A Type Theory for Strictly Unital ∞-Categories

Eric Finster David Reutter Alex Rice Jamie Vicary

LICS 2022

UNIVERSITY OF CAMBRIDGE

Infinity categories contain:

An overview of (globular) infinity categories

Infinity categories contain:

- Objects x, y, z

An overview of (globular) infinity categories

Infinity categories contain:

- Objects x, y, z
- 1-arrows:

$$x \xrightarrow{f} y$$

An overview of (globular) infinity categories

Infinity categories contain:

- Objects x, y, z
- 1-arrows:

$$x \xrightarrow{f} y$$

- 2-arrows:

$$\begin{tikzcd} x \arrow[r, bend left, "f"] \arrow[r, bend right, "g"'] \arrow[r, phantom, "\Uparrow\alpha"] & y \end{tikzcd}$$

Infinity categories contain:

- Objects x, y, z
- 1-arrows:

$$x \xrightarrow{f} y$$

- 2-arrows:

$$\begin{tikzcd} x \arrow[r, bend left, "f"] \arrow[r, bend right, "g"'] \arrow[r, phantom, "\Uparrow\alpha"] & y \end{tikzcd}$$

- ...

An overview of (globular) infinity categories

Infinity categories contain:

- Objects x, y, z
- 1-arrows:

$$x \xrightarrow{f} y$$

- 2-arrows:

$$\begin{tikzcd} x \arrow[r, bend left, "f"] \arrow[r, bend right, "g"'] \arrow[r, phantom, "\Uparrow\alpha"] & y \end{tikzcd}$$

- ...

Our arrows are *Globular*.

Infinity categories contain:

- Objects x, y, z
- 1-arrows:

$$x \xrightarrow{f} y$$

- 2-arrows:

$$\begin{tikzcd} x \arrow[r, bend left=50, "f"] \arrow[r, bend right=50, "g"'] \arrow[r, phantom, "\Uparrow\alpha"] & y \end{tikzcd}$$

- ...

Our arrows are *Globular*.

Compositions:

$$\begin{tikzcd} x \arrow[r, bend left=50, "f"] \arrow[r, bend right=50, "g"'] \arrow[r, phantom, "\Uparrow\alpha"] & y \arrow[r, bend left=50, "h"] \arrow[r, bend right=50, "i"'] \arrow[r, phantom, "\Uparrow\beta"] & z \end{tikzcd}$$

$$\begin{tikzcd} x \arrow[r, bend left=70, "f"] \arrow[r, "h" description] \arrow[r, bend right=70, "g"'] \arrow[r, phantom, shift left=3, "\Uparrow\beta"] \arrow[r, phantom, shift right=3, "\Uparrow\alpha"] & y \end{tikzcd}$$

Infinity categories contain:

- Objects x, y, z
- 1-arrows:

$$x \xrightarrow{f} y$$

- 2-arrows:

$$\begin{tikzcd} x \arrow[r, bend left, "f"] \arrow[r, bend right, "g"'] \arrow[r, phantom, "\Uparrow\alpha"] & y \end{tikzcd}$$

- ...

Our arrows are *Globular*.

Compositions:

$$\begin{tikzcd} x \arrow[r, bend left, "f"] \arrow[r, bend right, "g"'] \arrow[r, phantom, "\Uparrow\alpha"] & y \arrow[r, bend left, "h"] \arrow[r, bend right, "i"'] \arrow[r, phantom, "\Uparrow\beta"] & z \end{tikzcd}$$

$$\begin{tikzcd} x \arrow[r, bend left=60, "f"] \arrow[r, "h" description] \arrow[r, bend right=60, "g"'] \arrow[r, phantom, bend left=30, "\Uparrow\beta"] \arrow[r, phantom, bend right=30, "\Uparrow\alpha"] & y \end{tikzcd}$$

Identities:

$$x \xrightarrow{\mathrm{id}_x} x$$

Strict Infinity Categories

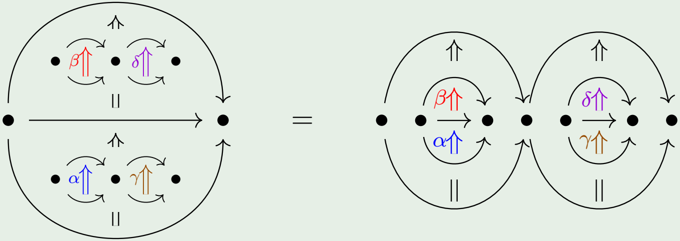
Associativity

$$w \xrightarrow{\;w \xrightarrow{f} x \xrightarrow{g} y\;} y \xrightarrow{h} z \quad = \quad w \xrightarrow{f} x \xrightarrow{\;x \xrightarrow{g} y \xrightarrow{h} z\;} z$$

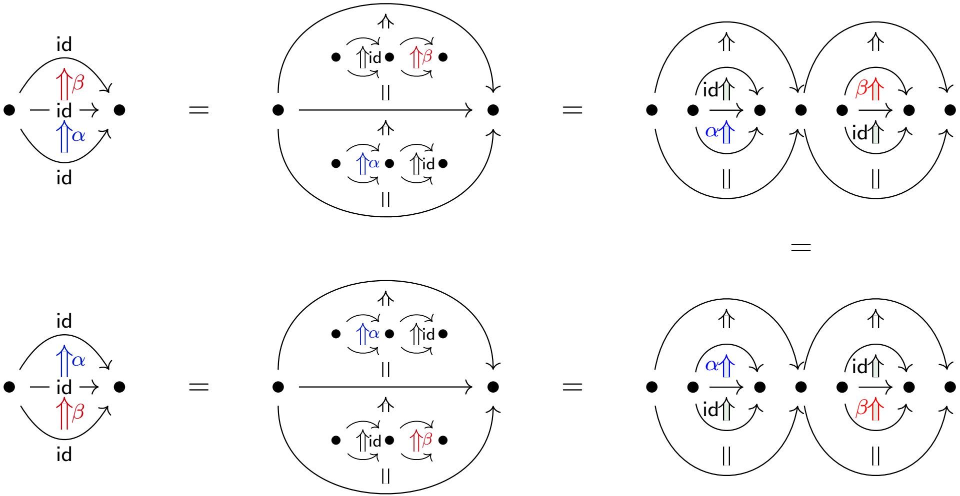
Unitality

$$\begin{tikzcd} x \arrow[r, bend left, "\mathrm{id}(x)"] \arrow[r, bend right, "\mathrm{id}(x)"'] \arrow[r, phantom, "\Uparrow \mathrm{id}(\mathrm{id}(x))"] & x \arrow[r, bend left, "g"] \arrow[r, bend right, "f"'] \arrow[r, phantom, "\Uparrow \alpha"] & y \end{tikzcd} \quad = \quad \begin{tikzcd} x \arrow[r, bend left, "g"] \arrow[r, bend right, "f"'] \arrow[r, phantom, "\Uparrow \alpha"] & y \end{tikzcd}$$

Interchange



Example: Eckmann-Hilton



Weakness

We have only so far talked about strict ∞-categories.

Weakness

We have only so far talked about strict ∞-categories.

In higher categories, non-equal arrows can be isomorphic.

In a weak higher category, the laws are only required to hold up to isomorphism.

Weakness

We have only so far talked about strict ∞-categories.

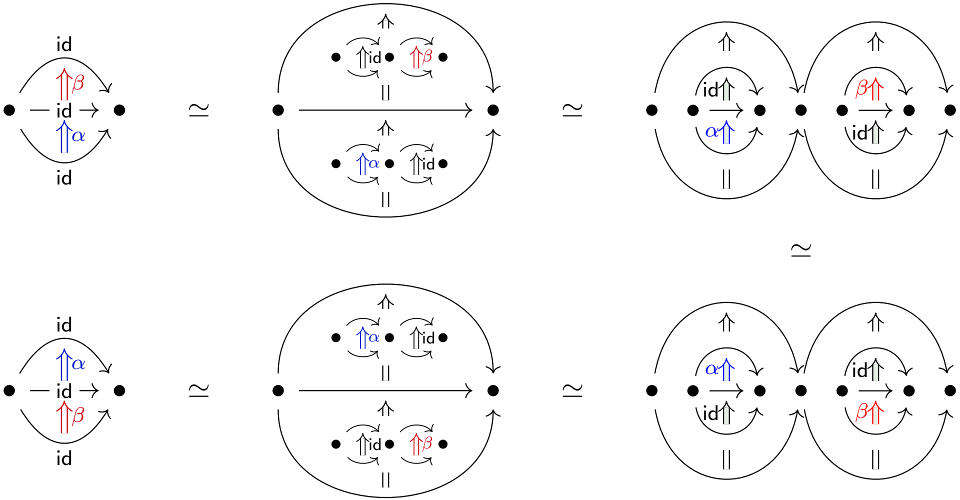
In higher categories, non-equal arrows can be isomorphic.

In a weak higher category, the laws are only required to hold up to isomorphism.

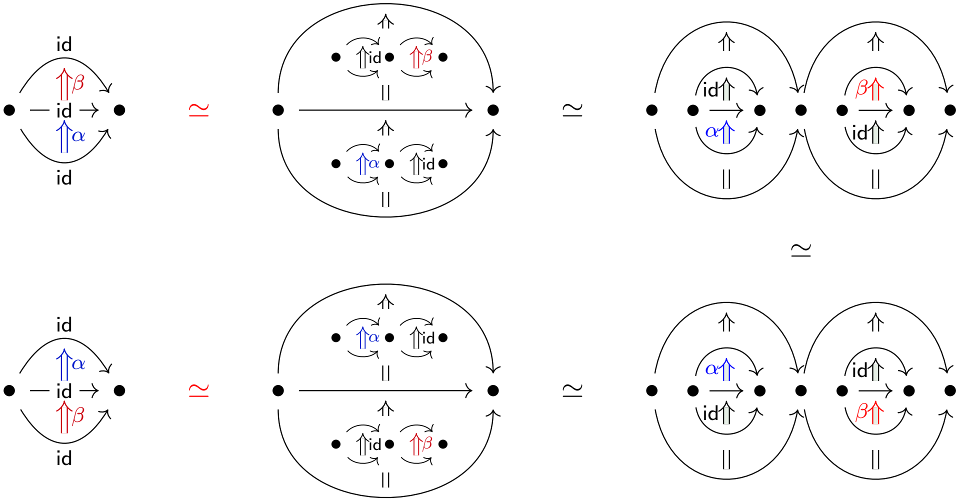
Many examples of higher categories are weak:

- Homotopy groupoids of topological spaces.
- Equality types in HoTT.
- Bicategory of categories and profunctors.

Example: Eckmann-Hilton



Example: Eckmann-Hilton



Weakness vs Strictness

Weak $\longleftrightarrow$ Strict

Weakness vs Strictness

Weak $\longleftrightarrow$ Strict

Harder to use

Easier to use

Weakness vs Strictness

Weak	$\longleftrightarrow$	Strict
Harder to use		Easier to use
More expressive		Less expressive

Weakness vs Strictness

Weak	$\longleftrightarrow$	Strict
Harder to use	?	Easier to use
More expressive		Less expressive

Weakness vs Strictness

Weak $\longleftarrow$ Semistrict $\longrightarrow$ Strict

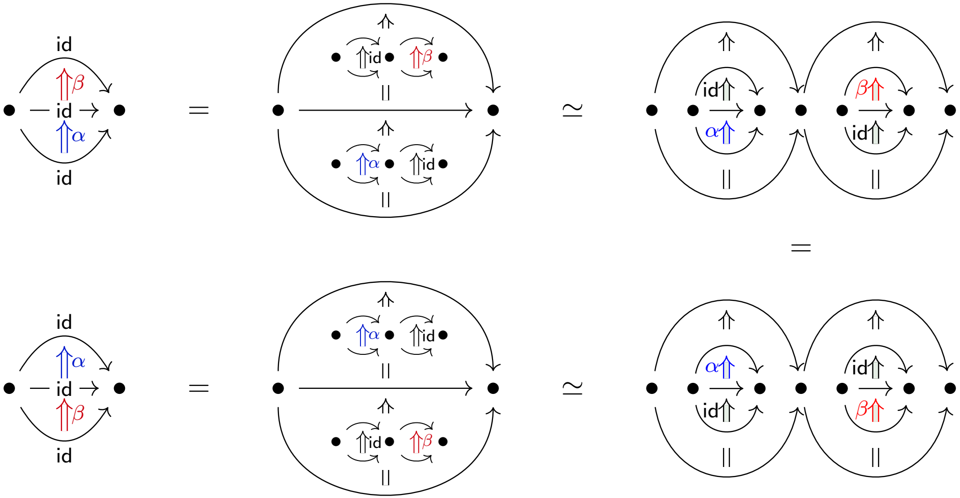
Harder to use	Easier to use
More expressive	Less expressive

Weakness vs Strictness

Weak ⟵	Semistrict	⟶ Strict
Harder to use	Lack of definitions	Easier to use
More expressive	Retains expressiveness?	Less expressive

- Catt [1] is a type theory for weak ∞-categories.
- Its terms are the possible operations in an ∞-category.
- By adding a definitional equality to Catt, we can unify certain operations.
- Catt$_{\mathsf{su}}$ is a new type theory based on Catt with strict units.

Example: Eckmann-Hilton



```
coh id C (x) : x => x
coh id2 C (x(f)y) : f => f
coh comp C (x(f)y(g)z) : x => z
coh vert C (x(f(a)g(b)h)y) : f => h
coh horiz C (x(f(a)g)y(h(b)k)z) : comp f h => comp g k

coh swap3 C (x(f(a)g)y(h(b)k)z)
  : vert (horiz a (id2 h)) (horiz (id2 g) b) =>
    vert (horiz (id2 f) b) (horiz a (id2 k))

let eh {C : Cat} {x :: C} (a :: id x => id x) (b :: id x => id x)
  : [ vert a b => vert b a ]
  = swap3 a b
```

Properties of $\mathsf{Catt}_{\mathsf{su}}$

- Equality in $\mathsf{Catt}_{\mathsf{su}}$ preserves typing.
- Equality is generated by a strongly-terminating, confluent reduction relation.
- Equality and type checking are decidable.
- All terms (of the same dimension) in a disc context are identified.
- Eckmann-Hilton and the Syllepsis have been formalised in $\mathsf{Catt}_{\mathsf{su}}$.

References

[1] Eric Finster and Samuel Mimram. "A Type-Theoretical Definition of Weak ωCategories". In: *Proceedings of LICS 2017*. arXiv:1706.02866. 2017.