Hybrid Systems & Complete Proofs

André Platzer

aplatzer@cs.cmu.edu

Computer Science Department
Carnegie Mellon University, Pittsburgh, PA
Outline

1. Hybrid Systems
2. Differential Dynamic Logic
   - Syntax
   - Semantics
   - Axiomatization
3. Continuous Completeness
   - Schematic Completeness
   - Expressibility and Rendition of Hybrid Programs
4. Discrete Completeness
   - Open Discrete Completeness
   - Closed Discrete Completeness
   - Semialgebraic Discrete Completeness of $dL + \Delta$
   - Discrete Completeness of $dL + \Delta$
   - Equi-expressible
   - Relative Decidable
5. Summary
Outline

1 Hybrid Systems

2 Differential Dynamic Logic
   - Syntax
   - Semantics
   - Axiomatization

3 Continuous Completeness
   - Schematic Completeness
   - Expressibility and Rendition of Hybrid Programs

4 Discrete Completeness
   - Open Discrete Completeness
   - Closed Discrete Completeness
   - Semialgebraic Discrete Completeness of dL +Δ
   - Discrete Completeness of dL +Δ
   - Equi-expressible
   - Relative Decidable

5 Summary
Hybrid Systems: e.g., Car Control

Challenge (Hybrid Systems)

Fixed rule describing state evolution with both

- Continuous dynamics (differential equations)
- Discrete dynamics (control decisions)
Challenge (Hybrid Systems)

*Fixed rule* describing state evolution with both

- Continuous dynamics (differential equations)
- Discrete dynamics (control decisions)
Proof theory: continuous = hybrid = discrete
Outline

1. Hybrid Systems
2. Differential Dynamic Logic
   - Syntax
   - Semantics
   - Axiomatization
3. Continuous Completeness
   - Schematic Completeness
   - Expressibility and Rendition of Hybrid Programs
4. Discrete Completeness
   - Open Discrete Completeness
   - Closed Discrete Completeness
   - Semialgebraic Discrete Completeness of dL +Δ
   - Discrete Completeness of dL +Δ
   - Equi-expressible
   - Relative Decidable
5. Summary
Hybrid Systems Analysis

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

$$\alpha \phi$$

$$[\alpha] \phi \phi^\alpha$$

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

(all runs)

$$x \neq m$$
Differential Dynamic Logic dL

Definition (Hybrid program)

\[ \alpha, \beta ::= x := e \mid ?Q \mid x' = f(x) \land Q \mid \alpha \cup \beta \mid \alpha; \beta \mid \alpha^* \]

Definition (Differential dynamic logic)

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

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

\[ \alpha \quad \alpha^* \]

\[ \beta \quad \alpha \cup \beta \]

\[ \alpha; \beta \]

\[ \mu \quad \beta \]

\[ v \quad v_1 \quad v_2 \]

\[ \omega \quad \omega_1 \quad \omega_2 \]

\[ \alpha \text{-span} \]
Definition (Hybrid program)

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

Definition (Differential dynamic logic)

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

\[ [\alpha]P \text{ is a shorthand for } (\alpha \cup \beta) \]
Definition (Hybrid program)

\[ \alpha, \beta ::= x := e \mid \text{?}Q \mid x' = f(x) \& Q \mid \alpha \cup \beta \mid \alpha;\beta \mid \alpha^* \]

Definition (Differential dynamic logic)

\[ P, Q ::= e \geq \tilde{e} \mid \neg P \mid P \land Q \mid P \lor Q \mid P \rightarrow Q \mid \forall x \, P \mid \exists x \, P \mid [\alpha]P \mid \langle \alpha \rangle P \]
## Differential Dynamic Logic dL: Semantics

### Definition (Hybrid program semantics) \([\cdot] : \text{HP} \to \wp(\mathcal{S} \times \mathcal{S})\)

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

### Definition (dL semantics) \([\cdot] : \text{Fml} \to \wp(\mathcal{S})\)

- \([e \geq \tilde{e}] = \{\omega : \omega[e] \geq \omega[\tilde{e}]\}\)
- \([\neg P] = [P]^c\)
- \([P \land Q] = [P] \cap [Q]\)
- \([\langle \alpha \rangle P] = [\alpha] \circ [P] = \{\omega : \nu \models P \text{ for some } \nu : (\omega, \nu) \in [\alpha]\}\)
- \([\llbracket \alpha \rrbracket P] = [\neg \langle \alpha \rangle \neg P] = \{\omega : \nu \models P \text{ for all } \nu : (\omega, \nu) \in [\alpha]\}\)
- \([\exists x P] = \{\omega : \omega_x^r \in [P] \text{ for some } r \in \mathbb{R}\}\)

*compositional semantics*
### Differential Dynamic Logic dL: Axiomatization

<table>
<thead>
<tr>
<th>Expression</th>
<th>Equations of Truth</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x := e )</td>
<td>( P(x) \leftrightarrow P(e) )</td>
</tr>
<tr>
<td>( ?Q )</td>
<td>( P \leftrightarrow (Q \rightarrow P) )</td>
</tr>
<tr>
<td>( x' = f(x) )</td>
<td>( P \leftrightarrow \forall t \geq 0 [x := y(t)] P ) (( y'(t) = f(y) ))</td>
</tr>
<tr>
<td>( { ; } )</td>
<td>( [\alpha ; \beta] P \leftrightarrow [\alpha][\beta] P )</td>
</tr>
<tr>
<td>( { [\cdot] } )</td>
<td>( [\alpha][\cdot] P \leftrightarrow \forall t \geq 0 [x := y(t)] P )</td>
</tr>
<tr>
<td>( { [\cdot]^* } )</td>
<td>( [\alpha]^* P \leftrightarrow P \land [\alpha][\alpha]^* P )</td>
</tr>
<tr>
<td>K</td>
<td>( [\alpha](P \rightarrow Q) \rightarrow ([\alpha] P \rightarrow [\alpha] Q) )</td>
</tr>
<tr>
<td>I</td>
<td>( [\alpha]^* P \leftrightarrow P \land [\alpha]^* (P \rightarrow [\alpha] P) )</td>
</tr>
<tr>
<td>C</td>
<td>( [\alpha]^* \forall v &gt; 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)) )</td>
</tr>
</tbody>
</table>

*LICS’12, JAR’17*
Differential Dynamic Logic $dL$: Axiomatization

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

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

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

[∪] \ [α ∪ β] P \leftrightarrow [α] P \land [β] P

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

[*] \ [α*] P \leftrightarrow P \land [α][α*] P

K \ [α](P \rightarrow Q) \rightarrow ([α] P \rightarrow [α] Q)

I \ [α*] P \leftrightarrow P \land [α*](P \rightarrow [α] P)

C \ [α*](\forall v > 0 (P(v) \rightarrow ⟨α⟩ P(v − 1))) \rightarrow \forall v (P(v) \rightarrow ⟨α*⟩ ∃v ≤ 0 P(v))

LICS’12, JAR’17
Differential Dynamic Logic $\text{dL}$: Axiomatization

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

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

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

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

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

$\ast \quad [\alpha^*]P \leftrightarrow P \land [\alpha][\alpha^*]P$

$K \quad [\alpha](P \rightarrow Q) \rightarrow ([\alpha]P \rightarrow [\alpha]Q)$

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

$C \quad [\alpha^*]\forall v > 0(P(v) \rightarrow \langle\alpha\rangle P(v - 1)) \rightarrow \forall v (P(v) \rightarrow \langle\alpha^*\rangle \exists v \leq 0 P(v))$
Differential Dynamic Logic dL: Axiomatization

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

- \forall 
  \[
  \frac{P}{\forall x P} 
  \]

- MP 
  \[
  \frac{P \rightarrow Q \quad P}{Q} 
  \]

rules of truth

LICS'12, JAR'17
Differential Dynamic Logic $dL$: Axiomatization

- **G** \( \frac{P}{[\alpha]P} \)
- **∀** \( \frac{P}{\forall x P} \)
- **MP** \( \frac{P \rightarrow Q \quad P}{Q} \)
- **B** \( \forall x [\alpha]P \rightarrow [\alpha]\forall x P \quad (x \not\in \alpha) \)
- **V** \( p \rightarrow [\alpha]p \quad (FV(p) \cap BV(\alpha) = \emptyset) \)
\[ \begin{align*} 
[x' = f(x) \& Q] P & \iff [x' = f(x)](P) 
\end{align*} \]
\[ [x' = f(x) \& Q]P \iff [x' = f(x)](P) \]
$\left[ x' = f(x) \& Q \right] P$

$\iff \left[ x' = f(x) \right] (\left[ x' = -f(x) \right] (Q \rightarrow P))$

revert flow,
check $Q$ backwards

$X$

$t'$

$Q$

$x' = f(x)$

$x' = -f(x)$
\[
[x' = f(x) \& Q] P \\
\leftrightarrow [x' = f(x)] ([x' = -f(x)](Q) \rightarrow P)
\]
\[
\begin{align*}
[x' = f(x) & Q]P
\iff \\
\forall t_0 = t [x' = f(x)] ([x' = -f(x)] (t \geq t_0 \rightarrow Q) \rightarrow P)
\end{align*}
\]
“There and Back Again” Axiom of dL

\[ [x' = f(x) \& Q] P \leftrightarrow \forall t_0 = t [x' = f(x)] (\lbrack x' = -f(x) \rbrack (t \geq t_0 \rightarrow Q) \rightarrow P) \]

Lemma

**Evolution domain axiomatizable**
Outline

1 Hybrid Systems

2 Differential Dynamic Logic
   - Syntax
   - Semantics
   - Axiomatization

3 Continuous Completeness
   - Schematic Completeness
   - Expressibility and Rendition of Hybrid Programs

4 Discrete Completeness
   - Open Discrete Completeness
   - Closed Discrete Completeness
   - Semialgebraic Discrete Completeness of $dL + \Delta$
   - Discrete Completeness of $dL + \Delta$
   - Equi-expressible
   - Relative Decidable

5 Summary
Theorem (Relative Completeness / Continuous) (JAR’08,LICS’12)

dL calculus is a sound & complete axiomatization of hybrid systems relative to differential equations:

\[ \models \phi \iff \text{Taut}_{\text{FOD}} \vdash \phi \]

Corollary (Complete Proof-theoretical Alignment)

proving: continuous = hybrid

Corollary (Compositionality)

hybrid systems can be verified by recursive decomposition

\[ \text{FOD} = \text{FOL} + [x' = f(x)]F \]
### Theorem (Relative Completeness / Continuous)  
(JAR’08,LICS’12)

\[ dL \text{ calculus is a sound & complete axiomatization of hybrid systems relative to differential equations:} \]

\[ \vdash \phi \text{ iff } \text{Taut}_{\text{FOD}} \vdash \phi \]

### Theorem (Relative Completeness / Discrete)  
(LICS’12)

\[ dL \text{ calculus is a sound & complete axiomatization of hybrid systems relative to discrete dynamics:} \]

\[ \vdash \phi \text{ iff } \text{Taut}_{\text{DL}} \vdash \phi \]

### Corollary (Complete Proof-theoretical Alignment)

proving: continuous = hybrid = discrete

### Corollary (Interdisciplinary Integrability)

“Discrete mathematics + continuous mathematics are integrable”
<table>
<thead>
<tr>
<th>Schematic Completeness</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Theorem (Relative Completeness / Continuous)</strong> (JAR’08,LICS’12)</td>
</tr>
<tr>
<td>( dL ) calculus is a sound &amp; complete axiomatization of hybrid systems relative to <strong>differential equations:</strong></td>
</tr>
<tr>
<td>( \vdash \phi \iff \text{Taut}_{\text{FOD}} \vdash \phi )</td>
</tr>
<tr>
<td><strong>Theorem (Relative Completeness / Discrete)</strong> (LICS’12)</td>
</tr>
<tr>
<td>( dL ) calculus is a sound &amp; complete axiomatization of hybrid systems relative to <strong>discrete dynamics:</strong></td>
</tr>
<tr>
<td>( \vdash \phi \iff \text{Taut}_{\text{DL}} \vdash \phi )</td>
</tr>
<tr>
<td><strong>Theorem (Schematic Completeness)</strong> (JAR’17)</td>
</tr>
<tr>
<td>( dL ) calculus is a sound &amp; complete axiomatization of hybrid systems relative to <strong>any (differentially) expressive logic</strong> ( L ):</td>
</tr>
<tr>
<td>( \vdash \phi \iff \text{Taut}_L \vdash \phi )</td>
</tr>
</tbody>
</table>

**Differentially expressive** |
\[
\forall \phi \in dL \exists \phi^b \in L \vdash \phi \iff \phi^b \text{ and } \forall \phi \in L \vdash_L \langle x' = f(x) \rangle \phi \iff (\langle x' = f(x) \rangle \phi)^b
\]
Proof of “continuous = hybrid = discrete”
**Schematic Completeness Proof**

**Proof Sketch (φ in NNF, induction on well-founded ≺) (JAR’17).**

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>φ first-order formula ⇒ φ ∈ L so ⊨_L φ if ⊨ φ (Also for ¬φ₁ by NNF)</td>
</tr>
<tr>
<td>1</td>
<td>φ ≡ φ₁ ∧ φ₂ ⇒ ⊨ φ₁ and ⊨ φ₂ (\overset{IH}{\Rightarrow}) ⊨_L φ₁ and ⊨_L φ₂ ⇒ ⊨_L φ₁ ∧ φ₂.</td>
</tr>
<tr>
<td>2</td>
<td>φ ≡ ∃x φ₂, ∀x φ₂, ⟨α⟩φ₂ or [α]φ₂ covered in next case with φ₁ ≡ false.</td>
</tr>
<tr>
<td>3</td>
<td>φ ≡ φ₁ ∨ ⟨[α]⟩φ₂ is (by associativity and commutativity to reorder):</td>
</tr>
<tr>
<td></td>
<td>φ₁ ∨ ⟨α⟩φ₂  φ₁ ∨ ∃x φ₂</td>
</tr>
<tr>
<td></td>
<td>φ₁ ∨ [α]φ₂  φ₁ ∨ ∀x φ₂</td>
</tr>
</tbody>
</table>

Then, φ₂ ≺ φ and φ₁ ≺ φ as less HP/quantifier. Let F ≡ ¬φ₁ and G ≡ φ₂ then ⊨ F → ⟨[α]⟩ G. Show ⊨_L F → ⟨[α]⟩ G, which derives ⊨_L φ₁ ∨ ⟨[α]⟩ φ₂.

\[ ⊨_L φ \text{ iff Taut}_L \vdash φ \]

≺ is lexicographic order of HP, formula, with L at the bottom.

---

André Platzer (CMU)
Proof Sketch (\(\phi\) in NNF, induction on well-founded \(\prec\)) (JAR’17).

4. \(\{\alpha\} \equiv \forall x\) with \(\models F \rightarrow \forall x G\), wlog \(x \not\in F\) by bound variable renaming.

Hence, \(\models F \rightarrow G \models \vdash F \rightarrow G \) as \((F \rightarrow G) \prec (F \rightarrow \forall x G)\) less \(\forall\).

\[
\begin{align*}
\forall & \frac{}{\forall x (F \rightarrow G)} \\
\forall & \frac{}{\forall x F \rightarrow \forall x G} \\
\forall\forall & \frac{}{F \rightarrow \forall x G}
\end{align*}
\]

5. \(\{\alpha\} \equiv \exists x\) with \(\models F \rightarrow \exists x G\). Have \(\models G^b \leftrightarrow G \models \models F \rightarrow \exists x (G^b) \models \vdash F \rightarrow \exists x (G^b) \) as \((F \rightarrow \exists x (G^b)) \prec (F \rightarrow \exists x G)\) as \(G^b \in L\).

Also \(\models G^b \leftrightarrow G \models \models G^b \rightarrow G \models \vdash G^b \rightarrow G\) since \((G^b \rightarrow G) \prec \phi\) as \(G^b \in L\).

\[
\begin{align*}
\forall & \frac{}{\forall x (G^b \rightarrow G)} \\
\forall & \frac{}{\exists x (G^b) \rightarrow \exists x G}
\end{align*}
\]

\(F \rightarrow \exists x (G^b)\)

\[
\begin{align*}
\forall & \frac{}{\exists x (G^b) \rightarrow \exists x G}
\end{align*}
\]

\(F \rightarrow \exists x G\)
Proof Sketch (φ in NNF, induction on well-founded ≺) (JAR’17).

6. \[ F \rightarrow \langle x' = f(x) \rangle G \] implies \[\vdash F \rightarrow (\langle x' = f(x) \rangle G^b)^b \]
   \[\vdash_L F \rightarrow (\langle x' = f(x) \rangle G^b)^b \] as \( (\langle x' = f(x) \rangle G^b)^b \in L \) is smaller.
   \[\vdash_L \langle x' = f(x) \rangle G^b \leftrightarrow (\langle x' = f(x) \rangle G^b)^b \] as L differentially expressive.
   By IH \[\vdash_L G^b \rightarrow G \] as \( G^b \in L \). So \[\vdash_L \langle x' = f(x) \rangle G^b \rightarrow \langle x' = f(x) \rangle G \] by M.
   Thus \[\vdash_L F \rightarrow \langle x' = f(x) \rangle G \] propositionally.

7. \[ F \rightarrow [?Q] G \] implies \[\vdash F \rightarrow (Q \rightarrow G) \]
   \[\vdash_L F \rightarrow (Q \rightarrow G) \] since \( (Q \rightarrow G) \prec [?Q] G \). Thus \[\vdash_L F \rightarrow [?Q] G \] as \([?Q] G \leftrightarrow (Q \rightarrow G) \) by [?].

8. \[ F \rightarrow [\beta \cup \gamma] G \] implies \[\vdash F \rightarrow [\beta] G \land [\gamma] G \]
   \[\vdash_L F \rightarrow [\beta] G \land [\gamma] G \] as \([\beta] G \land [\gamma] G \prec [\beta \cup \gamma] G \) has smaller HP. Thus \[\vdash_L F \rightarrow [\beta \cup \gamma] G \] by [\cup].

9. \[ F \rightarrow [\beta ; \gamma] G \] implies \[\vdash F \rightarrow [\beta][\gamma] G \]
   \[\vdash_L F \rightarrow [\beta][\gamma] G \] as \([\beta][\gamma] G \prec [\beta ;\gamma] G \) has smaller HP. Thus \[\vdash_L F \rightarrow [\beta ; \gamma] G \] by [;].
Proof Sketch ($\phi$ in NNF, induction on well-founded $\prec$) (JAR'17).

\[ \models F \rightarrow [y := \theta]G. \text{ Rename bound variable to fresh variable } x \text{ where } G^x_y \text{ is the result of uniformly renaming } y \text{ to } x \text{ in } G: \]

\[ F \rightarrow \forall x (x = \theta \rightarrow G^x_y) \]

\[ \models [x := \theta]G^x_y \]

\[ F \rightarrow [y := \theta]G \]

using the derivable equational form of the assignment axiom $[:=]$

\[ [:=] = [x := f]P \leftrightarrow \forall x (x = f \rightarrow P) \]

Only used equivalences, so premise valid iff conclusion valid.

\[ \models F \rightarrow \forall x (x = \theta \rightarrow G^x_y) \supset \vdash L F \rightarrow \forall x (x = \theta \rightarrow G^x_y) \text{ as } \]

\[ (F \rightarrow \forall x (x = \theta \rightarrow G^x_y)) \prec (F \rightarrow [y := \theta]G) \text{ has less modalities.} \]
Proof Sketch (\(\phi\) in NNF, induction on well-founded \(\prec\)) \(\text{(JAR'17).}\)

\[\vdash F \rightarrow [\beta^*]G.\] Formula \([\beta^*]G\) is loop invariant as \(\vdash [\beta^*]G \rightarrow [\beta][\beta^*]G.\)

\[J \equiv ([\beta^*]G) \downarrow\] equivalent loop invariant in simpler \(L\)

Then \(\vdash F \rightarrow J\) and \(\vdash J \rightarrow G\) \(\text{IH}\) \(\vdash_L F \rightarrow J\) and \(\vdash_L J \rightarrow G\) since \((F \rightarrow J) \prec \phi\) and \((J \rightarrow G) \prec \phi\) as \(J \in L\) is smaller.

Moreover \(\vdash J \rightarrow [\beta]J\) \(\text{IH}\) \(\vdash_L J \rightarrow [\beta]J\) since \(\beta\) has less loops than \(\beta^*\).

\[
\begin{align*}
J \rightarrow [\beta]J & \quad \text{ind} \quad J \rightarrow [\beta^*]J & \quad \text{M[\cdot]} [\beta^*]J \rightarrow [\beta^*]G \\
F \rightarrow J & \quad \text{MP} & \quad J \rightarrow [\beta^*]G & \quad \text{MP} & \quad F \rightarrow [\beta^*]G
\end{align*}
\]
Proof Sketch ($\phi$ in NNF, induction on well-founded $\prec$) (JAR’17).

$\models F \rightarrow \langle \beta^* \rangle G$. Let $x = \text{FV}(\langle \beta^* \rangle G)$. Since $\langle \beta^* \rangle G$ is a least pre-fixpoint:

$$
\models \forall x (G \lor \langle \beta \rangle p(x) \rightarrow p(x)) \rightarrow (\langle \beta^* \rangle G \rightarrow p(x))
$$

As $\models F \rightarrow \langle \beta^* \rangle G$ also $\models \forall x (G \lor \langle \beta \rangle p(x) \rightarrow p(x)) \rightarrow (F \rightarrow p(x))$ \(\xrightarrow{\text{IH}}\) $\models L \forall x (G \lor \langle \beta \rangle p(x) \rightarrow p(x)) \rightarrow (F \rightarrow p(x))$ as

$$(\forall x (G \lor \langle \beta \rangle p(x) \rightarrow p(x)) \rightarrow (F \rightarrow p(x))) \prec \phi.$$ \(\sigma = \{p(x) \mapsto \langle \beta^* \rangle G\}\)

admissible since $\text{FV}(\sigma) = \emptyset$ as $x = \text{FV}(\langle \beta^* \rangle G)$ and since $p$ is fresh:

\[
\begin{align*}
\forall x (G \lor \langle \beta \rangle p(x) \rightarrow p(x)) & \rightarrow (F \rightarrow p(x)) & \text{[\(\star\)]}, \langle \cdot \rangle \\
\forall x (G \lor \langle \beta \rangle \langle \beta^* \rangle G \rightarrow \langle \beta^* \rangle G) & \rightarrow (F \rightarrow \langle \beta^* \rangle G) & \forall \\
\forall x (G \lor \langle \beta \rangle \langle \beta^* \rangle G \rightarrow \langle \beta^* \rangle G) & \rightarrow (F \rightarrow \langle \beta^* \rangle G) & \forall \\
\end{align*}
\]

Note: could also use modified $(\langle \beta^* \rangle G)^{\flat}$ with convergence rule con.
### Theorem (Relative Completeness / Continuous) (JAR’08, LICS’12)

**dL calculus** is a sound & complete axiomatization of hybrid systems relative to differential equations:

\[ \models \phi \iff \text{Taut}_{FOD} \vdash \phi \]
Continuous Completeness

**Theorem (Relative Completeness / Continuous) (JAR’08,LICS’12)**

\( dL \) calculus is a sound & complete axiomatization of hybrid systems relative to *differential equations*:

\[ \models \phi \iff \text{Taut}_{\text{FOD}} \vdash \phi \]

**Theorem (Schematic Completeness) (JAR’17)**

\( dL \) calculus is a sound & complete axiomatization of hybrid systems relative to *any (differentially) expressive logic* \( L \):

\[ \models \phi \iff \text{Taut}_{\text{L}} \vdash \phi \]

**Differentially expressive**

\[ \forall \phi \in dL \exists \phi^b \in L \models \phi \leftrightarrow \phi^b \text{ and } \forall \phi \in L \vdash \langle x' = f(x) \rangle \phi \leftrightarrow (\langle x' = f(x) \rangle \phi)^b \]
Lemma (dL Expressibility in FOD)

\[ \forall \phi \in dL \exists \phi^b \in FOD \models \phi \leftrightarrow \phi^b \quad \text{and} \quad \forall \phi \in FOD \vdash_L \langle x' = f(x) \rangle \phi \leftrightarrow (\langle x' = f(x) \rangle \phi)^b \]

Proof Sketch.

1. Strong enough invariants and variants expressible in dL!
2. dL expressible in FOD?
3. Finite FOD formula characterizing unbounded hybrid repetition.
4. FOD characterizes \( \mathbb{R} \)-Gödel encoding (pairing/unpairing on \( \mathbb{R} \)).
5. FOD characterizes HP transitions.
6. FOD expresses dL formulas.

\[ FOD = \text{FOL}_{\mathbb{R}} + [x' = f(x)]F \]
Lemma (dL Expressibility in FOD)

∀φ ∈ dL ∃φ^♭ ∈ FOD ⊨ φ ↔ φ^♭ and ∀φ ∈ FOD ⊢ₗ ⟨x' = f(x)⟩ φ ↔ (⟨x' = f(x)⟩ φ)^♭

Proof Sketch.

1. Strong enough invariants and variants expressible in dL!
2. dL expressible in FOD?
3. Finite FOD formula characterizing unbounded hybrid repetition.
4. FOD characterizes \( \mathbb{R} \)-Gödel encoding (pairing/unpairing on \( \mathbb{R} \)).
5. FOD characterizes HP transitions.
6. FOD expresses dL formulas.

\[ \text{FOD} = \text{FOL}_\mathbb{R} + [x' = f(x)]F \]
Lemma (dL Expressibility in FOD)

\[ \forall \phi \in dL \exists \phi^b \in FOD \models \phi \iff \phi^b \quad \text{and} \quad \forall \phi \in FOD \vdash L \langle x' = f(x) \rangle \phi \iff (\langle x' = f(x) \rangle \phi)^b \]

Proof Sketch.

1. Strong enough invariants and variants expressible in dL!
2. dL expressible in FOD?
3. Finite FOD formula characterizing unbounded hybrid repetition.
4. FOD characterizes \( \mathbb{R} \)-Gödel encoding (pairing/unpairing on \( \mathbb{R} \)).
5. FOD characterizes HP transitions.
6. FOD expresses dL formulas.

\[ FOD = FOL_\mathbb{R} + [x' = f(x)]F \]
### Lemma (dL Expressibility in FOD)

\[ \forall \phi \in dL \exists \phi^b \in FOD \models \phi \leftrightarrow \phi^b \quad \text{and} \quad \forall \phi \in FOD \models L(x') = f(x) \phi \leftrightarrow (L(x') = f(x) \phi)^b \]

### Proof Sketch.

1. Strong enough invariants and variants expressible in dL!
2. dL expressible in FOD?
3. Finite FOD formula characterizing unbounded hybrid repetition.
4. FOD characterizes \( \mathbb{R} \)-Gödel encoding (pairing/unpairing on \( \mathbb{R} \)).
5. FOD characterizes HP transitions.
6. FOD expresses dL formulas.

\[ FOD = \text{FOL}_{\mathbb{R}} + [x' = f(x)]F \]
Continuous Completeness Proof

\[ \text{FOD} = \text{FOL}_\mathbb{R} + [x' = f(x)]F \]

Proof Sketch (\(\mathbb{R}\)-Gödel encoding).

FOD characterizes constructive bijection \(\mathbb{R} \to \mathbb{R}^2\)
FOD = FOL_\mathbb{R} + [x' = f(x)]F

Proof Sketch (\mathbb{R}-Gödel encoding).

FOD characterizes constructive bijection \mathbb{R} \rightarrow \mathbb{R}^2
FOD = FOL\(\mathbb{R}\) + \([x' = f(x)]F\)

Proof Sketch (\(\mathbb{R}\)-Gödel encoding).

FOD characterizes constructive bijection \(\mathbb{R} \rightarrow \mathbb{R}^2\)
FOD = FOL_\mathbb{R} + [x' = f(x)]F

Proof Sketch (\mathbb{R}-Gödel encoding).

FOD characterizes constructive bijection \mathbb{R} \rightarrow \mathbb{R}^2 not differentiable, Morayne!
Continuous Completeness Proof

\[ \text{FOD} = \text{FOL}_\mathbb{R} + [x' = f(x)]F \]

**Proof Sketch (\(\mathbb{R}\)-Gödel encoding).**

FOD characterizes constructive bijection \(\mathbb{R} \rightarrow \mathbb{R}^2\)

\[ \sum_{i=0}^{\infty} \frac{a_i}{2^i} = a_0.a_1a_2\ldots \quad \sum_{i=0}^{\infty} \left( \frac{a_i}{2^{2i-1}} + \frac{b_i}{2^{2i}} \right) = a_0b_0.a_1b_1a_2b_2\ldots \]

\[ \sum_{i=0}^{\infty} \frac{b_i}{2^i} = b_0.b_1b_2\ldots \]
FOD = FOL_\mathbb{R} + [x' = f(x)]F

Proof Sketch (\mathbb{R}-Gödel encoding).

FOD characterizes constructive bijection \mathbb{R} \to \mathbb{R}^2

\[ \sum_{i=0}^{\infty} \frac{a_i}{2^i} = a_0.a_1a_2\ldots \]
\[ \sum_{i=0}^{\infty} \frac{b_i}{2^i} = b_0.b_1b_2\ldots \]

\[ Z_j^{(n)} = z \text{ is } j\text{th } \mathbb{R} \text{ of } n \text{ reals } Z \]

\[ \text{at}(Z, n, j, z) \leftrightarrow \forall i: \mathbb{Z} \text{ digit}(z, i) = \text{digit}(Z, n(i - 1) + j) \land n > 0 \land n, j \in \mathbb{N} \]
\[ \text{digit}(a, i) = \text{intpart}(2\text{frac}(2^{i-1}a)) \]
\[ \text{intpart}(a) = a - \text{frac}(a) \]
\[ \text{frac}(a) = z \leftrightarrow \exists i: \mathbb{Z} \ z = a - i \land -1 < z \land z < 1 \land az \geq 0 \quad \text{“keep sign”} \]
Continuous Completeness Proof

\[ \text{FOD} = \text{FOL}_\mathbb{R} + [x' = f(x)]F \]

Proof Sketch (\(\mathbb{R}\)-Gödel encoding).

FOD characterizes constructive bijection \(\mathbb{R} \rightarrow \mathbb{R}^2\)

\[
\sum_{i=0}^{\infty} \frac{a_i}{2^i} = a_0.a_1a_2\ldots 
\]
\[
\sum_{i=0}^{\infty} \frac{b_i}{2^i} = b_0.b_1b_2\ldots 
\]

\[
\sum_{i=0}^{\infty} \left( \frac{a_i}{2^{2i-1}} + \frac{b_i}{2^{2i}} \right) = a_0b_0.a_1b_1a_2b_2\ldots 
\]

\[
\text{at}(Z, n, j, z) \iff \forall i: \mathbb{Z} \ \text{digit}(z, i) = \text{digit}(Z, n(i - 1) + j) \land n > 0 \land n, j \in \mathbb{N} \\
\text{digit}(a, i) = \text{intpart}(2 \text{frac}(2^{i-1}a)) \\
\text{intpart}(a) = a - \text{frac}(a) \\
\text{frac}(a) = z \iff \exists i: \mathbb{Z} \ z = a - i \land -1 < z \land z < 1 \land az \geq 0 \quad \text{“keep sign”} 
\]
Continuous Completeness Proof

\[ \text{FOD} = \text{FOL}_\mathbb{R} + [x' = f(x)]F \]

Proof Sketch (\(\mathbb{R}\)-Gödel encoding).

FOD characterizes constructive bijection \(\mathbb{R} \rightarrow \mathbb{R}^2\)

\[
\sum_{i=0}^\infty \frac{a_i}{2^i} = a_0.a_1a_2\ldots \\
\sum_{i=0}^\infty \frac{b_i}{2^i} = b_0.b_1b_2\ldots \\
\sum_{i=0}^\infty \left( \frac{a_i}{2^{2i-1}} + \frac{b_i}{2^{2i}} \right) = a_0b_0.a_1b_1a_2b_2\ldots
\]

\[
2^i = z \leftrightarrow i \geq 0 \land \langle x := 1; t := 0; x' = x \ln 2, t' = 1 \rangle (t = i \land x = z) \\
\lor i < 0 \land \langle x := 1; t := 0; x' = -x \ln 2, t' = -1 \rangle (t = i \land x = z) \\
\ln 2 = z \leftrightarrow \langle x := 1; t := 0; x' = x, t' = 1 \rangle (x = 2 \land t = z)
\]

syntactic abbreviation without recursion
Lemma (dL Expressibility in FOD)

\[ \forall \phi \in \text{dL} \exists \phi^b \in \text{FOD} \models \phi \iff \phi^b \quad \text{and} \quad \forall \phi \in \text{FOD} \vdash_L \langle x' = f(x) \rangle \phi \iff (\langle x' = f(x) \rangle \phi)^b \]

Proof Sketch.

1. Strong enough invariants and variants expressible in dL!
2. dL expressible in FOD?
3. Finite FOD formula characterizing unbounded hybrid repetition.
4. FOD characterizes \( \mathbb{R} \)-Gödel encoding (pairing/unpairing on \( \mathbb{R} \)).
5. FOD characterizes HP transitions.
6. FOD expresses dL formulas.

\[ \text{FOD} = \text{FOL}_{\mathbb{R}} + [x' = f(x)]F \]
Lemma (dL Expressibility in FOD)

\[ \forall \varphi \in dL \exists \varphi^b \in FOD \models \varphi \iff \varphi^b \text{ and } \forall \varphi \in FOD \vdash_L \langle x' = f(x) \rangle \varphi \iff (\langle x' = f(x) \rangle \varphi)^b \]

Proof Sketch.

1. Strong enough invariants and variants expressible in dL!
2. dL expressible in FOD?
3. Finite FOD formula characterizing unbounded hybrid repetition.
4. FOD characterizes \( \mathbb{R} \)-Gödel encoding (pairing/unpairing on \( \mathbb{R} \)).
5. FOD characterizes HP transitions.
6. FOD expresses dL formulas.

\[ \text{FOD} = \text{FOL}_\mathbb{R} + [x' = f(x)]F \]
Lemma (dL Expressibility in FOD)

\[ \forall \phi \in \text{dL} \exists \phi^b \in \text{FOD} \models \phi \iff \phi^b \text{ and } \forall \phi \in \text{FOD} \vdash_L \langle x' = f(x) \rangle \phi \iff (\langle x' = f(x) \rangle \phi)^b \]

Proof Sketch.

1. Strong enough invariants and variants expressible in dL!
2. dL expressible in FOD?
3. Finite FOD formula characterizing unbounded hybrid repetition.
4. FOD characterizes \( \mathbb{R} \)-Gödel encoding (pairing/unpairing on \( \mathbb{R} \)).
5. FOD characterizes HP transitions.
6. FOD expresses dL formulas.

\[ \text{FOD} = \text{FOL}_{\mathbb{R}} + [x' = f(x)]F \]
Lemma (Program rendition)

∀α∈HP with variables among x = x₁, ..., xₖ ∃Σᵢ(x, v)∈FOD with variables among distinct x = x₁, ..., xₖ and v = v₁, ..., vₖ:

⊢ Σᵢ(x, v) ↔ ⟨α⟩x = v

Proof Sketch (by induction on α).

Sₓᵢ := θ(x, v) ≡ vᵢ = θ ∧ ⋀ⱼ≠i(vⱼ = xⱼ)

Sₓ′ := θ(x, v) ≡ ⟨x′ = θ⟩v = x

Sₓ′ := θ & Q(x, v) ≡ ∃t(t=0 ∧ ⟨x′ = θ, t′=1⟩(v=x ∧ [x′=−θ, t′=−1](t≥0→Q)))

S?Q(x, v) ≡ v = x ∧ Q

Σβ∪γ(x, v) ≡ Σβ(x, v) ∨ Σγ(x, v)

Σβ;γ(x, v) ≡ ∃z(Σβ(x, z) ∧ Σγ(z, v))

Σβ∗(x, v) ≡ ∃Z ∃n:N (Z₁⁽ⁿ⁾=x ∧ Zₙ⁽ⁿ⁾=v ∧ ∀i:N (1 ≤ i < n → Σβ(Zᵢ⁽ⁿ⁾, Zᵢ₊₁⁽ⁿ⁾)))
Lemma (Program rendition)

∀α ∈ HP with variables among x = x₁, ..., xₖ
∃I_α(x, v) ∈ FOD with variables among distinct x = x₁, ..., xₖ and v = v₁, ..., vₖ:

I_α(x, v) ↔ ⟨α⟩ x = v

Proof Sketch (by induction on α).

I_{x_i} := θ(x, v) ≡ vᵢ = θ ∧ \bigwedge_{j \neq i}(v_j = x_j)

I_{x'} := θ(x, v) ≡ ⟨x' = θ⟩ v = x

I_{x' = θ \& Q}(x, v) ≡ ∃t(t = 0 ∧ ⟨x' = θ, t' = 1⟩(v = x ∧ [x' = −θ, t' = −1](t ≥ 0 → Q)))

I_{?Q}(x, v) ≡ v = x ∧ Q

I_β \cup \gamma(x, v) ≡ I_β(x, v) ∨ I_γ(x, v)

I_β;\gamma(x, v) ≡ ∃z(I_β(x, z) ∧ I_γ(z, v))

I_β^*(x, v) ≡ ∃Z ∃n: \mathbb{N}(Z_1^{(n)} = x ∧ Z_n^{(n)} = v ∧ ∀i: \mathbb{N}(1 ≤ i < n → I_β(Z_i^{(n)}, Z_{i+1}^{(n)}))))
Lemma (Program rendition)

\[ \forall \alpha \in \text{HP} \text{ with variables among } x = x_1, \ldots, x_k \ \exists \mathcal{I}_\alpha(x, v) \in \text{FOD with variables among distinct } x = x_1, \ldots, x_k \text{ and } v = v_1, \ldots, v_k : \ \models \mathcal{I}_\alpha(x, v) \leftrightarrow \langle \alpha \rangle x = v \]

Proof Sketch (by induction on \( \alpha \)).

\[ \mathcal{I}_{x' = \theta \& Q}(x, v) \equiv \exists t \left( t = 0 \land \langle x' = \theta, t' = 1 \rangle (v = x \land [x' = -\theta, t' = -1](t \geq 0 \rightarrow Q)) \right) \]

\[ \equiv \exists t \exists r \left( t = 0 \land \langle x' = \theta, t' = 1 \rangle (v = x \land r = t) \land \forall x \forall t \left( x = v \land t = r \rightarrow [x' = -\theta, t' = -1](t \geq 0 \rightarrow Q) \right) \right) \]

\[ x' = \theta \]

\[ x' = -\theta \]

revert flow and time check \( Q \) backwards
Lemma (dL Expressibility in FOD)

\[ \forall \phi \in \text{dL} \ \exists \phi^b \in \text{FOD} \models \phi \leftrightarrow \phi^b \]

Proof (by induction on \( \phi \)).

1. \( \phi \) first-order, then \( \phi^b := \phi \) already is a FOD-formula.
2. \( \phi \equiv \phi \lor \psi \xRightarrow{IH} \) have \( \phi^b, \psi^b \) such that \( \models \phi \leftrightarrow \phi^b \) and \( \models \psi \leftrightarrow \psi^b \). By congruence \( \models (\phi \lor \psi) \leftrightarrow (\phi^b \lor \psi^b) \) giving \( \models \phi \leftrightarrow \phi^b \) for \( \phi^b \equiv \phi^b \lor \psi^b \).
3. Likewise for propositional connectives or quantifiers.
4. \( \phi \equiv \langle \alpha \rangle \psi \) uses \( \models \langle \alpha \rangle \psi \leftrightarrow \exists v (S_\alpha(x, v) \land \psi^b_{\frac{v}{x}}) \)
5. \( \phi \equiv [\alpha] \psi \) uses \( \models [\alpha] \psi \leftrightarrow \forall v (S_\alpha(x, v) \rightarrow \psi^b_{\frac{v}{x}}) \)
Outline

1 Hybrid Systems

2 Differential Dynamic Logic
   - Syntax
   - Semantics
   - Axiomatization

3 Continuous Completeness
   - Schematic Completeness
   - Expressibility and Rendition of Hybrid Programs

4 Discrete Completeness
   - Open Discrete Completeness
   - Closed Discrete Completeness
   - Semialgebraic Discrete Completeness of dL +Δ
   - Discrete Completeness of dL +Δ
   - Equi-expressible
   - Relative Decidable

5 Summary
Discrete Completeness

Theorem (Relative Completeness / Continuous) (JAR’08, LICS’12)

\( d\text{L calculus} \) is a sound & complete axiomatization of hybrid systems relative to differential equations:

\( \models \phi \iff \text{Taut}_{\text{FOD}} \vdash \phi \)

Theorem (Relative Completeness / Discrete) (LICS’12)

\( d\text{L calculus} \) is a sound & complete axiomatization of hybrid systems relative to discrete dynamics:

\( \models \phi \iff \text{Taut}_{\text{DL}} \vdash \phi \)

Corollary (Complete Proof-theoretical Alignment)

proving: continuous = hybrid = discrete
Continuous Dynamics ≠ Discrete Dynamics

\[ x' = \frac{x}{4}F \]
Continuous Dynamics $\neq$ Discrete Dynamics

\[ [x' = \frac{x}{4}] F \quad (x := x + h \frac{x}{4})^* ] F \]
Continuous Dynamics ≠ Discrete Dynamics

\[ x' = \frac{x}{4} \] \( F \) ⇄ \[ (x := x + h \frac{x}{4})^* \] \( F \)
Continuous Dynamics \neq \text{Discrete Dynamics}

\[
\begin{align*}
[x' &= \frac{x}{4}] F \\
[(x := x + \frac{h x}{4})^*] F
\end{align*}
\]
Continuous Dynamics $\neq$ Discrete Dynamics

\[
\begin{align*}
[x' = \frac{x}{4}]F & \quad [(x := x + h\frac{x}{4})^*]F
\end{align*}
\]

\[
\begin{align*}
x(t) & \\
t & 0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6
\end{align*}
\]

\[
\begin{align*}
x & 1 \quad 1.25 \quad 1.56 \quad 1.95 \quad 2.44 \quad 3.05 \quad 3.81 \quad 4.76
\end{align*}
\]

\[
\begin{align*}
h = 1 \quad h = 2 \quad h = 4
\end{align*}
\]
Continuous Dynamics $\neq$ Discrete Dynamics

\[
\begin{align*}
[x' &= \frac{x}{4}]F \\
[(x := x + h\frac{x}{4})^*]F
\end{align*}
\]
Continuous Dynamics \neq \text{Discrete Dynamics}

\[ x' = \frac{x}{4}F \quad \text{vs.} \quad [(x := x + h \frac{x}{4})]F \]

\[ x \]

\[ t \]

\[ h = \frac{1}{2}, h = 1, h = 2, h = 4 \]

\[ e^{\frac{t}{4}} \]

\[ t_0 \]

\[ 1 \]
Continuous Dynamics $\neq$ Discrete Dynamics

\[ x' = \frac{x}{4} \]
\[ F \Rightarrow \left[ (x := x + h\frac{x}{4})^* \right] F \]
Continuous Dynamics ≠ Discrete Dynamics

\[
\frac{x'}{4} \neq (x := x + \frac{x}{4})^* F
\]
Discrete Euler Approximation Axiom

\[ \Delta \]

\[ [x' = f(x)] F \]

\[ \leftarrow \exists h_0 > 0 \forall 0 < h < h_0 [(x := x + hf(x))^*] F \]
Discrete Euler Approximation Axiom \( \Delta \)

\[
\begin{align*}
\Delta & \quad [x' = f(x)]F \\
& \iff \exists h_0 > 0 \forall 0 < h < h_0 \ [(x := x + hf(x))^*]F
\end{align*}
\]

Example (Incomplete, not global)

\[
\models x^2 + y^2 \leq 1 \rightarrow [x' = y, y' = -x] x^2 + y^2 \leq 1.1
\]
Discrete Euler Approximation Axiom

\[ [x' = f(x)] F \]
\[ \leftarrow \exists h_0 > 0 \forall 0 < h < h_0 \left[ (x := x + hf(x))^* \right] F \]

(closed)

Example (Unsound for open $F$, only in closure)

\[ \not\models x = 1 \land y = 0 \rightarrow [x' = y, y' = -x] (x \leq 0 \rightarrow x^2 + y^2 > 1) \]
Discrete Euler Approximation Axiom $\triangleleft$

$\triangleleft [x' = f(x)]F$

$\triangleleft \exists h_0 > 0 \forall 0 < h < h_0 [(x := x + hf(x))^*]F$

(closed)

Example (Incomplete, not global)

$\vdash x^2 + y^2 \leq 1 \rightarrow [x' = y, y' = -x]x^2 + y^2 \leq 1.1$
Discrete Euler Approximation Axiom \( \rightarrow \Delta \)

\[
\Delta [x' = f(x)] F \Rightarrow \forall t \geq 0 \exists h_0 > 0 \forall 0 < h < h_0 [(x := x + hf(x))^*](t \geq 0 \rightarrow F)
\]
Discrete Euler Approximation Axiom $\vec{\Delta}$

\[ [x' = f(x)] F \rightarrow \forall t \geq 0 \exists h_0 > 0 \forall 0 < h < h_0 \left[ (x := x + hf(x))^* \right] (t \geq 0 \rightarrow F) \]

Example (Converse of $\vec{\Delta}$ unsound for open $F$ closed $F$ by $\vec{\Delta}$)

\[ \not\equiv x = 1 \land y = 0 \rightarrow [x' = y, y' = -x] (x \leq 0 \rightarrow x^2 + y^2 > 1) \]
Discrete Euler Approximation Axiom $\Delta$

$[x' = f(x)]F \rightarrow \forall t \geq 0 \exists h_0 > 0 \forall 0 < h < h_0 [(x := x + hf(x))^*](t \geq 0 \rightarrow F)$ (open)

Example (Unsound for closed $F$, only holds in the limit)

$\models x^2 + y^2 = 1 \rightarrow [x' = y, y' = -x]x^2 + y^2 = 1$
Discrete Euler Approximation Axiom \( \Delta \)

\[ x' = f(x) F \]

\( \leftrightarrow \forall t \geq 0 \exists \varepsilon > 0 \exists h_0 > 0 \forall 0 < h < h_0 \left[ (x := x + hf(x))^* \right] (t \geq 0 \rightarrow \neg \mathcal{U}_\varepsilon (\neg F)) \]
Discrete Euler Approximation Axiom \( \Delta \)  

\[ \begin{align*} 
\Delta & \quad [x' = f(x)] F \\
\iff & \quad \forall t \geq 0 \exists \varepsilon > 0 \exists h_0 > 0 \forall 0 < h < h_0 [(x := x + hf(x))^*](t \geq 0 \rightarrow \neg \mathcal{U}_\varepsilon (\neg F)) 
\end{align*} \]

Example ()  

\[ \models x^2 + y^2 < 1 \rightarrow [x' = y, y' = -x] x^2 + y^2 < 1.1 \]
Discrete Euler Approximation Axiom

\[ [x' = f(x)] F \]
\[ \leftrightarrow \forall t \geq 0 \exists \epsilon > 0 \exists h_0 > 0 \forall 0 < h < h_0 [(x := x + hf(x))^*] (t \geq 0 \rightarrow \neg U_\epsilon (\neg F)) \]

Example (Incomplete for closed $F$)

\[ \models x^2 + y^2 \leq 1 \rightarrow [x' = y, y' = -x] x^2 + y^2 \leq 1 \]
Discrete Euler Approximation Axiom $\Delta$

\[
\begin{align*}
\leftarrow & \Delta \\
\iff & [x' = f(x)] F \\
\iff & \forall t \geq 0 \exists \varepsilon > 0 \exists h_0 > 0 \forall 0 < h < h_0 [(x := x + hf(x))^*] (t \geq 0 \rightarrow \neg U_\varepsilon (\neg F))
\end{align*}
\]

Example (Incomplete for closed $F$)

\[
\vdash x^2 + y^2 \leq 1 \rightarrow [x' = y, y' = -x] x^2 + y^2 \leq 1
\]
Proof: Partial Covering for Solution, Approximation
Proof: Partial Covering for Solution, Approximation

domain for error bound

covering of neighborhoods has finite subcovering since $x([0, t])$ compact
Proof: Partial Covering for Solution, Approximation

- Domain for error bound
- Covering of neighborhoods has finite subcovering
- Since \( x([0, t]) \) compact
- \( \not\Rightarrow \epsilon \) neighborhoods safe
Proof: Discrete Euler Approximation Axiom

\[ \Delta \quad [x' = f(x)]F \iff \exists h_0 > 0 \forall 0 < h < h_0 [(x := x + hf(x))^*]F \quad \text{(closed)} \]

Proof Sketch.

1. \( \omega \models \exists h_0 > 0 \forall 0 < h < h_0 [(x := x + hf(x))^*]F \quad \hat{x}^n = x \) at iteration \( n \)
2. \( x \in C^2([0, t]) \) solves \( x' = f(x) \) and \( x(0) = \omega \). NTS \( x(t) \models F \)
3. \( f \in C^1 \) locally Lipschitz iff Lipschitz on compact subsets \( \iff \) loc. compact
4. Fix \( E > 0 \). Let \( L \) Lipschitz constant of \( f \in C^1 \) on compact image \( U \) def \( = \bigcup_{q \in x([0, t])} \overline{U}_E(q) \) of \( x([0, t]) \times \overline{U}_E(0) \) under +.
   \[ \|x(nh) - \hat{x}^n\| \leq h \max_{\zeta \in [0, t]} \|x''(\zeta)\| \frac{e^{Lt} - 1}{2L} < \varepsilon < E \quad \text{for small} \quad (h \ll 1) \]
   \[ \|x(t) - x(nh)\| \stackrel{\text{MVT}}{=} \|x'(\xi)\|(t - nh) \leq \max_{\xi \in [0, t]} \|f(x(\xi))\|(t - nh) < \varepsilon \quad (h \ll 1) \]
   \[ \|x(t) - \hat{x}^n\| \leq \|x(t) - x(nh)\| + \|x(nh) - \hat{x}^n\| < 2\varepsilon \quad (h \ll 1) \]
5. A subseq. of \( \hat{x}^n \to x(t) \) as \( h \to 0 \) and \( \hat{x}^n \models F \) closed so \( x(t) \models F \). \( \square \)
Proof: Discrete Euler Approximation Axiom \( \Delta \)

\[ [x' = f(x)] F \rightarrow \forall t \geq 0 \exists h_0 > 0 \forall 0 < h < h_0 \left[ (x := x + hf(x))^* \right] (t \geq 0 \rightarrow F) \] (open)

Proof Sketch.

1. \( \omega \models [x' = f(x)] F \) \quad \( \hat{x}^n = x \) at iteration \( n \)
2. \( x \in C^2([0, t]) \) solves \( x' = f(x) \) and \( x(0) = \omega \). Compact \( x([0, t]) \subseteq F \) open
3. \( 0 < E < \inf_{q \in x([0, t])} d(q, [F]^C) \) has compact \( U \overset{\text{def}}{=} \overline{U}E(x([0, t])) \) in \( F \).
4. Let \( L \) Lipschitz constant of \( f \in C^1 \) on compact \( U \).
   \[
   \| x(nh) - \hat{x}^n \| \leq h \max_{\zeta \in [0, t]} \| x''(\zeta) \| \frac{e^{Lt} - 1}{2L} < \varepsilon < E \quad \text{for small } (h \ll 1)
   \]
   \[
   \| x(t) - x(nh) \| \overset{\text{MVT}}{=} \| x'(\zeta) \| (t - nh) \leq \max_{\zeta \in [0, t]} \| f(x(\zeta)) \| (t - nh) < \varepsilon \quad (h \ll 1)
   \]
   \[
   \| x(t) - \hat{x}^n \| \leq \| x(t) - x(nh) \| + \| x(nh) - \hat{x}^n \| < 2\varepsilon \quad (h \ll 1)
   \]
5. \( \omega \models \exists h_0 > 0 \forall 0 < h < h_0 \left[ (x := x + hf(x))^* \right] (t \geq 0 \rightarrow F) \) for \( h \ll 1, nh \leq t \)
   as \( \hat{x}^n \models F \) for \( h \ll 1, nh \leq t \) by 4a since \( t \geq 0 \) after loop iff \( nh \leq t \) before \( \square \)
Proof: Discrete Euler Approximation Axiom

\[ x' = f(x) \] \( F \iff \forall t \geq 0 \exists \varepsilon > 0 \exists h_0 > 0 \forall 0 < h < h_0 [(x := x + hf(x))^*](t \geq 0 \to \neg \mathcal{U}_\varepsilon(\neg F)) \]

Proof Sketch. (open).

1. “\( \to \)” \( \omega \models [x' = f(x)] F \) ("\( \leftarrow \)" derives from \( \Delta \) as \( \neg \mathcal{U}_\varepsilon(\neg F) \) closed)
2. \( x \in C^2([0, t]) \) solves \( x' = f(x) \) and \( x(0) = \omega \). Compact \( x([0, t]) \subseteq F \) open
3. \( 0 < E < \inf_{q \in x([0, t])} d(q, [F]^C) \) has compact \( U \overset{\text{def}}{=} \overline{\mathcal{U}}_E(x([0, t])) \) in \( F \).
4. \( \omega \models [x' = f(x)](t \geq 0 \to \forall z (\|z - x\| < E \to F(z))) \) by (3)
   \[
   \|x(nh) - \hat{x}^n\| \leq h \max_{\zeta \in [0, t]} \|x''(\zeta)\| \frac{e^{Lt} - 1}{2L} < \varepsilon < E \quad \text{for small} \ (h \ll 1)
   \]
   \[
   \|x(t) - x(nh)\| \overset{\text{MVT}}{=} \|x'(\xi)\| (t - nh) \leq \max_{\xi \in [0, t]} \|f(x(\xi))\| (t - nh) < \varepsilon \quad (h \ll 1)
   \]
   \[
   \|x(t) - \hat{x}^n\| \leq \|x(t) - x(nh)\| + \|x(nh) - \hat{x}^n\| < 2\varepsilon \quad (h \ll 1)
   \]
5. \( \|x(nh) - z\| \leq \|x(nh) - \hat{x}^n\| + \|\hat{x}^n - z\| < 2\varepsilon \leq E \) for \( h \ll 1 \), \( \|\hat{x}^n - z\| < \varepsilon \).
6. \( F(z) \) true at these \( z \) by (4).
7. \( n \)-th iterate \( \omega_n \models t \geq 0 \to \forall z (\|z - x\| < \varepsilon \to F(z)) \) as \( \omega_n \models t \geq 0 \) iff \( \omega \models nh \leq t \)

\( \neg \mathcal{U}_\varepsilon(\neg F) \)
Δ axiom for open $F$, but $F$ may be closed
Discrete Euler Approximation Axiom \( \rightarrow \leftarrow \)

\[ x' = f(x) \]
\( \rightarrow \leftarrow \)
\[ \forall t \geq 0 \exists \varepsilon > 0 \exists h_0 > 0 \forall 0 < h < h_0 [(x := x + hf(x))^*] (t \geq 0 \rightarrow \neg \mathcal{U}_\varepsilon(\neg F)) \]

Example (Incomplete for closed \( F \))

\[ \models x^2 + y^2 \leq 1 \rightarrow [x' = y, y' = -x] x^2 + y^2 \leq 1 \]
\[ \hat{U} \quad [x' = f(x)] F \leftrightarrow \forall \varepsilon > 0 [x' = f(x)] \mathcal{U}_\varepsilon (F) \quad (\iff B,V,G,K) \]
Closed Discrete Completeness (derivable)

\[
\hat{U} \quad [x' = f(x)] F \iff \forall \varepsilon > 0 [x' = f(x)] \mathcal{U}_\varepsilon(F)
\]

( \iff B,V,G,K)

Example (Closed \sim Quantified Open)

\[\models x^2 + y^2 \leq 1 \rightarrow [x' = y, y' = -x] x^2 + y^2 \leq 1\]
Closed Discrete Completeness

\[ \dot{x}' = f(x) \] \iff \forall \varepsilon > 0 [x' = f(x)] \varUpsilon(\varepsilon)(F) \quad (\iff B,V,G,K)

Example (Closed \sim \text{Quantified Open})

\[ \models x^2 + y^2 \leq 1 \rightarrow [x' = y, y' = -x] \forall \varepsilon > 0 x^2 + y^2 < 1 + \varepsilon \]
Closed Discrete Completeness

$\hat{U}[x' = f(x)]F \iff \forall \varepsilon > 0[x' = f(x)]\mathcal{U}_\varepsilon(F)$

(derivable)

Example (Closed $\sim$ Quantified Open)

$\models x^2 + y^2 \leq 1 \rightarrow \forall \varepsilon > 0[x' = y, y' = -x]x^2 + y^2 < 1 + \varepsilon$
$\Delta$ axiom for open/closed $F$, but otherwise?
Example (Locally Closed $\sim\rightarrow$ Open, Closed)

$\models O \land C \rightarrow [x' = y, y' = -x](O \land C)$
Locally Closed Discrete Completeness (derivable)

\[
[] \land [\alpha](O \land C) \leftrightarrow [\alpha]O \land [\alpha]C
\]

( \Leftarrow K)

Example (Locally Closed \simrightarrow Open, Closed)

\[\models O \land C \rightarrow [x' = y, y' = -x](O \land C)\]
Locally Closed Discrete Completeness (derivable)

\[ [] \land [\alpha](O \land C) \leftrightarrow [\alpha]O \land [\alpha]C \]

( \leftrightarrow \text{K})

Example (Locally Closed $\sim$ Open, Closed)

\[ O \land C \rightarrow [x' = y, y' = -x]O \land [x' = y, y' = -x]C \]
$\tilde{U} \quad [x' = f(x)](O \lor C) \iff \forall \varepsilon > 0 [x' = f(x)](O \lor U_{\varepsilon}(C)) \quad (\iff B,V,G,K)$
Semialgebraic Discrete Completeness

\[ \bar{U} \quad [x' = f(x)](O \lor C) \iff \forall \varepsilon > 0 [x' = f(x)](O \lor \exists \varepsilon \in C) \quad (\iff B, V, G, K) \]

Example ((Open \lor Closed) \sim Quantified Open)

\[ \models O \lor C \rightarrow [x' = y, y' = -x](O \lor C) \]
Semialgebraic Discrete Completeness (derivable)

\[ \dot{x}' = f(x)(O \lor C) \iff \forall \varepsilon > 0 \dot{x}' = f(x)(O \lor U_\varepsilon(C)) \quad (\leftarrow \text{B,V,G,K}) \]

Example ((Open \lor Closed) \sim \text{Quantified Open})

\[ \models O \lor C \rightarrow [x' = y, y' = -x](O \lor \forall \varepsilon > 0 U_\varepsilon(C)) \]
Semialgebraic Discrete Completeness (derivable)

\[ x' = f(x)(O \lor C) \iff \forall \varepsilon > 0 [x' = f(x)(O \lor \mathcal{U}_\varepsilon(C))] \quad (\iff B,V,G,K) \]

Example ((Open \lor Closed) \sim Quantified Open)

\[ \models O \lor C \rightarrow [x' = y, y' = -x] \forall \varepsilon > 0 (O \lor \mathcal{U}_\varepsilon(C)) \]
Semialgebraic Discrete Completeness (derivable)

\[ x' = f(x) \] \( (O \lor C) \iff \forall \varepsilon > 0 \ [x' = f(x)] \ (O \lor \mathcal{U}_\varepsilon(C)) \quad (\iff B, V, G, K) \]

Example ((Open \lor Closed) \sim \text{Quantified Open})

\[ \models O \lor C \rightarrow \forall \varepsilon > 0 \ [x' = y, y' = -x] \ (O \lor \mathcal{U}_\varepsilon(C)) \]
Δ axiom for semialgebraic $F$, but otherwise?
Discrete Completeness

Theorem (Relative Completeness / Continuous) (JAR’08, LICS’12)

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

⊢ φ implies Taut_{FOD} ⊨ φ

Theorem (Relative Completeness / Discrete) (LICS’12)

dL calculus ⊕ ∆ is a sound & complete axiomatization of hybrid systems relative to discrete dynamics.

⊢ φ implies Taut_{DL} ⊨ φ

Proof.

1. dL/ODE complete ⇒ suffices ⊨ φ implies Taut_{DL} ⊨ φ for φ ∈ FOD
2. [x’ = f(x)]F for first-order F see previous slides.
3. propositional connectives and quantifiers see schematic completeness.
4. ⊢_{DL} ⟨x’ = f(x)⟩F ↔ (⟨x’ = f(x)⟩F)^♭ see previous slides.
**Theorem (Equi-expressibility) (LICS’12)**

$dL$ (constructively) expressible in $FOD$ and in $DL$:

\[
\forall \phi \exists \phi^b \in FOD \models \phi \leftrightarrow \phi^b \\
\forall \phi \exists \phi^\# \in DL \models \phi \leftrightarrow \phi^\#
\]
Theorem (Equi-expressibility) (LICS’12)

dL (constructively) expressible in FOD and in DL:

\[ \forall \phi \exists \phi^b \in \text{FOD} \models \phi \leftrightarrow \phi^b \]

\[ \forall \phi \exists \phi^# \in \text{DL} \models \phi \leftrightarrow \phi^# \]
Theorem (Equi-expressibility) (LICS’12)

dL (constructively) expressible in FOD and in DL:

\[ \forall \phi \exists \phi^b \in \text{FOD} \models \phi \leftrightarrow \phi^b \]
\[ \forall \phi \exists \phi^# \in \text{DL} \models \phi \leftrightarrow \phi^# \]
Theorem (Equi-expressibility) (LICS’12)

dL (constructively) expressible in FOD and in DL:

\[ \forall \phi \exists \phi^b \in \text{FOD} \models \phi \iff \phi^b \]

\[ \forall \phi \exists \phi^\# \in \text{DL} \models \phi \iff \phi^\# \]

hybrid

discrete

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

dL (constructively) expressible in FOD and in DL:

\[ \forall \phi \exists \phi^b \in FOD \models \phi \iff \phi^b \]

\[ \forall \phi \exists \phi^\# \in DL \models \phi \iff \phi^\# \]

hybrid

discrete

continuous
Equi-expressibility

Theorem (Equi-expressibility) (LICS’12)

dL (constructively) expressible in FOD and in DL:

\[
\forall \phi \exists \phi^b \in \text{FOD} \models \phi \iff \phi^b
\]

\[
\forall \phi \exists \phi^# \in \text{DL} \models \phi \iff \phi^#
\]
Equi-expressibility

Theorem (Equi-expressibility) (LICS'12)

dL (constructively) expressible in FOD and in DL:

\[ \forall \phi \exists \phi^b \in FOD \models \phi \leftrightarrow \phi^b \]

\[ \forall \phi \exists \phi^\# \in DL \models \phi \leftrightarrow \phi^\# \]
Theorem (Equi-expressibility) (LICS’12)

dL (constructively) expressible in FOD and in DL:

$$\forall \phi \exists \phi^b \in FOD \models \phi \leftrightarrow \phi^b$$
$$\forall \phi \exists \phi^\# \in DL \models \phi \leftrightarrow \phi^\#$$
Theorem (Relative Decidability) (LICS’12)

Validity of $dL$ sentences is decidable relative to FOD or relative to DL.

Proof.

1. Let $\phi$ a sentence in $dL$ ($\text{FV}(\phi) = \emptyset$) and $\omega$ a state.
2. Either $\omega \models \phi$ or $\omega \not\models \phi$. So either $\omega \models \phi$ or $\omega \models \neg \phi$.
3. By coincidence, $\omega \models \phi$ iff $\nu \models \phi$ for arbitrary $\nu$, as $\text{FV}(\phi)$, no symbols.
4. Either $\vdash \phi$ or $\vdash \neg \phi$.
5. Either $\vdash_L \phi$ or $\vdash_L \neg \phi$ by completeness relative to $L = \text{FOD}$, $L = \text{DL}$. \hfill $\square$
Outline

1. Hybrid Systems
2. Differential Dynamic Logic
   - Syntax
   - Semantics
   - Axiomatization
3. Continuous Completeness
   - Schematic Completeness
   - Expressibility and Rendition of Hybrid Programs
4. Discrete Completeness
   - Open Discrete Completeness
   - Closed Discrete Completeness
   - Semialgebraic Discrete Completeness of $\mathcal{dL} + \Delta$
   - Discrete Completeness of $\mathcal{dL} + \Delta$
   - Equi-expressible
   - Relative Decidable
5. Summary
Complete Proof Theory of Hybrid Systems

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

proof-theoretical alignment
continuous = hybrid = discrete

System

Hybrid
Continuous
Discrete
Complete Proof Theory of Hybrid Systems

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

proof-theoretical alignment
continuous = hybrid = discrete

System

Continuous

Hybrid

Discrete
André Platzer.
The complete proof theory of hybrid systems.
In LICS [6], pages 541–550.

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.
Logics of dynamical systems.
In LICS [6], pages 13–24.