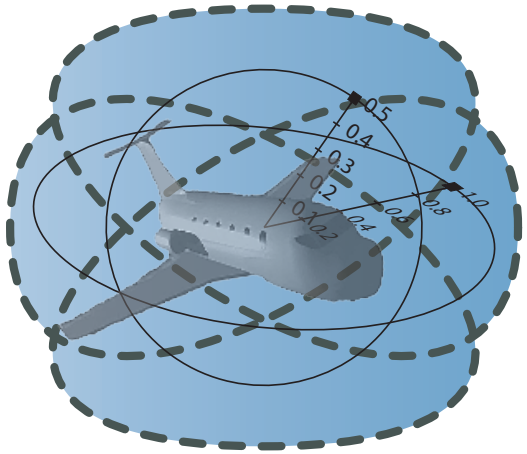
25: Distributed Systems & Hybrid Systems

15-424: Foundations of Cyber-Physical Systems

André Platzer

Carnegie Mellon University, Pittsburgh, PA



Outline

Complex Physical Systems:

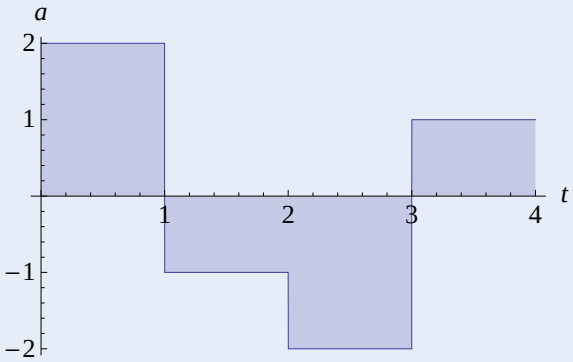
Q: I want to verify my car

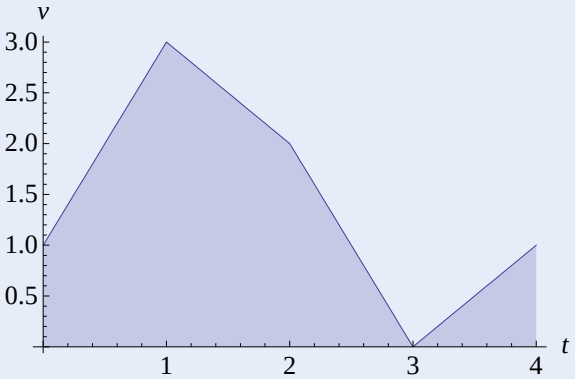
Challenge

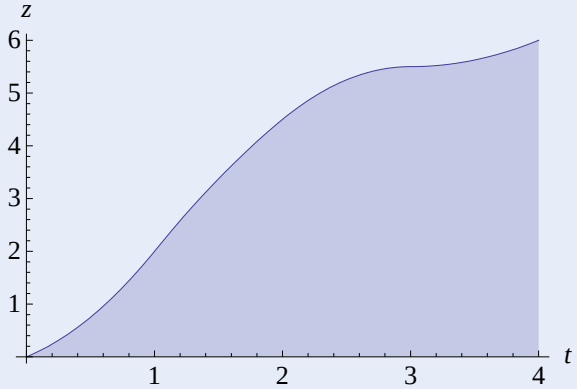
Q: I want to verify my car A: Hybrid systems

Challenge (Hybrid Systems)

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



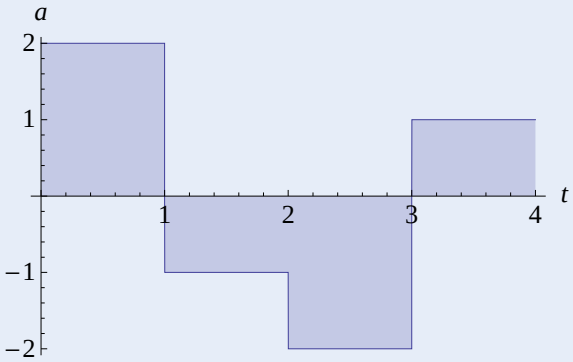


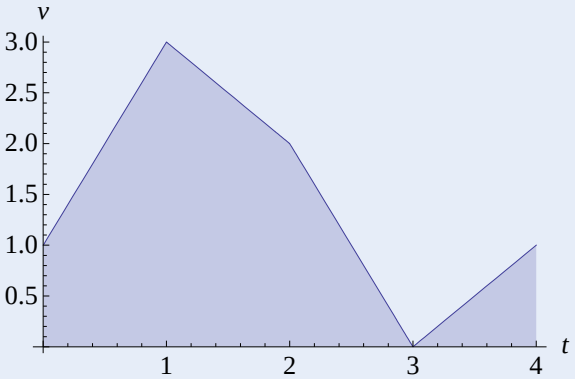


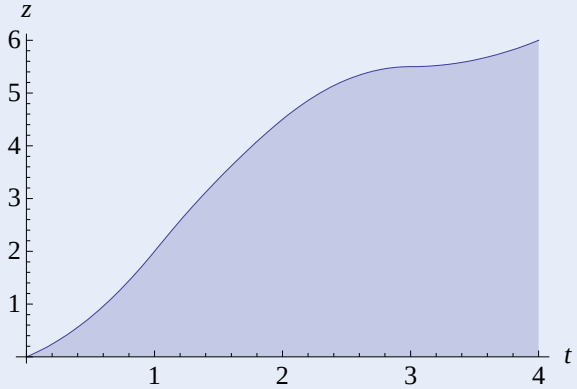
Q: I want to verify my car A: Hybrid systems Q: But there's a lot of cars!

Challenge (Hybrid Systems)

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



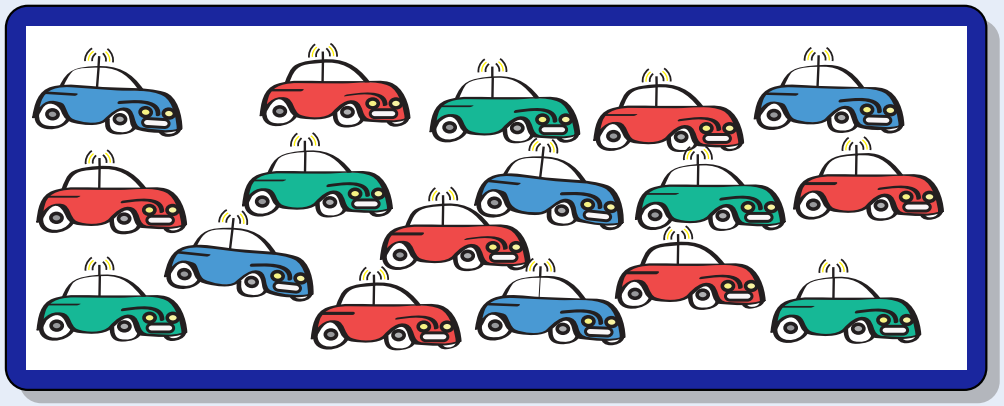




Complex Physical Systems:

Q: I want to verify a lot of cars

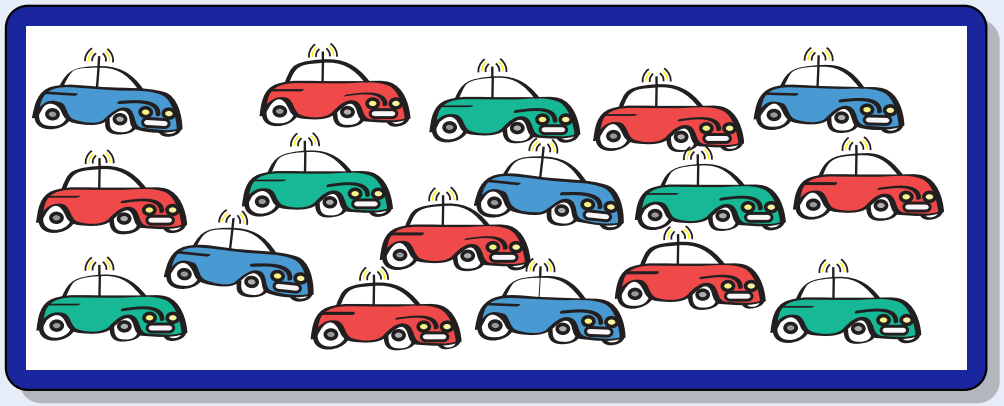
Challenge

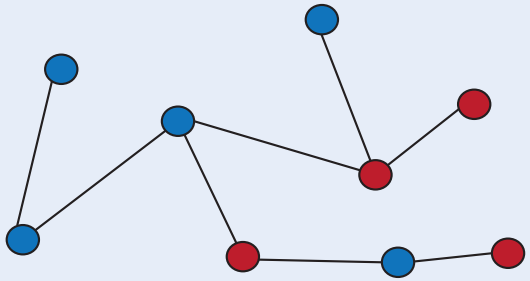


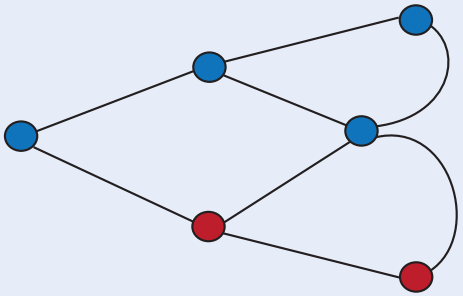
Q: I want to verify a lot of cars A: Distributed systems

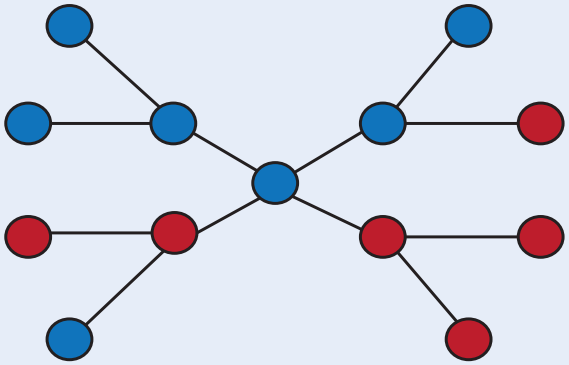
Challenge (Distributed Systems)

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





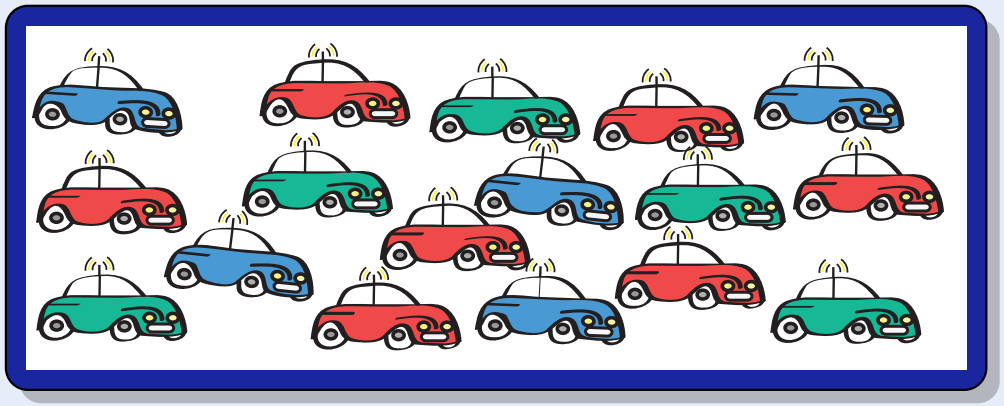


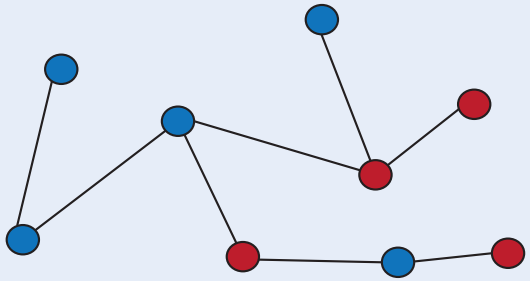


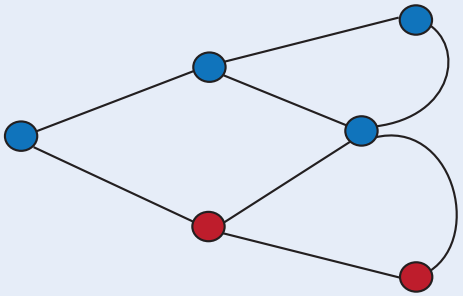
Q: I want to verify a lot of cars A: Distributed systems Q: But they move!

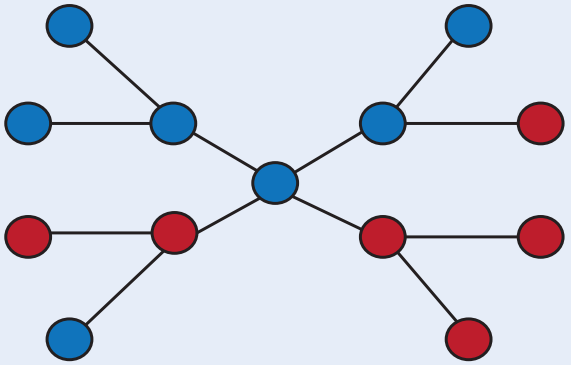
Challenge (Distributed Systems)

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





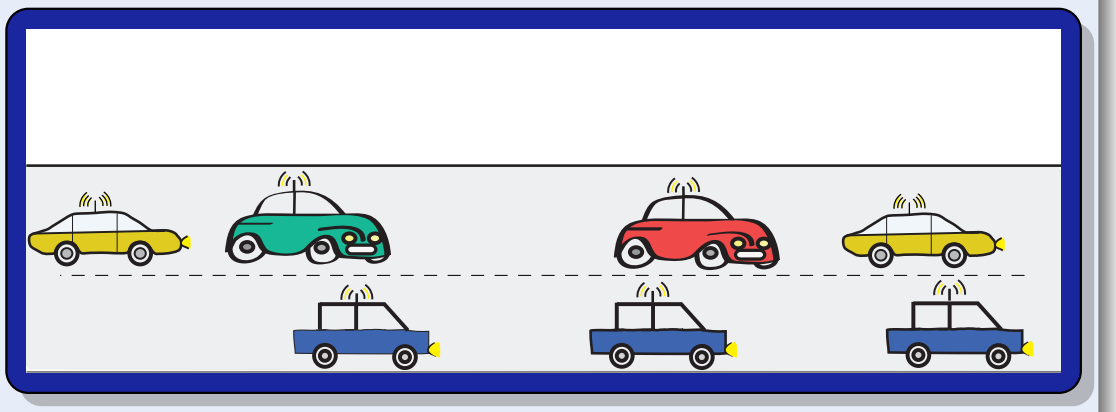




Complex Physical Systems:

Q: I want to verify lots of moving cars

Challenge

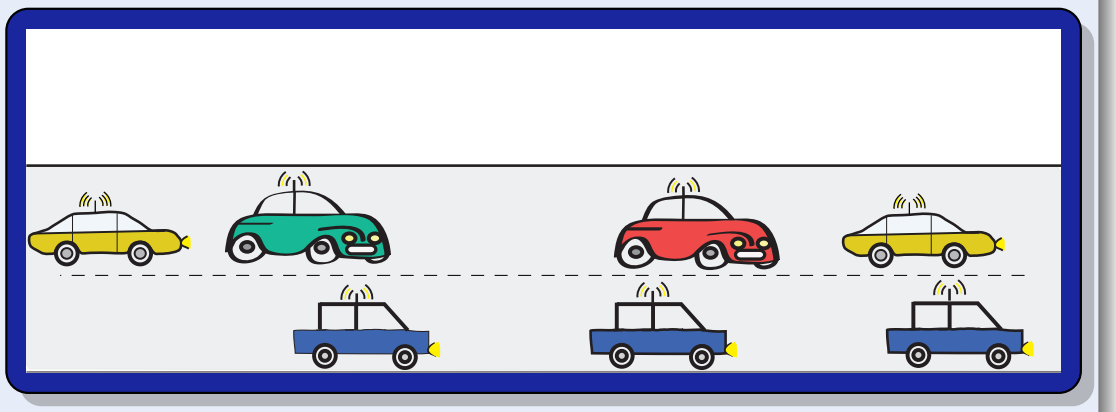


Complex Physical Systems: Distributed Hybrid Systems

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)

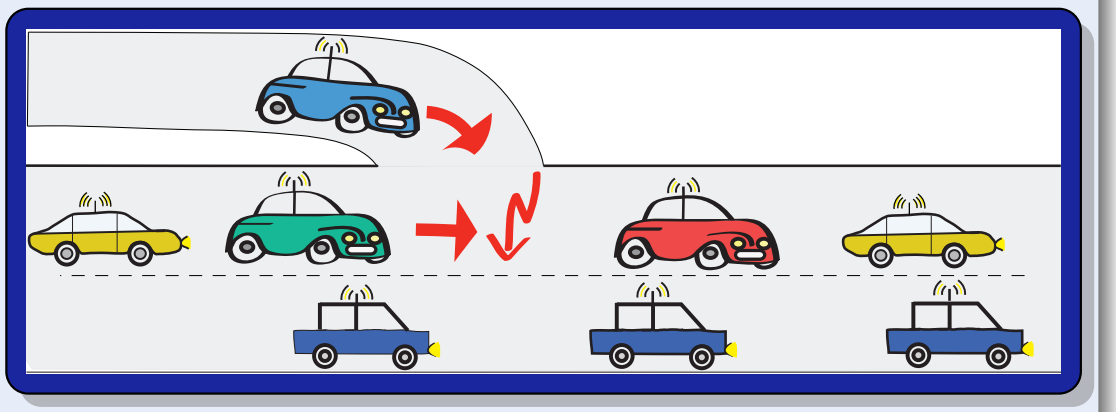


Complex Physical Systems: Distributed Hybrid Systems

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)

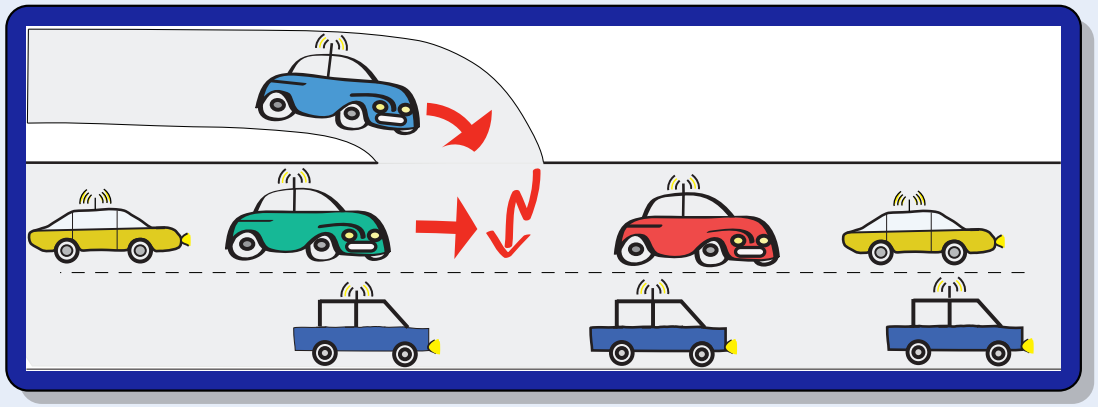


Complex Physical Systems: Distributed Hybrid Systems

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: 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_{hs}^{srt} [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: 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

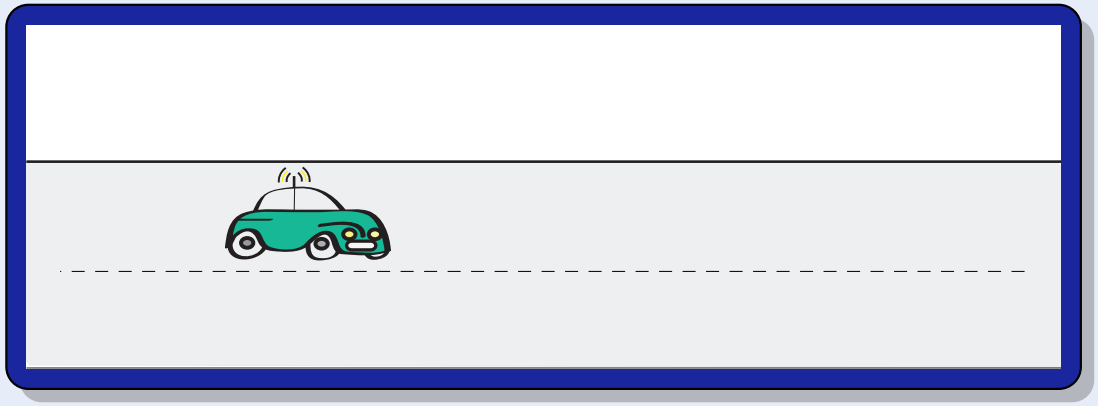
Outline (Conceptual Approach)

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)

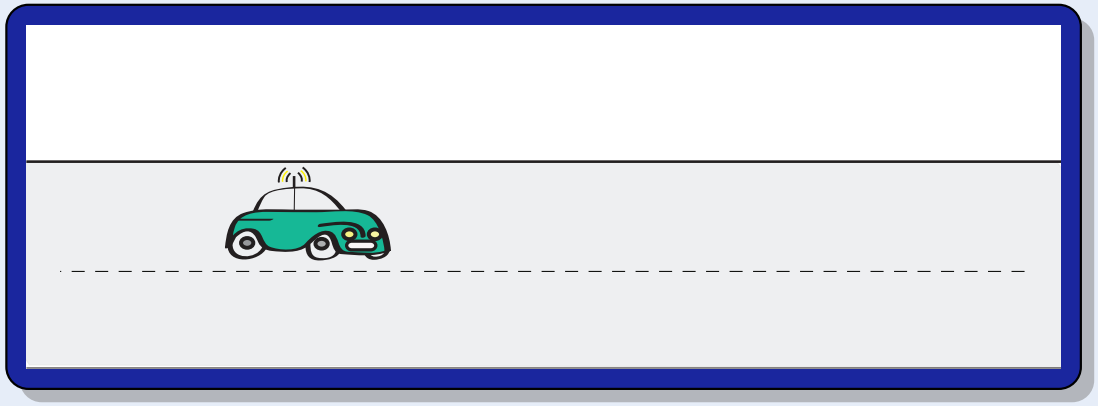


Model for Distributed Hybrid Systems

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)

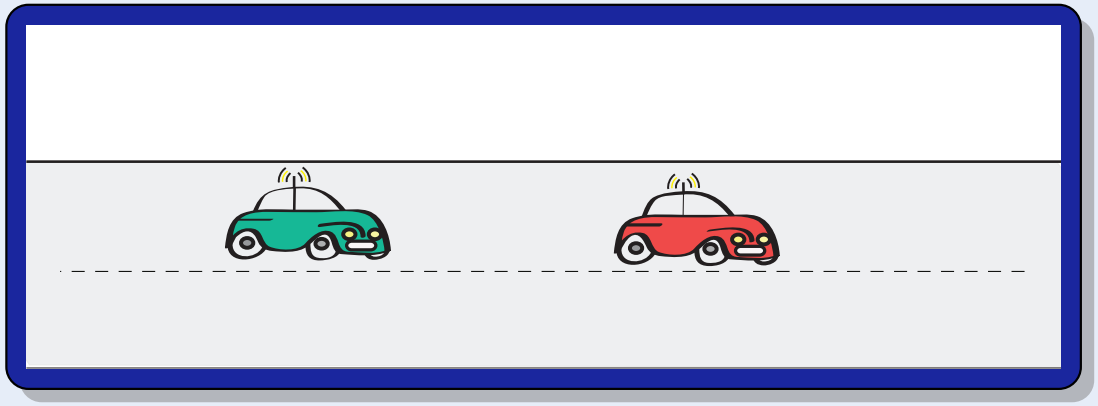


Model for Distributed Hybrid Systems

Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

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

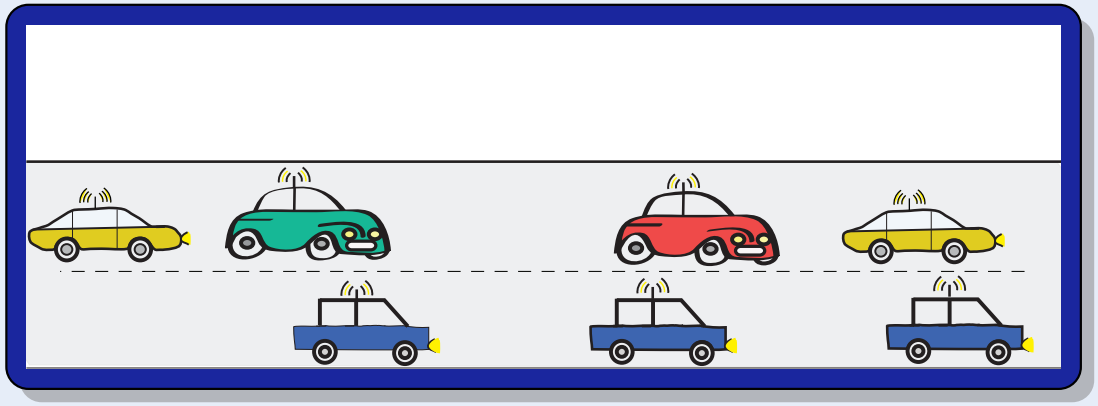


Model for Distributed Hybrid Systems

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)

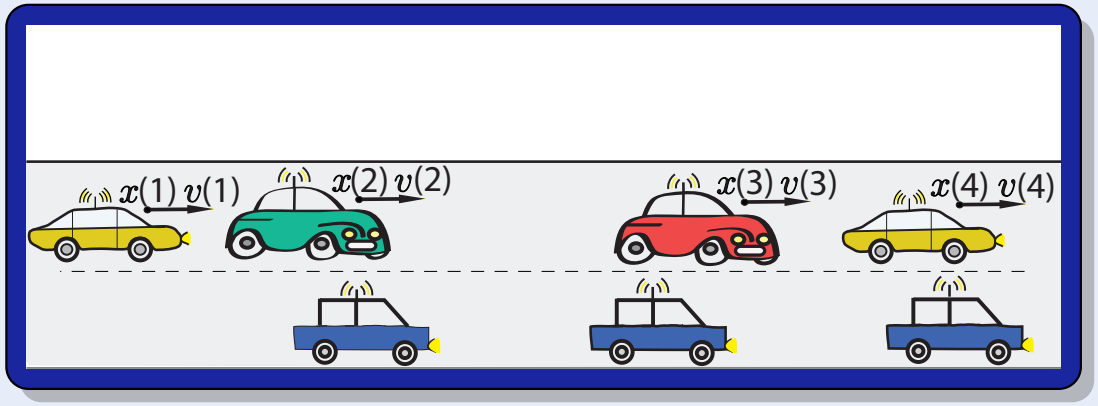


Model for Distributed Hybrid Systems

Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

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

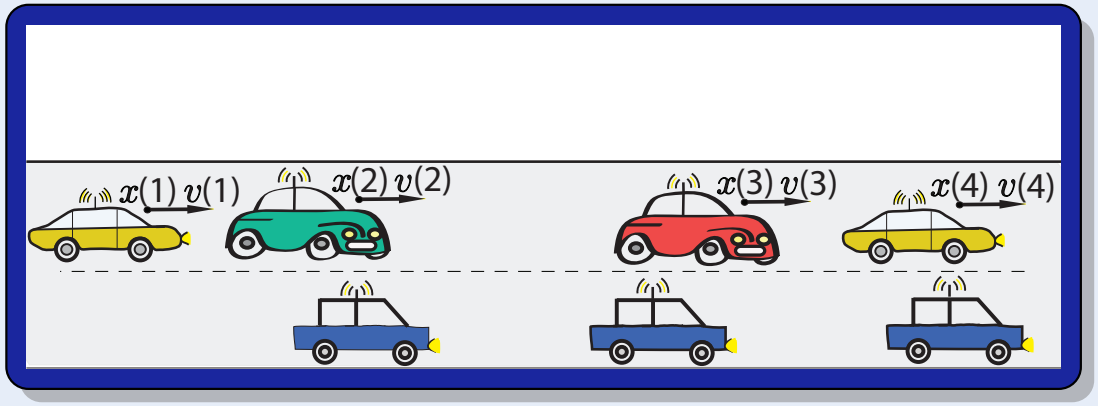


Model for Distributed Hybrid Systems

Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

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



Model for Distributed Hybrid Systems

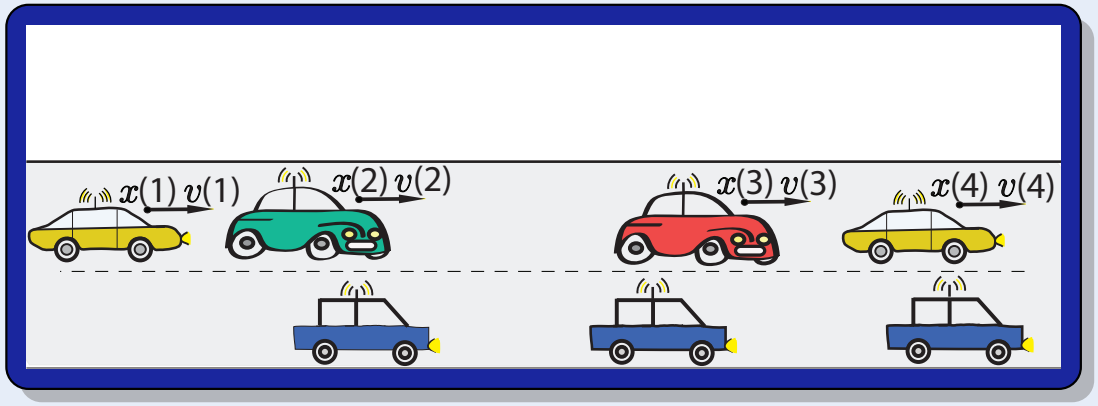
Q: How to model distributed hybrid systems

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)



Model for Distributed Hybrid Systems

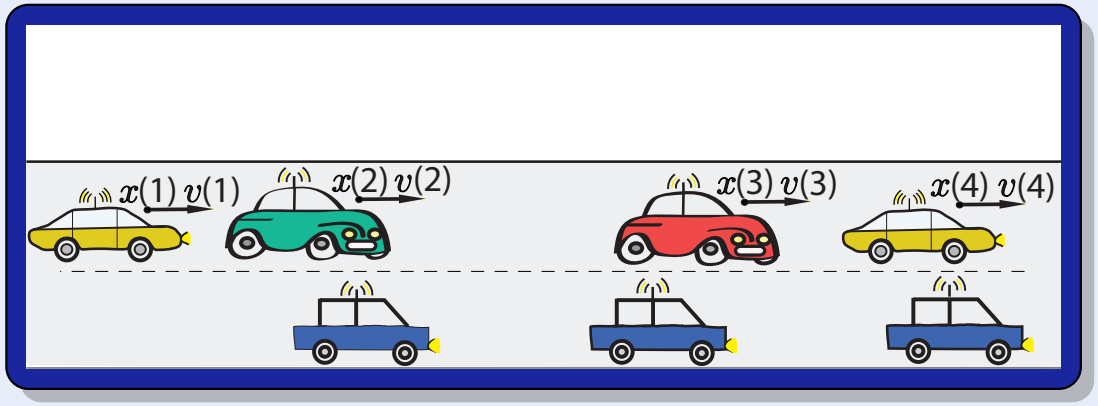
Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

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

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

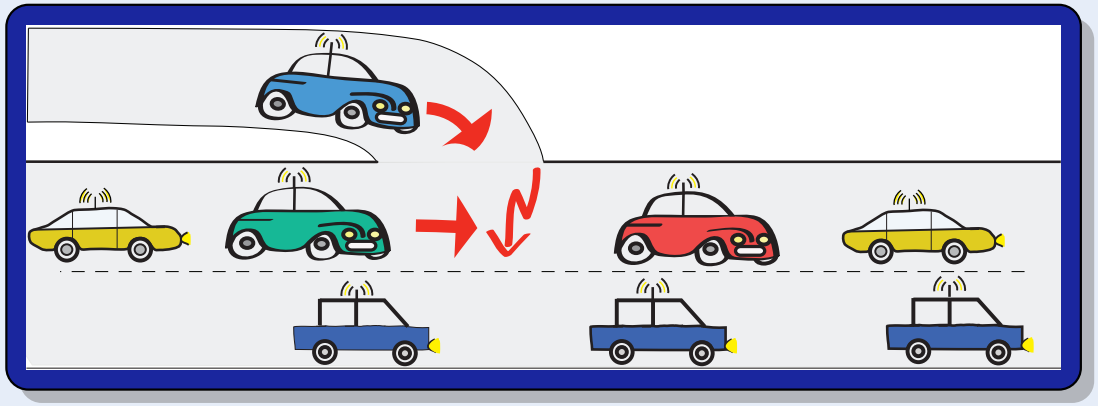
- Structural dynamics (communication/coupling)
  $\ell(i) := \mathit{carInFrontOf}(i)$



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)

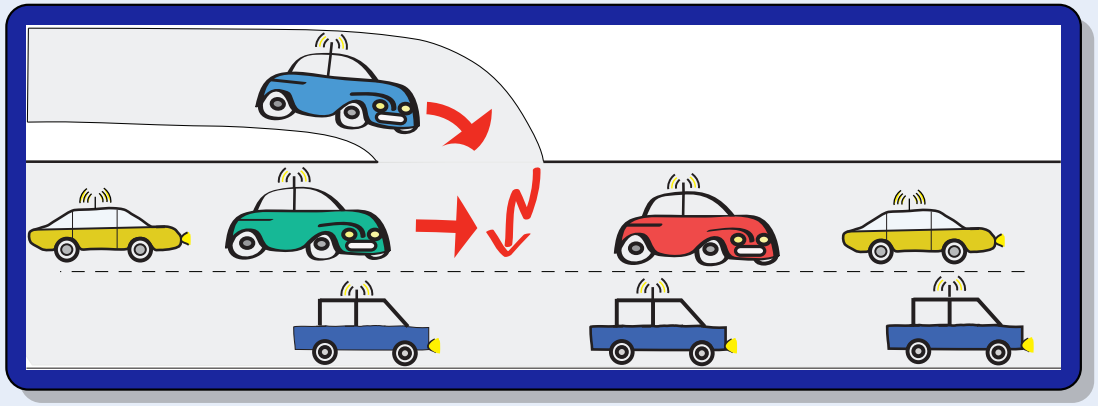
$\forall i\, a(i) := \texttt{if} \,..\, \texttt{then}\, A\, \texttt{else} - b$

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

  $\ell(i) := \mathit{carInFrontOf}(i)$

- Dimensional dynamics (appearance)

  $n := \texttt{new}\, \mathit{Car}$

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

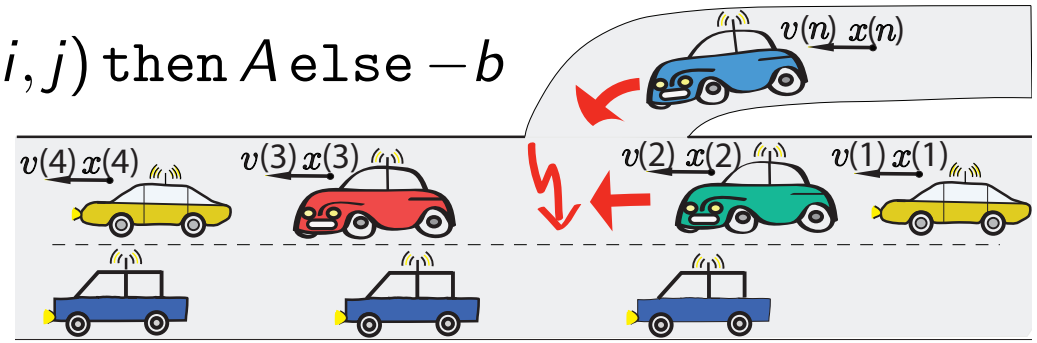
Definition (Quantified hybrid program α)

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

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

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

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

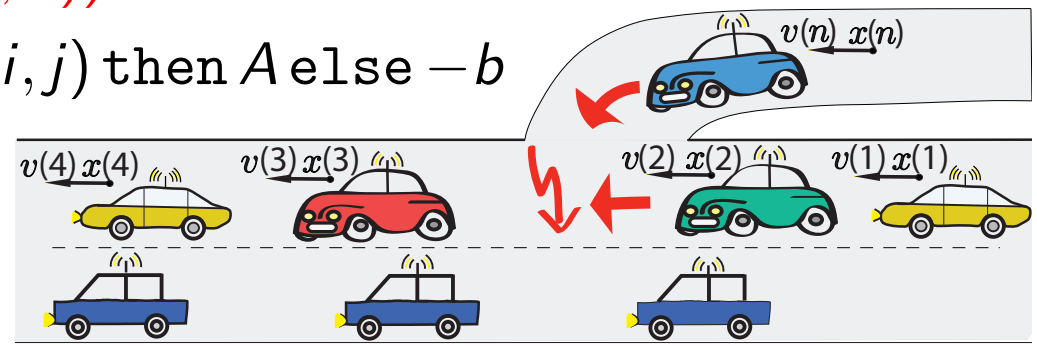


Quantified Differential Dynamic Logic Qd$\mathcal{L}$: Syntax

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

$$
\begin{aligned}
DCCS &\equiv (appear; ctrl; drive)^* \\
appear &\equiv n := \texttt{new}\ C;\quad ?(\forall j : C\ far(j, n)) \\
ctrl &\equiv \forall i : C\ a(i) := \texttt{if}\ \forall j : C\ far(i, j)\ \texttt{then}\ A\ \texttt{else}\ -b \\
drive &\equiv \forall i : C\ x(i)'' = a(i)
\end{aligned}
$$



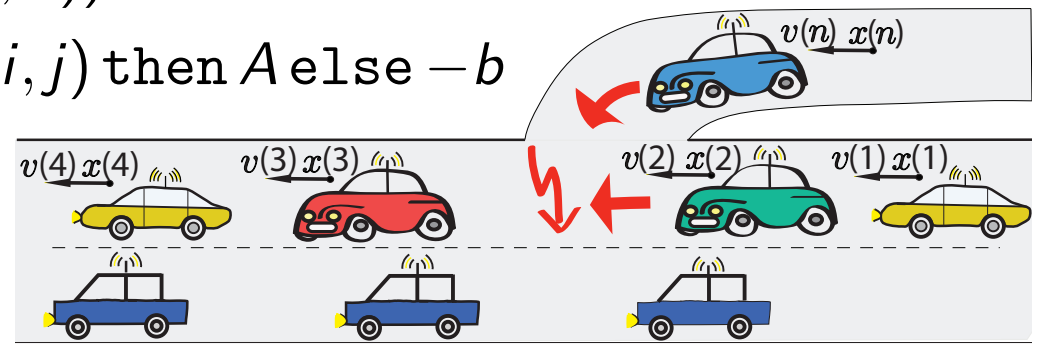
Quantified Differential Dynamic Logic Qd$\mathcal{L}$: Syntax

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

$$
\begin{aligned}
DCCS &\equiv (appear; ctrl; drive)^* \\
appear &\equiv n := \texttt{new}\ C;\quad ?(\forall j : C\ far(j, n)) \\
ctrl &\equiv \forall i : C\ a(i) := \texttt{if}\ \forall j : C\ far(i, j)\ \texttt{then}\ A\ \texttt{else}\ -b \\
drive &\equiv \forall i : C\ x(i)'' = a(i)
\end{aligned}
$$

$\texttt{new}\ C$ is definable!

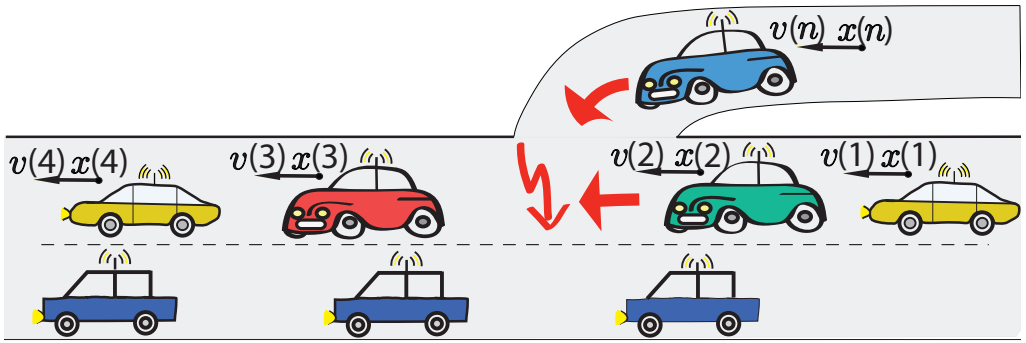


Quantified Differential Dynamic Logic Qd$\mathcal{L}$: Syntax

Definition (Qd$\mathcal{L}$ 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)

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

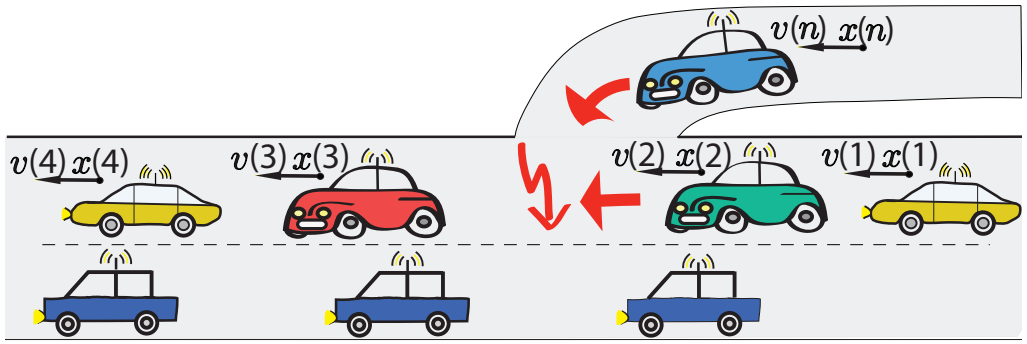


Quantified Differential Dynamic Logic Qd$\mathcal{L}$: Syntax

Definition (Qd$\mathcal{L}$ 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)

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



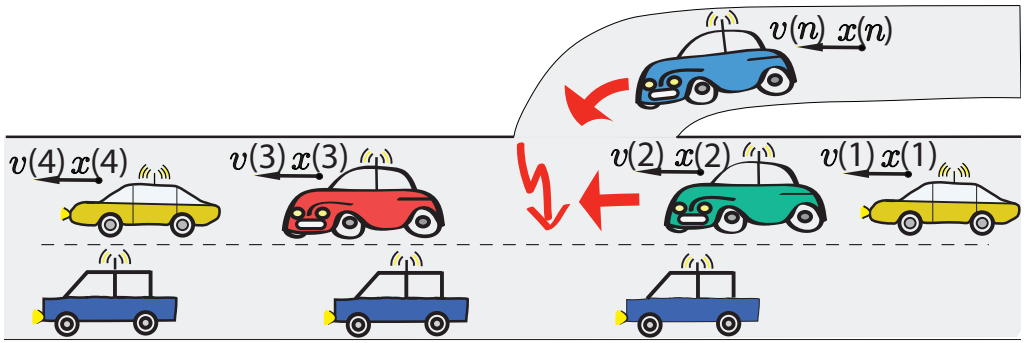
Quantified Differential Dynamic Logic Qd$\mathcal{L}$: Syntax

Definition (Qd$\mathcal{L}$ 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)

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

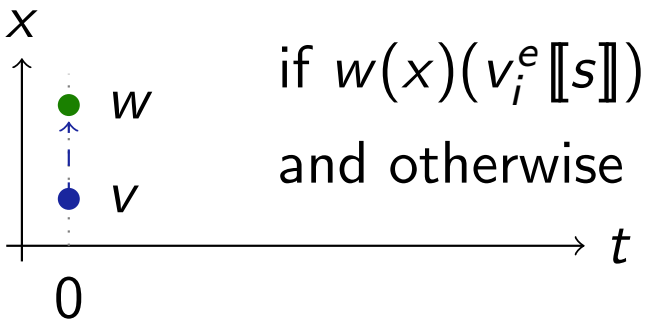
$$far(i,j) \ \equiv \ i \neq j \ \rightarrow \ x(i) < x(j) \wedge v(i) \leq v(j) \wedge a(i) \leq a(j)$$
$$\vee\, x(i) > x(j) \wedge v(i) \geq v(j) \wedge a(i) \geq a(j) \ldots$$



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

Definition (Quantified hybrid program α: transition semantics)

$$v \xrightarrow{\forall i : C \; x(s) := \theta} w$$



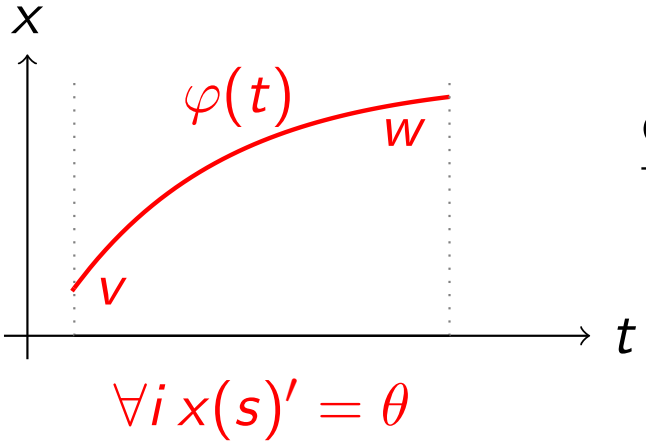
if $w(x)(v_i^e[\![s]\!]) = v_i^e[\![\theta]\!]$ (for all e)

and otherwise unchanged

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

Definition (Quantified hybrid program α: transition semantics)

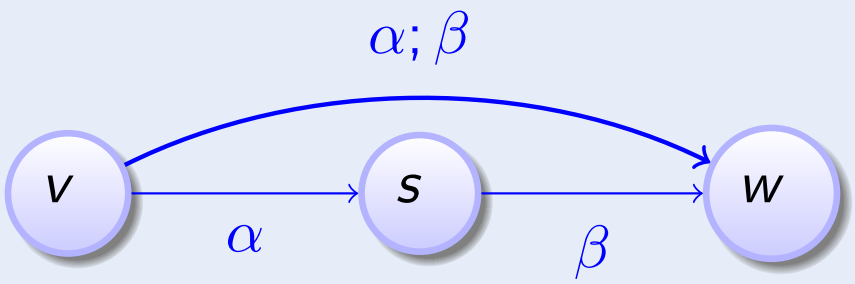
$$v \xrightarrow{\forall i : C \; x(s)' = \theta} w$$



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

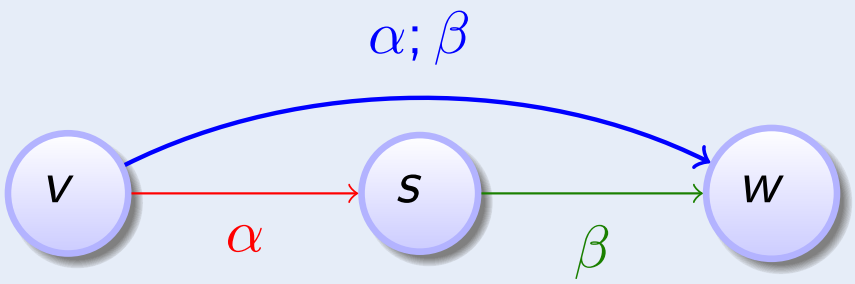
Quantified Differential Dynamic Logic Qd$\mathcal{L}$: Semantics

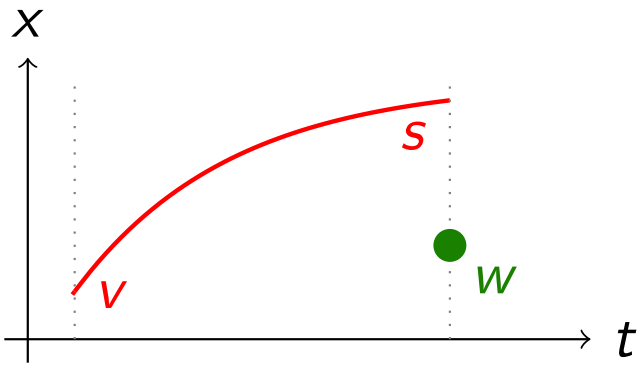
Definition (Quantified hybrid program α: transition semantics)



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

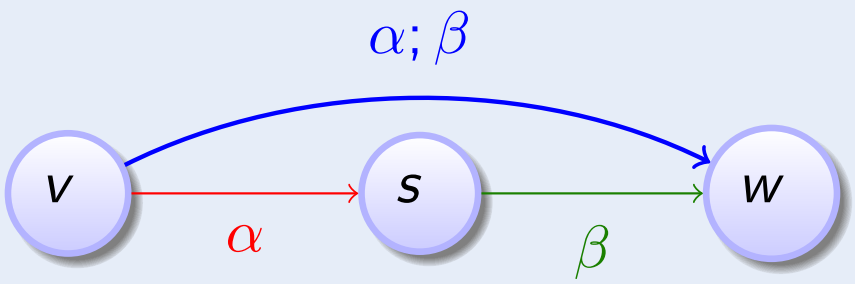
Definition (Quantified hybrid program α: transition semantics)

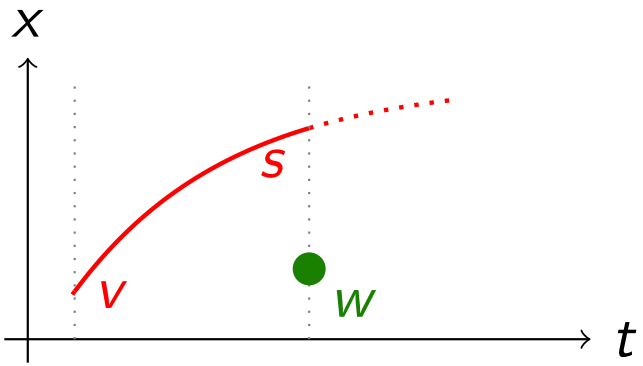




Quantified Differential Dynamic Logic Qd$\mathcal{L}$: Semantics

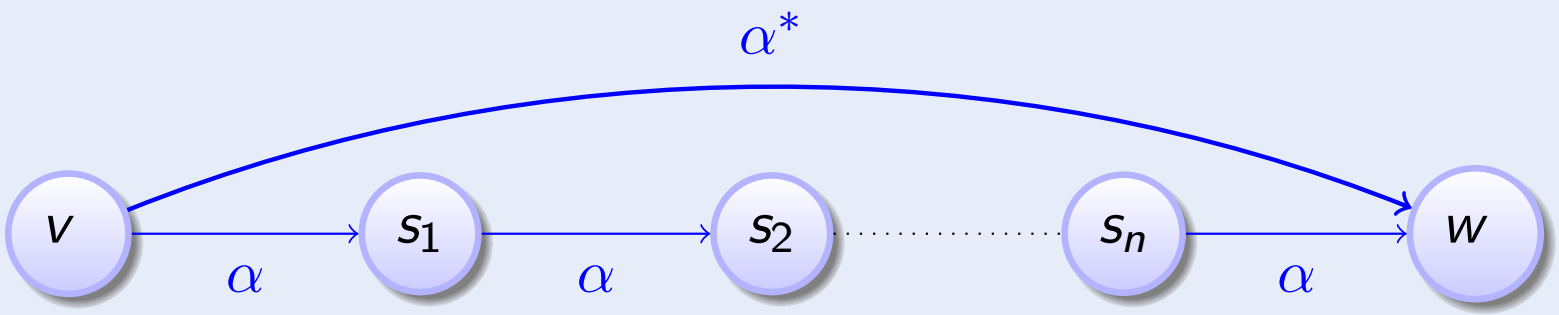
Definition (Quantified hybrid program α: transition semantics)





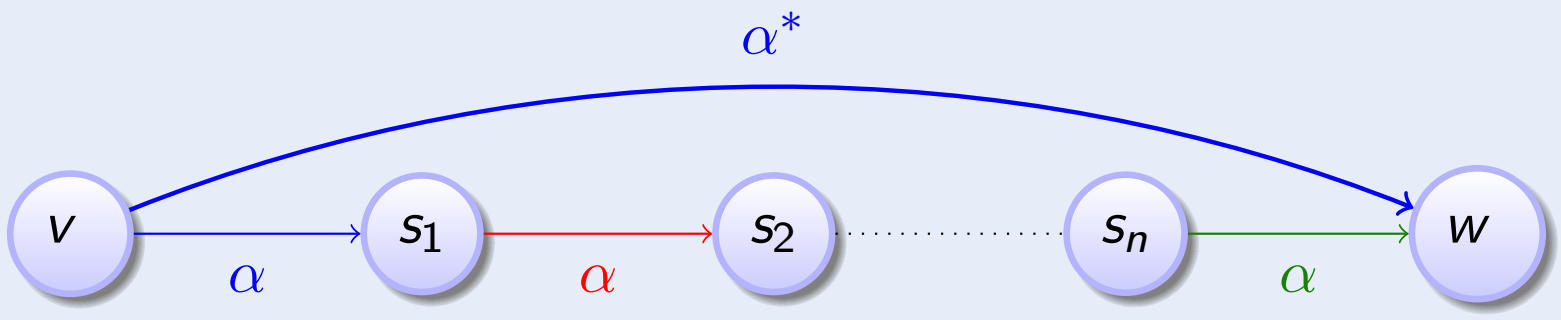
Quantified Differential Dynamic Logic Qd$\mathcal{L}$: Semantics

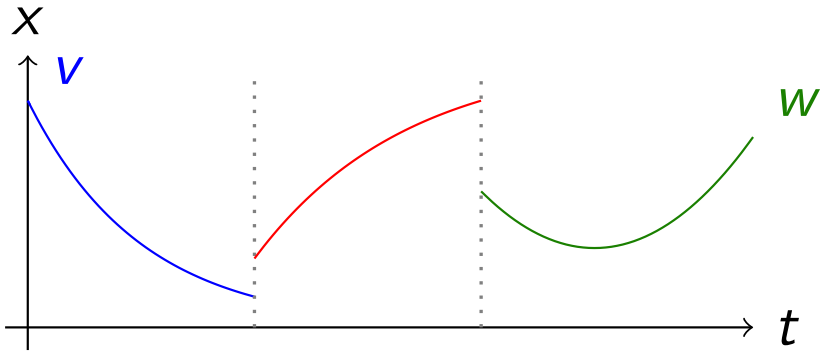
Definition (Quantified hybrid program α: transition semantics)



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

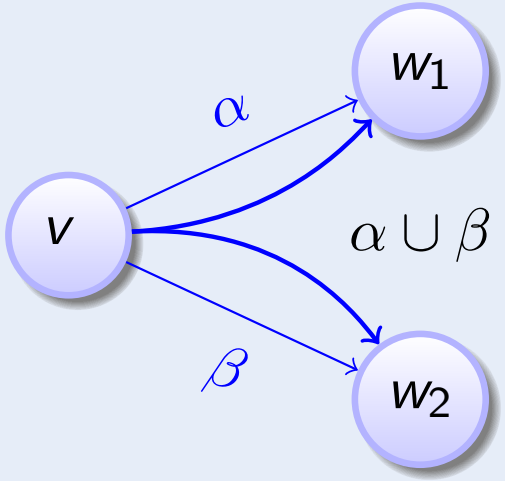
Definition (Quantified hybrid program α: transition semantics)





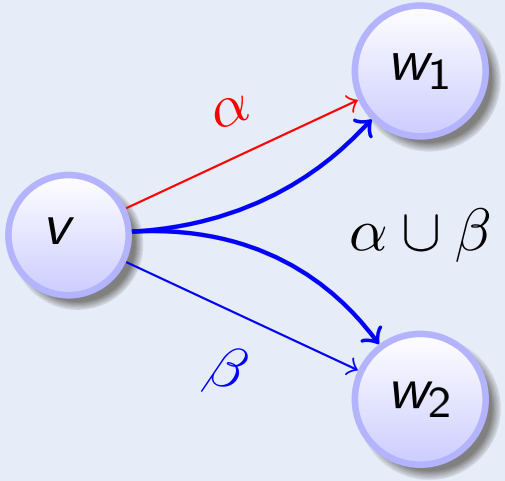
Quantified Differential Dynamic Logic Qd$\mathcal{L}$: Semantics

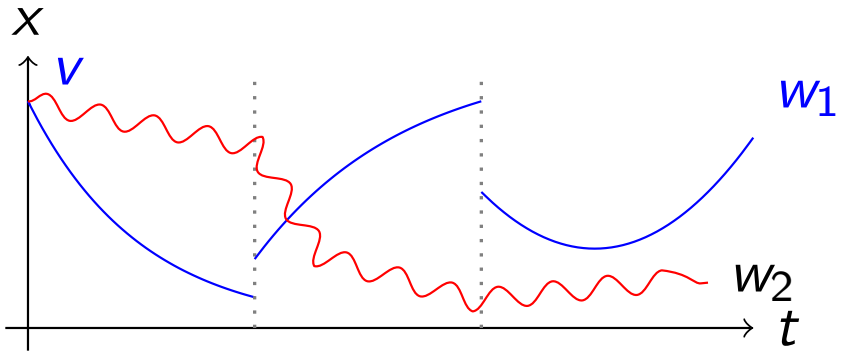
Definition (Quantified hybrid program α: transition semantics)



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

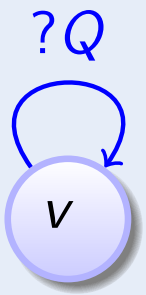
Definition (Quantified hybrid program α: transition semantics)



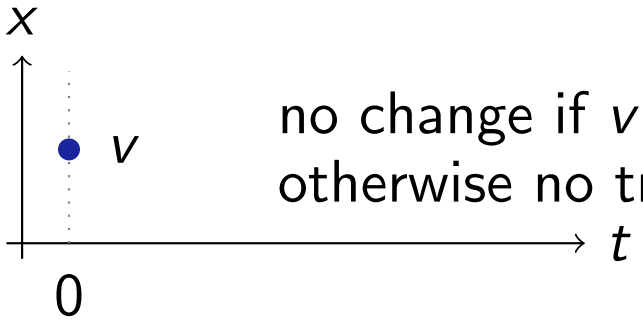


Quantified Differential Dynamic Logic Qd$\mathcal{L}$: Semantics

Definition (Quantified hybrid program α: transition semantics)



if $v \models Q$

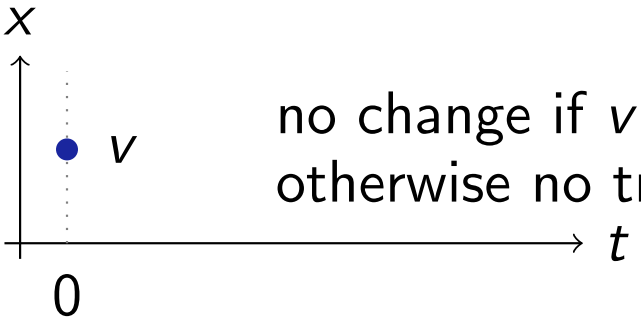


no change if $v \models Q$
otherwise no transition

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

Definition (Quantified hybrid program α: transition semantics)

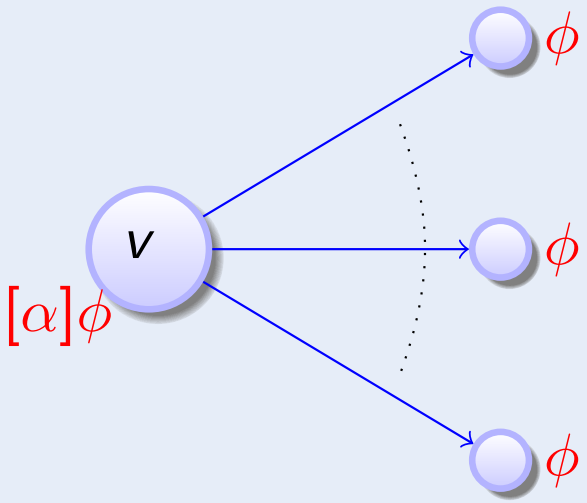
if $v \not\models Q$



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

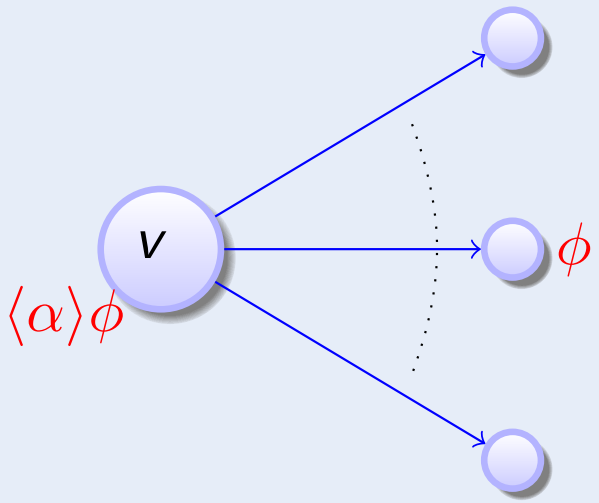
Quantified Differential Dynamic Logic Qd$\mathcal{L}$: Semantics

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



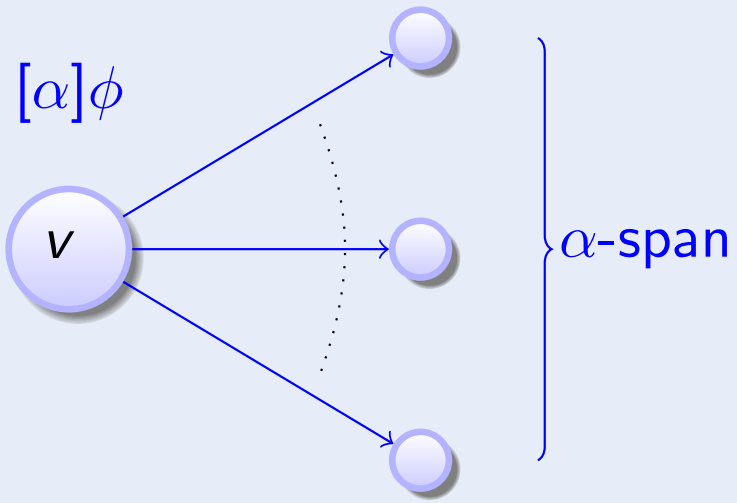
Quantified Differential Dynamic Logic Qd$\mathcal{L}$: Semantics

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



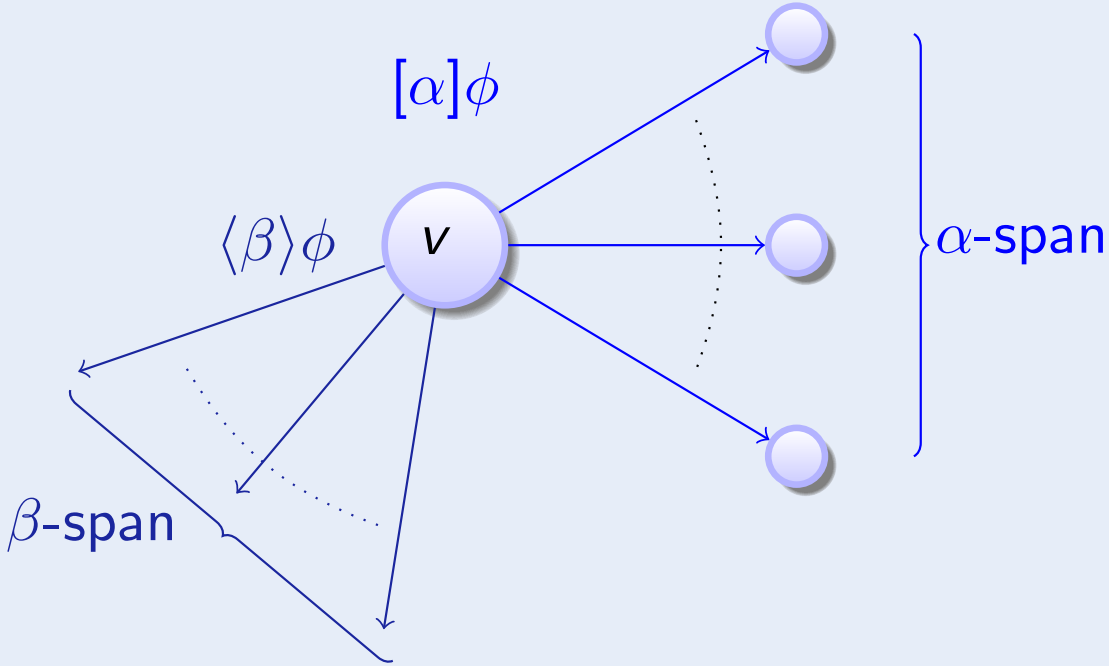
Quantified Differential Dynamic Logic Qd$\mathcal{L}$: Semantics

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



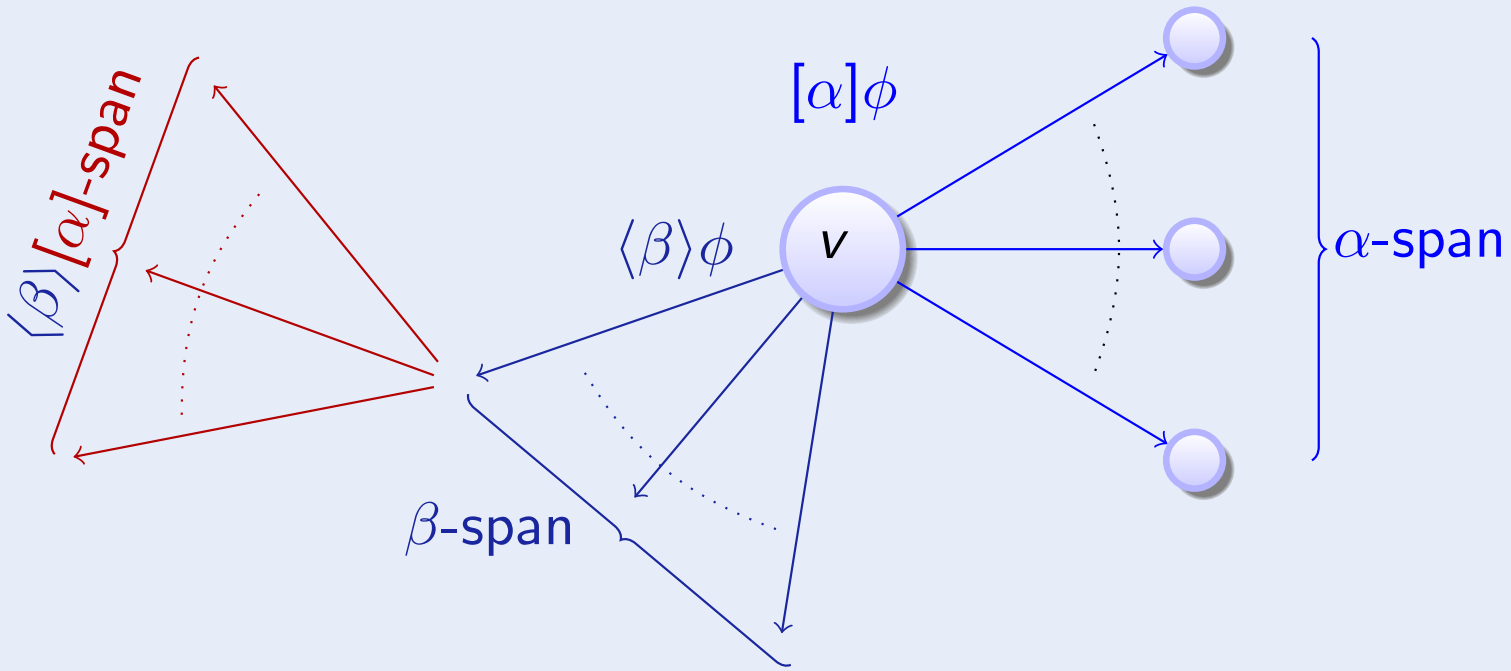
Quantified Differential Dynamic Logic Qd$\mathcal{L}$: Semantics

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



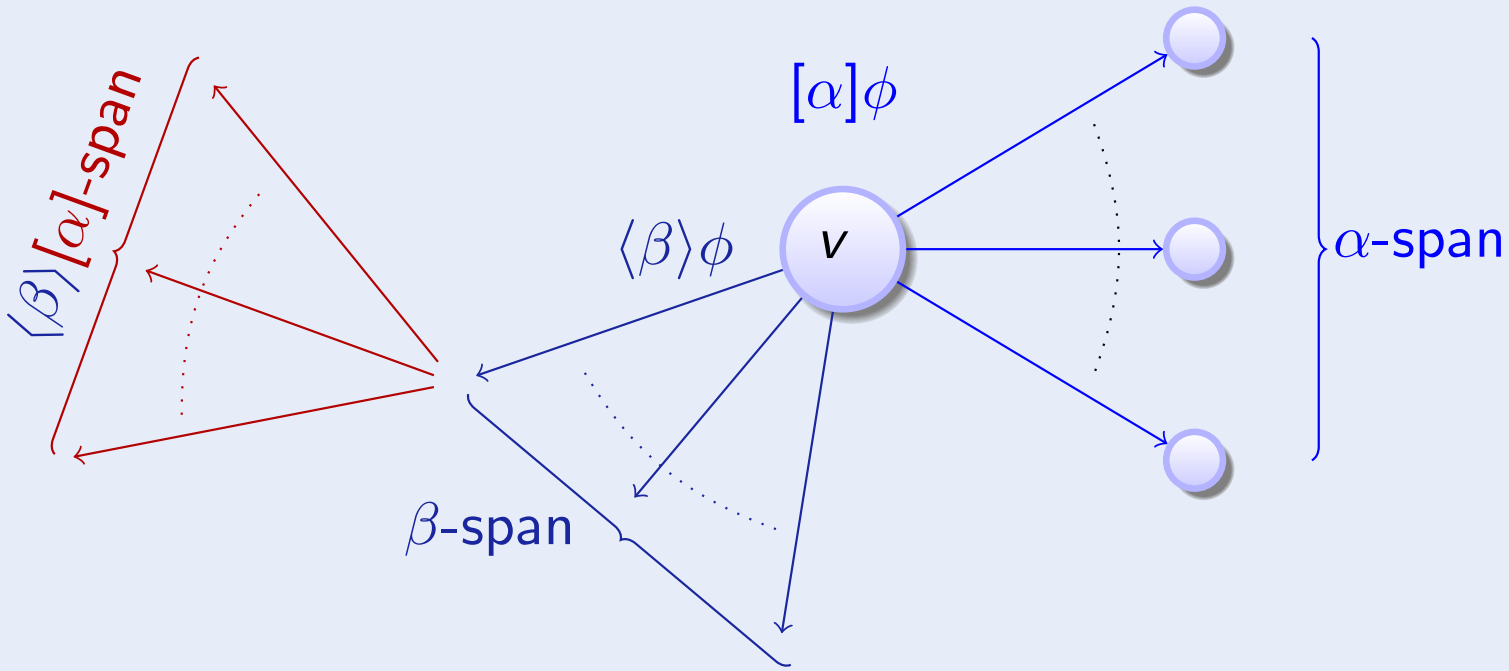
Quantified Differential Dynamic Logic Qd$\mathcal{L}$: Semantics

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



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

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

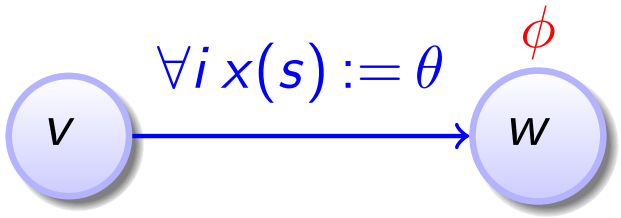


compositional semantics $\Rightarrow$ compositional calculus!

Outline (Verification Approach)

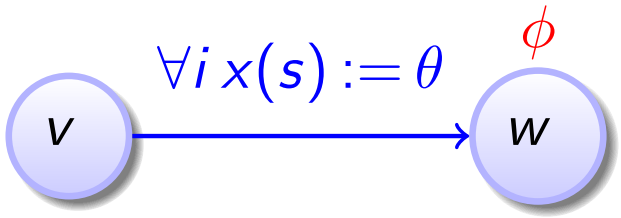
Proof Calculus for Quantified Differential Dynamic Logic

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



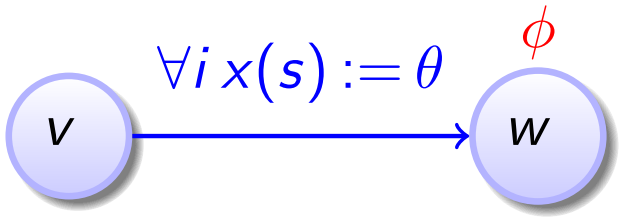
Proof Calculus for Quantified Differential Dynamic Logic

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



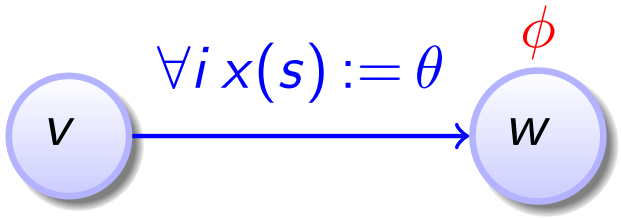
Proof Calculus for Quantified Differential Dynamic Logic

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



Proof Calculus for Quantified Differential Dynamic Logic

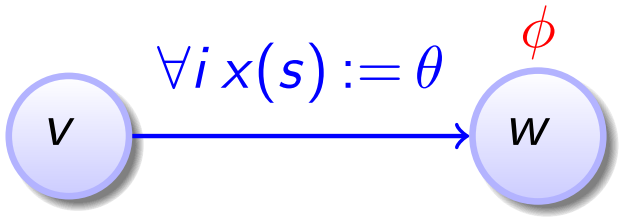
$$\frac{\forall i\,(i = [\forall i\, x(i) := \theta]u \to \phi(\theta))}{\phi([\forall i\, x(i) := \theta]x(u))}$$



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

Proof Calculus for Quantified Differential Dynamic Logic

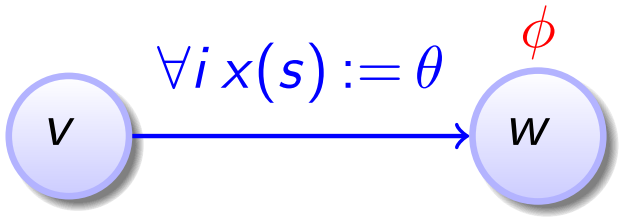
$$\frac{\forall i\,(i = [\forall i\, x(i) := \theta]u \to \phi(\theta))}{\phi([\forall i\, x(i) := \theta]x(u))}$$



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

Proof Calculus for Quantified Differential Dynamic Logic

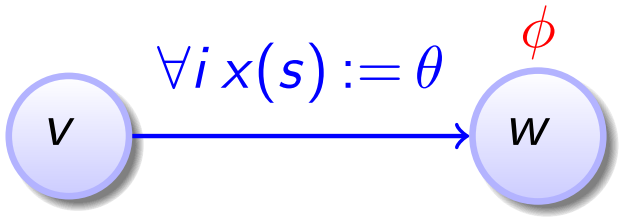
$$\frac{\forall i\,(i = [\forall i\, x(i) := \theta]u \to \phi(\theta))}{\phi([\forall i\, x(i) := \theta]x(u))}$$



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

Proof Calculus for Quantified Differential Dynamic Logic

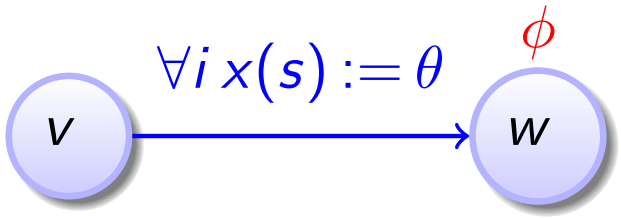
$$\frac{\forall i\,(i = [\forall i\, x(i) := \theta]u \to \phi(\theta))}{\phi([\forall i\, x(i) := \theta]x(u))}$$



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

Proof Calculus for Quantified Differential Dynamic Logic

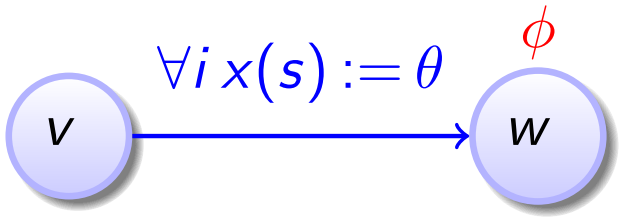
$$\frac{\forall i\,(i = [\forall i\, x(i) := \theta]u \to \phi(\theta))}{\phi([\forall i\, x(i) := \theta]x(u))}$$



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

Proof Calculus for Quantified Differential Dynamic Logic

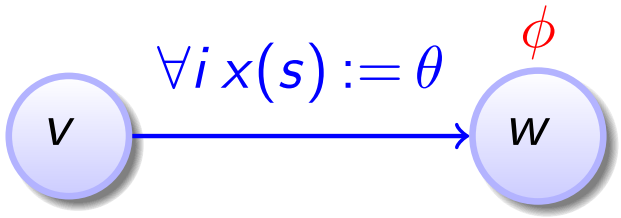
$$\frac{\forall i\,(i = [\forall i\, x(i) := \theta]u \to \phi(\theta))}{\phi([\forall i\, x(i) := \theta]x(u))}$$



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

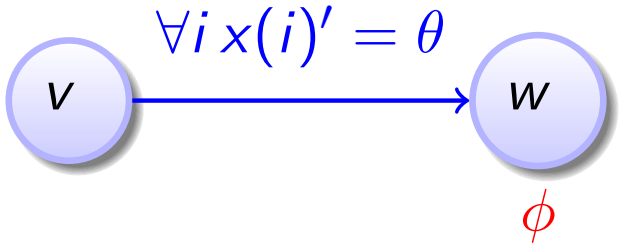
Proof Calculus for Quantified Differential Dynamic Logic

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



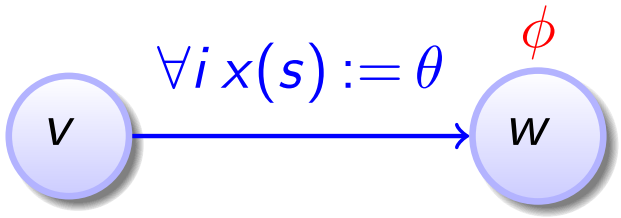
$$\frac{\texttt{if}\ \exists i\, s = [\mathcal{A}]u\ \texttt{then}\ \forall i\,(s = [\mathcal{A}]u \to \phi(\theta))\ \texttt{else}\ \phi(x([\mathcal{A}]u))}{\phi([\underbrace{\forall i\, x(s) := \theta}_{\mathcal{A}}]x(u))}$$

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



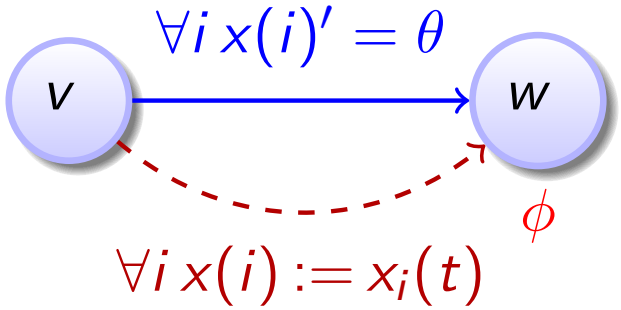
Proof Calculus for Quantified Differential Dynamic Logic

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



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

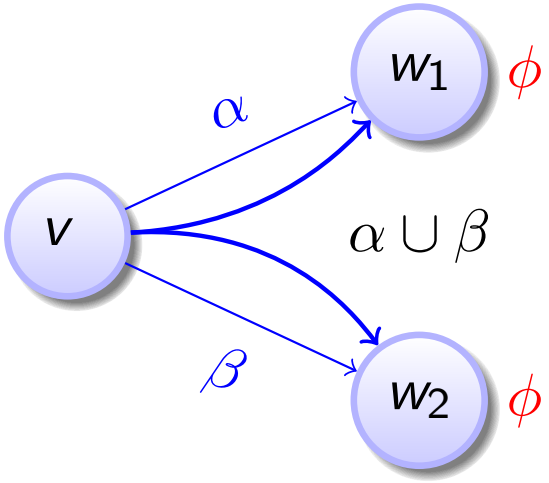
$$\frac{\forall t{\geq}0\,[\forall i\, x(i) := x_i(t)]\phi}{[\forall i\, x(i)' = \theta]\phi}$$



compositional semantics $\Rightarrow$ compositional rules!

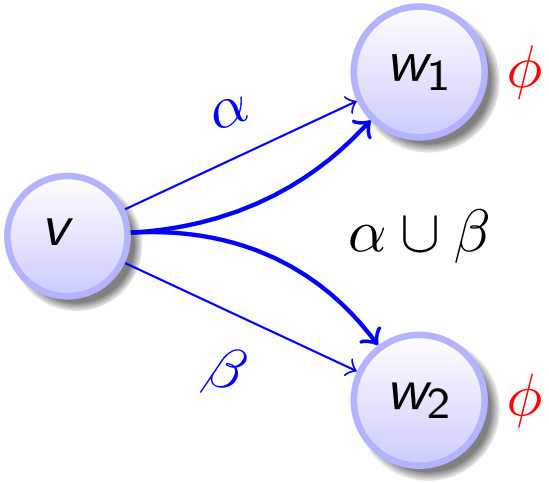
Proof Calculus for Quantified Differential Dynamic Logic

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

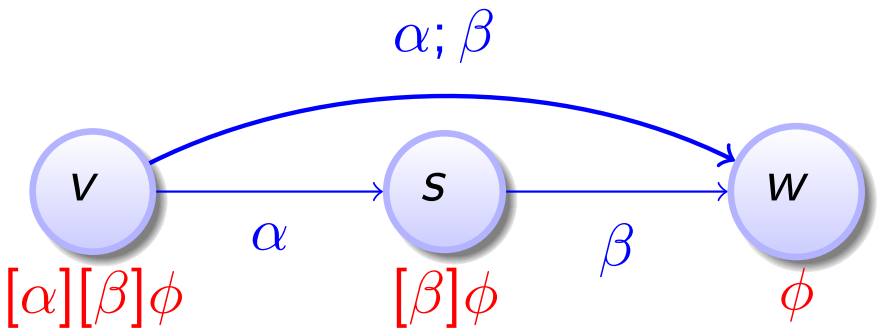


Proof Calculus for Quantified Differential Dynamic Logic

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

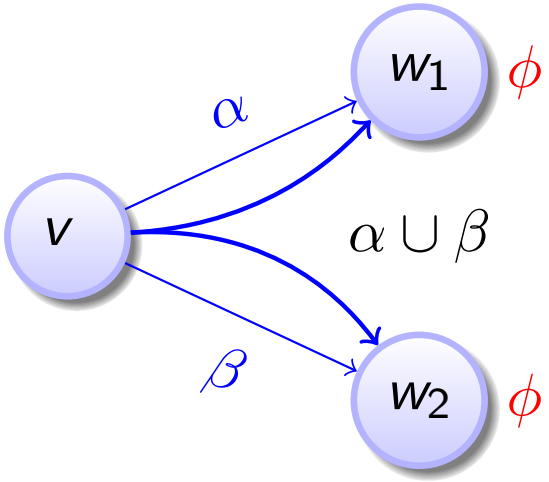


$$\frac{[\alpha][\beta]\phi}{[\alpha;\beta]\phi}$$

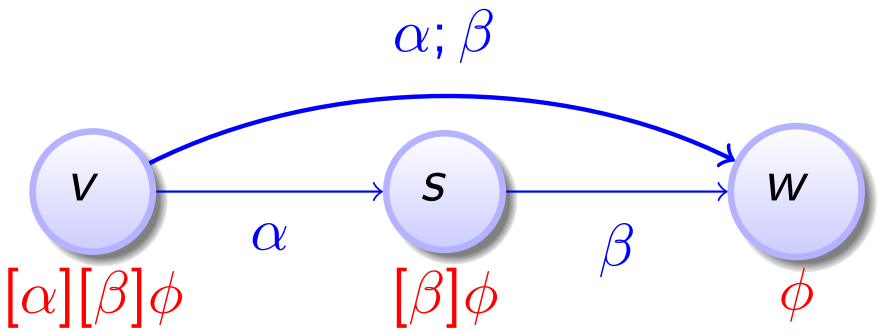


Proof Calculus for Quantified Differential Dynamic Logic

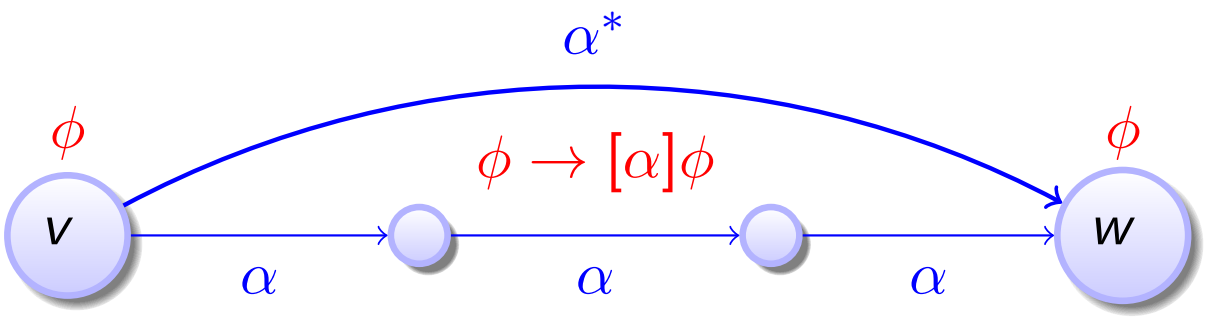
$$\frac{[\alpha]\phi \wedge [\beta]\phi}{[\alpha \cup \beta]\phi}$$



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

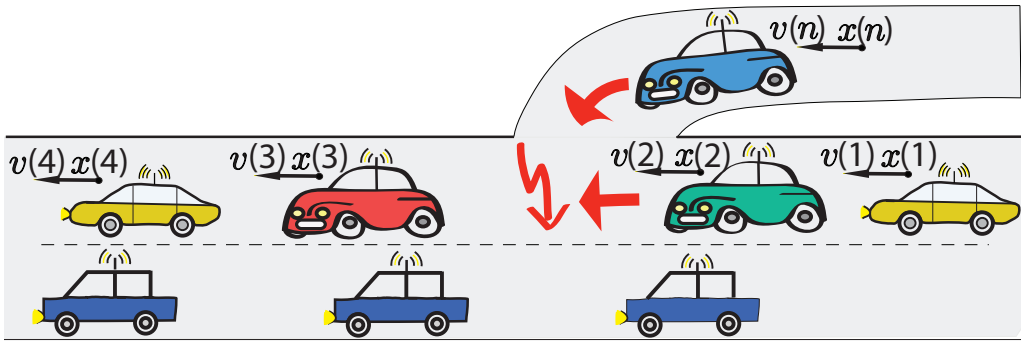


$$\frac{\phi \quad (\phi \to [\alpha]\phi)}{[\alpha^*]\phi}$$



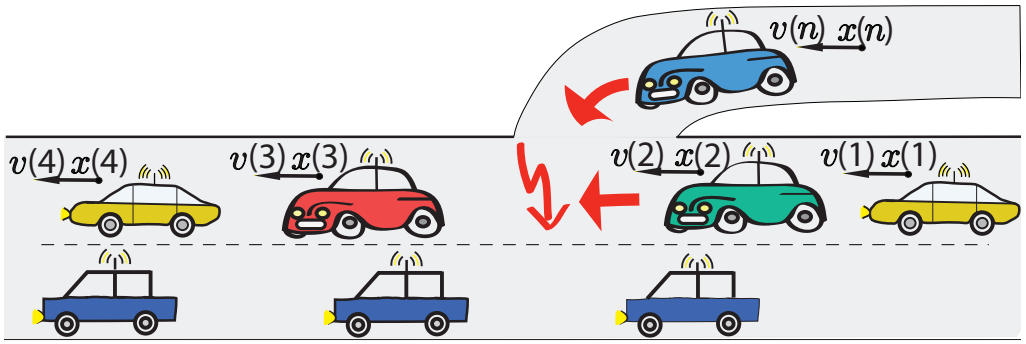
Deduction Modulo with Free Variables & Skolemization

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



Deduction Modulo with Free Variables & Skolemization

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

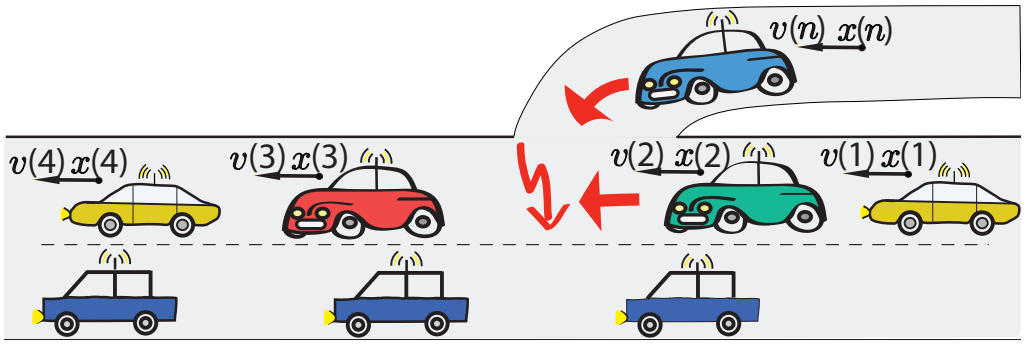


Deduction Modulo with Free Variables & Skolemization

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

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

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



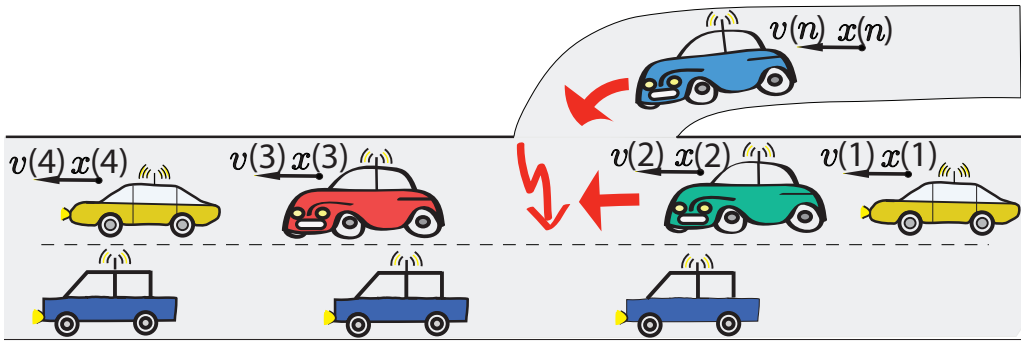
Deduction Modulo with Free Variables & Skolemization

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

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

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

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



Deduction Modulo with Free Variables & Skolemization

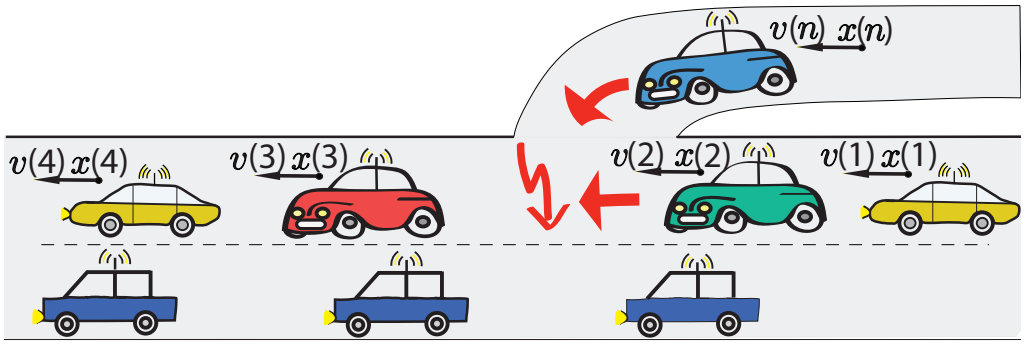
$$\forall i{\neq}j\, x(i){\neq}x(j), s{\geq}0 \rightarrow [\forall i\, x(i) := -\tfrac{b}{2}s^2 + v(i)s + x(i)]\, \forall j{\neq}k\, x(j){\neq}x(k)$$

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

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

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

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



Deduction Modulo with Free Variables & Skolemization

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

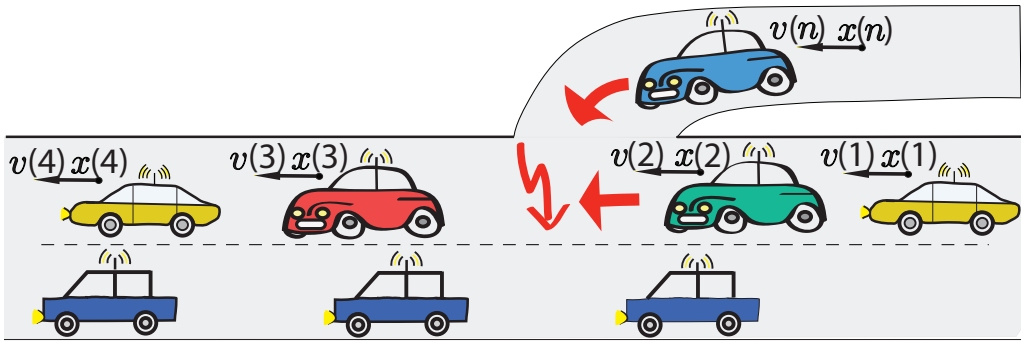
$$\forall i{\neq}j\, x(i){\neq}x(j), s{\geq}0 \rightarrow [\forall i\, x(i) := -\tfrac{b}{2}s^2 + v(i)s + x(i)]\, \forall j{\neq}k\, x(j){\neq}x(k)$$

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

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

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

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



Deduction Modulo with Free Variables & Skolemization

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

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

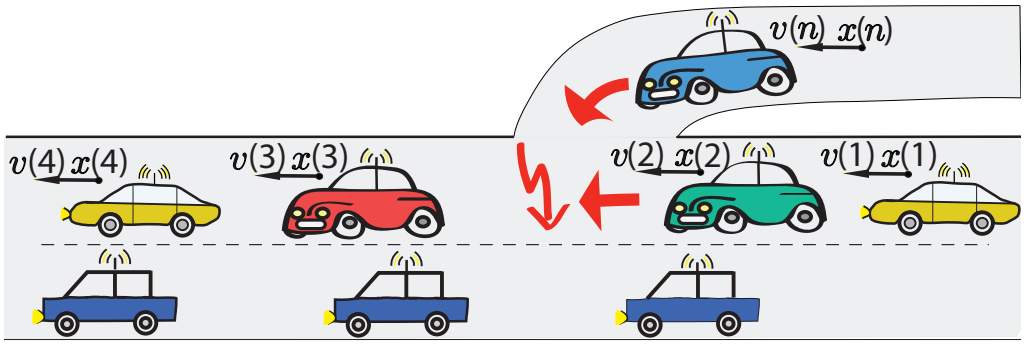
$$\forall i{\neq}j\, x(i){\neq}x(j), s{\geq}0 \to [\forall i\, x(i) := -\tfrac{b}{2}s^2 + v(i)s + x(i)]\, \forall j{\neq}k\, x(j){\neq}x(k)$$

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

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

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

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



Deduction Modulo with Free Variables & Skolemization

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

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

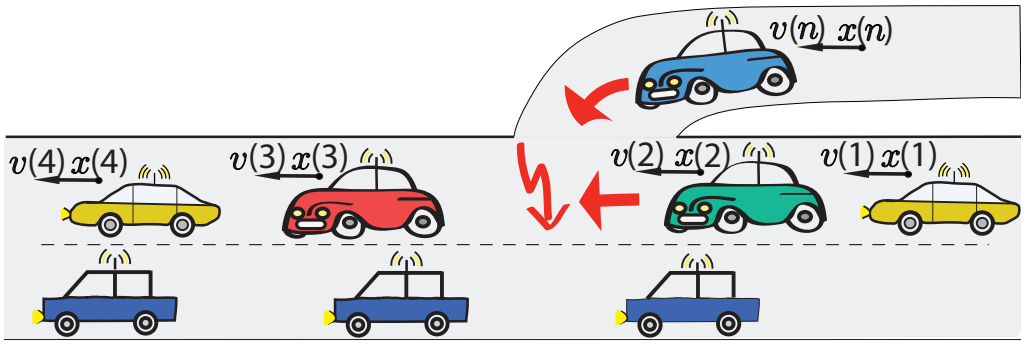
$$\forall i{\neq}j\, x(i){\neq}x(j), s{\geq}0 \rightarrow [\forall i\, x(i) := -\tfrac{b}{2}s^2 + v(i)s + x(i)]\, \forall j{\neq}k\, x(j){\neq}x(k)$$

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

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

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

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



Deduction Modulo with Free Variables & Skolemization

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

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

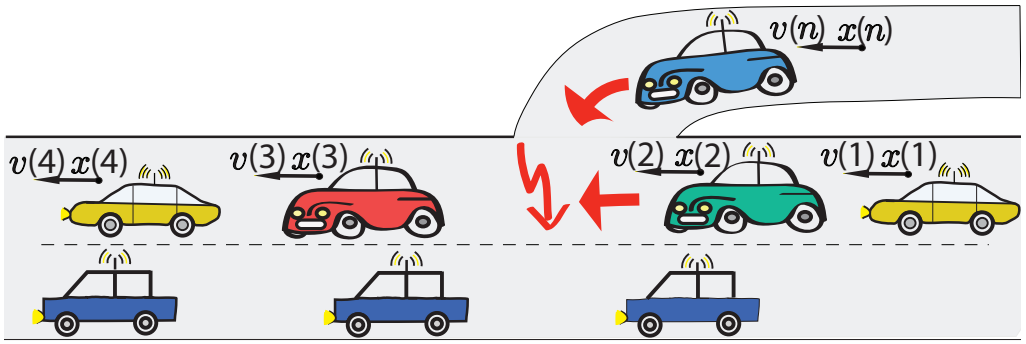
$$\forall i{\neq}j\, x(i){\neq}x(j), s{\geq}0 \rightarrow [\forall i\, x(i) := -\tfrac{b}{2}s^2 + v(i)s + x(i)]\, \forall j{\neq}k\, x(j){\neq}x(k)$$

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

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

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

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



Deduction Modulo with Free Variables & Skolemization

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

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

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

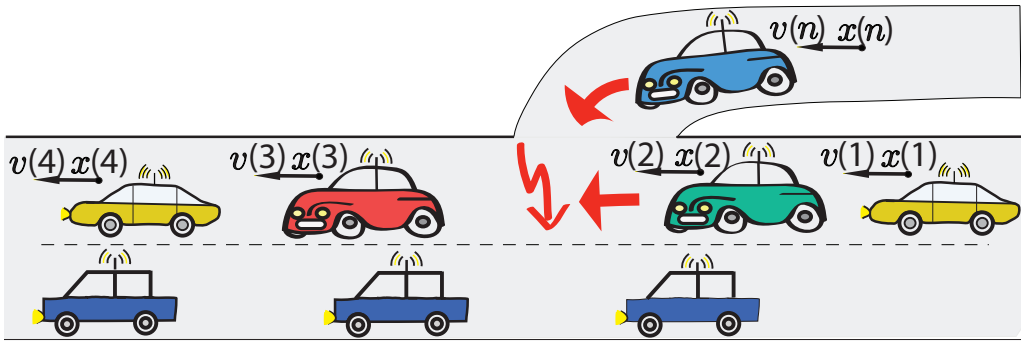
$$\forall i \neq j\; x(i) \neq x(j), s \geq 0 \rightarrow [\forall i\; x(i) := -\tfrac{b}{2}s^2 + v(i)s + x(i)]\,\forall j \neq k\; x(j) \neq x(k)$$

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

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

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

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



Deduction Modulo with Free Variables & Skolemization

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

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

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

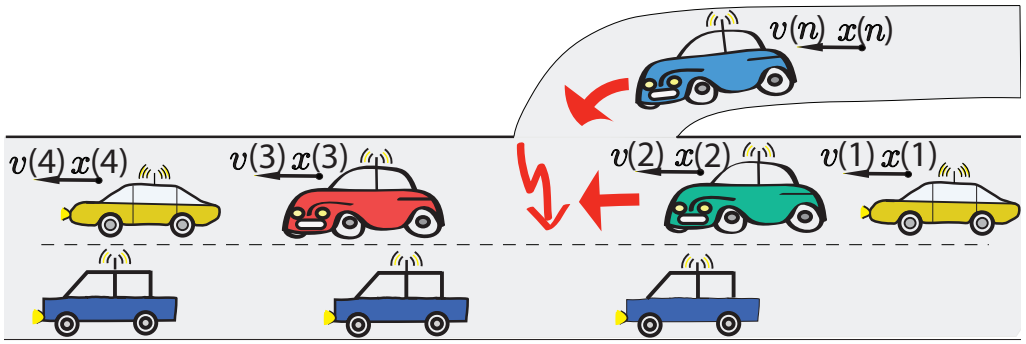
$$\forall i \neq j\; x(i) \neq x(j), s \geq 0 \rightarrow [\forall i\; x(i) := -\tfrac{b}{2}s^2 + v(i)s + x(i)]\; \forall j \neq k\; x(j) \neq x(k)$$

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

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

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

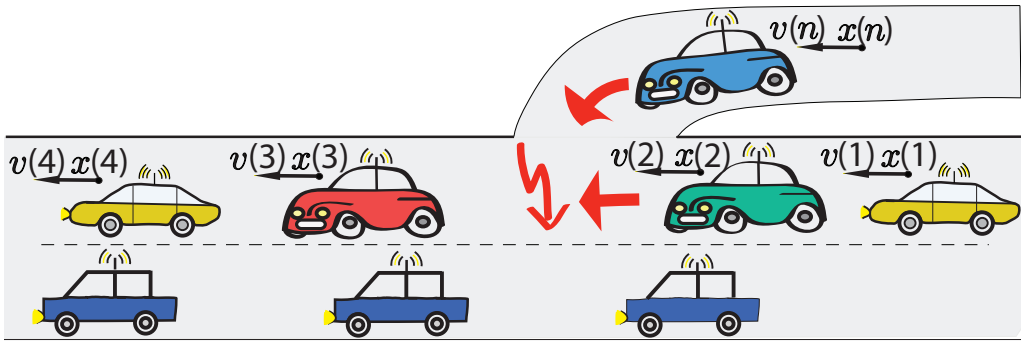
$$\forall i \neq j\; x(i) \neq x(j) \rightarrow [\forall i\; x(i)'' = -b]\; \forall j \neq k\; x(j) \neq x(k)$$



Actual Existence and Creation

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

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

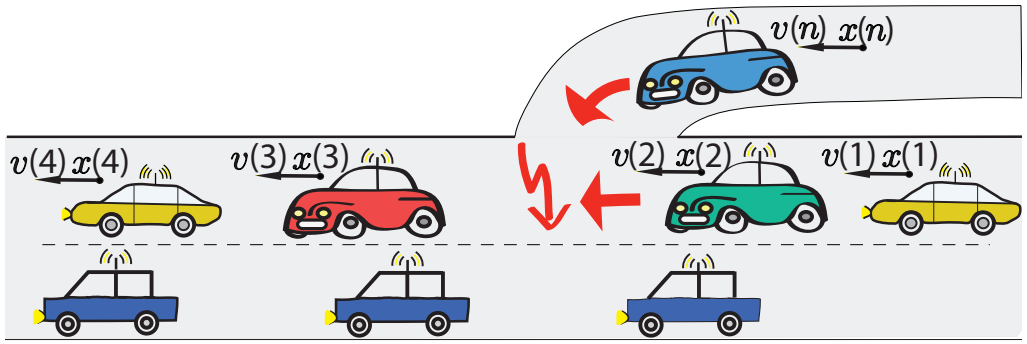


Actual Existence and Creation

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

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

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

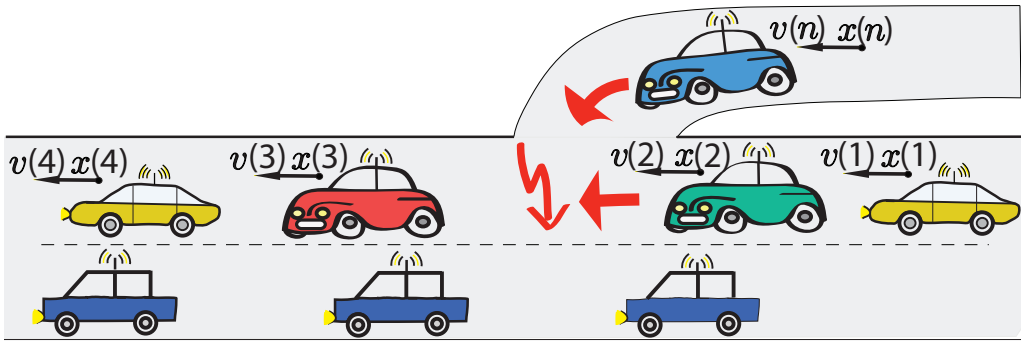


Actual Existence and Creation

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

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

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

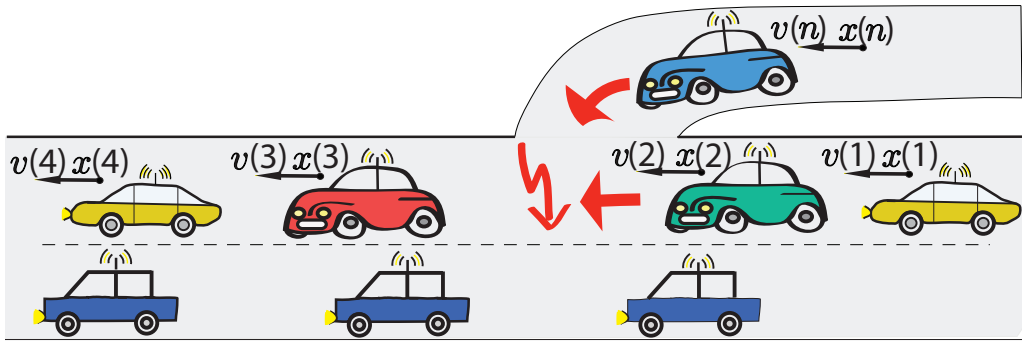


Actual Existence and Creation

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

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

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

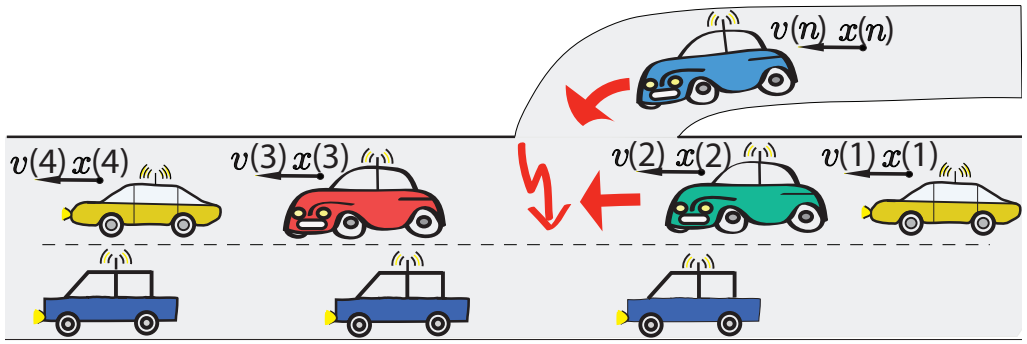


Actual Existence and Creation

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

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

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



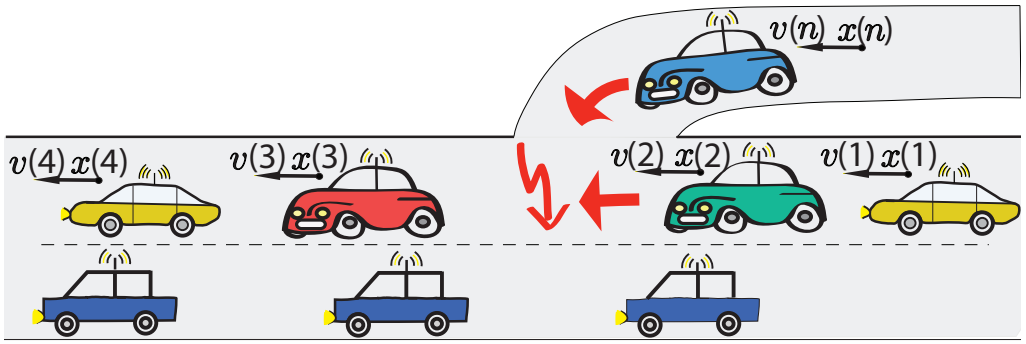
Actual Existence and Creation

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

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

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

$$\begin{aligned} \forall i : C!\;\phi \;&\equiv \\ \forall i : C!\;f(s) := \theta \;&\equiv \\ \forall i : C!\;f(s)' = \theta \;&\equiv \end{aligned}$$



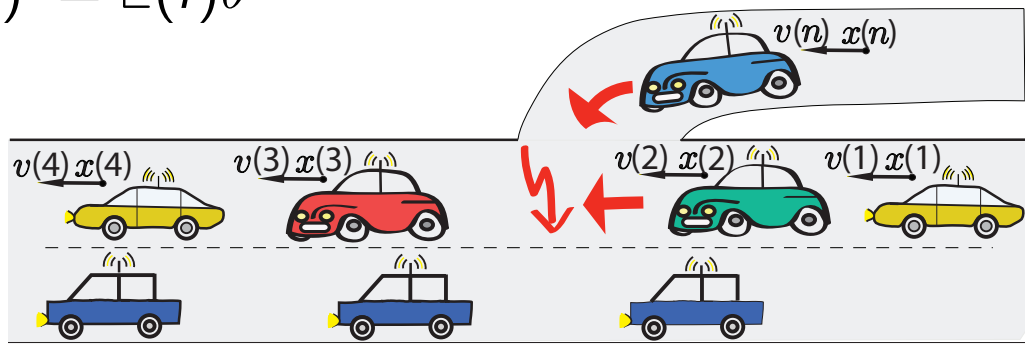
Actual Existence and Creation

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

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

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

$$\begin{aligned} \forall i : C!\ \phi &\equiv \forall i : C\ (\mathbb{E}(i) = 1 \rightarrow \phi) \\ \forall i : C!\ f(s) := \theta &\equiv \forall i : C\ f(s) := (\texttt{if}\ \mathbb{E}(i) = 1\ \texttt{then}\ \theta\ \texttt{else}\ f(s)) \\ \forall i : C!\ f(s)' = \theta &\equiv \forall i : C\ f(s)' = \mathbb{E}(i)\theta \end{aligned}$$



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.

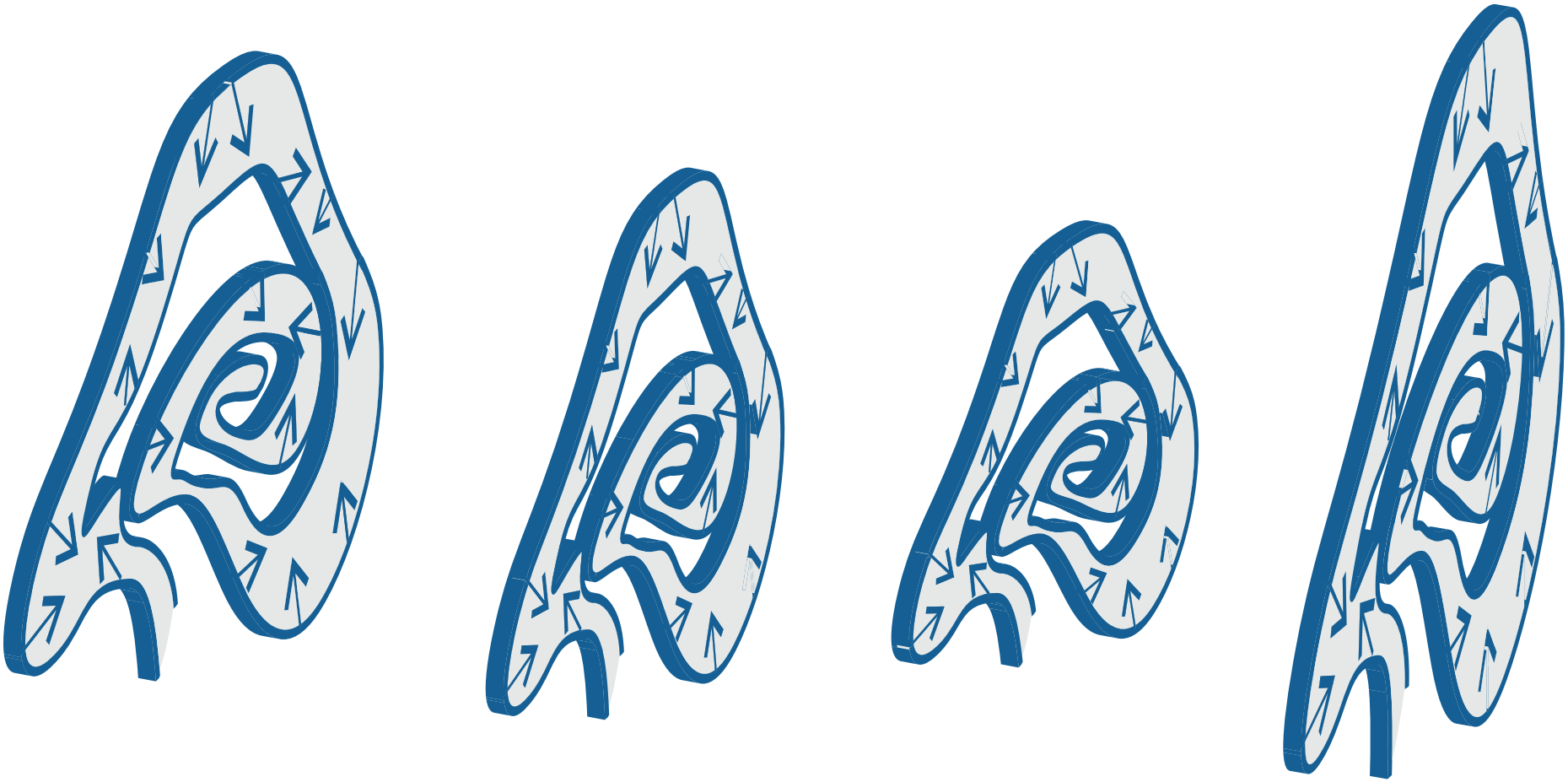
Corollary (Proof-theoretical Alignment)

proving distributed hybrid systems = proving dynamical systems!

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

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

A Simple Proof with Quantified Differential Invariants

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

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

A Simple Proof with Quantified Differential Invariants

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

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

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

A Simple Proof with Quantified Differential Invariants

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

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

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

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

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

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

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

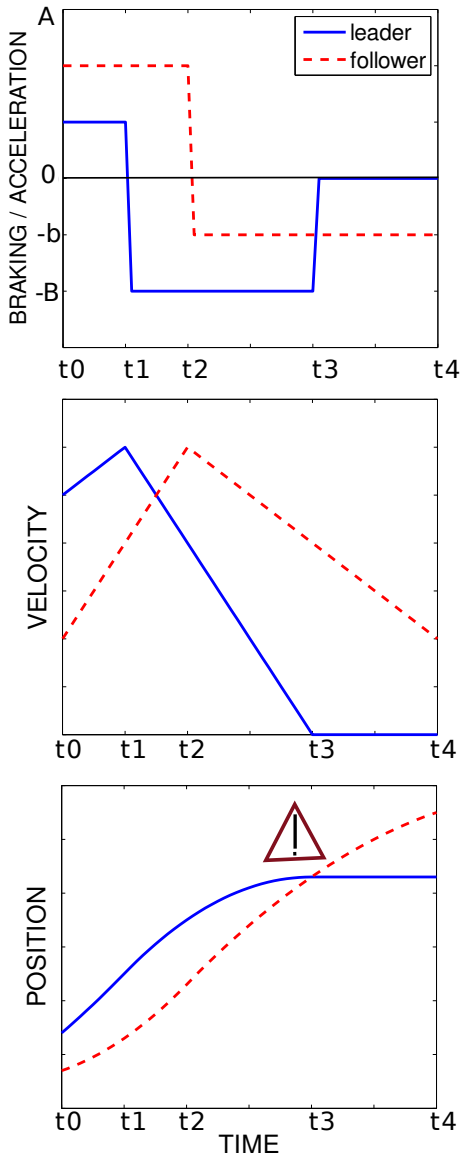
Challenge: Local lane dynamics

- 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:

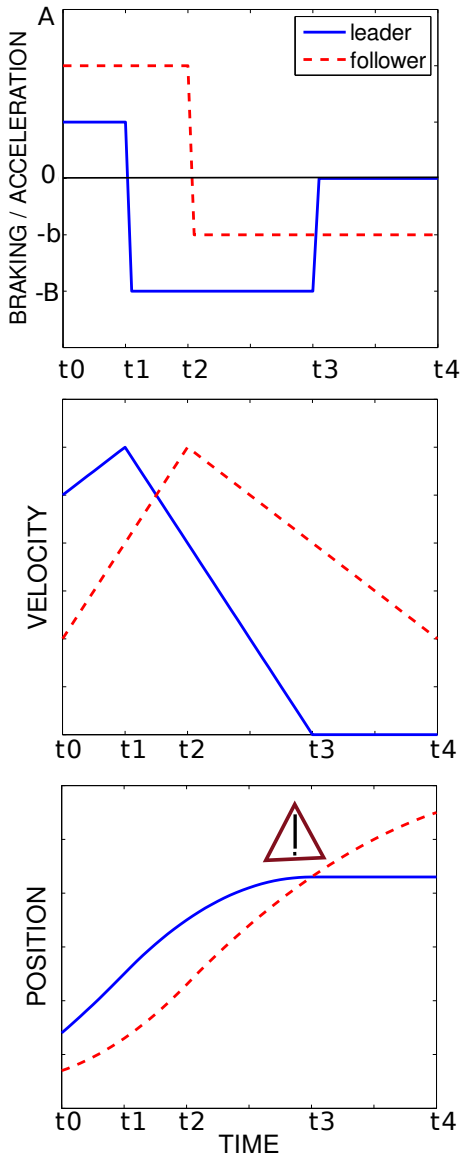


Car Control: Local Lane Control Challenge

Challenge: Local lane dynamics

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

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



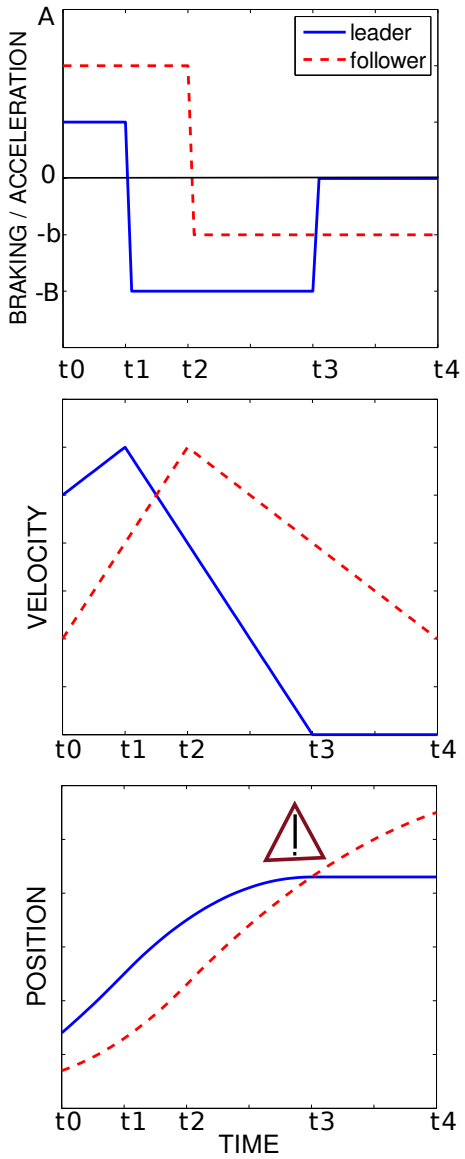
Car Control: Local Lane Control Challenge

Challenge: Local lane dynamics

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

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

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



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

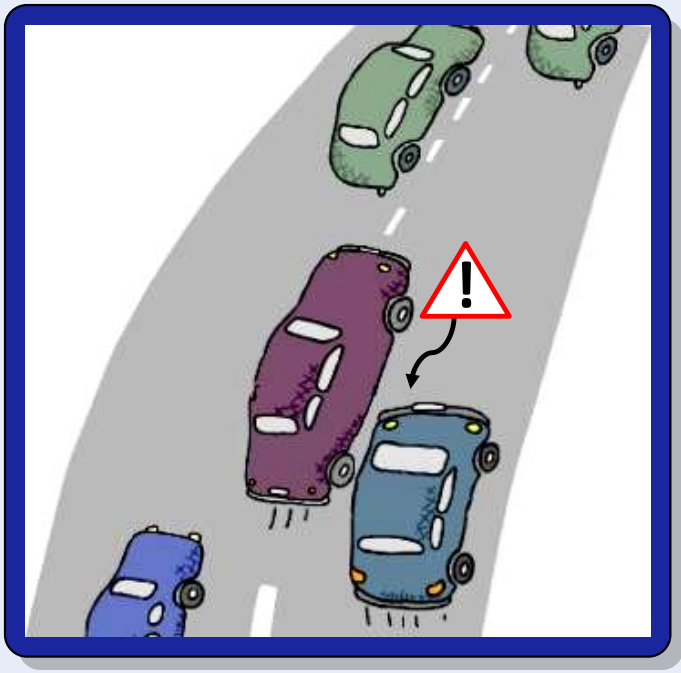
Challenge: Local highway dynamics

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

Car Control: Local Highway Control Challenge

Challenge: Local highway dynamics

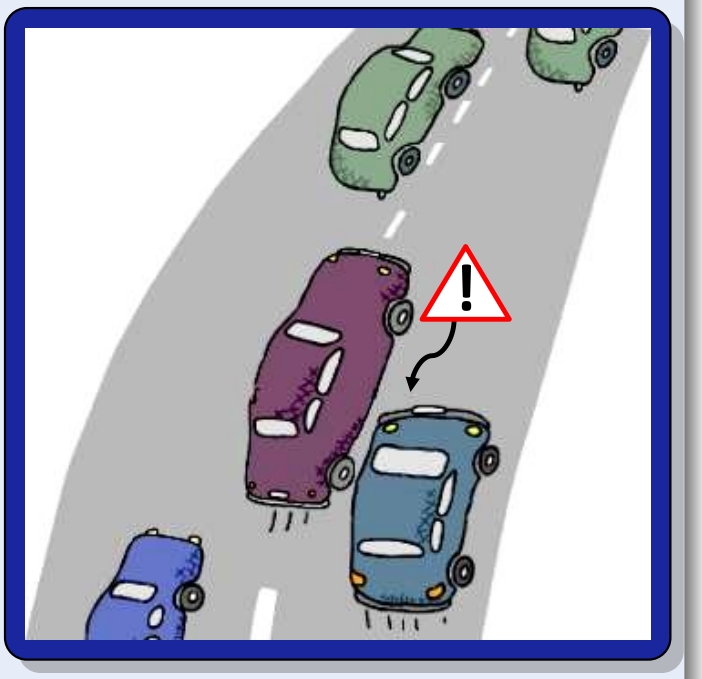
- All controllers for arbitrarily many differential equations respect separation locally on highway.
- For each lane: all controllers for the differential equations respect separation even if cars appear or disappear.



Car Control: Local Highway Control Challenge

Challenge: Local highway dynamics

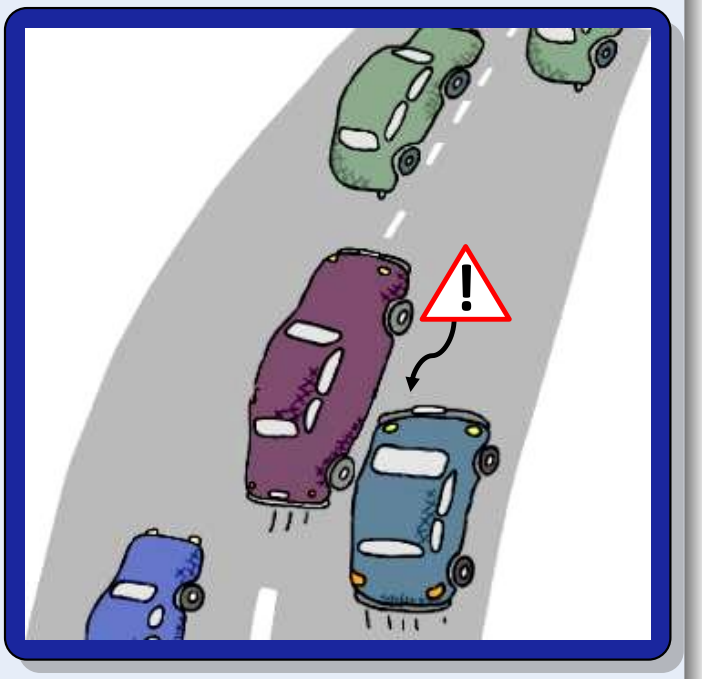
- All controllers for arbitrarily many differential equations respect separation locally on highway.
- For each lane: all controllers for the differential equations respect separation even if cars appear or disappear.
- Each car safe behind all others, even if new cars appear or disappear.



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.



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

Car Control: Global Highway Control Challenge

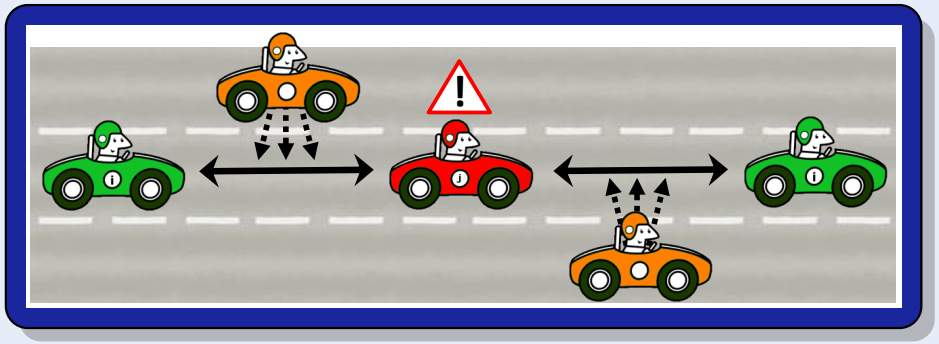
Challenge: Global highway dynamics

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

Car Control: Global Highway Control Challenge

Challenge: Global highway dynamics

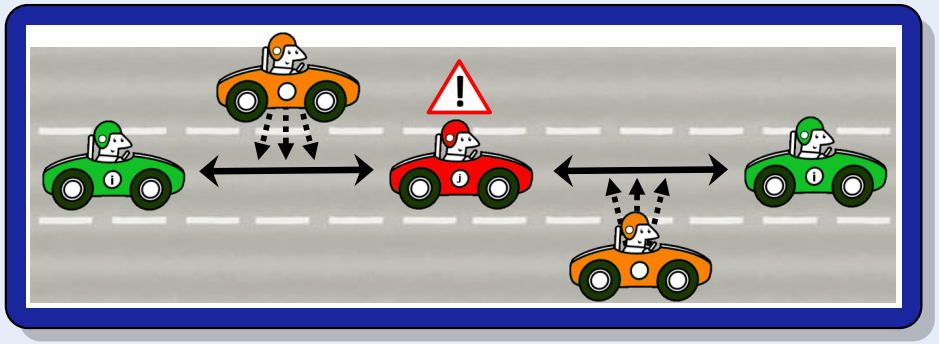
- All controllers for arbitrarily many differential equations respect separation globally on highway.
- All controllers for the differential equations respect separation even if cars switch lanes.



Car Control: Global Highway Control Challenge

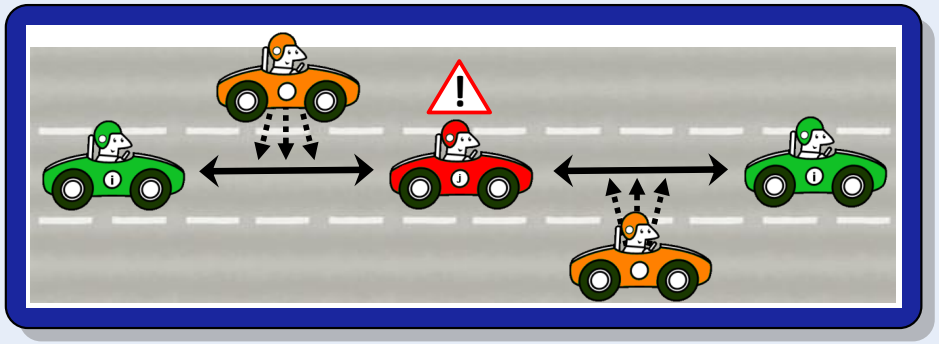
Challenge: Global highway dynamics

- 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.



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.

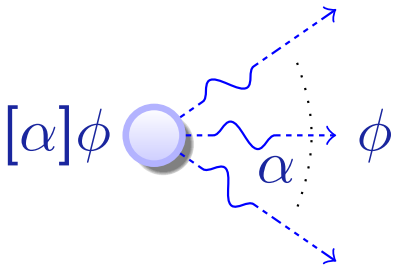
$$[\forall l\,(n := \texttt{new}\ C;\ \forall i\ a(i) := ctrl;\ \forall i\ x(i)'' = a(i))^*]\,\forall l\,\forall i,j\ i \ll j$$

Outline

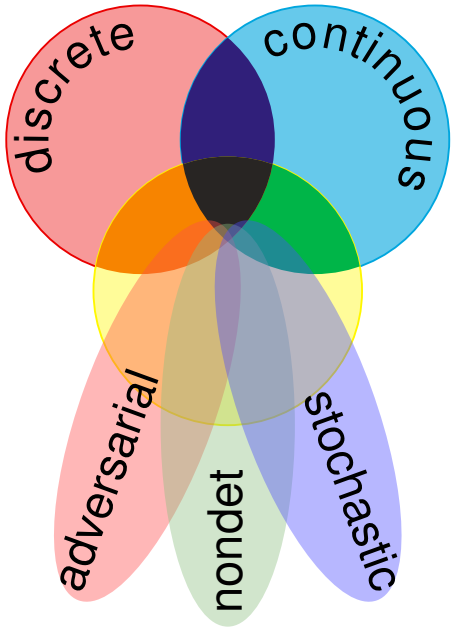
Conclusions

quantified differential dynamic logic

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



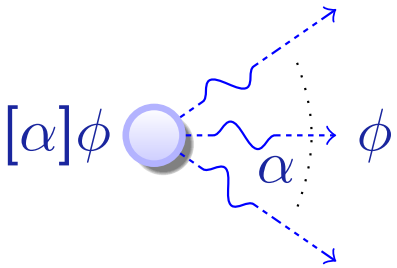
- Distributed hybrid systems everywhere
- System model and semantics
- Logic for distributed hybrid systems
- Compositional proof calculus
- First verification approach
- Sound & complete / diff. eqn.
- Quantified differential invariants
- Distributed car control verified
- Distributed aircraft control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



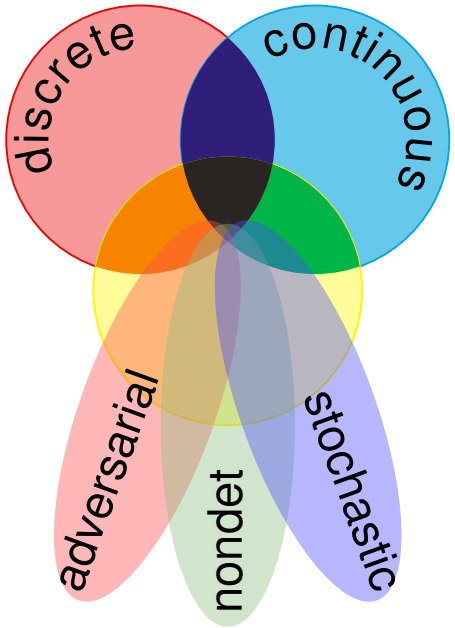
Conclusions

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.
- Quantified differential invariants
- Distributed car control verified
- Distributed aircraft control verified



André Platzer.
A complete axiomatization of quantified differential dynamic logic for distributed hybrid systems.
Log. Meth. Comput. Sci., 8(4):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 [12], 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. Springer, 1995.

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 [12], pages 460–475.

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.