Heterogeneous Dynamic Logic

Provability Modulo Theories

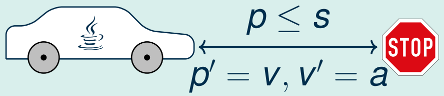
PLDI'26

Samuel Teuber, Mattias Ulbrich, André Platzer, Bernhard Beckert | 19.06.2026

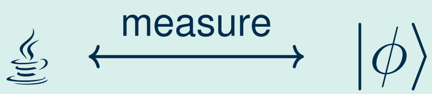
Heterogeneous Systems

Many (Software) Systems are **inherently heterogeneous**

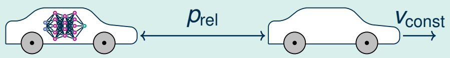
Software + CPS (our case study)



Classical + Quantum Software Klamroth et al. 2023



Neural Network + CPS NeurIPS'24

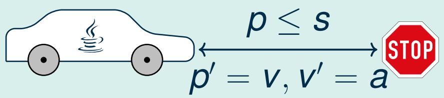


$\cdots$

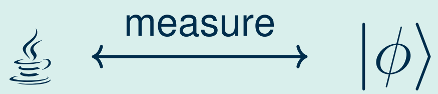
Heterogeneous Systems

Many (Software) Systems are **inherently heterogeneous**

Software + CPS (our case study)



Classical + Quantum Software Klamroth et al. 2023



Neural Network + CPS NeurIPS'24

$\dots$

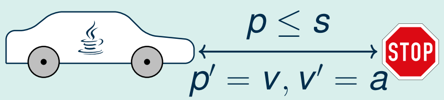
Verifying heterogeneous systems benefits from **combining verification techniques**

$\Rightarrow \mathcal{O}\left(N^2\right)$ custom solutions?

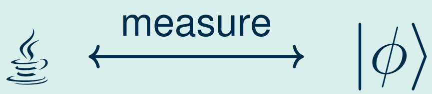
Heterogeneous Systems

Many (Software) Systems are **inherently heterogeneous**

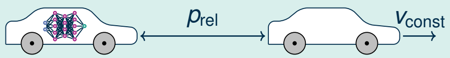
Software + CPS (our case study)



Classical + Quantum Software Klamroth et al. 2023



Neural Network + CPS NeurIPS'24



...

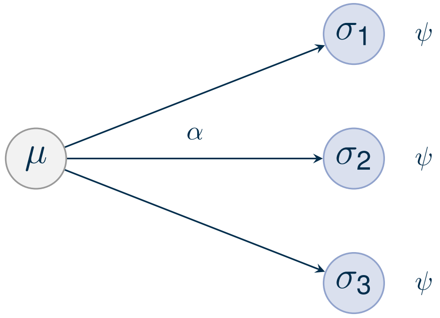
Verifying heterogeneous systems benefits from **combining verification techniques**
$\Rightarrow \mathcal{O}\left(N^2\right)$ custom solutions?

How do we do this in a rigorous manner for program logics?

What is Dynamic Logic?

Modal logic with **programs** as modalities:

$[\alpha]\,\psi$ — all runs of α satisfy ψ

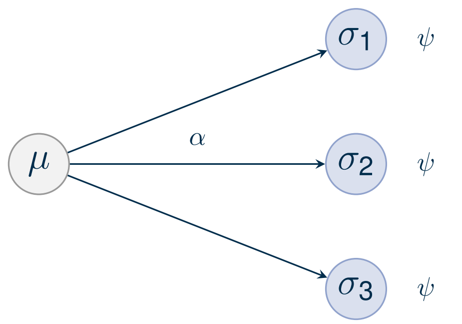


$\mu \models [\alpha]\,\psi$

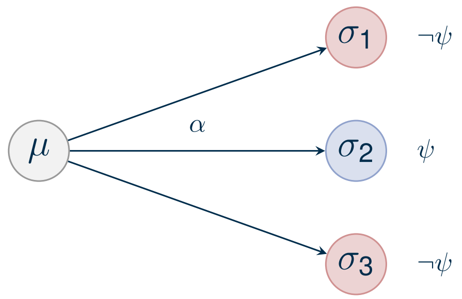
What is Dynamic Logic?

Modal logic with **programs** as modalities:

$[\alpha]\,\psi$	all runs of α satisfy ψ
$\langle\alpha\rangle\,\psi \equiv \neg[\alpha]\,\neg\psi$	some run of α satisfies ψ



$\mu \models [\alpha]\,\psi$



$\mu \models \langle\alpha\rangle\,\psi$

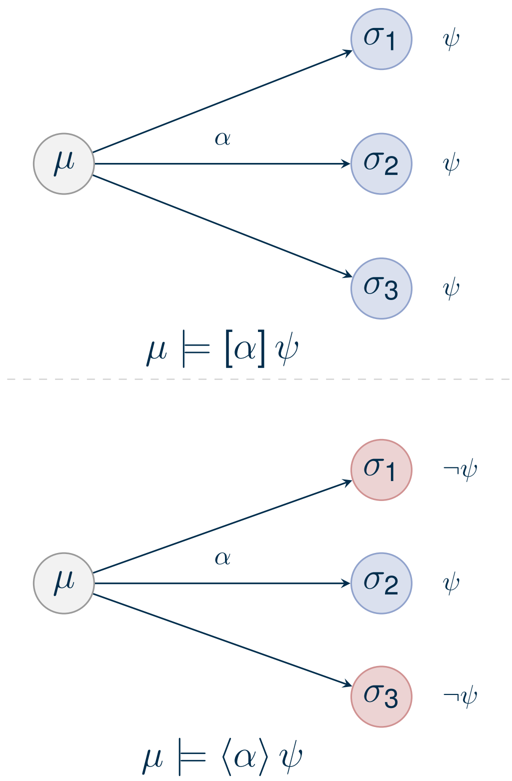
What is Dynamic Logic?

Modal logic with **programs** as modalities:

$[\alpha]\,\psi$	all runs of α satisfy ψ
$\langle\alpha\rangle\,\psi \equiv \neg[\alpha]\,\neg\psi$	some run of α satisfies ψ

Hoare logic is a special case

$$\{\phi\}\ \alpha\ \{\psi\} \quad \equiv \quad \phi\ \rightarrow\ [\alpha]\,\psi$$



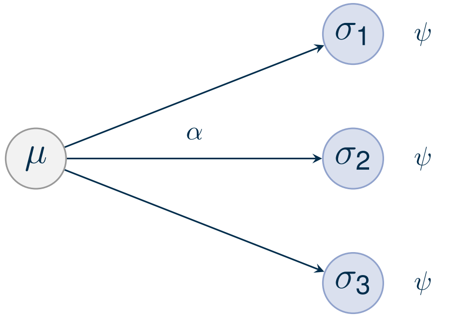
What is Dynamic Logic?

Modal logic with **programs** as modalities:

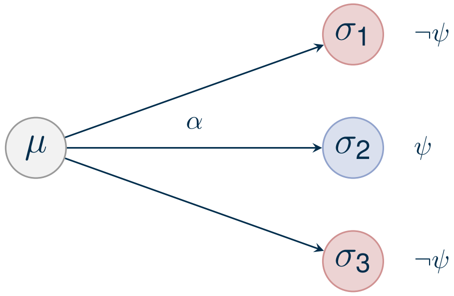
$[\alpha]\,\psi$	all runs of α satisfy ψ
$\langle\alpha\rangle\,\psi \equiv \neg[\alpha]\,\neg\psi$	some run of α satisfies ψ

Hoare logic is a special case

$$\{\phi\}\;\alpha\;\{\psi\} \quad\equiv\quad \phi \;\rightarrow\; [\alpha]\,\psi$$



$\mu \models [\alpha]\,\psi$



$\mu \models \langle\alpha\rangle\,\psi$

$$\langle\cdot\rangle\psi \qquad \langle\cdot\rangle\langle\cdot\rangle\psi \qquad [\cdot]\,[\cdot]\,\psi \qquad \langle\cdot\rangle\,[\cdot]\,\psi \qquad [\cdot]\,\langle\cdot\rangle\psi \qquad \langle\cdot\rangle\langle\cdot\rangle\langle\cdot\rangle\psi \qquad \cdots$$

Contribution

Heterogeneous Dynamic Logic

Contribution

Heterogeneous Dynamic Logic

Frame individual dynamic logics as **dynamic theories** ( )

Contribution

Heterogeneous Dynamic Logic

Frame individual dynamic logics as **dynamic theories** ( )

Lifting

Extend theory with features and calculus

Contribution

Heterogeneous Dynamic Logic

Frame individual dynamic logics as **dynamic theories** ()

Lifting

Extend theory with features and calculus

Combination

Reasoning across theories

Contribution

Heterogeneous Dynamic Logic

Frame individual dynamic logics as **dynamic theories** ( )

Lifting

Extend theory with features and calculus

Combination

Reasoning across theories

BYOP

Contribution

Heterogeneous Dynamic Logic

Frame individual dynamic logics as **dynamic theories** ( )

Lifting

Extend theory with features and calculus

Combination

Reasoning across theories

BYOP (*Bring Your Own Proof*)

$$\dfrac{\dfrac{\dfrac{\text{Proof for System 1}}{\phi_1 \vdash [p_1]\ \psi_1}}{\cdots} \qquad \cdots \qquad \dfrac{\dfrac{\text{Proof for System 2}}{\phi_2 \vdash [p_2]\ \psi_2}}{\cdots}}{\dfrac{\cdots}{\vdash \phi \to [(p_1; \ldots; p_2)^*]\ \psi}}$$

Contribution

Heterogeneous Dynamic Logic

Frame individual dynamic logics as **dynamic theories** ( )

Lifting

Extend theory with features and calculus

Combination

Reasoning across theories

BYOP (*Bring Your Own Proof*)

$$\dfrac{\dfrac{\dfrac{\dfrac{\text{Proof for System 1}}{\phi_1 \vdash [p_1]\ \psi_1}}{\cdots} \qquad \cdots \qquad \dfrac{\dfrac{\text{Proof for System 2}}{\phi_2 \vdash [p_2]\ \psi_2}}{\cdots}}{\cdots}}{\vdash \phi \rightarrow [(p_1; \ldots ; p_2)^*]\ \psi}$$

- Calculus formalized in Isabelle/HOL

Contribution

Heterogeneous Dynamic Logic

Frame individual dynamic logics as **dynamic theories** ()

Lifting

Extend theory with features and calculus

BYOP (*Bring Your Own Proof*)

$$\dfrac{\dfrac{\dfrac{\text{Proof for System 1}}{\phi_1 \vdash [p_1]\ \psi_1}}{\cdots} \quad \cdots \quad \dfrac{\dfrac{\text{Proof for System 2}}{\phi_2 \vdash [p_2]\ \psi_2}}{\cdots}}{\dfrac{\cdots}{\vdash \phi \rightarrow [(p_1; \ldots; p_2)^*]\ \psi}}$$

Combination

Reasoning across theories

- Calculus formalized in Isabelle/HOL
- Relative Completeness of Calculi

What is a Dynamic *Theory*?

Core Idea: Abstracting Away Programs and Atoms

$$y \geq 0 \ \land \ x \geq 0 \rightarrow [\quad (x{:=}x{+}1)^{*} \quad] (\ x \geq 0 \ \land \ y \geq 0)$$

Core Idea: Abstracting Away Programs and Atoms

[redacted] $\wedge$ [redacted] $\to [$ [redacted] $]\,($ [redacted] $\wedge$ [redacted] $)$

Core Idea: Abstracting Away Programs and Atoms

$\forall x$ [redacted] $\land$ [redacted] $\rightarrow [$ [redacted] $](\,$[redacted] $\land$ [redacted]$)$

Core Idea: Abstracting Away Programs and Atoms

$$\forall x \; \boxed{a_1} \wedge \boxed{a_2} \rightarrow [\; \boxed{p} \;](\; \boxed{b_1} \wedge \boxed{b_2} \;)$$

Definition (Dynamic Signature)

A *dynamic signature* $\Lambda = (\Lambda_V, \Lambda_A, \Lambda_P)$ is determined by:

Variables Λ_V	Atoms Λ_A	Programs Λ_P
$\{x\}$	$\{a_1, a_2, b_1, b_2\}$	$\{p\}$

Core Idea: Abstracting Away Programs and Atoms

$$\forall x \; \boxed{a_1} \;\wedge\; \boxed{a_2} \rightarrow [\; \boxed{p} \;]\,(\; \boxed{b_1} \;\wedge\; \boxed{b_2}\;)$$

Definition (Dynamic Signature)

A *dynamic signature* $\Lambda = (\Lambda_V, \Lambda_A, \Lambda_P)$ is determined by:

Variables Λ_V	Atoms Λ_A	Programs Λ_P
$\{x\}$	$\{a_1, a_2, b_1, b_2\}$	$\{p\}$

Definition (Syntax of Formulas)

With $p \in \Lambda_P$, $a \in \Lambda_A$, $v \in \Lambda_V$:

$$F, G \quad ::= \quad a \mid \neg F \mid F \wedge G \mid \forall v\ F \mid [p]\ F$$

Semantics: Domains of Computation $\mathcal{D}$

State Space

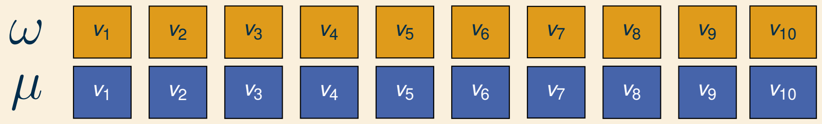
- Universe $\mathcal{U}$
- State Space $\mathcal{S}$
- Variable Evaluation $\mathcal{V} : \mathcal{S} \times \Lambda_V \to \mathcal{U}$

Semantics: Domains of Computation $\mathcal{D}$

State Space

- Universe $\mathcal{U}$
- State Space $\mathcal{S}$
- Variable Evaluation $\mathcal{V} : \mathcal{S} \times \Lambda_V \to \mathcal{U}$

$\mathcal{S}$ Independence



Semantics: Domains of Computation $\mathcal{D}$

State Space

- Universe $\mathcal{U}$
- State Space $\mathcal{S}$
- Variable Evaluation $\mathcal{V} : \mathcal{S} \times \Lambda_V \to \mathcal{U}$

$\mathcal{S}$ Independence

$$\overbrace{v_1 \quad v_2 \quad v_3 \quad v_4 \quad v_5}^{V}$$

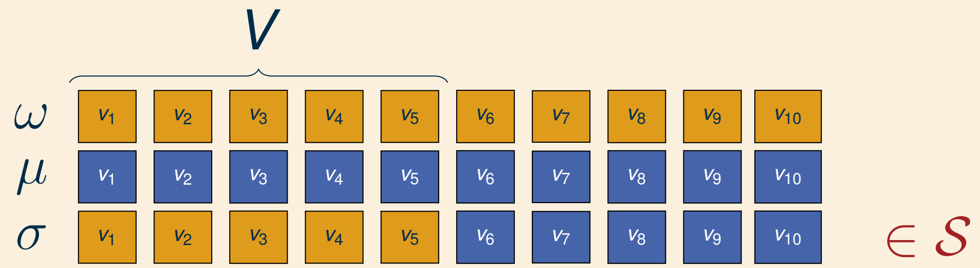
ω	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}
μ	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}

Semantics: Domains of Computation $\mathcal{D}$

State Space

- Universe $\mathcal{U}$
- State Space $\mathcal{S}$
- Variable Evaluation $\mathcal{V} : \mathcal{S} \times \Lambda_V \to \mathcal{U}$

$\mathcal{S}$ Independence



Semantics: Domains of Computation $\mathcal{D}$

State Space

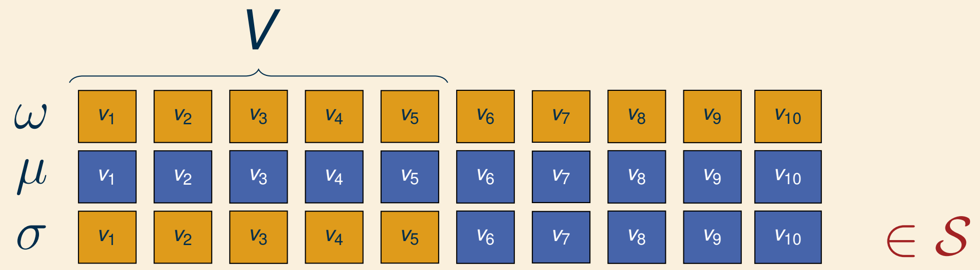
- Universe $\mathcal{U}$
- State Space $\mathcal{S}$
- Variable Evaluation $\mathcal{V} : \mathcal{S} \times \Lambda_V \to \mathcal{U}$

Atoms

For every atom $a \in \Lambda_A$:

- Atom Evaluation $\mathcal{A}(a) \subseteq \mathcal{S}$
- Free Variables $\mathbf{FV}_A(a) \subseteq \Lambda_V$

$\mathcal{S}$ Independence

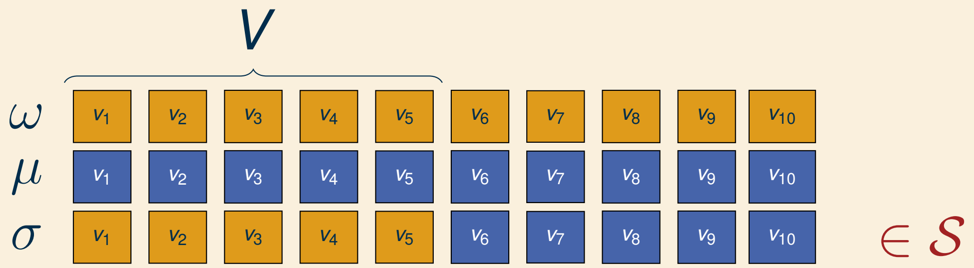


Semantics: Domains of Computation $\mathcal{D}$

State Space

- Universe $\mathcal{U}$
- State Space $\mathcal{S}$
- Variable Evaluation $\mathcal{V} : \mathcal{S} \times \Lambda_V \to \mathcal{U}$

$\mathcal{S}$ Independence



Atoms

For every atom $a \in \Lambda_A$:

- Atom Evaluation $\mathcal{A}(a) \subseteq \mathcal{S}$
- Free Variables $\mathbf{FV}_A(a) \subseteq \Lambda_V$

Programs

For every program $p \in \Lambda_P$:

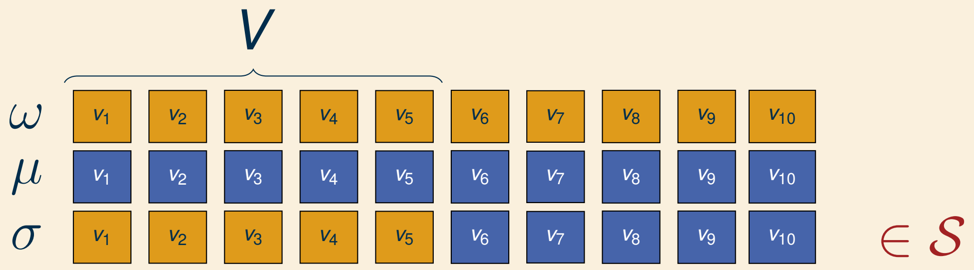
- Program Evaluation $\mathcal{P}(p) \subseteq \mathcal{S}^2$
- Free Variables $\mathbf{FV}_P(p) \subseteq \Lambda_V$

Semantics: Domains of Computation $\mathcal{D}$

State Space

- Universe $\mathcal{U}$
- State Space $\mathcal{S}$
- Variable Evaluation $\mathcal{V} : \mathcal{S} \times \Lambda_V \to \mathcal{U}$

$\mathcal{S}$ Independence



Atoms

For every atom $a \in \Lambda_A$:

- Atom Evaluation $\mathcal{A}(a) \subseteq \mathcal{S}$
- Free Variables $\mathbf{FV}_A(a) \subseteq \Lambda_V$

Programs

For every program $p \in \Lambda_P$:

- Program Evaluation $\mathcal{P}(p) \subseteq \mathcal{S}^2$
- Free Variables $\mathbf{FV}_P(p) \subseteq \Lambda_V$

Additional Assumptions (formalized)

Free variables *correct* | Free variables finite | $\mathcal{P}$ extensional w.r.t. $\mathcal{V}$

Dynamic Theory: Formula Semantics

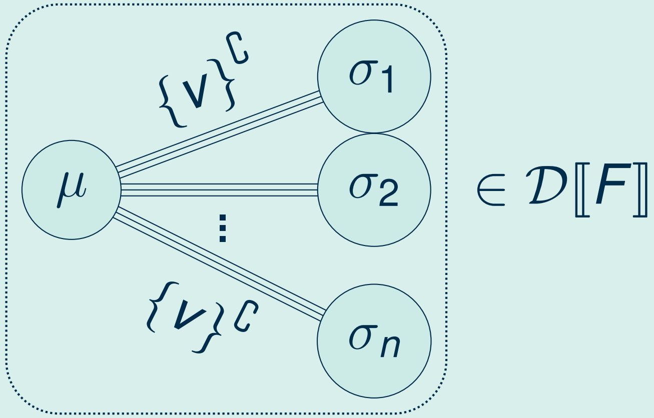
Atom Semantics

$$\mathcal{D}[\![a]\!] = \mathcal{A}(a)$$

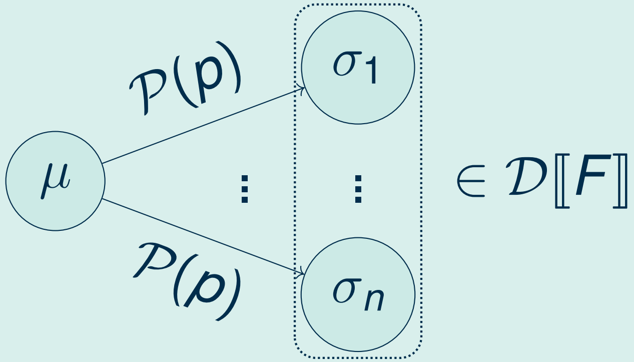
Propositional Semantics

$$\mathcal{D}[\![\neg F]\!] = \mathcal{S} \setminus \mathcal{D}[\![F]\!]$$
$$\mathcal{D}[\![F \wedge G]\!] = \mathcal{D}[\![F]\!] \cap \mathcal{D}[\![G]\!]$$

$\mu \in \mathcal{D}[\![\forall v\ F]\!]$



$\mu \in \mathcal{D}[\![[p]\ \ F]\!]$



Dynamic Theory Proof Rules and Axioms

Challenge:
No general syntactic analysis of free/bound variables for formulas and programs

Dynamic Theory Proof Rules and Axioms

Challenge:
No general syntactic analysis of free/bound variables for formulas and programs

Solution:
Semantic definitions for free variables FV_Δ and bound variables BV_Δ Platzer 2017
$\Rightarrow$ Use any (syntactic) superset for proofs (e.g. $\mathbf{FV}_P$)

Dynamic Theory Proof Rules and Axioms

Challenge:
No general syntactic analysis of free/bound variables for formulas and programs

Solution:
Semantic definitions for free variables FV_Δ and bound variables BV_Δ Platzer 2017
$\Rightarrow$ Use any (syntactic) superset for proofs (e.g. $\mathbf{FV}_P$)

$$(G)\ \frac{\vdash\ \phi}{\vdash\ [\alpha]\ \phi}$$

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

(B) $(\forall v\ [\alpha]\phi) \leftrightarrow ([\alpha]\forall v\ \phi)$ $\quad(v \notin \mathrm{FV}_\Delta(\alpha) \cup \mathrm{BV}_\Delta(\alpha))$

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

Dynamic Theory Proof Rules and Axioms

Challenge:
No general syntactic analysis of free/bound variables for formulas and programs

Solution:
Semantic definitions for free variables FV_Δ and bound variables BV_Δ Platzer 2017
$\Rightarrow$ Use any (syntactic) superset for proofs (e.g. $\mathbf{FV}_P$)

$$(G)\ \frac{\vdash\ \phi}{\vdash\ [\alpha]\ \phi}$$

(V)	$\phi \to [\alpha]\phi$	$(\mathrm{FV}_\Delta(\phi) \cap \mathrm{BV}_\Delta(\alpha) = \emptyset)$
(B)	$(\forall v\ [\alpha]\phi) \leftrightarrow ([\alpha]\forall v\ \phi)$	$(v \notin \mathrm{FV}_\Delta(\alpha) \cup \mathrm{BV}_\Delta(\alpha))$
(K)	$([\alpha](\phi \to \psi) \to ([\alpha]\phi \to [\alpha]\psi))$	
(E)	$[\alpha]\phi \leftrightarrow [\beta]\phi$	(for KAT equality $\alpha \cong \beta$)

Instantiations of Dynamic Theories

- Propositional Dynamic Logic Fischer et al. 1977
- Differential Dynamic Logic (AFP formalization) Platzer 2008; Bohrer 2017
- JavaDL (w/o formalized proof) Beckert et al. 2016

Instantiations of Dynamic Theories

- Propositional Dynamic Logic — Fischer et al. 1977
- Differential Dynamic Logic (AFP formalization) — Platzer 2008; Bohrer 2017
- JavaDL (w/o formalized proof) — Beckert et al. 2016

Simpler Example

Assignments for semi ring expressions (e.g. $\mathbb{N}, \mathbb{Z}$ or $\mathbb{R}$):

$$y > 0 \to [x := 2 \cdot y] \; x > 0$$

Instantiations of Dynamic Theories

- Propositional Dynamic Logic Fischer et al. 1977
- Differential Dynamic Logic (AFP formalization) Platzer 2008; Bohrer 2017
- JavaDL (w/o formalized proof) Beckert et al. 2016

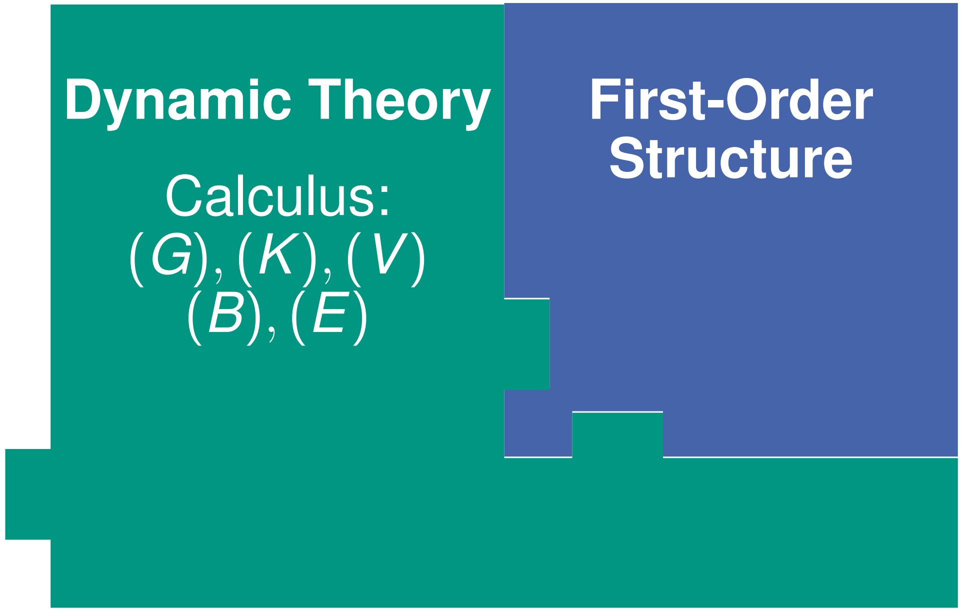
Simpler Example

Assignments for semi ring expressions (e.g. $\mathbb{N}, \mathbb{Z}$ or $\mathbb{R}$):

$$y > 0 \to [x := 2 \cdot y] \; x > 0$$

What is the use?

Dynamic Theories: Overview



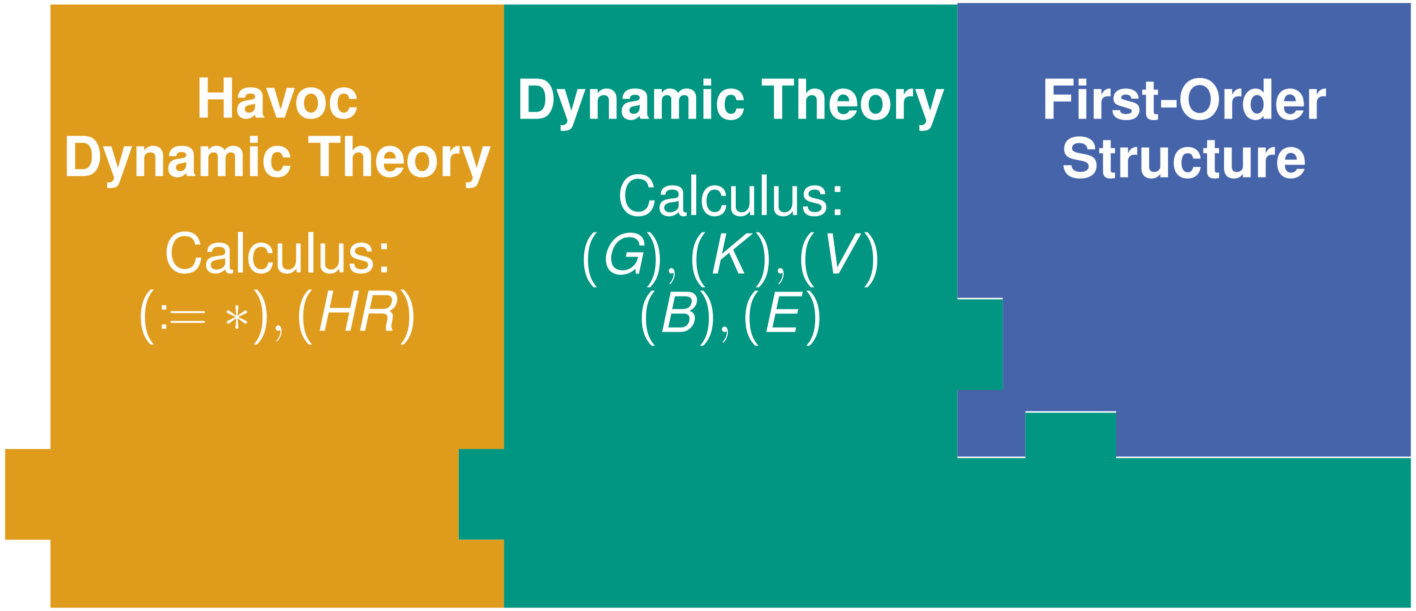
Results

- Dynamic Logic Formulas
- Coincidence & Bounded Effect Lemmas
- Proven Axioms: G, K, V, B, E

Assumptions

- Programs
- Free Variables & Extensionality

Dynamic Theories: Lifting



Proven Axioms

$$(:= *) \quad [v := *]\,\phi \leftrightarrow \forall v\, \phi$$

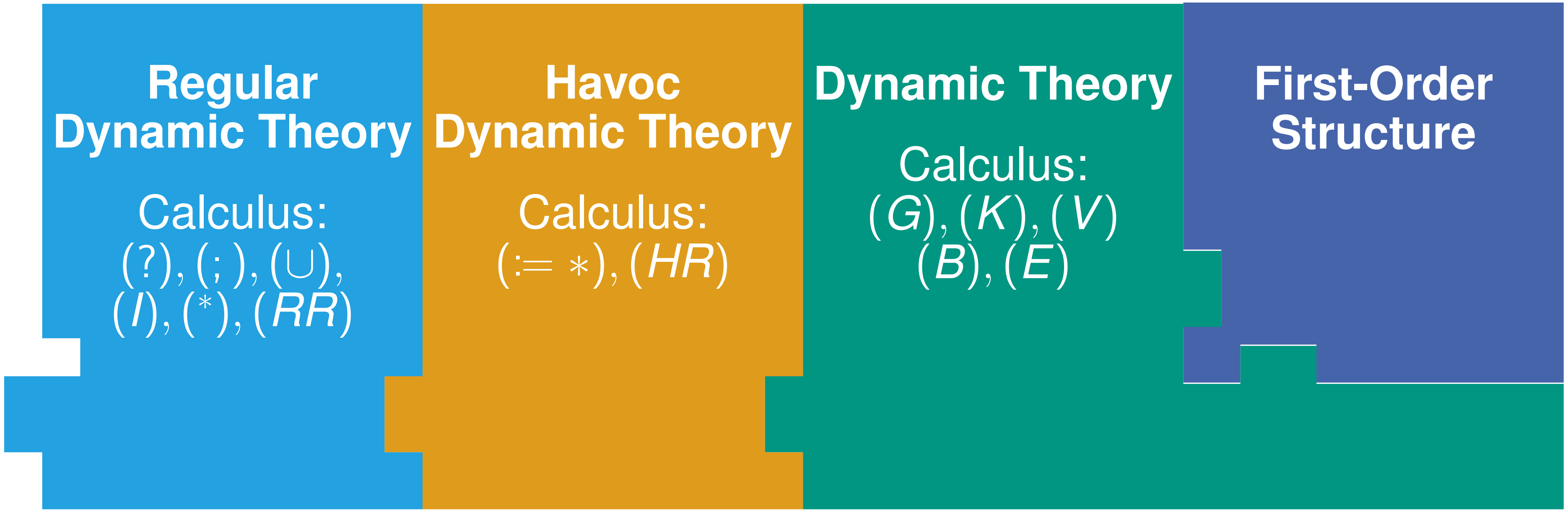
$$(HR) \quad \frac{\vdash_\Delta \phi}{\vdash_{\Delta^{:=}} \phi} \quad \text{for } \phi \in \mathrm{Fml}_\Delta$$

Havoc Lifting

Extended Language ($r \in \Lambda_P$):

$$p ::= r \mid x := *$$

Dynamic Theories: Lifting



Proven Axioms (and many more)

$$(I) \quad [\alpha^*](\phi \to [\alpha]\phi) \to (\phi \to [\alpha^*]\phi)$$

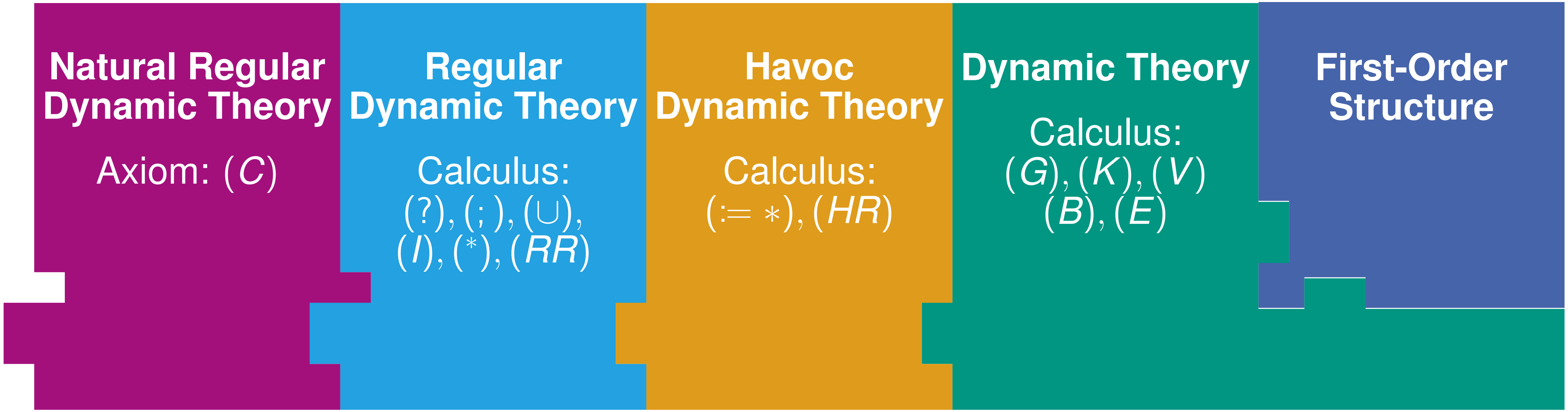
$$(RR) \quad \frac{\vdash_\Delta \phi}{\vdash_{\Delta^{\text{reg}}} \phi} \quad \text{for } \phi \in \mathrm{Fml}_\Delta$$

Regular Lifting

Extended Language ($r \in \Lambda_P$):

$$p, q ::= r \mid ?(P) \mid p; q \mid p \cup q \mid (p)^*$$

Dynamic Theories: Lifting



Proven Axiom

Loop Convergence

Assumptions

- Formula for “is positive”
- Formula for “is equal”
- Formula for “w is v plus one”

Semi Ring DL

Without Lifting

$$x > 0 \land y > 0 \rightarrow [x := x + y]\, x > 0$$

Semi Ring DL

Without Lifting

$$x > 0 \land y > 0 \to [x := x + y]\, x > 0$$

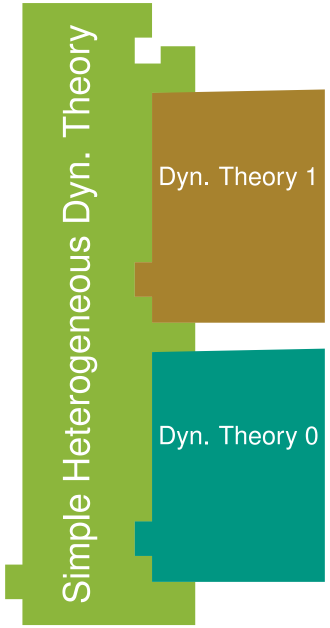
With Lifting (Havoc + Regular)

$$x > 0 \to \left[(y := *; ?(y > 0) ; (x := x + y))^*\right] x > 0$$

Combination

Dynamic Logic Combination: Overview

Dynamic Logic Combination: Overview



Combination: Syntax & Semantics

Variables: $\Lambda_V^{(01)} = \Lambda_V^{(0)} \dot\cup \Lambda_V^{(1)}$
Programs: $\Lambda_P^{(01)} = \Lambda_P^{(0)} \dot\cup \Lambda_P^{(1)}$

Combination: Syntax & Semantics

Variables: $\Lambda_V^{(01)} = \Lambda_V^{(0)} \dot\cup \Lambda_V^{(1)}$
Programs: $\Lambda_P^{(01)} = \Lambda_P^{(0)} \dot\cup \Lambda_P^{(1)}$
Universe: $\mathcal{U}^{(01)} = \mathcal{U}^{(0)} \cup \mathcal{U}^{(1)}$

Combination: Syntax & Semantics

Variables: $\Lambda_V^{(01)} = \Lambda_V^{(0)} \dot\cup \Lambda_V^{(1)}$
Programs: $\Lambda_P^{(01)} = \Lambda_P^{(0)} \dot\cup \Lambda_P^{(1)}$
Universe: $\mathcal{U}^{(01)} = \mathcal{U}^{(0)} \cup \mathcal{U}^{(1)}$
State Space: $\mathcal{S}^{(01)} := \mathcal{S}^{(0)} \times \mathcal{S}^{(1)}$

Combination: Syntax & Semantics

Variables: $\Lambda_V^{(01)} = \Lambda_V^{(0)} \dot{\cup} \Lambda_V^{(1)}$
Programs: $\Lambda_P^{(01)} = \Lambda_P^{(0)} \dot{\cup} \Lambda_P^{(1)}$
Universe: $\mathcal{U}^{(01)} = \mathcal{U}^{(0)} \cup \mathcal{U}^{(1)}$
State Space: $\mathcal{S}^{(01)} := \mathcal{S}^{(0)} \times \mathcal{S}^{(1)}$
Variable Evaluation by origin:

$$\mathcal{V}^{(01)}\left(\left(\mu^{(0)}, \mu^{(1)}\right), v\right) = \begin{cases} \mathcal{V}^{(0)}\left(\mu^{(0)}, v\right) & v \in \Lambda_V^{(0)} \\ \mathcal{V}^{(1)}\left(\mu^{(1)}, v\right) & v \in \Lambda_V^{(1)} \end{cases}$$

Combination: Syntax & Semantics

Variables: $\Lambda_V^{(01)} = \Lambda_V^{(0)} \dot\cup \Lambda_V^{(1)}$
Programs: $\Lambda_P^{(01)} = \Lambda_P^{(0)} \dot\cup \Lambda_P^{(1)}$
Universe: $\mathcal{U}^{(01)} = \mathcal{U}^{(0)} \cup \mathcal{U}^{(1)}$
State Space: $\mathcal{S}^{(01)} := \mathcal{S}^{(0)} \times \mathcal{S}^{(1)}$
Variable Evaluation by origin:

$$\mathcal{V}^{(01)}\left(\left(\mu^{(0)}, \mu^{(1)}\right), v\right) = \begin{cases} \mathcal{V}^{(0)}\left(\mu^{(0)}, v\right) & v \in \Lambda_V^{(0)} \\ \mathcal{V}^{(1)}\left(\mu^{(1)}, v\right) & v \in \Lambda_V^{(1)} \end{cases}$$

Atom Evaluation: By origin + **Additional atoms** for combined state space

Combination: Syntax & Semantics

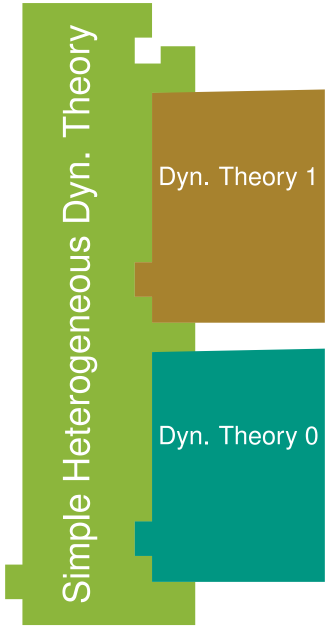
Variables: $\Lambda_V^{(01)} = \Lambda_V^{(0)} \dot{\cup} \Lambda_V^{(1)}$
Programs: $\Lambda_P^{(01)} = \Lambda_P^{(0)} \dot{\cup} \Lambda_P^{(1)}$
Universe: $\mathcal{U}^{(01)} = \mathcal{U}^{(0)} \cup \mathcal{U}^{(1)}$
State Space: $\mathcal{S}^{(01)} := \mathcal{S}^{(0)} \times \mathcal{S}^{(1)}$
Variable Evaluation by origin:

$$\mathcal{V}^{(01)}\left(\left(\mu^{(0)}, \mu^{(1)}\right), v\right) = \begin{cases} \mathcal{V}^{(0)}\left(\mu^{(0)}, v\right) & v \in \Lambda_V^{(0)} \\ \mathcal{V}^{(1)}\left(\mu^{(1)}, v\right) & v \in \Lambda_V^{(1)} \end{cases}$$

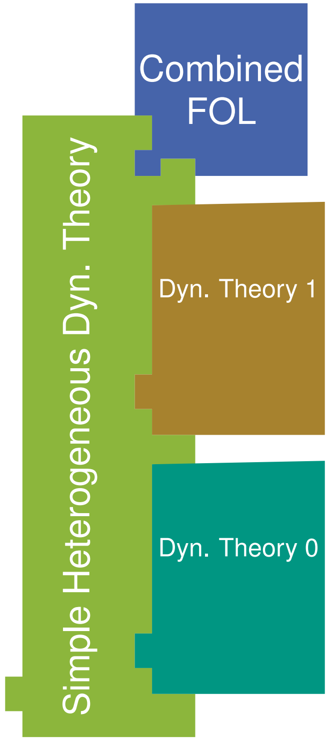
Atom Evaluation: By origin + **Additional atoms** for combined state space
Program Evaluation for $p^{(0)} \in \Lambda^{(0)}$:

$$\left(\left(\mu^{(0)}, \mu^{(1)}\right), \left(\sigma^{(0)}, \sigma^{(1)}\right)\right) \in \mathcal{P}^{(01)}\left(p^{(i)}\right) \text{ iff } \left(\mu^{(0)}, \sigma^{(0)}\right) \in \mathcal{P}^{(0)}\left(p^{(0)}\right) \text{ and } \mu^{(1)} \equiv_{\Lambda_V^{(1)}} \sigma^{(1)}$$

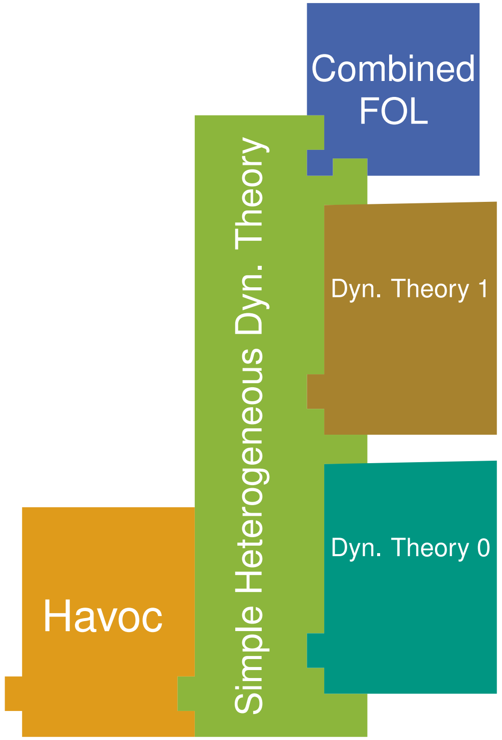
Dynamic Logic Combination: Overview



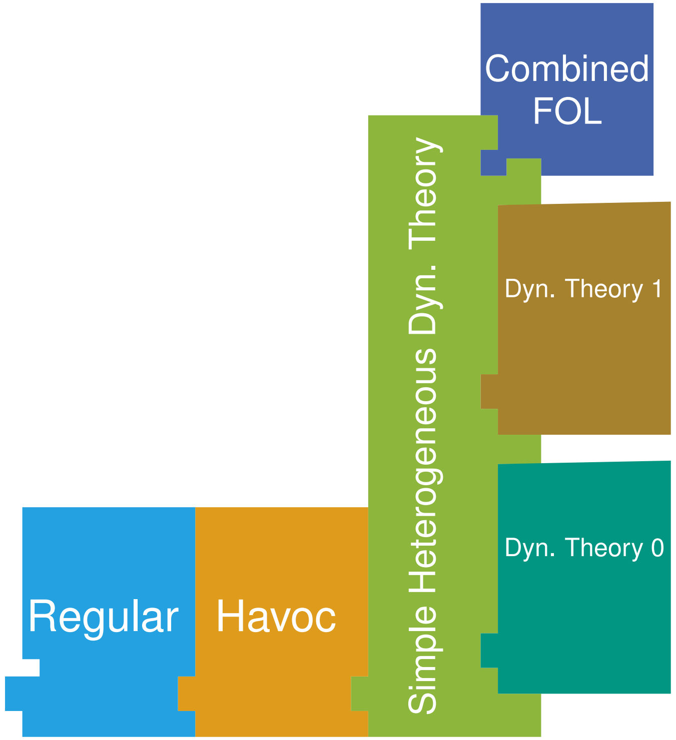
Dynamic Logic Combination: Overview



Dynamic Logic Combination: Overview



Dynamic Logic Combination: Overview



Combination: Calculus Rules

$$(HR0)\ \frac{\vdash_{\Delta^{(0)}} \phi}{\vdash_{\hat{\Delta}} \phi} \qquad \text{for } \phi \in \mathrm{Fml}_{\Delta^{(0)}}$$

$$(HR1)\ \frac{\vdash_{\Delta^{(1)}} \phi}{\vdash_{\hat{\Delta}} \phi} \qquad \text{for } \phi \in \mathrm{Fml}_{\Delta^{(1)}}$$

Combination: Calculus Rules

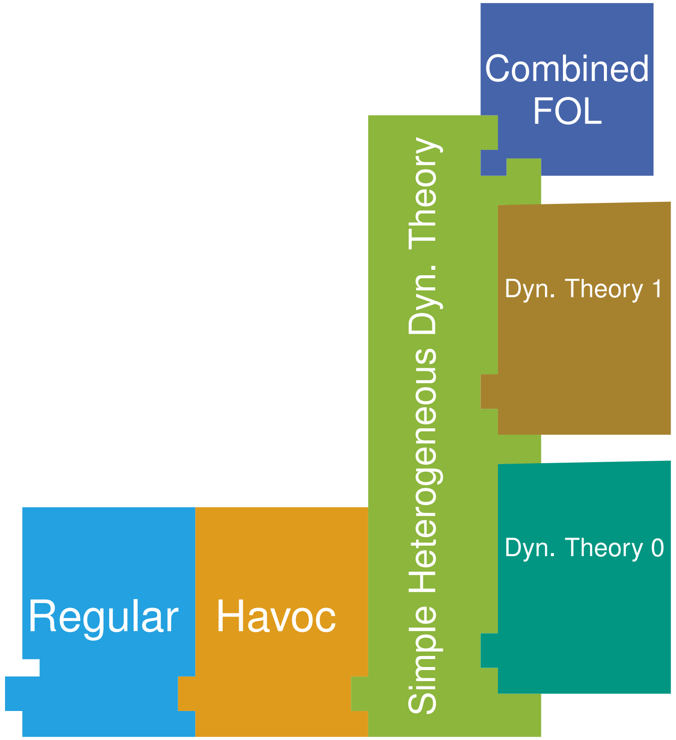
$$(HR0)\ \frac{\vdash_{\Delta^{(0)}} \phi}{\vdash_{\hat{\Delta}} \phi} \qquad \text{for } \phi \in \mathrm{Fml}_{\Delta^{(0)}}$$

$$(HR1)\ \frac{\vdash_{\Delta^{(1)}} \phi}{\vdash_{\hat{\Delta}} \phi} \qquad \text{for } \phi \in \mathrm{Fml}_{\Delta^{(1)}}$$

Derivable:

$$(F0)\quad (\psi \to [\alpha]\,\xi) \to (\phi \wedge \psi \to [\alpha]\,(\phi \wedge \xi)) \qquad \text{for } \alpha \in \Lambda_P^{(0)}, \phi \in \mathrm{Fml}_{\Lambda^{(1)}}$$

Dynamic Logic Combination: Overview



$$\underbrace{x > 0}_{\text{DT 0}} \rightarrow [\,\underbrace{x := 2 \cdot x}_{\text{DT 0}};\, y := *;\, ?(\underbrace{x}_{\text{DT 0}} \underbrace{\doteq}_{\text{FOL}} \underbrace{y}_{\text{DT 0}});\, \underbrace{y' = 2}_{\text{DT 1}}\,]\underbrace{y > 0}_{\text{DT 1}}$$

Completeness

Can I prove all valid formulas with this calculus?

Relative Completeness

Can I prove all valid formulas with this calculus?

Relativity Assumption:
Oracles for modality-free formulas

Relative Completeness

Can I prove all valid formulas with this calculus?

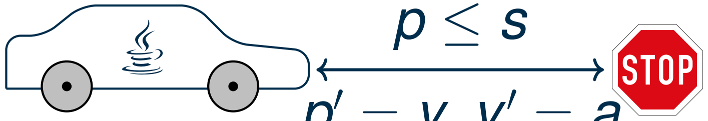
Relativity Assumption:
Oracles for modality-free formulas

Key Result

Relative completeness **preserved** by lifting/combination
(under reasonable assumptions)

(Pen and Paper Proofs Only)

Heterogeneous Dynamic Logic: A first example



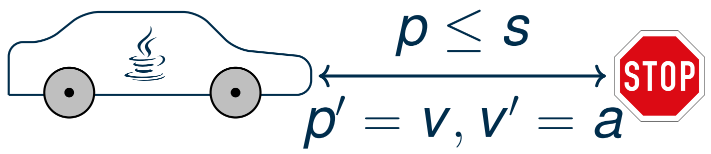
$p \leq s$

$p' = v, v' = a$

- Controller: Java
- Environment: Hybrid Program

Approach:
Heterogeneous Theory over JavaDL & $d\mathcal{L}$ with predicates toReal and round

Heterogeneous Dynamic Logic: A first example



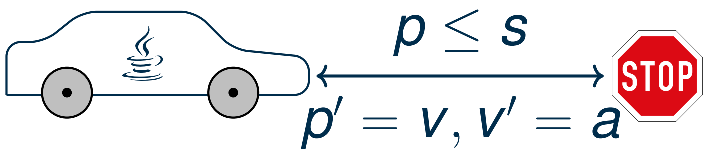
- Controller: Java
- Environment: Hybrid Program

Approach:
Heterogeneous Theory over JavaDL & $d\mathcal{L}$
with predicates toReal and round

$$\begin{pmatrix} \texttt{java\_pre} \wedge \\ dl\_pre \ \wedge \ \mathrm{coupling} \end{pmatrix} \rightarrow \Big[\big(\underbrace{\texttt{ctrl()}}_{\text{JavaDL}}; \mathrm{actuation}; \underbrace{env}_{d\mathcal{L}}; \mathrm{sensor}\big)^*\Big] p \leq s$$

$$\mathrm{actuation} \equiv \underbrace{a}_{d\mathcal{L}} := *;\ ?\big(toReal(a * 100, \overbrace{\texttt{this.acc}}^{\text{JavaDL}})\big)$$

Heterogeneous Dynamic Logic: A first example



- Controller: Java
- Environment: Hybrid Program

Approach:
Heterogeneous Theory over JavaDL & $d\mathcal{L}$
with predicates toReal and round

$$\begin{pmatrix} \texttt{java\_pre} \wedge \\ \mathit{dl\_pre} \wedge \mathrm{coupling} \end{pmatrix} \rightarrow \Big[\big(\underbrace{\texttt{ctrl()}}_{\text{JavaDL}};\mathrm{actuation};\underbrace{\mathit{env}}_{d\mathcal{L}};\mathrm{sensor}\big)^*\Big]p \leq s$$

$$\mathrm{actuation} \equiv \underbrace{a}_{d\mathcal{L}} := *;\ ?\big(\mathit{toReal}(a * 100, \overbrace{\texttt{this.acc}}^{\text{JavaDL}})\big)$$

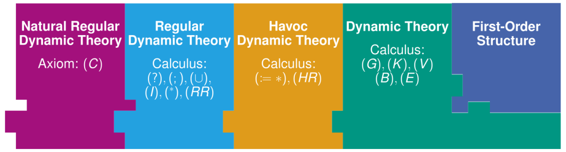
Proof relies on JavaDL & $d\mathcal{L}$ guarantees derived in KeY & KeYmaera X

Conclusion

- **Dynamic Theories**

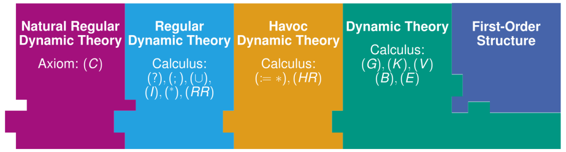
Conclusion

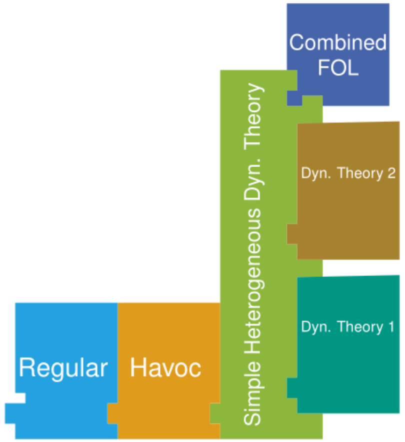
- Dynamic Theories
- Lifting



Conclusion

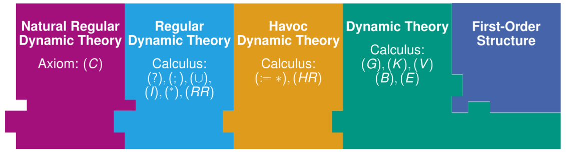
- **Dynamic Theories**
- **Lifting**
- **Combination**: BYOP

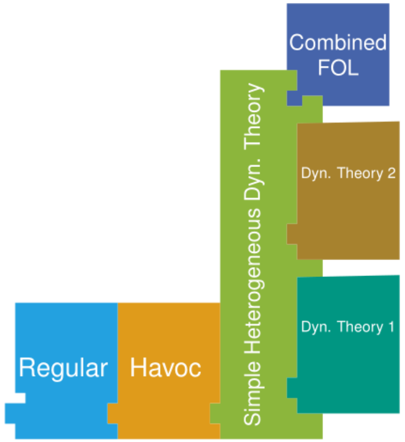




Conclusion

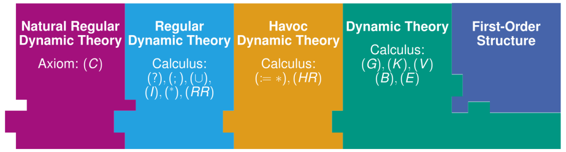
- **Dynamic Theories**
- **Lifting**
- **Combination**: BYOP
- **Relative Completeness**

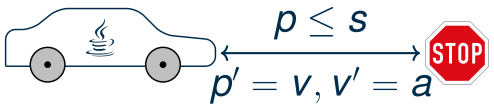


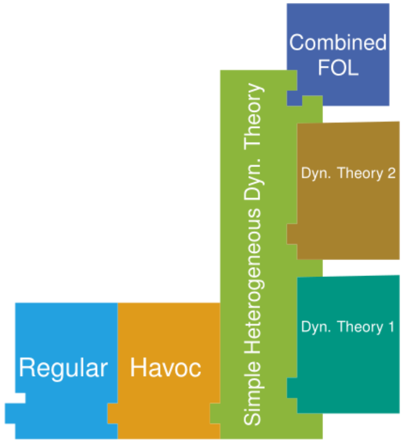


Conclusion

- **Dynamic Theories**
- **Lifting**
- **Combination**: BYOP
- **Relative Completeness**
- **Case Study**

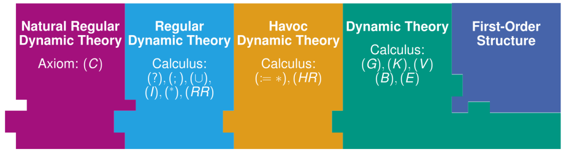


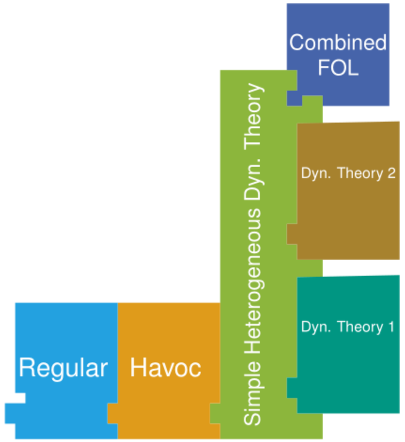


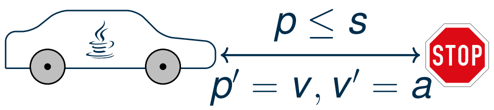


Conclusion

- **Dynamic Theories**
- **Lifting**
- **Combination**: BYOP
- **Relative Completeness**
- **Case Study**







Future Work

- More logics and case studies
- Integration with Neural Network verification ICFP'26

References I

[1] Bernhard Beckert, Vladimir Klebanov, and Benjamin Weiß. “Dynamic Logic for Java”. In: *Deductive Software Verification - The KeY Book - From Theory to Practice*. Ed. by Wolfgang Ahrendt et al. Lecture Notes in Computer Science. Springer, 2016, pp. 49–106. DOI: `10.1007/978-3-319-49812-6\_3`. URL: `https://doi.org/10.1007/978-3-319-49812-6%5C_3`.

[2] Bernhard Beckert and André Platzer. “Dynamic Logic with Non-rigid Functions: A Basis for Object-oriented Program Verification.”. In: *IJCAR*. Ed. by Ulrich Furbach and Natarajan Shankar. Vol. 4130. LNCS. ISSN: 0302-9743. Springer, 2006, pp. 266–280. ISBN: 3-540-37187-7. DOI: `10.1007/11814771_23`.

[3] Rose Bohrer. “Differential Dynamic Logic”. In: *Archive of Formal Proofs* (Feb. 2017). `https://isa-afp.org/entries/Differential_Dynamic_Logic.html`, Formal proof development. ISSN: 2150-914x.

References II

[4] Ádám Darvas, Reiner Hähnle, and David Sands. “A Theorem Proving Approach to Analysis of Secure Information Flow”. In: *Security in Pervasive Computing, Second International Conference, SPC 2005, Boppard, Germany, April 6-8, 2005, Proceedings*. Ed. by Dieter Hutter and Markus Ullmann. Lecture Notes in Computer Science. Springer, 2005, pp. 193–209. DOI: `10.1007/978-3-540-32004-3\_20`. URL: `https://doi.org/10.1007/978-3-540-32004-3%5C_20`.

[5] Daniel Drodt and Reiner Hähnle. “RustyDL: A Program Logic for Rust”. In: *Formal Methods - 27th International Symposium, FM 2026, Tokyo, Japan, May 18-22, 2026, Proceedings, Part I*. Ed. by Augusto Sampaio and Marielle Stoelinga. Lecture Notes in Computer Science. Springer, 2026, pp. 357–377. DOI: `10.1007/978-3-032-26204-2\_19`. URL: `https://doi.org/10.1007/978-3-032-26204-2%5C_19`.

[6] Michael J. Fischer and Richard E. Ladner. “Propositional Modal Logic of Programs (Extended Abstract)”. In: *Proceedings of the 9th Annual ACM Symposium on Theory of Computing, May 4-6, 1977, Boulder, Colorado, USA*. Ed. by John E. Hopcroft, Emily P. Friedman, and Michael A. Harrison. ACM, 1977, pp. 286–294. DOI: `10.1145/800105.803418`. URL: `https://doi.org/10.1145/800105.803418`.

References III

[7] Jonas Klamroth et al. "QIn: Enabling Formal Methods to Deal with Quantum Circuits". In: *IEEE International Conference on Quantum Software, QSW 2023, Chicago, IL, USA, July 2-8, 2023*. Ed. by Shaukat Ali et al. IEEE, 2023, pp. 175–185. DOI: 10.1109/QSW59989.2023.00029.

[8] André Platzer. *A Complete Uniform Substitution Calculus for Differential Dynamic Logic*. Vol. 59. arXiv: 1601.06183 Publication Title: Journal of Automated Reasoning Issue: 2 ISSN: 15730670. Springer Netherlands, 2017. ISBN: 1081701693851. DOI: 10.1007/s10817-016-9385-1.

[9] André Platzer. "Differential Dynamic Logic for Hybrid Systems". In: *J. Autom. Reason.* 41.2 (2008), pp. 143–189. DOI: 10.1007/s10817-008-9103-8. URL: https://doi.org/10.1007/s10817-008-9103-8.

[10] André Platzer. "Differential Logic for Reasoning about Hybrid Systems". In: *HSCC*. Ed. by Alberto Bemporad, Antonio Bicchi, and Giorgio Buttazzo. Vol. 4416. LNCS. Springer, 2007, pp. 746–749. ISBN: 978-3-540-71492-7. DOI: 10.1007/978-3-540-71493-4_75.

References IV

[11] Gerhard Schellhorn and Wolfgang Ahrendt. “Reasoning about Abstract State Machines: The WAM Case Study”. In: *J. Univers. Comput. Sci.* 3.4 (1997), pp. 377–413. DOI: `10.3217/JUCS-003-04-0377`. URL: `https://doi.org/10.3217/jucs-003-04-0377`.

[NeurIPS’24] Samuel Teuber, Stefan Mitsch, and André Platzer. “Provably Safe Neural Network Controllers via Differential Dynamic Logic”. In: *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems*. NeurIPS 2024 (Vancouver, BC, Canada, Dec. 10–15, 2024). Ed. by Amir Globersons et al. 2024.

[12] Mattias Ulbrich. “Dynamic Logic for an Intermediate Language: Verification, Interaction and Refinement”. In: (2013).

Backup

The Many Uses of Dynamic Logic

There exist a **multitude** of dynamic logics for verification of systems:

The Many Uses of Dynamic Logic

There exist a **multitude** of dynamic logics for verification of systems:

- Abstract State Machines — Schellhorn et al. 1997
- JavaDL — Beckert et al. 2006
- Differential Dynamic Logic — Platzer 2007
- Unstructured Dynamic Logic — Ulbrich 2013
- RustyDL — Drodt et al. 2026

The Many Uses of Dynamic Logic

There exist a **multitude** of dynamic logics for verification of systems:

- Abstract State Machines — Schellhorn et al. 1997
- JavaDL — Beckert et al. 2006
- Differential Dynamic Logic — Platzer 2007
- Unstructured Dynamic Logic — Ulbrich 2013
- RustyDL — Drodt et al. 2026

Many Properties can be formalized as Dynamic Logic formula

$\phi \rightarrow [\alpha]\,\psi$	$\phi \rightarrow \langle\alpha\rangle\psi$
Safety	Liveness

The Many Uses of Dynamic Logic

There exist a **multitude** of dynamic logics for verification of systems:

- Abstract State Machines — Schellhorn et al. 1997
- JavaDL — Beckert et al. 2006
- Differential Dynamic Logic — Platzer 2007
- Unstructured Dynamic Logic — Ulbrich 2013
- RustyDL — Drodt et al. 2026

Many Properties can be formalized as Dynamic Logic formula

$\phi \to [\alpha]\,\psi$
Safety

$\phi \to \langle\alpha\rangle\psi$
Liveness

$\phi \to [\alpha]\,\langle\beta\rangle\mathrm{coupling}$
Refinement
Ulbrich 2013

$\forall \mathrm{low}\ \exists \mathrm{res}\ \forall \mathrm{high}\ \langle\alpha\rangle\ (\mathrm{low} \doteq \mathrm{res})$
Information Flow
Darvas et al. 2005

Simulation of validity for $\Lambda^{(i)}$-formulas

Corollary

Let $\hat{\Delta}$ be a Heterogeneous Dynamic Theory over Dynamic Theories $\Delta^{(0)}, \Delta^{(1)}$ and let P be a $\Lambda^{(i)}$-formula.
P is $\Delta^{(i)}$-valid iff P is $\hat{\Delta}$-valid.

This allows us to **reuse** the calculi for the individual logics!

Meta Properties of Dynamic Logics

(Pen and Paper Proofs Only)

$\mathit{Fin}(\Delta)$	Finite Support
$\mathit{FolExp}(\Delta)$	FOL Expressiveness
$\mathit{Göd}(\Delta)$	Gödel Expressiveness
$\mathit{FolRelCom}(\Delta)$	Relatively complete w.r.t. box and diamond formulas
$\mathit{RelCom}(\Delta)$	Relative Completeness
$\mathit{Co}(\Delta)$	Communicating heterogeneous dynamic theory *ability to pass integer-like values between theories*

Meta Properties of Dynamic Logics

(Pen and Paper Proofs Only)

$\mathit{Fin}(\Delta)$	Finite Support
$\mathit{FolExp}(\Delta)$	FOL Expressiveness
$\mathit{Göd}(\Delta)$	Gödel Expressiveness
$\mathit{FolRelCom}(\Delta)$	Relatively complete w.r.t. box and diamond formulas
$\mathit{RelCom}(\Delta)$	**Relative Completeness**
$\mathit{Co}(\Delta)$	Communicating heterogeneous dynamic theory *ability to pass integer-like values between theories*

A Calculus of Relative Completeness

(Pen and Paper Proofs Only)

By Definition:

$$\mathit{FolExp}(\Delta) \Longrightarrow \mathit{Fin}(\Delta)$$

Theorem B.1:

$$\mathit{FolExp}(\Delta) \land \mathit{FolRelCom}(\Delta) \Longrightarrow \mathit{RelCom}(\Delta)$$

Results for Havoc Lifting:

$\mathit{Fin}(\Delta) \Longrightarrow \mathit{Fin}(\Delta^{:=*})$	Lemma B.5
$\mathit{Göd}(\Delta) \Longrightarrow \mathit{Göd}(\Delta^{:=*})$	Lemma B.5
$\mathit{FolExp}(\Delta) \Longrightarrow \mathit{FolExp}(\Delta^{:=*})$	Lemma B.6
$\mathit{RelCom}(\Delta) \Longrightarrow \mathit{FolRelCom}(\Delta^{:=*})$	Lemma B.7

Results for Regular Program Lifting:

$\mathit{Fin}(\Delta) \Longrightarrow \mathit{Fin}(\Delta^{\mathrm{reg}})$	Lemma B.9
$\mathit{Göd}(\Delta) \Longrightarrow \mathit{Göd}(\Delta^{\mathrm{reg}})$	Lemma B.9
$\mathit{Göd}(\Delta) \land \mathit{FolExp}(\Delta) \Longrightarrow \mathit{FolExp}(\Delta^{\mathrm{reg}})$	Lemma B.10
$\mathit{Göd}(\Delta) \land \mathit{FolExp}(\Delta) \land \mathit{RelCom}(\Delta) \Longrightarrow \mathit{FolRelCom}(\Delta^{\mathrm{reg}})$	Lemma B.11, B.12

Results for Heterogeneous Dynamic Theories:

$\mathit{Fin}\left(\Delta^{(0)}\right) \land \mathit{Fin}\left(\Delta^{(1)}\right) \Longrightarrow \mathit{Fin}\left(\Delta^{(01)}\right)$	Lemma B.4
$\mathit{Göd}\left(\Delta^{(0)}\right) \land \mathit{Göd}\left(\Delta^{(1)}\right) \land \mathit{Co}\left(\Delta^{(01)}\right) \Longrightarrow \mathit{Göd}\left(\Delta^{(01)}\right)$	Lemma B.13
$\mathit{FolExp}\left(\Delta^{(0)}\right) \land \mathit{FolExp}\left(\Delta^{(1)}\right) \land \mathit{Co}\left(\Delta^{(01)}\right) \Longrightarrow \mathit{FolExp}\left(\Delta^{(01)}\right)$	Lemma B.14
$\left(\begin{array}{l} \mathit{FolExp}(\Delta^{(0)}) \land \mathit{RelCom}(\Delta^{(0)}) \land \\ \mathit{FolExp}(\Delta^{(1)}) \land \mathit{RelCom}(\Delta^{(1)}) \land \mathit{Co}(\Delta^{(01)}) \end{array}\right) \Longrightarrow \mathit{FolRelCom}\left(\Delta^{(01)}\right)$	Lemma B.15, B.16

Key Result

Relative completeness **preserved** by lifting/combination (under reasonable assumptions)