A - functor from paths to group

Baez seminar

1/22/02

Gauge Theory

Want $A(X) \in G$, $G$ same group "gauge group"

$A(\delta)A(\delta') = A(\delta \cdot \delta')$

category: objects are points, paths

We can think of a group as a category:

objects: 1 object

morphisms: maps from elt to itself

(all morphisms are invertible)

2 category (paths between paths)

$\alpha : \delta \Rightarrow \delta'$

$G$ will be a "2-group" - a 2-category

w/ one object and all morphisms and 2 morphisms are equivalences.

We need this if string theory is actually true -

"a particle moving" is really a string moving like
Recall—Any topological space $X$ gives an $n$-category $\mathcal{T}(X)$ where
- objects are pts. of $X$
- morphisms are paths in $X$
- 2-morphisms are paths of paths
  etc.

If we're studying $2$-categories, we don't need the etc.

Note: We have a way to, given any $n$,
take an $n$-category and chop off
top part to get an $n$-category.

Decategorification:

From an $n$-category $C$ we get an
$n$-category $C[n]$ with:
- objects in $C[n] = \text{objects in } C$
- morphisms in $C[n] = \text{morphisms in } C$
how to get an n-category, we need everything above n to be an identity.

So - define

* n-morphisms in $C[n] = \text{equivalence classes of n-morphisms in } C$.

where given 2 n-morphisms

$f, g : x \to y \text{ in } C$, we say $f \sim g$ if there is an equivalence 

$\alpha : f \Rightarrow g$

(i.e., $\exists \alpha^{-1} : g \Rightarrow f$ that's not a true inverse, but inverse up to equivalence)

* j-morphisms in $C[n] \ j > n$ are identities

what we're faced with now - we've handed the decategorification of some mathematical thing and we want to know what the categorification is.

Recall in $\pi(X)$, all morphisms are equivalences.

Fact: If C is an $n$-groupoid (all $j$-morphisms are equivalences) then $C[n]$ is an $n$-groupoid ($n$-cat w/ all $j$-morphisms equivalences).
So: $\pi(X)$ gives by decategorification an $n$-groupoid: $\mathbf{T}_n(X)$.

$n=0$ $\mathbf{T}_0(X)$ is a set

A zero morphism is an object. We form equivalence classes of pts where 2 points are equivalent if connected by a path.

Equivalence classes of points, where $x \sim y$ if there's a path from $x$ to $y$. We call this the set of path-components.

Decategorification – we only care if we can get from $x$ to $y$, we don’t care how (or how many ways).
note: working up to equivalences gives no eqns at upper levels

\( n=1 \quad T_1(x) \) is a groupoid (no base pt for groupoid)

ie) A category where all morphisms have inverses.

- objects = points of \( X \)
- morphisms = homotopy classes of paths
  (we regard \( f \circ g \) if can get to one from other than deformation/homotopy)
  here we take equiv. composition must be associative on the nose classes

\[
\begin{array}{c}
\text{Fundamental group (only contractible loops at } x) \\
\pi_1(X,x) = \text{hom}(x,x)
\end{array}
\]

the fund gp is a microcosm of the fund. groupoid (only restrict ourselves to loops at a pt)
\( T_1(X) \) is called the fundamental groupoid.

\( n=2 \quad \Pi_2(X) \) is a 2-groupoid.

Example: \( S^2 \)

\[ \alpha : 1_x \Rightarrow 1_x \]

This path of paths tells us there is a 2-dim'l hole.

So \( \alpha \neq 1_{2x} \)

Goal: Tell what \( \Pi_n(X) \) tells us about our space.

\( \Pi_2(X) \) includes the info of \( \Pi_1(X) \) (fund. grp.) and \( \Pi_2(X) \) (homotopy class) and how they relate.

To-Do List

1) \( \Pi_2 \) is not right for categorified gauge theory. We need "thin \( \Pi_2 \)."

2) Understand 2-groupoids:
   - strict vs. weak
   - laws are up to something
   - hold on the nose

3) Understand 2-groups: A 2-groupoid w/ only 1 object.
Just as a group was a groupoid (cat. we see morphisms equivalence)
we one object,
define a 2-group to be a 2-groupoid
we one object.
Again we have strict vs. weak.

**Strict 2-categories**

A strict 2-category \( \mathcal{C} \) is:
1. a set \( \mathcal{C}_0 \) of objects
2. for any \( x, y \in \mathcal{C}_0 \), a set of morphisms \( f : x \rightarrow y \)
3. for any morphisms \( f, g : x \rightarrow y \), a set of 2-morphisms \( \alpha : f \Rightarrow g \)
4. operations satisfying various laws "on the nose":
   1. \( f \circ g \) gives composition of morphisms
      \[ f : x \rightarrow y, \quad g : y \rightarrow z \]
      gives \( fg : x \rightarrow z \)
   2. \( 1_x : x \rightarrow x \)
      (there are binary & nullary composition)

   comp. of 2 or 6 things

   satisfying:

   "associativity" \( (fg)h = f(gh) \)

   \( 1_xf = f \) left unit law

   chop up into 0-diml cell, \( e \)
\[ f \circ 1_y = f \quad \text{right unit law} \]

vertical composition \( \alpha, \beta \) give of 2-morphisms \( \alpha \circ \beta \)

\[ x \quad f \quad y \]
gives identity 2-morphisms \( f \) gives \( 1_f : f \Rightarrow f \)

satisfying

associativity

left/right unit laws for vertical comp.

\[ f \circ h \]

horizontal comp.
\( \alpha : f \Rightarrow g, \beta : h \Rightarrow i \)
give \( \alpha \cdot \beta : fg \Rightarrow hi \)

satisfying

associativity

left/right unit laws
(Note - we can prove everything about whiskering from what we've defined.)

**AND**

\[
\begin{array}{ccc}
\alpha & \cdot & \beta \\
\downarrow & & \downarrow \\
\alpha' & \cdot & \beta' \\
\end{array}
\]

interchange law for vert/horiz composition

\[(\alpha \cdot \beta) \cdot (\delta \cdot \epsilon) = (\alpha \cdot \delta) \cdot (\beta \cdot \epsilon)\]

But we can write this defn more simply:

If we categorify a category, we should get a 2-category.

**Defn**

A category is a set \( C \) of objects,

\( \forall x, y \in C \), a set \( \text{hom}(x, y) \) of morphisms,

\( \forall x \in C \), an identity element \( 1_x \in \text{hom}(x, x) \)

\( \forall x, y, z \) a composition function

\( \circ : \text{hom}(x, y) \times \text{hom}(y, z) \rightarrow \text{hom}(x, z) \)

Satisfying left/right unit laws i.e. associativity holds.

Now - we can categorify this object to get a 2-category.
"Categorified defn of category"

we get:

- A set \( C \) of objects
- A objects \( x, y \in C \), a category \( \text{hom}(x, y) \) of morphisms
- A objects \( x \in C \), an identity object \( 1_x \in \text{hom}(x, x) \) (element becomes an object)
- A \( x, y, z \), a composition function
  \( o : \text{hom}(x, y) \times \text{hom}(y, z) \to \text{hom}(x, z) \)

st left/right unit laws and assoc. hold
  on the nose (strict).

This above defn is what we call
  a 2-category.

(had we categorified everything correctly
  we'd get a double-category)

let's show this is the same as our
defn of 2-category on prev pg.

right unit laws
objects of $\text{ham}(x,y)$

morphisms of $\text{ham}(x,y)$

These look like morphisms $\alpha, \beta$ - morphisms

Since $\text{ham}(x,y)$ is a category, we can compose the double arrows, so we get vertical composition.

Now $\text{ham}(x,y) \times \text{ham}(y,z)$ is a product of categories

On objects, $\alpha : \text{ham}(x,y) \times \text{ham}(y,z) \to \text{ham}(x,z)$

On objects, $\alpha : \text{ham}(x,y) \times \text{ham}(y,z) \to \text{ham}(x,z)$

$\alpha$ is a morphism, so $\alpha$ sends the object $x \to f$.

Hence $\alpha$ gives $x \to f$.

$\alpha$ gives $\text{normal comp.}$
on morphisms, composition gives

\[(\alpha, \beta) \in \text{hom}(x, y) \times \text{hom}(y, z)\]

gives

\[x \circ (\alpha \circ \beta) \circ z\]

horizontal composition

We need to check the laws ...

most interesting: the interchange law

\[\text{a functor} \quad \text{vertical camp is from fact that } \text{hom}(x, y) \text{ a cat.}

means we can do in either order

vert. camp \*check this*

then horiz. camp

or vice versa