# d$\mathcal{L}_\iota$: Definite Descriptions in Differential Dynamic Logic

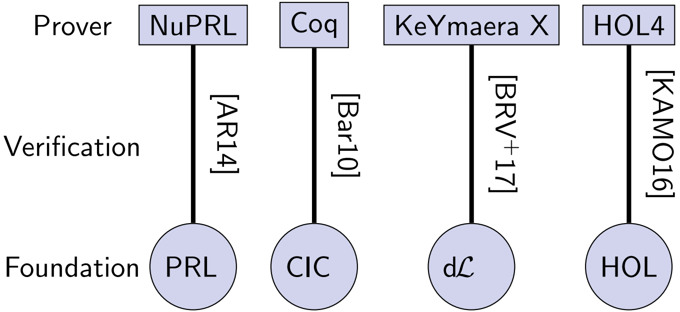
**Brandon Bohrer**, Manuel Fernández, and André Platzer

Logical Systems Lab
Computer Science Department
Carnegie Mellon University

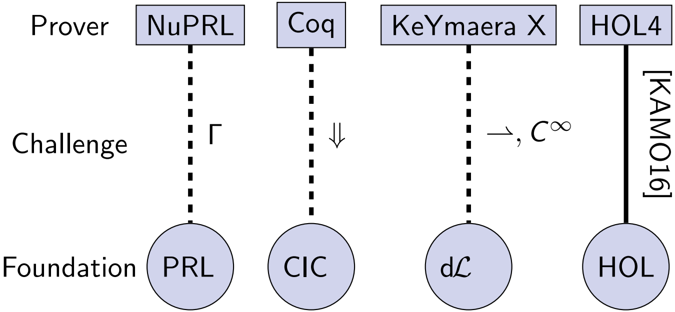
CADE-27
August 29 2019

# Outline

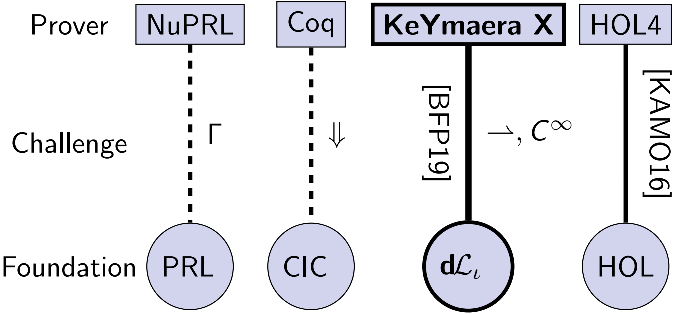
# We Can Trust Theorem Provers



# We Can *Almost* Trust Theorem Provers



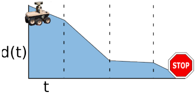
# We Help d$\mathcal{L}$ Foundation Catch Up



# Safety-Critical CPS Deserve Proofs

Planes

Drones



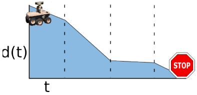

Robots

*How can we design cyber-physical systems people can bet their lives on? – Jeanette Wing*

# d$\mathcal{L}$ + KeYmaera X Provides Proofs

Planes

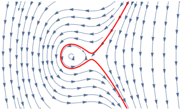
Drones




Robots

```
SQR_LOOP bcf     STATUS,C
         rrf     X,f
  ; Task 3B: IF Carry == 1
         btfss   STATUS,C
          goto   SQR_CONT
         movf    X_COPY_L,w
         addwf   SQUARE+1,f
         btfsc   STATUS,C
          incf   SQUARE,f
         movf    X_COPY_H,w
         addwf   SQUARE,f
```

Discrete Control



Continuous Dynamics

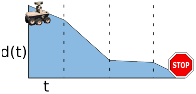
$$\frac{\Gamma \vdash A}{\Gamma \vdash A \wedge B}$$

Syntactic Proof

# d$\mathcal{L}$ + KeYmaera X Provides Proofs

Planes

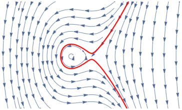
Drones




Robots

```
SQR_LOOP bcf     STATUS,C
         rrf     X,f
   ; Task 3B: IF Carry == 1
         btfss   STATUS,C
          goto   SQR_CONT
         movf    X_COPY_L,w
         addwf   SQUARE+1,f
         btfsc   STATUS,C
          incf   SQUARE,f
         movf    X_COPY_H,w
         addwf   SQUARE,f
```

Discrete Control



Continuous Dynamics

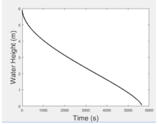
$$\frac{\Gamma \vdash A}{\Gamma \vdash A \wedge B}$$

Syntactic Proof

*How do proofs cope when control, dynamics are partial, discontinuous?*

# Outline

# Example System: Robot Water Cooler

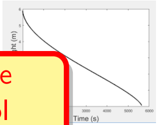



$$\alpha_B \equiv \Big\{\{\{?h > 0;\ a := 1\} \ \cup \ a := 0\};$$
$$h' = -\sqrt{2gh}\frac{a}{A} \ \&\ h \geq 0\Big\}^*$$

## Proposition (Leakiness)

$$g > 0 \land h = h_0 \land h_0 > 0 \land A > 0 \rightarrow [\alpha_B](h \leq h_0)$$

# Example System: Robot Water Cooler



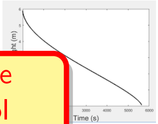

Choose control case

$$\alpha_B \equiv \Big\{ \{ \{?h > 0;\ a := 1\} \ \cup\ a := 0 \};$$
$$h' = -\sqrt{2gh}\frac{a}{A} \,\&\, h \geq 0 \Big\}^*$$

**Proposition (Leakiness)**

$$g > 0 \land h = h_0 \land h_0 > 0 \land A > 0 \rightarrow [\alpha_B](h \leq h_0)$$

# Example System: Robot Water Cooler




Test
$h > 0$

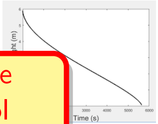
Choose
control
case

$$\alpha_B \equiv \Big\{\{\{?h > 0;\ a := 1\} \ \cup \ a := 0\};$$
$$h' = -\sqrt{2gh}\frac{a}{A} \ \&\ h \geq 0\Big\}^*$$

## Proposition (Leakiness)

$$g > 0 \land h = h_0 \land h_0 > 0 \land A > 0 \rightarrow [\alpha_B](h \leq h_0)$$

# Example System: Robot Water Cooler




Test $h > 0$

Set $a$ to 1

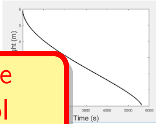
Choose control case

$$\alpha_B \equiv \Big\{\{\{?h > 0;\ a := 1\} \cup a := 0\};$$
$$h' = -\sqrt{2gh}\frac{a}{A} \,\&\, h \geq 0\Big\}^*$$

## Proposition (Leakiness)

$$g > 0 \land h = h_0 \land h_0 > 0 \land A > 0 \rightarrow [\alpha_B](h \leq h_0)$$

# Example System: Robot Water Cooler




Test $h > 0$

Set $a$ to 1

Choose control case

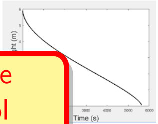
Evolve physics

$$\alpha_B \equiv \Big\{\{\{?h > 0;\ a := 1\} \cup a := 0\};\ h' = -\sqrt{2gh}\,\frac{a}{A} \,\&\, h \geq 0\Big\}^*$$

## Proposition (Leakiness)

$$g > 0 \land h = h_0 \land h_0 > 0 \land A > 0 \rightarrow [\alpha_B](h \leq h_0)$$

# Example System: Robot Water Cooler




Test $h > 0$

Set $a$ to 1

Choose control case

Evolve physics

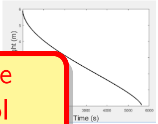
$$\alpha_B \equiv \Big\{\{\{?h > 0;\ a := 1\} \cup a := 0\};$$
$$h' = -\sqrt{2gh}\frac{a}{A} \,\&\, h \geq 0\Big\}^*$$

Proposition (Leakiness)

F.O. Arithmetic

$$g > 0 \land h = h_0 \land h_0 > 0 \land A > 0 \rightarrow [\alpha_B](h \leq h_0)$$

# Example System: Robot Water Cooler




Test $h > 0$

Set $a$ to 1

Choose control case

$$\alpha_B \equiv \Big\{\{\{?h > 0;\ a := 1\} \ \cup\ a := 0\};$$

Evolve physics

$$h' = -\sqrt{2gh}\frac{a}{A} \,\&\, h \geq 0\Big\}^*$$

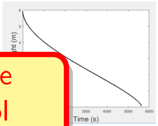
Proposition (Le

Conjunction

F.O. Arithmetic

$$g > 0 \wedge h = h_0 \wedge h_0 > 0 \wedge A > 0 \rightarrow [\alpha_B](h \leq h_0)$$

# Example System: Robot Water Cooler




Test $h > 0$

Set $a$ to 1

Choose control case

$$\alpha_B \equiv \Big\{\{\{?h > 0;\ a := 1\} \cup a := 0\};$$

Evolve physics

$$h' = -\sqrt{2gh}\frac{a}{A} \ \&\ h \geq 0\Big\}^*$$

Proposition (Le

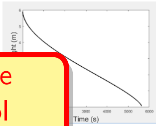
Conjunction

Implication

F.O. Arithmetic

$$g > 0 \land h = h_0 \land h_0 > 0 \land A > 0 \rightarrow [\alpha_B](h \leq h_0)$$

# Example System: Robot Water Cooler




Test $h > 0$

Set $a$ to 1

Choose control case

$$\alpha_B \equiv \Big\{\{\{?h > 0;\ a := 1\} \cup a := 0\};$$

Evolve physics

$$h' = -\sqrt{2gh}\frac{a}{A}\ \&\ h \geq 0\Big\}^*$$

Proposition (Le

Conjunction

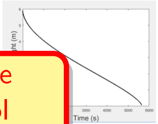
Implication

All runs

F.O. Arithmetic

$$g > 0 \land h = h_0 \land h_0 > 0 \land A > 0 \rightarrow [\alpha_B](h \leq h_0)$$

# Example System: Robot Water Cooler




Test $h > 0$

Set $a$ to 1

Choose control case

$$\alpha_B \equiv \Big\{\{\{?h > 0;\ a := 1\} \cup a := 0\};$$

Evolve physics

$$h' = -\sqrt{2gh}\frac{a}{A}\ \&\ h \geq 0\Big\}^*$$

Proposition (Le

Conjunction

Implication

All runs

F.O. Arithmetic

$$g > 0 \land h = h_0 \land h_0 > 0 \land A > 0 \rightarrow [\alpha_B](h \leq h_0)$$

# Outline

# d$\mathcal{L}$ Needs Lots of Extensions

Definition (d$\mathcal{L}$ Terms)

$$\theta, \eta ::= x \mid q \mid \theta + \eta \mid \theta \cdot \eta \mid (\theta)'$$

# d$\mathcal{L}$ Needs Lots of Extensions

Definition (d$\mathcal{L}$ Terms)

$$\theta, \eta ::= x \mid q \mid \theta + \eta \mid \theta \cdot \eta \mid (\theta)'$$
$$\mid \theta/\eta$$

# d$\mathcal{L}$ Needs Lots of Extensions

Definition (d$\mathcal{L}$ Terms)

$$\theta, \eta ::= x \mid q \mid \theta + \eta \mid \theta \cdot \eta \mid (\theta)' \\ \mid \theta/\eta \mid \sqrt{\theta}$$

# d$\mathcal{L}$ Needs Lots of Extensions

Definition (d$\mathcal{L}$ Terms)

$$\theta, \eta ::= x \mid q \mid \theta + \eta \mid \theta \cdot \eta \mid (\theta)'$$
$$\mid \theta/\eta \mid \sqrt{\theta} \mid \mathsf{max}(\theta, \mathit{eta}) \mid \mathsf{min}(\theta, \eta) \mid |\theta| \mid (\mathsf{if}(\phi)(\theta)\mathsf{else}(\eta))$$

# d$\mathcal{L}$ Needs Lots of Extensions

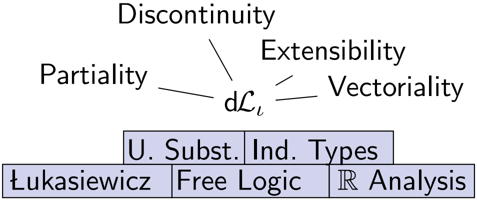
Definition (d$\mathcal{L}$ Terms)

$$\begin{aligned}\theta, \eta ::= {} & x \mid q \mid \theta + \eta \mid \theta \cdot \eta \mid (\theta)' \\ & \mid \theta/\eta \mid \sqrt{\theta} \mid \mathsf{max}(\theta, eta) \mid \mathsf{min}(\theta, \eta) \mid |\theta| \mid (\mathsf{if}(\phi)(\theta)\mathsf{else}(\eta)) \\ & \mid \sin(\theta) \mid \cos(\theta) \mid (\theta, \eta) \mid \pi_1\theta \mid \pi_2\theta \mid \mathsf{in}\mathbb{R}(\theta) \mid \mathsf{isT}(\theta) \\ & \mid \mathsf{map2}(T, f(x, y)) \mid \mathsf{zip}(L_1, L_2) \mid \left(L_1 \vec{+} L_2\right) \mid L_1 \vec{\cdot} L_2\end{aligned}$$

# d$\mathcal{L}_\iota$ Generalizes Foundations

## Definition (d$\mathcal{L}_\iota$ Terms)

$$\theta, \eta ::= \cdots \mid (\theta, \eta) \mid \iota x\, \phi(x)$$




Examples:

$$(\text{if}(\phi)(\theta_1)\text{else}(\theta_2)) = \iota x\, (\phi \land x{=}\theta_1) \lor (\neg\phi \land x{=}\theta_2)$$

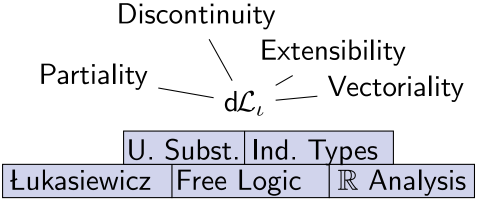
$$\sqrt{\theta} = \iota x\, (x^2{=}\theta \land x \geq 0) \quad \theta_1/\theta_2 = \iota x\, (x \cdot \theta_2{=}\theta_1)$$

# $d\mathcal{L}_\iota$ Generalizes Foundations

**Definition ($d\mathcal{L}_\iota$ Terms)**

Pairing

$$\theta, \eta ::= \cdots \mid (\theta, \eta) \mid \iota x\, \phi(x)$$




Examples:

$$(\text{if}(\phi)(\theta_1)\text{else}(\theta_2)) = \iota x\, (\phi \wedge x{=}\theta_1) \vee (\neg\phi \wedge x{=}\theta_2)$$

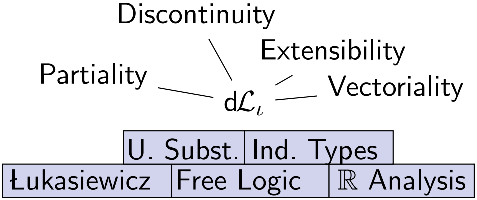
$$\sqrt{\theta} = \iota x\, (x^2{=}\theta \wedge x \geq 0) \qquad \theta_1/\theta_2 = \iota x\, (x \cdot \theta_2{=}\theta_1)$$

# d$\mathcal{L}_\iota$ Generalizes Foundations

Definition (d$\mathcal{L}_\iota$ Terms)

Pairing | Unique $x$ s.t. $\phi$

$$\theta, \eta ::= \cdots \mid (\theta, \eta) \mid \iota x \, \phi(x)$$

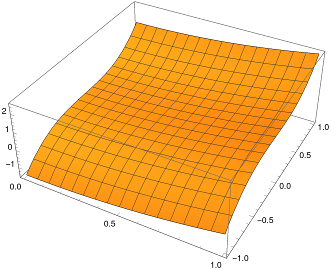



Examples:

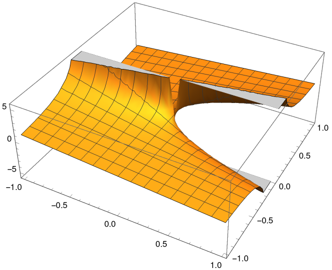
$$(\text{if}(\phi)(\theta_1)\text{else}(\theta_2)) = \iota x \, (\phi \land x{=}\theta_1) \lor (\neg\phi \land x{=}\theta_2)$$

$$\sqrt{\theta} = \iota x \, (x^2{=}\theta \land x \geq 0) \quad \theta_1/\theta_2 = \iota x \, (x \cdot \theta_2{=}\theta_1)$$

# Term Semantics

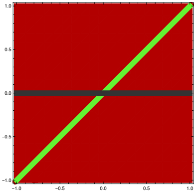



$d\mathcal{L}$



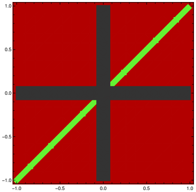

$d\mathcal{L}_\iota$

Compare



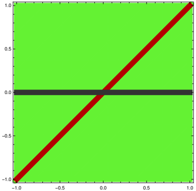
$x/y = 1$

And



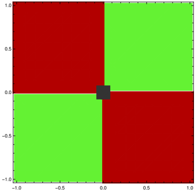
$x/y \geq 1 \wedge y/x \geq 1$

Not

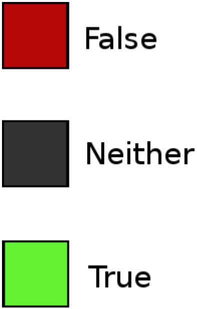


$\neg(x/y = 1)$

Or



$x/y \geq 1 \vee y/x \geq 1$



# Outline

# Program Axioms Decompose Dynamics

$$[:=] \quad ([x := f]p(x) \leftrightarrow p(f))$$

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

$$\langle\cup\rangle \quad \langle a \cup b\rangle P \leftrightarrow (\langle a\rangle P \vee \langle b\rangle P)$$

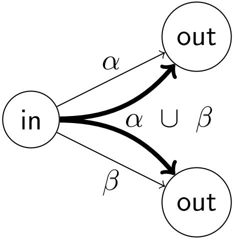



Figure: Selected Program Axioms ($d\mathcal{L}_\iota$)

# Program Axioms Decompose Dynamics

$$[:=] \quad ([x := f]p(x) \leftrightarrow p(f)) \leftarrow \mathrm{E}(f)$$

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

$$\langle \cup \rangle \quad \langle a \cup b \rangle P \leftrightarrow (\langle a \rangle P \vee \langle b \rangle P)$$

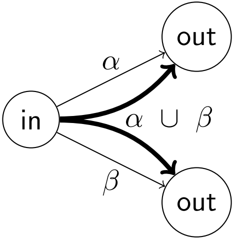



Figure: Selected Program Axioms ($\mathrm{d}\mathcal{L}_\iota$)

# Program Axioms Decompose Dynamics

$$[:=] \quad ([x := f]p(x) \leftrightarrow p(f)) \leftarrow \mathrm{E}(f)$$

Denotes

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

$$\langle\cup\rangle \quad \langle a \cup b\rangle P \leftrightarrow (\langle a\rangle P \vee \langle b\rangle P)$$

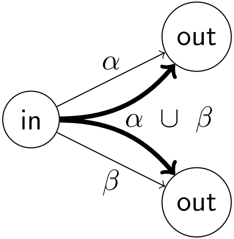



Figure: Selected Program Axioms ($d\mathcal{L}_\iota$)

# Program Axioms Decompose Dynamics

$$[:=] \quad ([x := f]p(x) \leftrightarrow p(f)) \leftarrow \mathrm{E}(f)$$

Denotes

$$[?] \quad [?Q]P \leftrightarrow (\mathrm{D}(Q) \to P)$$

Definitely true

$$\langle \cup \rangle \quad \langle a \cup b \rangle P \leftrightarrow (\langle a \rangle P \vee \langle b \rangle P)$$

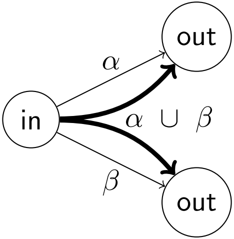



Figure: Selected Program Axioms ($\mathrm{d}\mathcal{L}_\iota$)

# Outline

## U. Subst is Clean Foundation

Axioms are single formulas, substitution is *explicit*:

$$\text{US} \quad \frac{\phi}{\sigma(\phi)}$$

Sound for *admissible* $\sigma$:

### Definition (Admissibility (d$\mathcal{L}$))

No new free variable ref. under **formula, program** binders

### Definition (Admissibility (d$\mathcal{L}_\iota$))

No new free variable ref. under **formula, program, term** binders

**Takeaway:** Admissibility generalizes cleanly to definite description

# Axiom Validity

**Proposition (Non-conservative extension)**
*Formula* $x \cdot x \geq 0$ *is valid in* $\mathrm{d}\mathcal{L}$ *but not* $\mathrm{d}\mathcal{L}_\iota$

**Proposition (Converse reducibility)**
*Exists linear-time* $T(\phi) : \mathrm{d}\mathcal{L} \to \mathrm{d}\mathcal{L}_\iota$ *where* $T(\phi)$ *valid iff* $\phi$ *valid.*

- Non-conservative implies soundness must be proved anew in $\mathrm{d}\mathcal{L}_\iota$ (but we proved it).
- $\mathrm{d}\mathcal{L}_\iota$ axioms are single formulas, so each case of soundness only needs to show validity of one single formula.
- Converse reducibility shows $\mathrm{d}\mathcal{L}_\iota$ supports all $\mathrm{d}\mathcal{L}$ theorems in theory and practice.

# Forward Reducibility

**Motivation:** What is the expressive power of d$\mathcal{L}_\iota$?

Theorem (Forward reducibility)

*Exists reduction* $T(\phi)$ : d$\mathcal{L}_\iota \to$ d$\mathcal{L}$

$$T(x' = \theta \,\&\, \phi) \rightsquigarrow sol(t) \wedge axioms_{sol}$$
$$T((x, y)) \rightsquigarrow \text{Gödel}_{\mathbb{R}}(x, y)$$
$$T(f(x)) \rightsquigarrow \text{Gödel}_{\mathbb{R}}(f(x))$$

# Forward Reducibility

**Motivation:** What is the expressive power of $\mathrm{d}\mathcal{L}_\iota$?

Theorem (Forward reducibility)

*Exists reduction* $T(\phi) : \mathrm{d}\mathcal{L}_\iota \to \mathrm{d}\mathcal{L}$

$$\begin{aligned} T(x' = \theta \,\&\, \phi) &\rightsquigarrow sol(t) \land axioms_{sol} \\ T((x, y)) &\rightsquigarrow \text{Gödel}_{\mathbb{R}}(x, y) \\ T(f(x)) &\rightsquigarrow \text{Gödel}_{\mathbb{R}}(f(x)) \end{aligned}$$

**Implication:** Reduction is hard, want $\mathrm{d}\mathcal{L}_\iota$ in practice.

# Takeaways

- d$\mathcal{L}_\iota$ (definite description) helped d$\mathcal{L}$ foundation catch up with *KeYmaera X* implementation.
- Theory is now ahead of implementation (vectors, function definitions, non-polynomial ODEs)
- Uniform substitution calculus generalizes smoothly to many logics

# References I


Abhishek Anand and Vincent Rahli, *Towards a formally verified proof assistant*, ITP (Gerwin Klein and Ruben Gamboa, eds.), LNCS, vol. 8558, Springer, 2014, pp. 27–44.

Bruno Barras, *Sets in Coq, Coq in sets*, J. Formalized Reasoning **3** (2010), no. 1, 29–48.

Brandon Bohrer, Manuel Fernandez, and André Platzer, *$dL_\iota$: Definite descriptions in differential dynamic logic*, CADE (Pascal Fontaine, ed.), LNCS, vol. 11716, Springer, 2019, pp. 94–110.

# References II


Brandon Bohrer, Vincent Rahli, Ivana Vukotic, Marcus Völp, and André Platzer, *Formally verified differential dynamic logic*, Certified Programs and Proofs - 6th ACM SIGPLAN Conference, CPP 2017, Paris, France, January 16-17, 2017 (New York) (Yves Bertot and Viktor Vafeiadis, eds.), ACM, 2017, pp. 208–221.

Ramana Kumar, Rob Arthan, Magnus O. Myreen, and Scott Owens, *Self-formalisation of higher-order logic: Semantics, soundness, and a verified implementation*, J. Autom. Reas. **56** (2016), no. 3, 221–259.