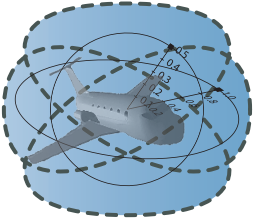
22: Axioms & Uniform Substitutions

15-424: Foundations of Cyber-Physical Systems

André Platzer

aplatzer@cs.cmu.edu
Computer Science Department
Carnegie Mellon University, Pittsburgh, PA

The Secret for Simpler Sound Hybrid Systems Provers



Outline

Outline

Can you trust a computer to control physics?

Can you trust a computer to control physics?

Rationale

1. Safety guarantees require analytic foundations.
2. Foundations revolutionized digital computer science & our society.
3. Need even stronger foundations when software reaches out into our physical world.

How can we provide people with cyber-physical systems they can bet their lives on? — Jeannette Wing

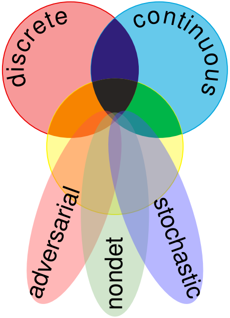
Cyber-physical Systems

CPS combine cyber capabilities with physical capabilities to solve problems that neither part could solve alone.

CPSs are Multi-Dynamical Systems

CPS Dynamics

CPS are characterized by multiple facets of dynamical systems.



CPS Compositions

CPS combine multiple simple dynamical effects.

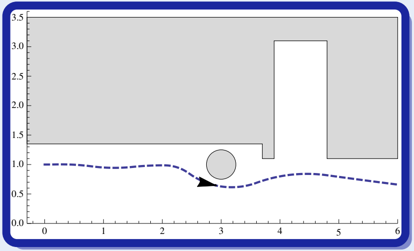
Tame Parts

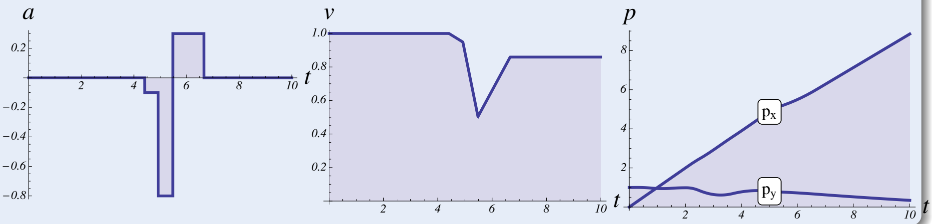
Exploiting compositionality tames CPS complexity.

Challenge (CPS)

Fixed rule describing state evolution with both

- Discrete dynamics (control decisions)
- Continuous dynamics (differential equations)



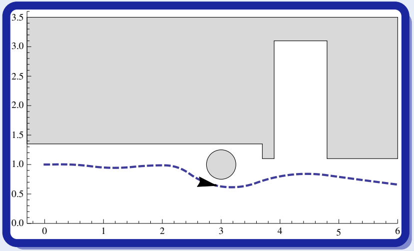


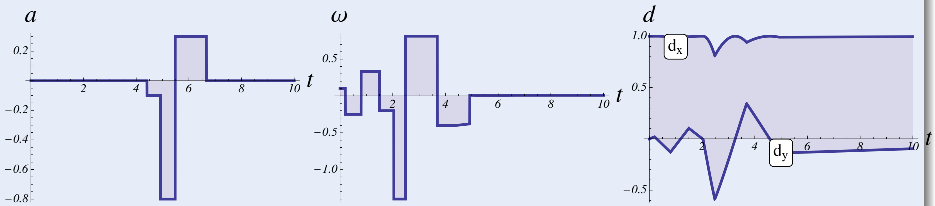
CPS Analysis

Challenge (CPS)

Fixed rule describing state evolution with both

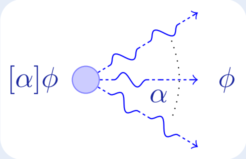
- Discrete dynamics (control decisions)
- Continuous dynamics (differential equations)





CPS Analysis

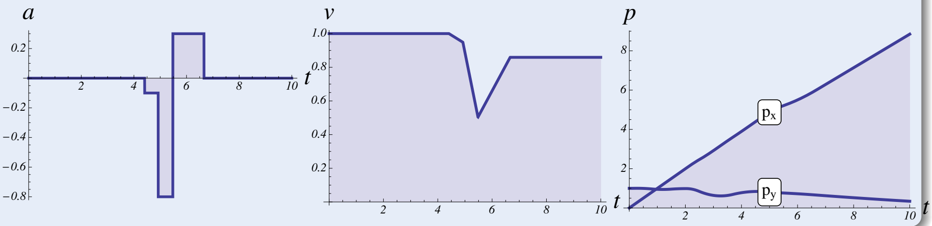
Differential Dynamic Logic



Seq. Compose

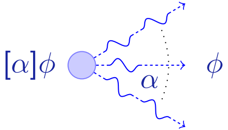
Nondet. Repeat

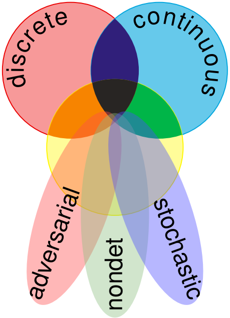
$$\underbrace{x \neq o \land b > 0}_{\text{init}} \rightarrow \Big[\Big(\underbrace{\mathtt{if}(\mathrm{tooClose}(x,o))\ a := -b}_{\text{discrete control}}\ ;\ \underbrace{x' = v, v' = a}_{\text{ODE}}\Big)^*\Big]\underbrace{x \neq o}_{\text{post}}$$



differential dynamic logic

$\mathsf{d}\mathcal{L} = \mathsf{DL} + \mathsf{HP}$



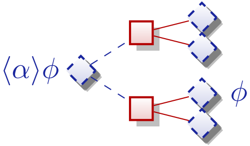


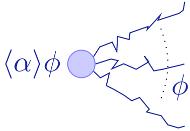
differential game logic

$\mathsf{dG}\mathcal{L} = \mathsf{GL} + \mathsf{HG}$

stochastic differential DL

$\mathsf{Sd}\mathcal{L} = \mathsf{DL} + \mathsf{SHP}$





quantified differential DL

$\mathsf{Qd}\mathcal{L} = \mathsf{FOL} + \mathsf{DL} + \mathsf{QHP}$

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

LICS'12,CADE'15,TOCL'15

Key Contributions

Q: How to build a prover with a small soundness-critical core?
A: Uniform substitution [Church]

Q: How to enable flexible yet sound reasoning?
A: Axioms with local meaning [Philosophy, Algebraic Geometry]

Q: What's the local meaning of a differential equation?
A: Differential forms [Differential Geometry]

Q: How to do hybrid systems proving?
A: Uniform substitution calculus for differential dynamic logic

Q: What's the impact of uniform substitution on a prover core?
A: $65\,989 \searrow 1\,677$ LOC (2.5%) [KeYmaera X]

Outline

Differential Dynamic Logic: Axiomatization

$[:=]$	$[x := \theta]\phi(x) \leftrightarrow \phi(\theta)$	(θ free for x in ϕ)
$[?]$	$[?\chi]\phi \leftrightarrow (\chi \rightarrow \phi)$	
$[\cup]$	$[\alpha \cup \beta]\phi \leftrightarrow [\alpha]\phi \wedge [\beta]\phi$	
$[;]$	$[\alpha; \beta]\phi \leftrightarrow [\alpha][\beta]\phi$	
$[^*]$	$[\alpha^*]\phi \leftrightarrow \phi \wedge [\alpha][\alpha^*]\phi$	
K	$[\alpha](\phi \rightarrow \psi) \rightarrow ([\alpha]\phi \rightarrow [\alpha]\psi)$	
I	$[\alpha^*](\phi \rightarrow [\alpha]\phi) \rightarrow (\phi \rightarrow [\alpha^*]\phi)$	
V	$\phi \rightarrow [\alpha]\phi$	$(FV(\phi) \cap BV(\alpha) = \emptyset)$
$[']$	$[x' = \theta]\phi \leftrightarrow \forall t{\geq}0\, [x := x(t)]\phi$	(t fresh and $x'(t) = \theta$)

LICS'12

Differential Dynamic Logic: Axioms

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

[?] $[?q]p \leftrightarrow (q \rightarrow p)$

[∪] $[a \cup b]p(\bar{x}) \leftrightarrow [a]p(\bar{x}) \land [b]p(\bar{x})$

[;] $[a; b]p(\bar{x}) \leftrightarrow [a][b]p(\bar{x})$

[*] $[a^*]p(\bar{x}) \leftrightarrow p(\bar{x}) \land [a][a^*]p(\bar{x})$

K $[a](p(\bar{x}) \rightarrow q(\bar{x})) \rightarrow ([a]p(\bar{x}) \rightarrow [a]q(\bar{x}))$

I $[a^*](p(\bar{x}) \rightarrow [a]p(\bar{x})) \rightarrow (p(\bar{x}) \rightarrow [a^*]p(\bar{x}))$

V $p \rightarrow [a]p$

CADE'15

Differential Dynamic Logic: Comparison

$[x := f]p(x) \leftrightarrow p(f)$	$[:=]$	$[x := \theta]\phi(x) \leftrightarrow \phi(\theta)$
$[?q]p \leftrightarrow (q \to p)$	$[?]$	$[?\chi]\phi \leftrightarrow (\chi \to \phi)$
$[a \cup b]p(\bar{x}) \leftrightarrow [a]p(\bar{x}) \land [b]p(\bar{x})$	$[\cup]$	$[\alpha \cup \beta]\phi \leftrightarrow [\alpha]\phi \land [\beta]\phi$
$[a; b]p(\bar{x}) \leftrightarrow [a][b]p(\bar{x})$	$[;]$	$[\alpha; \beta]\phi \leftrightarrow [\alpha][\beta]\phi$
$[a^*]p(\bar{x}) \leftrightarrow p(\bar{x}) \land [a][a^*]p(\bar{x})$	$[^*]$	$[\alpha^*]\phi \leftrightarrow \phi \land [\alpha][\alpha^*]\phi$
$[a](p(\bar{x}) \to q(\bar{x})) \to ([a]p(\bar{x}) \to [a]q(\bar{x}))$	K	$[\alpha](\phi \to \psi) \to ([\alpha]\phi \to [\alpha]\psi)$
$[a^*](p(\bar{x}) \to [a]p(\bar{x})) \to (p(\bar{x}) \to [a^*]p(\bar{x}))$	I	$[\alpha^*](\phi \to [\alpha]\phi) \to (\phi \to [\alpha^*]\phi)$
$p \to [a]p$	V	$\phi \to [\alpha]\phi$
	$[']$	$[x' = \theta]\phi \leftrightarrow \forall t \geq 0\, [x := x(t)]\phi$
CADE'15		LICS'12

Differential Dynamic Logic: Comparison

$[x := f]p(x) \leftrightarrow p(f)$	$[:=]$	$[x := \theta]\phi(x) \leftrightarrow \phi(\theta)$
$[?q]p \leftrightarrow (q \rightarrow p)$	$[?]$	$[?\chi]\phi \leftrightarrow (\chi \rightarrow \phi)$
Axiom: $[a \cup b]p(\bar{x}) \leftrightarrow [a]p(\bar{x}) \wedge [b]p(\bar{x})$	$[\cup]$	Schema: $[\alpha \cup \beta]\phi \leftrightarrow [\alpha]\phi \wedge [\beta]\phi$
$[a;b]p(\bar{x}) \leftrightarrow [a][b]p(\bar{x})$	$[;]$	$[\alpha;\beta]\phi \leftrightarrow [\alpha][\beta]\phi$
$[a^*]p(\bar{x}) \leftrightarrow p(\bar{x}) \wedge [a][a^*]p(\bar{x})$	$[^*]$	$[\alpha^*]\phi \leftrightarrow \phi \wedge [\alpha][\alpha^*]\phi$
$[a](p(\bar{x}) \rightarrow q(\bar{x})) \rightarrow ([a]p(\bar{x}) \rightarrow [a]q(\bar{x}))$	K	$[\alpha](\phi \rightarrow \psi) \rightarrow ([\alpha]\phi \rightarrow [\alpha]\psi)$
$[a^*](p(\bar{x}) \rightarrow [a]p(\bar{x})) \rightarrow (p(\bar{x}) \rightarrow [a^*]p(\bar{x}))$	I	$[\alpha^*](\phi \rightarrow [\alpha]\phi) \rightarrow (\phi \rightarrow [\alpha^*]\phi)$
Axiom: $p \rightarrow [a]p$	V	Schema: $\phi \rightarrow [\alpha]\phi$
	$[']$	$[x' = \theta]\phi \leftrightarrow \forall t \geq 0\, [x := x(t)]\phi$
CADE'15		LICS'12

Axiom vs. Axiom Schema

Axiom

$[a \cup b]p(\bar{x}) \leftrightarrow [a]p(\bar{x}) \land [b]p(\bar{x})$

Schema

$[\alpha \cup \beta]\phi \leftrightarrow [\alpha]\phi \land [\beta]\phi$

Axiom

$p \rightarrow [a]p$

Schema

$\phi \rightarrow [\alpha]\phi \quad \ldots$

Axiom vs. Axiom Schema

Axiom

$[a \cup b]p(\bar{x}) \leftrightarrow [a]p(\bar{x}) \land [b]p(\bar{x})$

Schema

$[\alpha \cup \beta]\phi \leftrightarrow [\alpha]\phi \land [\beta]\phi$

Pattern match formulas for shape $\alpha \cup \beta$

Placeholder α schema variable matcher

Same instance of ϕ in all places

Axiom

$p \to [a]p$

Schema

$\phi \to [\alpha]\phi \quad \dots$

Axiom vs. Axiom Schema

Axiom

$[a \cup b]p(\bar{x}) \leftrightarrow [a]p(\bar{x}) \land [b]p(\bar{x})$

Schema

$[\alpha \cup \beta]\phi \leftrightarrow [\alpha]\phi \land [\beta]\phi$

Pattern match formulas for shape $\alpha \cup \beta$

Placeholder α schema variable matcher

Same instance of ϕ in all places

Axiom

$p \rightarrow [a]p$

Schema

$\phi \rightarrow [\alpha]\phi \quad \dots$

- $x = 0 \rightarrow [y' = 5]x = 0$
- $x = y \rightarrow [y' = 5]x = y$
- $x = z \rightarrow [y' = 5]x = z$

Axiom vs. Axiom Schema

Axiom

$[a \cup b]p(\bar{x}) \leftrightarrow [a]p(\bar{x}) \land [b]p(\bar{x})$

Schema

$[\alpha \cup \beta]\phi \leftrightarrow [\alpha]\phi \land [\beta]\phi$

Pattern match formulas for shape $\alpha \cup \beta$

Placeholder α schema variable matcher

Same instance of ϕ in all places

Axiom

$p \to [a]p$

Schema

$\phi \to [\alpha]\phi \quad \dots$

special vs. degenerate instances

✓ $x = 0 \to [y' = 5]x = 0$

× $x = y \to [y' = 5]x = y$

✓ $x = z \to [y' = 5]x = z$

rule out by side conditions

Axiom vs. Axiom Schema: Formula vs. Algorithm

1 Formula

Axiom

$[a \cup b]p(\bar{x}) \leftrightarrow [a]p(\bar{x}) \wedge [b]p(\bar{x})$

Generic formula.
No exceptions.

Algorithm

Schema

$[\alpha \cup \beta]\phi \leftrightarrow [\alpha]\phi \wedge [\beta]\phi$

Pattern match formulas for shape $\alpha \cup \beta$

Placeholder α schema variable matcher

Same instance of ϕ in all places

Axiom

$p \rightarrow [a]p$

Schema

$\phi \rightarrow [\alpha]\phi \quad \ldots$

special vs. degenerate instances

✓ $x = 0 \rightarrow [y' = 5]x = 0$

× $x = y \rightarrow [y' = 5]x = y$

✓ $x = z \rightarrow [y' = 5]x = z$

rule out by side conditions

Generic Formulas in Axioms are like Generic Points

An analogy from algebraic geometry

Axiom schemata with side conditions are like concrete points

$\exists x\, ax^2 + bx + c = 0$ iff $b^2 \geq 4ac$ except $a = 0$

This Way

Generic Formulas in Axioms are like Generic Points

An analogy from algebraic geometry

Axiom schemata with side conditions are like concrete points

$\exists x\, ax^2 + bx + c = 0$ iff $b^2 \geq 4ac$ except $a = 0$ except $b = 0$

This Way

Generic Formulas in Axioms are like Generic Points

An analogy from algebraic geometry

Axiom schemata | with side conditions are like | concrete points

$\exists x\, ax^2 + bx + c = 0$ iff $b^2 \geq 4ac$ except $a = 0$ except $b = 0$ except $c = 0$

This Way

An analogy from algebraic geometry

Axiom schemata with side conditions are like concrete points

$\exists x\, ax^2 + bx + c = 0$ iff $b^2 \geq 4ac$ except $a = 0$ except $b = 0$ except $c = 0$

This Way

Axioms Generic formulas in axioms are like generic points

$ax^2 + bx + c = 0$ iff $x = -b \pm \sqrt{b^2 - 4ac}/(2a)$

Paying attention during substitutions to avoid degenerates (no $/0$, $\sqrt{-1}$)

Axioms vs. Axiom Schemata: Philosophy Affects Provers

- ✓ Soundness easier: literal formula, not instantiation mechanism
- ✓ An axiom is one formula. Axiom schema is a decision algorithm.
- ✓ Generic formula, not some shape with characterization of exceptions
- ✓ No schema variable or meta variable algorithms
- ✓ No matching mechanisms / unification in prover kernel
- ✓ No side condition subtlety or occurrence pattern checks (per schema)
- × Need other means of instantiating axioms: uniform substitution (US)
- ✓ US + renaming: isolate static semantics
- ✓ US independent from axioms: modular logic vs. prover separation
- ✓ More flexible by syntactic contextual equivalence
- × Extra proofs branches since instantiation is explicit proof step

Axioms vs. Axiom Schemata: Philosophy Affects Provers

- ✓ Soundness easier: literal formula, not instantiation mechanism
- ✓ An axiom is one formula. Axiom schema is a decision algorithm.
- ✓ Generic formula, not some shape with characterization of exceptions
- ✓ No schema variable or meta variable algorithms
- ✓ No matching mechanisms / unification in prover kernel
- ✓ No side condition subtlety or occurrence pattern checks (per schema)
- × Need other means of instantiating axioms: uniform substitution (US)
- ✓ US + renaming: isolate static semantics
- ✓ US independent from axioms: modular logic vs. prover separation
- ✓ More flexible by syntactic contextual equivalence
- × Extra proofs branches since instantiation is explicit proof step

---

$\sum$ Net win for soundness since significantly simpler prover

KeYmaera X Kernel is a Microkernel for Soundness

	≈LOC		
KeYmaera X	1 677		hybrid prover
KeYmaera	65 989		
KeY	51 328		Java
HOL Light	396		general math
Isabelle/Pure	8 113		
Nuprl	15 000	+ 50 000	
Coq	20 000		
HSolver	20 000		hybrid verifier
Flow*	25 000		
PHAVer	30 000		
dReal	50 000	+ millions	
SpaceEx	100 000		
HyCreate2	6 081	+ user model analysis	

Disclaimer: These self-reported estimates of the soundness-critical lines of code + rules are to be taken with a grain of salt. Different languages, capabilities, styles

Uniform Substitution

Theorem (Soundness) replace all occurrences of $p(\cdot)$

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

provided $FV(\sigma|_{\Sigma(\theta)}) \cap BV(\otimes(\cdot)) = \emptyset$ *for each operation* $\otimes(\theta)$ *in* ϕ

i.e. bound variables $U = \mathrm{BV}(\otimes(\cdot))$ of operator $\otimes$
are not free in the substitution on its argument θ (U-admissible)

$$\text{US}\ \frac{[a \cup b]p(\bar{x}) \leftrightarrow [a]p(\bar{x}) \land [b]p(\bar{x})}{[x := x+1 \cup x' = 1]x \geq 0 \leftrightarrow [x := x+1]x \geq 0 \land [x' = 1]x \geq 0}$$

Theorem (Soundness) replace all occurrences of $p(\cdot)$

$$US \; \frac{\phi}{\sigma(\phi)}$$

provided $FV(\sigma|_{\Sigma(\theta)}) \cap BV(\otimes(\cdot)) = \emptyset$ *for each operation* $\otimes(\theta)$ *in* ϕ

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

are not free in the substitution on its argument θ (U-admissible)

Uniform substitution σ replaces all occurrences of $p(\theta)$ for any θ by $\psi(\theta)$

function $f(\theta)$ for any θ by $\eta(\theta)$

quantifier $C(\phi)$ for any ϕ by $\psi(\theta)$

program const. a by α

$$\text{US}\,\frac{[a \cup b]p(\bar{x}) \leftrightarrow [a]p(\bar{x}) \land [b]p(\bar{x})}{[x := x+1 \cup x' = 1]x \geq 0 \leftrightarrow [x := x+1]x \geq 0 \land [x' = 1]x \geq 0}$$

$$
\begin{array}{rcll}
\sigma(f(\theta)) & \overset{\text{def}}{=} & & \text{for function symbol } f \in \sigma \\
& \equiv & & \\
\sigma(\theta + \eta) & = & & \\
\sigma((\theta)') & = & & \\
\hline
\sigma(p(\theta)) & \equiv & & \text{for predicate symbol } p \in \sigma \\
\sigma(C(\phi)) & \equiv & & \\
\sigma(\phi \land \psi) & \equiv & & \\
\sigma(\forall x\, \phi) & = & & \\
\sigma([\alpha]\phi) & = & & \\
\hline
\sigma(a) & \equiv & & \text{for program constant } a \in \sigma \\
\sigma(x := \theta) & \equiv & & \\
\sigma(x' = f(x) \,\&\, Q) & \equiv & & \\
\sigma(\alpha \cup \beta) & \equiv & & \\
\sigma(\alpha; \beta) & \equiv & & \\
\sigma(\alpha^*) & \equiv & &
\end{array}
$$

Uniform Substitution: Definition expanded explicitly

$$
\begin{array}{rcll}
\sigma(f(\theta)) & = & (\sigma(f))(\sigma(\theta)) & \text{for function symbol } f \in \sigma \\
 & \stackrel{\text{def}}{=} & \{\cdot \mapsto \sigma(\theta)\}(\sigma f(\cdot)) & \\
\sigma(\theta + \eta) & = & & \\
\sigma((\theta)') & = & & \\
\hline
\sigma(p(\theta)) & \equiv & & \text{for predicate symbol } p \in \sigma \\
\sigma(C(\phi)) & \equiv & & \\
\sigma(\phi \land \psi) & \equiv & & \\
\sigma(\forall x\, \phi) & = & & \\
\sigma([\alpha]\phi) & = & & \\
\hline
\sigma(a) & \equiv & & \text{for program constant } a \in \sigma \\
\sigma(x := \theta) & \equiv & & \\
\sigma(x' = f(x)\,\&\, Q) & \equiv & & \\
\sigma(\alpha \cup \beta) & \equiv & & \\
\sigma(\alpha; \beta) & \equiv & & \\
\sigma(\alpha^*) & \equiv & &
\end{array}
$$

Uniform Substitution: Definition expanded explicitly

$$
\begin{aligned}
\sigma(f(\theta)) &= (\sigma(f))(\sigma(\theta)) && \text{for function symbol } f \in \sigma \\
&\overset{\text{def}}{=} \{\cdot \mapsto \sigma(\theta)\}(\sigma f(\cdot)) \\
\sigma(\theta + \eta) &= \sigma(\theta) + \sigma(\eta) \\
\sigma((\theta)') &= \\
\hline
\sigma(p(\theta)) &\equiv && \text{for predicate symbol } p \in \sigma \\
\sigma(C(\phi)) &\equiv \\
\sigma(\phi \land \psi) &\equiv \\
\sigma(\forall x\, \phi) &= \\
\sigma([\alpha]\phi) &= \\
\hline
\sigma(a) &\equiv && \text{for program constant } a \in \sigma \\
\sigma(x := \theta) &\equiv \\
\sigma(x' = f(x) \,\&\, Q) &\equiv \\
\sigma(\alpha \cup \beta) &\equiv \\
\sigma(\alpha; \beta) &\equiv \\
\sigma(\alpha^*) &\equiv
\end{aligned}
$$

Uniform Substitution: Definition expanded explicitly

$$
\begin{array}{rcll}
\sigma(f(\theta)) &=& (\sigma(f))(\sigma(\theta)) & \text{for function symbol } f \in \sigma \\
&\overset{\text{def}}{=}& \{\cdot \mapsto \sigma(\theta)\}(\sigma f(\cdot)) & \\
\sigma(\theta + \eta) &=& \sigma(\theta) + \sigma(\eta) & \\
\sigma((\theta)') &=& (\sigma(\theta))' & \text{if } \sigma\ \mathcal{V} \cup \mathcal{V}'\text{-admissible for } \theta \\
\hline
\sigma(p(\theta)) &\equiv& & \text{for predicate symbol } p \in \sigma \\
\sigma(C(\phi)) &\equiv& & \\
\sigma(\phi \land \psi) &\equiv& & \\
\sigma(\forall x\, \phi) &=& & \\
\sigma([\alpha]\phi) &=& & \\
\hline
\sigma(a) &\equiv& & \text{for program constant } a \in \sigma \\
\sigma(x := \theta) &\equiv& & \\
\sigma(x' = f(x) \,\&\, Q) &\equiv& & \\
\sigma(\alpha \cup \beta) &\equiv& & \\
\sigma(\alpha; \beta) &\equiv& & \\
\sigma(\alpha^*) &\equiv& &
\end{array}
$$

Uniform Substitution: Definition expanded explicitly

$$
\begin{array}{rcll}
\sigma(f(\theta)) & = & (\sigma(f))(\sigma(\theta)) & \text{for function symbol } f \in \sigma \\
& \overset{\text{def}}{=} & \{\cdot \mapsto \sigma(\theta)\}(\sigma f(\cdot)) & \\
\sigma(\theta + \eta) & = & \sigma(\theta) + \sigma(\eta) & \\
\sigma((\theta)') & = & (\sigma(\theta))' & \text{if } \sigma \ \mathcal{V} \cup \mathcal{V}'\text{-admissible for } \theta \\
\hline
\sigma(p(\theta)) & \equiv & (\sigma(p))(\sigma(\theta)) & \text{for predicate symbol } p \in \sigma \\
\sigma(C(\phi)) & \equiv & & \\
\sigma(\phi \land \psi) & \equiv & & \\
\sigma(\forall x\, \phi) & = & & \\
\sigma([\alpha]\phi) & = & & \\
\hline
\sigma(a) & \equiv & & \text{for program constant } a \in \sigma \\
\sigma(x := \theta) & \equiv & & \\
\sigma(x' = f(x) \,\&\, Q) & \equiv & & \\
\sigma(\alpha \cup \beta) & \equiv & & \\
\sigma(\alpha; \beta) & \equiv & & \\
\sigma(\alpha^*) & \equiv & &
\end{array}
$$

Uniform Substitution: Definition expanded explicitly

$$
\begin{array}{rcll}
\sigma(f(\theta)) & = & (\sigma(f))(\sigma(\theta)) & \text{for function symbol } f \in \sigma \\
& \stackrel{\text{def}}{=} & \{\cdot \mapsto \sigma(\theta)\}(\sigma f(\cdot)) & \\
\sigma(\theta + \eta) & = & \sigma(\theta) + \sigma(\eta) & \\
\sigma((\theta)') & = & (\sigma(\theta))' & \text{if } \sigma\ \mathcal{V} \cup \mathcal{V}'\text{-admissible for } \theta \\
\hline
\sigma(p(\theta)) & \equiv & (\sigma(p))(\sigma(\theta)) & \text{for predicate symbol } p \in \sigma \\
\sigma(C(\phi)) & \equiv & \sigma(C)(\sigma(\phi)) & \text{if } \sigma\ \mathcal{V} \cup \mathcal{V}'\text{-admissible for } \phi,\ C \in \\
\sigma(\phi \wedge \psi) & \equiv & & \\
\sigma(\forall x\, \phi) & = & & \\
\sigma([\alpha]\phi) & = & & \\
\hline
\sigma(a) & \equiv & & \text{for program constant } a \in \sigma \\
\sigma(x := \theta) & \equiv & & \\
\sigma(x' = f(x)\,\&\, Q) & \equiv & & \\
\sigma(\alpha \cup \beta) & \equiv & & \\
\sigma(\alpha; \beta) & \equiv & & \\
\sigma(\alpha^*) & \equiv & &
\end{array}
$$

Uniform Substitution: Definition expanded explicitly

$$
\begin{aligned}
\sigma(f(\theta)) &= (\sigma(f))(\sigma(\theta)) && \text{for function symbol } f \in \sigma \\
&\stackrel{\text{def}}{=} \{\cdot \mapsto \sigma(\theta)\}(\sigma f(\cdot)) \\
\sigma(\theta + \eta) &= \sigma(\theta) + \sigma(\eta) \\
\sigma((\theta)') &= (\sigma(\theta))' && \text{if } \sigma\ \mathcal{V} \cup \mathcal{V}'\text{-admissible for } \theta \\
\hline
\sigma(p(\theta)) &\equiv (\sigma(p))(\sigma(\theta)) && \text{for predicate symbol } p \in \sigma \\
\sigma(C(\phi)) &\equiv \sigma(C)(\sigma(\phi)) && \text{if } \sigma\ \mathcal{V} \cup \mathcal{V}'\text{-admissible for } \phi,\ C \in \\
\sigma(\phi \land \psi) &\equiv \sigma(\phi) \land \sigma(\psi) \\
\sigma(\forall x\, \phi) &= \\
\sigma([\alpha]\phi) &= \\
\hline
\sigma(a) &\equiv && \text{for program constant } a \in \sigma \\
\sigma(x := \theta) &\equiv \\
\sigma(x' = f(x)\, \&\, Q) &\equiv \\
\sigma(\alpha \cup \beta) &\equiv \\
\sigma(\alpha; \beta) &\equiv \\
\sigma(\alpha^*) &\equiv
\end{aligned}
$$

Uniform Substitution: Definition expanded explicitly

$$
\begin{array}{rcll}
\sigma(f(\theta)) & = & (\sigma(f))(\sigma(\theta)) & \text{for function symbol } f \in \sigma \\
 & \overset{\text{def}}{=} & \{\cdot \mapsto \sigma(\theta)\}(\sigma f(\cdot)) & \\
\sigma(\theta + \eta) & = & \sigma(\theta) + \sigma(\eta) & \\
\sigma((\theta)') & = & (\sigma(\theta))' & \text{if } \sigma\ \mathcal{V} \cup \mathcal{V}'\text{-admissible for } \theta \\
\hline
\sigma(p(\theta)) & \equiv & (\sigma(p))(\sigma(\theta)) & \text{for predicate symbol } p \in \sigma \\
\sigma(C(\phi)) & \equiv & \sigma(C)(\sigma(\phi)) & \text{if } \sigma\ \mathcal{V} \cup \mathcal{V}'\text{-admissible for } \phi,\ C \in \\
\sigma(\phi \wedge \psi) & \equiv & \sigma(\phi) \wedge \sigma(\psi) & \\
\sigma(\forall x\, \phi) & = & \forall x\, \sigma(\phi) & \text{if } \sigma\ \{x\}\text{-admissible for } \phi \\
\sigma([\alpha]\phi) & = & & \\
\hline
\sigma(a) & \equiv & & \text{for program constant } a \in \sigma \\
\sigma(x := \theta) & \equiv & & \\
\sigma(x' = f(x)\,\&\, Q) & \equiv & & \\
\sigma(\alpha \cup \beta) & \equiv & & \\
\sigma(\alpha; \beta) & \equiv & & \\
\sigma(\alpha^*) & \equiv & &
\end{array}
$$

Uniform Substitution: Definition expanded explicitly

$$
\begin{array}{rcll}
\sigma(f(\theta)) &=& (\sigma(f))(\sigma(\theta)) & \text{for function symbol } f \in \sigma \\
&\overset{\text{def}}{=}& \{\cdot \mapsto \sigma(\theta)\}(\sigma f(\cdot)) & \\
\sigma(\theta + \eta) &=& \sigma(\theta) + \sigma(\eta) & \\
\sigma((\theta)') &=& (\sigma(\theta))' & \text{if } \sigma\ \mathcal{V} \cup \mathcal{V}'\text{-admissible for } \theta \\
\hline
\sigma(p(\theta)) &\equiv& (\sigma(p))(\sigma(\theta)) & \text{for predicate symbol } p \in \sigma \\
\sigma(C(\phi)) &\equiv& \sigma(C)(\sigma(\phi)) & \text{if } \sigma\ \mathcal{V} \cup \mathcal{V}'\text{-admissible for } \phi,\ C \in \\
\sigma(\phi \wedge \psi) &\equiv& \sigma(\phi) \wedge \sigma(\psi) & \\
\sigma(\forall x\, \phi) &=& \forall x\, \sigma(\phi) & \text{if } \sigma\ \{x\}\text{-admissible for } \phi \\
\sigma([\alpha]\phi) &=& [\sigma(\alpha)]\sigma(\phi) & \text{if } \sigma\ BV(\sigma(\alpha))\text{-admissible for } \phi \\
\hline
\sigma(a) &\equiv& & \text{for program constant } a \in \sigma \\
\sigma(x := \theta) &\equiv& & \\
\sigma(x' = f(x)\, \&\, Q) &\equiv& & \\
\sigma(\alpha \cup \beta) &\equiv& & \\
\sigma(\alpha; \beta) &\equiv& & \\
\sigma(\alpha^*) &\equiv& &
\end{array}
$$

$$
\begin{array}{rcll}
\sigma(f(\theta)) & = & (\sigma(f))(\sigma(\theta)) & \text{for function symbol } f \in \sigma \\
& \stackrel{\text{def}}{=} & \{\cdot \mapsto \sigma(\theta)\}(\sigma f(\cdot)) & \\
\sigma(\theta + \eta) & = & \sigma(\theta) + \sigma(\eta) & \\
\sigma((\theta)') & = & (\sigma(\theta))' & \text{if } \sigma\ \mathcal{V} \cup \mathcal{V}'\text{-admissible for } \theta \\
\hline
\sigma(p(\theta)) & \equiv & (\sigma(p))(\sigma(\theta)) & \text{for predicate symbol } p \in \sigma \\
\sigma(C(\phi)) & \equiv & \sigma(C)(\sigma(\phi)) & \text{if } \sigma\ \mathcal{V} \cup \mathcal{V}'\text{-admissible for } \phi,\ C \in \\
\sigma(\phi \land \psi) & \equiv & \sigma(\phi) \land \sigma(\psi) & \\
\sigma(\forall x\, \phi) & = & \forall x\, \sigma(\phi) & \text{if } \sigma\ \{x\}\text{-admissible for } \phi \\
\sigma([\alpha]\phi) & = & [\sigma(\alpha)]\sigma(\phi) & \text{if } \sigma\ \mathrm{BV}(\sigma(\alpha))\text{-admissible for } \phi \\
\hline
\sigma(a) & \equiv & \sigma a & \text{for program constant } a \in \sigma \\
\sigma(x := \theta) & \equiv & & \\
\sigma(x' = f(x)\,\&\, Q) & \equiv & & \\
\sigma(\alpha \cup \beta) & \equiv & & \\
\sigma(\alpha; \beta) & \equiv & & \\
\sigma(\alpha^*) & \equiv & &
\end{array}
$$

Uniform Substitution: Definition expanded explicitly

$$
\begin{array}{rcll}
\sigma(f(\theta)) &=& (\sigma(f))(\sigma(\theta)) & \text{for function symbol } f \in \sigma \\
&\overset{\text{def}}{=}& \{\cdot \mapsto \sigma(\theta)\}(\sigma f(\cdot)) & \\
\sigma(\theta + \eta) &=& \sigma(\theta) + \sigma(\eta) & \\
\sigma((\theta)') &=& (\sigma(\theta))' & \text{if } \sigma\ \mathcal{V} \cup \mathcal{V}'\text{-admissible for } \theta \\
\hline
\sigma(p(\theta)) &\equiv& (\sigma(p))(\sigma(\theta)) & \text{for predicate symbol } p \in \sigma \\
\sigma(C(\phi)) &\equiv& \sigma(C)(\sigma(\phi)) & \text{if } \sigma\ \mathcal{V} \cup \mathcal{V}'\text{-admissible for } \phi,\ C \in \\
\sigma(\phi \wedge \psi) &\equiv& \sigma(\phi) \wedge \sigma(\psi) & \\
\sigma(\forall x\, \phi) &=& \forall x\, \sigma(\phi) & \text{if } \sigma\ \{x\}\text{-admissible for } \phi \\
\sigma([\alpha]\phi) &=& [\sigma(\alpha)]\sigma(\phi) & \text{if } \sigma\ \mathrm{BV}(\sigma(\alpha))\text{-admissible for } \phi \\
\hline
\sigma(a) &\equiv& \sigma a & \text{for program constant } a \in \sigma \\
\sigma(x := \theta) &\equiv& x := \sigma(\theta) & \\
\sigma(x' = f(x)\,\&\, Q) &\equiv& & \\
\sigma(\alpha \cup \beta) &\equiv& & \\
\sigma(\alpha; \beta) &\equiv& & \\
\sigma(\alpha^*) &\equiv& &
\end{array}
$$

Uniform Substitution: Definition expanded explicitly

$$
\begin{array}{rcll}
\sigma(f(\theta)) & = & (\sigma(f))(\sigma(\theta)) & \text{for function symbol } f \in \sigma \\
 & \stackrel{\text{def}}{=} & \{\cdot \mapsto \sigma(\theta)\}(\sigma f(\cdot)) & \\
\sigma(\theta + \eta) & = & \sigma(\theta) + \sigma(\eta) & \\
\sigma((\theta)') & = & (\sigma(\theta))' & \text{if } \sigma\ \mathcal{V} \cup \mathcal{V}'\text{-admissible for } \theta \\
\hline
\sigma(p(\theta)) & \equiv & (\sigma(p))(\sigma(\theta)) & \text{for predicate symbol } p \in \sigma \\
\sigma(C(\phi)) & \equiv & \sigma(C)(\sigma(\phi)) & \text{if } \sigma\ \mathcal{V} \cup \mathcal{V}'\text{-admissible for } \phi,\ C \in \\
\sigma(\phi \land \psi) & \equiv & \sigma(\phi) \land \sigma(\psi) & \\
\sigma(\forall x\, \phi) & = & \forall x\, \sigma(\phi) & \text{if } \sigma\ \{x\}\text{-admissible for } \phi \\
\sigma([\alpha]\phi) & = & [\sigma(\alpha)]\sigma(\phi) & \text{if } \sigma\ \mathrm{BV}(\sigma(\alpha))\text{-admissible for } \phi \\
\hline
\sigma(a) & \equiv & \sigma a & \text{for program constant } a \in \sigma \\
\sigma(x := \theta) & \equiv & x := \sigma(\theta) & \\
\sigma(x' = f(x)\ \&\ Q) & \equiv & x' = \sigma(f(x))\ \&\ \sigma(Q) & \text{if } \sigma\ \{x, x'\}\text{-admissible for } f(x), Q \\
\sigma(\alpha \cup \beta) & \equiv & & \\
\sigma(\alpha; \beta) & \equiv & & \\
\sigma(\alpha^*) & \equiv & &
\end{array}
$$

Uniform Substitution: Definition expanded explicitly

$$
\begin{array}{rcll}
\sigma(f(\theta)) & = & (\sigma(f))(\sigma(\theta)) & \text{for function symbol } f \in \sigma \\
 & \stackrel{\text{def}}{=} & \{\cdot \mapsto \sigma(\theta)\}(\sigma f(\cdot)) & \\
\sigma(\theta + \eta) & = & \sigma(\theta) + \sigma(\eta) & \\
\sigma((\theta)') & = & (\sigma(\theta))' & \text{if } \sigma\ \mathcal{V} \cup \mathcal{V}'\text{-admissible for } \theta \\
\hline
\sigma(p(\theta)) & \equiv & (\sigma(p))(\sigma(\theta)) & \text{for predicate symbol } p \in \sigma \\
\sigma(C(\phi)) & \equiv & \sigma(C)(\sigma(\phi)) & \text{if } \sigma\ \mathcal{V} \cup \mathcal{V}'\text{-admissible for } \phi,\ C \in \\
\sigma(\phi \land \psi) & \equiv & \sigma(\phi) \land \sigma(\psi) & \\
\sigma(\forall x\, \phi) & = & \forall x\, \sigma(\phi) & \text{if } \sigma\ \{x\}\text{-admissible for } \phi \\
\sigma([\alpha]\phi) & = & [\sigma(\alpha)]\sigma(\phi) & \text{if } \sigma\ BV(\sigma(\alpha))\text{-admissible for } \phi \\
\hline
\sigma(a) & \equiv & \sigma a & \text{for program constant } a \in \sigma \\
\sigma(x := \theta) & \equiv & x := \sigma(\theta) & \\
\sigma(x' = f(x)\,\&\, Q) & \equiv & x' = \sigma(f(x))\,\&\, \sigma(Q) & \text{if } \sigma\ \{x, x'\}\text{-admissible for } f(x), Q \\
\sigma(\alpha \cup \beta) & \equiv & \sigma(\alpha) \cup \sigma(\beta) & \\
\sigma(\alpha; \beta) & \equiv & & \\
\sigma(\alpha^*) & \equiv & &
\end{array}
$$

Uniform Substitution: Definition expanded explicitly

$$
\begin{array}{rcll}
\sigma(f(\theta)) & = & (\sigma(f))(\sigma(\theta)) & \text{for function symbol } f \in \sigma \\
 & \overset{\text{def}}{=} & \{\cdot \mapsto \sigma(\theta)\}(\sigma f(\cdot)) & \\
\sigma(\theta + \eta) & = & \sigma(\theta) + \sigma(\eta) & \\
\sigma((\theta)') & = & (\sigma(\theta))' & \text{if } \sigma\ \mathcal{V} \cup \mathcal{V}'\text{-admissible for } \theta \\
\hline
\sigma(p(\theta)) & \equiv & (\sigma(p))(\sigma(\theta)) & \text{for predicate symbol } p \in \sigma \\
\sigma(C(\phi)) & \equiv & \sigma(C)(\sigma(\phi)) & \text{if } \sigma\ \mathcal{V} \cup \mathcal{V}'\text{-admissible for } \phi,\ C \in \\
\sigma(\phi \land \psi) & \equiv & \sigma(\phi) \land \sigma(\psi) & \\
\sigma(\forall x\, \phi) & = & \forall x\, \sigma(\phi) & \text{if } \sigma\ \{x\}\text{-admissible for } \phi \\
\sigma([\alpha]\phi) & = & [\sigma(\alpha)]\sigma(\phi) & \text{if } \sigma\ \mathrm{BV}(\sigma(\alpha))\text{-admissible for } \phi \\
\hline
\sigma(a) & \equiv & \sigma a & \text{for program constant } a \in \sigma \\
\sigma(x := \theta) & \equiv & x := \sigma(\theta) & \\
\sigma(x' = f(x) \,\&\, Q) & \equiv & x' = \sigma(f(x)) \,\&\, \sigma(Q) & \text{if } \sigma\ \{x, x'\}\text{-admissible for } f(x), Q \\
\sigma(\alpha \cup \beta) & \equiv & \sigma(\alpha) \cup \sigma(\beta) & \\
\sigma(\alpha; \beta) & \equiv & \sigma(\alpha); \sigma(\beta) & \text{if } \sigma\ \mathrm{BV}(\sigma(\alpha))\text{-admissible for } \beta \\
\sigma(\alpha^*) & \equiv & &
\end{array}
$$

$\sigma(f(\theta))$	$=$	$(\sigma(f))(\sigma(\theta))$	for function symbol $f \in \sigma$
	$\stackrel{\text{def}}{=}$	$\{\cdot \mapsto \sigma(\theta)\}(\sigma f(\cdot))$	
$\sigma(\theta + \eta)$	$=$	$\sigma(\theta) + \sigma(\eta)$	
$\sigma((\theta)')$	$=$	$(\sigma(\theta))'$	if σ $\mathcal{V} \cup \mathcal{V}'$-admissible for θ
$\sigma(p(\theta))$	$\equiv$	$(\sigma(p))(\sigma(\theta))$	for predicate symbol $p \in \sigma$
$\sigma(C(\phi))$	$\equiv$	$\sigma(C)(\sigma(\phi))$	if σ $\mathcal{V} \cup \mathcal{V}'$-admissible for ϕ, $C \in$
$\sigma(\phi \wedge \psi)$	$\equiv$	$\sigma(\phi) \wedge \sigma(\psi)$	
$\sigma(\forall x\, \phi)$	$=$	$\forall x\, \sigma(\phi)$	if σ $\{x\}$-admissible for ϕ
$\sigma([\alpha]\phi)$	$=$	$[\sigma(\alpha)]\sigma(\phi)$	if σ $\mathrm{BV}(\sigma(\alpha))$-admissible for ϕ
$\sigma(a)$	$\equiv$	σa	for program constant $a \in \sigma$
$\sigma(x := \theta)$	$\equiv$	$x := \sigma(\theta)$	
$\sigma(x' = f(x) \,\&\, Q)$	$\equiv$	$x' = \sigma(f(x)) \,\&\, \sigma(Q)$	if σ $\{x, x'\}$-admissible for $f(x)$, Q
$\sigma(\alpha \cup \beta)$	$\equiv$	$\sigma(\alpha) \cup \sigma(\beta)$	
$\sigma(\alpha; \beta)$	$\equiv$	$\sigma(\alpha); \sigma(\beta)$	if σ $\mathrm{BV}(\sigma(\alpha))$-admissible for β
$\sigma(\alpha^*)$	$\equiv$	$(\sigma(\alpha))^*$	if σ $\mathrm{BV}(\sigma(\alpha))$-admissible for α

Uniform Substitution: Examples

$$\frac{[x := f]p(x) \leftrightarrow p(f)}{[x := x+1]x \neq x \leftrightarrow x+1 \neq x} \qquad \sigma = \{f \mapsto x+1, p(\cdot) \mapsto (\cdot \neq x)\}$$

$$\frac{[x := f]p(x) \leftrightarrow p(f)}{[x := x^2][(z := x{+}z)^*; z := x{+}yz]y{\geq}x \leftrightarrow [(z := x^2{+}z^*); z := x^2{+}yz]y{\geq}x^2}$$

with $\sigma = \{f \mapsto x^2, p(\cdot) \mapsto [(z := \cdot + z)^*; z := \cdot + yz]y \geq \cdot\}$

$$\frac{p \rightarrow [a]p}{x \geq 0 \rightarrow [x' = -1]x \geq 0} \qquad \sigma = \{a \mapsto x' = -1, p \mapsto x \geq 0\}$$

$$\frac{(-x)^2 \geq 0}{[x' = -1](-x)^2 \geq 0} \quad \text{by} \quad \frac{p(\bar{x})}{[a]p(\bar{x})} \quad \sigma = \{a \mapsto x' = -1, p(\cdot) \mapsto (-\cdot)^2 \geq 0\}$$

Uniform Substitution: Examples

$$\frac{[x:=f]p(x) \leftrightarrow p(f)}{[x:=x+1]x \neq x \leftrightarrow x+1 \neq x} \qquad \sigma = \{f \mapsto x+1, p(\cdot) \mapsto (\cdot \neq x)\}$$

$$\frac{[x:=f]p(x) \leftrightarrow p(f)}{[x:=x^2][(z:=x+z)^*; z:=x+yz]y \geq x \leftrightarrow [(z:=x^2+z^*); z:=x^2+yz]y \geq x^2}$$

with $\sigma = \{f \mapsto x^2, p(\cdot) \mapsto [(z:=\cdot+z)^*; z:=\cdot+yz]y \geq \cdot\}$

$$\frac{p \to [a]p}{x \geq 0 \to [x'=-1]x \geq 0} \qquad \sigma = \{a \mapsto x'=-1, p \mapsto x \geq 0\}$$

$$\frac{(-x)^2 \geq 0}{[x'=-1](-x)^2 \geq 0} \quad \text{by} \quad \frac{p(\bar{x})}{[a]p(\bar{x})} \quad \sigma = \{a \mapsto x'=-1, p(\cdot) \mapsto (-\cdot)^2 \geq 0\}$$

Uniform Substitution: Examples

BV

Clash

FV

$$\frac{[x := f]p(x) \leftrightarrow p(f)}{[x := x+1]x \neq x \leftrightarrow x+1 \neq x} \qquad \sigma = \{f \mapsto x+1, p(\cdot) \mapsto (\cdot \neq x)\}$$

$$\frac{[x := f]p(x) \leftrightarrow p(f)}{[x := x^2][(z := x+z)^*; z := x+yz]y \geq x \leftrightarrow [(z := x^2+z^*); z := x^2+yz]y \geq x^2}$$

with $\sigma = \{f \mapsto x^2, p(\cdot) \mapsto [(z := \cdot + z)^*; z := \cdot + yz]y \geq \cdot\}$

$$\frac{p \to [a]p}{x \geq 0 \to [x' = -1]x \geq 0} \qquad \sigma = \{a \mapsto x' = -1, p \mapsto x \geq 0\}$$

$$\frac{(-x)^2 \geq 0}{[x' = -1](-x)^2 \geq 0} \quad \text{by} \quad \frac{p(\bar{x})}{[a]p(\bar{x})} \qquad \sigma = \{a \mapsto x' = -1, p(\cdot) \mapsto (-\cdot)^2 \geq 0\}$$

Clash

$$\frac{[x := f]p(x) \leftrightarrow p(f)}{[x := x+1]x \neq x \leftrightarrow x+1 \neq x} \qquad \sigma = \{f \mapsto x+1, p(\cdot) \mapsto (\cdot \neq x)\}$$

$$\frac{[x := f]p(x) \leftrightarrow p(f)}{[x := x^2][(z := x+z)^*; z := x+yz]y \geq x \leftrightarrow [(z := x^2+z^*); z := x^2+yz]y \geq x^2}$$

with $\sigma = \{f \mapsto x^2, p(\cdot) \mapsto [(z := \cdot + z)^*; z := \cdot + yz]y \geq \cdot\}$

$$\frac{p \rightarrow [a]p}{x \geq 0 \rightarrow [x' = -1]x \geq 0} \qquad \sigma = \{a \mapsto x' = -1, p \mapsto x \geq 0\}$$

$$\frac{(-x)^2 \geq 0}{[x' = -1](-x)^2 \geq 0} \quad \text{by} \quad \frac{p(\bar{x})}{[a]p(\bar{x})} \qquad \sigma = \{a \mapsto x' = -1, p(\cdot) \mapsto (-\cdot)^2 \geq 0\}$$

Uniform Substitution: Examples

$$\frac{[x := f]p(x) \leftrightarrow p(f)}{[x := x+1]x \neq x \leftrightarrow x+1 \neq x} \qquad \sigma = \{f \mapsto x+1, p(\cdot) \mapsto (\cdot \neq x)\}$$

Clash

$$\frac{[x := f]p(x) \leftrightarrow p(f)}{[x := x^2][(z := x+z)^*; z := x+yz]y \geq x \leftrightarrow [(z := x^2+z^*); z := x^2+yz]y \geq x^2}$$

Correct

with $\sigma = \{f \mapsto x^2, p(\cdot) \mapsto [(z := \cdot + z)^*; z := \cdot + yz]y \geq \cdot\}$

$$\frac{p \to [a]p}{x \geq 0 \to [x' = -1]x \geq 0} \qquad \sigma = \{a \mapsto x' = -1, p \mapsto x \geq 0\}$$

$$\frac{(-x)^2 \geq 0}{[x' = -1](-x)^2 \geq 0} \quad \text{by} \quad \frac{p(\bar{x})}{[a]p(\bar{x})} \qquad \sigma = \{a \mapsto x' = -1, p(\cdot) \mapsto (-\cdot)^2 \geq 0\}$$

Uniform Substitution: Examples

$$\frac{[x := f]p(x) \leftrightarrow p(f)}{[x := x+1]x \neq x \leftrightarrow x+1 \neq x} \qquad \sigma = \{f \mapsto x+1, p(\cdot) \mapsto (\cdot \neq x)\}$$

Clash

$$\frac{[x := f]p(x) \leftrightarrow p(f)}{[x := x^2][(z := x+z)^*; z := x+yz]y \geq x \leftrightarrow [(z := x^2+z^*); z := x^2+yz]y \geq x^2}$$

Correct

with $\sigma = \{f \mapsto x^2, p(\cdot) \mapsto [(z := \cdot + z)^*; z := \cdot + yz]y \geq \cdot\}$

$$\frac{[a]p}{x \geq 0 \rightarrow [x' = -1]x \geq 0} \qquad \sigma = \{a \mapsto x' = -1, p \mapsto x \geq 0\}$$

BV

Clash

FV

$$\frac{(-x)^2 \geq 0}{[x' = -1](-x)^2 \geq 0} \quad \text{by} \quad \frac{p(\bar{x})}{[a]p(\bar{x})} \qquad \sigma = \{a \mapsto x' = -1, p(\cdot) \mapsto (-\cdot)^2 \geq 0\}$$

Uniform Substitution: Examples

$$\frac{[x := f]p(x) \leftrightarrow p(f)}{[x := x+1]x \neq x \leftrightarrow x+1 \neq x}$$ Clash

$$\sigma = \{f \mapsto x+1, p(\cdot) \mapsto (\cdot \neq x)\}$$

$$\frac{[x := f]p(x) \leftrightarrow p(f)}{[x := x^2][(z := x+z)^*; z := x+yz]y \geq x \leftrightarrow [(z := x^2+z^*); z := x^2+yz]y \geq x^2}$$ Correct

with $\sigma = \{f \mapsto x^2, p(\cdot) \mapsto [(z := \cdot + z)^*; z := \cdot + yz]y \geq \cdot\}$

$$\frac{p \to [a]p}{x \geq 0 \to [x' = -1]x \geq 0}$$ Clash

$$\sigma = \{a \mapsto x' = -1, p \mapsto x \geq 0\}$$

$$\frac{(-x)^2 \geq 0}{[x' = -1](-x)^2 \geq 0}$$ Correct by $\dfrac{p(\bar{x})}{[a]p(\bar{x})}$

$$\sigma = \{a \mapsto x' = -1, p(\cdot) \mapsto (-\cdot)^2 \geq 0\}$$

Uniform Substitution: Contextual Congruence Example

$$\text{CE}\ \frac{p(\bar{x}) \leftrightarrow q(\bar{x})}{C(p(\bar{x})) \leftrightarrow C(q(\bar{x}))}$$

$$\text{CE}\ \frac{[x := x^2]x \leq 1 \leftrightarrow x^2 \leq 1}{[x' = x^3 \cup x' = -1][x := x^2]x \leq 1 \leftrightarrow [x' = x^3 \cup x' = -1]x^2 \leq 1}$$

Uniform Substitution: Contextual Congruence Example

$$\text{CE}\ \frac{p(\bar{x}) \leftrightarrow q(\bar{x})}{C(p(\bar{x})) \leftrightarrow C(q(\bar{x}))}$$

$$\text{CE}\ \frac{[x := x^2]x \leq 1 \leftrightarrow x^2 \leq 1}{[x' = x^3 \cup x' = -1][x := x^2]x \leq 1 \leftrightarrow [x' = x^3 \cup x' = -1]x^2 \leq 1}$$

Theorem (Soundness) $(\mathrm{FV}(\sigma) = \emptyset)$

$\dfrac{\phi_1 \quad \dots \quad \phi_n}{\psi}$ *locally sound* *implies* $\dfrac{\sigma(\phi_1) \quad \dots \quad \sigma(\phi_n)}{\sigma(\psi)}$ *locally sound*

Uniform Substitution: Contextual Congruence Example

$$\text{CE}\ \frac{p(\bar{x}) \leftrightarrow q(\bar{x})}{C(p(\bar{x})) \leftrightarrow C(q(\bar{x}))}$$

$$\text{CE}\ \frac{[x := x^2]x \leq 1 \leftrightarrow x^2 \leq 1}{[x' = x^3 \cup x' = -1][x := x^2]x \leq 1 \leftrightarrow [x' = x^3 \cup x' = -1]x^2 \leq 1}$$

Theorem (Soundness) $(\mathrm{FV}(\sigma) = \emptyset)$

$$\frac{\phi_1 \quad \ldots \quad \phi_n}{\psi}\ \textit{locally sound} \quad \textit{implies} \quad \frac{\sigma(\phi_1) \quad \ldots \quad \sigma(\phi_n)}{\sigma(\psi)}\ \textit{locally sound}$$

Locally sound

The conclusion is valid in any interpretation I in which the premises are.

Correctness of Uniform Substitutions

"Syntactic uniform substitution = semantic replacement"

Lemma (Uniform substitution lemma)

Uniform substitution σ *and its adjoint interpretation* $\sigma^*_u I$ *to* σ *for* I, u *have the same semantics:*

$$[\![\sigma(\theta)]\!] I u \;=\; [\![\theta]\!] \sigma^*_u I u$$

$$u \in [\![\sigma(\phi)]\!] I \text{ iff } u \in [\![\phi]\!] \sigma^*_u I$$

$$(u, w) \in [\![\sigma(\alpha)]\!] I \text{ iff } (u, w) \in [\![\alpha]\!] \sigma^*_u I$$

$$\begin{tikzcd}
\theta \arrow[r, mapsto, "\sigma"'] \arrow[drr, mapsto, bend right, "\sigma^*_u I"] & \sigma(\theta) \arrow[r, mapsto, "I"'] & [\![\sigma(\theta)]\!] I u \arrow[d, equal] \\
& & [\![\theta]\!] \sigma^*_u I u
\end{tikzcd}$$

Solving Differential Equations? By Axiom Schema?

$[']$ $[x' = \theta]\phi \leftrightarrow \forall t{\geq}0\, [x := x(t)]\phi$ $\qquad$ (t fresh and $x'(t) = \theta$)

LICS'12

Solving Differential Equations? By Axiom Schema?

$[']$ $[x' = \theta]\phi \leftrightarrow \forall t{\geq}0\,[x := x(t)]\phi$ $\qquad$ (t fresh and $x'(t) = \theta$)

Axiom schema with side conditions:

1. Occurs check: t fresh
2. Solution check: $x(\cdot)$ solves the ODE $x'(t) = \theta$ with $x(\cdot)$ plugged in for x in θ
3. Initial value check: $x(\cdot)$ solves the symbolic IVP $x(0) = x$

Quite nontrivial soundness-critical algorithms ...

LICS'12

Solving Differential Equations? By Axiom Schema?

$[']$ $[x' = \theta]\phi \leftrightarrow \forall t{\geq}0\, [x := x(t)]\phi$ $\quad$ (t fresh and $x'(t) = \theta$)

Axiom schema with side conditions:

1. Occurs check: t fresh
2. Solution check: $x(\cdot)$ solves the ODE $x'(t) = \theta$ with $x(\cdot)$ plugged in for x in θ
3. Initial value check: $x(\cdot)$ solves the symbolic IVP $x(0) = x$
4. $x(\cdot)$ covers all solutions parametrically

Quite nontrivial soundness-critical algorithms ...

LICS'12

Differential Equation Axioms & Differential Axioms

DW $[x' = f(x) \,\&\, q(x)]q(x)$

DC $\big([x' = f(x) \,\&\, q(x)]p(x) \leftrightarrow [x' = f(x) \,\&\, q(x) \land r(x)]p(x)\big)$
$\leftarrow [x' = f(x) \,\&\, q(x)]r(x)$

DE $[x' = f(x) \,\&\, q(x)]p(x, x') \leftrightarrow [x' = f(x) \,\&\, q(x)][x' := f(x)]p(x, x')$

DI $[x' = f(x) \,\&\, q(x)]p(x) \leftarrow \big(q(x) \to p(x) \land [x' = f(x) \,\&\, q(x)](p(x))'\big)$

DG $[x' = f(x) \,\&\, q(x)]p(x) \leftrightarrow \exists y\, [x' = f(x), y' = a(x)y + b(x) \,\&\, q(x)]p(x)$

DS $[x' = f \,\&\, q(x)]p(x) \leftrightarrow \forall t{\geq}0\, \big((\forall 0{\leq}s{\leq}t\; q(x{+}fs)) \to [x := x{+}ft]p(x)\big)$

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

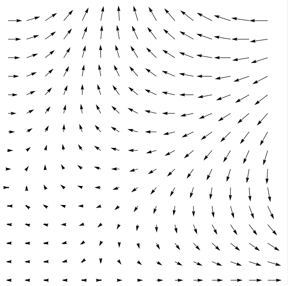
$+'$ $(f(\bar{x}) + g(\bar{x}))' = (f(\bar{x}))' + (g(\bar{x}))'$

$\cdot'$ $(f(\bar{x}) \cdot g(\bar{x}))' = (f(\bar{x}))' \cdot g(\bar{x}) + f(\bar{x}) \cdot (g(\bar{x}))'$

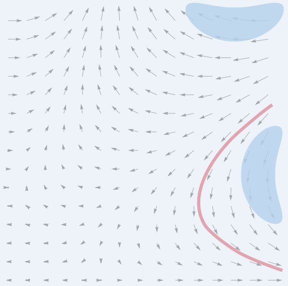
$\circ'$ $[y := g(x)][y' := 1]\big((f(g(x)))' = (f(y))' \cdot (g(x))'\big)$

CADE'15

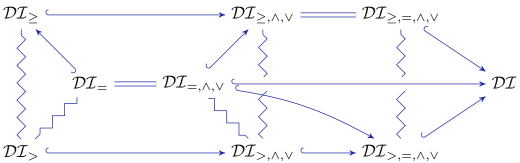
Differential Invariant



Differential Cut



Differential Ghost



Logic

Provability theory

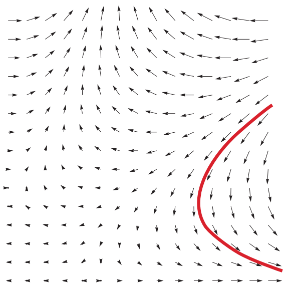
Math

Characteristic PDE

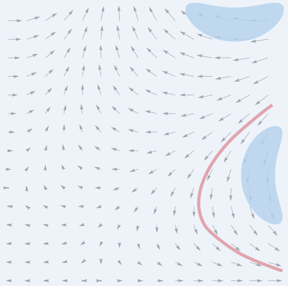
JLogComput'10,CAV'08,FMSD'09,LMCS'12,LICS'12,ITP'12,CADE'15

Differential Invariants for Differential Equations

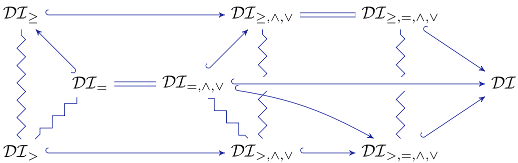
Differential Invariant



Differential Cut



Differential Ghost



Logic

Provability theory

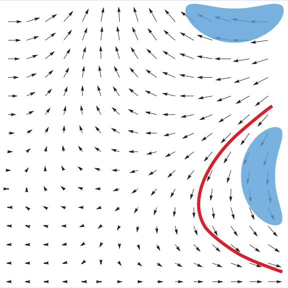
Math

Characteristic PDE

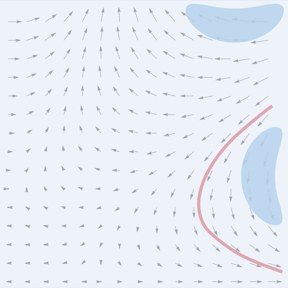
JLogComput'10,CAV'08,FMSD'09,LMCS'12,LICS'12,ITP'12,CADE'15

Differential Invariants for Differential Equations

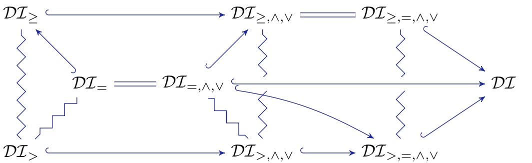
Differential Invariant



Differential Cut



Differential Ghost



Logic

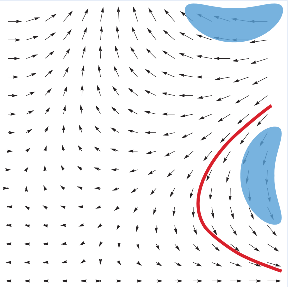
Provability theory

Math

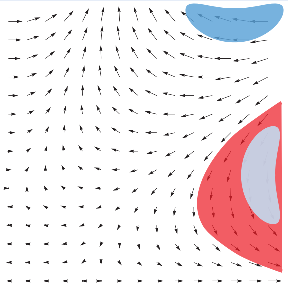
Characteristic PDE

JLogComput'10,CAV'08,FMSD'09,LMCS'12,LICS'12,ITP'12,CADE'15

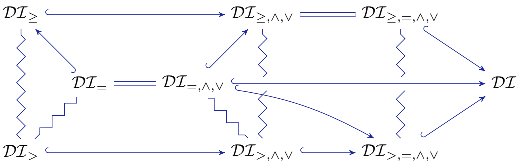
Differential Invariant



Differential Cut



Differential Ghost



Logic

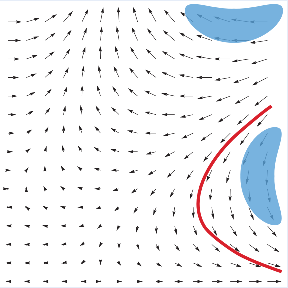
Provability theory

Math

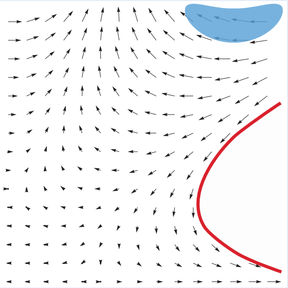
Characteristic PDE

JLogComput'10,CAV'08,FMSD'09,LMCS'12,LICS'12,ITP'12,CADE'15

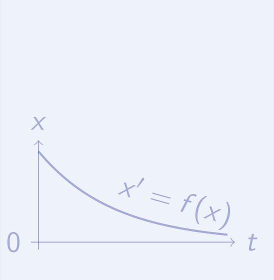
Differential Invariant

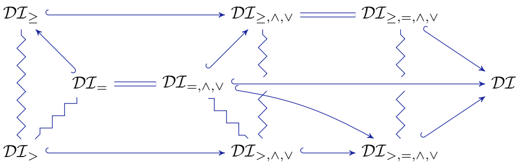


Differential Cut



Differential Ghost





Logic

Provability theory

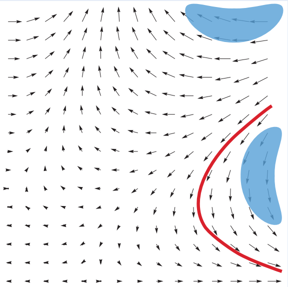
Math

Character-istic PDE

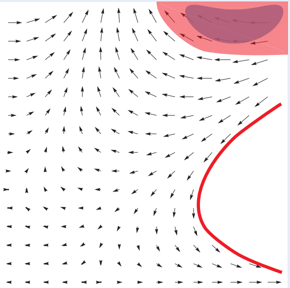
JLogComput'10,CAV'08,FMSD'09,LMCS'12,LICS'12,ITP'12,CADE'15

Differential Invariants for Differential Equations

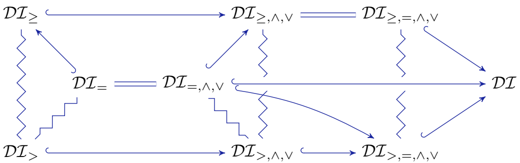
Differential Invariant



Differential Cut



Differential Ghost



Logic

Provability theory

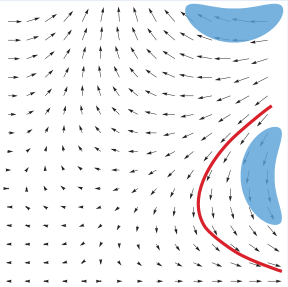
Math

Characteristic PDE

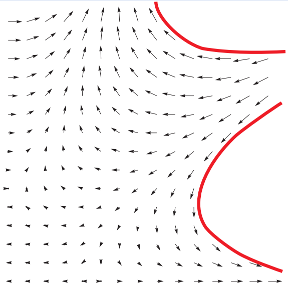
JLogComput'10,CAV'08,FMSD'09,LMCS'12,LICS'12,ITP'12,CADE'15

Differential Invariants for Differential Equations

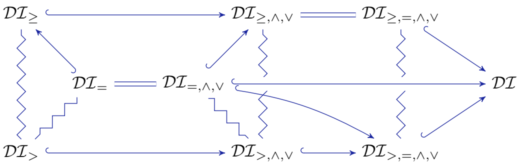
Differential Invariant



Differential Cut



Differential Ghost



Logic

Provability theory

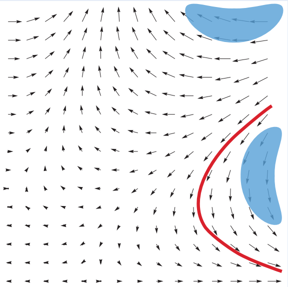
Math

Character-istic PDE

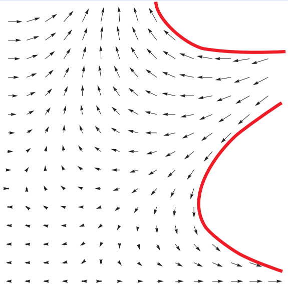
JLogComput'10,CAV'08,FMSD'09,LMCS'12,LICS'12,ITP'12,CADE'15

Differential Invariants for Differential Equations

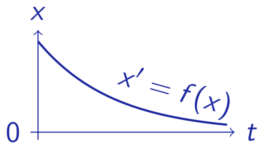
Differential Invariant

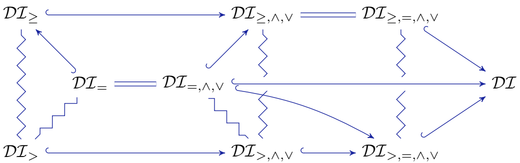


Differential Cut



Differential Ghost





Logic

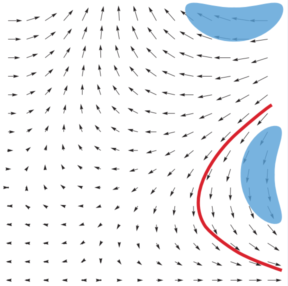
Provability theory

Math

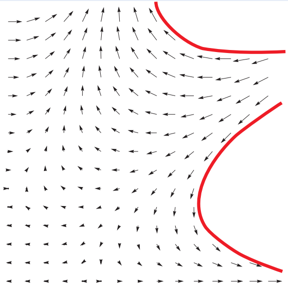
Characteristic PDE

JLogComput'10,CAV'08,FMSD'09,LMCS'12,LICS'12,ITP'12,CADE'15

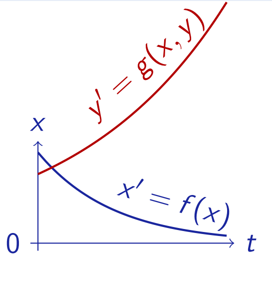
Differential Invariant

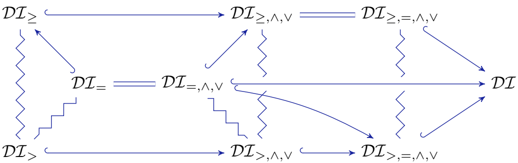


Differential Cut



Differential Ghost





Logic

Provability theory

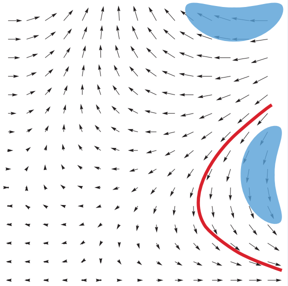
Math

Characteristic PDE

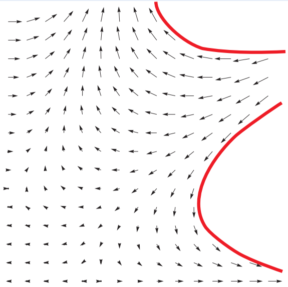
JLogComput'10,CAV'08,FMSD'09,LMCS'12,LICS'12,ITP'12,CADE'15

Differential Invariants for Differential Equations

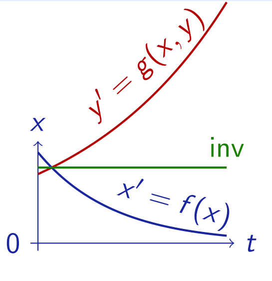
Differential Invariant

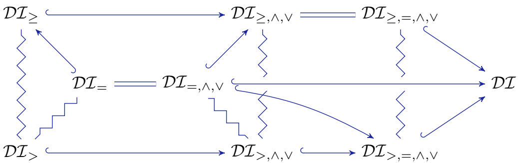


Differential Cut



Differential Ghost





Logic

Provability theory

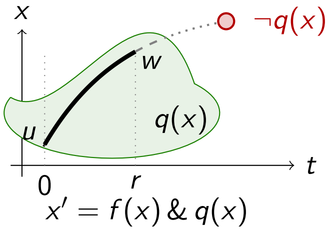
Math

Character-istic PDE

JLogComput'10,CAV'08,FMSD'09,LMCS'12,LICS'12,ITP'12,CADE'15

Axiom (Differential Weakening) (CADE'15)

DW $[x' = f(x) \,\&\, q(x)]q(x)$

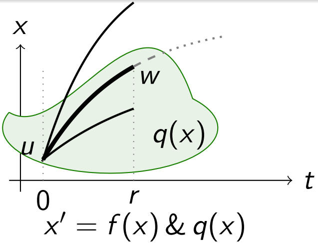


Differential equations cannot leave their evolution domains. Implies:

$$[x' = f(x) \,\&\, q(x)]p(x) \leftrightarrow [x' = f(x) \,\&\, q(x)]\big(q(x) \rightarrow p(x)\big)$$

Axiom (Differential Cut) (CADE'15)

DC $\big([x' = f(x) \,\&\, q(x)]p(x) \leftrightarrow [x' = f(x) \,\&\, q(x) \land r(x)]p(x)\big)$
$\leftarrow [x' = f(x) \,\&\, q(x)]r(x)$

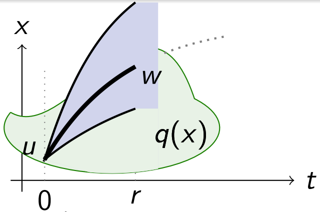


DC is a cut for differential equations.
DC is a differential modal modus ponens K.
Can't leave $r(x)$, then might as well restrict state space to $r(x)$.

Differential Equation Axioms

Axiom (Differential Cut) (CADE'15)

DC $$\big([x' = f(x) \,\&\, q(x)]p(x) \leftrightarrow [x' = f(x) \,\&\, q(x) \wedge r(x)]p(x)\big) \leftarrow [x' = f(x) \,\&\, q(x)]r(x)$$



$x' = f(x) \,\&\, q(x)$

DC is a cut for differential equations.

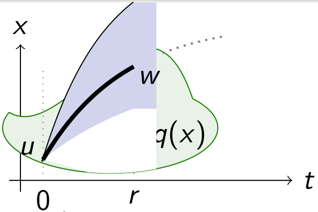
DC is a differential modal modus ponens K.

Can't leave $r(x)$, then might as well restrict state space to $r(x)$.

Differential Equation Axioms

Axiom (Differential Cut) (CADE'15)

DC $\quad ([x' = f(x) \,\&\, q(x)]p(x) \leftrightarrow [x' = f(x) \,\&\, q(x) \land r(x)]p(x))$
$\quad \leftarrow [x' = f(x) \,\&\, q(x)]r(x)$



$x' = f(x) \,\&\, q(x)$

DC is a cut for differential equations.

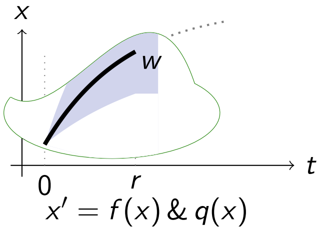
DC is a differential modal modus ponens K.

Can't leave $r(x)$, then might as well restrict state space to $r(x)$.

Differential Equation Axioms

Axiom (Differential Cut) (CADE'15)

$$\text{DC} \quad ([x' = f(x) \,\&\, q(x)]p(x) \leftrightarrow [x' = f(x) \,\&\, q(x) \wedge r(x)]p(x)) \leftarrow [x' = f(x) \,\&\, q(x)]r(x)$$



DC is a cut for differential equations.

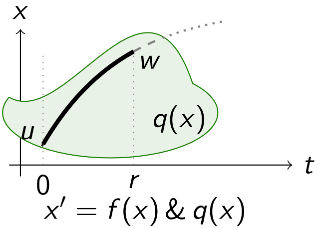
DC is a differential modal modus ponens K.

Can't leave $r(x)$, then might as well restrict state space to $r(x)$.

Differential Equation Axioms

Axiom (Differential Cut) (CADE'15)

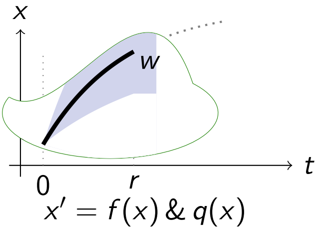
DC $\quad ([x' = f(x) \,\&\, q(x)]p(x) \leftrightarrow [x' = f(x) \,\&\, q(x) \land r(x)]p(x))$
$\leftarrow [x' = f(x) \,\&\, q(x)]r(x)$

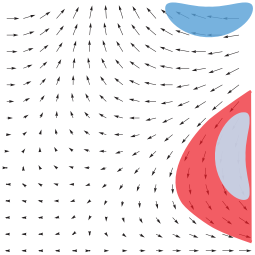


DC is a cut for differential equations.
DC is a differential modal modus ponens K.
Can't leave $r(x)$, then might as well restrict state space to $r(x)$.

Axiom (Differential Cut) (CADE'15)

DC $\quad ([x' = f(x) \,\&\, q(x)]p(x) \leftrightarrow [x' = f(x) \,\&\, q(x) \land r(x)]p(x))$
$\quad \leftarrow [x' = f(x) \,\&\, q(x)]r(x)$

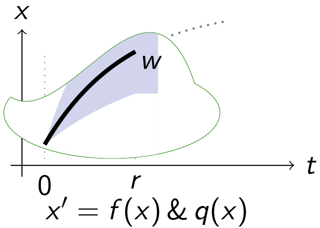


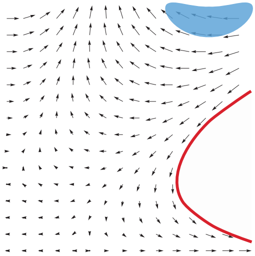


DC is a cut for differential equations.
DC is a differential modal modus ponens K.
Can't leave $r(x)$, then might as well restrict state space to $r(x)$.

Axiom (Differential Cut) (CADE'15)

DC $\quad ([x' = f(x)\,\&\, q(x)]p(x) \leftrightarrow [x' = f(x)\,\&\, q(x) \land r(x)]p(x))$
$\quad \leftarrow [x' = f(x)\,\&\, q(x)]r(x)$

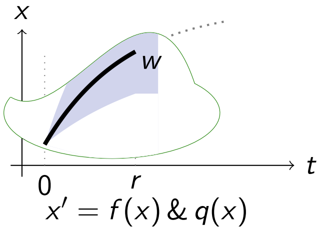


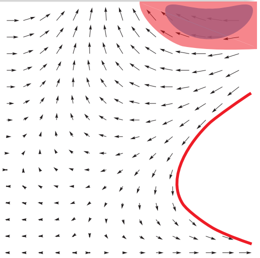


DC is a cut for differential equations.
DC is a differential modal modus ponens K.
Can't leave $r(x)$, then might as well restrict state space to $r(x)$.

Axiom (Differential Cut) (CADE'15)

DC $\quad ([x' = f(x)\,\&\,q(x)]p(x) \leftrightarrow [x' = f(x)\,\&\,q(x) \wedge r(x)]p(x))$
$\leftarrow [x' = f(x)\,\&\,q(x)]r(x)$

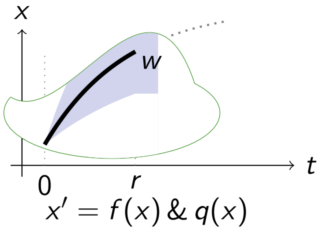


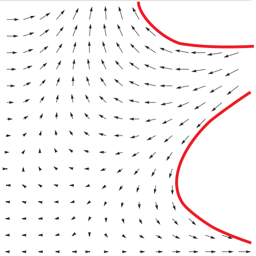


DC is a cut for differential equations.
DC is a differential modal modus ponens K.
Can't leave $r(x)$, then might as well restrict state space to $r(x)$.

Axiom (Differential Cut) (CADE'15)

DC $$([x' = f(x) \,\&\, q(x)]p(x) \leftrightarrow [x' = f(x) \,\&\, q(x) \land r(x)]p(x)) \\ \leftarrow [x' = f(x) \,\&\, q(x)]r(x)$$

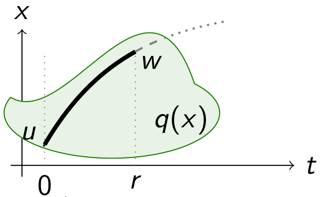




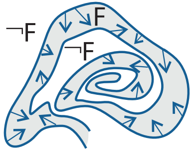
DC is a cut for differential equations.
DC is a differential modal modus ponens K.
Can't leave $r(x)$, then might as well restrict state space to $r(x)$.

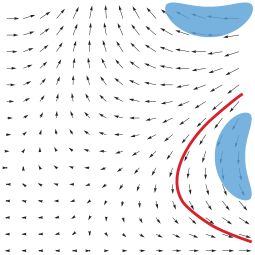
Axiom (Differential Invariant) (CADE'15)

$$\text{DI} \quad [x' = f(x)\,\&\,q(x)]p(x) \leftarrow \big(q(x) \rightarrow p(x) \land [x' = f(x)\,\&\,q(x)](p(x))'\big)$$



$x' = f(x)\,\&\,q(x)$

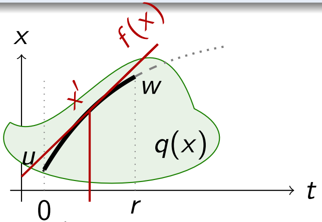




Differential invariant: $p(x)$ true now and its differential $(p(x))'$ true always
What's the differential of a formula???
What's the meaning of a differential term ... in a state???

Axiom (Differential Effect) (CADE'15)

DE $[x' = f(x) \,\&\, q(x)]p(x,x') \leftrightarrow [x' = f(x) \,\&\, q(x)][x' := f(x)]p(x,x')$

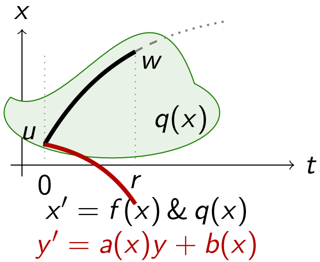


$x' = f(x) \,\&\, q(x)$

Effect of differential equation on differential symbol x'
$[x' := f(x)]$ instantly mimics continuous effect $[x' = f(x)]$ on x'
$[x' := f(x)]$ selects vector field $x' = f(x)$ for subsequent differentials

Axiom (Differential Ghost) (CADE'15)

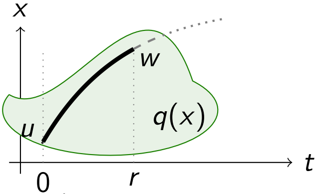
DG $[x' = f(x) \,\&\, q(x)]p(x) \leftrightarrow \exists y\, [x' = f(x), y' = a(x)y + b(x) \,\&\, q(x)]p(x)$



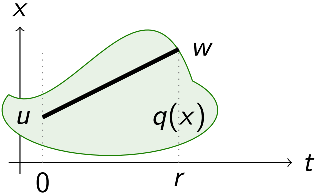
Differential ghost/auxiliaries: extra differential equations that exist
Can cause new invariants
"Dark matter" counterweight to balance conserved quantities

Axiom (Differential Solution) (CADE'15)

DS $[x' = f \,\&\, q(x)]p(x) \leftrightarrow \forall t{\geq}0\, \big((\forall 0{\leq}s{\leq}t\; q(x{+}fs)) \rightarrow [x := x{+}ft]p(x)\big)$



$x' = f(x) \,\&\, q(x)$



$x' = f \,\&\, q(x)$

Differential solutions: solve differential equations with DG,DC and inverse companions

$$\text{DI}\ \dfrac{}{x \cdot x \geq 1 \rightarrow [x' = x^3]\, x \cdot x \geq 1}$$

1. DI proves a property of an ODE inductively by its differentials

$$\dfrac{\dfrac{}{[x' = x^3](x \cdot x \geq 1)'}\;\text{DE}}{x \cdot x \geq 1 \rightarrow [x' = x^3]x \cdot x \geq 1}\;\text{DI}$$

Example: Differential Invariants Don't Solve. Prove!

1. DI proves a property of an ODE inductively by its differentials
2. DE exports vector field, possibly after DW exports evolution domain

$$\dfrac{\dfrac{\dfrac{}{[x' = x^3][x' := x^3](x \cdot x \geq 1)'}\text{CE}}{[x' = x^3](x \cdot x \geq 1)'}\text{DE}}{x \cdot x \geq 1 \rightarrow [x' = x^3]x \cdot x \geq 1}\text{DI}$$

Example: Differential Invariants Don't Solve. Prove!

1. DI proves a property of an ODE inductively by its differentials
2. DE exports vector field, possibly after DW exports evolution domain
3. CE+CQ reason efficiently in Equivalence or eQuational context

$$
\dfrac{\dfrac{\dfrac{\text{G}\dfrac{}{[x'=x^3][x':=x^3]x'\cdot x+x\cdot x'\geq 0} \quad \dfrac{}{(x\cdot x\geq 1)' \leftrightarrow x'\cdot x+x\cdot x'\geq 0}}{[x'=x^3][x':=x^3](x\cdot x\geq 1)'}\,\text{CE}}{[x'=x^3](x\cdot x\geq 1)'}\,\text{DE}}{x\cdot x\geq 1 \rightarrow [x'=x^3]x\cdot x\geq 1}\,\text{DI}
$$

Example: Differential Invariants Don't Solve. Prove!

1. DI proves a property of an ODE inductively by its differentials
2. DE exports vector field, possibly after DW exports evolution domain
3. CE+CQ reason efficiently in Equivalence or eQuational context
4. G isolates postcondition

$$
\text{DI}\;\cfrac{\text{DE}\;\cfrac{\text{CE}\;\cfrac{\text{G}\;\cfrac{[{}'{:=}]\;\cfrac{}{[x' := x^3]x'\cdot x + x\cdot x' \geq 0}}{[x' = x^3][x' := x^3]x'\cdot x + x\cdot x' \geq 0} \qquad \cfrac{}{(x\cdot x \geq 1)' \leftrightarrow x'\cdot x + x\cdot x' \geq 0}}{[x' = x^3][x' := x^3](x\cdot x \geq 1)'}}{[x' = x^3](x\cdot x \geq 1)'}}{x\cdot x \geq 1 \rightarrow [x' = x^3]x\cdot x \geq 1}
$$

Example: Differential Invariants Don't Solve. Prove!

1. DI proves a property of an ODE inductively by its differentials
2. DE exports vector field, possibly after DW exports evolution domain
3. CE+CQ reason efficiently in Equivalence or eQuational context
4. G isolates postcondition
5. [′:=] differential substitution uses vector field

$$
\text{DI}\;\dfrac{\text{DE}\;\dfrac{\text{CE}\;\dfrac{\text{G}\;\dfrac{[\,':=]\;\dfrac{\mathbb{R}\;\dfrac{}{x^3\cdot x + x\cdot x^3 \geq 0}}{[x' := x^3]x'\cdot x + x\cdot x' \geq 0}}{[x' = x^3][x' := x^3]x'\cdot x + x\cdot x' \geq 0} \qquad \dfrac{}{(x\cdot x \geq 1)' \leftrightarrow x'\cdot x + x\cdot x' \geq 0}}{[x' = x^3][x' := x^3](x\cdot x \geq 1)'}}{[x' = x^3](x\cdot x \geq 1)'}}{x\cdot x \geq 1 \rightarrow [x' = x^3]x\cdot x \geq 1}
$$

1. DI proves a property of an ODE inductively by its differentials
2. DE exports vector field, possibly after DW exports evolution domain
3. CE+CQ reason efficiently in Equivalence or eQuational context
4. G isolates postcondition
5. $[':=]$ differential substitution uses vector field

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{*}{x^3 \cdot x + x \cdot x^3 \geq 0}\,\mathbb{R}}{[x' := x^3]x' \cdot x + x \cdot x' \geq 0}\,[':=]}{[x' = x^3][x' := x^3]x' \cdot x + x \cdot x' \geq 0}\,\mathrm{G} \qquad \overline{(x \cdot x \geq 1)' \leftrightarrow x' \cdot x + x \cdot x' \geq 0}}{[x' = x^3][x' := x^3](x \cdot x \geq 1)'}\,\mathrm{CE}}{[x' = x^3](x \cdot x \geq 1)'}\,\mathrm{DE}}{x \cdot x \geq 1 \rightarrow [x' = x^3]x \cdot x \geq 1}\,\mathrm{DI}
$$

Example: Differential Invariants Don't Solve. Prove!

1. DI proves a property of an ODE inductively by its differentials
2. DE exports vector field, possibly after DW exports evolution domain
3. CE+CQ reason efficiently in Equivalence or eQuational context
4. G isolates postcondition
5. [′:=] differential substitution uses vector field

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{*}{x^3\cdot x + x\cdot x^3 \geq 0}\;\mathbb{R}}{[x' := x^3]x'\cdot x + x\cdot x' \geq 0}\;[':=]}{[x' = x^3][x' := x^3]x'\cdot x + x\cdot x' \geq 0}\;\mathrm{G} \qquad \dfrac{\dfrac{}{(x\cdot x)' \geq 0 \leftrightarrow x'\cdot x + x\cdot x' \geq 0}\;\mathrm{CQ}}{(x\cdot x \geq 1)' \leftrightarrow x'\cdot x + x\cdot x' \geq 0}}{[x' = x^3][x' := x^3](x\cdot x \geq 1)'}\;\mathrm{CE}}{[x' = x^3](x\cdot x \geq 1)'}\;\mathrm{DE}}{x\cdot x \geq 1 \rightarrow [x' = x^3]x\cdot x \geq 1}\;\mathrm{DI}$$

1. DI proves a property of an ODE inductively by its differentials
2. DE exports vector field, possibly after DW exports evolution domain
3. CE+CQ reason efficiently in Equivalence or eQuational context
4. G isolates postcondition
5. $[':=]$ differential substitution uses vector field

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{*}{x^3\cdot x + x\cdot x^3 \geq 0}\,\mathbb{R}}{[x' := x^3]x'\cdot x + x\cdot x' \geq 0}\,[':=]}{[x'=x^3][x':=x^3]x'\cdot x+x\cdot x'\geq 0}\,G \qquad \dfrac{\dfrac{\dfrac{}{(x\cdot x)' = x'\cdot x+x\cdot x'}}{(x\cdot x)'\geq 0 \leftrightarrow x'\cdot x+x\cdot x'\geq 0}\,CQ}{(x\cdot x\geq 1)' \leftrightarrow x'\cdot x + x\cdot x' \geq 0}}{[x'=x^3][x':=x^3](x\cdot x\geq 1)'}\,CE}{[x'=x^3](x\cdot x\geq 1)'}\,DE}{x\cdot x\geq 1 \rightarrow [x'=x^3]x\cdot x\geq 1}\,DI$$

1. DI proves a property of an ODE inductively by its differentials
2. DE exports vector field, possibly after DW exports evolution domain
3. CE+CQ reason efficiently in Equivalence or eQuational context
4. G isolates postcondition
5. $[' {:=}]$ differential substitution uses vector field

$$
\text{DI}\;\dfrac{\text{DE}\;\dfrac{\text{CE}\;\dfrac{\text{G}\;\dfrac{[' {:=}]\;\dfrac{\mathbb{R}\;\dfrac{*}{x^3\cdot x + x\cdot x^3 \geq 0}}{[x' := x^3]x'\cdot x + x\cdot x' \geq 0}}{[x' = x^3][x' := x^3]x'\cdot x + x\cdot x' \geq 0} \qquad \dfrac{\text{CQ}\;\dfrac{\dfrac{\text{US}\;\dfrac{}{(x\cdot x)' = (x)'\cdot x + x\cdot (x)'}}{(x\cdot x)' = x'\cdot x + x\cdot x'}}{(x\cdot x)' \geq 0 \leftrightarrow x'\cdot x + x\cdot x' \geq 0}}{(x\cdot x \geq 1)' \leftrightarrow x'\cdot x + x\cdot x' \geq 0}}{[x' = x^3][x' := x^3](x\cdot x \geq 1)'}}{[x' = x^3](x\cdot x \geq 1)'}}{x\cdot x \geq 1 \rightarrow [x' = x^3]x\cdot x \geq 1}
$$

Example: Differential Invariants Don't Solve. Prove!

1. DI proves a property of an ODE inductively by its differentials
2. DE exports vector field, possibly after DW exports evolution domain
3. CE+CQ reason efficiently in Equivalence or eQuational context
4. G isolates postcondition
5. $[':=]$ differential substitution uses vector field
6. $\cdot'$ differential computations are axiomatic (US)

$$
\text{DI}\ \dfrac{\text{DE}\ \dfrac{\text{CE}\ \dfrac{\text{G}\ \dfrac{[':=]\ \dfrac{\mathbb{R}\ \dfrac{*}{x^3\cdot x + x\cdot x^3 \geq 0}}{[x' := x^3]x'\cdot x + x\cdot x' \geq 0}}{[x' = x^3][x' := x^3]x'\cdot x + x\cdot x' \geq 0} \qquad \dfrac{\dfrac{\text{CQ}\ \dfrac{\dfrac{\text{US}\ \dfrac{\cdot'\ \dfrac{}{(f(\bar{x})\cdot g(\bar{x}))' = (f(\bar{x}))'\cdot g(\bar{x}) + f(\bar{x})\cdot(g(\bar{x}))'}}{(x\cdot x)' = (x)'\cdot x + x\cdot(x)'}}{(x\cdot x)' = x'\cdot x + x\cdot x'}}{(x\cdot x)' \geq 0 \leftrightarrow x'\cdot x + x\cdot x' \geq 0}}{(x\cdot x \geq 1)' \leftrightarrow x'\cdot x + x\cdot x' \geq 0}}{}}{[x' = x^3][x' := x^3](x\cdot x \geq 1)'}}{[x' = x^3](x\cdot x \geq 1)'}}{x\cdot x \geq 1 \rightarrow [x' = x^3]x\cdot x \geq 1}
$$

1. DI proves a property of an ODE inductively by its differentials
2. DE exports vector field, possibly after DW exports evolution domain
3. CE+CQ reason efficiently in Equivalence or eQuational context
4. G isolates postcondition
5. $[':=]$ differential substitution uses vector field
6. $\cdot'$ differential computations are axiomatic (US)

$$
\dfrac{
  \dfrac{
    \dfrac{
      \dfrac{
        \dfrac{
          \dfrac{*}{x^3\cdot x + x\cdot x^3 \geq 0}\,\mathbb{R}
        }{[x' := x^3]x'\cdot x + x\cdot x' \geq 0}\,[':=]
      }{[x' = x^3][x' := x^3]x'\cdot x + x\cdot x' \geq 0}\,\mathrm{G}
      \qquad
      \dfrac{
        \dfrac{
          \dfrac{
            \dfrac{
              \dfrac{*}{(f(\bar{x})\cdot g(\bar{x}))' = (f(\bar{x}))'\cdot g(\bar{x}) + f(\bar{x})\cdot (g(\bar{x}))'}\,\cdot'
            }{(x\cdot x)' = (x)'\cdot x + x\cdot (x)'}\,\mathrm{US}
          }{(x\cdot x)' = x'\cdot x + x\cdot x'}
        }{(x\cdot x)' \geq 0 \leftrightarrow x'\cdot x + x\cdot x' \geq 0}\,\mathrm{CQ}
      }{(x\cdot x \geq 1)' \leftrightarrow x'\cdot x + x\cdot x' \geq 0}
    }{[x' = x^3][x' := x^3](x\cdot x \geq 1)'}\,\mathrm{CE}
  }{[x' = x^3](x\cdot x \geq 1)'}\,\mathrm{DE}
}{x\cdot x \geq 1 \rightarrow [x' = x^3]x\cdot x \geq 1}\,\mathrm{DI}
$$

Example: Soundly Solving Differential Equations

1. DG introduces time t, DC cuts solution in, that DI proves and
2. DW exports to postcondition
3. inverse DC removes evolution domain constraints
4. inverse DG removes original ODE
5. DS solves remaining ODE for time

$$*$$

$$\mathbb{R}\ \frac{}{\phi \rightarrow \forall s {\geq} 0\, (x_0 + \frac{a}{2}s^2 + v_0 s \geq 0)}$$

$$[:=]\ \frac{}{\phi \rightarrow \forall s {\geq} 0\, [t := 0 + 1s] x_0 + \frac{a}{2}t^2 + v_0 t \geq 0}$$

$$\text{DS}\ \frac{}{\phi \rightarrow [t' = 1] x_0 + \frac{a}{2}t^2 + v_0 t \geq 0}$$

$$\text{DG}\ \frac{}{\phi \rightarrow [v' = a, t' = 1] x_0 + \frac{a}{2}t^2 + v_0 t \geq 0}$$

$$\text{DG}\ \frac{}{\phi \rightarrow [x' = v, v' = a, t' = 1] x_0 + \frac{a}{2}t^2 + v_0 t \geq 0}$$

$$\text{DC}\ \frac{}{\phi \rightarrow [x' = v, v' = a, t' = 1 \,\&\, v = v_0 + at] x_0 + \frac{a}{2}t^2 + v_0 t \geq 0}$$

$$\text{DC}\ \frac{}{\phi \rightarrow [x' = v, v' = a, t' = 1 \,\&\, v = v_0 + at \land x = x_0 + \frac{a}{2}t^2 + v_0 t] x_0 + \frac{a}{2}t^2 + v_0 t \geq 0}$$

$$\text{G,K}\ \frac{}{\phi \rightarrow [x' = v, v' = a, t' = 1 \,\&\, v = v_0 + at \land x = x_0 + \frac{a}{2}t^2 + v_0 t](x{=}x_0{+}\frac{a}{2}t^2{+}v_0 t \rightarrow x{\geq}0)}$$

$$\text{DW}\ \frac{}{\phi \rightarrow [x' = v, v' = a, t' = 1 \,\&\, v = v_0 + at \land x = x_0 + \frac{a}{2}t^2 + v_0 t] x \geq 0}$$

$$\text{DC}\ \frac{}{\phi \rightarrow [x' = v, v' = a, t' = 1 \,\&\, v = v_0 + at] x \geq 0}$$

$$\text{DC}\ \frac{}{\phi \rightarrow [x' = v, v' = a, t' = 1] x \geq 0}$$

$$\frac{}{\phi \rightarrow \exists t\, [x' = v, v' = a, t' = 1] x \geq 0}$$

$$\text{DG}\ \frac{}{\phi \rightarrow [x' = v, v' = a] x \geq 0}$$

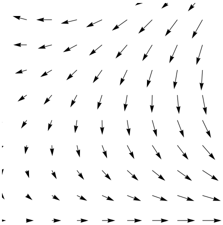
The Meaning of Prime

The Meaning of Prime

Semantics $[\![(\theta)']\!] I u =$

The Meaning of Prime

Semantics $[\![(\theta)']\!] I u =$

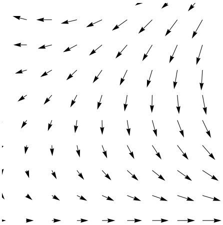


depends on the differential equation?

The Meaning of Prime

Semantics

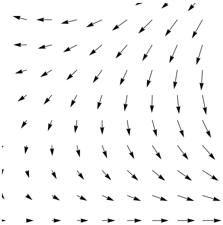
$$[\![(\theta)']\!]Iu =$$



depends on the differential equation?
well-defined in isolated state u at all?

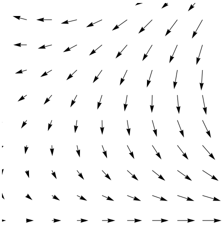
Semantics

$$[\![(\theta)']\!]Iu = \sum_x u(x')\frac{\partial[\![\theta]\!]}{\partial x}(u)$$



depends on the differential equation?
well-defined in isolated state u at all?

Semantics $[\![(\theta)']\!]Iu = \sum_x u(x')\frac{\partial[\![\theta]\!]}{\partial x}(u)$

 $\to \mathbb{R}$

depends on the differential equation?
well-defined in isolated state u at all?

Differential Substitution Lemmas

Lemma (Differential lemma)

If $I, \varphi \models x' = f(x) \land Q$ *for duration* $r > 0$*, then for all* $0 \leq \zeta \leq r$*:*

Syntactic $$[\![(\theta)']\!] I\varphi(\zeta) = \frac{\mathrm{d}[\![\theta]\!] I\varphi(t)}{\mathrm{d}t}(\zeta)$$ Analytic

Lemma (Differential assignment)

If $I, \varphi \models x' = f(x) \land Q$ *then* $I, \varphi \models \phi \leftrightarrow [x' := f(x)]\phi$

Lemma (Derivations)

$$(f(\bar{x}) + g(\bar{x}))' = (f(\bar{x}))' + (g(\bar{x}))'$$

$$(f(\bar{x}) \cdot g(\bar{x}))' = (f(\bar{x}))' \cdot g(\bar{x}) + f(\bar{x}) \cdot (g(\bar{x}))'$$

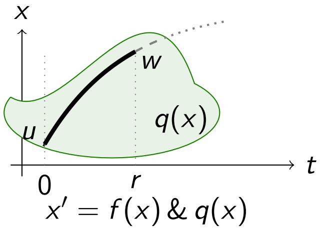
$$[y := f(\bar{x})][y' := 1]\big((f(f(\bar{x})))' = (f(y))' \cdot (f(\bar{x}))'\big) \qquad \textit{for } y, y' \notin f(\bar{x})$$

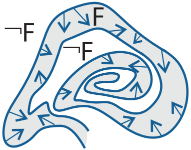
$$(f)' = 0 \qquad \textit{for arity 0 functions/nu}$$

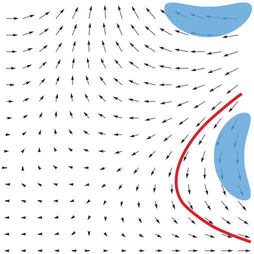
Outline

Axiom (Differential Invariant) (CADE'15)

DI $[x' = f(x) \,\&\, q(x)]p(x) \leftarrow \big(q(x) \rightarrow p(x) \land [x' = f(x) \,\&\, q(x)](p(x))'\big)$







Differential invariant: $p(x)$ true now and its differential $(p(x))'$ true always
What's the differential of a formula???
What's the meaning of a differential term . . . in a state???

Differential-form Differential Dynamic Logic: Syntax

Definition (Hybrid program α)

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

Definition (d$\mathcal{L}$ Formula ϕ)

$$\theta \geq \eta \mid p(\theta_1,\dots,\theta_k) \mid \neg\phi \mid \phi \wedge \psi \mid \forall x\, \phi \mid \exists x\, \phi \mid [\alpha]\phi \mid \langle\alpha\rangle\phi$$

Definition (Term θ)

$$x \mid x' \mid f(\theta_1,\dots,\theta_k) \mid \theta + \eta \mid \theta \cdot \eta \mid (\theta)'$$

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

Definition (Hybrid program α)

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

Definition (d$\mathcal{L}$ Formula ϕ)

$$\theta \geq \eta \mid p(\theta_1, \ldots, \theta_k) \mid \neg\phi \mid \phi \wedge \psi \mid \forall x\, \phi \mid \exists x\, \phi \mid [\alpha]\phi \mid \langle\alpha\rangle\phi$$

Definition (Term θ)

$$x \mid x' \mid f(\theta_1, \ldots, \theta_k) \mid \theta + \eta \mid \theta \cdot \eta \mid (\theta)'$$

All Reals | Some Reals | All Runs | Some Runs

Differential-form Differential Dynamic Logic: Syntax

Program Constant | Discrete Assign | Differential Assign

Definition (Hybrid program α)

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

Definition (d$\mathcal{L}$ Formula ϕ)

$$\theta \geq \eta \mid p(\theta_1, \ldots, \theta_k) \mid \neg\phi \mid \phi \wedge \psi \mid \forall x\, \phi \mid \exists x\, \phi \mid [\alpha]\phi \mid \langle\alpha\rangle\phi$$

Definition (Term θ)

$$x \mid x' \mid f(\theta_1, \ldots, \theta_k) \mid \theta + \eta \mid \theta \cdot \eta \mid (\theta)'$$

Differential Symbol | Differential

Differential-form Differential Dynamic Logic: Semantics

Definition (Term semantics) $(\llbracket\cdot\rrbracket : \mathrm{Trm} \to (\mathcal{S} \to \mathbb{R}))$

$$\llbracket(\theta)'\rrbracket Iu \;=\; \sum_x u(x')\frac{\partial\llbracket\theta\rrbracket I}{\partial x}(u) = \sum_x u(x')\frac{\partial\llbracket\theta\rrbracket Iu_x^X}{\partial X}$$

Definition (d$\mathcal{L}$ semantics) $(\llbracket\cdot\rrbracket : \mathrm{Fml} \to \wp(\mathcal{S}))$

$$\begin{aligned}
\llbracket C(\phi)\rrbracket I &= I(C)\big(\llbracket\phi\rrbracket I\big)\\
\llbracket\langle\alpha\rangle\phi\rrbracket I &= \llbracket\alpha\rrbracket I \circ \llbracket\phi\rrbracket I\\
\llbracket[\alpha]\phi\rrbracket I &= \llbracket\neg\langle\alpha\rangle\neg\phi\rrbracket I
\end{aligned}$$

Definition (Program semantics) $(\llbracket\cdot\rrbracket : \mathrm{HP} \to \wp(\mathcal{S}\times\mathcal{S}))$

$$\begin{aligned}
\llbracket x' = f(x)\,\&\,Q\rrbracket I &= \{(\varphi(0)|_{\{x'\}^\complement}, \varphi(r)) \;:\; I,\varphi \models x' = f(x) \wedge Q\}\\
\llbracket\alpha\cup\beta\rrbracket I &= \llbracket\alpha\rrbracket I \cup \llbracket\beta\rrbracket I\\
\llbracket\alpha;\beta\rrbracket I &= \llbracket\alpha\rrbracket I \circ \llbracket\beta\rrbracket I\\
\llbracket\alpha^*\rrbracket I &= \big(\llbracket\alpha\rrbracket I\big)^* = \bigcup_{n\in\mathbb{N}} \llbracket\alpha^n\rrbracket I
\end{aligned}$$

Differential Substitution Lemmas

Lemma (Differential lemma)

If $I, \varphi \models x' = f(x) \land Q$ *for duration* $r > 0$*, then for all* $0 \leq \zeta \leq r$*:*

Syntactic $$[\![(\eta)']\!] I\varphi(\zeta) = \frac{\mathrm{d}[\![\eta]\!] I\varphi(t)}{\mathrm{d}t}(\zeta)$$ Analytic

Lemma (Differential assignment)

If $I, \varphi \models x' = f(x) \land Q$ *then* $I, \varphi \models \phi \leftrightarrow [x' := f(x)]\phi$

Lemma (Derivations)

$$\begin{aligned}
(\theta + \eta)' &= (\theta)' + (\eta)' \\
(\theta \cdot \eta)' &= (\theta)' \cdot \eta + \theta \cdot (\eta)' \\
[y := \theta][y' := 1]\big((f(\theta))' &= (f(y))' \cdot (\theta)'\big) && \textit{for } y, y' \notin \theta \\
(f)' &= 0 && \textit{for arity 0 functions/numbers } f
\end{aligned}$$

Differential Equation Axioms & Differential Axioms

DW $[x' = f(x) \,\&\, q(x)]q(x)$

DC $\big([x' = f(x) \,\&\, q(x)]p(x) \leftrightarrow [x' = f(x) \,\&\, q(x) \land r(x)]p(x)\big)$
$\leftarrow [x' = f(x) \,\&\, q(x)]r(x)$

DE $[x' = f(x) \,\&\, q(x)]p(x, x') \leftrightarrow [x' = f(x) \,\&\, q(x)][x' := f(x)]p(x, x')$

DI $[x' = f(x) \,\&\, q(x)]p(x) \leftarrow \big(q(x) \to p(x) \land [x' = f(x) \,\&\, q(x)](p(x))'\big)$

DG $[x' = f(x) \,\&\, q(x)]p(x) \leftrightarrow \exists y \, [x' = f(x), y' = a(x)y + b(x) \,\&\, q(x)]p(x)$

DS $[x' = f \,\&\, q(x)]p(x) \leftrightarrow \forall t {\geq} 0 \, \big((\forall 0 {\leq} s {\leq} t \; q(x{+}fs)) \to [x := x{+}ft]p(x)\big)$

$[' {:=}]$ $[x' := f]p(x') \leftrightarrow p(f)$

$+'$ $(f(\bar{x}) + g(\bar{x}))' = (f(\bar{x}))' + (g(\bar{x}))'$

$\cdot'$ $(f(\bar{x}) \cdot g(\bar{x}))' = (f(\bar{x}))' \cdot g(\bar{x}) + f(\bar{x}) \cdot (g(\bar{x}))'$

$\circ'$ $[y := g(x)][y' := 1]\big((f(g(x)))' = (f(y))' \cdot (g(x))'\big)$

CADE'15

Differential Dynamic Logic: Axioms

$$\text{G}\ \frac{p(\bar{x})}{[a]p(\bar{x})}$$

$$\forall\ \frac{p(x)}{\forall x\, p(x)}$$

$$\text{MP}\ \frac{p \rightarrow q \quad p}{q}$$

$$\text{CT}\ \frac{f(\bar{x}) = g(\bar{x})}{c(f(\bar{x})) = c(g(\bar{x}))}$$

$$\text{CQ}\ \frac{f(\bar{x}) = g(\bar{x})}{p(f(\bar{x})) \leftrightarrow p(g(\bar{x}))}$$

$$\text{CE}\ \frac{p(\bar{x}) \leftrightarrow q(\bar{x})}{C(p(\bar{x})) \leftrightarrow C(q(\bar{x}))}$$

CADE'15

1. DI proves a property of an ODE inductively by its differentials
2. DE exports vector field, possibly after DW exports evolution domain
3. CE+CQ reason efficiently in Equivalence or eQuational context
4. G isolates postcondition
5. $[':=]$ differential substitution uses vector field

$$\text{DI}\;\dfrac{\text{DE}\;\dfrac{\text{CE}\;\dfrac{\text{G}\;\dfrac{[':=]\;\dfrac{\mathbb{R}\;\dfrac{*}{x^3\cdot x + x\cdot x^3 \geq 0}}{[x' := x^3]x'\cdot x + x\cdot x' \geq 0}}{[x' = x^3][x' := x^3]x'\cdot x + x\cdot x' \geq 0} \qquad \dfrac{\text{CQ}\;\dfrac{\dfrac{}{(x\cdot x)' = x'\cdot x + x\cdot x'}}{(x\cdot x)' \geq 0 \leftrightarrow x'\cdot x + x\cdot x' \geq 0}}{(x\cdot x \geq 1)' \leftrightarrow x'\cdot x + x\cdot x' \geq 0}}{[x' = x^3][x' := x^3](x\cdot x \geq 1)'}}{[x' = x^3](x\cdot x \geq 1)'}}{x\cdot x \geq 1 \rightarrow [x' = x^3]x\cdot x \geq 1}$$

Example: Syntactic Contextual Congruence by US

$$\text{CQ}\ \frac{f(\bar{x}) = g(\bar{x})}{p(f(\bar{x})) \leftrightarrow p(g(\bar{x}))}$$

$$\text{CQ}\ \frac{(x \cdot x)' = x' \cdot x + x \cdot x'}{(x \cdot x)' \geq 0 \leftrightarrow x' \cdot x + x \cdot x' \geq 0}$$

$$\text{CE}\ \frac{p(\bar{x}) \leftrightarrow q(\bar{x})}{C(p(\bar{x})) \leftrightarrow C(q(\bar{x}))}$$

$$\text{CE}\ \frac{(x \cdot x \geq 1)' \leftrightarrow x' \cdot x + x \cdot x' \geq 0}{[x' = x^3][x' := x^3](x \cdot x \geq 1)' \leftrightarrow [x' = x^3][x' := x^3]x' \cdot x + x \cdot x' \geq 0}$$

Example: Syntactic Contextual Congruence by US

$$\text{CQ}\ \frac{f(\bar{x}) = g(\bar{x})}{p(f(\bar{x})) \leftrightarrow p(g(\bar{x}))}$$

$$\text{CQ}\ \frac{(x \cdot x)' = x' \cdot x + x \cdot x'}{(x \cdot x)' \geq 0 \leftrightarrow x' \cdot x + x \cdot x' \geq 0}$$

with $\sigma \approx p(\cdot) \mapsto \cdot \geq 0, f(\cdot) \mapsto ((\cdot) \cdot (\cdot))', g(\cdot) \mapsto (\cdot') \cdot (\cdot) + (\cdot) \cdot (\cdot')$

$$\text{CE}\ \frac{p(\bar{x}) \leftrightarrow q(\bar{x})}{C(p(\bar{x})) \leftrightarrow C(q(\bar{x}))}$$

$$\text{CE}\ \frac{(x \cdot x \geq 1)' \leftrightarrow x' \cdot x + x \cdot x' \geq 0}{[x' = x^3][x' := x^3](x \cdot x \geq 1)' \leftrightarrow [x' = x^3][x' := x^3]x' \cdot x + x \cdot x' \geq 0}$$

with
$\sigma \approx C(\_) \mapsto [x' = x^3][x' := x^3]\_, p(\bar{x}) \mapsto ((\cdot)(\cdot) \geq 1)', q(\bar{x}) \mapsto \cdot' \cdot + \cdot \cdot'$

Example: Syntactic Contextual Congruence by US

$$\text{CQ}\ \frac{f(\bar{x}) = g(\bar{x})}{p(f(\bar{x})) \leftrightarrow p(g(\bar{x}))}$$

$$\text{CQ}\ \frac{(x \cdot x)' = x' \cdot x + x \cdot x'}{(x \cdot x)' \geq 0 \leftrightarrow x' \cdot x + x \cdot x' \geq 0}$$

with $\sigma \approx p(\cdot) \mapsto \cdot \geq 0, f(\bar{x}) \mapsto (x \cdot x)', g(\bar{x}) \mapsto x' \cdot x + x \cdot x'$

$$\text{CE}\ \frac{p(\bar{x}) \leftrightarrow q(\bar{x})}{C(p(\bar{x})) \leftrightarrow C(q(\bar{x}))}$$

$$\text{CE}\ \frac{(x \cdot x \geq 1)' \leftrightarrow x' \cdot x + x \cdot x' \geq 0}{[x' = x^3][x' := x^3](x \cdot x \geq 1)' \leftrightarrow [x' = x^3][x' := x^3]x' \cdot x + x \cdot x' \geq 0}$$

with $\sigma \approx C(\_) \mapsto [x' = x^3] p(\bar{x}) \mapsto (x \cdot x \geq 1)', q(\bar{x}) \mapsto x' \cdot x + x \cdot x' \geq 0$

Example: Differential Invariants

$$\text{CE}\ \overline{\qquad [x' = x^3][x' := x^3](x \cdot x \geq 1)' \qquad}$$

$$\text{DE}\ \overline{\qquad [x' = x^3](x \cdot x \geq 1)' \qquad}$$

$$\text{DI}\ \overline{\qquad x \cdot x \geq 1 \rightarrow [x' = x^3] x \cdot x \geq 1 \qquad}$$

1. Free function $j(x,x')$ for parametric differential computation

$$
\dfrac{\dfrac{\dfrac{\text{G}\ \dfrac{}{[x'=x^3][x':=x^3]j(x,x')\geq 0} \qquad \dfrac{}{(x\cdot x\geq 1)'\leftrightarrow j(x,x')\geq 0}}{[x'=x^3][x':=x^3](x\cdot x\geq 1)'}\ \text{CE}}{[x'=x^3](x\cdot x\geq 1)'}\ \text{DE}}{x\cdot x\geq 1\rightarrow [x'=x^3]x\cdot x\geq 1}\ \text{DI}
$$

1. Free function $j(x, x')$ for parametric differential computation
2. Again G,$[':=]$ to isolate differentially substituted postcondition

$$\text{DI}\,\dfrac{\text{DE}\,\dfrac{\text{CE}\,\dfrac{\text{G}\,\dfrac{[':=]\,\dfrac{}{[x' := x^3]j(x,x') \geq 0}}{[x' = x^3][x' := x^3]j(x,x') \geq 0} \qquad (x\cdot x \geq 1)' \leftrightarrow j(x,x') \geq 0}{[x' = x^3][x' := x^3](x\cdot x \geq 1)'}}{[x' = x^3](x\cdot x \geq 1)'}}{x\cdot x \geq 1 \rightarrow [x' = x^3]x\cdot x \geq 1}$$

1. Free function $j(x,x')$ for parametric differential computation
2. Again G,$[':=]$ to isolate differentially substituted postcondition

$$
\dfrac{
  \dfrac{
    \dfrac{
      \dfrac{j(x,x^3) \geq 0}{[x' := x^3]j(x,x') \geq 0}\;[':=]
    }{[x' = x^3][x' := x^3]j(x,x') \geq 0}\;\mathrm{G}
    \qquad
    \dfrac{}{(x\cdot x \geq 1)' \leftrightarrow j(x,x') \geq 0}
  }{[x' = x^3][x' := x^3](x\cdot x \geq 1)'}\;\mathrm{CE}
}{
  \dfrac{[x' = x^3](x\cdot x \geq 1)'}{x\cdot x \geq 1 \rightarrow [x' = x^3]x\cdot x \geq 1}\;\mathrm{DI}
}\;\mathrm{DE}
$$

1. Free function $j(x, x')$ for parametric differential computation
2. Again G,[$'$:=] to isolate differentially substituted postcondition
3. Construct parametric $j(x, x')$ by axiomatic differential computation

$$
\text{DI}\;\dfrac{\text{DE}\;\dfrac{\text{CE}\;\dfrac{\text{G}\;\dfrac{[':=]\;\dfrac{j(x,x^3)\geq 0}{[x' := x^3]j(x,x')\geq 0}}{[x'=x^3][x':=x^3]j(x,x')\geq 0}\qquad \text{CQ}\;\dfrac{\dfrac{}{(x\cdot x)'\geq 0 \leftrightarrow j(x,x')\geq 0}}{(x\cdot x\geq 1)'\leftrightarrow j(x,x')\geq 0}}{[x'=x^3][x':=x^3](x\cdot x\geq 1)'}}{[x'=x^3](x\cdot x\geq 1)'}}{x\cdot x\geq 1 \rightarrow [x'=x^3]x\cdot x\geq 1}
$$

Example: Differential Invariants — Parametric

1. Free function $j(x,x')$ for parametric differential computation
2. Again G,$[':=]$ to isolate differentially substituted postcondition
3. Construct parametric $j(x,x')$ by axiomatic differential computation

$$
\text{DI}\;\dfrac{\text{DE}\;\dfrac{\text{CE}\;\dfrac{\text{G}\;\dfrac{[':=]\;\dfrac{j(x,x^3)\geq 0}{[x':=x^3]j(x,x')\geq 0}}{[x'=x^3][x':=x^3]j(x,x')\geq 0}\qquad \dfrac{\text{CQ}\;\dfrac{(x\cdot x)'=j(x,x')}{(x\cdot x)'\geq 0 \leftrightarrow j(x,x')\geq 0}}{(x\cdot x\geq 1)'\leftrightarrow j(x,x')\geq 0}}{[x'=x^3][x':=x^3](x\cdot x\geq 1)'}}{[x'=x^3](x\cdot x\geq 1)'}}{x\cdot x\geq 1 \rightarrow [x'=x^3]x\cdot x\geq 1}
$$

1. Free function $j(x,x')$ for parametric differential computation
2. Again G,$[':=]$ to isolate differentially substituted postcondition
3. Construct parametric $j(x,x')$ by axiomatic differential computation
4. USR instantiates proof by $\{j(x,x') \mapsto x' \cdot x + x \cdot x'\}$

$$
\dfrac{\dfrac{\dfrac{j(x,x^3)\geq 0}{[x':=x^3]j(x,x')\geq 0}\,[':=]}{[x'=x^3][x':=x^3]j(x,x')\geq 0}\,\mathrm{G} \qquad \dfrac{\dfrac{(x\cdot x)'=j(x,x')}{(x\cdot x)'\geq 0 \leftrightarrow j(x,x')\geq 0}\,\mathrm{CQ}}{(x\cdot x\geq 1)' \leftrightarrow j(x,x')\geq 0}}{\dfrac{\dfrac{[x'=x^3][x':=x^3](x\cdot x\geq 1)'}{[x'=x^3](x\cdot x\geq 1)'}\,\mathrm{DE}}{x\cdot x\geq 1 \rightarrow [x'=x^3]x\cdot x\geq 1}\,\mathrm{DI}}\,\mathrm{CE}
$$

$$
\dfrac{\dfrac{}{x^3\cdot x + x\cdot x^3 \geq 0}\,\mathbb{R} \qquad \dfrac{}{(x\cdot x)' = x'\cdot x + x\cdot x'}\,x'}{x\cdot x\geq 1 \rightarrow [x'=x^3]x\cdot x\geq 1}\,\mathrm{USR}
$$

1. Free function $j(x,x')$ for parametric differential computation
2. Again G,[$'$:=] to isolate differentially substituted postcondition
3. Construct parametric $j(x,x')$ by axiomatic differential computation
4. USR instantiates proof by $\{j(x,x') \mapsto x' \cdot x + x \cdot x'\}$

$$\cfrac{\cfrac{\cfrac{j(x,x^3) \geq 0}{[x' := x^3]j(x,x') \geq 0}\,['{:=}]}{[x' = x^3][x' := x^3]j(x,x') \geq 0}\,\mathrm{G} \qquad \cfrac{\cfrac{(x\cdot x)' = j(x,x')}{(x\cdot x)' \geq 0 \leftrightarrow j(x,x') \geq 0}\,\mathrm{CQ}}{(x\cdot x \geq 1)' \leftrightarrow j(x,x') \geq 0}}{\cfrac{\cfrac{[x' = x^3][x' := x^3](x\cdot x \geq 1)'}{[x' = x^3](x\cdot x \geq 1)'}\,\mathrm{DE}}{x\cdot x \geq 1 \to [x' = x^3]x\cdot x \geq 1}\,\mathrm{DI}}\,\mathrm{CE}$$

$$\cfrac{\cfrac{*}{x^3\cdot x + x\cdot x^3 \geq 0}\,\mathbb{R} \qquad \cfrac{}{(x\cdot x)' = x'\cdot x + x\cdot x'}\,x'}{x\cdot x \geq 1 \to [x' = x^3]x\cdot x \geq 1}\,\mathrm{USR}$$

Example: Differential Invariants

1. Free function $j(x,x')$ for parametric differential computation
2. Again G,$[':=]$ to isolate differentially substituted postcondition
3. Construct parametric $j(x,x')$ by axiomatic differential computation
4. USR instantiates proof by $\{j(x,x') \mapsto x' \cdot x + x \cdot x'\}$

$$\text{DI}\;\dfrac{\text{DE}\;\dfrac{\text{CE}\;\dfrac{\text{G}\;\dfrac{[':=]\;\dfrac{j(x,x^3) \geq 0}{[x' := x^3]j(x,x') \geq 0}}{[x' = x^3][x' := x^3]j(x,x') \geq 0} \qquad \dfrac{\text{CQ}\;\dfrac{(x\cdot x)' = j(x,x')}{(x\cdot x)' \geq 0 \leftrightarrow j(x,x') \geq 0}}{(x\cdot x \geq 1)' \leftrightarrow j(x,x') \geq 0}}{[x' = x^3][x' := x^3](x\cdot x \geq 1)'}}{[x' = x^3](x\cdot x \geq 1)'}}{x\cdot x \geq 1 \to [x' = x^3]x\cdot x \geq 1}$$

$$\text{USR}\;\dfrac{\mathbb{R}\;\dfrac{*}{x^3\cdot x + x\cdot x^3 \geq 0} \qquad x'\;\dfrac{\text{US}\;\dfrac{}{(x\cdot x)' = (x)'\cdot x + x\cdot(x)'}}{(x\cdot x)' = x'\cdot x + x\cdot x'}}{x\cdot x \geq 1 \to [x' = x^3]x\cdot x \geq 1}$$

1. Free function $j(x,x')$ for parametric differential computation
2. Again G,$[':=]$ to isolate differentially substituted postcondition
3. Construct parametric $j(x,x')$ by axiomatic differential computation
4. USR instantiates proof by $\{j(x,x') \mapsto x'\cdot x + x\cdot x'\}$

$$
\text{DI}\,\dfrac{\text{DE}\,\dfrac{\text{CE}\,\dfrac{\text{G}\,\dfrac{[':=]\,\dfrac{j(x,x^3)\geq 0}{[x':=x^3]j(x,x')\geq 0}}{[x'=x^3][x':=x^3]j(x,x')\geq 0} \qquad \dfrac{\text{CQ}\,\dfrac{(x\cdot x)' = j(x,x')}{(x\cdot x)'\geq 0 \leftrightarrow j(x,x')\geq 0}}{(x\cdot x\geq 1)' \leftrightarrow j(x,x')\geq 0}}{[x'=x^3][x':=x^3](x\cdot x\geq 1)'}}{[x'=x^3](x\cdot x\geq 1)'}}{x\cdot x\geq 1 \rightarrow [x'=x^3]x\cdot x\geq 1}
$$

$$
\text{USR}\,\dfrac{\mathbb{R}\,\dfrac{*}{x^3\cdot x + x\cdot x^3\geq 0} \qquad x'\,\dfrac{\text{US}\,\dfrac{\cdot'\,\dfrac{}{(f(\bar{x})\cdot g(\bar{x}))' = (f(\bar{x}))'\cdot g(\bar{x}) + f(\bar{x})\cdot(g(\bar{x}))'}}{(x\cdot x)' = (x)'\cdot x + x\cdot(x)'}}{(x\cdot x)' = x'\cdot x + x\cdot x'}}{x\cdot x\geq 1 \rightarrow [x'=x^3]x\cdot x\geq 1}
$$

1. Free function $j(x,x')$ for parametric differential computation
2. Again G,$[':=]$ to isolate differentially substituted postcondition
3. Construct parametric $j(x,x')$ by axiomatic differential computation
4. USR instantiates proof by $\{j(x,x') \mapsto x'\cdot x + x\cdot x'\}$

$$
\cfrac{\cfrac{\cfrac{\cfrac{j(x,x^3)\geq 0}{[x':=x^3]j(x,x')\geq 0}\,[':=]}{[x'=x^3][x':=x^3]j(x,x')\geq 0}\,\mathrm{G} \qquad \cfrac{\cfrac{(x{\cdot}x)'=j(x,x')}{(x{\cdot}x)'\geq 0 \leftrightarrow j(x,x')\geq 0}\,\mathrm{CQ}}{(x{\cdot}x\geq 1)' \leftrightarrow j(x,x')\geq 0}}{[x'=x^3][x':=x^3](x{\cdot}x\geq 1)'}\,\mathrm{CE}}{\cfrac{[x'=x^3](x{\cdot}x\geq 1)'}{x{\cdot}x\geq 1 \to [x'=x^3]x{\cdot}x\geq 1}\,\mathrm{DI}}\,\mathrm{DE}
$$

$$
\cfrac{\cfrac{*}{x^3{\cdot}x+x{\cdot}x^3\geq 0}\,\mathbb{R} \qquad \cfrac{\cfrac{\cfrac{*}{(f(\bar{x}){\cdot}g(\bar{x}))'=(f(\bar{x}))'{\cdot}g(\bar{x})+f(\bar{x}){\cdot}(g(\bar{x}))'}\,\cdot'}{(x{\cdot}x)'=(x)'{\cdot}x+x{\cdot}(x)'}\,\mathrm{US}}{(x{\cdot}x)'=x'{\cdot}x+x{\cdot}x'}\,x'}{x{\cdot}x\geq 1 \to [x'=x^3]x{\cdot}x\geq 1}\,\mathrm{USR}
$$

$$\dfrac{\dfrac{\dfrac{}{[x' = x^3][x' := x^3](x\cdot x \geq 1)'}\text{CE}}{[x' = x^3](x\cdot x \geq 1)'}\text{DE}}{x\cdot x \geq 1 \rightarrow [x' = x^3]x\cdot x \geq 1}\text{DI}$$

1. Start with identity differential computation result

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\quad(x\cdot x)' = (x\cdot x)'\quad}\;\mathbb{R}}{}\;.'}{}\;x'}{}\;\text{CT}}{[x' = x^3][x' := x^3](x\cdot x \geq 1)'}\;\text{CE}}{[x' = x^3](x\cdot x \geq 1)'}\;\text{DE}
$$

$$
\dfrac{[x' = x^3](x\cdot x \geq 1)'}{x\cdot x \geq 1 \rightarrow [x' = x^3]x\cdot x \geq 1}\;\text{DI}
$$

1. Start with identity differential computation result which proves

$$
\cfrac{\mathrm{CE}\;\cfrac{}{[x'=x^3][x':=x^3](x\cdot x\geq 1)'}}{\mathrm{DE}\;\cfrac{[x'=x^3](x\cdot x\geq 1)'}{\mathrm{DI}\;\; x\cdot x\geq 1 \rightarrow [x'=x^3]x\cdot x\geq 1}}
$$

$$
\cfrac{\mathrm{CT}\;\cfrac{x'\;\cfrac{.'\;\cfrac{\mathbb{R}\;\cfrac{*}{(x\cdot x)'=(x\cdot x)'}}{}}{}}{}}{}
$$

1. Start with identity differential computation result which proves
2. Construct differential computation result forward by $\cdot'$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{*}{(x\cdot x)' = (x\cdot x)'}\ \mathbb{R}}{(x\cdot x)' = (x)'\cdot x + x\cdot(x)'}\ \cdot'}{}\ x'}{}\ \mathrm{CT}}{}$$

$$\dfrac{\dfrac{\dfrac{}{[x' = x^3][x' := x^3](x\cdot x \geq 1)'}\ \mathrm{CE}}{[x' = x^3](x\cdot x \geq 1)'}\ \mathrm{DE}}{x\cdot x \geq 1 \rightarrow [x' = x^3]x\cdot x \geq 1}\ \mathrm{DI}$$

1. Start with identity differential computation result which proves
2. Construct differential computation result forward by $\cdot'$ x'

$$
\dfrac{
  \dfrac{
    \dfrac{
      \dfrac{
        \dfrac{*}{(x\cdot x)' = (x\cdot x)'}\;\mathbb{R}
      }{(x\cdot x)' = (x)'\cdot x + x\cdot (x)'}\;\cdot'
    }{(x\cdot x)' = x'\cdot x + x\cdot x'}\;x'
  }{}\;\mathrm{CT}
}{}
$$

$$
\dfrac{
  \dfrac{
    \dfrac{}{[x' = x^3][x' := x^3](x\cdot x \geq 1)'}\;\mathrm{CE}
  }{[x' = x^3](x\cdot x \geq 1)'}\;\mathrm{DE}
}{x\cdot x \geq 1 \rightarrow [x' = x^3]x\cdot x \geq 1}\;\mathrm{DI}
$$

1. Start with identity differential computation result which proves
2. Construct differential computation result forward by $\cdot'$ x'
3. Embed differential computation result forward by CT

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{*}{(x\cdot x)' = (x\cdot x)'}\,\mathbb{R}}{(x\cdot x)' = (x)'\cdot x + x\cdot(x)'}\,\cdot'}{(x\cdot x)' = x'\cdot x + x\cdot x'}\,x'}{(x\cdot x)' \geq 0 \leftrightarrow x'\cdot x + x\cdot x' \geq 0}\,\mathrm{CT}}{}
$$

$$
\dfrac{\dfrac{\dfrac{}{[x' = x^3][x' := x^3](x\cdot x \geq 1)'}\,\mathrm{CE}}{[x' = x^3](x\cdot x \geq 1)'}\,\mathrm{DE}}{x\cdot x \geq 1 \rightarrow [x' = x^3]x\cdot x \geq 1}\,\mathrm{DI}
$$

1. Start with identity differential computation result which proves
2. Construct differential computation result forward by $\cdot'$ x'
3. Embed differential computation result forward by CT
4. Construct differential invariant computation result forward accordingly

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{*}{(x\cdot x)' = (x\cdot x)'}\,\mathbb{R}}{(x\cdot x)' = (x)'\cdot x + x\cdot(x)'}\,\cdot'}{(x\cdot x)' = x'\cdot x + x\cdot x'}\,x'}{(x\cdot x)' \geq 0 \leftrightarrow x'\cdot x + x\cdot x' \geq 0}\,\mathrm{CT}}{(x\cdot x \geq 1)' \leftrightarrow x'\cdot x + x\cdot x' \geq 0}}{[x' = x^3][x' := x^3](x\cdot x \geq 1)'}\,\mathrm{CE}}{[x' = x^3](x\cdot x \geq 1)'}\,\mathrm{DE}}{x\cdot x \geq 1 \rightarrow [x' = x^3]x\cdot x \geq 1}\,\mathrm{DI}
$$

1. Start with identity differential computation result which proves
2. Construct differential computation result forward by $\cdot'$ x'
3. Embed differential computation result forward by CT
4. Construct differential invariant computation result forward accordingly
5. Resume backward proof with result computed by forward proof right

$$
\dfrac{\dfrac{\dfrac{[x'=x^3][x':=x^3]x'\cdot x+x\cdot x'\geq 0 \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{*}{(x\cdot x)'=(x\cdot x)'}\mathbb{R}}{(x\cdot x)'=(x)'\cdot x+x\cdot(x)'}\cdot'}{(x\cdot x)'=x'\cdot x+x\cdot x'}x'}{(x\cdot x)'\geq 0 \leftrightarrow x'\cdot x+x\cdot x'\geq 0}\mathrm{CT}}{(x\cdot x\geq 1)'\leftrightarrow x'\cdot x+x\cdot x'\geq 0}}{[x'=x^3][x':=x^3](x\cdot x\geq 1)'}\mathrm{G},\mathrm{CE}}{[x'=x^3](x\cdot x\geq 1)'}\mathrm{DE}}{x\cdot x\geq 1\rightarrow[x'=x^3]x\cdot x\geq 1}\mathrm{DI}
$$

1. Start with identity differential computation result which proves
2. Construct differential computation result forward by $\cdot'$ x'
3. Embed differential computation result forward by CT
4. Construct differential invariant computation result forward accordingly
5. Resume backward proof with result computed by forward proof right

$$
\mathrm{DI}\,\dfrac{\mathrm{DE}\,\dfrac{\mathrm{CE}\,\dfrac{\mathrm{G}\,\dfrac{[{}':=]\,\dfrac{}{[x':=x^3]x'\cdot x+x\cdot x'\geq 0}}{[x'=x^3][x':=x^3]x'\cdot x+x\cdot x'\geq 0}\qquad \dfrac{\mathrm{CT}\,\dfrac{x'\,\dfrac{\cdot'\,\dfrac{\mathbb{R}\,\dfrac{*}{(x\cdot x)'=(x\cdot x)'}}{(x\cdot x)'=(x)'\cdot x+x\cdot(x)'}}{(x\cdot x)'=x'\cdot x+x\cdot x'}}{(x\cdot x)'\geq 0\leftrightarrow x'\cdot x+x\cdot x'\geq 0}}{(x\cdot x\geq 1)'\leftrightarrow x'\cdot x+x\cdot x'\geq 0}}{[x'=x^3][x':=x^3](x\cdot x\geq 1)'}}{[x'=x^3](x\cdot x\geq 1)'}}{x\cdot x\geq 1\rightarrow[x'=x^3]x\cdot x\geq 1}
$$

1. Start with identity differential computation result which proves
2. Construct differential computation result forward by $\cdot'$ x'
3. Embed differential computation result forward by CT
4. Construct differential invariant computation result forward accordingly
5. Resume backward proof with result computed by forward proof right

$$
\begin{prooftree}
\AxiomC{}
\RightLabel{$\mathbb{R}$}
\UnaryInfC{$x^3 \cdot x + x \cdot x^3 \geq 0$}
\RightLabel{$[':=]$}
\UnaryInfC{$[x' := x^3] x' \cdot x + x \cdot x' \geq 0$}
\RightLabel{G}
\UnaryInfC{$[x' = x^3][x' := x^3] x' \cdot x + x \cdot x' \geq 0$}
\AxiomC{$*$}
\RightLabel{$\mathbb{R}$}
\UnaryInfC{$(x \cdot x)' = (x \cdot x)'$}
\RightLabel{$\cdot'$}
\UnaryInfC{$(x \cdot x)' = (x)' \cdot x + x \cdot (x)'$}
\RightLabel{x'}
\UnaryInfC{$(x \cdot x)' = x' \cdot x + x \cdot x'$}
\RightLabel{CT}
\UnaryInfC{$(x \cdot x)' \geq 0 \leftrightarrow x' \cdot x + x \cdot x' \geq 0$}
\UnaryInfC{$(x \cdot x \geq 1)' \leftrightarrow x' \cdot x + x \cdot x' \geq 0$}
\RightLabel{CE}
\BinaryInfC{$[x' = x^3][x' := x^3](x \cdot x \geq 1)'$}
\RightLabel{DE}
\UnaryInfC{$[x' = x^3](x \cdot x \geq 1)'$}
\RightLabel{DI}
\UnaryInfC{$x \cdot x \geq 1 \rightarrow [x' = x^3] x \cdot x \geq 1$}
\end{prooftree}
$$

1. Start with identity differential computation result which proves
2. Construct differential computation result forward by $\cdot'$ x'
3. Embed differential computation result forward by CT
4. Construct differential invariant computation result forward accordingly
5. Resume backward proof with result computed by forward proof right

$$
\begin{prooftree}
\AxiomC{$*$}
\RightLabel{$\mathbb{R}$}
\UnaryInfC{$x^3\cdot x + x\cdot x^3 \geq 0$}
\RightLabel{$[':=]$}
\UnaryInfC{$[x' := x^3]x'\cdot x + x\cdot x' \geq 0$}
\RightLabel{G}
\UnaryInfC{$[x' = x^3][x' := x^3]x'\cdot x+x\cdot x'\geq 0$}
\AxiomC{$*$}
\RightLabel{$\mathbb{R}$}
\UnaryInfC{$(x\cdot x)' = (x\cdot x)'$}
\RightLabel{$\cdot'$}
\UnaryInfC{$(x\cdot x)' = (x)'\cdot x + x\cdot (x)'$}
\RightLabel{x'}
\UnaryInfC{$(x\cdot x)' = x'\cdot x + x\cdot x'$}
\RightLabel{CT}
\UnaryInfC{$(x\cdot x)' \geq 0 \leftrightarrow x'\cdot x + x\cdot x' \geq 0$}
\UnaryInfC{$(x\cdot x \geq 1)' \leftrightarrow x'\cdot x + x\cdot x' \geq 0$}
\RightLabel{CE}
\BinaryInfC{$[x' = x^3][x' := x^3](x\cdot x \geq 1)'$}
\RightLabel{DE}
\UnaryInfC{$[x' = x^3](x\cdot x \geq 1)'$}
\RightLabel{DI}
\UnaryInfC{$x\cdot x \geq 1 \rightarrow [x' = x^3]x\cdot x \geq 1$}
\end{prooftree}
$$

Uniform Substitution

Theorem (Soundness) replace all occurrences of $p(\cdot)$

$$US \;\; \frac{\phi}{\sigma(\phi)}$$

provided $FV(\sigma|_{\Sigma(\theta)}) \cap BV(\otimes(\cdot)) = \emptyset$ *for each operation* $\otimes(\theta)$ *in* ϕ

i.e. bound variables $U = \mathrm{BV}(\otimes(\cdot))$ of operator $\otimes$
are not free in the substitution on its argument θ (U-admissible)

Uniform substitution σ replaces all occurrences of $p(\theta)$ for any θ by $\psi(\theta)$
function $f(\theta)$ for any θ by $\eta(\theta)$
quantifier $C(\phi)$ for any ϕ by $\psi(\theta)$
program const. a by α

$$\text{US}\;\frac{[a \cup b]p(\bar{x}) \leftrightarrow [a]p(\bar{x}) \wedge [b]p(\bar{x})}{[x := x+1 \cup x' = 1]x \geq 0 \leftrightarrow [x := x+1]x \geq 0 \wedge [x' = 1]x \geq 0}$$

Theorem (Soundness) replace all occurrences of $p(\cdot)$

Modular interface: Prover vs. Logic

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

provided $FV(\sigma|_{\Sigma(\theta)}) \cap BV(\otimes(\cdot)) = \emptyset$ *for each operation* $\otimes(\theta)$ *in* ϕ

i.e. bound variables $U = \mathrm{BV}(\otimes(\cdot))$ of operator $\otimes$
are not free in the substitution on its argument θ (U-admissible)

Uniform substitution σ replaces all occurrences of $p(\theta)$ for any θ by $\psi(\theta)$
function $f(\theta)$ for any θ by $\eta(\theta)$
quantifier $C(\phi)$ for any ϕ by $\psi(\theta)$
program const. a by α

$$\text{US}\ \frac{[a \cup b]p(\bar{x}) \leftrightarrow [a]p(\bar{x}) \land [b]p(\bar{x})}{[x := x+1 \cup x' = 1]x \geq 0 \leftrightarrow [x := x+1]x \geq 0 \land [x' = 1]x \geq 0}$$

Correctness of Static Semantics

Lemma (Bound effect lemma) (Only BV(·) change)

If $(u, w) \in [\![\alpha]\!]I$, *then* $u = w$ *on* $BV(\alpha)^\complement$.

Lemma (Coincidence lemma) (Only FV(·) determine truth)

If $u = \tilde{u}$ *on* $FV(\theta)$ *and* $I = J$ *on* $\Sigma(\theta)$, *then*

$$[\![\theta]\!]Iu = [\![\theta]\!]J\tilde{u}$$

$$u \in [\![\phi]\!]I \text{ iff } \tilde{u} \in [\![\phi]\!]J$$

$$\begin{tikzcd}
u \arrow[r, "\alpha"'] \arrow[r, bend left, equal, "\text{on } BV(\alpha)^\complement"] \arrow[d, equal, "\text{on } V \supseteq FV(\alpha)"'] & w \arrow[d, equal, "\text{on } V \cup MBV(\alpha)"] \\
\tilde{u} \arrow[r, dashed, "\alpha", "\exists"'] \arrow[r, bend right, equal, "\text{on } BV(\alpha)^\complement"'] & \tilde{w}
\end{tikzcd}$$

Correctness of Static Semantics

Lemma (Bound effect lemma) (Only BV(·) change)

If $(u, w) \in [\![\alpha]\!]I$, then $u = w$ on $BV(\alpha)^\complement$.

Lemma (Coincidence lemma) (Only FV(·) determine truth)

If $u = \tilde{u}$ on $FV(\theta)$ and $I = J$ on $\Sigma(\theta)$, then
$$[\![\theta]\!]Iu = [\![\theta]\!]J\tilde{u}$$
$$u \in [\![\phi]\!]I \text{ iff } \tilde{u} \in [\![\phi]\!]J$$

$$\begin{tikzcd}[column sep=huge, row sep=huge]
u \arrow[r, "\alpha"'] \arrow[r, bend left, equal, "\text{on } BV(\alpha)^\complement"] \arrow[d, equal, "\text{on } V \supseteq FV(\alpha)"'] & w \arrow[d, equal, "\text{on } V \cup MBV(\alpha)"] \\
\tilde{u} \arrow[r, dashed, "\alpha", "\exists"'] \arrow[r, bend right, equal, "\text{on } BV(\alpha)^\complement"'] & \tilde{w}
\end{tikzcd}$$

Differential Dynamic Logic d$\mathcal{L}$: Static Semantics

$$
\begin{aligned}
\mathsf{FV}((\theta)') \\
\mathsf{FV}(p(\theta_1,\dots,\theta_k)) \\
\mathsf{FV}(C(\phi)) \\
\mathsf{FV}(\phi \land \psi) \\
\mathsf{FV}(\forall x\, \phi) = \mathsf{FV}(\exists x\, \phi) \\
\mathsf{FV}([\alpha]\phi) = \mathsf{FV}(\langle\alpha\rangle\phi) \\
\mathsf{FV}(a) \\
\mathsf{FV}(x := \theta) = \mathsf{FV}(x' := \theta) \\
\mathsf{FV}(?Q) \\
\mathsf{FV}(x' = f(x) \,\&\, Q) \\
\mathsf{FV}(\alpha \cup \beta) \\
\mathsf{FV}(\alpha;\beta) \\
\mathsf{FV}(\alpha^*)
\end{aligned}
$$

Differential Dynamic Logic dℒ: Static Semantics

$$
\begin{array}{rcl}
\mathrm{FV}((\theta)') &=& \mathrm{FV}(\theta) \\
\hline
\mathrm{FV}(p(\theta_1,\dots,\theta_k)) &=& \mathrm{FV}(\theta_1) \cup \dots \cup \mathrm{FV}(\theta_k) \\
\mathrm{FV}(C(\phi)) &=& \mathcal{V} \cup \mathcal{V}' \\
\mathrm{FV}(\phi \land \psi) &=& \mathrm{FV}(\phi) \cup \mathrm{FV}(\psi) \\
\mathrm{FV}(\forall x\, \phi) = \mathrm{FV}(\exists x\, \phi) &=& \mathrm{FV}(\phi) \setminus \{x\} \\
\mathrm{FV}([\alpha]\phi) = \mathrm{FV}(\langle\alpha\rangle\phi) &=& \mathrm{FV}(\alpha) \cup (\mathrm{FV}(\phi) \setminus \mathrm{BV}(\alpha)) \\
\hline
\mathrm{FV}(a) &=& \mathcal{V} \cup \mathcal{V}' \qquad \text{for program const. } a \\
\mathrm{FV}(x := \theta) = \mathrm{FV}(x' := \theta) &=& \mathrm{FV}(\theta) \\
\mathrm{FV}(?Q) &=& \mathrm{FV}(Q) \\
\mathrm{FV}(x' = f(x) \,\&\, Q) &=& \{x\} \cup \mathrm{FV}(f(x)) \cup \mathrm{FV}(Q) \\
\mathrm{FV}(\alpha \cup \beta) &=& \mathrm{FV}(\alpha) \cup \mathrm{FV}(\beta) \\
\mathrm{FV}(\alpha;\beta) &=& \mathrm{FV}(\alpha) \cup (\mathrm{FV}(\beta) \setminus \mathrm{BV}(\alpha)) \\
\mathrm{FV}(\alpha^*) &=& \mathrm{FV}(\alpha)
\end{array}
$$

Differential Dynamic Logic dℒ: Static Semantics

$$
\begin{array}{rll}
\mathrm{FV}((\theta)') & = \mathrm{FV}(\theta) \cup \mathrm{FV}(\theta)' & \text{caution} \\
\hline
\mathrm{FV}(p(\theta_1, \dots, \theta_k)) & = \mathrm{FV}(\theta_1) \cup \dots \cup \mathrm{FV}(\theta_k) & \\
\mathrm{FV}(C(\phi)) & = \mathcal{V} \cup \mathcal{V}' & \\
\mathrm{FV}(\phi \land \psi) & = \mathrm{FV}(\phi) \cup \mathrm{FV}(\psi) & \\
\mathrm{FV}(\forall x\, \phi) = \mathrm{FV}(\exists x\, \phi) & = \mathrm{FV}(\phi) \setminus \{x\} & \\
\mathrm{FV}([\alpha]\phi) = \mathrm{FV}(\langle\alpha\rangle\phi) & = \mathrm{FV}(\alpha) \cup (\mathrm{FV}(\phi) \setminus \mathrm{MBV}(\alpha)) & \text{caution} \\
\hline
\mathrm{FV}(a) & = \mathcal{V} \cup \mathcal{V}' & \text{for program const. } a \\
\mathrm{FV}(x := \theta) = \mathrm{FV}(x' := \theta) & = \mathrm{FV}(\theta) & \\
\mathrm{FV}(?Q) & = \mathrm{FV}(Q) & \\
\mathrm{FV}(x' = f(x) \,\&\, Q) & = \{x\} \cup \mathrm{FV}(f(x)) \cup \mathrm{FV}(Q) & \\
\mathrm{FV}(\alpha \cup \beta) & = \mathrm{FV}(\alpha) \cup \mathrm{FV}(\beta) & \\
\mathrm{FV}(\alpha; \beta) & = \mathrm{FV}(\alpha) \cup (\mathrm{FV}(\beta) \setminus \mathrm{MBV}(\alpha)) & \text{caution} \\
\mathrm{FV}(\alpha^*) & = \mathrm{FV}(\alpha) &
\end{array}
$$

$$
\begin{aligned}
\mathrm{BV}(\theta \geq \eta) = \mathrm{BV}(p(\theta_1, \ldots, \theta_k)) \\
\mathrm{BV}(C(\phi)) \\
\mathrm{BV}(\phi \land \psi) \\
\mathrm{BV}(\forall x\, \phi) = \mathrm{BV}(\exists x\, \phi) \\
\mathrm{BV}([\alpha]\phi) = \mathrm{BV}(\langle\alpha\rangle\phi) \\
\mathrm{BV}(a) \\
\mathrm{BV}(x := \theta) \\
\mathrm{BV}(x' := \theta) \\
\mathrm{BV}(?Q) \\
\mathrm{BV}(x' = f(x)\, \&\, Q) \\
\mathrm{BV}(\alpha \cup \beta) = \mathrm{BV}(\alpha; \beta) \\
\mathrm{BV}(\alpha^*)
\end{aligned}
$$

Differential Dynamic Logic d$\mathcal{L}$: Static Semantics

$$
\begin{aligned}
\mathrm{BV}(\theta \geq \eta) = \mathrm{BV}(p(\theta_1,\dots,\theta_k)) &= \emptyset \\
\mathrm{BV}(C(\phi)) &= \mathcal{V} \cup \mathcal{V}' \\
\mathrm{BV}(\phi \land \psi) &= \mathrm{BV}(\phi) \cup \mathrm{BV}(\psi) \\
\mathrm{BV}(\forall x\, \phi) = \mathrm{BV}(\exists x\, \phi) &= \{x\} \cup \mathrm{BV}(\phi) \\
\mathrm{BV}([\alpha]\phi) = \mathrm{BV}(\langle\alpha\rangle\phi) &= \mathrm{BV}(\alpha) \cup \mathrm{BV}(\phi) \\
\hline
\mathrm{BV}(a) &= \mathcal{V} \cup \mathcal{V}' \qquad \text{for program constant } a \\
\mathrm{BV}(x := \theta) &= \{x\} \\
\mathrm{BV}(x' := \theta) &= \{x'\} \\
\mathrm{BV}(?Q) &= \emptyset \\
\mathrm{BV}(x' = f(x) \,\&\, Q) &= \{x, x'\} \\
\mathrm{BV}(\alpha \cup \beta) = \mathrm{BV}(\alpha ; \beta) &= \mathrm{BV}(\alpha) \cup \mathrm{BV}(\beta) \\
\mathrm{BV}(\alpha^*) &= \mathrm{BV}(\alpha)
\end{aligned}
$$

Differential Dynamic Logic d$\mathcal{L}$: Static Semantics

$$\begin{aligned}
\mathsf{MBV}(a) \\
\mathsf{MBV}(\alpha) \\
\mathsf{MBV}(\alpha \cup \beta) \\
\mathsf{MBV}(\alpha;\beta) \\
\mathsf{MBV}(\alpha^*)
\end{aligned}$$

Differential Dynamic Logic d$\mathcal{L}$: Static Semantics

$$
\begin{aligned}
\mathrm{MBV}(a) &= \emptyset && \text{for program constant } a\\
\mathrm{MBV}(\alpha) &= \mathrm{BV}(\alpha) && \text{for other atomic HPs } \alpha\\
\mathrm{MBV}(\alpha \cup \beta) &= \mathrm{MBV}(\alpha) \cap \mathrm{MBV}(\beta)\\
\mathrm{MBV}(\alpha;\beta) &= \mathrm{MBV}(\alpha) \cup \mathrm{MBV}(\beta)\\
\mathrm{MBV}(\alpha^*) &= \emptyset
\end{aligned}
$$

Correctness of Static Semantics

Lemma (Bound effect lemma) (Only BV(·) change)

If $(u,w) \in [\![\alpha]\!]I$, *then* $u = w$ *on* $BV(\alpha)^\complement$.

Lemma (Coincidence lemma) (Only FV(·) determine truth)

If $u = \tilde{u}$ *on* $FV(\theta)$ *and* $I = J$ *on* $\Sigma(\theta)$, *then*

$$[\![\theta]\!]Iu = [\![\theta]\!]J\tilde{u}$$
$$u \in [\![\phi]\!]I \text{ iff } \tilde{u} \in [\![\phi]\!]J$$

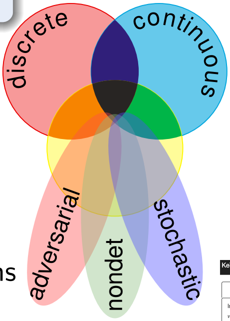
$$\begin{tikzcd}
u \arrow[r, "\alpha"'] \arrow[r, bend left, equal, "\text{on } BV(\alpha)^\complement"] \arrow[d, equal, "\text{on } V \supseteq FV(\alpha)"'] & w \arrow[d, equal, "\text{on } V \cup MBV(\alpha)"] \\
\tilde{u} \arrow[r, dashed, "\alpha", "\exists"'] \arrow[r, bend right, equal, "\text{on } BV(\alpha)^\complement"'] & \tilde{w}
\end{tikzcd}$$

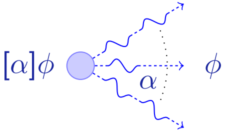
Outline

Uniform Substitution for Differential Dynamic Logic

differential dynamic logic

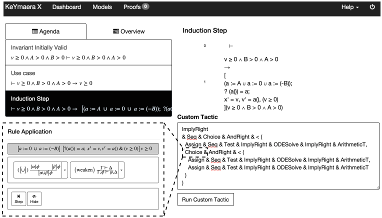
$d\mathcal{L} = \mathrm{DL} + \mathrm{HP}$





- Multi-dynamical systems
- Differential forms
  $\rightsquigarrow$ local axioms of ODEs
- Uniform substitution
  $\rightsquigarrow$ modular generic axioms
  (not schemata)
- Modular: Logic $\parallel$ Prover
- Straightforward to implement
- Tactics regain efficiency
- Fast contextual equivalence

KeYmaera X



Key Contributions

Q: How to build a prover with a small soundness-critical core?

A: Uniform substitution [Church]

Q: How to enable flexible yet sound reasoning?

A: Axioms with local meaning [Philosophy, Algebraic Geometry]

Q: What's the local meaning of a differential equation?

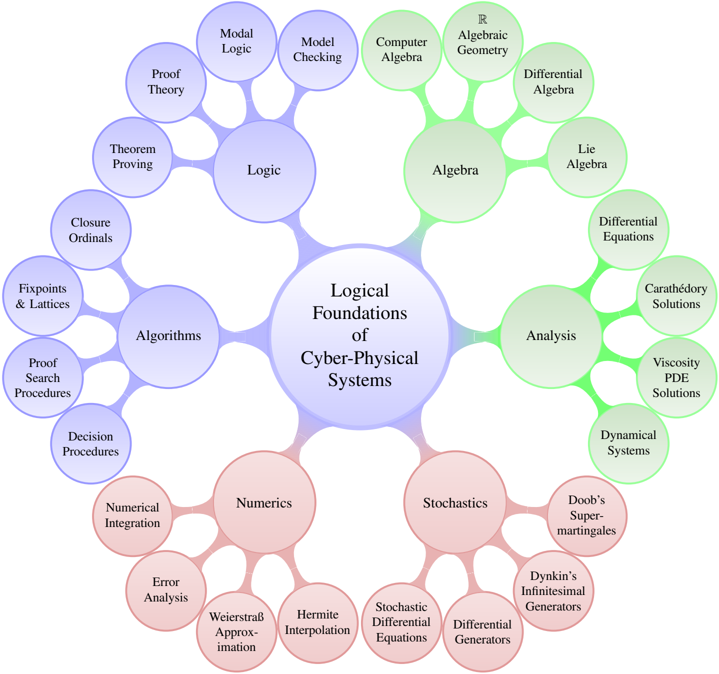
A: Differential forms [Differential Geometry]

Q: How to do hybrid systems proving?

A: Uniform substitution calculus for differential dynamic logic

Q: What's the impact of uniform substitution on a prover core?

A: 65 989 $\searrow$ 1 677 LOC (2.5%) [KeYmaera X]


Logical Foundations of Cyber-Physical Systems
Logic
Theorem Proving
Proof Theory
Modal Logic
Model Checking
Algebra
Computer Algebra
ℝ Algebraic Geometry
Differential Algebra
Lie Algebra
Analysis
Differential Equations
Carathédory Solutions
Viscosity PDE Solutions
Dynamical Systems
Stochastics
Doob's Super-martingales
Dynkin's Infinitesimal Generators
Differential Generators
Stochastic Differential Equations
Numerics
Hermite Interpolation
Weierstraß Approx-imation
Error Analysis
Numerical Integration
Algorithms
Decision Procedures
Proof Search Procedures
Fixpoints & Lattices
Closure Ordinals

KeYmaera X Kernel is a Microkernel for Soundness

	≈LOC		
KeYmaera X	1 677		hybrid prover
KeYmaera	65 989		
KeY	51 328		Java
HOL Light	396		general math
Isabelle/Pure	8 113		
Nuprl	15 000	+ 50 000	
Coq	20 000		
HSolver	20 000		hybrid verifier
Flow*	25 000		
PHAVer	30 000		
dReal	50 000	+ millions	
SpaceEx	100 000		
HyCreate2	6 081	+ user model analysis	

Disclaimer: These self-reported estimates of the soundness-critical lines of code + rules are to be taken with a grain of salt. Different languages, capabilities, styles

André Platzer.
A uniform substitution calculus for differential dynamic logic.
In Amy Felty and Aart Middeldorp, editors, *CADE*, volume 9195 of *LNCS*, pages 467–481. Springer, 2015.
doi:10.1007/978-3-319-21401-6_32.

André Platzer.
A uniform substitution calculus for differential dynamic logic.
CoRR, abs/1503.01981, 2015.
arXiv:1503.01981.

André Platzer.
A complete uniform substitution calculus for differential dynamic logic.
CoRR, abs/1507.04943, 2016.
arXiv:1507.04943.

André Platzer.
Logics of dynamical systems.
In LICS [15], pages 13–24.

doi:10.1109/LICS.2012.13.

André Platzer.
Differential dynamic logic for hybrid systems.
J. Autom. Reas., 41(2):143–189, 2008.
doi:10.1007/s10817-008-9103-8.

André Platzer.
Differential game logic.
ACM Trans. Comput. Log., 17(1):1:1–1:51, 2015.
doi:10.1145/2817824.

André Platzer.
The complete proof theory of hybrid systems.
In LICS [15], pages 541–550.
doi:10.1109/LICS.2012.64.

André Platzer.
A complete axiomatization of quantified differential dynamic logic for distributed hybrid systems.
Log. Meth. Comput. Sci., 8(4):1–44, 2012.

Special issue for selected papers from CSL'10.
doi:10.2168/LMCS-8(4:17)2012.

André Platzer.
Stochastic differential dynamic logic for stochastic hybrid programs.
In Nikolaj Bjørner and Viorica Sofronie-Stokkermans, editors, *CADE*, volume 6803 of *LNCS*, pages 431–445. Springer, 2011.
doi:10.1007/978-3-642-22438-6_34.

André Platzer.
Differential-algebraic dynamic logic for differential-algebraic programs.
J. Log. Comput., 20(1):309–352, 2010.
doi:10.1093/logcom/exn070.

André Platzer and Edmund M. Clarke.
Computing differential invariants of hybrid systems as fixedpoints.
In Aarti Gupta and Sharad Malik, editors, *CAV*, volume 5123 of *LNCS*, pages 176–189. Springer, 2008.
doi:10.1007/978-3-540-70545-1_17.

André Platzer and Edmund M. Clarke.

Computing differential invariants of hybrid systems as fixedpoints.
Form. Methods Syst. Des., 35(1):98–120, 2009.
Special issue for selected papers from CAV'08.
doi:10.1007/s10703-009-0079-8.

André Platzer.
The structure of differential invariants and differential cut elimination.
Log. Meth. Comput. Sci., 8(4):1–38, 2012.
doi:10.2168/LMCS-8(4:16)2012.

André Platzer.
A differential operator approach to equational differential invariants.
In Lennart Beringer and Amy Felty, editors, *ITP*, volume 7406 of *LNCS*, pages 28–48. Springer, 2012.
doi:10.1007/978-3-642-32347-8_3.

Proceedings of the 27th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2012, Dubrovnik, Croatia, June 25–28, 2012. IEEE, 2012.

Differential Dynamic Logic: Axioms

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

$$[?]\ [?q]p \leftrightarrow (q \rightarrow p)$$

$$[\cup]\ [a \cup b]p(\bar{x}) \leftrightarrow [a]p(\bar{x}) \wedge [b]p(\bar{x})$$

$$[;]\ [a;b]p(\bar{x}) \leftrightarrow [a][b]p(\bar{x})$$

$$[^*]\ [a^*]p(\bar{x}) \leftrightarrow p(\bar{x}) \wedge [a][a^*]p(\bar{x})$$

$$\mathrm{K}\ [a](p(\bar{x}) \rightarrow q(\bar{x})) \rightarrow ([a]p(\bar{x}) \rightarrow [a]q(\bar{x}))$$

$$\mathrm{I}\ [a^*](p(\bar{x}) \rightarrow [a]p(\bar{x})) \rightarrow (p(\bar{x}) \rightarrow [a^*]p(\bar{x}))$$

$$\mathrm{V}\ p \rightarrow [a]p$$

CADE'15

Differential Dynamic Logic: Axioms

$$\text{G}\quad \frac{p(\bar{x})}{[a]p(\bar{x})}$$

$$\forall\quad \frac{p(x)}{\forall x\, p(x)}$$

$$\text{MP}\quad \frac{p \rightarrow q \quad p}{q}$$

$$\text{CT}\quad \frac{f(\bar{x}) = g(\bar{x})}{c(f(\bar{x})) = c(g(\bar{x}))}$$

$$\text{CQ}\quad \frac{f(\bar{x}) = g(\bar{x})}{p(f(\bar{x})) \leftrightarrow p(g(\bar{x}))}$$

$$\text{CE}\quad \frac{p(\bar{x}) \leftrightarrow q(\bar{x})}{C(p(\bar{x})) \leftrightarrow C(q(\bar{x}))}$$

CADE'15

DW $[x' = f(x) \,\&\, q(x)]q(x)$

DC $\big([x' = f(x) \,\&\, q(x)]p(x) \leftrightarrow [x' = f(x) \,\&\, q(x) \land r(x)]p(x)\big) \leftarrow [x' = f(x) \,\&\, q(x)]r(x)$

DE $[x' = f(x) \,\&\, q(x)]p(x, x') \leftrightarrow [x' = f(x) \,\&\, q(x)][x' := f(x)]p(x, x')$

DI $[x' = f(x) \,\&\, q(x)]p(x) \leftarrow \big(q(x) \rightarrow p(x) \land [x' = f(x) \,\&\, q(x)](p(x))'\big)$

DG $[x' = f(x) \,\&\, q(x)]p(x) \leftrightarrow \exists y\, [x' = f(x), y' = a(x)y + b(x) \,\&\, q(x)]p(x)$

DS $[x' = f \,\&\, q(x)]p(x) \leftrightarrow \forall t{\geq}0\, \big((\forall 0{\leq}s{\leq}t\; q(x{+}fs)) \rightarrow [x := x{+}ft]p(x)\big)$

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

$+'$ $(f(\bar{x}) + g(\bar{x}))' = (f(\bar{x}))' + (g(\bar{x}))'$

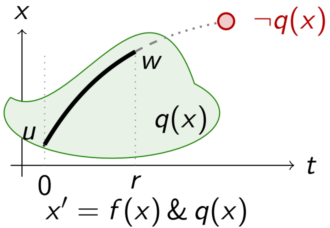
$\cdot'$ $(f(\bar{x}) \cdot g(\bar{x}))' = (f(\bar{x}))' \cdot g(\bar{x}) + f(\bar{x}) \cdot (g(\bar{x}))'$

$\circ'$ $[y := g(x)][y' := 1]\big((f(g(x)))' = (f(y))' \cdot (g(x))'\big)$

CADE'15

Axiom (Differential Weakening) (CADE'15)

DW $[x' = f(x)\,\&\,q(x)]q(x)$

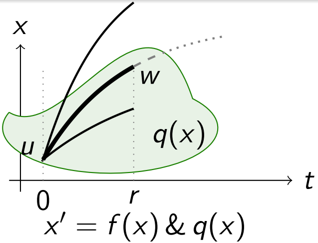


Differential equations cannot leave their evolution domains. Implies:

$$[x' = f(x)\,\&\,q(x)]p(x) \leftrightarrow [x' = f(x)\,\&\,q(x)]\big(q(x) \rightarrow p(x)\big)$$

Axiom (Differential Cut) (CADE'15)

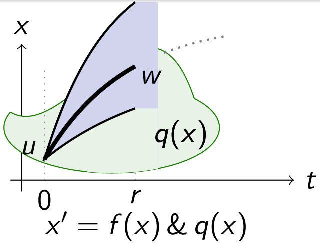
DC $\big([x' = f(x)\,\&\, q(x)]p(x) \leftrightarrow [x' = f(x)\,\&\, q(x) \land r(x)]p(x)\big)$
$\leftarrow [x' = f(x)\,\&\, q(x)]r(x)$



DC is a cut for differential equations.
DC is a differential modal modus ponens K.
Can't leave $r(x)$, then might as well restrict state space to $r(x)$.

Axiom (Differential Cut) (CADE'15)

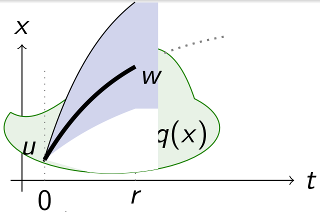
DC $\big([x' = f(x)\,\&\,q(x)]p(x) \leftrightarrow [x' = f(x)\,\&\,q(x) \land r(x)]p(x)\big)$
$\leftarrow [x' = f(x)\,\&\,q(x)]r(x)$



DC is a cut for differential equations.
DC is a differential modal modus ponens K.
Can't leave $r(x)$, then might as well restrict state space to $r(x)$.

Axiom (Differential Cut) (CADE'15)

DC $$([x' = f(x)\,\&\,q(x)]p(x) \leftrightarrow [x' = f(x)\,\&\,q(x) \land r(x)]p(x))$$
$$\leftarrow [x' = f(x)\,\&\,q(x)]r(x)$$



$x' = f(x)\,\&\,q(x)$

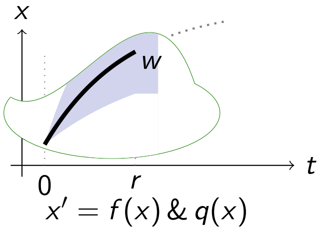
DC is a cut for differential equations.

DC is a differential modal modus ponens K.

Can't leave $r(x)$, then might as well restrict state space to $r(x)$.

Axiom (Differential Cut) (CADE'15)

DC $$([x' = f(x) \,\&\, q(x)]p(x) \leftrightarrow [x' = f(x) \,\&\, q(x) \wedge r(x)]p(x))$$
$$\leftarrow [x' = f(x) \,\&\, q(x)]r(x)$$

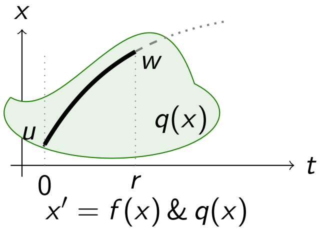


DC is a cut for differential equations.
DC is a differential modal modus ponens K.
Can't leave $r(x)$, then might as well restrict state space to $r(x)$.

Differential Equation Axioms

Axiom (Differential Cut) (CADE'15)

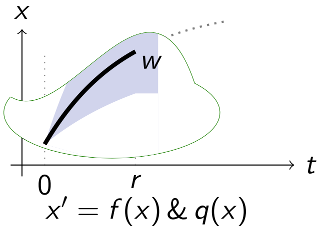
DC $\quad ([x' = f(x)\,\&\, q(x)]p(x) \leftrightarrow [x' = f(x)\,\&\, q(x) \wedge r(x)]p(x))$
$\leftarrow [x' = f(x)\,\&\, q(x)]r(x)$

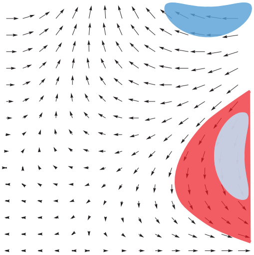


DC is a cut for differential equations.
DC is a differential modal modus ponens K.
Can't leave $r(x)$, then might as well restrict state space to $r(x)$.

Axiom (Differential Cut) (CADE'15)

DC $\big([x' = f(x)\,\&\,q(x)]p(x) \leftrightarrow [x' = f(x)\,\&\,q(x) \land r(x)]p(x)\big)$
$\leftarrow [x' = f(x)\,\&\,q(x)]r(x)$





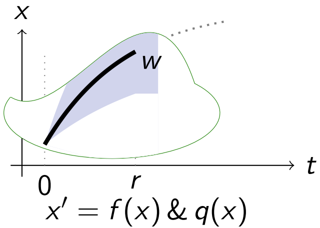
DC is a cut for differential equations.

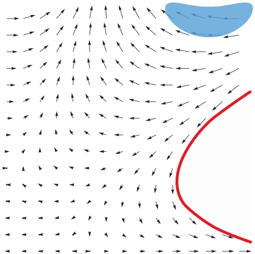
DC is a differential modal modus ponens K.

Can't leave $r(x)$, then might as well restrict state space to $r(x)$.

Axiom (Differential Cut) (CADE'15)

DC $\quad ([x' = f(x) \,\&\, q(x)]p(x) \leftrightarrow [x' = f(x) \,\&\, q(x) \land r(x)]p(x))$
$\leftarrow [x' = f(x) \,\&\, q(x)]r(x)$





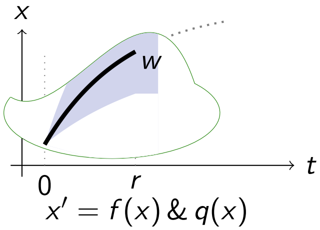
DC is a cut for differential equations.

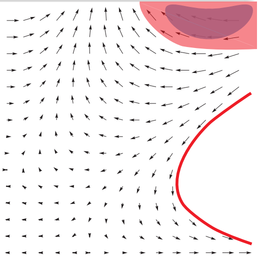
DC is a differential modal modus ponens K.

Can't leave $r(x)$, then might as well restrict state space to $r(x)$.

Axiom (Differential Cut) (CADE'15)

DC $\quad ([x' = f(x) \,\&\, q(x)]p(x) \leftrightarrow [x' = f(x) \,\&\, q(x) \land r(x)]p(x))$
$\quad \leftarrow [x' = f(x) \,\&\, q(x)]r(x)$

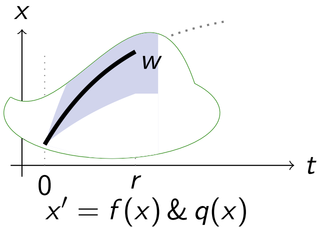


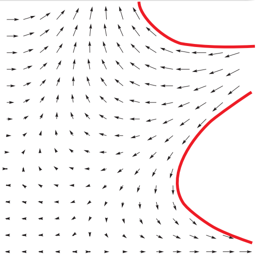


DC is a cut for differential equations.
DC is a differential modal modus ponens K.
Can't leave $r(x)$, then might as well restrict state space to $r(x)$.

Axiom (Differential Cut) (CADE'15)

DC $$([x' = f(x)\,\&\,q(x)]p(x) \leftrightarrow [x' = f(x)\,\&\,q(x) \wedge r(x)]p(x)) \leftarrow [x' = f(x)\,\&\,q(x)]r(x)$$





DC is a cut for differential equations.

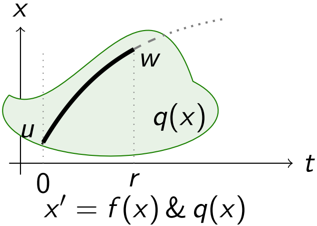
DC is a differential modal modus ponens K.

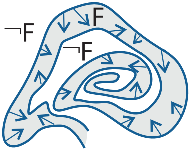
Can't leave $r(x)$, then might as well restrict state space to $r(x)$.

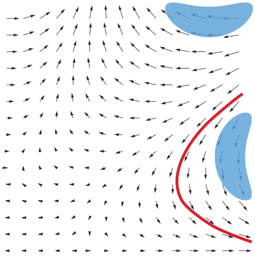
Differential Equation Axioms

Axiom (Differential Invariant) (CADE'15)

DI $[x' = f(x) \,\&\, q(x)]p(x) \leftarrow \big(q(x) \rightarrow p(x) \land [x' = f(x) \,\&\, q(x)](p(x))'\big)$







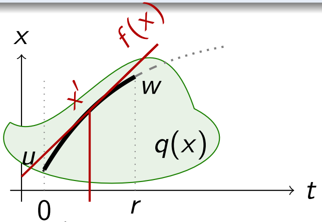
Differential invariant: $p(x)$ true now and its differential $(p(x))'$ true always

What's the differential of a formula???

What's the meaning of a differential term ... in a state???

Axiom (Differential Effect) (CADE'15)

DE $[x' = f(x)\,\&\,q(x)]p(x,x') \leftrightarrow [x' = f(x)\,\&\,q(x)][x' := f(x)]p(x,x')$



$x' = f(x)\,\&\,q(x)$

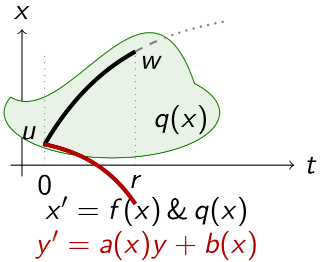
Effect of differential equation on differential symbol x'

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

$[x' := f(x)]$ selects vector field $x' = f(x)$ for subsequent differentials

Axiom (Differential Ghost) (CADE'15)

DG $[x' = f(x) \,\&\, q(x)]p(x) \leftrightarrow \exists y\, [x' = f(x), y' = a(x)y + b(x) \,\&\, q(x)]p(x)$



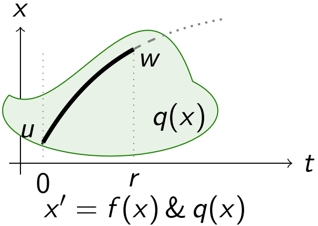
Differential ghost/auxiliaries: extra differential equations that exist

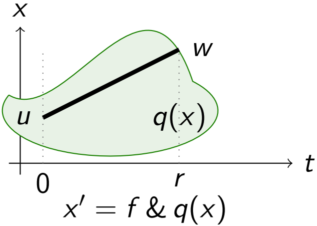
Can cause new invariants

"Dark matter" counterweight to balance conserved quantities

Axiom (Differential Solution) (CADE'15)

DS $[x' = f \,\&\, q(x)]p(x) \leftrightarrow \forall t{\geq}0\,\big((\forall 0{\leq}s{\leq}t\; q(x{+}fs)) \rightarrow [x := x{+}ft]p(x)\big)$





Differential solutions: solve differential equations
with DG,DC and inverse companions

Differential-form Differential Dynamic Logic: Semantics

Definition (Term semantics) $(\llbracket\cdot\rrbracket : \mathrm{Trm} \to (\mathcal{S} \to \mathbb{R}))$

$$\llbracket(\theta)'\rrbracket I u = \sum_x u(x') \frac{\partial \llbracket\theta\rrbracket I}{\partial x}(u) = \sum_x u(x') \frac{\partial \llbracket\theta\rrbracket I u_x^X}{\partial X}$$

Definition (d$\mathcal{L}$ semantics) $(\llbracket\cdot\rrbracket : \mathrm{Fml} \to \wp(\mathcal{S}))$

$$\begin{aligned}
\llbracket C(\phi)\rrbracket I &= I(C)\big(\llbracket\phi\rrbracket I\big)\\
\llbracket\langle\alpha\rangle\phi\rrbracket I &= \llbracket\alpha\rrbracket I \circ \llbracket\phi\rrbracket I\\
\llbracket[\alpha]\phi\rrbracket I &= \llbracket\neg\langle\alpha\rangle\neg\phi\rrbracket I
\end{aligned}$$

Definition (Program semantics) $(\llbracket\cdot\rrbracket : \mathrm{HP} \to \wp(\mathcal{S} \times \mathcal{S}))$

$$\begin{aligned}
\llbracket x' = f(x) \,\&\, Q\rrbracket I &= \{(\varphi(0)|_{\{x'\}^\complement}, \varphi(r)) \;:\; I, \varphi \models x' = f(x) \land Q\}\\
\llbracket\alpha \cup \beta\rrbracket I &= \llbracket\alpha\rrbracket I \cup \llbracket\beta\rrbracket I\\
\llbracket\alpha;\beta\rrbracket I &= \llbracket\alpha\rrbracket I \circ \llbracket\beta\rrbracket I\\
\llbracket\alpha^*\rrbracket I &= \big(\llbracket\alpha\rrbracket I\big)^* = \bigcup_{n\in\mathbb{N}} \llbracket\alpha^n\rrbracket I
\end{aligned}$$

Differential-form Differential Dynamic Logic: Semantics

Definition (Term semantics) $(\llbracket\cdot\rrbracket : \mathrm{Trm} \to (\mathcal{S} \to \mathbb{R}))$

$$
\begin{aligned}
\llbracket x \rrbracket I u &= u(x) && \text{for variable } x \in \mathcal{V}\\
\llbracket x' \rrbracket I u &= u(x') && \text{for differential symbol } x' \in \mathcal{V}'\\
\llbracket f(\theta_1,\dots,\theta_k) \rrbracket I u &= I(f)\big(\llbracket \theta_1 \rrbracket I u,\dots,\llbracket \theta_k \rrbracket I u\big) && \text{for function symbol } f\\
\llbracket \theta + \eta \rrbracket I u &= \llbracket \theta \rrbracket I u + \llbracket \eta \rrbracket I u\\
\llbracket \theta \cdot \eta \rrbracket I u &= \llbracket \theta \rrbracket I u \cdot \llbracket \eta \rrbracket I u\\
\llbracket (\theta)' \rrbracket I u &= \sum_x u(x') \frac{\partial \llbracket \theta \rrbracket I}{\partial x}(u) && = \sum_x u(x') \frac{\partial \llbracket \theta \rrbracket I u_x^X}{\partial X}
\end{aligned}
$$

Definition (d$\mathcal{L}$ semantics) $(\llbracket\cdot\rrbracket : \mathrm{Fml} \to \wp(\mathcal{S}))$

$$
\begin{aligned}
\llbracket C(\phi) \rrbracket I &= I(C)\big(\llbracket \phi \rrbracket I\big)\\
\llbracket \langle\alpha\rangle\phi \rrbracket I &= \llbracket \alpha \rrbracket I \circ \llbracket \phi \rrbracket I\\
\llbracket [\alpha]\phi \rrbracket I &= \llbracket \neg\langle\alpha\rangle\neg\phi \rrbracket I
\end{aligned}
$$

Definition (Program semantics) $(\llbracket\cdot\rrbracket : \mathrm{HP} \to \wp(\mathcal{S} \times \mathcal{S}))$

Definition (Term semantics) $(\llbracket\cdot\rrbracket : \mathrm{Trm} \to (\mathcal{S} \to \mathbb{R}))$

$$\llbracket(\theta)'\rrbracket Iu = \sum_x u(x')\frac{\partial\llbracket\theta\rrbracket I}{\partial x}(u) = \sum_x u(x')\frac{\partial\llbracket\theta\rrbracket Iu_x^X}{\partial X}$$

Definition (d$\mathcal{L}$ semantics) $(\llbracket\cdot\rrbracket : \mathrm{Fml} \to \wp(\mathcal{S}))$

$$\begin{aligned}
\llbracket\theta \geq \eta\rrbracket I &= \{u : \llbracket\theta\rrbracket Iu \geq \llbracket\eta\rrbracket Iu\}\\
\llbracket p(\theta_1,\dots,\theta_k)\rrbracket I &= \{u : (\llbracket\theta_1\rrbracket Iu,\dots,\llbracket\theta_k\rrbracket Iu) \in I(p)\}\\
\llbracket C(\phi)\rrbracket I &= I(C)\big(\llbracket\phi\rrbracket I\big)\\
\llbracket\neg\phi\rrbracket I &= (\llbracket\phi\rrbracket I)^{\complement}\\
\llbracket\phi \land \psi\rrbracket I &= \llbracket\phi\rrbracket I \cap \llbracket\psi\rrbracket I\\
\llbracket\exists x\,\phi\rrbracket I &= \{u \in \mathcal{S} : u_x^r \in \llbracket\phi\rrbracket I \text{ for some } r \in \mathbb{R}\}\\
\llbracket\langle\alpha\rangle\phi\rrbracket I &= \llbracket\alpha\rrbracket I \circ \llbracket\phi\rrbracket I = \{u : w \in \llbracket\phi\rrbracket I \text{ for some } w\ (u,w) \in \llbracket\alpha\rrbracket I\}\\
\llbracket[\alpha]\phi\rrbracket I &= \llbracket\neg\langle\alpha\rangle\neg\phi\rrbracket I = \{u : w \in \llbracket\phi\rrbracket I \text{ for all } w\ (u,w) \in \llbracket\alpha\rrbracket I\}
\end{aligned}$$

Definition (Program semantics) $(\llbracket\cdot\rrbracket : \mathrm{HP} \to \wp(\mathcal{S} \times \mathcal{S}))$

Differential-form Differential Dynamic Logic: Semantics

Definition (Term semantics) $(\llbracket\cdot\rrbracket : \mathrm{Trm} \to (\mathcal{S} \to \mathbb{R}))$

$$\llbracket(\theta)'\rrbracket Iu \;=\; \sum_x u(x')\frac{\partial\llbracket\theta\rrbracket I}{\partial x}(u) = \sum_x u(x')\frac{\partial\llbracket\theta\rrbracket Iu_x^X}{\partial X}$$

Definition (d$\mathcal{L}$ semantics) $(\llbracket\cdot\rrbracket : \mathrm{Fml} \to \wp(\mathcal{S}))$

$$\begin{aligned}
\llbracket C(\phi)\rrbracket I &= I(C)\big(\llbracket\phi\rrbracket I\big)\\
\llbracket\langle\alpha\rangle\phi\rrbracket I &= \llbracket\alpha\rrbracket I \circ \llbracket\phi\rrbracket I\\
\llbracket[\alpha]\phi\rrbracket I &= \llbracket\neg\langle\alpha\rangle\neg\phi\rrbracket I
\end{aligned}$$

Definition (Program semantics) $(\llbracket\cdot\rrbracket : \mathrm{HP} \to \wp(\mathcal{S} \times \mathcal{S}))$

$$\begin{aligned}
\llbracket a\rrbracket I &= I(a)\\
\llbracket x := \theta\rrbracket I &= \{(u,w) \;:\; w = u \text{ except } \llbracket x\rrbracket Iw = \llbracket\theta\rrbracket Iu\}\\
\llbracket x' := \theta\rrbracket I &= \{(u,w) \;:\; w = u \text{ except } \llbracket x'\rrbracket Iw = \llbracket\theta\rrbracket Iu\}\\
\llbracket ?Q\rrbracket I &= \{(u,u) \;:\; u \in \llbracket Q\rrbracket I\}\\
\llbracket x' = f(x)\,\&\,Q\rrbracket I &= \{(\varphi(0)|_{\{x'\}^\complement}, \varphi(r)) \;:\; I,\varphi \models x' = f(x) \land Q\}\\
\llbracket \alpha \cup \beta\rrbracket I &= \llbracket\alpha\rrbracket I \cup \llbracket\beta\rrbracket I
\end{aligned}$$