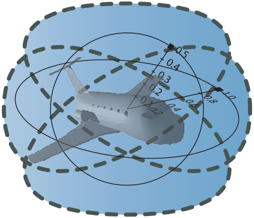
Quantified Differential Dynamic Logic for Distributed Hybrid Systems

André Platzer

Carnegie Mellon University, Pittsburgh, PA



Outline

1. Motivation
2. Quantified Differential Dynamic Logic 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
4. Conclusions

Complex Physical Systems:

Q: I want to verify my car

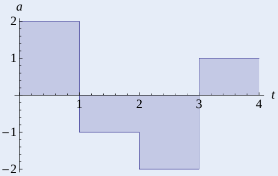
Challenge

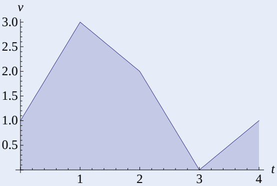
Complex Physical Systems: Hybrid Systems

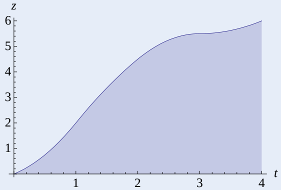
Q: I want to verify my car A: Hybrid systems

Challenge (Hybrid Systems)

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



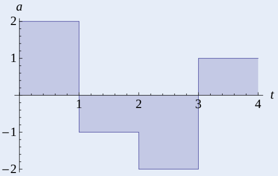


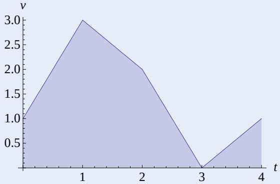


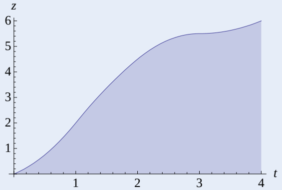
Q: I want to verify my car A: Hybrid systems Q: But there's a lot of cars!

Challenge (Hybrid Systems)

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



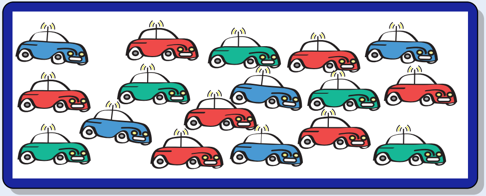




Complex Physical Systems:

Q: I want to verify a lot of cars

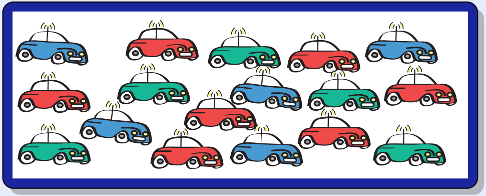
Challenge

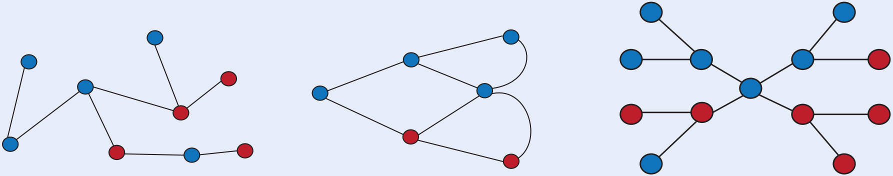


Q: I want to verify a lot of cars A: Distributed systems

Challenge (Distributed Systems)

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



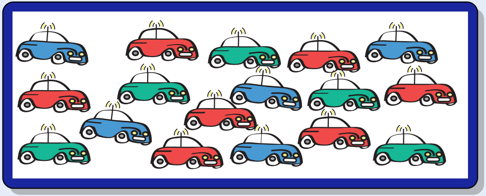


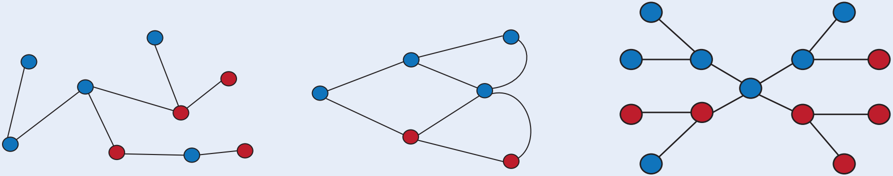
Complex Physical Systems: Distributed Systems

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)

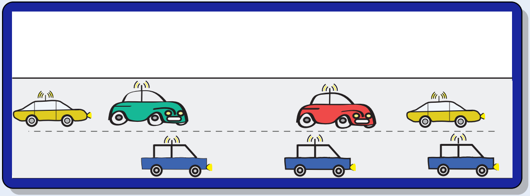




Complex Physical Systems:

Q: I want to verify lots of moving cars

Challenge

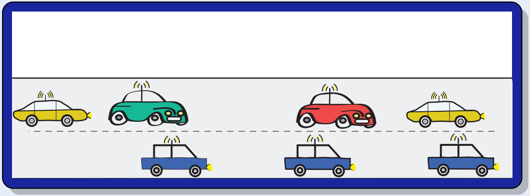


Complex Physical Systems: Distributed Hybrid Systems

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)

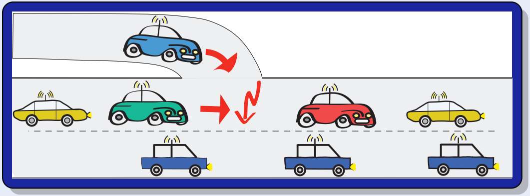


Complex Physical Systems: Distributed Hybrid Systems

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)

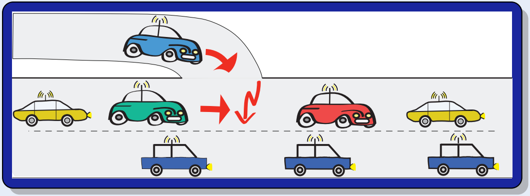


Complex Physical Systems: Distributed Hybrid Systems

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:

Shift [DGV96] The Hybrid System Simulation Programming Language

Hybrid CSP [CJR95] Semantics in Extended Duration Calculus

HyPA [CR05] Translate fragment into normal form.

χ process algebra [vBMR$^+$06] Simulation, translation of fragments to PHAVER, UPPAAL

R-Charon [KSPL06] Modeling Language for Reconfigurable Hybrid Systems

Φ-calculus [Rou04] Semantics in rich set theory

ACP$^{srt}_{hs}$ [BM05] Modeling language proposal

OBSHS [MS06] Partial random simulation of objects

State of the Art: Modeling and Simulation

Shift [DGV96] The Hybrid System Simulation Programming Language

R-Charon [KSPL06] Modeling Language for Reconfigurable Hybrid Systems

Hybrid CSP [CJR95] Semantics in Extended Duration Calculus

Φ-calculus [Rou04] Semantics in rich set theory

HyPA [CR05] Translate fragment into normal form.

ACP_{hs}^{srt} [BM05] Modeling language proposal

χ process algebra [vBMR$^+$06] Simulation, translation of fragments to PHAVER, UPPAAL

OBSHS [MS06] Partial random simulation of objects

State of the Art: Modeling and Simulation

No formal verification of distributed hybrid systems

Shift [DGV96] The Hybrid System Simulation Programming Language

Hybrid CSP [CJR95] Semantics in Extended Duration Calculus

HyPA [CR05] Translate fragment into normal form.

χ process algebra [vBMR+06] Simulation, translation of fragments to PHAVER, UPPAAL

R-Charon [KSPL06] Modeling Language for Reconfigurable Hybrid Systems

Φ-calculus [Rou04] Semantics in rich set theory

ACP^{srt}_{hs} [BM05] Modeling language proposal

OBSHS [MS06] Partial random simulation of objects

Contributions

1. System model and semantics for distributed hybrid systems: QHP
2. Specification and verification logic: Qd$\mathcal{L}$
3. Proof calculus for Qd$\mathcal{L}$
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

Outline

1. Motivation

2. Quantified Differential Dynamic Logic 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

4. Conclusions

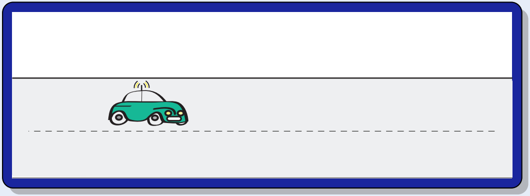
Outline (Conceptual Approach)

1. Motivation
2. Quantified Differential Dynamic Logic 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
4. Conclusions

Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

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

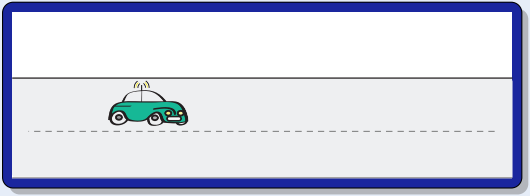


Model for Distributed Hybrid Systems

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)

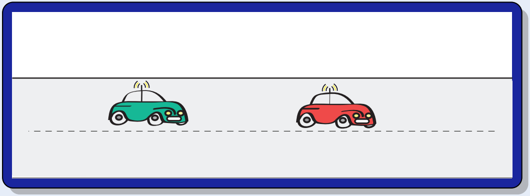


Model for Distributed Hybrid Systems

Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

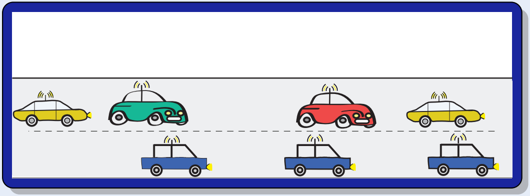
- Continuous dynamics (differential equations)
  $$x'' = a$$
- Discrete dynamics (control decisions)
  $a := \texttt{if} .. \texttt{then}\, A\, \texttt{else} -b$
- Structural dynamics (communication/coupling)



Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

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

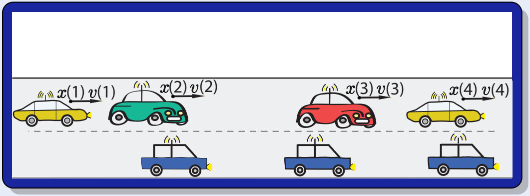


Model for Distributed Hybrid Systems

Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

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

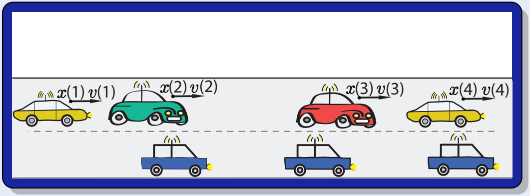


Model for Distributed Hybrid Systems

Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

- Continuous dynamics (differential equations)
  $$x(i)'' = a(i)$$
- 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)

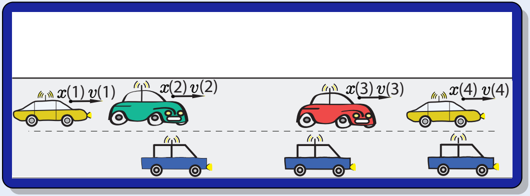
- Continuous dynamics (differential equations)

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

- Discrete dynamics (control decisions)

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

- Structural dynamics (communication/coupling)



Model for Distributed Hybrid Systems

Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

- Continuous dynamics (differential equations)

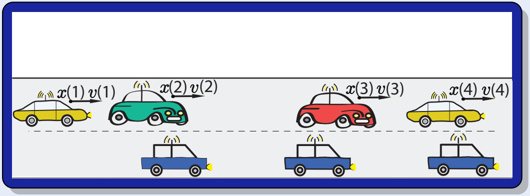
  $\forall i\; x(i)'' = a(i)$

- Discrete dynamics (control decisions)

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

- Structural dynamics (communication/coupling)

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

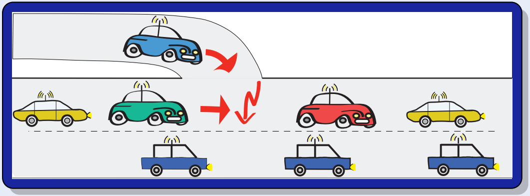


Model for Distributed Hybrid Systems

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)

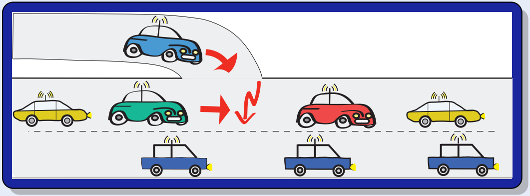


Model for Distributed Hybrid Systems

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



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
$?\chi$	(conditional execution)	jump & test
$\alpha; \beta$	(seq. composition)	Kleene algebra
$\alpha \cup \beta$	(nondet. choice)	Kleene algebra
α^*	(nondet. repetition)	Kleene algebra

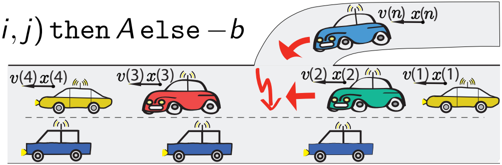
Definition (Quantified hybrid program α)

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

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

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

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



Definition (Quantified hybrid program α)

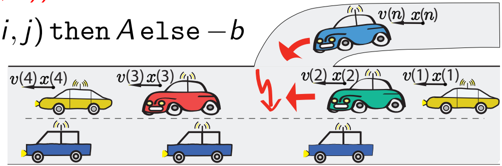
$\forall i : C\ x(s)' = \theta$	(quantified ODE)	
$\forall i : C\ x(s) := \theta$	(quantified assignment)	jump & test
$?\chi$	(conditional execution)	jump & test
$\alpha; \beta$	(seq. composition)	Kleene algebra
$\alpha \cup \beta$	(nondet. choice)	Kleene algebra
α^*	(nondet. repetition)	Kleene algebra

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

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

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

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



Quantified Differential Dynamic Logic 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
$?\chi$	(conditional execution)	
$\alpha;\beta$	(seq. composition)	Kleene algebra
$\alpha \cup \beta$	(nondet. choice)	
α^*	(nondet. repetition)	

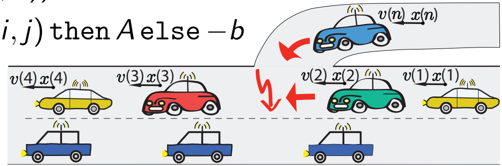
$$DCCS \equiv (appear; ctrl; drive)^*$$

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

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

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

new C is definable!



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

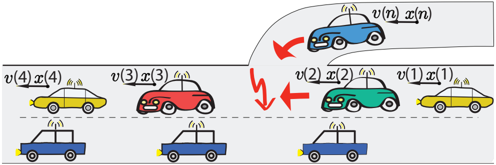
Definition ($\mathsf{Qd}\mathcal{L}$ Formula ϕ)

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

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

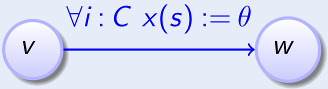
$$\forall i, j : C \ \mathit{far}(i,j) \rightarrow [(\mathit{appear}\,;\mathit{ctrl}\,;\mathit{drive})^*]\ \forall i{\neq}j : C \ x(i) \neq x(j)$$

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

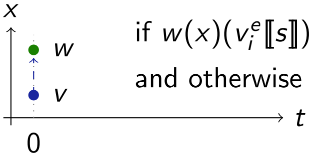


Quantified Differential Dynamic Logic QdL: Semantics

Definition (Quantified hybrid program α: transition semantics)



Details

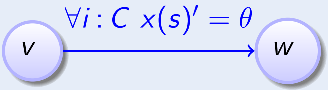


if $w(x)(v_i^e[\![s]\!]) = v_i^e[\![\theta]\!]$ (for all e)

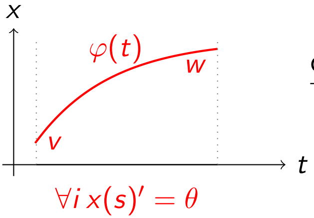
and otherwise unchanged

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

Definition (Quantified hybrid program α: transition semantics)



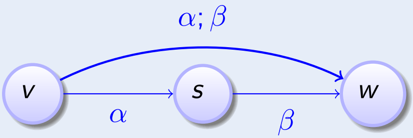
▸ Details ▸▸



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

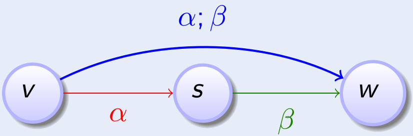
Quantified Differential Dynamic Logic Qd$\mathcal{L}$: Semantics

Definition (Quantified hybrid program α: transition semantics)

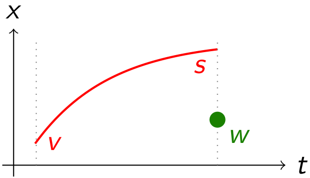


▸ Details ⏩

Definition (Quantified hybrid program α: transition semantics)

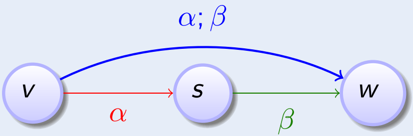


▸ Details ⏩

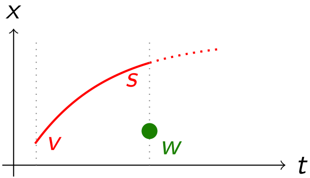


Quantified Differential Dynamic Logic Qd$\mathcal{L}$: Semantics

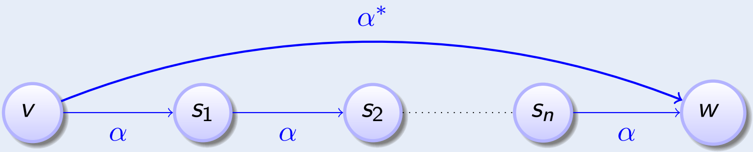
Definition (Quantified hybrid program α: transition semantics)



▸ Details ▸▸

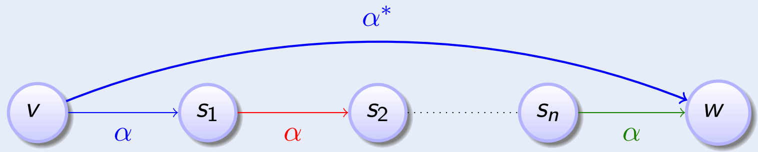


Definition (Quantified hybrid program α: transition semantics)

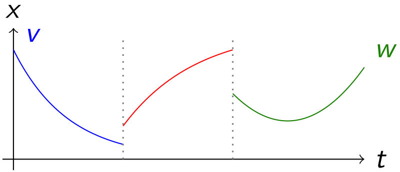


Details

Definition (Quantified hybrid program α: transition semantics)

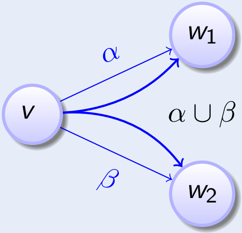


▸ Details ▸▸



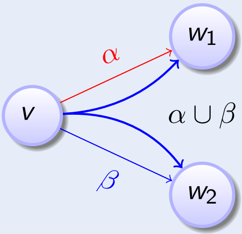
Quantified Differential Dynamic Logic Qd$\mathcal{L}$: Semantics

Definition (Quantified hybrid program α: transition semantics)

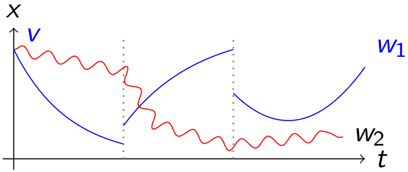


▸ Details ⏩

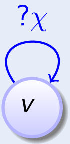
Definition (Quantified hybrid program α: transition semantics)



Details ▸▸

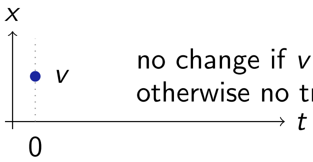


Definition (Quantified hybrid program α: transition semantics)



if $v \models \chi$

Details



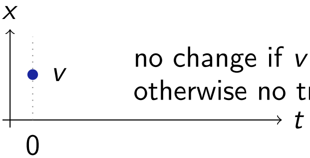
no change if $v \models \chi$
otherwise no transition

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

Definition (Quantified hybrid program α: transition semantics)

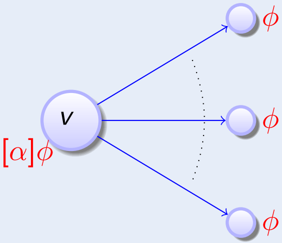
if $v \not\models \chi$

Details



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

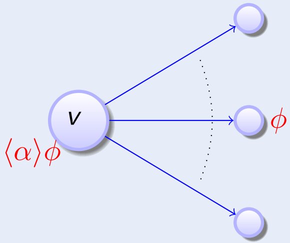
Definition (Q$d\mathcal{L}$ Formula ϕ)



Details

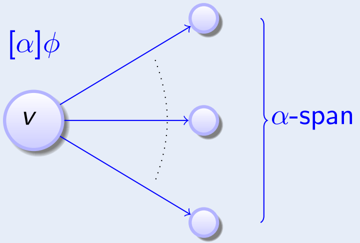
Quantified Differential Dynamic Logic Q$d\mathcal{L}$: Semantics

Definition (Q$d\mathcal{L}$ Formula ϕ)



Details

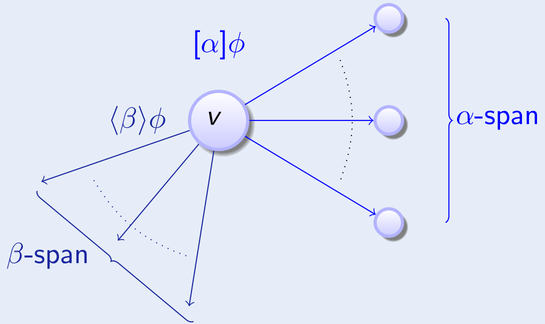
Definition (QdL Formula ϕ)



Details

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

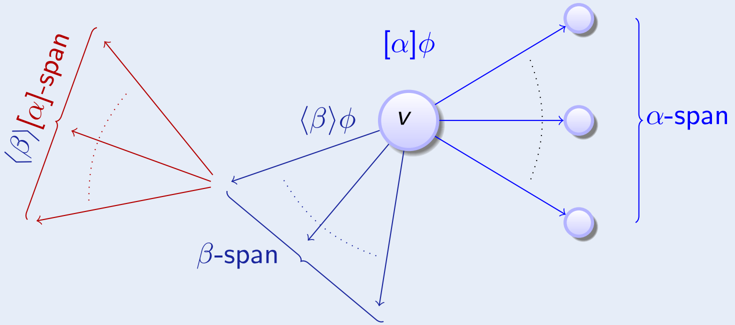
Definition (Qd$\mathcal{L}$ Formula ϕ)



Details

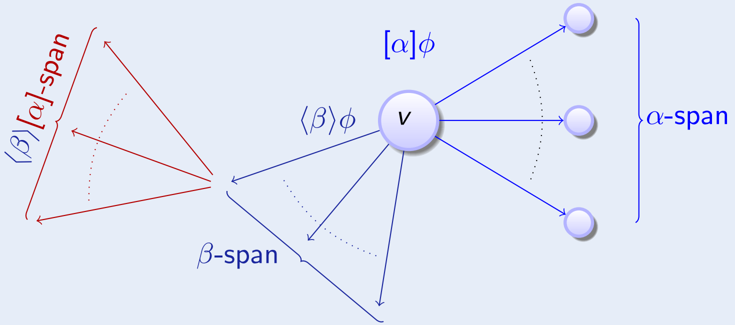
Quantified Differential Dynamic Logic Qd$\mathcal{L}$: Semantics

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



Details

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



Details

compositional semantics $\Rightarrow$ compositional calculus!

Outline (Verification Approach)

1. Motivation

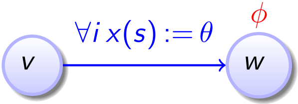
2. Quantified Differential Dynamic Logic Qd$\mathcal{L}$
   - Design
   - Syntax
   - Semantics

3. Proof Calculus for Distributed Hybrid Systems
   - Compositional Verification Calculus
   - Deduction Modulo with Free Variables & Skolemization
   - Actual Existence and Creation
   - Soundness and Completeness

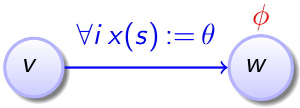
4. Conclusions

Proof Calculus for Quantified Differential Dynamic Logic

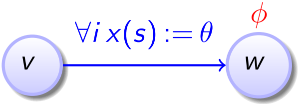
$$\frac{\texttt{if}\ \exists i\ s = u\ \texttt{then}\ \forall i\ (s = u \rightarrow \phi(\theta))\ \texttt{else}\ \phi(x(u))}{\phi([\underbrace{\forall i\ x(s) := \theta}]x(u))}$$



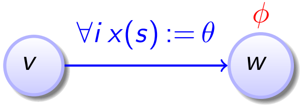
$$\frac{\texttt{if}\ \exists i\, s = u\ \texttt{then}\ \forall i\, (s = u \rightarrow \phi(\theta))\ \texttt{else}\ \phi(x(u))}{\phi([\underbrace{\forall i\, x(s) := \theta}]x(u))}$$



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

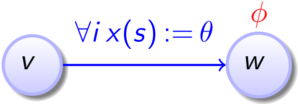


$$\frac{\texttt{if}\ \exists i\, s = u\ \texttt{then}\ \forall i\, (s = u \to \phi(\theta))\ \texttt{else}\ \phi(x(u))}{\phi([\underbrace{\forall i\, x(s) := \theta}]x(u))}$$



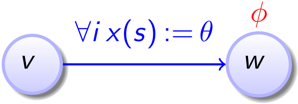
Proof Calculus for Quantified Differential Dynamic Logic

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

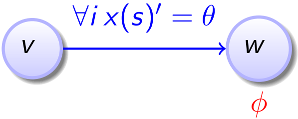


Proof Calculus for Quantified Differential Dynamic Logic

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

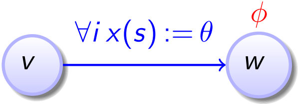


$$\frac{\exists t{\geq}0\ \langle \forall i\, \mathcal{S}(t)\rangle \phi}{\langle \forall i\, x(s)' = \theta\rangle \phi}$$

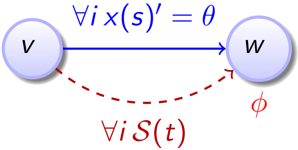


Proof Calculus for Quantified Differential Dynamic Logic

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



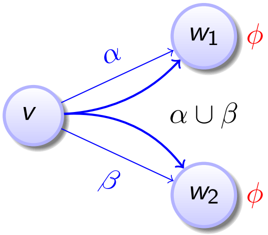
$$\frac{\exists t \geq 0\ \langle \forall i\, \mathcal{S}(t)\rangle \phi}{\langle \forall i\, x(s)' = \theta \rangle \phi}$$



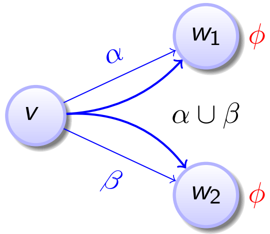
Proof Calculus for Quantified Differential Dynamic Logic

compositional semantics $\Rightarrow$ compositional rules!

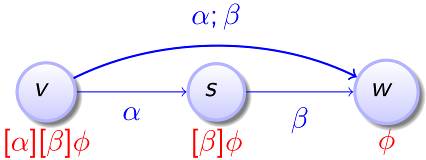
$$\frac{[\alpha]\phi \land [\beta]\phi}{[\alpha \cup \beta]\phi}$$



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

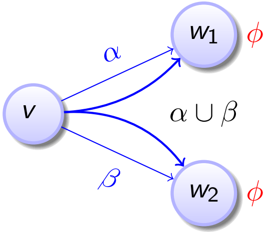


$$\frac{[\alpha][\beta]\phi}{[\alpha; \beta]\phi}$$

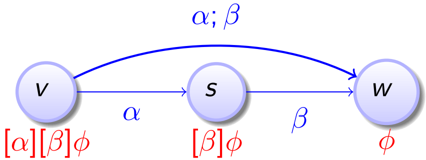


Proof Calculus for Quantified Differential Dynamic Logic

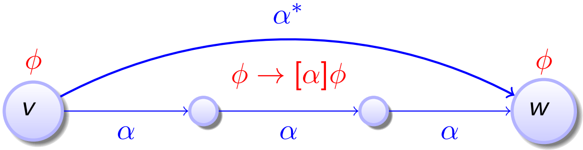
$$\frac{[\alpha]\phi \land [\beta]\phi}{[\alpha \cup \beta]\phi}$$



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

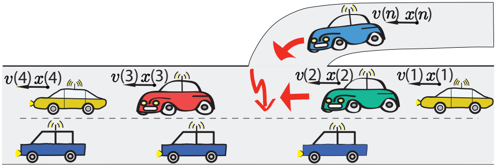


$$\frac{\phi \quad (\phi \to [\alpha]\phi)}{[\alpha^*]\phi}$$



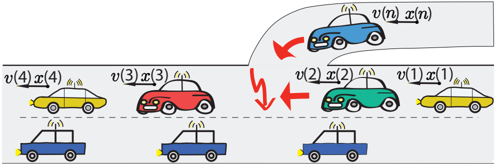
Deduction Modulo with Free Variables & Skolemization

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

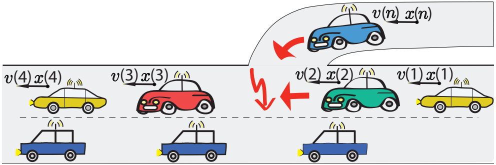


Deduction Modulo with Free Variables & Skolemization

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



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



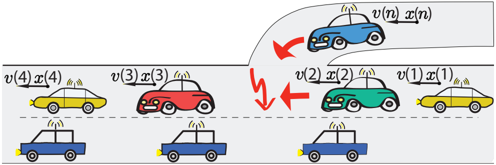
Deduction Modulo with Free Variables & Skolemization

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

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

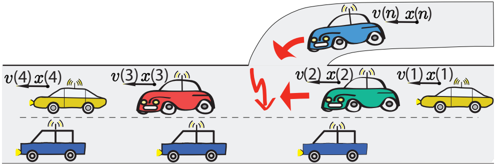
$$\forall i{\neq}j\, x(i){\neq}x(j) \rightarrow [\forall i\, x(i)' = v(i), v(i)' = -b]\, \forall j{\neq}k\, x(j){\neq}x(k)$$

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



Deduction Modulo with Free Variables & Skolemization

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



Deduction Modulo with Free Variables & Skolemization

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

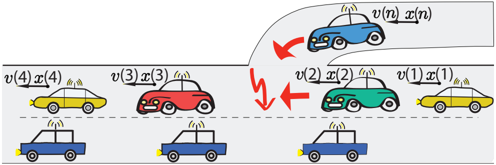
$$\forall i{\neq}j\, x(i){\neq}x(j), s{\geq}0 \rightarrow [\forall i\, x(i) := -\tfrac{b}{2}s^2 + v(i)s + x(i)]\, \forall j{\neq}k\, x(j){\neq}x(k)$$

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

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

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

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



Deduction Modulo with Free Variables & Skolemization

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

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

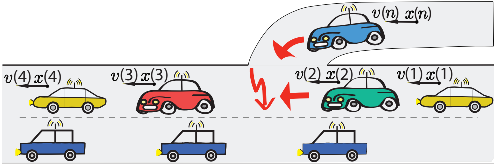
$$\forall i{\neq}j\, x(i){\neq}x(j), s{\geq}0 \rightarrow [\forall i\, x(i) := -\tfrac{b}{2}s^2 + v(i)s + x(i)]\, \forall j{\neq}k\, x(j){\neq}x(k)$$

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

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

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

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



Deduction Modulo with Free Variables & Skolemization

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

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

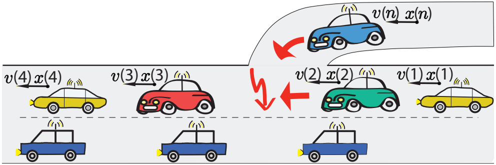
$$\forall i{\neq}j\, x(i){\neq}x(j), s{\geq}0 \rightarrow [\forall i\, x(i) := -\tfrac{b}{2}s^2 + v(i)s + x(i)]\, \forall j{\neq}k\, x(j){\neq}x(k)$$

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

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

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

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



Deduction Modulo with Free Variables & Skolemization

$$\forall i{\neq}j\, x(i){\neq}x(j) \rightarrow \forall j{\neq}k\, (x(j){\leq}x(k){\wedge}v(j){\leq}v(k) \vee x(j){\geq}x(k){\wedge}v(j){\geq}v(k))$$

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

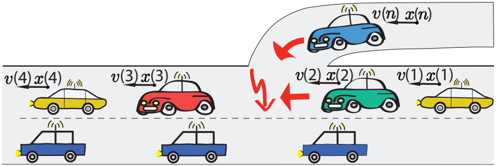
$$\forall i{\neq}j\, x(i){\neq}x(j), s{\geq}0 \rightarrow [\forall i\, x(i) := -\tfrac{b}{2}s^2 + v(i)s + x(i)]\, \forall j{\neq}k\, x(j){\neq}x(k)$$

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

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

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

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



Deduction Modulo with Free Variables & Skolemization

$$\forall X, Y, V, W\,(X\neq Y \rightarrow X\leq Y\wedge V\leq W \vee X\geq Y\wedge V\geq W)$$

$$\forall i\neq j\; x(i)\neq x(j) \rightarrow \forall j\neq k\,(x(j)\leq x(k)\wedge v(j)\leq v(k) \vee x(j)\geq x(k)\wedge v(j)\geq v(k))$$

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

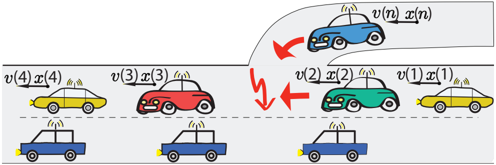
$$\forall i\neq j\; x(i)\neq x(j), s\geq 0 \rightarrow [\forall i\; x(i) := -\tfrac{b}{2}s^2 + v(i)s + x(i)]\,\forall j\neq k\; x(j)\neq x(k)$$

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

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

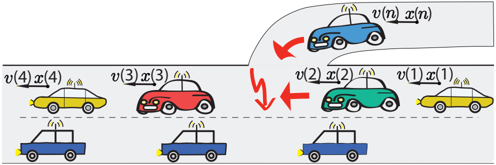
$$\forall i\neq j\; x(i)\neq x(j) \rightarrow [\forall i\; x(i)' = v(i), v(i)' = -b]\,\forall j\neq k\; x(j)\neq x(k)$$

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



Deduction Modulo with Free Variables & Skolemization

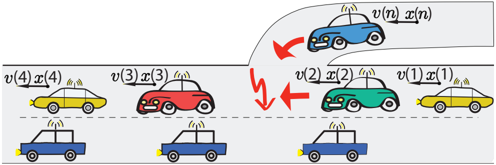
$$\frac{\forall X, Y, V, W\, (X \neq Y \to X \leq Y \land V \leq W \lor X \geq Y \land V \geq W)}{\dfrac{\forall i{\neq}j\, x(i){\neq}x(j) \to \forall j{\neq}k\, (x(j){\leq}x(k) \land v(j){\leq}v(k) \lor x(j){\geq}x(k) \land v(j){\geq}v(k))}{\dfrac{\forall i{\neq}j\, x(i){\neq}x(j), s{\geq}0 \to \forall j{\neq}k\, (-\frac{b}{2}s^2 + v(j)s + x(j) \neq -\frac{b}{2}s^2 + v(k)s + x(k))}{\dfrac{\forall i{\neq}j\, x(i){\neq}x(j), s{\geq}0 \to [\forall i\, x(i) := -\frac{b}{2}s^2 + v(i)s + x(i)]\, \forall j{\neq}k\, x(j){\neq}x(k)}{\dfrac{\forall i{\neq}j\, x(i){\neq}x(j) \to s{\geq}0 \to [\forall i\, x(i) := -\frac{b}{2}s^2 + v(i)s + x(i)]\, \forall j{\neq}k\, x(j){\neq}x(k)}{\dfrac{\forall i{\neq}j\, x(i){\neq}x(j) \to \forall t{\geq}0\, [\forall i\, x(i) := -\frac{b}{2}t^2 + v(i)t + x(i)] \forall j{\neq}k\, x(j){\neq}x(k)}{\dfrac{\forall i{\neq}j\, x(i){\neq}x(j) \to [\forall i\, x(i)' = v(i), v(i)' = -b]\, \forall j{\neq}k\, x(j){\neq}x(k)}{\forall i{\neq}j\, x(i){\neq}x(j) \to [\forall i\, x(i)'' = -b]\, \forall j{\neq}k\, x(j){\neq}x(k)}}}}}}}$$



Actual Existence and Creation

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

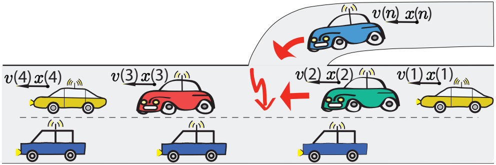
$$\mathrm{E}(i) = \begin{cases} 0 & \text{if } i \text{ denotes a possible object} \\ 1 & \text{if } i \text{ denotes an actively existing objects} \end{cases}$$



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

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

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

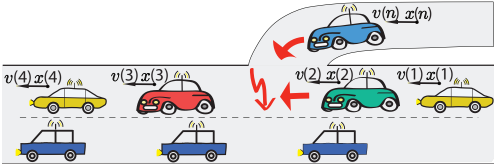


Actual Existence and Creation

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

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

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

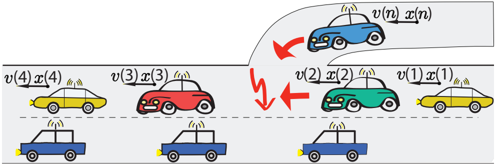


Actual Existence and Creation

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

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

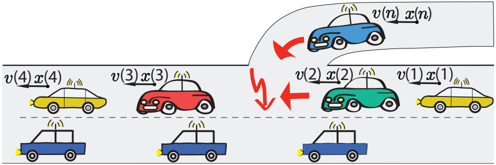
$$\frac{[(\forall j : C \;\; n := j);\;\; ?(\mathrm{E}(n) = 0);\qquad\qquad]\phi}{[n := \mathtt{new}\ C]\phi}$$



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

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

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



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

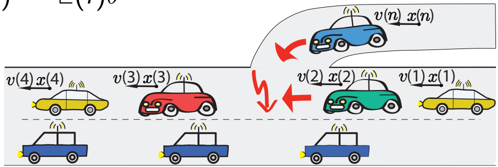
$$\mathbb{E}(i) = \begin{cases} 0 & \text{if } i \text{ denotes a possible object} \\ 1 & \text{if } i \text{ denotes an actively existing objects} \end{cases}$$

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

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

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

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



Soundness and Completeness

Theorem (Relative Completeness)

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

Proof 16p.

Soundness and Completeness

Theorem (Relative Completeness)

Qd$\mathcal{L}$ *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!

Soundness and Completeness

Theorem (Relative Completeness)

Qd$\mathcal{L}$ 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!

Corollary (Yes, we can!)

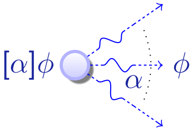
distributed hybrid systems can be verified by recursive decomposition

Outline

Conclusions

quantified differential dynamic logic

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

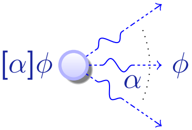


- Distributed hybrid systems everywhere
- System model and semantics
- Logic for distributed hybrid systems
- Compositional proof calculus
- First verification approach
- Sound & complete / diff. eqn.
- Simple distributed car control verified

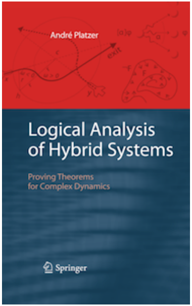
Conclusions

quantified differential dynamic logic

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



- Distributed hybrid systems everywhere
- System model and semantics
- Logic for distributed hybrid systems
- Compositional proof calculus
- First verification approach
- Sound & complete / diff. eqn.
- Simple distributed car control verified



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

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. Springer, 1995.

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

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.

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.

Fabian Kratz, Oleg Sokolsky, George J. Pappas, and Insup Lee.
R-Charon, a modeling language for reconfigurable hybrid systems.
In Hespanha and Tiwari [HT06], pages 392–406.

José Meseguer and Raman Sharykin.
Specification and analysis of distributed object-based stochastic hybrid systems.
In Hespanha and Tiwari [HT06], pages 460–475.

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.

A complete axiomatization of quantified differential dynamic logic for distributed hybrid systems.
Logical Methods in Computer Science, 2012.
Special issue for selected papers from CSL'10.

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.

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.