Displayed Monoidal Categories for the Semantics of Linear Logic

Benedikt Ahrens Ralph Matthes **Niels van der Weide**
Kobe Wullaert

January 16, 2024

This Session: Formalizing Category Theory

What is category theory?

- Category is an abstract **framework** for mathematics
- Generalizes a common pattern: we have **objects** and **morphisms**
- Also used to characterize certain constructions (products, exponentials, ...)
- Useful in the study of **semantics of logic and programming languages**

Category Theory and Semantics

Curry-Howard-Lambek correspondence

Logic	Programming language	Category Theory
Formula	Type	Object
Proof	Program	Morphism
Connective	Type Constructor	Categorical structure

Category Theory and Semantics

Curry-Howard-Lambek correspondence

Logic	Programming language	Category Theory
Formula	Type	Object
Proof	Program	Morphism
Connective	Type Constructor	Categorical structure

Note: categorical structure is described via **universal properties** whereas connective/type constructors are described via **introduction and elimination rules**.

The Many Flavors of Category Theory

- To model advanced logics or programming languages, we need more **categorical structure**
- For this reason, many different kinds of categorical structure have been studied
- Throughout this session we will see various kinds of categories (monoidal categories, double categories, ∞-categories)
- In this talk: **monoidal categories**

Monoidal Categories, what are they?

Basically: **Monoid**al **category** = **Monoid** + **category**

Monoidal Categories, what are they?

Basically: **Monoid**al **category** = **Monoid** + **category**
A **category** consists of

- objects
- morphisms
- we have identity and composition operation

Monoidal Categories, what are they?

Basically: **Monoid**al **category** = **Monoid** + **category**

A **category** consists of

- objects
- morphisms
- we have identity and composition operation

A **monoidal category** is a category with a **multiplication** $\otimes$.

- given objects x, y, we have an object $x \otimes y$
- given morphisms $f : x \to x'$ and $g : y \to y'$, we have a morphism $f \otimes g : x \otimes y \to x' \otimes y'$

We require $\otimes$ to be associative and unital in a weak sense.

Examples and Applications of monoidal categories

There are many examples of monoidal categories

- Sets (and functions) with the binary product
- Sets (and functions) with the binary coproduct
- Sets (and **relations**) with the binary product
- Abelian groups with the tensor product

Examples and Applications of monoidal categories

There are many examples of monoidal categories

- Sets (and functions) with the binary product
- Sets (and functions) with the binary coproduct
- Sets (and **relations**) with the binary product
- Abelian groups with the tensor product

Monoidal categories are used in

- the semantics of linear logic
- quantum theory
- domain theory and algebraic effects (smash products)

Examples and Applications of monoidal categories

There are many examples of monoidal categories

- Sets (and functions) with the binary product
- Sets (and functions) with the binary coproduct
- Sets (and **relations**) with the binary product
- Abelian groups with the tensor product

Monoidal categories are used in

- **the semantics of linear logic**
- quantum theory
- domain theory and algebraic effects (smash products)

In this talk: we are interested in the **semantics of linear logic**

Monoidal Categories and Linear Logic

$$\frac{x : \mathcal{C} \qquad y : \mathcal{C}}{x \otimes y : \mathcal{C}}$$

$$\frac{f : x \to x' \qquad g : y \to y'}{f \otimes g : x \otimes y \to x' \otimes y'}$$

(a) Monoidal Categories

$$\frac{\varphi : \mathsf{Prop} \qquad \psi : \mathsf{Prop}}{\varphi \otimes \psi : \mathsf{Prop}}$$

$$\frac{\Gamma \vdash \varphi \qquad \Delta \vdash \psi}{\Gamma \otimes \Delta \vdash \varphi \otimes \psi}$$

(b) Linear Logic

Challenge!

- In the study of linear logic, one encounters **complicated models**
- Lafont's original model uses **comonoids**
- Other models uses **Eilenberg-Moore categories**

Challenge: how do we formalize complicated monoidal categories in a modular way?

Our paper

- We introduce **displayed monoidal categories**
- We use them to construct complicated monoidal categories in a **modular** way
- Nice application of dependent types to category theory
- Formalized using **Coq** and the **UniMath library**

This talk

I will start by illustrating the problem

- Models of linear logic
- Category of comonoids

Then I will discuss displayed categories

- What are displayed categories
- Modularly constructing categories
- Displayed monoidal categories

Models of Linear Logic

Key feature of linear logic:

- All hypotheses must be used **precisely once**
- So: no copying and deletion

Models of Linear Logic

Key feature of linear logic:

- All hypotheses must be used **precisely once**
- So: no copying and deletion

Intuitionistic linear logic has 3 main connectives:

- linear conjunction: $\otimes$
- linear implication: $\multimap$
- bang modality: ! (you can duplicate assumptions under a !)

Models of Linear Logic

Key feature of linear logic:

- All hypotheses must be used **precisely once**
- So: no copying and deletion

Intuitionistic linear logic has 3 main connectives:

- linear conjunction: $\otimes$
- linear implication: $\multimap$
- bang modality: ! (you can duplicate assumptions under a !)

For the semantics:

- the linear conjunction and implication are interpreted via a symmetric monoidal closed category
- the bang modality is more complicated and various proposals have been made

Linear-non-linear models: Intuition

- We have a linear world where we cannot duplicate assumptions
- We have a cartesian world where we can duplicate assumptions
- The ! modality jumps from the linear world to the cartesian world and back

Linear-non-linear models: Precisely

A linear-non-linear model is an adjunction

$$\mathbb{C} \underset{\longleftarrow}{\overset{\longrightarrow}{}} \mathbb{L}$$

where $\mathbb{L}$ is a symmetric monoidal category ($\otimes$ and $\multimap$) and $\mathbb{C}$ is a cartesian category (we can copy and delete hypotheses).
We interpret ! as $\mathbb{L} \to \mathbb{C} \to \mathbb{L}$.

The relation model

The relation model of linear logic

$$\text{Comonoid(Rel)} \underset{C}{\overset{U}{\rightleftarrows}} \text{Rel}$$

Key construction: monoidal category of comonoids

Complicated Monoidal Categories: Comonoids

A **comonoid** (x, ε, δ) in a monoidal category $\mathcal{C}$ consists of

- an object $x : \mathcal{C}$
- a comultiplication $\varepsilon : x \to x \otimes x$
- a counit $\delta : x \to \mathbf{1}$
- Laws: coassociativity and counitality.

Complicated Monoidal Categories: Comonoids

A **comonoid** (x, ε, δ) in a monoidal category $\mathcal{C}$ consists of

- an object $x : \mathcal{C}$
- a comultiplication $\varepsilon : x \to x \otimes x$
- a counit $\delta : x \to \mathbf{1}$
- Laws: coassociativity and counitality.

For the tensor, we need to consider comonoids as a whole
This does not allow for code reuse (i.e., complicated structures of which comonoids form substructure)

Interlude: Group Structures

We can use the following strategy to define the notion of groups.

1. Given a set X, define the type of **group structures** over X
2. A group is a set together with a group structure

This means we define the notion of groups in **2 steps**.

Interlude: Group Structures

We can use the following strategy to define the notion of groups.

1. Given a set X, define the type of **group structures** over X
2. A group is a set together with a group structure

This means we define the notion of groups in **2 steps**.
Displayed categories formalize this idea for categories

Displayed Categories

A displayed category over a category $\mathcal{C}$ consists of

- For every object $x : \mathcal{C}$, a type of structures over x
- For all morphisms $f : x \to y$ and structures S_x and S_y for x and y respectively, a type of structure-preserving maps

Displayed Categories: Example

The displayed category of groups over sets:

- For every set X, a type of group structures for X
- For all functions $f : X \to Y$ and group structures G_X and G_Y, a type expressing that f is a homomorphism

Building Complicated Structures from Simpler Ones

Displayed categories give **modularity**, because we can **untangle** and **stratify** structures.

Basically: build up complicated structures from simpler structures

Building Complicated Structures from Simpler Ones

Displayed categories give **modularity**, because we can **untangle** and **stratify** structures.
Basically: build up complicated structures from simpler structures
For example:

- Product of displayed categories (combines structures)

$$\frac{f : X \to \mathrm{Type} \qquad g : X \to \mathrm{Type}}{h(x) = f(x) \times g(x)}$$

- Adding a destructor (i.e. coalgebra structure)

$$f(x) = x \to x^n$$

We can reason about the these parts **independently**, and we can reuse the results in larger proofs.

Displayed Monoidal Categories, but what are they?

Displayed monoidal categories

$=$

Displayed categories $+$ monoidal categories

Displayed Monoidal Categories, but what are they?

Displayed monoidal categories

$=$

Displayed categories $+$ monoidal categories

Note: there also needs to be a suitable interaction between the two concepts

Displayed Monoidal Categories, but *what* are they?

Let S be a displayed category over $\mathcal{C}$.

$$\frac{x : \mathcal{C} \qquad y : \mathcal{C}}{x \otimes y : \mathcal{C}}$$

(a) Monoidal Categories

$$\frac{x : \mathcal{C} \qquad \overline{x} : S_x \qquad y : \mathcal{C} \qquad \overline{y} : S_y}{\overline{x} \otimes \overline{y} : S_{x \otimes y}}$$

(b) Displayed Monoidal Categories

Comonoids using displayed monoidal categories

Main idea:

- We define a displayed monoidal category that adds a destructor $x \to F(x)$ for a lax monoidal functor F
- This way we acquire the counit ε and the comultiplication δ
- We define the full subcategory via a displayed monoidal category, and that gives us the laws

So: we build up the category of comonoids via smaller pieces and we reason about those smaller parts

Conclusion

- Main take-away: displayed monoidal categories are a technique to modularly build monoidal categories
- In the paper, we define and study displayed monoidal categories
- We apply it to a case study arising from linear logic
- They make the formalization of complicated monoidal categories more convenient and nicer
- Key examples: category of comonoids, Eilenberg-Moore category

Check our paper:
`https://dl.acm.org/doi/abs/10.1145/3636501.3636956`.