Monoidal Categories

We interpret objects of categories as systems and the morphisms as the processes in between them. A monoidal category has additional structure allowing us to consider processes that occur in *parallel*, as well as sequentially. One could interpret this in the following ways:

- letting independent physical systems evolve simultaneously;
- running computer algorithms in parallel;
- taking products or sums of algebraic or geometric structures;
- using separate proofs of and to construct a proof of the conjunction (and).

Monoidal Structure

It is perhaps surprising that a non-trivial theory can be developed from such simple intuition. But, in fact, some interesting general issues quickly arise. For example, let , , and be processes, and write for parallel composition. One might say that should be equal to , as they are different ways to express the same arrangement of systems.

However, this is too strong and might not be the case. For example, if , , and are Hilbert spacesHilbert spaces, and is the usual tensor producttensor product, then these two composite Hilbert spaces are not *exactly equal*, they are only *isomorphic*.

Now we have a new problem, what equations should these isomorphisms satisfy? The theory of monoidal categories is formulated to deal with these issues.

Monoidal Category

A *monoidal category* is a categorycategory equipped with the following data:

- A *tensor product* functorfunctor .
- A *unit object* .
- An *associator* natural isomorphismnatural isomorphism .
- A *left unitor* natural isomorphismnatural isomorphism .
- A *right unitor* natural isomorphismnatural isomorphism

This data must satisfy the triangle and pentagon equations:

center

The naturality conditions for , , and are shown also, diagrammatically:

$$
\begin{tikzcd}
(A \otimes B) \otimes C \arrow[r, "\alpha_{A,B,C}"] \arrow[d, "(f \otimes g) \otimes h"] & A \otimes (B \otimes C) \arrow[d, "f \otimes (g \otimes h)"'] \\
(A' \otimes B') \otimes C' \arrow[r, "\alpha_{A',B',C'}"] & A' \otimes (B' \otimes C')
\end{tikzcd}
\qquad
\begin{tikzcd}
I \otimes A \arrow[r, "\lambda_A"] \arrow[d, "I \otimes f"] & A \arrow[d, "f"] & A \otimes I \arrow[l, "\rho_A"'] \arrow[d, "f \otimes I"'] \\
I \otimes B \arrow[r, "\lambda_B"'] & B & B \otimes I \arrow[l, "\rho_B"]
\end{tikzcd}
$$

The tensor unit object represents the 'trivial' or 'empty' system. This interpretation comes from the unitors and , which witness the fact that the object is 'just as good as', or isomorphicisomorphic to, the objects and .

Each of the triangle and pentagon equations say that two particular ways of 'reorganizing' a system is equal. Surprisingly, this implies that *any* two 'reorganizations' are equal. This is the content of the coherence theorem detailed below.

What the pentagon equation tells us is that the two ways of essentially "rebracketing" the expression both compose into the same object. It also tells us that there are two isomorphisms between and , but these two isomorphisms are equal and compose to the same object.

The triangle equation says that inserting *units* into this is compatible with the content of the pentagon equation too. For example, you can insert or remove the empty string from any word, and that's okay.

Coherence for Monoidal Categories

Given the data of a monoidal category, if the pentagon and triangle equations hold, then any well-typed equation built from , , , and their inverses hold.

In particular, the pentagon and triangle equations imply

Coherence

An intuitive example for the coherence theorem is that if I can prove that , then I can deduce that I can change the brackets however I want on .

Coherence is the fundamental motivating idea for monoidal categories, and gives an answer to the question we posed earlierearlier.

Let's take a look at the monoidal structure we can give to $\textbf{Hilb}$.

Monoidal Structure in .

In the monoidal category , and by restriction :

- **The Tensor Product** is the tensor product of Hilbert spaces, as defined earlierearlier.
- **The Unit Object** is the one dimensional Hilbert space .
- **Associators** are the unique linear maps satisfying for all , , .
- **Left Unitors** are the unique linear maps with for all .
- **Right Unitors** are the unique linear maps with for all .

Although we call the functor of a monoidal category a tensor product, that does not mean we have to choose the actual tensor producttensor product of a Hilbert space. There are other monoidal structures we could choose; a good example is the direct sumdirect sum of Hilbert Spaces. However, the tensor product we have defined previously has a special status, since it describes the state space of a composite system in quantum theoryit describes the state space of a composite system in quantum theory.

While is relevant for *quantum* computation, the monoidal category $\textbf{Set}$ is relevant for *classical* computation. We now add the monoidal structure:

Monoidal Structure in

- **The Tensor Product** is the Cartesian product of sets, written , acting on functions and as .
- **The Unit Object** is a chosen singleton set .
- **Left Unitors** are the functions .
- **Right Unitors** are the functions .

The Cartesian product in is a categorical productcategorical product. This is an example of a general phenomenon: if a category has productsproducts and a terminal objectterminal object, then these furnish the category with monoidal structure. The same is true for coproductscoproducts and initial objectsinitial objects, which in are given by disjoint union.