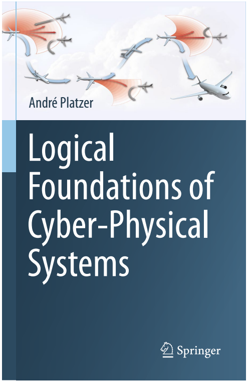
# 18: Axioms & Uniform Substitutions

## Logical Foundations of Cyber-Physical Systems


André Platzer

Logical Foundations of Cyber-Physical Systems

Springer

Stefan Mitsch

Carnegie Mellon University
Computer Science Department

# Outline

1. Learning Objectives
2. Axioms Versus Axiom Schemata
3. Differential Dynamic Logic with Interpretations
    - Syntax
    - Semantics
4. Uniform Substitution
    - Uniform Substitution Application
    - Uniform Substitution Lemmas
5. Axiomatic Proof Calculus for dL
6. Summary

# Outline

# Learning Objectives

## Axioms & Uniform Substitutions

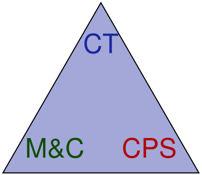
axiom vs. axiom schema

algorithmic impact of philosophical difference

local meaning of axioms

generic axioms like generic points

uniform substitution



meaning of differentials

parsimonious CPS reasoning impl.

modular impl. of logic $\|$ prover

# Outline

1. Learning Objectives

2. Axioms Versus Axiom Schemata

3. Differential Dynamic Logic with Interpretations
   - Syntax
   - Semantics

4. Uniform Substitution
   - Uniform Substitution Application
   - Uniform Substitution Lemmas

5. Axiomatic Proof Calculus for dL

6. Summary

# Differential Dynamic Logic: Axiomatization

**Part I**

$[:=]$ $[x := \theta]\phi(x) \leftrightarrow \phi(\theta)$

$[?]$ $[?\chi]\phi \leftrightarrow (\chi \rightarrow \phi)$

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

$[;]$ $[\alpha;\beta]\phi \leftrightarrow [\alpha][\beta]\phi$

$[^*]$ $[\alpha^*]\phi \leftrightarrow \phi \wedge [\alpha][\alpha^*]\phi$

K $[\alpha](\phi \rightarrow \psi) \rightarrow ([\alpha]\phi \rightarrow [\alpha]\psi)$

I $[\alpha^*]\phi \leftrightarrow \phi \wedge [\alpha^*](\phi \rightarrow [\alpha]\phi)$

V $\phi \rightarrow [\alpha]\phi$

$[']$ $[x' = \theta]\phi \leftrightarrow \forall t \geq 0\, [x := y(t)]\phi$

# Differential Dynamic Logic: Axiomatization

Part I

$[:=]\ [x := \theta]\phi(x) \leftrightarrow \phi(\theta)$ $\quad(\theta$ free for $x$ in $\phi)$

$[?]\ [?\chi]\phi \leftrightarrow (\chi \rightarrow \phi)$

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

$[;]\ [\alpha;\beta]\phi \leftrightarrow [\alpha][\beta]\phi$

$[^*]\ [\alpha^*]\phi \leftrightarrow \phi \wedge [\alpha][\alpha^*]\phi$

$\mathrm{K}\ [\alpha](\phi \rightarrow \psi) \rightarrow ([\alpha]\phi \rightarrow [\alpha]\psi)$

$\mathrm{I}\ [\alpha^*]\phi \leftrightarrow \phi \wedge [\alpha^*](\phi \rightarrow [\alpha]\phi)$

$\mathrm{V}\ \phi \rightarrow [\alpha]\phi$ $\quad(FV(\phi) \cap BV(\alpha) = \emptyset)$

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

# Axiom Schema

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

V $\phi \to [\alpha]\phi$

[:=] $[x := \theta]\phi(x) \leftrightarrow \phi(\theta)$

# Axiom Schema Matches Many Formulas

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

- $[x := x+1 \cup x' = x^2]\, x \geq 0 \leftrightarrow [x := x+1]\, x \geq 0 \land [x' = x^2]\, x \geq 0$
- $[x' = 5 \cup x' = -x]\, x^2 \geq 5 \leftrightarrow [x' = 5]\, x^2 \geq 5 \land [x' = -x]\, x^2 \geq 5$
- $[v := v{+}1; x' = v \cup x' = 2]x{\geq}5 \leftrightarrow [v := v{+}1; x' = v]x{\geq}5 \land [x' = 2]x{\geq}4$

V $\phi \rightarrow [\alpha]\phi$

[:=] $[x := \theta]\phi(x) \leftrightarrow \phi(\theta)$

# Axiom Schema Matches Many Formulas

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

✓ $[x := x+1 \cup x' = x^2]\, x \geq 0 \leftrightarrow [x := x+1]\, x \geq 0 \land [x' = x^2]\, x \geq 0$

✓ $[x' = 5 \cup x' = -x]\, x^2 \geq 5 \leftrightarrow [x' = 5]\, x^2 \geq 5 \land [x' = -x]\, x^2 \geq 5$

× $[v := v+1; x' = v \cup x' = 2]x \geq 5 \leftrightarrow [v := v+1; x' = v]x \geq 5 \land [x' = 2]x \geq 4$

V $\phi \rightarrow [\alpha]\phi$

$[:=]$ $[x := \theta]\phi(x) \leftrightarrow \phi(\theta)$

# Axiom Schema Matches Many Formulas

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

Match shape $\alpha \cup \beta$

Schema variable $\alpha$ match

Same $\phi$ everywhere

$= x +$ $x^2] x$ $x := x + 1] x \geq 0 \wedge [x' = x^2] x \geq 0$

$= 5 \cup$ $x^2 \geq$ $= 5] x^2 \geq 5 \wedge [x' = -x] x^2 \geq 5$

$= v+$ $\cup x' =$ $\leftrightarrow [v := v+1; x' = v] x \geq 5 \wedge [x' = 2] x \geq 4$

V $\phi \rightarrow [\alpha]\phi$

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

[:=] $[x := \theta]\phi(x) \leftrightarrow \phi(\theta)$

# Axiom Schema Matches Many Formulas

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

Match shape $\alpha \cup \beta$

Schema variable $\alpha$ match

Same $\phi$ everywhere

$= x +$ … $x^2]\, x$ … $x := x + 1]\, x \geq 0 \wedge [x' = x^2]\, x \geq 0$

$= 5 \cup$ … $x^2 \geq$ … $= 5]\, x^2 \geq 5 \wedge [x' = -x]\, x^2 \geq 5$

$= v +$ … $\cup\, x' =$ … $\leftrightarrow [v := v+1; x' = v]x \geq 5 \wedge [x' = 2]x \geq 4$

V $\phi \rightarrow [\alpha]\phi$

✓ $y \geq 0 \rightarrow [x' = -5]\, y \geq 0$

× $x \geq 0 \rightarrow [x' = -5]\, x \geq 0$

✓ $y \geq z \rightarrow [x' = -5]\, y \geq z$

[:=] $[x := \theta]\phi(x) \leftrightarrow \phi(\theta)$

# Axiom Schema Matches Many Formulas But Not All

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

Match shape $\alpha\cup\beta$

Schema variable $\alpha$ match

Same $\phi$ everywhere

$= x + \quad x^2]\,x \quad x := x+1]\,x \geq 0 \wedge [x' = x^2]\,x \geq 0$

$= 5 \cup \quad x^2 \geq \quad = 5]\,x^2 \geq 5 \wedge [x' = -x]\,x^2 \geq 5$

$= v+ \quad \cup x' = \quad \leftrightarrow [v := v+1; x' = v]x\geq5 \wedge [x' = 2]x\geq4$

V $\phi \rightarrow [\alpha]\phi$ $\quad (FV(\phi)\cap BV(\alpha) = \emptyset)$

✓ $y \geq 0 \rightarrow [x' = -5]\,y \geq 0$

× $x \geq 0 \rightarrow [x' = -5]\,x \geq 0$

✓ $y \geq z \rightarrow [x' = -5]\,y \geq z$

rule out by side conditions

[:=] $[x := \theta]\phi(x) \leftrightarrow \phi(\theta)$

- $[x := x+y]\,x \leq y^2 \leftrightarrow x+y \leq y^2$
- $[x := x+y][y := 5]\,x\geq0 \leftrightarrow [y := 5]\,x+y\geq0$
- $[y := 2b][(x := x+y; x' = y)^*]\,x\geq y \leftrightarrow [(x := x+2b; x' = 2b)^*]\,x\geq2b$
- $[x := x+y][x := x+1]\,x\geq0 \leftrightarrow [x := x+y+1]\,x\geq0$

# Axiom Schema Matches Many Formulas But Not All

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

$= x +$ $x^2] x$ $x := x + 1] x \geq 0 \wedge [x' = x^2] x \geq 0$

$= 5 \cup$ $x^2 \geq$ $= 5] x^2 \geq 5 \wedge [x' = -x] x^2 \geq 5$

$= v +$ $\cup x' =$ $\leftrightarrow [v := v+1; x' = v] x \geq 5 \wedge [x' = 2] x \geq 4$

Match shape $\alpha \cup \beta$

Schema variable $\alpha$ match

Same $\phi$ everywhere

V $\phi \rightarrow [\alpha]\phi$ $(FV(\phi) \cap BV(\alpha) = \emptyset)$

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

rule out by side conditions

[:=] $[x := \theta]\phi(x) \leftrightarrow \phi(\theta)$

- ✓ $[x := x + y] x \leq y^2 \leftrightarrow x + y \leq y^2$
- × $[x := x + y][y := 5] x \geq 0 \leftrightarrow [y := 5] x + y \geq 0$
- ✓ $[y := 2b][(x := x+y; x' = y)^*] x \geq y \leftrightarrow [(x := x+2b; x' = 2b)^*] x \geq 2b$
- ✓ $[x := x + y][x := x + 1] x \geq 0 \leftrightarrow [x := x + y + 1] x \geq 0$

# Axiom Schema Matches Many Formulas But Not All

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

Match shape $\alpha\cup\beta$

Schema variable $\alpha$ match

Same $\phi$ everywhere

$=x+$ $x^2]\,x$ $x := x+1]\,x \geq 0 \wedge [x' = x^2]\,x \geq 0$

$=5\cup$ $x^2 \geq$ $=5]\,x^2 \geq 5 \wedge [x' = -x]\,x^2 \geq 5$

$=v+$ $\cup x'$ $\leftrightarrow [v := v+1; x' = v]x \geq 5 \wedge [x' = 2]x \geq 4$

V $\phi \rightarrow [\alpha]\phi$ $\quad (FV(\phi)\cap BV(\alpha) = \emptyset)$

✓ $y \geq 0 \rightarrow [x' = -5]\,y \geq 0$

× $x \geq 0 \rightarrow [x' = -5]\,x \geq 0$

✓ $y \geq z \rightarrow [x' = -5]\,y \geq z$

rule out by side conditions

$[:=]$ $[x := \theta]\phi(x) \leftrightarrow \phi(\theta)$ $\quad (\theta$ free for $x$ in $\phi)$

Match all free $x$ occurrences

Replace by $\theta$ everywhere

no $x$ occurrence where $\theta$ bound

✓ $y]\,x \leq y^2$ $y^2$

× $y][y := 5]$ $:= 5]\,x + y \geq 0$

✓ $(x := x+y$ $x \geq y \leftrightarrow [(x := x+2b; x'$ $2b$

✓ $y][x := x+$ $\leftrightarrow [x := x+y+1]\,x \geq 0$

# Axiom Schema Matches Many Formulas But Not All

Algorithm

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

Match shape $\alpha \cup \beta$

Schema variable $\alpha$ match

Same $\phi$ everywhere

$= x +$ $x^2] x$ $x := x + 1] x \geq 0 \wedge [x' = x^2] x \geq 0$

$= 5 \cup$ $x^2 \geq$ $= 5] x^2 \geq 5 \wedge [x' = -x] x^2 \geq 5$

$= v+$ $\cup x' =$ $\leftrightarrow [v := v+1; x' = v] x \geq 5 \wedge [x' = 2] x \geq 4$

V $\phi \rightarrow [\alpha]\phi$ $(FV(\phi) \cap BV(\alpha) = \emptyset)$

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

rule out by side conditions

[:=] $[x := \theta]\phi(x) \leftrightarrow \phi(\theta)$ $(\theta$ free for $x$ in $\phi)$

Match all free $x$ occurrences

Replace by $\theta$ everywhere

no $x$ occurrence where $\theta$ bound

- ✓ $y] x \leq y^2$ $y^2$
- × $y][y := 5]$ $:= 5] x + y \geq 0$
- ✓ $(x := x+y$ $x \geq y \leftrightarrow [(x := x+2b; x'$ $\geq 2b$
- ✓ $y][x := x +$ $\leftrightarrow [x := x + y + 1] x \geq 0$

# Axiom Schema Side Conditions: ODE Solving

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

# Axiom Schema Side Conditions: ODE Solving

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

Axiom schema with side conditions:

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

# Axiom Schema Side Conditions: ODE Solving

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

Axiom schema with side conditions:

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

# Axiom Schema Side Conditions: ODE Solving

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

Axiom schema with side conditions:

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

# Axiom Schema Side Conditions: ODE Solving

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

Axiom schema with side conditions:

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

Quite nontrivial soundness-critical side condition algorithms . . .

# What Axioms Want

V $\phi \rightarrow [\alpha]\phi$

# What Axioms Want

V $\phi \to [\alpha]\phi$ V $p \to [a]p$

V predicate symbol $p$ of arity 0 has no bound variable of HP $a$ free
"Formula $p$ has no explicit permission to depend on anything"
(except implicitly on what doesn't change in $a$ anyhow)

V program constant symbol $a$ could have arbitrary behavior

# What Axioms Want

V $\phi \to [\alpha]\phi$ V $p \to [a]p$

[:=] $[x := \theta]\phi(x) \leftrightarrow \phi(\theta)$

- predicate symbol *p* of arity 0 has no bound variable of HP *a* free
  “Formula *p* has no explicit permission to depend on anything”
  (except implicitly on what doesn’t change in *a* anyhow)

- program constant symbol *a* could have arbitrary behavior

# What Axioms Want

V $\phi \rightarrow [\alpha]\phi$ V $p \rightarrow [a]p$

[:=] $[x := \theta]\phi(x) \leftrightarrow \phi(\theta)$ [:=] $[x := c]p(x) \leftrightarrow p(c)$

V predicate symbol *p* of arity 0 has no bound variable of HP *a* free
“Formula *p* has no explicit permission to depend on anything”
(except implicitly on what doesn’t change in *a* anyhow)

[:=] predicate symbol *p* of arity 1 has different arguments in different places
“Formula $p(x)$ has explicit permission to depend on *x*”

[:=] function symbol *c* of arity 0 takes no arguments

V program constant symbol *a* could have arbitrary behavior

# What Axioms Want

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

V $\phi \rightarrow [\alpha]\phi$ V $p \rightarrow [a]p$

$[:=]$ $[x := \theta]\phi(x) \leftrightarrow \phi(\theta)$ $[:=]$ $[x := c]p(x) \leftrightarrow p(c)$

V predicate symbol $p$ of arity 0 has no bound variable of HP $a$ free
“Formula $p$ has no explicit permission to depend on anything”
(except implicitly on what doesn’t change in $a$ anyhow)

$[:=]$ predicate symbol $p$ of arity 1 has different arguments in different places
“Formula $p(x)$ has explicit permission to depend on $x$”

$[:=]$ function symbol $c$ of arity 0 takes no arguments

V program constant symbol $a$ could have arbitrary behavior

# What Axioms Want

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

V $\phi \rightarrow [\alpha]\phi$ V $p \rightarrow [a]p$

$[:=]$ $[x:=\theta]\phi(x) \leftrightarrow \phi(\theta)$ $[:=]$ $[x:=c]p(x) \leftrightarrow p(c)$

V predicate symbol *p* of arity 0 has no bound variable of HP *a* free
“Formula *p* has no explicit permission to depend on anything”
(except implicitly on what doesn’t change in *a* anyhow)

$[:=]$ predicate symbol *p* of arity 1 has different arguments in different places
“Formula $p(x)$ has explicit permission to depend on *x*”

$[\cup]$ predicate symbol *p* of arity *n* takes all variables $\bar{x}$ as arguments
“Formula $p(\bar{x})$ has explicit permission to depend on all variables $\bar{x}$”

$[:=]$ function symbol *c* of arity 0 takes no arguments

V program constant symbol *a* could have arbitrary behavior

# Outline

1. Learning Objectives
2. Axioms Versus Axiom Schemata
3. Differential Dynamic Logic with Interpretations
   - Syntax
   - Semantics
4. Uniform Substitution
   - Uniform Substitution Application
   - Uniform Substitution Lemmas
5. Axiomatic Proof Calculus for dL
6. Summary

# Differential Dynamic Logic with Interpretations: Syntax

## Definition (Hybrid program $\alpha$)

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

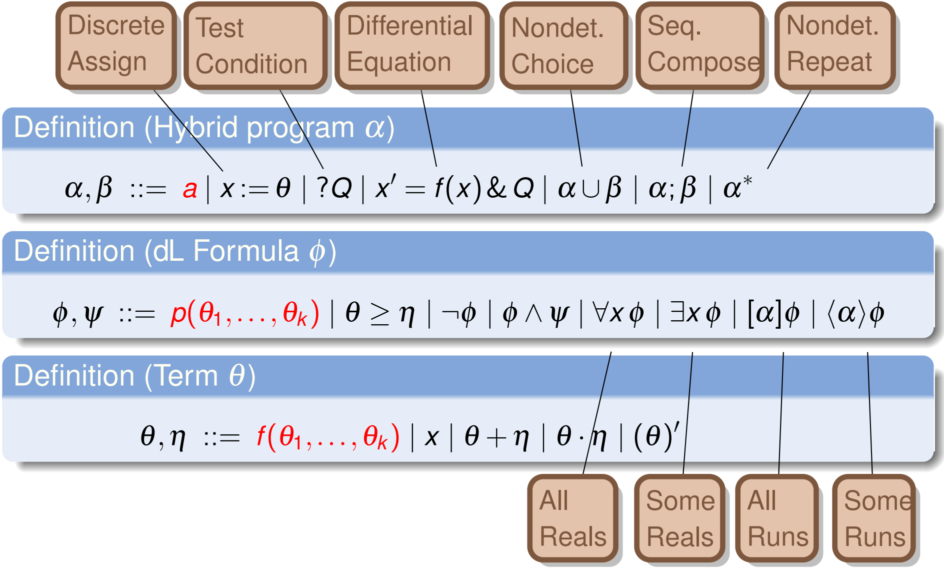
## Definition (dL Formula $\phi$)

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

## Definition (Term $\theta$)

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

# Differential Dynamic Logic with Interpretations: Syntax



### Definition (Hybrid program $\alpha$)

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

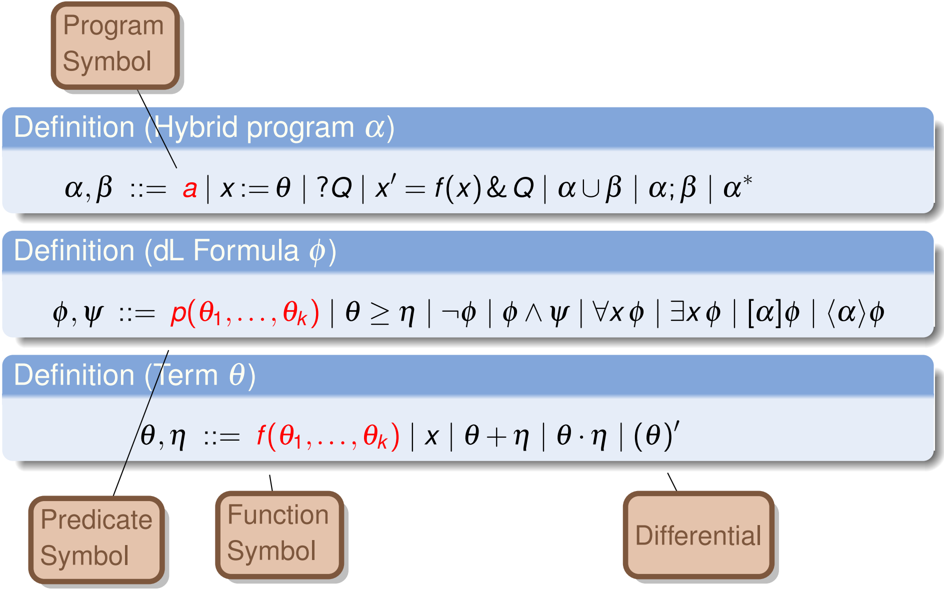
### Definition (dL Formula $\phi$)

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

### Definition (Term $\theta$)

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

# Differential Dynamic Logic with Interpretations: Syntax



Program Symbol

**Definition (Hybrid program $\alpha$)**

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

**Definition (dL Formula $\phi$)**

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

**Definition (Term $\theta$)**

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

Predicate Symbol

Function Symbol

Differential

# Differential Dynamic Logic with Interpretations: Semantics

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

$$\begin{aligned} \omega\llbracket f(\theta_1,\dots,\theta_k)\rrbracket &= I(f)\big(\omega\llbracket\theta_1\rrbracket,\dots,\omega\llbracket\theta_k\rrbracket\big) \qquad I(f) : \mathbb{R}^k \to \mathbb{R} \text{ smooth} \\ \omega\llbracket(\theta)'\rrbracket &= \sum_x \omega(x')\frac{\partial\llbracket\theta\rrbracket}{\partial x}(\omega) \end{aligned}$$

**Definition (dL semantics)** $(\llbracket\cdot\rrbracket : \mathrm{Fml} \to \wp(\mathscr{S}))$

$$\begin{aligned} \llbracket p(\theta_1,\dots,\theta_k)\rrbracket &= \{\omega : (\omega\llbracket\theta_1\rrbracket,\dots,\omega\llbracket\theta_k\rrbracket) \in I(p)\} \qquad I(p) \subseteq \mathbb{R}^k \\ \llbracket\langle\alpha\rangle\phi\rrbracket &= \llbracket\alpha\rrbracket \circ \llbracket\phi\rrbracket \end{aligned}$$

$P$ valid iff $\omega \in \llbracket P\rrbracket$ for all states $\omega$ of all interpretations $I$

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

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

# Soundness Proofs for Axioms

**Lemma (V vacuous axiom)**

$$\text{V} \quad p \rightarrow [a]p$$

**Lemma ([:=] assignment axiom)**

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

# Soundness Proofs for Axioms

**Lemma (V vacuous axiom)**

$$\text{V} \quad p \rightarrow [a]p$$

**Proof.**

Truth of an arity 0 predicate symbol *p* depends only on interpretation *I*.

1. *I* interprets *p* as *true*: $\omega \in [\![p]\!]$ for all $\omega$, so $\omega \in [\![[a]p]\!]$ especially.
2. *I* interprets *p* as *false*: $\omega \notin [\![p]\!]$ for all $\omega$, so $p \rightarrow [a]p$ vacuously. □

**Lemma ([:=] assignment axiom)**

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

**Proof.**

*p* is *true* of *x* after assigning the new value *c* to *x* ($\omega \in [\![[x := c]p(x)]\!]$) iff *p* is *true* of the new value *c* ($\omega \in [\![p(c)]\!]$). □

# Outline

1. Learning Objectives
2. Axioms Versus Axiom Schemata
3. Differential Dynamic Logic with Interpretations
    - Syntax
    - Semantics
4. Uniform Substitution
    - Uniform Substitution Application
    - Uniform Substitution Lemmas
5. Axiomatic Proof Calculus for dL
6. Summary

# Uniform Substitution

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

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

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

# Uniform Substitution

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

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

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

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

# Uniform Substitution

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

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

Uniform substitution $\sigma$ replaces all occurrences of $p(\theta)$ for any $\theta$ by $\psi(\theta)$
function sym. $f(\theta)$ for any $\theta$ by $\eta(\theta)$
program sym. $a$ by $\alpha$

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

# Uniform Substitution: First-Order Examples

$$\frac{(\neg\neg p) \leftrightarrow p}{(\neg\neg[x' = x^2]\, x \geq 0) \leftrightarrow [x' = x^2]\, x \geq 0} \qquad \sigma = \{p \mapsto [x' = x^2]\, x \geq 0\}$$

$$\frac{(\forall x\, p) \leftrightarrow p}{\forall x\, (x \geq 0) \leftrightarrow x \geq 0} \qquad \sigma = \{p \mapsto x \geq 0\}$$

$$\frac{(\forall x\, p) \leftrightarrow p}{\forall x\, (y \geq 0) \leftrightarrow y \geq 0} \qquad \sigma = \{p \mapsto y \geq 0\}$$

# Uniform Substitution: First-Order Examples

$$\frac{(\neg\neg p) \leftrightarrow p}{(\neg\neg[x' = x^2]\,x \geq 0) \leftrightarrow [x' = x^2]\,x \geq 0} \qquad \sigma = \{p \mapsto [x' = x^2]\,x \geq 0\}$$

Correct

$$\frac{(\forall x\, p) \leftrightarrow p}{\forall x\,(x \geq 0) \leftrightarrow x \geq 0} \qquad \sigma = \{p \mapsto x \geq 0\}$$

$$\frac{(\forall x\, p) \leftrightarrow p}{\forall x\,(y \geq 0) \leftrightarrow y \geq 0} \qquad \sigma = \{p \mapsto y \geq 0\}$$

# Uniform Substitution: First-Order Examples

Correct

$$\frac{(\neg\neg p) \leftrightarrow p}{(\neg\neg[x' = x^2]\, x \geq 0) \leftrightarrow [x' = x^2]\, x \geq 0} \qquad \sigma = \{p \mapsto [x' = x^2]\, x \geq 0\}$$

BV

Clash

FV

$$\frac{(\forall x\, p) \leftrightarrow p}{\forall x\, (x \geq 0) \leftrightarrow x \geq 0} \qquad \sigma = \{p \mapsto x \geq 0\}$$

$$\frac{(\forall x\, p) \leftrightarrow p}{\forall x\, (y \geq 0) \leftrightarrow y \geq 0} \qquad \sigma = \{p \mapsto y \geq 0\}$$

# Uniform Substitution: First-Order Examples

$$\frac{(\neg\neg p) \leftrightarrow p}{(\neg\neg[x' = x^2]\, x \geq 0) \leftrightarrow [x' = x^2]\, x \geq 0}$$

Correct

$$\sigma = \{p \mapsto [x' = x^2]\, x \geq 0\}$$

$$\frac{(\forall x\, p) \leftrightarrow p}{\forall x\, (x \geq 0) \leftrightarrow x \geq 0}$$

Clash

$$\sigma = \{p \mapsto x \geq 0\}$$

$$\frac{(\forall x\, p) \leftrightarrow p}{\forall x\, (y \geq 0) \leftrightarrow y \geq 0}$$

Correct

$$\sigma = \{p \mapsto y \geq 0\}$$

# Uniform Substitution: Argument Examples

$$\frac{[x:=c]p(x) \leftrightarrow p(c)}{[x:=x^2-1]\,x \geq 0 \leftrightarrow x^2-1 \geq 0} \qquad \sigma = \{c \mapsto x^2-1, p(\cdot) \mapsto (\cdot \geq 0)\}$$

$$\frac{[x:=c]p(x) \leftrightarrow p(c)}{[x:=x^2-1]x \geq x \leftrightarrow x^2-1 \geq x} \qquad \sigma = \{c \mapsto x^2-1, p(\cdot) \mapsto (\cdot \geq x)\}$$

$$\frac{[x:=c]p(x) \leftrightarrow p(c)}{[x:=x^2-1]\,x \geq x \leftrightarrow x^2-1 \geq x^2-1} \qquad \sigma = \{c \mapsto x^2-1, p(\cdot) \mapsto (\cdot \geq \cdot)\}$$

$$\frac{[x:=c]p(x) \leftrightarrow p(c)}{[x:=x^2-1]\,x \geq y \leftrightarrow x^2-1 \geq y} \qquad \sigma = \{c \mapsto x^2-1, p(\cdot) \mapsto (\cdot \geq y)\}$$

# Uniform Substitution: Argument Examples

$$\frac{[x := c]p(x) \leftrightarrow p(c)}{[x := x^2 - 1]\, x \geq 0 \leftrightarrow x^2 - 1 \geq 0} \qquad \sigma = \{c \mapsto x^2 - 1, p(\cdot) \mapsto (\cdot \geq 0)\}$$

Correct

$$\frac{[x := c]p(x) \leftrightarrow p(c)}{[x := x^2 - 1]x \geq x \leftrightarrow x^2 - 1 \geq x} \qquad \sigma = \{c \mapsto x^2 - 1, p(\cdot) \mapsto (\cdot \geq x)\}$$

$$\frac{[x := c]p(x) \leftrightarrow p(c)}{[x := x^2 - 1]\, x \geq x \leftrightarrow x^2 - 1 \geq x^2 - 1} \qquad \sigma = \{c \mapsto x^2 - 1, p(\cdot) \mapsto (\cdot \geq \cdot)\}$$

$$\frac{[x := c]p(x) \leftrightarrow p(c)}{[x := x^2 - 1]\, x \geq y \leftrightarrow x^2 - 1 \geq y} \qquad \sigma = \{c \mapsto x^2 - 1, p(\cdot) \mapsto (\cdot \geq y)\}$$

# Uniform Substitution: Argument Examples

Correct

$$\frac{[x:=c]p(x) \leftrightarrow p(c)}{[x:=x^2-1]\,x \geq 0 \leftrightarrow x^2-1 \geq 0} \qquad \sigma = \{c \mapsto x^2-1, p(\cdot) \mapsto (\cdot \geq 0)\}$$

$$\frac{[x:=c]p(x) \leftrightarrow p(c)}{[x:=x^2-1]x \geq x \leftrightarrow x^2-1 \geq x} \qquad \sigma = \{c \mapsto x^2-1, p(\cdot) \mapsto (\cdot \geq x)\}$$

$$\frac{[x:=c]p(x) \leftrightarrow p(c)}{[x:=x^2-1]\,x \geq x \leftrightarrow x^2-1 \geq x^2-1} \qquad \sigma = \{c \mapsto x^2-1, p(\cdot) \mapsto (\cdot \geq \cdot)\}$$

$$\frac{[x:=c]p(x) \leftrightarrow p(c)}{[x:=x^2-1]\,x \geq y \leftrightarrow x^2-1 \geq y} \qquad \sigma = \{c \mapsto x^2-1, p(\cdot) \mapsto (\cdot \geq y)\}$$

# Uniform Substitution: Argument Examples

Correct

$$\frac{[x:=c]p(x) \leftrightarrow p(c)}{[x:=x^2-1]\,x \geq 0 \leftrightarrow x^2-1 \geq 0} \qquad \sigma = \{c \mapsto x^2-1, p(\cdot) \mapsto (\cdot \geq 0)\}$$

BV

Clash

FV

$$\frac{[x:=c]p(x) \leftrightarrow p(c)}{[x:=x^2-1]x \geq x \leftrightarrow x^2-1 \geq x} \qquad \sigma = \{c \mapsto x^2-1, p(\cdot) \mapsto (\cdot \geq x)\}$$

$$\frac{[x:=c]p(x) \leftrightarrow p(c)}{[x:=x^2-1]\,x \geq x \leftrightarrow x^2-1 \geq x^2-1} \qquad \sigma = \{c \mapsto x^2-1, p(\cdot) \mapsto (\cdot \geq \cdot)\}$$

$$\frac{[x:=c]p(x) \leftrightarrow p(c)}{[x:=x^2-1]\,x \geq y \leftrightarrow x^2-1 \geq y} \qquad \sigma = \{c \mapsto x^2-1, p(\cdot) \mapsto (\cdot \geq y)\}$$

# Uniform Substitution: Argument Examples

$$\frac{[x := c]p(x) \leftrightarrow p(c)}{[x := x^2-1]\, x \geq 0 \leftrightarrow x^2-1 \geq 0} \qquad \sigma = \{c \mapsto x^2-1, p(\cdot) \mapsto (\cdot \geq 0)\}$$

Correct

$$\frac{[x := c]p(x) \leftrightarrow p(c)}{[x := x^2-1] x \geq x \leftrightarrow x^2-1 \geq x} \qquad \sigma = \{c \mapsto x^2-1, p(\cdot) \mapsto (\cdot \geq x)\}$$

Clash

$$\frac{[x := c]p(x) \leftrightarrow p(c)}{[x := x^2-1]\, x \geq x \leftrightarrow x^2-1 \geq x^2-1} \qquad \sigma = \{c \mapsto x^2-1, p(\cdot) \mapsto (\cdot \geq \cdot)\}$$

Correct

$$\frac{[x := c]p(x) \leftrightarrow p(c)}{[x := x^2-1]\, x \geq y \leftrightarrow x^2-1 \geq y} \qquad \sigma = \{c \mapsto x^2-1, p(\cdot) \mapsto (\cdot \geq y)\}$$

# Uniform Substitution: Argument Examples

$$\frac{[x:=c]p(x) \leftrightarrow p(c)}{[x:=x^2-1]\,x \geq 0 \leftrightarrow x^2-1 \geq 0}$$

Correct

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

$$\frac{[x:=c]p(x) \leftrightarrow p(c)}{[x:=x^2-1]x \geq x \leftrightarrow x^2-1 \geq x}$$

Clash

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

$$\frac{[x:=c]p(x) \leftrightarrow p(c)}{[x:=x^2-1]\,x \geq x \leftrightarrow x^2-1 \geq x^2-1}$$

Correct

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

$$\frac{[x:=c]p(x) \leftrightarrow p(c)}{[x:=x^2-1]\,x \geq y \leftrightarrow x^2-1 \geq y}$$

Correct

$$\sigma = \{c \mapsto x^2-1, p(\cdot) \mapsto (\cdot \geq y)\}$$

# Uniform Substitution

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

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

Uniform substitution $\sigma$ replaces all occurrences of $p(\theta)$ for any $\theta$ by $\psi(\theta)$
function sym. $f(\theta)$ for any $\theta$ by $\eta(\theta)$
program sym. $a$ by $\alpha$

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

# Uniform Substitution

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

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

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

i.e. bound variables $U = \mathrm{BV}(\otimes(\cdot))$ of **no** operator $\otimes$
are free in the substitution on its argument $\theta$ ($U$-admissible)

Uniform substitution $\sigma$ replaces all occurrences of $p(\theta)$ for any $\theta$ by $\psi(\theta)$
function sym. $f(\theta)$ for any $\theta$ by $\eta(\theta)$
program sym. $a$ by $\alpha$

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

# 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* $\phi$

i.e. bound variables $U = \mathrm{BV}(\otimes(\cdot))$ of **no** operator $\otimes$
are free in the substitution on its argument $\theta$ ($U$-admissible)

If you bind a free variable, you go to logic jail!

Uniform substitution $\sigma$ replaces all occurrences of $p(\theta)$ for any $\theta$ by $\psi(\theta)$
function sym. $f(\theta)$ for any $\theta$ by $\eta(\theta)$
program sym. $a$ by $\alpha$

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

# Uniform Substitution: Recursive Application

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

# Uniform Substitution: Recursive Application

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

# Uniform Substitution: Recursive Application

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

# Uniform Substitution: Recursive Application

$$
\begin{aligned}
\sigma(x) &= x && \text{for variable } x \in \mathscr{V} \\
\sigma(f(\theta)) &= (\sigma(f))(\sigma(\theta)) && \text{for function symbol } f \in \sigma \\
&\stackrel{\text{def}}{=} \{\cdot \mapsto \sigma(\theta)\}(\sigma f(\cdot)) \\
\sigma(\theta + \eta) &= \sigma(\theta) + \sigma(\eta) \\
\sigma((\theta)') &=
\end{aligned}
$$

$$
\begin{aligned}
\sigma(p(\theta)) &\equiv && \text{for predicate symbol } p \in \sigma \\
\sigma(\phi \land \psi) &\equiv \\
\sigma(\forall x \, \phi) &= \\
\sigma([\alpha]\phi) &=
\end{aligned}
$$

$$
\begin{aligned}
\sigma(a) &\equiv && \text{for program symbol } a \in \sigma \\
\sigma(x := \theta) &\equiv \\
\sigma(x' = \theta \,\&\, Q) &\equiv \\
\sigma(?Q) &\equiv \\
\sigma(\alpha \cup \beta) &\equiv \\
\sigma(\alpha ; \beta) &\equiv \\
\sigma(\alpha^*) &\equiv
\end{aligned}
$$

# Uniform Substitution: Recursive Application

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

# Uniform Substitution: Recursive Application

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

# Uniform Substitution: Recursive Application

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

# Uniform Substitution: Recursive Application

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

# Uniform Substitution: Recursive Application

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

# Uniform Substitution: Recursive Application

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

# Uniform Substitution: Recursive Application

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

# Uniform Substitution: Recursive Application

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

# Uniform Substitution: Recursive Application

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

# Uniform Substitution: Recursive Application

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

# Uniform Substitution: Recursive Application

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

# Uniform Substitution: Recursive Application

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

# Uniform Substitution: Examples

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

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

$$\sigma = \{c \mapsto x^2, p(\cdot) \mapsto [(y:=\cdot+y)^*](\cdot \geq y)\}$$

$$\frac{p \rightarrow [a]p}{x \geq 0 \rightarrow [x'=-5]x \geq 0} \quad \sigma = \{a \mapsto x'=-5, p \mapsto x \geq 0\}$$

$$\frac{p \rightarrow [a]p}{y \geq 0 \rightarrow [x'=-5]y \geq 0} \quad \sigma = \{a \mapsto x'=-5, p \mapsto y \geq 0\}$$

# Uniform Substitution: Examples

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

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

$$\sigma = \{c \mapsto x^2, p(\cdot) \mapsto [(y := \cdot + y)^*](\cdot \geq y)\}$$

$$\frac{p \rightarrow [a]p}{x \geq 0 \rightarrow [x' = -5]x \geq 0} \quad \sigma = \{a \mapsto x' = -5, p \mapsto x \geq 0\}$$

$$\frac{p \rightarrow [a]p}{y \geq 0 \rightarrow [x' = -5]y \geq 0} \quad \sigma = \{a \mapsto x' = -5, p \mapsto y \geq 0\}$$

# Uniform Substitution: Examples

BV

Clash

FV

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

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

$$\sigma = \{c \mapsto x^2, p(\cdot) \mapsto [(y := \cdot + y)^*](\cdot \geq y)\}$$

$$\frac{p \to [a]p}{x \geq 0 \to [x' = -5]x \geq 0} \qquad \sigma = \{a \mapsto x' = -5, p \mapsto x \geq 0\}$$

$$\frac{p \to [a]p}{y \geq 0 \to [x' = -5]y \geq 0} \qquad \sigma = \{a \mapsto x' = -5, p \mapsto y \geq 0\}$$

# Uniform Substitution: Examples

Clash

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

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

$$\sigma = \{c \mapsto x^2, p(\cdot) \mapsto [(y := \cdot + y)^*](\cdot \geq y)\}$$

$$\frac{p \rightarrow [a]p}{x \geq 0 \rightarrow [x' = -5]x \geq 0} \qquad \sigma = \{a \mapsto x' = -5, p \mapsto x \geq 0\}$$

$$\frac{p \rightarrow [a]p}{y \geq 0 \rightarrow [x' = -5]y \geq 0} \qquad \sigma = \{a \mapsto x' = -5, p \mapsto y \geq 0\}$$

# Uniform Substitution: Examples

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

Clash

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

Correct

$$\sigma = \{c \mapsto x^2, p(\cdot) \mapsto [(y := \cdot + y)^*](\cdot \geq y)\}$$

$$\frac{p \rightarrow [a]p}{x \geq 0 \rightarrow [x' = -5]x \geq 0} \qquad \sigma = \{a \mapsto x' = -5, p \mapsto x \geq 0\}$$

$$\frac{p \rightarrow [a]p}{y \geq 0 \rightarrow [x' = -5]y \geq 0} \qquad \sigma = \{a \mapsto x' = -5, p \mapsto y \geq 0\}$$

# Uniform Substitution: Examples

Clash

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

Correct

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

$$\sigma = \{c \mapsto x^2, p(\cdot) \mapsto [(y := \cdot + y)^*](\cdot \geq y)\}$$

BV

Clash

FV

$$\frac{[a]p}{x \geq 0 \rightarrow [x' = -5]x \geq 0} \qquad \sigma = \{a \mapsto x' = -5, p \mapsto x \geq 0\}$$

$$\frac{p \rightarrow [a]p}{y \geq 0 \rightarrow [x' = -5]y \geq 0} \qquad \sigma = \{a \mapsto x' = -5, p \mapsto y \geq 0\}$$

# Uniform Substitution: Examples

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

Clash

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

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

Correct

$$\sigma = \{c \mapsto x^2, p(\cdot) \mapsto [(y := \cdot + y)^*](\cdot \geq y)\}$$

$$\frac{p \to [a]p}{x \geq 0 \to [x' = -5]x \geq 0}$$

Clash

$$\sigma = \{a \mapsto x' = -5, p \mapsto x \geq 0\}$$

$$\frac{p \to [a]p}{y \geq 0 \to [x' = -5]y \geq 0}$$

Correct

$$\sigma = \{a \mapsto x' = -5, p \mapsto y \geq 0\}$$

# Uniform Substitution

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

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

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

i.e. bound variables $U = \mathrm{BV}(\otimes(\cdot))$ of **no** operator $\otimes$
are free in the substitution on its argument $\theta$ ($U$-admissible)

If you bind a free variable, you go to logic jail!

Uniform substitution $\sigma$ replaces all occurrences of $p(\theta)$ for any $\theta$ by $\psi(\theta)$
function sym. $f(\theta)$ for any $\theta$ by $\eta(\theta)$
program sym. $a$ by $\alpha$

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

# Correctness of Uniform Substitutions

“Syntactic uniform substitution = semantic replacement”

## Lemma (Uniform substitution lemma)

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

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

$$\begin{tikzcd}
\phi \arrow[rr, mapsto, "\sigma"'] \arrow[rrrrdd, mapsto, bend right, "\sigma^*_\omega I" description] & & \sigma(\phi) \arrow[rr, mapsto, "I"'] & & \omega \in I[\![\sigma(\phi)]\!] \arrow[dd, equal] \\
& \sigma \arrow[rd, mapsto] & & I \arrow[ld, mapsto] & \\
& & {} & & \omega \in \sigma^*_\omega I[\![\phi]\!]
\end{tikzcd}$$

$$\sigma^*_\omega I(f) : \mathbb{R} \to \mathbb{R};\, d \mapsto I^d_\cdot \omega[\![\sigma f(\cdot)]\!]$$

$$\sigma^*_\omega I(p) = \{d \in \mathbb{R} \;:\; \omega \in I^d_\cdot[\![\sigma p(\cdot)]\!]\}$$

$$\sigma^*_\omega I(a) = I[\![\sigma a]\!]$$

# Uniform Substitution

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

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

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

**Proof.**

If premise $\phi$ valid, i.e. $\omega \in I[\![\phi]\!]$ in all $I, \omega$

Then conclusion $\sigma(\phi)$ valid, because $\omega \in I[\![\sigma(\phi)]\!]$ iff $\omega \in \sigma^*_\omega I[\![\phi]\!]$ □

# Outline

1. Learning Objectives
2. Axioms Versus Axiom Schemata
3. Differential Dynamic Logic with Interpretations
   - Syntax
   - Semantics
4. Uniform Substitution
   - Uniform Substitution Application
   - Uniform Substitution Lemmas
5. **Axiomatic Proof Calculus for dL**
6. Summary

# Differential Dynamic Logic: Comparison

**Part I**

**Part IV**

$$[:=]\ [x := \theta]\phi(x) \leftrightarrow \phi(\theta)$$

$$[?]\ [?\chi]\phi \leftrightarrow (\chi \rightarrow \phi)$$

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

$$[;]\ [\alpha;\beta]\phi \leftrightarrow [\alpha][\beta]\phi$$

$$[^*]\ [\alpha^*]\phi \leftrightarrow \phi \land [\alpha][\alpha^*]\phi$$

$$\mathsf{K}\ [\alpha](\phi \rightarrow \psi) \rightarrow ([\alpha]\phi \rightarrow [\alpha]\psi)$$

$$\mathsf{I}\ [\alpha^*]\phi \leftrightarrow \phi \land [\alpha^*](\phi \rightarrow [\alpha]\phi)$$

$$\mathsf{V}\ \phi \rightarrow [\alpha]\phi$$

$$[']\ [x' = f(x)]\phi \leftrightarrow \forall t{\geq}0\,[x := y(t)]\phi$$

# Differential Dynamic Logic: Comparison

| Part I | Part IV |
|---|---|
| [:=] $[x:=\theta]\phi(x) \leftrightarrow \phi(\theta)$ | [:=] $[x:=c]p(x) \leftrightarrow p(c)$ |
| [?] $[?\chi]\phi \leftrightarrow (\chi \rightarrow \phi)$ | [?] $[?q]p \leftrightarrow (q \rightarrow p)$ |
| [∪] $[\alpha \cup \beta]\phi \leftrightarrow [\alpha]\phi \land [\beta]\phi$ | [∪] $[a \cup b]p(\bar{x}) \leftrightarrow [a]p(\bar{x}) \land [b]p(\bar{x})$ |
| [;] $[\alpha;\beta]\phi \leftrightarrow [\alpha][\beta]\phi$ | [;] $[a;b]p(\bar{x}) \leftrightarrow [a][b]p(\bar{x})$ |
| [*] $[\alpha^*]\phi \leftrightarrow \phi \land [\alpha][\alpha^*]\phi$ | [*] $[a^*]p(\bar{x}) \leftrightarrow p(\bar{x}) \land [a][a^*]p(\bar{x})$ |
| K $[\alpha](\phi \rightarrow \psi) \rightarrow ([\alpha]\phi \rightarrow [\alpha]\psi)$ | K $[a](p(\bar{x}) \rightarrow q(\bar{x})) \rightarrow ([a]p(\bar{x}) \rightarrow [a]q(\bar{x}))$ |
| I $[\alpha^*]\phi \leftrightarrow \phi \land [\alpha^*](\phi \rightarrow [\alpha]\phi)$ | I $[a^*]p(\bar{x}) \leftrightarrow p(\bar{x}) \land [a^*](p(\bar{x}) \rightarrow [a]p(\bar{x}))$ |
| V $\phi \rightarrow [\alpha]\phi$ | V $p \rightarrow [a]p$ |
| [′] $[x' = f(x)]\phi \leftrightarrow \forall t \geq 0\, [x := y(t)]\phi$ | |

# Differential Dynamic Logic: Comparison

| | Infinite axiom schema | | Axiom = one formula |
|---|---|---|---|
| [:=] | $[x := \theta]\phi(x) \leftrightarrow \phi(\theta)$ | [:=] | $[x := c]p(x) \leftrightarrow p(c)$ |
| [?] | $[?\chi]\phi \leftrightarrow (\chi \rightarrow \phi)$<br>Schema | [?] | $[?q]p \leftrightarrow (q \rightarrow p)$<br>Axiom |
| [∪] | $[\alpha \cup \beta]\phi \leftrightarrow [\alpha]\phi \wedge [\beta]\phi$ | [∪] | $[a \cup b]p(\bar{x}) \leftrightarrow [a]p(\bar{x}) \wedge [b]p(\bar{x})$ |
| [;] | $[\alpha;\beta]\phi \leftrightarrow [\alpha][\beta]\phi$ | [;] | $[a;b]p(\bar{x}) \leftrightarrow [a][b]p(\bar{x})$ |
| [*] | $[\alpha^*]\phi \leftrightarrow \phi \wedge [\alpha][\alpha^*]\phi$ | [*] | $[a^*]p(\bar{x}) \leftrightarrow p(\bar{x}) \wedge [a][a^*]p(\bar{x})$ |
| K | $[\alpha](\phi \rightarrow \psi) \rightarrow ([\alpha]\phi \rightarrow [\alpha]\psi)$ | K | $[a](p(\bar{x}) \rightarrow q(\bar{x})) \rightarrow ([a]p(\bar{x}) \rightarrow [a]q(\bar{x}))$ |
| I | $[\alpha^*]\phi \leftarrow$ Schema $^*](\phi \rightarrow [\alpha]\phi)$ | I | $[a^*]p(\bar{x}$ Axiom $\bar{x}) \wedge [a^*](p(\bar{x}) \rightarrow [a]p(\bar{x}))$ |
| V | $\phi \rightarrow [\alpha]\phi$ | V | $p \rightarrow [a]p$ |
| ['] | $[x' = f(x)]\phi \leftrightarrow \forall t \geq 0\, [x := y(t)]\phi$ | | |

# Example Proof

$$\dfrac{}{j(x) \vdash [(v := 2 \cup v := x); x' = v]x > 0}\ [;]$$

# Example Proof

$$\sigma = \{a \mapsto (v:=2 \cup v:=x), b \mapsto x' = v, p(\bar{x}) \mapsto x > 0\}$$

$$\text{US}\frac{[a;b]p(\bar{x}) \leftrightarrow [a][b]p(\bar{x})}{[(v:=2 \cup v:=x);x'=v]x>0 \leftrightarrow [(v:=2 \cup v:=x)][x'=v]x>0}$$

$$[;]\dfrac{[\cup]\dfrac{}{j(x) \vdash [v:=2 \cup v:=x][x'=v]x>0}}{j(x) \vdash [(v:=2 \cup v:=x);x'=v]x>0}$$

# Example Proof

$$\sigma = \{a \mapsto v:=2, b \mapsto v:=x, p(\bar{x}) \mapsto [x'=v]x>0\}$$

$$\text{US}\,\dfrac{[a\cup b]p(\bar{x}) \leftrightarrow [a]p(\bar{x}) \land [b]p(\bar{x})}{[v:=2\cup v:=x][x'=v]x>0 \leftrightarrow [v:=2][x'=v]x>0 \land [v:=x][x'=v]x>0}$$

$${\scriptstyle[;]}\,\dfrac{{\scriptstyle[\cup]}\,\dfrac{{\scriptstyle[:=]}\,\dfrac{}{j(x) \vdash [v:=2][x'=v]x>0 \land [v:=x][x'=v]x>0}}{j(x) \vdash [v:=2\cup v:=x][x'=v]x>0}}{j(x) \vdash [(v:=2\cup v:=x);x'=v]x>0}$$

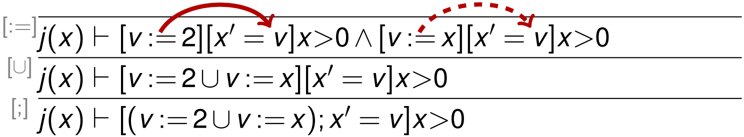
# Example Proof

$$\sigma = \{c \mapsto 2, p(\cdot) \mapsto [x'=\cdot]x>0\}$$

$$\dfrac{[v:=c]p(v) \leftrightarrow p(c)}{[v:=2][x'=v]x>0 \leftrightarrow [x'=2]x>0}$$

$$\sigma = \{c \mapsto x, p(\cdot) \mapsto [x'=\cdot]x>0\}$$

$$\dfrac{[v:=c]p(v) \leftrightarrow p(c)}{[v:=x][x'=v]x>0 \leftrightarrow [x'=x]x>0}$$



$$
{\scriptstyle[;]}\;\dfrac{{\scriptstyle[\cup]}\;\dfrac{{\scriptstyle[:=]}\;\dfrac{}{j(x) \vdash [v:=2][x' = v]x>0 \land [v:=x][x' = v]x>0}}{j(x) \vdash [v:=2 \cup v:=x][x' = v]x>0}}{j(x) \vdash [(v:=2 \cup v:=x); x' = v]x>0}
$$

# Example Proof

$$\sigma = \{c \mapsto 2, p(\cdot) \mapsto [x'{=}\cdot]x>0\}$$

$$\frac{[v:=c]p(v) \leftrightarrow p(c)}{[v:=2][x'{=}v]x>0 \leftrightarrow [x'{=}2]x>0}$$

$$\sigma = \{c \mapsto x, p(\cdot) \mapsto [x'{=}\cdot]x>0\}$$

$$\frac{[v:=c]p(v) \leftrightarrow p(c)}{[v:=x][x'{=}v]x>0 \leftrightarrow [x'{=}x]x>0} \;\lightning$$

$${\scriptstyle [;]}\dfrac{{\scriptstyle [\cup]}\dfrac{{\scriptstyle [:=]}\dfrac{{\scriptstyle [']}\dfrac{}{j(x) \vdash [x' = 2]x>0 \land [v:=x][x' = v]x>0}}{j(x) \vdash [v:=2][x' = v]x>0 \land [v:=x][x' = v]x>0}}{j(x) \vdash [v:=2 \cup v:=x][x' = v]x>0}}{j(x) \vdash [(v:=2 \cup v:=x); x' = v]x>0}$$

# Example Proof

$$\sigma = \{c \mapsto v, p(\cdot) \mapsto \cdot>0\}$$

*v* can't have ODE

$$\text{US}\,\frac{[x' = c]p(x) \leftrightarrow \forall t{\geq}0\,[x := x+ct]p(x)}{[x' = v]x>0 \leftrightarrow \forall t{\geq}0\,[x := x+vt]x>0}$$

$$\begin{array}{ll}
\hline
{[:=]} & j(x) \vdash \forall t{\geq}0\,[x := x+2t]x>0 \land [v := x]\forall t{\geq}0\,[x := x+vt]x>0 \\ \hline
{[']} & j(x) \vdash [x' = 2]x>0 \land [v := x][x' = v]x>0 \\ \hline
{[:=]} & j(x) \vdash [v := 2][x' = v]x>0 \land [v := x][x' = v]x>0 \\ \hline
{[\cup]} & j(x) \vdash [v := 2 \cup v := x][x' = v]x>0 \\ \hline
{[;]} & j(x) \vdash [(v := 2 \cup v := x); x' = v]x>0
\end{array}$$

# Example Proof

$$\sigma = \{c \mapsto x, p(\cdot) \mapsto \forall t{\geq}0\, [x := x+(\cdot)t]x>0\}$$

$$\text{US}\,\dfrac{[v := c]p(v) \leftrightarrow p(c)}{[v := x]\forall t{\geq}0\, [x := x+vt]x>0 \leftrightarrow \forall t{\geq}0\, [x := x+xt]x>0}$$

$$
{\scriptstyle [;]}\dfrac{
{\scriptstyle [\cup]}\dfrac{
{\scriptstyle [:=]}\dfrac{
{\scriptstyle [']}\dfrac{
{\scriptstyle [:=]}\dfrac{
{\scriptstyle [:=]}\dfrac{}{j(x) \vdash \forall t{\geq}0\, x+2t>0 \land \forall t{\geq}0\, [x := x+xt]x>0}
}{j(x) \vdash \forall t{\geq}0\, [x := x+2t]x>0 \land [v := x]\forall t{\geq}0\, [x := x+vt]x>0}
}{j(x) \vdash [x' = 2]x>0 \land [v := x][x' = v]x>0}
}{j(x) \vdash [v := 2][x' = v]x>0 \land [v := x][x' = v]x>0}
}{j(x) \vdash [v := 2 \cup v := x][x' = v]x>0}
}{j(x) \vdash [(v := 2 \cup v := x); x' = v]x>0}
$$

# Example Proof

$$\sigma = \{c \mapsto x+xt, p(\cdot) \mapsto \cdot>0\}$$

$$\text{US}\,\dfrac{[x:=c]p(x) \leftrightarrow p(c)}{[x:=x+xt]x>0 \leftrightarrow x+xt>0}$$

$$\begin{array}{rl}
 & j(x) \vdash \forall t{\geq}0\, x+2t>0 \land \forall t{\geq}0\, x+xt>0 \\ \cline{2-2}
{[:=]} & j(x) \vdash \forall t{\geq}0\, x+2t>0 \land \forall t{\geq}0\, [x:=x+xt]x>0 \\ \cline{2-2}
{[:=]} & j(x) \vdash \forall t{\geq}0\, [x:=x+2t]x>0 \land [v:=x]\forall t{\geq}0\, [x:=x+vt]x>0 \\ \cline{2-2}
{[']} & j(x) \vdash [x'=2]x>0 \land [v:=x][x'=v]x>0 \\ \cline{2-2}
{[:=]} & j(x) \vdash [v:=2][x'=v]x>0 \land [v:=x][x'=v]x>0 \\ \cline{2-2}
{[\cup]} & j(x) \vdash [v:=2 \cup v:=x][x'=v]x>0 \\ \cline{2-2}
{[;]} & j(x) \vdash [(v:=2 \cup v:=x);x'=v]x>0
\end{array}$$

# Example Proof

$$
\begin{array}{r l}
 & j(x) \vdash \forall t{\geq}0\, x{+}2t{>}0 \land \forall t{\geq}0\, x{+}xt{>}0 \\ \hline
{[:=]} & j(x) \vdash \forall t{\geq}0\, x{+}2t{>}0 \land \forall t{\geq}0\, [x:=x{+}xt]x{>}0 \\ \hline
{[:=]} & j(x) \vdash \forall t{\geq}0\, [x:=x{+}2t]x{>}0 \land [v:=x]\forall t{\geq}0\, [x:=x{+}vt]x{>}0 \\ \hline
{[']} & j(x) \vdash [x'=2]x{>}0 \land [v:=x][x'=v]x{>}0 \\ \hline
{[:=]} & j(x) \vdash [v:=2][x'=v]x{>}0 \land [v:=x][x'=v]x{>}0 \\ \hline
{[\cup]} & j(x) \vdash [v:=2 \cup v:=x][x'=v]x{>}0 \\ \hline
{[;]} & j(x) \vdash [(v:=2 \cup v:=x); x'=v]x{>}0
\end{array}
$$

# Example Proof

Summarize:

$$\frac{j(x) \vdash \forall t{\geq}0\, x{+}2t{>}0 \land \forall t{\geq}0\, x{+}xt{>}0}{j(x) \vdash [(v := 2 \cup v := x); x' = v]x > 0}$$

# Example Proof

Summarize:

$$\frac{j(x) \vdash \forall t{\geq}0\, x{+}2t{>}0 \land \forall t{\geq}0\, x{+}xt{>}0}{j(x) \vdash [(v := 2 \cup v := x); x' = v]x > 0}$$

Using $\sigma = \{j(\cdot) \mapsto \cdot{>}0\}$ on above derived rule proves:

$$\dfrac{\mathbb{R}\ \dfrac{*}{x > 0 \vdash \forall t{\geq}0\, x + 2t > 0 \land \forall t{\geq}0\, x + xt > 0}}{x > 0 \vdash [(v := 2 \cup v := x); x' = v]x > 0}\ \text{USR}$$

# Outline

1. Learning Objectives
2. Axioms Versus Axiom Schemata
3. Differential Dynamic Logic with Interpretations
   - Syntax
   - Semantics
4. Uniform Substitution
   - Uniform Substitution Application
   - Uniform Substitution Lemmas
5. Axiomatic Proof Calculus for dL
6. **Summary**

# Axiom vs. Axiom Schema: 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

# Axiom vs. Axiom Schema: 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

---

∑ Net win for soundness since significantly simpler prover

# Differential Dynamic Logic: Comparison

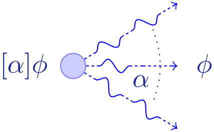
| Part I | Part IV |
|---|---|
| $[:=]\ [x:=\theta]\phi(x) \leftrightarrow \phi(\theta)$ | $[:=]\ [x:=c]p(x) \leftrightarrow p(c)$ |
| $[?]\ [?\chi]\phi \leftrightarrow (\chi \rightarrow \phi)$ | $[?]\ [?q]p \leftrightarrow (q \rightarrow p)$ |
| $[\cup]\ [\alpha \cup \beta]\phi \leftrightarrow [\alpha]\phi \land [\beta]\phi$ | $[\cup]\ [a \cup b]p(\bar{x}) \leftrightarrow [a]p(\bar{x}) \land [b]p(\bar{x})$ |
| $[;]\ [\alpha;\beta]\phi \leftrightarrow [\alpha][\beta]\phi$ | $[;]\ [a;b]p(\bar{x}) \leftrightarrow [a][b]p(\bar{x})$ |
| $[^*]\ [\alpha^*]\phi \leftrightarrow \phi \land [\alpha][\alpha^*]\phi$ | $[^*]\ [a^*]p(\bar{x}) \leftrightarrow p(\bar{x}) \land [a][a^*]p(\bar{x})$ |
| $\mathrm{K}\ [\alpha](\phi \rightarrow \psi) \rightarrow ([\alpha]\phi \rightarrow [\alpha]\psi)$ | $\mathrm{K}\ [a](p(\bar{x}) \rightarrow q(\bar{x})) \rightarrow ([a]p(\bar{x}) \rightarrow [a]q(\bar{x}))$ |
| $\mathrm{I}\ [\alpha^*]\phi \leftrightarrow \phi \land [\alpha^*](\phi \rightarrow [\alpha]\phi)$ | $\mathrm{I}\ [a^*]p(\bar{x}) \leftrightarrow p(\bar{x}) \land [a^*](p(\bar{x}) \rightarrow [a]p(\bar{x}))$ |
| $\mathrm{V}\ \phi \rightarrow [\alpha]\phi$ | $\mathrm{V}\ p \rightarrow [a]p$ |
| $[']\ [x' = f(x)]\phi \leftrightarrow \forall t \geq 0\, [x := y(t)]\phi$ | |

# Uniform Substitution for Differential Dynamic Logic

**differential dynamic logic**

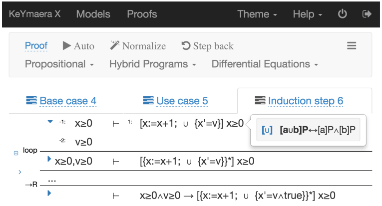
dL = DL + HP

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



- Uniform substitution
  ⇝ axioms not schemata
- Modular: Logic ‖ Prover
- Straightforward to implement
- Prover microkernel
- Sound & complete / ODE
- Fast contextual equivalence

## KeYmaera X



```
-1: x≥0   ⊢  1: [x:=x+1; ∪ {x'=v}] x≥0
-2: v≥0
loop
x≥0,v≥0   ⊢  [{x:=x+1; ∪ {x'=v}}*] x≥0
...
→R
          ⊢  x≥0∧v≥0 → [{x:=x+1; ∪ {x'=v∧true}}*] x≥0
```

[∪] **[a∪b]P**↔[a]P∧[b]P

# Uniform Substitution of Rules and Proofs

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

# Uniform Substitution of Rules and Proofs

G $\dfrac{p(\bar{x})}{[a]p(\bar{x})}$ implies $\dfrac{x^2 \geq 0}{[x := x+1; (x' = x \cup x' = -2)]x^2 \geq 0}$

**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 of Rules and Proofs

$$\text{G}\ \frac{p(\bar{x})}{[a]p(\bar{x})} \qquad \text{implies} \qquad \frac{x^2 \geq 0}{[x:=x+1;(x'=x \cup x'=-2)]x^2 \geq 0}$$

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

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

**Locally sound**

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

# Uniform Substitution of Rules and Proofs

$$\text{G}\ \frac{p(\bar{x})}{[a]p(\bar{x})} \qquad \text{implies} \qquad \frac{x^2 \geq 0}{[x := x+1; (x' = x \cup x' = -2)]x^2 \geq 0}$$

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

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

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

**Locally sound**

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

# Uniform Substitution of Rules and Proofs

G $\dfrac{p(\bar{x})}{[a]p(\bar{x})}$ implies $\dfrac{x^2 \geq 0}{[x := x+1; (x' = x \cup x' = -2)]x^2 \geq 0}$

CQ $\dfrac{f() = g()}{p(f()) \leftrightarrow p(g())}$ implies $\dfrac{2x - x = x}{[x' = v]2x - x \geq 0 \leftrightarrow [x' = v]x \geq 0}$

**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*

**Locally sound**

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

# Outline

7 Differential Axioms
- Differential Equation and Differential Axioms
- Differential Substitution Lemmas
- Contextual Congruences
- Static Semantics
- Summary

# Axiom Schema Side Conditions: ODE Solving

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

Axiom schema with side conditions:

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

Quite nontrivial soundness-critical side condition algorithms . . .

# Uniform Substitution

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

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

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

function sym. $f(\theta)$ for any $\theta$ by $\eta(\theta)$

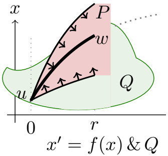
program sym. $a$ by $\alpha$

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

# Differential Invariants for Differential Equations

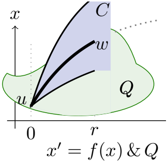
### Differential Invariant

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



### Differential Cut

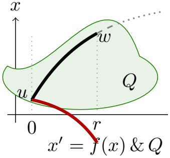
$$\frac{P \vdash [x' = f(x) \,\&\, Q]C \quad P \vdash [x' = f(x) \,\&\, Q \wedge C]P}{P \vdash [x' = f(x) \,\&\, Q]P}$$



### Differential Ghost

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

if new $y' = g(x,y)$ has long enough solution



# Differential Equation Axioms & Differential Axioms

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

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

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

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

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 $[x' = c \,\&\, q(x)]p(x) \leftrightarrow \forall t \geq 0 \, \big((\forall 0 \leq s \leq t \, q(x + cs)) \rightarrow [x := x + ct]p(x)\big)$

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

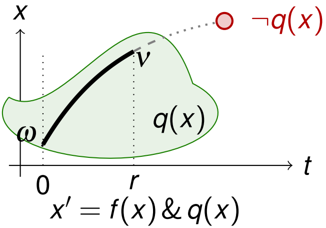
$\cdot'$ $(f(\bar{x}) \cdot g(\bar{x}))' = (f(\bar{x}))' \cdot g(\bar{x}) + f(\bar{x}) \cdot (g(\bar{x}))'$

$c'$ $(c)' = 0$

# Differential Equation Axioms

## Axiom (Differential Weakening) (JAR'17)

$$\text{DW}\quad [x' = f(x) \,\&\, q(x)]\big(q(x) \to p(x)\big) \leftrightarrow [x' = f(x) \,\&\, q(x)]p(x)$$



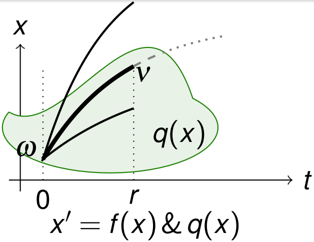
Differential equations cannot leave their evolution domains. Derives from:

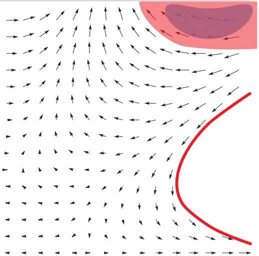
$$\text{DW}\quad [x' = f(x) \,\&\, q(x)]q(x)$$

# Differential Equation Axioms

## Axiom (Differential Cut) (JAR'17)

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.

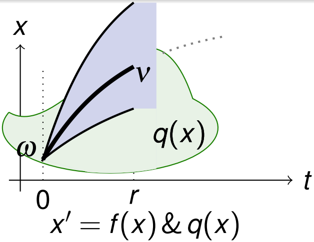
DC is a differential modal modus ponens K.

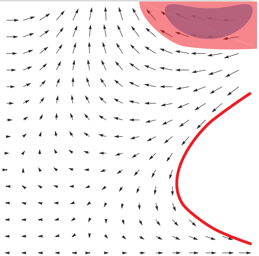
Can't leave $r(x)$, then might as well restrict state space to $r(x)$.

# Differential Equation Axioms

## Axiom (Differential Cut) (JAR'17)

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.

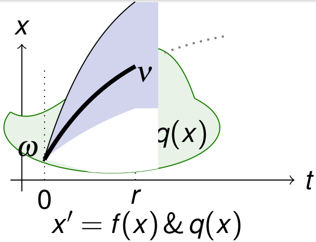
DC is a differential modal modus ponens K.

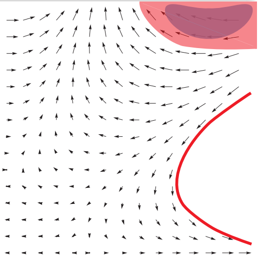
Can't leave $r(x)$, then might as well restrict state space to $r(x)$.

# Differential Equation Axioms

**Axiom (Differential Cut)** (JAR’17)

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.

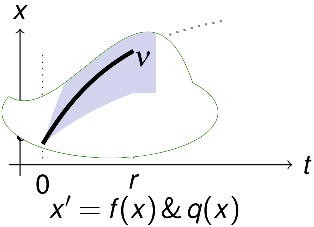
DC is a differential modal modus ponens K.

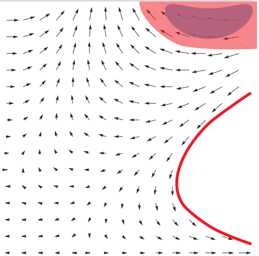
Can’t leave $r(x)$, then might as well restrict state space to $r(x)$.

# Differential Equation Axioms

## Axiom (Differential Cut) (JAR'17)

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.

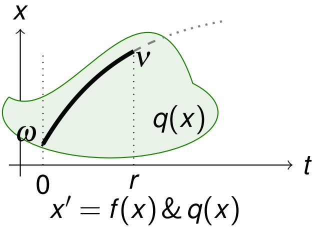
DC is a differential modal modus ponens K.

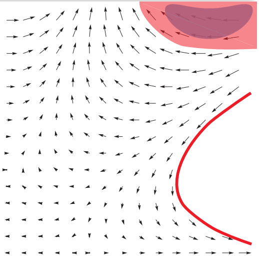
Can't leave $r(x)$, then might as well restrict state space to $r(x)$.

# Differential Equation Axioms

**Axiom (Differential Cut)** (JAR'17)

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.

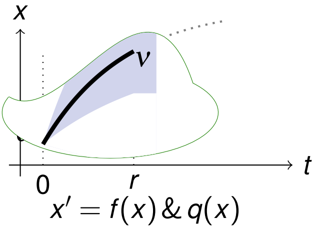
DC is a differential modal modus ponens K.

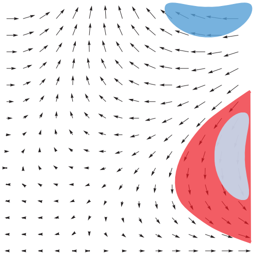
Can't leave $r(x)$, then might as well restrict state space to $r(x)$.

# Differential Equation Axioms

## Axiom (Differential Cut) (JAR'17)

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





DC is a cut for differential equations.

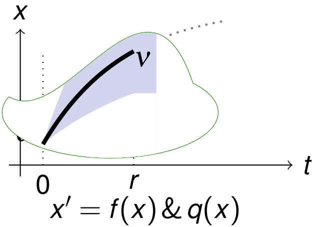
DC is a differential modal modus ponens K.

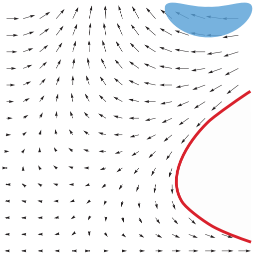
Can't leave $r(x)$, then might as well restrict state space to $r(x)$.

# Differential Equation Axioms

**Axiom (Differential Cut)** (JAR'17)

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.

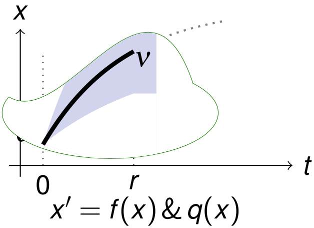
DC is a differential modal modus ponens K.

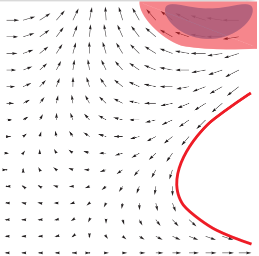
Can't leave $r(x)$, then might as well restrict state space to $r(x)$.

# Differential Equation Axioms

### Axiom (Differential Cut) (JAR'17)

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.

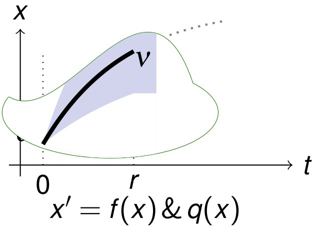
DC is a differential modal modus ponens K.

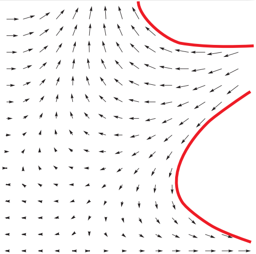
Can't leave $r(x)$, then might as well restrict state space to $r(x)$.

# Differential Equation Axioms

**Axiom (Differential Cut)** (JAR'17)

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.

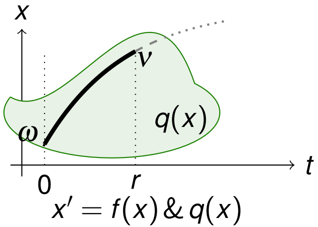
DC is a differential modal modus ponens K.

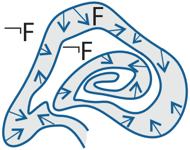
Can't leave $r(x)$, then might as well restrict state space to $r(x)$.

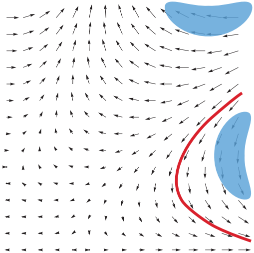
# Differential Equation Axioms

**Axiom (Differential Invariant)** (JAR'17)

DI $([x' = f(x) \,\&\, q(x)]p(x) \leftrightarrow [?q(x)]p(x)) \leftarrow [x' = f(x) \,\&\, q(x)](p(x))'$







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

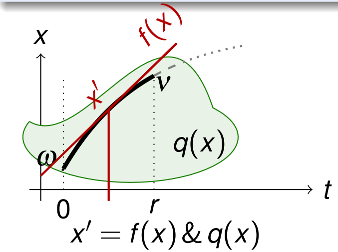
What's the differential of a formula???

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

# Differential Equation Axioms

## Axiom (Differential Effect) (JAR'17)

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



Effect of differential equation on differential symbol $x'$

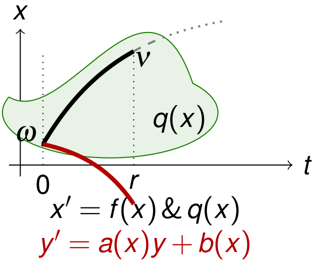
$[x' := f(x)]$ instantly mimics continuous effect $[x' = f(x)]$ on $x'$

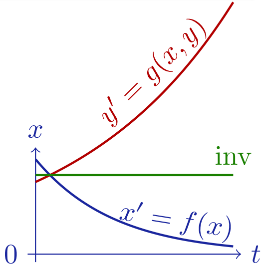
$[x' := f(x)]$ selects vector field $x' = f(x)$ for subsequent differentials

# Differential Equation Axioms

## Axiom (Differential Ghost) (JAR'17)

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

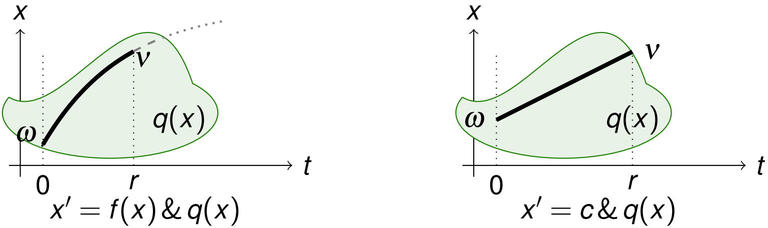
Can cause new invariants

“Dark matter” counterweight to balance conserved quantities

# Differential Equation Axioms

## Axiom (Differential Solution) (JAR'17)

DS $[x' = c \,\&\, q(x)]p(x) \leftrightarrow \forall t{\geq}0 \left((\forall 0{\leq}s{\leq}t\, q(x{+}cs)) \rightarrow [x := x{+}ct]p(x)\right)$



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

# 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 assignment uses vector field
6. $\cdot'$ differential computations are axiomatic (US)

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

# Differential Substitution Lemmas

## Lemma (Differential lemma)

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

Syntactic

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

Analytic

## Lemma (Differential assignment)

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

## Lemma (Derivations)

$$\begin{aligned} (f(\bar{x}) + g(\bar{x}))' &= (f(\bar{x}))' + (g(\bar{x}))' \\ (f(\bar{x}) \cdot g(\bar{x}))' &= (f(\bar{x}))' \cdot g(\bar{x}) + f(\bar{x}) \cdot (g(\bar{x}))' \\ (c)' &= 0 \qquad \textit{for arity 0 functions } c \end{aligned}$$

# Differential Substitution Lemmas

### Lemma (Differential lemma)

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

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

### Lemma (Differential assignment)

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

### Lemma (Derivations)

$$(\theta + \eta)' = (\theta)' + (\eta)'$$

$$(\theta \cdot \eta)' = (\theta)' \cdot \eta + \theta \cdot (\eta)'$$

$$(c)' = 0$$ *for arity 0 functions c*

# Differential Equation Axioms & Differential Axioms

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

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

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

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 $[x' = c \,\&\, q(x)]p(x) \leftrightarrow \forall t {\geq} 0 \,\big((\forall 0 {\leq} s {\leq} t\, q(x + cs)) \to [x := x + ct]p(x)\big)$

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

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

$c'$ $(c)' = 0$

# 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 assignment uses vector field

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

# Example: Contextual Congruence Reasoning by US

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

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

$$\text{CE}\ \frac{P \leftrightarrow Q}{C(P) \leftrightarrow C(Q)}$$

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

# Example: Contextual Congruence Reasoning by US

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

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

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

$$\text{CE}\ \frac{P \leftrightarrow Q}{C(P) \leftrightarrow C(Q)}$$

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

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

# Example: Differential Invariants Parametric

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

# Example: Differential Invariants Parametric

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

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

# Example: Differential Invariants

## Parametric

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

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

# Example: Differential Invariants

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

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

# 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

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

# 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

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

# 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
4. USR instantiates proof by $\{j(x,x') \mapsto x'\cdot x + x\cdot x'\}$

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

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

# 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 $\{j(x,x') \mapsto x' \cdot x + x \cdot x'\}$

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

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

# 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
4. USR instantiates proof by $\{j(x, x') \mapsto x' \cdot x + x \cdot x'\}$

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

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

# 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 $\{j(x,x') \mapsto x'\cdot x + x\cdot x'\}$

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

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

# 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
4. USR instantiates proof by $\{j(x,x') \mapsto x' \cdot x + x \cdot x'\}$

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

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

# Example: Differential Invariants

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

# Example: Differential Invariants

1. Start with identity differential computation result

$$
\begin{array}{ll}
\mathbb{R} & \hline (x\cdot x)' = (x\cdot x)' \\
.' & \hline \\
x' & \hline \\
\text{CT} & \hline \\
 & \hline \\
\text{CE} & \hline \vdash [x' = x^3][x' := x^3]{\color{red}(x\cdot x \geq 1)'} \\
\text{DE} & \hline \vdash [x' = x^3](x\cdot x \geq 1)' \\
\text{DI} & \hline x\cdot x \geq 1 \vdash [x' = x^3]x\cdot x \geq 1
\end{array}
$$

# Example: Differential Invariants

Computation

1. Start with identity differential computation result which proves

$$\begin{array}{ll}
 & * \\ \hline
\mathbb{R} & (x\cdot x)' = (x\cdot x)' \\ \hline
.' & \\ \hline
x' & \\ \hline
\mathrm{CT} & \\ \hline
 & \\ \hline
\mathrm{CE} & \vdash [x' = x^3][x' := x^3](x\cdot x \geq 1)' \\ \hline
\mathrm{DE} & \vdash [x' = x^3](x\cdot x \geq 1)' \\ \hline
\mathrm{DI} & x\cdot x \geq 1 \vdash [x' = x^3]x\cdot x \geq 1
\end{array}$$

# Example: Differential Invariants Computation

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

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

# Example: Differential Invariants Computation

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

$$\begin{array}{rc}
 & * \\ \hline
\mathbb{R} & (x\cdot x)' = (x\cdot x)' \\ \hline
\cdot' & (x\cdot x)' = (x)'\cdot x + x\cdot(x)' \\ \hline
x' & (x\cdot x)' = x'\cdot x + x\cdot x' \\ \hline
\text{CT} & \\ \hline
 &
\end{array}$$

$$\begin{array}{rr}
\hline
\text{CE} & \vdash [x' = x^3][x' := x^3](x\cdot x \geq 1)' \\ \hline
\text{DE} & \vdash [x' = x^3](x\cdot x \geq 1)' \\ \hline
\text{DI} & x\cdot x \geq 1 \vdash [x' = x^3]x\cdot x \geq 1
\end{array}$$

# Example: Differential Invariants Computation

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

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

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

# Example: Differential Invariants

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

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

# Example: Differential Invariants

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

$$
{\scriptstyle \mathrm{DI}}\dfrac{{\scriptstyle \mathrm{DE}}\dfrac{{\scriptstyle \mathrm{CE}}\dfrac{{\scriptstyle \mathrm{G}}\dfrac{}{\vdash [x'=x^3][x':=x^3]{\color{green}x'\cdot x+x\cdot x'\geq 0}} \qquad \begin{array}{c}{\scriptstyle \mathrm{CT}}\dfrac{{\scriptstyle x'}\dfrac{{\scriptstyle \cdot'}\dfrac{{\scriptstyle \mathbb{R}}\dfrac{*}{(x\cdot x)'=(x\cdot x)'}}{(x\cdot x)'=(x)'\cdot x+x\cdot(x)'}}{(x\cdot x)'=x'\cdot x+x\cdot x'}}{(x\cdot x)'\geq 0\leftrightarrow x'\cdot x+x\cdot x'\geq 0}\\ {\color{red}(x\cdot x\geq 1)'}\leftrightarrow{\color{green}x'\cdot x+x\cdot x'\geq 0}\end{array}}{\vdash [x'=x^3][x':=x^3]{\color{red}(x\cdot x\geq 1)'}}}{\vdash [x'=x^3](x\cdot x\geq 1)'}}{x\cdot x\geq 1\vdash [x'=x^3]x\cdot x\geq 1}
$$

# Example: Differential Invariants — Computation

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

$$
\text{DI}\dfrac{\text{DE}\dfrac{\text{CE}\dfrac{\text{G}\dfrac{\text{[:=]}\dfrac{}{\vdash [x' := x^3]x'\cdot x + x\cdot x' \geq 0}}{\vdash [x' = x^3][x' := x^3]x'\cdot x + x\cdot x' \geq 0} \qquad \text{CT}\dfrac{x'\dfrac{\cdot'\dfrac{\mathbb{R}\dfrac{*}{(x\cdot x)' = (x\cdot x)'}}{(x\cdot x)' = (x)'\cdot x + x\cdot (x)'}}{(x\cdot x)' = x'\cdot x + x\cdot x'}}{\begin{array}{c}(x\cdot x)' \geq 0 \leftrightarrow x'\cdot x + x\cdot x' \geq 0\\ (x\cdot x \geq 1)' \leftrightarrow x'\cdot x + x\cdot x' \geq 0\end{array}}}{\vdash [x' = x^3][x' := x^3](x\cdot x \geq 1)'}}{\vdash [x' = x^3](x\cdot x \geq 1)'}}{x\cdot x \geq 1 \vdash [x' = x^3]x\cdot x \geq 1}
$$

# Example: Differential Invariants

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

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

# Example: Differential Invariants

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

$$\text{DI}\;\dfrac{\text{DE}\;\dfrac{\text{CE}\;\dfrac{\text{G}\;\dfrac{[:=]\;\dfrac{\mathbb{R}\;\dfrac{*}{\vdash x^3\cdot x+x\cdot x^3\geq 0}}{\vdash [x':=x^3]x'\cdot x+x\cdot x'\geq 0}}{\vdash [x'=x^3][x':=x^3]x'\cdot x+x\cdot x'\geq 0}\qquad \text{CT}\;\dfrac{x'\;\dfrac{\cdot'\;\dfrac{\mathbb{R}\;\dfrac{*}{(x\cdot x)'=(x\cdot x)'}}{(x\cdot x)'=(x)'\cdot x+x\cdot(x)'}}{(x\cdot x)'=x'\cdot x+x\cdot x'}}{\begin{array}{c}(x\cdot x)'\geq 0\leftrightarrow x'\cdot x+x\cdot x'\geq 0\\ (x\cdot x\geq 1)'\leftrightarrow x'\cdot x+x\cdot x'\geq 0\end{array}}}{\vdash [x'=x^3][x':=x^3](x\cdot x\geq 1)'}}{\vdash [x'=x^3](x\cdot x\geq 1)'}}{x\cdot x\geq 1\vdash [x'=x^3]x\cdot x\geq 1}$$

# Uniform Substitution

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

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

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

i.e. bound variables $U = \mathrm{BV}(\otimes(\cdot))$ of **no** operator $\otimes$
are free in the substitution on its argument $\theta$ ($U$-admissible)

If you bind a free variable, you go to logic jail!

Uniform substitution $\sigma$ replaces all occurrences of $p(\theta)$ for any $\theta$ by $\psi(\theta)$
function sym. $f(\theta)$ for any $\theta$ by $\eta(\theta)$
program sym. $a$ by $\alpha$

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

# Uniform Substitution

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

Modular interface:
Prover vs. Logic

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

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

i.e. bound variables $U = \mathrm{BV}(\otimes(\cdot))$ of **no** operator $\otimes$
are free in the substitution on its argument $\theta$ ($U$-admissible)

If you bind a free variable, you go to logic jail!

Uniform substitution $\sigma$ replaces all occurrences of $p(\theta)$ for any $\theta$ by $\psi(\theta)$
function sym. $f(\theta)$ for any $\theta$ by $\eta(\theta)$
program sym. $a$ by $\alpha$

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

# Correctness of Static Semantics

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

*If* $(\omega,\nu) \in [\![\alpha]\!]$*, then* $\omega = \nu$ *on* $BV(\alpha)^\complement$.

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

*If* $\omega = \tilde{\omega}$ *on* $FV(\theta)$ *and* $I = J$ *on* $\Sigma(\theta)$*, then* $\omega[\![\theta]\!] = \tilde{\omega}[\![\theta]\!]$

*If* $\omega = \tilde{\omega}$ *on* $FV(\phi)$ $\omega \in [\![\phi]\!]$ *iff* $\tilde{\omega} \in J[\![\phi]\!]$

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

# Correctness of Static Semantics

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

*If* $(\omega, \nu) \in [\![\alpha]\!]$, *then* $\omega = \nu$ *on* $BV(\alpha)^\complement$.

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

*If* $\omega = \tilde\omega$ *on* $FV(\theta)$ *and* $I = J$ *on* $\Sigma(\theta)$, *then* $\omega[\![\theta]\!] = \tilde\omega[\![\theta]\!]$

*If* $\omega = \tilde\omega$ *on* $FV(\phi)$ $\omega \in [\![\phi]\!]$ *iff* $\tilde\omega \in J[\![\phi]\!]$

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

# Differential Dynamic Logic dL: Static Semantics

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

# Differential Dynamic Logic dL: Static Semantics

$$
\begin{array}{rcll}
\mathsf{FV}((\theta)') &=& \mathsf{FV}(\theta) & \\
\hline
\mathsf{FV}(p(\theta_1,\dots,\theta_k)) &=& \mathsf{FV}(\theta_1) \cup \dots \cup \mathsf{FV}(\theta_k) & \\
\mathsf{FV}(\phi \wedge \psi) &=& \mathsf{FV}(\phi) \cup \mathsf{FV}(\psi) & \\
\mathsf{FV}(\forall x\, \phi) = \mathsf{FV}(\exists x\, \phi) &=& \mathsf{FV}(\phi) \setminus \{x\} & \\
\mathsf{FV}([\alpha]\phi) = \mathsf{FV}(\langle\alpha\rangle\phi) &=& \mathsf{FV}(\alpha) \cup (\mathsf{FV}(\phi) \setminus \mathsf{BV}(\alpha)) & \\
\hline
\mathsf{FV}(a) &=& \mathscr{V} & \text{for program symbol } a \\
\mathsf{FV}(x := \theta) &=& \mathsf{FV}(\theta) & \\
\mathsf{FV}(?Q) &=& \mathsf{FV}(Q) & \\
\mathsf{FV}(x' = \theta \,\&\, Q) &=& \{x\} \cup \mathsf{FV}(\theta) \cup \mathsf{FV}(Q) & \\
\mathsf{FV}(\alpha \cup \beta) &=& \mathsf{FV}(\alpha) \cup \mathsf{FV}(\beta) & \\
\mathsf{FV}(\alpha ; \beta) &=& \mathsf{FV}(\alpha) \cup (\mathsf{FV}(\beta) \setminus \mathsf{BV}(\alpha)) & \\
\mathsf{FV}(\alpha^*) &=& \mathsf{FV}(\alpha) &
\end{array}
$$

# Differential Dynamic Logic dL: Static Semantics

$$\mathsf{FV}((\theta)') = \mathsf{FV}(\theta) \cup \mathsf{FV}(\theta)'$$ caution

---

$$\mathsf{FV}(p(\theta_1,\ldots,\theta_k)) = \mathsf{FV}(\theta_1) \cup \cdots \cup \mathsf{FV}(\theta_k)$$

$$\mathsf{FV}(\phi \land \psi) = \mathsf{FV}(\phi) \cup \mathsf{FV}(\psi)$$

$$\mathsf{FV}(\forall x\, \phi) = \mathsf{FV}(\exists x\, \phi) = \mathsf{FV}(\phi) \setminus \{x\}$$

$$\mathsf{FV}([\alpha]\phi) = \mathsf{FV}(\langle\alpha\rangle\phi) = \mathsf{FV}(\alpha) \cup (\mathsf{FV}(\phi) \setminus \mathsf{MBV}(\alpha))$$ caution

---

$$\mathsf{FV}(a) = \mathscr{V}$$ for program symbol $a$

$$\mathsf{FV}(x := \theta) = \mathsf{FV}(\theta)$$

$$\mathsf{FV}(?Q) = \mathsf{FV}(Q)$$

$$\mathsf{FV}(x' = \theta \,\&\, Q) = \{x\} \cup \mathsf{FV}(\theta) \cup \mathsf{FV}(Q)$$

$$\mathsf{FV}(\alpha \cup \beta) = \mathsf{FV}(\alpha) \cup \mathsf{FV}(\beta)$$

$$\mathsf{FV}(\alpha;\beta) = \mathsf{FV}(\alpha) \cup (\mathsf{FV}(\beta) \setminus \mathsf{MBV}(\alpha))$$ caution

$$\mathsf{FV}(\alpha^*) = \mathsf{FV}(\alpha)$$

# Differential Dynamic Logic dL: Static Semantics

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

# Differential Dynamic Logic dL: Static Semantics

$$
\begin{aligned}
\mathrm{BV}(\theta \geq \eta) = \mathrm{BV}(p(\theta_1,\dots,\theta_k)) &= \emptyset \\
\mathrm{BV}(\phi \land \psi) &= \mathrm{BV}(\phi) \cup \mathrm{BV}(\psi) \\
\mathrm{BV}(\forall x\, \phi) = \mathrm{BV}(\exists x\, \phi) &= \{x\} \cup \mathrm{BV}(\phi) \\
\mathrm{BV}([\alpha]\phi) = \mathrm{BV}(\langle\alpha\rangle\phi) &= \mathrm{BV}(\alpha) \cup \mathrm{BV}(\phi)
\end{aligned}
$$

---

$$
\begin{aligned}
\mathrm{BV}(a) &= \mathscr{V} && \text{for program symbol } a \\
\mathrm{BV}(x := \theta) &= \{x\} \\
\mathrm{BV}(?Q) &= \emptyset \\
\mathrm{BV}(x' = \theta \,\&\, Q) &= \{x, x'\} \\
\mathrm{BV}(\alpha \cup \beta) = \mathrm{BV}(\alpha;\beta) &= \mathrm{BV}(\alpha) \cup \mathrm{BV}(\beta) \\
\mathrm{BV}(\alpha^*) &= \mathrm{BV}(\alpha)
\end{aligned}
$$

---

# Differential Dynamic Logic dL: Static Semantics

$$
\begin{aligned}
\mathsf{BV}(\theta \geq \eta) = \mathsf{BV}(p(\theta_1, \dots, \theta_k)) &= \emptyset \\
\mathsf{BV}(\phi \land \psi) &= \mathsf{BV}(\phi) \cup \mathsf{BV}(\psi) \\
\mathsf{BV}(\forall x\, \phi) = \mathsf{BV}(\exists x\, \phi) &= \{x\} \cup \mathsf{BV}(\phi) \\
\mathsf{BV}([\alpha]\phi) = \mathsf{BV}(\langle\alpha\rangle\phi) &= \mathsf{BV}(\alpha) \cup \mathsf{BV}(\phi)
\end{aligned}
$$

---

$$
\begin{aligned}
\mathsf{BV}(a) &= \mathscr{V} && \text{for program symbol } a \\
\mathsf{BV}(x := \theta) &= \{x\} \\
\mathsf{BV}(?Q) &= \emptyset \\
\mathsf{BV}(x' = \theta \,\&\, Q) &= \{x, x'\} \\
\mathsf{BV}(\alpha \cup \beta) = \mathsf{BV}(\alpha;\beta) &= \mathsf{BV}(\alpha) \cup \mathsf{BV}(\beta) \\
\mathsf{BV}(\alpha^*) &= \mathsf{BV}(\alpha)
\end{aligned}
$$

---

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

# Differential Dynamic Logic dL: Static Semantics

$$\begin{aligned}
\mathsf{BV}(\theta \geq \eta) = \mathsf{BV}(p(\theta_1,\dots,\theta_k)) &= \emptyset\\
\mathsf{BV}(\phi \land \psi) &= \mathsf{BV}(\phi) \cup \mathsf{BV}(\psi)\\
\mathsf{BV}(\forall x\, \phi) = \mathsf{BV}(\exists x\, \phi) &= \{x\} \cup \mathsf{BV}(\phi)\\
\mathsf{BV}([\alpha]\phi) = \mathsf{BV}(\langle\alpha\rangle\phi) &= \mathsf{BV}(\alpha) \cup \mathsf{BV}(\phi)
\end{aligned}$$

---

$$\begin{aligned}
\mathsf{BV}(a) &= \mathscr{V} && \text{for program symbol } a\\
\mathsf{BV}(x := \theta) &= \{x\}\\
\mathsf{BV}(?Q) &= \emptyset\\
\mathsf{BV}(x' = \theta \,\&\, Q) &= \{x, x'\}\\
\mathsf{BV}(\alpha \cup \beta) = \mathsf{BV}(\alpha;\beta) &= \mathsf{BV}(\alpha) \cup \mathsf{BV}(\beta)\\
\mathsf{BV}(\alpha^*) &= \mathsf{BV}(\alpha)
\end{aligned}$$

---

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

# Correctness of Static Semantics

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

*If* $(\omega,\nu) \in [\![\alpha]\!]$, *then* $\omega = \nu$ *on* $BV(\alpha)^\complement$.

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

*If* $\omega = \tilde{\omega}$ *on* $FV(\theta)$ *and* $I = J$ *on* $\Sigma(\theta)$, *then* $\omega[\![\theta]\!] = \tilde{\omega}[\![\theta]\!]$

*If* $\omega = \tilde{\omega}$ *on* $FV(\phi)$ $\omega \in [\![\phi]\!]$ *iff* $\tilde{\omega} \in J[\![\phi]\!]$

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

# Uniform Substitution

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

Modular interface:
Prover vs. Logic

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

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

i.e. bound variables $U = \text{BV}(\otimes(\cdot))$ of **no** operator $\otimes$
are free in the substitution on its argument $\theta$ ($U$-admissible)

If you bind a free variable, you go to logic jail!

Uniform substitution $\sigma$ replaces all occurrences of $p(\theta)$ for any $\theta$ by $\psi(\theta)$
function sym. $f(\theta)$ for any $\theta$ by $\eta(\theta)$
program sym. $a$ by $\alpha$

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

# Differential Equation Axioms & Differential Axioms

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

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

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

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 $[x' = c \,\&\, q(x)]p(x) \leftrightarrow \forall t \geq 0 \,\big((\forall 0 \leq s \leq t\, q(x + cs)) \rightarrow [x := x + ct]p(x)\big)$

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

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

$c'$ $(c)' = 0$

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

André Platzer.
A complete uniform substitution calculus for differential dynamic logic.
*J. Autom. Reas.*, 59(2):219–265, 2017.
`doi:10.1007/s10817-016-9385-1`.

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, Berlin, 2015. Springer.
`doi:10.1007/978-3-319-21401-6_32`.

André Platzer.
The complete proof theory of hybrid systems.
In LICS [7], pages 541–550.
`doi:10.1109/LICS.2012.64`.

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

André Platzer.
Logics of dynamical systems.
In LICS [7], pages 13–24.
`doi:10.1109/LICS.2012.13`.

*Logic in Computer Science (LICS), 2012 27th Annual IEEE Symposium on*, Los Alamitos, 2012. IEEE.

# Differential Dynamic Logic with Interpretations: Semantics

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

$$\begin{aligned}
\omega\llbracket f(\theta_1,\dots,\theta_k)\rrbracket &= I(f)\big(\omega\llbracket\theta_1\rrbracket,\dots,\omega\llbracket\theta_k\rrbracket\big) \qquad I(f) : \mathbb{R}^k \to \mathbb{R} \text{ smooth}\\
\omega\llbracket(\theta)'\rrbracket &= \sum_x \omega(x')\frac{\partial\llbracket\theta\rrbracket}{\partial x}(\omega)
\end{aligned}$$

**Definition (dL semantics)** $(\llbracket\cdot\rrbracket : \mathrm{Fml} \to \wp(\mathscr{S}))$

$$\begin{aligned}
\llbracket p(\theta_1,\dots,\theta_k)\rrbracket &= \{\omega : (\omega\llbracket\theta_1\rrbracket,\dots,\omega\llbracket\theta_k\rrbracket) \in I(p)\} \qquad I(p) \subseteq \mathbb{R}^k\\
\llbracket\langle\alpha\rangle\phi\rrbracket &= \llbracket\alpha\rrbracket \circ \llbracket\phi\rrbracket
\end{aligned}$$

$P$ valid iff $\omega \in \llbracket P\rrbracket$ for all states $\omega$ of all interpretations $I$

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

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

# Differential Dynamic Logic with Interpretations: Semantics

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

$$\begin{aligned}
\omega\llbracket x\rrbracket &= \omega(x) && \text{for variable } x \in \mathscr{V}\\
\omega\llbracket \theta + \eta\rrbracket &= \omega\llbracket\theta\rrbracket + \omega\llbracket\eta\rrbracket\\
\omega\llbracket \theta \cdot \eta\rrbracket &= \omega\llbracket\theta\rrbracket \cdot \omega\llbracket\eta\rrbracket\\
\omega\llbracket f(\theta_1,\dots,\theta_k)\rrbracket &= I(f)\big(\omega\llbracket\theta_1\rrbracket,\dots,\omega\llbracket\theta_k\rrbracket\big) && I(f) : \mathbb{R}^k \to \mathbb{R} \text{ smooth}\\
\omega\llbracket(\theta)'\rrbracket &= \sum_x \omega(x')\frac{\partial\llbracket\theta\rrbracket}{\partial x}(\omega)
\end{aligned}$$

### Definition (dL semantics) $(\llbracket\cdot\rrbracket : \mathrm{Fml} \to \wp(\mathscr{S}))$

$$\begin{aligned}
\llbracket p(\theta_1,\dots,\theta_k)\rrbracket &= \{\omega \,:\, (\omega\llbracket\theta_1\rrbracket,\dots,\omega\llbracket\theta_k\rrbracket) \in I(p)\} \quad I(p) \subseteq \mathbb{R}^k\\
\llbracket\langle\alpha\rangle\phi\rrbracket &= \llbracket\alpha\rrbracket \circ \llbracket\phi\rrbracket\\
\llbracket[\alpha]\phi\rrbracket &= \llbracket\neg\langle\alpha\rangle\neg\phi\rrbracket
\end{aligned}$$

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

$$\begin{aligned}
\llbracket a\rrbracket &= I(a) && I(a) \subseteq \mathscr{S} \times \mathscr{S}\\
\llbracket x' = f(x) \,\&\, Q\rrbracket &= \{(\varphi(0)|_{\{x'\}^\complement}, \varphi(r)) \,:\, \varphi \models x' = f(x) \land Q\}\\
\llbracket\alpha \cup \beta\rrbracket &= \llbracket\alpha\rrbracket \cup \llbracket\beta\rrbracket\\
\llbracket\alpha;\beta\rrbracket &= \llbracket\alpha\rrbracket \circ \llbracket\beta\rrbracket
\end{aligned}$$

# Differential Dynamic Logic with Interpretations: Semantics

**Definition (Term semantics)** $(\llbracket\cdot\rrbracket : \text{Trm} \to (\mathscr{S} \to \mathbb{R}))$

$$
\begin{aligned}
\omega\llbracket f(\theta_1,\dots,\theta_k)\rrbracket &= I(f)\big(\omega\llbracket\theta_1\rrbracket,\dots,\omega\llbracket\theta_k\rrbracket\big) \qquad I(f) : \mathbb{R}^k \to \mathbb{R} \text{ smooth}\\
\omega\llbracket(\theta)'\rrbracket &= \sum_x \omega(x')\frac{\partial\llbracket\theta\rrbracket}{\partial x}(\omega)
\end{aligned}
$$

**Definition (dL semantics)** $(\llbracket\cdot\rrbracket : \text{Fml} \to \wp(\mathscr{S}))$

$$
\begin{aligned}
\llbracket\theta \geq \eta\rrbracket &= \{\omega : \omega\llbracket\theta\rrbracket \geq \omega\llbracket\eta\rrbracket\}\\
\llbracket p(\theta_1,\dots,\theta_k)\rrbracket &= \{\omega : (\omega\llbracket\theta_1\rrbracket,\dots,\omega\llbracket\theta_k\rrbracket) \in I(p)\} \qquad I(p) \subseteq \mathbb{R}^k\\
\llbracket\neg\phi\rrbracket &= (\llbracket\phi\rrbracket)^{\complement}\\
\llbracket\phi \wedge \psi\rrbracket &= \llbracket\phi\rrbracket \cap \llbracket\psi\rrbracket\\
\llbracket\exists x\,\phi\rrbracket &= \{\omega \in \mathscr{S} : \omega_x^r \in \llbracket\phi\rrbracket \text{ for some } r \in \mathbb{R}\}\\
\llbracket\langle\alpha\rangle\phi\rrbracket &= \llbracket\alpha\rrbracket \circ \llbracket\phi\rrbracket = \{\omega : \nu \in \llbracket\phi\rrbracket \text{ for some } \nu\ (\omega,\nu) \in \llbracket\alpha\rrbracket\}\\
\llbracket[\alpha]\phi\rrbracket &= \llbracket\neg\langle\alpha\rangle\neg\phi\rrbracket = \{\omega : \nu \in \llbracket\phi\rrbracket \text{ for all } \nu\ (\omega,\nu) \in \llbracket\alpha\rrbracket\}
\end{aligned}
$$

**Definition (Program semantics)** $(\llbracket\cdot\rrbracket : \text{HP} \to \wp(\mathscr{S} \times \mathscr{S}))$

$$
\begin{aligned}
\llbracket a\rrbracket &= I(a) \qquad I(a) \subseteq \mathscr{S} \times \mathscr{S}\\
\llbracket x' = f(x)\,\&\,Q\rrbracket &= \{(\varphi(0)|_{\{x'\}^{\complement}}, \varphi(r)) : \varphi \models x' = f(x) \wedge Q\}\\
\llbracket\alpha \cup \beta\rrbracket &= \llbracket\alpha\rrbracket \cup \llbracket\beta\rrbracket
\end{aligned}
$$

# Differential Dynamic Logic with Interpretations: Semantics

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

$$\begin{aligned} \omega\llbracket f(\theta_1,\dots,\theta_k)\rrbracket &= I(f)\big(\omega\llbracket\theta_1\rrbracket,\dots,\omega\llbracket\theta_k\rrbracket\big) \qquad I(f):\mathbb{R}^k \to \mathbb{R} \text{ smooth} \\ \omega\llbracket(\theta)'\rrbracket &= \sum_x \omega(x')\frac{\partial\llbracket\theta\rrbracket}{\partial x}(\omega) \end{aligned}$$

## Definition (dL semantics) $(\llbracket\cdot\rrbracket : \mathrm{Fml} \to \wp(\mathscr{S}))$

$$\begin{aligned} \llbracket p(\theta_1,\dots,\theta_k)\rrbracket &= \{\omega : (\omega\llbracket\theta_1\rrbracket,\dots,\omega\llbracket\theta_k\rrbracket) \in I(p)\} \qquad I(p) \subseteq \mathbb{R}^k \\ \llbracket\langle\alpha\rangle\phi\rrbracket &= \llbracket\alpha\rrbracket \circ \llbracket\phi\rrbracket \\ \llbracket[\alpha]\phi\rrbracket &= \llbracket\neg\langle\alpha\rangle\neg\phi\rrbracket \end{aligned}$$

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

$$\begin{aligned} \llbracket a\rrbracket &= I(a) \qquad I(a) \subseteq \mathscr{S} \times \mathscr{S} \\ \llbracket x := \theta\rrbracket &= \{(\omega,\nu) : \nu = \omega \text{ except } \nu\llbracket x\rrbracket = \omega\llbracket\theta\rrbracket\} \\ \llbracket ?Q\rrbracket &= \{(\omega,\omega) : \omega \in \llbracket Q\rrbracket\} \\ \llbracket x' = f(x)\,\&\,Q\rrbracket &= \{(\varphi(0)|_{\{x'\}^\complement},\varphi(r)) : \varphi \models x' = f(x) \land Q\} \\ \llbracket\alpha\cup\beta\rrbracket &= \llbracket\alpha\rrbracket \cup \llbracket\beta\rrbracket \\ \llbracket\alpha;\beta\rrbracket &= \llbracket\alpha\rrbracket \circ \llbracket\beta\rrbracket \\ \llbracket\alpha^*\rrbracket &= \big(\llbracket\alpha\rrbracket\big)^* = \bigcup_{n\in\mathbb{N}}\llbracket\alpha^n\rrbracket \end{aligned}$$