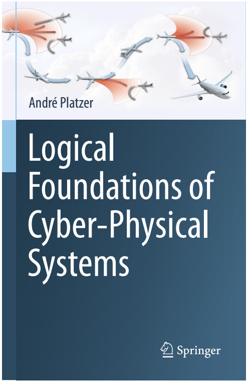
18: Axioms & Uniform Substitutions

Logical Foundations of Cyber-Physical Systems



André Platzer

Karlsruhe Institute of Technology
Department of Informatics

Computer Science Department
Carnegie Mellon University

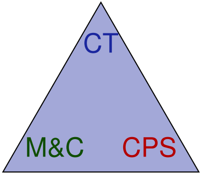
Outline

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

Part I

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

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

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

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

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

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

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

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

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

Part I

$$[:=]\ [x := \theta]\phi(x) \leftrightarrow \phi(\theta) \qquad (\theta \text{ free for } x \text{ in } \phi)$$

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

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

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

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

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

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

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

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

Axiom Schema

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

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

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

Axiom Schema Matches Many Formulas

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

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

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

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

Axiom Schema Matches Many Formulas

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

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

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

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

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

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

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

Match shape $\alpha \cup \beta$

Schema variable α match

Same ϕ everywhere

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

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

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

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 α match

Same ϕ every-where

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

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

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

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

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

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

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

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

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

Match shape $\alpha \cup \beta$

Schema variable α match

Same ϕ 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 $\phi \to [\alpha]\phi$ $\qquad (FV(\phi) \cap BV(\alpha) = \emptyset)$

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

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

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

Match shape $\alpha \cup \beta$

Schema variable α match

Same ϕ 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 $\phi \to [\alpha]\phi$ $\qquad (FV(\phi) \cap BV(\alpha) = \emptyset)$

✓ $y \geq 0 \to [x' = -5]\, y \geq 0$

× $x \geq 0 \to [x' = -5]\, x \geq 0$

✓ $y \geq z \to [x' = -5]\, y \geq z$

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

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

Match shape $\alpha\cup\beta$

Schema variable α match

Same ϕ 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 $\phi \rightarrow [\alpha]\phi$ $\qquad (FV(\phi)\cap BV(\alpha) = \emptyset)$

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

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

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

rule out by side conditions

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

Match all free x occurrences

Replace by θ everywhere

no x occurrence where θ bound

✓ … $y]\,x \leq y^2$ … $\leq y^2$

× … $y][y := 5]$ … $:= 5]\,x + y \geq 0$

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

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

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

Algorithm

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

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

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

Match shape $\alpha\cup\beta$

Schema variable α match

Same ϕ everywhere

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 $\phi)$

✓ … $y]\,x \leq y^2$ … $\leq y^2$

× … $y][y := 5]$ … $:= 5]\,x + y{\geq}0$

✓ … $(x := x+y$ … $]\,x{\geq}y \leftrightarrow [(x := x+2b;\, x'$ … ${\geq}2b$

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

Match all free x occurrences

Replace by θ everywhere

no x occurrence where θ bound

Axiom Schema Side Conditions: ODE Solving

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

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 θ
3. Initial value check: $y(\cdot)$ solves the symbolic IVP $y(0) = x$

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 θ
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

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

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 θ
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 ϕ

Quite nontrivial soundness-critical side condition algorithms …

What Axioms Want

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

What Axioms Want

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

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

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

- 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 $\phi \rightarrow [\alpha]\phi$ 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

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

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

$[\cup]$ $[\alpha \cup \beta]\phi \leftrightarrow [\alpha]\phi \land [\beta]\phi$	$[\cup]$ $[a \cup b]p(\bar{x}) \leftrightarrow [a]p(\bar{x}) \land [b]p(\bar{x})$
V $\phi \to [\alpha]\phi$	V $p \to [a]p$
$[:=]$ $[x := \theta]\phi(x) \leftrightarrow \phi(\theta)$	$[:=]$ $[x := c]p(x) \leftrightarrow p(c)$

- 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

Definition (Hybrid program α)

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

Definition (dL Formula ϕ)

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

Definition (Term θ)

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

Differential Dynamic Logic with Interpretations: Syntax

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

Definition (Hybrid program α)

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

Definition (dL Formula ϕ)

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

Definition (Term θ)

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

All Reals | Some Reals | All Runs | Some Runs

Program Symbol

Definition (Hybrid program α)

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

Definition (dL Formula ϕ)

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

Definition (Term θ)

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

Predicate Symbol

Function Symbol

Differential

Definition (Term semantics) ($[\![\cdot]\!] : \mathrm{Trm} \to (\mathscr{S} \to \mathbb{R})$)

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

Definition (dL semantics) ($[\![\cdot]\!] : \mathrm{Fml} \to \wp(\mathscr{S})$)

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

P valid iff $\omega \in [\![P]\!]$ for all states ω of all interpretations I

Definition (Program semantics) ($[\![\cdot]\!] : \mathrm{HP} \to \wp(\mathscr{S} \times \mathscr{S})$)

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

Soundness Proofs for Axioms

Lemma (V vacuous axiom)

$$V \quad p \rightarrow [a]p$$

Lemma ([:=] assignment axiom)

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

Soundness Proofs for Axioms

Lemma (V vacuous axiom)

$$V \quad p \rightarrow [a]p$$

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 ω, so $\omega \in [\![[a]p]\!]$ especially.
2. I interprets p as *false*: $\omega \notin [\![p]\!]$ for all ω, 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

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

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

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

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

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

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

Uniform Substitution: 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

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

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

Correct

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

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

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

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

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

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

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

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

Correct

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

BV

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

Clash

FV

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

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

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

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

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

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

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

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

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

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 σ replaces all occurrences of $p(\theta)$ for any θ by $\psi(\theta)$
function sym. $f(\theta)$ for any θ by $\eta(\theta)$
program sym. a by α

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

Uniform Substitution

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

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

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

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

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

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

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* ϕ

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

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

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

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

Uniform Substitution: Recursive Application

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

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

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

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

Uniform Substitution: Recursive Application

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

Uniform Substitution: Recursive Application

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

Uniform Substitution: Recursive Application

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

Uniform Substitution: Recursive Application

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

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

Uniform Substitution: Recursive Application

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

Uniform Substitution: Recursive Application

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

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

Uniform Substitution: Recursive Application

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

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

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

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

Uniform Substitution: Examples

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

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

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

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

Uniform Substitution: Examples

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

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

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

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

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

Uniform Substitution: Examples

BV

Clash

FV

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

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

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

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

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

Uniform Substitution: Examples

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

Clash

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

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

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

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

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

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

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

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

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

Uniform Substitution: Examples

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

Clash

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

Correct

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

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

BV

Clash

FV

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

Uniform Substitution: Examples

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

Clash

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

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

Correct

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

Clash

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

Correct

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* ϕ

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

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

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

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

Correctness of Uniform Substitutions

"Syntactic uniform substitution = semantic replacement"

Lemma (Uniform substitution lemma) (JAR'17)

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

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

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

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

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

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

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* ϕ

Proof.

If premise ϕ valid, i.e. $\omega \in I[\![\phi]\!]$ in all I, ω
Then conclusion $\sigma(\phi)$ valid, because $\omega \in I[\![\sigma(\phi)]\!]$ iff $\omega \in \sigma^*_\omega I[\![\phi]\!]$ □

Outline

Part I

Part IV

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

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

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

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

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

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

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

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

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

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

Infinite axiom schema

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

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

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

Schema

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

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

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

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

Schema

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

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

Axiom = one formula

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

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

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

Axiom

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

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

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

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

Axiom

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

Example Proof

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

Example Proof

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

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

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

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

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

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

Example Proof

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

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

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

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

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

Example Proof

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

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

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

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

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

Example Proof

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

v can't have ODE

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

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

Example Proof

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

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

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

Example Proof

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

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

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

Example Proof

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

Summarize:

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

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:

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

Outline

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

---

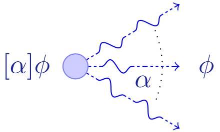
∑ Net win for soundness since significantly simpler prover

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

differential dynamic logic

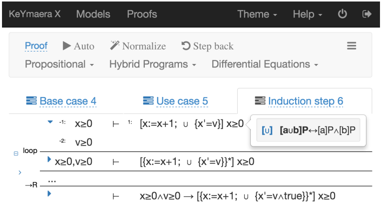
$\mathsf{dL} = \mathsf{DL} + \mathsf{HP}$

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



- Uniform substitution
  $\rightsquigarrow$ axioms not schemata
- Modular: Logic $\|$ Prover
- Straightforward to implement
- Prover microkernel
- Sound & complete / ODE
- Fast contextual equivalence

KeYmaera X



Uniform Substitution of Rules and Proofs

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

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

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

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) $(\text{FV}(\sigma) = \emptyset)$

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

Locally sound

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

Uniform Substitution of Rules and Proofs

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

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

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

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

Locally sound

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

Uniform Substitution of Rules and Proofs

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 θ
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 ϕ

Quite nontrivial soundness-critical side condition algorithms ...

Uniform Substitution

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

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

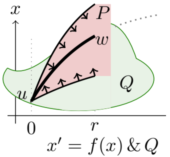
Uniform substitution σ replaces all occurrences of $p(\theta)$ for any θ by $\psi(\theta)$
function sym. $f(\theta)$ for any θ by $\eta(\theta)$
program sym. a by α

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

Differential Invariants for Differential Equations

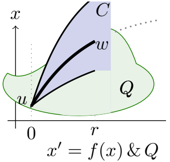
Differential Invariant

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



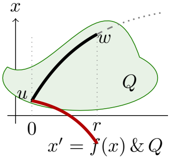
Differential Cut

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



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

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

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

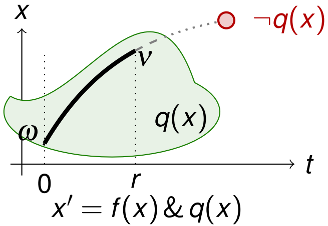
$+'$ $(f(\bar{x}) + g(\bar{x}))' = (f(\bar{x}))' + (g(\bar{x}))'$

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

c' $(c)' = 0$

Axiom (Differential Weakening) (JAR'17)

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

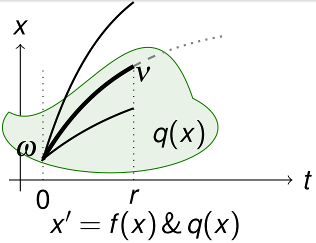


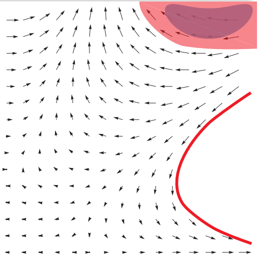
Differential equations cannot leave their evolution domains. Derives from:

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

Axiom (Differential Cut) (JAR'17)

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





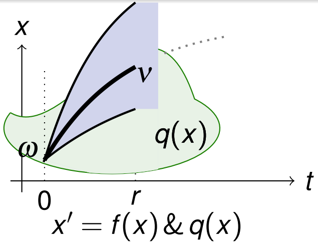
DC is a cut for differential equations.

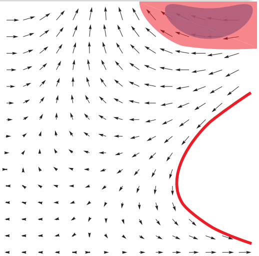
DC is a differential modal modus ponens K.

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

Axiom (Differential Cut) (JAR'17)

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

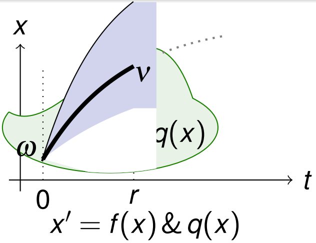


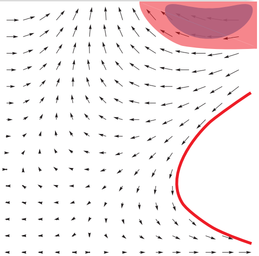


DC is a cut for differential equations.
DC is a differential modal modus ponens K.
Can't leave $r(x)$, then might as well restrict state space to $r(x)$.

Axiom (Differential Cut) (JAR'17)

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

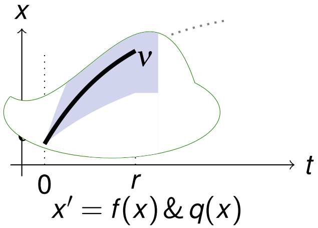


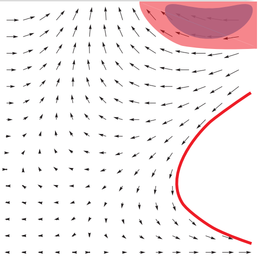


DC is a cut for differential equations.
DC is a differential modal modus ponens K.
Can't leave $r(x)$, then might as well restrict state space to $r(x)$.

Axiom (Differential Cut) (JAR'17)

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

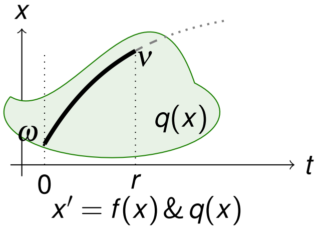


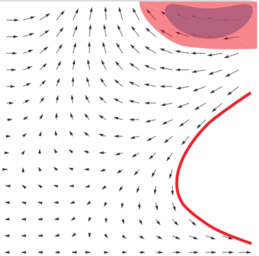


DC is a cut for differential equations.
DC is a differential modal modus ponens K.
Can't leave $r(x)$, then might as well restrict state space to $r(x)$.

Axiom (Differential Cut) (JAR'17)

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

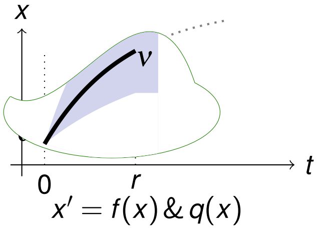


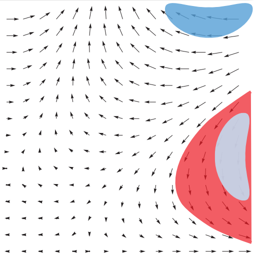


DC is a cut for differential equations.
DC is a differential modal modus ponens K.
Can't leave $r(x)$, then might as well restrict state space to $r(x)$.

Axiom (Differential Cut) (JAR'17)

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





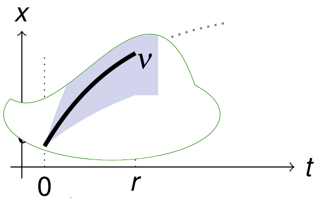
DC is a cut for differential equations.

DC is a differential modal modus ponens K.

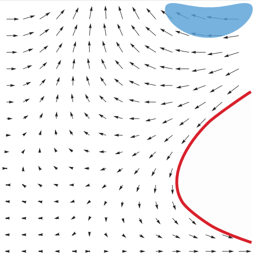
Can't leave $r(x)$, then might as well restrict state space to $r(x)$.

Axiom (Differential Cut) (JAR'17)

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



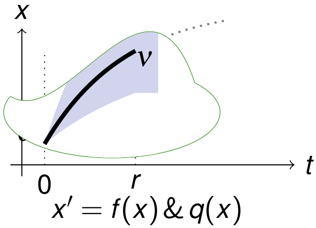
$x' = f(x)\,\&\,q(x)$

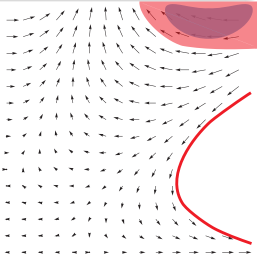


DC is a cut for differential equations.
DC is a differential modal modus ponens K.
Can't leave $r(x)$, then might as well restrict state space to $r(x)$.

Axiom (Differential Cut) (JAR'17)

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





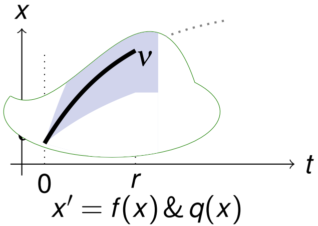
DC is a cut for differential equations.

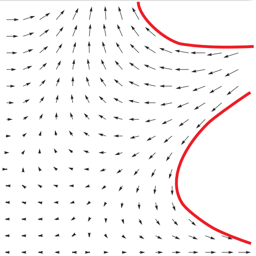
DC is a differential modal modus ponens K.

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

Axiom (Differential Cut) (JAR'17)

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

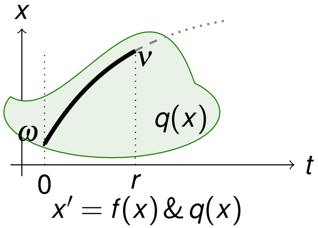


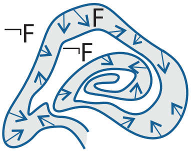


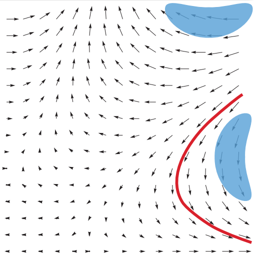
DC is a cut for differential equations.
DC is a differential modal modus ponens K.
Can't leave $r(x)$, then might as well restrict state space to $r(x)$.

Axiom (Differential Invariant) (JAR'17)

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







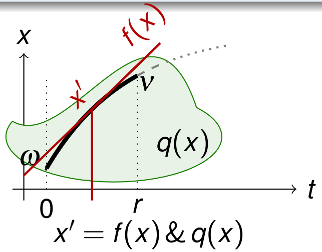
Differential invariant: if $p(x)$ true now and if differential $(p(x))'$ true always

What's the differential of a formula???

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

Axiom (Differential Effect) (JAR'17)

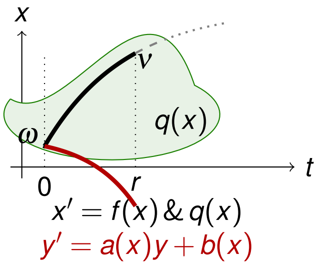
DE $[x' = f(x) \,\&\, q(x)]p(x,x') \leftrightarrow [x' = f(x) \,\&\, q(x)][x' := f(x)]p(x,x')$

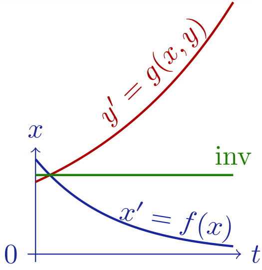


Effect of differential equation on differential symbol x'
$[x' := f(x)]$ instantly mimics continuous effect $[x' = f(x)]$ on x'
$[x' := f(x)]$ selects vector field $x' = f(x)$ for subsequent differentials

Axiom (Differential Ghost) (JAR'17)

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





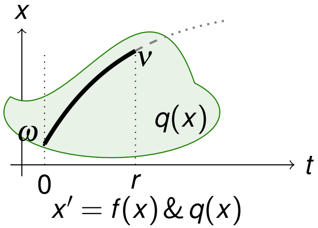
Differential ghost/auxiliaries: extra differential equations that exist

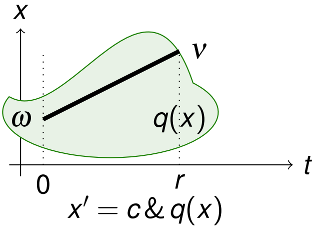
Can cause new invariants

"Dark matter" counterweight to balance conserved quantities

Axiom (Differential Solution) (JAR'17)

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





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

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)

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

Differential Substitution Lemmas

Lemma (Differential lemma)

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

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

Lemma (Differential assignment)

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

Lemma (Derivations)

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

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

Lemma (Differential assignment)

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

Lemma (Derivations)

$$(\theta + \eta)' = (\theta)' + (\eta)'$$
$$(\theta \cdot \eta)' = (\theta)' \cdot \eta + \theta \cdot (\eta)'$$
$$(c)' = 0 \qquad \textit{for arity 0 functions } c$$

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

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' $(c)' = 0$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

1. Start with identity differential computation result

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

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

1. Start with identity differential computation result which proves

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

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

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

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

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

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

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

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

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

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

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

Example: Differential Invariants

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

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

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

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

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

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

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

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

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

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* ϕ

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

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

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

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

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* ϕ

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

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

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

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

Correctness of Static Semantics

Lemma (Bound effect lemma) (Only BV($\cdot$) change)

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

Lemma (Coincidence lemma) (Only FV($\cdot$) 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* $\tilde{\omega} \in J[\![\phi]\!]$

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

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* $\tilde\omega \in J[\![\phi]\!]$

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

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

Differential Dynamic Logic dL: Static Semantics

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

Differential Dynamic Logic dL: Static Semantics

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

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

Differential Dynamic Logic dL: Static Semantics

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

Differential Dynamic Logic dL: Static Semantics

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

$$
\begin{aligned}
\mathsf{BV}(\theta \geq \eta) = \mathsf{BV}(p(\theta_1,\dots,\theta_k)) &= \emptyset \\
\mathsf{BV}(\phi \land \psi) &= \mathsf{BV}(\phi) \cup \mathsf{BV}(\psi) \\
\mathsf{BV}(\forall x\, \phi) = \mathsf{BV}(\exists x\, \phi) &= \{x\} \cup \mathsf{BV}(\phi) \\
\mathsf{BV}([\alpha]\phi) = \mathsf{BV}(\langle\alpha\rangle\phi) &= \mathsf{BV}(\alpha) \cup \mathsf{BV}(\phi) \\
\hline
\mathsf{BV}(a) &= \mathscr{V} && \text{for program symbol } a \\
\mathsf{BV}(x := \theta) &= \{x\} \\
\mathsf{BV}(?Q) &= \emptyset \\
\mathsf{BV}(x' = \theta \,\&\, Q) &= \{x, x'\} \\
\mathsf{BV}(\alpha \cup \beta) = \mathsf{BV}(\alpha;\beta) &= \mathsf{BV}(\alpha) \cup \mathsf{BV}(\beta) \\
\mathsf{BV}(\alpha^*) &= \mathsf{BV}(\alpha) \\
\hline
\mathsf{MBV}(a) &= \emptyset && \text{program symbol } a \\
\mathsf{MBV}(\alpha) &= \mathsf{BV}(\alpha) && \text{other atomic HPs } \alpha \\
\mathsf{MBV}(\alpha \cup \beta) &= \mathsf{MBV}(\alpha) \cap \mathsf{MBV}(\beta) \\
\mathsf{MBV}(\alpha;\beta) &= \mathsf{MBV}(\alpha) \cup \mathsf{MBV}(\beta) \\
\mathsf{MBV}(\alpha^*) &= \emptyset
\end{aligned}
$$

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* $\tilde\omega \in J[\![\phi]\!]$

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

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* ϕ

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

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

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

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

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

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' $(c)' = 0$

André Platzer.
Logical Foundations of Cyber-Physical Systems.
Springer, Cham, 2018.
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.

Definition (Term semantics) ($[\![\cdot]\!] : \mathrm{Trm} \to (\mathscr{S} \to \mathbb{R})$)

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

Definition (dL semantics) ($[\![\cdot]\!] : \mathrm{Fml} \to \wp(\mathscr{S})$)

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

P valid iff $\omega \in [\![P]\!]$ for all states ω of all interpretations I

Definition (Program semantics) ($[\![\cdot]\!] : \mathrm{HP} \to \wp(\mathscr{S} \times \mathscr{S})$)

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

Differential Dynamic Logic with Interpretations: Semantics

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

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

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

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

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

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

Definition (Term semantics) ($[\![\cdot]\!] : \mathrm{Trm} \to (\mathscr{S} \to \mathbb{R})$)

$$\omega[\![f(\theta_1,\dots,\theta_k)]\!] = I(f)\big(\omega[\![\theta_1]\!],\dots,\omega[\![\theta_k]\!]\big) \qquad I(f) : \mathbb{R}^k \to \mathbb{R} \text{ smooth}$$

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

Definition (dL semantics) ($[\![\cdot]\!] : \mathrm{Fml} \to \wp(\mathscr{S})$)

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

Definition (Program semantics) ($[\![\cdot]\!] : \mathrm{HP} \to \wp(\mathscr{S} \times \mathscr{S})$)

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

Differential Dynamic Logic with Interpretations: Semantics

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

$$\omega\llbracket f(\theta_1,\dots,\theta_k)\rrbracket = I(f)\big(\omega\llbracket\theta_1\rrbracket,\dots,\omega\llbracket\theta_k\rrbracket\big) \qquad I(f) : \mathbb{R}^k \to \mathbb{R} \text{ smooth}$$

$$\omega\llbracket(\theta)'\rrbracket = \sum_x \omega(x')\frac{\partial\llbracket\theta\rrbracket}{\partial x}(\omega)$$

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

$$\llbracket p(\theta_1,\dots,\theta_k)\rrbracket = \{\omega : (\omega\llbracket\theta_1\rrbracket,\dots,\omega\llbracket\theta_k\rrbracket) \in I(p)\} \qquad I(p) \subseteq \mathbb{R}^k$$

$$\llbracket\langle\alpha\rangle\phi\rrbracket = \llbracket\alpha\rrbracket \circ \llbracket\phi\rrbracket$$

$$\llbracket[\alpha]\phi\rrbracket = \llbracket\neg\langle\alpha\rangle\neg\phi\rrbracket$$

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

$$\llbracket a\rrbracket = I(a) \qquad I(a) \subseteq \mathscr{S} \times \mathscr{S}$$

$$\llbracket x := \theta\rrbracket = \{(\omega,\nu) : \nu = \omega \text{ except } \nu\llbracket x\rrbracket = \omega\llbracket\theta\rrbracket\}$$

$$\llbracket ?Q\rrbracket = \{(\omega,\omega) : \omega \in \llbracket Q\rrbracket\}$$

$$\llbracket x' = f(x)\,\&\,Q\rrbracket = \{(\varphi(0)|_{\{x'\}^\complement},\varphi(r)) : \varphi \models x' = f(x) \land Q\}$$

$$\llbracket\alpha\cup\beta\rrbracket = \llbracket\alpha\rrbracket \cup \llbracket\beta\rrbracket$$

$$\llbracket\alpha;\beta\rrbracket = \llbracket\alpha\rrbracket \circ \llbracket\beta\rrbracket$$

$$\llbracket\alpha^*\rrbracket = \big(\llbracket\alpha\rrbracket\big)^* = \bigcup_{n\in\mathbb{N}} \llbracket\alpha^n\rrbracket$$