A Syntax for Strictly Associative and Unital ∞-Categories

Eric Finster Alex Rice Jamie Vicary

LICS 2024, Tallinn

UNIVERSITY OF CAMBRIDGE

CATT is a type theory for ∞-categories.

The Type Theory CATT

CATT is a type theory for ∞-categories.

Globular ∞-categories contain higher dimensional arrows.

2-cell $\alpha : f \to g$

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

The Type Theory Catt

Catt is a type theory for ∞-categories.

Globular ∞-categories contain higher dimensional arrows.

Terms of Catt correspond to operations.

2-cell $\alpha : f \to g$

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

The Type Theory CATT

CATT is a type theory for ∞-categories.

Globular ∞-categories contain higher dimensional arrows.

Terms of CATT correspond to operations.

A type gives the boundary of a term.

2-cell $\alpha : f \to g$

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

Terms in CATT

Terms represent the possible operations in a globular ∞-category.

Terms built over *pasting diagrams*.

Compound terms using substitutions.

Terms in CATT

Terms represent the possible operations in a globular ∞-category.

Terms built over *pasting diagrams*.

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

Compound terms using substitutions.

Terms represent the possible operations in a globular ∞-category.

Terms built over *pasting diagrams*.

Compound terms using substitutions.

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

$$\Gamma := \ x \xrightarrow{f} y \xrightarrow{g} z$$

$$f * g := \mathsf{coh}\ (\Gamma : x \to z)$$

Terms in CATT

Terms represent the possible operations in a globular ∞-category.

Terms built over *pasting diagrams*.

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

$$\Gamma := \quad x \xrightarrow{f} y \xrightarrow{g} z$$

$$f * g := \mathsf{coh}\ (\Gamma : x \to z)$$

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

$$\alpha \otimes \beta := \mathsf{coh}\ (\Delta : f * g \to h * i)$$

Compound terms using substitutions.

Terms represent the possible operations in a globular ∞-category.

Terms built over *pasting diagrams*.

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

$$\Gamma := \quad x \xrightarrow{f} y \xrightarrow{g} z$$

$$f * g := \mathsf{coh}\ (\Gamma : x \to z)$$

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

$$\alpha \otimes \beta := \mathsf{coh}\ (\Delta : f * g \to h * i)$$

Compound terms using substitutions.

$$s[\![\sigma]\!]$$

Terms in Catt

Terms represent the possible operations in a globular ∞-category.

Terms built over *pasting diagrams*.

$$\mathrm{coh}\ (\Gamma : s \to t)$$

$$\Gamma := \begin{tikzcd} x \arrow[r, "f"] & y \arrow[r, "g"] & z \end{tikzcd}$$

$$f * g := \mathrm{coh}\ (\Gamma : x \to z)$$

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

$$\alpha \otimes \beta := \mathrm{coh}\ (\Delta : f * g \to h * i)$$

Compound terms using substitutions.

$$s[\![\sigma]\!]$$

$$\sigma := \langle f \mapsto a, g \mapsto (b * c) \rangle$$

$$a * (b * c) := (f * g)[\![\sigma]\!]$$

The Associator

$$\Delta := \quad x \xrightarrow{a} y \xrightarrow{b} z \xrightarrow{c} w$$

The Associator

$$\Delta := \quad x \xrightarrow{a} y \xrightarrow{b} z \xrightarrow{c} w$$

$$\alpha_{a,b,c} := \mathsf{coh}\left(\Delta : (a * b) * c \to a * (b * c)\right)$$

Semistrictness

The theory described by CATT is *Weak*.

Laws of categories are given by equivalence.

$$\text{Weak} \longleftrightarrow \text{Strict}$$

Semistrictness

The theory described by $\textsc{Catt}$ is *Weak*.

Laws of categories are given by equivalence.

Weak $\longleftrightarrow$ Strict

Harder to use | Easier to use

Semistrictness

The theory described by CATT is *Weak*.

Laws of categories are given by equivalence.

Weak $\longleftrightarrow$ Strict

Harder to use	Easier to use
More expressive	Less expressive

Semistrictness

The theory described by CATT is *Weak*.

Laws of categories are given by equivalence.

$$\text{Weak} \longleftarrow\!\text{————————} \quad \text{Semistrict} \longrightarrow \text{Strict}$$

Harder to use	Easier to use
More expressive	Less expressive

The theory described by CATT is *Weak*.

Laws of categories are given by equivalence.

Weak ⟵ Semistrict ⟶ Strict

$\textsc{Catt}_{\mathsf{su}}$

Harder to use

Easier to use

More expressive

Less expressive

The theory described by $\textsc{Catt}$ is *Weak*.

Laws of categories are given by equivalence.

Weak ⟵ Semistrict ⟶ Strict

$\textsc{Catt}_{\mathsf{su}}$

$\textsc{Catt}_{\mathsf{sua}}$

Harder to use | Easier to use

More expressive | Less expressive

Strictness

(Semi)strictness allows more operations to be defined.

Strictness

(Semi)strictness allows more operations to be defined.

$$
\begin{tikzcd}
u \arrow[r, "c"] & v \arrow[r, "d"] & w \\
x \arrow[u, "f"] \arrow[r, "a"'] & y \arrow[u, "g"'] \arrow[ul, Rightarrow, "A"] \arrow[r, "b"'] & z \arrow[u, "h"'] \arrow[ul, Rightarrow, "B"]
\end{tikzcd}
$$

(Semi)strictness allows more operations to be defined.

$$\begin{tikzcd}
u \arrow[r, "c"] & v \arrow[r, "d"] & w \\
x \arrow[u, "f"] \arrow[r, "a"'] & y \arrow[u, "g"'] \arrow[ul, Rightarrow, "A"] \arrow[r, "b"'] & z \arrow[u, "h"'] \arrow[ul, Rightarrow, "B"]
\end{tikzcd}$$

$$(\mathrm{id}_a \otimes B) * (A \otimes \mathrm{id}_d) : (a * b) * h \to f * (c * d)$$

$\textsc{Catt}$ has trivial equality.

$\textsc{Catt}_{\mathsf{su}}$ has disc removal, endo-coherence removal, and pruning.

In $\textsc{Catt}_{\mathsf{sua}}$, pruning is replaced by insertion.

$$\textsc{Catt}_{\mathsf{sua}} := \textsc{Catt} + \text{insertion} + \text{disc removal} + \text{endo-coherence removal}$$

Insertion Rule

$$(a * b) * c \ =_{\mathsf{sua}} \ a * b * c \equiv \mathsf{coh}\,(x \xrightarrow{a} y \xrightarrow{b} z \xrightarrow{c} w : x \to w)$$

Insertion Rule

$$(a * b) * c \;\; =_{\mathsf{sua}} \;\; a * b * c \equiv \mathsf{coh}\ (x \xrightarrow{a} y \xrightarrow{b} z \xrightarrow{c} w : x \to w)$$

Recalling $(a * b) * c \equiv (f * g)[\![\langle f \mapsto a, g \mapsto b * c \rangle]\!]$:

$$x \xrightarrow{\;f\;} y \xrightarrow{\;b*c\;} z \qquad\qquad x' \xrightarrow{\;f'\;} y' \xrightarrow{\;g'\;} z'$$

is sent to:

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

$$\mathsf{coh}\ (\Delta : s \to t)[\![\sigma]\!] \quad x \in \Delta \quad x[\![\sigma]\!] \equiv \mathsf{coh}\ (\Theta : u \to v)[\![\tau]\!]$$

$$\begin{tikzcd}
& \Delta \arrow[dr, bend left, "\sigma"] & \\
\Theta \arrow[rr, bend right, "\tau"'] & & \Gamma
\end{tikzcd}$$

Universal Property of Insertion

$$\mathsf{coh}\ (\Delta : s \to t)[\![\sigma]\!] \quad x \in \Delta \quad x[\![\sigma]\!] \equiv \mathsf{coh}\ (\Theta : u \to v)[\![\tau]\!] \quad (+\ \text{syntactic side condition})$$

$$\begin{tikzcd}
 & \Delta \arrow[d, bend left, "\sigma"] \\
\Theta \arrow[r, bend right, "\tau"'] & \Gamma
\end{tikzcd}$$

Universal Property of Insertion

$$\mathsf{coh}\;(\Delta : s \to t)[\![\sigma]\!] \quad x \in \Delta \quad x[\![\sigma]\!] \equiv \mathsf{coh}\;(\Theta : u \to v)[\![\tau]\!] \quad (+\text{ syntactic side condition})$$

$$\begin{tikzcd}
 & \Delta \arrow[dr, bend left, "\sigma"] & \\
\Theta \arrow[rr, bend right, "\tau"'] & \color{red}{\Delta \ll_x \Theta} & \Gamma
\end{tikzcd}$$

Universal Property of Insertion

$$\mathsf{coh}\ (\Delta : s \to t)[\![\sigma]\!] \quad x \in \Delta \quad x[\![\sigma]\!] \equiv \mathsf{coh}\ (\Theta : u \to v)[\![\tau]\!] \quad (+\ \text{syntactic side condition})$$

$$\begin{tikzcd}
 & \Delta \arrow[ddr, bend left, "\sigma"] & \\
\Theta \arrow[r, "\iota"'] \arrow[drr, bend right, "\tau"'] & \Delta \ll_x \Theta & \\
 & & \Gamma
\end{tikzcd}$$

Universal Property of Insertion

$$\mathsf{coh}\ (\Delta : s \to t)[\![\sigma]\!] \quad x \in \Delta \quad x[\![\sigma]\!] \equiv \mathsf{coh}\ (\Theta : u \to v)[\![\tau]\!] \quad (+\ \text{syntactic side condition})$$

$$\begin{tikzcd}
& \Delta \arrow[d, "\color{red}\kappa"] \arrow[ddr, bend left, "\sigma"] & \\
\Theta \arrow[r, "\iota"'] \arrow[drr, bend right, "\tau"'] & \Delta \ll_x \Theta & \\
& & \Gamma
\end{tikzcd}$$

Universal Property of Insertion

$$\mathsf{coh}\ (\Delta : s \to t)[\![\sigma]\!] \quad x \in \Delta \quad x[\![\sigma]\!] \equiv \mathsf{coh}\ (\Theta : u \to v)[\![\tau]\!] \quad (+\ \text{syntactic side condition})$$

$$\begin{tikzcd}
\color{red}{D} \arrow[r, "\color{red}{x}"] \arrow[d, "\color{red}{\mathsf{coh}\ (\Theta : u \to v)}"'] & \Delta \arrow[d, "\kappa"] \arrow[ddr, bend left, "\sigma"] & \\
\Theta \arrow[r, "\iota"'] \arrow[drr, bend right, "\tau"'] & \Delta \ll_x \Theta & \\
 & & \Gamma
\end{tikzcd}$$

$$\mathsf{coh}\ (\Delta : s \to t)[\![\sigma]\!] \quad x \in \Delta \quad x[\![\sigma]\!] \equiv \mathsf{coh}\ (\Theta : u \to v)[\![\tau]\!] \quad (+\ \text{syntactic side condition})$$

$$\begin{tikzcd}
D \arrow[r, "x"] \arrow[d, "\mathsf{coh}\ (\Theta : u \to v)"'] & \Delta \arrow[d, "\kappa"] \arrow[ddr, bend left, "\sigma"] & \\
\Theta \arrow[r, "\iota"'] \arrow[drr, bend right, "\tau"'] & \Delta \ll_x \Theta \arrow[dr, "\color{red}{\sigma \ll_x \tau}"] & \\
& & \Gamma
\end{tikzcd}$$

Universal Property of Insertion

$$\mathsf{coh}\ (\Delta : s \to t)[\![\sigma]\!] \quad x \in \Delta \quad x[\![\sigma]\!] \equiv \mathsf{coh}\ (\Theta : u \to v)[\![\tau]\!] \quad (+\ \text{syntactic side condition})$$

$$\begin{tikzcd}
D \arrow[r, "x"] \arrow[d, "\mathsf{coh}\ (\Theta : u \to v)"'] & \Delta \arrow[d, "\kappa"] \arrow[ddr, bend left, "\sigma"] & \\
\Theta \arrow[r, "\iota"'] \arrow[drr, bend right, "\tau"'] & \Delta \ll_x \Theta \arrow[ul, phantom, "\ulcorner", very near start] \arrow[dr, "\sigma \ll_x \tau"] & \\
& & \Gamma
\end{tikzcd}$$

$$\mathsf{coh}\ (\Delta : s \to t)[\![\sigma]\!] \quad x \in \Delta \quad x[\![\sigma]\!] \equiv \mathsf{coh}\ (\Theta : u \to v)[\![\tau]\!] \quad (+\ \text{syntactic side condition})$$

$$\begin{tikzcd}
D \arrow[r, "x"] \arrow[d, "\mathsf{coh}\ (\Theta : u \to v)"'] & \Delta \arrow[d, "\kappa"] \arrow[ddr, bend left, "\sigma"] & \\
\Theta \arrow[r, "\iota"'] \arrow[drr, bend right, "\tau"'] & \Delta \ll_x \Theta \arrow[ul, phantom, "\ulcorner", very near start] \arrow[dr, dashed, "\sigma \ll_x \tau"] & \\
 & & \Gamma
\end{tikzcd}$$

$$\mathsf{coh}\ (\Delta : s \to t)[\![\sigma]\!] = \mathsf{coh}\ (\Delta \ll_x \Theta : s[\![\kappa]\!] \to t[\![\kappa]\!])[\![\sigma \ll_x \tau]\!]$$

Normalisation

Equality in $\textsc{Catt}_{\mathsf{sua}}$ is decidable.

$\textsc{Catt}_{\mathsf{sua}}$ has unique normal forms.

Obtained by reduction system.

Termination: Assign *syntactic complexity* to terms.

Confluence: Encode various constructions in Agda.

Normalisation

Equality in $\textsc{Catt}_{\mathsf{sua}}$ is decidable.

$\textsc{Catt}_{\mathsf{sua}}$ has unique normal forms.

Obtained by reduction system.

Termination: Assign *syntactic complexity* to terms.

Confluence: Encode various constructions in Agda.

Type checking is decidable.

We provide a interpreter which:

- Provides tools for construction terms.
- Type checks terms.
- Reduces terms to $\textsc{Catt}_{\mathsf{sua}}$ normal form.

Summary

We introduce the type theory $\textsc{Catt}_{\mathsf{sua}}$.

$\textsc{Catt}_{\mathsf{sua}}$ models strictly unital and associative ∞-categories.

$\textsc{Catt}_{\mathsf{sua}}$ terms are simpler than their $\textsc{Catt}$ or $\textsc{Catt}_{\mathsf{su}}$ equivalents.

Normal forms for $\textsc{Catt}_{\mathsf{sua}}$ are obtained via a reduction system:

This reduction system is strongly terminating and confluent.

Try our interpreter for $\textsc{Catt}_{\mathsf{sua}}$:

`https://github.com/alexarice/catt-strict`

Thank you for listening.