Modal Reasoning = Metric Reasoning, via Lawvere

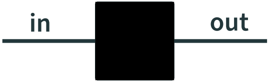
Francesco Gavazzo (joint work with Ugo Dal Lago)

University of Bologna & INRIA Sophia Antipolis

A Long Introduction

Why This Talk?

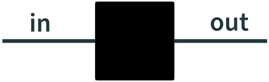
Extensional properties of programs



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

Why This Talk?

Extensional properties of programs



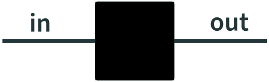
- Does the program terminate?
- Does the program raise errors?
- What the program computes

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**

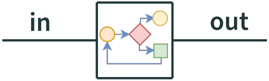
- Relations between input-output
- Do not care how output is produced
- Same IO behaviour implies equivalent programs

Mathematical foundation

- Type theory
- Denotational Semantics
- Program equivalence

Why This Talk?

Intensional properties of programs



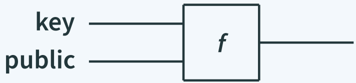
Focus on how programs compute

Is the program efficient?

Is the program secure?

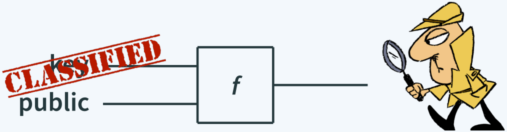
Is the program robust wrt variations in the input?

Example 1: Information-Flow



Q. Is *f* **secure**?

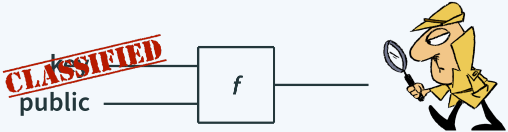
Example 1: Information-Flow



Q. Is *f* **secure**?

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

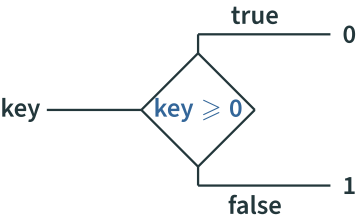
Example 1: Information-Flow



Q. Is *f* **secure**?

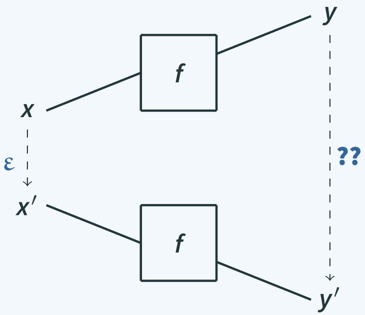
secure = classified information cannot **flow-out** of programs

Example



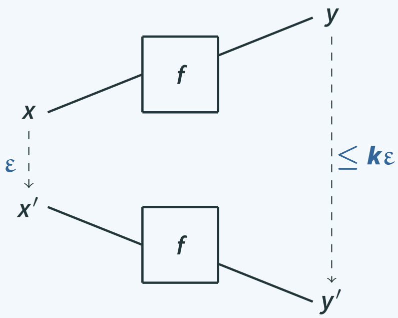
is **not** secure

Example 2: Program Sensitivity



Q. Is *f* **robust** to **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?

Intensional Program Analysis

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

Type Theory

Information-flow	Sensitivity
key: $[\text{secret}]\tau$	f: $[k]\tau \to \sigma$

Q. How to reason about programs intensionally?

Intensional Program Analysis

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

Type Theory

Information-flow	Sensitivity
key: $[\text{secret}]\tau$	f: $[k]\tau \to \sigma$

Q. How to reason about programs intensionally?

☞ **Program Equivalence**

Program Equivalence

Goal: Identify programs with the same operational behaviour

Goal: Identify programs with the same operational behaviour

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

- HO Arithmetic

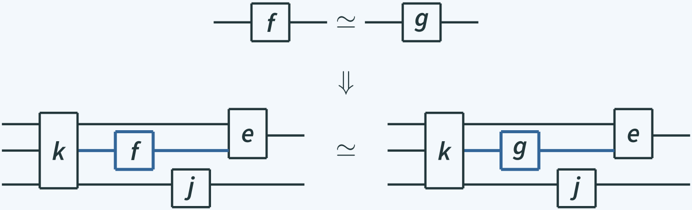
$$\lambda x.\lambda f.f(x + 0) \simeq \lambda x.\lambda f.f(x)$$

- Structural equivalences

$$\begin{pmatrix} \text{let} & x = a \\ & y = b \\ \text{in} & f(x) \end{pmatrix} \simeq \text{let } x = a \text{ in } f(x)$$

Program Equivalence

Main feature: compositionality



Program Equivalence

Q. Program equivalence for intensional program analysis?

Q. Program equivalence for intensional program analysis?

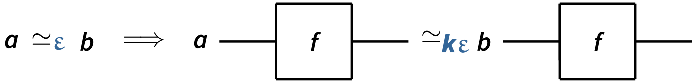
Non-Interference (Abadi et al., 1999)

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

An observer with **public permission** cannot infer whether the first input is key_1 or key_2

Program Equivalence

Metric-preservation (Reed & Pierce, 2010)



***k*-robust** programs preserve **approximate equivalence** up-to a factor *k*

Summing-up

Two intensional analyses of programs

All analyses performed in languages with suitable type systems

Intensional properties of programs via program equivalence

Security	Sensitivity
$[\text{secret}]\tau$	$[k]\tau$
Non-interference	Metric-preservation

Summing-up

Two intensional analyses of programs

All analyses performed in languages with suitable type systems

Intensional properties of programs via program equivalence

Security	Sensitivity
$[\text{secret}]\tau$	$[k]\tau$
Non-interference	Metric-preservation

Further examples: dead-code analysis, strictness analysis, resource/usage analysis, . . .

Summing-up

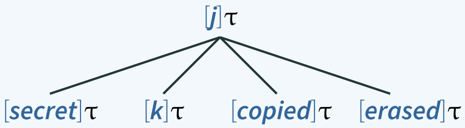
Q. Can we give a **uniform account** of all these phenomena?

Summing-up

Q. Can we give a **uniform account** of all these phenomena?

Type Systems

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



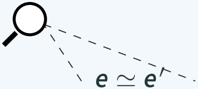
Program Equivalence

This talk

Intensional vs Extensional PE

Extensional Program equivalence

Programs are equivalent for any observer

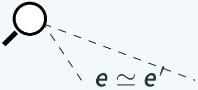


$e \simeq e'$

Intensional vs Extensional PE

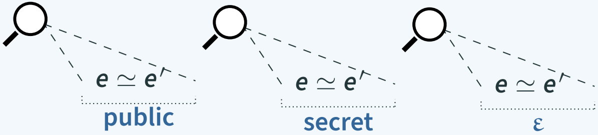
Extensional Program equivalence

Programs are equivalent for any observer



Intensional Program equivalence

Programs are equivalent wrt observers' features



Intensional vs Extensional PE

Intensional Program equivalence

Programs are equivalent wrt observers' features

$\text{key}_1 \simeq \text{key}_2 : [\text{secret}]\tau$

- depend on the observer's permission
- public permission $\Longrightarrow$ $\text{key}_1 \simeq \text{key}_2$
- secret permission $\Longrightarrow$ $\text{key}_1 \not\simeq \text{key}_2$

This Talk

Program equivalence for **graded modal types**

$w \Vdash e \simeq e'$

- Eq. wrt possible worlds
- Intensionality as in logic

Metric Reasoning

Intensional equivalence

$\equiv$

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)
- Sensitivty (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)
- Sensitivty (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

Graded Modal Types

Goal. Program equivalence for languages with **graded modal types**

Linearity $\rightarrow$ Data as resources

$$\nvdash \lambda x.(x, x) : \tau \rightarrow \tau \times \tau \qquad \nvdash \lambda x.\lambda y.x : \tau \rightarrow \sigma \rightarrow \tau$$

Modalities $\rightarrow$ Code manipulations

$[\mathbf{int}]\tau$	code can be copied and erased
$[k]\tau$	code can be used k-times
$[\mathbf{secret}]\tau$	code cannot contains unclassified info

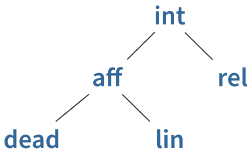
Graded Modal Types

Types	$\tau ::= \ldots \mid \tau \multimap \tau \mid [j]\tau$
Values	$a ::= \ldots \mid \mathsf{box}\ a$
Expressions	$e ::= \ldots \mid \mathsf{let\ box}\ x = a\ \mathsf{in}\ e$

Grade algebra	S4 modality
$(\mathcal{J}, \leqslant, +, *, \mathbf{0}, \mathbf{1}, \infty)$	$[j]\tau$

Example

Resource Usage



Security

Sensitivity

$([0, \infty], \leqslant, +, \cdot, 0, 1)$

Graded Modal Types

Types	$\tau ::= \ldots \mid \tau \multimap \tau \mid [j]\tau$
Values	$a ::= \ldots \mid \mathsf{box}\ a$
Expressions	$e ::= \ldots \mid \mathsf{let\ box}\ x = a\ \mathsf{in}\ e$

Grade algebra	S4 modality
$(\mathcal{J}, \leqslant, +, *, \mathbf{0}, \mathbf{1}, \infty)$	$[j]\tau$

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

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: $e \Downarrow a$

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

$$\vdots$$

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

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?

Extensional PE	**Intensional PE**
$e \mathrel{R} e'$	$w \Vdash e \mathrel{R} e'$
$R \subseteq Exp \times Exp$	$R : W \to \mathcal{P}(Exp \times Exp)$
Relations	Relations over possible worlds

Program Equivalence

Goal. identify programs with the same operational and intensional behaviour

Q. How to capture intensionality?

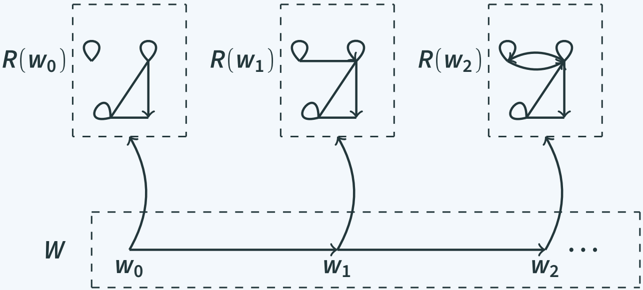
Extensional PE	**Intensional PE**
$e\ R\ e'$	$w \Vdash e\ R\ e'$
$R \subseteq Exp \times Exp$	$R : W \to \mathcal{P}(Exp \times Exp)$
Relations	Relations over possible worlds

Possible worlds $=$ Monoidal preoprder $(W, \leqslant, \bullet, \varepsilon)$

Semantics substructural logic (Urquhart, 1972; Routley & Meyer, 1973)

Category W-Rel

- Objects: $X, Y, \ldots$
- Arrows: $R : (W, \leqslant) \to (\mathcal{P}(X \times Y), \subseteq)$



Applicative Bisimilarity

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. λ-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 $R \subseteq Exp \times Exp$ s.t.

$$e \mathrel{R} e' \text{ and } e \Downarrow a \implies e' \Downarrow a' \text{ and } a \mathrel{R} a'$$

$$\lambda x.e \mathrel{R} \lambda x.e' \implies \forall a.\ e[x := a] \mathrel{R} e'[x := a]$$

Applicative Bisimilarity

Applicative Bisimilarity (Abramsky, 1990)

The **largest** symmetric $R \subseteq Exp \times Exp$ s.t.

$$
\begin{aligned}
e \mathrel{R} e' \text{ and } e \Downarrow a &\implies e' \Downarrow a' \text{ and } a \mathrel{R} a' \\
\lambda x.e \mathrel{R} \lambda x.e' &\implies \forall a.\ e[x := a] \mathrel{R} e'[x := a]
\end{aligned}
$$

Modal Applicative Bisimilarity

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

$$
\begin{aligned}
w \Vdash e \mathrel{R} e' \text{ and } e \Downarrow a &\implies e' \Downarrow a' \text{ and } w \Vdash a \mathrel{R} a' \\
w \Vdash \lambda x.e \mathrel{R} \lambda x.e' &\implies \forall a.\ w \Vdash e[x := a] \mathrel{R} e'[x := a] \\
w \Vdash \mathbf{box}\ a \mathrel{R} \mathbf{box}\ a' &\implies \textbf{???}
\end{aligned}
$$

Modal Applicative Bisimilarity

Idea. Modal types act on possible worlds

Modal Applicative Bisimilarity

Idea. Modal types act on possible worlds

Sensitivity

$\varepsilon \Vdash \mathsf{box}\, a \simeq \mathsf{box}\, a' : [k]\tau \iff \exists\delta.\ \varepsilon \geqslant k\delta.\ \text{and}\ \delta \Vdash a \simeq a' : \tau$

Modal Applicative Bisimilarity

Idea. Modal types act on possible worlds

Sensitivity

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

Security

$\mathit{public} \Vdash \text{box}\ a \simeq \text{box}\ a' : [\mathit{secret}]\tau \iff \text{always}$

$\mathit{secret} \Vdash \text{box}\ a \simeq \text{box}\ a' : [\mathit{secret}]\tau \iff \mathit{secret} \Vdash a \simeq a' : \tau$

Modal Applicative Bisimilarity

Idea. Modal types act on possible worlds

Sensitivity

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

Security

$\mathit{public} \Vdash \mathsf{box}\ a \simeq \mathsf{box}\ a' : [\mathit{secret}]\tau \iff \text{always}$

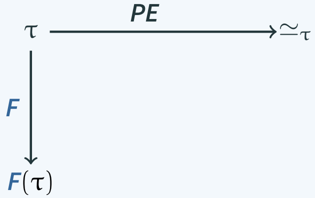
$\mathit{secret} \Vdash \mathsf{box}\ a \simeq \mathsf{box}\ a' : [\mathit{secret}]\tau \iff \mathit{secret} \Vdash a \simeq a' : \tau$

Q. How do we generalise these constructions?

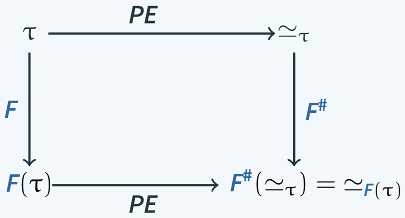
Relation Lifting

$$\tau \xrightarrow{\quad\quad PE \quad\quad} \simeq_\tau$$

Relation Lifting



Relation Lifting



Moral. Need ways to **extend** constructions on types/**sets** to **relations**

Relation Lifting

Lax Extension (Barr, 1970; Thijs, 1996)

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

$$\Gamma(R); \Gamma(S) \subseteq \Gamma(R; S)$$

$$F(f) \subseteq \Gamma(f)$$

$$F(f)^{\mathsf{T}} \subseteq \Gamma(f^{\mathsf{T}})$$

$$R \subseteq S \implies \Gamma(R) \subseteq \Gamma(S)$$

Functor $\mapsto$ **Lax Functor**

$$\begin{bmatrix} \mathsf{Set} \\ \downarrow F \\ \mathsf{Set} \end{bmatrix} \mapsto \begin{bmatrix} W\text{-}\mathsf{Rel} \\ \downarrow \Gamma^{\#}(F) \\ W\text{-}\mathsf{Rel} \end{bmatrix}$$

Relation Lifting

Q. What about **modal types** $[j]\tau$?

Relation Lifting

Q. What about **modal types** $[j]\tau$?

Modal types $=$ **Graded comonadic lax extension** of the **identity** comonad

- ☞ Graded comonadic $=$ graded S4 modalities
- ☞ Identity $=$ act on possible worlds only

Relation Lifting

Graded Comonadic Lax Extension

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

$$\Delta_1(R) \subseteq R$$

$$\Delta_{j*k}(R) \subseteq \Delta_j(\Delta_k(R))$$

$$\Delta_j(R) \otimes \Delta_j(S) \subseteq \Delta_j(R \otimes S)$$

$$\Delta_{j+k}(R) \subseteq dup^{\mathsf{T}}; (\Delta_j(R) \otimes \Delta_j(S)); dup$$

Identity Comonad $\mapsto$ **Graded Lax Monoidal Comonad**

$$\left[\begin{array}{c} \mathsf{Set} \\ \downarrow ID \\ \mathsf{Set} \end{array}\right] \mapsto \left[\begin{array}{c} W\text{-Rel} \\ \downarrow \Delta^{\#}(ID) \\ W\text{-Rel} \end{array}\right]$$

Modal Applicative Bisimilarity

Modal Applicative Bisimilarity

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

$$w \Vdash e \, R \, e' : \tau \text{ and } e \Downarrow a \implies e' \Downarrow a' \text{ and } w \Vdash a \, R \, a' : \tau$$

$$w \Vdash f \, R \, f' : \tau \to \tau' \implies \forall a. \, w' \Vdash fa \, R \, f'a : \tau'$$

$$w \Vdash \mathsf{box}\, a \, R \, \mathsf{box}\, a' : [j]\tau \implies w \Vdash a \, \Delta_j(R) \, a' : \tau$$

Modal Applicative Bisimilarity

Modal Applicative Bisimilarity

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

$$w \Vdash e \, R \, e' : \tau \text{ and } e \Downarrow a \implies e' \Downarrow a' \text{ and } w \Vdash a \, R \, a' : \tau$$

$$w \Vdash f \, R \, f' : \tau \to \tau' \implies \forall a. \, w' \Vdash fa \, R \, f'a : \tau'$$

$$w \Vdash \mathsf{box}\, a \, R \, \mathsf{box}\, a' : [j]\tau \implies w \Vdash a \, \Delta_j(R) \, a' : \tau$$

Theorem (Compositionality)

Modal applicative bisimilarity is **compositional**

$$\frac{x :_j \tau \vdash e, e' : \tau' \quad v \Vdash e \simeq e' \quad w \Vdash a \, \Delta_j(\simeq) \, a' : \tau}{w \bullet v \Vdash e[x := a] \simeq e'[x := a'] : \tau'}$$

Metric Preservation, Non-Interference, etc

Metric Preservation (Reed & Pierce, 2010)

$$\mathcal{J} = [0, \infty] = W$$

$$\frac{x :_k \tau \vdash f : \tau' \qquad \varepsilon \Vdash a \simeq a' : \tau}{k\varepsilon \Vdash f[x := a] \simeq f[x := a'] : \tau'}$$

Non-Interference (Abadi et al., 1999)

$$\mathcal{J} = \{public \leqslant secret\} = W$$

$$x :_{secret} \tau \vdash f : \tau' \;\;\&\;\; a, a' : \tau \Rightarrow public \Vdash f[x := a] \simeq f[x := a'] : \tau'$$

Modal Reasoning = Metric Reasoning, via Lawvere

Program Distance

Relations	***W*-relations**
$e \simeq e'$	$w \Vdash e \simeq e'$
Equivalence	Intensional equivalence

Program Distance

Relations	W-relations	Distances
$e \simeq e'$	$w \Vdash e \simeq e'$	$\delta(e, e') = \varepsilon$
Equivalence	Intensional equivalence	Pseudometric

Program Distance

Relations	W-relations	Distances
$e \simeq e'$	$w \Vdash e \simeq e'$	$\delta(e, e') = \varepsilon$
Equivalence	Intensional equivalence	Pseudometric

Goal. Intensional equivalence $\equiv$ Program distance

- ☞ Solid theory of program distance
- ☞ Combined effects and coeffects

From Equivalences to Distances

Relations	Distances
$e \simeq e'$	$\delta(e, e') = \varepsilon$
$\{false, true\}$	$[0, \infty]$
Equivalence	Pseudometric

From Equivalences to Distances

Relations	Distances	Quantale-relations
$e \simeq e'$	$\delta(e, e') = \varepsilon$	$\delta(e, e') = v$
$\{false, true\}$	$[0, \infty]$	$(V, \leqslant, \otimes, k)$
Equivalence	Pseudometric	V-equivalences

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

Quantale (Rosenthal, 1990)

A complete lattice $(V, \leqslant)$ with a monoid structure $(V, \otimes, k)$

$$v \otimes \bigvee_i u_i = \bigvee_i (v \otimes u_i) \qquad \bigvee_i v_i \otimes u = \bigvee_i (v_i \otimes u)$$

Quantale-relations

Example

Boolean	Lawvere	Strong Lawvere
$(\{\mathit{false}, \mathit{true}\}, \leqslant, \wedge, \top)$	$([0, \infty], \geqslant, +, 0)$	$([0, \infty], \geqslant, \max, 0)$
3-element chain	**Powerset**	**Left cont. distributions**
$\{\bot, k, \top\}$	$\mathcal{P}(X)$	$f : [0, \infty] \to [0, 1]$

Example

Monotone *W*-predicates

$p : (W, \leqslant, \bullet, \varepsilon) \to (2, \leqslant)$

Quantale-relations

Category V-Rel

- Objects: $X, Y, \dots$
- Arrows: $\alpha : X \times Y \to \mathsf{V}$

Identity

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

Composition

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

Quantale-relations

Category V-Rel

- Objects: $X, Y, \dots$
- Arrows: $\alpha : X \times Y \to \mathsf{V}$

Identity

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

Composition

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

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

Boolean	Lawvere	Strong Lawvere
Transitivity	Triangle Inequality	Strong TI
Equivalence	Pseudometric	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 δ

The largest V-relation α s.t.

$$\vdots$$

$$\alpha_{\tau \to \tau'}(\lambda x.f, \lambda x.f') \leqslant \bigwedge_{a} \alpha_{\tau}(f[x := a], f'[x := a])$$

$$\alpha_{[j]\tau}(\mathbf{box}\, a, \mathbf{box}\, a') \leqslant \Delta_j(\alpha_\tau)(a, a')$$

Bisimilarity Distance

Comonadic Lax Extension

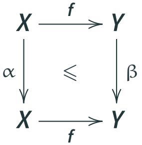
$\Delta_j : \mathsf{V}\text{-}\mathsf{Rel}(X, Y) \to \mathsf{V}\text{-}\mathsf{Rel}(X, Y)$

Main Example

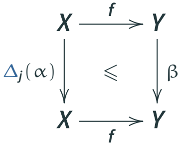
$\mathsf{V} = \mathcal{J} = [0, \infty]$

$\Delta_j(\alpha)(x, y) = j \cdot \alpha(x, y)$

Non-expansive



Lipschitz-continuous



Bisimilarity Distance

Theorem (Abstract Metric Preservation)

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

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

Theorem

For $\mathsf{V} = (W, \leqslant, \bullet, \varepsilon) \to (2, \leqslant)$,

$$\text{AMP} \implies \text{Compositionality}$$

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

Summing Up

What do we gain from AMP?

Summing Up

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

$$(\ell := \mathbf{2}; !\ell + \mathbf{3}) \oplus_{\frac{1}{3}} (\ell := \mathbf{3}; !\ell - \mathbf{1})$$

Summing Up

What do we gain from AMP? Combined effects and coeffects

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

$$(\ell := 2; !\ell + 3) \oplus_{\frac{1}{3}} (\ell := 3; !\ell - 1)$$

Bisimilarity distance using monadic lax extension

Monadic Lax Extension

$\Gamma : \mathsf{V}\text{-}\mathsf{Rel}(X, Y) \to \mathsf{V}\text{-}\mathsf{Rel}(T(X), T(Y))$

Lax distributive law

$\Delta_r \circ \Gamma \subseteq \Gamma \circ \Delta_r$

Abstract metric preservation theorem

References

Abadi, M., Banerjee, A., Heintze, N., & Riecke, J. G. (1999). A core calculus of dependency. In *POPL '99, proceedings of the 26th ACM SIGPLAN-SIGACT symposium on principles of programming languages, san antonio, tx, usa, january 20-22, 1999* (pp. 147–160).

Abel, A., & Bernardy, J. (2020). A unified view of modalities in type systems. *Proc. ACM Program. Lang.*, *4*(ICFP), 90:1–90:28.

Abramsky, S. (1990). The lazy lambda calculus. In D. Turner (Ed.), *Research topics in functional programming* (pp. 65–117). Addison Wesley.

Atkey, R. (2018). Syntax and semantics of quantitative type theory. In *Proceedings of the 33rd annual ACM/IEEE symposium on logic in computer science, LICS 2018, oxford, uk, july 09-12, 2018* (pp. 56–65).

Barr, M. (1970). Relational algebras. *Lect. Notes Math.*, *137*, 39–55.

Brunel, A., Gaboardi, M., Mazza, D., & Zdancewic, S. (2014). A core quantitative coeffect calculus. In *Programming languages and systems - 23rd european symposium on programming, ESOP 2014, held as part of the european joint conferences on theory and practice of software, ETAPS 2014, grenoble, france, april 5-13, 2014, proceedings* (pp. 351–370).

Gaboardi, M., Katsumata, S., Orchard, D. A., Breuvart, F., & Uustalu, T. (2016). Combining effects and coeffects via grading. In *Proceedings of the 21st ACM SIGPLAN international conference on functional programming, ICFP 2016, nara, japan, september 18-22, 2016* (pp. 476–489).

Gavazzo, F. (2018). Quantitative behavioural reasoning for higher-order effectful programs: Applicative distances. In *Proc. of LICS 2018* (pp. 452–461).

Ghica, D. R., & Smith, A. I. (2014). Bounded linear types in a resource semiring. In *Programming languages and systems - 23rd european symposium on programming, ESOP 2014, held as part of the european joint conferences on theory and practice of software, ETAPS 2014, grenoble, france, april 5-13, 2014, proceedings* (pp. 331–350).

Girard, J., Scedrov, A., & Scott, P. J. (1992). Bounded linear logic: A modular approach to polynomial-time computability. *Theor. Comput. Sci.*, *97*(1), 1–66.

Hofmann, D., Seal, G., & Tholen, W. (Eds.). (2014). *Monoidal topology. a categorical approach to order, metric, and topology* (No. 153). Cambridge University Press.

Hofmann, M. (1999). Linear types and non-size-increasing polynomial time computation. In *14th annual IEEE symposium on logic in computer science, trento, italy, july 2-5, 1999* (pp. 464–473).

Lago, U. D., & Hofmann, M. (2009). Bounded linear logic, revisited. In *Typed lambda calculi and applications, 9th international conference, TLCA 2009, brasilia, brazil, july 1-3, 2009. proceedings* (pp. 80–94).

Lawvere, F. (1973). Metric spaces, generalized logic, and closed categories. *Rend. Sem. Mat. Fis. Milano*, *43*, 135–166.

Orchard, D., Liepelt, V., & Eades III, H. (2019). Quantitative program reasoning with graded modal types. *Proc. ACM Program. Lang.*, *3*(ICFP), 110:1–110:30.

Petricek, T., Orchard, D. A., & Mycroft, A. (2014). Coeffects: a calculus of context-dependent computation. In *Proceedings of the 19th ACM SIGPLAN international conference on functional programming, gothenburg, sweden, september 1-3, 2014* (pp. 123–135).

Reed, J., & Pierce, B. C. (2010). Distance makes the types grow stronger: a calculus for differential privacy. In *Proceeding of the 15th ACM SIGPLAN international conference on functional programming, ICFP 2010, baltimore, maryland, usa, september 27-29, 2010* (pp. 157–168).

Rosenthal, K. (1990). *Quantales and their applications*. Longman Scientific & Technical.

Routley, R., & Meyer, R. K. (1973). The semantics of entailment. In H. Leblanc (Ed.), *Truth, syntax and modality* (Vol. 68, p. 199 - 243). Elsevier.

Thijs, A. (1996). *Simulation and fixpoint semantics*. Rijksuniversiteit Groningen.

Urquhart, A. (1972). Semantics for relevant logics. *J. Symb. Log.*, *37*(1), 159–169.

Volpano, D. M., Irvine, C. E., & Smith, G. (1996). A sound type system for secure flow analysis. *Journal of Computer Security*, *4*(2/3), 167–188.

Wood, J., & Atkey, R. (2020). A linear algebra approach to linear metatheory. *CoRR*, *abs/2005.02247*. Retrieved from *https://arxiv.org/abs/2005.02247*