Dynamic Logic for Dynamical Systems

André Platzer

Carnegie Mellon University

Summer School Marktoberdorf 2017
Outline

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 Axioms
   - Example: Differential Ghosts

5. Applications

6. Summary
CPS are Multi-Dynamical Systems

Hybrid Systems / Games / Stochastic / Distributed Hybrid Systems

Differential Dynamic Logic

- Syntax
- Semantics
- Example: Car Control Design

Dynamic Axioms for Dynamical Systems

- Axiomatics
- Example: Safe Car Control
- Soundness and Completeness

Differential Invariants for Differential Equations

- Differential Axioms
- Example: Differential Ghosts

Applications

Summary
Cyber-Physical Systems Analysis: Aircraft Example

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?

André Platzer (CMU)
Dynamic Logic for Dynamical Systems
MOD’17
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
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
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 Axioms
   - Example: Differential Ghosts

5. Applications

6. Summary

André Platzer (CMU)
Dynamic Logics for Dynamical Systems

- **dL** = DL + HP
- **dGL** = GL + HG
- **SdL** = DL + SHP
- **QdL** = FOL + DL + QHP

[JAR'08, CADE'11, LMCS'12, LICS'12, LICS'12, TOCL'15, CADE'15, JAR'17, TOCL'17]

André Platzer (CMU)
**Logical Trinity**

**Axiomatics**

**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 trafo.
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?

André Platzer (CMU)
CPS Analysis

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

\[ [\alpha] \varphi \mapsto \varphi \]

\[ \alpha \]

\[ \varphi \]

\[ \alpha \]

\[ \varphi \]

\[ [\alpha] \varphi \]

\[ \alpha \]

\[ \varphi \]

\[ [\alpha] \varphi \]

\[ \alpha \]

\[ \varphi \]

\[ (JAR’08, LICS’12) \]
CPS Analysis

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

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

\[ x \neq m \]

\[ \begin{align*}
  x &
  \begin{cases}
    1, & 0 \leq t < 2.5 \\
    2, & 2.5 \leq t < 3.5 \\
    3, & 3.5 \leq t < 4.5 \\
    4, & 4.5 \leq t < 5.5 \\
    5, & 5.5 \leq t < 6.5 \\
    6, & 6.5 \leq t < 7.5 \\
    7, & t \geq 7.5
  \end{cases}
\end{align*} \]

\[ \begin{align*}
  v &
  \begin{cases}
    1, & 0 \leq t < 2 \\
    2, & 2 \leq t < 4 \\
    3, & 4 \leq t < 6 \\
    4, & 6 \leq t < 8 \\
    5, & 8 \leq t < 10 \\
    6, & t \geq 10
  \end{cases}
\end{align*} \]

\[ \begin{align*}
  m &
  \begin{cases}
    1, & 0 \leq t < 2 \\
    2, & 2 \leq t < 4 \\
    3, & 4 \leq t < 6 \\
    4, & 6 \leq t < 8 \\
    5, & 8 \leq t < 10 \\
    6, & t \geq 10
  \end{cases}
\end{align*} \]
Concept (Differential Dynamic Logic) (JAR’08,LICS’12)

\[ \alpha \varphi \Rightarrow \varphi \]

\[ \square x \neq m \Rightarrow x \neq m \]

André Platzer (CMU)
Dynamic Logic for Dynamical Systems
MOD’17 9 / 59
Concept (Differential Dynamic Logic) (JAR’08, LICS’12)

\[ [\alpha] \varphi \implies \varphi \]

\[ x \neq m \]

\[ [\varphi] x \neq m \]

\[ x \neq m \]

\[ x \neq m \]
CPS Analysis

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

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

\[ [x \neq m] \]

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

ODE

André Platzer (CMU)

Dynamic Logic for Dynamical Systems

MOD'17 9 / 59
CPS Analysis

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

\[ [\alpha] \varphi \]

\[ \varphi \alpha \]

\[ x \neq m \]

\[ [\varphi] x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ a := -b \]

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

\[ a \]

\[ t \]

\[ 0 \]

\[ 2 \]

\[ 4 \]

\[ 6 \]

\[ 8 \]

\[ 10 \]

\[ m \]

\[ t \]

\[ 0 \]

\[ 2 \]

\[ 4 \]

\[ 6 \]

\[ 8 \]

\[ 10 \]

\[ t \]

\[ 0 \]

\[ 2 \]

\[ 4 \]

\[ 6 \]

\[ 8 \]

\[ 10 \]
CPS Analysis

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

\([\alpha] \varphi \rightarrow \varphi\)

\([x \neq m] x \neq m\)

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

André Platzer (CMU)

Dynamic Logic for Dynamical Systems

MOD’17 9 / 59
(JAR’08, LICS’12)

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

seq. compose

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

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

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

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

seq. compose  

nondet. repeat

André Platzer (CMU)

Dynamic Logic for Dynamical Systems
CPS Analysis

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

\([\alpha]\phi \rightarrow \phi_{\alpha}\)

\([x \neq m]\)

\([\alpha]\phi \rightarrow \phi_{\alpha}\)

\([x \neq m]x \neq m\)

\([x \neq m]x \neq m\)

\([x \neq m]x \neq m\)

\([x \neq m]x \neq m\)

\([x \neq m]x \neq m\)

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

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

$\alpha \varphi \rightarrow \varphi$

$x \neq m \wedge b > 0 \rightarrow \left[ \left( \begin{array}{c} \text{if}(\text{SB}(x, m)) \rightarrow a := -b \right) \; ; \; x' = v, v' = a \right]^{*} \xrightarrow{} x \neq m$

init

all runs

post

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

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

\[ [\alpha] \varphi \iff \varphi \]

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

non-det. choice
CPS Analysis

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

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

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

Test

Non-deterministic choice

\[ \alpha \]

\[ \varphi \]

\[ \neg \]

\[ SB \]

\[ x \]

\[ m \]

\[ b \]

\[ > 0 \]

\[ \land \]

\[ \rightarrow \]

\[ [ \]

\[ ) \]

\[ \ast \]

\[ \rightarrow \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ \rightarrow \]

\[ \rightarrow \]

\[ post \]

\[ init \]

\[ a \]

\[ v \]

\[ m \]

\[ t \]

\[ a \]

\[ v \]

\[ m \]

\[ t \]
CPS Analysis

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

\[ [\alpha] \varphi \]

\[ \alpha \]

\[ \varphi \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

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

\[ x \neq m \]

\[ x \neq m \]

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

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]
Definition (Hybrid program $\alpha$)

\[
\begin{align*}
    x & := f(x) \mid ?Q \mid x' = f(x) \land Q \mid \alpha \cup \beta \mid \alpha; \beta \mid \alpha^*
\end{align*}
\]

Definition (dL Formula $P$)

\[
\begin{align*}
    e \geq \bar{e} \mid \neg P \mid P \land Q \mid \forall x P \mid \exists x P \mid [\alpha]P \mid \langle \alpha \rangle P
\end{align*}
\]
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 $P$)**

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


All Reals  Some Reals  All Runs  Some Runs

JAR’08, LICS’12, JAR’17
### Differential Dynamic Logic dL: Semantics

**Definition (Hybrid program semantics)**

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

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

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

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

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

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

**Definition (dL semantics)**

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

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

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

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

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

\[ [\exists x P] = \{ \omega : \omega_x^r \in [P] \text{ for some } r \in \mathbb{R} \} \]
Differential Dynamic Logic $dL$: Semantics

\[ x := e \]

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

\[ ?Q \]

If $\omega \in \llbracket Q \rrbracket$

Otherwise no transition

\[ \omega \text{ no change if } \omega \in \llbracket Q \rrbracket \]

\[ \omega \text{ otherwise no transition} \]
Differential Dynamic Logic $\mathsf{dL}$: Semantics

\[ \omega \xrightarrow{x := e} \nu \]

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

\[ ?Q \xrightarrow{\text{if } \omega \in \llbracket Q \rrbracket} \omega \]

\[ ?Q \xrightarrow{\text{if } \omega \not\in \llbracket Q \rrbracket} \omega \]

\[ \nu \text{ if } \nu(x) = \omega[\llbracket e \rrbracket] \text{ and } \nu(z) = \omega(z) \text{ for } z \neq x \]

\[ \omega \text{ no change if } \omega \in \llbracket Q \rrbracket \]

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

\[ x := e \]

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

\[ \omega \text{ no change if } \omega \in \llbracket Q \rrbracket \]

\[ \omega \text{ if } \omega \not\in \llbracket Q \rrbracket \]

\[ \omega \text{ if } \nu(z) = \omega(z) \text{ for } z \neq x \]

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

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

\[ \omega \text{ no change if } \omega \in \llbracket Q \rrbracket \]

\[ \omega \text{ if } \omega \not\in \llbracket Q \rrbracket \]

\[ \omega \text{ no change if } \omega \in \llbracket Q \rrbracket \]

\[ \omega \text{ otherwise no transition} \]
Differential Dynamic Logic dL: Semantics

\[ x := e \]

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

\[ ?Q \]

\[ \omega \]

\[ \omega \]

\[ \omega \]

\[ \omega \]

\[ \text{André Platzer (CMU)} \]

Dynamic Logic for Dynamical Systems

MOD’17 12 / 59
Differential Dynamic Logic dL: Semantics

\[
\begin{align*}
\omega &\xrightarrow{\alpha} \nu_1 \\
\omega &\xrightarrow{\beta} \nu_2 \\
\alpha &; \beta
\end{align*}
\]

\[
\begin{align*}
\omega &\xrightarrow{\alpha} \omega_1 \\
\omega_1 &\xrightarrow{\alpha} \omega_2 \\
\omega_2 &\xrightarrow{\alpha} \nu \\
\omega &\xrightarrow{\alpha^*} \nu
\end{align*}
\]
Differential Dynamic Logic $dL$: Semantics

\[
\begin{align*}
&\omega \\
&\quad \xleftarrow{\alpha \cup \beta} \nu_1 \\
&\quad \xrightarrow{\beta} \nu_2 \\
&\quad \xleftarrow{\alpha ; \beta} \\
&\omega \xleftarrow{\alpha} \mu \xrightarrow{\beta} \nu \\
&\quad \xrightarrow{\omega_1} \xrightarrow{\omega_2} \nu \\
&(\alpha ; \beta)^* \\
&\omega \xleftarrow{\alpha} \omega_1 \xrightarrow{\alpha} \omega_2 \xrightarrow{\alpha} \nu \\
\end{align*}
\]
Definition (dL Formulas)

\[ [\alpha]P \]

Compositional semantics \(\Rightarrow\) compositional proofs!
Definition (dL Formulas)

\[ \omega \langle \alpha \rangle P \]

compositional semantics \(\Rightarrow\) compositional proofs!

André Platzer (CMU)
Definition (dL Formulas)

\[ [\alpha]P \]

\( \omega \)

\( \alpha\)-span

André Platzer (CMU)
Definition (dL Formulas)

\[ [\alpha]P \]

\[ \langle \beta \rangle P \]

\( \omega \)

\( \alpha\)-span

\( \beta\)-span

\( [\alpha]P \)

Compositional semantics \( \Rightarrow \) Compositional proofs!
Definition (dL Formulas)

$\langle \beta \rangle P$

$\langle \beta \rangle P$

$[\alpha]P$

$\beta$-span

$\alpha$-span

$\omega$
compositional semantics $\Rightarrow$ compositional proofs!
Repeat control decisions

Example (Single car \(s\))

\[
((a := A \cup a := -b); \{x' = v, v' = a}\)^* 
\]
How does this model brake?

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

$$((a := A \cup a := -b); \{x' = v, v' = a\})^*$$
Velocity bound $v \geq 0$ in evolution domain

Example (Single car $car_s$)

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

Acceleration not always safe

Example (Single car $car_s$)

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

Acceleration condition \(?Q\)

Example (Single car \(car_s\))

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

\[ Q \equiv \]

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

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

Example (\text{ Safely stays before traffic light } m)

\[ A \geq 0 \land b > 0 \rightarrow [\text{car}_\varepsilon] x \leq m \]
\[ Q \equiv 2b(m - x) \geq v^2 + (A + b)(A + 2v) \]

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

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

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

\[ v^2 \leq 2b(m - x) \land A \geq 0 \land b > 0 \rightarrow [\text{car}_\varepsilon] x \leq m \]
Ex: Car Control Properties

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

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

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

Example (Safe stays before traffic light \( m \))

\[ v^2 \leq 2b(m - x) \land A \geq 0 \land b > 0 \rightarrow [\text{car}_\varepsilon] x \leq m \]
Ex: Car Control Properties

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

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

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

Example (Live, can move everywhere)

\[ \varepsilon > 0 \& A > 0 \& b > 0 \rightarrow \forall p \exists m \langle \text{car}_\varepsilon \rangle x \geq p \]
car ≡ (ctrl ; drive)∗

ctrl ≡ (?Q_A; a:= A)
∪ (?Q_b; a:= −b)

drive ≡ t := 0; \{x' = v, v' = a, t' = 1 & v ≥ 0 ∧ t ≤ \varepsilon\}
car \equiv (\text{ctrl} ; \text{drive})^*

\text{ctrl} \equiv (\mathcal{Q}_A ; a \leftarrow A)
\cup (\mathcal{Q}_b ; a \leftarrow -b)

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

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

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

\[ \text{drive} \equiv t := 0 ; \{ x' = v, v' = a, t' = 1 \land v \geq 0 \land t \leq \epsilon \} \]
car ≡ (ctrl ; drive)∗

ctrl ≡ (?QA; a := A) 
          U (?Qb; a := −b)

drive ≡ t := 0; \{x' = v, v' = a, t' = 1 \& v ≥ 0 \& t ≤ ε\}
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\}
car ≡ (ctrl ; drive)*

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

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

car ≡ (ctrl ; drive)*

ctrl ≡ (?Q_A; a := A)
    ∪ (?Q_b; a := −b)

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

car \equiv (ctrl ; drive)^*

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 \wedge t \leq \varepsilon \}
car $\equiv (ctrl \; ; \; drive)^*$

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

drive $\equiv t:=0; \{x'=v, \; v'=a, \; t'=1 \& \; v\geq0 \land t\leq\varepsilon\}$
Example (Bouncing Ball)
Example (Bouncing Ball)

\( \{ x' = v, v' = -g \} \)
Example (Bouncing Ball)

\[
\begin{align*}
x' &= v, \\
v' &= -g & x \geq 0
\end{align*}
\]
Ex: The Acrophobic Bouncing Ball

Example (Bouncing Ball)

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

\[ \{ x' = v, \ v' = -g & x \geq 0 \}; \]

\[ \text{if}(x = 0) \ v := -cv) \]

André Platzer (CMU)
Example (Bouncing Ball)

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

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

\[ Q(x, y) \equiv (x + w - o x)^2 + (y - v - o y)^2 \neq v^2 + w^2 \]

\[ Q_0 \equiv (o x - x)w \neq (o y - y)v \]

Obstacle not on tangential circle
Obstacle not on ray \((x, y) + R(v, w)\)

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

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

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

\[(x, y) \neq o \rightarrow \left[ \left( \left( Q_{-1} ; \omega := -1 \cup Q_{1} ; \omega := 1 \cup Q_{0} ; \omega := 0 \right) ; \right. \right. \left. \left. \{ x' = v, y' = w, v' = \omega w, w' = -\omega v \} \right) \right] \ (x, y) \neq o \]
Example (dL-based model-predictive control design)

\[ \forall \nu \geq 0 \land A \geq 0 \land b > 0 \rightarrow \]

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

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

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

\[
[x' = v, v' = -b] x \leq m \land v \geq 0 \land A \geq 0 \land b > 0 \to \\
[(\begin{array}{c}
\text{?} \\
\text{?}
\end{array}) \\
\begin{array}{c}
\cup a := A) \\
\cup a := -b)
\end{array} \\
t := 0; \{x' = v, v' = a, t' = 1 \land v \geq 0 \land t \leq \varepsilon\})^*] x \leq m
\]
Ex: Car Control Design

Logical MPC

Example (dL-based model-predictive control design)

\[
[x' = v, \dot{v}' = -b] \quad x \leq m \land v \geq 0 \land A \geq 0 \land b > 0 \rightarrow \\
[( ((
\begin{array}{l}
t := 0; \{ x' = v, \dot{v}' = a, t' = 1 \land v \geq 0 \land t \leq \varepsilon \}^* 
\end{array})^* \]

André Platzer (CMU)
Example (\(\text{dL-based model-predictive control design}\))

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

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

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

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

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

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

\[
v^2 \leq 2b(m-x) \land v \geq 0 \land A \geq 0 \land b > 0 \rightarrow \\
[([([t:=0; x' = v, v' = A, t' = 1 \& v \geq 0 \land t \leq \varepsilon][x' = v, v' = -b]x \leq m) ; \\
    a := A) \\
\cup a := -b); \\
    t := 0; \{x' = v, v' = a, t' = 1 \& v \geq 0 \land t \leq \varepsilon\}^*)] x \leq m
\]
Example (dL-based model-predictive control design)

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

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

\[ a := A ;
\]

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

\[ t := 0; \{ x' = v, v' = a, t' = 1 \land v \geq 0 \land t \leq \varepsilon \}^* \] x \leq m

\( a \)

\( x \)

\( v \)

\( t \)

\( m \)
Example (dL-based model-predictive control design)

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

\[ \left( (2b(m - x) \geq v^2 + (A + b)(A\varepsilon^2 + 2\varepsilon v)) ; \right] \\
\]

\[ a := A \right \}
\]

∪ \[ a := -b \right) ; \]

\[ t := 0; \{ x' = v, v' = a, t' = 1 \land v \geq 0 \land t \leq \varepsilon \}^* \right) x \leq m \]
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 Axioms
   - Example: Differential Ghosts

5. Applications

6. Summary
Differential Dynamic Logic: Axiomatization

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

\[\text{[?] } [?Q]P \leftrightarrow (Q \rightarrow P)\]

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

\[\text{[∪]} \ [\alpha \cup \beta]P \leftrightarrow [\alpha]P \land [\beta]P\]

\[\text{[;]} \ [\alpha; \beta]P \leftrightarrow [\alpha][\beta]P\]

\[\text{[*]} \ [\alpha^*]P \leftrightarrow P \land [\alpha][\alpha^*]P\]

\[\text{K } [\alpha](P \rightarrow Q) \rightarrow ([\alpha]P \rightarrow [\alpha]Q)\]

\[\text{I } [\alpha^*]P \leftrightarrow P \land [\alpha^*](P \rightarrow [\alpha]P)\]

\[\text{C } [\alpha^*] \forall v > 0 (P(v) \rightarrow \langle \alpha \rangle P(v-1)) \rightarrow \forall v (P(v) \rightarrow \langle \alpha^* \rangle \exists v \leq 0 P(v))\]
\[ [\mathrel{:=}] \ [x := e] \rho(x) \iff \]

\[ \omega \quad \xrightarrow{x := e} \quad v \quad \rho(x) \]
$[\equiv] \ [x := e] \rho(x) \leftrightarrow \rho(e)$
Dynamic Axioms for Dynamical Systems

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

\[ \text{['] } [x' = f(x)] \rho(x) \leftrightarrow \]

\[ x := e \quad \rightarrow \quad v \]

\[ p(e) \quad \rightarrow \quad p(x) \]

\[ x' = f(x) \quad \rightarrow \quad v \]

\[ p(x) \]
Dynamic Axioms for Dynamical Systems

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

\[ \text{[\textprime]} \ [x' = f(x)] p(x) \leftrightarrow [x := y(t)] p(x) \]
[$\cdot \cdot \cdot]$ $[x := e] p(x) \leftrightarrow p(e)$

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

\[ x := e \]

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

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

[\[\prime\]] \( [x' = f(x) \& q(x)] \rho(x) \leftrightarrow \forall t \geq 0 ([x := y(t)] \rho(x)) \)
Dynamic Axioms for Dynamical Systems

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

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

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

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

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

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

\[ \text{[?] [} ?Q \text{]} P \leftrightarrow \]

\[ \text{if } \omega \in [Q] \]
Dynamic Axioms for Dynamical Systems

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

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

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

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

\text{if } \omega \in \{Q\}
compositional semantics $\Rightarrow$ compositional proofs
\([\bigcup] [\alpha \cup \beta] P \leftrightarrow \]
\[ \bigcup \ [\alpha \cup \beta]P \leftrightarrow [\alpha]P \land [\beta]P \]
\[ \bigcup \left[ \alpha \cup \beta \right] P \leftrightarrow \left[ \alpha \right] P \land \left[ \beta \right] P \]

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

\[ \omega \rightarrow \alpha \rightarrow \omega \]
\[ \omega \rightarrow \beta \rightarrow \omega \]
\[ \beta \rightarrow \omega \]
\[ \alpha \rightarrow \mu \]
\[ \beta \rightarrow \nu \]
\[ \alpha \cup \beta \rightarrow \omega \]
\[ \alpha \cup \beta \rightarrow \nu \]
\[ \alpha ; \beta \rightarrow \omega \]
\[ \alpha ; \beta \rightarrow \nu \]
$$\bigcup [\alpha \cup \beta]P \leftrightarrow [\alpha]P \land [\beta]P$$

$$\mathbf{[;]} [\alpha;\beta]P \leftrightarrow [\alpha][\beta]P$$
\[
[\cup] \ [\alpha \cup \beta] P \leftrightarrow [\alpha] P \land [\beta] P
\]

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

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

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

\[\ast\] \([\alpha^\ast]P \leftrightarrow P \land [\alpha][\alpha^\ast]P\]
[∀] [α ∪ β]P ↔ [α]P ∧ [β]P

[;] [α; β]P ↔ [α][β]P

[|] [α*]P ↔ P ∧
[\bigcup] \ [\alpha \cup \beta] P \leftrightarrow [\alpha] P \land [\beta] P

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

\vdash \ [\alpha^*] P \leftrightarrow P \land (P \rightarrow [\alpha] P)
Dynamic Axioms for Dynamical Systems

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

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

\[ \vdash [\alpha^*]P \leftrightarrow P \land [\alpha^*](P \to [\alpha]P) \]
Dynamic Axioms for Dynamical Systems

\[
\boxtimes \quad [\alpha \cup \beta]P \leftrightarrow [\alpha]P \land [\beta]P
\]

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

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

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

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

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

\[
| \quad [\alpha^*]P \leftrightarrow P \land [\alpha^*](P \rightarrow [\alpha]P)
\]
Dynamic Axioms for Dynamical Systems

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

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

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

\[
\lfloor \alpha; \beta \rfloor P \iff [\alpha][\beta] P
\]

\[
\lfloor [\alpha^*] P \iff P \land [\alpha^*](P \to [\alpha] P)
\]
Proof Rule: Loop Invariants

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

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

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

Lemma (Loop invariant rule is derived)

\[
\frac{\Gamma \vdash J, \Delta \quad J \vdash [\alpha]J \quad J \vdash P}{\Gamma \vdash [\alpha^*]P, \Delta}
\]
**Proof Rule: Loop Invariants**

\[
\begin{array}{c}
P \\
G \frac{[\alpha]P}{[\alpha^*]P} \\
\end{array}
\]

**Lemma (Loop invariant rule is derived)**

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

**Proof (Derived rule).**

\[
\begin{array}{c}
J \vdash [\alpha]J \\
G \frac{J \vdash J \land [\alpha^*](J \rightarrow [\alpha]J)}{J \vdash [\alpha^*]J} \\
\end{array}
\]

\[
\begin{array}{c}
J \vdash P \\
M[\cdot] \frac{[\alpha^*]J \vdash [\alpha^*]P}{[\alpha]P \rightarrow [\alpha]Q}
\end{array}
\]

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

\[
G \quad \frac{P}{[\alpha]P} \quad I \quad [\alpha^*]P \iff P \land [\alpha^*](P \rightarrow [\alpha]P) \quad M[\cdot] \quad \frac{P \rightarrow Q}{[\alpha]P \rightarrow [\alpha]Q}
\]

Lemma (Loop invariant rule is derived)

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

Proof (Derived rule).

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

Finding invariant \( J \) can be a challenge.
Misplaced \([\alpha^*]\) 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

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

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

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 (at the bottom) to premises with remaining subgoals (top) until no more subgoals (\( \ast \))

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

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

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

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 \textit{conclusion} (at the bottom) to \textit{premises} with remaining subgoals (top) until no more subgoals (*)
Example Proof: Safe Braking

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

\[
\begin{align*}
\Gamma & \vdash \Delta \quad \text{shape of conjecture to prove} \\
\Gamma & \text{is list of all available assumptions} \\
\Delta & \text{disjunction needs to be proved from assumptions } \Gamma \\
\text{Proof reduces desired } & \text{conclusion (at the bottom)} \\
\text{to } & \text{premises with remaining subgoals (top) until no more subgoals (\( \ast \))}
\end{align*}
\]
Example Proof: Safe Braking

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

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

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 \textbf{conclusion} (at the bottom) to \textbf{premises} with remaining subgoals (top) until no more subgoals (*)

\[ \text{sequent} \]
\[ \text{antecedent} \]
\[ \text{succeedent} \]
Example Proof: Safe Braking

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

\[ J(x, v) \vdash \forall t \geq 0 \left( -\frac{b}{2} t^2 + vt + x \leq m \right) \]

\[ \text{QE} \]

\[ J(x, v) \vdash \forall t \geq 0 \left[ x := -\frac{b}{2} t^2 + vt + x \right] J(x, v) \]

\[ [:=] \]

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

\[ ['] \]

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

\[ [:=] \]

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

\[ [] \]

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

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 (at the bottom) to premises with remaining subgoals (top) until no more subgoals (\( \ast \))

\( \Delta \text{ shape of conjecture to prove} \)
(\( \Gamma \) antecedent
\( \Delta \) succedent)

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

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

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

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 (\(*\))

\text{sequent} \quad \text{antecedent} \quad \text{succedent}
Example Proof: Safe Braking

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

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

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

\[
\vdash \neg SB \rightarrow (A \epsilon + v) \leq 2b(m - A^2 \epsilon^2 - v \epsilon - x)
\]

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

\[
[:=] J(x, v) \vdash \neg SB \rightarrow \forall t \geq 0 (t \leq \epsilon \rightarrow J(x', v', a', t', 1 \& t \leq \epsilon))
\]
Example Proof: Safe Acceleration

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

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

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

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

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

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

\[ J(x, v) \vdash [\neg SB]J(x, v) \]

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

André Platzer (CMU)
Dynamic Logic for Dynamical Systems
MOD'17 26 / 59
Example Proof: Safe Acceleration

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

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

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

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

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

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

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

André Platzer (CMU)
Dynamic Logic for Dynamical Systems

\[
\begin{align*}
\therefore & \quad J(x, v) \vdash \neg SB \rightarrow \forall t \geq 0 \ (t \leq \varepsilon \rightarrow [x := \frac{A}{2} t^2 + vt + x] J(x, v)) \\
\,' & \quad J(x, v) \vdash \neg SB \rightarrow [x' = v, v' = A, t' = 1 \& t \leq \varepsilon] J(x, v) \\
\therefore & \quad J(x, v) \vdash \neg SB \rightarrow [a := A][x' = v, v' = a, t' = 1 \& t \leq \varepsilon] J(x, v) \\
\therefore & \quad J(x, v) \vdash [\neg SB][a := A; (x' = v, v' = a, t' = 1 \& t \leq \varepsilon)] J(x, v) \\
\therefore & \quad J(x, v) \vdash [\neg SB; a := A; (x' = v, v' = a, t' = 1 \& t \leq \varepsilon)] J(x, v)
\end{align*}
\]
\[ J(x, v) \equiv v^2 \leq 2b(m - x) \]

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

\[
\begin{align*}
J(x, v) & \vdash [a := -b][x'' = a\.\phantom{a}]. J(x, v) \land \lnot SB; a := A][x'' = a\.\phantom{a}]. J(x, v) \\
\cup & \vdash [a := -b \cup \lnot SB; a := A][x'' = a, t' = 1 & t \leq \varepsilon] J(x, v) \]
\]

\[
\begin{align*}
\text{loop} & \vdash [(a := -b \cup \lnot SB; a := A); x'' = a, t' = 1 & t \leq \varepsilon) \ast] J(x, v) \\
\end{align*}
\]
\( J(x, v) \equiv v^2 \leq 2b(m - x) \)

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

---

**Example Proof: Safe Driving**

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

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

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

\[ J(x, v) \vdash [(a := -b \cup \neg SB; a := A); x'' = a, t' = 1 \land t \leq \varepsilon]^* J(x, v) \]
\[ J(x, v) \equiv v^2 \leq 2b(m - x) \]
\[ SB \equiv 2b(m - x) < v^2 + (A + b)(A\varepsilon^2 + 2\varepsilon v) \]

---

**Example Proof: Safe Driving**

**Proof is deterministic “follow your nose”**.

**Synthesize invariant** \( J(x, v) \) and parameter constraint \( SB \).

**\( J(x, v) \) is a predicate symbol to prove only once and instantiate later.**

**First looking at proofs of smaller pieces is often effective.**
Example (Bouncing Ball)

\[ x = H \geq 0 \land \ldots \rightarrow \left[ \{ x' = v, v' = -g \land x \geq 0 \} \right. \]

\[ \text{if}(x = 0) \ v := -cv \right]^* \ 0 \leq x \leq H \]
Ex: The Ball Discovered a Crack in the Fabric of Time

Example (Bouncing Ball if $g > 0 \land 1 \geq c \geq 0 \land v = 0$)

$$x = H \geq 0 \land \ldots \rightarrow \left[ \left\{ x' = v, v' = -g \land x \geq 0 \right\};
\begin{align*}
\text{if}(x = 0) & v := -cv
\end{align*}\right] 0 \leq x \leq H$$
Example (Runaround Robot)

\[(x, y) \neq o \rightarrow \left[ \left( (\neg Q_{-1}; \omega := -1 \cup \neg Q_1; \omega := 1 \cup \neg Q_0; \omega := 0); \right) \left\{ x' = v, y' = w, v' = \omega w, w' = -\omega v \right\} \right]^* \right] (x, y) \neq o \]
Example (Runaround Robot)

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

\[ \models P \text{ iff } \text{FODE} \vdash_{dL} P \]
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
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)

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
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) \textit{(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)

\(dL\) 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

Theorem (Equi-expressibility) (LICS’12)

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

Theorem (Relative Decidability) (LICS’12)

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

Model

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

KeYmaera X generates proofs

ModelPlex proof

Proof search

KeYmaera X

Trustworthy

Flexible

Customizable

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

\[
\text{US} \quad \frac{\phi}{\sigma(\phi)}
\]

provided \( \text{FV}(\sigma|_{\Sigma(\theta)}) \cap \text{BV}(\otimes(\cdot)) = \emptyset \) 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 \( \theta \)

\( (U\text{-admissible}) \)

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

Obstacle Avoidance + Ground Navigation

Airborne Collision Avoidance (ACAS X)

Train Control Brakes

Ship Cooling

André Platzer (CMU)

Dynamic Logic for Dynamical Systems
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 Axioms
   - Example: Differential Ghosts

5. **Applications**

6. **Summary**
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

Theorem (Equi-expressibility) (LICS’12)

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

Theorem (Relative Decidability) (LICS’12)

Validity of \( \text{dL} \) 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.

\[
\begin{align*}
x'' &= -x \\
\text{has } x(t) &= \sin(t) = t - \frac{t^3}{3!} + \frac{t^5}{5!} - \frac{t^7}{7!} + \frac{t^9}{9!} - \ldots \\
x''(t) &= e^{t^2} \\
\text{has no elementary closed-form solution}
\end{align*}
\]
Intuition for Differential Invariants

Differential Invariant

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

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

Differential Invariant

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

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

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

Differential Invariant

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

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

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

Differential Invariant

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

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

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

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

\[ \neg F \]

Next step is undefined for ODEs. But 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

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

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

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

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

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

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

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

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

\[ \vdash [v' := w] [w' := -v] \]

\[ 2 vv' + 2ww' - 2rr' = 0 \]

\[ v^2 + w^2 - r^2 = 0 \]

\[ \vdash [v' = w, w' = -v] \]

\[ v^2 + w^2 - r^2 = 0 \]

\[ v^2 + w^2 - r^2 = 0 \]

\[ \vdash [v' = w, w' = -v] \]

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

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

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

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

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

\[
\begin{array}{c}
\text{\(\mathbb{R}\)}
\hline
\vdash 2v(w) + 2w(-v) = 0 \\
\vdash [v' := w][w' := -v] 2vv' + 2ww' - 2rr' = 0 \\
\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] v^2 + w^2 - r^2 = 0
\end{array}
\]
Differential Invariants for Differential Equations

Differential Invariant

Differential Cut

Differential Ghost

Logic

Provability theory

Math

Characteristic PDE

JLogComput’10, FMSD’09, LMCS’12, LICS’12, ITP’12, JAR’17, LICS’18
Differential Invariants for Differential Equations

Differential Invariant

Differential Cut

Differential Ghost

Logic

Provability theory

Math

Characteristic PDE

JLogComput’10,FMSD’09,LMCS’12,LICS’12,ITP’12,JAR’17,LICS’18
Differential Invariants for Differential Equations

Differential Invariant

Differential Cut

Differential Ghost

Logic
Provability theory

Math
Characteristic PDE

JLogComput’10, FMSD’09, LMCS’12, LICS’12, ITP’12, JAR’17, LICS’18
Differential Invariants for Differential Equations

Differential Invariant

Differential Cut

Differential Ghost

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

\( \text{inv} \)

\( \text{DI} \geq \text{DI} = \land \lor \)

\( \text{DI} > \text{DI} > = \land \lor \)

Logic

Provability theory

Math

Characteristic PDE

JLogComput’10, FMSD’09, LMCS’12, LICS’12, ITP’12, JAR’17, LICS’18

André Platzer (CMU)
Dynamic Logic for Dynamical Systems
MOD’17 38 / 59
Differential Invariants for Differential Equations

Differential Invariant

\[ x' = f(x) \]

Differential Cut

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

Differential Ghost

[Graph and equations related to differential invariants, cuts, and ghosts]

Logic

Provability theory

Math

Characteristic PDE

JLogComput’10, FMSD’09, LMCS’12, LICS’12, ITP’12, JAR’17, LICS’18

André Platzer (CMU)

Dynamic Logic for Dynamical Systems

MOD’17 38 / 59
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, FMSD’09, LMCS’12, LICS’12, ITP’12, JAR’17, LICS’18
Differential Invariants for Differential Equations

Differential Invariant

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

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

Differential Cut

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

Differential Ghost

Logic

Provability theory

Math

Characteristic PDE

JLogComput’10, FMSD’09, LMCS’12, LICS’12, ITP’12, JAR’17, LICS’18

André Platzer (CMU)

Dynamic Logic for Dynamical Systems

MOD’17 38 / 59
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, FMSD’09, LMCS’12, LICS’12, ITP’12, JAR’17, LICS’18

André Platzer (CMU) Dynamic Logic for Dynamical Systems
Differential Invariants for Differential Equations

**Differential Invariant**

**Differential Cut**

**Differential Ghost**

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

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

\[ \text{inv} \]

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

\[ x' = f(x) \]

**Logic**

- Provability theory

**Math**

- Characteristic PDE

\[ DI \geq, =, \&, \lor \]

\[ DI >, =, \&, \lor \]

\[ DI >, >, =, \&, \lor \]

\[ DI =, =, \&, \lor \]

\[ DI >, =, \&, \lor \]

JLogComput’10, FMSD’09, LMCS’12, LICS’12, ITP’12, JAR’17, LICS’18
Differential Invariant

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

Differential Cut

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

Differential Ghost

\[
P \iff \exists y G \quad G \vdash [x' = f(x), y' = g(x, y) \land Q]G \\
\frac{}{P \vdash [x' = f(x) \land Q]P}
\]

deductive power adds $\text{DI} \prec \text{DC} \prec \text{DG}$
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 long enough solution

JLogComput’10, LMCS’12, LICS’12, JAR’17, LICS’18
\[ \omega^2 x^2 + y^2 \leq c^2 \vdash [x' = y, y' = -\omega^2 x - 2d \omega y \land \omega \geq 0 \land d \geq 0] \omega^2 x^2 + y^2 \leq c^2 \]
\[ \omega^2 x^2 + y^2 \leq c^2 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y \land \omega \geq 0 \land d \geq 0] \omega^2 x^2 + y^2 \leq c^2 \]
\[ \omega \geq 0 \land d \geq 0 \vdash [x' := y][y' := -\omega^2 x - 2d\omega y] 2\omega^2 x x' + 2y y' \leq 0 \]

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

* 

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

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

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

increasingly damped oscillator
ω ≥ 0 ∧ d ≥ 0 ⊢ [x' := y] [y' := −ω^2 x − 2d ω y] 2ω^2 xx' + 2yy' ≤ 0

ω^2 x^2 + y^2 ≤ c^2 ⊢ [x' = y, y' = −ω^2 x − 2d ω y, d' = 7 & ω ≥ 0 ∧ d ≥ 0] ω^2 x^2 + y^2 ≤ c^2

ω^2 x^2 + y^2 ≤ c^2 ⊢ [x' = y, y' = −ω^2 x − 2d ω y, d' = 7 & ω ≥ 0] ω^2 x^2 + y^2 ≤ c^2

* 

ω ≥ 0 ⊢ 7 ≥ 0

ω ≥ 0 ⊢ [d' := 7] d' ≥ 0

d ≥ 0 ⊢ [x' = y, y' = −ω^2 x − 2d ω y, d' = 7 & ω ≥ 0] d' ≥ 0

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

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

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

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

* 

\( \omega \geq 0 \vdash 7 \geq 0 \)

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

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

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

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

increasingly damped oscillator
Differential Cuts for Differential Equations

\(*\)

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

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

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

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

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

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

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

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

\[ \omega[(e)'] = \]

\[ \omega[(e)'] = \sum x \omega(x') \partial [ [e] ] \partial x (\omega) \]

Axioms

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

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

\[ (c())' = 0 \text{ for constants/numbers} \]

\[ x' = x' \text{ for variables } x \in V \]

ODE

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

\[ \{ \phi(0), \phi(r) \} : \phi | = x' = f(x) \wedge Q \text{ for some } \phi : [0, r] \to S, \text{ some } r \in \mathbb{R} \]

\[ \phi(z)(x') = d\phi(t)(x) dt(z) \]

...
Differentials

Syntax

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

Semantics

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

**Syntax**

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

**Semantics**

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

no time!
Differentials

**Syntax**

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

**Semantics**

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

→ \( \mathbb{R} \)
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**

1. \((e + k)' = (e)' + (k)'\)
2. \((e \cdot k)' = (e)' \cdot k + e \cdot (k)'\)
3. \((c())' = 0\) for constants/numbers \(c()\)
4. \((x)' = x'\) for variables \(x \in \mathcal{V}\)
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**

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

**ODE**

\[
[x' = f(x) \land Q] = \{(\varphi(0), \varphi(r)) : \varphi \models x' = f(x) \land Q \}
\text{ for some } \varphi : [0, r] \to \mathcal{S}, \text{ some } r \in \mathbb{R}
\]

\[
\varphi(z)(x') = \frac{d \varphi(t)(x)}{dt}(z) \quad \ldots
\]
## 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 $\varphi \models P \leftrightarrow [x' := f(x)]P$

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

- $(e + k)' = (e)' + (k)'$
- $(e \cdot k)' = (e)' \cdot k + e \cdot (k)'$
- $(c())' = 0$
- $(x)' = x'$

for constants/numbers $c()$

for variables $x \in V$

André Platzer (CMU)
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 $0 \leq z \leq r$, $FV(e) \subseteq \{x\}$:

**Syntactic**

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

**Analytic**

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

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

**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 $x \in \mathcal{V}$
<table>
<thead>
<tr>
<th>Lemma (Differential lemma)</th>
<th>(Differential value vs. Time-derivative)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \phi \models x' = f(x) \land Q ) for duration ( r &gt; 0 ), then for all ( 0 \leq z \leq r ), ( FV(e) \subseteq {x} ):</td>
<td></td>
</tr>
<tr>
<td>( \phi(z)[(e)'] = \frac{d\phi(t)[e]}{dt}(z) )</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Lemma (Differential assignment)</th>
<th>(Effect on Differentials)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( DE [x' = f(x) \land Q]P \leftrightarrow [x' = f(x) \land Q][x' := f(x)]P )</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Lemma (Derivations)</th>
<th>(Equations of Differentials)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( +' ) ( (e + k)' = (e)' + (k)' )</td>
<td></td>
</tr>
<tr>
<td>( \cdot' ) ( (e \cdot k)' = (e)' \cdot k + e \cdot (k)' )</td>
<td></td>
</tr>
<tr>
<td>( c' ) ( (c())' = 0 )</td>
<td></td>
</tr>
<tr>
<td>( x' ) ( (x)' = x' )</td>
<td></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 $0 \leq z \leq r$, $FV(e) \subseteq \{x\}$:

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

Lemma (Differential assignment) (Effect on Differentials)

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

Axiomatics

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

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

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

\[
\text{DC} \quad ([x' = f(x) & Q]P \leftrightarrow [x' = f(x) & Q \land C]P) \leftarrow [x' = f(x) & Q]C
\]

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

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

\[
\text{DG} \quad [x' = f(x) & Q]P \leftrightarrow \exists y [x' = f(x), y' = a(x)y + b(x) & Q]P
\]

\[
\text{DS} \quad [x' = c() & Q]P \leftrightarrow \forall t \geq 0 ((\forall 0 \leq s \leq t q(x + c()s)) \rightarrow [x := x + c()t]P)
\]

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

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

\[
\circ [y := g(x)][y' := 1]((f(g(x)))' = (f(y))' \cdot (g(x))')
\]
Differential Equation Axioms

Axiom (Differential Weakening) (JAR’17)

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

\[ Q \]

Differential equations cannot leave their evolution domains. Implies:

\[ [x' = f(x) \& Q] P \leftrightarrow [x' = f(x) \& Q](Q \rightarrow P) \]
Differential Equation Axioms

Axiom (Differential Cut) (JAR’17)

\[ \text{DC} \left( [x' = f(x) \& Q] P \leftrightarrow [x' = f(x) \& Q \land C] P \right) \]
\[ \leftrightarrow [x' = f(x) \& Q] C \]

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.
Differential Equation Axioms

Axiom (Differential Invariant) (JAR’17)

\[ \text{DI} \left( [x' = f(x) \& Q] P \leftrightarrow [\?? Q] P \right) \leftarrow [x' = f(x) \& Q] (P)' \]

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

What’s the differential of a formula???
What’s the meaning of a differential term . . . in a state???
Axiom (Differential Effect) (JAR’17)

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

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
**Differential Equation Axioms**

**Axiom (Differential Ghost)** (JAR’17)

\[
\begin{align*}
\text{DG} \quad [x' = f(x) & Q] P \iff & \exists y [x' = f(x), y' = a(x)y + b(x) & Q] P
\end{align*}
\]

**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 ghost proof)

MR
\[ x > 0 \vdash \exists y \left[ x' = -x, y' = \cdots \right] x > 0 \]

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

Example (Differential ghost proof)

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

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

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

DG
\[ x > 0 \vdash [x' = -x] x > 0 \]
Example (Differential ghost proof)

\[
\begin{align*}
\mathbb{R} & \vdash xy^2 = 1 \implies x > 0 \\
\exists \mathbb{R}, \text{cut} & \vdash x > 0 \implies \exists y [x' = -x, y' = \ \ ] xy^2 = 1 \\
\text{MR} & \vdash x > 0 \implies \exists y [x' = -x, y' = \ ] x > 0 \\
\text{DG} & \vdash x > 0 \implies [x' = -x] x > 0
\end{align*}
\]
Example: Differential Ghosts

Example (Differential ghost proof)

\[ x > 0 \vdash [x' = -x] x' y^2 + x 2 y y' = 0 \]

\[ \begin{align*}
\forall \mathbb{R} \quad xy^2 = 1 & \vdash x > 0 \\
\exists \mathbb{R}, \text{cut} & \\
x > 0 \vdash \exists y [x' = -x, y' = \_\_ \_ \_ \_ \_] xy^2 = 1 \\
\exists \mathbb{R}, \text{cut} & \\
x > 0 \vdash \exists y [x' = -x, y' = \_\_ \_ \_ \_ \_] x > 0 \\
\exists \mathbb{R}, \text{cut} & \\
x > 0 \vdash [x' = -x] x > 0 \\
\end{align*} \]
Example: Differential Ghosts

Example (Differential ghost proof)

\[
\begin{align*}
\begin{array}{c}
\text{Example (Differential ghost proof)} \\
\text{Example (Differential ghost proof)}
\end{array}
\end{align*}
\]

\[
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\text{Example (Differential ghost proof)} \\
\text{Example (Differential ghost proof)}
\end{array}
\end{array}
\end{align*}
\]

\[
\begin{align*}
\mathbb{R} & \quad \vdash -xy^2 + 2xy = 0 \\
[=] & \quad \vdash [x':=-x][y':=\phantom{5}\phantom{5}]x'y^2 + x2yy' = 0 \\
* & \quad \vdash \phantom{5}\phantom{5}\phantom{5} \\
dl & \quad \vdash \phantom{5}\phantom{5}\phantom{5} \\
x^2y=1 & \quad \vdash [x' = -x, y' = \phantom{5}\phantom{5}]xy^2 = 1 \\
\exists R, c \text{ut} & \quad \exists y [x' = -x, y' = \phantom{5}\phantom{5}]xy^2 = 1 \\
x > 0 & \quad \exists y [x' = -x, y' = \phantom{5}\phantom{5}]x > 0 \\
\exists R, c \text{ut} & \quad \exists y [x' = -x, y' = \phantom{5}\phantom{5}]x > 0 \\
x > 0 & \quad \vdash [x' = -x]x > 0 \\
\end{align*}
\]

\[
\begin{align*}
\mathbb{R} & \quad \vdash -xy^2 + 2xy = 0 \\
[=] & \quad \vdash [x':=-x][y':=\phantom{5}\phantom{5}]x'y^2 + x2yy' = 0 \\
* & \quad \vdash \phantom{5}\phantom{5}\phantom{5} \\
dl & \quad \vdash \phantom{5}\phantom{5}\phantom{5} \\
x^2y=1 & \quad \vdash [x' = -x, y' = \phantom{5}\phantom{5}]xy^2 = 1 \\
\exists R, c \text{ut} & \quad \exists y [x' = -x, y' = \phantom{5}\phantom{5}]xy^2 = 1 \\
x > 0 & \quad \exists y [x' = -x, y' = \phantom{5}\phantom{5}]x > 0 \\
\exists R, c \text{ut} & \quad \exists y [x' = -x, y' = \phantom{5}\phantom{5}]x > 0 \\
x > 0 & \quad \vdash [x' = -x]x > 0 \\
\end{align*}
\]

\[
\begin{align*}
\mathbb{R} & \quad \vdash -xy^2 + 2xy = 0 \\
[=] & \quad \vdash [x':=-x][y':=\phantom{5}\phantom{5}]x'y^2 + x2yy' = 0 \\
* & \quad \vdash \phantom{5}\phantom{5}\phantom{5} \\
dl & \quad \vdash \phantom{5}\phantom{5}\phantom{5} \\
x^2y=1 & \quad \vdash [x' = -x, y' = \phantom{5}\phantom{5}]xy^2 = 1 \\
\exists R, c \text{ut} & \quad \exists y [x' = -x, y' = \phantom{5}\phantom{5}]xy^2 = 1 \\
x > 0 & \quad \exists y [x' = -x, y' = \phantom{5}\phantom{5}]x > 0 \\
\exists R, c \text{ut} & \quad \exists y [x' = -x, y' = \phantom{5}\phantom{5}]x > 0 \\
x > 0 & \quad \vdash [x' = -x]x > 0 \\
\end{align*}
\]
Example (Differential Ghost proof)

\[
\begin{align*}
\mathbb{R} & \quad \vdash -xy^2 + 2xy = 0 \\
[::] & \quad \vdash [x'=-x, y'=\text{blue}] x'y^2 + x2yy' = 0 \\
\star & \quad \vdash [x'=-x, y'=\text{blue}] x'y^2 = 1 \\
\text{dl} & \quad \vdash x > 0 \implies \exists y [x'=-x, y'=\text{blue}] xy^2 = 1 \\
\text{MR} & \quad \vdash x > 0 \implies \exists y [x'=-x, y'=\text{blue}] x > 0 \\
\text{DG} & \quad \vdash x > 0 \implies [x'=-x] x > 0
\end{align*}
\]
Example: Differential Ghosts

Example ( Differential ghost proof)

Example ( Differential ghost proof)

\[
\begin{align*}
\mathbb{R} & \vdash -xy^2 + 2xy\frac{y}{2} = 0 \\
[=] & \vdash [x' = -x][y' = \frac{y}{2}] x'y^2 + x2yy' = 0 \\
\ast & \vdash [x' = -x, y' = \frac{y}{2}] xy^2 = 1 \\
\exists \mathbb{R}, \text{cut} & \vdash \exists y [x' = -x, y' = \frac{y}{2}] xy^2 = 1 \\
\exists \mathbb{R}, \text{cut} & \vdash x > 0 \vdash \exists y [x' = -x, y' = \frac{y}{2}] x > 0 \\
\exists \mathbb{R}, \text{cut} & \vdash x > 0 \vdash [x' = -x] x > 0
\end{align*}
\]
Example: Differential Ghosts

Example (Diffential ghost proof)

\[ \mathbb{R} \vdash -xy^2 + 2xy \frac{y}{2} = 0 \]
\[ [:=] \quad \vdash [x' = -x, y' = \frac{y}{2}] x'y^2 + x2yy' = 0 \]
\[ \mathbb{R} \vdash xy^2 = 1 \quad \exists \mathbb{R}, \text{cut} \]
\[ \vdash x > 0 \quad \exists y [x' = -x, y' = \frac{y}{2}] xy^2 = 1 \]
\[ \text{MR} \quad x > 0 \vdash \exists y [x' = -x, y' = \frac{y}{2}] x > 0 \]
\[ \text{DG} \quad x > 0 \vdash [x' = -x] x > 0 \]

\[
\begin{align*}
\gamma' &= \frac{\gamma}{\alpha} \\
X' &= -X \\
X_0 &\rightarrow X \\
0 &\rightarrow t
\end{align*}
\]
Example (Differential ghost proof)

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

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

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

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

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

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

\[ x > 0 \vdash [x' = -x] x > 0 \]
Ex: Parachute Open or Keep Closed

Conservatively bounded next velocity above parachute's limit velocity.

Limit by differential ghost:

\[ y' = -p^2 (v - \sqrt{g/p}) y_2 (v + \sqrt{g/p}) \]

\[ v \geq v_{\text{old}}(t) - gt \quad \text{if closed} \]
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)

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

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

Conservatively bounded next velocity above parachute’s limit velocity.

Limit by differential ghost:

\[ y' = -\frac{p^2}{v - \sqrt{g/p}} \]

\[ v > 0 \]

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

Example (Parachute)

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

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

Conservatively bounded next velocity above parachute’s limit velocity.

Example (Parachute)

\[ m < -\sqrt{g/p} \rightarrow \left[ (\exists(Q \land r = a) \cup r := p); t := 0; \right. \]

\[ \left. \{ x' = v, v' = -g + rv^2, t' = 1 \land t \leq T \land x \geq 0 \land v < 0 \}^* \right] \]

\[ (x = 0 \rightarrow v \geq m) \]
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)

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

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

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

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

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

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

Example (Parachute)

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

1. Identified safe region for each advisory symbolically
2. Proved safety for hybrid systems flight model in KeYmaera X
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.
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 ($\approx 899 \times 10^6$ 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 (≈899 $10^6$ counterexamples).

Safe Version: Action Issued = CL1500
Followed by Most Extreme Up/Down-sense Available

ACAS X issues Maintain advisory instead of CL1500
Application Highlights

Obstacle Avoidance + Ground Navigation

Airborne Collision Avoidance (ACAS X)

Train Control Brakes

Ship Cooling

\[ k \cdot u = v_0 - v \]

\[ x' = f(x) \]

\[ y' = g(x, y) \]
Verified CPS Applications: Trains & Airplanes

ICFEM’09, JAIS’14, TACAS’15, EMSOFT’15, FM’09, HSCC’11, HSCC’13, TACAS’14, RSSRail’17

André Platzer (CMU)
Dynamic Logic for Dynamical Systems
MOD’17 53 / 59
Verified CPS Applications: Cars

FM’11, LMCS’12, ICCPS’12, ITSC’11, ITSC’13, IJCAR’12

Dynamic Logic for Dynamical Systems
Verified CPS Applications: lfcps.org/course/

undergrads in Foundations of Cyber-Physical Systems course

André Platzer (CMU)  Dynamic Logic for Dynamical Systems  MOD’17  53 / 59
KeYmaera X aXiomatic Tactical Theorem Prover for CPS

Proof search

KeYmaera X generates proofs

ModelPlex proof

Model Safety

Compliance Monitor

Model

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
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 Axioms
   - Example: Differential Ghosts

5. Applications

6. Summary
Acknowledgments

Logical Systems Lab at Carnegie Mellon University
Yong Kiam Tan, Brandon Bohrer, Nathan Fulton, Sarah Loos
Stefan Mitsch, Khalil Ghorbal, Jean-Baptiste Jeannin, Andrew Sogokon
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

differential dynamic logic
\[ dL = DL + HP \]

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

Numerous wonders remain to be discovered

KeYmaera X

KeYmaera X is a tool for formal verification of cyber-physical systems. It combines differential dynamic logic (dL) with hybrid programs (HP) to analyze complex systems involving both discrete and continuous dynamics.

KeYmaera X supports reasoning about a variety of system properties, including those involving adversarial, nondeterministic, and stochastic behaviors. It provides a framework for proving safety, liveness, and other properties of complex systems, facilitating the development of more reliable and secure technologies.
Future CPS Challenges

Numerous wonders remain to be discovered

- Scalable continuous stochastics  
  CADE’11
- Concurrent CPS  
- Real arithmetic: Scalable and verified  
  CADE’09
- Verified CPS implementations, ModelPlex  
  FMSD’16
- Correct CPS execution  
- CPS-conducive tactic languages+libraries  
  ITP’17
- Tactics exploiting CPS structure/linearity/. . .
- Invariant generation  
  FMSD’09 TACAS’14
- Tactics & proofs for reachable set computations
- Parallel proof search & disprovers
- Correct model transformation  
  FM’14
- Inspiring applications

CPSs deserve proofs as safety evidence!
Differential Invariant Soundness Proof

- Differential Radical Invariants
### 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 $0 \leq z \leq r$, $FV(e) \subseteq \{x\}$:

<table>
<thead>
<tr>
<th>Syntactic</th>
<th>Analytic</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varphi(z)[(e)'] = \frac{d\varphi(t)[e]}{dt}(z)$</td>
<td>$\varphi(z)[(e)'] = \frac{d\varphi(t)[e]}{dt}(z)$</td>
</tr>
</tbody>
</table>

#### 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)

- $(e + k)' = (e)' + (k)'$
- $(e \cdot k)' = (e)' \cdot k + e \cdot (k)'$
- $(c())' = 0$
- $(x)' = x'$

for constants/numbers $c()$ for variables $x \in \mathcal{V}$
**Lemma (Differential lemma)** (Differential value vs. Time-derivative)

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

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

**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 $\varphi : [0, r] \rightarrow \mathcal{S}$ 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 \( 0 \leq z \leq r \), \( FV(e) \subseteq \{x\} \):

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

**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 \( \varphi : [0, r] \rightarrow \mathcal{I} \) 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)

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

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

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

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

### Semantics

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

### Definition (Hybrid program semantics) $\llbracket\cdot\rrbracket : \text{HP} \to \wp(S \times S)$

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

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 $0 \leq z \leq r$, $FV(e) \subseteq \{x\}$:

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

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

Semantic

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

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 $\varphi : [0, r] \rightarrow \mathcal{I}$ 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**

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

**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 $\varphi : [0, r] \rightarrow \mathcal{S}$ where $\varphi(z)(x') \overset{\text{def}}{=} \frac{d\varphi(t)(x)}{dt}(z)\)$$
Theorem (Differential radical invariant characterization)

\[ h = 0 \rightarrow \bigwedge_{i=1}^{N-1} h^{(i)}_p = 0 \]

\[ h = 0 \rightarrow [x' = p] h = 0 \]

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

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

Corollary (Algebraic Invariants Decidable)

Algebraic invariants of algebraic differential equations are decidable.
Example: Longitudinal Dynamics of an Airplane

Study (6th Order Longitudinal Flight Equations)

\[ u' = \frac{X}{m} - g \sin(\theta) - qw \quad \text{axial velocity} \]

\[ w' = \frac{Z}{m} + g \cos(\theta) + qu \quad \text{vertical velocity} \]

\[ x' = \cos(\theta)u + \sin(\theta)w \quad \text{range} \]

\[ z' = -\sin(\theta)u + \cos(\theta)w \quad \text{altitude} \]

\[ \theta' = q \quad \text{pitch angle} \]

\[ q' = \frac{M}{I_{yy}} \quad \text{pitch rate} \]

\[ X : \text{thrust along } u \quad Z : \text{thrust along } w \quad M : \text{thrust moment for } w \]

\[ g : \text{gravity} \quad m : \text{mass} \quad I_{yy} : \text{inertia second diagonal} \]

with Khalil Ghorbal TACAS’14
Result (DRI Automatically Generates Invariant Functions)

\[
\frac{M_z}{I_{yy}} + g\theta + \left( \frac{X}{m} - qw \right) \cos(\theta) + \left( \frac{Z}{m} + qu \right) \sin(\theta) \\
\frac{M_x}{I_{yy}} - \left( \frac{Z}{m} + qu \right) \cos(\theta) + \left( \frac{X}{m} - qw \right) \sin(\theta) \\
- q^2 + \frac{2M\theta}{I_{yy}}
\]
André Platzer.
Logic & proofs for cyber-physical systems.
doi:10.1007/978-3-319-40229-1_3.

André Platzer.
Logics of dynamical systems.
In LICS [29], pages 13–24.

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

André Platzer.
Differential dynamic logic for hybrid systems.
André Platzer.
A complete uniform substitution calculus for differential dynamic logic.

André Platzer.
Differential game logic.

André Platzer.
Differential hybrid games.

André Platzer.
The complete proof theory of hybrid systems.
In LICS [29], pages 541–550.
doi:10.1109/LICS.2012.64.

André Platzer.
A complete axiomatization of quantified differential dynamic logic for distributed hybrid systems.

Special issue for selected papers from CSL’10.

**André Platzer.**

Stochastic differential dynamic logic for stochastic hybrid programs.
doi:10.1007/978-3-642-22438-6_34.

**André Platzer.**

A uniform substitution calculus for differential dynamic logic.
In Felty and Middeldorp [30], pages 467–481.
doi:10.1007/978-3-319-21401-6_32.

**Nathan Fulton, Stefan Mitsch, Jan-David Quesel, Marcus Völp, and André Platzer.**

KeYmaera X: An axiomatic tactical theorem prover for hybrid systems.
In Felty and Middeldorp [30], pages 527–538.
Stefan Mitsch and André Platzer.
ModelPlex: Verified runtime validation of verified cyber-physical system models.
Special issue of selected papers from RV’14.

Stefan Mitsch, Khalil Ghorbal, David Vogelbacher, and André Platzer.
Formal verification of obstacle avoidance and navigation of ground robots.

André Platzer and Jan-David Quesel.
European Train Control System: A case study in formal verification.
Stefan Mitsch, Marco Gario, Christof J. Budnik, Michael Golm, and André Platzer.
Formal verification of train control with air pressure brakes.
doi:10.1007/978-3-319-68499-4_12.

Jean-Baptiste Jeannin, Khalil Ghorbal, Yanni Kouskoulas, Aurora Schmidt, Ryan Gardner, Stefan Mitsch, and André Platzer.
A formally verified hybrid system for safe advisories in the next-generation airborne collision avoidance system.

André Platzer.
Differential-algebraic dynamic logic for differential-algebraic programs.

André Platzer and Edmund M. Clarke.
Computing differential invariants of hybrid systems as fixedpoints. 
Special issue for selected papers from CAV’08.

André Platzer.
The structure of differential invariants and differential cut elimination. 

André Platzer.
A differential operator approach to equational differential invariants. 
In Lennart Beringer and Amy Felty, editors, *ITP*, volume 7406 of *LNCS*, 
doi:10.1007/978-3-642-32347-8_3.

André Platzer and Yong Kiam Tan.
Differential equation axiomatization: The impressive power of differential ghosts. 


Khalil Ghorbal and André Platzer.
Characterizing algebraic invariants by differential radical invariants.  

Thomas A. Henzinger.  
The theory of hybrid automata.  
doi:10.1109/LICS.1996.561342.

Jennifer M. Davoren and Anil Nerode.  
Logics for hybrid systems.  
doi:10.1109/5.871305.


Amy Felty and Aart Middeldorp, editors.  
doi:10.1007/978-3-319-21401-6.