Distributive laws for Lawvere theories

Eugenia Cheng

University of Sheffield

CT2011

Plan

1. Introduction

Plan

1. Introduction
2. Lawvere theories

Plan

1. Introduction
2. Lawvere theories
3. Distributive laws for monads

Plan

1. Introduction
2. Lawvere theories
3. Distributive laws for monads
4. Three ways to do it

Plan

1. Introduction
2. Lawvere theories
3. Distributive laws for monads
4. Three ways to do it
5. Comparison.

1. Introduction

Distributive laws give us a way of combining two or more types of algebraic structure expressed as monads.

1. Introduction

Distributive laws give us a way of combining two or more types of algebraic structure expressed as monads.

E.g. monoids and abelian groups ⟶ rings

1. Introduction

Distributive laws give us a way of combining two or more types of algebraic structure expressed as monads.

E.g. monoids and abelian groups $\longrightarrow$ rings

Question

What's a distributive law for Lawvere theories?

1. Introduction

Distributive laws give us a way of combining two or more types of algebraic structure expressed as monads.

E.g. monoids and abelian groups $\longrightarrow$ rings

Question

What's a distributive law for Lawvere theories?

- Lawvere theories correspond to finitary monads on **Set**.

1. Introduction

Distributive laws give us a way of combining two or more types of algebraic structure expressed as monads.

E.g. monoids and abelian groups $\longrightarrow$ rings

Question

What's a distributive law for Lawvere theories?

- Lawvere theories correspond to finitary monads on **Set**.
- Lawvere theories are themselves monads in a certain bicategory.

1. Introduction

Distributive laws give us a way of combining two or more types of algebraic structure expressed as monads.

E.g. monoids and abelian groups ⟶ rings

Question

What's a distributive law for Lawvere theories?

- Lawvere theories correspond to finitary monads on **Set**.
- Lawvere theories are themselves monads in a certain bicategory.

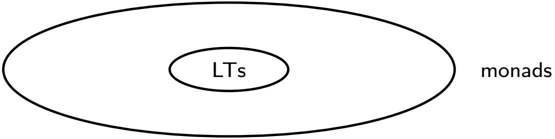
—So we can look for distributive laws between these monads.

1. Introduction

- A monad on $\mathcal{V}$ only gives algebras in $\mathcal{V}$.
- A Lawvere theory gives models in any finite-product category.

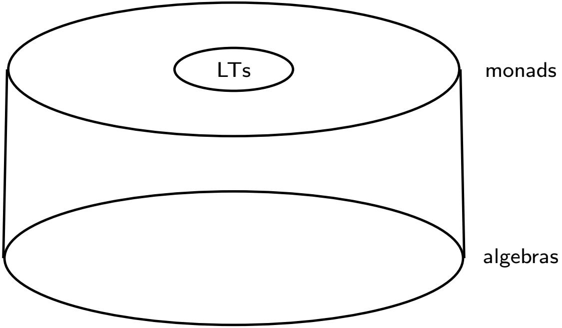
1. Introduction

- A monad on $\mathcal{V}$ only gives algebras in $\mathcal{V}$.
- A Lawvere theory gives models in any finite-product category.



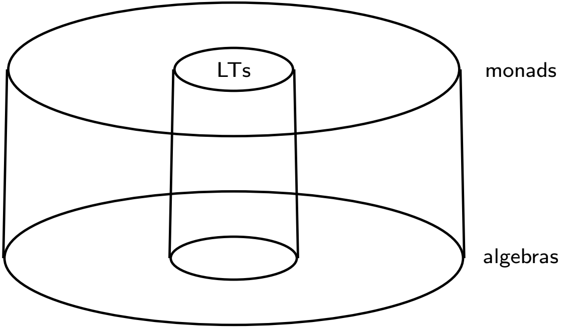
1. Introduction

- A monad on $\mathcal{V}$ only gives algebras in $\mathcal{V}$.
- A Lawvere theory gives models in any finite-product category.



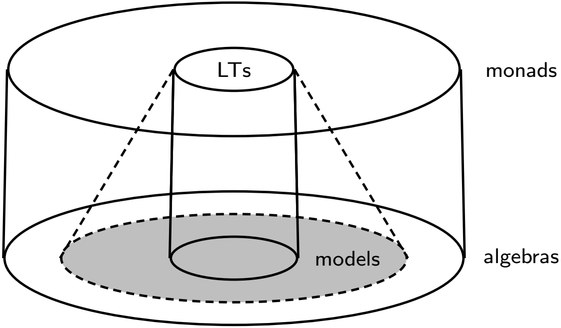
1. Introduction

- A monad on $\mathcal{V}$ only gives algebras in $\mathcal{V}$.
- A Lawvere theory gives models in any finite-product category.



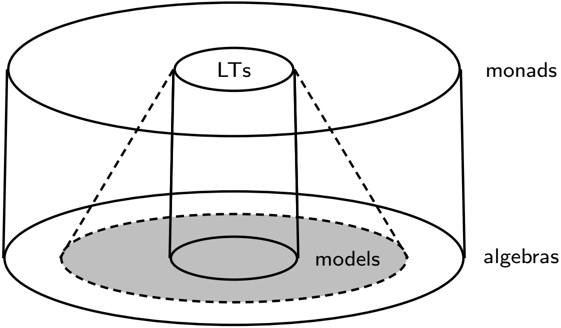
1. Introduction

- A monad on $\mathcal{V}$ only gives algebras in $\mathcal{V}$.
- A Lawvere theory gives models in any finite-product category.



1. Introduction

- A monad on $\mathcal{V}$ only gives algebras in $\mathcal{V}$.
- A Lawvere theory gives models in any finite-product category.



Example

Distributive law for monoids over abelian groups
$\longrightarrow$ rings internal to *any* finite-product category $\mathcal{V}$.

2. Lawvere theories

2. Lawvere theories

Idea

Encapsulate an algebraic theory in a category $\mathbb{L}$.

2. Lawvere theories

Idea

Encapsulate an algebraic theory in a category $\mathbb{L}$.

- The objects of $\mathbb{L}$ are the natural numbers, our *arities*.
- A morphism $k \longrightarrow 1$ is an operation of arity k.
- A morphism $k \longrightarrow m$ is m operations of arity k.

2. Lawvere theories

Idea

Encapsulate an algebraic theory in a category $\mathbb{L}$.

- The objects of $\mathbb{L}$ are the natural numbers, our *arities*.
- A morphism $k \longrightarrow 1$ is an operation of arity k.
- A morphism $k \longrightarrow m$ is m operations of arity k.

We use $\mathbb{F}$ a skeleton of **FinSet** (finite sets and functions).

2. Lawvere theories

Idea

Encapsulate an algebraic theory in a category $\mathbb{L}$.

- The objects of $\mathbb{L}$ are the natural numbers, our *arities*.
- A morphism $k \longrightarrow 1$ is an operation of arity k.
- A morphism $k \longrightarrow m$ is m operations of arity k.

We use $\mathbb{F}$ a skeleton of **FinSet** (finite sets and functions).

Definition

A Lawvere theory is a small category $\mathbb{L}$ with finite products, equipped with a strict identity-on-objects functor

$$\mathbb{F}^{\mathrm{op}} \longrightarrow \mathbb{L}.$$

2. Lawvere theories

Idea

Encapsulate an algebraic theory in a category $\mathbb{L}$.

- The objects of $\mathbb{L}$ are the natural numbers, our *arities*.
- A morphism $k \longrightarrow 1$ is an operation of arity k.
- A morphism $k \longrightarrow m$ is m operations of arity k.

We use $\mathbb{F}$ a skeleton of **FinSet** (finite sets and functions).

Definition

A Lawvere theory is a small category $\mathbb{L}$ with finite products, equipped with a strict identity-on-objects functor

$$\mathbb{F}^{\mathrm{op}} \longrightarrow \mathbb{L}.$$

Note: in $\mathbb{F}^{\mathrm{op}}$ the object m is the *product* of m copies of 1.

Note

We are allowed to forget and repeat variables.

2. Lawvere theories

Note

We are allowed to forget and repeat variables.

Example

2-ary operations in the theory of monoids

Note

We are allowed to forget and repeat variables.

Example

2-ary operations in the theory of monoids

- (non-Σ) operads: only one i.e. ab

2. Lawvere theories

Note

We are allowed to forget and repeat variables.

Example

2-ary operations in the theory of monoids

- (non-Σ) operads: only one i.e. ab
- Lawvere theory: $ab, a, a^2, b, b^2,\ aba,\ ab^3a^5, \ldots$

2. Lawvere theories

Note

We are allowed to forget and repeat variables.

Example

2-ary operations in the theory of monoids

- (non-Σ) operads: only one i.e. ab
- Lawvere theory: $ab, a, a^2, b, b^2,\ aba,\ ab^3a^5, \ldots$
  i.e. everything in the free monad on $\{a, b\}$.

2. Lawvere theories

Note

We are allowed to forget and repeat variables.

Example

2-ary operations in the theory of monoids

- (non-Σ) operads: only one i.e. ab
- Lawvere theory: $ab, a, a^2, b, b^2,\ aba,\ ab^3a^5, \ldots$
  i.e. everything in the free monad on $\{a, b\}$.

A morphism $3 \longrightarrow 2$ is two 3-ary operations e.g.

$$(ab, a^3), (a^2b, abc), \ldots$$

Note

We are allowed to forget and repeat variables.

Example

2-ary operations in the theory of monoids

- (non-Σ) operads: only one i.e. ab
- Lawvere theory: $ab, a, a^2, b, b^2,\ aba,\ ab^3a^5, \ldots$ i.e. everything in the free monad on $\{a, b\}$.

A morphism $3 \longrightarrow 2$ is two 3-ary operations e.g.

$$(ab, a^3), (a^2b, abc), \ldots$$

Composition: $3 \xrightarrow{\{ab, a^3\}} 2 \xrightarrow{x^2y} 1$

2. Lawvere theories

Note

We are allowed to forget and repeat variables.

Example

2-ary operations in the theory of monoids

- (non-Σ) operads: only one i.e. ab
- Lawvere theory: $ab, a, a^2, b, b^2,\ aba,\ ab^3a^5, \ldots$
  i.e. everything in the free monad on $\{a, b\}$.

A morphism $3 \longrightarrow 2$ is two 3-ary operations e.g.

$$(ab, a^3), (a^2b, abc), \ldots$$

Composition: $\underbrace{3 \xrightarrow{\{ab, a^3\}} 2 \xrightarrow{x^2y} 1}_{ab.ab.a^3}$

2. Lawvere theories

We have many arities for the “same” operation.

arity	operation
3	$a, b, c \longrightarrow abc$
4	$a, b, c, d \longrightarrow abc$
5	$a, b, c, d, e \longrightarrow abc$
$\vdots$	

2. Lawvere theories

We have many arities for the “same” operation.

arity		operation	
3	a, b, c	$\longrightarrow$	abc
4	a, b, c, d	$\longrightarrow$	abc
5	a, b, c, d, e	$\longrightarrow$	abc
$\vdots$			

$$\begin{tikzcd}
3 \arrow[r] & 1 \\
4 \arrow[u, dashed] \arrow[ur] & \\
5 \arrow[u, dashed] \arrow[uur] & \\
\vdots \arrow[u, dashed] &
\end{tikzcd}$$

These are all related by forgetting variables i.e. via projections in $\mathbb{F}^{\mathrm{op}}$.

Generalisations

Generalisations

- use $\mathbb{F} = \mathbf{FinSet}$ instead of a skeleton

2. Lawvere theories

Generalisations

- use $\mathbb{F} = \mathbf{FinSet}$ instead of a skeleton
- put $\mathcal{P}$ = "free finite product category" 2-monad
  note that $\mathbf{FinSet}^{\mathrm{op}}$ is $\mathcal{P}1$
  —could use $\mathcal{P}\mathbb{A}$ to get "typed" theory

2. Lawvere theories

Generalisations

- use $\mathbb{F} = \mathbf{FinSet}$ instead of a skeleton
- put $\mathcal{P} =$ "free finite product category" 2-monad
  note that $\mathbf{FinSet}^{\mathrm{op}}$ is $\mathcal{P}1$
  —could use $\mathcal{P}\mathbb{A}$ to get "typed" theory
- could just say a Lawvere theory is *any* finite product category $\mathbb{C}$

2. Lawvere theories

Generalisations

- use $\mathbb{F} = \mathbf{FinSet}$ instead of a skeleton
- put $\mathcal{P} =$ “free finite product category” 2-monad
  note that $\mathbf{FinSet}^{\mathrm{op}}$ is $\mathcal{P}1$
  —could use $\mathcal{P}\mathbb{A}$ to get “typed” theory
- could just say a Lawvere theory is *any* finite product category $\mathbb{C}$
- could do finite limits instead of just products.

2. Lawvere theories

Models $\equiv$ algebras

A model for $\mathbb{L}$ in a finite-product category $\mathcal{C}$ is a finite-product preserving functor

$$\begin{tikzcd}
\mathbb{L} \arrow[r] & \mathcal{C}
\end{tikzcd}$$

2. Lawvere theories

Models ≡ algebras

A model for $\mathbb{L}$ in a finite-product category $\mathcal{C}$ is a finite-product preserving functor

$$\mathbb{L} \longrightarrow \mathcal{C}$$

Idea

$$1 \longmapsto A \in \mathcal{C} \quad \text{underlying data}$$

Models ≡ algebras

A model for $\mathbb{L}$ in a finite-product category $\mathcal{C}$ is a finite-product preserving functor

$$\mathbb{L} \longrightarrow \mathcal{C}$$

Idea

$$\begin{array}{lcll} 1 & \longmapsto & A \in \mathcal{C} & \text{underlying data} \\ k & \longmapsto & A^k & \end{array}$$

2. Lawvere theories

Models $\equiv$ algebras

A model for $\mathbb{L}$ in a finite-product category $\mathcal{C}$ is a finite-product preserving functor

$$\mathbb{L} \longrightarrow \mathcal{C}$$

Idea

$$\begin{tikzcd} 1 \arrow[r, mapsto] & A \in \mathcal{C} \end{tikzcd} \quad \text{underlying data}$$

$$\begin{tikzcd} k \arrow[r, mapsto] & A^k \end{tikzcd}$$

operation of arity k

$$\begin{tikzcd} k \arrow[d] & & A^k \arrow[d] \\ 1 \arrow[rr, phantom, "\longmapsto" description, pos=0.5, yshift=4ex] & & A \end{tikzcd}$$

Lawvere theories vs monads

Lawvere theories vs monads

Lawvere theory	monad

Lawvere theories vs monads

Lawvere theory		monad
morphism $k \longrightarrow 1$	"k-ary operation"	

Lawvere theories vs monads

set of k elements

Lawvere theory		monad
morphism $k \longrightarrow 1$	“k-ary operation”	element of $T([k])$

Lawvere theories vs monads

set of k elements

Lawvere theory		monad
morphism $k \longrightarrow 1$	“k-ary operation”	element of $T([k])$ i.e. $1 \longrightarrow T([k]) \in \mathbf{Set}$

Lawvere theories vs monads

set of k elements

Lawvere theory		monad
morphism $k \longrightarrow 1$	"k-ary operation"	element of $T([k])$ i.e. $1 \longrightarrow T([k]) \in \mathbf{Set}$
morphism $k \longrightarrow m$	"m operations of arity k"	

Lawvere theories vs monads

set of k elements

Lawvere theory		monad
morphism $k \longrightarrow 1$	"k-ary operation"	element of $T([k])$ i.e. $1 \longrightarrow T([k]) \in \mathbf{Set}$
morphism $k \longrightarrow m$	"m operations of arity k"	m elements of $T([k])$

Lawvere theories vs monads

set of k elements

Lawvere theory		monad
morphism $k \longrightarrow 1$	“k-ary operation”	element of $T([k])$ i.e. $1 \longrightarrow T([k]) \in \mathbf{Set}$
morphism $k \longrightarrow m$	“m operations of arity k”	m elements of $T([k])$ i.e. $[m] \longrightarrow T([k]) \in \mathbf{Set}$

Lawvere theories vs monads

set of k elements

Lawvere theory		monad
morphism $k \longrightarrow 1$	“k-ary operation”	element of $T([k])$ i.e. $1 \longrightarrow T([k]) \in \mathbf{Set}$
morphism $k \longrightarrow m$	“m operations of arity k”	m elements of $T([k])$ i.e. $[m] \longrightarrow T([k]) \in \mathbf{Set}$ i.e. $[m] \longrightarrow [k] \in \mathbf{Kl}T$

Lawvere theories vs monads

set of k elements

Lawvere theory		monad
morphism $k \longrightarrow 1$	“k-ary operation”	element of $T([k])$ i.e. $1 \longrightarrow T([k]) \in \mathbf{Set}$
morphism $k \longrightarrow m$	“m operations of arity k”	m elements of $T([k])$ i.e. $[m] \longrightarrow T([k]) \in \mathbf{Set}$ i.e. $[m] \longrightarrow [k] \in \mathbf{Kl}T$

Idea

Lawvere theories are related to monads via the Kleisli category.

2. Lawvere theories

Definition

Monad T on $\mathbf{Set} \longrightarrow$ Lawvere theory $\mathbb{L}_T$

$$\mathbb{L}_T \;=\; \text{full subcategory of } (\mathbf{Kl}T)^{\mathrm{op}} \text{ whose objects are \emph{finite} sets.}$$

2. Lawvere theories

Definition

Monad T on **Set** ⟶ Lawvere theory $\mathbb{L}_T$

$$\mathbb{L}_T \;=\; \text{full subcategory of } (\mathbf{Kl}T)^{\mathrm{op}} \text{ whose objects are } \textit{finite} \text{ sets.}$$

Lawvere theory $\mathbb{L}$ ⟶ monad $T_{\mathbb{L}}$ on **Set**

$$T_{\mathbb{L}}X = \int^{n \in \mathbb{F}^{\mathrm{op}}} \mathbb{L}(n,1) \times X^n.$$

Definition

Monad T on **Set** $\longrightarrow$ Lawvere theory $\mathbb{L}_T$

$$\mathbb{L}_T \quad = \quad \text{full subcategory of } (\mathbf{Kl}\,T)^{\text{op}}$$
$$\text{whose objects are \textit{finite} sets.}$$

Lawvere theory $\mathbb{L}$ $\longrightarrow$ monad $T_{\mathbb{L}}$ on **Set**

$$T_{\mathbb{L}}X = \int^{n \in \mathbb{F}^{\text{op}}} \mathbb{L}(n,1) \times X^n.$$

Theorem

This gives a correspondence between Lawvere theories and *finitary* monads on **Set**.

3. Distributive laws for monads

3. Distributive laws for monads

Idea

Given monads S and T on $\mathcal{C}$, can we make TS into a monad?

3. Distributive laws for monads

Idea

Given monads S and T on $\mathcal{C}$, can we make TS into a monad?

$$\begin{tikzcd} TSTS \arrow[r, "?"] & TTSS \arrow[r, "\mu^T \mu^S"] & TS \end{tikzcd}$$

3. Distributive laws for monads

Idea

Given monads S and T on $\mathcal{C}$, can we make TS into a monad?

$$\begin{tikzcd} TSTS \arrow[r, "?"] & TTSS \arrow[r, "\mu^T \mu^S"] & TS \end{tikzcd}$$

Definition (Beck)

A distributive law of monads S over T consists of a natural transformation

$$\lambda : ST \Rightarrow TS$$

satisfying some axioms.

3. Distributive laws for monads

Idea

Given monads S and T on $\mathcal{C}$, can we make TS into a monad?

$$\begin{tikzcd} TSTS \arrow[r, "?"] & TTSS \arrow[r, "\mu^T \mu^S"] & TS \end{tikzcd}$$

Definition (Beck)

A distributive law of monads S over T consists of a natural transformation

$$\lambda : ST \Rightarrow TS$$

satisfying some axioms.

- **Formal theory of monads (Street)**
  Do this inside any bicategory, not just **Cat**.

3. Distributive laws for monads

Idea

Given monads S and T on $\mathcal{C}$, can we make TS into a monad?

$$\begin{tikzcd} TSTS \arrow[r, "?"] & TTSS \arrow[r, "\mu^T \mu^S"] & TS \end{tikzcd}$$

Definition (Beck)

A distributive law of monads S over T consists of a natural transformation

$$\lambda : ST \Rightarrow TS$$

satisfying some axioms.

- **Formal theory of monads (Street)**
  Do this inside any bicategory, not just **Cat**.
- **Iterated distributive laws (Cheng)**
  Combine n monads with distributive laws and Yang-Baxter condition.

3. Distributive laws for monads

Examples

monoid + abelian group → ring

horizontal composition + vertical composition → 2-category

3. Distributive laws for monads

Examples

monoid + abelian group $\longrightarrow$ ring

horizontal composition + vertical composition $\longrightarrow$ 2-category

Or combining more structures:

$$\left.\begin{array}{c}\text{0-composition}\\ +\\ \text{1-composition}\\ +\\ \vdots\\ +\\ (n-1)\text{-composition}\end{array}\right\}\longrightarrow n\text{-category}$$

A point of view

- The monad TS says we can express all structure as "S-structure followed by T-structure".

3. Distributive laws for monads

A point of view

- The monad TS says we can express all structure as "S-structure followed by T-structure".
- The distributive law $ST \longrightarrow TS$ says "if we had it the other way round we could switch it over".

3. Distributive laws for monads

A point of view

- The monad TS says we can express all structure as "S-structure followed by T-structure".
- The distributive law $ST \longrightarrow TS$ says "if we had it the other way round we could switch it over".

For Lawvere theories

We want a way of combining $\mathbb{A}$ and $\mathbb{B}$ to give $\mathbb{BA}$ corresponding to a distributive law of monads

$$T_{\mathbb{A}}T_{\mathbb{B}} \longrightarrow T_{\mathbb{B}}T_{\mathbb{A}}$$

3. Distributive laws for monads

A point of view

- The monad TS says we can express all structure as "S-structure followed by T-structure".
- The distributive law $ST \longrightarrow TS$ says "if we had it the other way round we could switch it over".

For Lawvere theories

We want a way of combining $\mathbb{A}$ and $\mathbb{B}$ to give $\mathbb{BA}$ corresponding to a distributive law of monads

$$T_{\mathbb{A}}T_{\mathbb{B}} \longrightarrow T_{\mathbb{B}}T_{\mathbb{A}}$$

with

$$T_{\mathbb{B}}T_{\mathbb{A}} = T_{\mathbb{BA}}$$

4. Three ways to do it

1. Factorisation systems over $\mathbb{F}^{\mathrm{op}}$.

   —Rosebrugh and Wood, Distributive laws and factorization (JPAA 2002)

4. Three ways to do it

1. Factorisation systems over $\mathbb{F}^{\mathrm{op}}$.

   —Rosebrugh and Wood, Distributive laws and factorization (JPAA 2002)

2. Profunctors internal to **Mon**.

   —Lack, Composing props (TAC 2004)

   —Akhvlediani, Composing Lawvere theories (CT2010)

4. Three ways to do it

1. Factorisation systems over $\mathbb{F}^{\mathrm{op}}$.

   —Rosebrugh and Wood, Distributive laws and factorization (JPAA 2002)

2. Profunctors internal to **Mon**.

   —Lack, Composing props (TAC 2004)

   —Akhvlediani, Composing Lawvere theories (CT2010)

3. Kleisli bicategory of $\mathcal{P}$ on profunctors

   —Hyland, Distributive laws (CLP 2010)

4. Three ways to do it

1. Factorisation systems over $\mathbb{F}^{\mathbf{op}}$

4. Three ways to do it

1. Factorisation systems over $\mathbb{F}^{\mathrm{op}}$

In the composite theory $\mathbb{B}\mathbb{A}$ every morphism can be expressed as a composite

$$\begin{tikzcd} {} \arrow[r, "\in\mathbb{A}"] & {} \arrow[r, "\in\mathbb{B}"] & {} \end{tikzcd}$$

4. Three ways to do it

1. Factorisation systems over $\mathbb{F}^{\mathrm{op}}$

In the composite theory $\mathbb{B}\mathbb{A}$ every morphism can be expressed as a composite

$$\xrightarrow{\in\mathbb{A}} \quad \xrightarrow{\in\mathbb{B}}$$

For example: × and +

The composite 3-ary operation $a(b+c)$ can be expressed as

4. Three ways to do it

1. Factorisation systems over $\mathbb{F}^{\mathrm{op}}$

In the composite theory $\mathbb{B}\mathbb{A}$ every morphism can be expressed as a composite

$$\begin{tikzcd} {} \arrow[r, "\in\mathbb{A}"] & {} \arrow[r, "\in\mathbb{B}"] & {} \end{tikzcd}$$

For example: $\times$ and $+$

The composite 3-ary operation $a(b+c)$ can be expressed as

$$\begin{tikzcd} 3 \arrow[dr, "{ab, ac}"'] & & 1 \\ & 2 \arrow[ur, "x+y"'] & \end{tikzcd}$$

4. Three ways to do it

1. Factorisation systems over $\mathbb{F}^{\mathbf{op}}$

In the composite theory $\mathbb{B}\mathbb{A}$ every morphism can be expressed as a composite

$$\xrightarrow{\in\mathbb{A}} \quad \xrightarrow{\in\mathbb{B}}$$

For example: × and +

The composite 3-ary operation $a(b+c)$ can be expressed as

$$\begin{tikzcd}
 & 3 \arrow[dr, "x+y"] & \\
3 \arrow[ur, "{ab, ac, ab^2c}"] \arrow[dr, "{ab, ac}"'] & & 1 \\
 & 2 \arrow[ur, "x+y"'] &
\end{tikzcd}$$

4. Three ways to do it

1. Factorisation systems over $\mathbb{F}^{\mathrm{op}}$

In the composite theory $\mathbb{BA}$ every morphism can be expressed as a composite

$$\xrightarrow{\in\mathbb{A}} \quad \xrightarrow{\in\mathbb{B}}$$

For example: × and +

The composite 3-ary operation $a(b+c)$ can be expressed as

$$\begin{tikzcd}
 & 3 \arrow[dd, dashed, "p_1, p_2"'] \arrow[dr, "x+y"] & \\
3 \arrow[ur, "{ab, ac, ab^2c}"] \arrow[dr, "{ab, ac}"'] & & 1 \\
 & 2 \arrow[ur, "x+y"'] &
\end{tikzcd}$$

4. Three ways to do it

1. Factorisation systems over $\mathbb{F}^{\mathrm{op}}$

In the composite theory $\mathbb{BA}$ every morphism can be expressed as a composite

$$\xrightarrow{\in\mathbb{A}} \quad \xrightarrow{\in\mathbb{B}}$$

For example: × and +

The composite 3-ary operation $a(b+c)$ can be expressed as

$$\begin{tikzcd}
 & 3 \arrow[dr, "x+y"] \arrow[dd, dashed, "{p_1, p_2}"'] & \\
3 \arrow[ur, "{ab, ac, \color{red}{ab^2c}}"] \arrow[dr, "{ab, ac}"'] & & 1 \\
 & 2 \arrow[ur, "x+y"'] &
\end{tikzcd}$$

—factorisations are only unique up to morphisms in $\mathbb{F}^{\mathrm{op}}$.

4. Three ways to do it

Appealing fact (Rosebrugh and Wood)

Strict factorisation systems are distributive laws in **Span**.

4. Three ways to do it

Appealing fact (Rosebrugh and Wood)

Strict factorisation systems are distributive laws in **Span**.

- $\mathbb{A}$ and $\mathbb{B}$ are categories i.e. monads in **Span**.
- $\mathbb{A}\mathbb{B} \longrightarrow \mathbb{B}\mathbb{A}$ makes $\mathbb{B}\mathbb{A}$ into a monad in **Span** i.e. a category.

4. Three ways to do it

Appealing fact (Rosebrugh and Wood)

Strict factorisation systems are distributive laws in **Span**.

- $\mathbb{A}$ and $\mathbb{B}$ are categories i.e. monads in **Span**.
- $\mathbb{A}\mathbb{B} \longrightarrow \mathbb{B}\mathbb{A}$ makes $\mathbb{B}\mathbb{A}$ into a monad in **Span** i.e. a category.

It is the pullback

$$\begin{tikzcd}
 & A \arrow[dl] \arrow[dr] & \\
\mathbf{ob}\,\mathbb{F} & & \mathbf{ob}\,\mathbb{F}
\end{tikzcd}$$

4. Three ways to do it

Appealing fact (Rosebrugh and Wood)

Strict factorisation systems are distributive laws in **Span**.

- $\mathbb{A}$ and $\mathbb{B}$ are categories i.e. monads in **Span**.
- $\mathbb{A}\mathbb{B} \longrightarrow \mathbb{B}\mathbb{A}$ makes $\mathbb{B}\mathbb{A}$ into a monad in **Span** i.e. a category.

It is the pullback

$$\begin{tikzcd}[column sep=small]
 & A \arrow[dl] \arrow[dr] & & B \arrow[dl] \arrow[dr] & \\
\mathbf{ob}\,\mathbb{F} & & \mathbf{ob}\,\mathbb{F} & & \mathbf{ob}\,\mathbb{F}
\end{tikzcd}$$

4. Three ways to do it

Appealing fact (Rosebrugh and Wood)

Strict factorisation systems are distributive laws in **Span**.

- $\mathbb{A}$ and $\mathbb{B}$ are categories i.e. monads in **Span**.
- $\mathbb{A}\mathbb{B} \longrightarrow \mathbb{B}\mathbb{A}$ makes $\mathbb{B}\mathbb{A}$ into a monad in **Span** i.e. a category.

It is the pullback

$$\begin{tikzcd}
 & & \cdot \arrow[dl] \arrow[dr] \arrow[dd, phantom, very near start, "\lrcorner" rotate=-45] & & \\
 & A \arrow[dl] \arrow[dr] & & B \arrow[dl] \arrow[dr] & \\
\mathbf{ob}\,\mathbb{F} & & \mathbf{ob}\,\mathbb{F} & & \mathbf{ob}\,\mathbb{F}
\end{tikzcd}$$

4. Three ways to do it

Appealing fact (Rosebrugh and Wood)
Strict factorisation systems are distributive laws in **Span**.

- $\mathbb{A}$ and $\mathbb{B}$ are categories i.e. monads in **Span**.
- $\mathbb{A}\mathbb{B} \longrightarrow \mathbb{B}\mathbb{A}$ makes $\mathbb{B}\mathbb{A}$ into a monad in **Span** i.e. a category.

It is the pullback

$$\begin{tikzcd}
 & & \cdot \arrow[dl] \arrow[dr] \arrow[d, phantom, "\vee", very near start] & & \\
 & A \arrow[dl] \arrow[dr] & & B \arrow[dl] \arrow[dr] & \\
\mathbf{ob}\,\mathbb{F} & & \mathbf{ob}\,\mathbb{F} & & \mathbf{ob}\,\mathbb{F}
\end{tikzcd}$$

composable pairs

$$k \xrightarrow{\in\mathbb{A}} l \xrightarrow{\in\mathbb{B}} m$$

4. Three ways to do it

Appealing fact (Rosebrugh and Wood)

Strict factorisation systems are distributive laws in **Span**.

- $\mathbb{A}$ and $\mathbb{B}$ are categories i.e. monads in **Span**.
- $\mathbb{A}\mathbb{B} \longrightarrow \mathbb{B}\mathbb{A}$ makes $\mathbb{B}\mathbb{A}$ into a monad in **Span**
  i.e. a category.

It is the pullback

$$\begin{tikzcd}
 & & \cdot \arrow[dl] \arrow[dr] \arrow[d, phantom, "\lrcorner" rotate=-45] & & \\
 & A \arrow[dl] \arrow[dr] & {} & B \arrow[dl] \arrow[dr] & \\
\mathbf{ob}\,\mathbb{F} & & \mathbf{ob}\,\mathbb{F} & & \mathbf{ob}\,\mathbb{F}
\end{tikzcd}$$

composable pairs

$$k \xrightarrow{\in\mathbb{A}} l \xrightarrow{\in\mathbb{B}} m$$

The distributive law tells us how to re-express a pair

$$k \xrightarrow{\in\mathbb{B}} l \xrightarrow{\in\mathbb{A}} m \qquad \text{as} \qquad k \xrightarrow{\in\mathbb{A}} l' \xrightarrow{\in\mathbb{B}} m$$

4. Three ways to do it

RW define *distributive laws over* $\mathcal{I}$ for $\mathcal{I}$ a groupoid
—ensures equivalence relation on composable pairs.

4. Three ways to do it

RW define *distributive laws over* $\mathcal{J}$ for $\mathcal{J}$ a groupoid —ensures equivalence relation on composable pairs.

However instead we can *generate* an equivalence relation.

4. Three ways to do it

RW define *distributive laws over* $\mathcal{J}$ for $\mathcal{J}$ a groupoid
—ensures equivalence relation on composable pairs.

However instead we can *generate* an equivalence relation.

Idea

4. Three ways to do it

RW define *distributive laws over* $\mathbb{J}$ for $\mathbb{J}$ a groupoid
—ensures equivalence relation on composable pairs.

However instead we can *generate* an equivalence relation.

Idea

Our original pullback

$$\begin{tikzcd}
 & & B \otimes A \arrow[dl] \arrow[dr] \arrow[d, phantom, very near start, "\vee"] & & \\
 & A \arrow[dl] \arrow[dr] & {} & B \arrow[dl] \arrow[dr] & \\
\mathbf{ob}\,\mathbb{F} & & \mathbf{ob}\,\mathbb{F} & & \mathbf{ob}\,\mathbb{F}
\end{tikzcd}$$

ignored the fact that $\mathbb{F}^{\mathrm{op}}$ is in both $\mathbb{A}$ and $\mathbb{B}$.

4. Three ways to do it

RW define *distributive laws over* $\mathcal{J}$ for $\mathcal{J}$ a groupoid
—ensures equivalence relation on composable pairs.

However instead we can *generate* an equivalence relation.

Idea

Our original pullback

$$\begin{tikzcd}
 & & B \otimes A \arrow[dl] \arrow[dr] \arrow[dd, phantom, very near start, "\lrcorner" {rotate=-45}] & & \\
 & A \arrow[dl] \arrow[dr] & & B \arrow[dl] \arrow[dr] & \\
\mathbf{ob}\,\mathbb{F} & & \mathbf{ob}\,\mathbb{F} & & \mathbf{ob}\,\mathbb{F}
\end{tikzcd}$$

ignored the fact that $\mathbb{F}^{\mathrm{op}}$ is in both $\mathbb{A}$ and $\mathbb{B}$.

So we want a coequaliser

$$\begin{tikzcd}
\mathbb{B} \otimes \mathbb{F}^{\mathrm{op}} \otimes \mathbb{A} \arrow[r, shift left, "\textsf{absorb } \mathbb{F}^{\mathrm{op}} \textsf{ into } \mathbb{A}"] \arrow[r, shift right, "\textsf{absorb } \mathbb{F}^{\mathrm{op}} \textsf{ into } \mathbb{B}"'] & \mathbb{B} \otimes \mathbb{A} \arrow[r] & \mathbb{B} \otimes_{\mathbb{F}^{\mathrm{op}}} \mathbb{A}
\end{tikzcd}$$

—looks like bimodules.

4. Three ways to do it

Definition 1

A distributive law of Lawvere theories $\mathbb{A}$ over $\mathbb{B}$ is a factorisation system over $\mathbb{F}^{\mathrm{op}}$ on the composite $\mathbb{B} \otimes \mathbb{A}$ in **Span**.

4. Three ways to do it

Definition 1

A distributive law of Lawvere theories $\mathbb{A}$ over $\mathbb{B}$ is a factorisation system over $\mathbb{F}^{\text{op}}$ on the composite $\mathbb{B} \otimes \mathbb{A}$ in **Span**.

$\equiv$ a distributive law of $\mathbb{A}$ over $\mathbb{B}$ expressed as monads in

$$\mathbf{Bim}(\mathbf{Span})$$

4. Three ways to do it

Definition 1

A distributive law of Lawvere theories $\mathbb{A}$ over $\mathbb{B}$ is a factorisation system over $\mathbb{F}^{\mathrm{op}}$ on the composite $\mathbb{B} \otimes \mathbb{A}$ in **Span**.

$\equiv$ a distributive law of $\mathbb{A}$ over $\mathbb{B}$ expressed as monads in

$$\mathbf{Bim}(\mathbf{Span}) \simeq \mathbf{Prof}.$$

4. Three ways to do it

Definition 1

A distributive law of Lawvere theories $\mathbb{A}$ over $\mathbb{B}$ is a factorisation system over $\mathbb{F}^{\mathrm{op}}$ on the composite $\mathbb{B} \otimes \mathbb{A}$ in **Span**.

$\equiv$ a distributive law of $\mathbb{A}$ over $\mathbb{B}$ expressed as monads in

$$\mathbf{Bim}(\mathbf{Span}) \simeq \mathbf{Prof}.$$

Aside on profunctors

4. Three ways to do it

Definition 1

A distributive law of Lawvere theories $\mathbb{A}$ over $\mathbb{B}$ is a factorisation system over $\mathbb{F}^{\mathrm{op}}$ on the composite $\mathbb{B} \otimes \mathbb{A}$ in **Span**.

$\equiv$ a distributive law of $\mathbb{A}$ over $\mathbb{B}$ expressed as monads in

$$\mathbf{Bim}(\mathbf{Span}) \simeq \mathbf{Prof}.$$

Aside on profunctors

Given categories $\mathbb{C}$ and $\mathbb{D}$, a profunctor $\mathbb{C} \xrightarrow{F} \mathbb{D}$ is a functor

$$\mathbb{D}^{\mathrm{op}} \times \mathbb{C} \xrightarrow{F} \mathbf{Set}.$$

4. Three ways to do it

Definition 1

A distributive law of Lawvere theories $\mathbb{A}$ over $\mathbb{B}$ is a factorisation system over $\mathbb{F}^{\mathrm{op}}$ on the composite $\mathbb{B} \otimes \mathbb{A}$ in **Span**.

$\equiv$ a distributive law of $\mathbb{A}$ over $\mathbb{B}$ expressed as monads in

$$\mathbf{Bim}(\mathbf{Span}) \simeq \mathbf{Prof}.$$

Aside on profunctors

Given categories $\mathbb{C}$ and $\mathbb{D}$, a profunctor $\mathbb{C} \xrightarrow{F} \mathbb{D}$ is a functor

$$\mathbb{D}^{\mathrm{op}} \times \mathbb{C} \xrightarrow{F} \mathbf{Set}.$$

A monad $\mathbb{C} \xrightarrow{A} \mathbb{C} \in \mathbf{Prof}$ corresponds to a category $\mathbb{A}$ equipped with an identity-on-objects functor

$$\mathbb{C} \longrightarrow \mathbb{A}.$$

4. Three ways to do it

Definition 1

A distributive law of Lawvere theories $\mathbb{A}$ over $\mathbb{B}$ is a factorisation system over $\mathbb{F}^{\mathrm{op}}$ on the composite $\mathbb{B} \otimes \mathbb{A}$ in **Span**.

$\equiv$ a distributive law of $\mathbb{A}$ over $\mathbb{B}$ expressed as monads in

$$\mathbf{Bim}(\mathbf{Span}) \simeq \mathbf{Prof}.$$

Aside on profunctors

Given categories $\mathbb{C}$ and $\mathbb{D}$, a profunctor $\mathbb{C} \overset{F}{\longrightarrow\!\!\!\!\!\!\!+\;\;} \mathbb{D}$ is a functor

$$\mathbb{D}^{\mathrm{op}} \times \mathbb{C} \xrightarrow{F} \mathbf{Set}.$$

A monad $\mathbb{C} \overset{A}{\longrightarrow\!\!\!\!\!\!\!+\;\;} \mathbb{C} \in \mathbf{Prof}$ corresponds to a category $\mathbb{A}$ equipped with an identity-on-objects functor

$$\mathbb{C} \longrightarrow \mathbb{A}.$$

So Lawvere theories arise as particular monads on $\mathbb{F}^{\mathrm{op}}$.

2. Prof(Mon) —internal profunctors in monoids

2. **Prof(Mon)** —internal profunctors in monoids

A monad $\mathbb{C} \longrightarrow\!\!\!\!\!\!\!\!\mapsto \mathbb{C}$ is now a *monoidal* category $\mathbb{A}$ equipped with an identity-on-objects *monoidal* functor $\mathbb{C} \longrightarrow \mathbb{A}$.

2. **Prof(Mon)** —internal profunctors in monoids

A monad $\mathbb{C} \longrightarrow\!\!\!\!\!\!\!\!\mapsto \mathbb{C}$ is now a *monoidal* category $\mathbb{A}$ equipped with an identity-on-objects *monoidal* functor $\mathbb{C} \longrightarrow \mathbb{A}$.

So again Lawvere theories arise as particular monads on $\mathbb{F}^{\mathrm{op}}$.

2. **Prof**(**Mon**) —internal profunctors in monoids

A monad $\mathbb{C} \longrightarrow\!\!\!\!\!\!\!\!\!+\quad \mathbb{C}$ is now a *monoidal* category $\mathbb{A}$ equipped with an identity-on-objects *monoidal* functor $\mathbb{C} \longrightarrow \mathbb{A}$.

So again Lawvere theories arise as particular monads on $\mathbb{F}^{\mathrm{op}}$.

Definition 2

A distributive law of Lawvere theories $\mathbb{A}$ over $\mathbb{B}$ is a distributive law in the bicategory **Prof**(**Mon**).

2. **Prof**(**Mon**) —internal profunctors in monoids

A monad $\mathbb{C} \longrightarrow\!\!\!\!\!\!\!+\;\; \mathbb{C}$ is now a *monoidal* category $\mathbb{A}$ equipped with an identity-on-objects *monoidal* functor $\mathbb{C} \longrightarrow \mathbb{A}$.

So again Lawvere theories arise as particular monads on $\mathbb{F}^{\mathrm{op}}$.

Definition 2

A distributive law of Lawvere theories $\mathbb{A}$ over $\mathbb{B}$ is a distributive law in the bicategory **Prof**(**Mon**).

Theorem

Such a distributive law makes $\mathbb{B} \otimes_{\mathbb{F}^{\mathrm{op}}} \mathbb{A}$ into a Lawvere theory. i.e. if $\mathbb{A}$ and $\mathbb{B}$ are finite-product categories, so is $\mathbb{B} \otimes_{\mathbb{F}^{\mathrm{op}}} \mathbb{A}$.

2. **Prof**(**Mon**) —internal profunctors in monoids

A monad $\mathbb{C} \xrightarrow{\;\;+\;\;} \mathbb{C}$ is now a *monoidal* category $\mathbb{A}$ equipped with an identity-on-objects *monoidal* functor $\mathbb{C} \longrightarrow \mathbb{A}$.

So again Lawvere theories arise as particular monads on $\mathbb{F}^{\mathrm{op}}$.

Definition 2

A distributive law of Lawvere theories $\mathbb{A}$ over $\mathbb{B}$ is a distributive law in the bicategory **Prof**(**Mon**).

$$\mathbb{A} \otimes_{\mathbb{F}^{\mathrm{op}}} \mathbb{B} \xrightarrow{\lambda} \mathbb{B} \otimes_{\mathbb{F}^{\mathrm{op}}} \mathbb{A}$$

Theorem

Such a distributive law makes $\mathbb{B} \otimes_{\mathbb{F}^{\mathrm{op}}} \mathbb{A}$ into a Lawvere theory. i.e. if $\mathbb{A}$ and $\mathbb{B}$ are finite-product categories, so is $\mathbb{B} \otimes_{\mathbb{F}^{\mathrm{op}}} \mathbb{A}$.

2. **Prof(Mon)** —internal profunctors in monoids

A monad $\mathbb{C} \longrightarrow\!\!\!\!\!\!\!\!+\;\;\; \mathbb{C}$ is now a *monoidal* category $\mathbb{A}$ equipped with an identity-on-objects *monoidal* functor $\mathbb{C} \longrightarrow \mathbb{A}$.

So again Lawvere theories arise as particular monads on $\mathbb{F}^{\mathrm{op}}$.

Definition 2

A distributive law of Lawvere theories $\mathbb{A}$ over $\mathbb{B}$ is a distributive law in the bicategory **Prof(Mon)**.

$$\mathbb{A} \otimes_{\mathbb{F}^{\mathrm{op}}} \mathbb{B} \xrightarrow{\lambda} \mathbb{B} \otimes_{\mathbb{F}^{\mathrm{op}}} \mathbb{A}$$

Theorem

Such a distributive law makes $\mathbb{B} \otimes_{\mathbb{F}^{\mathrm{op}}} \mathbb{A}$ into a Lawvere theory. i.e. if $\mathbb{A}$ and $\mathbb{B}$ are finite-product categories, so is $\mathbb{B} \otimes_{\mathbb{F}^{\mathrm{op}}} \mathbb{A}$.

Proof

- Bare hands, or

2. Prof(Mon) —internal profunctors in monoids

A monad $\mathbb{C} \longrightarrow\!\!\!\!\!\!\!+\;\; \mathbb{C}$ is now a *monoidal* category $\mathbb{A}$ equipped with an identity-on-objects *monoidal* functor $\mathbb{C} \longrightarrow \mathbb{A}$.

So again Lawvere theories arise as particular monads on $\mathbb{F}^{\mathrm{op}}$.

Definition 2

A distributive law of Lawvere theories $\mathbb{A}$ over $\mathbb{B}$ is a distributive law in the bicategory **Prof**(**Mon**).

$$\mathbb{A} \otimes_{\mathbb{F}^{\mathrm{op}}} \mathbb{B} \xrightarrow{\lambda} \mathbb{B} \otimes_{\mathbb{F}^{\mathrm{op}}} \mathbb{A}$$

Theorem

Such a distributive law makes $\mathbb{B} \otimes_{\mathbb{F}^{\mathrm{op}}} \mathbb{A}$ into a Lawvere theory. i.e. if $\mathbb{A}$ and $\mathbb{B}$ are finite-product categories, so is $\mathbb{B} \otimes_{\mathbb{F}^{\mathrm{op}}} \mathbb{A}$.

Proof

- Bare hands, or
- The free finite-product category 2-monad on **Prof**.

4. Three ways to do it

3. Free finite-product category 2-monad approach.

4. Three ways to do it

3. Free finite-product category 2-monad approach.

- Let $\mathcal{P}$ be the free finite-product category 2-monad on **Cat**.
- $\mathcal{P}$ extends to **Prof** via a distributive law.
- Let $\mathbf{Prof}_{\mathcal{P}}$ be the Kleisli bicategory for the extended $\mathcal{P}$.

3. Free finite-product category 2-monad approach.

- Let $\mathcal{P}$ be the free finite-product category 2-monad on **Cat**.
- $\mathcal{P}$ extends to **Prof** via a distributive law.
- Let $\mathbf{Prof}_{\mathcal{P}}$ be the Kleisli bicategory for the extended $\mathcal{P}$.

Then monads on 1 in $\mathbf{Prof}_{\mathcal{P}}$ are *precisely* Lawvere theories.

3. Free finite-product category 2-monad approach.

- Let $\mathcal{P}$ be the free finite-product category 2-monad on **Cat**.
- $\mathcal{P}$ extends to **Prof** via a distributive law.
- Let $\mathbf{Prof}_{\mathcal{P}}$ be the Kleisli bicategory for the extended $\mathcal{P}$.

Then monads on 1 in $\mathbf{Prof}_{\mathcal{P}}$ are *precisely* Lawvere theories.

- A monad in $\mathbf{Prof}_{\mathcal{P}}$ is a profunctor $1 \longrightarrow\!\!\!\!\!\!\mapsto \mathcal{P}1$

3. Free finite-product category 2-monad approach.

- Let $\mathcal{P}$ be the free finite-product category 2-monad on **Cat**.
- $\mathcal{P}$ extends to **Prof** via a distributive law.
- Let $\mathbf{Prof}_{\mathcal{P}}$ be the Kleisli bicategory for the extended $\mathcal{P}$.

Then monads on 1 in $\mathbf{Prof}_{\mathcal{P}}$ are *precisely* Lawvere theories.

- A monad in $\mathbf{Prof}_{\mathcal{P}}$ is a profunctor $1 \xrightarrow{\;\;+\;\;} \mathcal{P}1$
  i.e. $\mathcal{P}1^{\mathrm{op}} \times 1 \longrightarrow \mathbf{Set}$

3. Free finite-product category 2-monad approach.

- Let $\mathcal{P}$ be the free finite-product category 2-monad on **Cat**.
- $\mathcal{P}$ extends to **Prof** via a distributive law.
- Let $\mathbf{Prof}_{\mathcal{P}}$ be the Kleisli bicategory for the extended $\mathcal{P}$.

$$\boxed{\text{Then monads on } 1 \text{ in } \mathbf{Prof}_{\mathcal{P}} \text{ are } \textit{precisely} \text{ Lawvere theories.}}$$

- A monad in $\mathbf{Prof}_{\mathcal{P}}$ is a profunctor $1 \xrightarrow{\;\;|\;\;} \mathcal{P}1$
  i.e. $\mathcal{P}1^{\mathrm{op}} \times 1 \longrightarrow \mathbf{Set}$
  i.e. $\mathbf{FinSet} \longrightarrow \mathbf{Set}$ a finitary monad.

3. Free finite-product category 2-monad approach.

- Let $\mathcal{P}$ be the free finite-product category 2-monad on **Cat**.
- $\mathcal{P}$ extends to **Prof** via a distributive law.
- Let $\mathbf{Prof}_{\mathcal{P}}$ be the Kleisli bicategory for the extended $\mathcal{P}$.

Then monads on 1 in $\mathbf{Prof}_{\mathcal{P}}$ are *precisely* Lawvere theories.

- A monad in $\mathbf{Prof}_{\mathcal{P}}$ is a profunctor $1 \longrightarrow\!\!\!\!\!\!\!+\;\; \mathcal{P}1$
  i.e. $\mathcal{P}1^{\mathrm{op}} \times 1 \longrightarrow \mathbf{Set}$
  i.e. $\mathbf{FinSet} \longrightarrow \mathbf{Set}$ a finitary monad.

Definition 3

3. Free finite-product category 2-monad approach.

- Let $\mathcal{P}$ be the free finite-product category 2-monad on **Cat**.
- $\mathcal{P}$ extends to **Prof** via a distributive law.
- Let $\mathbf{Prof}_{\mathcal{P}}$ be the Kleisli bicategory for the extended $\mathcal{P}$.

Then monads on 1 in $\mathbf{Prof}_{\mathcal{P}}$ are *precisely* Lawvere theories.

- A monad in $\mathbf{Prof}_{\mathcal{P}}$ is a profunctor $1 \xrightarrow{\;+\;} \mathcal{P}1$
  i.e. $\mathcal{P}1^{\mathrm{op}} \times 1 \longrightarrow \mathbf{Set}$
  i.e. $\mathbf{FinSet} \longrightarrow \mathbf{Set}$ a finitary monad.

Definition 3

A distributive law of Lawvere theories $\mathbb{A}$ over $\mathbb{B}$ is a distributive law in the bicategory $\mathbf{Prof}_{\mathcal{P}}$.

5. Comparison

5. Comparison

Claim

These three methods all give the same answer as a distributive law between the associated monads.

5. Comparison

Claim

These three methods all give the same answer as a distributive law between the associated monads.

Idea

5. Comparison

Claim

These three methods all give the same answer as a distributive law between the associated monads.

Idea

Compare

- Finitary monads $\mathbf{Set} \longrightarrow \mathbf{Set}$ in $\mathbf{CAT}$

5. Comparison

Claim
These three methods all give the same answer as a distributive law between the associated monads.

Idea

Compare

- Finitary monads $\mathbf{Set} \longrightarrow \mathbf{Set}$ in $\mathbf{CAT}$
- Lawvere theories as
  1. monads $\mathbb{F} \xrightarrow{\;\;+\;\;} \mathbb{F}$ in $\mathbf{Prof}$
  2. monads $\mathbb{F} \xrightarrow{\;\;+\;\;} \mathbb{F}$ in $\mathbf{Prof}(\mathbf{Mon})$
  3. monads $1 \xrightarrow{\;\;+\;\;} 1$ in $\mathbf{Prof}_{\mathcal{P}}$.

5. Comparison

Claim

These three methods all give the same answer as a distributive law between the associated monads.

Idea

Compare

- Finitary monads $\mathbf{Set} \longrightarrow \mathbf{Set}$ in $\mathbf{CAT}$ $\qquad \mathbf{Set} \xrightarrow{T} \mathbf{Set}$
- Lawvere theories as
  1. monads $\mathbb{F} \relbar\joinrel\mapstochar\joinrel\rightarrow \mathbb{F}$ in $\mathbf{Prof}$
  2. monads $\mathbb{F} \relbar\joinrel\mapstochar\joinrel\rightarrow \mathbb{F}$ in $\mathbf{Prof}(\mathbf{Mon})$
  3. monads $1 \relbar\joinrel\mapstochar\joinrel\rightarrow 1$ in $\mathbf{Prof}_{\mathcal{P}}$.

5. Comparison

Claim
These three methods all give the same answer as a distributive law between the associated monads.

Idea

Compare

- Finitary monads $\mathbf{Set} \longrightarrow \mathbf{Set}$ in $\mathbf{CAT}$
- Lawvere theories as
  1. monads $\mathbb{F} \relbar\joinrel\mapstochar\joinrel\rightarrow \mathbb{F}$ in $\mathbf{Prof}$
  2. monads $\mathbb{F} \relbar\joinrel\mapstochar\joinrel\rightarrow \mathbb{F}$ in $\mathbf{Prof}(\mathbf{Mon})$
  3. monads $1 \relbar\joinrel\mapstochar\joinrel\rightarrow 1$ in $\mathbf{Prof}_{\mathcal{P}}$.

$$\begin{tikzcd}
 & & \mathbf{Set} \arrow[rr, "T"] & & \mathbf{Set} & & \\
 & & & {} \arrow[d, mapsto] & & & \\
 \mathbb{F} \arrow[rr, "I_*", "\shortmid" marking] & & \mathbf{Set} \arrow[rr, "T_*", "\shortmid" marking] & {} & \mathbf{Set} \arrow[rr, "I^*", "\shortmid" marking] & & \mathbb{F}
\end{tikzcd}$$

5. Comparison

5. Comparison

$$\mathbf{Prof}_{\mathcal{P}}(1,1)$$

$$\mathbf{Prof}(\mathbb{F},\mathbb{F})$$

$$\mathbf{Prof}(\mathbf{Mon})(\mathbb{F},\mathbb{F})$$

Monads

$$\mathbf{Prof}_{\mathcal{P}}(1,1)$$

$$\mathbf{Prof}(\mathbb{F},\mathbb{F})$$

$$\mathbf{Prof}(\mathbf{Mon})(\mathbb{F},\mathbb{F})$$

Monads

$\mathbf{Prof}_{\mathcal{P}}(1,1)$ Lawvere theories

$\mathbf{Prof}(\mathbb{F},\mathbb{F})$

$\mathbf{Prof}(\mathbf{Mon})(\mathbb{F},\mathbb{F})$

5. Comparison

	Monads
$\mathbf{Prof}_{\mathcal{P}}(1,1)$	Lawvere theories
$\mathbf{Prof}(\mathbb{F},\mathbb{F})$	id-on-objects functors $\mathbb{F} \longrightarrow \mathbb{A}$
$\mathbf{Prof}(\mathbf{Mon})(\mathbb{F},\mathbb{F})$	

5. Comparison

	Monads
$\mathbf{Prof}_{\mathcal{P}}(1,1)$	Lawvere theories
$\mathbf{Prof}(\mathbb{F},\mathbb{F})$	id-on-objects functors $\mathbb{F} \longrightarrow \mathbb{A}$
$\mathbf{Prof}(\mathbf{Mon})(\mathbb{F},\mathbb{F})$	id-on-objects monoidal functors $\mathbb{F} \longrightarrow \mathbb{A}$

5. Comparison

		Monads
	$\mathbf{Prof}_{\mathcal{P}}(1,1)$	Lawvere theories
$\mathbf{CAT}_f(\mathbf{Set},\mathbf{Set})$	$\mathbf{Prof}(\mathbb{F},\mathbb{F})$	id-on-objects functors $\mathbb{F} \longrightarrow \mathbb{A}$
	$\mathbf{Prof}(\mathbf{Mon})(\mathbb{F},\mathbb{F})$	id-on-objects monoidal functors $\mathbb{F} \longrightarrow \mathbb{A}$

5. Comparison

	Monads
$\mathbf{Prof}_{\mathcal{P}}(1,1)$	Lawvere theories
$\mathbf{CAT}_f(\mathbf{Set},\mathbf{Set}) \xrightarrow{\mathsf{f+f}} \mathbf{Prof}(\mathbb{F},\mathbb{F})$	id-on-objects functors $\mathbb{F} \longrightarrow \mathbb{A}$
$\mathbf{Prof}(\mathbf{Mon})(\mathbb{F},\mathbb{F})$	id-on-objects monoidal functors $\mathbb{F} \longrightarrow \mathbb{A}$

5. Comparison

Monads

$$\begin{tikzcd} & \mathbf{Prof}_{\mathcal{P}}(1,1) \\ \mathbf{CAT}_f(\mathbf{Set},\mathbf{Set}) \arrow[ur, "\text{equivalence}"] \arrow[r, "\text{f+f}"] & \mathbf{Prof}(\mathbb{F},\mathbb{F}) \\ & \mathbf{Prof}(\mathbf{Mon})(\mathbb{F},\mathbb{F}) \end{tikzcd}$$

Lawvere theories

id-on-objects functors
$\mathbb{F} \longrightarrow \mathbb{A}$

id-on-objects
monoidal functors
$\mathbb{F} \longrightarrow \mathbb{A}$

5. Comparison

Monads

$$\begin{tikzcd}
& \mathbf{Prof}_{\mathcal{P}}(1,1) \arrow[d, "\text{forgetful}"] \\
& \mathbf{Prof}(\mathcal{P}1,\mathcal{P}1) \arrow[d, phantom, "\wr|"] \\
\mathbf{CAT}_f(\mathbf{Set},\mathbf{Set}) \arrow[uur, "\text{equivalence}"] \arrow[r, "\text{f+f}"] & \mathbf{Prof}(\mathbb{F},\mathbb{F})
\end{tikzcd}$$

Lawvere theories

id-on-objects functors
$\mathbb{F} \longrightarrow \mathbb{A}$

$\mathbf{Prof}(\mathbf{Mon})(\mathbb{F},\mathbb{F})$

id-on-objects
monoidal functors
$\mathbb{F} \longrightarrow \mathbb{A}$

5. Comparison

$$\begin{tikzcd}
 & \mathbf{Prof}_{\mathcal{P}}(1,1) \arrow[d, "\text{forgetful}"] \\
 & \mathbf{Prof}(\mathcal{P}1, \mathcal{P}1) \arrow[d, phantom, "\wr|"] \\
\mathbf{CAT}_f(\mathbf{Set}, \mathbf{Set}) \arrow[uur, "\text{equivalence}"] \arrow[r, "\text{f+f}"] \arrow[dr, dashed, "\text{f+f}"'] & \mathbf{Prof}(\mathbb{F}, \mathbb{F}) \\
 & \mathbf{Prof}(\mathbf{Mon})(\mathbb{F}, \mathbb{F}) \arrow[u, "\text{forgetful}"']
\end{tikzcd}$$

Monads

Lawvere theories

id-on-objects functors
$\mathbb{F} \longrightarrow \mathbb{A}$

id-on-objects
monoidal functors
$\mathbb{F} \longrightarrow \mathbb{A}$

Key points

5. Comparison

Key points

- The functors send T to $\mathbb{L}_T$.

5. Comparison

Key points

- The functors send T to $\mathbb{L}_T$.
- By pseudo-functoriality distributive laws map to distributive laws, and

$$\mathbb{L}_T \circ \mathbb{L}_S \cong \mathbb{L}_{TS}$$

5. Comparison

Key points

- The functors send T to $\mathbb{L}_T$.
- By pseudo-functoriality distributive laws map to distributive laws, and

$$\mathbb{L}_T \circ \mathbb{L}_S \cong \mathbb{L}_{TS}$$

- Moreover the functors are full and faithful, so given Lawvere theories on the right, *any* distributive law between them corresponds to one on the left.

5. Comparison

$$\begin{tikzcd}
 & \mathbf{Prof}_{\mathcal{P}}(1,1) \arrow[d, "\text{forgetful}"] \\
 & \mathbf{Prof}(\mathcal{P}1,\mathcal{P}1) \arrow[d, phantom, sloped, "\simeq"] \\
\mathbf{CAT}_f(\mathbf{Set},\mathbf{Set}) \arrow[uur, "\text{equivalence}"] \arrow[r, "\text{f+f}"] \arrow[dr, dashed, "\text{f+f}"'] & \mathbf{Prof}(\mathbb{F},\mathbb{F}) \\
 & \mathbf{Prof}(\mathbf{Mon})(\mathbb{F},\mathbb{F}) \arrow[u, "\text{forgetful}"']
\end{tikzcd}$$

Monads

Lawvere theories

id-on-objects functors
$\mathbb{F} \longrightarrow \mathbb{A}$

id-on-objects
monoidal functors
$\mathbb{F} \longrightarrow \mathbb{A}$

5. Comparison

Key points

- The functors are monoidal, and send T to $\mathbb{L}_T$.
- By pseudo-functoriality distributive laws map to distributive laws, and

$$\mathbb{L}_T \circ \mathbb{L}_S \cong \mathbb{L}_{TS}$$

- Moreover the functors are full and faithful, so given Lawvere theories on the right, *any* distributive law between them corresponds to one on the left.

Key points

- The functors are monoidal, and send T to $\mathbb{L}_T$.
- By pseudo-functoriality distributive laws map to distributive laws, and

$$\mathbb{L}_T \circ \mathbb{L}_S \cong \mathbb{L}_{TS}$$

- Moreover the functors are full and faithful, so given Lawvere theories on the right, *any* distributive law between them corresponds to one on the left.

So we have three equivalent notions of distributive laws for Lawvere theories, which correspond to distributive laws between the associated monads.