A Type Theoretic Approach to Semistrict Higher Categories

Alex Rice

12$^{\text{th}}$ May 2022

Outline

1. Globular Infinity Categories
2. Weak Infinity Categories
3. Semistrict infinity categories

Globular Sets

Globular sets are one natural shape of higher categories.

Globular Sets

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

- A set of objects G.

Globular Sets

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

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

Globular Sets

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

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

$$\begin{tikzcd} x \arrow[r, bend left=50, "g"{name=G}] \arrow[r, bend right=50, "f"'{name=F}] & y \arrow[from=F, to=G, Rightarrow, "\alpha"'] \end{tikzcd}$$

Globular Sets

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

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

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

- ...

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

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

$$\begin{tikzcd} x \arrow[r, bend left=50, "g"{name=U}] \arrow[r, bend right=50, "f"'{name=D}] & y \arrow[from=D, to=U, Rightarrow, "\alpha"'] \end{tikzcd}$$

- ...

Definition

A *Globular Set* is a set G with a globular set $G_{x,y}$ for each pair of objects $x, y \in G$.

Compostition in Globular Sets

Composition of 1 cells

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

Compostition in Globular Sets

Composition of 1 cells

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

Composition of 2 cells

Composition along a 1-boundary:
$$\begin{tikzcd}
\bullet \arrow[r, bend left=60, ""{name=U}] \arrow[r, ""{name=M}] \arrow[r, bend right=60, ""{name=D}] & \bullet
\arrow[from=M, to=U, phantom, "\beta\Uparrow"] \arrow[from=D, to=M, phantom, "\alpha\Uparrow"]
\end{tikzcd}$$

Compostition in Globular Sets

Composition of 1 cells

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

Composition of 2 cells

Composition along a 1-boundary:
$$\begin{tikzcd} \bullet \arrow[r, bend left=80, ""{name=T}] \arrow[r, ""{name=M}] \arrow[r, bend right=80, ""{name=B}] & \bullet \arrow[from=M, to=T, phantom, "\beta\Uparrow"] \arrow[from=B, to=M, phantom, "\alpha\Uparrow"] \end{tikzcd}$$

Composition along a 0-boundary:
$$\begin{tikzcd} \bullet \arrow[r, bend left=50, ""{name=A1}] \arrow[r, bend right=50, ""{name=A2}] \arrow[from=A2, to=A1, phantom, "\alpha\Uparrow"] & \bullet \arrow[r, bend left=50, ""{name=B1}] \arrow[r, bend right=50, ""{name=B2}] \arrow[from=B2, to=B1, phantom, "\beta\Uparrow"] & \bullet \end{tikzcd}$$

In a *strict infinity category* we have binary composition of n-cells for along a k boundary for all $k < n$.

Composition

If f and g are n-cells with the k-target of f equalling the k-source of g then there is an n-cell $f \circ_k g$.

In a *strict infinity category* we have binary composition of n-cells for along a k boundary for all $k < n$.

Composition

If f and g are n-cells with the k-target of f equalling the k-source of g then there is an n-cell $f \circ_k g$.

Identities

For each n-cell f there is an $(n+1)$-cell $\mathrm{id}_f : f \to f$.

Strict Infinity Categories - Associativity

If $0 \leq k < n$ and f, g, and h are n-cells then:

$$f \circ_k (g \circ_k h) = (f \circ_k g) \circ_k h$$

If $0 \leq k < n$ and f, g, and h are n-cells then:

$$f \circ_k (g \circ_k h) = (f \circ_k g) \circ_k h$$

Associativity of 1-cells

Given $\bullet \xrightarrow{f} \bullet \xrightarrow{g} \bullet \xrightarrow{h} \bullet$ we have:

$$f \circ_0 (g \circ_0 h) = (f \circ_0 g) \circ_0 h$$

Strict Infinity Categories - Identities

If $0 \leq k < n$ and f is an n-cell with k-source x and k-target y then:

$$\mathrm{id}^{n-k}(x) \circ_k f = f = f \circ_k \mathrm{id}^{n-k}(y)$$

Strict Infinity Categories - Identities

If $0 \leq k < n$ and f is an n-cell with k-source x and k-target y then:

$$\mathrm{id}^{n-k}(x) \circ_k f = f = f \circ_k \mathrm{id}^{n-k}(y)$$

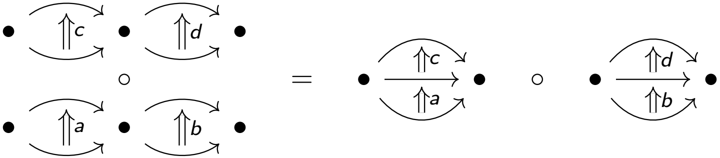
Identity on 2-cell

Given $f, g : x \to y$ and $\alpha : f \to g$ we have:

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

If $0 \leq q < p < n$ and a, b, c, d are n-cells then:

$$(a \circ_p b) \circ_q (c \circ_p d) = (a \circ_q c) \circ_p (b \circ_q d)$$



Strict Infinity Categories - Interchange

If $0 \leq q < p < n$ and a, b, c, d are n-cells then:

$$(a \circ_p b) \circ_q (c \circ_p d) = (a \circ_q c) \circ_p (b \circ_q d)$$

$$\begin{tikzcd}
\bullet \arrow[r, bend left] \arrow[r, bend right] \arrow[r, phantom, "\Uparrow c"] & \bullet \arrow[r, bend left] \arrow[r, bend right] \arrow[r, phantom, "\Uparrow d"] & \bullet \\
 & \circ & \\
\bullet \arrow[r, bend left] \arrow[r, bend right] \arrow[r, phantom, "\Uparrow a"] & \bullet \arrow[r, bend left] \arrow[r, bend right] \arrow[r, phantom, "\Uparrow b"] & \bullet
\end{tikzcd}
=
\begin{tikzcd}
\bullet \arrow[r, bend left=60] \arrow[r] \arrow[r, bend right=60] \arrow[r, phantom, shift left=2ex, "\Uparrow c"] \arrow[r, phantom, shift right=2ex, "\Uparrow a"] & \bullet & \circ & \bullet \arrow[r, bend left=60] \arrow[r] \arrow[r, bend right=60] \arrow[r, phantom, shift left=2ex, "\Uparrow d"] \arrow[r, phantom, shift right=2ex, "\Uparrow b"] & \bullet
\end{tikzcd}$$

Further if $f \circ_k g$ is well defined then:

$$\mathrm{id}_f \circ_k \mathrm{id}(g) = \mathrm{id}(f \circ_k g)$$

Monoidal Categories

Monoidal categories are instances of infinity categories.

Definition (Monoidal category)

A *monoidal category* is a category $\mathcal{C}$ equipped with a functor $\otimes : \mathcal{C} \times \mathcal{C} \to \mathcal{C}$ and a unit object I satisfying some conditions.

Monoidal Categories

Monoidal categories are instances of infinity categories.

Definition (Monoidal category)

A *monoidal category* is a category $\mathcal{C}$ equipped with a functor $\otimes : \mathcal{C} \times \mathcal{C} \to \mathcal{C}$ and a unit object I satisfying some conditions.

A strict infinity category with one object and no non-identity n-cells for n higher than 2 is a strict monoidal category.

Weakness

If a category has all products and a terminal object, then it can be given the structure of a monoidal category.

Weakness

If a category has all products and a terminal object, then it can be given the structure of a monoidal category.

Set

The category **Set** is a monoidal category with $\otimes$ given by cartesian product and unit object given by the singleton set.

Weakness

If a category has all products and a terminal object, then it can be given the structure of a monoidal category.

Set

The category **Set** is a monoidal category with $\otimes$ given by cartesian product and unit object given by the singleton set.

The monoidal product in **Set** is *not* strict.

Weak Infinity Categories

In a weak infinity category, we only require that the various laws hold up to isomorphism.
However many isomorphisms can exist between two cells. We require that these isomorphisms be *coherent*.

In a weak infinity category, we only require that the various laws hold up to isomorphism.
However many isomorphisms can exist between two cells. We require that these isomorphisms be *coherent*.

- For a 1-cell $f : x \to y$, there are unitors $\lambda_f : \mathrm{id}_x \circ f \to f$ and $\rho_f : f \circ \mathrm{id}_y$.
- λ_{id_x} and ρ_{id_x} are both arrows $\mathrm{id}_x \circ \mathrm{id}_x \to \mathrm{id}_x$. We can ask that they be isomorphic.
- This isomorphism will also be subject to coherence conditions.

Weak Infinity Categories

In a weak infinity category, we only require that the various laws hold up to isomorphism.
However many isomorphisms can exist between two cells. We require that these isomorphisms be *coherent*.

- For a 1-cell $f : x \to y$, there are unitors $\lambda_f : \mathrm{id}_x \circ f \to f$ and $\rho_f : f \circ \mathrm{id}_y$.
- λ_{id_x} and ρ_{id_x} are both arrows $\mathrm{id}_x \circ \mathrm{id}_x \to \mathrm{id}_x$. We can ask that they be isomorphic.
- This isomorphism will also be subject to coherence conditions.

It quickly becomes apparent that we need a more uniform way to package this coherence data.

Pasting Diagrams

A *pasting diagram* represents a composition that can be done in an infinity category.

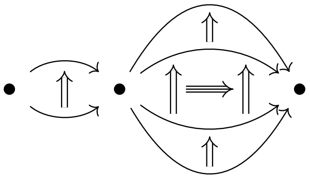
A *pasting diagram* represents a composition that can be done in an infinity category.

The compositions we have already seen form pasting diagrams.

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

$$\begin{tikzcd}
\bullet \arrow[r, bend left] \arrow[r, bend right] \arrow[r, phantom, "\alpha\Uparrow"] & \bullet \arrow[r, bend left] \arrow[r, bend right] \arrow[r, phantom, "\beta\Uparrow"] & \bullet
\end{tikzcd}$$

We can also form more complicated compositions as pasting diagrams.



Motto for Pasting Diagrams

- Pasting diagrams for 1-categories are simply chains of 1-cells:

$$\bullet \qquad\qquad \bullet \xrightarrow{f} \bullet \xrightarrow{g} \bullet \qquad\qquad \bullet \to \bullet \to \cdots \to \bullet$$

Motto for Pasting Diagrams

- Pasting diagrams for 1-categories are simply chains of 1-cells:

$$\bullet \qquad\qquad \bullet \xrightarrow{f} \bullet \xrightarrow{g} \bullet \qquad\qquad \bullet \to \bullet \to \cdots \to \bullet$$

- In a 1-category, any pasting diagram has a composite.

Motto for Pasting Diagrams

- Pasting diagrams for 1-categories are simply chains of 1-cells:

$$\bullet \qquad\qquad \bullet \xrightarrow{f} \bullet \xrightarrow{g} \bullet \qquad\qquad \bullet \to \bullet \to \cdots \to \bullet$$

- In a 1-category, any pasting diagram has a composite.
- Further, there is exactly 1 composite.

Motto for Pasting Diagrams

- Pasting diagrams for 1-categories are simply chains of 1-cells:

$$\bullet \qquad\qquad \bullet \xrightarrow{f} \bullet \xrightarrow{g} \bullet \qquad\qquad \bullet \to \bullet \to \cdots \to \bullet$$

- In a 1-category, any pasting diagram has a composite.
- Further, there is exactly 1 composite.
- In a strict infinity category, every (higher dimensional) pasting diagram has exactly one composite.

Motto for Pasting Diagrams

- Pasting diagrams for 1-categories are simply chains of 1-cells:

$$\bullet \qquad\qquad \bullet \xrightarrow{f} \bullet \xrightarrow{g} \bullet \qquad\qquad \bullet \to \bullet \to \cdots \to \bullet$$

- In a 1-category, any pasting diagram has a composite.
- Further, there is exactly 1 composite.
- In a strict infinity category, every (higher dimensional) pasting diagram has exactly one composite.
- For weak infinity categories, we weaken the exactness condition to uniqueness up to isomorphism.

Weak Composition

Motto for weak infinity categories

- Every pasting diagram has a composite
- Given 2 parallel arrows s, t generated from the whole pasting diagram, there is a higher dimensional arrow from s to t.

Weak Composition

Motto for weak infinity categories

- Every pasting diagram has a composite
- Given 2 parallel arrows s, t generated from the whole pasting diagram, there is a higher dimensional arrow from s to t.

Taking the composite of the diagram:

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

gives the composite $f \circ g$.

Over the singleton pasting diagram

$$x$$

and taking $s = x$ and $t = x$ we get a term from x to x representing the identity on x.

CaTT is a type theory for *weak infinity categories*. It allows us to build and describe the operations of an infinity category.

CaTT

CaTT is a type theory for *weak infinity categories*. It allows us to build and describe the operations of an infinity category.

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

- Contexts: Represent the *generating* data of an infinity category.

CaTT is a type theory for *weak infinity categories*. It allows us to build and describe the operations of an infinity category.

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

- Contexts: Represent the *generating* data of an infinity category.
- Terms: A term in a context Γ represents a *word* that can be built from the generators in Γ.

CaTT is a type theory for *weak infinity categories*. It allows us to build and describe the operations of an infinity category.

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

- Contexts: Represent the *generating* data of an infinity category.
- Terms: A term in a context Γ represents a *word* that can be built from the generators in Γ.
- Types: A type contains all the information of the *sources* and *targets* for a term.

CaTT is a type theory for *weak infinity categories*. It allows us to build and describe the operations of an infinity category.

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

- Contexts: Represent the *generating* data of an infinity category.
- Terms: A term in a context Γ represents a *word* that can be built from the generators in Γ.
- Types: A type contains all the information of the *sources* and *targets* for a term.
- Substitutions: A substitution is a *morphism* between contexts.

Types

Types in CaTT capture the structure of a globular set. The type of a term contains the data of all of its sources and targets.

Types

Types in CaTT capture the structure of a globular set. The type of a term contains the data of all of its sources and targets.

A term in a globular set is either a 0-cell, or an $(n+1)$-cell between two parallel n-cells.

Types have 2 constructors, the star constructor and the arrow constructor.

Types

Types in CaTT capture the structure of a globular set. The type of a term contains the data of all of its sources and targets.

A term in a globular set is either a 0-cell, or an $(n+1)$-cell between two parallel n-cells.

Types have 2 constructors, the star constructor and the arrow constructor.

- If a term is a 0-cell in our infinity category, then it has type $\star$.

Types

Types in CaTT capture the structure of a globular set. The type of a term contains the data of all of its sources and targets.

A term in a globular set is either a 0-cell, or an $(n+1)$-cell between two parallel n-cells.

Types have 2 constructors, the star constructor and the arrow constructor.

- If a term is a 0-cell in our infinity category, then it has type $\star$.
- Otherwise a term is an $(n+1)$-cell between parallel n-cells f and g, in which case it has type:

$$f \to_A g$$

  where A is the (common) type of f and g.

The Coh Constructor

The crucial part of CaTT is the Coh constructor, which captures the motto for weak composition.

Motto for weak infinity categories

- Every pasting diagram has a composite
- Given 2 parallel arrows s, t generated from the whole pasting diagram, there is a higher dimensional arrow from s to t.

The Coh Constructor

The crucial part of CaTT is the Coh constructor, which captures the motto for weak composition.

Motto for weak infinity categories

- Every pasting diagram has a composite
- Given 2 parallel arrows s, t generated from the whole pasting diagram, there is a higher dimensional arrow from s to t.

$$\mathsf{coh}\ (\Gamma : s \to t)[\sigma]$$

The Coh Constructor

The crucial part of CaTT is the Coh constructor, which captures the motto for weak composition.

Motto for weak infinity categories

- Every pasting diagram has a composite
- Given 2 parallel arrows s, t generated from the whole pasting diagram, there is a higher dimensional arrow from s to t.

$$\mathsf{coh}\ (\Gamma : s \to t)[\sigma]$$

- Γ is a pasting diagram, and is represented by a context.

The Coh Constructor

The crucial part of CaTT is the Coh constructor, which captures the motto for weak composition.

Motto for weak infinity categories

- Every pasting diagram has a composite
- Given 2 parallel arrows s, t generated from the whole pasting diagram, there is a higher dimensional arrow from s to t.

$$\mathsf{coh}\ (\Gamma : s \to t)[\sigma]$$

- Γ is a pasting diagram, and is represented by a context.
- s and t are two parallel terms, which can be represented as a type.

The Coh Constructor

The crucial part of CaTT is the Coh constructor, which captures the motto for weak composition.

Motto for weak infinity categories

- Every pasting diagram has a composite
- Given 2 parallel arrows s, t generated from the whole pasting diagram, there is a higher dimensional arrow from s to t.

$$\mathsf{coh}\ (\Gamma : s \to t)[\sigma]$$

- Γ is a pasting diagram, and is represented by a context.
- s and t are two parallel terms, which can be represented as a type.
- σ labels the pasting diagram with (compatible) terms, and can be represented as a substitution.

Examples

Identity

Let t be a 1 dimensional term. The identity on t is:

$$\mathsf{coh}\ (x \xrightarrow{f} y : f \to_{x \to_{\star} y} f)[\sigma]$$

where σ maps f to t.

Examples

Identity

Let t be a 1 dimensional term. The identity on t is:

$$\mathsf{coh}\ (x \xrightarrow{f} y : f \to_{x \to_\star y} f)[\sigma]$$

where σ maps f to t.

1-composition

Let $s : x \to_\star y$ and $t : y \to_\star z$ be terms. Their composite is given by:

$$\mathsf{coh}\ (x \xrightarrow{f} y \xrightarrow{g} z : x \to_\star z)[\sigma]$$

where $\sigma(x) = x$, $\sigma(y) = y$, $\sigma(z) = z$, $\sigma(f) = s$, $\sigma(g), = t$.

Examples

Take the context $\Gamma = w \xrightarrow{f} x \xrightarrow{g} y \xrightarrow{h} z$.

The *associator* is given by:

$$\mathrm{coh}\ (\Gamma : (f \circ g) \circ h \to_{w \to_\star z} f \circ (g \circ h))[\mathrm{id}]$$

Strictification

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

Strictification

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.

Strictification

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.

Semistrictness

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

Semistrictness

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

Therefore, we look for *semistrict* definitions of infinity categories.

Semistrictness

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

Semistrictness

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 ∞-**Cat**
Associators	✓
Unitors	✓
Interchangers	✓

Semistrictness

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 ∞-**Cat**	Simpson
Associators	✓	✓
Unitors	✓	
Interchangers	✓	✓

Semistrictness

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 ∞-**Cat**	Simpson
Associators	✓	✓
Unitors	✓	
Interchangers	✓	✓

Semistrictness

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 ∞-**Cat**	Simpson	Grey
Associators	✓	✓	✓
Unitors	✓		✓
Interchangers	✓	✓	

Type theories for semistrict languages

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. Further by showing our reduction is terminating and confluent, we obtain a language for the operations which has decidable type checking and equality.

- $\mathsf{CaTT}_{\mathsf{su}}$: Has strict units. Generated by the pruning operation.
- $\mathsf{CaTT}_{\mathsf{sa}}$: Has strict associators. Generated by the insertion operation.
- $\mathsf{CaTT}_{\mathsf{sua}}$ (Work in Progress): Combines the previous two theories.

Example - Syllepsis

- Given two scalars $a, b : id_x \to id_x$, by the Eckmann Hilton argument we have an isomorphism $\mathrm{EH}_{f,g} : a \circ_1 b \simeq b \circ_1 a$.
- In fact, there are two such isomorphisms, $\mathrm{EH}_{a,b}$ and $\mathrm{EH}^{-1}_{b,a}$, that need not be themselves isomorphic.
- If the whole problem is suspended one dimension higher, then there is a morphism called the syllepsis between these.

Example - Syllepsis

- Given two scalars $a, b : id_x \to id_x$, by the Eckmann Hilton argument we have an isomorphism $\mathrm{EH}_{f,g} : a \circ_1 b \simeq b \circ_1 a$.
- In fact, there are two such isomorphisms, $\mathrm{EH}_{a,b}$ and $\mathrm{EH}_{b,a}^{-1}$, that need not be themselves isomorphic.
- If the whole problem is suspended one dimension higher, then there is a morphism called the syllepsis between these.

	CaTT	$\mathrm{CaTT}_{\mathrm{su}}$	$\mathrm{CaTT}_{\mathrm{sua}}$
Eckmann-Hilton	297	15	15
Syllepsis	NA	675	397

Figure: Coh constructors in Eckmann-Hilton and Syllepsis

Further work

- Finish proving metatheorems for $\mathsf{CaTT}_{\mathsf{sua}}$.
- Equivalence of Theories.
- More semistrict type theories, including one for Simpson-like semistrictness.
- Bridging the gap between CaTT and graphical methods.