# Coherence and normalisation-by-evaluation for bicategorical cartesian closed structure

Marcelo Fiore[†] and Philip Saville[*]

[†]University of Cambridge
Department of Computer Science
and Technology

[*]University of Edinburgh
School of Informatics

Cartesian closed bicategories:

- Generalised species and cartesian distributors
  (linear logic, higher category theory)
  (Fiore, Gambino, Hyland, Winskel), (Fiore & Joyal)
- Categorical algebra (operads)
  (Gambino & Joyal)
- Game semantics (concurrent games)
  (Yamada & Abramsky, Winskel *et al.*, Paquet)
- ‘STLC with explicit substitution and invertible $\beta\eta$-rewrites’
  (free cartesian closed bicategory)
  (Fiore & S., LICS 2019)

Cartesian closed bicategories:

- Generalised species and cartesian distributors
  (linear logic, higher category theory)
  (Fiore, Gambino, Hyland, Winskel), (Fiore & Joyal)
- Categorical algebra (operads)
  (Gambino & Joyal)
- Game semantics (concurrent games)
  (Yamada & Abramsky, Winskel *et al.*, Paquet)
- 'STLC with explicit substitution and invertible $\beta\eta$-rewrites'
  (free cartesian closed bicategory)
  (Fiore & S., LICS 2019)

In the paper:

two proofs of coherence $\rightsquigarrow$ 'all (pasting) diagrams commute'

Cartesian closed bicategories:

- Generalised species and cartesian distributors
  (linear logic, higher category theory)
  (Fiore, Gambino, Hyland, Winskel), (Fiore & Joyal)
- Categorical algebra (operads)
  (Gambino & Joyal)
- Game semantics (concurrent games)
  (Yamada & Abramsky, Winskel *et al.*, Paquet)
- 'STLC with explicit substitution and invertible $\beta\eta$-rewrites'
  (free cartesian closed bicategory)
  (Fiore & S., LICS 2019)

In the paper:

two proofs of coherence $\rightsquigarrow$ 'all (pasting) diagrams commute'

Consequences:

Cartesian closed bicategories:

- Generalised species and cartesian distributors
  (linear logic, higher category theory)
  (Fiore, Gambino, Hyland, Winskel), (Fiore & Joyal)
- Categorical algebra (operads)
  (Gambino & Joyal)
- Game semantics (concurrent games)
  (Yamada & Abramsky, Winskel *et al.*, Paquet)
- 'STLC with explicit substitution and invertible $\beta\eta$-rewrites'
  (free cartesian closed bicategory)
  (Fiore & S., LICS 2019)

In the paper:

two proofs of coherence $\rightsquigarrow$ 'all (pasting) diagrams commute'

Consequences:

1. Modulo $\equiv$ there is at most one rewrite $t \rightsquigarrow_{\beta\eta} t'$

Cartesian closed bicategories:

- Generalised species and cartesian distributors
  (linear logic, higher category theory)
  (Fiore, Gambino, Hyland, Winskel), (Fiore & Joyal)
- Categorical algebra (operads)
  (Gambino & Joyal)
- Game semantics (concurrent games)
  (Yamada & Abramsky, Winskel *et al.*, Paquet)
- 'STLC with explicit substitution and invertible $\beta\eta$-rewrites'
  (free cartesian closed bicategory)
  (Fiore & S., LICS 2019)

In the paper:

two proofs of coherence $\rightsquigarrow$ 'all (pasting) diagrams commute'

Consequences:

1. Modulo $\equiv$ there is at most one rewrite $t \rightsquigarrow_{\beta\eta} t'$
2. Constructions in cc-bicategories simplify to STLC

# Cartesian closed bicategories

## Composition by universal property ⇒ bicategory

In a category $\mathbb{C}$ with pullbacks:

1. objects: objects of $\mathbb{C}$,
2. 1-cells $A \rightsquigarrow B$: spans $(A \leftarrow X \rightarrow B)$,
3. 2-cells: commutative squares

$$\begin{tikzcd}
 & X \arrow[dl] \arrow[dr] \arrow[dd, dashed, "h"] & \\
A & & B \\
 & Y \arrow[ul] \arrow[ur] &
\end{tikzcd}$$

Composition defined by pullback:

$$\begin{tikzcd}
 & & X \times_B Y \arrow[dl] \arrow[dr] \arrow[dd, phantom, very near start, "\lrcorner" rotate=-45] & & \\
 & X \arrow[dl] \arrow[dr] & & Y \arrow[dl] \arrow[dr] & \\
A & & B & & C
\end{tikzcd}$$

## Composition by universal property ⇒ bicategory

In a category $\mathbb{C}$ with pullbacks:

1. objects: objects of $\mathbb{C}$,
2. 1-cells $A \rightsquigarrow B$: spans $(A \leftarrow X \rightarrow B)$,
3. 2-cells: commutative squares

$$\begin{tikzcd}
 & X \arrow[dl] \arrow[dr] \arrow[dd, dashed, "h"] & \\
A & & B \\
 & Y \arrow[ul] \arrow[ur] &
\end{tikzcd}$$

Composition defined by pullback: $\rightsquigarrow$ associative up to iso

$$\begin{tikzcd}
 & & X \times_B Y \arrow[dl] \arrow[dr] \arrow[dd, phantom, very near start, "\lrcorner"] & & \\
 & X \arrow[dl] \arrow[dr] & & Y \arrow[dl] \arrow[dr] & \\
A & & B & & C
\end{tikzcd}$$

# Bicategories

# Bicategories

- Objects $X, Y, Z, \dots$

# Bicategories

- Objects $X, Y, Z, \ldots$
- Morphisms (*1-cells*) $f : X \to Y$

# Bicategories

- Objects $X, Y, Z, \ldots$
- Morphisms (*1-cells*) $f : X \to Y$
- 2-cells $$\begin{tikzcd} X \arrow[r, bend left, "f"{name=U}] \arrow[r, bend right, "f'"'{name=D}] & Y \arrow[from=U, to=D, Rightarrow, "\alpha"] \end{tikzcd}$$

# Bicategories

- Objects $X, Y, Z, \dots$
- Morphisms (*1-cells*) $f : X \to Y$
- 2-cells $$\begin{tikzcd} X \arrow[r, bend left, "f"{name=U}] \arrow[r, bend right, "f'"'{name=D}] & Y \arrow[from=U, to=D, phantom, "\Downarrow \alpha"] \end{tikzcd}$$
- Invertible 2-cells witnessing the axioms

$$(h \circ g) \circ f \xRightarrow{\mathbf{a}_{h,g,f}} h \circ (g \circ f)$$

$$\mathrm{Id}_X \circ f \overset{\mathbf{l}_f}{\Longrightarrow} f$$

$$g \circ \mathrm{Id}_X \overset{\mathbf{r}_g}{\Longrightarrow} g$$

# Bicategories

- Objects $X, Y, Z, \ldots$
- Morphisms (*1-cells*) $f : X \to Y$
- 2-cells $$\begin{tikzcd} X \arrow[r, bend left, "f"{name=U}] \arrow[r, bend right, "f'"'{name=D}] & Y \arrow[from=U, to=D, Rightarrow, "\alpha"] \end{tikzcd}$$
- Invertible 2-cells witnessing the axioms

$$(h \circ g) \circ f \xRightarrow{\mathbf{a}_{h,g,f}} h \circ (g \circ f)$$

$$\mathrm{Id}_X \circ f \overset{\mathbf{l}_f}{\Longrightarrow} f$$

$$g \circ \mathrm{Id}_X \overset{\mathbf{r}_g}{\Longrightarrow} g$$

subject to a triangle law and pentagon law.

# Bicategories

- Objects $X, Y, Z, \ldots$
- Morphisms (*1-cells*) $f : X \to Y$
- 2-cells $$\begin{tikzcd} X \arrow[r, bend left, "f"{name=U}] \arrow[r, bend right, "f'"'{name=D}] & Y \arrow[from=U, to=D, Rightarrow, "\alpha"] \end{tikzcd}$$
- Invertible 2-cells witnessing the axioms

$$(h \circ g) \circ f \xRightarrow{\mathbf{a}_{h,g,f}} h \circ (g \circ f)$$

$$\mathrm{Id}_X \circ f \overset{\mathbf{l}_f}{\Longrightarrow} f$$

$$g \circ \mathrm{Id}_X \overset{\mathbf{r}_g}{\Longrightarrow} g$$

subject to a triangle law and pentagon law.

⇝ 'Monoidal category with many objects'

### Bicategories everywhere (not all cartesian closed!)

1. Monoidal category = one-object bicategory
2. 2-category = bicategory with $\mathsf{a}, \mathsf{l}, \mathsf{r}$ all $\mathrm{id}$
3. $\mathrm{Span}(\mathbb{C})$
4. Proof-relevant relations
5. Profunctors (distributors)
6. Polynomial functors (W-types, ornaments, containers, . . . )
7. Concurrent games

# Cartesian closed bicategories

Bicategories $\mathcal{B}$ equipped with

# Cartesian closed bicategories

Bicategories $\mathcal{B}$ equipped with families of equivalences

$$\mathcal{B}(X, A_1 \times A_2) \quad \simeq \quad \mathcal{B}(X, A_1) \times \mathcal{B}(X, A_2)$$

$$\mathcal{B}(X, A \Rightarrow B) \quad \simeq \quad \mathcal{B}(X \times A, B)$$

**NB:** Differ from the 'cartesian bicategories' of Carboni and Walters!

# Cartesian closed bicategories

Bicategories $\mathcal{B}$ equipped with families of equivalences

$$\begin{tikzcd}
\mathcal{B}(X, A_1 \times A_2) \arrow[r, bend left, "(\pi_{\mathbf{1}} \circ -, \pi_{\mathbf{2}} \circ -)"] \arrow[r, phantom, "\perp\simeq"] & \mathcal{B}(X, A_1) \times \mathcal{B}(X, A_2) \arrow[l, bend left, "\langle -, = \rangle"]
\end{tikzcd}$$

(pairing)

$$\begin{tikzcd}
\mathcal{B}(X, A \Rightarrow B) \arrow[r, bend left, "\mathrm{eval}_{A,B} \circ (- \times A)"] \arrow[r, phantom, "\perp\simeq"] & \mathcal{B}(X \times A, B) \arrow[l, bend left, "\lambda"]
\end{tikzcd}$$

(currying)

**NB:** Differ from the ‘cartesian bicategories’ of Carboni and Walters!

# Cartesian closed bicategories

Bicategories $\mathcal{B}$ equipped with families of equivalences

$$\begin{tikzcd}
\mathcal{B}(X, A_1 \times A_2) \arrow[r, bend left, "(\pi_1 \circ -, \pi_2 \circ -)"] \arrow[r, phantom, "\perp\simeq"] & \mathcal{B}(X, A_1) \times \mathcal{B}(X, A_2) \arrow[l, bend left, "\langle -, = \rangle"]
\end{tikzcd}$$

(pairing)

$$\begin{tikzcd}
\mathcal{B}(X, A \Rightarrow B) \arrow[r, bend left, "\mathrm{eval}_{A,B} \circ (- \times A)"] \arrow[r, phantom, "\perp\simeq"] & \mathcal{B}(X \times A, B) \arrow[l, bend left, "\lambda"]
\end{tikzcd}$$

(currying)

$$\pi_i \circ \langle f_1, f_2 \rangle \overset{\cong}{\Longrightarrow} f_i \qquad g \overset{\cong}{\Longrightarrow} \langle \pi_1 \circ g, \pi_2 \circ g \rangle$$

# Cartesian closed bicategories

Bicategories $\mathcal{B}$ equipped with families of equivalences

$$\begin{tikzcd}
\mathcal{B}(X, A_1 \times A_2) \arrow[r, bend left, "(\pi_{\mathbf{1}} \circ -{,} \pi_{\mathbf{2}} \circ -)"] \arrow[r, phantom, "\perp\simeq"] & \mathcal{B}(X, A_1) \times \mathcal{B}(X, A_2) \arrow[l, bend left, "\langle -{,} = \rangle"]
\end{tikzcd}$$

(pairing)

$$\begin{tikzcd}
\mathcal{B}(X, A \Rightarrow B) \arrow[r, bend left, "\mathrm{eval}_{A,B} \circ (- \times A)"] \arrow[r, phantom, "\perp\simeq"] & \mathcal{B}(X \times A, B) \arrow[l, bend left, "\lambda"]
\end{tikzcd}$$

(currying)

$$\pi_i \circ \langle f_1, f_2 \rangle \xRightarrow{\cong} f_i \qquad g \xRightarrow{\cong} \langle \pi_1 \circ g, \pi_2 \circ g \rangle$$

$$\mathrm{eval}_{A,B} \circ (\lambda f \times A) \xRightarrow{\cong} f \qquad g \xRightarrow{\cong} \lambda(\mathrm{eval}_{A,B} \circ (g \times A))$$

# Cartesian closed bicategories

Bicategories $\mathcal{B}$ equipped with families of equivalences

$$\begin{tikzcd}
\mathcal{B}(X, A_1 \times A_2) \arrow[rr, bend left, "(\pi_1 \circ -, \pi_2 \circ -)"] \arrow[r, phantom, "\perp\simeq"] & {} & \mathcal{B}(X, A_1) \times \mathcal{B}(X, A_2) \arrow[ll, bend left, "\langle -, = \rangle"]
\end{tikzcd}$$

(pairing)

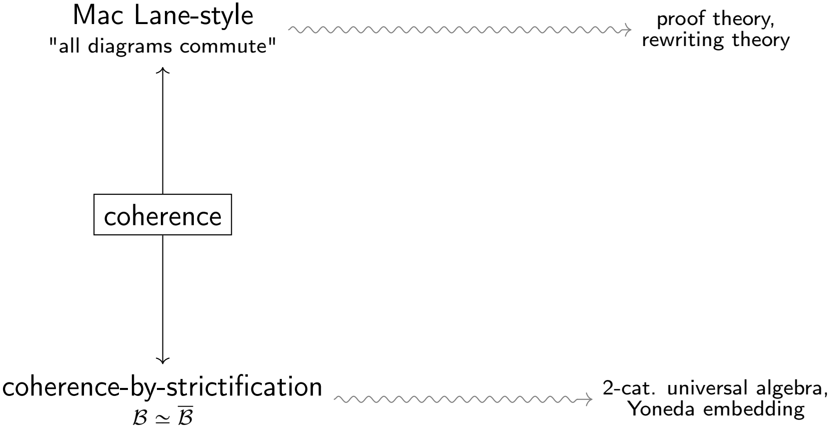
$$\begin{tikzcd}
\mathcal{B}(X, A \Rightarrow B) \arrow[rr, bend left, "\mathrm{eval}_{A,B} \circ (- \times A)"] \arrow[r, phantom, "\perp\simeq"] & {} & \mathcal{B}(X \times A, B) \arrow[ll, bend left, "\lambda"]
\end{tikzcd}$$

(currying)

$$\pi_i \circ \langle f_1, f_2 \rangle \overset{\cong}{\Longrightarrow} f_i \qquad g \overset{\cong}{\Longrightarrow} \langle \pi_1 \circ g, \pi_2 \circ g \rangle$$

$$\begin{tikzcd}
 & \pi_i \circ \langle \pi_1 \circ g, \pi_2 \circ g \rangle \arrow[dr, Rightarrow, "\beta_i^{\times}"] & \\
\pi_i \circ g \arrow[ur, Rightarrow, "\pi_i \circ \eta^{\times}"] \arrow[rr, Rightarrow, "\mathrm{id}"'] & & \pi_i \circ g
\end{tikzcd}$$

# Coherence



Mac Lane-style

"all diagrams commute"

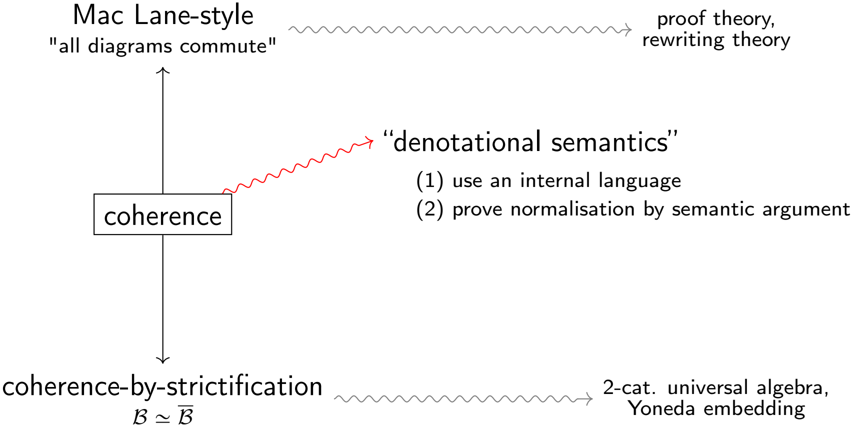
proof theory,
rewriting theory

coherence

coherence-by-strictification

$\mathcal{B} \simeq \overline{\mathcal{B}}$

2-cat. universal algebra,
Yoneda embedding



Mac Lane-style

"all diagrams commute"

proof theory,
rewriting theory

“denotational semantics”

(1) use an internal language

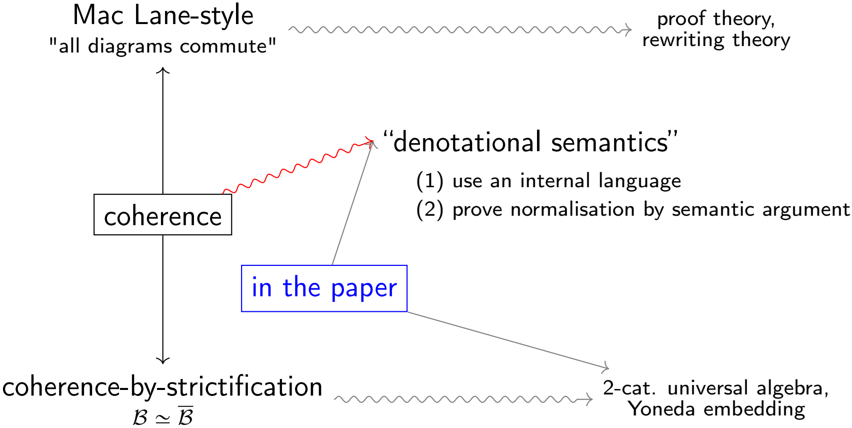
(2) prove normalisation by semantic argument

coherence

coherence-by-strictification

$\mathcal{B} \simeq \overline{\mathcal{B}}$

2-cat. universal algebra,
Yoneda embedding



Mac Lane-style

"all diagrams commute"

proof theory,
rewriting theory

"denotational semantics"

(1) use an internal language

(2) prove normalisation by semantic argument

coherence

in the paper

coherence-by-strictification

$\mathcal{B} \simeq \overline{\mathcal{B}}$

2-cat. universal algebra,
Yoneda embedding

$\boxed{\text{coherence}} \rightsquigarrow$ “denotational semantics”

(1) use an internal language
(2) prove normalisation by semantic argument

- builds on categorical & type-theoretic intuition
- *once set up* about as hard as categorical proof

Theorem (Coherence of cc-bicategories)

*For any* $f, f' : X \to Y$ *in the free cc-bicategory on a graph, there exists at most one 2-cell* $\tau : f \Rightarrow f'$*.* □

### Theorem (Coherence of cc-bicategories)

*For any $f, f' : X \to Y$ in the free cc-bicategory on a graph, there exists at most one 2-cell $\tau : f \Rightarrow f'$.* ☐

Consequence 1:

$$\boxed{\text{modulo} \equiv \text{there is at most one rewrite } t \rightsquigarrow_{\beta\eta} t'}$$

### Theorem (Coherence of cc-bicategories)

*For any* $f, f' : X \to Y$ *in the free cc-bicategory on a graph, there exists at most one 2-cell* $\tau : f \Rightarrow f'$*.* □

Consequence 1:

$$\boxed{\text{modulo} \equiv \text{there is at most one rewrite } t \rightsquigarrow_{\beta\eta} t'}$$

Consequence 2:

$$\boxed{\text{can use STLC for constructions in cc-bicategories}}$$

Theorem (Coherence of cc-bicategories)

*For any* $f, f' : X \to Y$ *in the free cc-bicategory on a graph, there exists at most one 2-cell* $\tau : f \Rightarrow f'$. □

Consequence 1:

modulo $\equiv$ there is at most one rewrite $t \rightsquigarrow_{\beta\eta} t'$

Consequence 2:

can use STLC for constructions in cc-bicategories

# The internal language for cc-bicategories, $\Lambda_{\mathrm{ps}}^{\times,\rightarrow}$ (Fiore & S., 2019)

## Judgements *c.f.* Seely, Hilken, Hirschowitz

| | | |
|---|---|---|
| *Terms* | $\Gamma \vdash t : A$ | (1-cells) |
| *Rewrites* | $\Gamma \vdash \tau : t \Rightarrow t' : A$ | (2-cells) |
| *Equations* | $\Gamma \vdash \tau \equiv \tau' : t \Rightarrow t' : A$ | |

# The internal language for cc-bicategories, $\Lambda_{\mathrm{ps}}^{\times,\rightarrow}$ (Fiore & S., 2019)

## Judgements *c.f.* Seely, Hilken, Hirschowitz

| | | |
|---|---|---|
| *Terms* | $\Gamma \vdash t : A$ | (1-cells) |
| *Rewrites* | $\Gamma \vdash \tau : t \Rightarrow t' : A$ | (2-cells) |
| *Equations* | $\Gamma \vdash \tau \equiv \tau' : t \Rightarrow t' : A$ | |

## Features

# The internal language for cc-bicategories, $\Lambda_{\mathrm{ps}}^{\times,\to}$ (Fiore & S., 2019)

## Judgements *c.f.* Seely, Hilken, Hirschowitz

| | | |
|---|---|---|
| *Terms* | $\Gamma \vdash t : A$ | (1-cells) |
| *Rewrites* | $\Gamma \vdash \tau : t \Rightarrow t' : A$ | (2-cells) |
| *Equations* | $\Gamma \vdash \tau \equiv \tau' : t \Rightarrow t' : A$ | |

## Features

- Weak composition enforced by *explicit substitution*

$$\frac{x_1 : A_1, \ldots, x_n : A_n \vdash t : B \qquad (\Delta \vdash u_i : A_i)_{i=1.,n}}{\Delta \vdash t\left\{x_i \mapsto u_i\right\} : B}$$

$$\frac{x_1 : A_1, \ldots, x_n : A_n \vdash \tau : t \Rightarrow t' : B \qquad (\Delta \vdash \sigma_i : u_i \Rightarrow u_i' : A_i)_{i=1,\ldots,n}}{\Delta \vdash \tau\left\{x_i \mapsto \sigma_i\right\} : t\left\{x_i \mapsto u_i\right\} \Rightarrow t'\left\{x_i \mapsto u_i'\right\} : B}$$

$\rightsquigarrow$ binds the variables $x_1, \ldots, x_n$

# The internal language for cc-bicategories, $\Lambda_{\mathrm{ps}}^{\times,\to}$ (Fiore & S., 2019)

## Judgements *c.f.* Seely, Hilken, Hirschowitz

| | | |
|---|---|---|
| *Terms* | $\Gamma \vdash t : A$ | (1-cells) |
| *Rewrites* | $\Gamma \vdash \tau : t \Rightarrow t' : A$ | (2-cells) |
| *Equations* | $\Gamma \vdash \tau \equiv \tau' : t \Rightarrow t' : A$ | |

## Features

- Weak composition enforced by *explicit substitution*
- STLC embeds as explicit-substitution free fragment

# The internal language for cc-bicategories, $\Lambda_{\mathrm{ps}}^{\times,\to}$ (Fiore & S., 2019)

## Judgements *c.f.* Seely, Hilken, Hirschowitz

| | | |
|---|---|---|
| *Terms* | $\Gamma \vdash t : A$ | (1-cells) |
| *Rewrites* | $\Gamma \vdash \tau : t \Rightarrow t' : A$ | (2-cells) |
| *Equations* | $\Gamma \vdash \tau \equiv \tau' : t \Rightarrow t' : A$ | |

## Features

- Weak composition enforced by *explicit substitution*
- STLC embeds as explicit-substitution free fragment
- Free property for syntactic model

$$\begin{tikzcd}
[\![\Gamma]\!] \arrow[r, bend left, "{[\![\Gamma \vdash t : A]\!]}"] \arrow[r, bend right, "{[\![\Gamma \vdash t' : A]\!]}"'] \arrow[r, phantom, "{\Downarrow [\![\Gamma \vdash \tau : t \Rightarrow t' : A]\!]}"] & [\![A]\!]
\end{tikzcd}$$

# The internal language for cc-bicategories, $\Lambda_{\mathrm{ps}}^{\times,\rightarrow}$ (Fiore & S., 2019)

## Judgements *c.f.* Seely, Hilken, Hirschowitz

| | | |
|---|---|---|
| *Terms* | $\Gamma \vdash t : A$ | (1-cells) |
| *Rewrites* | $\Gamma \vdash \tau : t \Rightarrow t' : A$ | (2-cells) |
| *Equations* | $\Gamma \vdash \tau \equiv \tau' : t \Rightarrow t' : A$ | |

## Features

- Weak composition enforced by *explicit substitution*
- STLC embeds as explicit-substitution free fragment
- Free property for syntactic model

$\rightsquigarrow$ A logic of program transformations (modulo equations)

$$\boxed{\text{there exists a rewrite } (\!|\, t \,|\!) \Rightarrow (\!|\, t' \,|\!) \quad \text{iff} \quad t \rightsquigarrow_{\beta\eta} t'}$$

embedding of STLC

# The internal language for cc-bicategories, $\Lambda_{\mathrm{ps}}^{\times,\to}$ (Fiore & S., 2019)

## Judgements *c.f.* Seely, Hilken, Hirschowitz

| | | |
|---|---|---|
| *Terms* | $\Gamma \vdash t : A$ | (1-cells) |
| *Rewrites* | $\Gamma \vdash \tau : t \Rightarrow t' : A$ | (2-cells) |
| *Equations* | $\Gamma \vdash \tau \equiv \tau' : t \Rightarrow t' : A$ | |

## Features

- Weak composition enforced by *explicit substitution*
- STLC embeds as explicit-substitution free fragment
- Free property for syntactic model

$\rightsquigarrow$ A logic of program transformations (modulo equations)

$$\boxed{\text{there exists a rewrite } (\!| t |\!) \Rightarrow (\!| t' |\!) \quad \text{iff} \quad t \rightsquigarrow_{\beta\eta} t'}$$

embedding of STLC

$\rightsquigarrow$ Coherence: modulo $\equiv$ there exists at most one rewrite $\tau$ such that $\Gamma \vdash \tau : t \Rightarrow t' : A$.

# The internal language for cc-bicategories, $\Lambda_{\mathrm{ps}}^{\times,\to}$ (Fiore & S., 2019)

## Judgements *c.f.* Seely, Hilken, Hirschowitz

| | | |
|---|---|---|
| *Terms* | $\Gamma \vdash t : A$ | (1-cells) |
| *Rewrites* | $\Gamma \vdash \tau : t \Rightarrow t' : A$ | (2-cells) |
| *Equations* | $\Gamma \vdash \tau \equiv \tau' : t \Rightarrow t' : A$ | |

## Features

- Weak composition enforced by *explicit substitution*
- STLC embeds as explicit-substitution free fragment
- Free property for syntactic model

$\rightsquigarrow$ A logic of program transformations (modulo equations)

$$\boxed{\text{there exists a rewrite } (\!| t |\!) \Rightarrow (\!| t' |\!) \quad \text{iff} \quad t \rightsquigarrow_{\beta\eta} t'}$$

embedding of STLC

$\rightsquigarrow$ Coherence: modulo $\equiv$ there exists at most one rewrite $\tau$ such that $\Gamma \vdash \tau : t \Rightarrow t' : A$.

# The internal language for cc-bicategories, $\Lambda_{\mathrm{ps}}^{\times,\rightarrow}$ (Fiore & S., 2019)

## Judgements *c.f.* Seely, Hilken, Hirschowitz

| | | |
|---|---|---|
| *Terms* | $\Gamma \vdash t : A$ | (1-cells) |
| *Rewrites* | $\Gamma \vdash \tau : t \Rightarrow t' : A$ | (2-cells) |
| *Equations* | $\Gamma \vdash \tau \equiv \tau' : t \Rightarrow t' : A$ | |

## Features

- Weak composition enforced by *explicit substitution*
- STLC embeds as explicit-substitution free fragment
- Free property for syntactic model

$\rightsquigarrow$ A logic of program transformations (modulo equations)

$$\boxed{\text{there exists a rewrite } (\!| t |\!) \Rightarrow (\!| t' |\!) \quad \text{iff} \quad t \rightsquigarrow_{\beta\eta} t'}$$

by coherence, must be unique

embedding of STLC

$\rightsquigarrow$ Coherence: modulo $\equiv$ there exists at most one rewrite $\tau$ such that $\Gamma \vdash \tau : t \Rightarrow t' : A$.

Theorem (Coherence of cc-bicategories)

*For any $f, f' : X \to Y$ in the free cc-bicategory on a graph, there exists at most one 2-cell $\tau : f \Rightarrow f'$.* ☐

Consequence 1:

modulo $\equiv$ there is at most one rewrite $t \rightsquigarrow_{\beta\eta} t'$

Consequence 2:

can use STLC for constructions in cc-bicategories

# Internal monoids

In a category with finite products:

$$1 \xrightarrow{e} M \xleftarrow{m} M \times M$$

$\boxed{\text{Unit law}}$

$$\begin{tikzcd}
1 \times M \arrow[r, "e \times M"] \arrow[dr, bend right, "\cong"'] & M \times M \arrow[d, "m"] & M \times 1 \arrow[l, "M \times e"'] \arrow[dl, bend left, "\cong"] \\
& M &
\end{tikzcd}$$

$\boxed{\text{Assoc. law}}$

$$\begin{tikzcd}
(M \times M) \times M \arrow[r, "\cong"] \arrow[d, "m \times M"'] & M \times (M \times M) \arrow[r, "M \times m"] & M \times M \arrow[d, "m"] \\
M \times M \arrow[rr, "m"'] & & M
\end{tikzcd}$$

# Internal monoids

In a category with finite products:

$$1 \xrightarrow{e} M \xleftarrow{m} M \times M$$

In **Set**: monoids
In **Cat**: strict monoidal categories

**Unit law**

$$\begin{tikzcd}
1 \times M \arrow[r, "e \times M"] \arrow[dr, bend right, "\cong"'] & M \times M \arrow[d, "m"] & M \times 1 \arrow[l, "M \times e"'] \arrow[dl, bend left, "\cong"] \\
 & M & 
\end{tikzcd}$$

**Assoc. law**

$$\begin{tikzcd}
(M \times M) \times M \arrow[r, "\cong"] \arrow[d, "m \times M"'] & M \times (M \times M) \arrow[r, "M \times m"] & M \times M \arrow[d, "m"] \\
M \times M \arrow[rr, "m"'] & & M
\end{tikzcd}$$

# Internal pseudomonoids

In **Cat**:

$$1 \xrightarrow{e} M \xleftarrow{m} M \times M$$

**Unit 2-cells**

$$\begin{tikzcd}
1 \times M \arrow[rr, "e \times M"] \arrow[ddrr, bend right, "\simeq"'] & & M \times M \arrow[dd, "m"] & & M \times 1 \arrow[ll, "M \times e"'] \arrow[ddll, bend left, "\simeq"] \\
 & \overset{\lambda}{\cong} & & \overset{\rho}{\cong} & \\
 & & M & &
\end{tikzcd}$$

data

**Assoc. 2-cell**

$$\begin{tikzcd}
(M \times M) \times M \arrow[r, "\simeq"] \arrow[dd, "m \times M"'] & M \times (M \times M) \arrow[r, "M \times m"] & M \times M \arrow[dd, "m"] \\
 & \overset{\alpha}{\cong} & \\
M \times M \arrow[rr, "m"'] & & M
\end{tikzcd}$$

# Internal pseudomonoids

In **Cat**:

$$1 \xrightarrow{e} M \xleftarrow{m} M \times M$$

Unit 2-cells

$$\begin{tikzcd}
1 \times M \arrow[r, "e \times M"] \arrow[dr, bend right, "\simeq"'] \arrow[dr, phantom, "\overset{\lambda}{\cong}", near end, shift left=4] & M \times M \arrow[d, "m"] & M \times 1 \arrow[l, "M \times e"'] \arrow[dl, bend left, "\simeq"] \arrow[dl, phantom, "\overset{\rho}{\cong}", near end, shift right=4] \\
& M &
\end{tikzcd}$$

data

Assoc. 2-cell

$$\begin{tikzcd}
(M \times M) \times M \arrow[r, "\simeq"] \arrow[d, "m \times M"'] & M \times (M \times M) \arrow[r, "M \times m"] \arrow[d, phantom, "\overset{\alpha}{\cong}"] & M \times M \arrow[d, "m"] \\
M \times M \arrow[rr, "m"'] & & M
\end{tikzcd}$$

+ triangle and pentagon laws ⇝ monoidal category

# Internal pseudomonoids

In **Cat**:

$$1 \xrightarrow{e} M \xleftarrow{m} M \times M$$

...likewise in any fp-bicategory

Unit 2-cells

$$\begin{tikzcd}
1 \times M \arrow[r, "e \times M"] \arrow[dr, bend right, "\simeq"'] \arrow[dr, phantom, "\overset{\lambda}{\cong}"] & M \times M \arrow[d, "m"] & M \times 1 \arrow[l, "M \times e"'] \arrow[dl, bend left, "\simeq"] \arrow[dl, phantom, "\overset{\rho}{\cong}"] \\
& M &
\end{tikzcd}$$

data

Assoc. 2-cell

$$\begin{tikzcd}
(M \times M) \times M \arrow[r, "\simeq"] \arrow[d, "m \times M"'] & M \times (M \times M) \arrow[r, "M \times m"] \arrow[d, phantom, "\overset{\alpha}{\cong}"] & M \times M \arrow[d, "m"] \\
M \times M \arrow[rr, "m"'] & & M
\end{tikzcd}$$

\+ triangle and pentagon laws

⇝ monoidal category

$\boxed{*}$ In a CCC every $[X \Rightarrow X]$ becomes a monoid:

$$\left(1 \xrightarrow{\mathrm{Id}_X} [X \Rightarrow X] \overset{\circ}{\leftarrow} [X \Rightarrow X] \times [X \Rightarrow X]\right)$$

$\boxed{?}$ In a cc-bicategory every $[X \Rightarrow X]$ becomes a pseudomonoid:

$$\left(1 \xrightarrow{\mathrm{Id}_X} [X \Rightarrow X] \overset{\circ}{\leftarrow} [X \Rightarrow X] \times [X \Rightarrow X]\right)$$

need to check laws (*i.e.* triangle + pentagon)

$\boxed{*}$ In a CCC every $[X \Rightarrow X]$ becomes a monoid:

$$\left(1 \xrightarrow{\mathrm{Id}_X} [X \Rightarrow X] \xleftarrow{\circ} [X \Rightarrow X] \times [X \Rightarrow X]\right)$$

$\boxed{?}$ In a cc-bicategory every $[X \Rightarrow X]$ becomes a pseudomonoid:

$$\left(1 \xrightarrow{\mathrm{Id}_X} [X \Rightarrow X] \xleftarrow{\circ} [X \Rightarrow X] \times [X \Rightarrow X]\right)$$

need to check laws (*i.e.* triangle + pentagon)

with coherence theorem $\rightsquigarrow$ can just use STLC!

$\boxed{*}$ In a CCC every $[X \Rightarrow X]$ becomes a monoid:

$$\left(1 \xrightarrow{\mathrm{Id}_X} [X \Rightarrow X] \overset{\circ}{\leftarrow} [X \Rightarrow X] \times [X \Rightarrow X]\right)$$

$\boxed{?}$ In a cc-bicategory every $[X \Rightarrow X]$ becomes a pseudomonoid:

$$\left(1 \xrightarrow{\mathrm{Id}_X} [X \Rightarrow X] \overset{\circ}{\leftarrow} [X \Rightarrow X] \times [X \Rightarrow X]\right)$$

need to check laws (*i.e.* triangle + pentagon)

with coherence theorem $\rightsquigarrow$ can just use STLC!

1. prove $\boxed{*}$ in STLC

$\boxed{*}$ In a CCC every $[X \Rightarrow X]$ becomes a monoid:

$$\left(1 \xrightarrow{\mathrm{Id}_X} [X \Rightarrow X] \xleftarrow{\circ} [X \Rightarrow X] \times [X \Rightarrow X]\right)$$

$\boxed{?}$ In a cc-bicategory every $[X \Rightarrow X]$ becomes a pseudomonoid:

$$\left(1 \xrightarrow{\mathrm{Id}_X} [X \Rightarrow X] \xleftarrow{\circ} [X \Rightarrow X] \times [X \Rightarrow X]\right)$$

need to check laws (*i.e.* triangle + pentagon)

with coherence theorem $\rightsquigarrow$ can just use STLC!

1. prove $\boxed{*}$ in STLC
2. $\beta\eta$-equalities $\rightsquigarrow$ structural 2-cells

$\boxed{*}$ In a CCC every $[X \Rightarrow X]$ becomes a monoid:

$$\left(1 \xrightarrow{\mathrm{Id}_X} [X \Rightarrow X] \overset{\circ}{\leftarrow} [X \Rightarrow X] \times [X \Rightarrow X]\right)$$

$\boxed{?}$ In a cc-bicategory every $[X \Rightarrow X]$ becomes a pseudomonoid:

$$\left(1 \xrightarrow{\mathrm{Id}_X} [X \Rightarrow X] \overset{\circ}{\leftarrow} [X \Rightarrow X] \times [X \Rightarrow X]\right)$$

need to check laws (*i.e.* triangle + pentagon)

with coherence theorem ⇝ can just use STLC!

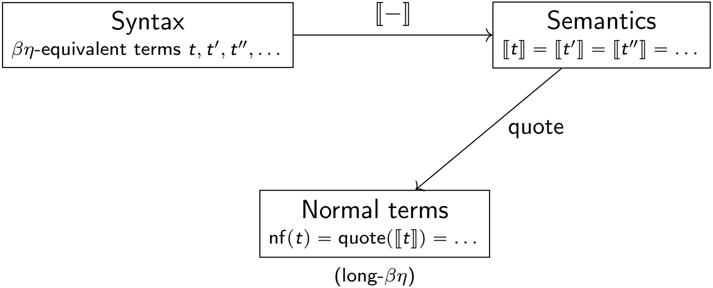
1. prove $\boxed{*}$ in STLC
2. $\beta\eta$-equalities ⇝ structural 2-cells
3. coherence guaranteed

# Normalisation-by-evaluation

# Normalisation-by-evaluation

Aim: find canonical representatives within each $\beta\eta$-equivalence class

Strategy:



**Syntax**

$\beta\eta$-equivalent terms $t, t', t'', \ldots$

$[\![-]\!]$

**Semantics**

$[\![t]\!] = [\![t']\!] = [\![t'']\!] = \ldots$

quote

**Normal terms**

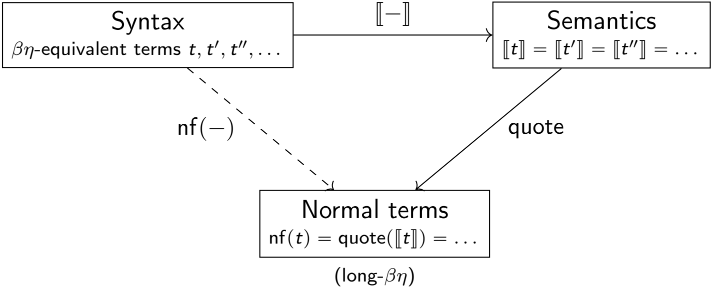
$\mathsf{nf}(t) = \mathsf{quote}([\![t]\!]) = \ldots$

(long-$\beta\eta$)

# Normalisation-by-evaluation

Aim: find canonical representatives within each $\beta\eta$-equivalence class

Strategy:



Syntax

$\beta\eta$-equivalent terms $t, t', t'', \dots$

$[\![-]\!]$

Semantics

$[\![t]\!] = [\![t']\!] = [\![t'']\!] = \dots$

$\mathsf{nf}(-)$

quote

Normal terms

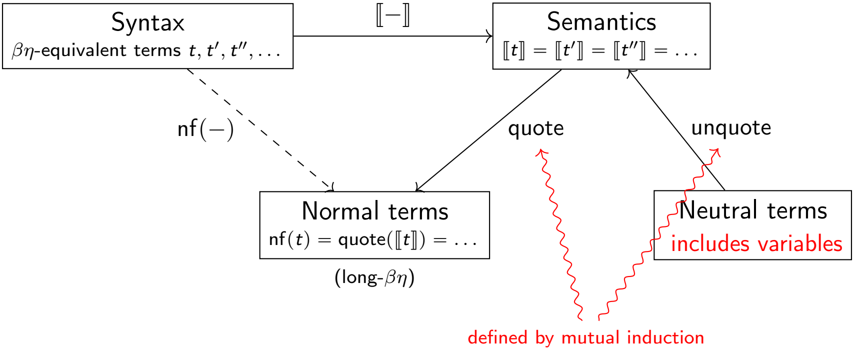
$\mathsf{nf}(t) = \mathsf{quote}([\![t]\!]) = \dots$

(long-$\beta\eta$)

# Normalisation-by-evaluation

Aim: find canonical representatives within each $\beta\eta$-equivalence class

Strategy:



Syntax

$\beta\eta$-equivalent terms $t, t', t'', \ldots$

$[\![-]\!]$

Semantics

$[\![t]\!] = [\![t']\!] = [\![t'']\!] = \ldots$

nf$(-)$

quote

unquote

Normal terms

nf$(t)$ = quote$([\![t]\!]) = \ldots$

(long-$\beta\eta$)

Neutral terms

includes variables

defined by mutual induction

# Normalisation-by-evaluation, categorically (Fiore 2002)

# Normalisation-by-evaluation, categorically (Fiore 2002)

Syntax as indexed presheaves over a *category of contexts* $\mathrm{Con}$:

$$\left.\begin{array}{r}\mathsf{neuts}_A : \Gamma \mapsto \{\text{neutral terms } t \text{ such that } \Gamma \vdash t : A\} \\ \mathsf{norms}_A : \Gamma \mapsto \{\text{normal terms } t \text{ such that } \Gamma \vdash t : A\}\end{array}\right\} : \mathrm{Con} \to \mathbf{Set}$$

# Normalisation-by-evaluation, categorically (Fiore 2002)

Syntax as indexed presheaves over a *category of contexts* $\mathrm{Con}$:

$$\left.\begin{array}{r}\mathsf{neuts}_A : \Gamma \mapsto \{\text{neutral terms } t \text{ such that } \Gamma \vdash t : A\} \\ \mathsf{norms}_A : \Gamma \mapsto \{\text{normal terms } t \text{ such that } \Gamma \vdash t : A\}\end{array}\right\} : \mathrm{Con} \to \mathbf{Set}$$

syntax glued to semantics by natural transformations

$$(\Gamma \vdash t : A) \mapsto s[\![\Gamma \vdash t : A]\!]$$

# Normalisation-by-evaluation, categorically (Fiore 2002)

Syntax as indexed presheaves over a *category of contexts* $\mathrm{Con}$:

$$\left.\begin{array}{r}\mathsf{neuts}_A : \Gamma \mapsto \{\text{neutral terms } t \text{ such that } \Gamma \vdash t : A\} \\ \mathsf{norms}_A : \Gamma \mapsto \{\text{normal terms } t \text{ such that } \Gamma \vdash t : A\}\end{array}\right\} : \mathrm{Con} \to \mathbf{Set}$$

syntax glued to semantics by natural transformations

$$(\Gamma \vdash t : A) \mapsto s[\![\Gamma \vdash t : A]\!]$$

$$s[\![-]\!] : \mathsf{neuts}_A \Rightarrow \mathbb{C}(s[\![-]\!], s[\![A]\!])$$

$$s[\![-]\!] : \mathsf{norms}_A \Rightarrow \mathbb{C}(s[\![-]\!], s[\![A]\!])$$

$s$ any interpretation of base types;

# Normalisation-by-evaluation, categorically (Fiore 2002)

Syntax as indexed presheaves over a *category of contexts* $\mathrm{Con}$:

$$\left.\begin{aligned} \mathsf{neuts}_A &: \Gamma \mapsto \{\text{neutral terms } t \text{ such that } \Gamma \vdash t : A\} \\ \mathsf{norms}_A &: \Gamma \mapsto \{\text{normal terms } t \text{ such that } \Gamma \vdash t : A\} \end{aligned}\right\} : \mathrm{Con} \to \mathbf{Set}$$

syntax glued to semantics by natural transformations

$$\begin{aligned} (\Gamma \vdash t : A) &\mapsto s[\![\Gamma \vdash t : A]\!] \\ s[\![-]\!] : \mathsf{neuts}_A &\Rightarrow \mathbb{C}(s[\![-]\!], s[\![A]\!]) \\ s[\![-]\!] : \mathsf{norms}_A &\Rightarrow \mathbb{C}(s[\![-]\!], s[\![A]\!]) \end{aligned}$$

$s$ any interpretation of
base types;
$\mathbb{C}$ any CCC

Strategy:

# Normalisation-by-evaluation, categorically (Fiore 2002)

Syntax as indexed presheaves over a *category of contexts* $\mathrm{Con}$:

$$\left.\begin{array}{r}\mathsf{neuts}_A : \Gamma \mapsto \{\text{neutral terms } t \text{ such that } \Gamma \vdash t : A\} \\ \mathsf{norms}_A : \Gamma \mapsto \{\text{normal terms } t \text{ such that } \Gamma \vdash t : A\}\end{array}\right\} : \mathrm{Con} \to \mathbf{Set}$$

syntax glued to semantics by natural transformations

$$(\Gamma \vdash t : A) \mapsto s[\![\Gamma \vdash t : A]\!]$$

$$s[\![-]\!] : \mathsf{neuts}_A \Rightarrow \mathbb{C}(s[\![-]\!], s[\![A]\!])$$

$$s[\![-]\!] : \mathsf{norms}_A \Rightarrow \mathbb{C}(s[\![-]\!], s[\![A]\!])$$

$s$ any interpretation of base types; $\mathbb{C}$ any CCC

Strategy:

# Normalisation-by-evaluation, categorically (Fiore 2002)

Syntax as indexed presheaves over a *category of contexts* $\mathrm{Con}$:

$$\left.\begin{aligned} \mathsf{neuts}_A &: \Gamma \mapsto \{\text{neutral terms } t \text{ such that } \Gamma \vdash t : A\} \\ \mathsf{norms}_A &: \Gamma \mapsto \{\text{normal terms } t \text{ such that } \Gamma \vdash t : A\} \end{aligned}\right\} : \mathrm{Con} \to \mathbf{Set}$$

syntax glued to semantics by natural transformations

$$\begin{aligned} (\Gamma \vdash t : A) &\mapsto s[\![\Gamma \vdash t : A]\!] \\ s[\![-]\!] : \mathsf{neuts}_A &\Rightarrow \mathbb{C}(s[\![-]\!], s[\![A]\!]) \\ s[\![-]\!] : \mathsf{norms}_A &\Rightarrow \mathbb{C}(s[\![-]\!], s[\![A]\!]) \end{aligned}$$

$s$ any interpretation of
base types;
$\mathbb{C}$ any CCC

Strategy:

1. define a cartesian closed glueing category $\mathbb{G}(\mathbb{C}, s)$

# Normalisation-by-evaluation, categorically (Fiore 2002)

Syntax as indexed presheaves over a *category of contexts* $\mathrm{Con}$:

$$\left.\begin{aligned} \mathsf{neuts}_A : \Gamma &\mapsto \{\text{neutral terms } t \text{ such that } \Gamma \vdash t : A\} \\ \mathsf{norms}_A : \Gamma &\mapsto \{\text{normal terms } t \text{ such that } \Gamma \vdash t : A\} \end{aligned}\right\} : \mathrm{Con} \to \mathbf{Set}$$

syntax glued to semantics by natural transformations

$$\begin{aligned} (\Gamma \vdash t : A) &\mapsto s[\![\Gamma \vdash t : A]\!] \\ s[\![-]\!] : \mathsf{neuts}_A &\Rightarrow \mathbb{C}(s[\![-]\!], s[\![A]\!]) \\ s[\![-]\!] : \mathsf{norms}_A &\Rightarrow \mathbb{C}(s[\![-]\!], s[\![A]\!]) \end{aligned}$$

$s$ any interpretation of base types;
$\mathbb{C}$ any CCC

Strategy:

1. define a cartesian closed glueing category $\mathbb{G}(\mathbb{C}, s)$
2. pick an interpretation $\mathrm{e}[\![-]\!]$ in $\mathbb{G}(\mathbb{C}, s)$

# Normalisation-by-evaluation, categorically (Fiore 2002)

Syntax as indexed presheaves over a *category of contexts* $\mathrm{Con}$:

$$\left.\begin{aligned} \mathsf{neuts}_A &: \Gamma \mapsto \{\text{neutral terms } t \text{ such that } \Gamma \vdash t : A\} \\ \mathsf{norms}_A &: \Gamma \mapsto \{\text{normal terms } t \text{ such that } \Gamma \vdash t : A\} \end{aligned}\right\} : \mathrm{Con} \to \mathbf{Set}$$

syntax glued to semantics by natural transformations

$$\begin{aligned} (\Gamma \vdash t : A) &\mapsto s[\![\Gamma \vdash t : A]\!] \\ s[\![-]\!] : \mathsf{neuts}_A &\Rightarrow \mathbb{C}(s[\![-]\!], s[\![A]\!]) \\ s[\![-]\!] : \mathsf{norms}_A &\Rightarrow \mathbb{C}(s[\![-]\!], s[\![A]\!]) \end{aligned}$$

$s$ any interpretation of base types;
$\mathbb{C}$ any CCC

Strategy:

1. define a cartesian closed glueing category $\mathbb{G}(\mathbb{C}, s)$
2. pick an interpretation $\mathrm{e}[\![-]\!]$ in $\mathbb{G}(\mathbb{C}, s)$
3. define quote and unquote as maps in this category

Strategy:

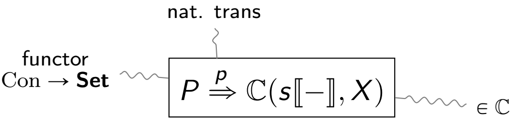
1. define a cartesian closed glueing category $\mathbb{G}(\mathbb{C}, s)$
2. pick an interpretation $\mathrm{e}[\![-]\!]$ in $\mathbb{G}(\mathbb{C}, s)$
3. define quote and unquote as maps in this category

# The glued CCC $\mathbb{G}(\mathbb{C}, s)$

**Objects**

$\mathbb{C}$ ccc,

$s$ : BaseTypes $\to \mathrm{Ob}\mathbb{C}$



nat. trans

functor

$\mathrm{Con} \to \textbf{Set}$

$$\boxed{P \overset{p}{\Rightarrow} \mathbb{C}(s[\![-]\!], X)}$$

$\in \mathbb{C}$

# The glued CCC $\mathbb{G}(\mathbb{C}, s)$

### Objects

$\mathbb{C}$ ccc,
$s$ : BaseTypes $\to \mathrm{Ob}\mathbb{C}$

nat. trans

functor
$\mathrm{Con} \to$ **Set**

$$\boxed{P \overset{p}{\Rightarrow} \mathbb{C}(s[\![-]\!], X)}$$

$\in \mathbb{C}$

### Morphisms

nat. trans $P \Rightarrow P'$

$$\begin{tikzcd}
P \arrow[r, "\kappa"] \arrow[d, "p"'] & P' \arrow[d, "p'"] \\
\mathbb{C}(s[\![-]\!], X) \arrow[r, "k \circ (-)"'] & \mathbb{C}(s[\![-]\!], X')
\end{tikzcd}$$

arrow $X \to X'$

# The glued CCC $\mathbb{G}(\mathbb{C}, s)$

## Objects

$\mathbb{C}$ ccc,
$s$ : BaseTypes $\to \mathrm{Ob}\mathbb{C}$

nat. trans

functor
$\mathrm{Con} \to$ **Set**

$$\boxed{P \overset{p}{\Rightarrow} \mathbb{C}(s[\![-]\!], X)}$$

$\in \mathbb{C}$

## Morphisms

nat. trans $P \Rightarrow P'$

"interpretation"

$$\begin{tikzcd}
\text{"syntax"} \arrow[r, squiggly, dash, red] & P \arrow[r, "\kappa"] \arrow[d, "p"'] & P' \arrow[d, "p'"] \\
\text{"semantics"} \arrow[r, squiggly, dash, red] & \mathbb{C}(s[\![-]\!], X) \arrow[r, "k \circ (-)"'] & \mathbb{C}(s[\![-]\!], X')
\end{tikzcd}$$

arrow $X \to X'$

# The glued CCC $\mathbb{G}(\mathbb{C}, s)$

**Objects**

$\mathbb{C}$ ccc, $s$ : BaseTypes $\to$ Ob$\mathbb{C}$

$$\begin{tikzcd}
 & \text{nat. trans} \arrow[d, squiggly] & \\
\text{functor} \atop \mathrm{Con} \to \textbf{Set} \arrow[r, squiggly] & \boxed{P \overset{p}{\Rightarrow} \mathbb{C}(s[\![-]\!], X)} \arrow[r, squiggly] & \in \mathbb{C}
\end{tikzcd}$$

**Morphisms**

$$\begin{tikzcd}
\text{"syntax"} \arrow[r, squiggly] & P \arrow[rr, "\kappa"] \arrow[dd, "p"'] & & P' \arrow[dd, "p'"] \\
\text{"interpretation"} \arrow[r, squiggly] & {} & & \\
\text{"semantics"} \arrow[r, squiggly] & \mathbb{C}(s[\![-]\!], X) \arrow[rr, "k \circ (-)"'] & & \mathbb{C}(s[\![-]\!], X')
\end{tikzcd}$$

# The glued CCC $\mathbb{G}(\mathbb{C}, s)$

**Objects**

$\mathbb{C}$ ccc, $s$ : BaseTypes $\to \mathrm{Ob}\mathbb{C}$

nat. trans

$$\underset{\mathrm{Con} \to \mathbf{Set}}{\text{functor}} \rightsquigarrow \boxed{P \overset{p}{\Rightarrow} \mathbb{C}(s[\![-]\!], X)} \leftsquigarrow \in \mathbb{C}$$

**Morphisms**

$$\begin{tikzcd}
\text{"syntax"} \arrow[r, squiggly, dash] & P \arrow[rr, "\kappa"] \arrow[dd, "p"'] & & P' \arrow[dd, "p'"] \\
\text{"interpretation"} \arrow[r, squiggly, dash] & {} & & \\
\text{"semantics"} \arrow[r, squiggly, dash] & \mathbb{C}(s[\![-]\!], X) \arrow[rr, "k \circ (-)"'] & & \mathbb{C}(s[\![-]\!], X')
\end{tikzcd}$$

$\rightsquigarrow$ for every type $A$,

$$\left.\begin{array}{l} \mathrm{neuts}_A \xRightarrow{s[\![-]\!]} \mathbb{C}(s[\![-]\!], s[\![A]\!]) \\ \mathrm{norms}_A \xRightarrow{s[\![-]\!]} \mathbb{C}(s[\![-]\!], s[\![A]\!]) \end{array}\right\} \in \mathbb{G}(\mathbb{C}, s)$$

Strategy:

1. define a cartesian closed glueing category $\mathbb{G}(\mathbb{C}, s)$
2. pick an interpretation $\mathrm{e}[\![-]\!]$ in $\mathbb{G}(\mathbb{C}, s)$
3. define quote and unquote as maps in this category

# Interpretation $\mathrm{e}$ in the glued CCC $\mathbb{G}(\mathbb{C}, s)$

$$\begin{aligned} \mathbb{C} &\text{ any ccc} \\ s : \mathrm{BaseTypes} &\to \mathrm{Ob}\mathbb{C} \end{aligned}$$

# Interpretation $\mathrm{e}$ in the glued CCC $\mathbb{G}(\mathbb{C}, s)$

$\mathbb{C}$ any ccc
$s : \mathsf{BaseTypes} \to \mathrm{Ob}\mathbb{C}$

$$\begin{tikzcd}
\bar{\mathrm{e}}[\![A]\!] \arrow[d, "\nu_A"] \\
\mathbb{C}(s[\![-]\!], s[\![A]\!])
\end{tikzcd}$$

# Interpretation $\mathrm{e}$ in the glued CCC $\mathbb{G}(\mathbb{C}, s)$

$\mathbb{C}$ any ccc
$s : \mathrm{BaseTypes} \to \mathrm{Ob}\mathbb{C}$

$$\begin{tikzcd}
\overline{\mathrm{e}}[\![\Gamma]\!] \arrow[r, "\overline{\mathrm{e}}[\![\Gamma \vdash t : A]\!]"] \arrow[d, "\nu_\Gamma"'] & \overline{\mathrm{e}}[\![A]\!] \arrow[d, "\nu_A"] \\
\mathbb{C}\big(s[\![-]\!], s[\![\Gamma]\!]\big) \arrow[r, "s[\![\Gamma \vdash t : A]\!] \circ (-)"'] & \mathbb{C}\big(s[\![-]\!], s[\![A]\!]\big)
\end{tikzcd}$$

# Interpretation $\mathrm{e}$ in the glued CCC $\mathbb{G}(\mathbb{C}, s)$

$\mathbb{C}$ any ccc

$s$ : BaseTypes → Ob$\mathbb{C}$

$$\begin{tikzcd}
\overline{\mathrm{e}}\llbracket \Gamma \rrbracket \arrow[r, "\overline{\mathrm{e}}\llbracket \Gamma \vdash t : A \rrbracket"] \arrow[d, "\nu_\Gamma"'] & \overline{\mathrm{e}}\llbracket A \rrbracket \arrow[d, "\nu_A"] \\
\mathbb{C}\big(s\llbracket - \rrbracket, s\llbracket \Gamma \rrbracket\big) \arrow[r, "s\llbracket \Gamma \vdash t : A \rrbracket \circ (-)"'] & \mathbb{C}\big(s\llbracket - \rrbracket, s\llbracket A \rrbracket\big)
\end{tikzcd}$$

$$\begin{tikzcd}
\mathsf{neuts}_A \arrow[r, "\mathsf{quote}_A"] \arrow[d, "s\llbracket - \rrbracket"'] & \overline{\mathrm{e}}\llbracket A \rrbracket \arrow[d, "\nu_A"] \\
\mathbb{C}\big(s\llbracket - \rrbracket, s\llbracket A \rrbracket\big) \arrow[r, equal, "\text{preserves } \beta\eta"'] & \mathbb{C}\big(s\llbracket - \rrbracket, s\llbracket A \rrbracket\big)
\end{tikzcd}$$

# Interpretation $\mathrm{e}$ in the glued CCC $\mathbb{G}(\mathbb{C}, s)$

$\mathbb{C}$ any ccc

$s$ : BaseTypes $\to \mathrm{Ob}\mathbb{C}$

$$\begin{tikzcd}
\bar{\mathrm{e}}\llbracket\Gamma\rrbracket \arrow[r, "\bar{\mathrm{e}}\llbracket\Gamma\vdash t:A\rrbracket"] \arrow[d, "\nu_\Gamma"'] & \bar{\mathrm{e}}\llbracket A\rrbracket \arrow[d, "\nu_A"] \\
\mathbb{C}\big(s\llbracket-\rrbracket, s\llbracket\Gamma\rrbracket\big) \arrow[r, "s\llbracket\Gamma\vdash t:A\rrbracket\circ(-)"'] & \mathbb{C}\big(s\llbracket-\rrbracket, s\llbracket A\rrbracket\big)
\end{tikzcd}$$

$$\begin{tikzcd}
\mathsf{neuts}_A \arrow[r, "\mathsf{quote}_A"] \arrow[d, "s\llbracket-\rrbracket"'] & \bar{\mathrm{e}}\llbracket A\rrbracket \arrow[d, "\nu_A"] \\
\mathbb{C}\big(s\llbracket-\rrbracket, s\llbracket A\rrbracket\big) \arrow[r, equal, "\text{preserves } \beta\eta"'] & \mathbb{C}\big(s\llbracket-\rrbracket, s\llbracket A\rrbracket\big)
\end{tikzcd}$$

$$\begin{tikzcd}
\bar{\mathrm{e}}\llbracket A\rrbracket \arrow[r, "\mathsf{unquote}_A"] \arrow[d, "\nu_A"'] & \mathsf{norms}_A \arrow[d, "s\llbracket-\rrbracket"] \\
\mathbb{C}\big(s\llbracket-\rrbracket, s\llbracket A\rrbracket\big) \arrow[r, equal, "\text{preserves } \beta\eta"'] & \mathbb{C}\big(s\llbracket-\rrbracket, s\llbracket A\rrbracket\big)
\end{tikzcd}$$

# Normalising $(\Gamma \vdash t : A)$

$$\mathsf{neuts}_\Gamma := \prod_{(x_i:A_i)\in\Gamma} \mathsf{neuts}_{A_i}$$

$$\mathsf{unquote}_\Gamma := \prod_{(x_i:A_i)\in\Gamma} \mathsf{unquote}_{A_i}$$

$$\begin{tikzcd}
\mathsf{neuts}_\Gamma \arrow[r, "\mathsf{unquote}_\Gamma"] \arrow[d, "\mathsf{s}\llbracket - \rrbracket"'] & \overline{\mathrm{e}}\llbracket \Gamma \rrbracket \arrow[r, "\overline{\mathrm{e}}\llbracket \Gamma \vdash t : A \rrbracket"] \arrow[d, "\nu_\Gamma"'] & \overline{\mathrm{e}}\llbracket A \rrbracket \arrow[r, "\mathsf{quote}_A"] \arrow[d, "\nu_A"] & \mathsf{norms}_A \arrow[d, "s\llbracket - \rrbracket"] \\
\mathbb{C}(s\llbracket - \rrbracket, s\llbracket \Gamma \rrbracket) \arrow[r, equal] & \mathbb{C}(s\llbracket - \rrbracket, s\llbracket \Gamma \rrbracket) \arrow[r, "s\llbracket \Gamma \vdash t : A \rrbracket \circ (-)"'] & \mathbb{C}(s\llbracket - \rrbracket, s\llbracket A \rrbracket) \arrow[r, equal] & \mathbb{C}(s\llbracket - \rrbracket, s\llbracket A \rrbracket)
\end{tikzcd}$$

# Normalising $(\Gamma \vdash t : A)$

$$\mathsf{neuts}_\Gamma := \prod_{(x_i : A_i) \in \Gamma} \mathsf{neuts}_{A_i}$$

$$\mathsf{unquote}_\Gamma := \prod_{(x_i : A_i) \in \Gamma} \mathsf{unquote}_{A_i}$$

$$\begin{tikzcd}
\mathsf{neuts}_\Gamma(\Gamma) \arrow[r, "\mathsf{unquote}_\Gamma"] \arrow[d, swap, "\mathsf{s}\llbracket - \rrbracket"] & \overline{\mathrm{e}}\llbracket \Gamma \rrbracket(\Gamma) \arrow[r, "\overline{\mathrm{e}}\llbracket \Gamma \vdash t : A \rrbracket"] \arrow[d, swap, "\nu_\Gamma"] & \overline{\mathrm{e}}\llbracket A \rrbracket(\Gamma) \arrow[r, "\mathsf{quote}_A"] \arrow[d, "\nu_A"] & \mathsf{norms}_A(\Gamma) \arrow[d, "s\llbracket - \rrbracket"] \\
\mathbb{C}(s\llbracket \Gamma \rrbracket, s\llbracket \Gamma \rrbracket) \arrow[r, equal] & \mathbb{C}(s\llbracket \Gamma \rrbracket, s\llbracket \Gamma \rrbracket) \arrow[r, swap, "s\llbracket \Gamma \vdash t : A \rrbracket \circ (-)"] & \mathbb{C}(s\llbracket \Gamma \rrbracket, s\llbracket A \rrbracket) \arrow[r, equal] & \mathbb{C}(s\llbracket \Gamma \rrbracket, s\llbracket A \rrbracket)
\end{tikzcd}$$

# Normalising $(\Gamma \vdash t : A)$

$$\mathsf{neuts}_{\Gamma} := \textstyle\prod_{(x_i : A_i) \in \Gamma} \mathsf{neuts}_{A_i}$$

$$\mathsf{unquote}_{\Gamma} := \textstyle\prod_{(x_i : A_i) \in \Gamma} \mathsf{unquote}_{A_i}$$

$$\begin{tikzcd}
\mathsf{neuts}_{\Gamma}(\Gamma) \arrow[r, "\mathsf{unquote}_{\Gamma}"] \arrow[rrr, bend left, "\mathsf{norm}^{\Gamma}_{t}"] \arrow[d, "\mathsf{s}\llbracket - \rrbracket"'] & \overline{\mathrm{e}}\llbracket \Gamma \rrbracket(\Gamma) \arrow[r, "\overline{\mathrm{e}}\llbracket \Gamma \vdash t : A \rrbracket"] \arrow[d, "\nu_{\Gamma}"'] & \overline{\mathrm{e}}\llbracket A \rrbracket(\Gamma) \arrow[r, "\mathsf{quote}_{A}"] \arrow[d, "\nu_{A}"] & \mathsf{norms}_{A}(\Gamma) \arrow[d, "\mathsf{s}\llbracket - \rrbracket"] \\
\mathbb{C}(s\llbracket \Gamma \rrbracket, s\llbracket \Gamma \rrbracket) \arrow[r, equal] & \mathbb{C}(s\llbracket \Gamma \rrbracket, s\llbracket \Gamma \rrbracket) \arrow[r, "s\llbracket \Gamma \vdash t : A \rrbracket \circ (-)"'] & \mathbb{C}(s\llbracket \Gamma \rrbracket, s\llbracket A \rrbracket) \arrow[r, equal] & \mathbb{C}(s\llbracket \Gamma \rrbracket, s\llbracket A \rrbracket)
\end{tikzcd}$$

Define $\mathsf{nf}(t) := \mathsf{norm}^{\Gamma}_{t}((\Gamma \vdash x_i : A_i)_{i=1,\dots,n})$.

# Normalising $(\Gamma \vdash t : A)$

$$\mathsf{neuts}_\Gamma := \textstyle\prod_{(x_i : A_i) \in \Gamma} \mathsf{neuts}_{A_i}$$

$$\mathsf{unquote}_\Gamma := \textstyle\prod_{(x_i : A_i) \in \Gamma} \mathsf{unquote}_{A_i}$$

$(x_i : A_i)_i$

$$\begin{tikzcd}
\mathsf{neuts}_\Gamma(\Gamma) \arrow[r, "\mathsf{unquote}_\Gamma"] \arrow[d, "s[\![-]\!]"'] \arrow[rrr, bend left, "\mathsf{norm}^\Gamma_t"] & \overline{\mathrm{e}}[\![\Gamma]\!](\Gamma) \arrow[r, "{\overline{\mathrm{e}}[\![\Gamma \vdash t : A]\!]}"] \arrow[d, "\nu_\Gamma"'] & \overline{\mathrm{e}}[\![A]\!](\Gamma) \arrow[r, "\mathsf{quote}_A"] \arrow[d, "\nu_A"] & \mathsf{norms}_A(\Gamma) \arrow[d, "s[\![-]\!]"] \\
\mathbb{C}(s[\![\Gamma]\!], s[\![\Gamma]\!]) \arrow[r, equal] & \mathbb{C}(s[\![\Gamma]\!], s[\![\Gamma]\!]) \arrow[r, "{s[\![\Gamma \vdash t : A]\!] \circ (-)}"'] & \mathbb{C}(s[\![\Gamma]\!], s[\![A]\!]) \arrow[r, equal] & \mathbb{C}(s[\![\Gamma]\!], s[\![A]\!])
\end{tikzcd}$$

Define $\mathsf{nf}(t) := \mathsf{norm}^\Gamma_t((\Gamma \vdash x_i : A_i)_{i=1,\ldots,n})$.

# Normalising $(\Gamma \vdash t : A)$

$$\mathsf{neuts}_\Gamma := \textstyle\prod_{(x_i : A_i) \in \Gamma} \mathsf{neuts}_{A_i}$$

$$\mathsf{unquote}_\Gamma := \textstyle\prod_{(x_i : A_i) \in \Gamma} \mathsf{unquote}_{A_i}$$

$$\begin{tikzcd}
\mathsf{neuts}_\Gamma(\Gamma) \arrow[r, "\mathsf{unquote}_\Gamma"] \arrow[rrr, bend left, "\mathsf{norm}^\Gamma_t"] \arrow[d, "\mathsf{s}[\![-]\!]"'] & \overline{\mathrm{e}}[\![\Gamma]\!](\Gamma) \arrow[r, "{\overline{\mathrm{e}}[\![\Gamma \vdash t : A]\!]}"] \arrow[d, "\nu_\Gamma"'] & \overline{\mathrm{e}}[\![A]\!](\Gamma) \arrow[r, "\mathsf{quote}_A"] \arrow[d, "\nu_A"] & \mathsf{norms}_A(\Gamma) \arrow[d, "\mathsf{s}[\![-]\!]"] \\
\mathbb{C}(s[\![\Gamma]\!], s[\![\Gamma]\!]) \arrow[r, equal, "\mathrm{id}_{s[\![\Gamma]\!]}"'] & \mathbb{C}(s[\![\Gamma]\!], s[\![\Gamma]\!]) \arrow[r, "{s[\![\Gamma \vdash t : A]\!] \circ (-)}"'] & \mathbb{C}(s[\![\Gamma]\!], s[\![A]\!]) \arrow[r, equal] & \mathbb{C}(s[\![\Gamma]\!], s[\![A]\!])
\end{tikzcd}$$

Define $\mathsf{nf}(t) := \mathsf{norm}^\Gamma_t((\Gamma \vdash x_i : A_i)_{i=1,\dots,n})$.

# Normalising $(\Gamma \vdash t : A)$

$$\mathsf{neuts}_\Gamma := \prod_{(x_i : A_i) \in \Gamma} \mathsf{neuts}_{A_i}$$

$$\mathsf{unquote}_\Gamma := \prod_{(x_i : A_i) \in \Gamma} \mathsf{unquote}_{A_i}$$

$$\begin{tikzcd}
\mathsf{neuts}_\Gamma(\Gamma) \arrow[r, "\mathsf{unquote}_\Gamma"] \arrow[d, "\mathsf{s}\llbracket - \rrbracket"'] \arrow[rrr, bend left, "\mathsf{norm}_t^\Gamma"] & \overline{\mathrm{e}}\llbracket \Gamma \rrbracket(\Gamma) \arrow[r, "\overline{\mathrm{e}}\llbracket \Gamma \vdash t : A \rrbracket"] \arrow[d, "\nu_\Gamma"'] & \overline{\mathrm{e}}\llbracket A \rrbracket(\Gamma) \arrow[r, "\mathsf{quote}_A"] \arrow[d, "\nu_A"] & \mathsf{norms}_A(\Gamma) \arrow[d, "\mathsf{s}\llbracket - \rrbracket"] \\
\mathbb{C}(s\llbracket \Gamma \rrbracket, s\llbracket \Gamma \rrbracket) \arrow[r, equal] & \mathbb{C}(s\llbracket \Gamma \rrbracket, s\llbracket \Gamma \rrbracket) \arrow[r, "s\llbracket \Gamma \vdash t : A \rrbracket \circ (-)"'] & \mathbb{C}(s\llbracket \Gamma \rrbracket, s\llbracket A \rrbracket) \arrow[r, equal, "s\llbracket \Gamma \vdash t : A \rrbracket"'] & \mathbb{C}(s\llbracket \Gamma \rrbracket, s\llbracket A \rrbracket)
\end{tikzcd}$$

Define $\mathsf{nf}(t) := \mathsf{norm}_t^\Gamma((\Gamma \vdash x_i : A_i)_{i=1,\ldots,n})$.

# Normalising $(\Gamma \vdash t : A)$

$$\mathsf{neuts}_\Gamma := \prod_{(x_i : A_i) \in \Gamma} \mathsf{neuts}_{A_i}$$

$$\mathsf{unquote}_\Gamma := \prod_{(x_i : A_i) \in \Gamma} \mathsf{unquote}_{A_i}$$

$$\begin{tikzcd}
\mathsf{neuts}_\Gamma(\Gamma) \arrow[r, "\mathsf{unquote}_\Gamma"] \arrow[d, "\mathsf{s}\llbracket - \rrbracket"'] \arrow[rrr, bend left, "\mathsf{norm}^\Gamma_t"] & \overline{\mathrm{e}}\llbracket \Gamma \rrbracket(\Gamma) \arrow[r, "\overline{\mathrm{e}}\llbracket \Gamma \vdash t : A \rrbracket"] \arrow[d, "\nu_\Gamma"'] & \overline{\mathrm{e}}\llbracket A \rrbracket(\Gamma) \arrow[r, "\mathsf{quote}_A"] \arrow[d, "\nu_A"] & \mathsf{norms}_A(\Gamma) \arrow[d, "\mathsf{s}\llbracket - \rrbracket"] & \mathsf{nf}(t) \\
\mathbb{C}(s\llbracket \Gamma \rrbracket, s\llbracket \Gamma \rrbracket) \arrow[r, equal] & \mathbb{C}(s\llbracket \Gamma \rrbracket, s\llbracket \Gamma \rrbracket) \arrow[r, "s\llbracket \Gamma \vdash t : A \rrbracket \circ (-)"'] & \mathbb{C}(s\llbracket \Gamma \rrbracket, s\llbracket A \rrbracket) \arrow[r, equal] & \mathbb{C}(s\llbracket \Gamma \rrbracket, s\llbracket A \rrbracket) \\
 & & s\llbracket \Gamma \vdash t : A \rrbracket & 
\end{tikzcd}$$

Define $\mathsf{nf}(t) := \mathsf{norm}^\Gamma_t((\Gamma \vdash x_i : A_i)_{i=1,\ldots,n})$.

# Normalising $(\Gamma \vdash t : A)$

$$\mathsf{neuts}_\Gamma := \textstyle\prod_{(x_i : A_i) \in \Gamma} \mathsf{neuts}_{A_i}$$

$$\mathsf{unquote}_\Gamma := \textstyle\prod_{(x_i : A_i) \in \Gamma} \mathsf{unquote}_{A_i}$$

$$\begin{tikzcd}
 & & & & \mathsf{nf}(t) \\
\mathsf{neuts}_\Gamma(\Gamma) \arrow[r, "\mathsf{unquote}_\Gamma"] \arrow[d, "\mathsf{s}\llbracket - \rrbracket"'] \arrow[rrr, bend left, "\mathsf{norm}^\Gamma_t"] & \overline{\mathrm{e}}\llbracket \Gamma \rrbracket(\Gamma) \arrow[r, "\overline{\mathrm{e}}\llbracket \Gamma \vdash t : A \rrbracket"] \arrow[d, "\nu_\Gamma"'] & \overline{\mathrm{e}}\llbracket A \rrbracket(\Gamma) \arrow[r, "\mathsf{quote}_A"] \arrow[d, "\nu_A"] & \mathsf{norms}_A(\Gamma) \arrow[d, "\mathsf{s}\llbracket - \rrbracket"] & \\
\mathbb{C}(s\llbracket \Gamma \rrbracket, s\llbracket \Gamma \rrbracket) \arrow[r, equal] & \mathbb{C}(s\llbracket \Gamma \rrbracket, s\llbracket \Gamma \rrbracket) \arrow[r, "s\llbracket \Gamma \vdash t : A \rrbracket \circ (-)"'] & \mathbb{C}(s\llbracket \Gamma \rrbracket, s\llbracket A \rrbracket) \arrow[r, equal] & \mathbb{C}(s\llbracket \Gamma \rrbracket, s\llbracket A \rrbracket) & \\
 & & & s\llbracket \Gamma \vdash t : A \rrbracket &
\end{tikzcd}$$

Define $\mathsf{nf}(t) := \mathsf{norm}^\Gamma_t((\Gamma \vdash x_i : A_i)_{i=1,\ldots,n})$.

Since $s\llbracket \Gamma \vdash \mathsf{nf}(t) : A \rrbracket = s\llbracket \Gamma \vdash t : A \rrbracket$ in every model, $\mathsf{nf}(t) =_{\beta\eta} t$.

# Proving coherence (this paper)

$$\begin{tikzcd}
\text{STLC neutrals inside } \Lambda_{\mathrm{ps}}^{\times,\to} \arrow[d] & & & \text{STLC normals inside } \Lambda_{\mathrm{ps}}^{\times,\to} \arrow[d] \\
\mathsf{neuts}_{\Gamma}^{\mathrm{ps}} \arrow[r, "\mathsf{unquote}_{\Gamma}"] \arrow[d, "\mathsf{s}\llbracket - \rrbracket"'] \arrow[dr, phantom, "\overset{\overline{u}_\Gamma}{\cong}"] & \overline{\mathrm{e}}\llbracket \Gamma \rrbracket \arrow[r, "\overline{\mathrm{e}}\llbracket \Gamma \vdash t : A \rrbracket"] \arrow[d] \arrow[dr, phantom, "\overset{\mathsf{w}_t}{\cong}"] & \overline{\mathrm{e}}\llbracket A \rrbracket \arrow[r, "\mathsf{quote}_A"] \arrow[d] \arrow[dr, phantom, "\overset{\overline{q}_A}{\cong}"] & \mathsf{norms}_{A}^{\mathrm{ps}} \arrow[d, "s\llbracket - \rrbracket"] \\
\mathcal{B}(s\llbracket - \rrbracket, s\llbracket \Gamma \rrbracket) \arrow[r, equal] & \mathcal{B}(s\llbracket - \rrbracket, s\llbracket \Gamma \rrbracket) \arrow[r, "s\llbracket \Gamma \vdash t : A \rrbracket \circ (-)"'] & \mathcal{B}(s\llbracket - \rrbracket, s\llbracket A \rrbracket) \arrow[r, equal] & \mathcal{B}(s\llbracket - \rrbracket, s\llbracket A \rrbracket)
\end{tikzcd}$$

# Proving coherence (this paper)

STLC neutrals
inside $\Lambda^{\times,\to}_{\mathrm{ps}}$

STLC normals
inside $\Lambda^{\times,\to}_{\mathrm{ps}}$

$$\begin{tikzcd}
\mathsf{neuts}^{\mathrm{ps}}_{\Gamma} \arrow[r, "\mathsf{unquote}_{\Gamma}"] \arrow[d, "\mathsf{s}[\![-]\!]"'] \arrow[rrr, bend left, "\mathsf{norm}_t"] \arrow[dr, phantom, "\overset{\overline{u}_{\Gamma}}{\cong}"] & \bar{\mathrm{e}}[\![\Gamma]\!] \arrow[r, "\bar{\mathrm{e}}[\![\Gamma \vdash t : A]\!]"] \arrow[d] \arrow[dr, phantom, "\overset{\mathsf{w}_t}{\cong}"] & \bar{\mathrm{e}}[\![A]\!] \arrow[r, "\mathsf{quote}_A"] \arrow[d] \arrow[dr, phantom, "\overset{\overline{q}_A}{\cong}"] & \mathsf{norms}^{\mathrm{ps}}_{A} \arrow[d, "s[\![-]\!]"] \\
\mathcal{B}(s[\![-]\!], s[\![\Gamma]\!]) \arrow[r, equal] & \mathcal{B}(s[\![-]\!], s[\![\Gamma]\!]) \arrow[r, "s[\![\Gamma \vdash t : A]\!] \circ (-)"'] & \mathcal{B}(s[\![-]\!], s[\![A]\!]) \arrow[r, equal] & \mathcal{B}(s[\![-]\!], s[\![A]\!])
\end{tikzcd}$$

# Proving coherence (this paper)

$$\begin{tikzcd}
\text{STLC neutrals inside } \Lambda_{\mathrm{ps}}^{\times,\to} \arrow[d] & & & \text{STLC normals inside } \Lambda_{\mathrm{ps}}^{\times,\to} \arrow[d] \\
\mathsf{neuts}^{\mathrm{ps}}_{\Gamma} \arrow[r, "\mathsf{unquote}_{\Gamma}"] \arrow[d, "\mathsf{s}\llbracket - \rrbracket"'] \arrow[rrr, bend left, "\mathsf{norm}_t"] \arrow[dr, phantom, "\overset{\overline{u}_{\Gamma}}{\cong}"] & \overline{\mathrm{e}}\llbracket \Gamma \rrbracket \arrow[r, "\overline{\mathrm{e}}\llbracket \Gamma \vdash t : A \rrbracket"] \arrow[d] \arrow[dr, phantom, "\overset{\mathrm{w}_t}{\cong}"] & \overline{\mathrm{e}}\llbracket A \rrbracket \arrow[r, "\mathsf{quote}_A"] \arrow[d] \arrow[dr, phantom, "\overset{\overline{q}_A}{\cong}"] & \mathsf{norms}^{\mathrm{ps}}_{A} \arrow[d, "s\llbracket - \rrbracket"] \\
\mathcal{B}(s\llbracket - \rrbracket, s\llbracket \Gamma \rrbracket) \arrow[r, equal] & \mathcal{B}(s\llbracket - \rrbracket, s\llbracket \Gamma \rrbracket) \arrow[r, "s\llbracket \Gamma \vdash t : A \rrbracket \circ (-)"'] & \mathcal{B}(s\llbracket - \rrbracket, s\llbracket A \rrbracket) \arrow[r, equal] & \mathcal{B}(s\llbracket - \rrbracket, s\llbracket A \rrbracket)
\end{tikzcd}$$

Instantiate at Γ, pass in variables $\rightsquigarrow$ $\boxed{s\llbracket t \rrbracket \cong s\llbracket \mathrm{nf}(t) \rrbracket}$

# Proving coherence (this paper)

STLC neutrals inside $\Lambda^{\times,\to}_{\mathrm{ps}}$

STLC normals inside $\Lambda^{\times,\to}_{\mathrm{ps}}$

$$\begin{tikzcd}
\mathrm{neuts}^{\mathrm{ps}}_{\Gamma} \arrow[r, "\mathsf{unquote}_{\Gamma}"] \arrow[d, "s\llbracket - \rrbracket"'] \arrow[rrr, bend left, "\mathsf{norm}_t"] \arrow[dr, phantom, "\overset{\overline{u}_{\Gamma}}{\cong}"] & \overline{\mathrm{e}}\llbracket \Gamma \rrbracket \arrow[r, "\overline{\mathrm{e}}\llbracket \Gamma \vdash t : A \rrbracket"] \arrow[d] \arrow[dr, phantom, "\overset{\mathsf{w}_t}{\cong}"] & \overline{\mathrm{e}}\llbracket A \rrbracket \arrow[r, "\mathsf{quote}_A"] \arrow[d] \arrow[dr, phantom, "\overset{\overline{q}_A}{\cong}"] & \mathrm{norms}^{\mathrm{ps}}_{A} \arrow[d, "s\llbracket - \rrbracket"] \\
\mathcal{B}(s\llbracket - \rrbracket, s\llbracket \Gamma \rrbracket) \arrow[r, equal] & \mathcal{B}(s\llbracket - \rrbracket, s\llbracket \Gamma \rrbracket) \arrow[r, "s\llbracket \Gamma \vdash t : A \rrbracket \circ (-)"'] & \mathcal{B}(s\llbracket - \rrbracket, s\llbracket A \rrbracket) \arrow[r, equal] & \mathcal{B}(s\llbracket - \rrbracket, s\llbracket A \rrbracket)
\end{tikzcd}$$

Instantiate at Γ, pass in variables $\rightsquigarrow$ $\boxed{s\llbracket t \rrbracket \cong s\llbracket \mathsf{nf}(t) \rrbracket}$

depends on $t$

$$s\llbracket t \rrbracket \xRightarrow{\cong} s\llbracket \mathsf{nf}(t) \rrbracket$$

$$s\llbracket t' \rrbracket \xRightarrow[\cong]{} s\llbracket \mathsf{nf}(t') \rrbracket$$

depends on $t'$

# Proving coherence (this paper)

$$\begin{tikzcd}[column sep=large]
\text{STLC neutrals inside } \Lambda_{\mathrm{ps}}^{\times,\to} \arrow[d] & & & \text{STLC normals inside } \Lambda_{\mathrm{ps}}^{\times,\to} \arrow[d] \\
\mathsf{neuts}^{\mathrm{ps}}_{\Gamma} \arrow[r, "\mathsf{unquote}_{\Gamma}"] \arrow[rrr, bend left, "\mathsf{norm}_t"] \arrow[d, "\mathsf{s}\llbracket - \rrbracket"'] \arrow[dr, phantom, "\overset{\overline{u}_{\Gamma}}{\cong}"] & \overline{\mathrm{e}}\llbracket \Gamma \rrbracket \arrow[r, "\overline{\mathrm{e}}\llbracket \Gamma \vdash t : A \rrbracket"] \arrow[d] \arrow[dr, phantom, "\overset{\mathsf{w}_t}{\cong}"] & \overline{\mathrm{e}}\llbracket A \rrbracket \arrow[r, "\mathsf{quote}_A"] \arrow[d] \arrow[dr, phantom, "\overset{\overline{q}_A}{\cong}"] & \mathsf{norms}^{\mathrm{ps}}_{A} \arrow[d, "s\llbracket - \rrbracket"] \\
\mathcal{B}(s\llbracket - \rrbracket, s\llbracket \Gamma \rrbracket) \arrow[r, equal] & \mathcal{B}(s\llbracket - \rrbracket, s\llbracket \Gamma \rrbracket) \arrow[r, "s\llbracket \Gamma \vdash t : A \rrbracket \circ (-)"'] & \mathcal{B}(s\llbracket - \rrbracket, s\llbracket A \rrbracket) \arrow[r, equal] & \mathcal{B}(s\llbracket - \rrbracket, s\llbracket A \rrbracket)
\end{tikzcd}$$

Instantiate at Γ, pass in variables $\rightsquigarrow$ $\boxed{s\llbracket t \rrbracket \cong s\llbracket \mathsf{nf}(t) \rrbracket}$

depends on $t$

$$\begin{tikzcd}
s\llbracket t \rrbracket \arrow[r, Rightarrow, "\cong"] \arrow[d, Rightarrow, "s\llbracket \tau \rrbracket"'] & s\llbracket \mathsf{nf}(t) \rrbracket \arrow[d, Rightarrow] \\
s\llbracket t' \rrbracket \arrow[r, Rightarrow, "\cong"'] & s\llbracket \mathsf{nf}(t') \rrbracket
\end{tikzcd}$$

depends on $t'$

condition on 2-cells
in glueing bicat

# Proving coherence (this paper)

$$\begin{tikzcd}
\text{STLC neutrals inside } \Lambda^{\times,\to}_{\mathrm{ps}} \arrow[d] & & & \text{STLC normals inside } \Lambda^{\times,\to}_{\mathrm{ps}} \arrow[d] \\
\mathrm{neuts}^{\mathrm{ps}}_{\Gamma} \arrow[r, "\mathrm{unquote}_{\Gamma}"] \arrow[rrr, bend left, "\mathrm{norm}_t"] \arrow[d, "s\llbracket - \rrbracket"'] \arrow[dr, phantom, "\overset{\overline{u}_{\Gamma}}{\cong}"] & \overline{\mathrm{e}}\llbracket \Gamma \rrbracket \arrow[r, "\overline{\mathrm{e}}\llbracket \Gamma \vdash t : A \rrbracket"] \arrow[d] \arrow[dr, phantom, "\overset{\mathrm{w}_t}{\cong}"] & \overline{\mathrm{e}}\llbracket A \rrbracket \arrow[r, "\mathrm{quote}_A"] \arrow[d] \arrow[dr, phantom, "\overset{\overline{q}_A}{\cong}"] & \mathrm{norms}^{\mathrm{ps}}_{A} \arrow[d, "s\llbracket - \rrbracket"] \\
\mathcal{B}(s\llbracket - \rrbracket, s\llbracket \Gamma \rrbracket) \arrow[r, equal] & \mathcal{B}(s\llbracket - \rrbracket, s\llbracket \Gamma \rrbracket) \arrow[r, "s\llbracket \Gamma \vdash t : A \rrbracket \circ (-)"'] & \mathcal{B}(s\llbracket - \rrbracket, s\llbracket A \rrbracket) \arrow[r, equal] & \mathcal{B}(s\llbracket - \rrbracket, s\llbracket A \rrbracket)
\end{tikzcd}$$

Instantiate at Γ, pass in variables $\rightsquigarrow$ $\boxed{s\llbracket t \rrbracket \cong s\llbracket \mathrm{nf}(t) \rrbracket}$

$$\begin{tikzcd}
s\llbracket t \rrbracket \arrow[r, Rightarrow, "\overset{\text{depends on } t}{\cong}"] \arrow[d, Rightarrow, "s\llbracket \tau \rrbracket"'] & s\llbracket \mathrm{nf}(t) \rrbracket \arrow[d, equal] \\
s\llbracket t' \rrbracket \arrow[r, Rightarrow, "\underset{\text{depends on } t'}{\cong}"'] & s\llbracket \mathrm{nf}(t') \rrbracket
\end{tikzcd}$$

since $\mathrm{norms}^{\mathrm{ps}}_{A}$ is
is a discrete category

# Proving coherence (this paper)

STLC neutrals inside $\Lambda_{\mathrm{ps}}^{\times,\to}$

STLC normals inside $\Lambda_{\mathrm{ps}}^{\times,\to}$

$$\begin{tikzcd}
\mathrm{neuts}^{\mathrm{ps}}_{\Gamma} \arrow[r, "\mathrm{unquote}_{\Gamma}"] \arrow[d, "s[\![-]\!]"'] \arrow[rrr, bend left, "\mathrm{norm}_t"] \arrow[dr, phantom, "\overset{\overline{u}_{\Gamma}}{\cong}"] & \overline{\mathrm{e}}[\![\Gamma]\!] \arrow[r, "{\overline{\mathrm{e}}[\![\Gamma \vdash t : A]\!]}"] \arrow[d] \arrow[dr, phantom, "\overset{\mathrm{w}_t}{\cong}"] & \overline{\mathrm{e}}[\![A]\!] \arrow[r, "\mathrm{quote}_A"] \arrow[d] \arrow[dr, phantom, "\overset{\overline{q}_A}{\cong}"] & \mathrm{norms}^{\mathrm{ps}}_{A} \arrow[d, "s[\![-]\!]"] \\
\mathcal{B}(s[\![-]\!], s[\![\Gamma]\!]) \arrow[r, equal] & \mathcal{B}(s[\![-]\!], s[\![\Gamma]\!]) \arrow[r, "{s[\![\Gamma \vdash t : A]\!] \circ (-)}"'] & \mathcal{B}(s[\![-]\!], s[\![A]\!]) \arrow[r, equal] & \mathcal{B}(s[\![-]\!], s[\![A]\!])
\end{tikzcd}$$

Instantiate at Γ, pass in variables $\rightsquigarrow$ $\boxed{s[\![t]\!] \cong s[\![\mathrm{nf}(t)]\!]}$

$$\begin{tikzcd}
t \arrow[r, Rightarrow, "\cong"] \arrow[d, Rightarrow, "\tau"'] & \mathrm{nf}(t) \arrow[d, equal] \\
t' \arrow[r, Rightarrow, "\cong"'] & \mathrm{nf}(t')
\end{tikzcd}$$

syntactic model

# Proving coherence (this paper)

STLC neutrals inside $\Lambda^{\times,\to}_{\mathrm{ps}}$

STLC normals inside $\Lambda^{\times,\to}_{\mathrm{ps}}$

$$\begin{tikzcd}
\mathsf{neuts}^{\mathrm{ps}}_{\Gamma} \arrow[r, "\mathsf{unquote}_{\Gamma}"] \arrow[rrr, bend left, "\mathsf{norm}_t"] \arrow[d, "\mathsf{s}[\![-]\!]"'] \arrow[dr, phantom, "\overset{\overline{u}_{\Gamma}}{\cong}"] & \overline{\mathrm{e}}[\![\Gamma]\!] \arrow[r, "\overline{\mathrm{e}}[\![\Gamma \vdash t : A]\!]"] \arrow[d] \arrow[dr, phantom, "\overset{\mathsf{w}_t}{\cong}"] & \overline{\mathrm{e}}[\![A]\!] \arrow[r, "\mathsf{quote}_A"] \arrow[d] \arrow[dr, phantom, "\overset{\overline{q}_A}{\cong}"] & \mathsf{norms}^{\mathrm{ps}}_{A} \arrow[d, "\mathsf{s}[\![-]\!]"] \\
\mathcal{B}(s[\![-]\!], s[\![\Gamma]\!]) \arrow[r, equal] & \mathcal{B}(s[\![-]\!], s[\![\Gamma]\!]) \arrow[r, "s[\![\Gamma \vdash t : A]\!] \circ (-)"'] & \mathcal{B}(s[\![-]\!], s[\![A]\!]) \arrow[r, equal] & \mathcal{B}(s[\![-]\!], s[\![A]\!])
\end{tikzcd}$$

Instantiate at Γ, pass in variables ⇝ $\boxed{s[\![t]\!] \cong s[\![\mathsf{nf}(t)]\!]}$

$$\begin{tikzcd}
t \arrow[r, Rightarrow, "\cong"] \arrow[d, Rightarrow, "\tau"'] & \mathsf{nf}(t) \arrow[d, equal] \\
t' \arrow[r, Rightarrow, "\cong"'] & \mathsf{nf}(t')
\end{tikzcd}$$

syntactic model

⇝ if $\tau : t \Rightarrow t'$ exists, it's unique (modulo $\equiv$)

# Coherence for cc-bicategorical structure

## Coherence for cc-bicategorical structure

$\rightsquigarrow$ cc-Bicategories are coherent

# Coherence for cc-bicategorical structure

$\rightsquigarrow$ cc-Bicategories are coherent

$\rightsquigarrow$ "Suffices" to work in a CCC

1. prove result in STLC
2. $\beta\eta$-equalities $\rightsquigarrow$ structural 2-cells
3. axioms guaranteed

## Coherence for cc-bicategorical structure

$\rightsquigarrow$ cc-Bicategories are coherent

$\rightsquigarrow$ "Suffices" to work in a CCC

1. prove result in STLC
2. $\beta\eta$-equalities $\rightsquigarrow$ structural 2-cells
3. axioms guaranteed

## Coherence via normalisation-by-evaluation

$\rightsquigarrow$ Coherence as a normalisation property

$\rightsquigarrow$ Normalisation proven semantically

$\rightsquigarrow$ If you work with universal properties enough

. . . higher-categorical proof builds on categorical proof

Future work: extend this to other structures

e.g. sums, dependent products, notions of initial algebra. . .

## Further reading

- *A type theory for cartesian closed bicategories*, LICS 2019
- *Relative full completeness for bicategorical cartesian closed structure*, FoSSaCS 2020
- *Cartesian closed bicategories: type theory and coherence*, PhD thesis, 2020