Dynamic Logic for Dynamical Systems

André Platzer

Carnegie Mellon University

Summer School Marktoberdorf 2017
1. CPS are Multi-Dynamical Systems
   - Hybrid Systems / Games / Stochastic / Distributed Hybrid Systems

2. Differential Dynamic Logic
   - Syntax
   - Semantics
   - Example: Car Control Design

3. Dynamic Axioms for Dynamical Systems
   - Axiomatics
   - Example: Safe Car Control
   - Soundness and Completeness

4. Differential Invariants for Differential Equations
   - Differential Invariants
   - Example: Elementary Differential Invariants
   - Differential Axioms
   - Example: Differential Ghosts

5. Applications

6. Summary
Outline (Introduction to CPS)

1. CPS are Multi-Dynamical Systems
   - Hybrid Systems / Games / Stochastic / Distributed Hybrid Systems

2. Differential Dynamic Logic
   - Syntax
   - Semantics
   - Example: Car Control Design

3. Dynamic Axioms for Dynamical Systems
   - Axiomatics
   - Example: Safe Car Control
   - Soundness and Completeness

4. Differential Invariants for Differential Equations
   - Differential Invariants
   - Example: Elementary Differential Invariants
   - Differential Axioms
   - Example: Differential Ghosts

5. Applications

6. Summary
Which control decisions are safe for aircraft collision avoidance?

Cyber-Physical Systems

CPSs combine cyber capabilities with physical capabilities to solve problems that neither part could solve alone.
CPSs Promise Transformative Impact!

Prospects: Safe & Efficient

- Driver assistance
  - Autonomous cars
- Pilot decision support
  - Autopilots / UAVs
- Train protection
  - Robots near humans

Prerequisite: CPSs need to be safe

How do we make sure CPSs make the world a better place?
Can you trust a computer to control physics?

Depends on how it has been programmed

And on what will happen if it malfunctions

Rationale

Safety guarantees require analytic foundations.

A common foundational core helps all application domains.

Foundations revolutionized digital computer science & our society.

Need even stronger foundations when software reaches out into our physical world.

CPSs deserve proofs as safety evidence!

André Platzer (CMU)
Dynamic Logic for Dynamical Systems
MOD’17 4 / 59
Can you trust a computer to control physics?

1. Depends on how it has been programmed
2. And on what will happen if it malfunctions

Rationale

1. Safety guarantees require analytic foundations.
2. A common foundational core helps all application domains.
3. Foundations revolutionized digital computer science & our society.
4. Need even stronger foundations when software reaches out into our physical world.

CPSs deserve proofs as safety evidence!
CPSs are Multi-Dynamical Systems

CPS Dynamics
CPS are characterized by multiple facets of dynamical systems.

CPS Compositions
CPS combines multiple simple dynamical effects.

Descriptive simplification

Tame Parts
Exploiting compositionality tames CPS complexity.

Analytic simplification

Andrè Platzer (CMU)
CPSs are Multi-Dynamical Systems

Hybrid systems

\[ HS = \text{discrete} + \text{ODE} \]

Hybrid games

\[ HG = HS + \text{adversary} \]

Stochastic hybrid systems

\[ SHS = HS + \text{stochastics} \]

Distributed hybrid systems

\[ DHS = HS + \text{distributed} \]
Outline (Modeling CPS)

1. CPS are Multi-Dynamical Systems
   - Hybrid Systems / Games / Stochastic / Distributed Hybrid Systems

2. Differential Dynamic Logic
   - Syntax
   - Semantics
   - Example: Car Control Design

3. Dynamic Axioms for Dynamical Systems
   - Axiomatics
   - Example: Safe Car Control
   - Soundness and Completeness

4. Differential Invariants for Differential Equations
   - Differential Invariants
   - Example: Elementary Differential Invariants
   - Differential Axioms
   - Example: Differential Ghosts

5. Applications

6. Summary
Dynamic Logics for Dynamical Systems

**Differential Dynamic Logic**

\[ dL = DL + HP \]

**Differential Game Logic**

\[ dGL = GL + HG \]

**Stochastic Differential Dynamic Logic**

\[ SdL = DL + SHP \]

**Quantified Differential Dynamic Logic**

\[ QdL = FOL + DL + QHP \]

JAR'08, CADE'11, LMCS'12, LICS'12, LICS'12

André Platzer (CMU)
Dynamic Logics for Dynamical Systems

- DL has been introduced for programs
  Pratt’76,Harel,Kozen
- Its real calling are dynamical systems
- DL excels at providing simple+elegant logical foundations for dynamical systems
- CPSs are multi-dynamical systems
- DL for CPS are multi-dynamical
**Syntax** defines the notation
What problems are we allowed to write down?

**Semantics** what carries meaning.
What real or mathematical objects does the syntax stand for?

**Axiomatics** internalizes semantic relations into universal syntactic transformations.
How does the semantics of $A$ relate to semantics of $A \land B$, syntactically? If $A$ is true, is $A \land B$ true, too? Conversely?
Concept (Differential Dynamic Logic)  

\([\alpha] \varphi \quad \varphi\)  

André Platzer (CMU)  
Dynamic Logic for Dynamical Systems  
MOD'17 9 / 59
CPS Analysis

Concept (Differential Dynamic Logic) (JAR’08, LICS’12)

\[ [\alpha] \varphi \rightarrow \varphi \]

\[ x \neq m \]

\( x \neq m \land b > 0 \)
Concept (Differential Dynamic Logic) (JAR’08, LICS’12)

\[ [\alpha] \varphi \rightarrow \varphi \]

\[ \Box x \neq m \rightarrow x \neq m \]

\[ x \neq m \]

\[ \alpha \]

\[ \begin{array}{c}
\begin{array}{cccccccc}
a & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
\hline
0.5 & & & & & & & \\
0.0 & & & & & & & \\
-0.5 & & & & & & & \\
-1.0 & & & & & & & \\
-1.5 & & & & & & & \\
-2.0 & & & & & & & \\
-2.5 & & & & & & &
\end{array}
\end{array} \]

\[ \begin{array}{c}
\begin{array}{cccccccc}
v & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
\hline
0 & & & & & & & \\
2 & & & & & & & \\
4 & & & & & & & \\
6 & & & & & & & \\
8 & & & & & & & \\
10 & & & & & & &
\end{array}
\end{array} \]

\[ \begin{array}{c}
\begin{array}{cccccccc}
x & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
\hline
0 & & & & & & & \\
2 & & & & & & & \\
4 & & & & & & & \\
6 & & & & & & & \\
8 & & & & & & & \\
10 & & & & & & &
\end{array}
\end{array} \]
CPS Analysis

Concept (Differential Dynamic Logic) (JAR’08, LICS’12)

\[ [\alpha] \varphi \]

\[ x \neq m \]

\[ [\alpha] x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]
CPS Analysis

Concept (Differential Dynamic Logic) (JAR'08, LICS'12)

\[ [\alpha] \varphi \]

\[ x \neq m \]

\[ x' = v, v' = a \]

ODE

\begin{align*}
  x(t) &\rightarrow [\text{init}] \quad [\text{if } SB(x,m) \quad a := -b] \quad [\text{f}]
\end{align*}
CPS Analysis

Concept (Differential Dynamic Logic) (JAR’08, LICS’12)

\[ [\alpha] \varphi \]

\[ x \neq m \]

\[ a := -b \]

\[ x' = v, v' = a \]

\[ x(t) \]

\[ a(t) \]

\[ \varphi \]

\[ \alpha \]

\[ \text{ODE assign} \]
Concept (Differential Dynamic Logic) (JAR’08, LICS’12)

\[ [\alpha] \varphi \quad \varphi \]

\[ x \neq m \]

\[ \text{if}(\text{SB}(x, m)) \quad a := -b \]

\[ x' = v, v' = a \]

\[ \text{test} \quad \text{assign} \]

\[ \text{ODE} \]

André Platzer (CMU)
Dynamic Logic for Dynamical Systems
Concept (Differential Dynamic Logic) (JAR’08, LICS’12)

\[ [\alpha] \varphi \quad \alpha \rightarrow \varphi \]

(seq. compose)

\[(\text{if}(SB(x, m)) \quad a := -b) \quad ; \quad x' = v, v' = a\]

(test assign)

ODE

André Platzer (CMU)
Dynamic Logic for Dynamical Systems
MOD’17 9 / 59
CPS Analysis

Concept (Differential Dynamic Logic) (JAR’08, LICS’12)

\[ [\alpha] \varphi \quad \varphi \]

\((\text{if}(\text{SB}(x, m)) \ a := -b) \ ; \ x' = v, v' = a)^*\)

- seq. compose
- nondet. repeat
- test
- assign
- ODE

\(x\)

\(a\)

\(m\)
Concept (Differential Dynamic Logic): $\left[\alpha\right] \phi \vdash \phi$

$(JAR'08, LICS'12)$

$\left[\alpha\right] x \neq m \quad x \neq m \quad x \neq m$

$\left[\alpha\right] \left( (\text{if}(SB(x, m)) \quad a := -b) ; \quad x' = v, v' = a \right)^* x \neq m$

$\alpha$

$v$

$m$

$x$

André Platzer (CMU)
CPS Analysis

Concept (Differential Dynamic Logic) (JAR’08, LICS’12)

\[ [\alpha] \phi \quad \alpha \quad \phi \]

\[
\begin{align*}
x \neq m \land b > 0 \rightarrow \left[ \left( (\text{if}(SB(x, m)) \ a := -b) ; \ x' = v, v' = a \right)^* \right] x \neq m
\end{align*}
\]

André Platzer (CMU)  Dynamic Logic for Dynamical Systems  MOD’17 9 / 59
CPS Analysis

Concept (Differential Dynamic Logic) (JAR’08, LICS’12)

\[ [\alpha] \varphi \]

\[ x \neq m \]

\[ \text{nondet. choice} \]

\[ x \neq m \land b > 0 \rightarrow [(\neg \text{SB}(x, m) \cup a := -b) ; x' = v, v' = a)^*] x \neq m \]

\[ \text{init} \]

\[ \text{post} \]

\[ a \]

\[ v \]

\[ x \]

\[ m \]
CPS Analysis

Concept (Differential Dynamic Logic) (JAR’08, LICS’12)

\[ [\alpha]\varphi \longrightarrow \varphi \]

\[ x \neq m \land b > 0 \rightarrow [((\neg \text{SB}(x, m) \cup a := -b) \land x' = v, v' = a)^*] \]

\[ x \neq m \]

\[ \text{test} \quad \text{nondet. choice} \]

\[ t \quad a \]

\[ t \quad v \]

\[ t \quad x \]

\[ m \]
CPS Analysis

Concept (Differential Dynamic Logic) (JAR’08, LICS’12)

\[
\text{init} : \neg \text{SB}(x, m) \cup a := -b ; \ x' = v, v' = a)^* \quad \text{post}
\]

\[
x \neq m 
\]

Hybrid program dynamics

Andre Platzer (CMU)
Differential Dynamic Logic dL: Syntax

Definition (Hybrid program $\alpha$)

\[
x := f(x) \mid ?Q \mid x' = f(x) \& Q \mid \alpha \cup \beta \mid \alpha;\beta \mid \alpha^\ast
\]

Definition (dL Formula $P$)

\[
e \geq \tilde{e} \mid \neg P \mid P \land Q \mid \forall x \ P \mid \exists x \ P \mid [\alpha]P \mid \langle \alpha \rangle P
\]
Differential Dynamic Logic dL: Syntax

Definition (Hybrid program $\alpha$)

\[ x := f(x) | ?Q | x' = f(x) & Q | \alpha \cup \beta | \alpha;\beta | \alpha^* \]

Definition (dL Formula $P$)

\[ e \geq \tilde{e} | \neg P | P \land Q | \forall x\ P | \exists x\ P | [\alpha]P | \langle \alpha \rangle P \]


JAR'08, LICS'12, JAR'17

André Platzer (CMU) Dynamic Logic for Dynamical Systems MOD'17 10 / 59
**Definition (Hybrid program semantics)** 

\[ [x := e] = \{ (\omega, \nu) : \nu = \omega \text{ except } \nu[x] = \omega[e] \} \]

\[ [?Q] = \{ (\omega, \omega) : \omega \in [Q] \} \]

\[ [x' = f(x)] = \{ (\varphi(0), \varphi(r)) : \varphi \models x' = f(x) \text{ for some duration } r \} \]

\[ [\alpha \cup \beta] = [\alpha] \cup [\beta] \]

\[ [\alpha; \beta] = [\alpha] \circ [\beta] \]

\[ [\alpha^*] = \bigcup_{n \in \mathbb{N}} [\alpha^n] \]

**Definition (dL semantics)** 

\[ [e \geq \tilde{e}] = \{ \omega : \omega[e] \geq \omega[\tilde{e}] \} \]

\[ [\neg P] = [P]^C \]

\[ [P \land Q] = [P] \cap [Q] \]

\[ [\langle \alpha \rangle P] = [\alpha] \circ [P] = \{ \omega : \nu \in [P] \text{ for some } \nu : (\omega, \nu) \in [\alpha] \} \]

\[ [[\alpha] P] = [\neg \langle \alpha \rangle \neg P] = \{ \omega : \nu \in [P] \text{ for all } \nu : (\omega, \nu) \in [\alpha] \} \]

\[ [\exists x P] = \{ \omega : \omega^r_x \in [P] \text{ for some } r \in \mathbb{R} \} \]

**compositional semantics**
Differential Dynamic Logic dL: Semantics

$$\omega \xrightarrow{x := e} \nu$$

$$\omega \xrightarrow{x' = f(x) \& Q} \nu$$

$$\omega \xrightarrow{?Q} \nu$$ if $$\omega \in \llbracket Q \rrbracket$$

$$\omega \xrightarrow{0 \rightarrow t}$$

$$\nu$$ if $$\nu(x) = \omega[e]$$ and $$\nu(z) = \omega(z)$$ for $$z \neq x$$

$$\omega$$ no change if $$\omega \in \llbracket Q \rrbracket$$

otherwise no transition
\[ x := e \]

\[ x' = f(x) \& Q \]

\[ ?Q \text{ if } \omega \in [Q] \]

\[ \omega \text{ no change if } \omega \in [Q] \text{ otherwise no transition} \]
Differential Dynamic Logic $dL$: Semantics

- $x := e$
- $x' = f(x) \& Q$
- $?Q$ if $\omega \in [Q]$
Differential Dynamic Logic $dL$: Semantics

$\omega \xrightarrow{x := e} \nu$

$\omega \xrightarrow{x' = f(x) \& Q} \nu$

?Q

if $\omega \in [Q]$}

$\omega \xrightarrow{\text{no change if } \omega \in [Q]} \nu$

$x \xrightarrow{\nu \text{ if } \nu(x) = \omega[e] \text{ and } \nu(z) = \omega(z) \text{ for } z \neq x} t$

$x \xrightarrow{\omega \text{ no change if } \omega \in [Q]} t$

$x' = f(x) \& Q$

$\nu$

$\omega$

$0$

$0$

$\omega$

$\nu$

$0$

$t$

$t$

$r$

$Q$

$\omega$

$\nu$

$\omega$

$0$

$t$

$\omega$

$\nu$

$\omega$

$0$

$t$
Differential Dynamic Logic dL: Semantics

\[ \omega \xrightarrow{\alpha} \nu_1 \xrightarrow{\alpha \cup \beta} \nu_2 \]

\[ \omega \xrightarrow{\alpha} \mu \xrightarrow{\beta} \nu \]

\[ \omega \xrightarrow{\alpha} \omega_1 \xrightarrow{\alpha} \omega_2 \xrightarrow{\alpha} \nu \]

\[ \omega \xrightarrow{\alpha^*} \nu \]

\[ x \]

\[ x \]

\[ x \]

\[ x \]

\[ t \]

\[ t \]

\[ t \]

André Platzer (CMU)
Differential Dynamic Logic $dL$: Semantics

$\omega \xrightarrow{\alpha} \nu_1 \xrightarrow{\beta} \nu_2$

$\omega \xrightarrow{\alpha} \mu \xrightarrow{\beta} \nu$

$\nu \xrightarrow{\alpha^*} \omega_1 \xrightarrow{\alpha} \omega_2 \xrightarrow{\alpha} \nu$

$\omega \xrightarrow{\alpha} \nu_1 \xrightarrow{\alpha^*} \nu_2 \xrightarrow{\alpha} \nu$

$\omega \xrightarrow{\alpha} \mu \xrightarrow{\beta} \nu$

$\omega \xrightarrow{\alpha} \nu_1 \xrightarrow{\alpha} \nu_2 \xrightarrow{\alpha} \nu$
Differential Dynamic Logic $dL$: Semantics

\[ \omega \xrightarrow{\alpha} \nu_1 \xrightarrow{\alpha \cup \beta} \nu_2 \]

\[ \omega \xrightarrow{\alpha} \mu \xrightarrow{\beta} \nu \]

\[ (\alpha; \beta)^* \]

\[ \omega \xrightarrow{\alpha} \omega_1 \xrightarrow{\alpha} \omega_2 \xrightarrow{\alpha} \nu \]

\[ \omega \xrightarrow{s} \nu \]

\[ \omega \xrightarrow{\alpha} \nu \]

\[ \nu_1 \]

\[ \nu_2 \]

\[ t \]
Definition (dL Formulas)

\[ \omega \]

\[ [\alpha]P \]

\[ \omega \text{ composes} \Rightarrow \text{compositional proofs!} \]
Definition (dL Formulas)

&omega; \langle \alpha \rangle P \Rightarrow \langle \beta \rangle P \beta \text{-span} \[ \alpha \text{-span} \] compositional semantics \Rightarrow \text{compositional proofs!}
Definition (dL Formulas)

\[ [\alpha]P \triangleq \left\{ \omega \mid \omega \text{-span} \right\} \]
Definition (dL Formulas)

\[ [\alpha]P \]

\[ \langle \beta \rangle P \]

\( \omega \)

\( \beta \)-span

\( \alpha \)-span

André Platzer (CMU)
Definition (dL Formulas)

\[ \omega^{\langle \alpha \rangle} P \]

- \( \langle \beta \rangle P \)
- \( [\alpha] P \)
- \( \beta \)-span
- \( \alpha \)-span

- \( \langle \beta \rangle^{\beta-\text{span}} \)
- \( [\alpha]^{\alpha-\text{span}} \)
Definition (dL Formulas)

\[
\omega \quad \langle \beta \rangle P \quad [\alpha] P \quad \beta\text{-span} \quad \langle \beta \rangle P \quad \omega \quad [\alpha] P \quad \beta\text{-span} \quad \langle \beta \rangle P \quad \omega \quad [\alpha] P \quad \beta\text{-span} \quad \langle \beta \rangle P \quad \omega \quad [\alpha] P \quad \beta\text{-span} \quad \langle \beta \rangle P \quad \omega \quad [\alpha] P \quad \beta\text{-span} \quad \langle \beta \rangle P \quad \omega \quad [\alpha] P \quad \beta\text{-span} \quad \langle \beta \rangle P \quad \omega \quad [\alpha] P \quad \beta\text{-span} \quad \langle \beta \rangle P \quad \omega \quad [\alpha] P \quad \beta\text{-span} \quad \langle \beta \rangle P \quad \omega \quad [\alpha] P \quad \beta\text{-span} \quad \langle \beta \rangle P \quad \omega \quad [\alpha] P \quad \beta\text{-span} \quad \langle \beta \rangle P \quad \omega \quad [\alpha] P \quad \beta\text{-span} \quad \langle \beta \rangle P \quad \omega \quad [\alpha] P \quad \beta\text{-span} \quad \langle \beta \rangle P
\]

compositional semantics \implies \text{compositional proofs!}
Ex: Car Control Programs

Repeat control decisions

Example (Single car \( car_s \))

\[
(( a \leftarrow A \cup a \leftarrow -b); \ x' = v, \ v' = a)^*
\]
Ex: Car Control Programs

How does this model brake?

Example (Single car $\text{car}_s$)

$((a := A \cup a := -b); x' = v, v' = a)^*$
Ex: Car Control Programs

Velocity bound $v \geq 0$ in evolution domain

Example (Single car $car_s$)

$$((a := A \cup a := -b); x' = v, v' = a \& v \geq 0)^*$$

![Graphs showing time evolution of variables a, v, x, m over time t]
Ex: Car Control Programs

Acceleration not always safe

Example (Single car \( \text{cars} \))

\[
((a := A \cup a := -b); \ x' = v, \ v' = a \& v \geq 0)^*
\]
Ex: Car Control Programs

Acceleration condition \(? Q\)

Example (Single car \(c_{ars}\))

\[
(((? Q; a := A) \cup a := -b); \ x' = v, v' = a \& v \geq 0)^
\]
Ex: Car Control Properties

Example (Single car $car_\varepsilon$ time-triggered)

\[
Q \equiv ((?Q; a := A) \cup a := -b); \ t := 0; \ x' = v, v' = a, t' = 1 \& v \geq 0 \land t \leq \varepsilon)^*
\]

Example (Safely stays before traffic light $m$)

\[
A \geq 0 \land b > 0 \rightarrow [car_\varepsilon] x \leq m
\]
\[ Q \equiv 2b(m - x) \geq v^2 + (A + b)(A\varepsilon^2 + 2\varepsilon v) \]

Example (Single car \(car_\varepsilon\) time-triggered)

\[
(((?Q; a := A) \cup a := -b); \ t := 0; \ x' = v, v' = a, t' = 1 \& v \geq 0 \& t \leq \varepsilon)^* \]

Example (\(\square\) Safely stays before traffic light \(m\))

\[ v^2 \leq 2b(m - x) \land A \geq 0 \land b > 0 \rightarrow [car_\varepsilon] x \leq m \]
\[ Q \equiv 2b(m - x) \geq v^2 + (A + b)(A\varepsilon^2 + 2\varepsilon v) \]

Example (Single car \( \text{car}_\varepsilon \) time-triggered)

\[
((((?Q; a := A) \cup a := -b); \ t := 0; \ x' = v, v' = a, t' = 1 & v \geq 0 \land t \leq \varepsilon))^*\]

Example (Live, can move everywhere)

\[
\varepsilon > 0 \land A > 0 \land b > 0 \rightarrow \forall p \exists m \langle \text{car}_\varepsilon \rangle x \geq p
\]
car \equiv (\text{ctrl} ; \text{drive})^*

\text{ctrl} \equiv (?Q_A; a := A)
\quad \cup (?Q_b; a := -b)

\text{drive} \equiv t := 0; \{x' = v, v' = a, t' = 1 \land v \geq 0 \land t \leq \varepsilon\}
Branching Transition Structure in Hybrid Programs

\[
\text{car} \equiv (\text{ctrl} ; \text{drive})^*
\]
\[
\text{ctrl} \equiv (?Q_A; a := A)
\]
\[
\quad \cup (?Q_b; a := -b)
\]
\[
\text{drive} \equiv t := 0; \{x' = v, v' = a, t' = 1 \& v \geq 0 \land t \leq \varepsilon\}
\]
Branching Transition Structure in Hybrid Programs

\[
\text{car} \equiv (\text{ctrl} ; \text{drive})^*
\]

\[
\text{ctrl} \equiv (?Q_A; a := A) \cup (?Q_b; a := -b)
\]

\[
\text{drive} \equiv t := 0; \{x' = v, v' = a, t' = 1 \land v \geq 0 \land t \leq \varepsilon\}
\]
Branching Transition Structure in Hybrid Programs

\[
\begin{align*}
\text{car} & \equiv (\text{ctrl} ; \text{drive})^* \\
\text{ctrl} & \equiv (?Q_A; a := A) \\
& \quad \cup (?Q_b; a := -b) \\
\text{drive} & \equiv t := 0; \{x' = v, v' = a, t' = 1 \land v \geq 0 \land t \leq \varepsilon\}
\end{align*}
\]

\[
\begin{align*}
& \quad x'' = a \\
& \quad t' = 1 \\
& \quad \land v \geq 0 \land t \leq \varepsilon
\end{align*}
\]
\[ \text{car} \equiv (\text{ctrl} ; \text{drive})^* \]
\[ \text{ctrl} \equiv (?Q_A; a := A) \]
\[ \cup (?Q_b; a := -b) \]
\[ \text{drive} \equiv t := 0; \{ x' = v, v' = a, t' = 1 \land v \geq 0 \land t \leq \varepsilon \} \]
Branching Transition Structure in Hybrid Programs

\[ ?Q_A \quad a := A \quad t := 0 \quad x'' = a \quad t' = 1 \quad \& \quad v \geq 0 \land t \leq \varepsilon \]

\[ a := -b \]

\[ ?Q_b \]

\[ car \equiv (ctrl ; drive)^* \]

\[ ctrl \equiv (?Q_A; a := A) \quad \cup \quad (?Q_b; a := -b) \]

\[ drive \equiv t := 0; \{x' = v, v' = a, t' = 1 \& v \geq 0 \land t \leq \varepsilon\} \]
Branching Transition Structure in Hybrid Programs

car ≡ (ctrl ; drive)*

ctrl ≡ (?QA; a := A) ∪ (?Qb; a := −b)

drive ≡ t := 0; \{x' = v, v' = a, t' = 1 & v \geq 0 \land t \leq \varepsilon\}
car ≡ (ctrl ; drive)*
ctrl ≡ (?QA; a := A)
  ∪ (?Qb; a := −b)

drive ≡ t := 0; \{x' = v, v' = a, t' = 1 \& v \geq 0 \land t \leq \varepsilon\}
car $\equiv (\text{ctrl} ; \text{drive})^*$

ctrl $\equiv (?Q_A; a := A) \\
\cup (?Q_b; a := -b)$

drive $\equiv t := 0; \{x' = v, v' = a, t' = 1 & v \geq 0 \land t \leq \varepsilon\}$
car $\equiv (\text{ctrl} ; \text{drive})^*$

ctrl $\equiv (?Q_A; a := A)$

$\cup (?Q_b; a := -b)$

drive $\equiv t := 0; \{x' = v, v' = a, t' = 1 & v \geq 0 \land t \leq \varepsilon\}$
Branching Transition Structure in Hybrid Programs

\[
\text{car} \equiv (\text{ctrl} ; \text{drive})^* \\
\text{ctrl} \equiv (\lnot Q_A ; a := A) \\
\quad \cup (\lnot Q_b ; a := -b) \\
\text{drive} \equiv t := 0; \{x' = v, v' = a, t' = 1 & v \geq 0 \land t \leq \varepsilon\} 
\]
Branching Transition Structure in Hybrid Programs

\[
\text{car} \equiv (\text{ctrl} ; \text{drive})^* \\
\text{ctrl} \equiv (\?Q_A; \; a := A) \\
\quad \cup (\?Q_b; \; a := -b) \\
\text{drive} \equiv t := 0; \{x' = v, v' = a, t' = 1 & v \geq 0 \land t \leq \varepsilon\} 
\]
car ≡ (ctrl ; drive)∗

ctrl ≡ (?QA; a := A)
∪ (?Qb; a := −b)

drive ≡ t := 0; \{x′ = v, v′ = a, t′ = 1 \& v ≥ 0 \land t ≤ ε\}
Branching Transition Structure in Hybrid Programs

\[
\begin{align*}
\text{car} & \equiv (\text{ctrl} ; \text{drive})^* \\
\text{ctrl} & \equiv (\neg Q_A; a := A) \cup (\neg Q_b; a := -b) \\
\text{drive} & \equiv t := 0; \{x' = v, v' = a, t' = 1 \land v \geq 0 \land t \leq \varepsilon\}
\end{align*}
\]
car \equiv (\text{ctrl} ; \text{drive})^*

\text{ctrl} \equiv (?Q_A; a := A)

\quad \cup (?Q_b; a := -b)

drive \equiv t := 0; \{x' = v, v' = a, t' = 1 \& v \geq 0 \land t \leq \varepsilon\}
Branching Transition Structure in Hybrid Programs

\[
\begin{align*}
\text{if}(Q) \alpha \text{ else } \beta & \equiv \\
\text{while}(Q) \alpha & \equiv \\
\text{car} & \equiv (\text{ctrl} ; \text{drive})^* \\
\text{ctrl} & \equiv (?Q_A; a := A) \\
& \quad \cup (?Q_b; a := -b) \\
\text{drive} & \equiv t := 0; \{x' = v, v' = a, t' = 1 \& v \geq 0 \land t \leq \varepsilon\}
\end{align*}
\]
If $Q$ then $\alpha$ else $\beta \equiv (\exists Q; \alpha) \cup (\exists \neg Q; \beta)$

while $Q$ $\alpha \equiv$

car $\equiv (\text{ctrl} \cdot \text{drive})^*$

ctrl $\equiv (\exists Q_A; a := A)$

$\cup (\exists Q_b; a := -b)$

drive $\equiv t := 0; \{x' = v, v' = a, t' = 1 & v \geq 0 \land t \leq \varepsilon\}$
Branching Transition Structure in Hybrid Programs

\[
\begin{align*}
\text{if}(Q) \alpha \text{ else } \beta & \equiv (\exists Q; \alpha) \cup (\exists \neg Q; \beta) \\
\text{while}(Q) \alpha & \equiv (\exists Q; \alpha)^*; \exists \neg Q
\end{align*}
\]

car \equiv (\text{ctrl} ; \text{drive})^*

ctrl \equiv (\exists Q_A; a := A) \\
\cup (\exists Q_b; a := -b)

drive \equiv t := 0; \{x' = v, v' = a, t' = 1 \& v \geq 0 \wedge t \leq \varepsilon\}
Example (Bouncing Ball)

\[ x' = v, \ v' = -g \]
Ex: The Acrophobic Bouncing Ball

Example (Bouncing Ball)

\[ x' = v, \quad v' = -g \quad \text{and} \quad x \geq 0 \]
Example (Bouncing Ball)

\[
x' = v, \quad v' = -g \quad \& \quad x \geq 0; \\
\text{if}(x = 0) \quad v := -cv
\]
Example (Bouncing Ball)

\[
(x' = v, \ v' = -g \ & \ x \geq 0; \\
\text{if}(x = 0) \ v := -cv)
\]
Example (Bouncing Ball)

\[
\begin{align*}
(x' &= v, 
\quad v' = -g \& x \geq 0; \\
\text{if}(x = 0) \ v := -cv)
\end{align*}
\]
Example (Bouncing Ball)

\[ x = H \geq 0 \land \ldots \rightarrow \left[ \left( x' = v, v' = -g \land x \geq 0; \right) \text{if}(x = 0) v := -cv \right] 0 \leq x \leq H \]
Ex: Runaround Robot

\[\begin{align*}
Q & \equiv (x + \omega - o x)^2 + (y - \omega - o y)^2 \neq v^2 + w^2 \\
Q_0 & \equiv (o x - x)w \neq (o y - y)v
\end{align*}\]
Example (Runaround Robot)

\[
\left( \left( \omega := -1 \cup \omega := 1 \cup \omega := 0 \right); \quad \{ x' = v, y' = w, v' = \omega w, w' = -\omega v \right) \right) \ast
\]
Example (Runaround Robot)

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

\[(x, y) \neq o \rightarrow [((?Q^{-1}; \omega := -1 \cup ?Q_1; \omega := 1 \cup ?Q_0; \omega := 0); \\
\{x' = v, y' = w, v' = \omega w, w' = -\omega v\})^*] (x, y) \neq o\]
Example (dL-based model-predictive control design)

\[ \wedge v \geq 0 \wedge A \geq 0 \wedge b > 0 \rightarrow \]

\[ (((\?
(?
\wedge a := A)
\cup a := -b);)
t := 0; x' = v, v' = a, t' = 1 & v \geq 0 \wedge t \leq \varepsilon) \) \] \[ x \leq m \]
Example (\(\text{dL-based model-predictive control design}\))

\[ ??? \land v \geq 0 \land A \geq 0 \land b > 0 \rightarrow \]

\[
\left( \left( \right) \cup a := A \right) \\
\cup a := -b; \\
t := 0; \ x' = v, v' = a, t' = 1 \& v \geq 0 \land t \leq \varepsilon \right)^* \]

\[ x \leq m \]
Example (dL-based model-predictive control design)

\[
[x' = v, v' = -b] x \leq m \land v \geq 0 \land A \geq 0 \land b > 0 \rightarrow \\
[((
    (? ;
    a := A)
    \cup a := -b);
    t := 0; x' = v, v' = a, t' = 1 \& v \geq 0 \land t \leq \varepsilon)^*] x \leq m
\]
Example (dL-based model-predictive control design)

\[
\begin{align*}
[x' = v, \ v' = -b]x & \leq m \land v \geq 0 \land A \geq 0 \land b > 0 \rightarrow \\
[( & ((
\begin{align*}
(\ & ???; \\
\ & a := A) \\
\cup a := -b); \\
\ & t := 0; \ x' = v, \ v' = a, \ t' = 1 \& v \geq 0 \land t \leq \varepsilon)^*] \ x \leq m
\end{align*}
\end{align*}
\]
Example (dL-based model-predictive control design)

\[ x' = v, v' = -b \land x \leq m \land v \geq 0 \land A \geq 0 \land b > 0 \rightarrow \]

\[
(((

(?[t := 0; x' = v, v' = A, t' = 1 \& v \geq 0 \land t \leq \varepsilon][x' = v, v' = -b]x \leq m ;

a := A) \\
\cup a := -b);

t := 0; x' = v, v' = a, t' = 1 \& v \geq 0 \land t \leq \varepsilon)^* \) x \leq m \]
Example (dL-based model-predictive control design)

\[
[x' = v, v' = -b] x \leq m \land v \geq 0 \land A \geq 0 \land b > 0 \rightarrow \\
[(([t := 0; x' = v, v' = A, t' = 1 \land v \geq 0 \land t \leq \varepsilon][x' = v, v' = -b] x \leq m ; \\
a := A)] \cup [a := -b]);
\\
t := 0; x' = v, v' = a, t' = 1 \land v \geq 0 \land t \leq \varepsilon)^*] x \leq m
\]
**Example (DL-based model-predictive control design)**

\[
v^2 \leq 2b(m - x) \land v \geq 0 \land A \geq 0 \land b > 0 \rightarrow \\
[([([]
(?[t := 0; x' = v, v' = A, t' = 1 \land v \geq 0 \land t \leq \varepsilon)] x' = v, v' = -b)] x \leq m ; \\
\text{a} := A)
\cup \text{a} := -b);
\]

\[t := 0; x' = v, v' = a, t' = 1 \land v \geq 0 \land t \leq \varepsilon)^* \] \( x \leq m \)
Example (dL-based model-predictive control design)

\[ v^2 \leq 2b(m-x) \land v \geq 0 \land A \geq 0 \land b > 0 \rightarrow \]

\[
(\exists [t := 0; x' = v, v' = A, t' = 1 \land v \geq 0 \land t \leq \varepsilon] [x' = v, v' = -b] x \leq m;\]

\[ a := A \]

\[ \cup a := -b); \]

\[ t := 0; x' = v, v' = a, t' = 1 \land v \geq 0 \land t \leq \varepsilon) \]

\[ x \leq m \]
Example (dL-based model-predictive control design)

\[ \begin{align*}
\nu^2 & \leq 2b(m - x) \land \nu \geq 0 \land A \geq 0 \land b > 0 \rightarrow \\
& \left( \left( ?2b(m - x) \geq \nu^2 + (A + b)(A\varepsilon^2 + 2\varepsilon\nu) \right) ; \\
& a := A \right) \\
& \cup a := -b \right) ; \\
& t := 0 ; \ x' = \nu , \ \nu' = a , \ t' = 1 \ & \nu \geq 0 \land t \leq \varepsilon \right) ^* \right) x \leq m
\end{align*} \]
Outline (Proving CPS)

1. CPS are Multi-Dynamical Systems
   - Hybrid Systems / Games / Stochastic / Distributed Hybrid Systems

2. Differential Dynamic Logic
   - Syntax
   - Semantics
   - Example: Car Control Design

3. Dynamic Axioms for Dynamical Systems
   - Axiomatics
   - Example: Safe Car Control
   - Soundness and Completeness

4. Differential Invariants for Differential Equations
   - Differential Invariants
   - Example: Elementary Differential Invariants
   - Differential Axioms
   - Example: Differential Ghosts

5. Applications

6. Summary
Differential Dynamic Logic: Axiomatization

\[ := \] \([x := e] P(x) \leftrightarrow P(e)\]

\[ ? \] \([? Q] P \leftrightarrow (Q \to P)\]

\[ ' \] \([x' = f(x)] P \leftrightarrow \forall t \geq 0 [x := y(t)] P\]

\[ \cup \] \([\alpha \cup \beta] P \leftrightarrow [\alpha] P \land [\beta] P\]

\[ ; \] \([\alpha; \beta] P \leftrightarrow [\alpha][\beta] P\]

\[ * \] \([\alpha^*] P \leftrightarrow P \land [\alpha][\alpha^*] P\]

\[ K \] \([\alpha](P \to Q) \to ([\alpha] P \to [\alpha] Q)\]

\[ I \] \([\alpha^*](P \to [\alpha] P) \to (P \to [\alpha^*] P)\]

\[ C \] \([\alpha^*]\forall v > 0 (P(v) \to \langle \alpha \rangle P(v-1)) \to \forall v (P(v) \to \langle \alpha^* \rangle \exists v \leq 0 P(v))\]

equations of truth

LICS'12, JAR'17
\[ := \quad [x := e] p(x) \leftrightarrow \]
Dynamic Axioms for Dynamical Systems

\[ \text{[:=]} \quad [x := e]p(x) \leftrightarrow p(e) \]
Dynamic Axioms for Dynamical Systems

\[
[\implies] \quad [x := e]p(x) \leftrightarrow p(e)
\]

\[
\labeled{\prime} [x' = f(x)]p(x) \leftrightarrow
\]

\[
\omega \quad \rightarrow \quad \nu
\]

\[
\begin{array}{c}
\omega \\
p(e)
\end{array} \quad \rightarrow \quad \begin{array}{c}
\nu \\
p(x)
\end{array}
\]

\[
\omega \quad \rightarrow \quad \nu
\]

\[
\begin{array}{c}
\omega \\
x := e
\end{array} \quad \rightarrow \quad \begin{array}{c}
\nu \\
p(x)
\end{array}
\]

\[
\omega \quad \rightarrow \quad \nu
\]

\[
\begin{array}{c}
\omega \\
x' = f(x)
\end{array} \quad \rightarrow \quad \begin{array}{c}
\nu \\
p(x)
\end{array}
\]

\[
\omega \quad \rightarrow \quad \nu
\]

\[
\begin{array}{c}
\omega \\
x := e
\end{array} \quad \rightarrow \quad \begin{array}{c}
\nu \\
p(x)
\end{array}
\]

\[
\omega \quad \rightarrow \quad \nu
\]

\[
\begin{array}{c}
\omega \\
x' = f(x)
\end{array} \quad \rightarrow \quad \begin{array}{c}
\nu \\
p(x)
\end{array}
\]

\[
\omega \quad \rightarrow \quad \nu
\]

\[
\begin{array}{c}
\omega \\
x := e
\end{array} \quad \rightarrow \quad \begin{array}{c}
\nu \\
p(x)
\end{array}
\]

\[
\omega \quad \rightarrow \quad \nu
\]

\[
\begin{array}{c}
\omega \\
x' = f(x)
\end{array} \quad \rightarrow \quad \begin{array}{c}
\nu \\
p(x)
\end{array}
\]
Dynamic Axioms for Dynamical Systems

$$[\text{:=}] \ [x := e]p(x) \leftrightarrow p(e)$$

$$\left[ ' \right] \ [x' = f(x)]p(x) \leftrightarrow [x := y(t)]p(x)$$
Dynamic Axioms for Dynamical Systems

\[ [\_] \quad [x := e]p(x) \leftrightarrow p(e) \]

\[ ['] \quad [x' = f(x)]p(x) \leftrightarrow \forall t \geq 0 [x := y(t)]p(x) \]
Dynamic Axioms for Dynamical Systems

\[[=] \ [x := e]p(x) \leftrightarrow p(e)\]

\[\overset{'}{[x' = f(x)]p(x)} \leftrightarrow \forall t \geq 0 [x := y(t)]p(x)\]

\[\overset{'}{[x' = f(x) & q(x)]p(x)} \leftrightarrow \forall t \geq 0 ([x := y(t)]p(x))\]
[\[\]:=\] $[x := e]p(x) \leftrightarrow p(e)$

[\[\]′] $[x' = f(x)]p(x) \leftrightarrow \forall t \geq 0 \ [x := y(t)]p(x)$

[\[\]′] $[x' = f(x) \& q(x)]p(x) \leftrightarrow \forall t \geq 0 \ (\forall 0 \leq s \leq t \ q(y(s)) \rightarrow [x := y(t)]p(x))$
Dynamic Axioms for Dynamical Systems

\[ [\text{:=}] \quad [x := e] p(x) \leftrightarrow p(e) \]

\[ ['] \quad [x' = f(x)] p(x) \leftrightarrow \forall t \geq 0 [x := y(t)] p(x) \]

\[ ['] \quad [x' = f(x) \land q(x)] p(x) \leftrightarrow \forall t \geq 0 (\forall 0 \leq s \leq t q(y(s)) \rightarrow [x := y(t)] p(x)) \]

\[ [?] \quad [? Q] P \leftrightarrow \]

\[ \text{if } \omega \in \llbracket Q \rrbracket \]
[\text{:=}] \quad [x := e]p(x) \iff p(e)

[\text{'}] \quad [x' = f(x)]p(x) \iff \forall t \geq 0 [x := y(t)]p(x)

[\text{'}] \quad [x' = f(x) \& q(x)]p(x) \iff \forall t \geq 0 (\forall 0 \leq s \leq t q(y(s)) \rightarrow [x := y(t)]p(x))

[?] \quad [?Q]P \iff (Q \rightarrow P)

\text{if } \omega \in [Q]
compositional semantics ⇒ compositional proofs
$$[\cup] \ [\alpha \cup \beta] P \leftrightarrow$$
$$\bigcup \quad [\alpha \cup \beta]P \leftrightarrow [\alpha]P \land [\beta]P$$
Dynamic Axioms for Dynamical Systems

\[ \bigcup [\alpha \cup \beta] P \leftrightarrow [\alpha] P \land [\beta] P \]

\[ ; [\alpha; \beta] P \leftrightarrow \]

\[ \omega \]

\[ \nu_1 P \]

\[ \alpha \]

\[ \beta \]

\[ \nu_2 P \]

\[ \alpha \cup \beta \]

\[ \omega \]

\[ \nu \]

\[ s \]

\[ P \]

\[ \alpha \]

\[ \beta \]
$\left[\bigcup\right] \ [\alpha \cup \beta]P \leftrightarrow [\alpha]P \land [\beta]P$

$[;] \ [\alpha; \beta]P \leftrightarrow [\alpha][\beta]P$
[∪] \([α ∪ β]P \leftrightarrow [α]P ∧ [β]P\)

[;] \([α; β]P \leftrightarrow [α][β]P\)

[*] \([α^*]P \leftrightarrow\)
[∪] \([α ∪ β]P \iff [α]P ∧ [β]P\)

[;] \([α; β]P \iff [α][β]P\)

[*] \([α^*]P \iff P ∧ [α][α^*]P\)
\[
[\cup] \quad [\alpha \cup \beta] P \leftrightarrow [\alpha] P \land [\beta] P
\]

\[
[;] \quad [\alpha; \beta] P \leftrightarrow [\alpha][\beta] P
\]

\[
[\neg \neg [\alpha^*] P \leftrightarrow P \land
\]

\[
\omega \quad \alpha \quad \nu_1 \\
\beta \quad \nu_2 \\
\alpha \cup \beta
\]

\[
\omega \quad \alpha \quad s \\
\beta \quad \nu \\
[\alpha][\beta] P \\
[\beta] P
\]

\[
\omega \quad \alpha \quad \nu \\
\alpha^* \\
P
\]
\[
\begin{align*}
\{\cup\} \quad [\alpha \cup \beta]P & \leftrightarrow [\alpha]P \land [\beta]P \\
\{;\} \quad [\alpha; \beta]P & \leftrightarrow [\alpha][\beta]P \\
\{\downarrow\} \quad [\alpha^*]P & \leftrightarrow P \land [\alpha^*](P \rightarrow [\alpha]P)
\end{align*}
\]
\begin{align*}
\bigcup \quad & [\alpha \cup \beta]P \leftrightarrow [\alpha]P \land [\beta]P \\
\; & [\alpha \; \beta]P \leftrightarrow [\alpha][\beta]P \\
\; & [\alpha^*]P \leftrightarrow P \land [\alpha^*](P \rightarrow [\alpha]P)
\end{align*}
\[ \cup \quad [\alpha \cup \beta]P \leftrightarrow [\alpha]P \land [\beta]P \]

\[ ; \quad [\alpha; \beta]P \leftrightarrow [\alpha][\beta]P \]

\[ \quad [\alpha^*]P \leftrightarrow P \land [\alpha^*](P \rightarrow [\alpha]P) \]
\[ [\bigcup] \quad [\alpha \cup \beta] P \leftrightarrow [\alpha] P \land [\beta] P \]

\[ [:] \quad [\alpha; \beta] P \leftrightarrow [\alpha][\beta] P \]

\[ [\alpha^*] P \leftrightarrow P \land [\alpha^*](P \rightarrow [\alpha] P) \]
\[
\bigcup [\alpha \cup \beta] P \leftrightarrow [\alpha] P \land [\beta] P
\]

\[
; \quad [\alpha; \beta] P \leftrightarrow [\alpha][\beta] P
\]

\[
\downarrow \quad [\alpha^*] P \leftrightarrow P \land [\alpha^*](P \rightarrow [\alpha] P)
\]
[∪] \( [α ∪ β]P \leftrightarrow [α]P ∧ [β]P \)

[;] \( [α; β]P \leftrightarrow [α][β]P \)

\( \vdash \) \( [α^*]P \leftrightarrow P ∧ [α^*](P → [α]P) \)
\[ \bigcup \quad [\alpha \cup \beta] P \leftrightarrow [\alpha] P \land [\beta] P \]

\[ ; \quad [\alpha; \beta] P \leftrightarrow [\alpha][\beta] P \]

\[ \land \quad [\alpha^*] P \leftrightarrow P \land [\alpha^*](P \rightarrow [\alpha] P) \]
Proof Rule: Loop Invariants

\[
\begin{align*}
P & \quad \frac{\alpha}{[\alpha]P} \\
[\alpha^*]P & \leftrightarrow P \land [\alpha^*](P \to [\alpha]P) \\
P & \rightarrow Q \quad \frac{\alpha}{[\alpha]P \to [\alpha]Q}
\end{align*}
\]

Lemma (Loop invariant rule is derived)

\[
\begin{array}{c}
\text{loop} \\
\Gamma \vdash J, \Delta \\
J \vdash [\alpha]J \\
J \vdash P \\
\hline
\Gamma \vdash [\alpha^*]P, \Delta
\end{array}
\]

Finding invariant \( J \) can be a challenge.

Misplaced \([\alpha^*]\) suggests that \( J \) needs to carry along info about \( \alpha^* \) history.
Proof Rule: Loop Invariants

\[ P \quad I \quad [\alpha^*]P \leftrightarrow P \land [\alpha^*](P \rightarrow [\alpha]P) \quad M[\cdot] \quad P \rightarrow Q \]

\[ [\alpha]P \rightarrow [\alpha]Q \]

Lemma (Loop invariant rule is derived)

\[ \text{loop} \quad \Gamma \vdash J, \Delta \quad J \vdash [\alpha]J \quad J \vdash P \]

\[ \Gamma \vdash [\alpha^*]P, \Delta \]

Proof (Derived rule).

\[ \text{cut} \quad \Gamma \vdash J, \Delta \quad J \vdash [\alpha^*]J \]

\[ \Gamma \vdash [\alpha]J \]

\[ J \vdash [\alpha]J \]

\[ \Gamma \vdash J \land [\alpha^*](J \rightarrow [\alpha]J) \]

\[ \Gamma \vdash [\alpha^*]P, \Delta \]

\[ J \vdash P \]

\[ M[\cdot] [\alpha^*]J \vdash [\alpha^*]P \]
Proof Rule: Loop Invariants

\[ \frac{P}{\alpha]P} \]

I \quad \lbrack \alpha^* \rceil P \leftrightarrow P \land (P \to \lbrack \alpha \rceil P) \quad \frac{P \to Q}{\alpha]P \to \lbrack \alpha \rceil Q}

Lemma (Loop invariant rule is derived)

\[ \frac{\Gamma \vdash J, \Delta \quad J \vdash \lbrack \alpha \rceil J \quad J \vdash P}{\Gamma \vdash \lbrack \alpha^* \rceil P, \Delta} \]

Proof (Derived rule).

\[ \frac{J \vdash \lbrack \alpha \rceil J}{\Gamma \vdash J, \Delta \quad J \vdash \lbrack \alpha \rceil J \quad J \vdash P}{\Gamma \vdash \lbrack \alpha^* \rceil P, \Delta} \]

Finding invariant \( J \) can be a challenge. Misplaced \( \lbrack \alpha^* \rceil \) suggests that \( J \) needs to carry along info about \( \alpha^* \) history.
The lion’s share of understanding comes from understanding what does change (variants/progress measures) and what doesn’t change (invariants).

- Invariants are a fundamental force of CS
- Variants are another fundamental force of CS
Example Proof: Safe Braking

\[ J(x, v) \equiv x \leq m \]

\[ J(x, v) \vdash [a := -b; (x' = v, v' = a)]J(x, v) \]

1. \( \Gamma \vdash \Delta \) shape of conjecture to prove
2. \( \Gamma \) is list of all available assumptions
3. \( \Delta \) disjunction needs to be proved from assumptions \( \Gamma \)
4. Proof reduces desired conclusion \textit{conclusion} (at the bottom) to premises with remaining subgoals (top) until no more subgoals (\(*\)
Example Proof: Safe Braking

\[ J(x, v) \equiv x \leq m \]

\[ \begin{array}{c}
[=]
J(x, v) \vdash [a := -b][x' = v, v' = a]J(x, v) \\
[;]
J(x, v) \vdash [a := -b; (x' = v, v' = a)]J(x, v)
\end{array} \]

1. \( \Gamma \vdash \Delta \) shape of conjecture to prove
2. \( \Gamma \) is list of all available assumptions
3. \( \Delta \) disjunction needs to be proved from assumptions \( \Gamma \)
4. Proof reduces desired conclusion (at the bottom) to premises with remaining subgoals (top) until no more subgoals (\( \ast \))
Example Proof: Safe Braking

\[ J(x, v) \equiv x \leq m \]

\begin{align*}
[1] & \quad J(x, v) \vdash [x' = v, v' = -b]J(x, v) \\
[;=] & \quad J(x, v) \vdash [a := -b][x' = v, v' = a]J(x, v) \\
[;] & \quad J(x, v) \vdash [a := -b; (x' = v, v' = a)]J(x, v)
\end{align*}

1. \( \Gamma \vdash \Delta \) shape of conjecture to prove \hfill sequent
2. \( \Gamma \) is list of all available assumptions \hfill antecedent
3. \( \Delta \) disjunction needs to be proved from assumptions \( \Gamma \) \hfill succedent
4. Proof reduces desired conclusion (at the bottom) to premises with remaining subgoals (top) until no more subgoals (*)
Example Proof: Safe Braking

\[ J(x, v) \equiv x \leq m \]

\[
\begin{align*}
[=] & \quad J(x, v) \vdash \forall t \geq 0 \left[ x := -\frac{b}{2} t^2 + vt + x \right] J(x, v) \\
[\prime] & \quad J(x, v) \vdash \left[ x' = v, v' = -b \right] J(x, v) \\
[=] & \quad J(x, v) \vdash \left[ a := -b \right] \left[ x' = v, v' = a \right] J(x, v) \\
[;] & \quad J(x, v) \vdash \left[ a := -b; (x' = v, v' = a) \right] J(x, v)
\end{align*}
\]

1. \( \Gamma \vdash \Delta \) shape of conjecture to prove
2. \( \Gamma \) is list of all available assumptions
3. \( \Delta \) disjunction needs to be proved from assumptions \( \Gamma \)
4. Proof reduces desired conclusion (at the bottom) to premises with remaining subgoals (top) until no more subgoals (\( \ast \))
Example Proof: Safe Braking

\[ J(x, v) \equiv x \leq m \]

\[
\begin{align*}
\text{QE} & \quad J(x, v) \vdash \forall t \geq 0 \left( -\frac{b}{2} t^2 + vt + x \leq m \right) \\
[:=] & \quad J(x, v) \vdash \forall t \geq 0 \left[ x := -\frac{b}{2} t^2 + vt + x \right] J(x, v) \\
[\prime] & \quad J(x, v) \vdash [x' = v, v' = -b] J(x, v) \\
[:=] & \quad J(x, v) \vdash [a := -b] [x' = v, v' = a] J(x, v) \\
[;] & \quad J(x, v) \vdash [a := -b; (x' = v, v' = a)] J(x, v)
\end{align*}
\]

1. **\( \Gamma \vdash \Delta \)** shape of conjecture to prove
2. **\( \Gamma \)** is list of all available assumptions
3. **\( \Delta \)** disjunction needs to be proved from assumptions \( \Gamma \)
4. Proof reduces desired **conclusion** (at the bottom) to **premises** with remaining subgoals (top) until no more subgoals (*)

Andre Platzer (CMU) Dynamic Logic for Dynamical Systems MOD’17 26 / 59
Example Proof: Safe Braking

\[ J(x, v) \equiv x \leq m \]

\[
\begin{align*}
J(x, v) & \vdash v^2 \leq 2b(m - x) \\
& \text{QE} \\
J(x, v) & \vdash \forall t \geq 0 \left(-\frac{b}{2}t^2 + vt + x \leq m\right) \\
\left[=\right] & \\
J(x, v) & \vdash \forall t \geq 0 \left[x := -\frac{b}{2}t^2 + vt + x\right]J(x, v) \\
\left[\right] & \\
J(x, v) & \vdash \left[x' = v, v' = -b\right]J(x, v) \\
\left[=\right] & \\
J(x, v) & \vdash \left[a := -b\right]\left[x' = v, v' = a\right]J(x, v) \\
\left[;\right] & \\
J(x, v) & \vdash \left[a := -b; (x' = v, v' = a)\right]J(x, v)
\end{align*}
\]

1. \( \Gamma \vdash \Delta \) shape of conjecture to prove
2. \( \Gamma \) is list of all available assumptions
3. \( \Delta \) disjunction needs to be proved from assumptions \( \Gamma \)
4. Proof reduces desired conclusion (at the bottom) to premises with remaining subgoals (top) until no more subgoals (\( \ast \))
Example Proof: Safe Braking

\[ J(x, v) \equiv v^2 \leq 2b(m - x) \]

\[
\begin{align*}
J(x, v) \vdash v^2 &\leq 2b(m - x) \\
\text{QE} &\quad J(x, v) \vdash \forall t \geq 0 \left( -\frac{b}{2} t^2 + vt + x \leq m \right) \\
[=] &\quad J(x, v) \vdash \forall t \geq 0 \left[ x := -\frac{b}{2} t^2 + vt + x \right] J(x, v) \\
\left[\right] &\quad J(x, v) \vdash \left[ x' = v, v' = -b \right] J(x, v) \\
[=] &\quad J(x, v) \vdash \left[ a := -b \right] \left[ x' = v, v' = a \right] J(x, v) \\
[;] &\quad J(x, v) \vdash \left[ a := -b; (x' = v, v' = a) \right] J(x, v)
\end{align*}
\]
Example Proof: Safe Acceleration

\[ J(x, v) \equiv v^2 \leq 2b(m - x) \]

\[ [; J(x, v) \vdash [? \neg SB; a := A; (x' = v, v' = a, t' = 1 \& t \leq \varepsilon)]J(x, v) \]
Example Proof: Safe Acceleration

\[ J(x, v) \equiv v^2 \leq 2b(m - x) \]

\[ J(x, v) \vdash [\neg SB][a := A; (x' = v, v' = a, t' = 1 & t \leq \varepsilon)]J(x, v) \]

\[ J(x, v) \vdash [\neg SB; a := A; (x' = v, v' = a, t' = 1 & t \leq \varepsilon)]J(x, v) \]
Example Proof: Safe Acceleration

\[ J(x, v) \equiv v^2 \leq 2b(m - x) \]

\[ J(x, v) \vdash \neg SB \rightarrow [a := A; (x' = v, v' = a, t' = 1 \& t \leq \varepsilon)]J(x, v) \]

\[ J(x, v) \vdash [?\neg SB][a := A; (x' = v, v' = a, t' = 1 \& t \leq \varepsilon)]J(x, v) \]

\[ J(x, v) \vdash [?\neg SB; a := A; (x' = v, v' = a, t' = 1 \& t \leq \varepsilon)]J(x, v) \]
Example Proof: Safe Acceleration

\[ J(x, v) \equiv v^2 \leq 2b(m - x) \]

\[
\begin{align*}
\text{[:=]} & \quad J(x, v) \vdash \neg \text{SB} \rightarrow [a := A][x' = v, v' = a, t' = 1 \& t \leq \varepsilon]J(x, v) \\
\text{[;]} & \quad J(x, v) \vdash \neg \text{SB} \rightarrow [a := A; (x' = v, v' = a, t' = 1 \& t \leq \varepsilon)]J(x, v) \\
\text{[?]} & \quad J(x, v) \vdash [?\neg \text{SB}][a := A; (x' = v, v' = a, t' = 1 \& t \leq \varepsilon)]J(x, v) \\
\text{[;]} & \quad J(x, v) \vdash [?\neg \text{SB}; a := A; (x' = v, v' = a, t' = 1 \& t \leq \varepsilon)]J(x, v)
\end{align*}
\]
Example Proof: Safe Acceleration

\[ J(x, v) \equiv v^2 \leq 2b(m - x) \]

\[ x \]

\[ v \]

\[ m \]

\[ ^{[\cdot]} \quad J(x, v) \vdash \neg SB \rightarrow [x' = v, v' = A, t' = 1 \& t \leq \varepsilon]J(x, v) \]

\[ ^{:=} \quad J(x, v) \vdash \neg SB \rightarrow [a := A][x' = v, v' = a, t' = 1 \& t \leq \varepsilon]J(x, v) \]

\[ ^{[\cdot]} \quad J(x, v) \vdash \neg SB \rightarrow [a := A; (x' = v, v' = a, t' = 1 \& t \leq \varepsilon)]J(x, v) \]

\[ ^{[?]} \quad J(x, v) \vdash [\neg SB][a := A; (x' = v, v' = a, t' = 1 \& t \leq \varepsilon)]J(x, v) \]

\[ ^{[\cdot]} \quad J(x, v) \vdash [\neg SB; a := A; (x' = v, v' = a, t' = 1 \& t \leq \varepsilon)]J(x, v) \]
\[ J(x, v) \equiv v^2 \leq 2b(m - x) \]

\[
\begin{align*}
\text{[=]} & \quad J(x, v) \vdash \neg SB \rightarrow \forall t \geq 0 \left( t \leq \varepsilon \rightarrow [x := \frac{A}{2} t^2 + vt + x] J(x, v) \right) \\
\text{[\textprime]} & \quad J(x, v) \vdash \neg SB \rightarrow [x' = v, v' = A, t' = 1 \& t \leq \varepsilon] J(x, v) \\
\text{[=]} & \quad J(x, v) \vdash \neg SB \rightarrow [a := A][x' = v, v' = a, t' = 1 \& t \leq \varepsilon] J(x, v) \\
\text{[\textdoubleprime]} & \quad J(x, v) \vdash \neg SB \rightarrow [a := A; (x' = v, v' = a, t' = 1 \& t \leq \varepsilon)] J(x, v) \\
\text{[\textquestionmark]} & \quad J(x, v) \vdash [\neg SB][a := A; (x' = v, v' = a, t' = 1 \& t \leq \varepsilon)] J(x, v) \\
\text{[\textdoubleprime]} & \quad J(x, v) \vdash [\neg SB; a := A; (x' = v, v' = a, t' = 1 \& t \leq \varepsilon)] J(x, v)
\end{align*}
\]
Example Proof: Safe Acceleration

\[ J(x, v) \equiv v^2 \leq 2b(m - x) \]

\[ J(x, v) \vdash \neg SB \rightarrow \forall t \geq 0 (t \leq \varepsilon \rightarrow J(\frac{A}{2} t^2 + vt + x, At + v)) \]

[\!] \[ J(x, v) \vdash \neg SB \rightarrow [x' = v, v' = A, t' = 1 \& t \leq \varepsilon] J(x, v) \]

[\!] \[ J(x, v) \vdash \neg SB \rightarrow [a := A][x' = v, v' = a, t' = 1 \& t \leq \varepsilon] J(x, v) \]

[\!] \[ J(x, v) \vdash \neg SB \rightarrow [a := A; (x' = v, v' = a, t' = 1 \& t \leq \varepsilon)] J(x, v) \]

[\?] \[ J(x, v) \vdash [? SB][a := A; (x' = v, v' = a, t' = 1 \& t \leq \varepsilon)] J(x, v) \]

[\!] \[ J(x, v) \vdash [? SB; a := A; (x' = v, v' = a, t' = 1 \& t \leq \varepsilon)] J(x, v) \]
Example Proof: Safe Acceleration

\[ J(x, v) \equiv v^2 \leq 2b(m - x) \]

\[
\begin{align*}
\text{QE} & \quad J(x, v) \vdash \neg SB \rightarrow \forall t \geq 0 \ (t \leq \epsilon \rightarrow (At + v)^2 \leq 2b(m - \frac{A}{2} t^2 - vt - x)) \\
\text{[=]} & \quad J(x, v) \vdash \neg SB \rightarrow \forall t \geq 0 \ (t \leq \epsilon \rightarrow J(\frac{A}{2} t^2 + vt + x, At + v)) \\
\text{[\prime]} & \quad J(x, v) \vdash \neg SB \rightarrow [x' = v, v' = A, t' = 1 \& t \leq \epsilon] J(x, v) \\
\text{[=]} & \quad J(x, v) \vdash \neg SB \rightarrow [a := A][x' = v, v' = a, t' = 1 \& t \leq \epsilon] J(x, v) \\
\text{[;]} & \quad J(x, v) \vdash [?\neg SB][a := A; (x' = v, v' = a, t' = 1 \& t \leq \epsilon)] J(x, v) \\
\text{[\?]} & \quad J(x, v) \vdash [?\neg SB; a := A; (x' = v, v' = a, t' = 1 \& t \leq \epsilon)] J(x, v) \\
\end{align*}
\]
Example Proof: Safe Acceleration

\[ J(x, v) \equiv v^2 \leq 2b(m - x) \]

\[ J(x, v) \vdash \neg \text{SB} \rightarrow (A\varepsilon + v)^2 \leq 2b(m - \frac{A}{2}\varepsilon^2 - v\varepsilon - x) \]

\[ \text{QE} \]

\[ J(x, v) \vdash \neg \text{SB} \rightarrow \forall t \geq 0 (t \leq \varepsilon \rightarrow (At + v)^2 \leq 2b(m - \frac{A}{2}t^2 - vt - x)) \]

\[ [\vdash] \]

\[ J(x, v) \vdash \neg \text{SB} \rightarrow \forall t \geq 0 (t \leq \varepsilon \rightarrow [x := \frac{A}{2}t^2 + vt + x]J(x, v)) \]

\[ [\vdash'] \]

\[ J(x, v) \vdash \neg \text{SB} \rightarrow [x' = v, v' = A, t' = 1 \& t \leq \varepsilon]J(x, v) \]

\[ [\vdash] \]

\[ J(x, v) \vdash \neg \text{SB} \rightarrow [a := A][x' = v, v' = a, t' = 1 \& t \leq \varepsilon]J(x, v) \]

\[ [\vdash'] \]

\[ J(x, v) \vdash [\neg \text{SB}]a := A; (x' = v, v' = a, t' = 1 \& t \leq \varepsilon]J(x, v) \]

\[ [\vdash] \]

\[ J(x, v) \vdash [\neg \text{SB}; a := A; (x' = v, v' = a, t' = 1 \& t \leq \varepsilon]J(x, v) \]
Example Proof: Safe Acceleration

\[ J(x, v) \equiv v^2 \leq 2b(m - x) \]

\[ SB \equiv 2b(m - x) < v^2 + (A + b)(A\varepsilon^2 + 2\varepsilon v) \]

\[ J(x, v) \vdash \neg SB \rightarrow (A\varepsilon + v)^2 \leq 2b(m - \frac{A}{2}\varepsilon^2 - v\varepsilon - x) \]

\[ \text{QE} \]

\[ J(x, v) \vdash \neg SB \rightarrow \forall t \geq 0 (t \leq \varepsilon \rightarrow (At + v)^2 \leq 2b(m - \frac{A}{2}t^2 - vt - x)) \]

\[ J(x, v) \vdash \neg SB \rightarrow \forall t \geq 0 (t \leq \varepsilon \rightarrow J(\frac{A}{2}t^2 + vt + x, At + v)) \]

\[ := \]

\[ J(x, v) \vdash \neg SB \rightarrow \forall t \geq 0 (t \leq \varepsilon \rightarrow [x := \frac{A}{2}t^2 + vt + x]J(x, v)) \]

\[ ['] \]

\[ J(x, v) \vdash \neg SB \rightarrow [x' = v, v' = A, t' = 1 \& t \leq \varepsilon]J(x, v) \]

\[ := \]

\[ J(x, v) \vdash \neg SB \rightarrow [a := A][x' = v, v' = a, t' = 1 \& t \leq \varepsilon]J(x, v) \]

\[ [:] \]

\[ J(x, v) \vdash \neg SB \rightarrow [a := A; (x' = v, v' = a, t' = 1 \& t \leq \varepsilon)]J(x, v) \]

\[ [?] \]

\[ J(x, v) \vdash [? \neg SB][a := A; (x' = v, v' = a, t' = 1 \& t \leq \varepsilon)]J(x, v) \]

\[ [:] \]

\[ J(x, v) \vdash [? \neg SB; a := A; (x' = v, v' = a, t' = 1 \& t \leq \varepsilon)]J(x, v) \]
$J(x, v) \equiv v^2 \leq 2b(m - x)$

$SB \equiv 2b(m - x) < v^2 + (A + b)(A\varepsilon^2 + 2\varepsilon v)$

Andrè Platzer (CMU)

Dynamic Logic for Dynamical Systems

MOD’17 26 / 59
Example Proof: Safe Driving

\[ J(x, v) \equiv v^2 \leq 2b(m - x) \]
\[ \text{SB} \equiv 2b(m - x) < v^2 + (A + b)(A\varepsilon^2 + 2\varepsilon v) \]

\[
\begin{align*}
\text{loop} & \\
J(x, v) & \vdash [(a := -b \cup ?\neg\text{SB}; a := A); x'' = a, t' = 1 & t \leq \varepsilon]J(x, v)
\end{align*}
\]
Example Proof: Safe Driving

\[
J(x, v) \equiv v^2 \leq 2b(m - x)
\]

\[
SB \equiv 2b(m - x) < v^2 + (A + b)(A\varepsilon^2 + 2\varepsilon v)
\]

\[
J(x, v) \vdash [a := -b \cup ?\neg SB; a := A][x'' = a, t' = 1 & t \leq \varepsilon]J(x, v)
\]

\[
J(x, v) \vdash [(a := -b \cup ?\neg SB; a := A); x'' = a, t' = 1 & t \leq \varepsilon]J(x, v)
\]

\[
\text{loop } J(x, v) \vdash [((a := -b \cup ?\neg SB; a := A); x'' = a, t' = 1 & t \leq \varepsilon)^*]J(x, v)
\]
Example Proof: Safe Driving

\[ J(x, v) \equiv v^2 \leq 2b(m - x) \]
\[ SB \equiv 2b(m - x) < v^2 + (A + b)(A\varepsilon^2 + 2\varepsilon v) \]

\[ J(x, v) \vdash [a := -b][x'' = a \ldots]J(x, v) \land [\neg SB; a := A][x'' = a \ldots]J(x, v) \]

[\cup] \hspace{1cm}
\[ J(x, v) \vdash [a := -b \cup \neg SB; a := A][x'' = a, t' = 1 \land t \leq \varepsilon]J(x, v) \]

[\Delta] \hspace{1cm}
\[ J(x, v) \vdash [(a := -b \cup \neg SB; a := A); x'' = a, t' = 1 \land t \leq \varepsilon]J(x, v) \]

[loop] \hspace{1cm}
\[ J(x, v) \vdash [(a := -b \cup \neg SB; a := A); x'' = a, t' = 1 \land t \leq \varepsilon]^*J(x, v) \]
Example Proof: Safe Driving

\[ J(x, v) \equiv v^2 \leq 2b(m - x) \]
\[ SB \equiv 2b(m - x) < v^2 + (A + b)(A\varepsilon^2 + 2\varepsilon v) \]

previous proofs for braking and acceleration
\[ J(x, v) \models [a := -b][x'' = a \ldots]J(x, v) \wedge [\neg SB; a := A][x'' = a \ldots]J(x, v) \]

\[ J(x, v) \models [a := -b \cup ?\neg SB; a := A][x'' = a, t' = 1 \& t \leq \varepsilon]J(x, v) \]

\[ J(x, v) \models [(a := -b \cup ?\neg SB; a := A); x'' = a, t' = 1 \& t \leq \varepsilon]J(x, v) \]

loop
\[ J(x, v) \models [(a := -b \cup ?\neg SB; a := A); x'' = a, t' = 1 \& t \leq \varepsilon)^*]J(x, v) \]
Example Proof: Safe Driving

\[ J(x, v) \equiv v^2 \leq 2b(m - x) \]
\[ SB \equiv 2b(m - x) < v^2 + (A + b)(A\varepsilon^2 + 2\varepsilon v) \]

<table>
<thead>
<tr>
<th>Previous proofs for braking and acceleration</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ J(x, v) \vdash [a := -b][x'' = a \ldots]J(x, v) \land [\neg SB; a := A][x'' = a \ldots]J(x, v) ]</td>
</tr>
<tr>
<td>[ J(x, v) \vdash [a := -b \cup \neg SB; a := A][x'' = a, t' = 1 &amp; t \leq \varepsilon]J(x, v) ]</td>
</tr>
<tr>
<td>[ J(x, v) \vdash [(a := -b \cup \neg SB; a := A); x'' = a, t' = 1 &amp; t \leq \varepsilon]J(x, v) ]</td>
</tr>
<tr>
<td>Loop: [ J(x, v) \vdash [((a := -b \cup \neg SB; a := A); x'' = a, t' = 1 &amp; t \leq \varepsilon)^*]J(x, v) ]</td>
</tr>
</tbody>
</table>

1. Proof is deterministic “follow your nose”.
2. Synthesize invariant \( J(x, v) \) and parameter constraint \( SB \).
3. \( J(x, v) \) is a predicate symbol to prove only once and instantiate later.
4. First looking at proofs of smaller pieces is often effective.
Example (Bouncing Ball)

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

\[
x = H \geq 0 \land \ldots \rightarrow \left[ (x' = v, v' = -g \land x \geq 0; \right. \\
\left. \text{if}(x = 0) \ v := -cv \right)^* \left. \right] 0 \leq x \leq H
\]
Ex: Runaround Robot

Example (Runaround Robot)

\((x, y) \neq o \rightarrow \left( (\mathcal{Q}_1; \omega := -1 \cup \mathcal{Q}_1; \omega := 1 \cup \mathcal{Q}_0; \omega := 0); \{x' = v, y' = w, v' = \omega w, w' = -\omega v}\right)^* \right) (x, y) \neq o
Ex: Runaround Robot

\[ Q_\omega \equiv (x + \frac{w}{\omega} - o_x)^2 + (y - \frac{v}{\omega} - o_y)^2 \neq v^2 + w^2 \]
\[ Q_0 \equiv (o_x - x)w \neq (o_y - y)v \]

1. Obstacle not on tangential circle
2. Obstacle not on ray \((x, y) + \mathbb{R}(v, w)\)

Example (Runaround Robot)

\[(x, y) \neq o \rightarrow \left[ (\text{?}Q_{-1}; \omega := -1 \cup \text{?}Q_1; \omega := 1 \cup \text{?}Q_0; \omega := 0) ; \{ x' = v , y' = w , v' = \omega w , w' = -\omega v \}^* \right] (x, y) \neq o \]
**Theorem (Sound & Complete) (JAR’08, LICS’12, JAR’17)**

dL calculus is a sound & complete axiomatization of hybrid systems relative to either differential equations or to discrete dynamics.

**Corollary (Complete Proof-theoretical Bridge)**

proving continuous = proving hybrid = proving discrete

\[ \vdash P \text{ iff } \text{FODE} \vdash_{dL} P \]
Theorem (Sound & Complete) (JAR’08, LICS’12, JAR’17)

dL calculus is a sound & complete axiomatization of hybrid systems relative to either differential equations or to discrete dynamics.

Corollary (Complete Proof-theoretical Bridge)

proving continuous = proving hybrid = proving discrete

ystem

Continuous

Discrete

Hybrid

System

André Platzer (CMU)
Theorem (Sound & Complete) (JAR’08, LICS’12, JAR’17)

dL calculus is a sound & complete axiomatization of hybrid systems relative to either differential equations or to discrete dynamics.

Proof 25pp

Corollary (Complete Proof-theoretical Bridge)
proving continuous = proving hybrid = proving discrete

System
Continuous
Hybrid
Discrete

Andr´e Platzer (CMU)
Theorem (Sound & Complete) \(\text{(JAR'08, LICS'12, JAR'17)}\)

dL calculus is a sound & complete axiomatization of hybrid systems relative to either differential equations or to discrete dynamics.

Corollary (Complete Proof-theoretical Bridge)

proving continuous = proving hybrid = proving discrete
Theorem (Sound & Complete) \( (\text{JAR'08, LICS'12, JAR'17}) \)

\[ \text{dL calculus is a sound & complete axiomatization of hybrid systems relative to either differential equations or to discrete dynamics.} \]

Corollary (Complete Proof-theoretical Bridge)

proving continuous = proving hybrid = proving discrete
Complete Proof Theory of Hybrid Systems

Theorem (Sound & Complete) (JAR’08, LICS’12, JAR’17)

\[ dL \text{ calculus is a sound & complete axiomatization of hybrid systems relative to either differential equations or to discrete dynamics. } \]

Corollary (Complete Proof-theoretical Bridge)

proving continuous = proving hybrid = proving discrete
Theorem (Sound & Complete) (JAR’08, LICS’12, JAR’17)

\( dL \) calculus is a sound & complete axiomatization of hybrid systems relative to either differential equations or to discrete dynamics.

Corollary (Complete Proof-theoretical Bridge)

proving continuous = proving hybrid = proving discrete
**Theorem (Sound & Complete)** (JAR’08, LICS’12, JAR’17)

\[ \text{dL calculus is a sound & complete axiomatization of hybrid systems relative to either differential equations or to discrete dynamics.} \]

**Corollary (Complete Proof-theoretical Bridge)**

proving continuous = proving hybrid = proving discrete

---

André Platzer (CMU)  
Dynamic Logic for Dynamical Systems  
MOD’17 29 / 59
Theorem (Sound & Complete) (JAR'08, LICS’12, JAR’17)

$dL$ calculus is a sound & complete axiomatization of hybrid systems relative to either differential equations or to discrete dynamics.

Proof 25pp

Corollary (Complete Proof-theoretical Bridge)

proving continuous $=$ proving hybrid $=$ proving discrete

Theorem (Equi-expressibility) (LICS’12)

$\forall P \in dL \ \exists P^b \in FODE \ \models P \leftrightarrow P^b$

$\forall P \in dL \ \exists P^\# \in DL \ \models P \leftrightarrow P^\#$

Theorem (Relative Decidability) (LICS’12)

Validity of $dL$ sentences is decidable relative to FOD or DL.
KeYmaera X: aXiomatic Tactical Theorem Prover for CPS

Proof search

KeYmaera X

Model

ctrl: \( a := -b \);
plant: \( x'' = a \)

generates proofs

ModelPlex proof

Proof search

Trustworthy

- Uniform substitution
- Sound & complete
- Small core: 1700 LOC

Flexible

- Proof automation
- Interactive UI
- Programmable

Customizable

- Scala+Java API
- Command line
- REST API

André Platzer (CMU)

Dynamic Logic for Dynamical Systems

MOD'17 30 / 59
Disclaimer: Self-reported estimates of the soundness-critical lines of code + rules
Uniform Substitution

Theorem (Soundness)

\[
\frac{\phi}{\sigma(\phi)}
\]

replace all occurrences of \( p(\cdot) \)

provided \( \text{FV}(\sigma|\Sigma(\theta)) \cap \text{BV}(\otimes(\theta)) = \emptyset \) for each operation \( \otimes(\theta) \) in \( \phi \)

i.e. bound variables \( U = \text{BV}(\otimes(\cdot)) \) of operator \( \otimes \)

are not free in the substitution on its argument \( \theta \) \hspace{1cm} (U-admissible)

\[
\begin{align*}
\text{US} \quad \frac{[a \cup b]p(\bar{x}) \leftrightarrow [a]p(\bar{x}) \land [b]p(\bar{x})}{[x := x + 1 \cup x' = 1]x \geq 0 \leftrightarrow [x := x + 1]x \geq 0 \land [x' = 1]x \geq 0}
\end{align*}
\]
Application Highlights

Obstacle Avoidance + Ground Navigation

Airborne Collision Avoidance (ACAS X)

Train Control Brakes

Ship Cooling

Acknowledgments

DARPA

National Aeronautics and Space Administration

National Science Foundation

Johns Hopkins Applied Physics Laboratory

Bosch
Outline (Proving ODEs in CPSs)

1. CPS are Multi-Dynamical Systems
   - Hybrid Systems / Games / Stochastic / Distributed Hybrid Systems

2. Differential Dynamic Logic
   - Syntax
   - Semantics
   - Example: Car Control Design

3. Dynamic Axioms for Dynamical Systems
   - Axiomatics
   - Example: Safe Car Control
   - Soundness and Completeness

4. Differential Invariants for Differential Equations
   - Differential Invariants
   - Example: Elementary Differential Invariants
   - Differential Axioms
   - Example: Differential Ghosts

5. Applications

6. Summary
Theorem (Sound & Complete) \( (\text{JAR}'08, \text{LICS}'12, \text{JAR}'17) \)

\( \mathsf{dL} \) calculus is a sound & complete axiomatization of hybrid systems relative to either differential equations or to discrete dynamics. \( \rightarrow \) Proof 25pp

Corollary (Complete Proof-theoretical Bridge)

proving continuous = proving hybrid = proving discrete
**Theorem (Sound & Complete)** (JAR’08, LICS’12, JAR’17)

\[ \text{dL calculus is a sound & complete axiomatization of hybrid systems relative to either differential equations or to discrete dynamics.} \]

**Corollary (Complete Proof-theoretical Bridge)**

proving continuous = proving hybrid = proving discrete

**Theorem (Equi-expressibility)** (LICS’12)

\[
\forall P \in \text{dL} \quad \exists P^b \in \text{FODE} \quad \models P \leftrightarrow P^b \\
\forall P \in \text{dL} \quad \exists P^\# \in \text{DL} \quad \models P \leftrightarrow P^\#
\]

**Theorem (Relative Decidability)** (LICS’12)

\[ \text{Validity of dL sentences is decidable relative to FOD or DL.} \]
Global Descriptive Power of Local Differential Equations

Descriptive power of differential equations

1. Simple differential equations describe complex physical processes.
2. Solution is a global description of the system evolution.
3. ODE is a local characterization.
4. Complexity difference between local description and global behavior.
5. Let’s exploit that phenomenon for proofs!
6. Reason locally about global behavior.

\[ x'' = -x \quad \text{has} \quad x(t) = \sin(t) = t - \frac{t^3}{3!} + \frac{t^5}{5!} - \frac{t^7}{7!} + \frac{t^9}{9!} - \ldots \]

\[ x'''(t) = e^{t^2} \quad \text{has no elementary closed-form solution} \]
Intuition for Differential Invariants

Differential Invariant

\(\Gamma \vdash J, \Delta \quad J \vdash ??? J \quad J \vdash P\)

\(\Gamma \vdash [x' = f(x)]P, \Delta\)

\(['] [x' = f(x)]P \leftrightarrow \forall t \geq 0 [x := y(t)]P \quad (y' = f(y), y(0) = x)\)
Intuition for Differential Invariants

Differential Invariant

\[ \Gamma \vdash J, \Delta \quad J \vdash ???J \quad J \vdash P \]

\[ \Gamma \vdash [x' = f(x)]P, \Delta \]

\[ ['] [x' = f(x)]P \leftrightarrow \forall t \geq 0 [x := y(t)]P \quad (y' = f(y), y(0) = x) \]
Intuition for Differential Invariants

Differential Invariant

\[ \Gamma \vdash J, \Delta \quad J \vdash ??? J \quad J \vdash P \]

\[ \Gamma \vdash [x' = f(x)]P, \Delta \]

\[ \left[ \begin{array}{c}
\left[ x' = f(x) \right] P \\
\leftrightarrow \forall t \geq 0 \left[ x := y(t) \right] P
\end{array} \right] \quad (y' = f(y), y(0) = x) \]
Intuition for Differential Invariants

Differential Invariant

\[ \Gamma \vdash J, \Delta \quad J \vdash ??? J \quad J \vdash P \]

\[ \Gamma \vdash [x' = f(x)]P, \Delta \]

Want: formula \( J \) remains true in the direction of the dynamics

\[ \neg F \quad \neg F \]

\[ [\'] [x' = f(x)]P \iff \forall t \geq 0 [x := y(t)]P \quad (y' = f(y), y(0) = x) \]

Don’t need to know where exactly the system evolves to. Just that it remains somewhere in \( J \).

Show: only evolves into directions in which formula \( J \) stays true.
Guiding Example

\[ v^2 + w^2 = r^2 \rightarrow [v' = w, w' = -v] \]

\[ v^2 + w^2 = r^2 \]
Guiding Example: Rotational Dynamics

\[ v^2 + w^2 = r^2 \rightarrow [v' = w, w' = -v] \ v^2 + w^2 = r^2 \]
Guiding Example: Rotational Dynamics

\[ v^2 + w^2 = r^2 \rightarrow [v' = w, w' = -v] \]

\[ v^2 + w^2 = r^2 \]

\[ \vdash v^2 + w^2 - r^2 = 0 \rightarrow [v' = w, w' = -v] \]
\[ v^2 + w^2 = r^2 \rightarrow [v' = w, w' = -v] v^2 + w^2 = r^2 \]

\[ \vdash v^2 + w^2 - r^2 = 0 \leftarrow [v' = w, w' = -v] v^2 + w^2 - r^2 = 0 \]

\[ \vdash v^2 + w^2 - r^2 = 0 \rightarrow [v' = w, w' = -v] v^2 + w^2 - r^2 = 0 \]
Guiding Example: Rotational Dynamics

\[ v^2 + w^2 = r^2 \rightarrow [v' = w, w' = -v] v^2 + w^2 = r^2 \]

\[
\begin{align*}
\vdash [v' := w][w' := -v] & \quad 2vv' + 2ww' - 2rr' = 0 \\
\vdash v^2 + w^2 - r^2 = 0 & \quad \vdash [v' = w, w' = -v] v^2 + w^2 - r^2 = 0 \\
\vdash v^2 + w^2 - r^2 = 0 \rightarrow [v' = w, w' = -v] & \quad v^2 + w^2 - r^2 = 0
\end{align*}
\]
Guiding Example: Rotational Dynamics

\[ v^2 + w^2 = r^2 \rightarrow [v' = w, w' = -v] \quad v^2 + w^2 = r^2 \]

\[
\begin{array}{c}
\rightarrow R \\
\begin{array}{c}
dl \\
2v(w) + 2w(-v) = 0 \quad \vdash [v' := w][w' := -v] 2v v' + 2w w' - 2rr' = 0 \\
[v' = w, w' = -v] v^2 + w^2 - r^2 = 0 \quad \vdash v^2 + w^2 - r^2 = 0 \rightarrow [v' = w, w' = -v] v^2 + w^2 - r^2 = 0
\end{array}
\end{array}
\]
Guiding Example: Rotational Dynamics

\[ v^2 + w^2 = r^2 \rightarrow [v' = w, w' = -v] \quad v^2 + w^2 = r^2 \]

\[ \begin{array}{c}
\text{dI} \\
\Rightarrow R
\end{array} \quad \begin{array}{c}
\vdash [v' = w, w' = -v] \quad 2v v' + 2w w' - 2r r' = 0 \\
\vdash 2v(w) + 2w(-v) = 0 \\
\vdash v^2 + w^2 - r^2 = 0 \rightarrow [v' = w, w' = -v] \quad v^2 + w^2 - r^2 = 0
\end{array} \]
Differential Invariants for Differential Equations

\[
x' = f(x)
\]

\[
y' = g(x, y)
\]

JLogComput’10, CAV’08, FMSD’09, LMCS’12, LICS’12, ITP’12, JAR’17
Differential Invariants for Differential Equations

\[ \frac{dx}{dt} = f(x) \]

\[ \frac{dy}{dt} = g(x, y) \]

\( \text{DI} \geq \text{DI}, \wedge, \vee \)

\( \text{DI} > \text{DI}, \wedge, \vee \)

Logic

Provability theory

Math

Characteristic PDE

JLogComput'10, CAV'08, FMSD'09, LMCS'12, LICS'12, ITP'12, JAR'17
Differential Invariants for Differential Equations

**Differential Invariant**

\[ x' = f(x) \]

**Differential Cut**

**Differential Ghost**

\[ x' = f(x) \]

\[ 0 \rightarrow t \]

**Logic Provability theory**

**Math Characteristic PDE**

JLogComput’10, CAV’08, FMSD’09, LMCS’12, LICS’12, ITP’12, JAR’17

Dynamic Logic for Dynamical Systems
Differential Invariants for Differential Equations

Differential Invariant

Differential Cut

Differential Ghost

Logic
Provability theory

Math
Characteristic PDE

JLogComput'10, CAV'08, FMSD'09, LMCS'12, LICS'12, ITP'12, JAR'17
Differential Invariants for Differential Equations

- **Differential Invariant**
  - $\dot{x} = f(x)$
  - $\dot{y} = g(x, y)$

- **Differential Cut**
  - $\text{DI} = \text{DI} >, \land, \lor$

- **Differential Ghost**
  - $x' = f(x)$
  - $t$

**Logic**
- Provability theory

**Math**
- Characteristic PDE

JLogComput'10, CAV'08, FMSD'09, LMCS'12, LICS'12, ITP'12, JAR'17
Differential Invariants for Differential Equations

\[\begin{align*}
D(t) &= t \\
\dot{x} &= f(x) \\
\dot{y} &= g(x, y)
\end{align*}\]

Logic

Provability theory

Math

Characteristic PDE

JLogComput’10, CAV’08, FMSD’09, LMCS’12, LICS’12, ITP’12, JAR’17

Andre Platzer (CMU)
Differential Invariants for Differential Equations

Differential Invariant

\[ x' = f(x) \]

Differential Cut

\[ y' = g(x, y) \]

Differential Ghost

\[ \text{Logic} \]

Provability theory

\[ \text{Math} \]

Characteristic PDE

JLogComput’10, CAV’08, FMSD’09, LMCS’12, LICS’12, ITP’12, JAR’17

André Platzer (CMU)

Dynamic Logic for Dynamical Systems

MOD’17 38 / 59
Differential Invariants for Differential Equations

Differential Invariant

Differential Cut

Differential Ghost

\[ x' = f(x) \]

Logic

Provability theory

Math

Characteristic PDE

JLogComput'10, CAV'08, FMSD'09, LMCS'12, LICS'12, ITP'12, JAR'17
Differential Invariants for Differential Equations

Differential Invariant

Differential Cut

Differential Ghost

\[ \frac{dx}{dt} = f(x) \]

\[ \frac{dy}{dt} = g(x,y) \]

Logic

Provability theory

Math

Characteristic PDE

JLogComput'10, CAV'08, FMSD'09, LMCS'12, LICS'12, ITP'12, JAR'17

André Platzer (CMU)
Differential Invariants for Differential Equations

\[ \dot{x} = f(x) \]
\[ \dot{y} = g(x, y) \]

Logic
Provability theory
Math
Characteristic PDE

DI\(\geq\) \(\wedge\) \(\vee\)

JLogComput’10, CAV’08, FMSD’09, LMCS’12, LICS’12, ITP’12, JAR’17
Differential Invariant

\[ Q \vdash [x' := f(x)](P)' \]

\[ P \vdash [x' = f(x) \& Q]P \]
Differential Invariant

\[
\begin{align*}
Q \vdash [x' := f(x)](P)' \\
P \vdash [x' = f(x) & Q]P
\end{align*}
\]

Differential Cut

\[
\begin{align*}
P \vdash [x' = f(x) & Q]C \\
P \vdash [x' = f(x) & Q \land C]P
\end{align*}
\]

\[
P \vdash [x' = f(x) & Q]P
\]

JLogComput'10, LMCS'12, LICS'12, JAR'17
Differential Invariant

\[
Q \vdash [x' := f(x)](P)'
\]

\[
\frac{P \vdash [x' = f(x) \land Q]P}{P \vdash [x' = f(x) \land Q]P}
\]

Differential Cut

\[
P \vdash [x' = f(x) \land Q]C \quad P \vdash [x' = f(x) \land Q \land C]P
\]

\[
\frac{P \vdash [x' = f(x) \land Q]P}{P \vdash [x' = f(x) \land Q]P}
\]

Differential Ghost

\[
P \leftrightarrow \exists y \ G \quad G \vdash [x' = f(x), y' = g(x, y) \land Q]G
\]

\[
\frac{P \vdash [x' = f(x) \land Q]P}{P \vdash [x' = f(x) \land Q]P}
\]
Differential Invariants for Differential Equations

Differential Invariant
\[
Q \vdash [x' := f(x)](P)'
\]
\[
P \vdash [x' = f(x) \& Q]P
\]

Differential Cut
\[
P \vdash [x' = f(x) \& Q]C
\]
\[
P \vdash [x' = f(x) \& Q \land C]P
\]
\[
P \vdash [x' = f(x) \& Q]P
\]

Differential Ghost
\[
P \leftrightarrow \exists y \ G
\]
\[
G \vdash [x' = f(x), y' = g(x, y) \& Q]G
\]
\[
P \vdash [x' = f(x) \& Q]P
\]

if new \( y' = g(x, y) \) has a global solution

JLogComput’10, LMCS’12, LICS’12, JAR’17
$\omega^2 x^2 + y^2 \leq c^2 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y \& \omega \geq 0 \land d \geq 0] \omega^2 x^2 + y^2 \leq c^2$
\[\omega^2 x^2 + y^2 \leq c^2 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y \& \omega \geq 0 \land d \geq 0] \omega^2 x^2 + y^2 \leq c^2\]
\[ \omega \geq 0 \land d \geq 0 \vdash [x' := y][y' := -\omega^2 x - 2d\omega y] 2\omega^2 xx' + 2yy' \leq 0 \]
\[ \omega^2 x^2 + y^2 \leq c^2 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y \land \omega \geq 0 \land d \geq 0] \omega^2 x^2 + y^2 \leq c^2 \]
\[
\omega \geq 0 \land d \geq 0 \vdash 2\omega^2 xy + 2y(-\omega^2 x - 2d\omega y) \leq 0
\]

\[
\omega \geq 0 \land d \geq 0 \vdash [x':=y][y':=-\omega^2 x - 2d\omega y] 2\omega^2 xx' + 2yy' \leq 0
\]

\[
\omega^2 x^2 + y^2 \leq c^2 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y \land \omega \geq 0 \land d \geq 0] \omega^2 x^2 + y^2 \leq c^2
\]

Damped oscillator
\[
\omega \geq 0 \land d \geq 0 \vdash 2\omega^2 xy + 2y(-\omega^2 x - 2d\omega y) \leq 0
\]

\[
\omega \geq 0 \land d \geq 0 \vdash [x' := y][y' := -\omega^2 x - 2d\omega y] 2\omega^2 xx' + 2yy' \leq 0
\]

\[
\omega^2 x^2 + y^2 \leq c^2 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y \land \omega \geq 0 \land d \geq 0] \omega^2 x^2 + y^2 \leq c^2
\]
\[ \omega \geq 0 \land d \geq 0 \vdash 2 \omega^2 x y + 2y(-\omega^2 x - 2d \omega y) \leq 0 \]

\[ \omega \geq 0 \land d \geq 0 \vdash [x':=y][y':=-\omega^2 x - 2d \omega y] 2\omega^2 xx' + 2yy' \leq 0 \]

\[ \omega^2 x^2 + y^2 \leq c^2 \vdash [x' = y, y' = -\omega^2 x - 2d \omega y \land \omega \geq 0 \land d \geq 0] \omega^2 x^2 + y^2 \leq c^2 \]
\[ \omega^2 x^2 + y^2 \leq c^2 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y, d' = 7 & \omega \geq 0] \omega^2 x^2 + y^2 \leq c^2 \]
\[
\omega^2 x^2 + y^2 \leq c^2 \vdash [x' = y, y' = -\omega^2 x - 2d \omega y, d' = 7 \& \omega \geq 0] \omega^2 x^2 + y^2 \leq c^2
\]
\[ \omega^2 x^2 + y^2 \leq c^2 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y, d' = 7 & \omega \geq 0 \land d \geq 0] \omega^2 x^2 + y^2 \leq c^2 \]

\[ \omega^2 x^2 + y^2 \leq c^2 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y, d' = 7 & \omega \geq 0] \omega^2 x^2 + y^2 \leq c^2 \]

increasingly damped oscillator
Differential Cuts for Differential Equations

\[
\omega^2 x^2 + y^2 \leq c^2 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y, d' = 7 \land \omega \geq 0 \land d \geq 0] \omega^2 x^2 + y^2 \leq c^2
\]

\[
\omega^2 x^2 + y^2 \leq c^2 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y, d' = 7 \land \omega \geq 0] \omega^2 x^2 + y^2 \leq c^2
\]

\[
d \geq 0 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y, d' = 7 \land \omega \geq 0] d \geq 0
\]

increasingly damped oscillator
\[ \omega^2 x^2 + y^2 \leq c^2 \vdash [x' = y, y' = -\omega^2 x - 2d \omega y, d' = 7 & \omega \geq 0] \omega^2 x^2 + y^2 \leq c^2 \]

\[ \omega^2 x^2 + y^2 \leq c^2 \vdash [x' = y, y' = -\omega^2 x - 2d \omega y, d' = 7 & \omega \geq 0] \omega^2 x^2 + y^2 \leq c^2 \]

\[ \omega \geq 0 \vdash [d' := 7] d' \geq 0 \]

\[ d \geq 0 \vdash [x' = y, y' = -\omega^2 x - 2d \omega y, d' = 7 & \omega \geq 0] d \geq 0 \]

increasingly damped oscillator
\[
\begin{align*}
\omega^2 x^2 + y^2 &\leq c^2 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y, d' = 7 &\omega \geq 0 \land d \geq 0] \omega^2 x^2 + y^2 &\leq c^2 \\
\omega^2 x^2 + y^2 &\leq c^2 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y, d' = 7 &\omega \geq 0] \omega^2 x^2 + y^2 &\leq c^2
\end{align*}
\]

\[
\begin{align*}
\omega \geq 0 \vdash 7 &\geq 0 \\
\omega \geq 0 \vdash [d' := 7] d' &\geq 0 \\
d \geq 0 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y, d' = 7 &\omega \geq 0] d &\geq 0
\end{align*}
\]

increasingly damped oscillator
\[ \omega^2 x^2 + y^2 \leq c^2 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y, d' = 7 \land \omega \geq 0 \land d \geq 0] \omega^2 x^2 + y^2 \leq c^2 \]

\[ \omega^2 x^2 + y^2 \leq c^2 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y, d' = 7 \land \omega \geq 0] \omega^2 x^2 + y^2 \leq c^2 \]

\[ \omega \geq 0 \vdash 7 \geq 0 \]

\[ \omega \geq 0 \vdash [d' := 7] \quad d' \geq 0 \]

\[ d \geq 0 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y, d' = 7 \land \omega \geq 0] \quad d \geq 0 \]

increasingly damped oscillator
\[ \omega \geq 0 \land d \geq 0 \vdash [x' = y][y' = -\omega^2 x - 2d \omega y] \ 2\omega^2 xx' + 2yy' \leq 0 \]

\[ \omega^2 x^2 + y^2 \leq c^2 \vdash [x' = y, y' = -\omega^2 x - 2d \omega y, d' = 7 \land \omega \geq 0 \land d \geq 0] \omega^2 x^2 + y^2 \leq c^2 \]

\[ * \]

\[ \omega \geq 0 \vdash 7 \geq 0 \]

\[ \omega \geq 0 \vdash [d' = 7]d' \geq 0 \]

\[ d \geq 0 \vdash [x' = y, y' = -\omega^2 x - 2d \omega y, d' = 7 \land \omega \geq 0] d \geq 0 \]

increasingly damped oscillator
\[ \omega \geq 0 \land d \geq 0 \vdash 2\omega^2 x y + 2y(-\omega^2 x - 2d\omega y) \leq 0 \]

\[ \omega \geq 0 \land d \geq 0 \vdash [x' := y][y' := -\omega^2 x - 2d\omega y] 2\omega^2 xx' + 2yy' \leq 0 \]

\[ \omega^2 x^2 + y^2 \leq c^2 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y, d' = 7 \land \omega \geq 0] \omega^2 x^2 + y^2 \leq c^2 \]

\[ \omega^2 x^2 + y^2 \leq c^2 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y, d' = 7 \land \omega \geq 0] \omega^2 x^2 + y^2 \leq c^2 \]

* 

\[ \omega \geq 0 \vdash 7 \geq 0 \]

\[ \omega \geq 0 \vdash [d' := 7] d' \geq 0 \]

\[ d \geq 0 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y, d' = 7 \land \omega \geq 0] d \geq 0 \]

increasingly damped oscillator
\begin{align*}
\omega \geq 0 \land d \geq 0 & \vdash 2\omega^2 xy + 2y(-\omega^2 x - 2d\omega y) \leq 0 \\
\omega \geq 0 \land d \geq 0 & \vdash [x' := y][y' := -\omega^2 x - 2d\omega y] \ 2\omega^2 xx' + 2yy' \leq 0 \\
\omega^2 x^2 + y^2 \leq c^2 & \vdash [x' = y, y' = -\omega^2 x - 2d\omega y, d' = 7 \land \omega \geq 0] \omega^2 x^2 + y^2 \leq c^2 \\
\omega \geq 0 & \vdash 7 \geq 0 \\
\omega \geq 0 & \vdash [d' := 7] \ d' \geq 0 \\
d \geq 0 & \vdash [x' = y, y' = -\omega^2 x - 2d\omega y, d' = 7 \land \omega \geq 0] \ d \geq 0
\end{align*}

increasingly damped oscillator
Could repeatedly diffcut in formulas to help the proof
Differentials

Syntax
\[ e ::= x \mid x' \mid c \mid e + k \mid e \cdot k \mid (e)' \]

Semantics
\[ \omega[(e)'] = \]
### Syntax
\[ e ::= x \mid x' \mid c \mid e + k \mid e \cdot k \mid (e)' \]

### Semantics
\[ \omega[(e)'] = \frac{d\omega[e]}{dt} \]
Differentials

Syntax
\[ e ::= x \mid x' \mid c \mid e + k \mid e \cdot k \mid (e)' \]

Semantics
\[ \omega[(e)'] = \frac{d\omega[e]}{dt} \text{ no time!} \]
Differentials

Syntax

\[ e ::= x \mid x' \mid c \mid e + k \mid e \cdot k \mid (e)' \]

Semantics

\[ \omega[(e)'] = \sum_x \omega(x') \frac{\partial[e]}{\partial x}(\omega) \rightarrow \mathbb{R} \]
Differentials

Syntax

\[ e ::= x \mid x' \mid c \mid e + k \mid e \cdot k \mid (e)' \]

Semantics

\[ \omega[(e)'] = \sum_x \omega(x') \frac{\partial[e]}{\partial x}(\omega) \]

Axioms

\begin{align*}
(e + k)' & = (e)' + (k)' \\
(e \cdot k)' & = (e)' \cdot k + e \cdot (k)' \\
(c())' & = 0 & \text{for constants/numbers } c() \\
(x)' & = x' & \text{for variables } x \in \mathcal{V}
\end{align*}
<table>
<thead>
<tr>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e ::= x \mid x' \mid c \mid e + k \mid e \cdot k \mid (e)'$</td>
</tr>
<tr>
<td><strong>Semantics</strong></td>
</tr>
<tr>
<td>$\omega[(e)'] = \sum_x \omega(x') \frac{\partial[e]}{\partial x}(\omega)$</td>
</tr>
<tr>
<td><strong>Axioms</strong></td>
</tr>
<tr>
<td>$(e + k)' = (e)' + (k)'$</td>
</tr>
<tr>
<td>$(e \cdot k)' = (e)' \cdot k + e \cdot (k)'$</td>
</tr>
<tr>
<td>$(c())' = 0$ for constants/numbers $c()$</td>
</tr>
<tr>
<td>$(x)' = x'$ for variables $x \in \mathcal{V}$</td>
</tr>
<tr>
<td><strong>ODE</strong></td>
</tr>
<tr>
<td>$[x' = f(x) &amp; Q] = {(\varphi(0), \varphi(r)) : \varphi \models x' = f(x) \land Q$ for some $\varphi : [0, r] \to \mathcal{S}$, some $r \in \mathbb{R}}$</td>
</tr>
<tr>
<td>$\varphi(z)(x') = \frac{d\varphi(t)(x)}{dt}(z)$ ...</td>
</tr>
</tbody>
</table>
Differential Substitution Lemmas

Lemma (Differential lemma) (Differential value vs. Time-derivative)

If $\varphi \models x' = f(x) \land Q$ for duration $r > 0$, then for all $0 \leq z \leq r$, $FV(e) \subseteq \{x\}$:

$$\varphi(z)[(e)'] = \left. \frac{d\varphi(t)[e]}{dt} \right|_{z}$$

Lemma (Differential assignment) (Effect on Differentials)

If $\varphi \models x' = f(x) \land Q$ then $\varphi \models P \leftrightarrow [x' := f(x)]P$

Lemma (Derivations) (Equations of Differentials)

\[(e + k)' = (e)' + (k)'
\[(e \cdot k)' = (e)' \cdot k + e \cdot (k)'
\[(c())' = 0 \quad \text{for constants/numbers } c()
\[(x)' = x' \quad \text{for variables } x \in \mathcal{V}

André Platzer (CMU)

Dynamic Logic for Dynamical Systems

MOD'17 43 / 59
Differential Substitution Lemmas

Lemma (Differential lemma) (Differential value vs. Time-derivative)
If $\varphi \models x' = f(x) \land Q$ for duration $r > 0$, then for all $0 \leq z \leq r$, $FV(e) \subseteq \{x\}$:

\[
\varphi(z)[(e)'] = \frac{d\varphi(t)[e]}{dt}(z)
\]

Lemma (Differential assignment) (Effect on Differentials)
If $\varphi \models x' = f(x) \land Q$ then $\varphi \models P \iff [x' := f(x)]P$

Lemma (Derivations) (Equations of Differentials)
\[
\begin{align*}
(e + k)' &= (e)' + (k)' \\
(e \cdot k)' &= (e)' \cdot k + e \cdot (k)' \\
(c())' &= 0 \\
(x)' &= x'
\end{align*}
\]
for constants/numbers $c()$
for variables $x \in \mathcal{V}$
**Lemma (Differential lemma) (Differential value vs. Time-derivative)**

If $\varphi \models x' = f(x) \land Q$ for duration $r > 0$, then for all $0 \leq z \leq r$, $FV(e) \subseteq \{x\}$:

$$\varphi(z)[(e)'] = \frac{d\varphi(t)[e]}{dt}(z)$$

**Lemma (Differential assignment) (Effect on Differentials)**

$$DE \ [x' = f(x) \land Q]P \leftrightarrow [x' = f(x) \land Q][x' := f(x)]P$$

**Lemma (Derivations) (Equations of Differentials)**

\begin{align*}
  +' & \quad (e + k)' = (e)' + (k)' \\
  \cdot' & \quad (e \cdot k)' = (e)' \cdot k + e \cdot (k)' \\
  c' & \quad (c())' = 0 \\
  x' & \quad (x)' = x'
\end{align*}
### Differential Substitution Lemmas \( \rightsquigarrow \) Proofs

#### Lemma (Differential lemma) (Differential value vs. Time-derivative)

If \( \varphi \models x' = f(x) \land Q \) for duration \( r > 0 \), then for all \( 0 \leq z \leq r \), \( FV(e) \subseteq \{x\} \):

\[
\varphi(z)[(e)'] = \frac{d\varphi(t)[e]}{dt}(z)
\]

#### Lemma (Differential assignment) (Effect on Differentials)

If \( \varphi \models x' = f(x) \land Q \) then \( \varphi \models P \iff [x' := f(x)]P \)

### Axiomatics

**DE** \( [x' = f(x) \land Q]P \leftrightarrow [x' = f(x) \land Q][x' := f(x)]P \)

**DI** \([x' = f(x) \land Q] \ e \geq 0 \iff [?Q]e \geq 0 \) \( \iff [x' = f(x) \land Q] (e)' \geq 0 \)
Differential Equation Axioms & Differential Axioms

\[ x' = f(x) \land Q \]

\[ P \iff [x' = f(x) \land Q]P \]

\[ \leftarrow \ [x' = f(x) \land Q]C \]

\[ P \iff [x' = f(x) \land Q][x' := f(x)]P \]

\[ ( [x' = f(x) \land Q]P \iff [? Q]P) \leftarrow [x' = f(x) \land Q](P') \]

\[ P \iff \exists y [x' = f(x), y' = a(x)y + b(x) \land Q]P \]

\[ P \iff \forall t \geq 0 \ ((\forall 0 \leq s \leq t q(x+c()s)) \rightarrow [x := x+c()t]P) \]

\[ +' (e + k)' = (e)' + (k)' \]

\[ \cdot' (e \cdot k)' = (e)' \cdot k + e \cdot (k)' \]

\[ o' [y := g(x)][y' := 1]((f(g(x)))' = (f(y))' \cdot (g(x))') \]
Differential Equations cannot leave their evolution domains. Implies:

\[
[x' = f(x) & Q]P \leftrightarrow [x' = f(x) & Q](Q \rightarrow P)
\]
### Axiom (Differential Cut)  
**DC**

\[
\begin{align*}
([x' = f(x) \& Q]P & \iff [x' = f(x) \& Q \land C]P) \\
\iff [x' = f(x) \& Q]C
\end{align*}
\]

---

**DC** is a cut for differential equations.  
**DC** is a differential modal modus ponens K.  
Can’t leave \( C \), then might as well restrict state space to \( C \).
Axiom (Differential Invariant) (JAR’17)

\[
\text{DI}\quad ([\dot{x} = f(x) \& Q]P \iff [?Q]P) \iff [\dot{x} = f(x) \& Q](P)'
\]

Differential invariant: if \( P \) true now and if differential \((P)’\) true always

What’s the differential of a formula???

What’s the meaning of a differential term . . . in a state???
Differential Equation Axioms

Axiom (Differential Effect) (JAR'17)

\[ \text{DE} \quad [x' = f(x) \& Q]P \iff [x' = f(x) \& Q][x' := f(x)]P \]

Effect of differential equation on differential symbol \( x' \)

\( [x' := f(x)] \) instantly mimics continuous effect \( [x' = f(x)] \) on \( x' \)

\( [x' := f(x)] \) selects vector field \( x' = f(x) \) for subsequent differentials
Axiom (Differential Ghost)

\[ DG \ [x' = f(x) & Q]P \iff \exists y \ [x' = f(x), y' = a(x)y + b(x) & Q]P \]

Differential ghost/auxiliaries: extra differential equations that exist
Can cause new invariants
“Dark matter” counterweight to balance conserved quantities
Example (Differential ghost proof)

\[ x > 0 \vdash [x' = -x] x > 0 \]
Example: Differential Ghosts

Example (Differential ghost proof)

MR
\[ x > 0 \vdash \exists y \ [x' = -x, y' = \overbrace{\text{Differential ghost}}^\text{cloud}] x > 0 \]

DG
\[ x > 0 \vdash [x' = -x] x > 0 \]

André Platzer (CMU)
Example: Differential Ghosts

Example (Differential ghost proof)

\[ \forall \in \mathbb{R} x y^2 = 1 \vdash x > 0 \quad \exists \in \mathbb{R}, \text{cut} \quad x > 0 \vdash \exists y [x' = -x, y' = y] x y^2 = 1 \]

MR

\[ x > 0 \vdash \exists y [x' = -x, y' = y] x > 0 \]

DG

\[ x > 0 \vdash [x' = -x] x > 0 \]
Example: Differential Ghosts

Example (Differential ghost proof)

\[
\begin{align*}
\mathbb{R}^{xy^2=1} &\vdash x > 0 & \exists R, \text{cut} &\quad x > 0 \vdash \exists y [x' = -x, y' = \text{\ding{51}}] xy^2 = 1 \\
\exists R, \text{cut} &\quad x > 0 \vdash \exists y [x' = -x, y' = \text{\ding{51}}] x > 0 \\
\text{MR} &\quad x > 0 \vdash \exists y [x' = -x, y' = \text{\ding{51}}] x > 0 \\
\text{DG} &\quad x > 0 \vdash [x' = -x] x > 0
\end{align*}
\]

\[x_0 \quad x \quad t \]
**Example (Differential ghost proof)**

\[
\begin{align*}
\text{Example: Differential Ghosts} \\
\text{Example (Differential ghost proof)} \\
\begin{array}{l}
\begin{array}{c}
\begin{array}{c}
\mathbb{R} \\
\times
\end{array}
\vdash
\begin{array}{c}
x y^2 = 1 \\
\times
\end{array}
\end{array}
\end{array}
\end{align*}
\]

\[
\begin{align*}
\begin{array}{l}
\begin{array}{c}
\begin{array}{c}
\mathbb{R} \\
\times
\end{array}
\vdash
\begin{array}{c}
x > 0 \\
\exists \mathbb{R}, \text{cut}
\end{array}
\end{array}
\end{array}
\end{align*}
\]

\[
\begin{align*}
\begin{array}{l}
\begin{array}{c}
\begin{array}{c}
\text{MR} \\
\text{DG}
\end{array}
\vdash
\begin{array}{c}
x > 0 \\
\exists y [x' = -x, y' = \text{cloud}] x > 0
\end{array}
\end{array}
\end{array}
\end{align*}
\]

\[
\begin{align*}
\begin{array}{l}
\begin{array}{c}
\text{MR} \\
\text{DG}
\end{array}
\vdash
\begin{array}{c}
x > 0 \\
\exists y [x' = -x, y' = \text{cloud}] x > 0
\end{array}
\end{array}
\end{align*}
\]

\[
\begin{align*}
\begin{array}{l}
\begin{array}{c}
\begin{array}{c}
\text{MR} \\
\text{DG}
\end{array}
\vdash
\begin{array}{c}
x > 0 \\
\exists y [x' = -x] x > 0
\end{array}
\end{array}
\end{array}
\end{align*}
\]
Example: Differential Ghosts

Example (Differential ghost proof)

\[ \begin{align*}
\exists \mathbb{R} x y^2 = 1 & \vdash x > 0 \\
\exists \mathbb{R}, \text{cut} \quad x > 0 & \vdash \exists y \ [x' = -x, y' = \_\_\_] xy^2 = 1 \\
\text{MR} \quad x > 0 & \vdash \exists y \ [x' = -x, y' = \_\_\_] x > 0 \\
\text{DG} \quad x > 0 & \vdash [x' = -x] x > 0
\end{align*} \]
Example (Differential ghost proof)

\[
\begin{align*}
\mathbb{R} & \quad \vdash -xy^2 + 2xy = 0 \\
[':=] & \quad \vdash [x' := -x, y' := y] x' y^2 + x2y y' = 0 \\
\ast & \quad \vdash [x' := -x, y' := y] \quad \vdash [x' := -x] \\
dl & \quad \vdash x > 0 \vdash \exists y [x' = -x, y' = y] xy^2 = 1 \\
\exists R, \text{cut} & \quad \vdash x > 0 \vdash \exists y [x' = -x, y' = y] xy^2 = 1 \\
MR & \quad \vdash x > 0 \vdash \exists y [x' = -x, y' = y] x > 0 \\
DG & \quad \vdash x > 0 \vdash [x' = -x] x > 0
\end{align*}
\]
**Example (☆ Differential ghost proof)**

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ \mathbb{R} ]</td>
<td>( \mathbb{R} ) ( \vdash -xy^2 + 2xy = 0 )</td>
</tr>
<tr>
<td>([':=])</td>
<td>( \vdash [x':=-x][y':=]x'y^2 + x2yy' = 0 )</td>
</tr>
<tr>
<td>dl</td>
<td>( xy^2=1 \vdash [x' = -x, y' = ] xy^2 = 1 )</td>
</tr>
<tr>
<td>( \mathbb{R} ) ( \vdash x &gt; 0 )</td>
<td>( \exists R, cut ) ( x &gt; 0 \vdash \exists y [x' = -x, y' = ] xy^2 = 1 )</td>
</tr>
<tr>
<td>MR</td>
<td>( x &gt; 0 \vdash \exists y [x' = -x, y' = ] x &gt; 0 )</td>
</tr>
<tr>
<td>DG</td>
<td>( x &gt; 0 \vdash [x' = -x] x &gt; 0 )</td>
</tr>
</tbody>
</table>

\[ x_0 \]

\( x' = -x \)

\( 0 \)

\( t \)
Example (Differential ghost proof)

\[
\begin{align*}
\text{dx} y^2 & = 1 \vdash x > 0 \\
\exists \text{R,cut} & \quad x > 0 \vdash \exists y [x' = -x, y' = \frac{y}{2}] x > 0 \\
\text{MR} & \quad x > 0 \vdash [x' = -x] x > 0
\end{align*}
\]

\[
\begin{align*}
\exists \text{R,cut} & \quad \vdash [x' = -x, y' = \frac{y}{2}] x y^2 \quad x > 0 \\
\text{dl} & \quad \vdash [x' = -x] x y^2 = 1 \\
\text{dI} & \quad \vdash -x y^2 + 2x y \frac{y}{2} = 0 \\
\text{'=} & \quad \vdash [x' = -x] y = 0 \\
\end{align*}
\]
Example: Differential Ghosts

**Example (Differential ghost proof)**

\[ \mathbb{R} \quad \vdash -xy^2 + 2xy\frac{y}{2} = 0 \]

\[ [\leftarrow:] \quad \vdash [x' = -x][y' = \frac{y}{2}] x'y^2 + 2yy' = 0 \]

\[ \mathbb{R} \quad xy^2 = 1 \quad \vdash [x' = -x, y' = \frac{y}{2}] xy^2 = 1 \]

\[ [\leftarrow:] \quad \vdash [x' = -x][y' = \frac{y}{2}] xy^2 = 1 \]

\[ \mathbb{R} \quad xy^2 = 1 \quad \vdash x > 0 \]

\[ \exists \mathbb{R}, \text{cut} \quad x > 0 \quad \vdash \exists y [x' = -x, y' = \frac{y}{2}] xy^2 = 1 \]

\[ \exists \mathbb{R}, \text{cut} \quad x > 0 \quad \vdash \exists y [x' = -x, y' = \frac{y}{2}] x > 0 \]

\[ \mathbb{R} \quad xy^2 = 1 \quad \vdash x > 0 \]

\[ \exists \mathbb{R}, \text{cut} \quad x > 0 \quad \vdash \exists y [x' = -x, y' = \frac{y}{2}] x > 0 \]

\[ \mathbb{R} \quad xy^2 = 1 \quad \vdash x > 0 \]

\[ \exists \mathbb{R}, \text{cut} \quad x > 0 \quad \vdash \exists y [x' = -x, y' = \frac{y}{2}] x > 0 \]
Example: Differential Ghosts

Example (Differential ghost proof)

\[ \mathbb{R} \quad \vdash -xy^2 + 2xy \frac{y}{2} = 0 \]

\[ ['=':] \]

\[ \vdash [x'=-x][y'=-\frac{y}{2}] x'y^2 + x2yy' = 0 \]

\[ \mathbb{R} \quad \vdash xy^2 = 1 \]

\[ \vdash [x'=-x, y'=-\frac{y}{2}] xy^2 = 1 \]

\[ \exists \mathbb{R, cut} \]

\[ x > 0 \quad \vdash \exists y \ [x' = -x, y' = \frac{y}{2}] xy^2 = 1 \]

\[ \exists \mathbb{R, cut} \]

\[ x > 0 \quad \vdash \exists y \ [x' = -x, y' = \frac{y}{2}] x > 0 \]

\[ x > 0 \quad \vdash [x' = -x] x > 0 \]

\[ \exists y \]

\[ x > 0 \quad \vdash \exists y \ [x' = -x] x > 0 \]

\[ x_0 \]

\[ xy^2 = 1 \]
Ex: Parachute Open or Keep Closed

Conservatively bounded next velocity above parachute’s limit velocity. Limit by differential ghost:

\[ y' = -\frac{p}{2} (v - \sqrt{\frac{g}{p}}) y^2 (v + \sqrt{\frac{g}{p}}) > 0 \] implies 

\[ v \geq v_{old} - gt \] if closed.
Example (Parachute)

\[
((? (Q \land r = a) \cup r := p); t := 0; \\
\{ x' = v, v' = -g + rv^2, t' = 1 \land t \leq T \land x \geq 0 \land v < 0 \})^*
\]
Example (Parachute)

\[ \rightarrow [ ( ( ? ( Q \land r = a ) \cup r := p ) ; t := 0 ; \\
\{ x' = v , v' = -g + rv^2 , t' = 1 \land t \leq T \land x \geq 0 \land v < 0 \} )^* ] \\
( x = 0 \rightarrow v \geq m ) \]
Ex: Parachute Open or Keep Closed

Example (Parachute)

$$\rightarrow [((? (Q \land r = a) \cup r := p); t := 0;
\{ x' = v, v' = -g + rv^2, t' = 1 & t \leq T \land x \geq 0 \land v < 0 \})^*]
(x = 0 \rightarrow v \geq m)$$
Ex: Parachute Open or Keep Closed

\[ Q \equiv v - gT > -\sqrt{g/p} \]

Conservatively bounded next velocity above parachute’s limit velocity.

Example (Parachute)

\[ m < -\sqrt{g/p} \rightarrow \left[ \left( (? (Q \land r = a) \cup r := p) ; t := 0 ; \right. \right. \]
\[ \left. \left. \{ x' = v, v' = -g + rv^2, t' = 1 \land t \leq T \land x \geq 0 \land v < 0 \} \right) ^* \right] \]

\[ (x = 0 \rightarrow v \geq m) \]
Ex: Parachute Open or Keep Closed

\[ Q \equiv v - gT > -\sqrt{g/p} \]

Conservatively bounded next velocity above parachute’s limit velocity.
Limit by differential ghost:

\[ y' = -\frac{p}{2}(v - \sqrt{g/p}) \quad y^2(v + \sqrt{g/p}) > 0 \]

Example (Parachute)

\[ m < -\sqrt{g/p} \rightarrow \left[ ((?Q \land r = a) \lor r := p); t := 0; \right. \]
\[ \left. \{x' = v, v' = -g + rv^2, t' = 1 \land t \leq T \land x \geq 0 \land v < 0\}^* \right] \]

\[ (x = 0 \rightarrow v \geq m) \]
Ex: Parachute Open or Keep Closed

\[ Q \equiv v - gT > -\sqrt{\frac{g}{p}} \]

Conservatively bounded next velocity above parachute’s limit velocity.

Limit by differential ghost:

\[ y' = -\frac{p}{2}(v - \sqrt{\frac{g}{p}}) \quad y^2(v + \sqrt{\frac{g}{p}}) > 0 \]

\[ v \geq v_{old} - gt \text{ if closed} \]

Example (Parachute)

\[ m < -\sqrt{\frac{g}{p}} \rightarrow (((Q \land r = a) \lor r := p); \ t := 0; \ \
\{x' = v, \ v' = -g + rv^2, \ t' = 1 \land t \leq T \land x \geq 0 \land v < 0\})^* \]

\[ (x = 0 \rightarrow v \geq m) \]
Outline (CPS Application Highlights)

1 CPS are Multi-Dynamical Systems
   - Hybrid Systems / Games / Stochastic / Distributed Hybrid Systems

2 Differential Dynamic Logic
   - Syntax
   - Semantics
   - Example: Car Control Design

3 Dynamic Axioms for Dynamical Systems
   - Axiomatics
   - Example: Safe Car Control
   - Soundness and Completeness

4 Differential Invariants for Differential Equations
   - Differential Invariants
   - Example: Elementary Differential Invariants
   - Differential Axioms
   - Example: Differential Ghosts

5 Applications

6 Summary
Airborne Collision Avoidance System ACAS X: Verify

- Developed by the FAA to replace current TCAS in aircraft
- Approximately optimizes Markov Decision Process on a grid
- Advisory from lookup tables with numerous 5D interpolation regions

- Identified safe region for each advisory symbolically
- Proved safety for hybrid systems flight model in KeYmaera X

TACAS'15, EMSOFT'15, STTT'17
ACAS X table comparison shows safe advisory in 97.7% of the 648,591,384,375 states compared (15,160,434,734 counterexamples).

ACAS X issues DNC advisory, which induces collision unless corrected

TACAS’15, EMSOFT’15, STTT’17
Conservative, so too many counterexamples
Settle for: safe for a little while, with safe future advisory possibility
Safeable advisory: a subsequent advisory can safely avoid collision

1. Identified safeable region for each advisory symbolically
2. Proved safety for hybrid systems flight model in KeYmaera X
ACAS X table comparison shows safeable advisory in more of the 648,591,384,375 states compared (31 to 899 $10^6$ counterexamples).

ACAS X issues Maintain advisory instead of CL1500
ACAS X table comparison shows safeable advisory in more of the 648,591,384,375 states compared (31 to 899 $10^6$ counterexamples).

ACAS X issues Maintain advisory instead of CL1500.
Application Highlights

Obstacle Avoidance + Ground Navigation

Airborne Collision Avoidance (ACAS X)

Train Control Brakes

Ship Cooling

\[ t_0 = t \]

\[ x' = f(x) \]

\[ y' = g(x, y) \]

Acknowledgments

André Platzer (CMU)
FM’11, LMCS’12, ICCPS’12, ITSC’11, ITSC’13, IJCAR’12
Verified CPS Applications: Robots

André Platzer (CMU)
Dynamic Logic for Dynamical Systems
HSCC’13, RSS’13, CADE’12, IJRR’17
undergrads in *Foundations of Cyber-Physical Systems* course
KeYmaera X aXiomatic Tactical Theorem Prover for CPS

Proof search

KeYmaera X

generates proofs

ModelPlex proof

Model

Safety

Compliance

Monitor

ctrl: \( a := -b \);
plant: \( x'' = a \)

Trustworthy

- Uniform substitution
- Sound & complete
- Small core: 1700 LOC

Flexible

- Proof automation
- Interactive UI
- Programmable

Customizable

- Scala+Java API
- Command line
- REST API

André Platzer (CMU)

Dynamic Logic for Dynamical Systems

MOD’17 54 / 59
Disclaimer: Self-reported estimates of the soundness-critical lines of code + rules
Outline (Dynamic Logic for Dynamical Systems)

1. CPS are Multi-Dynamical Systems
   - Hybrid Systems / Games / Stochastic / Distributed Hybrid Systems

2. Differential Dynamic Logic
   - Syntax
   - Semantics
   - Example: Car Control Design

3. Dynamic Axioms for Dynamical Systems
   - Axiomatics
   - Example: Safe Car Control
   - Soundness and Completeness

4. Differential Invariants for Differential Equations
   - Differential Invariants
   - Example: Elementary Differential Invariants
   - Differential Axioms
   - Example: Differential Ghosts

5. Applications

6. Summary
Acknowledgments

Students and postdocs of the Logical Systems Lab at Carnegie Mellon
Brandon Bohrer, Nathan Fulton, Sarah Loos, João Martins, Yong Kiam Tan
Khalil Ghorbal, Jean-Baptiste Jeannin, Stefan Mitsch
Logical foundations make a big difference for CPS, and vice versa

\[ dL = DL + HP \]

- Strong analytic foundations
- Practical reasoning advances
- Significant applications
- Catalyze many science areas

1. Multi-dynamical systems
2. Combine simple dynamics
3. Tame complexity
4. Complete axiomatization

Numerous wonders remain to be discovered
Logical foundations make a big difference for CPS, and vice versa.

\[
[\alpha] \varphi \quad \alpha \quad \varphi
\]

- Strong analytic foundations
- Practical reasoning advances
- Significant applications
- Catalyze many science areas

Numerous wonders remain to be discovered.
Future CPS Challenges

Numerous wonders remain to be discovered

- Scalable continuous stochastics
- Concurrent CPS
- Real arithmetic: Scalable and verified
- Verified CPS implementations, ModelPlex
- Correct CPS execution
- CPS-conducive tactic languages+libraries
- Tactics exploiting CPS structure/linearity/
- Invariant generation
- Tactics & proofs for reachable set computations
- Parallel proof search & disprovers
- Correct model transformation
- Inspiring applications

CPSs deserve proofs as safety evidence!

CADE’11
CADE’09
FMSD’16
ITP’17

André Platzer (CMU)                  Dynamic Logic for Dynamical Systems
MOD’17  58 / 59
Outline

7 Differential Invariant Soundness Proof
- Differential Radical Invariants
### Lemma (Differential lemma)  (Differential value vs. Time-derivative)

If $\varphi \models x' = f(x) \land Q$ for duration $r > 0$, then for all $0 \leq z \leq r$, $FV(e) \subseteq \{x\}$:

- **Syntactic**
  
  $$
  \varphi(z)[\lceil (e)' \rceil] = \frac{d\varphi(t)[e]}{dt}(z)
  $$

- **Analytic**

### Lemma (Differential assignment)  (Effect on Differentials)

If $\varphi \models x' = f(x) \land Q$ then $\varphi \models P \iff [x' := f(x)]P$

### Lemma (Derivations)  (Equations of Differentials)

\[
\begin{align*}
(e + k)' &= (e)' + (k)' \\
(e \cdot k)' &= (e)' \cdot k + e \cdot (k)' \\
(c())' &= 0 \\
(x)' &= x'
\end{align*}
\]

for constants/numbers $c()$

for variables $x \in \mathcal{V}$
Soundness Proof

Lemma (Differential lemma) (Differential value vs. Time-derivative)

If $\varphi \models x' = f(x) \land Q$ for duration $r > 0$, then for all $0 \leq z \leq r$, $FV(e) \subseteq \{x\}$:

$$\varphi(z)[(e)'] = \frac{d\varphi(t)[e]}{dt}(z)$$

Semantics

$$\omega[(e)'] = \sum_x \omega(x') \frac{\partial[e]}{\partial x}(\omega)$$

Definition (Hybrid program semantics)

$$[x' = f(x) \land Q] = \{ (\varphi(0), \varphi(r)) : \varphi(z) \models x' = f(x) \land Q \text{ for all } 0 \leq z \leq r \}$$

for a $\varphi : [0, r] \rightarrow S$ where $\varphi(z)(x') \overset{\text{def}}{=} \frac{d\varphi(t)(x)}{dt}(z)$
Soundness Proof

Lemma (Differential lemma) (Differential value vs. Time-derivative)

If $\varphi \models x' = f(x) \land Q$ for duration $r > 0$, then for all $0 \leq z \leq r$, $FV(e) \subseteq \{x\}$:

$$\varphi(z)[(e)'] = \frac{d\varphi(t)[e]}{dt}(z)$$

Semantics

$$\omega[(e)'] = \sum_x \omega(x') \frac{\partial[e]}{\partial x}(\omega)$$

Definition (Hybrid program semantics) ($[\cdot] : HP \rightarrow \wp(S \times S)$)

$$[x' = f(x) \land Q] = \{(\varphi(0), \varphi(r)) : \varphi(z) \models x' = f(x) \land Q \text{ for all } 0 \leq z \leq r \}
\text{ for a } \varphi : [0, r] \rightarrow S \text{ where } \varphi(z)(x') \overset{\text{def}}{=} \frac{d\varphi(t)(x)}{dt}(z)$$
Soundness Proof

**Lemma (Differential lemma) (Differential value vs. Time-derivative)**

If $\varphi \models x' = f(x) \land Q$ for duration $r > 0$, then for all $0 \leq z \leq r$, $FV(e) \subseteq \{x\}$:

$$\varphi(z)[(e)'] = \frac{d\varphi(t)[e]}{dt}(z)$$

$$\frac{d\varphi(t)[e]}{dt}(z) \overset{\text{chain}}{=} \sum_x \frac{\partial[e]}{\partial x}(\varphi(z)) \frac{d\varphi(t)(x)}{dt}(z)$$

**Semantics**

$$\omega[(e)'] = \sum_x \omega(x') \frac{\partial[e]}{\partial x}(\omega)$$

**Definition (Hybrid program semantics)**

$$[x' = f(x) \land Q] = \{ (\varphi(0), \varphi(r)) : \varphi(z) \models x' = f(x) \land Q \text{ for all } 0 \leq z \leq r \}$$

for a $\varphi : [0, r] \rightarrow S$ where $\varphi(z)(x') \overset{\text{def}}{=} \frac{d\varphi(t)(x)}{dt}(z)$
Soundness Proof

Lemma (Differential lemma) (Differential value vs. Time-derivative)

If $\phi \models x' = f(x) \land Q$ for duration $r > 0$, then for all $0 \leq z \leq r$, $FV(e) \subseteq \{x\}$:

$$\phi(z)[(e)'] = \frac{d\phi(t)[e]}{dt}(z)$$

$$\frac{d\phi(t)[e]}{dt}(z) \overset{\text{chain}}{=} \sum_x \frac{\partial[e]}{\partial x}(\phi(z)) \frac{d\phi(t)(x)}{dt}(z)$$

Semantics

$$\omega[(e)'] = \sum_x \omega(x') \frac{\partial[e]}{\partial x}(\omega)$$

Definition (Hybrid program semantics)

$$[x' = f(x) \land Q] = \{(\phi(0), \phi(r)) : \phi(z) \models x' = f(x) \land Q \text{ for all } 0 \leq z \leq r \text{ for a } \varphi : [0, r] \to S \text{ where } \varphi(z)(x') \overset{\text{def}}{=} \frac{d\varphi(t)(x)}{dt}(z)\}$$
Soundness Proof

**Lemma (Differential lemma) (Differential value vs. Time-derivative)**

If $\varphi \models x' = f(x) \land Q$ for duration $r > 0$, then for all $0 \leq z \leq r$, $FV(e) \subseteq \{x\}$:

$$
\varphi(z)[(e)'] = \frac{d\varphi(t)[e]}{dt}(z)
$$

$$
\frac{d\varphi(t)[e]}{dt}(z) \overset{\text{chain}}{=} \sum_x \frac{\partial[e]}{\partial x}(\varphi(z)) \frac{d\varphi(t)(x)}{dt}(z) = \sum_x \frac{\partial[e]}{\partial x}(\varphi(z)) \varphi(z)(x')
$$

**Semantics**

$$
\omega[(e)'] = \sum_x \omega(x') \frac{\partial[e]}{\partial x}(\omega)
$$

**Definition (Hybrid program semantics)**

$$
[x' = f(x) \land Q] = \{(\varphi(0), \varphi(r)) : \varphi(z) \models x' = f(x) \land Q \text{ for all } 0 \leq z \leq r
$$

for a $\varphi : [0, r] \to S$ where $\varphi(z)(x') \overset{\text{def}}{=} \frac{d\varphi(t)(x)}{dt}(z)$
Lemma (Differential lemma) (Differential value vs. Time-derivative)

If $\varphi \models x' = f(x) \land Q$ for duration $r > 0$, then for all $0 \leq z \leq r$, $FV(e) \subseteq \{x\}$:

$$\varphi(z)[(e)'] = \frac{d\varphi(t)[e]}{dt}(z)$$

$$\frac{d\varphi(t)[e]}{dt}(z) = \text{chain} = \sum_x \frac{\partial[e]}{\partial x}(\varphi(z)) \frac{d\varphi(t)(x)}{dt}(z) = \sum_x \frac{\partial[e]}{\partial x}(\varphi(z))\varphi(z)(x')$$

Semantics

$$\varphi(z)[(e)'] = \sum_x \varphi(z)(x') \frac{\partial[e]}{\partial x}(\varphi(z))$$

Definition (Hybrid program semantics) (\[
\square \cdot \] : HP $\rightarrow \wp(S \times S))$

\[\square [x' = f(x) \& Q] = \{(\varphi(0), \varphi(r)) : \varphi(z) \models x' = f(x) \land Q \text{ for all } 0 \leq z \leq r \text{ for a } \varphi : [0, r] \rightarrow S \text{ where } \varphi(z)(x') \overset{\text{def}}{=} \frac{d\varphi(t)(x)}{dt}(z)\}\]
Theorem (Differential radical invariant characterization)

\[
\begin{align*}
 h = 0 \rightarrow & \bigwedge_{i=1}^{N-1} h_p^{(i)} = 0 \\
 h = 0 \rightarrow & [x' = p] h = 0
\end{align*}
\]

characterizes all algebraic invariants, where \( N = \text{ord} \sqrt[\cdot]{h} \), i.e.

\[
 h_p^{(N)} = \sum_{i=0}^{N-1} g_i h_p^{(i)} \quad (g_i \in \mathbb{R}[x]) \quad h_p^{(i+1)} = [x' := p](h_p^{(i)})'
\]

Corollary (Algebraic Invariants Decidable)

Algebraic invariants of algebraic differential equations are decidable.

with Khalil Ghorbal TACAS'14
Case Study: Longitudinal Dynamics of an Airplane

Study (6th Order Longitudinal Flight Equations)

\[
\begin{align*}
    u' &= \frac{X}{m} - g \sin(\theta) - qw \quad \text{axial velocity} \\
    w' &= \frac{Z}{m} + g \cos(\theta) + qu \quad \text{vertical velocity} \\
    x' &= \cos(\theta)u + \sin(\theta)w \quad \text{range} \\
    z' &= -\sin(\theta)u + \cos(\theta)w \quad \text{altitude} \\
    \theta' &= q \quad \text{pitch angle} \\
    q' &= \frac{M}{I_{yy}} \quad \text{pitch rate}
\end{align*}
\]

\(X\) : thrust along \(u\) \quad \(Z\) : thrust along \(w\) \quad \(M\) : thrust moment for \(w\) \\
\(g\) : gravity \quad \(m\) : mass \quad \(I_{yy}\) : inertia second diagonal
Result (DRI Automatically Generates Invariant Functions)

\[
\frac{M_z}{l_{yy}} + g\theta + \left(\frac{X}{m} - qw\right)\cos(\theta) + \left(\frac{Z}{m} + qu\right)\sin(\theta)
\]

\[
\frac{M_x}{l_{yy}} - \left(\frac{Z}{m} + qu\right)\cos(\theta) + \left(\frac{X}{m} - qw\right)\sin(\theta)
\]

\[-q^2 + \frac{2M\theta}{l_{yy}}\]
Case Study: Dubins Dynamics of 2 Airplanes

Result (DRI Automatically Generates Invariants)

\[ \omega_1 = 0 \land \omega_2 = 0 \rightarrow v_2 \sin \vartheta x = (v_2 \cos \vartheta - v_1)y > p(v_1 + v_2) \]

\[ \omega_1 \neq 0 \lor \omega_2 \neq 0 \rightarrow -\omega_1 \omega_2 (x^2 + y^2) + 2v_2 \omega_1 \sin \vartheta x + 2(v_1 \omega_2 - v_2 \omega_1 \cos \vartheta)y \]

\[ + 2v_1 v_2 \cos \vartheta > 2v_1 v_2 + 2p(v_2|\omega_1| + v_1|\omega_2|) + p^2|\omega_1 \omega_2| \]