Towards a Hybrid Dynamic Logic for Hybrid Dynamic Systems

André Platzer[1,2]

[1]Carnegie Mellon University, Pittsburgh, PA, USA

[2]University of Oldenburg, Department of Computing Science, Germany
aplatzer@cs.cmu.edu

LICS International Workshop on Hybrid Logic 2006

Towards a Hybrid Dynamic Logic for Hybrid Dynamic Systems

André Platzer[1,2]

[1]Carnegie Mellon University, Pittsburgh, PA, USA

[2]University of Oldenburg, Department of Computing Science, Germany
aplatzer@cs.cmu.edu

LICS International Workshop on Hybrid Logic 2006

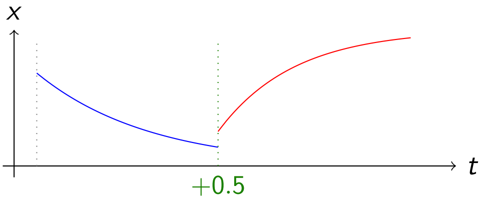
Hybrid Dynamic Systems

Hybrid Dynamic Logic

Logic with state-references and program-modalities

Hybrid Dynamic Systems

Hybrid dynamic systems are subject to both continuous evolution along differential equations and discrete change.



Hybrid Dynamic Systems

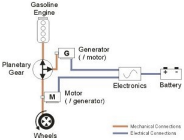
Hybrid Dynamic Systems

Hybrid dynamic systems are subject to both continuous evolution along differential equations and discrete change.

Example (Safety-Critical)

- Car / train / aircraft / chemical process / artificial pancreas
- discrete: digital controller of plant
- continuous: physical model of plant

Hybrid Dynamic Systems: Verification



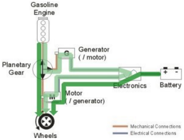
Hybrid Dynamic Systems

Hybrid dynamic systems are subject to both continuous evolution along differential equations and discrete change.

Challenges (Compositional Verification)

1. Verify intricate dynamics in isolation
2. Integrability of local correctness

Hybrid Dynamic Systems: Verification



Hybrid Dynamic Systems

Hybrid dynamic systems are subject to both continuous evolution along differential equations and discrete change.

Challenges (Compositional Verification)

1. Verify intricate dynamics in isolation
2. Integrability of local correctness
   1. state-based reasoning: (transition to abstract state i)
   2. introspection: (statement about other state $@_i\phi$)

Outline

1. Motivation
2. The Logic $d\mathcal{L}_h$
   - Syntax
   - Semantics
   - Compositional Introspection
3. The $d\mathcal{L}_h$ Calculus
   - Sequent Calculus
   - State-based Reasoning
   - Soundness & Co
4. Conclusions & Future Work

Outline

1. Motivation
2. The Logic $d\mathcal{L}_h$
   - Syntax
   - Semantics
   - Compositional Introspection
3. The $d\mathcal{L}_h$ Calculus
   - Sequent Calculus
   - State-based Reasoning
   - Soundness & Co
4. Conclusions & Future Work

The Logic $d\mathcal{L}_h$: Syntax

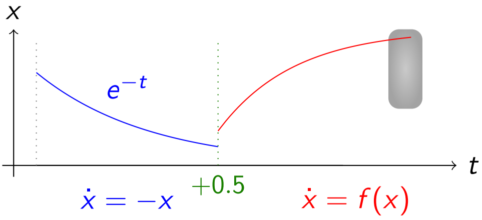
$$d\mathcal{L}_h \text{ formulas} = \text{first-order logic} + \underbrace{\text{dynamic logic}}_{[\alpha]\phi,\ \ \langle\alpha\rangle\phi} + \text{hybrid logic}$$

Definition (System actions α)

$\dot{x} = f(x)$	(continuous evolution)
$x := \theta$	(discrete mode switch)
$\phi?$	(conditional execution)
$\alpha;\gamma$	(seq. composition)
$\alpha \cup \gamma$	(nondet. choice)
α^*	(nondet. repetition)

▸ Details

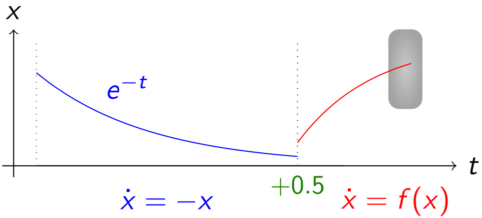
$d\mathcal{L}_h$ Semantics: Hybrid System Evolution



$$x > 1 \quad \rightarrow \quad \langle \dot{x} = -x; x := x + 0.5; \dot{x} = f(x) \rangle \text{ safe}$$

Details

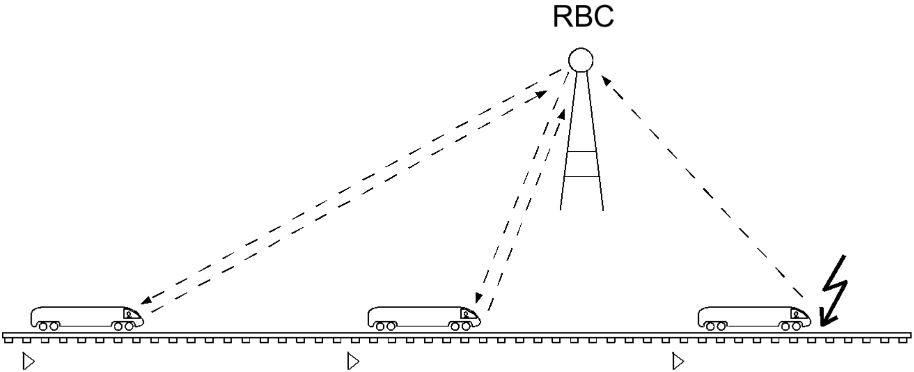
$d\mathcal{L}_h$ Semantics: Hybrid System Evolution



$$x > 1 \quad \rightarrow \quad \langle \dot{x} = -x; x := x + 0.5; \dot{x} = f(x) \rangle \text{ safe}$$

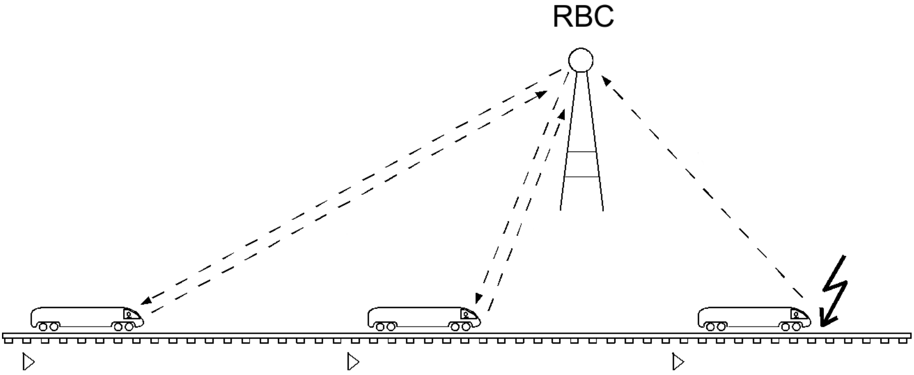
Details

Compositional Introspection in ETCS Braking



$$[\text{poll-sensor};\ a := \text{accel-sys};\quad \ddot{z} = a](z \geq m \rightarrow @_i slope)$$

Compositional Introspection in ETCS Braking



$$[\text{poll-sensor};\ a := \text{accel-sys};\ \color{red}{i?};\ \ddot{z} = a](z \geq m \rightarrow @_{\color{red}{i}}\,slope)$$

Outline

Sequent Calculus (excerpt)

$$\text{(R1)} \quad \frac{@_i\langle x := \theta\rangle j \vdash @_i F_x^\theta}{@_i\langle x := \theta\rangle j \vdash @_j F}$$

$$\text{(R2)} \quad \frac{@_i\langle\alpha\rangle a, @_a\phi \vdash}{@_i\langle\alpha\rangle\phi \vdash}$$

$$\text{(R3)} \quad \frac{@_i\exists t{\geq}0\,\langle x := y_x(t)\rangle\phi \vdash}{@_i\langle\dot{x} = f(x)\rangle\phi \vdash}$$

where y_x solution of IVP $\begin{bmatrix} \dot{x} = & f(x) \\ x(0) = & x \end{bmatrix}$

Priority: R3>R2>R1

State-based Reasoning for Compositional Verification

$$
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{*}{@_t\langle a := \text{-}b\rangle r, @_t\langle \ddot{z} = \text{-}b\rangle cr \vdash @_t\langle \ddot{z} = \text{-}b\rangle z \geq m}
}{@_t\langle a := \text{-}b\rangle r, @_r\langle \ddot{z} = a\rangle cr \vdash @_t\langle \ddot{z} = \text{-}b\rangle z \geq m}
\quad
\dfrac{\dots}{@_t\langle c_2?; \dots\rangle r \vdash \dots}
}{@_t(\langle a := \text{-}b\rangle r \lor \langle c_2?; a := 0.1\rangle r), @_r\langle \ddot{z} = a\rangle cr \vdash @_t\langle \ddot{z} = \text{-}b\rangle z \geq m}
}{@_t\langle a := \text{-}b \cup (c_2?; a := 0.1)\rangle r, @_r\langle \ddot{z} = a\rangle cr \vdash @_t\langle \ddot{z} = \text{-}b\rangle z \geq m}
}{@_t\langle a := \text{-}b \cup (c_2?; a := 0.1)\rangle\langle \ddot{z} = a\rangle cr \vdash @_t\langle \ddot{z} = \text{-}b\rangle z \geq m}
}{@_t\langle \text{accel}\rangle cr \vdash @_t\langle \ddot{z} = \text{-}b\rangle z \geq m}
}{@_t\neg\langle \ddot{z} = \text{-}b\rangle z \geq m, @_s\langle \text{tctl}\rangle t, @_t\langle \text{accel}\rangle cr \vdash}
}{@_s[\text{tctl}]\neg\langle \ddot{z} = \text{-}b\rangle z \geq m, @_s\langle \text{tctl}\rangle t, @_t\langle \text{accel}\rangle cr \vdash}
}{@_s[\text{tctl}]\neg\langle \ddot{z} = \text{-}b\rangle z \geq m, @_s\langle \text{tctl}\rangle\langle \text{accel}\rangle cr \vdash}
}{@_s[\text{tctl}]\neg\langle \ddot{z} = \text{-}b\rangle z \geq m, @_s\langle \text{tctl}; \text{accel}\rangle cr \vdash}
}{@_s[\text{tctl}]\neg\langle \ddot{z} = \text{-}b\rangle z \geq m \vdash @_s\neg\langle \text{tctl}; \text{accel}\rangle cr}
$$

Abbreviations: $c_2 \equiv (m - z \geq 2e)$ and $\text{accel} \equiv \big(a := \text{-}b \cup (c_2?; a := 0.1)\big); \ddot{z} = a$

State-based Reasoning for Compositional Verification

$$*$$

$$@_t\langle a := \text{-}b\rangle r, @_t\langle \ddot{z} = \text{-}b\rangle cr \vdash @_t\langle \ddot{z} = \text{-}b\rangle z \geq m$$

$$@_t\langle a := \text{-}b\rangle r, @_r\langle \ddot{z} = a\rangle cr \vdash @_t\langle \ddot{z} = \text{-}b\rangle z \geq m \qquad \frac{\ldots}{@_t\langle c_2?; \ldots\rangle r \vdash \ldots}$$

$$@_t(\langle a := \text{-}b\rangle r \vee \langle c_2?; a := 0.1\rangle r), @_r\langle \ddot{z} = a\rangle cr \vdash @_t\langle \ddot{z} = \text{-}b\rangle z \geq m$$

$$@_t\langle a := \text{-}b \cup (c_2?; a := 0.1)\rangle r, @_r\langle \ddot{z} = a\rangle cr \vdash @_t\langle \ddot{z} = \text{-}b\rangle z \geq m$$

$$@_t\langle a := \text{-}b \cup (c_2?; a := 0.1)\rangle\langle \ddot{z} = a\rangle cr \vdash @_t\langle \ddot{z} = \text{-}b\rangle z \geq m$$

$$@_t\langle \text{accel}\rangle cr \vdash @_t\langle \ddot{z} = \text{-}b\rangle z \geq m$$

$$@_t\neg\langle \ddot{z} = \text{-}b\rangle z \geq m, @_s\langle \text{tctl}\rangle t, @_t\langle \text{accel}\rangle cr \vdash$$

$$@_s[\text{tctl}]\neg\langle \ddot{z} = \text{-}b\rangle z \geq m, @_s\langle \text{tctl}\rangle t, @_t\langle \text{accel}\rangle cr \vdash$$

$$@_s[\text{tctl}]\neg\langle \ddot{z} = \text{-}b\rangle z \geq m, @_s\langle \text{tctl}\rangle\langle \text{accel}\rangle cr \vdash$$

$$@_s[\text{tctl}]\neg\langle \ddot{z} = \text{-}b\rangle z \geq m, @_s\langle \text{tctl}; \text{accel}\rangle cr \vdash$$

$$@_s[\text{tctl}]\neg\langle \ddot{z} = \text{-}b\rangle z \geq m \vdash @_s\neg\langle \text{tctl}; \text{accel}\rangle cr$$

Abbreviations: $c_2 \equiv (m{-}z{\geq}2e)$ and $\text{accel} \equiv \big(a := \text{-}b \cup (c_2?; a := 0.1)\big); \ddot{z} = a$

State-based Reasoning for Compositional Verification

$$\dfrac{\dfrac{\dfrac{\dfrac{*}{@_t\langle \ddot{z} = -b\rangle s, @_s crash \vdash @_s z \geq m}}{@_t\langle \ddot{z} = -b\rangle s, @_s crash \vdash @_t\langle \ddot{z} = -b\rangle z \geq m}}{@_t\langle a := -b\rangle r, @_t\langle \ddot{z} = -b\rangle crash \vdash @_t\langle \ddot{z} = -b\rangle z \geq m}}{@_t\langle a := -b\rangle r, @_r\langle \ddot{z} = a\rangle crash \vdash @_t\langle \ddot{z} = -b\rangle z \geq m}$$

State-based Reasoning for Compositional Verification

$$\dfrac{\dfrac{\dfrac{\dfrac{*}{@_t\langle \ddot{z} = \text{-}b\rangle s, @_s crash \vdash @_s z \geq m}}{@_t\langle \ddot{z} = \text{-}b\rangle s, @_s crash \vdash @_t\langle \ddot{z} = \text{-}b\rangle z \geq m}}{@_t\langle a := \text{-}b\rangle r, @_t\langle \ddot{z} = \text{-}b\rangle crash \vdash @_t\langle \ddot{z} = \text{-}b\rangle z \geq m}}{@_t\langle a := \text{-}b\rangle r, @_r\langle \ddot{z} = a\rangle crash \vdash @_t\langle \ddot{z} = \text{-}b\rangle z \geq m}$$

State-based Reasoning for Compositional Verification

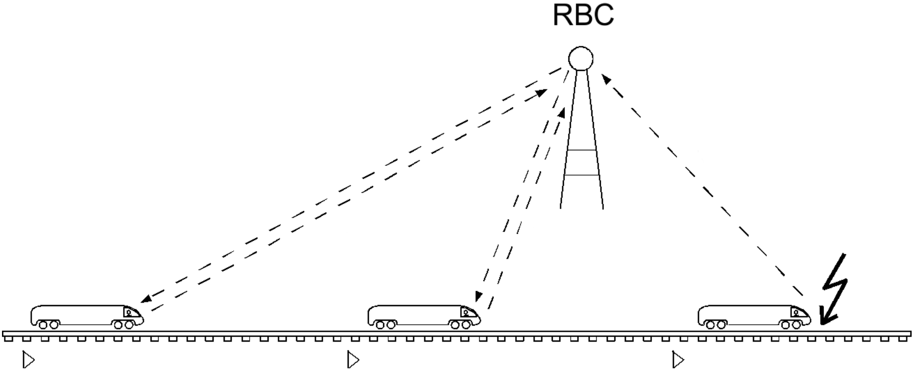
$$\dfrac{\dfrac{\dfrac{\dfrac{*}{@_t\langle \ddot{z} = \text{-}b\rangle s, @_s crash \vdash @_s z \geq m}}{@_t\langle \ddot{z} = \text{-}b\rangle s, @_s crash \vdash @_t\langle \ddot{z} = \text{-}b\rangle z \geq m}}{@_t\langle a := \text{-}b\rangle r, @_t\langle \ddot{z} = \text{-}b\rangle crash \vdash @_t\langle \ddot{z} = \text{-}b\rangle z \geq m}}{@_t\langle a := \text{-}b\rangle r, @_r\langle \ddot{z} = a\rangle crash \vdash @_t\langle \ddot{z} = \text{-}b\rangle z \geq m}$$

State-based Reasoning for Compositional Verification

$$\dfrac{\dfrac{\dfrac{\dfrac{*}{@_t\langle \ddot{z} = -b\rangle s, @_s crash \vdash @_s z \geq m}}{@_t\langle \ddot{z} = -b\rangle s, @_s crash \vdash @_t\langle \ddot{z} = -b\rangle z \geq m}}{@_t\langle a := -b\rangle r, @_t\langle \ddot{z} = -b\rangle crash \vdash @_t\langle \ddot{z} = -b\rangle z \geq m}}{@_t\langle a := -b\rangle r, @_r\langle \ddot{z} = a\rangle crash \vdash @_t\langle \ddot{z} = -b\rangle z \geq m}$$

State-based Reasoning for Compositional Verification

$$\dfrac{\dfrac{\dfrac{\dfrac{*}{@_t\langle \ddot{z} = \text{-}b\rangle s, @_s crash \vdash @_s z \geq m}}{@_t\langle \ddot{z} = \text{-}b\rangle s, @_s crash \vdash @_t\langle \ddot{z} = \text{-}b\rangle z \geq m}}{@_t\langle a := \text{-}b\rangle r, @_t\langle \ddot{z} = \text{-}b\rangle crash \vdash @_t\langle \ddot{z} = \text{-}b\rangle z \geq m}}{@_t\langle a := \text{-}b\rangle r, @_r\langle \ddot{z} = a\rangle crash \vdash @_t\langle \ddot{z} = \text{-}b\rangle z \geq m}$$



Soundness & Reduction

Theorem (Soundness)

d$\mathcal{L}_h$ *calculus is sound.*

Remark (Incompleteness)

(unbounded) d$\mathcal{L}_h$ *logic is inherently incomplete.*

Proposition (Reducibility)

d$\mathcal{L}_h$ *is reducible to* d$\mathcal{L}$.

Proof (Sketch): states characterised by variable assignments

$$i \rightsquigarrow \vec{i} = \vec{x}$$

$$@_i\phi \rightsquigarrow \langle \vec{x} := \vec{i} \rangle \phi$$

Outline

Future Work

- Levels of completeness
- Parallel systems
- Verification tool

Conclusions

- Challenges (Hybrid Dynamic Systems)
  1. Verify intricate dynamics in isolation
  2. Integrability of local correctness
- $d\mathcal{L}_h$ is a hybrid dynamic logic extending $d\mathcal{L}$ for compositionality:
  - State-based reasoning
  - Introspection
- Calculus with goal-directed interface to mathematical problem solving

Outline

5. The Logic $d\mathcal{L}_h$ (Details)
   - Hybrid Dynamic Logic vs. Hybrid Dynamic Systems
   - Syntax
   - Semantics

6. Appendix
   - ETCS in Mathematica
   - Flexible Verification Language

Hybrid Dynamic Logic vs. Hybrid Dynamic Systems

dynamic logic	$:=$	logic with program-modalities
dynamic system	$:=$	states vary along ODE
hybrid logic	$:=$	logic with state-references
hybrid system	$:=$	interacting discrete & continuous behaviour

The Logic $d\mathcal{L}_h$: Syntax

Definition (Formulas ϕ)

$\neg, \wedge, \vee, \rightarrow, \leftrightarrow, \; \forall x, \exists x, \; =, \geq, \leq, \; +, \cdot$	(first-order part)
$[\alpha]\phi, \quad \langle\alpha\rangle\phi$	(dynamic part)
$i, \quad @_i\phi$	(hybrid part)

Definition (System actions α)

$x := \theta$	(discrete mode switch)
$\dot{x} = \theta$	(continuous evolution)
$\phi?$	(conditional execution)
$\alpha;\gamma$	(seq. composition)
$\alpha \cup \gamma$	(nondet. choice)
α^*	(nondet. repetition)

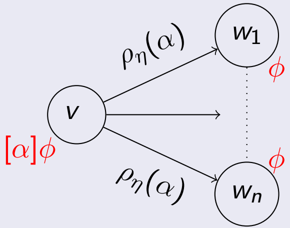
Return

The Logic $d\mathcal{L}_h$: Semantics

Definition (Formulas ϕ)

$$
\begin{aligned}
val_\eta(v, [\alpha]\phi) = true \quad &:\iff \quad val_\eta(w, \phi) = true \ \ \forall w \text{ with } (v, w) \in \rho_\eta(\alpha) \\
val_\eta(v, \langle\alpha\rangle\phi) = true \quad &:\iff \quad val_\eta(w, \phi) = true \ \ \exists w \text{ with } (v, w) \in \rho_\eta(\alpha) \\
val_\eta(v, i) = true \quad &:\iff \quad \eta(i) = v \\
val_\eta(v, @_i\phi) = true \quad &:\iff \quad val_\eta(\eta(i), \phi) = true
\end{aligned}
$$

Definition (System actions α)



The Logic d$\mathcal{L}_h$: Semantics

Definition (Formulas ϕ)

$$
\begin{aligned}
val_\eta(v,[\alpha]\phi) = true &\quad :\iff \quad val_\eta(w,\phi) = true \;\; \forall w \text{ with } (v,w) \in \rho_\eta(\alpha)\\
val_\eta(v,\langle\alpha\rangle\phi) = true &\quad :\iff \quad val_\eta(w,\phi) = true \;\; \exists w \text{ with } (v,w) \in \rho_\eta(\alpha)\\
val_\eta(v,i) = true &\quad :\iff \quad \eta(i) = v\\
val_\eta(v,@_i\phi) = true &\quad :\iff \quad val_\eta(\eta(i),\phi) = true
\end{aligned}
$$

Definition (System actions α)

$$
\begin{aligned}
(v,w) \in \rho_\eta(x := \theta) &\quad :\iff \quad w = v[x \mapsto val_\eta(v,\theta)]\\
(v,w) \in \rho_\eta(\dot{x} = f(x)) &\quad :\iff \quad \text{“}\tfrac{\mathrm{d}}{\mathrm{d}\tau} val_\eta(\cdot,x)(\zeta) = val_\eta(\zeta,f(x)) \quad \forall \zeta \in (v,\\
\rho_\eta(\phi?) &\quad = \quad \{(v,v) \;:\; val_\eta(v,\phi) = true\}\\
\rho_\eta(\alpha;\gamma) &\quad = \quad \rho_\eta(\alpha) \circ \rho_\eta(\gamma)\\
\rho_\eta(\alpha \cup \gamma) &\quad = \quad \rho_\eta(\alpha) \cup \rho_\eta(\gamma)\\
(v,w) \in \rho_\eta(\alpha^*) &\quad :\iff \quad \exists \; v \xrightarrow{\rho_\eta(\alpha)} s_1 \xrightarrow{\rho_\eta(\alpha)} \cdots\cdots\cdots \xrightarrow{\rho_\eta(\alpha)} w
\end{aligned}
$$

The Logic $d\mathcal{L}_h$: Semantics

Definition (Formulas ϕ)

$$val_\eta(v,[\alpha]\phi) = true \quad :\Longleftrightarrow \quad val_\eta(w,\phi) = true \;\; \forall w \text{ with } (v,w) \in \rho_\eta(\alpha)$$
$$val_\eta(v,\langle\alpha\rangle\phi) = true \quad :\Longleftrightarrow \quad val_\eta(w,\phi) = true \;\; \exists w \text{ with } (v,w) \in \rho_\eta(\alpha)$$
$$val_\eta(v,i) = true \quad :\Longleftrightarrow \quad \eta(i) = v$$
$$val_\eta(v,@_i\phi) = true \quad :\Longleftrightarrow \quad val_\eta(\eta(i),\phi) = true$$

Definition (System actions α)

$$(v,w) \in \rho_\eta(x := \theta) \quad :\Longleftrightarrow \quad w = v[x \mapsto val_\eta(v,\theta)]$$
$$(v,w) \in \rho_\eta(\dot{x} = f(x)) \quad :\Longleftrightarrow \quad \text{“}\frac{\mathrm{d}}{\mathrm{d}\tau} val_\eta(\cdot,x)(\zeta) = val_\eta(\zeta,f(x)) \quad \forall \zeta \in (v,$$
$$\rho_\eta(\phi?) \quad = \quad \{(v,v) \;:\; val_\eta(v,\phi) = true\}$$
$$\rho_\eta(\alpha;\gamma) \quad = \quad \rho_\eta(\alpha) \circ \rho_\eta(\gamma)$$
$$\rho_\eta(\alpha \cup \gamma) \quad = \quad \rho_\eta(\alpha) \cup \rho_\eta(\gamma)$$
$$(v,w) \in \rho_\eta(\alpha^*) \quad :\Longleftrightarrow \quad \exists \;\; v \xrightarrow{\rho_\eta(\alpha)} s_1 \xrightarrow{\rho_\eta(\alpha)} \cdots\cdots\cdots \xrightarrow{\rho_\eta(\alpha)} w$$

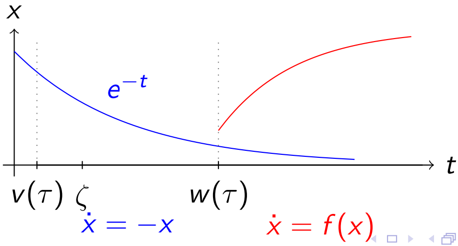
The Logic $d\mathcal{L}_h$: Semantics

Definition (System actions α)

$$(v, w) \in \rho_\eta(\dot{x} = f(x)) \quad :\iff \quad \text{“}\frac{\mathrm{d}}{\mathrm{d}\tau} val_\eta(t, x)(\zeta) = val_\eta(\zeta, f(x)) \quad \forall \zeta \in (v,$$

$$:\iff \quad \exists f : [v(\tau), w(\tau)] \to \text{Int}$$

- $\gamma_x(\zeta) \;:=\; val_\eta(f(\zeta), x)$ continuous on $[v(\tau), w(\tau)]$
- $\dot{\gamma}_x(\zeta) = \gamma_{f(x)}(\zeta)$, $\forall \zeta \in (v(\tau), w(\tau))$
- γ_y constant $\forall y \neq x$ and $f(v(\tau)) = v, f(w(\tau)) = w$

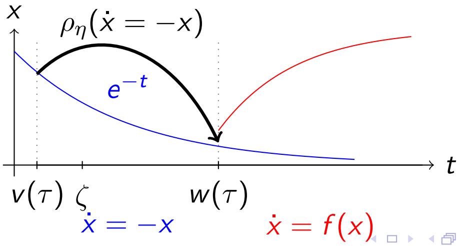


Definition (System actions α)

$$(v, w) \in \rho_\eta(\dot{x} = f(x)) \quad :\Longleftrightarrow \quad \text{“}\frac{\mathrm{d}}{\mathrm{d}\tau} val_\eta(t, x)(\zeta) = val_\eta(\zeta, f(x)) \quad \forall \zeta \in (v,$$

$$:\Longleftrightarrow \quad \exists f : [v(\tau), w(\tau)] \to \mathrm{Int}$$

- $\gamma_x(\zeta) \;:=\; val_\eta(f(\zeta), x)$ continuous on $[v(\tau), w(\tau)]$
- $\dot{\gamma}_x(\zeta) = \gamma_{f(x)}(\zeta)$, $\forall \zeta \in (v(\tau), w(\tau))$
- γ_y constant $\forall y \neq x$ and $f(v(\tau)) = v$, $f(w(\tau)) = w$



Outline

5 The Logic $d\mathcal{L}_h$ (Details)
- Hybrid Dynamic Logic vs. Hybrid Dynamic Systems
- Syntax
- Semantics

6 Appendix
- ETCS in Mathematica
- Flexible Verification Language

ETCS: Movement Authority with Mathematica

antecedent ⇒ <IVP>query

$\text{antecedent} = (z|m|b) \in \text{Reals} \wedge 0 < \text{z0} < m \wedge b > 0 \wedge \text{v0} > 0;$

$\text{ODE} = z''[t] == -b;$

$\text{IVP} = \{\text{ODE}, z[0] == \text{z0}, z'[0] == \text{v0}\};$

$\text{dsol} = \text{Simplify}[\text{DSolve}[\text{IVP}, z[t], t]]$

$\text{query} = z[t] == m;$

$$\left\{\left\{z[t] \to -\frac{bt^2}{2} + t\text{v0} + \text{z0}\right\}\right\}$$

$(\text{query}/.\text{dsol})[[1]]$

$\text{Reduce}[\text{Assuming}[\text{antecedent}, \text{Exists}[t, t \geq 0\&\&t \in \text{Reals}, \text{Assuming}[\text{antecedent}, \%]]], t, \text{Reals}]$

$\text{Simplify}[\%, \text{antecedent}]$

$$-\frac{bt^2}{2} + t\text{v0} + \text{z0} == m$$

$$\left(m < \text{z0}\&\&\left(\left(\text{v0} < 0\&\&b \geq \frac{\text{v0}^2}{2m-2\text{z0}}\right) \middle\| (\text{v0} \geq 0\&\&b > 0)\right)\right) \Bigg\|$$

$$m == \text{z0} \Bigg\| \left(m > \text{z0}\&\&\left((\text{v0} \leq 0\&\&b < 0) \middle\| \left(\text{v0} > 0\&\&b \leq \frac{\text{v0}^2}{2m-2\text{z0}}\right)\right)\right)$$

$$2b(m - \text{z0}) \leq \text{v0}^2$$

Flexible Verification Language

Example (Verification Tasks)

1. System verification problem (flat / compositional)
$$b \geq 10 \rightarrow [\alpha]z \leq m$$
2. (Compositional) refinement
$$[S]\langle C\rangle \mathit{safe}$$
3. Abstraction
$$f < \epsilon \;\rightarrow\; ([\tilde{\alpha}]\phi \rightarrow [\alpha]\phi)$$
4. Level of detail or "layered" time models
$$[x := 4]\phi \;\rightarrow\; [\dot{t} = 1; x := 4](t \leq 5 \rightarrow \phi)$$