4/7/03

**Category Theory**

category: things, homomorphisms bet those things

- in many categories - we can "combine" 2 objects
  - ex) \( V \otimes W \) or \( V \oplus W \) for \( V, W \) v. spaces \( \otimes, \oplus \)
  - gives monoidal category.

But - we want to do things for morphisms as well!

**objects**

- \( V, W \in \text{Vect} \)
- give \( V \otimes W \in \text{Vect} \)

**morphisms**

- \( f: V \to V', \ g: W \to W' \)
- linear maps give \( f \circ g: V \otimes W \to V' \otimes W' \)
- \( f \circ g(v \otimes w) = f(v) \otimes g(w) \)

We draw morphisms as:

\[ \begin{array}{c}
\circ \vline \\
\text{v} \\
\circ \\
\text{w} \\
\end{array} \quad \circ \vline \\
\text{v'} \\
\circ \\
\text{w'} \]

"parallel"

**circuits:**

- parallel: \( \circ \vline \text{v} \quad \circ \vline \text{w} \)
- series: \( \circ \vline \text{v} \quad \circ \vline \text{w} \)

**circuits:**

- electrical engineering
Monoids

**Defn:**

A monoid is a set \( M \) with an associative, binary operation \( \cdot \) called multiplication, together with a left/right unit \( 1 \in M \).

**Ex:** \((\mathbb{N}, +)\)

We'll now replace the word "set" by other words, to transpose this idea into other contexts.

So - we'll rewrite above defn. as

A monoid is a set \( M \) together with a function \( \cdot : M \times M \to M \) and a function \( i : 1 \to M \)

- We've taken the unit 1 above and turned it into the function \( i \) +
- So \( i(+) = 1 \in M \)

such that (write out conditions in terms of commutative diagrams)

\[
\begin{array}{ccc}
M \times M \times M & \xrightarrow{\cdot} & M \times M \\
\downarrow & & \downarrow \\
M \times M & \xrightarrow{\cdot} & M \\
\end{array}
\]

commutes (this is assoc. law)

\[
\begin{array}{ccc}
M & \xrightarrow{\sim} & M \times 1 \\
\downarrow & & \downarrow \\
M \times M & \xrightarrow{\cdot} & M \times M \\
\end{array}
\]

commutes \((x \cdot 1_m = x)\)

\[
\begin{array}{ccc}
M & \xrightarrow{\sim} & M \times 1 \\
\downarrow & & \downarrow \\
M \times M & \xrightarrow{\cdot} & M \times M \\
\end{array}
\]

left unit law
and corresponding right unit law:

1-elt set \( \overset{i \times 1_m}{\rightarrow} 1 \times M \xleftarrow{\sim} M \)

\[
\begin{array}{c}
1 \times 1_m \downarrow \quad \circ \quad \downarrow 1_m \\
M \times M \quad \quad \quad \quad M
\end{array}
\]

(right unit law)

This defn allows us to generalize a monoid to other categories!

**monoid object:**

We use this defn to internalize the defn of monoid — replace \textbf{set} a \textbf{function}
by \textbf{object} of \textbf{C} and \textbf{morphism} of \textbf{C}

where \textbf{C} is any \textbf{category w/ finite products}
(or more generally, in any \textbf{monoidal category})

\textbf{Note:} Not true that terminal obj. is the \textbf{unit object} for product.

Ex) In Vect — terminal obj. is 1-dim l. v. space
and unit for \(\otimes\) is ground field, \(\mathbb{K}\)

Ex) If \(C\) is Vect, a monoid object in \(C\) is an (associative) algebra.

Ex) If \(C\) is Top, a monoid object in \(C\) is a \textbf{topological monoid}.

\(1 = \text{-pt set}\) (like \text{topological grp})
Category \( \text{op} \) — formally turn all arrows around, all objects remain the same.

\( f: x \to y \) in \( C \) is \( f: y \to x \) in \( C \)  
\( f \circ g \) in \( C \) is \( g \circ f \) in \( C \)

Ex) If \( C = \text{Vect} \), a monoid in \( C \) is a coalgebra.

Note — Doing \( \text{op} \) twice gets us back where we've started.

\( X \) is whatever — space, set

Defn: A \( \text{co-} X \) in \( C \) is an \( X \) in \( C \text{op} \).

Ex) In \( C = \text{Set} \), a monoid in \( C \) is a co-monoid.

Try: Classify all comonoids.

Ex) In \( C = \text{Cat} \) — category of all small categories.

(Problem: set of all sets that contain themselves — paradox)

A monoid in \( C \) is a strict monoidal category.
The good ones are the weak ones!

Defn: A strict monoidal category consists of a category \( M \), a functor

\[ \otimes: M \times M \to M \]

a functor

\[ \mathbf{I}: \mathbf{I} \to M \]

\[ \mathbf{I} \text{ is the terminal category (terminal obj.)} \]

\[ \text{I is the category w/ 1 elt and 1 morphism: } 1_x \]

This is the unit for \( \otimes \).
\( i(I) = 1 \in M \)

S.t. \( \otimes \) is associative \& left \& right unit laws hold, i.e. diagrams on prev pg commute.

Thus: \( (x \otimes y) \otimes z = x \otimes (y \otimes z) \quad x, y, z \in M \)

\( 1 \otimes x = x = x \otimes 1 \)

But - in most categories these are only isomorphic!

Ex) In \( \text{Vect} \), \( (V \otimes W) \otimes Z \cong V \otimes (W \otimes Z) \)

\( C \otimes V \cong V \)

In practice - we should weaken this defn. 
so that these eqns become isos:

Defn:

\[ \text{associator: } \alpha_{x, y, z}: (x \otimes y) \otimes z \xrightarrow{\sim} x \otimes (y \otimes z) \]

\[ \text{left/right units } \begin{cases} \ell_x: 1 \otimes x & \xrightarrow{\sim} x \\ r_x: x \otimes 1 & \xrightarrow{\sim} x \end{cases} \]

But this still isn't good enough! It's not okay to just change eqns to isos.
In any category w/ way of glomming 2 things together, \( @ \), etc we can form a weak monoidal category.

For example — in Vect, there are 5 ways to parenthesize 4 things:

\[
\begin{align*}
(w \otimes x) \otimes (y \otimes z) & \\
(w \otimes (x \otimes y)) \otimes z & \\
(w \otimes (x \otimes y)) \otimes z & \\
(w \otimes (x \otimes y)) \otimes z & \\
(w \otimes (x \otimes y)) \otimes z & \\
\end{align*}
\]

We want a unique iso bet things!

Just one way for them to be the same.

Otherwise — if not unique, the two things have different ways of being the same.

This diagram commutes in Vect.

So — since just adding \( \alpha \), \( l \), \( r \) isn’t enough, we also demand that the above pentagon commutes (pentagon id). We also get a diagram for \( l \), \( r \) which we demand commutes.

Then, by MacLane’s coherence thm, we get:

All diagrams built using \( \otimes \), \( \times \), \( l \), \( r \) commute.

Adding in these diagrams we get the defn. of a weak monoidal category.
weak monoidal categories — what show up in nature.

Groups are used more often than monoids - so we do above internalization to get a group object — which has a functor:

\[ \text{inv: } G \rightarrow G. \]

So— if we do this all for groups, we get "groups in C" and if \( C = \text{Cat} \), we get a "groupal category" or strict 2-group.

What we really want are "weak 2-groups"