

# A Type Theory for Strictly Associative $\infty$ -Categories

Alex Rice   Eric Finster   Jamie Vicary

SYCO 10



UNIVERSITY OF  
CAMBRIDGE

- 1 Weak Globular Infinity Categories
- 2 Type Theories for Infinity Categories
- 3 Strict Associators

*Globular sets* are one natural shape of higher categories.

*Globular sets* are one natural shape of higher categories. They contain:

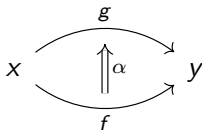
- A set of objects or 0-cells  $G$ .

*Globular sets* are one natural shape of higher categories. They contain:

- A set of objects or 0-cells  $G$ .
- For each pair of objects  $x, y \in G$ , a set of arrows or 1-cells with source  $x$  and target  $y$ .

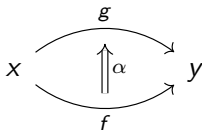
*Globular sets* are one natural shape of higher categories. They contain:

- A set of objects or 0-cells  $G$ .
- For each pair of objects  $x, y \in G$ , a set of arrows or 1-cells with source  $x$  and target  $y$ .
- For each pair of parallel arrows  $f, g$ , a set of 2-arrows (or 2-cells) from  $f$  to  $g$ .



*Globular sets* are one natural shape of higher categories. They contain:

- A set of objects or 0-cells  $G$ .
- For each pair of objects  $x, y \in G$ , a set of arrows or 1-cells with source  $x$  and target  $y$ .
- For each pair of parallel arrows  $f, g$ , a set of 2-arrows (or 2-cells) from  $f$  to  $g$ .



- ...

## Composition of 1 cells

$$\bullet \xrightarrow{f} \bullet \xrightarrow{g} \bullet$$



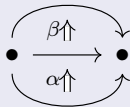
# Composition in Globular Sets

## Composition of 1 cells

$$\bullet \xrightarrow{f} \bullet \xrightarrow{g} \bullet$$

## Composition of 2 cells

Composition along a 1-boundary:



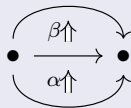
# Composition in Globular Sets

## Composition of 1 cells

$$\bullet \xrightarrow{f} \bullet \xrightarrow{g} \bullet$$

## Composition of 2 cells

Composition along a 1-boundary:



Composition along a 0-boundary:



- In strict category theory, we add equalities between certain arrows.
- In higher category theory we can instead require that equivalences exist between certain arrows.

- In strict category theory, we add equalities between certain arrows.
- In higher category theory we can instead require that equivalences exist between certain arrows.

## Coherence

- For a 1-cell  $f : x \rightarrow y$ , there are unitors  $\lambda_f : \text{id}_x \circ f \rightarrow f$  and  $\rho_f : f \circ \text{id}_y$ .
- $\lambda_{\text{id}_x}$  and  $\rho_{\text{id}_x}$  are both arrows  $\text{id}_x \circ \text{id}_x \rightarrow \text{id}_x$ .
- These should be equivalent.

- *Strict* categories are easier to work with while there are more examples of *weak* categories.

- *Strict* categories are easier to work with while there are more examples of *weak* categories.
- All weak monoidal categories and all weak 2-categories are equivalent to a strict version of themselves.

- *Strict* categories are easier to work with while there are more examples of *weak* categories.
- All weak monoidal categories and all weak 2-categories are equivalent to a strict version of themselves.
- However this is no longer possible at dimensions 3 and higher.

- Since full strictification is not possible, we want to do the best possible.



- Since full strictification is not possible, we want to do the best possible.
- Therefore, we look for *semistrict* definitions of infinity categories.

- Since full strictification is not possible, we want to do the best possible.
- Therefore, we look for *semistrict* definitions of infinity categories.
- We can strictify:

---

Associators

Unitors

Interchangers

- Since full strictification is not possible, we want to do the best possible.
- Therefore, we look for *semistrict* definitions of infinity categories.
- We can strictify:

Strict $\infty$ - <b>Cat</b>	
Associators	✓
Unitors	✓
Interchangers	✓

- Since full strictification is not possible, we want to do the best possible.
- Therefore, we look for *semistrict* definitions of infinity categories.
- We can strictify:

	Strict $\infty$ - <b>Cat</b>	Simpson
Associators	✓	✓
Unitors	✓	
Interchangers	✓	✓

- Since full strictification is not possible, we want to do the best possible.
- Therefore, we look for *semistrict* definitions of infinity categories.
- We can strictify:

	Strict $\infty$ - <b>Cat</b>	Simpson	Grey
Associators	✓	✓	✓
Unitors	✓		✓
Interchangers	✓	✓	

- Since full strictification is not possible, we want to do the best possible.
- Therefore, we look for *semistrict* definitions of infinity categories.
- We can strictify:

	Strict $\infty$ - <b>Cat</b>	Simpson	Grey	CaTT <sub>su</sub> <sup>1</sup>
Associators	✓	✓	✓	
Unitors	✓		✓	✓
Interchangers	✓	✓		

---

<sup>1</sup>Finster, Reutter, et al., *A Type Theory for Strictly Unital  $\infty$ -Categories*

- Since full strictification is not possible, we want to do the best possible.
- Therefore, we look for *semistrict* definitions of infinity categories.
- We can strictify:

	Strict $\infty$ - <b>Cat</b>	Simpson	Grey	CaTT <sub>su</sub> <sup>1</sup>	CaTT <sub>sa</sub> <sup>2</sup>
Associators	✓	✓	✓		✓
Unitors	✓		✓	✓	
Interchangers	✓	✓			

---

<sup>1</sup>Finster, Reutter, et al., *A Type Theory for Strictly Unital  $\infty$ -Categories*

<sup>2</sup>Finster, R., and Vicary, *A Type Theory for Strictly Associative Infinity Categories*

CaTT is a type theory for *weak infinity categories*<sup>3</sup>.

---

<sup>3</sup>Finster and Mimram, *A Type-Theoretical Definition of Weak  $\omega$ -Categories*.



CaTT is a type theory for *weak infinity categories*<sup>3</sup>.

There are 4 pieces of syntax, all defined by mutual induction:

- Contexts: Generating data of an infinity category.

---

<sup>3</sup>Finster and Mimram, *A Type-Theoretical Definition of Weak  $\omega$ -Categories*.

CaTT is a type theory for *weak infinity categories*<sup>3</sup>.

There are 4 pieces of syntax, all defined by mutual induction:

- Contexts: Generating data of an infinity category.
- Terms: Operations in an infinity category.

---

<sup>3</sup>Finster and Mimram, *A Type-Theoretical Definition of Weak  $\omega$ -Categories*.

CaTT is a type theory for *weak infinity categories*<sup>3</sup>.

There are 4 pieces of syntax, all defined by mutual induction:

- Contexts: Generating data of an infinity category.
- Terms: Operations in an infinity category.
- Types: Source and Target for a term.

---

<sup>3</sup>Finster and Mimram, *A Type-Theoretical Definition of Weak  $\omega$ -Categories*.

CaTT is a type theory for *weak infinity categories*<sup>3</sup>.

There are 4 pieces of syntax, all defined by mutual induction:

- Contexts: Generating data of an infinity category.
- Terms: Operations in an infinity category.
- Types: Source and Target for a term.
- Substitutions: A mapping from variables of one context to terms of another.

---

<sup>3</sup>Finster and Mimram, *A Type-Theoretical Definition of Weak  $\omega$ -Categories*.

Types in CaTT have 2 constructors.

Types in CaTT have 2 constructors.

- The  $\star$  constructor takes no arguments.  
A term of type  $\star$  represents a 0-cell.

Types in CaTT have 2 constructors.

- The  $\star$  constructor takes no arguments.  
A term of type  $\star$  represents a 0-cell.
- The *arrow* constructor takes 2 terms and a type as arguments.  
A term of type  $s \rightarrow_A t$  has source  $s$ , target  $t$  and lower dimensional sources and targets given by  $A$ .

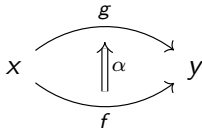
Types in CaTT have 2 constructors.

- The  $\star$  constructor takes no arguments.  
A term of type  $\star$  represents a 0-cell.
- The *arrow* constructor takes 2 terms and a type as arguments.  
A term of type  $s \rightarrow_A t$  has source  $s$ , target  $t$  and lower dimensional sources and targets given by  $A$ .



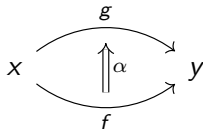
Types in CaTT have 2 constructors.

- The  $\star$  constructor takes no arguments.  
A term of type  $\star$  represents a 0-cell.
- The *arrow* constructor takes 2 terms and a type as arguments.  
A term of type  $s \rightarrow_A t$  has source  $s$ , target  $t$  and lower dimensional sources and targets given by  $A$ .



Types in CaTT have 2 constructors.

- The  $\star$  constructor takes no arguments.  
A term of type  $\star$  represents a 0-cell.
- The *arrow* constructor takes 2 terms and a type as arguments.  
A term of type  $s \rightarrow_A t$  has source  $s$ , target  $t$  and lower dimensional sources and targets given by  $A$ .



$$\alpha : f \rightarrow_{x \rightarrow_{\star} y} g$$

Contexts consist of a list of pairs of variable names and types.

Contexts consist of a list of pairs of variable names and types.

## Disc contexts

For each natural number we can define the *disc context*  $D_n$ .

$D_0$



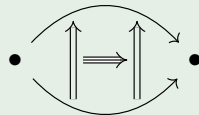
$D_1$



$D_2$



$D_3$



$$D_2 := x : \star, y : \star, f : x \rightarrow_{\star} y, g : x \rightarrow_{\star} y, \alpha : f \rightarrow_{x \rightarrow_{\star} y} g$$

Composition can be done with the `coh` constructor.

## `coh` constructor

Given:

- A context  $\Gamma$  - the shape of the composition,
- A type  $A$  in  $\Gamma$  - the boundary of the composition,
- A substitution  $\sigma : \Gamma \rightarrow \Delta$  - the terms to be composed,

we get a term in  $\Delta$ :

$$\text{coh } (\Gamma : A)[\sigma]$$

The contexts for which the `coh` constructor is well typed are called *pasting contexts*

# Example composition

Suppose we have:

$$\bullet \xrightarrow{f} \bullet \xrightarrow{g} \bullet \xrightarrow{h} \bullet$$

# Example composition

Suppose we have:

$$\bullet \xrightarrow{f} \bullet \xrightarrow{g} \bullet \xrightarrow{h} \bullet$$

Let  $\Gamma = \bullet \xrightarrow{a} \bullet \xrightarrow{b} \bullet$ .  $\Gamma$  is a pasting context. Then:

$$f \cdot g := \text{coh } (\Gamma : x \rightarrow z)[a \mapsto f, \\ b \mapsto g]$$

# Example composition

Suppose we have:

$$\bullet \xrightarrow{f} \bullet \xrightarrow{g} \bullet \xrightarrow{h} \bullet$$

Let  $\Gamma = \bullet \xrightarrow{a} \bullet \xrightarrow{b} \bullet$ .  $\Gamma$  is a pasting context. Then:

$$f \cdot g := \text{coh } (\Gamma : x \rightarrow z) [a \mapsto f, \\ b \mapsto g]$$

$$(f \cdot g) \cdot h := \text{coh } (\Gamma : x \rightarrow z) [a \mapsto f \cdot g, \\ b \mapsto h]$$



- CaTT as we have presented it has no non-trivial equality and no computation.
- The idea is to implement a reduction relation that unifies the operations we want to strictify.
- By doing this we obtain a type theory for which the models are semistrict categories.

$\text{CaTT}_{\text{sa}}$  has a definitional equality based on an operation we call insertion.

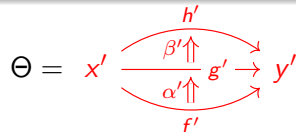
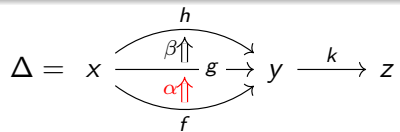
1-associator

$$x \xrightarrow{f} y \xrightarrow{g} z \qquad x' \xrightarrow{f'} y' \xrightarrow{g'} z'$$


is sent to:

$$x \xrightarrow{f} x' \xrightarrow{f'} y' \xrightarrow{g'} z'$$

# Components of insertion



# Components of insertion

$$\Delta = x \begin{array}{c} \xrightarrow{h} \\ \beta \uparrow \uparrow \\ \xrightarrow{g} \\ \alpha \uparrow \uparrow \\ \xrightarrow{f} \end{array} y \xrightarrow{k} z$$

$$\Theta = x' \begin{array}{c} \xrightarrow{h'} \\ \beta' \uparrow \uparrow \\ \xrightarrow{g'} \\ \alpha' \uparrow \uparrow \\ \xrightarrow{f'} \end{array} y' \xrightarrow{k} z$$

$$\Delta \ll_{\alpha} \Theta = x' \begin{array}{c} \xrightarrow{h} \\ \beta \uparrow \uparrow \\ \xrightarrow{h'} \\ \beta' \uparrow \uparrow \\ \xrightarrow{g'} \\ \alpha' \uparrow \uparrow \\ \xrightarrow{f'} \end{array} y' \xrightarrow{k} z$$

# Components of insertion

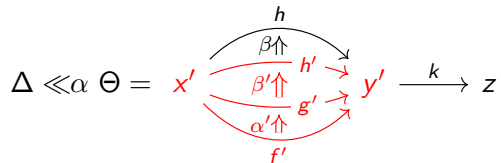
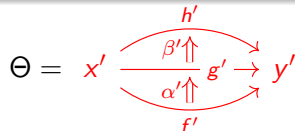
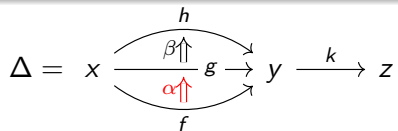
$$\Delta = x \begin{array}{c} \xrightarrow{h} \\ \beta \uparrow \uparrow \\ \xrightarrow{g} \\ \alpha \uparrow \uparrow \\ \xrightarrow{f} \end{array} y \xrightarrow{k} z$$

$$\Theta = x' \begin{array}{c} \xrightarrow{h'} \\ \beta' \uparrow \uparrow \\ \xrightarrow{g'} \\ \alpha' \uparrow \uparrow \\ \xrightarrow{f'} \end{array} y' \xrightarrow{k} z$$

$$\Delta \ll_{\alpha} \Theta = x' \begin{array}{c} \xrightarrow{h} \\ \beta \uparrow \uparrow \\ \xrightarrow{h'} \\ \beta' \uparrow \uparrow \\ \xrightarrow{g'} \\ \alpha' \uparrow \uparrow \\ \xrightarrow{f'} \end{array} y' \xrightarrow{k} z$$

$$\iota : \Theta \rightarrow \Delta \ll_{\alpha} \Theta$$

# Components of insertion



$$\iota : \Theta \rightarrow \Delta \ll_{\alpha} \Theta$$

$$\kappa : \Delta \rightarrow \Delta \ll_{\alpha} \Theta$$

# Components of insertion

$$\Delta = x \begin{array}{c} \xrightarrow{h} \\ \beta \uparrow \uparrow \\ \xrightarrow{g} \\ \alpha \uparrow \uparrow \\ \xrightarrow{f} \end{array} y \xrightarrow{k} z$$

$$\Theta = x' \begin{array}{c} \xrightarrow{h'} \\ \beta' \uparrow \uparrow \\ \xrightarrow{g'} \\ \alpha' \uparrow \uparrow \\ \xrightarrow{f'} \end{array} y'$$

$$\Delta \ll_{\alpha} \Theta = x' \begin{array}{c} \xrightarrow{h} \\ \beta \uparrow \uparrow \\ \xrightarrow{h'} \\ \beta' \uparrow \uparrow \\ \xrightarrow{g'} \\ \alpha' \uparrow \uparrow \\ \xrightarrow{f'} \end{array} y' \xrightarrow{k} z$$

$$\iota : \Theta \rightarrow \Delta \ll_{\alpha} \Theta$$

$$\kappa : \Delta \rightarrow \Delta \ll_{\alpha} \Theta$$

Given  $\sigma : \Delta \rightarrow \Gamma$  and  $\tau : \Theta \rightarrow \Gamma$  we get:

$$\sigma \ll_{\alpha} \tau : \Delta \ll_{\alpha} \Theta \rightarrow \Gamma$$

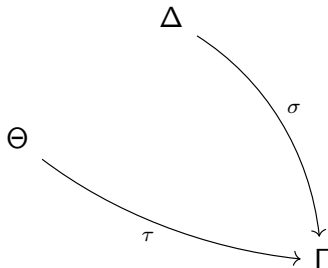
# Universal Property of Insertion

Insertion also satisfies a *universal property*. Suppose we have  $\text{coh } (\Delta : A)[\sigma]$  where  $\sigma(\alpha) = \text{coh } (\Theta : B)[\tau]$ .



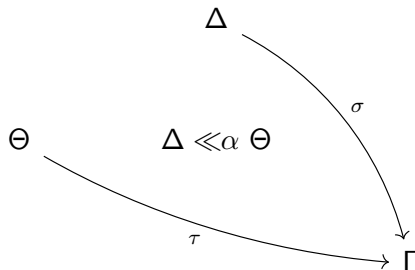
# Universal Property of Insertion

Insertion also satisfies a *universal property*. Suppose we have  $\text{coh } (\Delta : A)[\sigma]$  where  $\sigma(\alpha) = \text{coh } (\Theta : B)[\tau]$ .



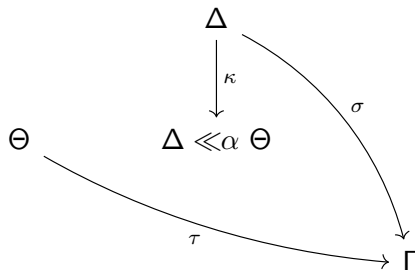
# Universal Property of Insertion

Insertion also satisfies a *universal property*. Suppose we have  $\text{coh } (\Delta : A)[\sigma]$  where  $\sigma(\alpha) = \text{coh } (\Theta : B)[\tau]$ .



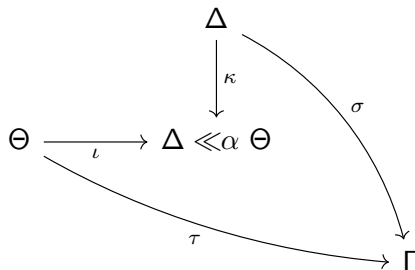
# Universal Property of Insertion

Insertion also satisfies a *universal property*. Suppose we have  $\text{coh } (\Delta : A)[\sigma]$  where  $\sigma(\alpha) = \text{coh } (\Theta : B)[\tau]$ .



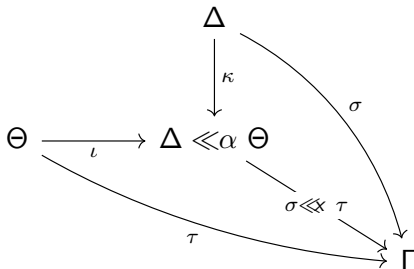
# Universal Property of Insertion

Insertion also satisfies a *universal property*. Suppose we have  $\text{coh } (\Delta : A)[\sigma]$  where  $\sigma(\alpha) = \text{coh } (\Theta : B)[\tau]$ .



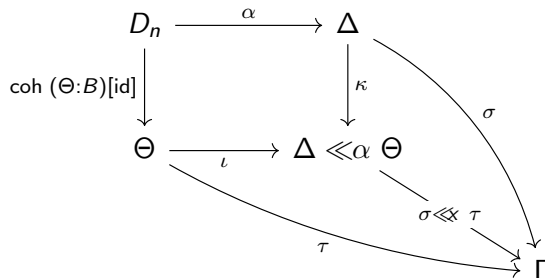
# Universal Property of Insertion

Insertion also satisfies a *universal property*. Suppose we have  $\text{coh}(\Delta : A)[\sigma]$  where  $\sigma(\alpha) = \text{coh}(\Theta : B)[\tau]$ .



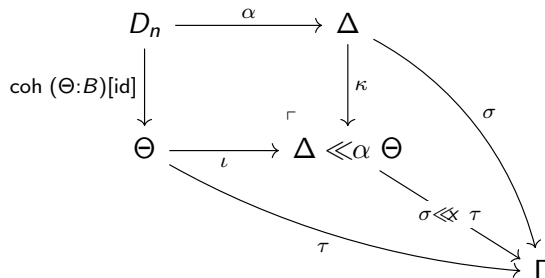
# Universal Property of Insertion

Insertion also satisfies a *universal property*. Suppose we have  $\text{coh}(\Delta : A)[\sigma]$  where  $\sigma(\alpha) = \text{coh}(\Theta : B)[\tau]$ .



# Universal Property of Insertion

Insertion also satisfies a *universal property*. Suppose we have  $\text{coh}(\Delta : A)[\sigma]$  where  $\sigma(\alpha) = \text{coh}(\Theta : B)[\tau]$ .



Insertion generates a reduction relation for  $\text{Catt}_{\text{sa}}$ :

$$\text{coh } (\Delta : A)[\sigma] \rightsquigarrow \text{coh } (\Delta \ll_{\alpha} \Theta : A[\![\kappa]\!])[\sigma \ll_{\alpha} \tau]$$

where  $\sigma(\alpha) = \text{coh } (\Delta : B)[\tau]$ .






Insertion generates a reduction relation for  $\text{Catt}_{\text{sa}}$ :

$$\text{coh } (\Delta : A)[\sigma] \rightsquigarrow \text{coh } (\Delta \ll_{\alpha} \Theta : A[\![\kappa]\!])[\sigma \ll_{\alpha} \tau]$$

where  $\sigma(\alpha) = \text{coh } (\Delta : B)[\tau]$ .

This reduction has been proven to have the following properties:

- Subject reduction
- Termination
- Confluence

-  Finster, Eric and Samuel Mimram. *A Type-Theoretical Definition of Weak  $\omega$ -Categories*. 2017. DOI: [10.1109/lics.2017.8005124](https://doi.org/10.1109/lics.2017.8005124). eprint: [1706.02866](https://arxiv.org/abs/1706.02866).
-  Finster, Eric, Alex R., and Jamie Vicary. *A Type Theory for Strictly Associative Infinity Categories*. 2021. arXiv: [2109.01513](https://arxiv.org/abs/2109.01513).
-  Finster, Eric, David Reutter, et al. *A Type Theory for Strictly Unital  $\infty$ -Categories*. Proceedings of the Thirty-Seventh Annual ACM/IEEE Symposium on Logic in Computer Science (LICS 2022). 2020. DOI: [10.1145/3531130.3533363](https://doi.org/10.1145/3531130.3533363). arXiv: [2007.08307](https://arxiv.org/abs/2007.08307).