N-Categories

- want to categorify notion of group (called 2-group)
- A 2-group will be a category
- 2-groups will be good for parallel transport on 2-manifolds
- more generally than N-Categories are

w-Categories

Defn: An w-category has:

- objects (0-morphisms)
- morphisms (1-morphisms) \( sf \bullet f \bullet +f \) normal
  (here we’d stop if we were just talking about categories)
- 2-morphisms

A morphism \( f \) has a source and target \( (sf, tf) \)
3-morphisms

\[ s(s(s(f))) \rightarrow t(t(f)) \]

Arrows coming around sphere are 2-morphisms

or - we can draw a 3-morphism as

\[
\begin{align*}
\text{4-morphisms:} \\
\end{align*}
\]

\[ s(f) \rightarrow t(f) \]
**Defn:** A **globular set** \( C \) is \( C_0, C_1, C_2, \ldots \)

where \( f \in C_n \) is a "\( n \)-globe" or "\( n \)-cell"

and functions

\[
\begin{array}{cccc}
C_0 & 
C_1 & 
C_2 & 
C_3 & 
C_4 & 
\ldots \\
\leftarrow s & \leftarrow s & \leftarrow s & \leftarrow s & \leftarrow s \\
\downarrow t & \downarrow t & \downarrow t & \downarrow t & \\
\end{array}
\]

s.t.

\[
s(s(f)) = s(+(f))
\]

\[
+(+(f)) = +(s(f))
\]

An \( \omega \)-category (strict or weak) will be a globular set \( C \) with a bunch of "globe-glomming" operations such as:

\[
\begin{array}{ccc}
\circ f & g \\
x & y & z \\
\end{array}
\]

**composition**

\[
\begin{array}{ccc}
\circ fg \\
x & z \\
\end{array}
\]

"composition"
In a 2-category: (we have many ways of composing)

```
```

"vertical composition"

or - "horizontal composition"

Ex)
3-cells have 3 kinds of composition:

1. \( x \xrightarrow{f} y \xrightarrow{g} z \) giving \( f_g \)

2. Arrows point toward something 1 dimension down.

3. \( a \xrightarrow{f} b \) giving \( ac \xrightarrow{f \cdot g} bd \)

4. Giving \( p \) and \( s \)
M. Batanin figured out a notation for all globe-glopping operations in an $\omega$-category.

In a strict $\omega$-category there is one globe-glopping operation for each $n, m \in \mathbb{N}$ st $n \leq m$ and each $n$-dimensional cell colony.

Exs of cell-colonies: (cell colony = these patterns which give relationships of sources/targets)

Ex)

2-dim'l cell colony
since cells of highest dim is 2

These pictures tell us relations about targets & sources

Ex)

3-d cell colony

These cell-colonies are related to trees:
2d cell-colony

Ex)

E. Tanin:

Cell colonies correspond to planar trees
finite

4 branches
from 4 one-cells on top.

What is the m \in \mathbb{N}?

Given an object \( x \), we want to get a morphism: we get \( \text{id}_x \)

\[ x \xrightarrow{\text{id}_x} x \]

zero all one-cell
In a strict $\mathcal{w}$-category, given a bunch of cells arranged in the shape of an $n$-dimensional cell colony and given $m \geq n$, our globe-glomming operation gives us an $m$-cell in our $\mathcal{w}$-category.

\[ n = 1 \]
\[ \xrightarrow{f} \quad \xrightarrow{g} \quad x, y, z \in C, \quad f, g \in C, \]

[cell-colony tells us:]

\[ t(f) = y, \quad s(g) = y. \]

This is a 1-dim'le cell category, so $m = 1$ gives a 1-cell.

\[ m = 2 \text{ gives a 2-cell} \]

identity cell

\[ x \xrightarrow{1_{fg}} y \]

\[ fg \]
m = 3
get identity of the identity of fg

We call the m-cell we get this way a "composite" m-cell.

The source and target of this composite are equal to certain composites of sources and targets of the cells we're composing. Following a rule too tedious to describe here.

e.g.

n = 3
m = 3
(output)

\[ c \cdot f : c \cdot a \rightarrow c \cdot b \]

Imagine f again going into the board.

c \cdot a : have composite
we've basically defined a w-category but we haven't specified any laws!

In a strict w-category, globe-glomming operations satisfy equational laws (like assoc, etc)
(this is evil)

What are they?
* They are all possible laws! (between globe-glomming operations)

We can glum globes all at once or a bit at a time and we always get the same result.

ie - have more than one way of doing something, do any way and get same thing.

Ex)

\[
\begin{array}{c}
\text{f} \\
\text{g} \\
\text{h}
\end{array}
\]

gives

\[
\begin{array}{c}
f \circ g \\
h
\end{array}
\]

all at once

or

\[
\begin{array}{c}
f \\
g \circ h
\end{array}
\]

a bit at a time.
ie) get half-assoc. law

\[(fg)h = f(gh)\]

so - associativity comes from 2 "half-associativity" laws.

Ex)

vertically compose: \((ab) \cdot (cd)\)

\[= \text{all at once}\]

--- horizontally compose: \((a \cdot c) \cdot (b \cdot c)\)

Then vertically compose result