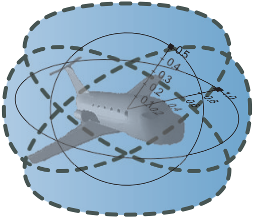
Logic of Distributed Hybrid Systems

André Platzer

Karlsruhe Institute of Technology

Carnegie Mellon University



Outline

1. Motivation
2. Quantified Differential Dynamic Logic $\mathsf{Qd}\mathcal{L}$
   - Design
   - Syntax
   - Semantics
3. Proof Calculus for Distributed Hybrid Systems
   - Compositional Verification Calculus
   - Deduction Modulo with Free Variables & Skolemization
   - Actual Existence and Creation
   - Soundness and Completeness
   - Quantified Differential Invariants
4. Applications
   - Distributed Car Control
   - Surgical Robot
5. Conclusions

Complex Physical Systems:

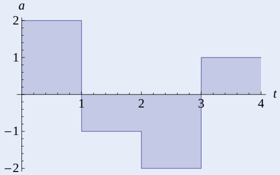
Q: I want to verify my car

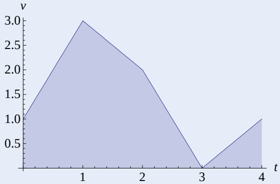
Challenge

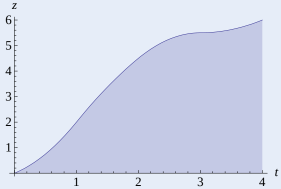
Q: I want to verify my car A: Hybrid systems

Challenge (Hybrid Systems)

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



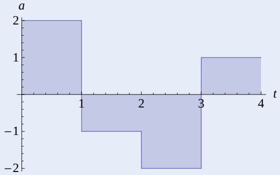


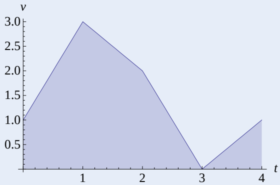


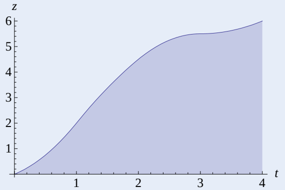
Q: I want to verify my car A: Hybrid systems Q: But there's a lot of cars!

Challenge (Hybrid Systems)

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



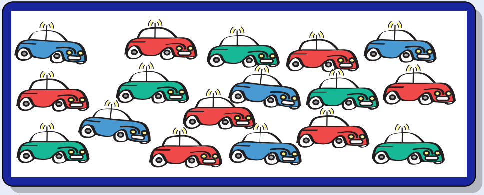




Complex Physical Systems:

Q: I want to verify a lot of cars

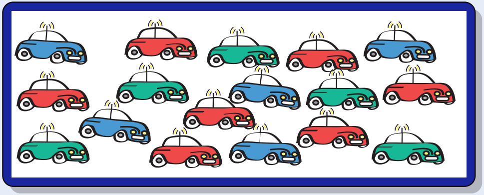
Challenge

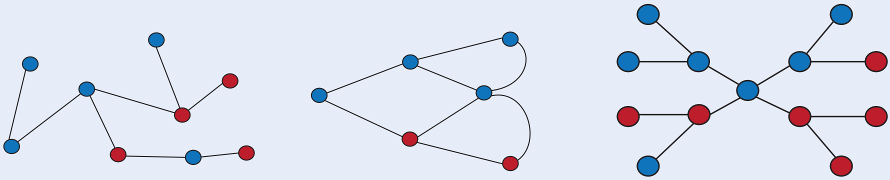


Q: I want to verify a lot of cars A: Distributed systems

Challenge (Distributed Systems)

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

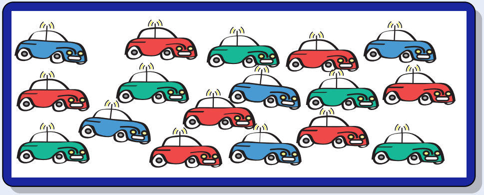


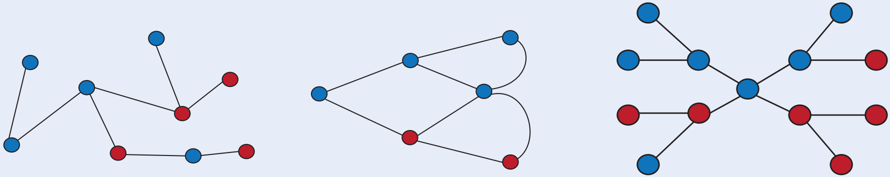


Q: I want to verify a lot of cars A: Distributed systems Q: But they move!

Challenge (Distributed Systems)

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

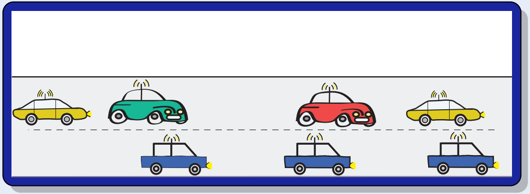




Complex Physical Systems:

Q: I want to verify lots of moving cars

Challenge

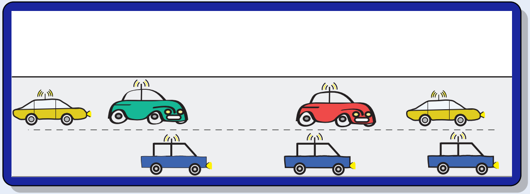


Complex Physical Systems: Distributed Hybrid Systems

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

Challenge (Distributed Hybrid Systems)

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

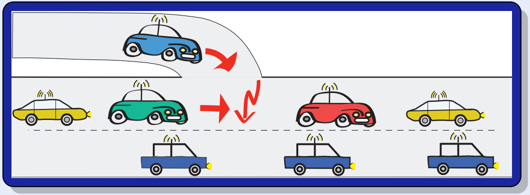


Complex Physical Systems: Distributed Hybrid Systems

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

Challenge (Distributed Hybrid Systems)

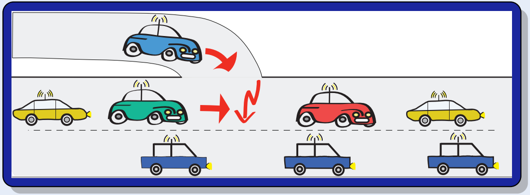
- Continuous dynamics (differential equations)
- Discrete dynamics (control decisions)
- Structural dynamics (remote communication)
- Dimensional dynamics (appearance)



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

Challenge (Distributed Hybrid Systems)

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



State of the Art: Modeling and Simulation

No formal verification of distributed hybrid systems

Shift [4] The Hybrid System Simulation Programming Language

R-Charon [5] Modeling Language for Reconfigurable Hybrid Systems

Hybrid CSP [6] Semantics in Extended Duration Calculus

Φ-calculus [9] Semantics in rich set theory

HyPA [7] Translate fragment into normal form.

ACP^{srt}_{hs} [10] Modeling language proposal

χ process algebra [8] Simulation, translation of fragments to PHAVER, UPPAAL

OBSHS [11] Partial random simulation of objects

Contributions

1. System model and semantics for distributed hybrid systems: QHP
2. Specification and verification logic: $\mathsf{Qd}\mathscr{L}$
3. Proof calculus for $\mathsf{Qd}\mathscr{L}$
4. First verification approach for distributed hybrid systems
5. Sound and complete axiomatization relative to differential equations
6. Prove collision freedom in a (simple) distributed car control system, where new cars may appear dynamically on the road
7. Logical foundation for analysis of distributed hybrid systems
8. Fundamental extension: first-order $x(i)$ versus primitive x

CSL'10,LMCS'12

Outline

1. Motivation
2. Quantified Differential Dynamic Logic Qd$\mathscr{L}$
   - Design
   - Syntax
   - Semantics
3. Proof Calculus for Distributed Hybrid Systems
   - Compositional Verification Calculus
   - Deduction Modulo with Free Variables & Skolemization
   - Actual Existence and Creation
   - Soundness and Completeness
   - Quantified Differential Invariants
4. Applications
   - Distributed Car Control
   - Surgical Robot
5. Conclusions

Outline (Conceptual Approach)

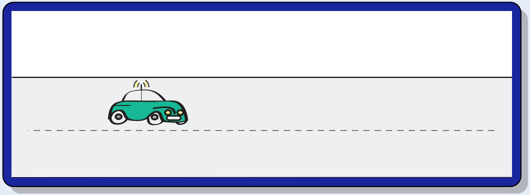
1. Motivation
2. Quantified Differential Dynamic Logic Qd$\mathscr{L}$
   - Design
   - Syntax
   - Semantics
3. Proof Calculus for Distributed Hybrid Systems
   - Compositional Verification Calculus
   - Deduction Modulo with Free Variables & Skolemization
   - Actual Existence and Creation
   - Soundness and Completeness
   - Quantified Differential Invariants
4. Applications
   - Distributed Car Control
   - Surgical Robot
5. Conclusions

Model for Distributed Hybrid Systems

Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

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

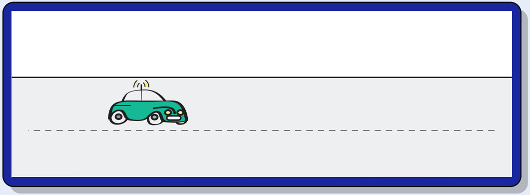


Model for Distributed Hybrid Systems

Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

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

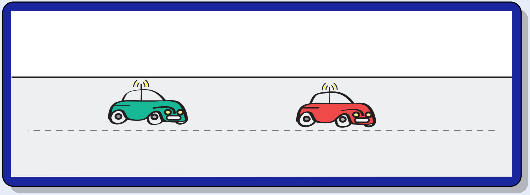


Model for Distributed Hybrid Systems

Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

- Continuous dynamics (differential equations)
  $x'' = a$
- Discrete dynamics (control decisions)
  $a := \texttt{if..then}\, A \,\texttt{else} -b$
- Structural dynamics (communication/coupling)

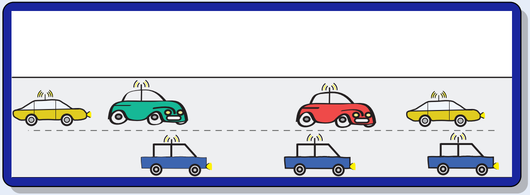


Model for Distributed Hybrid Systems

Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

- Continuous dynamics (differential equations)
  $x'' = a$
- Discrete dynamics (control decisions)
  $a := \mathtt{if}\,..\,\mathtt{then}\, A\, \mathtt{else} - b$
- Structural dynamics (communication/coupling)

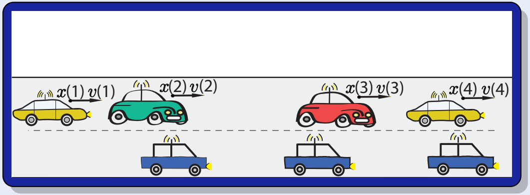


Model for Distributed Hybrid Systems

Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

- Continuous dynamics (differential equations)
  $x'' = a$
- Discrete dynamics (control decisions)
  $a := \mathtt{if}\,..\,\mathtt{then}\, A\, \mathtt{else} - b$
- Structural dynamics (communication/coupling)



Model for Distributed Hybrid Systems

Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

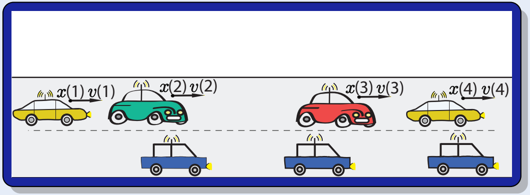
- Continuous dynamics (differential equations)

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

- Discrete dynamics (control decisions)

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

- Structural dynamics (communication/coupling)

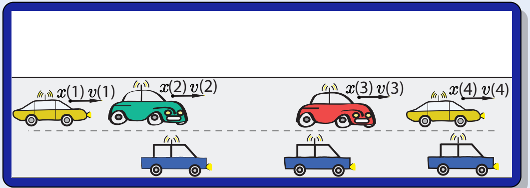


Model for Distributed Hybrid Systems

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)
  $$\forall i\, a(i) := \texttt{if..then}\, A\, \texttt{else} -b$$
- Structural dynamics (communication/coupling)

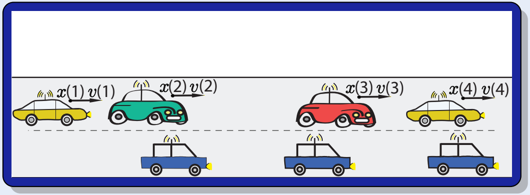


Model for Distributed Hybrid Systems

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)
  $$\forall i\, a(i) := \texttt{if}\,..\,\texttt{then}\, A\, \texttt{else} -b$$
- Structural dynamics (communication/coupling)
  $$\ell(i) := \mathit{carInFrontOf(i)}$$



Model for Distributed Hybrid Systems

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

Model (Distributed Hybrid Systems)

- Continuous dynamics (differential equations)

  $\forall i\, x(i)'' = a(i)$

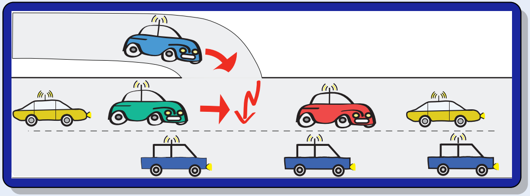
- Discrete dynamics (control decisions)

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

- Structural dynamics (communication/coupling)

  $\ell(i) := carInFrontOf(i)$

- Dimensional dynamics (appearance)

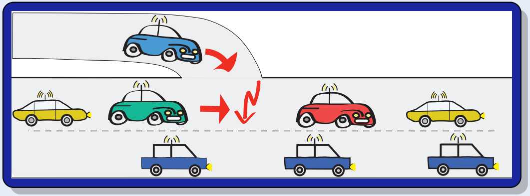


Model for Distributed Hybrid Systems

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)
  $\forall i\, a(i) := \mathtt{if}\,..\,\mathtt{then}\, A\, \mathtt{else} -b$
- Structural dynamics (communication/coupling)
  $\ell(i) := carInFrontOf(i)$
- Dimensional dynamics (appearance)
  $n := \mathtt{new}\, Car$



Quantified Differential Dynamic Logic $\mathrm{Qd}\mathcal{L}$: Syntax

Definition (Quantified hybrid program α)

$\forall i : C\ x(i)' = \theta$	(quantified ODE)	
$\forall i : C\ x(i) := \theta$	(quantified assignment)	jump & test
$?Q$	(conditional execution)	jump & test
$\alpha ; \beta$	(seq. composition)	Kleene algebra
$\alpha \cup \beta$	(nondet. choice)	Kleene algebra
α^*	(nondet. repetition)	Kleene algebra

Quantified Differential Dynamic Logic Qd$\mathscr{L}$: Syntax

Definition (Quantified hybrid program α)

$\forall i : C\ x(s)' = \theta$	(quantified ODE)	
$\forall i : C\ x(s) := \theta$	(quantified assignment)	jump & test
$?Q$	(conditional execution)	
$\alpha ; \beta$	(seq. composition)	
$\alpha \cup \beta$	(nondet. choice)	Kleene algebra
α^*	(nondet. repetition)	

Quantified Differential Dynamic Logic Qd$\mathcal{L}$: Syntax

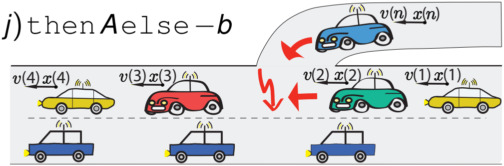
Definition (Quantified hybrid program α)

$\forall i : C\ x(s)' = \theta$	(quantified ODE)	
$\forall i : C\ x(s) := \theta$	(quantified assignment)	jump & test
$?Q$	(conditional execution)	
$\alpha ; \beta$	(seq. composition)	Kleene algebra
$\alpha \cup \beta$	(nondet. choice)	
α^*	(nondet. repetition)	

$$DCCS \equiv (ctrl; drive)^*$$

$$ctrl \equiv \forall i : C\ a(i) := \texttt{if}\ \forall j : C\ far(i,j)\ \texttt{then}\ A\ \texttt{else}\ -b$$

$$drive \equiv \forall i : C\ x(i)'' = a(i)$$



Quantified Differential Dynamic Logic $\mathsf{Qd}\mathcal{L}$: Syntax

Definition (Quantified hybrid program α)

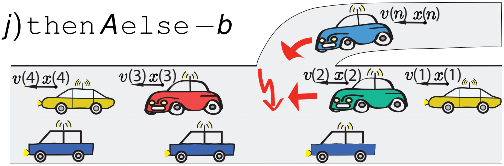
$\forall i : C\ x(s)' = \theta$	(quantified ODE)	
$\forall i : C\ x(s) := \theta$	(quantified assignment)	jump & test
$?Q$	(conditional execution)	
$\alpha ; \beta$	(seq. composition)	Kleene algebra
$\alpha \cup \beta$	(nondet. choice)	
α^*	(nondet. repetition)	

$$DCCS \equiv (appear; ctrl; drive)^*$$

$$appear \equiv n := \mathtt{new}\ C;\ \ ?(\forall j : C\ far(j,n))$$

$$ctrl \equiv \forall i : C\ a(i) := \mathtt{if}\ \forall j : C\ far(i,j)\ \mathtt{then}\ A\ \mathtt{else}\ -b$$

$$drive \equiv \forall i : C\ x(i)'' = a(i)$$



Definition (Quantified hybrid program α)

$\forall i : C\ x(s)' = \theta$	(quantified ODE)	
$\forall i : C\ x(s) := \theta$	(quantified assignment)	jump & test
$?Q$	(conditional execution)	
$\alpha; \beta$	(seq. composition)	
$\alpha \cup \beta$	(nondet. choice)	Kleene algebra
α^*	(nondet. repetition)	

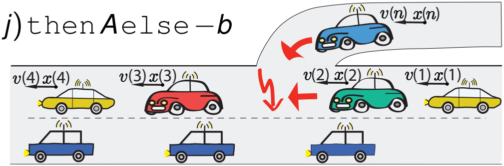
$$DCCS \equiv (appear; ctrl; drive)^*$$

$$appear \equiv n := \mathtt{new}\ C;\ \ ?(\forall j : C\ far(j, n))$$

$$ctrl \equiv \forall i : C\ a(i) := \mathtt{if}\ \forall j : C\ far(i, j)\ \mathtt{then}\ A\ \mathtt{else}\ -b$$

$$drive \equiv \forall i : C\ x(i)'' = a(i)$$

new C is definable!



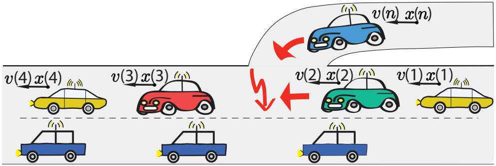
Quantified Differential Dynamic Logic Qd$\mathscr{L}$: Syntax

Definition (Qd$\mathscr{L}$ Formula ϕ)

$\neg, \wedge, \vee, \rightarrow, \forall x, \exists x, \; =, \geq, +, \cdot$ ($\mathbb{R}$-first-order part)

$[\alpha]\phi, \quad \langle\alpha\rangle\phi$ (dynamic part)

$$[(appear; ctrl; drive)^*]\ \forall i{\neq}j : C\ x(i) \neq x(j)$$

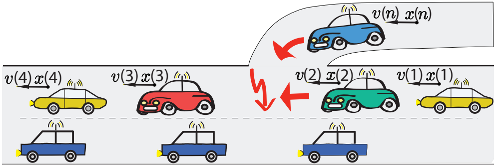


Quantified Differential Dynamic Logic Qd$\mathscr{L}$: Syntax

Definition (Qd$\mathscr{L}$ Formula ϕ)

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

$$\forall i, j : C \; far(i,j) \rightarrow [(appear; ctrl; drive)^*] \, \forall i{\neq}j : C \; x(i) \neq x(j)$$



Quantified Differential Dynamic Logic Qd$\mathscr{L}$: Syntax

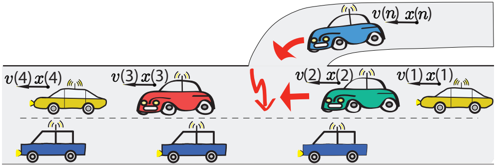
Definition (Qd$\mathscr{L}$ Formula ϕ)

$\neg, \wedge, \vee, \rightarrow, \forall x, \exists x, \; =, \geq, \; +, \cdot$ ($\mathbb{R}$-first-order part)

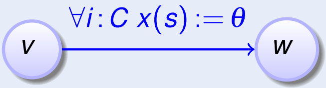
$[\alpha]\phi, \quad \langle\alpha\rangle\phi$ (dynamic part)

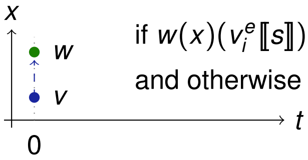
$$\forall i, j : C \; far(i,j) \rightarrow [(appear; ctrl; drive)^*] \, \forall i \neq j : C \; x(i) \neq x(j)$$

$$far(i,j) \equiv i \neq j \rightarrow x(i) < x(j) \wedge v(i) \leq v(j) \wedge a(i) \leq a(j)$$
$$\vee \, x(i) > x(j) \wedge v(i) \geq v(j) \wedge a(i) \geq a(j) \ldots$$



Definition (Quantified hybrid program α: transition semantics)



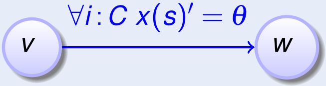


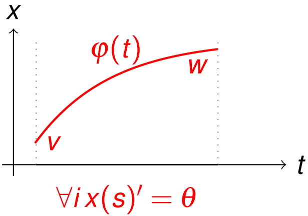
if $w(x)(v_i^e[\![s]\!]) = v_i^e[\![\theta]\!]$ (for all e)

and otherwise unchanged

Quantified Differential Dynamic Logic Qd$\mathcal{L}$: Semantics

Definition (Quantified hybrid program α: transition semantics)

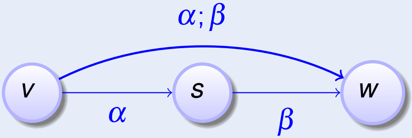




$$\frac{\mathrm{d}\,\varphi(t)_i^e[\![x(s)]\!]}{\mathrm{d}t}(\zeta) = \varphi(\zeta)_i^e[\![\theta]\!] \quad \text{(for all } e\text{)}$$

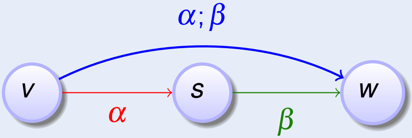
Quantified Differential Dynamic Logic Q$d\mathcal{L}$: Semantics

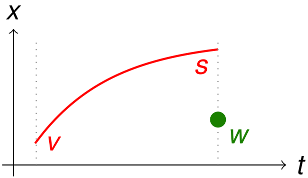
Definition (Quantified hybrid program α: transition semantics)



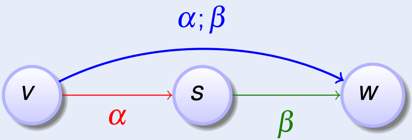
Quantified Differential Dynamic Logic Qd$\mathscr{L}$: Semantics

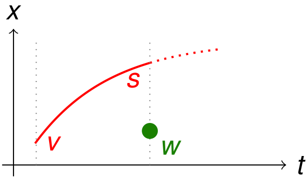
Definition (Quantified hybrid program α: transition semantics)





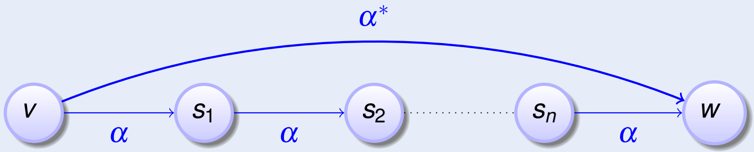
Definition (Quantified hybrid program α: transition semantics)





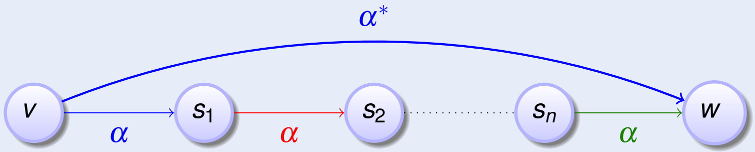
Quantified Differential Dynamic Logic Qd$\mathscr{L}$: Semantics

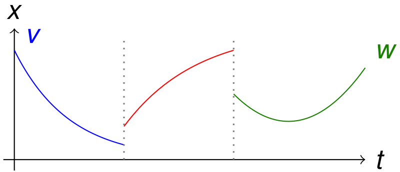
Definition (Quantified hybrid program α: transition semantics)



Quantified Differential Dynamic Logic Qd$\mathcal{L}$: Semantics

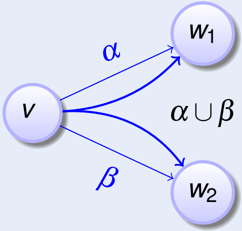
Definition (Quantified hybrid program α: transition semantics)



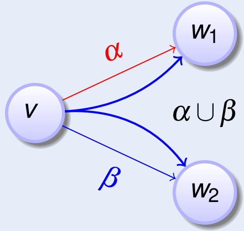


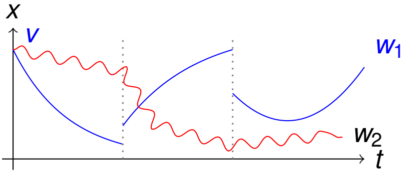
Quantified Differential Dynamic Logic Qd$\mathscr{L}$: Semantics

Definition (Quantified hybrid program α: transition semantics)



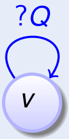
Definition (Quantified hybrid program α: transition semantics)



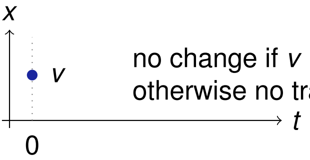


Quantified Differential Dynamic Logic Qd$\mathscr{L}$: Semantics

Definition (Quantified hybrid program α: transition semantics)



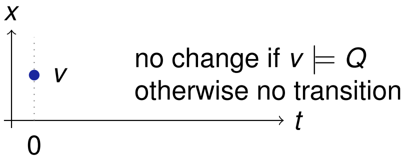
if $v \models Q$



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

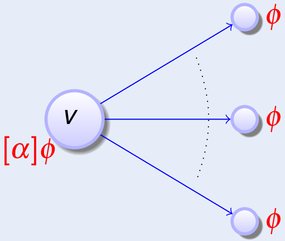
Definition (Quantified hybrid program α: transition semantics)

if $v \not\models Q$



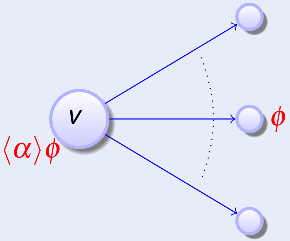
Quantified Differential Dynamic Logic Qd$\mathscr{L}$: Semantics

Definition (Qd$\mathscr{L}$ Formula ϕ)



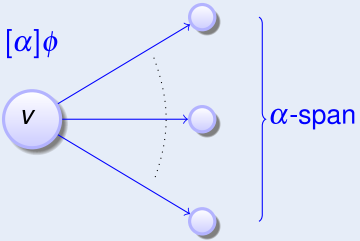
Quantified Differential Dynamic Logic Qd$\mathscr{L}$: Semantics

Definition (Qd$\mathscr{L}$ Formula ϕ)

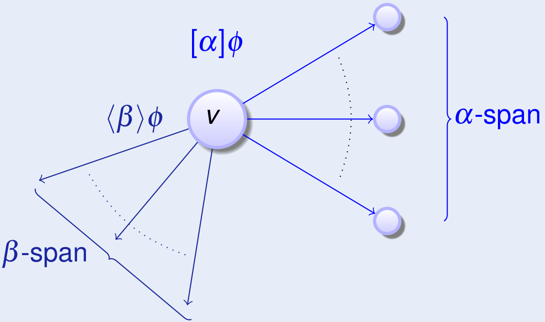


Quantified Differential Dynamic Logic Qd$\mathscr{L}$: Semantics

Definition (Qd$\mathscr{L}$ Formula ϕ)

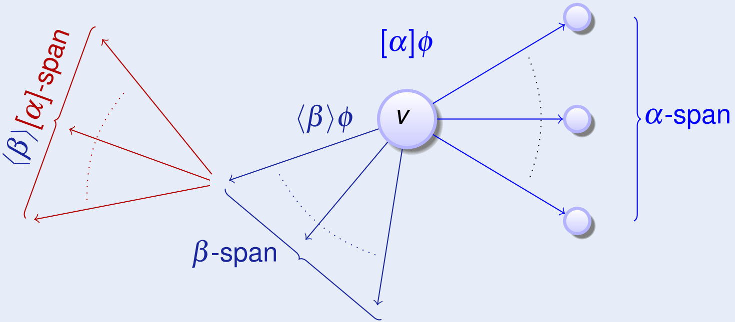


Definition ($Qd\mathcal{L}$ Formula ϕ)



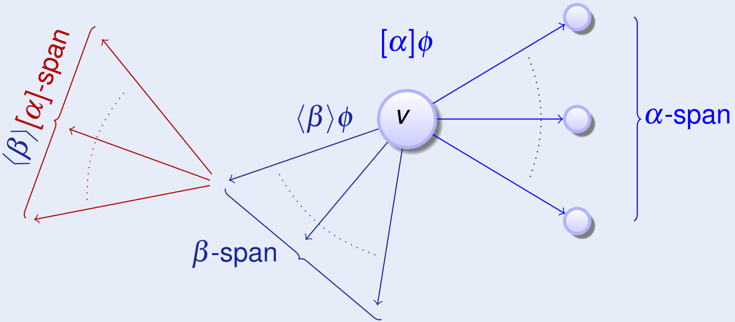
Quantified Differential Dynamic Logic Qd$\mathcal{L}$: Semantics

Definition (Qd$\mathcal{L}$ Formula ϕ)



Quantified Differential Dynamic Logic Qd$\mathscr{L}$: Semantics

Definition (Qd$\mathscr{L}$ Formula ϕ)

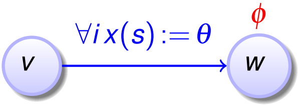


compositional semantics $\Rightarrow$ compositional calculus!

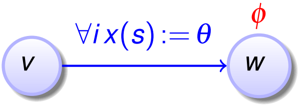
Outline (Verification Approach)

Proof Calculus for Quantified Differential Dynamic Logic

$$\overline{\phi([\forall i\, x(i) := \theta]x(u))}$$

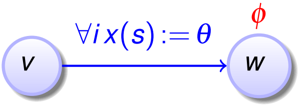


$$\frac{\forall i\,(i = u \rightarrow \phi(\theta))}{\phi([\forall i\, x(i) := \theta]x(u))}$$



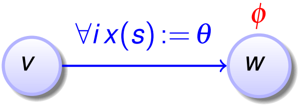
Proof Calculus for Quantified Differential Dynamic Logic

$$\frac{\forall i\,(i = [\forall i\, x(i) := \theta]u \rightarrow \phi(\theta))}{\phi([\forall i\, x(i) := \theta]x(u))}$$



Proof Calculus for Quantified Differential Dynamic Logic

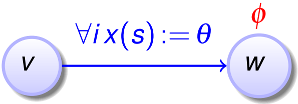
$$\frac{\forall i\,(i = [\forall i\, x(i) := \theta]u \rightarrow \phi(\theta))}{\phi([\forall i\, x(i) := \theta]x(u))}$$



$$\frac{}{\phi([\underbrace{\forall i\, x(s) := \theta}]x(u))}$$

$$\frac{\forall i\,(i = [\forall i\, x(i) := \theta]u \to \phi(\theta))}{\phi([\forall i\, x(i) := \theta]x(u))}$$

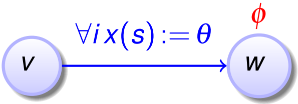
$$\frac{\texttt{if}\ \exists i\, s = u\ \texttt{then}\ \forall i\,(s = u \to \phi(\theta))\ \texttt{else}\ \phi(x(u))}{\phi([\underbrace{\forall i\, x(s) := \theta}]x(u))}$$



Proof Calculus for Quantified Differential Dynamic Logic

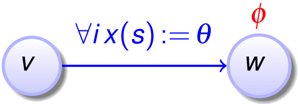
$$\frac{\forall i\,(i = [\forall i\, x(i) := \theta]u \to \phi(\theta))}{\phi([\forall i\, x(i) := \theta]x(u))}$$

$$\frac{\texttt{if}\ \exists i\, s = u\ \texttt{then}\ \forall i\,(s = u \to \phi(\theta))\ \texttt{else}\ \phi(x(u))}{\phi([\underbrace{\forall i\, x(s) := \theta}]x(u))}$$



$$\frac{\forall i\,(i = [\forall i\, x(i) := \theta]u \to \phi(\theta))}{\phi([\forall i\, x(i) := \theta]x(u))}$$

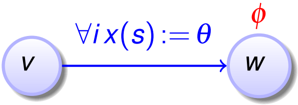
$$\frac{\texttt{if}\ \exists i\, s = u\ \texttt{then}\ \forall i\,(s = u \to \phi(\theta))\ \texttt{else}\ \phi(x(u))}{\phi(\underbrace{[\forall i\, x(s) := \theta]}x(u))}$$



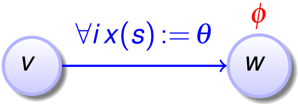
Proof Calculus for Quantified Differential Dynamic Logic

$$\frac{\forall i\,(i = [\forall i\, x(i) := \theta]u \to \phi(\theta))}{\phi([\forall i\, x(i) := \theta]x(u))}$$

$$\frac{\texttt{if}\ \exists i\, s = u\ \texttt{then}\ \forall i\,(s = u \to \phi(\theta))\ \texttt{else}\ \phi(x(u))}{\phi(\underbrace{[\forall i\, x(s) := \theta]}x(u))}$$

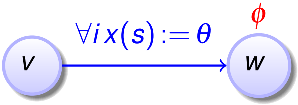


$$\frac{\forall i\,(i=[\forall i\,x(i):=\theta]u \rightarrow \phi(\theta))}{\phi([\forall i\,x(i):=\theta]x(u))}$$

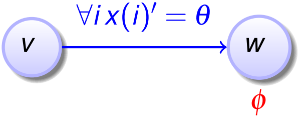


$$\frac{\mathtt{if}\ \exists i\, s=[\mathscr{A}]u\ \mathtt{then}\ \forall i\,(s=[\mathscr{A}]u \rightarrow \phi(\theta))\ \mathtt{else}\ \phi(x([\mathscr{A}]u))}{\phi(\underbrace{[\forall i\,x(s):=\theta]}_{\mathscr{A}}x(u))}$$

$$\frac{\forall i\,(i = [\forall i\, x(i) := \theta]u \to \phi(\theta))}{\phi([\forall i\, x(i) := \theta]x(u))}$$

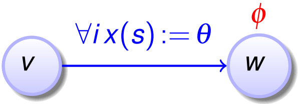


$$\frac{\texttt{if}\ \exists i\, s = [\mathscr{A}]u\ \texttt{then}\ \forall i\,(s = [\mathscr{A}]u \to \phi(\theta))\ \texttt{else}\ \phi(x([\mathscr{A}]u))}{\phi(\underbrace{[\forall i\, x(s) := \theta]}_{\mathscr{A}}x(u))}$$



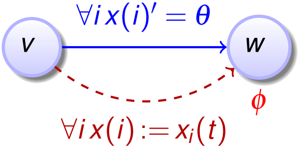
$$\frac{\forall t{\geq}0\,[\forall i\, x(i) := x_i(t)]\phi}{[\forall i\, x(i)' = \theta]\phi}$$

$$\frac{\forall i\,(i = [\forall i\, x(i) := \theta]u \to \phi(\theta))}{\phi([\forall i\, x(i) := \theta]x(u))}$$



$$\frac{\mathtt{if}\ \exists i\, s = [\mathscr{A}]u\ \mathtt{then}\ \forall i\,(s = [\mathscr{A}]u \to \phi(\theta))\ \mathtt{else}\ \phi(x([\mathscr{A}]u))}{\phi(\underbrace{[\forall i\, x(s) := \theta]}_{\mathscr{A}}x(u))}$$

$$\frac{\forall t{\geq}0\, [\forall i\, x(i) := x_i(t)]\phi}{[\forall i\, x(i)' = \theta]\phi}$$

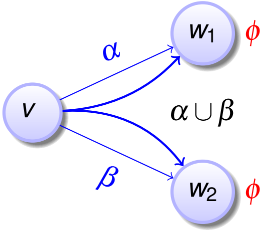


Proof Calculus for Quantified Differential Dynamic Logic

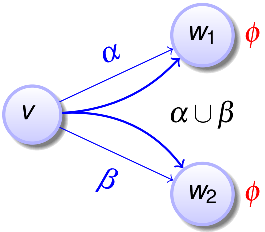
compositional semantics $\Rightarrow$ compositional rules!

Proof Calculus for Quantified Differential Dynamic Logic

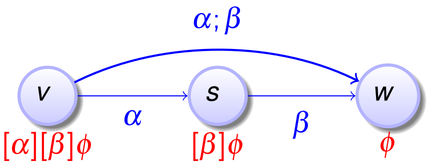
$$\frac{[\alpha]\phi \land [\beta]\phi}{[\alpha \cup \beta]\phi}$$



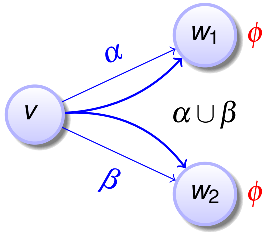
$$\frac{[\alpha]\phi \land [\beta]\phi}{[\alpha \cup \beta]\phi}$$



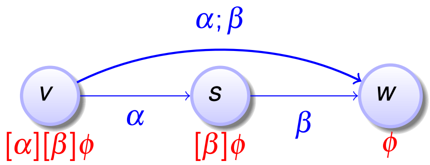
$$\frac{[\alpha][\beta]\phi}{[\alpha;\beta]\phi}$$



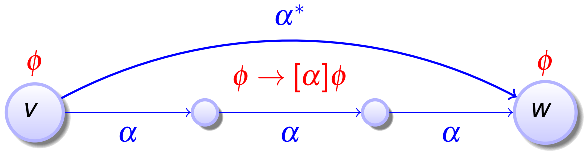
$$\frac{[\alpha]\phi \wedge [\beta]\phi}{[\alpha \cup \beta]\phi}$$



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

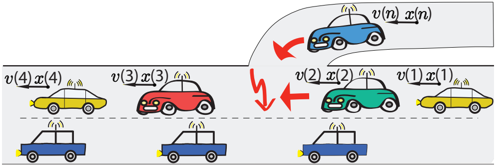


$$\frac{\phi \quad (\phi \rightarrow [\alpha]\phi)}{[\alpha^*]\phi}$$

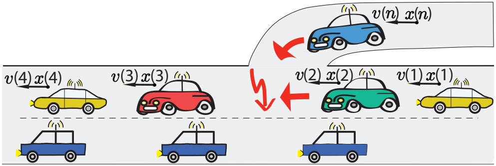


Deduction Modulo with Free Variables & Skolemization

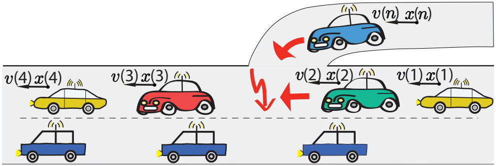
$$\forall i{\neq}j\, x(i){\neq}x(j) \rightarrow [\forall i\, x(i)'' = -b]\forall j{\neq}k\, x(j){\neq}x(k)$$



$$\frac{\forall i{\neq}j\, x(i){\neq}x(j) \rightarrow [\forall i\, x(i)' = v(i), v(i)' = -b]\forall j{\neq}k\, x(j){\neq}x(k)}{\forall i{\neq}j\, x(i){\neq}x(j) \rightarrow [\forall i\, x(i)'' = -b]\forall j{\neq}k\, x(j){\neq}x(k)}$$



$$\dfrac{\dfrac{\forall i{\neq}j\, x(i){\neq}x(j) \rightarrow \forall t{\geq}0\, [\forall i\, x(i) := -\frac{b}{2}t^2 + v(i)t + x(i)]\forall j{\neq}k\, x(j){\neq}x(k)}{\forall i{\neq}j\, x(i){\neq}x(j) \rightarrow [\forall i\, x(i)' = v(i), v(i)' = -b]\forall j{\neq}k\, x(j){\neq}x(k)}}{\forall i{\neq}j\, x(i){\neq}x(j) \rightarrow [\forall i\, x(i)'' = -b]\forall j{\neq}k\, x(j){\neq}x(k)}$$

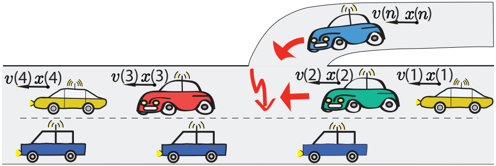


$$\forall i{\neq}j\, x(i){\neq}x(j) \to s{\geq}0 \to [\forall i\, x(i) := -\frac{b}{2}s^2 + v(i)s + x(i)]\,\forall j{\neq}k\, x(j){\neq}x(k)$$

$$\forall i{\neq}j\, x(i){\neq}x(j) \to \forall t{\geq}0\, [\forall i\, x(i) := -\frac{b}{2}t^2 + v(i)t + x(i)]\forall j{\neq}k\, x(j){\neq}x(k)$$

$$\forall i{\neq}j\, x(i){\neq}x(j) \to [\forall i\, x(i)' = v(i), v(i)' = -b]\,\forall j{\neq}k\, x(j){\neq}x(k)$$

$$\forall i{\neq}j\, x(i){\neq}x(j) \to [\forall i\, x(i)'' = -b]\,\forall j{\neq}k\, x(j){\neq}x(k)$$



Deduction Modulo with Free Variables & Skolemization

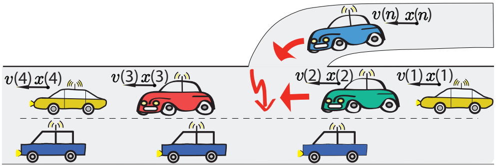
$$\forall i{\neq}j\, x(i){\neq}x(j), s{\geq}0 \to [\forall i\, x(i) := -\frac{b}{2}s^2 + v(i)s + x(i)]\forall j{\neq}k\, x(j){\neq}x(k)$$

$$\forall i{\neq}j\, x(i){\neq}x(j) \to s{\geq}0 \to [\forall i\, x(i) := -\frac{b}{2}s^2 + v(i)s + x(i)]\forall j{\neq}k\, x(j){\neq}x(k)$$

$$\forall i{\neq}j\, x(i){\neq}x(j) \to \forall t{\geq}0\, [\forall i\, x(i) := -\frac{b}{2}t^2 + v(i)t + x(i)]\forall j{\neq}k\, x(j){\neq}x(k)$$

$$\forall i{\neq}j\, x(i){\neq}x(j) \to [\forall i\, x(i)' = v(i), v(i)' = -b]\forall j{\neq}k\, x(j){\neq}x(k)$$

$$\forall i{\neq}j\, x(i){\neq}x(j) \to [\forall i\, x(i)'' = -b]\forall j{\neq}k\, x(j){\neq}x(k)$$



Deduction Modulo with Free Variables & Skolemization

$$\forall i{\neq}j\, x(i){\neq}x(j), s{\geq}0 \rightarrow \forall j{\neq}k\, \left(-\tfrac{b}{2}s^2 + v(j)s + x(j) \neq -\tfrac{b}{2}s^2 + v(k)s + x(k)\right)$$

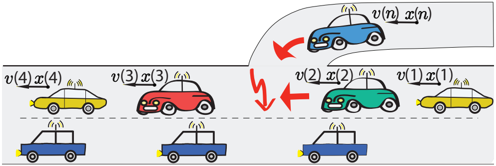
$$\forall i{\neq}j\, x(i){\neq}x(j), s{\geq}0 \rightarrow [\forall i\, x(i) := -\tfrac{b}{2}s^2 + v(i)s + x(i)]\, \forall j{\neq}k\, x(j){\neq}x(k)$$

$$\forall i{\neq}j\, x(i){\neq}x(j) \rightarrow s{\geq}0 \rightarrow [\forall i\, x(i) := -\tfrac{b}{2}s^2 + v(i)s + x(i)]\, \forall j{\neq}k\, x(j){\neq}x(k)$$

$$\forall i{\neq}j\, x(i){\neq}x(j) \rightarrow \forall t{\geq}0\, [\forall i\, x(i) := -\tfrac{b}{2}t^2 + v(i)t + x(i)]\forall j{\neq}k\, x(j){\neq}x(k)$$

$$\forall i{\neq}j\, x(i){\neq}x(j) \rightarrow [\forall i\, x(i)' = v(i), v(i)' = -b]\, \forall j{\neq}k\, x(j){\neq}x(k)$$

$$\forall i{\neq}j\, x(i){\neq}x(j) \rightarrow [\forall i\, x(i)'' = -b]\, \forall j{\neq}k\, x(j){\neq}x(k)$$



Deduction Modulo with Free Variables & Skolemization

$$\forall i{\neq}j\, x(i){\neq}x(j) \rightarrow \forall j{\neq}k \quad \forall s{\geq}0(-\tfrac{b}{2}s^2+v(j)s+x(j) \neq -\tfrac{b}{2}s^2+v(k)s+x(k))$$

$$\forall i{\neq}j\, x(i){\neq}x(j), s{\geq}0 \rightarrow \forall j{\neq}k\, (-\tfrac{b}{2}s^2+v(j)s+x(j) \neq -\tfrac{b}{2}s^2+v(k)s+x(k))$$

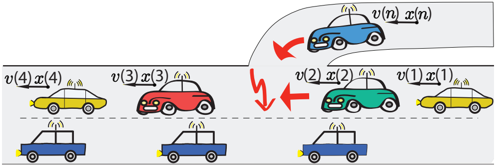
$$\forall i{\neq}j\, x(i){\neq}x(j), s{\geq}0 \rightarrow [\forall i\, x(i) := -\tfrac{b}{2}s^2+v(i)s+x(i)]\forall j{\neq}k\, x(j){\neq}x(k)$$

$$\forall i{\neq}j\, x(i){\neq}x(j) \rightarrow s{\geq}0 \rightarrow [\forall i\, x(i) := -\tfrac{b}{2}s^2+v(i)s+x(i)]\forall j{\neq}k\, x(j){\neq}x(k)$$

$$\forall i{\neq}j\, x(i){\neq}x(j) \rightarrow \forall t{\geq}0\, [\forall i\, x(i) := -\tfrac{b}{2}t^2+v(i)t+x(i)]\forall j{\neq}k\, x(j){\neq}x(k)$$

$$\forall i{\neq}j\, x(i){\neq}x(j) \rightarrow [\forall i\, x(i)' = v(i), v(i)' = -b]\forall j{\neq}k\, x(j){\neq}x(k)$$

$$\forall i{\neq}j\, x(i){\neq}x(j) \rightarrow [\forall i\, x(i)'' = -b]\forall j{\neq}k\, x(j){\neq}x(k)$$



$$\forall i{\neq}j\, x(i){\neq}x(j) \to \forall j{\neq}k\, \mathrm{QE}\forall s{\geq}0\left(-\tfrac{b}{2}s^2+v(j)s+x(j) \neq -\tfrac{b}{2}s^2+v(k)s+x(k)\right)$$

$$\forall i{\neq}j\, x(i){\neq}x(j), s{\geq}0 \to \forall j{\neq}k\left(-\tfrac{b}{2}s^2+v(j)s+x(j) \neq -\tfrac{b}{2}s^2+v(k)s+x(k)\right)$$

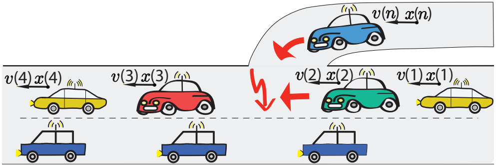
$$\forall i{\neq}j\, x(i){\neq}x(j), s{\geq}0 \to [\forall i\, x(i) := -\tfrac{b}{2}s^2+v(i)s+x(i)]\forall j{\neq}k\, x(j){\neq}x(k)$$

$$\forall i{\neq}j\, x(i){\neq}x(j) \to s{\geq}0 \to [\forall i\, x(i) := -\tfrac{b}{2}s^2+v(i)s+x(i)]\forall j{\neq}k\, x(j){\neq}x(k)$$

$$\forall i{\neq}j\, x(i){\neq}x(j) \to \forall t{\geq}0\, [\forall i\, x(i) := -\tfrac{b}{2}t^2+v(i)t+x(i)]\forall j{\neq}k\, x(j){\neq}x(k)$$

$$\forall i{\neq}j\, x(i){\neq}x(j) \to [\forall i\, x(i)' = v(i), v(i)' = -b]\forall j{\neq}k\, x(j){\neq}x(k)$$

$$\forall i{\neq}j\, x(i){\neq}x(j) \to [\forall i\, x(i)'' = -b]\forall j{\neq}k\, x(j){\neq}x(k)$$



Deduction Modulo with Free Variables & Skolemization

$$\forall i{\neq}j\, x(i){\neq}x(j) \to \forall j{\neq}k\, (x(j){\leq}x(k) \land v(j){\leq}v(k) \lor x(j){\geq}x(k) \land v(j){\geq}v(k))$$

$$\forall i{\neq}j\, x(i){\neq}x(j), s{\geq}0 \to \forall j{\neq}k\, (-\tfrac{b}{2}s^2 + v(j)s + x(j) \neq -\tfrac{b}{2}s^2 + v(k)s + x(k))$$

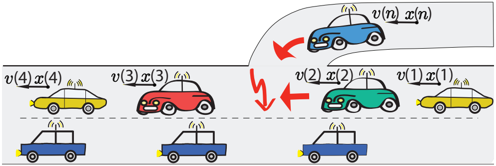
$$\forall i{\neq}j\, x(i){\neq}x(j), s{\geq}0 \to [\forall i\, x(i) := -\tfrac{b}{2}s^2 + v(i)s + x(i)]\, \forall j{\neq}k\, x(j){\neq}x(k)$$

$$\forall i{\neq}j\, x(i){\neq}x(j) \to s{\geq}0 \to [\forall i\, x(i) := -\tfrac{b}{2}s^2 + v(i)s + x(i)]\, \forall j{\neq}k\, x(j){\neq}x(k)$$

$$\forall i{\neq}j\, x(i){\neq}x(j) \to \forall t{\geq}0\, [\forall i\, x(i) := -\tfrac{b}{2}t^2 + v(i)t + x(i)]\, \forall j{\neq}k\, x(j){\neq}x(k)$$

$$\forall i{\neq}j\, x(i){\neq}x(j) \to [\forall i\, x(i)' = v(i), v(i)' = -b]\, \forall j{\neq}k\, x(j){\neq}x(k)$$

$$\forall i{\neq}j\, x(i){\neq}x(j) \to [\forall i\, x(i)'' = -b]\, \forall j{\neq}k\, x(j){\neq}x(k)$$



Deduction Modulo with Free Variables & Skolemization

$$\forall X, Y, V, W\, (X \neq Y \rightarrow X \leq Y \land V \leq W \lor X \geq Y \land V \geq W)$$

$$\forall i \neq j\, x(i) \neq x(j) \rightarrow \forall j \neq k\, (x(j) \leq x(k) \land v(j) \leq v(k) \lor x(j) \geq x(k) \land v(j) \geq v(k))$$

$$\forall i \neq j\, x(i) \neq x(j), s \geq 0 \rightarrow \forall j \neq k\, (-\tfrac{b}{2}s^2 + v(j)s + x(j) \neq -\tfrac{b}{2}s^2 + v(k)s + x(k))$$

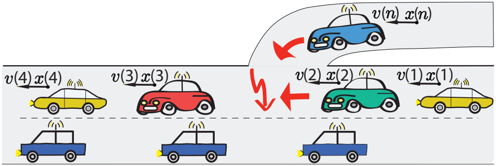
$$\forall i \neq j\, x(i) \neq x(j), s \geq 0 \rightarrow [\forall i\, x(i) := -\tfrac{b}{2}s^2 + v(i)s + x(i)]\, \forall j \neq k\, x(j) \neq x(k)$$

$$\forall i \neq j\, x(i) \neq x(j) \rightarrow s \geq 0 \rightarrow [\forall i\, x(i) := -\tfrac{b}{2}s^2 + v(i)s + x(i)]\, \forall j \neq k\, x(j) \neq x(k)$$

$$\forall i \neq j\, x(i) \neq x(j) \rightarrow \forall t \geq 0\, [\forall i\, x(i) := -\tfrac{b}{2}t^2 + v(i)t + x(i)]\, \forall j \neq k\, x(j) \neq x(k)$$

$$\forall i \neq j\, x(i) \neq x(j) \rightarrow [\forall i\, x(i)' = v(i), v(i)' = -b]\, \forall j \neq k\, x(j) \neq x(k)$$

$$\forall i \neq j\, x(i) \neq x(j) \rightarrow [\forall i\, x(i)'' = -b]\, \forall j \neq k\, x(j) \neq x(k)$$



Deduction Modulo with Free Variables & Skolemization

$$\forall X,Y,V,W\,(X\neq Y \to X\leq Y\land V\leq W \lor X\geq Y\land V\geq W)$$

$$\forall i{\neq}j\, x(i)\neq x(j) \to \forall j{\neq}k\,(x(j)\leq x(k)\land v(j)\leq v(k) \lor x(j)\geq x(k)\land v(j)\geq v(k))$$

$$\forall i{\neq}j\, x(i)\neq x(j), s\geq 0 \to \forall j{\neq}k\,(-\tfrac{b}{2}s^2+v(j)s+x(j) \neq -\tfrac{b}{2}s^2+v(k)s+x(k))$$

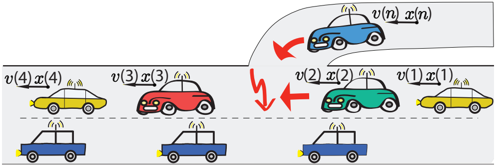
$$\forall i{\neq}j\, x(i)\neq x(j), s\geq 0 \to [\forall i\, x(i) := -\tfrac{b}{2}s^2+v(i)s+x(i)]\forall j{\neq}k\, x(j)\neq x(k)$$

$$\forall i{\neq}j\, x(i)\neq x(j) \to s\geq 0 \to [\forall i\, x(i) := -\tfrac{b}{2}s^2+v(i)s+x(i)]\forall j{\neq}k\, x(j)\neq x(k)$$

$$\forall i{\neq}j\, x(i)\neq x(j) \to \forall t\geq 0\,[\forall i\, x(i) := -\tfrac{b}{2}t^2+v(i)t+x(i)]\forall j{\neq}k\, x(j)\neq x(k)$$

$$\forall i{\neq}j\, x(i)\neq x(j) \to [\forall i\, x(i)' = v(i), v(i)' = -b]\forall j{\neq}k\, x(j)\neq x(k)$$

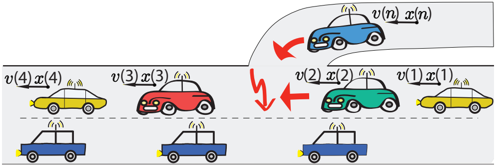
$$\forall i{\neq}j\, x(i)\neq x(j) \to [\forall i\, x(i)'' = -b]\forall j{\neq}k\, x(j)\neq x(k)$$



Actual Existence and Creation

Actual Existence Function $\mathrm{E}(\cdot)$

$$\mathrm{E}(i) = \begin{cases} 0 & \text{if } i \text{ denotes a possible object} \\ 1 & \text{if } i \text{ denotes an actively existing objects} \end{cases}$$

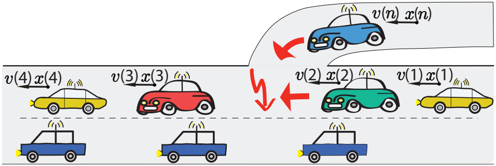


Actual Existence and Creation

Actual Existence Function $\mathrm{E}(\cdot)$

$$\mathrm{E}(i) = \begin{cases} 0 & \text{if } i \text{ denotes a possible object} \\ 1 & \text{if } i \text{ denotes an actively existing objects} \end{cases}$$

$$\frac{}{[n := \mathtt{new}\, C]\phi}$$

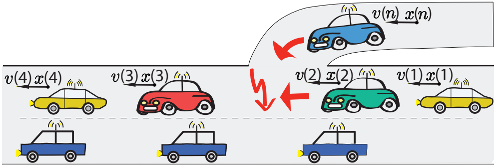


Actual Existence and Creation

Actual Existence Function $\mathrm{E}(\cdot)$

$$\mathrm{E}(i) = \begin{cases} 0 & \text{if } i \text{ denotes a possible object} \\ 1 & \text{if } i \text{ denotes an actively existing objects} \end{cases}$$

$$\frac{[(\forall j : C\ n := j);\qquad\qquad]\phi}{[n := \mathtt{new}\ C]\phi}$$

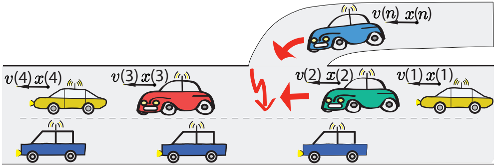


Actual Existence and Creation

Actual Existence Function $\mathrm{E}(\cdot)$

$$\mathrm{E}(i) = \begin{cases} 0 & \text{if } i \text{ denotes a possible object} \\ 1 & \text{if } i \text{ denotes an actively existing objects} \end{cases}$$

$$\frac{[(\forall j : C\ n := j);\ ?(\mathrm{E}(n) = 0);\qquad]\phi}{[n := \mathtt{new}\ C]\phi}$$

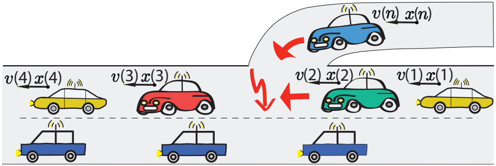


Actual Existence and Creation

Actual Existence Function $\mathrm{E}(\cdot)$

$$\mathrm{E}(i) = \begin{cases} 0 & \text{if } i \text{ denotes a possible object} \\ 1 & \text{if } i \text{ denotes an actively existing objects} \end{cases}$$

$$\frac{[(\forall j : C\ n := j);\ ?(\mathrm{E}(n) = 0);\ \mathrm{E}(n) := 1]\phi}{[n := \mathtt{new}\ C]\phi}$$



Actual Existence Function $\mathrm{E}(\cdot)$

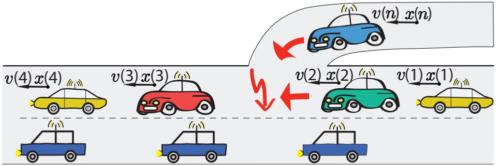
$$\mathrm{E}(i) = \begin{cases} 0 & \text{if } i \text{ denotes a possible object} \\ 1 & \text{if } i \text{ denotes an actively existing objects} \end{cases}$$

$$\frac{[(\forall j : C\ n := j);\ ?(\mathrm{E}(n) = 0);\ \mathrm{E}(n) := 1]\phi}{[n := \mathtt{new}\ C]\phi}$$

$$\forall i : C!\ \phi\ \equiv$$

$$\forall i : C!\ f(s) := \theta\ \equiv$$

$$\forall i : C!\ f(s)' = \theta\ \equiv$$



Actual Existence and Creation

Actual Existence Function $\mathrm{E}(\cdot)$

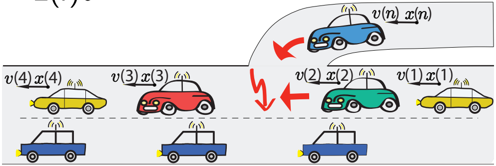
$$\mathrm{E}(i) = \begin{cases} 0 & \text{if } i \text{ denotes a possible object} \\ 1 & \text{if } i \text{ denotes an actively existing objects} \end{cases}$$

$$\frac{[(\forall j : C\ n := j);\ ?(\mathrm{E}(n) = 0);\ \mathrm{E}(n) := 1]\phi}{[n := \mathtt{new}\ C]\phi}$$

$$\forall i : C!\ \phi \ \equiv\ \forall i : C\ (\mathrm{E}(i) = 1 \rightarrow \phi)$$

$$\forall i : C!\ f(s) := \theta \ \equiv\ \forall i : C\ f(s) := (\mathtt{if}\ \mathrm{E}(i) = 1\ \mathtt{then}\ \theta\ \mathtt{else}\ f(s))$$

$$\forall i : C!\ f(s)' = \theta \ \equiv\ \forall i : C\ f(s)' = \mathrm{E}(i)\theta$$



Soundness and Completeness

Theorem (Relative Completeness) (LMCS'12)

QdL *calculus is a sound & complete axiomatisation of distributed hybrid systems relative to quantified differential equations.*

▸ Proof 16p.

Soundness and Completeness

Theorem (Relative Completeness) (LMCS'12)

Qd$\mathscr{L}$ *calculus is a sound & complete axiomatisation of distributed hybrid systems relative to quantified differential equations.*

Proof 16p.

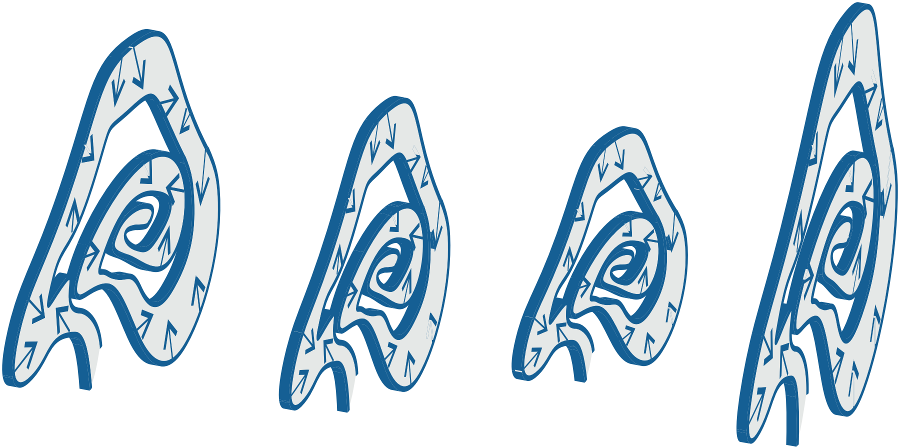
Corollary (Proof-theoretical Alignment)

proving distributed hybrid systems = proving dynamical systems!

Quantified Differential Invariants

Theorem (Quantified Differential Invariant) (HSCC'11)

$$\text{(QdI)} \quad \frac{Q \rightarrow [\forall i : C\ f(i)' := \theta] F'}{F \rightarrow [\forall i : C\ f(i)' = \theta \,\&\, Q] F} \quad \textit{is sound}$$



A Simple Proof with Quantified Differential Invariants

$$\frac{}{\forall i\!:\!C\; 2x(i)^3 \geq 1 \rightarrow [\forall i\!:\!C\; x(i)' = x(i)^2 + x(i)^4 + 2]\forall i\!:\!C\; 2x(i)^3 \geq 1}$$

A Simple Proof with Quantified Differential Invariants

$$\dfrac{[\forall i : C\ x(i)' := x(i)^2 + x(i)^4 + 2](\forall i : C\ 2x(i)^3 \geq 0)'}{\forall i : C\ 2x(i)^3 \geq 1 \rightarrow [\forall i : C\ x(i)' = x(i)^2 + x(i)^4 + 2]\forall i : C\ 2x(i)^3 \geq 1}$$

A Simple Proof with Quantified Differential Invariants

$$\frac{\dfrac{}{[\forall i : C\ x(i)' := x(i)^2 + x(i)^4 + 2]\forall i : C\ (2x(i)^3)' \geq 0}}{\dfrac{[\forall i : C\ x(i)' := x(i)^2 + x(i)^4 + 2](\forall i : C\ 2x(i)^3 \geq 0)'}{\forall i : C\ 2x(i)^3 \geq 1 \rightarrow [\forall i : C\ x(i)' = x(i)^2 + x(i)^4 + 2]\forall i : C\ 2x(i)^3 \geq 1}}$$

A Simple Proof with Quantified Differential Invariants

$$\dfrac{\dfrac{\dfrac{[\forall i{:}C\ x(i)' := x(i)^2 + x(i)^4 + 2]\forall i{:}C\ 6x(i)^2 x(i)' \geq 0}{[\forall i{:}C\ x(i)' := x(i)^2 + x(i)^4 + 2]\forall i{:}C\ (2x(i)^3)' \geq 0}}{[\forall i{:}C\ x(i)' := x(i)^2 + x(i)^4 + 2](\forall i{:}C\ 2x(i)^3 \geq 0)'}}{\forall i{:}C\ 2x(i)^3 \geq 1 \rightarrow [\forall i{:}C\ x(i)' = x(i)^2 + x(i)^4 + 2]\forall i{:}C\ 2x(i)^3 \geq 1}$$

A Simple Proof with Quantified Differential Invariants

$$\forall i{:}C\ 6x(i)^2(x(i)^2 + x(i)^4 + 2) \geq 0$$

$$[\forall i{:}C\ x(i)' := x(i)^2 + x(i)^4 + 2]\forall i{:}C\ 6x(i)^2 x(i)' \geq 0$$

$$[\forall i{:}C\ x(i)' := x(i)^2 + x(i)^4 + 2]\forall i{:}C\ (2x(i)^3)' \geq 0$$

$$[\forall i{:}C\ x(i)' := x(i)^2 + x(i)^4 + 2](\forall i{:}C\ 2x(i)^3 \geq 0)'$$

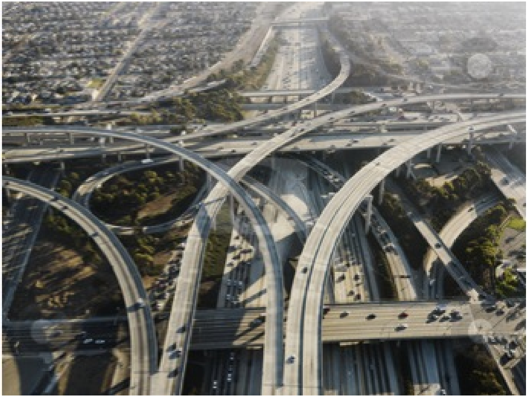
$$\forall i{:}C\ 2x(i)^3 \geq 1 \rightarrow [\forall i{:}C\ x(i)' = x(i)^2 + x(i)^4 + 2]\forall i{:}C\ 2x(i)^3 \geq 1$$

A Simple Proof with Quantified Differential Invariants

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\textit{true}}{\forall i : C\ 6x(i)^2(x(i)^2 + x(i)^4 + 2) \geq 0}}{[\forall i : C\ x(i)' := x(i)^2 + x(i)^4 + 2]\forall i : C\ 6x(i)^2 x(i)' \geq 0}}{[\forall i : C\ x(i)' := x(i)^2 + x(i)^4 + 2]\forall i : C\ (2x(i)^3)' \geq 0}}{[\forall i : C\ x(i)' := x(i)^2 + x(i)^4 + 2](\forall i : C\ 2x(i)^3 \geq 0)'}}{\forall i : C\ 2x(i)^3 \geq 1 \rightarrow [\forall i : C\ x(i)' = x(i)^2 + x(i)^4 + 2]\forall i : C\ 2x(i)^3 \geq 1}$$

Outline

Driver's License Test for Robotic Cars?



Driver's License Test for Robotic Cars?

Driver's License Test for Robotic Cars? Proof!

Car Control: Local Lane Control Challenge

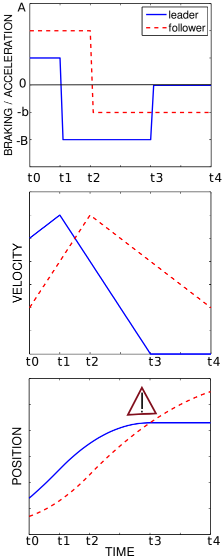
Challenge: Local lane dynamics

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

Car Control: Local Lane Control Challenge

Challenge: Local lane dynamics

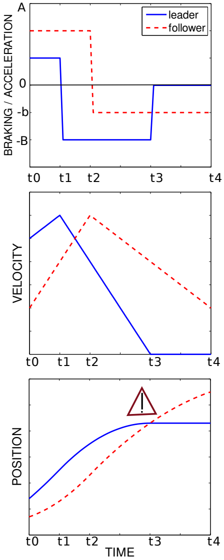
- A car controller for a differential equation respects separation of local lane.
- Follower car maintains safe distance to leader:



Challenge: Local lane dynamics

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

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



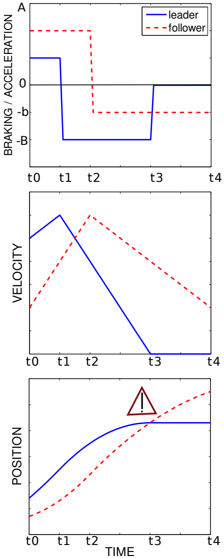
Car Control: Local Lane Control Challenge

Challenge: Local lane dynamics

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

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

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



Car Control: Global Lane Control Challenge

Challenge: Global lane dynamics

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

Car Control: Global Lane Control Challenge

Challenge: Global lane dynamics

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

Car Control: Global Lane Control Challenge

Challenge: Global lane dynamics

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

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

Car Control: Local Highway Control Challenge

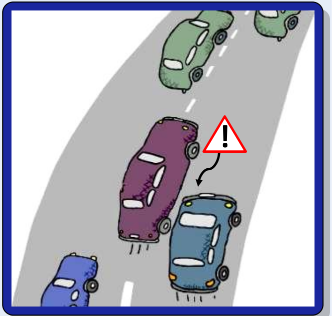
Challenge: Local highway dynamics

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

Car Control: Local Highway Control Challenge

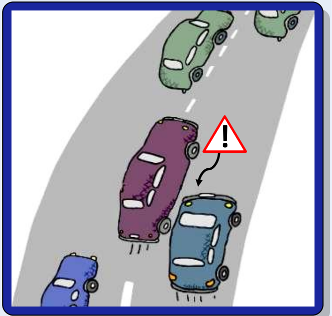
Challenge: Local highway dynamics

- 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

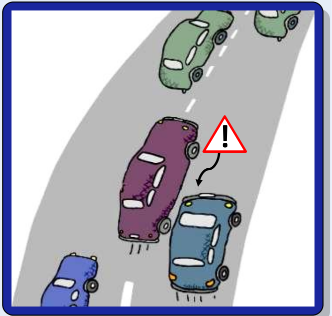
- All controllers for arbitrarily many differential equations respect separation locally on highway.
- For each lane: all controllers for the differential equations respect separation even if cars appear or disappear.
- Each car safe behind all others, even if new cars appear or disappear.



Car Control: Local Highway Control Challenge

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.



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

Car Control: Global Highway Control Challenge

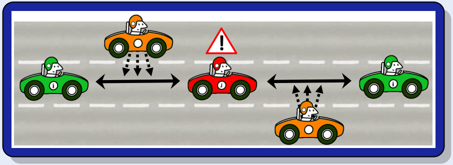
Challenge: Global highway dynamics

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

Car Control: Global Highway Control Challenge

Challenge: Global highway dynamics

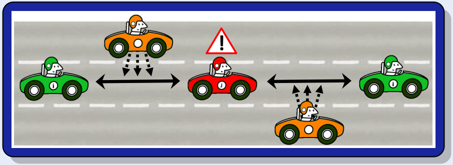
- All controllers for arbitrarily many differential equations respect separation globally on highway.
- All controllers for the differential equations respect separation even if cars switch lanes.



Car Control: Global Highway Control Challenge

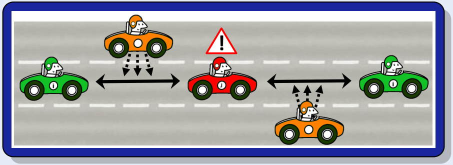
Challenge: Global highway dynamics

- 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.



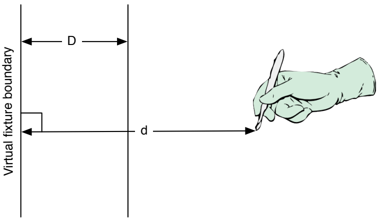
Challenge: Global highway dynamics

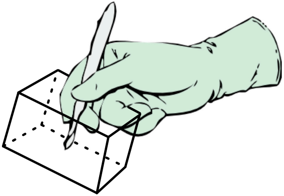
- All controllers for arbitrarily many differential equations respect separation globally on highway.
- All controllers for the differential equations respect separation even if cars switch lanes.
- On all lanes, all car safe behind all others on their lanes, even if cars switch lanes.

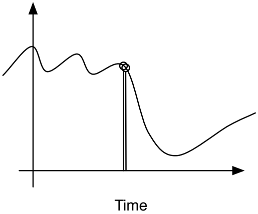


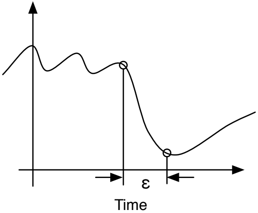
$$[\forall I\,(n := \mathtt{new}\,C;\ \forall i\, a(i) := ctrl;\ \forall i\, x(i)'' = a(i))^*]\,\forall I \forall i,j\; i \ll j$$

Surgical Robot Verification: Skull-base Surgery

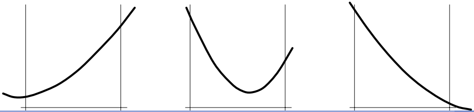








Redesign to predictive control



- Negligible lag?

HSCC'13

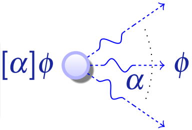
Outline

1. Motivation
2. Quantified Differential Dynamic Logic Qd$\mathscr{L}$
   - Design
   - Syntax
   - Semantics
3. Proof Calculus for Distributed Hybrid Systems
   - Compositional Verification Calculus
   - Deduction Modulo with Free Variables & Skolemization
   - Actual Existence and Creation
   - Soundness and Completeness
   - Quantified Differential Invariants
4. Applications
   - Distributed Car Control
   - Surgical Robot
5. Conclusions

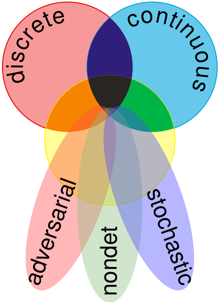
Conclusions

quantified differential dynamic logic

$$\text{Qd}\mathscr{L} = \text{FOL} + \text{DL} + \text{QHP}$$

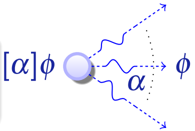


- Distributed hybrid systems everywhere
- System model and semantics
- Logic for distributed hybrid systems
- Compositional proof calculus
- First verification approach
- Sound & complete / diff. eqn.
- Quantified differential invariants
- Distributed car control verified
- Distributed aircraft control verified
- Robot verified for many obstacles

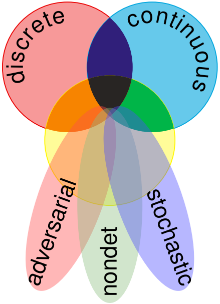


quantified differential dynamic logic

$$\mathrm{Qd}\mathscr{L} = \mathrm{FOL} + \mathrm{DL} + \mathrm{QHP}$$

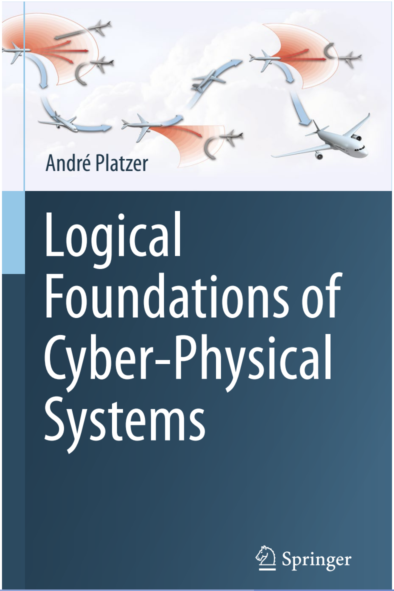


- Distributed hybrid systems everywhere
- System model and semantics
- Logic for distributed hybrid systems
- Compositional proof calculus
- First verification approach
- Sound & complete / diff. eqn.
- Quantified differential invariants
- Distributed car control verified
- Distributed aircraft control verified
- Robot verified for many obstacles



A. Platzer. *Logical Foundations of Cyber-Physical Systems*. Springer 2018

- **I Part: Elementary Cyber-Physical Systems**
- 2. Differential Equations & Domains
- 3. Choice & Control
- 4. Safety & Contracts
- 5. Dynamical Systems & Dynamic Axioms
- 6. Truth & Proof
- 7. Control Loops & Invariants
- 8. Events & Responses
- 9. Reactions & Delays
- **II Part: Differential Equations Analysis**
- 10. Differential Equations & Differential Invariants
- 11. Differential Equations & Proofs
- 12. Ghosts & Differential Ghosts
- 13. Differential Invariants & Proof Theory
- **III Part: Adversarial Cyber-Physical Systems**
- 14-17. Hybrid Systems & Hybrid Games
- **IV Part: Comprehensive CPS Correctness**



André Platzer.
A complete axiomatization of quantified differential dynamic logic for distributed hybrid systems.
Log. Meth. Comput. Sci., 8(4:17):1–44, 2012.
Special issue for selected papers from CSL'10.

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.

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

Akash Deshpande, Aleks Göllü, and Pravin Varaiya.
SHIFT: A formalism and a programming language for dynamic networks of hybrid automata.

In Panos J. Antsaklis, Wolf Kohn, Anil Nerode, and Shankar Sastry, editors, *Hybrid Systems*, volume 1273 of *LNCS*, pages 113–133. Springer, 1996.

Fabian Kratz, Oleg Sokolsky, George J. Pappas, and Insup Lee.
R-Charon, a modeling language for reconfigurable hybrid systems.
In Hespanha and Tiwari [18], pages 392–406.

Zhou Chaochen, Wang Ji, and Anders P. Ravn.
A formal description of hybrid systems.
In Rajeev Alur, Thomas A. Henzinger, and Eduardo D. Sontag, editors, *Hybrid Systems*, volume 1066 of *LNCS*, pages 511–530, Berlin, 1995. Springer.

Pieter J. L. Cuijpers and Michel A. Reniers.
Hybrid process algebra.
J. Log. Algebr. Program., 62(2):191–245, 2005.

D. A. van Beek, Ka L. Man, Michel A. Reniers, J. E. Rooda, and Ramon R. H. Schiffelers.
Syntax and consistent equation semantics of hybrid Chi.

J. Log. Algebr. Program., 68(1-2):129–210, 2006.

William C. Rounds.
A spatial logic for the hybrid π-calculus.
In Rajeev Alur and George J. Pappas, editors, *HSCC*, volume 2993 of *LNCS*, pages 508–522. Springer, 2004.

Jan A. Bergstra and C. A. Middelburg.
Process algebra for hybrid systems.
Theor. Comput. Sci., 335(2-3):215–280, 2005.

José Meseguer and Raman Sharykin.
Specification and analysis of distributed object-based stochastic hybrid systems.
In Hespanha and Tiwari [18], pages 460–475.

Yanni Kouskoulas, David W. Renshaw, André Platzer, and Peter Kazanzides.
Certifying the safe design of a virtual fixture control algorithm for a surgical robot.
In Belta and Ivancic [17], pages 263–272.

Sarah M. Loos, André Platzer, and Ligia Nistor.
Adaptive cruise control: Hybrid, distributed, and now formally verified.
In Michael Butler and Wolfram Schulte, editors, *FM*, volume 6664 of *LNCS*, pages 42–56, Berlin, 2011. Springer.

Sarah M. Loos, David W. Renshaw, and André Platzer.
Formal verification of distributed aircraft controllers.
In Belta and Ivancic [17], pages 125–130.

Stefan Mitsch, Khalil Ghorbal, David Vogelbacher, and André Platzer.
Formal verification of obstacle avoidance and navigation of ground robots.
I. J. Robotics Res., 36(12):1312–1340, 2017.

André Platzer.
Logical Foundations of Cyber-Physical Systems.
Springer, Cham, 2018.

Calin Belta and Franjo Ivancic, editors.

Hybrid Systems: Computation and Control (part of CPS Week 2013), HSCC'13, Philadelphia, PA, USA, April 8-13, 2013, New York, 2013. ACM.

João P. Hespanha and Ashish Tiwari, editors.
Hybrid Systems: Computation and Control, 9th International Workshop, HSCC 2006, Santa Barbara, CA, USA, March 29-31, 2006, Proceedings, volume 3927 of *LNCS*. Springer, 2006.