Composing Lawvere Theories

Andrei Akhvlediani

Oxford University Computing Laboratory

June 15, 2010

Motivation and background

We describe a uniform framework for composing PROPs and (countable) Lawvere theories. There are several reasons for why we wish to do this.

- Maths: We gain the ability to form complex algebraic structures from simple ones in both the Cartesian and the monoidal settings.

We describe a uniform framework for composing PROPs and (countable) Lawvere theories. There are several reasons for why we wish to do this.

- Maths: We gain the ability to form complex algebraic structures from simple ones in both the Cartesian and the monoidal settings.
- Computer Science: (Countable) Lawvere theories are used to model computational effects such as exceptions, input/output and nondeterminism in functional programming languages. Each such effect has an associated theory. In order to model a language with several interacting effects, one needs methods for combining those theories.

We describe a uniform framework for composing PROPs and (countable) Lawvere theories. There are several reasons for why we wish to do this.

- Maths: We gain the ability to form complex algebraic structures from simple ones in both the Cartesian and the monoidal settings.
- Computer Science: (Countable) Lawvere theories are used to model computational effects such as exceptions, input/output and nondeterminism in functional programming languages. Each such effect has an associated theory. In order to model a language with several interacting effects, one needs methods for combining those theories.
- Quantum Information In the Abramsky-Coecke approach to Quantum Information, one uses various interacting algebras/coalgebras and their string diagrammatic representation to model aspects of quantum mechanics. Expressing those algebraic structures as composites of simpler ones allows us to gain insight into the normal forms of their graphical representation, greatly simplifying reasoning using those structures and assisting with the creation of automated reasoning tools (e.g. Quantomatic: http://dream.inf.ed.ac.uk/projects/quantomatic).

Outline

1. Recall the definition of distributive laws between monads (in Cat)
2. Define Lawvere theories
3. Define PROPs
4. Describe Span(Mon) as a setting for composing strict monoidal categories
5. Describe Prof(Mon) as a setting for composing strict monoidal categories with extra structure
6. Show that Lawvere theories can be composed in Prof(Mon)
7. Discuss the generation of distributive laws and look at an example

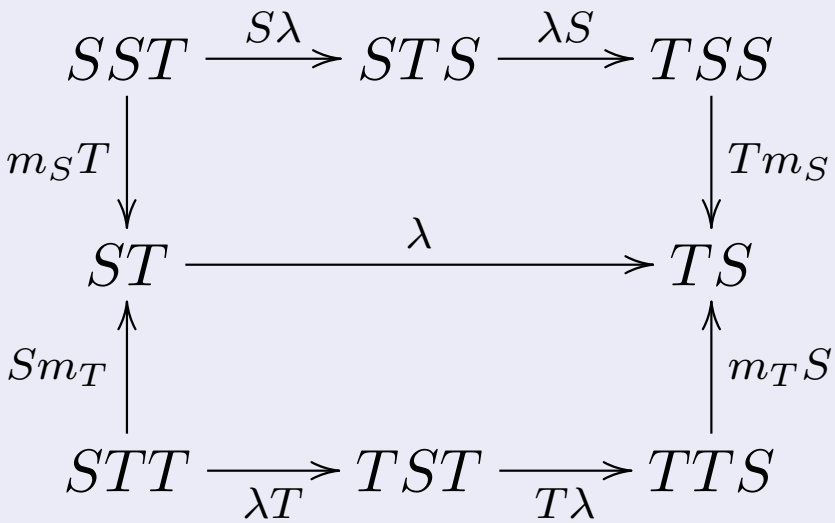
Distributive Laws

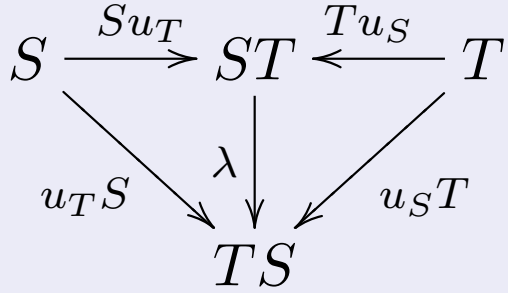
Distributive Law

Let $T, S : \mathcal{C} \to \mathcal{C}$ be monads. A *distributive law* of S over T is a nat. trans.

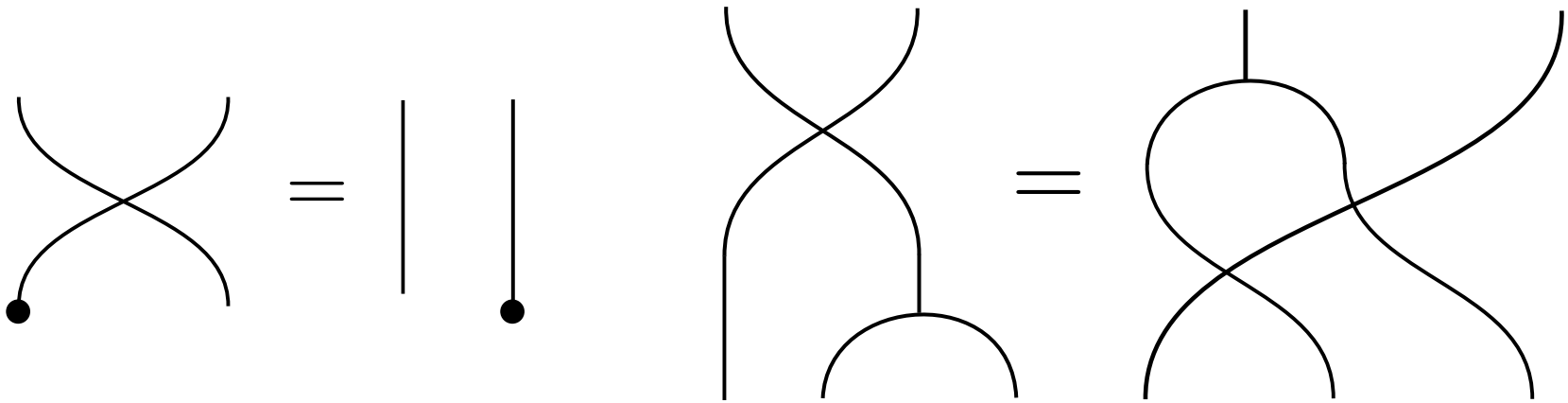
$$\lambda : ST \to TS$$

making the following diagrams commute:





The axioms, visually



Lawvere Theories and PROPs

- Let $\mathbb{F}$ denote the skeletal category of *finite* sets and functions; Let $\mathbb{C}$ denote the skeletal category of *countable* sets and functions.

- Let $\mathbb{F}$ denote the skeletal category of *finite* sets and functions; Let $\mathbb{C}$ denote the skeletal category of *countable* sets and functions.
- A *Lawvere theory* consists of a small category L with finite products and a strict finite-product preserving identity-on-objects functor $I : \mathbb{F}^{\mathrm{op}} \to L$. For a countable Lawvere theory replace $\mathbb{F}$ by $\mathbb{C}$ and finite by countable.

- Let $\mathbb{F}$ denote the skeletal category of *finite* sets and functions; Let $\mathbb{C}$ denote the skeletal category of *countable* sets and functions.
- A *Lawvere theory* consists of a small category L with finite products and a strict finite-product preserving identity-on-objects functor $I : \mathbb{F}^{\mathrm{op}} \to L$. For a countable Lawvere theory replace $\mathbb{F}$ by $\mathbb{C}$ and finite by countable.
- Let $\mathcal{C}$ be a finite product category. An *L-model* is a finite product preserving functor $L \to \mathcal{C}$.

- Let $\mathbb{F}$ denote the skeletal category of *finite* sets and functions; Let $\mathbb{C}$ denote the skeletal category of *countable* sets and functions.
- A *Lawvere theory* consists of a small category L with finite products and a strict finite-product preserving identity-on-objects functor $I : \mathbb{F}^{\mathrm{op}} \to L$. For a countable Lawvere theory replace $\mathbb{F}$ by $\mathbb{C}$ and finite by countable.
- Let $\mathcal{C}$ be a finite product category. An *L-model* is a finite product preserving functor $L \to \mathcal{C}$.
- An *L-homomorphism* between L-algebras A and B, is a natural transformation $\alpha : A \to B$.

- Let $\mathbb{F}$ denote the skeletal category of *finite* sets and functions; Let $\mathbb{C}$ denote the skeletal category of *countable* sets and functions.
- A *Lawvere theory* consists of a small category L with finite products and a strict finite-product preserving identity-on-objects functor $I : \mathbb{F}^{\mathrm{op}} \to L$. For a countable Lawvere theory replace $\mathbb{F}$ by $\mathbb{C}$ and finite by countable.
- Let $\mathcal{C}$ be a finite product category. An *L-model* is a finite product preserving functor $L \to \mathcal{C}$.
- An *L-homomorphism* between L-algebras A and B, is a natural transformation $\alpha : A \to B$.
- L-algebras and their homomorphisms form a category $\mathrm{Mod}(L, \mathcal{C})$.

- Let $\mathbb{F}$ denote the skeletal category of *finite* sets and functions; Let $\mathbb{C}$ denote the skeletal category of *countable* sets and functions.
- A *Lawvere theory* consists of a small category L with finite products and a strict finite-product preserving identity-on-objects functor $I : \mathbb{F}^{\mathrm{op}} \to L$. For a countable Lawvere theory replace $\mathbb{F}$ by $\mathbb{C}$ and finite by countable.
- Let $\mathcal{C}$ be a finite product category. An *L-model* is a finite product preserving functor $L \to \mathcal{C}$.
- An *L-homomorphism* between L-algebras A and B, is a natural transformation $\alpha : A \to B$.
- L-algebras and their homomorphisms form a category $\mathrm{Mod}(L, \mathcal{C})$.

A famous example:

$$\mathrm{Mod}(\mathrm{FreeGrp}^{\mathrm{op}}, \mathrm{Set}) \simeq \mathrm{Grp}.$$

- Let $\mathbb{P}$ denote the free symmetric monoidal category on 1. That is, the objects of $\mathbb{P}$ are the natural numbers and $\mathbb{P}(n, m) = S(n)$ if $n = m$ and is empty otherwise.

- Let $\mathbb{P}$ denote the free symmetric monoidal category on 1. That is, the objects of $\mathbb{P}$ are the natural numbers and $\mathbb{P}(n, m) = S(n)$ if $n = m$ and is empty otherwise.
- A *PROP* is a strict symmetric monoidal category $\mathbb{T}$ with a strict monoidal identity-on-objects functor $\mathbb{P} \to \mathbb{T}$.

- Let $\mathbb{P}$ denote the free symmetric monoidal category on 1. That is, the objects of $\mathbb{P}$ are the natural numbers and $\mathbb{P}(n, m) = S(n)$ if $n = m$ and is empty otherwise.
- A *PROP* is a strict symmetric monoidal category $\mathbb{T}$ with a strict monoidal identity-on-objects functor $\mathbb{P} \longrightarrow \mathbb{T}$.
- A $\mathbb{T}$*-model* in a symmetric monoidal category $\mathcal{C}$ is a symmetric monoidal functor $\mathbb{T} \longrightarrow \mathcal{C}$.

- Let $\mathbb{P}$ denote the free symmetric monoidal category on 1. That is, the objects of $\mathbb{P}$ are the natural numbers and $\mathbb{P}(n, m) = S(n)$ if $n = m$ and is empty otherwise.
- A *PROP* is a strict symmetric monoidal category $\mathbb{T}$ with a strict monoidal identity-on-objects functor $\mathbb{P} \to \mathbb{T}$.
- A $\mathbb{T}$*-model* in a symmetric monoidal category $\mathcal{C}$ is a symmetric monoidal functor $\mathbb{T} \to \mathcal{C}$.
- A $\mathbb{T}$*-homomorphism* between models A and B is a symmetric monoidal natural transformation between them.

- Let $\mathbb{P}$ denote the free symmetric monoidal category on 1. That is, the objects of $\mathbb{P}$ are the natural numbers and $\mathbb{P}(n, m) = S(n)$ if $n = m$ and is empty otherwise.
- A *PROP* is a strict symmetric monoidal category $\mathbb{T}$ with a strict monoidal identity-on-objects functor $\mathbb{P} \to \mathbb{T}$.
- A $\mathbb{T}$*-model* in a symmetric monoidal category $\mathcal{C}$ is a symmetric monoidal functor $\mathbb{T} \to \mathcal{C}$.
- A $\mathbb{T}$*-homomorphism* between models A and B is a symmetric monoidal natural transformation between them.
- $\mathbb{T}$-models and their homomorphisms form a category $\mathrm{Mod}(\mathbb{T}, \mathcal{C})$.

- Let $\mathbb{P}$ denote the free symmetric monoidal category on 1. That is, the objects of $\mathbb{P}$ are the natural numbers and $\mathbb{P}(n, m) = S(n)$ if $n = m$ and is empty otherwise.
- A *PROP* is a strict symmetric monoidal category $\mathbb{T}$ with a strict monoidal identity-on-objects functor $\mathbb{P} \to \mathbb{T}$.
- A $\mathbb{T}$*-model* in a symmetric monoidal category $\mathcal{C}$ is a symmetric monoidal functor $\mathbb{T} \to \mathcal{C}$.
- A $\mathbb{T}$*-homomorphism* between models A and B is a symmetric monoidal natural transformation between them.
- $\mathbb{T}$-models and their homomorphisms form a category $\mathrm{Mod}(\mathbb{T}, \mathcal{C})$.

For example, the category $\mathbb{F}$ is a PROP. Its algebras in $\mathcal{C}$ are the commutative monoids in $\mathcal{C}$.

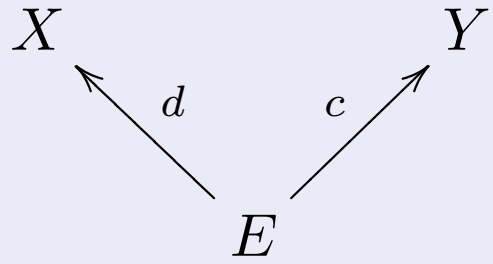
Span and Prof

Let $\mathcal{C}$ be a category with pullbacks. The bicategory of *internal Spans* in $\mathcal{C}$ consists of the following data:

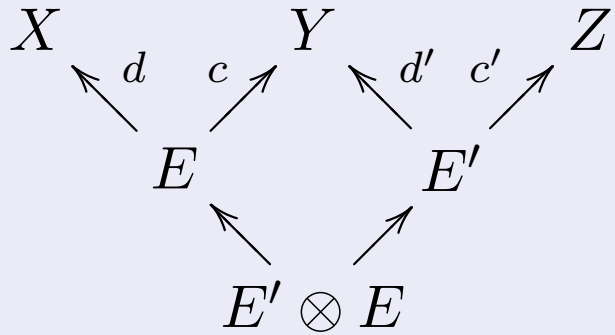
- 0-cells: objects of $\mathcal{C}$

Let $\mathcal{C}$ be a category with pullbacks. The bicategory of *internal Spans* in $\mathcal{C}$ consists of the following data:

- 0-cells: objects of $\mathcal{C}$
- 1-cells: spans, i.e.

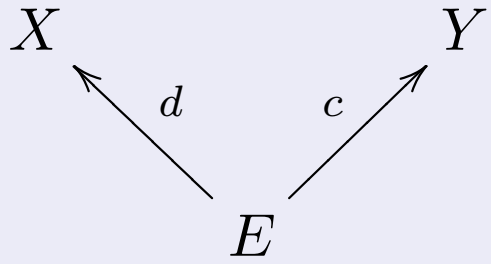


Composition of 1-cells is given by pullback:

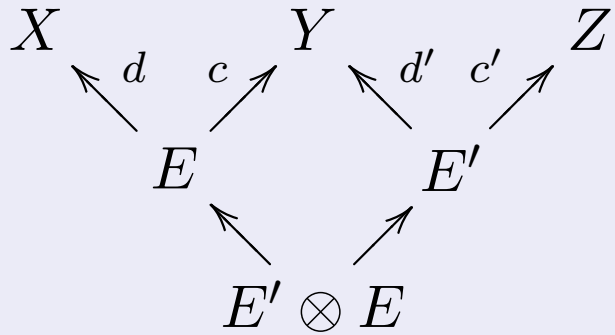


Let $\mathcal{C}$ be a category with pullbacks. The bicategory of *internal Spans* in $\mathcal{C}$ consists of the following data:

- 0-cells: objects of $\mathcal{C}$
- 1-cells: spans, i.e.



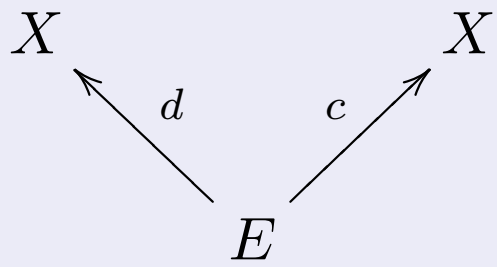
Composition of 1-cells is given by pullback:



- 2-cells: morphisms of spans

Take $\mathcal{C} = \mathrm{Mon}$. The monads in $\mathrm{Span}(\mathrm{Mon})$ are strict monoidal categories:

A monad is a span

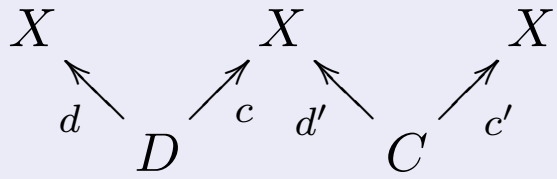


- X is the set of objects;
- E is the set of morphisms;
- d and c are the domain and codomain maps;
- $E \otimes E \to E$ is composition and $X \to E$ is given by the identities.
- monoidal structure comes from Mon.

Thus categories can be composed using distributive laws [Rosebrugh, Wood 2002]

Distributive laws in Span($\mathcal{C}$):

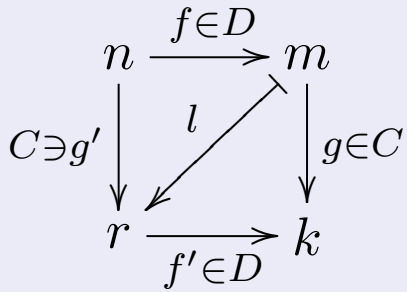
Let $\mathcal{C}$ and $\mathcal{D}$ be categories, with $\mathrm{ob}\mathcal{C} = X = \mathrm{ob}\mathcal{D}$ viewed as spans:



A *distributive law of* $\mathcal{C}$ *over* $\mathcal{D}$ is a $\mathcal{C}$-morphism

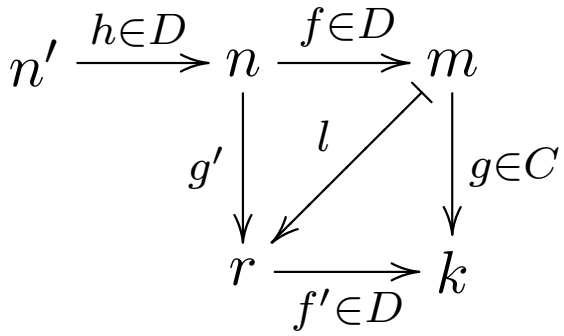
$$l : C \otimes D \to D \otimes C$$

which we depict by

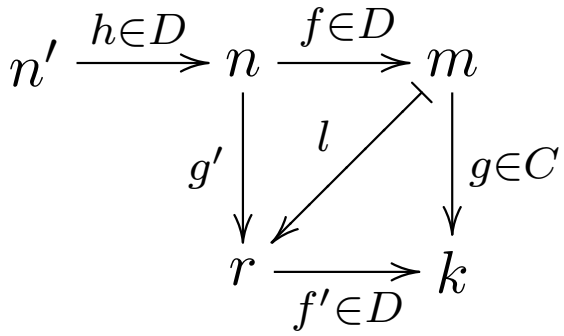


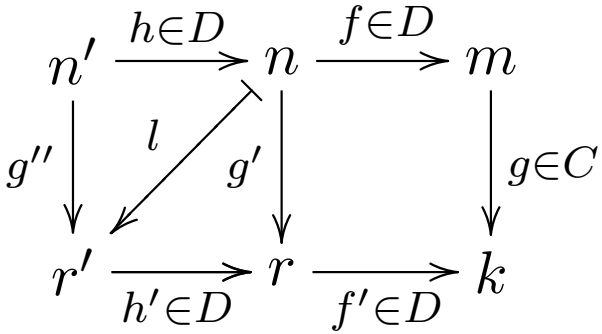
This 2-cell interacts nicely with the multiplications and units of the monads C and D - i.e. with the composition and identities of the underlying categories:

Distributive laws in Span($\mathcal{C}$)

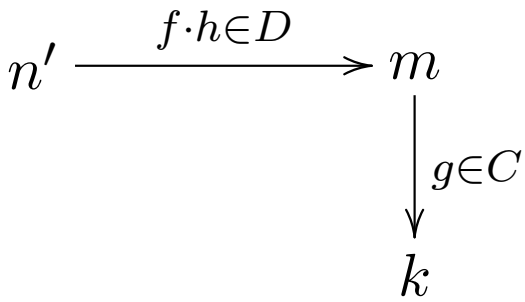


Distributive laws in Span($\mathcal{C}$)

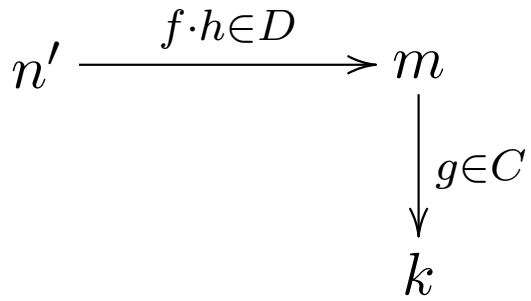


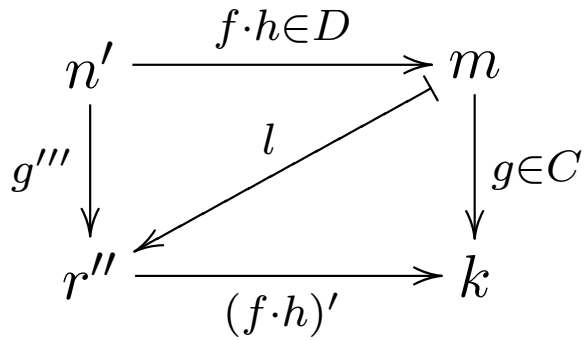


Distributive laws in Span($\mathcal{C}$)

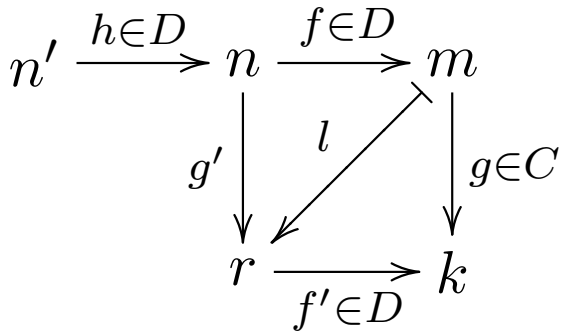


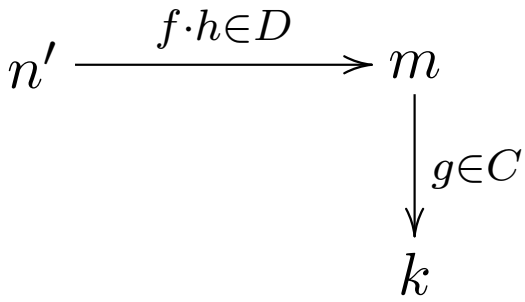
Distributive laws in Span($\mathcal{C}$)

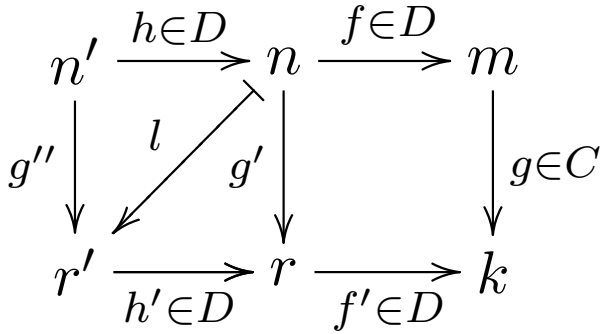


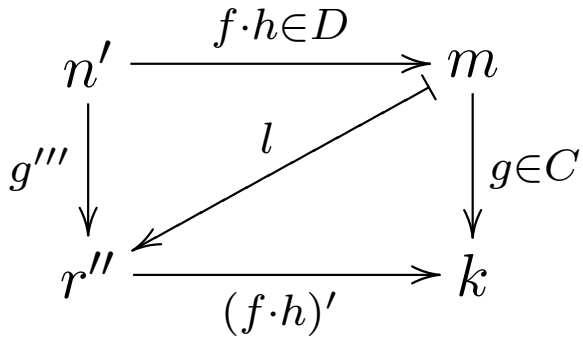


Distributive laws in Span($\mathcal{C}$)

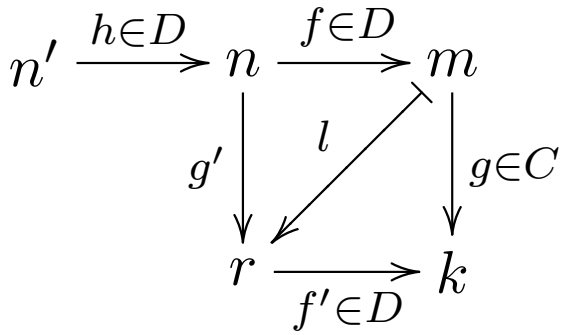


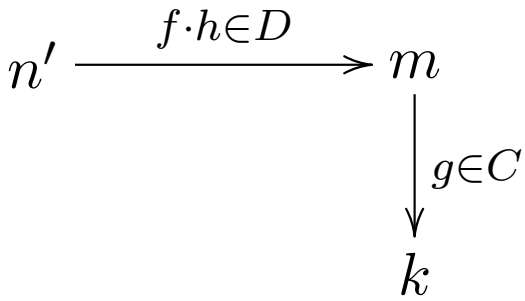


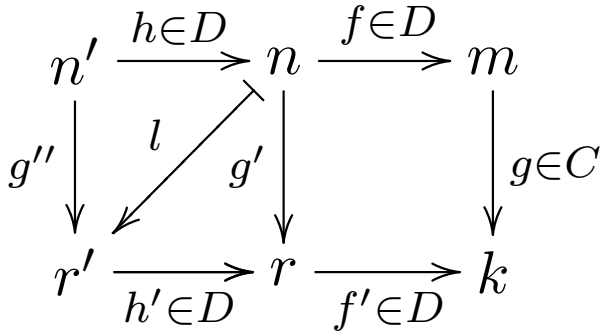


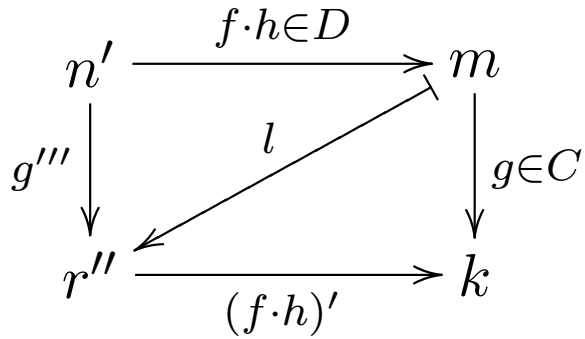


Distributive laws in Span($\mathcal{C}$)









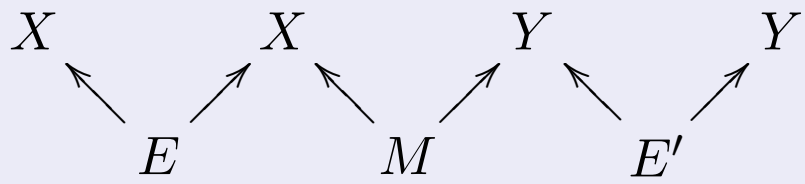
$$r' = r'',\ (f \cdot h)' = f' \cdot h',\ g''' = g''.$$

To compose categories with structure, we move from $\mathrm{Span}(\mathcal{C})$ to $\mathrm{Prof}(\mathcal{C})$:

- 0-cells: Monads in $\mathrm{Span}(\mathcal{C})$; when $\mathcal{C} = \mathrm{Mon}$ just strict monoidal categories.

To compose categories with structure, we move from $\mathrm{Span}(\mathcal{C})$ to $\mathrm{Prof}(\mathcal{C})$:

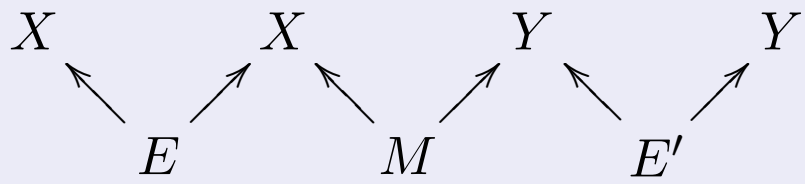
- 0-cells: Monads in $\mathrm{Span}(\mathcal{C})$; when $\mathcal{C} = \mathrm{Mon}$ just strict monoidal categories.
- 1-cells: *$\mathcal{C}$-internal profunctors* $M : \mathcal{E}' \to \mathcal{E}$, i.e., spans



that are equipped with left and right actions: $\alpha : E' \otimes M \to M$ and $\beta : M \otimes E \to M$ that make the triple (M, α, β) into an E'-E-bimodule.

To compose categories with structure, we move from $\mathrm{Span}(\mathcal{C})$ to $\mathrm{Prof}(\mathcal{C})$:

- 0-cells: Monads in $\mathrm{Span}(\mathcal{C})$; when $\mathcal{C} = \mathrm{Mon}$ just strict monoidal categories.
- 1-cells: *$\mathcal{C}$-internal profunctors* $M : \mathcal{E}' \to \mathcal{E}$, i.e., spans

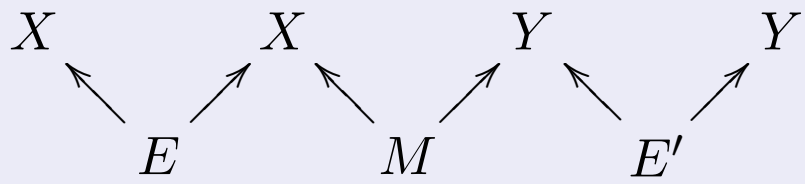


  that are equipped with left and right actions: $\alpha : E' \otimes M \to M$ and $\beta : M \otimes E \to M$ that make the triple (M, α, β) into an E'-E-bimodule.
- Given $M' : \mathcal{E}'' \to \mathcal{E}'$ and $M : \mathcal{E}' \to \mathcal{E}$ the composite module is given by the coequalizer in $\mathcal{C}$:

$$M' \otimes E' \otimes M \underset{1 \otimes \alpha}{\overset{\beta' \otimes 1}{\rightrightarrows}} M' \otimes M \xrightarrow{\kappa} M' \otimes_{\mathcal{E}'} M$$

To compose categories with structure, we move from $\mathrm{Span}(\mathcal{C})$ to $\mathrm{Prof}(\mathcal{C})$:

- 0-cells: Monads in $\mathrm{Span}(\mathcal{C})$; when $\mathcal{C} = \mathrm{Mon}$ just strict monoidal categories.
- 1-cells: *$\mathcal{C}$-internal profunctors* $M : \mathcal{E}' \to \mathcal{E}$, i.e., spans



  that are equipped with left and right actions: $\alpha : E' \otimes M \to M$ and $\beta : M \otimes E \to M$ that make the triple (M, α, β) into an E'-E-bimodule.
- Given $M' : \mathcal{E}'' \to \mathcal{E}'$ and $M : \mathcal{E}' \to \mathcal{E}$ the composite module is given by the coequalizer in $\mathcal{C}$:

$$M' \otimes E' \otimes M \underset{1 \otimes \alpha}{\overset{\beta' \otimes 1}{\rightrightarrows}} M' \otimes M \xrightarrow{\kappa} M' \otimes_{\mathcal{E}'} M$$

- 2-cells: bimodule homomorphisms.

Examples

Take $\mathcal{C} = \mathsf{Ab}$ - the category of abelian groups.

Examples

Take $\mathcal{C} = \mathrm{Ab}$ - the category of abelian groups.

- The monads on 1 in $\mathrm{Span}(\mathrm{Ab})$ are single object Abelian categories - or simply rings.

Examples

Take $\mathcal{C} = \mathrm{Ab}$ - the category of abelian groups.

- The monads on 1 in $\mathrm{Span}(\mathrm{Ab})$ are single object Abelian categories - or simply rings.
- With R and S rings, a 1-cell between them in $\mathrm{Prof}(\mathrm{Ab})$ is just an R-S-bimodule.

Examples

Take $\mathcal{C} = \mathrm{Ab}$ - the category of abelian groups.

- The monads on 1 in $\mathrm{Span}(\mathrm{Ab})$ are single object Abelian categories - or simply rings.
- With R and S rings, a 1-cell between them in $\mathrm{Prof}(\mathrm{Ab})$ is just an R-S-bimodule.
- Composition is the tensor product of bimodules.

Examples

Take $\mathcal{C} = \mathrm{Ab}$ - the category of abelian groups.

- The monads on 1 in $\mathrm{Span}(\mathrm{Ab})$ are single object Abelian categories - or simply rings.
- With R and S rings, a 1-cell between them in $\mathrm{Prof}(\mathrm{Ab})$ is just an R-S-bimodule.
- Composition is the tensor product of bimodules.

Take $\mathcal{C} = \mathrm{Mon}$. And let L be a Lawvere theory. Then $L : \mathbb{F}^{\mathrm{op}} \longrightarrow \mathbb{F}^{\mathrm{op}}$ is a monad on $\mathbb{F}^{\mathrm{op}}$ in $\mathrm{Prof}(\mathrm{Mon})$:

Examples

Take $\mathcal{C} = \mathrm{Ab}$ - the category of abelian groups.

- The monads on 1 in $\mathrm{Span}(\mathrm{Ab})$ are single object Abelian categories - or simply rings.
- With R and S rings, a 1-cell between them in $\mathrm{Prof}(\mathrm{Ab})$ is just an R-S-bimodule.
- Composition is the tensor product of bimodules.

Take $\mathcal{C} = \mathrm{Mon}$. And let L be a Lawvere theory. Then $L : \mathbb{F}^{\mathrm{op}} \longrightarrow \mathbb{F}^{\mathrm{op}}$ is a monad on $\mathbb{F}^{\mathrm{op}}$ in $\mathrm{Prof}(\mathrm{Mon})$:

- Multiplication and unit are given by the composition and identities of the underlying category;

Examples

Take $\mathcal{C} = \mathrm{Ab}$ - the category of abelian groups.

- The monads on 1 in $\mathrm{Span}(\mathrm{Ab})$ are single object Abelian categories - or simply rings.
- With R and S rings, a 1-cell between them in $\mathrm{Prof}(\mathrm{Ab})$ is just an R-S-bimodule.
- Composition is the tensor product of bimodules.

Take $\mathcal{C} = \mathrm{Mon}$. And let L be a Lawvere theory. Then $L : \mathbb{F}^{\mathrm{op}} \longrightarrow \mathbb{F}^{\mathrm{op}}$ is a monad on $\mathbb{F}^{\mathrm{op}}$ in $\mathrm{Prof}(\mathrm{Mon})$:

- Multiplication and unit are given by the composition and identities of the underlying category;
- Actions: $L \otimes \mathbb{F}^{\mathrm{op}} \longrightarrow L, \quad (\pi, l) \mapsto l \cdot \pi.$

The following Theorem allows us to express various categories with structure as monads in Prof(Mon):

Theorem

Let $\mathcal{C}$ and $\mathcal{D}$ be *identity-on-objects* *categories. There is a 1-1 correspondence between* *functors*

$$F : \mathcal{C} \to \mathcal{D}$$

and monads $\mathcal{C} \to \mathcal{C}$ in Prof( ).

The following Theorem allows us to express various categories with structure as monads in Prof(Mon):

Theorem

Let $\mathcal{C}$ and $\mathcal{D}$ be strict monoidal categories. There is a 1-1 correspondence between strict identity-on-objects monoidal functors

$$F : \mathcal{C} \to \mathcal{D}$$

and monads $\mathcal{C} \to \mathcal{C}$ in Prof(Mon).

The following Theorem allows us to express various categories with structure as monads in Prof(Mon):

Theorem

Let $\mathcal{C}$ and $\mathcal{D}$ be strict monoidal categories. There is a 1-1 correspondence between strict identity-on-objects monoidal functors

$$F : \mathcal{C} \to \mathcal{D}$$

and monads $\mathcal{C} \to \mathcal{C}$ in Prof(Mon).

- $\mathbb{F}^{\mathrm{op}} \to L$ - Lawvere theories become monads on $\mathbb{F}^{\mathrm{op}}$;

The following Theorem allows us to express various categories with structure as monads in Prof(Mon):

Theorem

Let $\mathcal{C}$ and $\mathcal{D}$ be strict monoidal categories. There is a 1-1 correspondence between strict identity-on-objects monoidal functors

$$F : \mathcal{C} \to \mathcal{D}$$

and monads $\mathcal{C} \to \mathcal{C}$ in Prof(Mon).

- $\mathbb{F}^{\mathrm{op}} \to L$ - Lawvere theories become monads on $\mathbb{F}^{\mathrm{op}}$;
- $\mathbb{C}^{\mathrm{op}} \to L$ - Countable Lawvere theories become monads on $\mathbb{C}^{\mathrm{op}}$;

The following Theorem allows us to express various categories with structure as monads in Prof(Mon):

Theorem

Let $\mathcal{C}$ and $\mathcal{D}$ be strict monoidal categories. There is a 1-1 correspondence between strict identity-on-objects monoidal functors

$$F : \mathcal{C} \to \mathcal{D}$$

and monads $\mathcal{C} \to \mathcal{C}$ in Prof(Mon).

- $\mathbb{F}^{\mathrm{op}} \to L$ - Lawvere theories become monads on $\mathbb{F}^{\mathrm{op}}$;
- $\mathbb{C}^{\mathrm{op}} \to L$ - Countable Lawvere theories become monads on $\mathbb{C}^{\mathrm{op}}$;
- $\mathbb{N} \to \mathbb{S}$ - PROs become monads on $\mathbb{N}$ (or just monads in Span(Mon));

The following Theorem allows us to express various categories with structure as monads in Prof(Mon):

Theorem

Let $\mathcal{C}$ and $\mathcal{D}$ be strict monoidal categories. There is a 1-1 correspondence between strict identity-on-objects monoidal functors

$$F : \mathcal{C} \to \mathcal{D}$$

and monads $\mathcal{C} \to \mathcal{C}$ in Prof(Mon).

- $\mathbb{F}^{\mathrm{op}} \to L$ - Lawvere theories become monads on $\mathbb{F}^{\mathrm{op}}$;
- $\mathbb{C}^{\mathrm{op}} \to L$ - Countable Lawvere theories become monads on $\mathbb{C}^{\mathrm{op}}$;
- $\mathbb{N} \to \mathbb{S}$ - PROs become monads on $\mathbb{N}$ (or just monads in Span(Mon));
- $\mathbb{P} \to \mathbb{T}$ - PROPs become monads on $\mathbb{P}$;

The following Theorem allows us to express various categories with structure as monads in Prof(Mon):

Theorem

Let $\mathcal{C}$ and $\mathcal{D}$ be strict monoidal categories. There is a 1-1 correspondence between strict identity-on-objects monoidal functors

$$F : \mathcal{C} \to \mathcal{D}$$

and monads $\mathcal{C} \to \mathcal{C}$ in Prof(Mon).

- $\mathbb{F}^{\mathrm{op}} \to L$ - Lawvere theories become monads on $\mathbb{F}^{\mathrm{op}}$;
- $\mathbb{C}^{\mathrm{op}} \to L$ - Countable Lawvere theories become monads on $\mathbb{C}^{\mathrm{op}}$;
- $\mathbb{N} \to \mathbb{S}$ - PROs become monads on $\mathbb{N}$ (or just monads in Span(Mon));
- $\mathbb{P} \to \mathbb{T}$ - PROPs become monads on $\mathbb{P}$;
- $\mathbb{B} \to \mathbb{R}$ - PROBs (braided PROPs) become monads on $\mathbb{B}$.

The following Theorem allows us to express various categories with structure as monads in Prof(Mon):

Theorem

Let $\mathcal{C}$ and $\mathcal{D}$ be strict monoidal categories. There is a 1-1 correspondence between strict identity-on-objects monoidal functors

$$F : \mathcal{C} \to \mathcal{D}$$

and monads $\mathcal{C} \to \mathcal{C}$ in Prof(Mon).

- $\mathbb{F}^{\mathrm{op}} \to L$ - Lawvere theories become monads on $\mathbb{F}^{\mathrm{op}}$;
- $\mathbb{C}^{\mathrm{op}} \to L$ - Countable Lawvere theories become monads on $\mathbb{C}^{\mathrm{op}}$;
- $\mathbb{N} \to \mathbb{S}$ - PROs become monads on $\mathbb{N}$ (or just monads in Span(Mon));
- $\mathbb{P} \to \mathbb{T}$ - PROPs become monads on $\mathbb{P}$;
- $\mathbb{B} \to \mathbb{R}$ - PROBs (braided PROPs) become monads on $\mathbb{B}$.

[S. Lack, 2004]

It turns out that the composite of two categories with structure X is again a category with structure X. We formulate this for X = countable products.

It turns out that the composite of two categories with structure X is again a category with structure X. We formulate this for X = countable products.

Theorem

Let L and L' have countable products and $\lambda : L' \otimes_{\mathbb{C}^{\mathrm{op}}} L \to L \otimes_{\mathbb{C}^{\mathrm{op}}} L'$ be a distributive law. Then $L \otimes_{\mathbb{C}^{\mathrm{op}}} L'$ also has countable products.

It turns out that the composite of two categories with structure X is again a category with structure X. We formulate this for X = countable products.

Theorem

Let L and L' have countable products and $\lambda : L' \otimes_{\mathbb{C}^{\mathrm{op}}} L \to L \otimes_{\mathbb{C}^{\mathrm{op}}} L'$ be a distributive law. Then $L \otimes_{\mathbb{C}^{\mathrm{op}}} L'$ also has countable products.

Lack's Theorem about composite algebras for PROPs carries through with practically no changes to (countable) Lawvere theories.

Theorem

An $L \otimes_{\mathbb{C}^{\mathrm{op}}} L'$-algebra structures on A consists of an L-algebras structure on A, and L'-algebra structure on A such that for all $f : n \to m \in L$, $g : m \to r \in L'$

$$\begin{array}{ccc} A^n & \xrightarrow{f} & A^m \\ {\scriptstyle g'}\downarrow & & \downarrow{\scriptstyle g} \\ A^k & \xrightarrow{f'} & A^r \end{array}$$

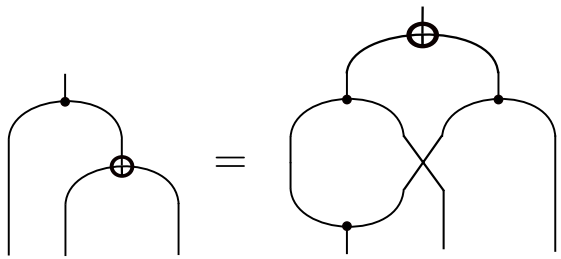
commutes.

Example

Let L be the Lawvere theory for semi-groups and L' the Lawvere theory for Abelian semigroups, given by the opposite of the category of free semigroups/abelian semigroups (on finitely many generators).
We can generate a distributive law

$$L \otimes_{\mathbb{F}^{\mathrm{op}}} L' \to L' \otimes_{\mathbb{F}^{op}} L$$

from the following equation



The composite theory is the Lawvere theory for semi-rings.

Summary and future work

- We saw that Prof(Mon) provides a good setting for composing categories with different types of monoidal structure.

- We saw that Prof(Mon) provides a good setting for composing categories with different types of monoidal structure.
- In particular (countable) Lawvere theories can be composed in a novel way with implications for computer science.

- We saw that Prof(Mon) provides a good setting for composing categories with different types of monoidal structure.
- In particular (countable) Lawvere theories can be composed in a novel way with implications for computer science.
- Just as Lawvere Theories model computational effects in classical computing, one can use a variation of PROPs to model interaction of a quantum system with the "classical environment". The modularity of such interactions is built-in into the theory. This is work in progress with Bob Coecke and Raymond Lal.

- We saw that Prof(Mon) provides a good setting for composing categories with different types of monoidal structure.
- In particular (countable) Lawvere theories can be composed in a novel way with implications for computer science.
- Just as Lawvere Theories model computational effects in classical computing, one can use a variation of PROPs to model interaction of a quantum system with the "classical environment". The modularity of such interactions is built-in into the theory. This is work in progress with Bob Coecke and Raymond Lal.
- To construct a collection of examples of composite PROPs/Lawvere theories we need the ability to generate distributive laws from (nice) rewriting systems.

References

- S. Abramsky, B. Coecke, A categorical semantics of quantum protocols, in *Proceedings of the 19th IEEE conference on Logic in Computer Science* (LiCS'04). IEEE Computer Science Press (2004).
- C. Coecke, D. Pavlovic, Quantum measurements without sums, in *The Mathematics of Quantum Computation and Technology* (2008).
- M. Hyland, G. Plotkin, J. Power. Combining Effects: Sum and Tensor *Theoretical Computer Science*, 357, (2006).
- S. Lack, Composing PROPs, *Theory and Applications of Categories*, Vol. 13, (2004).
- E. Moggi, Notions of computation and monads. *Information And Computation*, 93(1), (1991).
- R. Rosebrugh, R. J. Wood, Distributive laws and factorization, *Journal of Pure and Applied Algebra*, Volume 175, (2002).