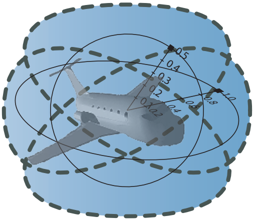
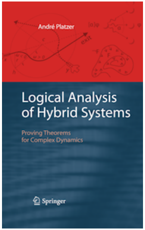
# Hybrid Systems Verification and Robotics

André Platzer

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

http://symbolaris.com/





# Outline

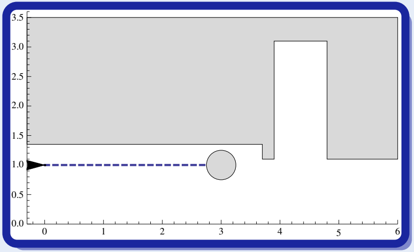
## Can you trust a computer to control physics?

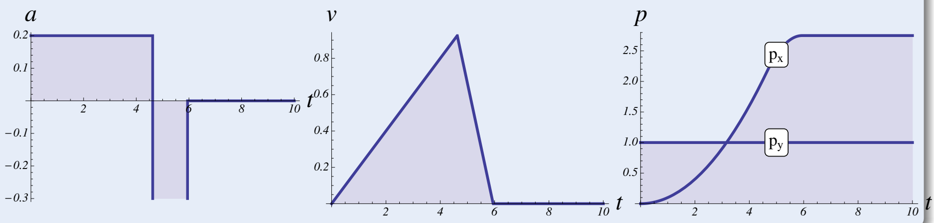
# Outline

## Challenge (Hybrid Systems)

Fixed rule describing state evolution with both

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

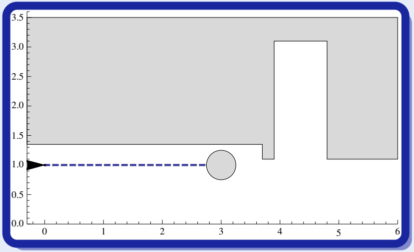


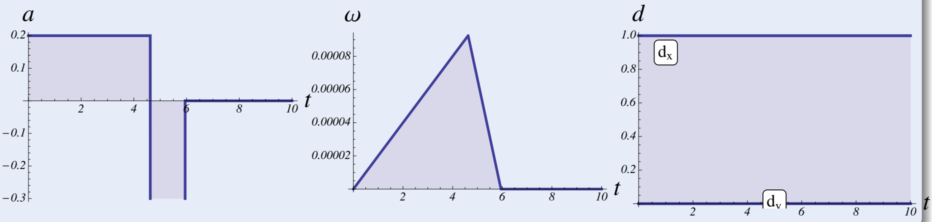



## Challenge (Hybrid Systems)

Fixed rule describing state evolution with both

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



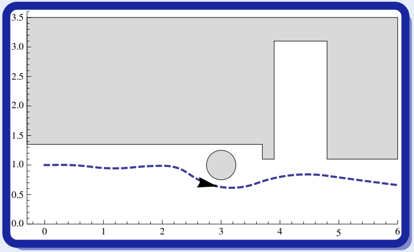



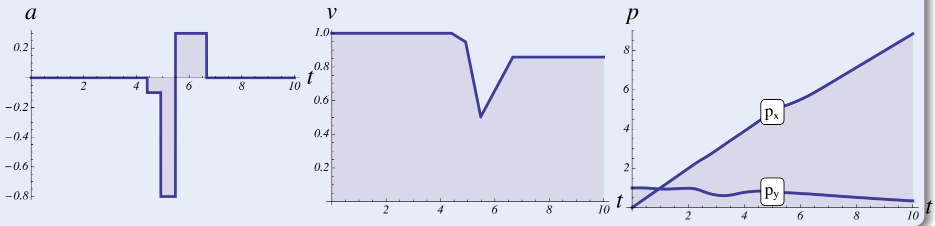
# Hybrid Systems Analysis

## Challenge (Hybrid Systems)

Fixed rule describing state evolution with both

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

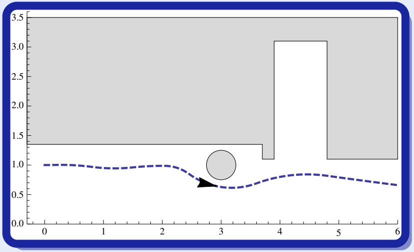


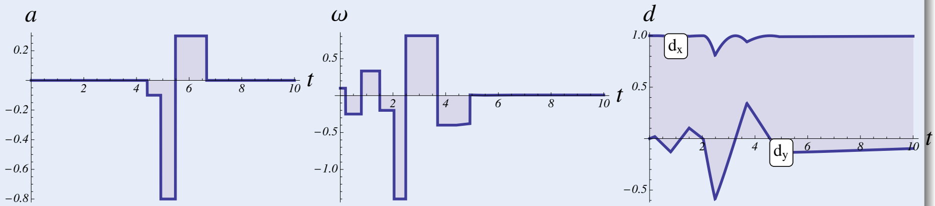



## Challenge (Hybrid Systems)

Fixed rule describing state evolution with both

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

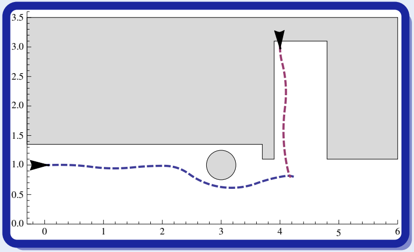


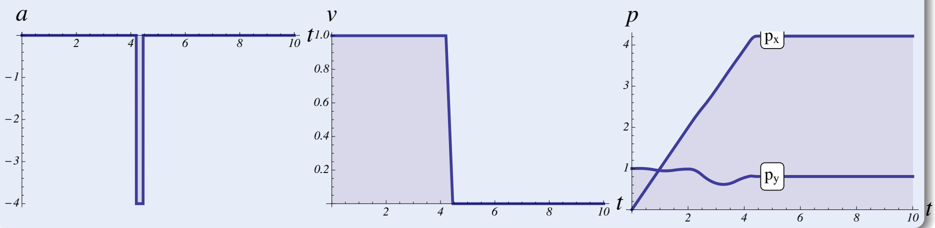



## Challenge (Hybrid Systems)

Fixed rule describing state evolution with both

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

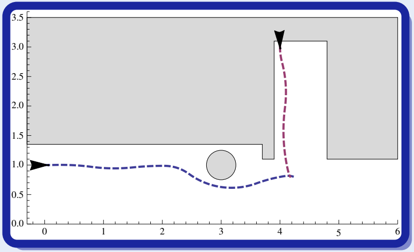


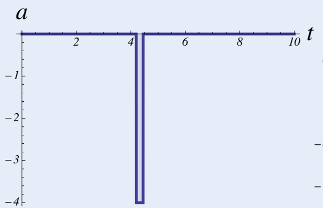


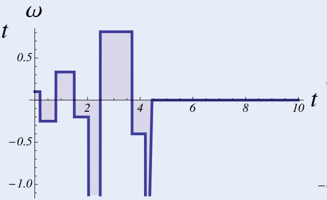
## Challenge (Hybrid Systems)

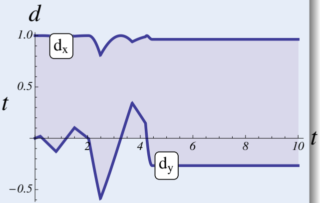
Fixed rule describing state evolution with both

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





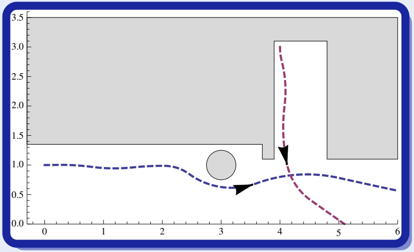



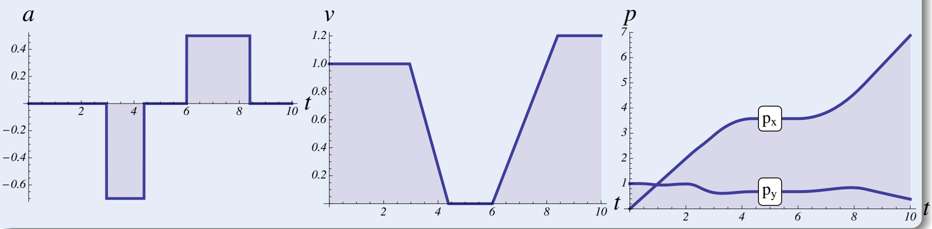



## Challenge (Hybrid Systems)

Fixed rule describing state evolution with both

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

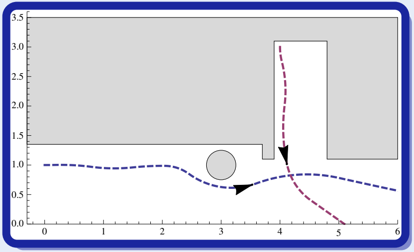


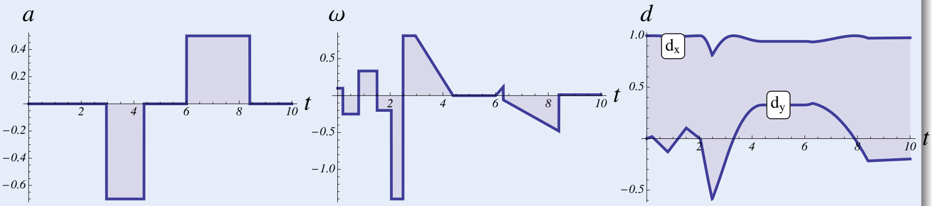


## Challenge (Hybrid Systems)

Fixed rule describing state evolution with both

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



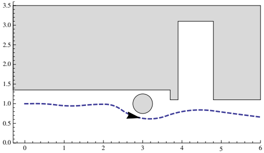


# Outline

# Logic for Hybrid Systems

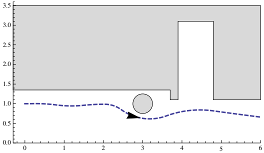
**differential dynamic logic**

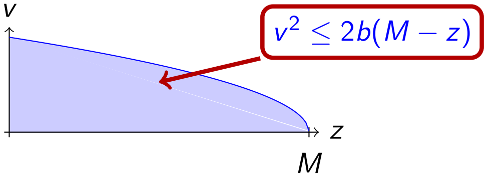
$$d\mathcal{L} = \qquad DL + HP$$



**differential dynamic logic**

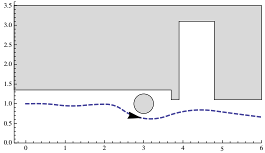
$d\mathcal{L} = \mathrm{FOL}_{\mathbb{R}}$

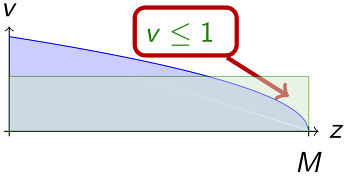


**differential dynamic logic**

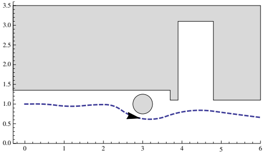
$d\mathcal{L} = \mathrm{FOL}_{\mathbb{R}}$

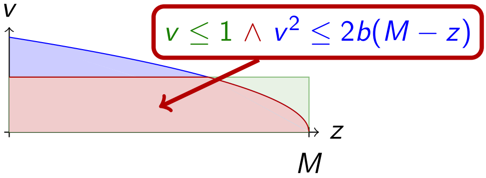


**differential dynamic logic**

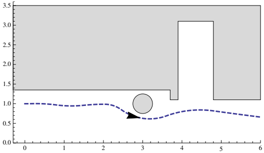
$d\mathcal{L} = \mathrm{FOL}_{\mathbb{R}}$

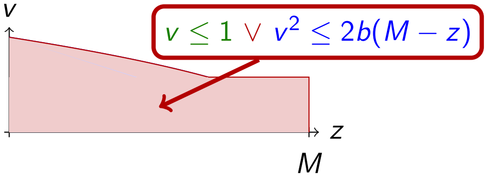


**differential dynamic logic**

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

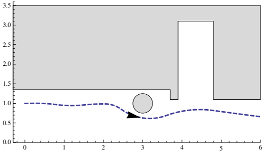


# Logic for Hybrid Systems

**differential dynamic logic**

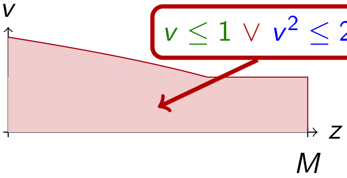
$\mathsf{d}\mathcal{L} = \mathrm{FOL}_{\mathbb{R}}$

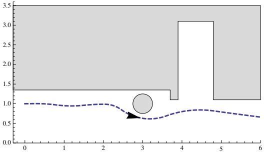



$\forall M\, \exists SB \ \ldots$

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

$v \leq 1 \lor v^2 \leq 2b(M - z)$

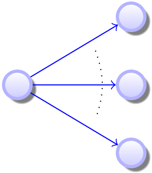





**differential dynamic logic**

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

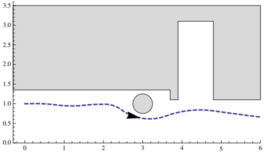
$v^2 \leq 2b$

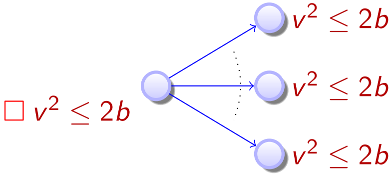


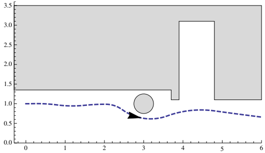
# Logic for Hybrid Systems

**differential dynamic logic**

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



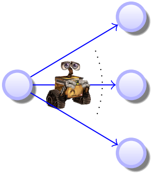


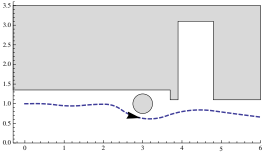



**differential dynamic logic**

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

[  ] $v^2 \leq 2b$

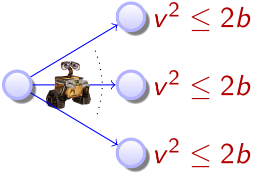




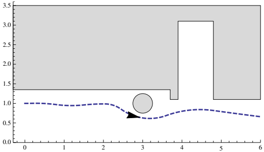

**differential dynamic logic**

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

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



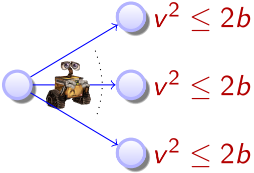
# Logic for Hybrid Systems

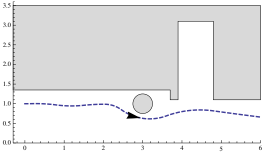



**differential dynamic logic**

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

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

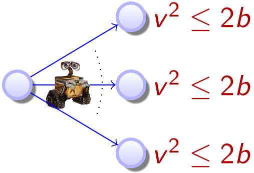




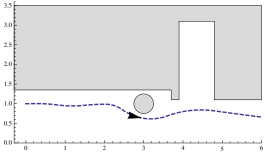

**differential dynamic logic**

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

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



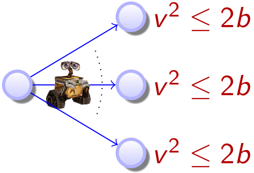
# Logic for Hybrid Systems



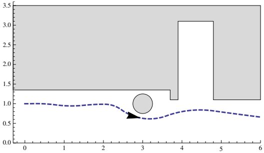

**differential dynamic logic**

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

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



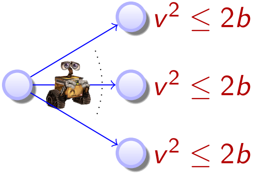
# Logic for Hybrid Systems




**differential dynamic logic**

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

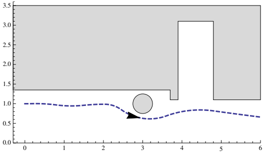
$$\mathcal{C} \rightarrow [\underbrace{\texttt{if}(z > SB)\, a := -b;\ z'' = a}_{\text{hybrid program}}]\, v^2 \leq 2b$$



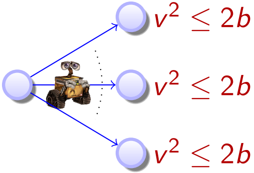

Initial condition

**differential dynamic logic**

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

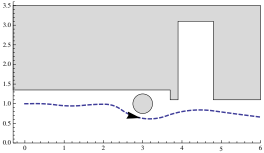



$$\mathcal{C} \rightarrow [\underbrace{\mathtt{if}(z > SB)\, a := -b;\ z'' = a}_{\text{hybrid program}}]\, v^2 \le 2b$$




Initial condition

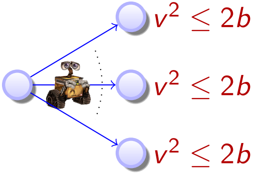
System dynamics



**differential dynamic logic**

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

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




Initial condition

System dynamics

Post condition

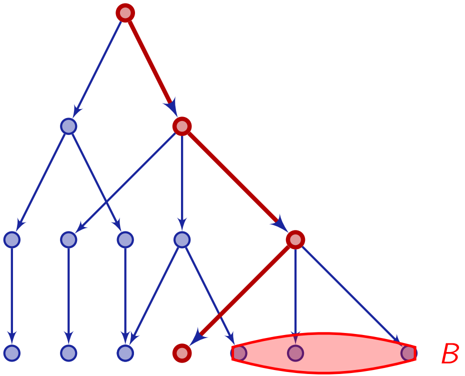
# Outline

Follow all transitions of the system
from a set of states
$\approx$ set-valued simulation

# Model Checking in a Nutshell

**Definition (Model Checking Problem)**

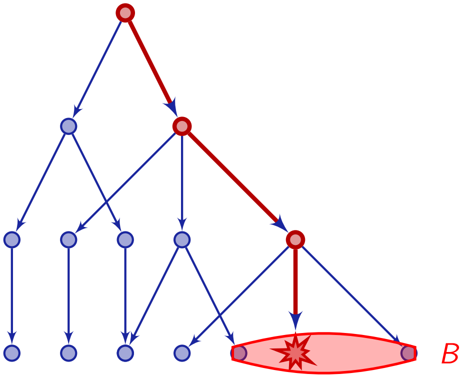
Given initial states $Q_0 \subseteq Q$ and bad states $B \subseteq Q$ for a transition system, check whether there is a trace from some $q_0 \in Q_0$ to some $q_b \in B$.



# Model Checking in a Nutshell

## Definition (Model Checking Problem)

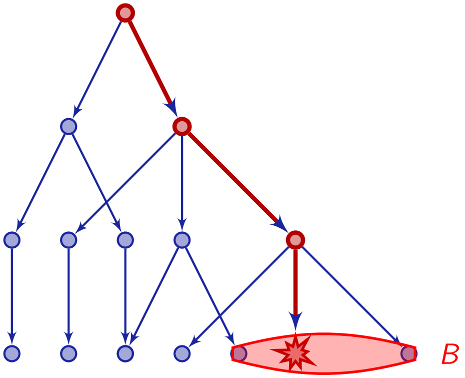
Given initial states $Q_0 \subseteq Q$ and bad states $B \subseteq Q$ for a transition system, check whether there is a trace from some $q_0 \in Q_0$ to some $q_b \in B$.



# Model Checking in a Nutshell

## Definition (Image Computation)

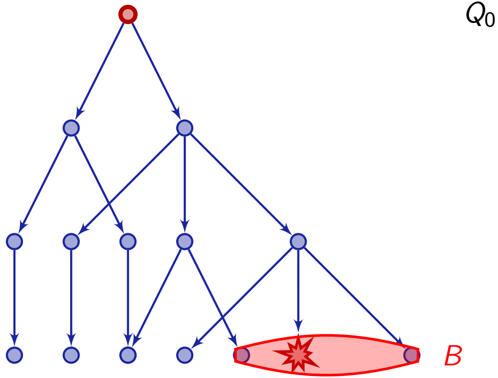
$$Post_A(Y) \ := \ \{q^+ \in Q \ : \ q \xrightarrow{a} q^+ \text{ for some } q \in Y, a \in A\}$$



# Model Checking in a Nutshell

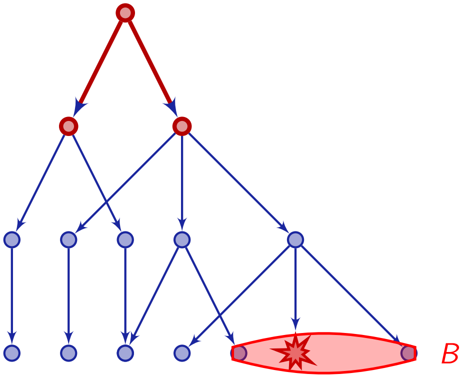
## Definition (Image Computation)

$$Post_A(Y) := \{q^+ \in Q : q \xrightarrow{a} q^+ \text{ for some } q \in Y, a \in A\}$$



**Definition (Image Computation)**

$$Post_A(Y) := \{q^+ \in Q : q \xrightarrow{a} q^+ \text{ for some } q \in Y, a \in A\}$$



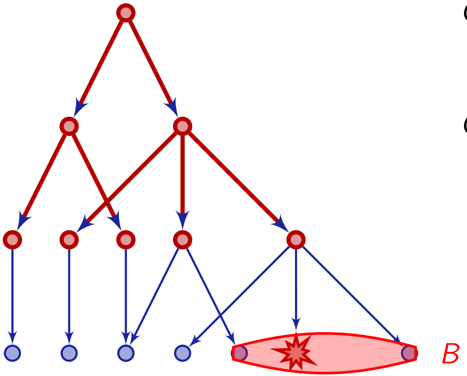

$Q_0$

$Post_A(Q_0)$

$Q_1 = Post_A(Q_0)$

## Definition (Image Computation)

$$Post_A(Y) := \{q^+ \in Q : q \xrightarrow{a} q^+ \text{ for some } q \in Y, a \in A\}$$




$Q_0$

$Post_A(Q_0)$

$Q_1 = Post_A(Q_0)$

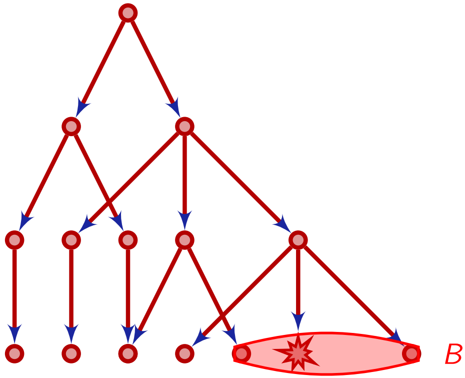
$Post_A(Q_1)$

$Q_2 = Post_A^2(Q_0)$

# Model Checking in a Nutshell

**Definition (Image Computation)**

$$Post_A(Y) \; := \; \{q^+ \in Q \; : \; q \xrightarrow{a} q^+ \text{ for some } q \in Y, a \in A\}$$




$Q_0$

$\downarrow Post_A(Q_0)$

$Q_1 = Post_A(Q_0)$

$\downarrow Post_A(Q_1)$

$Q_2 = Post_A^2(Q_0)$

$\downarrow Post_A(Q_2)$

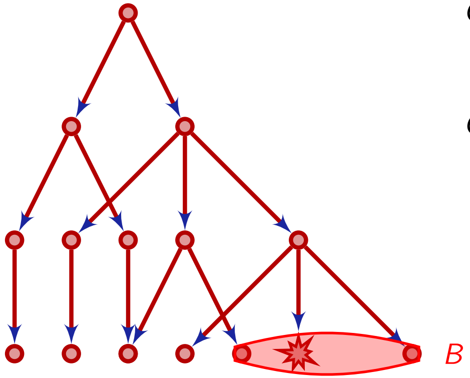
$Q_3 = Post_A^3(Q_0)$

# Model Checking in a Nutshell

## Definition (Image Computation)

$$Post_A(Y) := \{q^+ \in Q : q \xrightarrow{a} q^+ \text{ for some } q \in Y, a \in A\}$$

$$Post^*_A(Y) := \bigcup_{n \in \mathbb{N}} Post^n_A(Y) = \mu Z.(Y \cup Z \cup Post_A(Z))$$




$Q_0$

$Post_A(Q_0)$

$Q_1 = Post_A(Q_0)$

$Post_A(Q_1)$

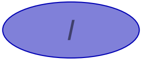
$Q_2 = Post^2_A(Q_0)$

$Post_A(Q_2)$

$Q_3 = Post^3_A(Q_0)$

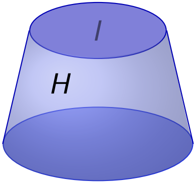
## Uncountably state spaces require extra care

# Image Computation in Hybrid Systems



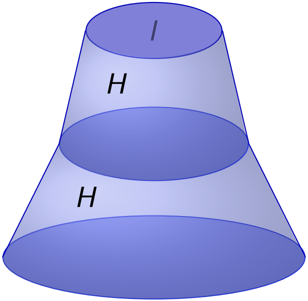

- Analyse image computation problem in hybrid systems
- Approximation refinement techniques and their limits
- Representation of regions in state space
- Numerical versus symbolic algorithms
  $1.421 \in \mathbb{Q}$ versus $x^2 + 2xy$ term computations

# Image Computation in Hybrid Systems



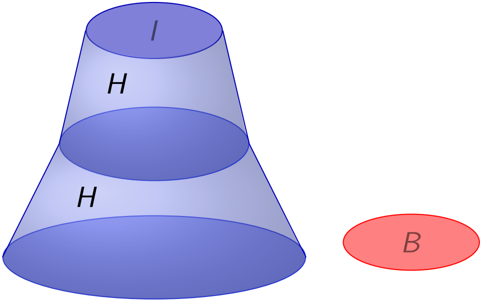

- Analyse image computation problem in hybrid systems
- Approximation refinement techniques and their limits
- Representation of regions in state space
- Numerical versus symbolic algorithms
  $1.421 \in \mathbb{Q}$ versus $x^2 + 2xy$ term computations

# Image Computation in Hybrid Systems



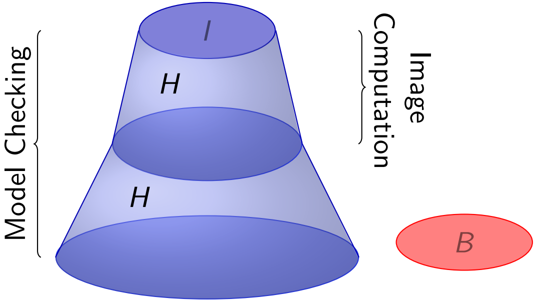

- Analyse image computation problem in hybrid systems
- Approximation refinement techniques and their limits
- Representation of regions in state space
- Numerical versus symbolic algorithms
  $1.421 \in \mathbb{Q}$ versus $x^2 + 2xy$ term computations

# Image Computation in Hybrid Systems




- Analyse image computation problem in hybrid systems
- Approximation refinement techniques and their limits
- Representation of regions in state space
- Numerical versus symbolic algorithms
  $1.421 \in \mathbb{Q}$ versus $x^2 + 2xy$ term computations

# Image Computation in Hybrid Systems

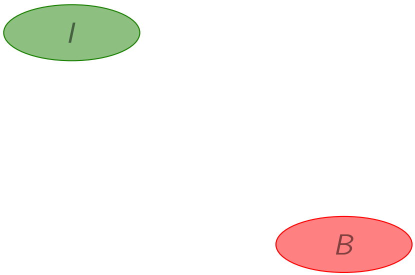



- Analyse image computation problem in hybrid systems
- Approximation refinement techniques and their limits
- Representation of regions in state space
- Numerical versus symbolic algorithms
  $1.421 \in \mathbb{Q}$ versus $x^2 + 2xy$ term computations

# AMC: Approximation Refinement Model Checking

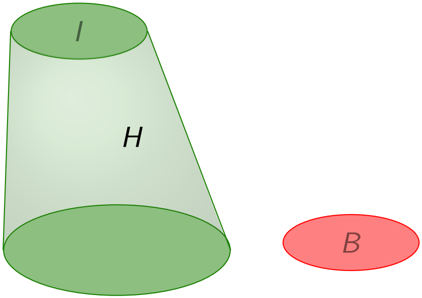
AMC($B$ reachable from $I$ in $H$):

1. $A :=$ approx($H$) uniformly
2. blur by uniform approximation error $+\epsilon$
3. check($B$ reachable from $I$ in $A + \epsilon$)
4. $B$ not reachable $\Rightarrow$ $H$ safe



AMC($B$ reachable from $I$ in $H$):

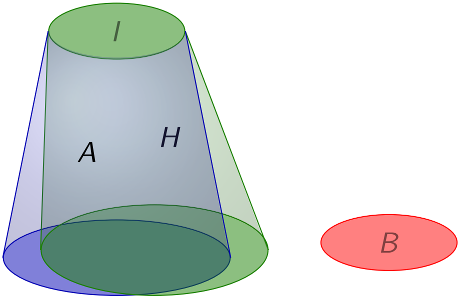
1. $A :=$ approx($H$) uniformly
2. blur by uniform approximation error $+\epsilon$
3. check($B$ reachable from $I$ in $A + \epsilon$)
4. $B$ not reachable $\Rightarrow$ $H$ safe



# AMC: Approximation Refinement Model Checking

AMC($B$ reachable from $I$ in $H$):

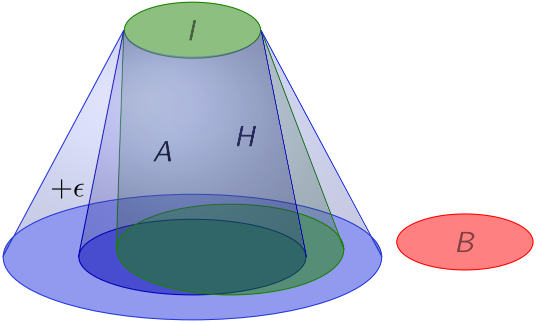
1. $A :=$ approx($H$) uniformly
2. blur by uniform approximation error $+\epsilon$
3. check($B$ reachable from $I$ in $A + \epsilon$)
4. $B$ not reachable $\Rightarrow$ $H$ safe



# AMC: Approximation Refinement Model Checking

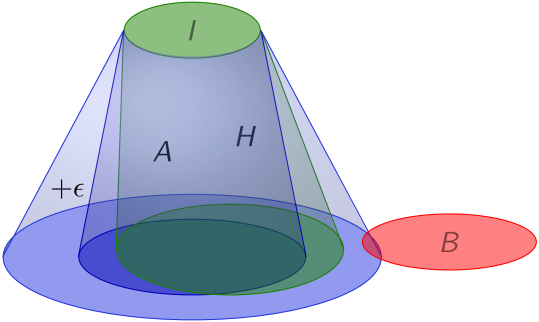
AMC($B$ reachable from $I$ in $H$):

1. $A :=$ approx($H$) uniformly
2. blur by uniform approximation error $+\epsilon$
3. check($B$ reachable from $I$ in $A + \epsilon$)
4. $B$ not reachable $\Rightarrow$ $H$ safe



AMC($B$ reachable from $I$ in $H$):

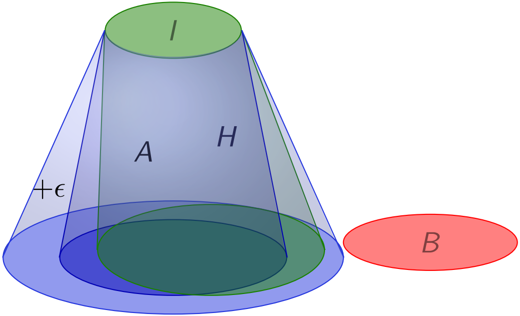
1. $A :=$ approx($H$) uniformly
2. blur by uniform approximation error $+\epsilon$
3. check($B$ reachable from $I$ in $A + \epsilon$)
4. $B$ not reachable $\Rightarrow$ $H$ safe



# AMC: Approximation Refinement Model Checking

AMC($B$ reachable from $I$ in $H$):

1. $A :=$ approx($H$) uniformly
2. blur by uniform approximation error $+\epsilon$
3. check($B$ reachable from $I$ in $A + \epsilon$)
4. $B$ not reachable $\Rightarrow$ $H$ safe



# AMC: Exact Image Computation

AMC($B$ reachable from $I$ in $H$):

1. $A :=$ approx($H$) uniformly
2. blur by uniform approximation error $+\epsilon$
3. check($B$ reachable from $I$ in $A + \epsilon$)
4. $B$ not reachable $\Rightarrow$ $H$ safe

**Proposition (Semialgebraic images)** (HSCC'07)

*check and blur can be implemented for*

- *$I$ and $B$ semialgebraic (propositional combinations of $p \geq 0$)*
- *$A$ with polynomial flows over $\mathbb{R}$*
- *$+$Piecewise definitions*
- *$+$Rational extensions (e.g. multivariate rational splines)*

# AMC: Image Approximation

AMC($B$ reachable from $I$ in $H$):

1. $A :=$ approx($H$) uniformly
2. blur by uniform approximation error $+\epsilon$
3. check($B$ reachable from $I$ in $A + \epsilon$)
4. $B$ not reachable $\Rightarrow$ $H$ safe

**Proposition (*Existence* of approximations) (HSCC'07)**

*approx exists for all uniform errors $\epsilon > 0$ when*

- *using polynomials to build $A$*
- *Flows $\varphi \in C(D, \mathbb{R}^n)$ of $H$*
- *$D \subset \mathbb{R} \times \mathbb{R}^n$ compact closure of an open set*

Approximation can solve problems without effective exact solution

Existence of solutions may be computationally insufficient

# Summary: Model Checking

- Image computation in hybrid systems model checking HSCC'07
  1. approx uniformly
  2. blur by uniform error
  3. check for $B$

| flows | approx / image computation |
|---|---|
| continuous | uniform approx exists, but... |
| smooth | undecidable by evaluation |
| bounded by $b$ | decidable |
| bound probabilities | probabilistically decidable |
| ODE $\ell$-Lipschitz | decidable |

- Combine numerical algorithms with symbolic analysis
- Roundabout maneuver unsafe

# Outline

## Verify using many simple symbolic proof steps

# Differential Dynamic Logic: Axiomatization

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

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

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

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

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

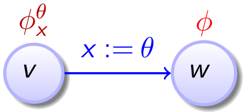
$$[^*] \quad [\alpha^*]\phi \leftrightarrow \phi \wedge [\alpha][\alpha^*]\phi$$

$$\mathsf{K} \quad [\alpha](\phi \rightarrow \psi) \rightarrow ([\alpha]\phi \rightarrow [\alpha]\psi)$$

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

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

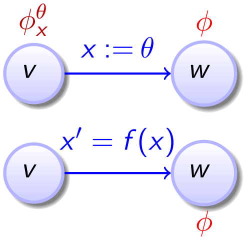
$$\frac{\phi_x^\theta}{[x := \theta]\phi}$$



# Proofs for Hybrid Systems

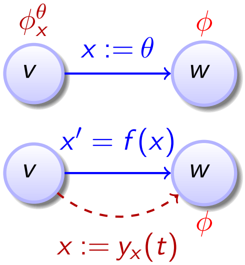
$$\frac{\phi_x^\theta}{[x := \theta]\phi}$$

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



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

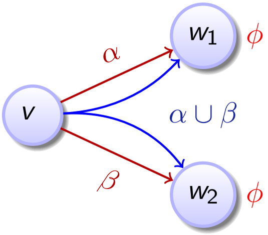
$$\frac{\forall t{\geq}0\, [x := y_x(t)]\phi}{[x' = f(x)]\phi}$$



compositional semantics $\Rightarrow$ compositional rules!

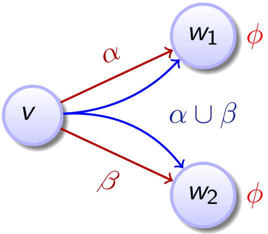
# Proofs for Hybrid Systems

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

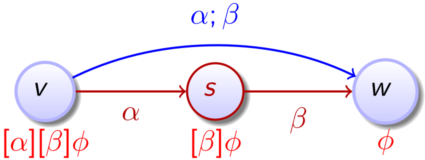


# Proofs for Hybrid Systems

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

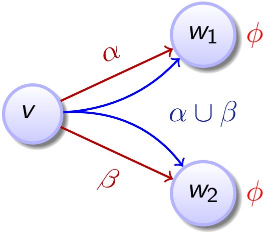



$$\frac{[\alpha][\beta]\phi}{[\alpha; \beta]\phi}$$

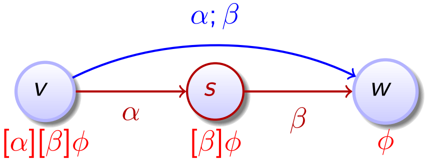


# Proofs for Hybrid Systems

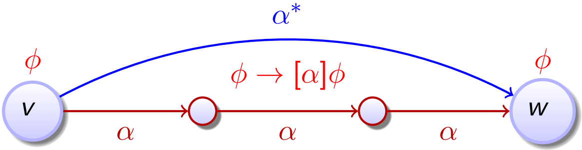
$$\frac{[\alpha]\phi \land [\beta]\phi}{[\alpha \cup \beta]\phi}$$



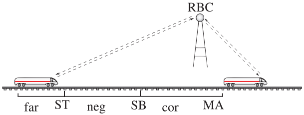

$$\frac{[\alpha][\beta]\phi}{[\alpha; \beta]\phi}$$




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



# Deduction Modulo (Free Variables for Automation)

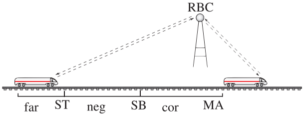



$$v \geq 0, z < m \rightarrow \exists t \geq 0 \, \langle z := -\frac{b}{2}t^2 + vt + z \rangle z > m$$

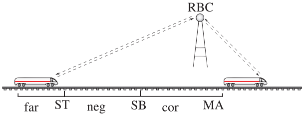
$$v \geq 0, z < m \rightarrow \langle z' = v, v' = -b \rangle z > m$$

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

# Deduction Modulo (Free Variables for Automation)




$$\dfrac{v \geq 0, z < m \to T \geq 0 \qquad \dfrac{v \geq 0, z < m \to -\frac{b}{2}T^2 + vT + z > m}{v \geq 0, z < m \to \langle z := -\frac{b}{2}T^2 + vT + z\rangle z > m}}{\dfrac{v \geq 0, z < m \to T \geq 0 \land \langle z := -\frac{b}{2}T^2 + vT + z\rangle z > m}{\dfrac{v \geq 0, z < m \to \exists t{\geq}0\, \langle z := -\frac{b}{2}t^2 + vt + z\rangle z > m}{\dfrac{v \geq 0, z < m \to \langle z' = v, v' = -b\rangle z > m}{v \geq 0 \land z < m \to \langle z' = v, v' = -b\rangle z > m}}}}$$




$$v \geq 0, z < m \rightarrow \quad \exists T\,(\ldots T \geq 0 \land -\frac{b}{2}T^2 + vT + z > m)$$

$$v \geq 0, z < m \rightarrow -\frac{b}{2}T^2 + vT + z > m$$

$$v \geq 0, z < m \rightarrow T \geq 0 \qquad v \geq 0, z < m \rightarrow \langle z := -\frac{b}{2}T^2 + vT + z\rangle z > m$$

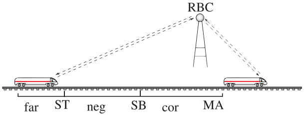
$$v \geq 0, z < m \rightarrow T \geq 0 \land \langle z := -\frac{b}{2}T^2 + vT + z\rangle z > m$$

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

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

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

# Deduction Modulo (Free Variables for Automation)




$$v \geq 0, z < m \rightarrow \mathrm{QE}\left(\exists T\,(\ldots\, T{\geq}0 \land -\tfrac{b}{2}T^2 + vT + z > m)\right)$$

$$v \geq 0, z < m \rightarrow -\tfrac{b}{2}T^2 + vT + z > m$$

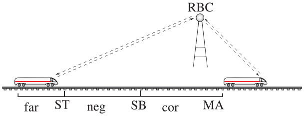
$$v \geq 0, z < m \rightarrow T{\geq}0 \qquad v \geq 0, z < m \rightarrow \langle z := -\tfrac{b}{2}T^2 + vT + z\rangle z > m$$

$$v \geq 0, z < m \rightarrow T \geq 0 \land \langle z := -\tfrac{b}{2}T^2 + vT + z\rangle z > m$$

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

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

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




$$v \geq 0, z < m \to v^2 > 2b(m - z)$$

$$v \geq 0, z < m \to -\frac{b}{2}T^2 + vT + z > m$$

$$v \geq 0, z < m \to T \geq 0 \qquad v \geq 0, z < m \to \langle z := -\frac{b}{2}T^2 + vT + z\rangle z > m$$

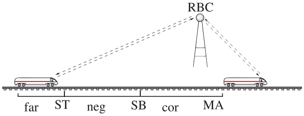
$$v \geq 0, z < m \to T \geq 0 \land \langle z := -\frac{b}{2}T^2 + vT + z\rangle z > m$$

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

$$v \geq 0, z < m \to \langle z' = v, v' = -b\rangle z > m$$

$$v \geq 0 \land z < m \to \langle z' = v, v' = -b\rangle z > m$$

- For requantification, not for unification




$$v \geq 0, z < m \to \mathrm{QE}\left(\exists T\,(\ldots T \geq 0 \land -\tfrac{b}{2}T^2 + vT + z > m)\right)$$

$$v \geq 0, z < m \to -\tfrac{b}{2}T^2 + vT + z > m$$

$$v \geq 0, z < m \to T \geq 0 \qquad v \geq 0, z < m \to \langle z := -\tfrac{b}{2}T^2 + vT + z\rangle z > m$$

$$v \geq 0, z < m \to T \geq 0 \land \langle z := -\tfrac{b}{2}T^2 + vT + z\rangle z > m$$

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

$$v \geq 0, z < m \to \langle z' = v, v' = -b\rangle z > m$$

$$v \geq 0 \land z < m \to \langle z' = v, v' = -b\rangle z > m$$

# Complete Proof Theory of Hybrid Systems

**Theorem (Continuous Relative Completeness) (J.Autom.Reas. 2008)**

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

▸ Proof 15pp

# Complete Proof Theory of Hybrid Systems

**Theorem (Continuous Relative Completeness) (J.Autom.Reas. 2008)**

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

▸ Proof 15pp

**Theorem (Discrete Relative Completeness) (LICS'12)**

d$\mathcal{L}$ *calculus is a sound & complete axiomatization of hybrid systems relative to discrete dynamics.*

▸ Proof +10pp

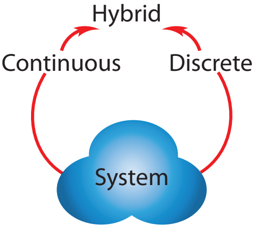
# Complete Proof Theory of Hybrid Systems

**Theorem (Continuous Relative Completeness) (J.Autom.Reas. 2008)**

d$\mathcal{L}$ *calculus is a sound & complete axiomatization of hybrid systems relative to differential equations.* ▸ Proof 15pp

**Theorem (Discrete Relative Completeness) (LICS'12)**

d$\mathcal{L}$ *calculus is a sound & complete axiomatization of hybrid systems relative to discrete dynamics.* ▸ Proof +10pp



# Complete Proof Theory of Hybrid Systems

**Theorem (Continuous Relative Completeness) (J.Autom.Reas. 2008)**

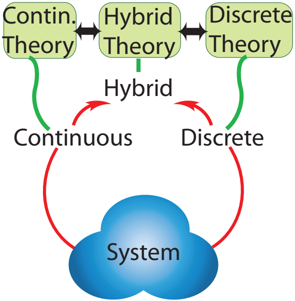
d$\mathcal{L}$ *calculus is a sound & complete axiomatization of hybrid systems relative to differential equations.*

▸ Proof 15pp

**Theorem (Discrete Relative Completeness) (LICS'12)**

d$\mathcal{L}$ *calculus is a sound & complete axiomatization of hybrid systems relative to discrete dynamics.*

▸ Proof +10pp



# Complete Proof Theory of Hybrid Systems

**Theorem (Continuous Relative Completeness) (J.Autom.Reas. 2008)**

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

▸ Proof 15pp

**Theorem (Discrete Relative Completeness) (LICS'12)**

d$\mathcal{L}$ *calculus is a sound & complete axiomatization of hybrid systems relative to discrete dynamics.*

▸ Proof +10pp

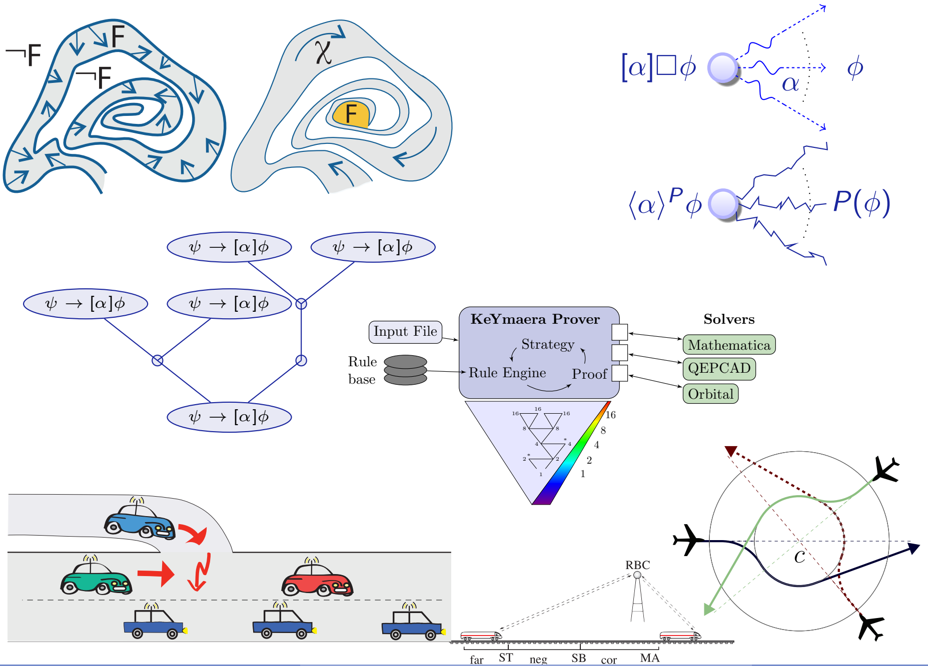
**Corollary (Relative Decidability)**

*Verification & synthesis decidable relative to differential equations.*

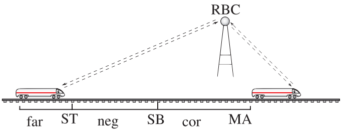
**Corollary (Relative Extension)**

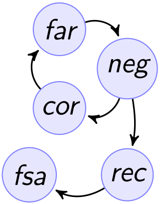
*All research on differential equations extends to hybrid systems.*

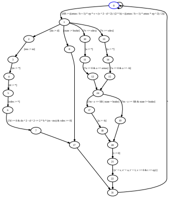
# Outline

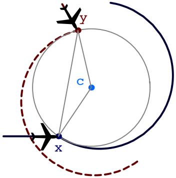


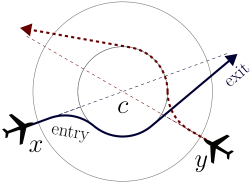
¬F
F
¬F
χ
F
$[\alpha]\Box\phi$
$\alpha$
$\phi$
$\langle\alpha\rangle^P\phi$
$P(\phi)$
$\psi \to [\alpha]\phi$
$\psi \to [\alpha]\phi$
$\psi \to [\alpha]\phi$
$\psi \to [\alpha]\phi$
$\psi \to [\alpha]\phi$
KeYmaera Prover
Input File
Rule base
Strategy
Rule Engine
Proof
Solvers
Mathematica
QEPCAD
Orbital
16
8
4
2
1
$c$
RBC
far
ST
neg
SB
cor
MA

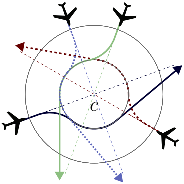

RBC
far
ST
neg
SB
cor
MA

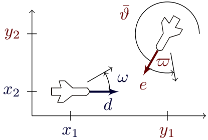

far
neg
cor
rec
fsa

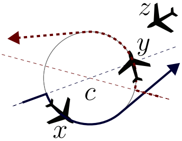


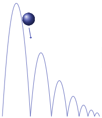

y
c
x

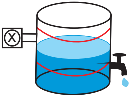

c
entry
exit
x
y


c

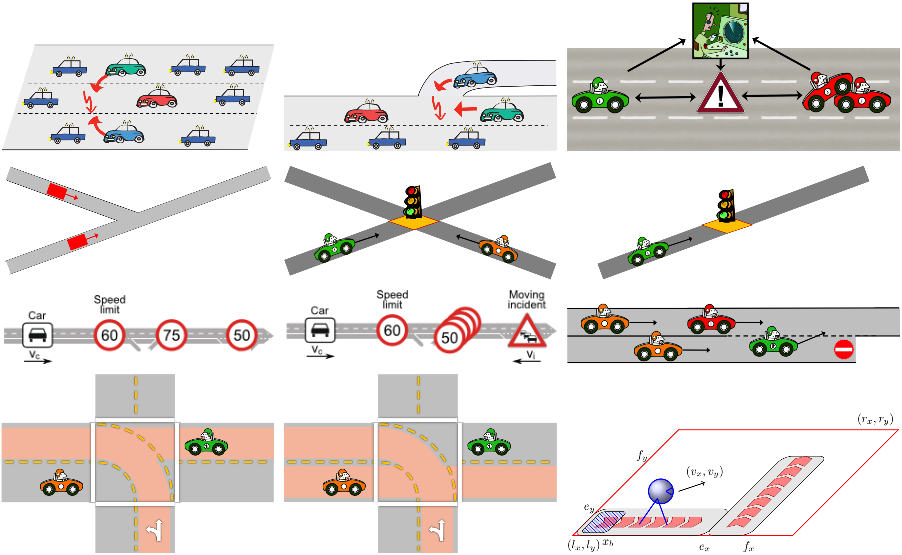

$y_2$
$x_2$
$x_1$
$y_1$
$\bar{\vartheta}$
$\omega$
$\varpi$
$e$
$d$

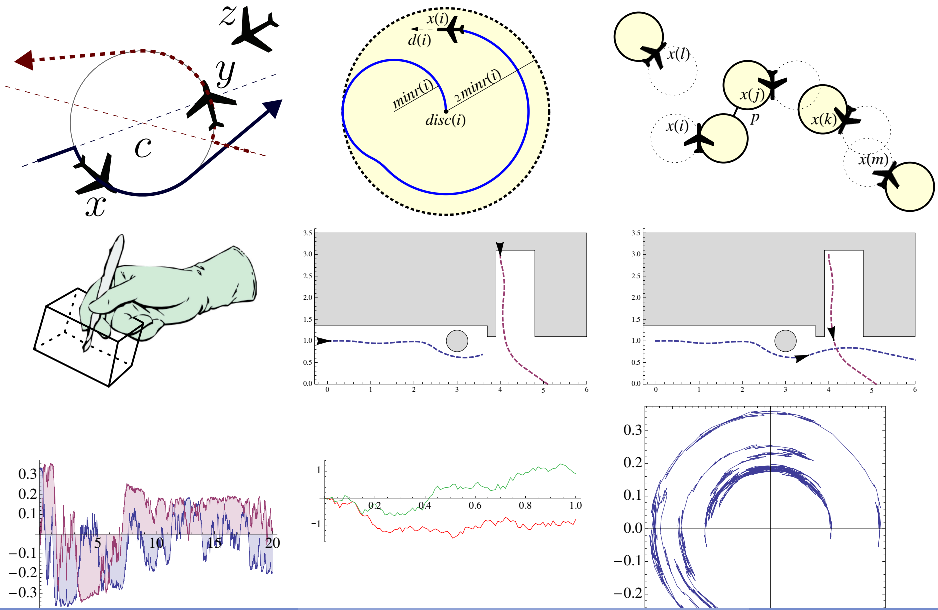

z
y
c
x





# Successful Hybrid Systems Proofs

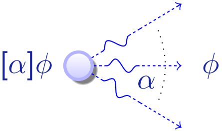


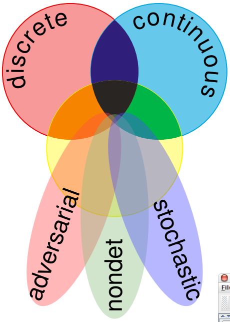

z
y
c
x
x(i)
d(i)
minr(i)
2minr(i)
disc(i)
x(l)
x(j)
p
x(k)
x(i)
x(m)

# Outline

# Hybrid Systems Verification and Robotics

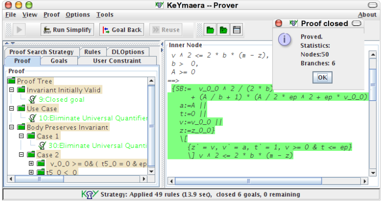
- Hybrid system models
- Discrete dynamics
- Continuous dynamics
- Correctness properties
- Safety, liveness ...

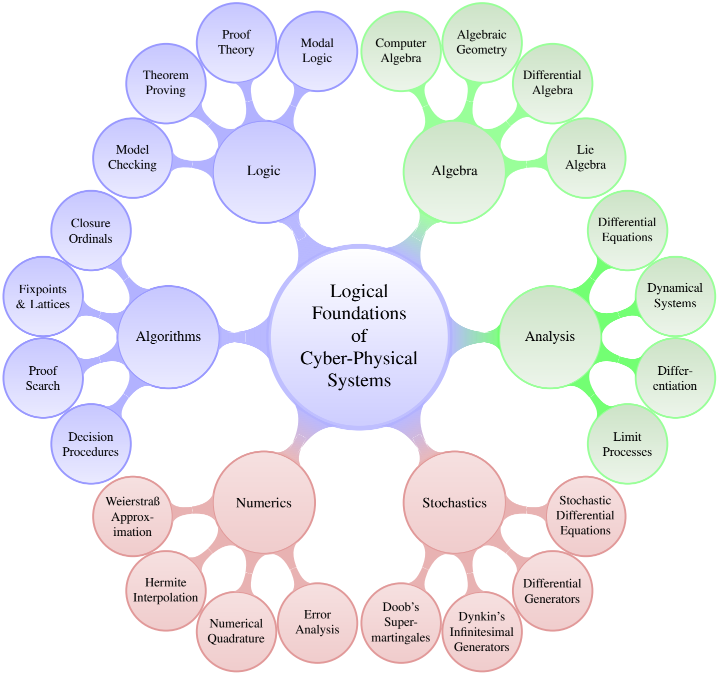


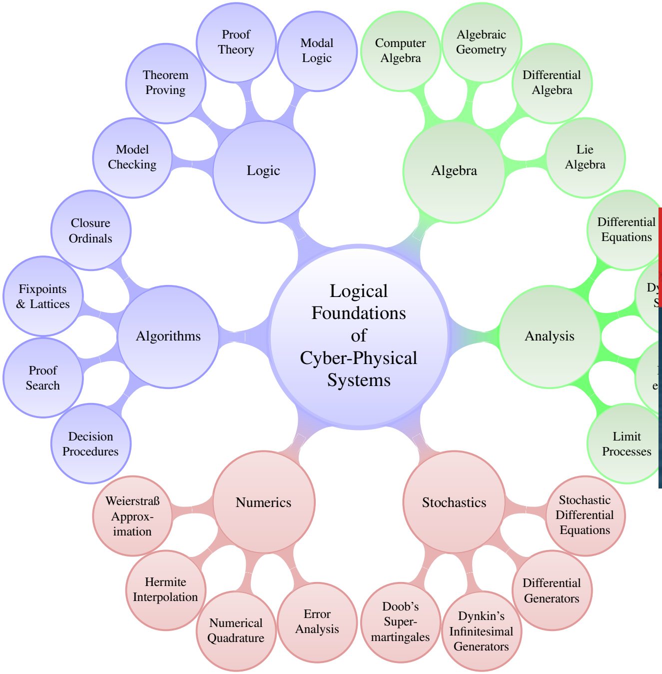




- Model checking
- Logic & proofs
- Cyber-physical systems
- Differential invariants

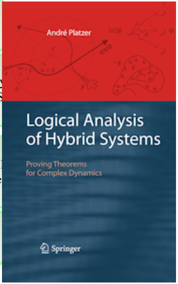
KeYmaera




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



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
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
Dynkin's Infinitesimal Generators
Doob's Super-martingales

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

Thomas A. Henzinger.
The theory of hybrid automata.
In *LICS*, pages 278–292, Los Alamitos, 1996. IEEE Computer Society.

Rajeev Alur.
Formal verification of hybrid systems.
In Samarjit Chakraborty, Ahmed Jerraya, Sanjoy K. Baruah, and Sebastian Fischmeister, editors, *EMSOFT*, pages 273–278. ACM, 2011.

André Platzer.
Logics of dynamical systems.
In *LICS*, pages 13–24. IEEE, 2012.

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

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

Springer, Heidelberg, 2010.

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.

Ian M. Mitchell and Jeremy A. Templeton.
A toolbox of Hamilton-Jacobi solvers for analysis of nondeterministic continuous and hybrid systems.
In Manfred Morari and Lothar Thiele, editors, *HSCC*, volume 3414 of *LNCS*, pages 480–494. Springer, 2005.

Stefan Ratschan and Zhikun She.
Safety verification of hybrid systems by constraint propagation-based abstraction refinement.
*Trans. on Embedded Computing Sys.*, 6(1):8, 2007.

Goran Frehse, Colas Le Guernic, Alexandre Donzé, Scott Cotton, Rajarshi Ray, Olivier Lebeltel, Rodolfo Ripado, Antoine Girard, Thao Dang, and Oded Maler.

SpaceEx: Scalable verification of hybrid systems.
In Ganesh Gopalakrishnan and Shaz Qadeer, editors, *CAV*, volume 6806 of *LNCS*, pages 379–395. Springer, 2011.

Goran Frehse.
PHAVer: algorithmic verification of hybrid systems past HyTech.
*STTT*, 10(3):263–279, 2008.

André Platzer and Edmund M. Clarke.
The image computation problem in hybrid systems model checking.
In Alberto Bemporad, Antonio Bicchi, and Giorgio Buttazzo, editors, *HSCC*, volume 4416 of *LNCS*, pages 473–486. Springer, 2007.

Pieter Collins.
Optimal semicomputable approximations to reachable and invariant sets.
*Theory Comput. Syst.*, 41(1):33–48, 2007.

Edmund M. Clarke, Ansgar Fehnker, Zhi Han, Bruce H. Krogh, Joël Ouaknine, Olaf Stursberg, and Michael Theobald.

Abstraction and counterexample-guided refinement in model checking of hybrid systems.
*Int. J. Found. Comput. Sci.*, 14(4):583–604, 2003.

Alongkrit Chutinan and Bruce H. Krogh.
Computational techniques for hybrid system verification.
*IEEE T. Automat. Contr.*, 48(1):64–75, 2003.

Carla Piazza, Marco Antoniotti, Venkatesh Mysore, Alberto Policriti, Franz Winkler, and Bud Mishra.
Algorithmic algebraic model checking I: Challenges from systems biology.
In Kousha Etessami and Sriram K. Rajamani, editors, *CAV*, volume 3576 of *LNCS*, pages 5–19. Springer, 2005.