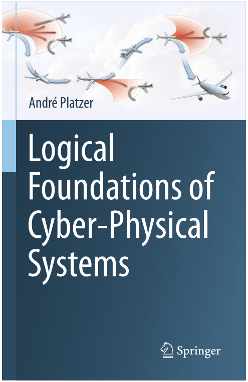
04: Safety & Contracts

Logical Foundations of Cyber-Physical Systems



André Platzer

Carnegie Mellon University
Computer Science Department

Outline

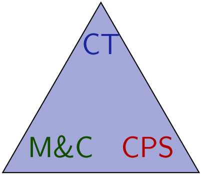
Outline

1. Learning Objectives
2. Quantum the Acrophobic Bouncing Ball
3. Contracts for CPS
   - Safety of Robots
   - Safety of Bouncing Balls
4. Logical Formulas for Hybrid Programs
5. Differential Dynamic Logic
   - Syntax
   - Semantics
   - Notational Convention
6. Identifying Requirements of a CPS
7. Summary

Learning Objectives

Safety & Contracts

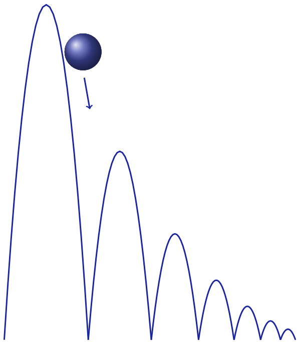
rigorous specification
contracts
preconditions
postconditions
differential dynamic logic



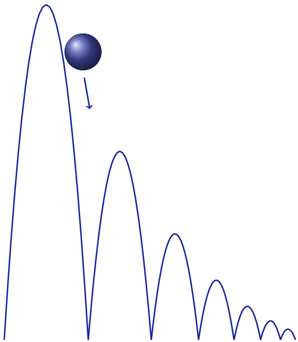
discrete+continuous
analytic specification

model semantics
reasoning principles

Outline

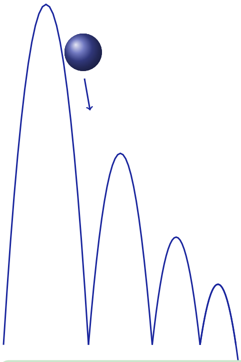


Example (Quantum the Bouncing Ball)



Example (Quantum the Bouncing Ball)

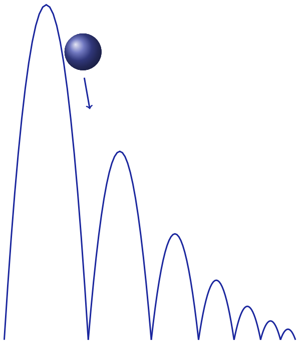
$$\{x' = v, v' = -g\}$$



Example (Quantum the Bouncing Ball)

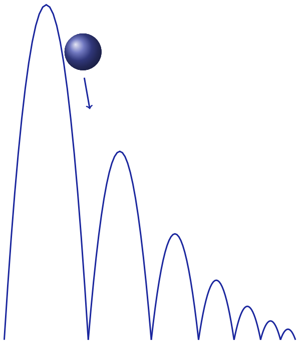
$$\{x' = v, v' = -g\}$$

Quantum the Acrophobic Bouncing Ball



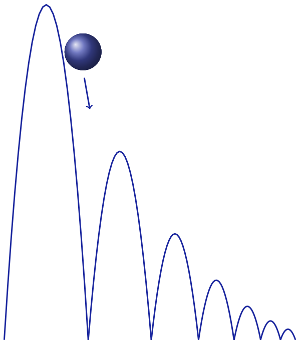
Example (Quantum the Bouncing Ball)

$$\{x' = v, v' = -g \,\&\, x \geq 0\}$$



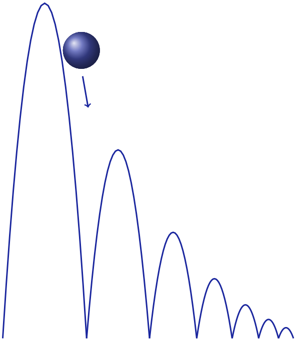
Example (Quantum the Bouncing Ball)

$$\{x' = v, v' = -g \,\&\, x \geq 0\};$$
$$\text{if}(x = 0)\ \ v := -cv$$



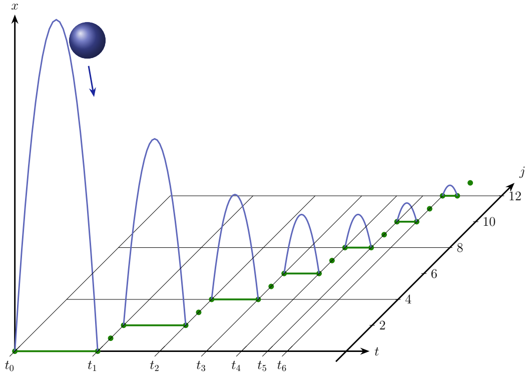
Example (Quantum the Bouncing Ball)

$$\big(\{x' = v, v' = -g \,\&\, x \geq 0\};$$
$$\text{if}(x = 0)\ \ v := -cv\big)^*$$



Example (Quantum the Bouncing Ball)

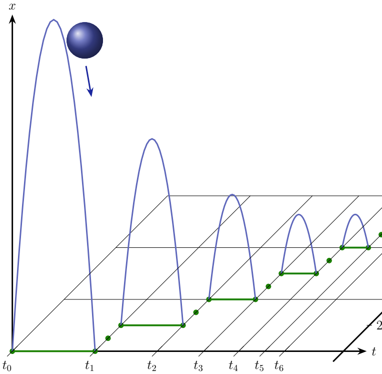
$$\big(\{x' = v, v' = -g \,\&\, x \geq 0\};$$
$$\text{if}(x = 0)\ \ v := -cv\big)^*$$

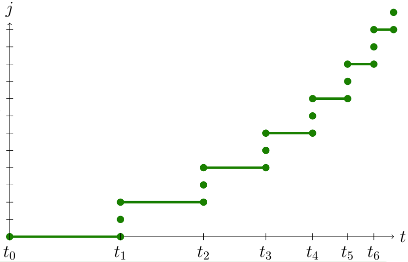


Example (Quantum the Bouncing Ball)

$$\big(\{x' = v, v' = -g \,\&\, x \geq 0\};$$
$$\text{if}(x = 0)\ \ v := -cv\big)^*$$

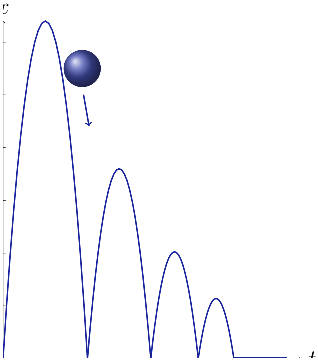
Quantum Discovered a Crack in the Fabric of Time





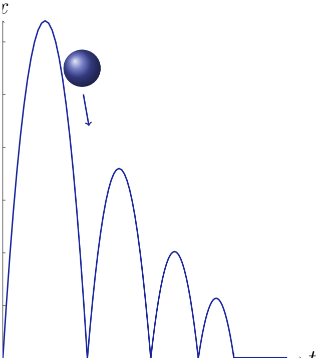
Example (Quantum the Bouncing Ball)

$$\big(\{x' = v, v' = -g \,\&\, x \geq 0\};$$
$$\text{if}(x = 0)\;\; v := -cv\big)^*$$



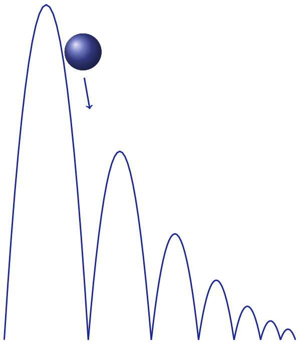
Example (Quantum the Bouncing Ball)

$$\big(\{x' = v, v' = -g \,\&\, x \geq 0\};$$
$$\text{if}(x = 0)\ \ v := -cv\big)^*$$



Example (Quantum the Bouncing Ball)

$$\left(\{x' = v, v' = -g \,\&\, x \geq 0\};\right.$$
$$\left.\text{if}(x = 0)\,(v := -cv \cup v := 0)\right)^*$$



Example (Quantum the Bouncing Ball)

$$\big(\{x' = v, v' = -g \,\&\, x \geq 0\};$$
$$\text{if}(x = 0)\ \ v := -cv\big)^*$$

Safety of Robots

Three Laws of Robotics — Isaac Asimov 1942

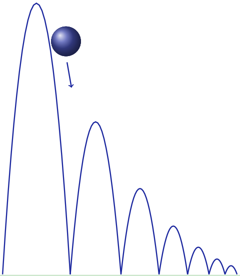
1. A robot may not injure a human being or, through inaction, allow a human being to come to harm.
2. A robot must obey the orders given to it by human beings, except where such orders would conflict with the First Law.
3. A robot must protect its own existence as long as such protection does not conflict with the First or Second Law.

Safety of Robots

Three Laws of Robotics — Isaac Asimov 1942

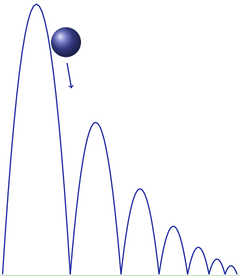
1. A robot may not injure a human being or, through inaction, allow a human being to come to harm.
2. A robot must obey the orders given to it by human beings, except where such orders would conflict with the First Law.
3. A robot must protect its own existence as long as such protection does not conflict with the First or Second Law.

Three Laws of Robotics are not the answer.
They are the inspiration!



Example (Quantum the Bouncing Ball)

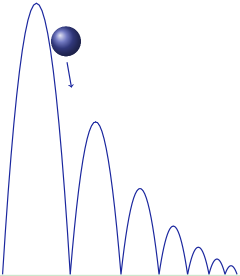
$$\left(\{x' = v, v' = -g \,\&\, x \geq 0\};\right.$$
$$\left.\text{if}(x = 0)\, v := -cv\right)^*$$



Example (Quantum the Bouncing Ball)

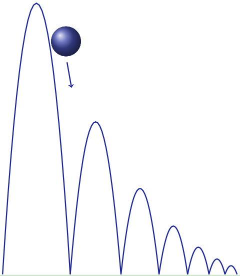
ensures$(0 \leq x)$

$$\left(\{x' = v, v' = -g \,\&\, x \geq 0\};\right.$$
$$\left.\text{if}(x = 0)\; v := -cv\right)^*$$



Example (Quantum the Bouncing Ball)

$$\mathbf{ensures}(0 \le x)$$
$$\mathbf{ensures}(x \le H)$$
$$\big(\{x' = v, v' = -g \,\&\, x \ge 0\};$$
$$\text{if}(x = 0)\; v := -cv\big)^*$$



Example (Quantum the Bouncing Ball)

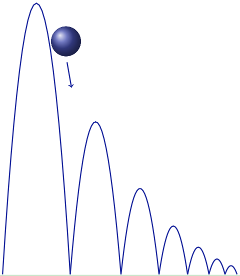
$\textbf{requires}(x = H)$

$\textbf{ensures}(0 \leq x)$

$\textbf{ensures}(x \leq H)$

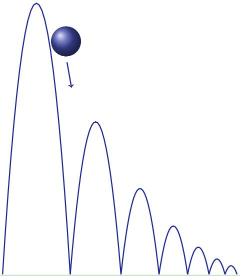
$\big(\{x' = v, v' = -g \,\&\, x \geq 0\};$

$\quad \text{if}(x = 0)\, v := -cv\big)^*$



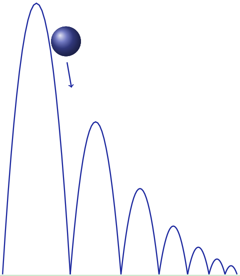
Example (Quantum the Bouncing Ball)

$$\mathbf{requires}(x = H)$$
$$\mathbf{requires}(0 \leq H)$$
$$\mathbf{ensures}(0 \leq x)$$
$$\mathbf{ensures}(x \leq H)$$
$$\big(\{x' = v, v' = -g \,\&\, x \geq 0\};$$
$$\quad \mathrm{if}(x = 0)\, v := -cv\big)^*$$



Example (Quantum the Bouncing Ball)

$$\begin{aligned}
&\textbf{requires}(x = H)\\
&\textbf{requires}(0 \le H)\\
&\textbf{ensures}(0 \le x)\\
&\textbf{ensures}(x \le H)\\
&\big(\{x' = v, v' = -g \,\&\, x \ge 0\};\\
&\quad \mathrm{if}(x = 0)\, v := -cv\big)^{*} \textbf{@invariant}(x \ge 0)
\end{aligned}$$



Example (Quantum the Bouncing Ball)

$$\mathbf{requires}(x = H)$$
$$\mathbf{requires}(0 \leq H)$$
$$\mathbf{ensures}(0 \leq x)$$
$$\mathbf{ensures}(x \leq H)$$
$$\big(\{x' = v, v' = -g \,\&\, x \geq 0\};$$
$$\text{if}(x = 0)\, v := -cv\big)^* \mathbf{@invariant}(x \geq 0)$$

Outline

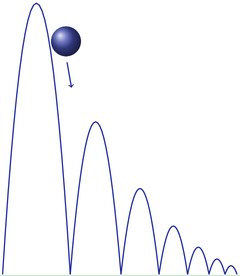
Contracts are Not Enough

CPS contracts are crucial for CPS safety.
We need to understand CPS programs and contracts and how we can convince ourselves that a CPS program respects its contract.

Contracts are at a disadvantage compared to full logic.

Logic is for Specification and Reasoning

1. Specification of a whole CPS program.
2. Analytic inspection of its parts.
3. Argumentative relations between contracts and program parts.
   "Yes, this CPS program meets its contract, and here's why ..."



Example (Quantum the Bouncing Ball)

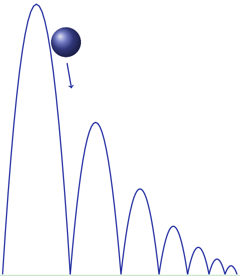
$\textbf{requires}(x = H)$

$\textbf{requires}(0 \leq H)$

$\textbf{ensures}(0 \leq x)$

$\textbf{ensures}(x \leq H)$

$$\big(\{x' = v, v' = -g \,\&\, x \geq 0\};\; \text{if}(x = 0)\, v := -cv\big)^*$$



Example (Quantum the Bouncing Ball)

requires$(x = H)$

requires$(0 \le H)$

ensures$(0 \le x)$

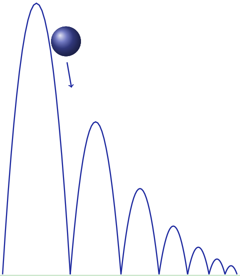
ensures$(x \le H)$

$\left(\{x' = v, v' = -g \,\&\, x \ge 0\};\right.$

$\left.\text{if}(x = 0)\, v := -cv\right)^*$

Precondition:

$x = H \land 0 \le H$ in FOL



Example (Quantum the Bouncing Ball)

requires$(x = H)$

requires$(0 \le H)$

ensures$(0 \le x)$

ensures$(x \le H)$

$\big(\{x' = v, v' = -g \,\&\, x \ge 0\};$

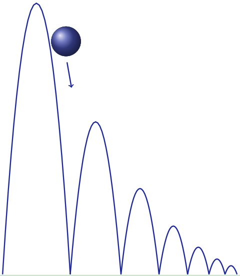
$\text{if}(x = 0)\ v := -cv\big)^*$

Precondition:

$x = H \land 0 \le H$ in FOL

Postcondition:

$0 \le x \land x \le H$ in FOL



Example (Quantum the Bouncing Ball)

requires$(x = H)$

requires$(0 \leq H)$

ensures$(0 \leq x)$

ensures$(x \leq H)$

$\big(\{x' = v, v' = -g \,\&\, x \geq 0\};$

$\quad \text{if}(x = 0)\, v := -cv\big)^*$

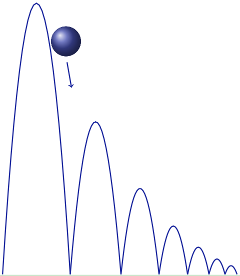
Precondition:

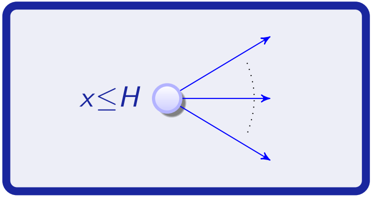
$x = H \land 0 \leq H$ in FOL

Postcondition:

$0 \leq x \land x \leq H$ in FOL

How to say post is true after all HP runs?





Example (Quantum the Bouncing Ball)

$\mathbf{requires}(x = H)$

$\mathbf{requires}(0 \le H)$

$\mathbf{ensures}(0 \le x)$

$\mathbf{ensures}(x \le H)$

$\big(\{x' = v, v' = -g \,\&\, x \ge 0\};$

$\quad \mathrm{if}(x = 0)\; v := -cv\big)^*$

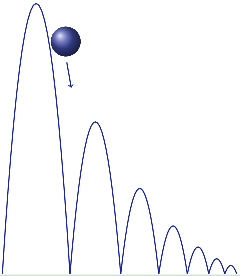
Precondition:

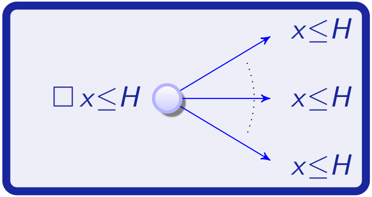
$x = H \land 0 \le H$ in FOL

Postcondition:

$0 \le x \land x \le H$ in FOL

Contracts for Quantum the Acrophobic Bouncing Ball





Example (Quantum the Bouncing Ball)

requires$(x = H)$

requires$(0 \leq H)$

ensures$(0 \leq x)$

ensures$(x \leq H)$

$\big(\{x' = v, v' = -g \,\&\, x \geq 0\};$

$\text{if}(x = 0)\; v := -cv\big)^*$

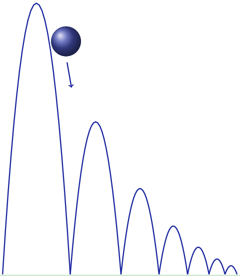
Precondition:

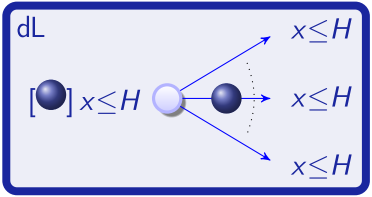
$x = H \land 0 \leq H$ in FOL

Postcondition:

$0 \leq x \land x \leq H$ in FOL

Contracts for Quantum the Acrophobic Bouncing Ball





Example (Quantum the Bouncing Ball)

requires$(x = H)$

requires$(0 \le H)$

ensures$(0 \le x)$

ensures$(x \le H)$

$\big(\{x' = v, v' = -g \,\&\, x \ge 0\};$

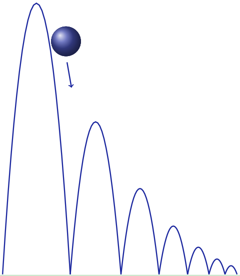
$\quad \mathrm{if}(x = 0)\; v := -cv\big)^*$

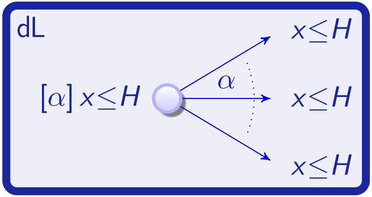
Precondition:

$x = H \land 0 \le H$ in FOL

Postcondition:

$0 \le x \land x \le H$ in FOL





Example (Quantum the Bouncing Ball)

requires$(x = H)$
requires$(0 \le H)$

Precondition:
$x = H \land 0 \le H$ in FOL

ensures$(0 \le x)$
ensures$(x \le H)$

Postcondition:
$0 \le x \land x \le H$ in FOL

$$\big(\{x' = v, v' = -g \,\&\, x \ge 0\};$$
$$\text{if}(x = 0)\; v := -cv\big)^*$$

$$[\big(\{x' = v, v' = -g \,\&\, x{\geq}0\}; \text{if}(x{=}0)\, v := -cv\big)^*]$$

Example (Quantum the Bouncing Ball)

requires$(x = H)$

requires$(0 \leq H)$

ensures$(0 \leq x)$

ensures$(x \leq H)$

$\big(\{x' = v, v' = -g \,\&\, x \geq 0\};$

$\text{if}(x = 0)\, v := -cv\big)^*$

Precondition:

$x = H \land 0 \leq H$ in FOL

Postcondition:

$0 \leq x \land x \leq H$ in FOL

$$[(\{x' = v, v' = -g \,\&\, x{\geq}0\}; \mathrm{if}(x{=}0)\, v := -cv)^*](x \leq H)$$

Example (Quantum the Bouncing Ball)

requires$(x = H)$
requires$(0 \leq H)$
ensures$(0 \leq x)$
ensures$(x \leq H)$
$(\{x' = v, v' = -g \,\&\, x \geq 0\};$
$\mathrm{if}(x = 0)\, v := -cv)^*$

Precondition:
$x = H \land 0 \leq H$ in FOL
Postcondition:
$0 \leq x \land x \leq H$ in FOL

$$[\big(\{x' = v, v' = -g \,\&\, x{\geq}0\}; \text{if}(x{=}0)\; v := -cv\big)^*](0 \leq x)$$
$$[\big(\{x' = v, v' = -g \,\&\, x{\geq}0\}; \text{if}(x{=}0)\; v := -cv\big)^*](x \leq H)$$

Example (Quantum the Bouncing Ball)

requires$(x = H)$
requires$(0 \leq H)$
ensures$(0 \leq x)$
ensures$(x \leq H)$
$\big(\{x' = v, v' = -g \,\&\, x \geq 0\};$
$\quad \text{if}(x = 0)\; v := -cv\big)^*$

Precondition:
$x = H \land 0 \leq H$ in FOL
Postcondition:
$0 \leq x \land x \leq H$ in FOL

$$[(\{x' = v, v' = -g \,\&\, x{\geq}0\}; \mathrm{if}(x{=}0)\, v := -cv)^*](0 \leq x)$$
$$[(\{x' = v, v' = -g \,\&\, x{\geq}0\}; \mathrm{if}(x{=}0)\, v := -cv)^*](x \leq H)$$
$$[(\{x' = v, v' = -g \,\&\, x{\geq}0\}; \mathrm{if}(x{=}0)\, v := -cv)^*](0{\leq}x \land x{\leq}H)$$

Example (Quantum the Bouncing Ball)

requires$(x = H)$
requires$(0 \leq H)$
ensures$(0 \leq x)$
ensures$(x \leq H)$
$(\{x' = v, v' = -g \,\&\, x \geq 0\};$
$\mathrm{if}(x = 0)\, v := -cv)^*$

Precondition:
$x = H \land 0 \leq H$ in FOL
Postcondition:
$0 \leq x \land x \leq H$ in FOL

$$
\begin{aligned}
& [(\{x' = v, v' = -g \,\&\, x{\ge}0\}; \mathrm{if}(x{=}0)\, v := -cv)^*](0 \le x) \\
\wedge\ & [(\{x' = v, v' = -g \,\&\, x{\ge}0\}; \mathrm{if}(x{=}0)\, v := -cv)^*](x \le H) \\
\leftrightarrow\ & [(\{x' = v, v' = -g \,\&\, x{\ge}0\}; \mathrm{if}(x{=}0)\, v := -cv)^*](0{\le}x \wedge x{\le}H)
\end{aligned}
$$

Example (Quantum the Bouncing Ball)

requires$(x = H)$

requires$(0 \le H)$

ensures$(0 \le x)$

ensures$(x \le H)$

$(\{x' = v, v' = -g \,\&\, x \ge 0\};$

$\mathrm{if}(x = 0)\, v := -cv)^*$

Precondition:

$x = H \wedge 0 \le H$ in FOL

Postcondition:

$0 \le x \wedge x \le H$ in FOL

$$[(\{x' = v, v' = -g \,\&\, x{\geq}0\}; \mathrm{if}(x{=}0)\, v := -cv)^*](0 \leq x)$$

Example (Quantum the Bouncing Ball)

requires$(x = H)$

requires$(0 \leq H)$

ensures$(0 \leq x)$

ensures$(x \leq H)$

$(\{x' = v, v' = -g \,\&\, x \geq 0\};$

$\mathrm{if}(x = 0)\, v := -cv)^*$

Precondition:

$x = H \land 0 \leq H$ in FOL

Postcondition:

$0 \leq x \land x \leq H$ in FOL

Contracts for Quantum the Acrophobic Bouncing Ball

$$x{=}H \to [(\{x' = v, v' = -g \,\&\, x{\geq}0\}; \text{if}(x{=}0)\, v := -cv)^*](0 \leq x)$$

Example (Quantum the Bouncing Ball)

requires$(x = H)$	Precondition:
requires$(0 \leq H)$	$x = H \land 0 \leq H$ in FOL
ensures$(0 \leq x)$	Postcondition:
ensures$(x \leq H)$	$0 \leq x \land x \leq H$ in FOL

$$(\{x' = v, v' = -g \,\&\, x \geq 0\};$$
$$\text{if}(x = 0)\, v := -cv)^*$$

$$0\leq x \land x=H \rightarrow [(\{x' = v, v' = -g \,\&\, x\geq 0\}; \text{if}(x=0)\, v := -cv)^*](0 \leq x)$$

Example (Quantum the Bouncing Ball)

requires$(x = H)$

requires$(0 \leq H)$

ensures$(0 \leq x)$

ensures$(x \leq H)$

$(\{x' = v, v' = -g \,\&\, x \geq 0\};$

$\text{if}(x = 0)\, v := -cv)^*$

Precondition:

$x = H \land 0 \leq H$ in FOL

Postcondition:

$0 \leq x \land x \leq H$ in FOL

Outline

1. Learning Objectives
2. Quantum the Acrophobic Bouncing Ball
3. Contracts for CPS
   - Safety of Robots
   - Safety of Bouncing Balls
4. Logical Formulas for Hybrid Programs
5. Differential Dynamic Logic
   - Syntax
   - Semantics
   - Notational Convention
6. Identifying Requirements of a CPS
7. Summary

Differential Dynamic Logic: Syntax

Definition (Syntax of differential dynamic logic)

The *formulas* of *differential dynamic logic* are defined by the grammar:

$$P, Q ::= e \geq \tilde{e} \mid \neg P \mid P \land Q \mid P \lor Q \mid P \rightarrow Q \mid \forall x\, P \mid \exists x\, P \mid [\alpha]P \mid \langle\alpha\rangle P$$

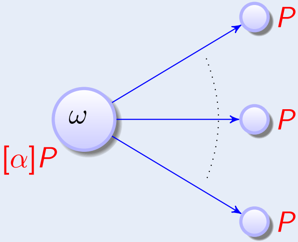
Differential Dynamic Logic: Syntax

Definition (Syntax of differential dynamic logic)

The *formulas* of *differential dynamic logic* are defined by the grammar:

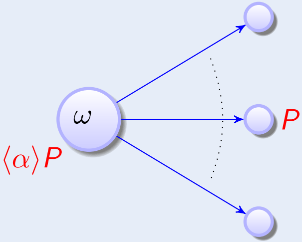
$$P, Q ::= e \geq \tilde{e} \mid \underbrace{\neg P}_{\text{Not}} \mid \underbrace{P \land Q}_{\text{And}} \mid \underbrace{P \lor Q}_{\text{Or}} \mid \underbrace{P \rightarrow Q}_{\text{Imply}} \mid \underbrace{\forall x\, P}_{\text{All reals}} \mid \underbrace{\exists x\, P}_{\text{Some real}} \mid \underbrace{[\alpha]P}_{\text{All runs}} \mid \underbrace{\langle\alpha\rangle P}_{\text{Some runs}}$$

Definition (dL Formulas)

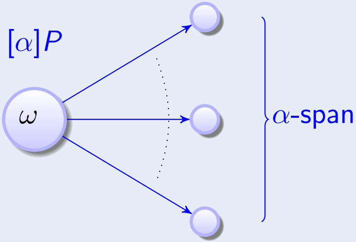


Differential Dynamic Logic dL: Semantics

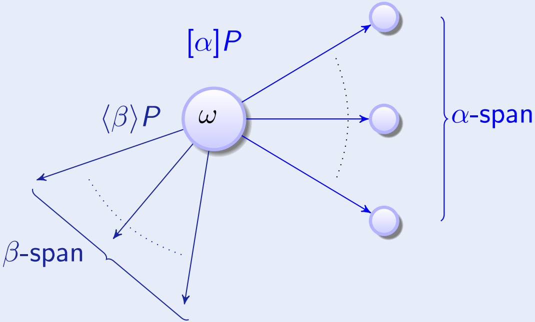
Definition (dL Formulas)



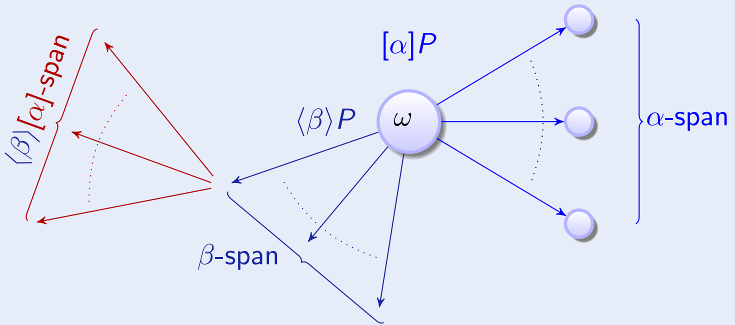
Definition (dL Formulas)



Definition (dL Formulas)



Definition (dL Formulas)



Definition (Syntax of differential dynamic logic)

The *formulas* of *differential dynamic logic* are defined by the grammar:

$$P, Q ::= e \geq \tilde{e} \mid \neg P \mid P \wedge Q \mid P \vee Q \mid P \rightarrow Q \mid \forall x\, P \mid \exists x\, P \mid [\alpha]P \mid \langle\alpha\rangle P$$

Definition (dL semantics) $([\![\cdot]\!] : \mathrm{Fml} \rightarrow \wp(\mathcal{S}))$

$$\begin{aligned}
[\![e \geq \tilde{e}]\!] &= \{\omega \;:\; \omega[\![e]\!] \geq \omega[\![\tilde{e}]\!]\} \\
[\![\neg P]\!] &= [\![P]\!]^{\complement} = \mathcal{S} \setminus [\![P]\!] \\
[\![P \wedge Q]\!] &= [\![P]\!] \cap [\![Q]\!] \\
[\![P \vee Q]\!] &= [\![P]\!] \cup [\![Q]\!] \\
[\![P \rightarrow Q]\!] &= [\![P]\!]^{\complement} \cup [\![Q]\!] \\
[\![\langle\alpha\rangle P]\!] &= [\![\alpha]\!] \circ [\![P]\!] = \{\omega \;:\; \nu \in [\![P]\!] \text{ for some } \nu :\; (\omega,\nu) \in [\![\alpha]\!]\} \\
[\![[\alpha]P]\!] &= [\![\neg\langle\alpha\rangle\neg P]\!] = \{\omega \;:\; \nu \in [\![P]\!] \text{ for all } \nu :\; (\omega,\nu) \in [\![\alpha]\!]\} \\
[\![\exists x\, P]\!] &= \{\omega \;:\; \omega_x^r \in [\![P]\!] \text{ for some } r \in \mathbb{R}\} \\
[\![\forall x\, P]\!] &= \{\omega \;:\; \omega_x^r \in [\![P]\!] \text{ for all } r \in \mathbb{R}\}
\end{aligned}$$

$$\omega_x^d(y) = \begin{cases} d & \text{if } y{=}x \\ \omega(y) & \text{if } y{\neq}x \end{cases}$$

Differential Dynamic Logic: Syntax & Semantics

$[\![P]\!]$	the set of states in which formula P is true
$\omega \in [\![P]\!]$	formula P is true in state ω, alias $\omega \models P$
$\models P$	formula P is valid, i.e., true in all states ω, i.e., $[\![P]\!] = \mathcal{S}$

Definition (dL semantics) $([\![\cdot]\!] : \mathrm{Fml} \to \wp(\mathcal{S}))$

$$
\begin{aligned}
[\![e \geq \tilde{e}]\!] &= \{\omega : \omega[\![e]\!] \geq \omega[\![\tilde{e}]\!]\} \\
[\![\neg P]\!] &= [\![P]\!]^{\complement} = \mathcal{S} \setminus [\![P]\!] \\
[\![P \land Q]\!] &= [\![P]\!] \cap [\![Q]\!] \\
[\![P \lor Q]\!] &= [\![P]\!] \cup [\![Q]\!] \\
[\![P \to Q]\!] &= [\![P]\!]^{\complement} \cup [\![Q]\!] \\
[\![\langle\alpha\rangle P]\!] &= [\![\alpha]\!] \circ [\![P]\!] = \{\omega : \nu \in [\![P]\!] \text{ for some } \nu : (\omega,\nu) \in [\![\alpha]\!]\} \\
[\![[\alpha]P]\!] &= [\![\neg\langle\alpha\rangle\neg P]\!] = \{\omega : \nu \in [\![P]\!] \text{ for all } \nu : (\omega,\nu) \in [\![\alpha]\!]\} \\
[\![\exists x\, P]\!] &= \{\omega : \omega_x^r \in [\![P]\!] \text{ for some } r \in \mathbb{R}\} \\
[\![\forall x\, P]\!] &= \{\omega : \omega_x^r \in [\![P]\!] \text{ for all } r \in \mathbb{R}\}
\end{aligned}
$$

$[\![P]\!]$	the set of states in which formula P is true
$\omega \in [\![P]\!]$	formula P is true in state ω, alias $\omega \models P$
$\models P$	formula P is valid, i.e., true in all states ω, i.e., $[\![P]\!] = \mathcal{S}$

$\exists d\, [x := 1; x'=d]x \geq 0$ and $\quad [x := x+1; x'=d]x \geq 0$ and $\quad \langle x'=d \rangle x \geq 0$

Definition (dL semantics) ($[\![\cdot]\!] : \mathrm{Fml} \to \wp(\mathcal{S})$)

$$
\begin{aligned}
[\![e \geq \tilde{e}]\!] &= \{\omega : \omega[\![e]\!] \geq \omega[\![\tilde{e}]\!]\} \\
[\![\neg P]\!] &= [\![P]\!]^{\complement} = \mathcal{S} \setminus [\![P]\!] \\
[\![P \wedge Q]\!] &= [\![P]\!] \cap [\![Q]\!] \\
[\![P \vee Q]\!] &= [\![P]\!] \cup [\![Q]\!] \\
[\![P \rightarrow Q]\!] &= [\![P]\!]^{\complement} \cup [\![Q]\!] \\
[\![\langle \alpha \rangle P]\!] &= [\![\alpha]\!] \circ [\![P]\!] = \{\omega : \nu \in [\![P]\!] \text{ for some } \nu : (\omega, \nu) \in [\![\alpha]\!]\} \\
[\![[\alpha] P]\!] &= [\![\neg \langle \alpha \rangle \neg P]\!] = \{\omega : \nu \in [\![P]\!] \text{ for all } \nu : (\omega, \nu) \in [\![\alpha]\!]\} \\
[\![\exists x\, P]\!] &= \{\omega : \omega_x^r \in [\![P]\!] \text{ for some } r \in \mathbb{R}\} \\
[\![\forall x\, P]\!] &= \{\omega : \omega_x^r \in [\![P]\!] \text{ for all } r \in \mathbb{R}\}
\end{aligned}
$$

$[\![P]\!]$	the set of states in which formula P is true
$\omega \in [\![P]\!]$	formula P is true in state ω, alias $\omega \models P$
$\models P$	formula P is valid, i.e., true in all states ω, i.e., $[\![P]\!] = \mathcal{S}$

$\models \exists d\, [x := 1; x'=d]x \geq 0$ and $\not\models [x := x+1; x'=d]x \geq 0$ and $\not\models \langle x'=d \rangle x \geq 0$

Definition (dL semantics) $([\![\cdot]\!] : \mathrm{Fml} \to \wp(\mathcal{S}))$

$$
\begin{aligned}
[\![e \geq \tilde{e}]\!] &= \{\omega : \omega[\![e]\!] \geq \omega[\![\tilde{e}]\!]\} \\
[\![\neg P]\!] &= [\![P]\!]^{\complement} = \mathcal{S} \setminus [\![P]\!] \\
[\![P \wedge Q]\!] &= [\![P]\!] \cap [\![Q]\!] \\
[\![P \vee Q]\!] &= [\![P]\!] \cup [\![Q]\!] \\
[\![P \to Q]\!] &= [\![P]\!]^{\complement} \cup [\![Q]\!] \\
[\![\langle \alpha \rangle P]\!] &= [\![\alpha]\!] \circ [\![P]\!] = \{\omega : \nu \in [\![P]\!] \text{ for some } \nu : (\omega, \nu) \in [\![\alpha]\!]\} \\
[\![[\alpha] P]\!] &= [\![\neg \langle \alpha \rangle \neg P]\!] = \{\omega : \nu \in [\![P]\!] \text{ for all } \nu : (\omega, \nu) \in [\![\alpha]\!]\} \\
[\![\exists x\, P]\!] &= \{\omega : \omega_x^r \in [\![P]\!] \text{ for some } r \in \mathbb{R}\} \\
[\![\forall x\, P]\!] &= \{\omega : \omega_x^r \in [\![P]\!] \text{ for all } r \in \mathbb{R}\}
\end{aligned}
$$

Notational Conventions: Precedence

Convention (Operator Precedence)

1. Unary operators (e.g., *, $\neg$, $\forall x, \exists x$, $[\alpha], \langle\alpha\rangle$) bind stronger than binary
2. $\wedge$ binds stronger than $\vee$, which binds stronger than $\rightarrow, \leftrightarrow$
3. ; binds stronger than $\cup$
4. Arithmetic operators $+, -, \cdot$ associate to the left
5. Logical and program operators associate to the right

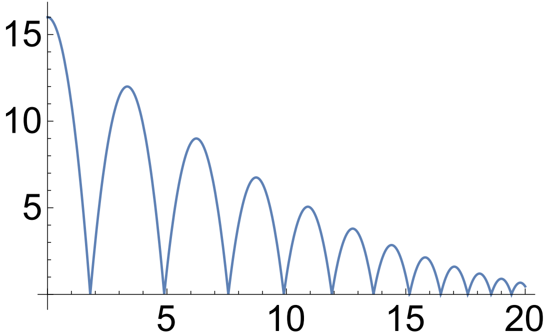
Example (Operator Precedence)

$[\alpha]P\wedge Q \equiv ([\alpha]P)\wedge Q \qquad \forall x\, P\wedge Q \equiv (\forall x\, P)\wedge Q \qquad \forall x\, P\rightarrow Q \equiv (\forall x\, P)\rightarrow Q$

$\alpha;\beta \cup \gamma \equiv (\alpha;\beta) \cup \gamma \qquad \alpha \cup \beta;\gamma \equiv \alpha \cup (\beta;\gamma) \qquad \alpha;\beta^* \equiv \alpha;(\beta^*)$

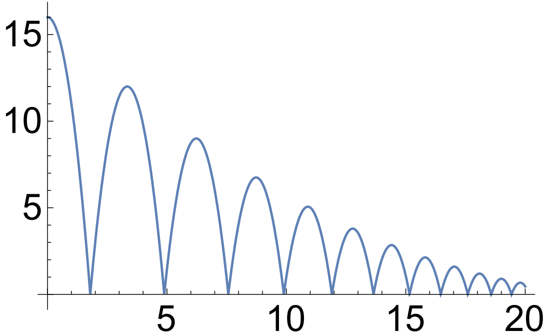
$P \rightarrow Q \rightarrow R \equiv P \rightarrow (Q \rightarrow R)$.

But $\rightarrow, \leftrightarrow$ expect explicit parentheses. Illegal: $P \rightarrow Q \leftrightarrow R \quad P \leftrightarrow Q \rightarrow R$



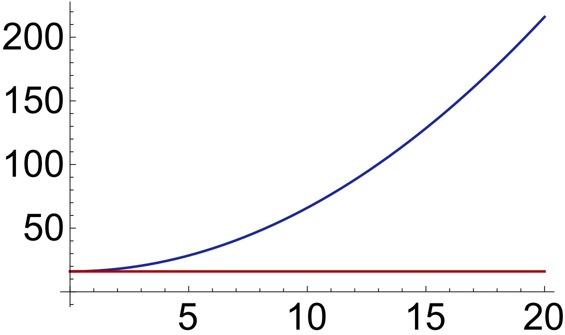
Example (Bouncing Ball)

$$\big(\{x' = v, v' = -g \,\&\, x \geq 0\};$$
$$\text{if}(x = 0)\; v := -cv\big)^*$$



Example (Bouncing Ball)

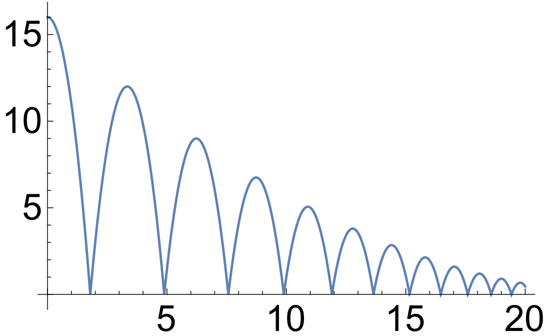
$$H=x\geq 0 \rightarrow \left[\left(\{x'=v, v'=-g \,\&\, x\geq 0\};\ \text{if}(x=0)\ v:=-cv\right)^*\right] 0\leq x\leq H$$



Not if $g < 0$ in anti-gravity

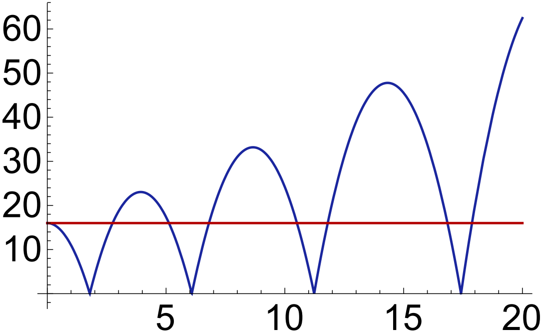
Example (Bouncing Ball)

$$H{=}x{\geq}0 \quad \rightarrow [(\{x' = v, v' = -g \,\&\, x \geq 0\};\ \mathrm{if}(x = 0)\ v := -cv)^*]\ 0{\leq}x{\leq}H$$



Example (Bouncing Ball)

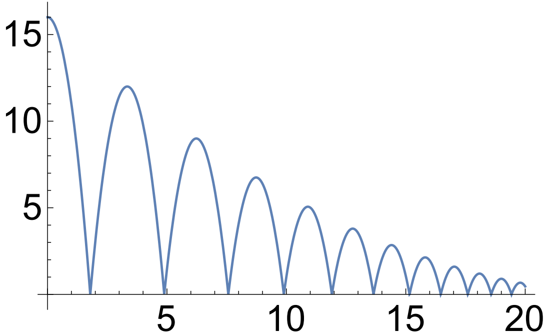
$$H{=}x{\geq}0 \land g{>}0 \rightarrow \big[\big(\{x' = v, v' = -g \,\&\, x \geq 0\};$$
$$\text{if}(x = 0)\; v := -cv\big)^*\big]\, 0{\leq}x{\leq}H$$



Not if $c > 1$ for anti-damping

Example (Bouncing Ball)

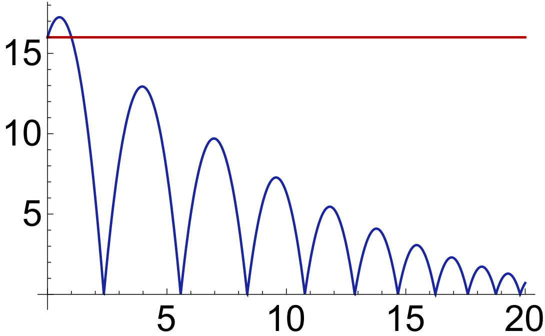
$$H = x \geq 0 \land g > 0 \to [(\{x' = v, v' = -g \,\&\, x \geq 0\};\ \text{if}(x = 0)\ v := -cv)^*]\ 0 \leq x \leq H$$



Example (Bouncing Ball)

$$1{\geq}c{\geq}0 \land H{=}x{\geq}0 \land g{>}0 \rightarrow \big[\big(\{x' = v, v' = -g \,\&\, x \geq 0\};$$
$$\text{if}(x = 0)\ v := -cv\big)^*\big]\ 0{\leq}x{\leq}H$$

Quantum the Acrophobic Bouncing Ball

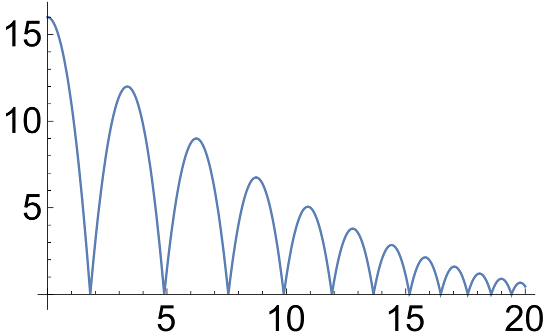


Not if $v > 0$ initial climbing

Example (Bouncing Ball)

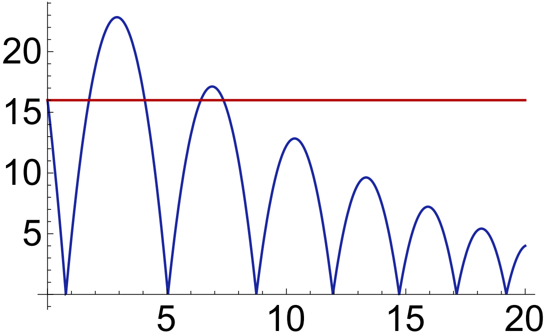
$$1\geq c\geq 0 \land H=x\geq 0 \land g>0 \rightarrow [(\{x'=v, v'=-g \,\&\, x\geq 0\};\ \text{if}(x=0)\ v:=-cv)^*]\ 0\leq x\leq H$$

Quantum the Acrophobic Bouncing Ball



Example (Bouncing Ball)

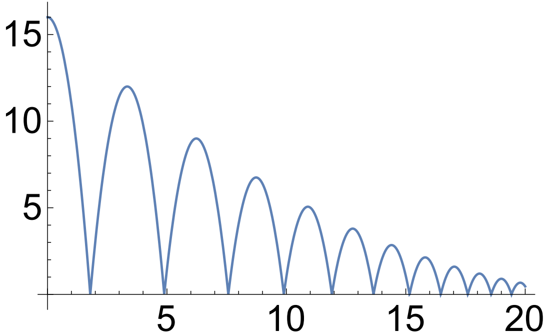
$$v \leq 0 \land 1 \geq c \geq 0 \land H = x \geq 0 \land g > 0 \rightarrow \big[\big(\{x' = v, v' = -g \,\&\, x \geq 0\};\ \text{if}(x = 0)\, v := -cv\big)^*\big]\, 0 \leq x \leq H$$



Not if $v \ll 0$ initial dribbling

Example (Bouncing Ball)

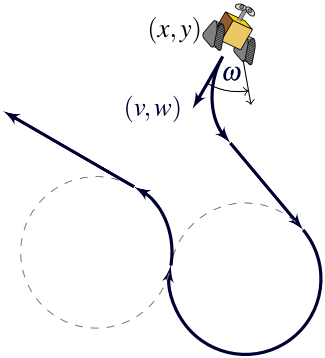
$$v \leq 0 \land 1 \geq c \geq 0 \land H = x \geq 0 \land g > 0 \rightarrow [(\{x' = v, v' = -g \,\&\, x \geq 0\}; \\ \text{if}(x = 0)\, v := -cv)^*]\, 0 \leq x \leq H$$

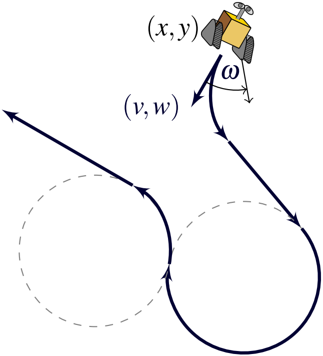


Example (Bouncing Ball)

$$v=0 \land 1\geq c\geq 0 \land H=x\geq 0 \land g>0 \rightarrow \left[\left(\{x' = v, v' = -g \,\&\, x \geq 0\};\right.\right.$$
$$\left.\left.\mathrm{if}(x = 0)\; v := -cv\right)^*\right] 0\leq x\leq H$$

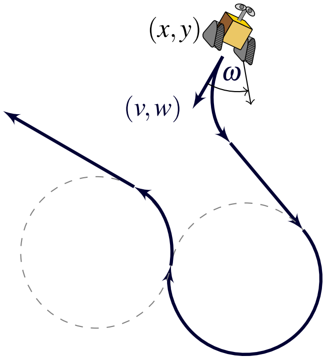
Ex: Runaround Robot





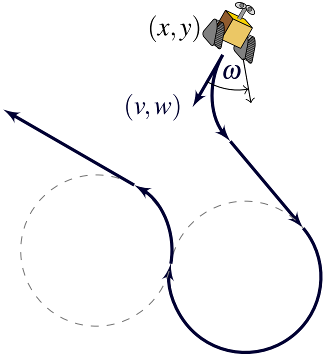
Example (Runaround Robot)

$$\big((\omega := -1 \cup \omega := 1 \cup \omega := 0);$$
$$\{x' = v, y' = w, v' = \omega w, w' = -\omega v\}\big)^*$$



Example (Runaround Robot)

$$(x,y) \neq o \rightarrow \Big[\Big((\omega := -1 \cup \omega := 1 \cup \omega := 0);\\ \{x' = v, y' = w, v' = \omega w, w' = -\omega v\}\Big)^*\Big]\, (x,y) \neq o$$



Example (Runaround Robot)

$$(x,y) \neq o \rightarrow \big[\big(\big(?Q_{-1}; \omega := -1 \cup ?Q_1; \omega := 1 \cup ?Q_0; \omega := 0\big);$$
$$\{x' = v, y' = w, v' = \omega w, w' = -\omega v\}\big)^*\big]\,(x,y) \neq o$$

Outline

Differential Dynamic Logic dL: Syntax

Definition (Hybrid program α)

$$\alpha, \beta \ ::= \ x := f(x) \mid ?Q \mid x' = f(x) \,\&\, Q \mid \alpha \cup \beta \mid \alpha ; \beta \mid \alpha^*$$

Definition (dL Formula P)

$$P, Q \ ::= \ e \geq \tilde{e} \mid \neg P \mid P \wedge Q \mid \forall x\, P \mid \exists x\, P \mid [\alpha]P \mid \langle \alpha \rangle P$$

Differential Dynamic Logic dL: Syntax

Discrete Assign | Test Condition | Differential Equation | Nondet. Choice | Seq. Compose | Nondet. Repeat

Definition (Hybrid program α)

$$\alpha, \beta \;::=\; x := f(x) \mid ?Q \mid x' = f(x) \,\&\, Q \mid \alpha \cup \beta \mid \alpha ; \beta \mid \alpha^*$$

Definition (dL Formula P)

$$P, Q \;::=\; e \geq \tilde{e} \mid \neg P \mid P \wedge Q \mid \forall x\, P \mid \exists x\, P \mid [\alpha] P \mid \langle \alpha \rangle P$$

All Reals | Some Reals | All Runs | Some Runs

Differential Dynamic Logic dL: Semantics

Definition (Hybrid program semantics) $([\![\cdot]\!] : \mathrm{HP} \to \wp(\mathcal{S} \times \mathcal{S}))$

$$
\begin{aligned}
[\![x := f(x)]\!] &= \{(\omega,\nu) : \nu = \omega \text{ except } \nu[\![x]\!] = \omega[\![f(x)]\!]\} \\
[\![?Q]\!] &= \{(\omega,\omega) : \omega \in [\![Q]\!]\} \\
[\![x' = f(x)]\!] &= \{(\varphi(0),\varphi(r)) : \varphi \models x' = f(x) \text{ for some duration } r\} \\
[\![\alpha \cup \beta]\!] &= [\![\alpha]\!] \cup [\![\beta]\!] \\
[\![\alpha;\beta]\!] &= [\![\alpha]\!] \circ [\![\beta]\!] \\
[\![\alpha^*]\!] &= [\![\alpha]\!]^* = \bigcup_{n\in\mathbb{N}} [\![\alpha^n]\!]
\end{aligned}
$$

compositional semantics

Definition (dL semantics) $([\![\cdot]\!] : \mathrm{Fml} \to \wp(\mathcal{S}))$

$$
\begin{aligned}
[\![e \geq \tilde{e}]\!] &= \{\omega : \omega[\![e]\!] \geq \omega[\![\tilde{e}]\!]\} \\
[\![\neg P]\!] &= [\![P]\!]^{\complement} \\
[\![P \land Q]\!] &= [\![P]\!] \cap [\![Q]\!] \\
[\![\langle\alpha\rangle P]\!] &= [\![\alpha]\!] \circ [\![P]\!] = \{\omega : \nu \in [\![P]\!] \text{ for some } \nu : (\omega,\nu) \in [\![\alpha]\!]\} \\
[\![[\alpha]P]\!] &= [\![\neg\langle\alpha\rangle\neg P]\!] = \{\omega : \nu \in [\![P]\!] \text{ for all } \nu : (\omega,\nu) \in [\![\alpha]\!]\} \\
[\![\exists x\, P]\!] &= \{\omega : \omega_x^r \in [\![P]\!] \text{ for some } r \in \mathbb{R}\}
\end{aligned}
$$

André Platzer.
Logical Foundations of Cyber-Physical Systems.
Springer, Switzerland, 2018.
URL: http://www.springer.com/978-3-319-63587-3,
doi:10.1007/978-3-319-63588-0.

André Platzer.
Logical Analysis of Hybrid Systems: Proving Theorems for Complex Dynamics.
Springer, Heidelberg, 2010.
doi:10.1007/978-3-642-14509-4.

André Platzer.
Logics of dynamical systems.
In *LICS*, pages 13–24, Los Alamitos, 2012. IEEE.
doi:10.1109/LICS.2012.13.

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

doi:10.1007/s10817-008-9103-8.

André Platzer.
A complete uniform substitution calculus for differential dynamic logic.
J. Autom. Reas., 59(2):219–265, 2017.
doi:10.1007/s10817-016-9385-1.