

Modular construction of sound-and-complete simplifiers for varieties of semi-rings

Ohad Kammar, LFCS, University of Edinburgh and
Amaury Mazoyer, École Normale Supérieure de Lyon



26th IFIP Working Group 2.11
October 13, 2026, Niagara-on-the-Lake, Ontario, Canada



THE UNIVERSITY of EDINBURGH
informatics lfcs

Laboratory for Foundations
of Computer Science



BayesCentre

supported by:



THE ROYAL
SOCIETY

Algebraic simplification

represent **equivalence classes** of expressions

$$1 + 3x + 6y + (-1) \in [0, 3, 6] \ni 3y + 3x + 3y \quad (\text{integers mod } \begin{cases} \text{commutativity} \\ \text{associativity} \\ \text{neutrality} \end{cases})$$

$$\mathbb{Z} \times \mathbb{N} \times \mathbb{N} \ni [a, n, k] \quad \text{represents} \quad a + nx + ky$$

Algebraic simplification

represent **equivalence classes** of expressions

$$1 + 3x + 6y + (-1) \in [0, 3, 6] \ni 3y + 3x + 3y \quad (\text{integers mod } \left\{ \begin{array}{l} \text{commutativity} \\ \text{associativity} \\ \text{neutrality} \end{array} \right.)$$

$$\mathbb{Z} \times \mathbb{N} \times \mathbb{N} \ni [a, n, k] \quad \text{represents} \quad a + nx + ky$$

Robust code generation

- ▶ **Partial evaluation** (Carette et al., 2009, tagless final)
esp. **partially-static data** structures (Yallop et al., 2018, frex)
- ▶ Rôle-proof **synthesis** (Allais et al., 2025, Agda/Idris frex)
- ▶ Program **synthesis** (Shepherd et al., 2026)

Algebraic simplification

represent **equivalence classes** of expressions

$$1 + 3x + 6y + (-1) \in [0, 3, 6] \ni 3y + 3x + 3y \quad (\text{integers mod } \begin{cases} \text{commutativity} \\ \text{associativity} \\ \text{neutrality} \end{cases})$$

$$\mathbb{Z} \times \mathbb{N} \times \mathbb{N} \ni [a, n, k] \quad \text{represents} \quad a + nx + ky$$

Extensions beyond ordinary algebra

- ▶ **NbE** with algebraic structure (Corbyn et al., 2022)
- ▶ 2nd and 3rd **Futamura** projections of **System T** (Brown, 2027, thesis)
uses frex for **second-order** algebraic theories

Modern algebra's core idea:

Theorem (extensional normalisation)

Let:

- ▶ $\mathcal{T}$ be an algebraic theory;
- ▶ $A = \langle \underline{A}, A \llbracket - \rrbracket \rangle$ be a free $\mathcal{T}$ -model with variables from X , i.e.: and $\text{var} : X \rightarrow \underline{A}$.

Then:

$$\begin{array}{ccc} \text{an equation } X \vdash t \equiv u \text{ is} & \iff & A \text{ \textbf{validates} it} \\ \text{\textbf{provable} in } \mathcal{T} & & A \llbracket t \rrbracket \text{var} = A \llbracket u \rrbracket \text{var} \end{array}$$

Takeaway

computable/effective representation of free models



normalisation/algebraic simplification **procedures**

Modern algebra's core idea:

Theorem (extensional normalisation)

Let:

- ▶ $\mathcal{T}$ be an algebraic theory;
- ▶ $A = \langle \underline{A}, A \llbracket - \rrbracket \rangle$ be a free $\mathcal{T}$ -model with variables from X , i.e.: and $\text{var} : X \rightarrow \underline{A}$.

Then:

$$\begin{array}{ccc} \text{an equation } X \vdash t \equiv u \text{ is} & \iff & A \text{ \textbf{validates} it} \\ \text{\textbf{provable} in } \mathcal{T} & & A \llbracket t \rrbracket \text{ var} = A \llbracket u \rrbracket \text{ var} \end{array}$$

NB: consequences beyond algebraic simplification

- ▶ **algebraic effects** in semantics (Plotkin and Power, 2002)
- ▶ **effect handlers** in functional programming (Bauer and Pretnar, 2015)
- ▶ **modular abstract syntax** with binding+substitution+holes (Fiore et al., 1999)

Modern algebra's core idea:

Theorem (extensional normalisation)

Let:

- ▶ $\mathcal{T}$ be an algebraic theory;
- ▶ $A = \langle \underline{A}, A \llbracket - \rrbracket \rangle$ be a free $\mathcal{T}$ -model with variables from X , i.e.: and $\text{var} : X \rightarrow \underline{A}$.

Then:

$$\begin{array}{ccc} \text{an equation } X \vdash t \equiv u \text{ is} & \iff & A \text{ \textbf{validates} it} \\ \text{\textbf{provable} in } \mathcal{T} & & A \llbracket t \rrbracket \text{var} = A \llbracket u \rrbracket \text{var} \end{array}$$

NB2: Monad connection

For $\mathbf{F}_{\mathcal{T}}X = \langle MX, \llbracket - \rrbracket \rangle$ free $\mathcal{T}$ -model over X via $\text{var} : X \rightarrow MX$:

- ▶ MX is a **monad** with var its unit;
- ▶ $\mathbf{F}_{\mathcal{T}}X$ are all the **algebraic effects** of it;
- ▶ $\mathcal{T}$ -models are all the **effect handlers** of them.

theory of algebraic simplifiers $\xRightarrow{\text{ought to give}}$ **building blocks** for algebraic simplifiers?

Initial results (Allais et al., 2023)

- ▶ Free involutive $\mathcal{T}$ -models over/extensions by X from those for $\mathcal{T}$ and $\text{Bool} \times X$
- ▶ Free $\mathcal{T}$ -homomorphism graph models/extensions over X from those of $\mathcal{T}$ for all Y

This talk

distributive combinations of theories and their simplifiers

Semiring $A = \langle \underline{A}, 1, (\cdot), 0, (+) \rangle$

- ▶ monoid $\langle \underline{A}, 1, (\cdot) \rangle$ that
- ▶ **distributes** over a commutative monoid $\langle \underline{A}, 0, (+) \rangle$:

$$0 \cdot x = 0 = x \cdot 0 \quad (x + y) \cdot z = x \cdot z + y \cdot z \quad x \cdot (y + z) = x \cdot y + x \cdot z$$

Semi-rings are what makes matrices 'work'.

Semiring $A = \langle \underline{A}, 1, (\cdot), 0, (+) \rangle$

- ▶ monoid $\langle \underline{A}, 1, (\cdot) \rangle$ that
- ▶ **distributes** over a commutative monoid $\langle \underline{A}, 0, (+) \rangle$:

$$0 \cdot x = 0 = x \cdot 0 \quad (x + y) \cdot z = x \cdot z + y \cdot z \quad x \cdot (y + z) = x \cdot y + x \cdot z$$

Semi-rings are what makes matrices 'work'.

Can vary the two algebraic structures to get:

- ▶ rings, i.e., addition has inverses;
- ▶ (semi-)rings without multiplicative unit or additive zero;
- ▶ commutative (semi-)rings;
- ▶ involution over multiplication; etc.

We'll consider the general situation.

Semiring $A = \langle \underline{A}, 1, (\cdot), 0, (+) \rangle$

- ▶ monoid $\langle \underline{A}, 1, (\cdot) \rangle$ that
- ▶ **distributes** over a commutative monoid $\langle \underline{A}, 0, (+) \rangle$:

$$0 \cdot x = 0 = x \cdot 0 \quad (x + y) \cdot z = x \cdot z + y \cdot z \quad x \cdot (y + z) = x \cdot y + x \cdot z$$

Semi-rings are what makes matrices 'work'.

Distributive combination $(\mathcal{T} \triangleleft \mathcal{S})$ -model $A = \langle \underline{A}, \mathcal{T}[-], \mathcal{S}[-] \rangle$

- ▶ $\mathcal{T}$ -model $\langle \underline{A}, \mathcal{T}[-] \rangle$; that
- ▶ **distributes** over an $\mathcal{S}$ -model $\langle \underline{A}, \mathcal{S}[-] \rangle$, i.e., for all:
 $\mathcal{T}$ -operation $(f : n) \in \mathcal{T}$, $\mathcal{S}$ -operation $(g : k)$ and position $1 \leq i \leq n$:

$$\begin{aligned} \mathcal{T} \llbracket f \rrbracket (x_1, \dots, x_{i-1}, \mathcal{S} \llbracket g \rrbracket (y_1, \dots, y_k), x_{i+1}, \dots, x_n) = \\ \mathcal{S} \llbracket g \rrbracket (\mathcal{T} \llbracket f \rrbracket (x_1, \dots, x_{i-1}, y_1, x_{i+1}, \dots, x_n), \dots, \\ \mathcal{T} \llbracket f \rrbracket (x_1, \dots, x_{i-1}, y_k, x_{i+1}, \dots, x_n)) \end{aligned}$$

Example distributive combinations

Multiplicative structure			Additive structure		
Theory	Comm. ¹	Inv. ³	Comm. Semigroup	Comm. Monoid	Abelian Group
Semigroup	Mere	Mere	$\mathcal{P}(\Sigma^+) \setminus \{\emptyset\}$	$\mathcal{P}(\Sigma^+)$	$\mathbb{Z} \langle \Sigma^+ \rangle$
		Inv. ⁴	$(\mathcal{P}(\Sigma^+) \setminus \{\emptyset\}, -^R)$	$(\mathcal{P}(\Sigma^+), -^R)$	$(\mathbb{Z} \langle \Sigma^+ \rangle, \text{rev})$
	Comm. ²	Mere	$2\mathbb{N}_{>0}$	$2\mathbb{N}$	$2\mathbb{Z}$
		Inv.	$\mathfrak{m}_n^\sigma \setminus \{0\}$	$\mathfrak{m}_n^\sigma$	$(2\mathbb{Z}[\mathfrak{i}], -)$
Monoid	Mere	Mere	$\mathcal{P}(\Sigma^*) \setminus \{\emptyset\},$ $M_n(\mathbb{N}_{>0})$	$\mathcal{P}(\Sigma^*), M_n(\mathbb{N})$	$M_n(\mathbb{Z})$
		Inv.	$(M_n(\mathbb{N}_{>0}), -^T),$ $(\mathcal{P}(\Sigma^*) \setminus \{\emptyset\}, -^R)$	$(M_n(\mathbb{N}), -^T),$ $(\mathcal{P}(\Sigma^*), -^R), \text{Rel}(X)$	$(M_n(\mathbb{Z}), -^T)$
	Comm.	Mere	$\mathbb{N}_{>0}$	$(\mathbb{N})$	$(\mathbb{Z}, \mathbb{R}, \mathbb{C})$
		Inv.	$\mathbb{N}_\sigma[X_1, \dots, X_n] \setminus \{0\}$	$\mathbb{N}_\sigma[X_1, \dots, X_n]$	$(\mathbb{C}, -),$ $\mathbb{Z}_\sigma[X_1, \dots, X_n]$
Group	Mere	Mere	$T_{n, \Delta_{>0}}(\mathbb{R})$	impossible nontrivially	impossible nontrivially
		Inv.	$(T_{n, \Delta_{>0}}(\mathbb{R}), -^\dagger)$	impossible nontrivially	impossible nontrivially
	Comm.	Mere	$\mathbb{R}_{>0}$	impossible nontrivially	impossible nontrivially
		Inv.	$(\mathbb{C}_{>0}, -)$	impossible nontrivially	impossible nontrivially

Legend: ¹Commutativity; ²Commutative; ³Involutivity; ⁴Involutive

Theorem (Fujii et al. 2026)

For the distributive combination $\mathcal{T} \triangleleft \mathcal{S}$, if $(\mathcal{T}, \mathcal{S})$ are **compatible**:

- ▶ Let A be free $\mathcal{T}$ -model on X ; and
- ▶ Let B be free $\mathcal{S}$ -model on $\underline{A}$.

Then B equipped with **$\mathcal{S}$ -convolution** of A is the free $(\mathcal{T} \triangleleft \mathcal{S})$ -model on X .

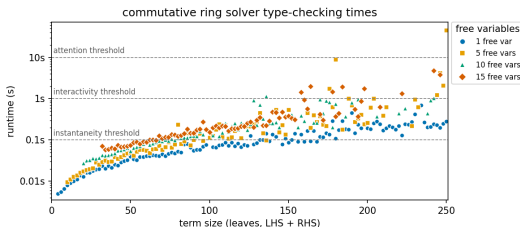
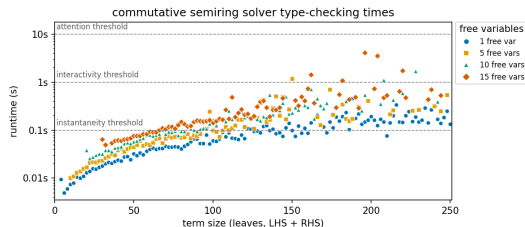
giving highly modular algebraic simplifiers

Especially for rôle-proof synthesis, this theorem gives dozens of **sound-and-complete** algebraic simplifiers.

Modularity is an issue, c.f. Rocq's ecosystem:

- ▶ uses **almost rings** (Grégoire and Mahboubi, 2005)
- ▶ one algebraic structure
- ▶ incomplete, but good enough, simplifier

Associated implementation doesn't suck for, e.g., responsive **type-level computation**:



Type-checking times for the commutative semiring and ring simplifiers.

Talk structure

- ▶ Introduction
- ▶ Why is it difficult?
- ▶ Generalised convolution
- ▶ Verbal substructure
- ▶ Compatibility

Known instances

$$T_{\text{semi-ring}} = T_{\text{commutative monoid}} \circ T_{\text{monoid}}$$

cf. category-theoretic concepts of **monad-monad distributive law** (Beck, 1969)

Monad-connection — notable interest for PL-semanticists

Known instances

$$T_{\text{semi-ring}} = T_{\text{commutative monoid}} \circ T_{\text{monoid}}$$

cf. category-theoretic concepts of **monad-monad distributive law** (Beck, 1969)

Monad-connection — notable interest for PL-semanticists

Impossibility results

Plotkin's counter-example (Varacca and Winskel, 2006)—**no** distributive law between

- ▶ probabilistic non-determinism: free Barycentric algebra / finite probability distribution monad
- ▶ non-determinism: free finite powerset monad

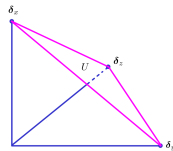
Systematically generalised by Zwart and Marsden (2022).

Distributive combination

Can be non-modular:

free (probability $\triangleleft$ non-determinism)-model carried by **credal sets** (Levi, 1980):

(finitely-generated) convex subsets of probability distributions



This approach

- ▶ Pioneered by Parlant (2020); Parlant et al. (2020); Dahlqvist et al. (2018)
- ▶ Fujii et al. (2026) connect it to Tronin's (2002) **verbal categories**, making the approach general and systematic

Limitations

- ▶ Only for monads over sets, cf. **Dash and Staton's (2021)** distributive combination for point processes:
 - probability distributions over measurable spaces
 - distributes over**
 - multisets / commutative monoids
- ▶ **inapplicable** theory when $\mathcal{S}$ is not **commutative/exchangeable**

Talk structure

- ▶ Introduction
- ▶ Why is it difficult?
- ▶ **Generalised convolution**
- ▶ Verbal substructure
- ▶ Compatibility

Synthetic convolution (folklore)

Given:

- monad $\mathcal{M} = \langle M, \text{val}, \text{let} \rangle$:

$$\frac{X \text{ set}}{MX \text{ set}} \quad \frac{x : X}{\text{val } x : MX} \quad \frac{a : MX \quad x : X \vdash b : MY}{\text{let } x = a \text{ in } b : MY}$$

- operation $f : X_1 \times \cdots \times X_n \rightarrow Y$

the $\mathcal{M}$ -convolution of f :

$$\text{Conv}_{\mathcal{M}} f : MX_1 \times \cdots \times MX_n \rightarrow MY$$

$$\text{Conv}_{\mathcal{M}} f(a_1, \dots, a_n) := \text{let } x_1 = a_1 \text{ in}$$

$$\vdots$$

$$\text{let } x_n = a_n \text{ in val } f(x_1, \dots, x_n)$$

Lift $A = \langle \underline{A}, \llbracket A \rrbracket - \rangle$ to $\text{Conv}_{\mathcal{M}} A = \langle M \underline{A}, \llbracket A \rrbracket f := \text{Conv}_{\mathcal{M}} f \rangle$

The generalised convolution, and its converse the **de-convolution** are common:

area	operation	convolution	de-convolution
scanning/lexing/parsing	string concat	language concat	Brzowski derivative
database theory	record concatenation	cartesian product	(some) queries
probability & statistics	trace concatenation	generative modelling	disintegration

also, collection programming with semi-ring dictionaries (**Shaikhha et al., 2022**).

Why 'convolution'?

Take:

- ▶ the Giry monad $DX = \{\text{probability measures over measurable space } X\}$
- ▶ $f : (+) : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$

For probability measures $\mu, \nu \in D\mathbb{R}$ with probability density functions $\varphi, \psi : \mathbb{R} \rightarrow [0, \infty]$

$$(\text{so } \mathbb{P}_\mu[E] = \int_E d\mathbf{x} \varphi \mathbf{x} \text{ and } \mathbb{P}_\nu[E] = \int_E d\mathbf{x} \psi \mathbf{x})$$

we have that $\mathbf{Conv}_{\text{Giry}}(\mu, \nu)$ has density function:

$$\varphi * \psi : \mathbb{R} \rightarrow [0, \infty] \quad (\varphi * \psi)t := \int_{-\infty}^{+\infty} d\mathbf{x} \varphi \mathbf{x} \cdot \psi(t - \mathbf{x})$$

and $\varphi * \psi$ is determined almost surely by $\mathbf{Conv}_{\text{Giry}}(\mu, \nu)$, and uniquely for continuous density functions.

Central question

Given:

- ▶ algebraic structure A ; and
- ▶ monad $\mathcal{S}$

which equations lift from A to $\mathbf{Conv}_{\mathcal{S}}A$?

$$A \models X \vdash l \equiv r \implies \mathbf{Conv}_{\mathcal{S}}A \models X \vdash l \equiv r$$

Example & non-example

For group $A = \langle \mathbb{Z}, 0, (+), (-) \rangle$, $\mathbf{Conv}_{\text{powerset}}A$ is a monoid:

- ▶ monoid axioms lift: $0 + x = x = x + 0$ $x + (y + z) = (x + y) + z$
- ▶ inverse axioms fail to lift: $x + (-x) = 0$:

$$\{1, -1\} \oplus (\ominus \{1, -1\}) = \{1, -1\} \oplus (\{-1, 1\}) = \{0, 2, -2\} \neq \{0\}$$

Central question

Given:

- ▶ algebraic structure A ; and
- ▶ monad $\mathcal{S}$

which equations lift from A to $\mathbf{Conv}_{\mathcal{S}}A$?

$$A \models X \vdash l \equiv r \implies \mathbf{Conv}_{\mathcal{S}}A \models X \vdash l \equiv r$$

- ▶ [Gautam \(1957\)](#) and [Manes and Mulry \(2007\)](#):
for $\mathcal{S}$ = powerset, exactly linear equations
- ▶ [Parlant \(2020\)](#) characterises for: commutative $\mathcal{S}$, commutative-affine $\mathcal{S}$, etc.
- ▶ [Fujii et al. \(2026\)](#) systematic theory for commutative $\mathcal{S}$

Fujii et al.'s (2026) approach

- ▶ Abstract over 'substructural manipulation of context':
Tronin's (2002) **verbal subcategories** $\mathbb{W}$
- ▶ Theories $\mathcal{T}$ **definable** by $\mathbb{W}$ -**allowed** axioms: $\mathcal{T} \perp_{\mathbb{W}}$
- ▶ $\mathbb{W}$ -**commutative monads** $\mathcal{S}$ preserve $\mathbb{W}$ -structure: $\perp_{\mathbb{W}} \mathcal{S}$.

The main theorem holds for $\mathcal{T} \perp_{\mathbb{W}} \mathcal{S}$ with $\mathbb{W}$ permitting variable exchange.

Talk structure

- ▶ Introduction
- ▶ Why is it difficult?
- ▶ Generalised convolution
- ▶ **Verbal substructure**
- ▶ Compatibility

Algebraic contexts $\Gamma, \Delta \in \mathbb{F}$

represent by natural numbers

variables in Γ are the numbers below Γ

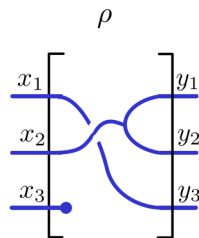
Write: $\Gamma = \{x_1, \dots, x_n\}$, $\Delta = \{y_1, \dots, y_k\}$

Renaming $\Gamma \vdash \rho : \Delta$

maps variables $y \in \Delta$ to variables $y[\rho] \in \Gamma$

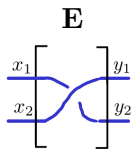
Example

$$\{x_1, x_2, x_3\} \vdash \rho : \begin{cases} x_2 \leftarrow y_1 \\ x_2 \leftarrow y_2 \\ x_1 \leftarrow y_3 \end{cases} : \{y_1, y_2, y_3\}$$

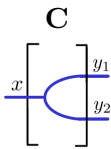


information flow graph notation

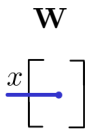
The three 'structural' rules are:



exchange

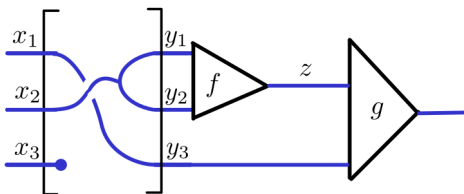


contraction

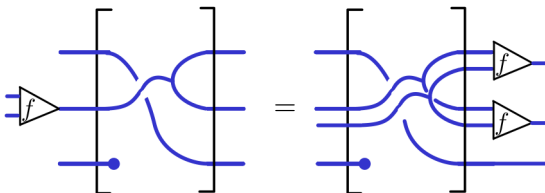


weakening

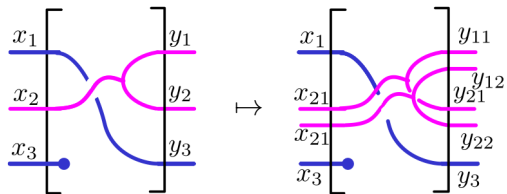
Renaming compose sequentially and ‘in-parallel’. We use them to depict structural manipulation on contexts of algebraic expressions:



Verbal commutativity (‘naturality’)



Verbal inflation



Verbal subcategory $\mathbb{W}$ (Tronin, 2002)

$\mathbb{W} \subseteq \mathbb{T}$ closed under:

- ▶ identities
- ▶ sequential and parallel composition
- ▶ verbal inflation

Verbal closure

S_{vrb} smallest verbal subcategory containing S :

- ▶ $\{\mathbf{E}\}_{\text{vrb}} = \mathbb{W}_{\text{bij}}$: bijective renamings
(all variable exchanges permitted)
- ▶ $\{\mathbf{E}, \mathbf{C}\}_{\text{vrb}} = \mathbb{W}_{\text{surj}}$: surjective renamings
(all variables are relevant)
- ▶ $\{\mathbf{E}, \mathbf{W}\}_{\text{vrb}} = \mathbb{W}_{\text{surj}}$: injective renamings
(affine use of variables)

Verbal subcategory $\mathbb{W}$ (Tronin, 2002)

$\mathbb{W} \subseteq \mathbb{T}$ closed under:

- ▶ identities
- ▶ sequential and parallel composition
- ▶ verbal inflation

NB: András Kovács will use **thinnings**:

$\{\mathbf{W}\}_{\text{vrb}}$: subset inclusions

Fujii et al.'s (2026) account only concerns **symmetric/exchangeable** verbal subcategories—those containing $\mathbf{E}$.

Verbal closure

S_{vrb} smallest verbal subcategory containing S :

- ▶ $\{\mathbf{E}\}_{\text{vrb}} = \mathbb{W}_{\text{bij}}$: bijective renamings
(all variable exchanges permitted)
- ▶ $\{\mathbf{E}, \mathbf{C}\}_{\text{vrb}} = \mathbb{W}_{\text{surj}}$: surjective renamings
(all variables are relevant)
- ▶ $\{\mathbf{E}, \mathbf{W}\}_{\text{vrb}} = \mathbb{W}_{\text{surj}}$: injective renamings
(affine use of variables)

Talk structure

- ▶ Introduction
- ▶ Why is it difficult?
- ▶ Generalised convolution
- ▶ Verbal substructure
- ▶ Compatibility

Linear terms $\Gamma \vdash_{\circ \Sigma} t$ over algebraic signature Σ

$$\frac{}{\{x\} \vdash_{\circ \Sigma} x} \quad \frac{\Gamma_1 \vdash_{\circ \Sigma} t_1, \dots, \Gamma_n \vdash_{\circ \Sigma} t_n}{\Gamma_1, \dots, \Gamma_n \vdash_{\circ \Sigma} f(t_1, \dots, t_n)} ((f : n) \in \Sigma)$$

Verbally definable equation over $\mathbb{W}$

Equation of the form $\Gamma \vdash t[\rho] \equiv t'[\rho']$ where:

- ▶ $\Delta \vdash_{\circ} t$ and $\Gamma \vdash_{\mathbb{W}} \rho : \Delta$
- ▶ $\Delta' \vdash_{\circ} t'$ and $\Gamma \vdash_{\mathbb{W}} \rho' : \Delta'$

Verbally definable theory over $\mathbb{W}$ (' $\mathbb{W}$ -operadic')

theory axiomatisable by verbally definable equations over $\mathbb{W}$

We then write $\mathcal{T} \perp_{\mathbb{W}}$

Theorem

Let $\mathcal{S}$ be a monad and $\mathbb{W} \ni \mathbf{E}$ a symmetric verbal subcategory. TFAE:

- For every algebraic signature Σ and Σ -algebra A , and every $\mathbb{W}$ -definable equation eq over Σ :

$$A \models \text{eq} \implies \mathbf{Conv}_{\mathcal{S}} A \models \text{eq}$$

- For every $\mathbb{W}$ -renaming $\Gamma \vdash_{\mathbb{W}} \rho : \Delta$ and set X :

$$\mathbf{Conv}_{\mathcal{S}} \rho = (\mathbf{Conv}_{\mathcal{S}} \text{id})^{\rho} : (\mathcal{S}X)^{\Gamma} \rightarrow \mathcal{S}(X^{\Delta})$$

In this case we write $\perp_{\mathbb{W}} \mathcal{S}$ and call $\mathcal{S}$ a **$\mathbb{W}$ -commutative** monad.

Theorem (Fujii et al. 2026; Parlant 2020)

A monad is:

- ▶ $\mathbb{W}_{bij}$ -commutative iff it is commutative.
- ▶ $\mathbb{W}_{inj}$ -commutative iff it is commutative and affine.
- ▶ $\mathbb{W}_{surj}$ -commutative iff it is commutative and relevant.

(see the paper for more characterisations)

Verbally compatible theory-monad $\mathcal{T} \perp_{\mathbb{W}} \mathcal{S}$

- ▶ $\mathbb{W}$ verbal subcategory;
- ▶ $\mathcal{T}$ is $\mathbb{W}$ -definable; and
- ▶ $\mathcal{S}$ is $\mathbb{W}$ -commutative

Theorem (Fujii et al. 2026)

For:

- ▶ algebraic theory $\mathcal{T}$;
- ▶ algebraic theory $\mathcal{S}$ (or merely monad $\mathcal{S}$);
- ▶ set X ;
- ▶ free $\mathcal{T}$ -model $A = \langle \underline{A}, A[-] \rangle$ over X via $\text{var}^{\mathcal{T}} : X \rightarrow \underline{A}$;
- ▶ free $\mathcal{S}$ -model $B = \langle \underline{B}, B[-] \rangle$ over $\underline{A}$ via $\text{var}^{\mathcal{S}} : \underline{A} \rightarrow \underline{B}$

If $(\mathcal{T}, \mathcal{S})$ are **compatible** then have free $(\mathcal{T} \triangleleft \mathcal{S})$ -model over X by:

$$\langle \underline{B}, \mathbf{Conv}_{\mathcal{S}} A, B[-] \rangle \quad X \xrightarrow{\text{var}^{\mathcal{T}}} \underline{A} \xrightarrow{\text{var}^{\mathcal{S}}} \underline{B}$$

where $\mathbf{Conv}_{\mathcal{S}} A$ is the **$\mathcal{S}$ -convolution** of A .

Summary

- ▶ Modular, sound-and-complete algebraic simplifiers for distributive combinations
- ▶ Suggestive performance for something free, esp. for rôle-proof synthesis
- ▶ Enabled by [Fujii et al.'s \(2026\)](#) analysis of verbal subcategories

Further work

- ▶ Relaxing the commutativity/exchange requirement through pre-operadic/pre-multi-categorical perspective
- ▶ Intensional normalisation via recourse to setoids
- ▶ Investigate runtime performance at scale
- ▶ What about free extensions? They require no structural axiomatisation.
- ▶ Integrate into the rest of the frex project (partial evaluation, program synthesis)

- Allais, G., Brady, E., Corbyn, N., Kammar, O., and Yallop, J. (2025). Frex: Dependently typed algebraic simplification. *Proceedings of the ACM on Programming Languages*, 9(ICFP):30–65.
- Allais, G., Corbyn, N., Kammar, O., Lindley, S., Valliappan, N., and Yallop, J. (2023). Modular construction of multi-sorted free extensions. extended abstract.
- Bauer, A. and Pretnar, M. (2015). Programming with algebraic effects and handlers. *Journal of Logical and Algebraic Methods in Programming*, 84(1):108–123. Special Issue: The 23rd Nordic Workshop on Programming Theory (NWPT 2011) Special Issue: Domains X, International workshop on Domain Theory and applications, Swansea, 5-7 September, 2011.
- Beck, J. (1969). Distributive laws. In Eckmann, B., editor, *Seminar on Triples and Categorical Homology Theory*, pages 119–140, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Brown, G. (2027). *Effective Free Extensions for Languages with Binding*. PhD thesis, University of Edinburgh.
- Carette, J., Kiselyov, O., and Shan, C. (2009). Finally tagless, partially evaluated: Tagless staged interpreters for simpler typed languages. *J. Funct. Program.*, 19(5):509–543.
- Corbyn, N., Kammar, O., Lindley, S., Valliappan, N., and Yallop, J. (2022). Normalization by evaluation with free extensions. extended abstract.
- Dahlqvist, F., Parlant, L., and Silva, A. (2018). Layer by layer – combining monads. In *Theoretical Aspects of Computing – ICTAC 2018*, volume 11187 of *Lecture Notes in Computer Science*, pages 153–172. Springer International Publishing.
- Dash, S. and Staton, S. (2021). A monad for probabilistic point processes. *Electronic Proceedings in Theoretical Computer Science*, 333:19–32.
- Fiore, M. P., Plotkin, G. D., and Turi, D. (1999). Abstract syntax and variable binding (extended abstract). In *Proc. 14th LICS Conf.*, pages 193–202. IEEE, Computer Society Press.
- Fujii, S., Tsai, Y. C., Montacute, Y., and Hasuo, I. (2026). Monads and Distributive Laws in Substructural Contexts. In Faggian, C. and Katoen, J.-P., editors, *41st Annual Symposium on Logic in Computer Science (LICS 2026)*, volume 380 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 45:1–45:28, Dagstuhl, Germany. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.
- Gautam, N. D. (1957). The validity of equations of complex algebras. *Archiv für mathematische Logik und Grundlagenforschung*, pages 117–124.
- Grégoire, B. and Mahboubi, A. (2005). Proving equalities in a commutative ring done right in coq. In Hurd, J. and Melham, T., editors, *Theorem Proving in Higher Order Logics*, pages 98–113, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Levi, I. (1980). *The Enterprise of Knowledge: An Essay on Knowledge, Credal Probability, and Chance*. MIT Press.
- Manes, E. and Mulry, P. (2007). Monad compositions I: General constructions and recursive distributive laws. *Theory and Applications of Categories*, 18:172–208.
- Parlant, L. (2020). *Monad Composition via Preservation of Algebras*. PhD thesis, University College London.
- Parlant, L., Rot, J., Silva, A., and Westerbaan, B. (2020). Preservation of Equations by Monoidal Monads. In Esparza, J. and Král', D., editors, *45th International Symposium on Mathematical Foundations of Computer Science (MFCS 2020)*, volume 170 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 77:1–77:14, Dagstuhl, Germany. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.
- Plotkin, G. and Power, J. (2002). Notions of computation determine monads. In Nielsen, M. and Engberg, U., editors, *Foundations of Software Science and Computation Structures*, pages 342–356, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Shaikhha, A., Huot, M., Smith, J., and Olteanu, D. (2022). Functional collection programming with semi-ring dictionaries. *Proc. ACM Program. Lang.*, 6(OOPSLA1):1–33.
- Shepherd, Z., Kammar, O., and Polgreen, E. (2026). Nsync: Normalised synthesis of computation.
- Tronin, S. (2002). Abstract clones and operads. *Siberian Mathematical Journal*, 43(4):746–755.
- Varacca, D. and Winskel, G. (2006). Distributing probability over non-determinism. *Mathematical Structures in Computer Science*, 16(1):87–113.
- Yallop, J., von Glehn, T., and Kammar, O. (2018). Partially-static data as free extension of algebras. *Proc. ACM Program. Lang.*, 2(ICFP).
- Zwart, M. and Marsden, D. (2022). No-go theorems for distributive laws. *Logical Methods in Computer Science*, 18(1):13:1–13:61.