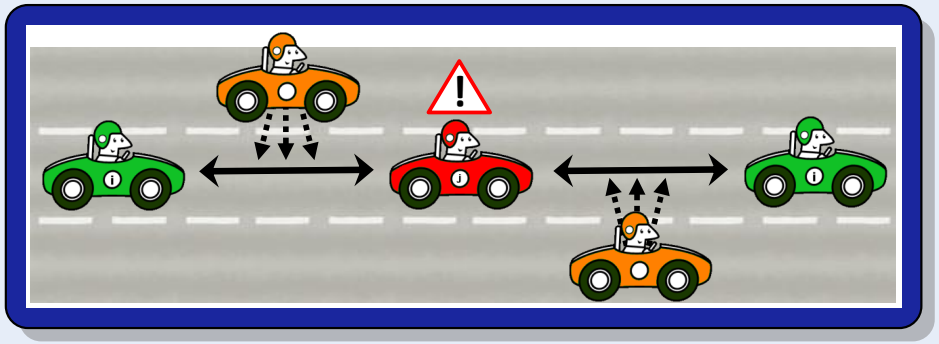
Challenge: Global highway dynamics

- All controllers for arbitrarily many differential equations respect separation globally on highway.
- All controllers for the differential equations respect separation even if cars switch lanes.

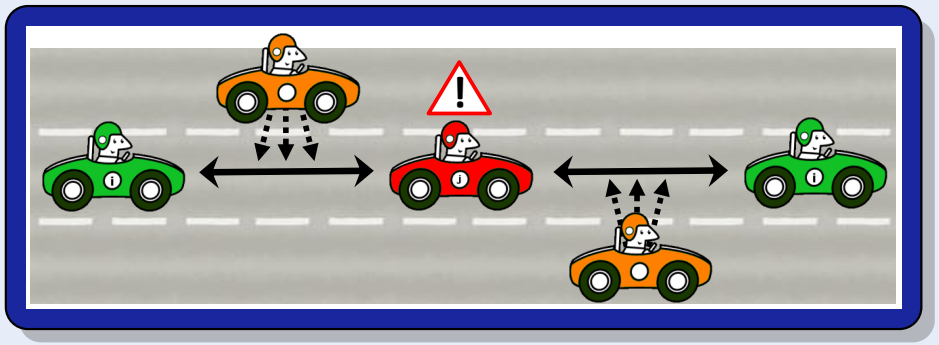


Challenge: Global highway dynamics

- All controllers for arbitrarily many differential equations respect separation globally on highway.
- All controllers for the differential equations respect separation even if cars switch lanes.
- On all lanes, all car safe behind all others on their lanes, even if cars switch lanes.



Challenge: Global highway dynamics



- All controllers for arbitrarily many differential equations respect separation globally on highway.
- All controllers for the differential equations respect separation even if cars switch lanes.
- On all lanes, all car safe behind all others on their lanes, even if cars switch lanes.

$$[\forall l\,(n := \texttt{new}\ C;\ \forall i\ a(i) := ctrl;\ \forall i\ x(i)'' = a(i))^*]\forall l\,\forall i,j\ i \ll j$$

7 Formal Details
- Soundness Proof
- Completeness Proof

8 Differential Algebraic Dynamic Logic DAL (Excerpt)
- Differential Invariants

9 Differential Temporal Dynamic Logic dTL (Excerpt)

10 Deduction Modulo Real Algebraic and Computer Algebraic Constraints

11 European Train Control System

12 Collision Avoidance Maneuvers in Air Traffic Control

13 Hybrid Automata Embedding

14 Distributed Hybrid Systems

15 Car Control Verification

16 Stochastic Hybrid Systems

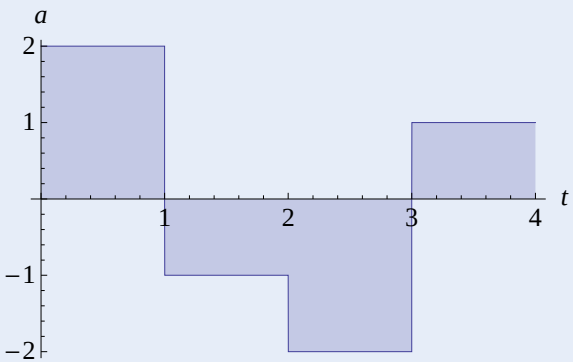
Q: I want to verify trains

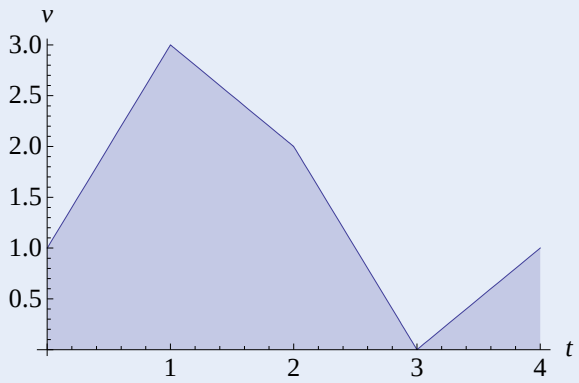
Challenge

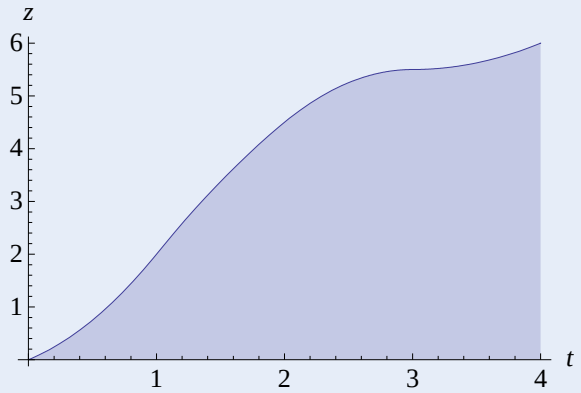
Q: I want to verify trains A: Hybrid systems

Challenge (Hybrid Systems)

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



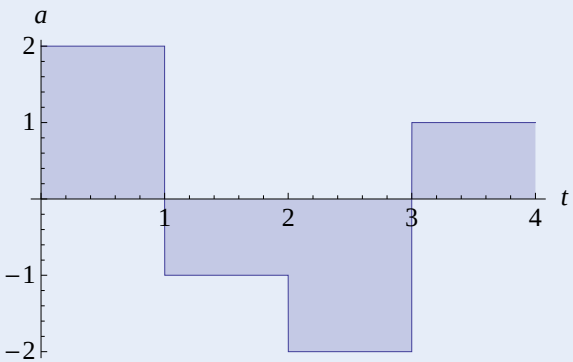


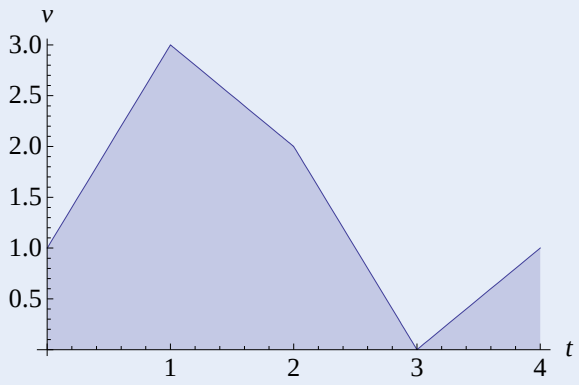


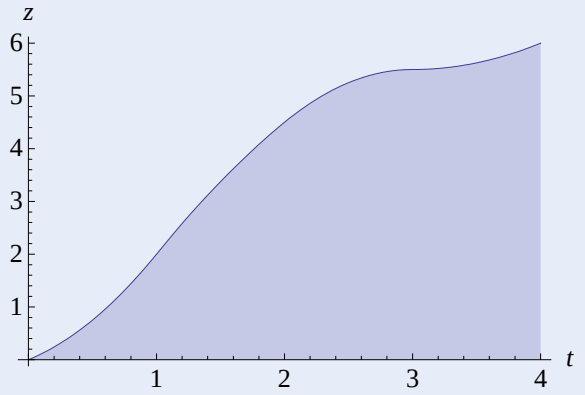
Q: I want to verify trains A: Hybrid systems Q: But there's uncertainties!

Challenge (Hybrid Systems)

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

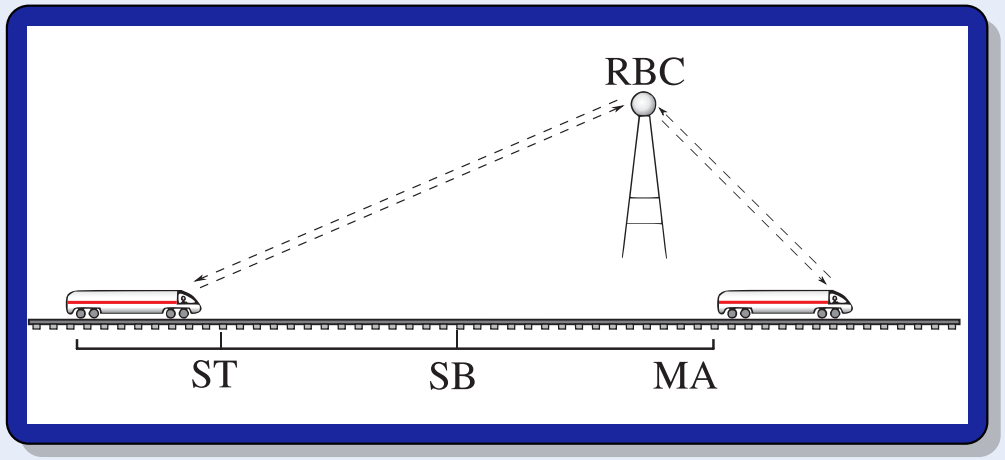






Q: I want to verify uncertain trains

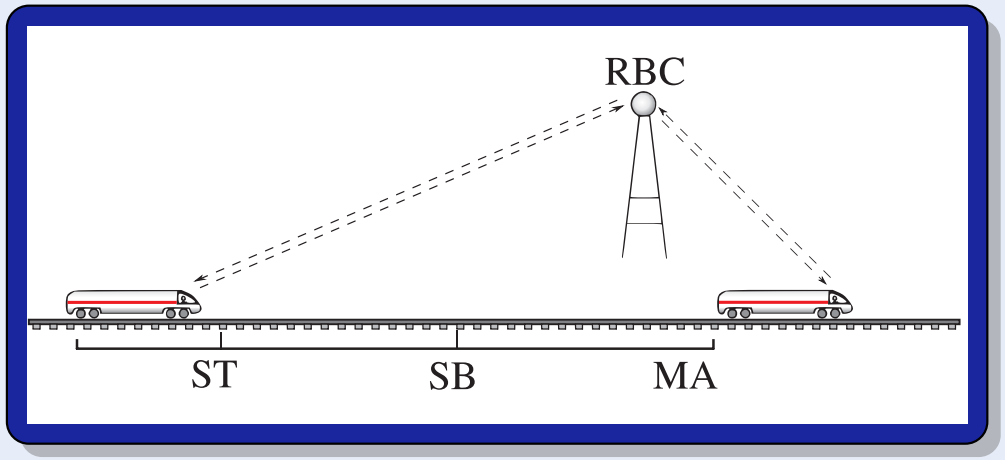
Challenge

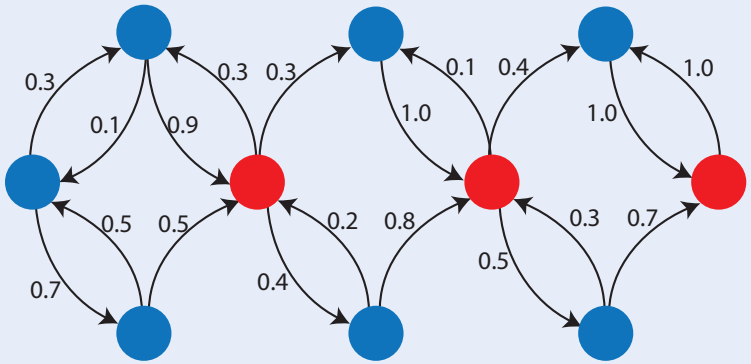


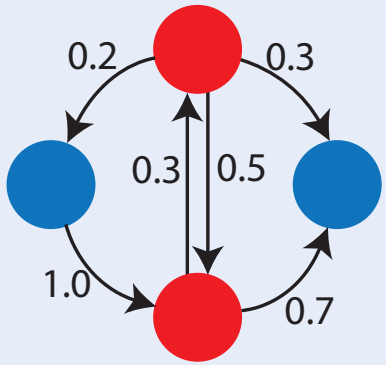
Q: I want to verify uncertain trains A: Markov chains

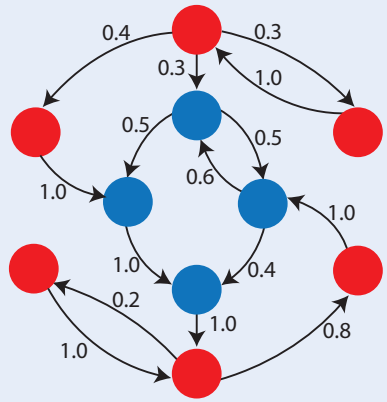
Challenge (Probabilistic Systems)

- Directed graph
  (Countable state space)
- Weighted edges
  (Transition probabilities)





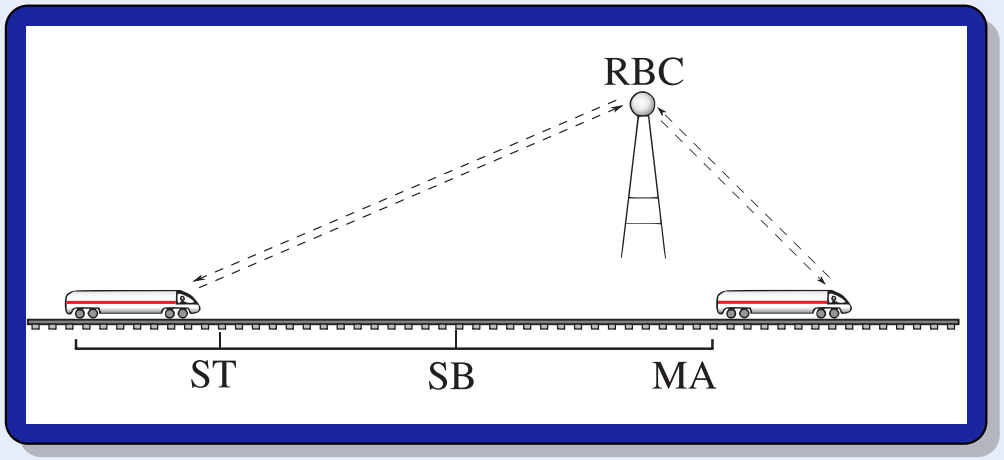


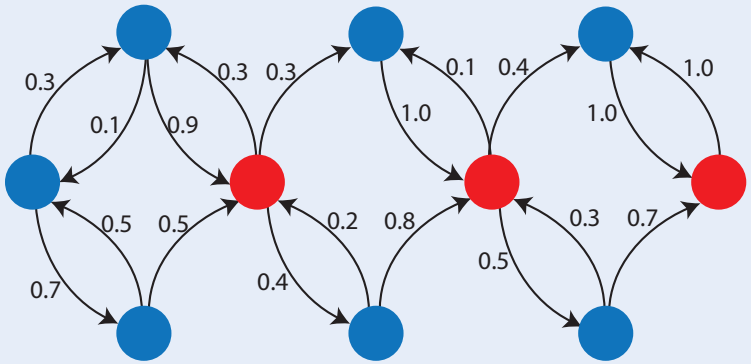


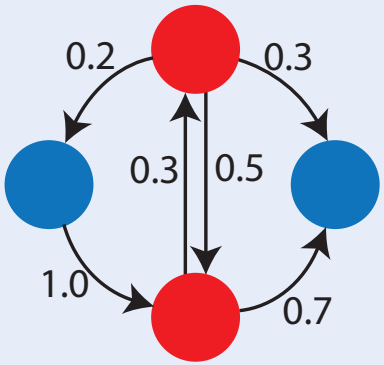
Q: I want to verify uncertain trains A: Markov chains Q: But trains move!

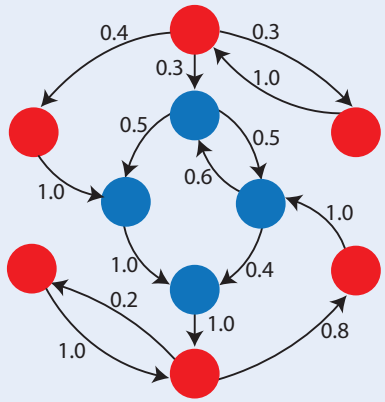
Challenge (Probabilistic Systems)

- Directed graph
  (Countable state space)
- Weighted edges
  (Transition probabilities)



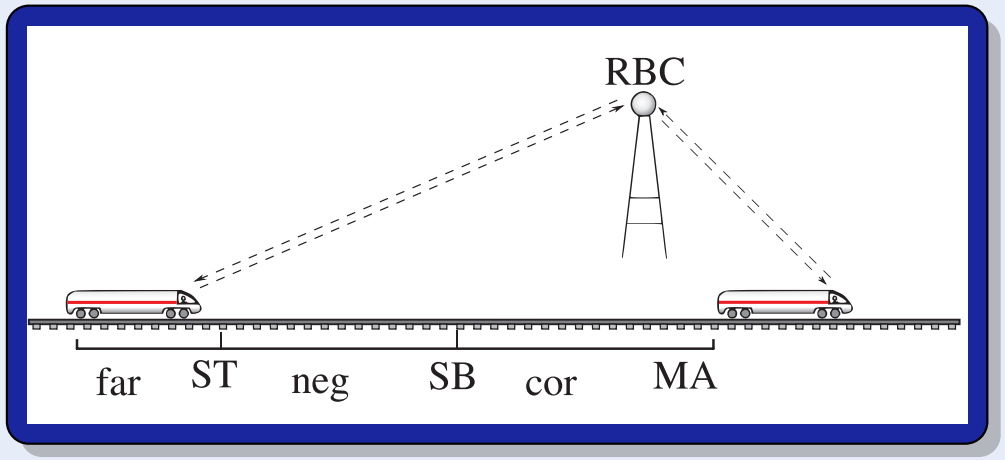






Q: I want to verify uncertain systems

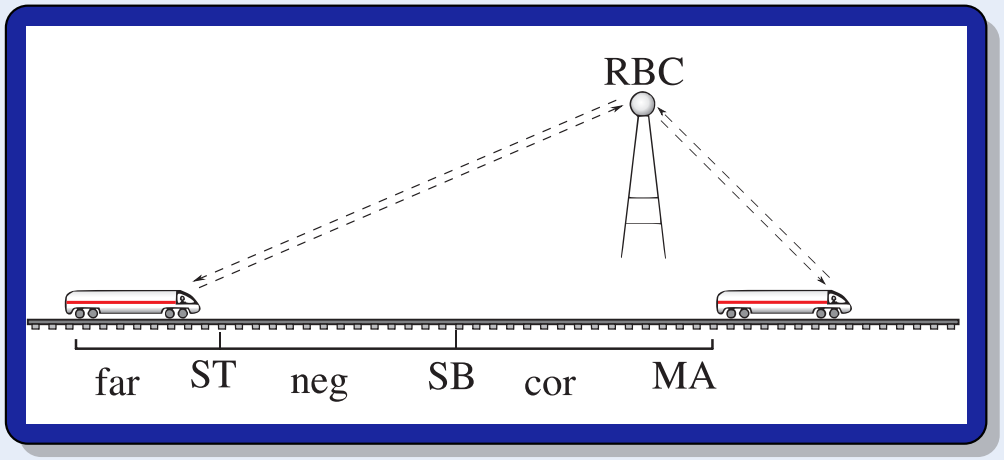
Challenge



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

Challenge (Stochastic Hybrid Systems)

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

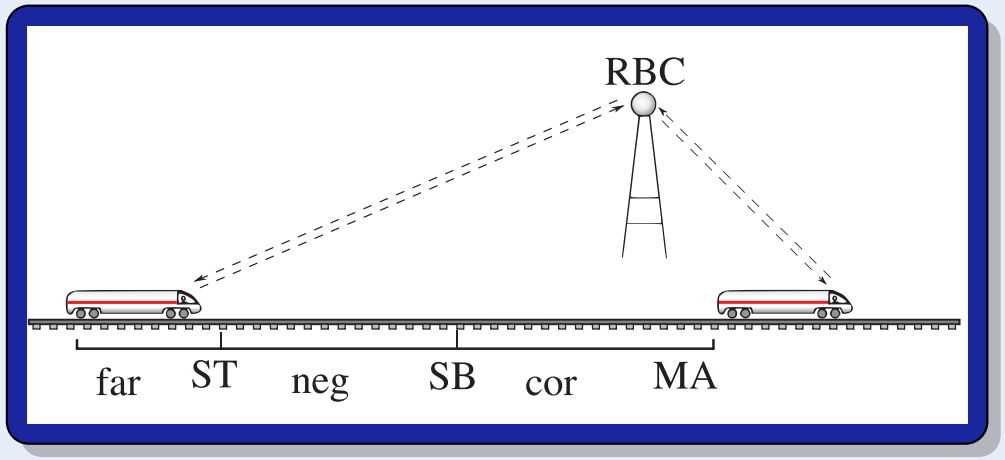


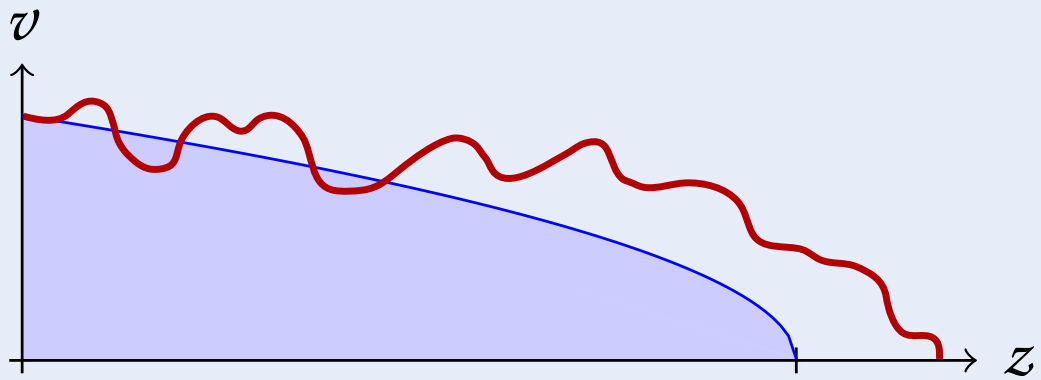
Q: I want to verify uncertain systems A: Stochastic hybrid systems

Challenge (Stochastic Hybrid Systems)

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

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

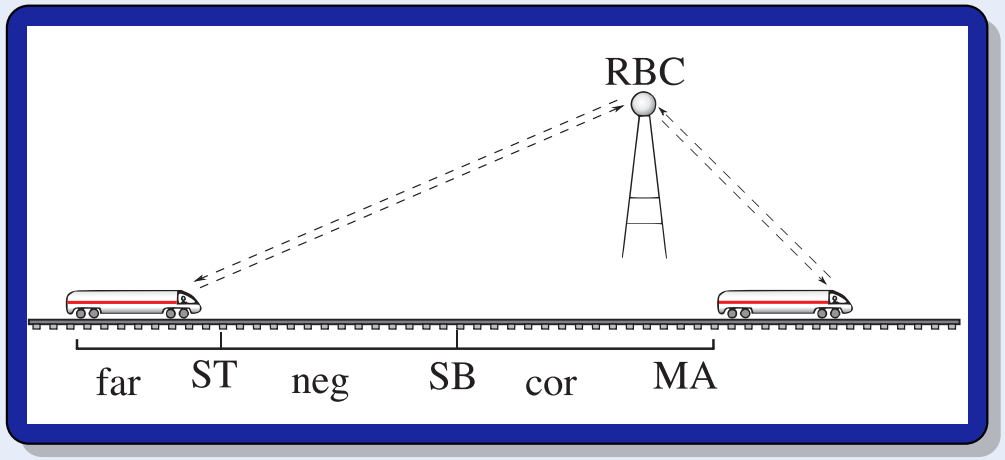


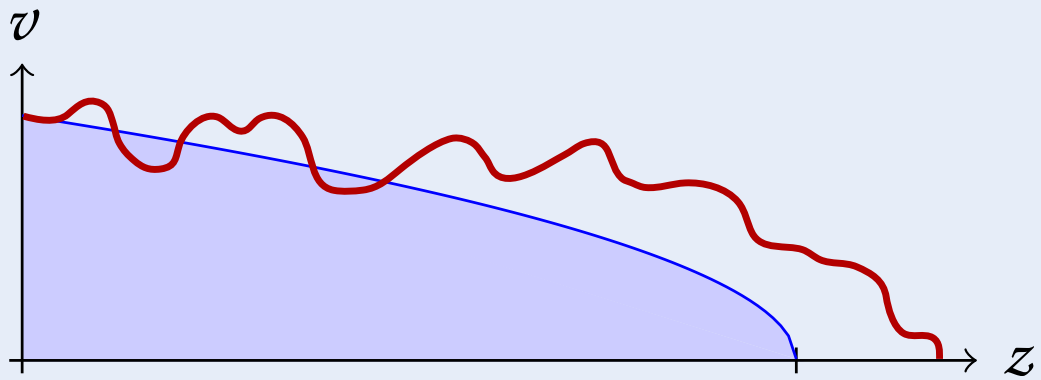


Q: I want to verify uncertain systems A: Stochastic hybrid systems Q: How?

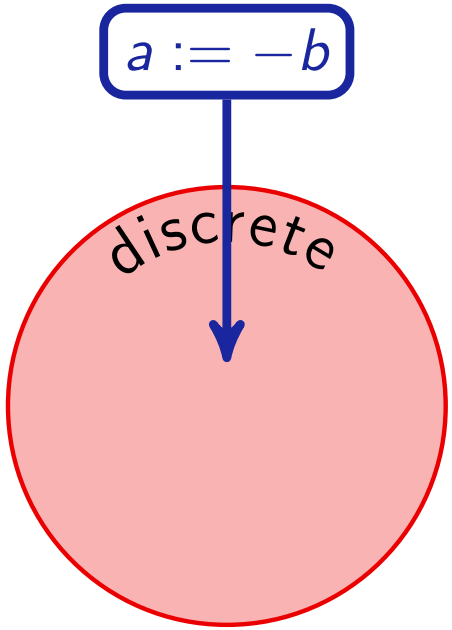
Challenge (Stochastic Hybrid Systems)

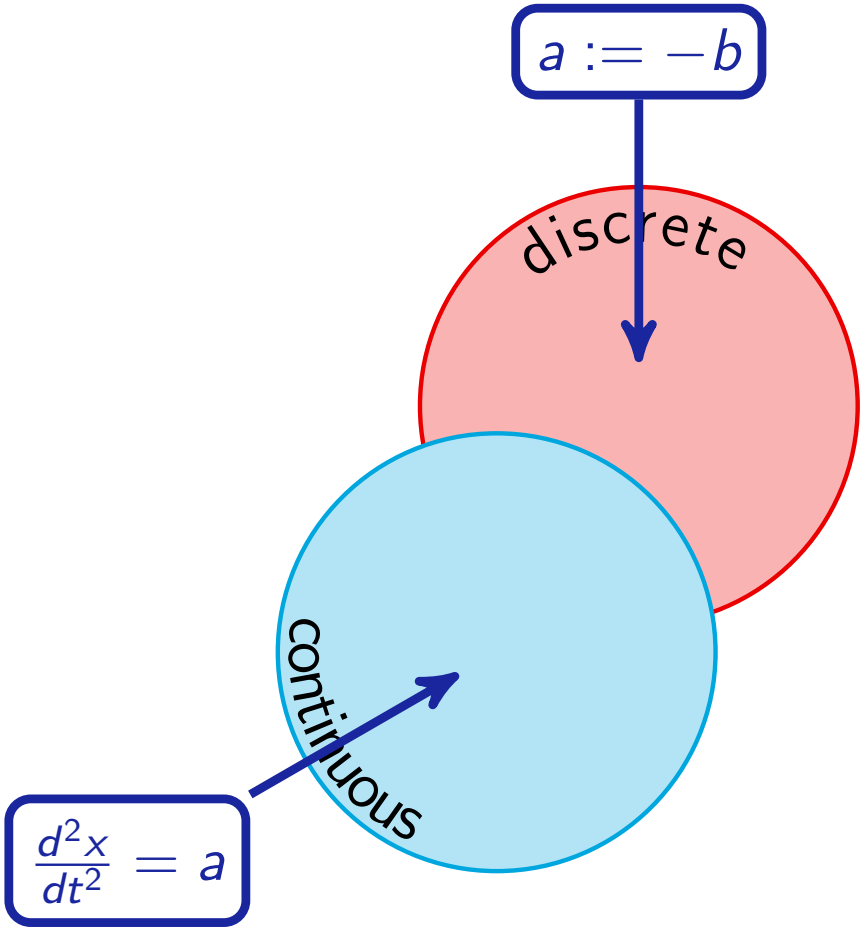
- Continuous dynamics (differential equations)
- Discrete dynamics (control decisions)
- Stochastic dynamics (uncertainty)
- Discrete stochastic (lossy communication)
- Continuous stochastic (wind, track)



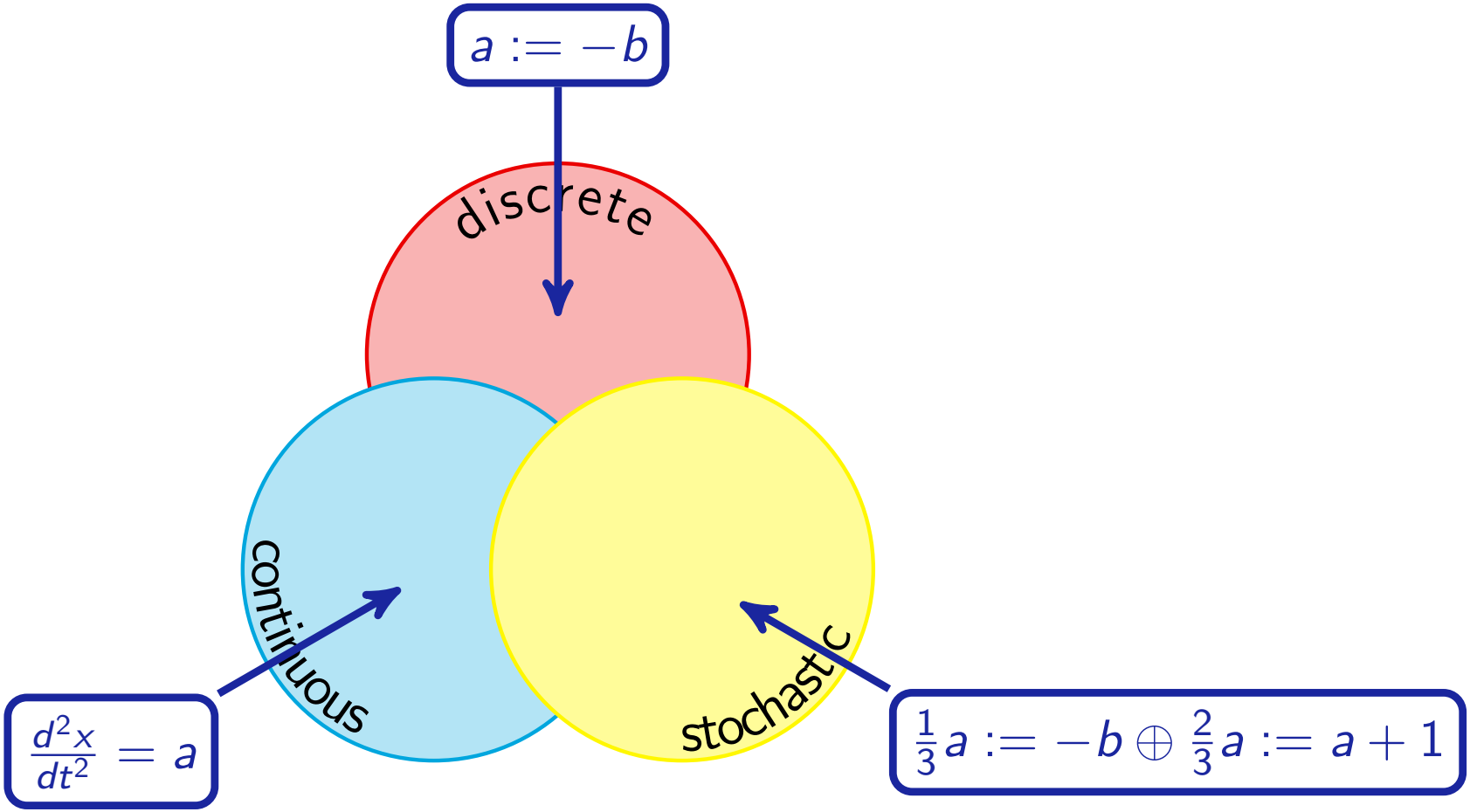


Model for Stochastic Hybrid Systems

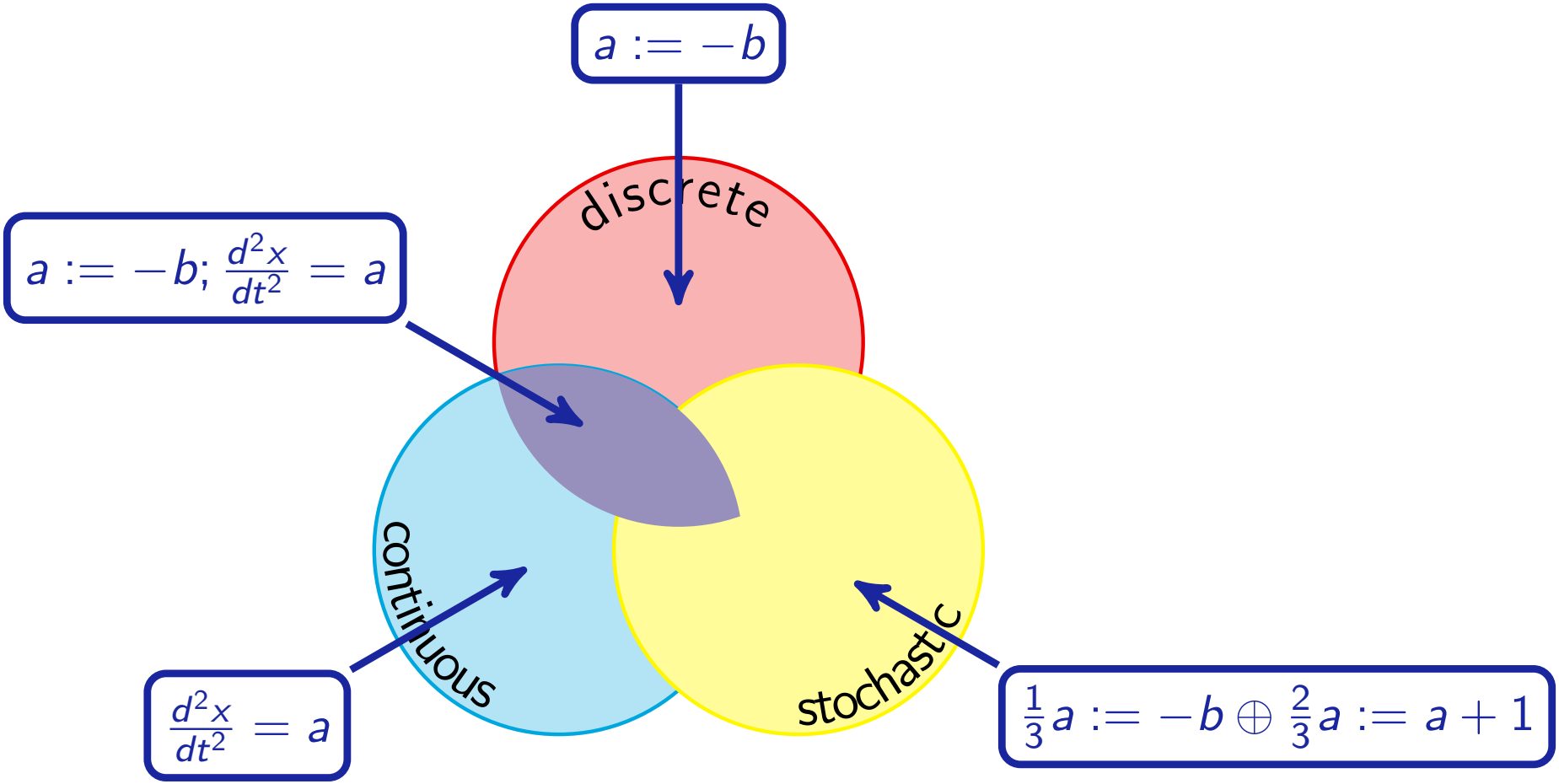



$a := -b$
discrete
continuous
$\frac{d^2x}{dt^2} = a$

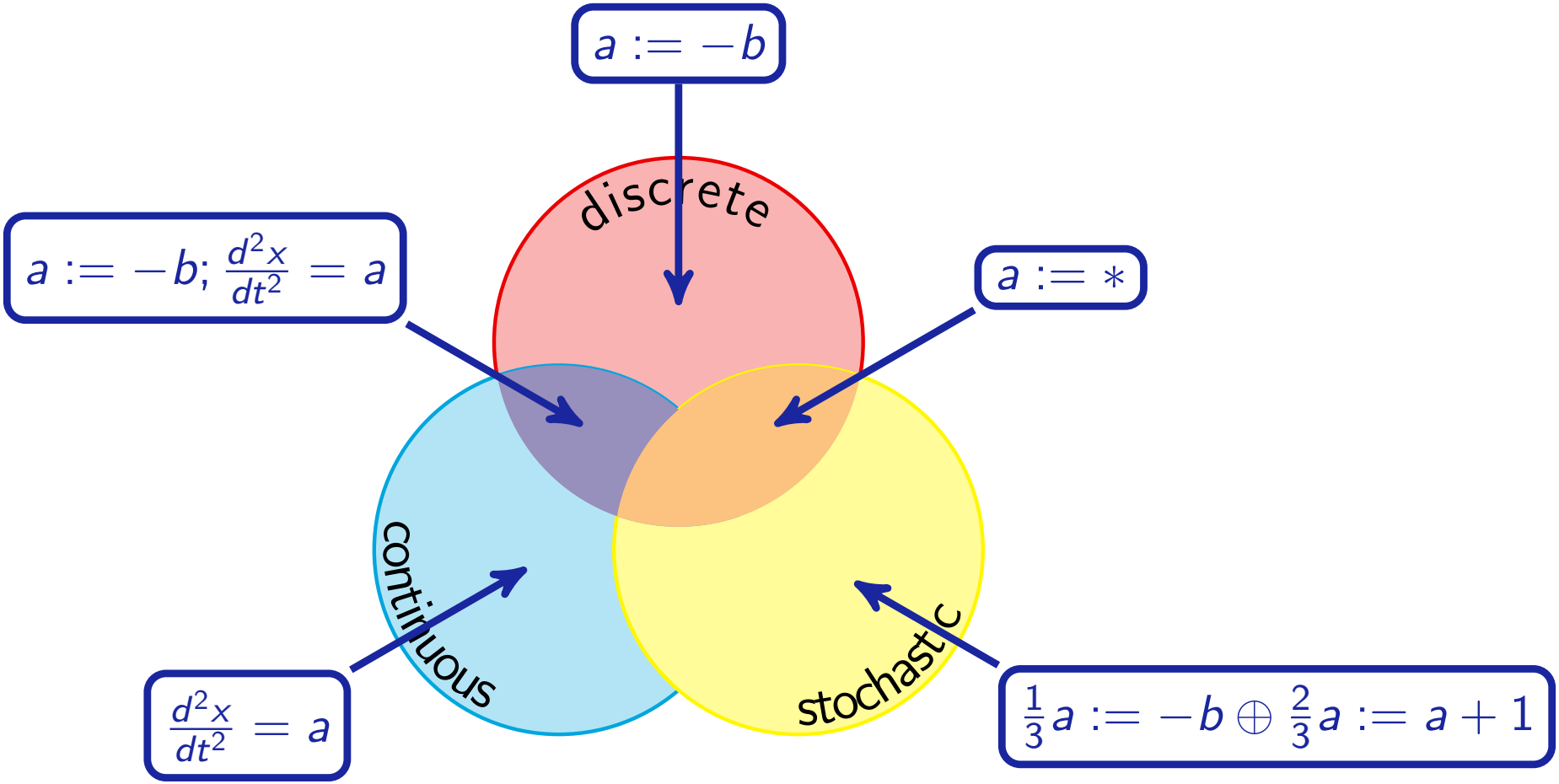
Model for Stochastic Hybrid Systems



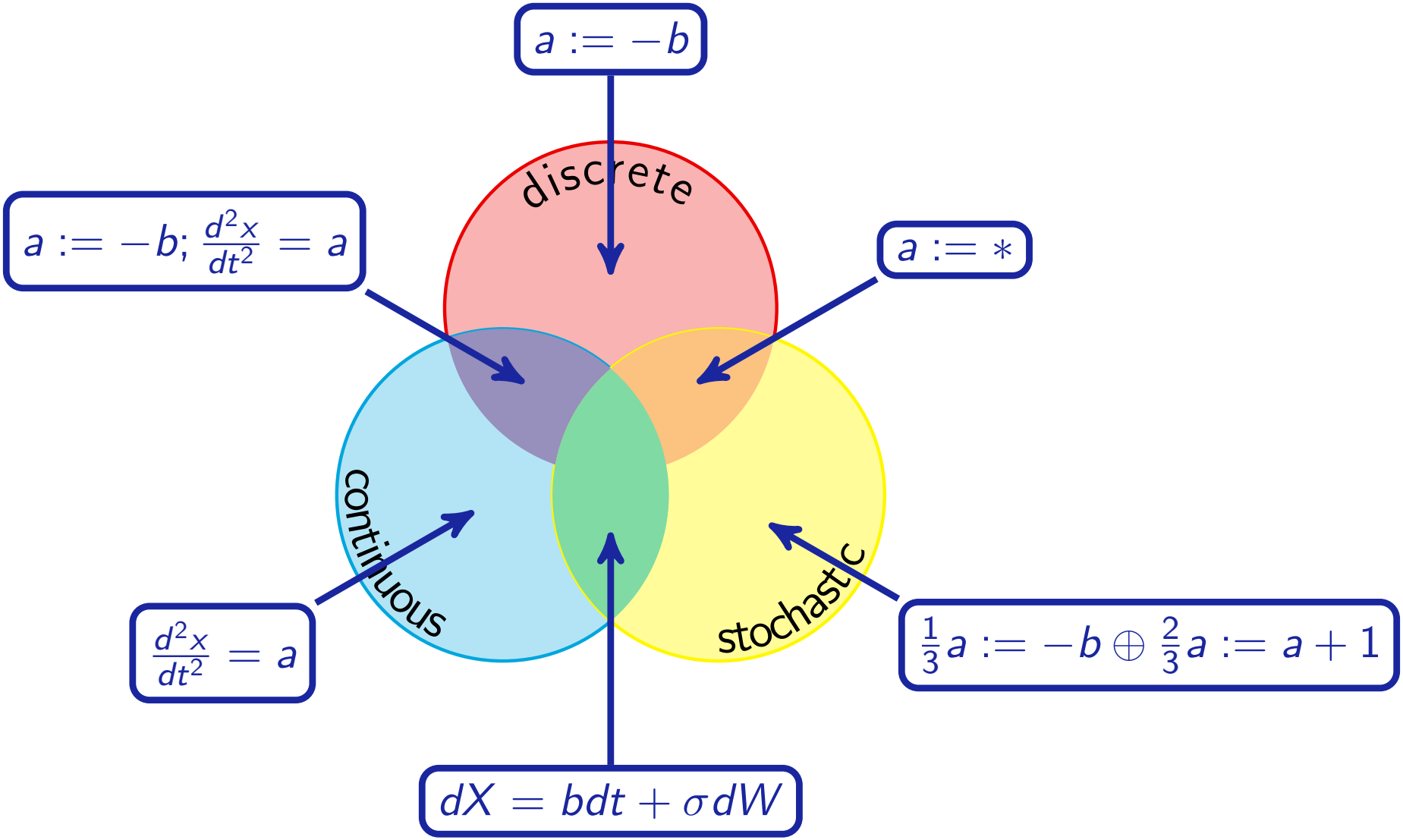
Model for Stochastic Hybrid Systems



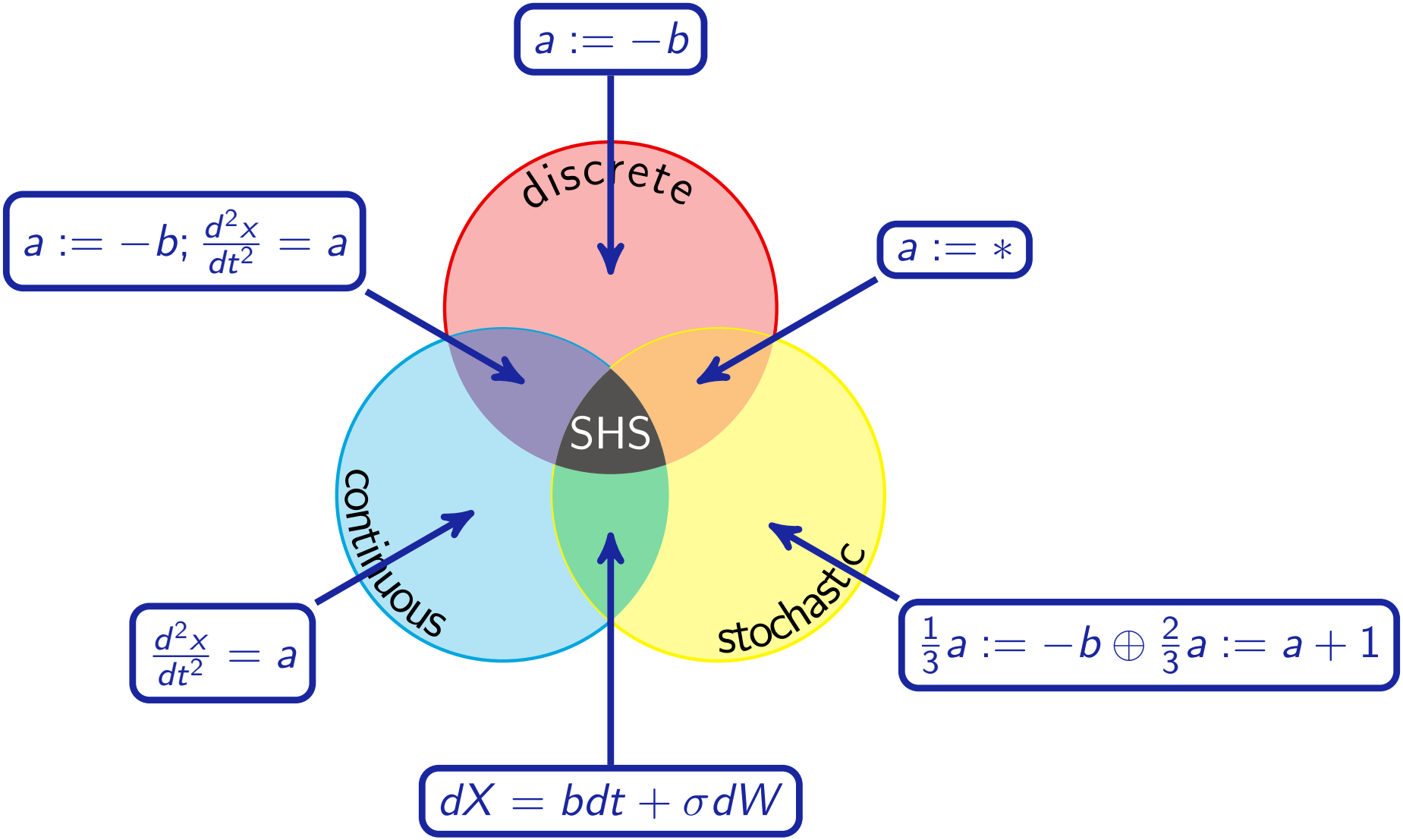
Model for Stochastic Hybrid Systems



Model for Stochastic Hybrid Systems

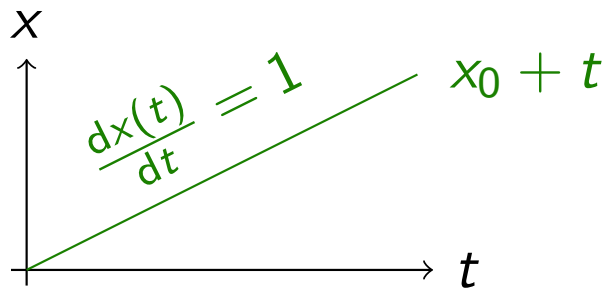


Model for Stochastic Hybrid Systems



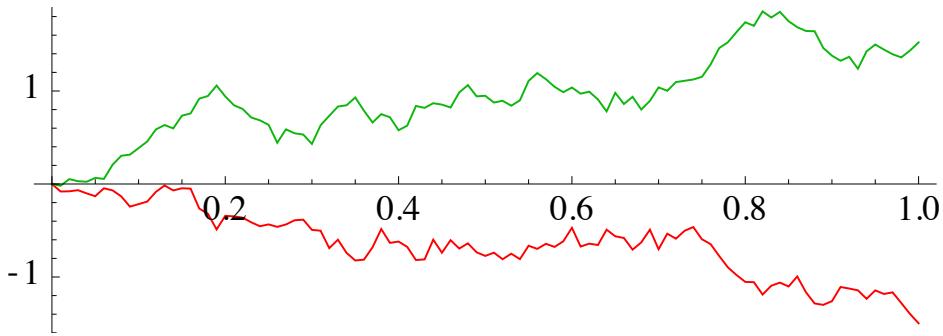
Definition (Ordinary differential equation (ODE))

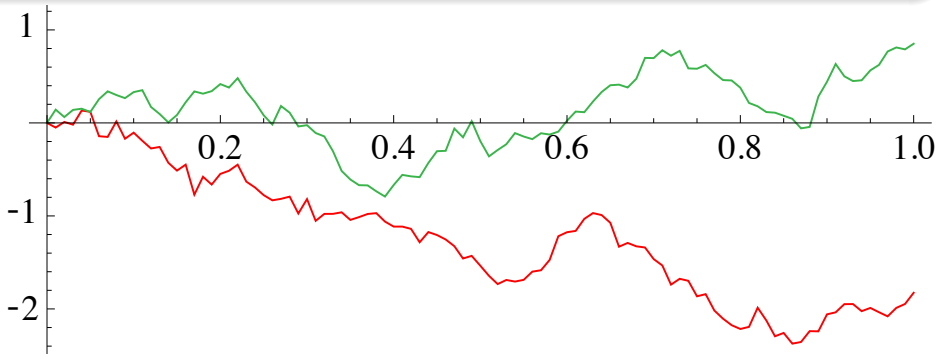
$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} = b(x(t)) \quad x(0) = x_0$$



Definition (Itō stochastic differential equation (SDE))

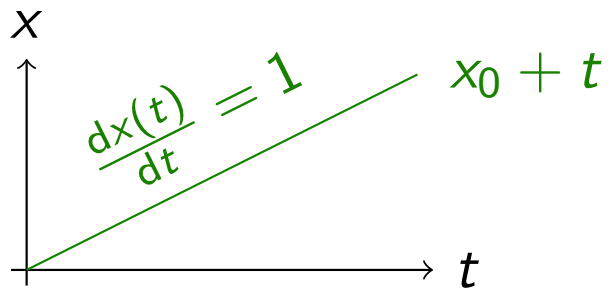
$$dX_t = b(X_t)dt + \sigma(X_t)dW_t \quad X_0 = Z$$





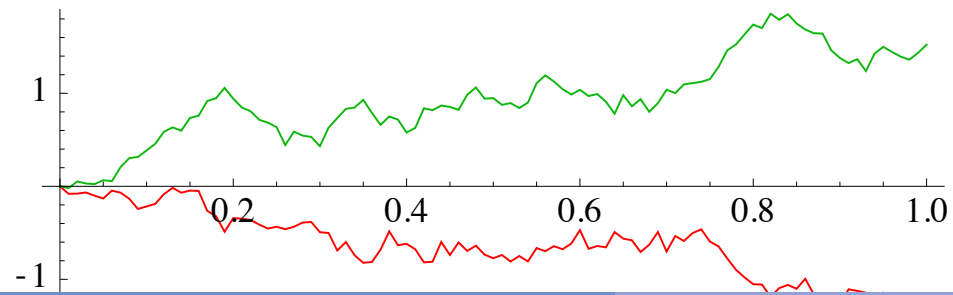
Definition (Ordinary differential equation (ODE))

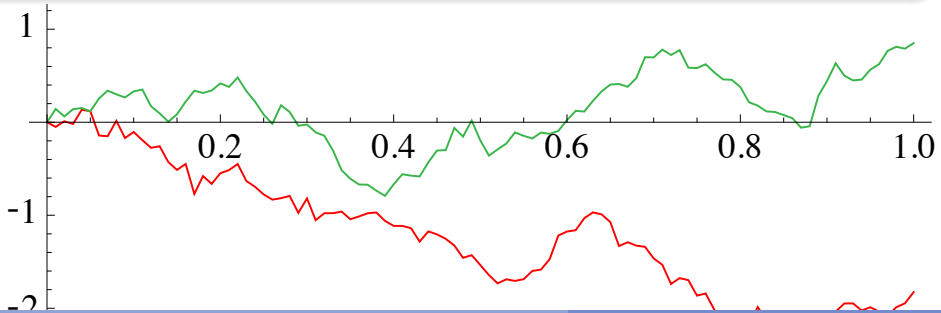
$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} = b(x(t)) \quad x(0) = x_0$$



Definition (Itō stochastic differential equation (SDE))

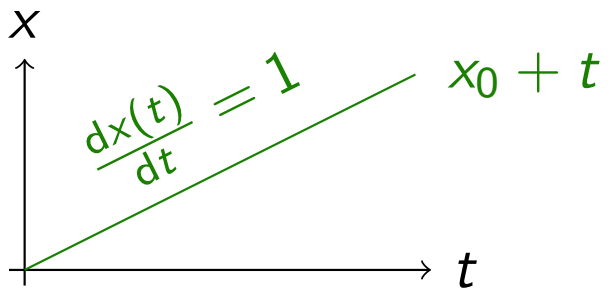
$$X_s = Z + \int_0^s dX_t = Z + \int_0^s b(X_t)dt + \int_0^s \sigma(X_t)dW_t$$





Definition (Ordinary differential equation (ODE))

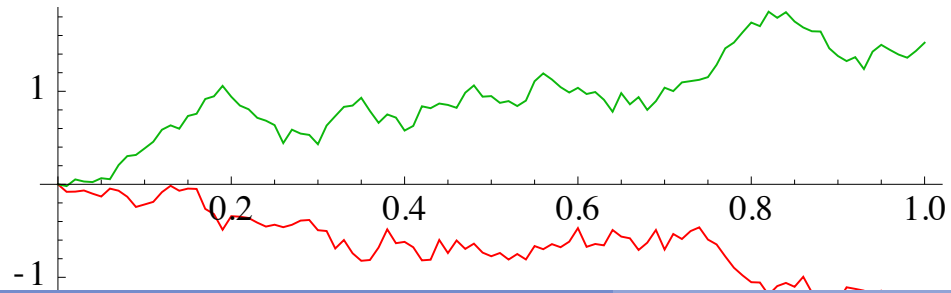
$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} = b(x(t)) \quad x(0) = x_0$$

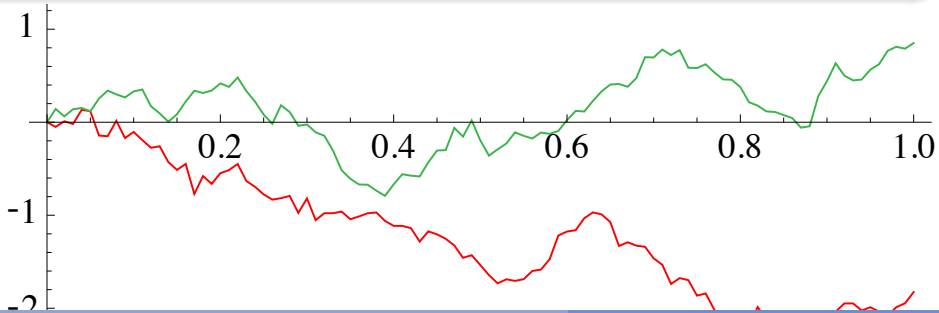


Calculus

Definition (Itō stochastic differential equation (SDE))

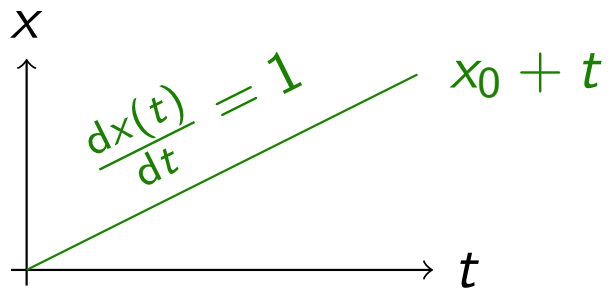
$$X_s = Z + \int_0^s dX_t = Z + \int_0^s b(X_t)dt + \int_0^s \sigma(X_t)dW_t$$

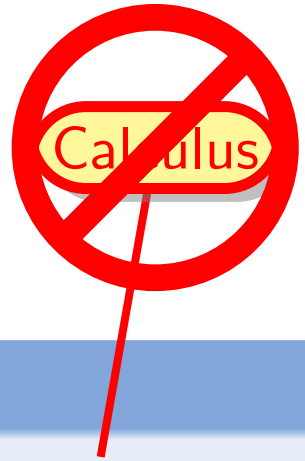




Definition (Ordinary differential equation (ODE))

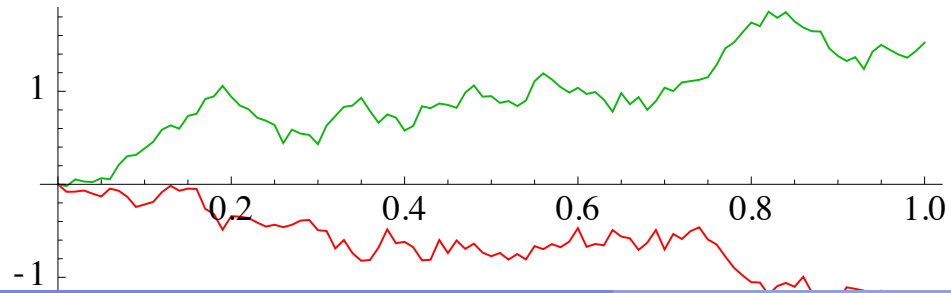
$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} = b(x(t)) \quad x(0) = x_0$$

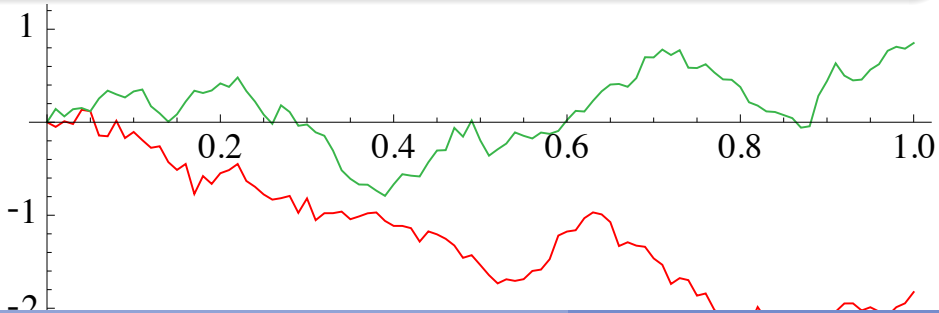




Definition (Itō stochastic differential equation (SDE))

$$X_s = Z + \int_0^s dX_t = Z + \int_0^s b(X_t)dt + \int_0^s \sigma(X_t)dW_t$$





Stochastic Hybrid Programs: Syntax

Definition (Stochastic hybrid program α)

$x := \theta$	(assignment)	jump & test
$x := *$	(random assignment)	
$?H$	(conditional execution)	
$dx = bdt + \sigma dW \,\&\, H$	(SDE)	
$\alpha; \beta$	(seq. composition)	algebra
$\lambda\alpha \oplus \nu\beta$	(convex combination)	
α^*	(nondet. repetition)	