Modal Reasoning = Metric Reasoning, via Lawvere

Francesco Gavazzo (joint work with Ugo Dal Lago)

University of Bologna & INRIA Sophia Antipolis
A Long Introduction
Why This Talk?

**Extensional properties of programs**

- Does the program terminate?
- Does the program raise errors?
- What the program computes
Why This Talk?

**Extensional properties of programs**

- Does the program **terminate**?
- Does the program raise **errors**?
- **What** the program computes

**Programs as black-boxes**

- Relations between **input-output**
- Do not care **how** output is produced
- Same IO behaviour implies **equivalent** programs
Why This Talk?

**Extensional properties of programs**
- Does the program terminate?
- Does the program raise errors?
- **What** the program computes

Programs as **black-boxes**
- Relations between input-output
- Do not care how output is produced
- Same IO behaviour implies equivalent programs

**Mathematical foundation**
- Type theory
- Denotational Semantics
- Program equivalence
Intensional properties of programs

Focus on how programs compute

Is the program efficient?
Is the program secure?
Is the program robust wrt variations in the input?
Example 1: Information-Flow

Q. Is $f$ secure?
Example 1: Information-Flow

Q. Is $f$ secure?

secure = classified information cannot flow-out of programs
**Example 1: Information-Flow**

Q. Is $f$ secure?

*secure* = classified information cannot **flow-out** of programs

Example

```
key \geq 0
```

- If $key \geq 0$, then the key is not secure.
- If $key < 0$, then the key is secure.

key: 0 (true) and 1 (false)
Example 2: Program Sensitivity

Q. Is \( f \) robust to \textit{variations} in the input?
Example 2: Program Sensitivity

Q. Is $f$ robust to variations in the input?

$k$-robustness (aka sensitivity) = errors in input are amplified at most of a factor $k$
Intensional Program Analysis

Q. How to guarantee intensional properties of programs?

Q. How to reason about programs intensionally?
Q. How to guarantee **intensional** properties of programs?

**Type Theory**

Information-flow key: \([\text{secret}]\tau\)  
Sensitivity: \(f: [k]\tau \rightarrow \sigma\)

Q. How to reason about programs **intensionally**?
**Intensional Program Analysis**

**Q.** How to guarantee **intensional** properties of programs?

**Type Theory**

- Information-flow key: $[secret]_\tau$
- Sensitivity: $f: [k]_\tau \rightarrow \sigma$

**Q.** How to reason about programs **intensionally**?

**Program Equivalence**
Goal: Identify programs with the same operational behaviour
Program Equivalence

**Goal:** Identify programs with the same operational behaviour

Applications in program **correctness, refactoring, and optimization**

- **HO Arithmetic**
  \[ \lambda x. \lambda f. f(x + 0) \simeq \lambda x. \lambda f. f(x) \]

- **Structural equivalences**
  \[ \left( \begin{array}{c}
  \text{let } x = a \\
  y = b \\
  \text{in } f(x)
  \end{array} \right) \simeq \text{let } x = a \text{ in } f(x) \]
Program Equivalence

Main feature: compositionality

\[ f \xrightarrow{\sim} g \]

\[ k \xrightarrow{\sim} \begin{array}{c} f \\ j \end{array} \xrightarrow{\sim} \begin{array}{c} g \\ j \end{array} \xrightarrow{\sim} e \]
Q. Program equivalence for intensional program analysis?
Q. Program equivalence for intensional program analysis?

**Non-Interference (Abadi et al., 1999)**

\[ \forall \text{key}_1, \text{key}_2 : [\text{secret}] \tau. \forall \text{public}. \]

An observer with **public permission** cannot infer whether the first input is \text{key}_1 or \text{key}_2.
Program Equivalence

Metric-preservation (Reed & Pierce, 2010)

\[ a \sim_\varepsilon b \implies a \xrightarrow{f} k \varepsilon b \xrightarrow{f} b \]

\textit{k-robust} programs preserve \texttt{approximate equivalence} up-to a factor \textit{k}
Summing-up

Two intensional analyses of programs

All analyses performed in languages with suitable type systems

Intensional properties of programs via program equivalence

<table>
<thead>
<tr>
<th>Security</th>
<th>Sensitivity</th>
</tr>
</thead>
<tbody>
<tr>
<td>[secret]_τ</td>
<td>[k]_τ</td>
</tr>
<tr>
<td>Non-interference</td>
<td>Metric-preservation</td>
</tr>
</tbody>
</table>
Two intensional analyses of programs

All analyses performed in languages with suitable type systems

Intensional properties of programs via program equivalence

Security
[secret]τ
Non-interference

Sensitivity
[k]τ
Metric-preservation

Further examples: dead-code analysis, strictness analysis, resource/usage analysis, ...
Q. Can we give a uniform account of all these phenomena?
Q. Can we give a uniform account of all these phenomena?

Type Systems

Graded modal types (Orchard et al., 2019; Gaboardi et al., 2016)

Program Equivalence

This talk
Intensional vs Extensional PE

Extensional Program equivalence
Programs are equivalent for any observer
Intensional vs Extensional PE

**Extensional Program equivalence**
Programs are equivalent for *any* observer

\[ e \simeq e' \]

**Intensional Program equivalence**
Programs are equivalent wrt observers’ features

\[ e \simeq e' \]
\[ e \simeq e' \]
\[ e \simeq e' \]
\[ \varepsilon \]
Intensional vs Extensional PE

Intensional Program equivalence
Programs are equivalent wrt observers’ features

\[\text{key}_1 \sim \text{key}_2 : [\text{secret}]_{\tau}\]

- Depend on the observer’s permission
- Public permission \(\implies\) \(\text{key}_1 \sim \text{key}_2\)
- Secret permission \(\implies\) \(\text{key}_1 \not\sim \text{key}_2\)
Program equivalence for graded modal types

\[ w \models e \approx e' \]

- Eq. wrt possible worlds
- Intensionality as in logic

Metric Reasoning
- Intensional equivalence
- Program distance

Abstract compositionality
- Non-interference
- Metric Preservation
Related Work
## Related Work

### Bounded Exponentials
- (Girard et al., 1992)
- **Resource-usage** (M. Hofmann, 1999)
- **Complexity** (Lago & Hofmann, 2009)
- **Sensitivity** (Reed & Pierce, 2010)

### Information-flow
- (Abadi et al., 1999)
- (Volpano et al., 1996)
Related Work

**Bounded Exponentials**
- (Girard et al., 1992)
- Resource-usage (M. Hofmann, 1999)
- Complexity (Lago & Hofmann, 2009)
- Sensitivity (Reed & Pierce, 2010)

**Information-flow**
- (Abadi et al., 1999)
- (Volpano et al., 1996)

**Graded and Quantitative Types**
- (Wood & Atkey, 2020)
- (Ghica & Smith, 2014)
- (Atkey, 2018)

**Coeffects**
- (Petricek et al., 2014)
- (Gaboardi et al., 2016)
- (Brunel et al., 2014)

How code can be manipulated
Related Work

**Graded Modal Types**
Modal types indexed by grades

Programming language **Granule** (Orchard et al., 2019)

**Graded (co)monadic denotational semantics** (Gaboardi et al., 2016)

**Logical relations** (Abel & Bernardy, 2020)
Modal Reasoning
Goal. Program equivalence for languages with \textit{graded modal types}

\textbf{Linearity} \rightarrow \textit{Data as resources}

\[ \forall \lambda x.(x,x) : \tau \rightarrow \tau \times \tau \quad \forall \lambda x.\lambda y.x : \tau \rightarrow \sigma \rightarrow \tau \]

\textbf{Modalities} \rightarrow \textit{Code manipulations}

\[ [\text{int}]\tau \quad \text{code can be copied and erased} \]
\[ [k]\tau \quad \text{code can be used } k\text{-times} \]
\[ [\text{secret}]\tau \quad \text{code cannot contain unclassified info} \]
Graded Modal Types

Types \( \tau ::= \ldots \mid \tau \rightarrow \tau \mid [j] \tau \)

Values \( a ::= \ldots \mid \text{box } a \)

Expressions \( e ::= \ldots \mid \text{let box } x = a \text{ in } e \)

Grade algebra \((J, \leq, +, *, 0, 1, \infty)\)

S4 modality \([j] \tau\)

Example

<table>
<thead>
<tr>
<th>Resource Usage</th>
<th>Security</th>
<th>Sensitivity</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>secret</td>
<td>([(0, \infty], \leq, +, *, 0, 1))</td>
</tr>
<tr>
<td>aff</td>
<td>public</td>
<td></td>
</tr>
<tr>
<td>dead</td>
<td>lin</td>
<td></td>
</tr>
</tbody>
</table>
Graded Modal Types

Types
\[ \tau ::= \ldots | \tau \rightarrow \tau | [j] \tau \]

Values
\[ a ::= \ldots | \text{box } a \]

Expressions
\[ e ::= \ldots | \text{let box } x = a \text{ in } e \]

Grade algebra
\[ (\mathcal{J}, \leq, +, *, 0, 1, \infty) \]

S4 modality
\[ [j] \tau \]

Graded Judgements
\[ x_1 : j_1 \tau_1, \ldots, x_n : j_n \tau_n \vdash e : \tau \]

\( e \) manipulates \( x_i \) according to \( j_i \)
Program Equivalence
Goal. Identify programs with the same operational and intensional behaviour
**Program Equivalence**

**Goal.** Identify programs with the same operational and intensional behaviour

**Operational Semantics**

Programs evaluate to values: $e \Downarrow a$

$$(\lambda x.e)a \mapsto e[x := a]$$

$$\vdots$$

let box $x = (\text{box } a)$ in $e \mapsto e[x := a]$
**Program Equivalence**

**Goal.** identify programs with the same operational and intensional behaviour

**Q.** How to capture intensionality?
**Program Equivalence**

**Goal.** Identify programs with the same operational and intensional behaviour.

**Q.** How to capture intensionality?

<table>
<thead>
<tr>
<th>Extensional PE</th>
<th>Intensional PE</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e R e'$</td>
<td>$w \models e R e'$</td>
</tr>
<tr>
<td>$R \subseteq \text{Exp} \times \text{Exp}$</td>
<td>$R : W \rightarrow \mathcal{P}(\text{Exp} \times \text{Exp})$</td>
</tr>
<tr>
<td>Relations</td>
<td>Relations over possible worlds</td>
</tr>
</tbody>
</table>
Program Equivalence

Goal. identify programs with the same operational and intensional behaviour

Q. How to capture intensionality?

- **Extensional PE**
  - $e R e'$
  - $R \subseteq \text{Exp} \times \text{Exp}$
  - Relations

- **Intensional PE**
  - $w \vdash e R e'$
  - $R : W \rightarrow \mathcal{P} (\text{Exp} \times \text{Exp})$
  - Relations over possible worlds

**Possible worlds** = Monoidal preorder $(W, \leq, \bullet, \varepsilon)$

Semantics substructural logic (Urquhart, 1972; Routley & Meyer, 1973)
**Categories of Relations**

**Category $W$-Rel**

- Objects: $X, Y, \ldots$
- Arrows: $R : (W, \leq) \rightarrow (\mathcal{P}(X \times Y), \subseteq)$
Goal. Define notions of equivalence

• Contextual/CIU equivalence
• Logical relations (Abel & Bernardy, 2020)
• Applicative bisimilarity
Applicative Bisimilarity

**Goal.** Define notions of equivalence
- Contextual/CIU equivalence
- Logical relations (Abel & Bernardy, 2020)
- Applicative bisimilarity

**Applicative Bisimilarity (Abramsky, 1990)**

**Idea.** \( \lambda \)-terms are functions

\[
f = g \iff \forall x. f(x) = g(x)
\]

\[
\lambda x. e \simeq \lambda x. e' \iff \forall a. e[x := a] \simeq e'[x := a]
\]

**Solution.** Coinduction
Applicative Bisimilarity

Applicative Bisimilarity (Abramsky, 1990)
The largest symmetric $R \subseteq \text{Exp} \times \text{Exp}$ s.t.

\[

e \, R \, e' \quad \text{and} \quad e \Downarrow a \quad \Rightarrow \quad e' \Downarrow a' \quad \text{and} \quad a \, R \, a'
\]

\[

\lambda x.e \, R \, \lambda x.e' \quad \Rightarrow \quad \forall a. \, e[x := a] \, R \, e'[x := a]
\]
Applicative Bisimilarity

Applicative Bisimilarity (Abramsky, 1990)
The largest symmetric \( R \subseteq \text{Exp} \times \text{Exp} \) s.t.

\[
e R e' \text{ and } e \Downarrow a \implies e' \Downarrow a' \text{ and } a R a' \\
\lambda x. e R \lambda x. e' \implies \forall a. e[x := a] R e'[x := a]
\]

Modal Applicative Bisimilarity
The largest symmetric \( W \)-relation \( R \) s.t.

\[
w \models e R e' \text{ and } e \Downarrow a \implies e' \Downarrow a' \text{ and } w \models a R a' \\
w \models \lambda x. e R \lambda x. e' \implies \forall a. w \models e[x := a] R e'[x := a] \\
w \models \text{box } a R \text{ box } a' \implies ???
\]
Idea. Modal types act on possible worlds
Modal Applicative Bisimilarity

Idea. Modal types act on possible worlds

Sensitivity

\[ \varepsilon \vdash \text{box } a \simeq \text{box } a' : [k]\tau \iff \exists \delta. \varepsilon \succeq k\delta. \text{ and } \delta \vdash a \simeq a' : \tau \]
Modal Applicative Bisimilarity

**Idea.** Modal types act on possible worlds

**Sensitivity**
\[ \varepsilon \vdash \text{box } a \simeq \text{box } a' : [k]\tau \iff \exists \delta. \varepsilon \geq k\delta. \text{ and } \delta \vdash a \simeq a' : \tau \]

**Security**
- **public** \[ \vdash \text{box } a \simeq \text{box } a' : [\text{secret}]\tau \iff \text{always} \]
- **secret** \[ \vdash \text{box } a \simeq \text{box } a' : [\text{secret}]\tau \iff \text{secret} \vdash a \simeq a' : \tau \]
Modal Applicative Bisimilarity

**Idea.** Modal types act on possible worlds

**Sensitivity**

$$\varepsilon \models \text{box } a \simeq \text{box } a' : [k]\tau \iff \exists \delta. \varepsilon \geq k\delta. \text{ and } \delta \models a \simeq a' : \tau$$

**Security**

- **public** \(\models\) box \(a \simeq a' : [\text{secret}]\tau \iff \text{always}
- **secret** \(\models\) box \(a \simeq a' : [\text{secret}]\tau \iff \text{secret } \models a \simeq a' : \tau

**Q.** How do we generalise these constructions?
Relation Lifting

\[ \tau \xrightarrow{PE} \sim_{\tau} \]
Relation Lifting

\[
\begin{array}{c}
\tau \\
F \\
F(\tau)
\end{array}
\xrightarrow{PE}
\xrightarrow{\sim_\tau}
\]
Relation Lifting

\[ \tau \xrightarrow{PE} \sim_{\tau} \]

\[ F \xrightarrow{PE} F(\tau) \xrightarrow{F^\#} F^\#(\sim_{\tau}) = \sim_{F(\tau)} \]

Moral. Need ways to **extend** constructions on types/sets to relations.
Lax Extension (Barr, 1970; Thijs, 1996)

A lax extension of $F : \text{Set} \rightarrow \text{Set}$, is a mapping $\Gamma : \mathcal{W}-\text{Rel}(X, Y) \rightarrow \mathcal{W}-\text{Rel}(F(X), F(Y))$ s.t.

- $\Gamma(R) ; \Gamma(S) \subseteq \Gamma(R; S)$
- $F(f) \subseteq \Gamma(f)$
- $F(f)^T \subseteq \Gamma(f^T)$
- $R \subseteq S \implies \Gamma(R) \subseteq \Gamma(S)$

Functor

\[
\begin{array}{c}
\text{Set} \\
\downarrow F \\
\text{Set}
\end{array}
\]

Lax Functor

\[
\begin{array}{c}
\mathcal{W}-\text{Rel} \\
\downarrow \Gamma^\#(F) \\
\mathcal{W}-\text{Rel}
\end{array}
\]
Q. What about modal types $[j] \tau$?
Q. What about modal types $[j] \tau$?

Modal types $= \text{Graded comonadic lax extension of the identity comonad}$

- Graded comonadic $= \text{graded S4 modalities}$
- Identity $= \text{act on possible worlds only}$
Relation Lifting

Graded Comonadic Lax Extension

A graded comonadic lax extension is a \( J \)-indexed family of lax extensions \( \Delta_j : W\text{-Rel}(X, Y) \to W\text{-Rel}(X, Y) \) antitone in \( J \) s.t.

\[
\Delta_1(R) \subseteq R \\
\Delta_{j \star k}(R) \subseteq \Delta_j(\Delta_k(R)) \\
\Delta_j(R) \otimes \Delta_j(S) \subseteq \Delta_j(R \otimes S) \\
\Delta_{j+k}(R) \subseteq \text{dup}^T; (\Delta_j(R) \otimes \Delta_j(S)); \text{dup}
\]

Identity Comonad

\[
\begin{bmatrix}
\text{Set} \\
\downarrow \text{ID} \\
\text{Set}
\end{bmatrix}
\]

Graded Lax Monoidal Comonad

\[
\begin{bmatrix}
W\text{-Rel} \\
\downarrow \Delta^#(\text{ID}) \\
W\text{-Rel}
\end{bmatrix}
\]
Modal Applicative Bisimilarity

The **largest** symmetric \( W \)-relation \( R \) s.t.

\[
    w \models e \mathrel{R} e' : \tau \text{ and } e \Downarrow a \quad \Rightarrow \quad e' \Downarrow a' \text{ and } w \models a \mathrel{R} a' : \tau
\]

\[
    w \models f \mathrel{R} f' : \tau \rightarrow \tau' \quad \Rightarrow \quad \forall a. \ w \models fa \mathrel{R} f'a : \tau'
\]

\[
    w \models \text{box } a \mathrel{R} \text{box } a' : [j] \tau \quad \Rightarrow \quad w \models a \Delta_j(R) a' : \tau
\]
Modal Applicative Bisimilarity

The **largest** symmetric $W$-relation $R$ s.t.

\[
\begin{align*}
\text{let } w &\vdash e \; R \; e' : \tau \text{ and } e \Downarrow a \implies e' \Downarrow a' \text{ and } w \vdash a \; R \; a' : \tau \\
w &\vdash f \; R \; f' : \tau \implies f a \; R \; f' a : \tau' \\
w &\vdash \text{box } a \; R \; \text{box } a' : [j] \tau \implies w \vdash a \; \Delta_j(R) \; a' : \tau
\end{align*}
\]

**Theorem (Compositionality)**

Modal applicative bisimilarity is **compositional**

\[
\begin{align*}
\text{let } &x : j \; \tau \vdash e, e' : \tau' \quad \nu \vdash e \simeq e' \quad w \vdash a \; \Delta_j(\simeq) \; a' : \tau \\
w \bullet \nu &\vdash e[x := a] \simeq e'[x := a'] : \tau'
\end{align*}
\]
Metric Preservation (Reed & Pierce, 2010)

\[ J = [0, \infty] = W \]

\[
\frac{x :k \tau \vdash f : \tau' \quad \varepsilon \vdash a \simeq a' : \tau}{k\varepsilon \vdash f[x := a] \simeq f[x := a'] : \tau'}
\]

Non-Interference (Abadi et al., 1999)

\[ J = \{public \leq secret\} = W \]

\[
x :secret \tau \vdash f : \tau' \quad \text{&} \quad a, a' : \tau \Rightarrow public \vdash f[x := a] \simeq f[x := a'] : \tau'
\]
Modal Reasoning = Metric Reasoning, via Lawvere
Program Distance

**Relations**

\[ e \simeq e' \]

**Equivalence**

**W-relations**

\[ w \models e \simeq e' \]

**Intensional equivalence**
## Program Distance

<table>
<thead>
<tr>
<th>Relations</th>
<th>$\mathcal{W}$-relations</th>
<th>Distances</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e \simeq e'$</td>
<td>$w \vdash e \simeq e'$</td>
<td>$\delta(e, e') = \varepsilon$</td>
</tr>
<tr>
<td><strong>Equivalence</strong></td>
<td><strong>Intensional equivalence</strong></td>
<td><strong>Pseudometric</strong></td>
</tr>
</tbody>
</table>
Program Distance

<table>
<thead>
<tr>
<th>Relations</th>
<th>$\mathcal{W}$-relations</th>
<th>Distances</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e \simeq e'$</td>
<td>$\mathcal{W} \models e \simeq e'$</td>
<td>$\delta(e, e') = \varepsilon$</td>
</tr>
</tbody>
</table>

Equivalence      Intensional equivalence  Pseudometric

**Goal.** Intensional equivalence $\equiv$ Program distance

- Solid theory of program distance
- Combined effects and coeffects
From Equivalences to Distances

<table>
<thead>
<tr>
<th>Relations</th>
<th>Distances</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e \sim e'$</td>
<td>$\delta(e, e') = \varepsilon$</td>
</tr>
<tr>
<td>${false, true}</td>
<td>[0, \infty]</td>
</tr>
</tbody>
</table>

Equivalence  Pseudometric
From Equivalences to Distances

<table>
<thead>
<tr>
<th>Relations</th>
<th>Distances</th>
<th>Quantale-relations</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e \simeq e'$</td>
<td>$\delta(e, e') = \varepsilon$</td>
<td>$\delta(e, e') = v$</td>
</tr>
<tr>
<td>${false, true}$</td>
<td>$[0, \infty]$</td>
<td>$(V, \leq, \otimes, k)$</td>
</tr>
<tr>
<td>Equivalence</td>
<td>Pseudometric</td>
<td>$V$-equivalences</td>
</tr>
</tbody>
</table>

(Generalised) metric spaces as enriched categories (Lawvere, 1973)

Quantale (Rosenthal, 1990)
A complete lattice $(V, \leq)$ with a monoid structure $(V, \otimes, k)$

$$v \otimes \bigvee_i u_i = \bigvee_i (v \otimes u_i)$$
$$\bigvee_i v_i \otimes u = \bigvee_i (v_i \otimes u)$$
### Quantale-relations

#### Example

<table>
<thead>
<tr>
<th>Boolean</th>
<th>Lawvere</th>
<th>Strong Lawvere</th>
</tr>
</thead>
<tbody>
<tr>
<td>({false, true}, \leq, \land, \top)</td>
<td>([0, \infty], \geq, +, 0)</td>
<td>([0, \infty], \geq, \text{max}, 0)</td>
</tr>
</tbody>
</table>

- 3-element chain: \{\bot, k, \top\}
- Powerset: \mathcal{P}(X)
- Left cont. distributions: \(f : [0, \infty] \rightarrow [0, 1]\)

#### Example

- Monotone \(W\)-predicates: \(p : (W, \leq, \bullet, \varepsilon) \rightarrow (2, \leq)\)
### Category $V$-Rel

- **Objects:** $X, Y, \ldots$
- **Arrows:** $\alpha : X \times Y \rightarrow V$

### Identity

\[
l(x, x) = k, \quad l(x, y) = \bot
\]

### Composition

\[
(\alpha; \beta)(x, z) = \bigvee_y \alpha(x, y) \otimes \beta(y, z)
\]
Quantale-relations

Category $V$-Rel

- Objects: $X, Y, \ldots$
- Arrows: $\alpha : X \times Y \to V$

Identity

$I(x, x) = k, I(x, y) = \bot$

Composition

$(\alpha; \beta)(x, z) = \bigvee_y \alpha(x, y) \otimes \beta(y, z)$

$(\alpha; \alpha)(x, z) \leq \alpha(x, z) \iff \inf_y \alpha(x, y) + \alpha(y, z) \geq \alpha(x, z) \iff \text{TI}$

- Boolean
- Transitivity
- Equivalence

- Lawvere
- Triangle Inequality

- Strong Lawvere
- Strong TI
- Ultra Pseudometric
Bisimilarity Distance

Rich literature on V-distances

- Monoidal topology (D. Hofmann et al., 2014)
- Effectful applicative bisimilarity (Gavazzo, 2018)
Bisimilarity Distance

Rich literature on V-distances

- Monoidal topology (D. Hofmann et al., 2014)
- Effectful applicative bisimilarity (Gavazzo, 2018)

Bisimilarity Distance $\delta$

The largest V-relation $\alpha$ s.t.

\[
\alpha_{\tau \rightarrow \tau'}(\lambda x.f, \lambda x.f') \leq \bigwedge_a \alpha_\tau(f[x := a], f'[x := a])
\]

\[
\alpha_{[j]_\tau}(\text{box } a, \text{box } a') \leq \Delta_j(\alpha_\tau)(a, a')
\]
Bisimilarity Distance

Comonadic Lax Extension

\[ \Delta_j : \text{V-Rel}(X, Y) \to \text{V-Rel}(X, Y) \]

Main Example

\[ V = \mathcal{J} = [0, \infty] \]

\[ \Delta_j(\alpha)(x, y) = j \cdot \alpha(x, y) \]

Non-expansive

\[
\begin{array}{c}
X \xrightarrow{f} Y \\
\alpha \downarrow \quad \leq \quad \downarrow \beta \\
X \xrightarrow{f} Y
\end{array}
\]

Lipschitz-continuous

\[
\begin{array}{c}
X \xrightarrow{f} Y \\
\Delta_j(\alpha) \downarrow \quad \leq \quad \downarrow \beta \\
X \xrightarrow{f} Y
\end{array}
\]
Theorem (Abstract Metric Preservation)

For $x : j \vdash e, e' : \tau'$ and $\vdash a, a' : \tau$, we have:

$$\Delta_j(\delta)(a, a') \otimes \delta(e, e') \leq \delta(e[x := a], e'[x := a'])$$

Theorem

For $V = (W, \leq, \bullet, \varepsilon) \rightarrow (2, \leq)$,

AMP $\implies$ Compositionality
Conclusion
Summing Up

**Intensional program equivalence for graded modal types**

**Compositionality theorem for modal applicative bisimilarity**

**Same results for other equivalences**

🔗 **Böhm tree-like equivalences**

\[ w \vdash BT(e) \equiv BT(e') \]

**Intensional equality as program distance**

🔗 **Abstract Metric Preservation**
What do we gain from AMP?
What do we gain from AMP? Combined effects and coeffects
Summing Up

What do we gain from AMP? **Combined effects and coeffects**

Add *algebraic operations* (random, print, lookup, . . .) and *monads*

\[(\ell := 2; !\ell + 3) \oplus_\frac{1}{3} (\ell := 3; !\ell - 1)\]
What do we gain from AMP? **Combined effects and coefficients**

Add **algebraic operations** (random, print, lookup, ...) and **monads**

\[(\ell := 2; !\ell + 3) \oplus \frac{1}{3} (\ell := 3; !\ell - 1)\]

Bisimilarity distance using **monadic lax extension**

**Monadic Lax Extension**
\[
\Gamma : V\text{-Rel}(X, Y) \rightarrow V\text{-Rel}(T(X), T(Y))
\]

**Lax distributive law**
\[
\Delta_r \circ \Gamma \subseteq \Gamma \circ \Delta_r
\]

**Abstract metric preservation theorem**


