Lax-Idempotent 2-Monads, Degrees of Relatedness, and Multilevel Type Theory

Andreas Nuyts

KU Leuven, Belgium

TYPES '23
Valencia, Spain
June 12, 2023

Degrees of Relatedness (RelDTT)

Nuyts and Devriese (2018) @ LICS

Degrees of Relatedness: Overview

- Parametricity is about **relations**,
- Equip types with **multiple**, **proof-relevant relations** $s \frown_i t$ indexed by **degree** i:
  - Just **one** for **small types** (Bool, $\mathbb{N} \to \mathbb{N}$, ...),
  - **More** for **larger** types ($U_0 \to U_0$, Grp, ...).
  - Proofs called ***i*-edges**.
- Describe **function behaviour** by saying how functions **influence degree** of relatedness,
- This explains
  - **parametricity**
  - **ad hoc polymorphism**
  - **. irrelevance**
  - **.. shape-irrelevance**
  - aspects of **algebra, unions, intersections, Prop, ...**

Degrees of Relatedness: Overview

- Parametricity is about **relations**,
- Equip types with multiple, **proof-relevant relations** $s \frown_i t$ indexed by **degree** i:
  - Just **one** for **small types** (Bool, $\mathbb{N} \to \mathbb{N}, \ldots$),
  - **More** for **larger** types ($U_0 \to U_0$, Grp, $\ldots$).
  - Proofs called ***i*-edges**.
- Describe **function behaviour** by saying how functions **influence degree** of relatedness,
- This explains
  - **parametricity**
  - **ad hoc polymorphism**
  - **. irrelevance**
  - **.. shape-irrelevance**
  - aspects of **algebra, unions, intersections, Prop, . . .**

Degrees of Relatedness: Overview

- Parametricity is about **relations**,
- Equip types with **multiple, proof-relevant relations** $s \frown_i t$ indexed by **degree** i:
  - Just **one** for **small types** (Bool, $\mathbb{N} \to \mathbb{N}, \ldots$),
  - **More** for **larger** types ($\mathsf{U}_0 \to \mathsf{U}_0$, Grp, $\ldots$).
  - Proofs called *i*-**edges**.
- Describe **function behaviour** by saying how functions **influence degree** of relatedness,
- This explains
  - **parametricity**
  - **ad hoc polymorphism**
  - **. irrelevance**
  - **.. shape-irrelevance**
  - aspects of **algebra, unions, intersections, Prop, . . .**

Degrees of Relatedness: Overview

- Parametricity is about **relations**,
- Equip types with **multiple, proof-relevant relations** $s \frown_i t$ indexed by **degree** i:
  - Just **one** for **small types** (Bool, $\mathbb{N} \to \mathbb{N}$, ...),
  - **More** for **larger** types ($U_0 \to U_0$, Grp, ...).
  - Proofs called i-**edges**.
- Describe **function behaviour** by saying how functions **influence degree** of relatedness,
- This explains
  - **parametricity**
  - **ad hoc polymorphism**
  - **. irrelevance**
  - **.. shape-irrelevance**
  - aspects of **algebra, unions, intersections, Prop, ...**

Degrees of Relatedness: Overview

- Parametricity is about **relations**,
- Equip types with **multiple, proof-relevant relations** $s \frown_i t$ indexed by **degree** i:
  - Just **one** for **small types** (Bool, $\mathbb{N} \to \mathbb{N}$, ...),
  - **More** for **larger** types ($U_0 \to U_0$, Grp, ...).
  - Proofs called ***i*-edges**.
- Describe **function behaviour** by saying how functions **influence degree** of relatedness,
- This explains
  - **parametricity**
  - **ad hoc polymorphism**
  - **. irrelevance**
  - **.. shape-irrelevance**
  - aspects of **algebra, unions, intersections, Prop, ...**

Degrees of Relatedness: Overview

- Parametricity is about **relations**,
- Equip types with **multiple, proof-relevant relations** $s \frown_i t$ indexed by **degree** i:
  - Just **one** for **small types** (Bool, $\mathbb{N} \to \mathbb{N}, \ldots$),
  - **More** for **larger** types ($\mathrm{U}_0 \to \mathrm{U}_0$, Grp, $\ldots$).
  - Proofs called ***i*-edges**.
- Describe **function behaviour** by saying how functions **influence degree** of relatedness,
- This explains
  - **parametricity**
  - **ad hoc polymorphism**
  - **. irrelevance**
  - **.. shape-irrelevance**
  - aspects of **algebra, unions, intersections, Prop, . . .**

Degrees of Relatedness: Overview

- Parametricity is about **relations**,
- Equip types with **multiple, proof-relevant relations** $s \frown_i t$ indexed by **degree** i:
  - Just **one** for **small types** (Bool, $\mathbb{N} \to \mathbb{N}$, ...),
  - **More** for **larger** types ($U_0 \to U_0$, Grp, ...).
  - Proofs called ***i*-edges**.
- Describe **function behaviour** by saying how functions **influence degree** of relatedness,
- This explains
  - **parametricity**
  - **ad hoc polymorphism**
  - **. irrelevance**
  - **.. shape-irrelevance**
  - aspects of **algebra, unions, intersections, Prop, ...**

Degrees of Relatedness: Overview

- Parametricity is about **relations**,
- Equip types with **multiple, proof-relevant relations** $s \frown_i t$ indexed by **degree** i:
  - Just **one** for **small types** (Bool, $\mathbb{N} \to \mathbb{N}$, ...),
  - **More** for **larger** types ($U_0 \to U_0$, Grp, ...).
  - Proofs called ***i*-edges**.
- Describe **function behaviour** by saying how functions **influence degree** of relatedness,
- This explains
  - **parametricity**
  - **ad hoc polymorphism**
  - **. irrelevance**
  - **.. shape-irrelevance**
  - aspects of **algebra, unions, intersections, Prop, ...**

Degrees of Relatedness: Four Laws

- **Reflexivity:** $(a : A) \frown_i^A (a : A)$
  (Semantically, prop. eq. = def. eq.)
- **Degradation:** $((a : A) \frown_i^R (b : B)) \to ((a : A) \frown_{i+1}^R (b : B))$
- **Dependency:** $(a : A) \frown_i^R (b : B)$ **presumes** $R : A \frown_{i+1}^{\mathrm{U}} B$
- **Identity extension:** $(a : A) \frown_0^A (b : A)$ means $a = b : A$.
  $\leadsto$ heterogeneous $\frown_0$ serves as heterogeneous equality.

Degrees of Relatedness: Four Laws

- **Reflexivity:** $(a = b : A) \to (a : A) \frown_i^A (b : A)$
  (Semantically, prop. eq. = def. eq.)
- **Degradation:** $((a : A) \frown_i^R (b : B)) \to ((a : A) \frown_{i+1}^R (b : B))$
- **Dependency:** $(a : A) \frown_i^R (b : B)$ **presumes** $R : A \frown_{i+1}^{\mathsf{U}} B$
- **Identity extension:** $(a : A) \frown_0^A (b : A)$ means $a = b : A$.
  $\leadsto$ heterogeneous $\frown_0$ serves as heterogeneous equality.

Degrees of Relatedness: Four Laws

- **Reflexivity:** $(a = b : A) \to (a : A) \frown_i^A (b : A)$
  (Semantically, prop. eq. = def. eq.)
- **Degradation:** $((a : A) \frown_i^R (b : B)) \to ((a : A) \frown_{i+1}^R (b : B))$
- **Dependency:** $(a : A) \frown_i^R (b : B)$ **presumes** $R : A \frown_{i+1}^{\mathsf{U}} B$
- **Identity extension:** $(a : A) \frown_0^A (b : A)$ means $a = b : A$.
  $\rightsquigarrow$ heterogeneous $\frown_0$ serves as heterogeneous equality.

Degrees of Relatedness: Four Laws

- **Reflexivity:** $(a = b : A) \to (a : A) \frown_i^A (b : A)$
  (Semantically, prop. eq. = def. eq.)
- **Degradation:** $((a : A) \frown_i^R (b : B)) \to ((a : A) \frown_{i+1}^R (b : B))$
- **Dependency:** $(a : A) \frown_i^R (b : B)$ **presumes** $R : A \frown_{i+1}^{\mathsf{U}} B$
- **Identity extension:** $(a : A) \frown_0^A (b : A)$ means $a = b : A$.
  $\leadsto$ heterogeneous $\frown_0$ serves as heterogeneous equality.

Degrees of Relatedness: Four Laws

- **Reflexivity:** $(a = b : A) \to (a : A) \frown_i^A (b : A)$
  (Semantically, prop. eq. = def. eq.)
- **Degradation:** $((a : A) \frown_i^R (b : B)) \to ((a : A) \frown_{i+1}^R (b : B))$
- **Dependency:** $(a : A) \frown_i^R (b : B)$ **presumes** $R : A \frown_{i+1}^{\mathrm{U}} B$
- **Identity extension:** $(a : A) \frown_0^A (b : A)$ means $a = b : A$.
  $\leadsto$ heterogeneous $\frown_0$ serves as heterogeneous equality.

Understanding degrees

$(a : A) \frown_0^A (b : A)$	Equality.
$(a : A) \frown_0^R (b : B)$	Heterogeneous equality along …
$R : (A : \mathsf{U}_0) \frown_1^{\mathsf{U}_0} (B : \mathsf{U}_0)$	Any relation R.
$P : (G : \mathsf{Grp}) \frown_1^{\mathsf{Grp}} (H : \mathsf{Grp})$	Any logical/algebraic relation P.
$Q : (G : \mathsf{Grp}) \frown_1^V (M : \mathsf{Monoid})$	Any logical/algebraic relation Q along …
$V : (\mathsf{Grp} : \mathsf{U}_1) \frown_2^{\mathsf{U}_1} (\mathsf{Monoid} : \mathsf{U}_1)$	V could specify that Q must respect e **and** $*$ (but it could ask Q to be anything).

Understanding degrees

$(a : A) \frown_0^A (b : A)$	Equality.
$(a : A) \frown_0^R (b : B)$	Heterogeneous equality along ...
$R : (A : \mathsf{U}_0) \frown_1^{\mathsf{U}_0} (B : \mathsf{U}_0)$	Any relation R.
$P : (G : \mathsf{Grp}) \frown_1^{\mathsf{Grp}} (H : \mathsf{Grp})$	Any logical/algebraic relation P.
$Q : (G : \mathsf{Grp}) \frown_1^V (M : \mathsf{Monoid})$	Any logical/algebraic relation Q along ...
$V : (\mathsf{Grp} : \mathsf{U}_1) \frown_2^{\mathsf{U}_1} (\mathsf{Monoid} : \mathsf{U}_1)$	V could specify that Q must respect e **and** $*$ (but it could ask Q to be anything).

Understanding degrees

$(a : A) \frown_0^A (b : A)$	Equality.
$(a : A) \frown_0^R (b : B)$	Heterogeneous equality along ...
$R : (A : U_0) \frown_1^{U_0} (B : U_0)$	Any relation R.
$P : (G : \mathrm{Grp}) \frown_1^{\mathrm{Grp}} (H : \mathrm{Grp})$	Any logical/algebraic relation P.
$Q : (G : \mathrm{Grp}) \frown_1^V (M : \mathrm{Monoid})$	Any logical/algebraic relation Q along ...
$V : (\mathrm{Grp} : U_1) \frown_2^{U_1} (\mathrm{Monoid} : U_1)$	V could specify that Q must respect e **and** $*$ (but it could ask Q to be anything).

Understanding degrees

$(a : A) \frown_0^A (b : A)$ Equality.

$(a : A) \frown_0^R (b : B)$ Heterogeneous equality along …

$R : (A : \mathsf{U}_0) \frown_1^{\mathsf{U}_0} (B : \mathsf{U}_0)$ Any relation R.

$P : (G : \mathsf{Grp}) \frown_1^{\mathsf{Grp}} (H : \mathsf{Grp})$ Any logical/algebraic relation P.

$Q : (G : \mathsf{Grp}) \frown_1^V (M : \mathsf{Monoid})$ Any logical/algebraic relation Q along …

$V : (\mathsf{Grp} : \mathsf{U}_1) \frown_2^{\mathsf{U}_1} (\mathsf{Monoid} : \mathsf{U}_1)$ V could specify that Q must respect e **and** $*$ (but it could ask Q to be anything).

Understanding degrees

$(a : A) \frown_0^A (b : A)$ — Equality.

$(a : A) \frown_0^R (b : B)$ — Heterogeneous equality along ...

$R : (A : \mathsf{U}_0) \frown_1^{\mathsf{U}_0} (B : \mathsf{U}_0)$ — Any relation R.

$P : (G : \mathsf{Grp}) \frown_1^{\mathsf{Grp}} (H : \mathsf{Grp})$ — Any logical/algebraic relation P.

$Q : (G : \mathsf{Grp}) \frown_1^V (M : \mathsf{Monoid})$ — Any logical/algebraic relation Q along ...

$V : (\mathsf{Grp} : \mathsf{U}_1) \frown_2^{\mathsf{U}_1} (\mathsf{Monoid} : \mathsf{U}_1)$ — V could specify that Q must respect e **and** $*$ (but it could ask Q to be anything).

Understanding degrees

$(a : A) \frown_0^A (b : A)$	Equality.
$(a : A) \frown_0^R (b : B)$	Heterogeneous equality along ...
$R : (A : U_0) \frown_1^{U_0} (B : U_0)$	Any relation R.
$P : (G : \mathrm{Grp}) \frown_1^{\mathrm{Grp}} (H : \mathrm{Grp})$	Any logical/algebraic relation P.
$Q : (G : \mathrm{Grp}) \frown_1^V (M : \mathrm{Monoid})$	Any logical/algebraic relation Q along ...
$V : (\mathrm{Grp} : U_1) \frown_2^{U_1} (\mathrm{Monoid} : U_1)$	V could specify that Q must respect e **and** $*$ (but it could ask Q to be anything).

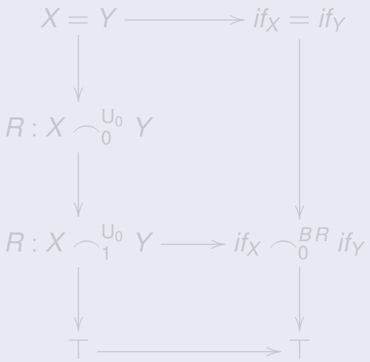
Understanding degrees

$(a : A) \frown_0^A (b : A)$	Equality.
$(a : A) \frown_0^R (b : B)$	Heterogeneous equality along …
$R : (A : \mathsf{U}_0) \frown_1^{\mathsf{U}_0} (B : \mathsf{U}_0)$	Any relation R.
$P : (G : \mathsf{Grp}) \frown_1^{\mathsf{Grp}} (H : \mathsf{Grp})$	Any logical/algebraic relation P.
$Q : (G : \mathsf{Grp}) \frown_1^V (M : \mathsf{Monoid})$	Any logical/algebraic relation Q along …
$V : (\mathsf{Grp} : \mathsf{U}_1) \frown_2^{\mathsf{U}_1} (\mathsf{Monoid} : \mathsf{U}_1)$	V could specify that Q must respect e **and** $*$ (but it could ask Q to be anything).

Understanding modalities (2)

par : types $\to$ values

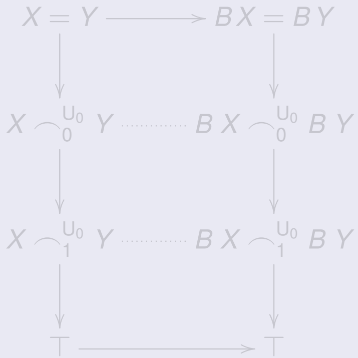
$$\mathit{if} : (\mathbf{par} \shortmid X : \mathsf{U}_0) \to B\, X$$



con : types $\to$ types

$$B : \mathsf{U}_0 \to \mathsf{U}_0$$

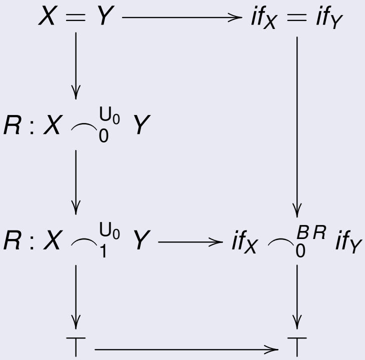
$$B\, X = \mathsf{Bool} \to X \to X \to X$$



Understanding modalities (2)

par : types → values

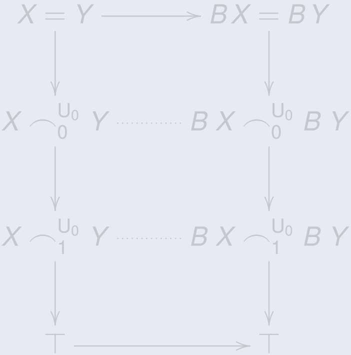
$$\mathit{if} : (\mathbf{par} \shortmid X : \mathsf{U}_0) \to B\,X$$



con : types → types

$$B : \mathsf{U}_0 \to \mathsf{U}_0$$

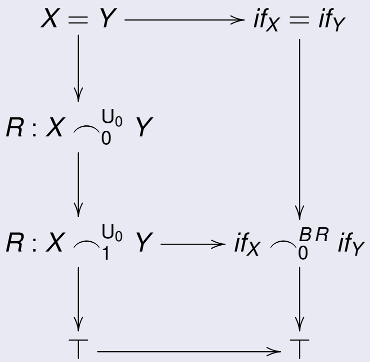
$$B\,X = \mathsf{Bool} \to X \to X \to X$$



Understanding modalities (2)

par : types $\to$ values

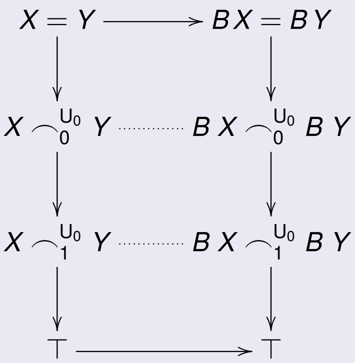
$$\mathit{if} : (\mathbf{par} \mid X : \mathsf{U}_0) \to B\,X$$



con : types $\to$ types

$$B : \mathsf{U}_0 \to \mathsf{U}_0$$

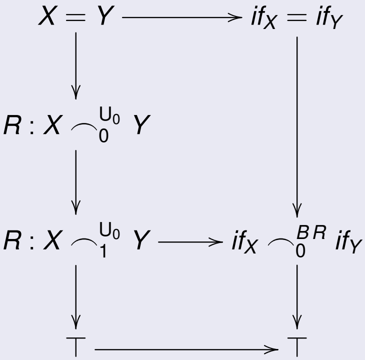
$$B\,X = \mathsf{Bool} \to X \to X \to X$$



Understanding modalities (2)

par : types → values

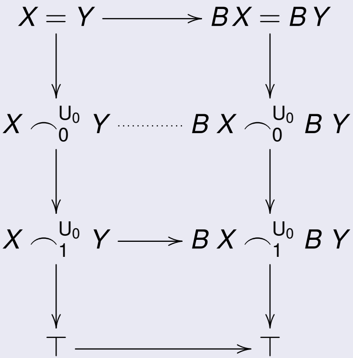
$$if : (\mathbf{par} \mid X : \mathrm{U}_0) \to B\,X$$



con : types → types

$$B : \mathrm{U}_0 \to \mathrm{U}_0$$

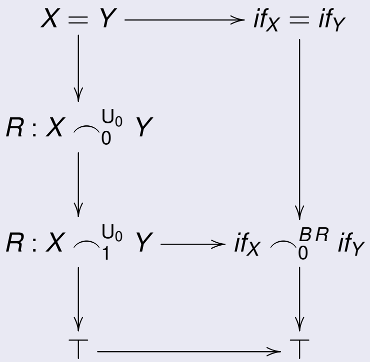
$$B\,X = \mathrm{Bool} \to X \to X \to X$$



Understanding modalities (2)

par : types → values

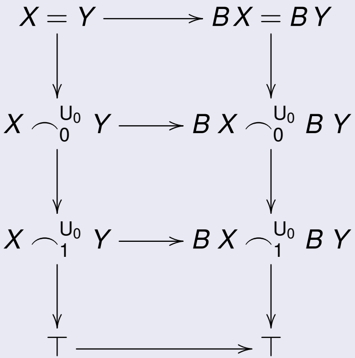
$$if : (\mathbf{par} \mid X : U_0) \to B\,X$$



con : types → types

$$B : U_0 \to U_0$$

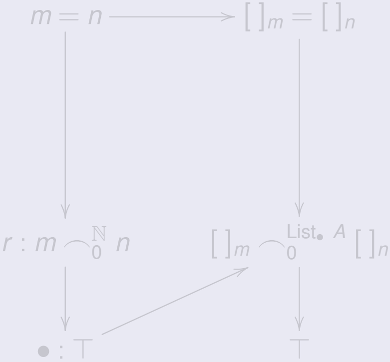
$$B\,X = \mathrm{Bool} \to X \to X \to X$$



Understanding modalities (1)

irr : values → values

$$[\,] : (\mathbf{irr} \mid n : \mathbb{N}) \to \mathrm{List}_n\, A$$

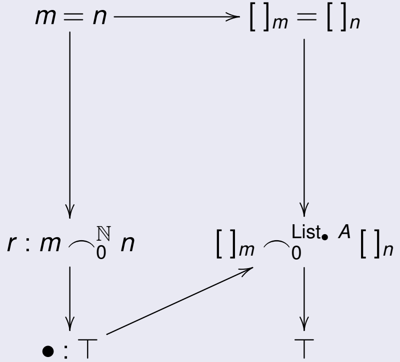


shi : values → types

$$\lambda n.\mathrm{List}_n\, A : (\mathbf{shi} \mid n : \mathbb{N}) \to \mathrm{U}_0$$

irr : values → values

$$[\,] : (\mathbf{irr} \mid n : \mathbb{N}) \to \mathrm{List}_n\ A$$



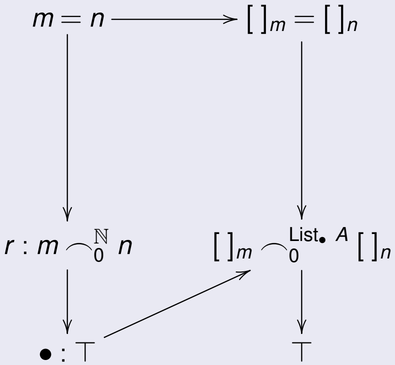
shi : values → types

$$\lambda n.\mathrm{List}_n\ A : (\mathbf{shi} \mid n : \mathbb{N}) \to \mathrm{U}_0$$

Understanding modalities (1)

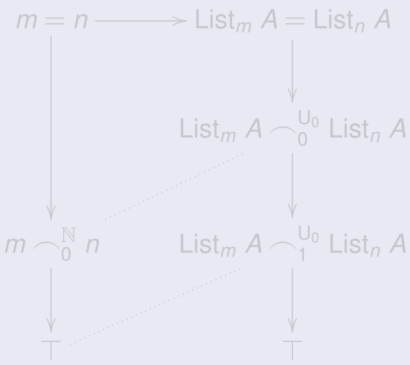
irr : values $\to$ values

$$[\,] : (\mathbf{irr} \mid n : \mathbb{N}) \to \mathrm{List}_n\ A$$



shi : values $\to$ types

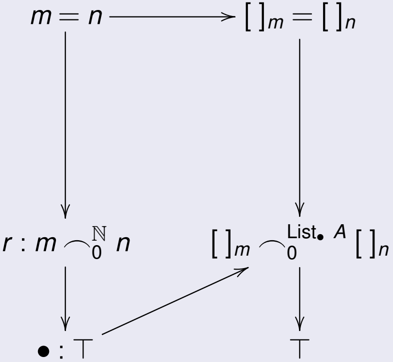
$$\lambda n.\mathrm{List}_n\ A : (\mathbf{shi} \mid n : \mathbb{N}) \to \mathrm{U}_0$$



Understanding modalities (1)

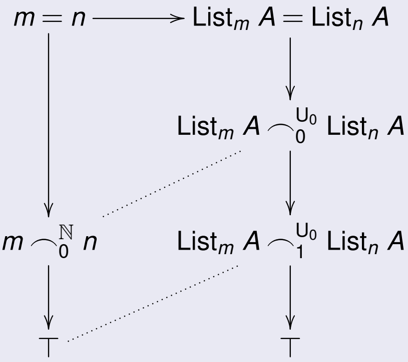
irr : values → values

$$[\,] : (\mathbf{irr} \mid n : \mathbb{N}) \to \mathrm{List}_n\, A$$



shi : values → types

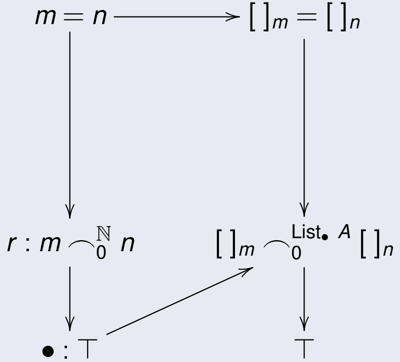
$$\lambda n.\mathrm{List}_n\, A : (\mathbf{shi} \mid n : \mathbb{N}) \to \mathrm{U}_0$$



Understanding modalities (1)

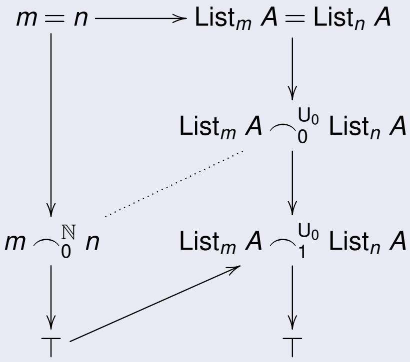
irr : values $\to$ values

$$[\,] : (\mathbf{irr} \mid n : \mathbb{N}) \to \mathrm{List}_n\, A$$



shi : values $\to$ types

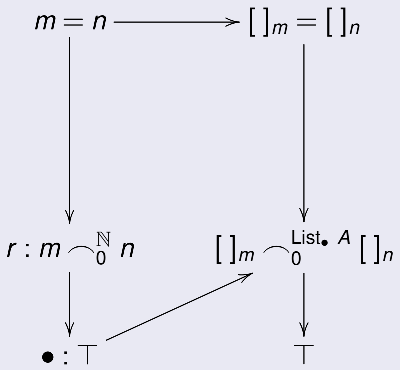
$$\lambda n.\mathrm{List}_n\, A : (\mathbf{shi} \mid n : \mathbb{N}) \to \mathrm{U}_0$$



Understanding modalities (1)

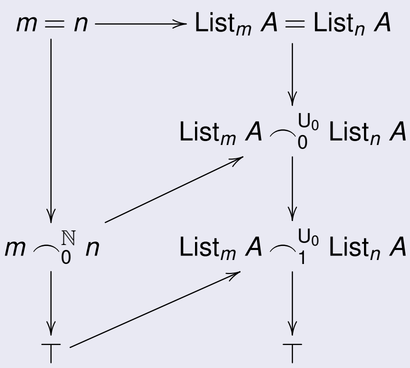
$\textbf{irr} : \text{values} \to \text{values}$

$$[\,] : (\textbf{irr} \mid n : \mathbb{N}) \to \text{List}_n\ A$$



$\textbf{shi} : \text{values} \to \text{types}$

$$\lambda n.\text{List}_n\ A : (\textbf{shi} \mid n : \mathbb{N}) \to \text{U}_0$$



The mode theory DoR

RelDTT can be built on an instance of MTT (Multimode Type Theory) with mode theory DoR:

- Modes are **depths** $p \in \mathbb{Z}_{\geq -1}$
- Modalities $\mu : p \to q$ are
  functions $\{0 \leq \ldots \leq q\} \to \{(=) \leq 0 \leq \ldots \leq p \leq \top\} : i \mapsto i \cdot \mu$
  where $f : (\mu \mid x : A) \to B(x)$ sends

$$(r : x \frown^{A}_{i \cdot \mu} y) \quad \to \quad f(x) \frown^{B(r)}_{i} f(y).$$

  Modal types:

$$\mathrm{mod}_\mu\, x \frown^{\langle \mu | A \rangle}_{i} \mathrm{mod}_\mu\, y \quad = \quad x \frown^{A}_{i \cdot \mu} y$$

- 2-cells are degree-wise inequalities.

Depth p is modelled in cubical sets with $p+1$ different dimension flavours.

The mode theory DoR

RelDTT can be built on an instance of MTT (Multimode Type Theory) with mode theory DoR:

- Modes are **depths** $p \in \mathbb{Z}_{\geq -1}$
- Modalities $\mu : p \to q$ are
  functions $\{0 \leq \ldots \leq q\} \to \{(=) \leq 0 \leq \ldots \leq p \leq \top\} : i \mapsto i \cdot \mu$
  where $f : (\mu \mid x : A) \to B(x)$ sends

$$(r : x \frown^{A}_{i \cdot \mu} y) \quad \to \quad f(x) \frown^{B(r)}_{i} f(y).$$

  Modal types:

$$\mathrm{mod}_{\mu}\, x \frown^{\langle \mu | A \rangle}_{i} \mathrm{mod}_{\mu}\, y \quad = \quad x \frown^{A}_{i \cdot \mu} y$$

- 2-cells are degree-wise inequalities.

Depth p is modelled in cubical sets with $p + 1$ different dimension flavours.

The mode theory DoR

RelDTT can be built on an instance of MTT (Multimode Type Theory) with mode theory DoR:

- Modes are **depths** $p \in \mathbb{Z}_{\geq -1}$
- Modalities $\mu : p \to q$ are
  functions $\{0 \leq \ldots \leq q\} \to \{(=) \leq 0 \leq \ldots \leq p \leq \top\} : i \mapsto i \cdot \mu$
  where $f : (\mu \mid x : A) \to B(x)$ sends

$$(r : x \frown^{A}_{i \cdot \mu} y) \quad \to \quad f(x) \frown^{B(r)}_{i} f(y).$$

Modal types:

$$\text{mod}_{\mu}\, x \frown^{\langle \mu | A \rangle}_{i} \text{mod}_{\mu}\, y \quad = \quad x \frown^{A}_{i \cdot \mu} y$$

- 2-cells are degree-wise inequalities.

Depth p is modelled in cubical sets with $p + 1$ different dimension flavours.

RelDTT can be built on an instance of MTT (Multimode Type Theory) with mode theory DoR:

- Modes are **depths** $p \in \mathbb{Z}_{\geq -1}$
- Modalities $\mu : p \to q$ are
  functions $\{0 \leq \ldots \leq q\} \to \{(=) \leq 0 \leq \ldots \leq p \leq \top\} : i \mapsto i \cdot \mu$
  where $f : (\mu \mid x : A) \to B(x)$ sends

$$(r : x \frown^{A}_{i \cdot \mu} y) \quad \to \quad f(x) \frown^{B(r)}_{i} f(y).$$

  Modal types:

$$\mathrm{mod}_\mu\, x \frown^{\langle \mu | A \rangle}_{i} \mathrm{mod}_\mu\, y \quad = \quad x \frown^{A}_{i \cdot \mu} y$$

- 2-cells are degree-wise inequalities.

Depth p is modelled in cubical sets with $p+1$ different dimension flavours.

The mode theory DoR

RelDTT can be built on an instance of MTT (Multimode Type Theory) with mode theory DoR:

- Modes are **depths** $p \in \mathbb{Z}_{\geq -1}$
- Modalities $\mu : p \to q$ are

  functions $\{0 \leq \ldots \leq q\} \to \{(=) \leq 0 \leq \ldots \leq p \leq \top\} : i \mapsto i \cdot \mu$

  where $f : (\mu \mid x : A) \to B(x)$ sends

$$(r : x \frown^{A}_{i \cdot \mu} y) \quad \to \quad f(x) \frown^{B(r)}_{i} f(y).$$

  Modal types:

$$\mathrm{mod}_{\mu}\, x \frown^{\langle \mu | A \rangle}_{i} \mathrm{mod}_{\mu}\, y \quad = \quad x \frown^{A}_{i \cdot \mu} y$$

- 2-cells are degree-wise inequalities.

Depth p is modelled in cubical sets with $p + 1$ different dimension flavours.

RelDTT can be built on an instance of MTT (Multimode Type Theory) with mode theory DoR:

- Modes are **depths** $p \in \mathbb{Z}_{\geq -1}$
- Modalities $\mu : p \to q$ are
  functions $\{0 \leq \ldots \leq q\} \to \{(=) \leq 0 \leq \ldots \leq p \leq \top\} : i \mapsto i \cdot \mu$
  where $f : (\mu \mid x : A) \to B(x)$ sends

$$(r : x \frown^{A}_{i \cdot \mu} y) \quad \to \quad f(x) \frown^{B(r)}_{i} f(y).$$

  Modal types:

$$\mathrm{mod}_\mu\, x \frown^{\langle \mu | A \rangle}_{i} \mathrm{mod}_\mu\, y \quad = \quad x \frown^{A}_{i \cdot \mu} y$$

- 2-cells are degree-wise inequalities.

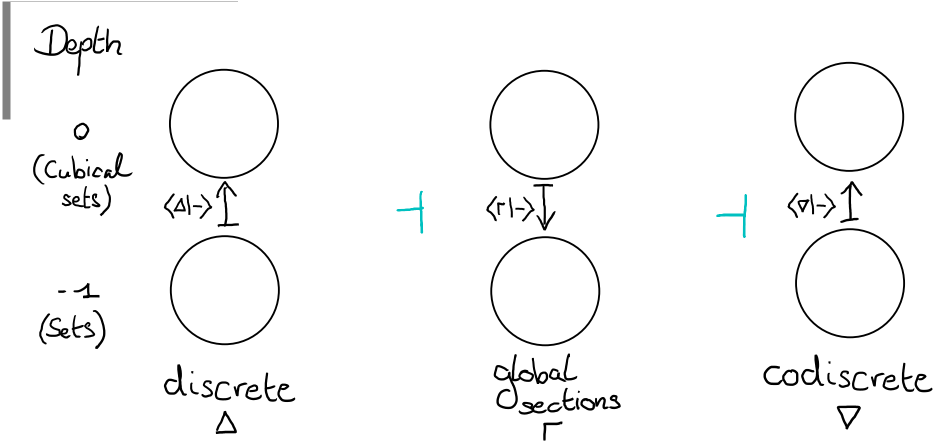
Depth p is modelled in cubical sets with $p+1$ different dimension flavours.

Depth

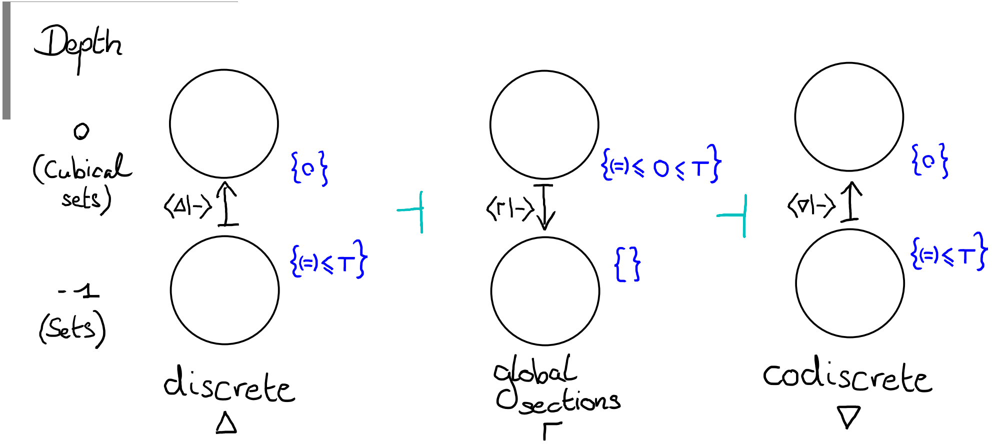
0
(Cubical sets)

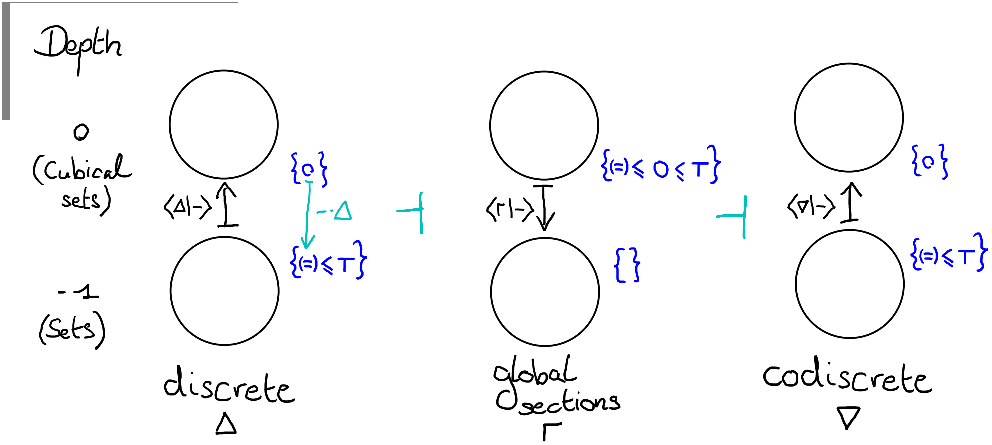
-1
(Sets)

Cohesion in RelDTT

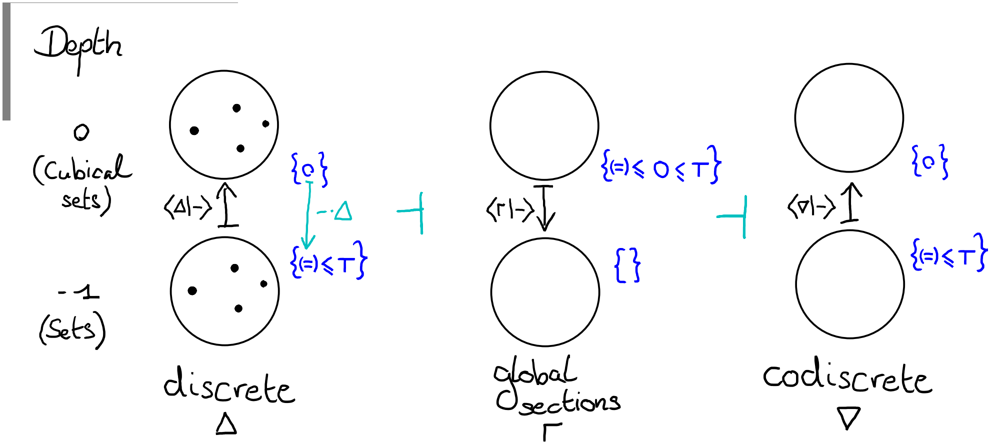


Cohesion in RelDTT

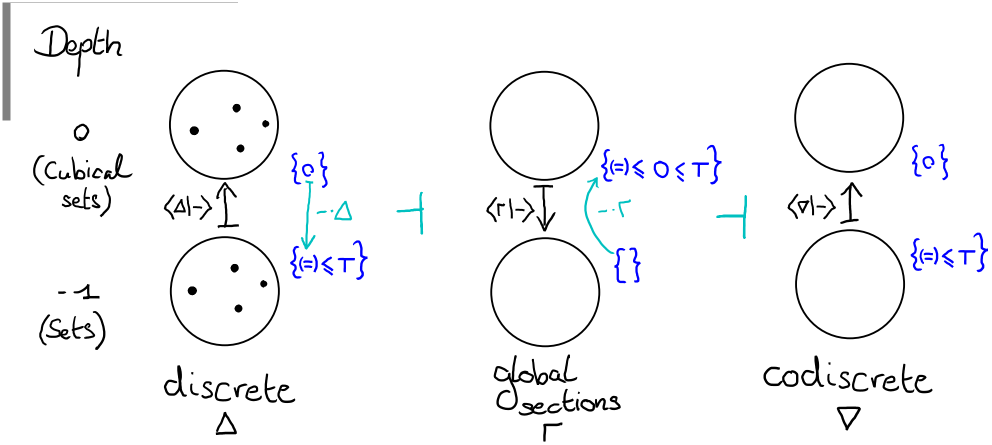


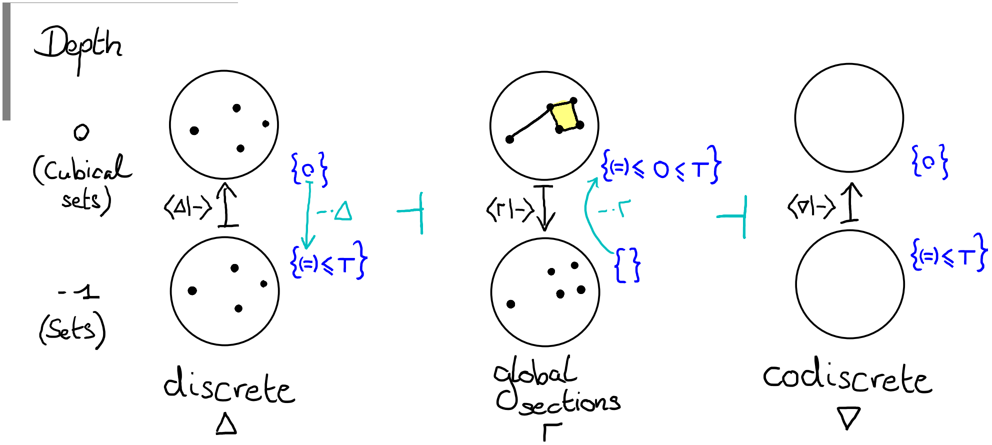

Depth
0
(Cubical sets)
-1
(Sets)
⟨Δ|-⟩
{0}
-·Δ
{(=)≤⊤}
discrete
Δ
⟨Γ|-⟩
{(=)≤0≤⊤}
{}
global sections
Γ
⟨∇|-⟩
{0}
{(=)≤⊤}
codiscrete
∇

Cohesion in RelDTT



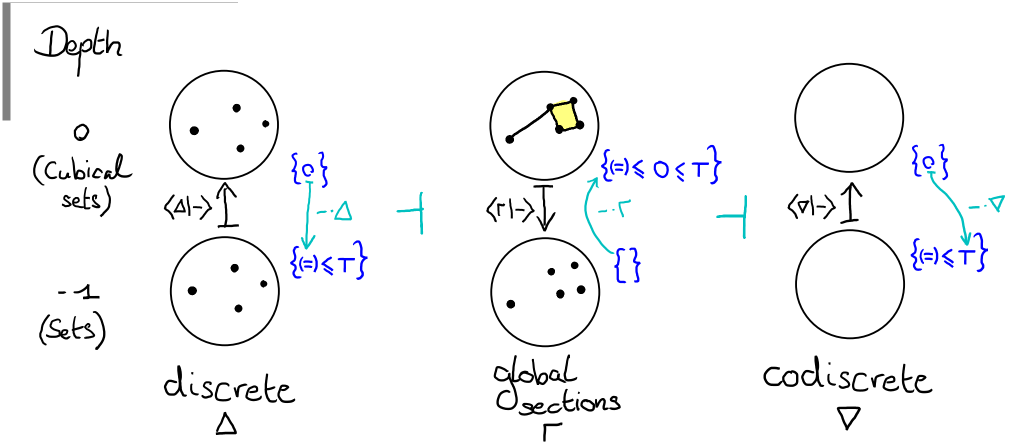
Cohesion in RelDTT



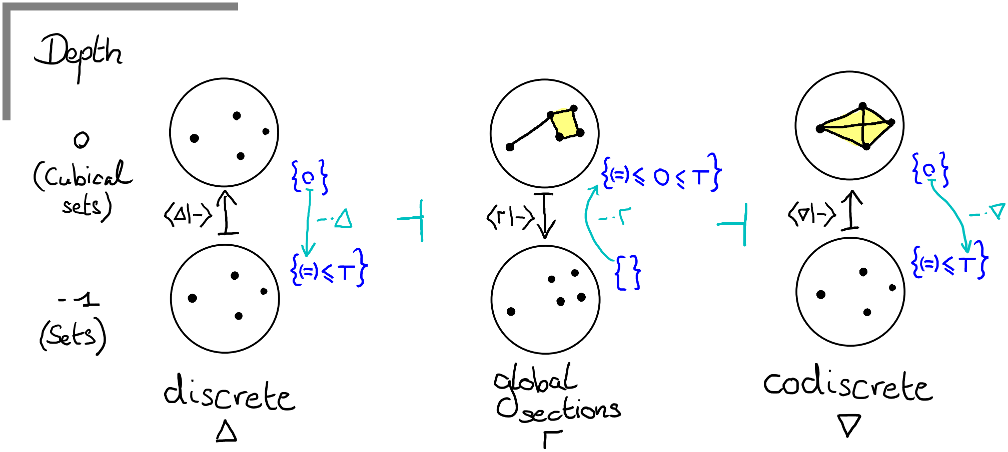


Depth
0
(Cubical sets)
-1
(Sets)
⟨Δ|-⟩
{0}
-·Δ
{(=) ≤ ⊤}
discrete
Δ
⊣
⟨Γ|-⟩
{(=) ≤ 0 ≤ ⊤}
-·Γ
{}
global sections
Γ
⊣
⟨∇|-⟩
{0}
{(=) ≤ ⊤}
codiscrete
∇

Cohesion in RelDTT



Cohesion in RelDTT



Two-Level Type Theory (2LTT)

Voevodsky (2013)
Altenkirch, Capriotti & Kraus (2016)
Annenkov, Capriotti, Kraus & Sattler (2017)

More specifically, **its general presheaf model**

Two-Level Type Theory (2LTT)

Voevodsky (2013)
Altenkirch, Capriotti & Kraus (2016)
Annenkov, Capriotti, Kraus & Sattler (2017)

More specifically, **its general presheaf model**

Inner Level

CwF $(\mathscr{C}, \mathrm{Ty}^i, \mathrm{Tm}^i)$

$\mathrm{Ty}^i \in \mathrm{Psh}(\mathscr{C})$

$\mathrm{Tm}^i \in \mathrm{Psh}\left(\int_{\mathscr{C}} \mathrm{Ty}^i\right)$

$\Gamma\ \mathsf{ctx}^i$

$\Gamma \vdash^i T\ \mathsf{type}$

$\Gamma \vdash^i t : T$

E.g inner system = HoTT

Outer Level

CwF $(\mathrm{Psh}(\mathscr{C}), \mathrm{Ty}^o, \mathrm{Tm}^o)$

$\vdash^o \mathrm{Ty}^i\ \mathsf{type}$

$X : \mathrm{Ty}^i \vdash^o \mathrm{Tm}^i(X)\ \mathsf{type}$

$\Rightarrow \mathbf{y}\Gamma\ \mathsf{ctx}^o$

$\Leftrightarrow \mathbf{y}\Gamma \vdash^o T : \mathrm{Ty}^i$

$\Leftrightarrow \mathbf{y}\Gamma \vdash^o t : \mathrm{Tm}^i(T)$

Talk about strict equality

Inner Level

CwF $(\mathscr{C}, \mathrm{Ty}^i, \mathrm{Tm}^i)$

$\mathrm{Ty}^i \in \mathrm{Psh}(\mathscr{C})$

$\mathrm{Tm}^i \in \mathrm{Psh}\left(\int_{\mathscr{C}} \mathrm{Ty}^i\right)$

$\Gamma\ \mathrm{ctx}^i$

$\Gamma \vdash^i T\ \mathrm{type}$

$\Gamma \vdash^i t : T$

E.g inner system = HoTT

Outer Level

CwF $(\mathrm{Psh}(\mathscr{C}), \mathrm{Ty}^o, \mathrm{Tm}^o)$

$\vdash^o \mathrm{Ty}^i\ \mathrm{type}$

$X : \mathrm{Ty}^i \vdash^o \mathrm{Tm}^i(X)\ \mathrm{type}$

$\Rightarrow \mathbf{y}\Gamma\ \mathrm{ctx}^o$

$\Leftrightarrow \mathbf{y}\Gamma \vdash^o T : \mathrm{Ty}^i$

$\Leftrightarrow \mathbf{y}\Gamma \vdash^o t : \mathrm{Tm}^i(T)$

Talk about strict equality

Inner Level

CwF $(\mathscr{C}, \mathrm{Ty}^i, \mathrm{Tm}^i)$

$\mathrm{Ty}^i \in \mathrm{Psh}(\mathscr{C})$

$\mathrm{Tm}^i \in \mathrm{Psh}\left(\int_{\mathscr{C}} \mathrm{Ty}^i\right)$

Γ ctx^i

$\Gamma \vdash^i T$ type

$\Gamma \vdash^i t : T$

E.g inner system = HoTT

Outer Level

CwF $(\mathrm{Psh}(\mathscr{C}), \mathrm{Ty}^o, \mathrm{Tm}^o)$

$\vdash^o \mathrm{Ty}^i$ type

$X : \mathrm{Ty}^i \vdash^o \mathrm{Tm}^i(X)$ type

$\Rightarrow \mathbf{y}\Gamma$ ctx^o

$\Leftrightarrow \mathbf{y}\Gamma \vdash^o T : \mathrm{Ty}^i$

$\Leftrightarrow \mathbf{y}\Gamma \vdash^o t : \mathrm{Tm}^i(T)$

Talk about strict equality

Inner Level

CwF $(\mathscr{C}, \mathrm{Ty}^i, \mathrm{Tm}^i)$

$\mathrm{Ty}^i \in \mathrm{Psh}(\mathscr{C})$

$\mathrm{Tm}^i \in \mathrm{Psh}\left(\int_{\mathscr{C}} \mathrm{Ty}^i\right)$

$\Gamma\ \mathrm{ctx}^i$

$\Gamma \vdash^i T$ type

$\Gamma \vdash^i t : T$

E.g inner system = HoTT

Outer Level

CwF $(\mathrm{Psh}(\mathscr{C}), \mathrm{Ty}^o, \mathrm{Tm}^o)$

$\vdash^o \mathrm{Ty}^i$ type

$X : \mathrm{Ty}^i \vdash^o \mathrm{Tm}^i(X)$ type

$\Rightarrow \mathbf{y}\Gamma\ \mathrm{ctx}^o$

$\Leftrightarrow \mathbf{y}\Gamma \vdash^o T : \mathrm{Ty}^i$

$\Leftrightarrow \mathbf{y}\Gamma \vdash^o t : \mathrm{Tm}^i(T)$

Talk about strict equality

Inner Level

CwF $(\mathscr{C}, \mathrm{Ty}^i, \mathrm{Tm}^i)$

$\mathrm{Ty}^i \in \mathrm{Psh}(\mathscr{C})$

$\mathrm{Tm}^i \in \mathrm{Psh}\left(\int_{\mathscr{C}} \mathrm{Ty}^i\right)$

$\Gamma\ \mathsf{ctx}^i$

$\Gamma \vdash^i T\ \mathsf{type}$

$\Gamma \vdash^i t : T$

E.g inner system = HoTT

Outer Level

CwF $(\mathrm{Psh}(\mathscr{C}), \mathrm{Ty}^o, \mathrm{Tm}^o)$

$\vdash^o \mathrm{Ty}^i\ \mathsf{type}$

$X : \mathrm{Ty}^i \vdash^o \mathrm{Tm}^i(X)\ \mathsf{type}$

$\Rightarrow \mathbf{y}\Gamma\ \mathsf{ctx}^o$

$\Leftrightarrow \mathbf{y}\Gamma \vdash^o T : \mathrm{Ty}^i$

$\Leftrightarrow \mathbf{y}\Gamma \vdash^o t : \mathrm{Tm}^i(T)$

Talk about strict equality

Inner Level

CwF $(\mathscr{C}, \mathrm{Ty}^i, \mathrm{Tm}^i)$

$\mathrm{Ty}^i \in \mathrm{Psh}(\mathscr{C})$

$\mathrm{Tm}^i \in \mathrm{Psh}\left(\int_{\mathscr{C}} \mathrm{Ty}^i\right)$

$\Gamma\ \mathsf{ctx}^i$

$\Gamma \vdash^i T\ \mathsf{type}$

$\Gamma \vdash^i t : T$

E.g inner system = HoTT

Outer Level

CwF $(\mathrm{Psh}(\mathscr{C}), \mathrm{Ty}^o, \mathrm{Tm}^o)$

$\vdash^o \mathrm{Ty}^i\ \mathsf{type}$

$X : \mathrm{Ty}^i \vdash^o \mathrm{Tm}^i(X)\ \mathsf{type}$

$\Rightarrow \mathbf{y}\Gamma\ \mathsf{ctx}^o$

$\Leftrightarrow \mathbf{y}\Gamma \vdash^o T : \mathrm{Ty}^i$

$\Leftrightarrow \mathbf{y}\Gamma \vdash^o t : \mathrm{Tm}^i(T)$

Talk about strict equality

Inner Level

CwF $(\mathscr{C}, \mathrm{Ty}^i, \mathrm{Tm}^i)$

$\mathrm{Ty}^i \in \mathrm{Psh}(\mathscr{C})$

$\mathrm{Tm}^i \in \mathrm{Psh}\left(\int_{\mathscr{C}} \mathrm{Ty}^i\right)$

$\Gamma\ \mathsf{ctx}^i$

$\Gamma \vdash^i T\ \mathsf{type}$

$\Gamma \vdash^i t : T$

E.g inner system = HoTT

Outer Level

CwF $(\mathrm{Psh}(\mathscr{C}), \mathrm{Ty}^o, \mathrm{Tm}^o)$

$\vdash^o \mathrm{Ty}^i\ \mathsf{type}$

$X : \mathrm{Ty}^i \vdash^o \mathrm{Tm}^i(X)\ \mathsf{type}$

$\Rightarrow \mathbf{y}\Gamma\ \mathsf{ctx}^o$

$\Leftrightarrow \mathbf{y}\Gamma \vdash^o T : \mathrm{Ty}^i$

$\Leftrightarrow \mathbf{y}\Gamma \vdash^o t : \mathrm{Tm}^i(T)$

Talk about strict equality

Inner Level

CwF $(\mathscr{C}, \mathrm{Ty}^i, \mathrm{Tm}^i)$

$\mathrm{Ty}^i \in \mathrm{Psh}(\mathscr{C})$

$\mathrm{Tm}^i \in \mathrm{Psh}\left(\int_{\mathscr{C}} \mathrm{Ty}^i\right)$

$\Gamma\ \mathsf{ctx}^i$

$\Gamma \vdash^i T\ \mathsf{type}$

$\Gamma \vdash^i t : T$

E.g inner system = HoTT

Outer Level

CwF $(\mathrm{Psh}(\mathscr{C}), \mathrm{Ty}^o, \mathrm{Tm}^o)$

$\vdash^o \mathrm{Ty}^i\ \mathsf{type}$

$X : \mathrm{Ty}^i \vdash^o \mathrm{Tm}^i(X)\ \mathsf{type}$

$\Rightarrow \mathbf{y}\Gamma\ \mathsf{ctx}^o$

$\Leftrightarrow \mathbf{y}\Gamma \vdash^o T : \mathrm{Ty}^i$

$\Leftrightarrow \mathbf{y}\Gamma \vdash^o t : \mathrm{Tm}^i(T)$

Talk about strict equality

Inner Level

CwF $(\mathscr{C}, \mathrm{Ty}^i, \mathrm{Tm}^i)$

$\mathrm{Ty}^i \in \mathrm{Psh}(\mathscr{C})$

$\mathrm{Tm}^i \in \mathrm{Psh}\left(\int_{\mathscr{C}} \mathrm{Ty}^i\right)$

$\Gamma\ \mathsf{ctx}^i$

$\Gamma \vdash^i T\ \mathsf{type}$

$\Gamma \vdash^i t : T$

E.g inner system = HoTT

Outer Level

CwF $(\mathrm{Psh}(\mathscr{C}), \mathrm{Ty}^o, \mathrm{Tm}^o)$

$\vdash^o \mathrm{Ty}^i\ \mathsf{type}$

$X : \mathrm{Ty}^i \vdash^o \mathrm{Tm}^i(X)\ \mathsf{type}$

$\Rightarrow \mathbf{y}\Gamma\ \mathsf{ctx}^o$

$\Leftrightarrow \mathbf{y}\Gamma \vdash^o T : \mathrm{Ty}^i$

$\Leftrightarrow \mathbf{y}\Gamma \vdash^o t : \mathrm{Tm}^i(T)$

Talk about strict equality

Inner Level

CwF $(\mathscr{C}, \mathrm{Ty}^i, \mathrm{Tm}^i)$

$\mathrm{Ty}^i \in \mathrm{Psh}(\mathscr{C})$

$\mathrm{Tm}^i \in \mathrm{Psh}\left(\int_{\mathscr{C}} \mathrm{Ty}^i\right)$

$\Gamma\ \mathsf{ctx}^i$

$\Gamma \vdash^i T\ \mathsf{type}$

$\Gamma \vdash^i t : T$

E.g inner system = HoTT

Outer Level

CwF $(\mathrm{Psh}(\mathscr{C}), \mathrm{Ty}^o, \mathrm{Tm}^o)$

$\vdash^o \mathrm{Ty}^i\ \mathsf{type}$

$X : \mathrm{Ty}^i \vdash^o \mathrm{Tm}^i(X)\ \mathsf{type}$

$\Rightarrow \mathbf{y}\Gamma\ \mathsf{ctx}^o$

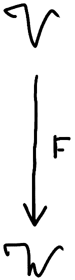
$\Leftrightarrow \mathbf{y}\Gamma \vdash^o T : \mathrm{Ty}^i$

$\Leftrightarrow \mathbf{y}\Gamma \vdash^o t : \mathrm{Tm}^i(T)$

Talk about strict equality

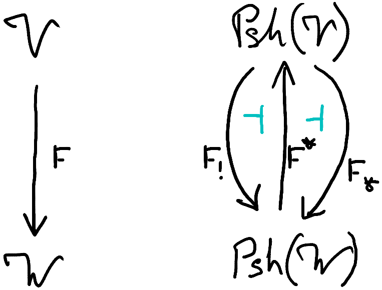
Lifting Functors

Multilevel Type Theory



- $F^* =$ precomposition by F.
- $F_!$ extends F along $\mathbf{y}$, i.e. $F_!\mathbf{y} \cong \mathbf{y}F$.

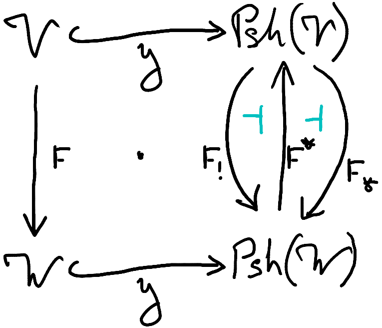
Lifting Functors



- $F^* =$ precomposition by F.
- $F_!$ extends F along $\mathbf{y}$, i.e. $F_! \mathbf{y} \cong \mathbf{y} F$.

Multilevel Type Theory

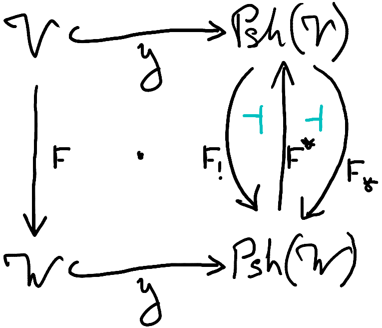
Lifting Functors



- $F^* =$ precomposition by F.
- $F_!$ extends F along $\mathbf{y}$, i.e. $F_!\mathbf{y} \cong \mathbf{y}F$.

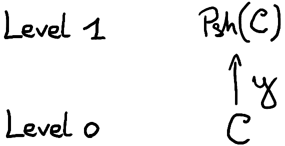
Multilevel Type Theory

Lifting Functors

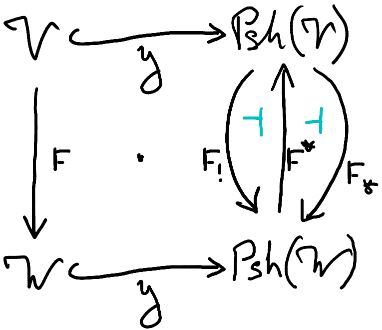


- $F^* =$ precomposition by F.
- $F_!$ extends F along $\mathbf{y}$, i.e. $F_!\mathbf{y} \cong \mathbf{y}F$.

Multilevel Type Theory

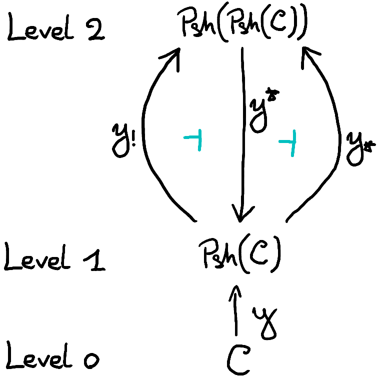


Lifting Functors

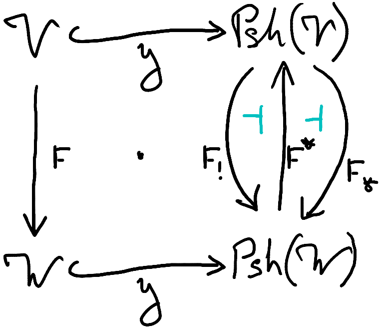


- $F^* =$ precomposition by F.
- $F_!$ extends F along $\mathbf{y}$, i.e. $F_! \mathbf{y} \cong \mathbf{y} F$.

Multilevel Type Theory

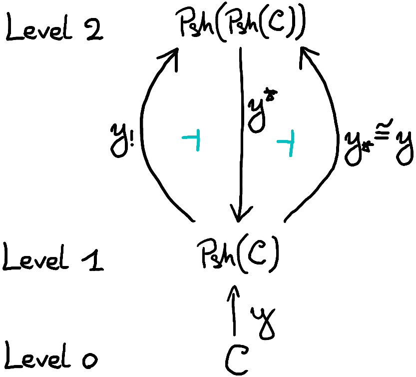


Lifting Functors



- $F^* =$ precomposition by F.
- $F_!$ extends F along $\mathbf{y}$, i.e. $F_! \mathbf{y} \cong \mathbf{y} F$.

Multilevel Type Theory



Multilevel TT: Case Study

- Level 0: $\star$ (point category is a CwF)
- Level 1: $\mathrm{Psh}(\star) \cong \mathsf{Set}$
  - $\mathbf{y} : \star \to \mathsf{Set} : \bullet \mapsto \top$
- Level 2: $\mathrm{Psh}(\mathrm{Psh}(\star)) \cong \mathrm{Psh}(\mathsf{Set})$
  - $\mathbf{y}^* : \mathrm{Psh}(\mathsf{Set}) \to \mathsf{Set}$ extracts points.

What is $\mathrm{Psh}(\mathsf{Set})$?

Presheaves over
finite
non-empty
linear orders
are
simplicial sets

Multilevel TT: Case Study

- Level 0: $\star$ (point category is a CwF)
- Level 1: $\mathrm{Psh}(\star) \cong \mathrm{Set}$
  - $\mathbf{y} : \star \to \mathrm{Set} : \bullet \mapsto \top$
- Level 2: $\mathrm{Psh}(\mathrm{Psh}(\star)) \cong \mathrm{Psh}(\mathrm{Set})$
  - $\mathbf{y}^* : \mathrm{Psh}(\mathrm{Set}) \to \mathrm{Set}$ extracts points.

What is $\mathrm{Psh}(\mathrm{Set})$?

Presheaves over
finite
non-empty
linear orders
are
simplicial sets

Multilevel TT: Case Study

- Level 0: $\star$ (point category is a CwF)
- Level 1: $\mathrm{Psh}(\star) \cong$ Set
  - $\mathbf{y} : \star \to \mathrm{Set} : \bullet \mapsto \top$
- Level 2: $\mathrm{Psh}(\mathrm{Psh}(\star)) \cong \mathrm{Psh}(\mathrm{Set})$
  - $\mathbf{y}^* : \mathrm{Psh}(\mathrm{Set}) \to \mathrm{Set}$ extracts points.

What is $\mathrm{Psh}(\mathrm{Set})$?

Presheaves over
finite
non-empty
linear orders

are

simplicial sets

Multilevel TT: Case Study

- Level 0: $\star$ (point category is a CwF)
- Level 1: $\mathrm{Psh}(\star) \cong \mathsf{Set}$
  - $\mathbf{y} : \star \to \mathsf{Set} : \bullet \mapsto \top$
- Level 2: $\mathrm{Psh}(\mathrm{Psh}(\star)) \cong \mathrm{Psh}(\mathsf{Set})$
  - $\mathbf{y}^* : \mathrm{Psh}(\mathsf{Set}) \to \mathsf{Set}$ extracts points.

What is $\mathrm{Psh}(\mathsf{Set})$?

Presheaves over

finite

non-empty

linear orders

are

simplicial sets

Multilevel TT: Case Study

- Level 0: $\star$ (point category is a CwF)
- Level 1: $\mathrm{Psh}(\star) \cong \mathsf{Set}$
  - $\mathbf{y} : \star \to \mathsf{Set} : \bullet \mapsto \top$
- Level 2: $\mathrm{Psh}(\mathrm{Psh}(\star)) \cong \mathrm{Psh}(\mathsf{Set})$
  - $\mathbf{y}^* : \mathrm{Psh}(\mathsf{Set}) \to \mathsf{Set}$ extracts points.

What is $\mathrm{Psh}(\mathsf{Set})$?

Presheaves over

finite

non-empty

linear orders

are

simplicial sets

Multilevel TT: Case Study

- Level 0: $\star$ (point category is a CwF)
- Level 1: $\mathrm{Psh}(\star) \cong \mathsf{Set}$
  - $\mathbf{y} : \star \to \mathsf{Set} : \bullet \mapsto \top$
- Level 2: $\mathrm{Psh}(\mathrm{Psh}(\star)) \cong \mathrm{Psh}(\mathsf{Set})$
  - $\mathbf{y}^* : \mathrm{Psh}(\mathsf{Set}) \to \mathsf{Set}$ extracts points.

What is $\mathrm{Psh}(\mathsf{Set})$?

Presheaves over

finite

~~non-empty~~

linear orders

are

~~semi~~ augmented simplicial sets

Multilevel TT: Case Study

- Level 0: $\star$ (point category is a CwF)
- Level 1: $\mathrm{Psh}(\star) \cong \mathsf{Set}$
  - $\mathbf{y} : \star \to \mathsf{Set} : \bullet \mapsto \top$
- Level 2: $\mathrm{Psh}(\mathrm{Psh}(\star)) \cong \mathrm{Psh}(\mathsf{Set})$
  - $\mathbf{y}^* : \mathrm{Psh}(\mathsf{Set}) \to \mathsf{Set}$ extracts points.

What is $\mathrm{Psh}(\mathsf{Set})$?

Presheaves over

finite

~~non-empty~~

~~linear orders~~ sets

are

symmetric

~~semi~~ augmented simplicial sets

Multilevel TT: Case Study

- Level 0: $\star$ (point category is a CwF)
- Level 1: $\mathrm{Psh}(\star) \cong \mathsf{Set}$
  - $\mathbf{y} : \star \to \mathsf{Set} : \bullet \mapsto \top$
- Level 2: $\mathrm{Psh}(\mathrm{Psh}(\star)) \cong \mathrm{Psh}(\mathsf{Set})$
  - $\mathbf{y}^* : \mathrm{Psh}(\mathsf{Set}) \to \mathsf{Set}$ extracts points.

What is $\mathrm{Psh}(\mathsf{Set})$?

Presheaves over

~~finite~~

~~non-empty~~

~~linear orders~~ sets

are

symmetric

~~semi~~ augmented simplicial sets

with ∞-dimensional simplices

Multilevel TT: Case Study

- Level 0: $\star$ (point category is a CwF)
- Level 1: $\mathrm{Psh}(\star) \cong \mathsf{Set}$
  - $\mathbf{y} : \star \to \mathsf{Set} : \bullet \mapsto \top$
- Level 2: $\mathrm{Psh}(\mathrm{Psh}(\star)) \cong \mathrm{Psh}(\mathsf{Set})$
  - $\mathbf{y}^* : \mathrm{Psh}(\mathsf{Set}) \to \mathsf{Set}$ extracts points.

What is $\mathrm{Psh}(\mathsf{Set})$?

Presheaves over

~~finite~~

~~non-empty~~

~~linear orders~~ sets

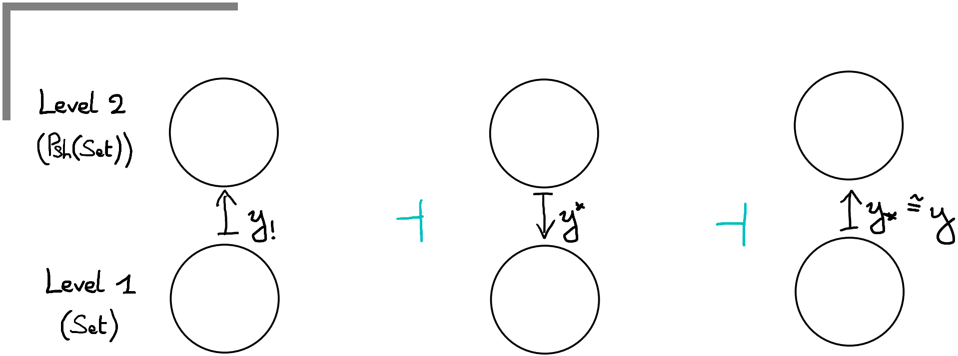
are

symmetric

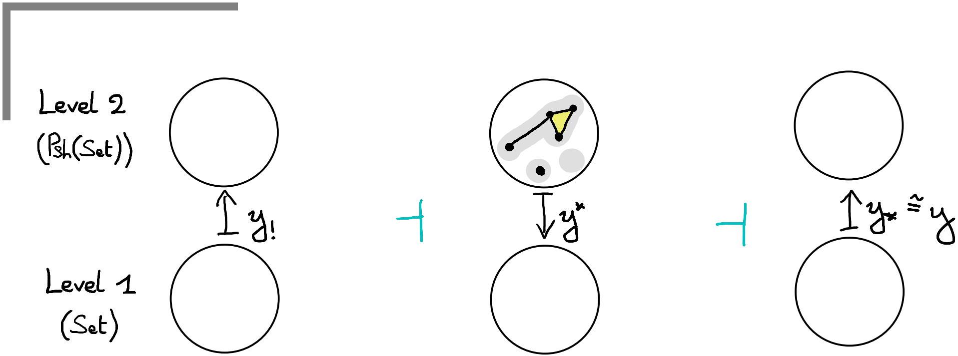
~~semi~~ augmented simplicial sets

with ∞-dimensional simplices

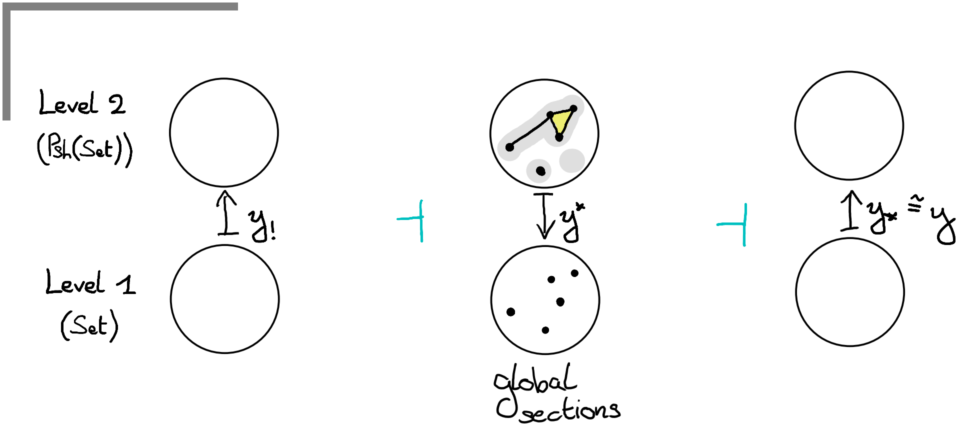
Cohesion in Multilevel TT



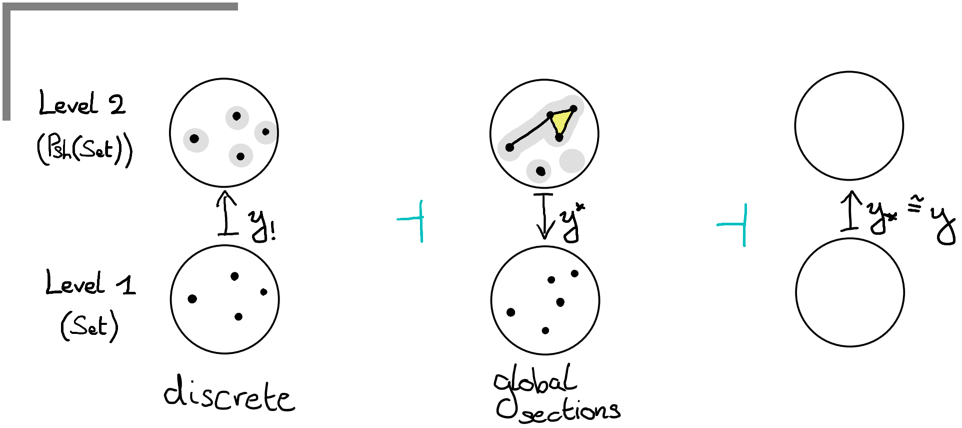
Cohesion in Multilevel TT



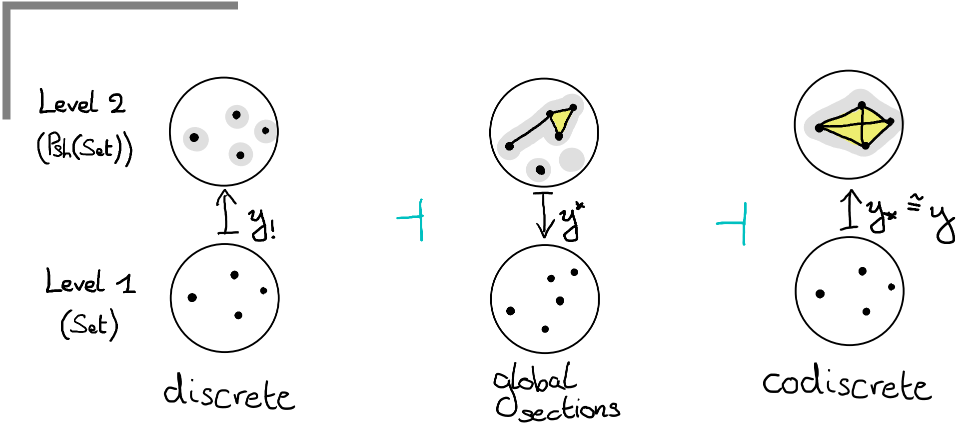
Cohesion in Multilevel TT



Cohesion in Multilevel TT



Cohesion in Multilevel TT



Degrees of Relatedness $\overset{?}{\sim}$ Multilevel TT

Goal: Formalize this correspondence, using **interface** of

Lax-idempotent 2-monads

Degrees of Relatedness $\stackrel{?}{\sim}$ Multilevel TT

Goal: Formalize this correspondence, using **interface** of

Lax-idempotent 2-monads

$$\text{Degrees of Relatedness} \overset{?}{\sim} \text{Multilevel TT}$$

Goal: Formalize this correspondence, using **interface** of

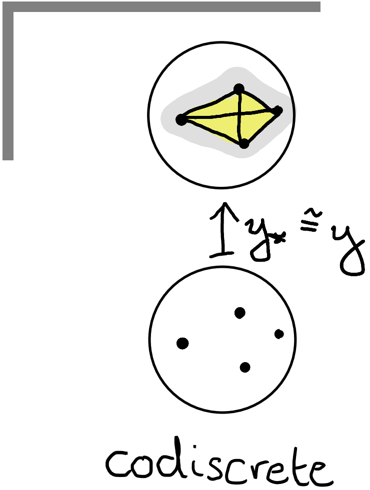
Lax-idempotent 2-monads

Why Psh : Cat → Cat is a Monad

- In Psh(Set), Yoneda-embeddings (representables) are **codiscrete**.
- Presheaves are colimits of codiscrete presheaves.
- $\mathrm{Psh}(\mathscr{C})$ is **freely** generated by $\mathbf{y} : \mathscr{C} \to \mathrm{Psh}(\mathscr{C})$ and **colimits**. (Free cocompletion.)
  - Like List X is freely generated by $[-] : X \to$ List X and [] and ++.

  It's a (2-)monad!
  - Unit $\eta = \mathbf{y} : \mathscr{C} \to \mathrm{Psh}(\mathscr{C})$,
  - (Pseudo)functorial action: $F \mapsto F_!$,
  - $\mu = \mathbf{y}^* : \mathrm{Psh}(\mathrm{Psh}(\mathscr{C})) \to \mathrm{Psh}(\mathscr{C})$
- It's a **lax-idempotent** 2-monad: $\mathrm{Psh}\,\eta \dashv \mu \dashv \eta\,\mathrm{Psh}$

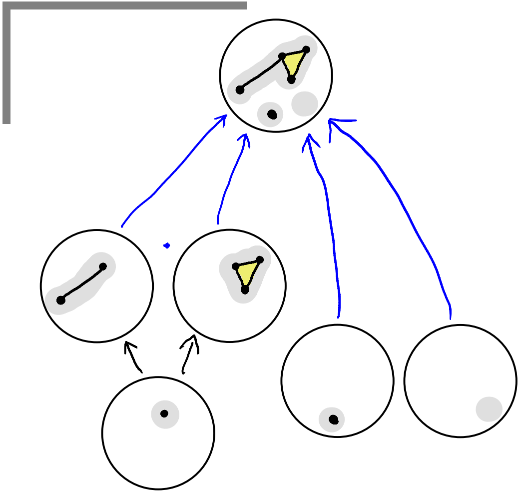
- In Psh(Set), Yoneda-embeddings (representables) are **codiscrete**.
- Presheaves are colimits of codiscrete presheaves.
- $\mathrm{Psh}(\mathscr{C})$ is **freely** generated by $\mathbf{y} : \mathscr{C} \to \mathrm{Psh}(\mathscr{C})$ and **colimits**. (Free cocompletion.)
  - Like List X is freely generated by $[-] : X \to$ List X and [] and ++.
- **It's a (2-)monad!**
  - Unit $\eta = \mathbf{y} : \mathscr{C} \to \mathrm{Psh}(\mathscr{C})$,
  - (Pseudo)functorial action: $F \mapsto F_!$,
  - $\mu = \mathbf{y}^* : \mathrm{Psh}(\mathrm{Psh}(\mathscr{C})) \to \mathrm{Psh}(\mathscr{C})$
- It's a **lax-idempotent** 2-monad: $\mathrm{Psh}\,\eta \dashv \mu \dashv \eta\,\mathrm{Psh}$



- In $\mathrm{Psh}(\mathrm{Set})$, Yoneda-embeddings (representables) are **codiscrete**.
- Presheaves are colimits of codiscrete presheaves.
- $\mathrm{Psh}(\mathscr{C})$ is **freely** generated by $\mathbf{y} : \mathscr{C} \to \mathrm{Psh}(\mathscr{C})$ and **colimits**. (Free cocompletion.)
  - Like $\mathrm{List}\,X$ is freely generated by $[-] : X \to \mathrm{List}\,X$ and [] and ++.

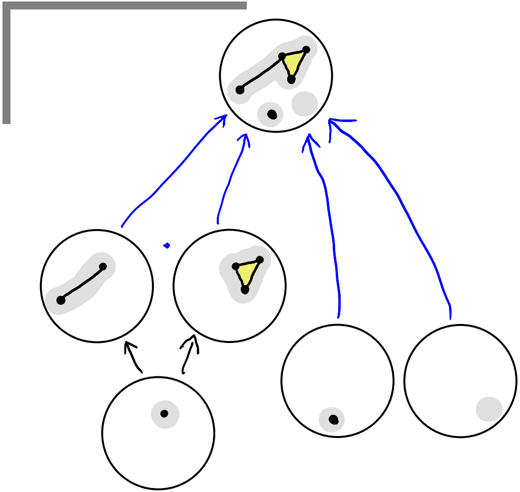
It's a (2-)monad!

  - Unit $\eta = \mathbf{y} : \mathscr{C} \to \mathrm{Psh}(\mathscr{C})$,
  - (Pseudo)functorial action: $F \mapsto F_!$,
  - $\mu = \mathbf{y}^* : \mathrm{Psh}(\mathrm{Psh}(\mathscr{C})) \to \mathrm{Psh}(\mathscr{C})$
- It's a **lax-idempotent** 2-monad: $\mathrm{Psh}\,\eta \dashv \mu \dashv \eta\,\mathrm{Psh}$



- In $\mathrm{Psh}(\mathrm{Set})$, Yoneda-embeddings (representables) are **codiscrete**.
- Presheaves are colimits of codiscrete presheaves.
- $\mathrm{Psh}(\mathscr{C})$ is **freely** generated by $\mathbf{y} : \mathscr{C} \to \mathrm{Psh}(\mathscr{C})$ and **colimits**. (Free cocompletion.)
  - Like List X is freely generated by $[-] : X \to \text{List } X$ and [] and ++.

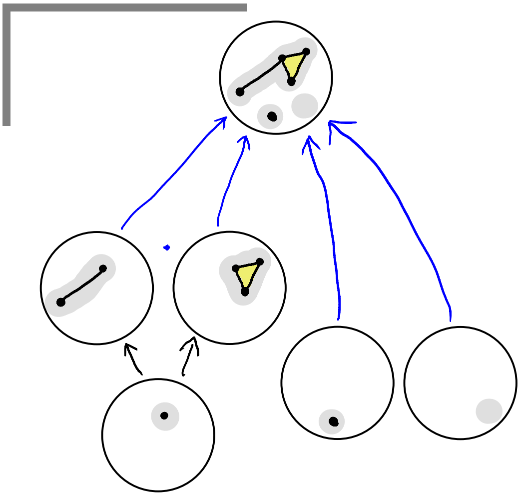
  It's a (2-)monad!
  - Unit $\eta = \mathbf{y} : \mathscr{C} \to \mathrm{Psh}(\mathscr{C})$,
  - (Pseudo)functorial action: $F \mapsto F_!$,
  - $\mu = \mathbf{y}^* : \mathrm{Psh}(\mathrm{Psh}(\mathscr{C})) \to \mathrm{Psh}(\mathscr{C})$
- It's a **lax-idempotent** 2-monad: $\mathrm{Psh}\,\eta \dashv \mu \dashv \eta\,\mathrm{Psh}$



Why Psh : Cat → Cat is a Monad

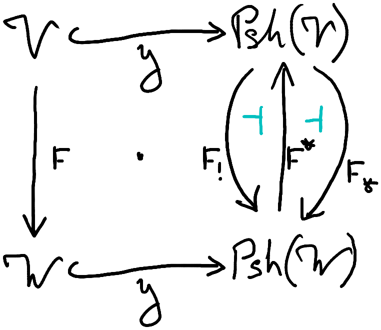
- In $\mathrm{Psh}(\mathrm{Set})$, Yoneda-embeddings (representables) are **codiscrete**.
- Presheaves are colimits of codiscrete presheaves.
- $\mathrm{Psh}(\mathscr{C})$ is **freely** generated by $\mathbf{y} : \mathscr{C} \to \mathrm{Psh}(\mathscr{C})$ and **colimits**.
  (Free cocompletion.)
  - Like List X is freely generated by $[-] : X \to \mathrm{List}\, X$ and [] and $++$.

  It's a (2-)monad!
  - Unit $\eta = \mathbf{y} : \mathscr{C} \to \mathrm{Psh}(\mathscr{C})$,
  - (Pseudo)functorial action: $F \mapsto F_!$,
  - $\mu = \mathbf{y}^* : \mathrm{Psh}(\mathrm{Psh}(\mathscr{C})) \to \mathrm{Psh}(\mathscr{C})$
- It's a **lax-idempotent** 2-monad:
  $\mathrm{Psh}\,\eta \dashv \mu \dashv \eta\,\mathrm{Psh}$



- In $\mathrm{Psh}(\mathrm{Set})$, Yoneda-embeddings (representables) are **codiscrete**.
- Presheaves are colimits of codiscrete presheaves.
- $\mathrm{Psh}(\mathscr{C})$ is **freely** generated by $\mathbf{y} : \mathscr{C} \to \mathrm{Psh}(\mathscr{C})$ and **colimits**. (Free cocompletion.)
    - Like List X is freely generated by $[-] : X \to \mathrm{List}\,X$ and $[]$ and $++$.

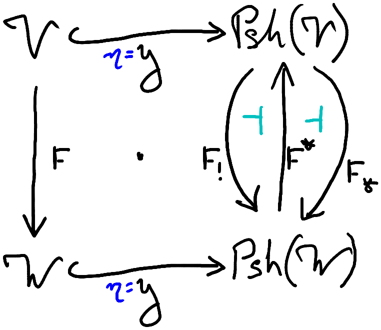
  It's a (2-)monad!
    - Unit $\eta = \mathbf{y} : \mathscr{C} \to \mathrm{Psh}(\mathscr{C})$,
    - (Pseudo)functorial action: $F \mapsto F_!$,
    - $\mu = \mathbf{y}^* : \mathrm{Psh}(\mathrm{Psh}(\mathscr{C})) \to \mathrm{Psh}(\mathscr{C})$
- It's a **lax-idempotent** 2-monad: $\mathrm{Psh}\,\eta \dashv \mu \dashv \eta\,\mathrm{Psh}$



Why Psh : Cat → Cat is a Monad

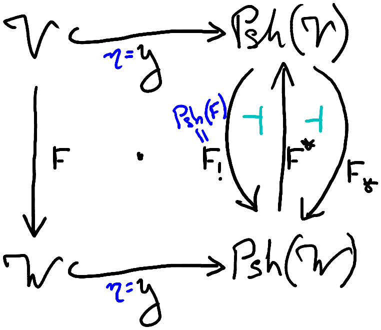
- In $\mathrm{Psh}(\mathrm{Set})$, Yoneda-embeddings (representables) are **codiscrete**.
- Presheaves are colimits of codiscrete presheaves.
- $\mathrm{Psh}(\mathscr{C})$ is **freely** generated by $\mathbf{y} : \mathscr{C} \to \mathrm{Psh}(\mathscr{C})$ and **colimits**. (Free cocompletion.)
  - Like List X is freely generated by $[-] : X \to \mathrm{List}\, X$ and $[]$ and $++$.

  It's a (2-)monad!
  - Unit $\eta = \mathbf{y} : \mathscr{C} \to \mathrm{Psh}(\mathscr{C})$,
  - (Pseudo)functorial action: $F \mapsto F_!$,
  - $\mu = \mathbf{y}^* : \mathrm{Psh}(\mathrm{Psh}(\mathscr{C})) \to \mathrm{Psh}(\mathscr{C})$
- It's a **lax-idempotent** 2-monad: $\mathrm{Psh}\, \eta \dashv \mu \dashv \eta\, \mathrm{Psh}$



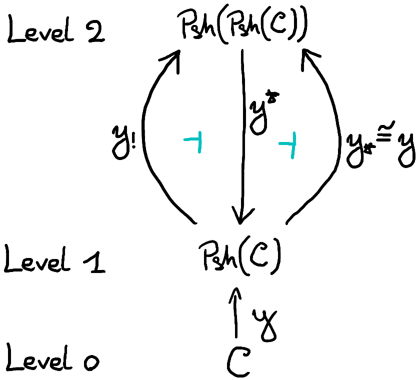
- In $\mathrm{Psh}(\mathrm{Set})$, Yoneda-embeddings (representables) are **codiscrete**.
- Presheaves are colimits of codiscrete presheaves.
- $\mathrm{Psh}(\mathscr{C})$ is **freely** generated by $\mathbf{y} : \mathscr{C} \to \mathrm{Psh}(\mathscr{C})$ and **colimits**. (Free cocompletion.)
  - Like List X is freely generated by $[-] : X \to \mathrm{List}\, X$ and $[]$ and $++$.

  It's a (2-)monad!
  - Unit $\eta = \mathbf{y} : \mathscr{C} \to \mathrm{Psh}(\mathscr{C})$,
  - (Pseudo)functorial action: $F \mapsto F_!$,
  - $\mu = \mathbf{y}^* : \mathrm{Psh}(\mathrm{Psh}(\mathscr{C})) \to \mathrm{Psh}(\mathscr{C})$
- It's a **lax-idempotent** 2-monad: $\mathrm{Psh}\,\eta \dashv \mu \dashv \eta\,\mathrm{Psh}$



- In $\mathrm{Psh}(\mathrm{Set})$, Yoneda-embeddings (representables) are **codiscrete**.
- Presheaves are colimits of codiscrete presheaves.
- $\mathrm{Psh}(\mathscr{C})$ is **freely** generated by $\mathbf{y} : \mathscr{C} \to \mathrm{Psh}(\mathscr{C})$ and **colimits**. (Free cocompletion.)
  - Like List X is freely generated by $[-] : X \to \mathrm{List}\, X$ and [] and $++$.

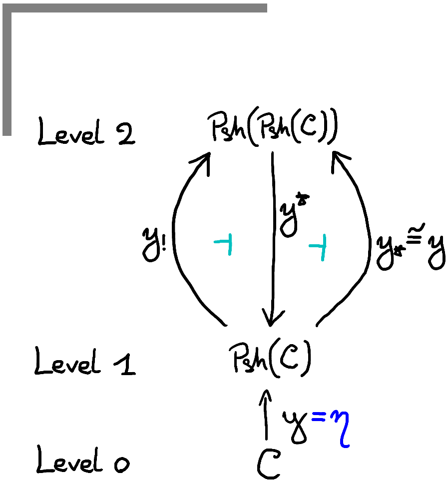
  It's a (2-)monad!
  - Unit $\eta = \mathbf{y} : \mathscr{C} \to \mathrm{Psh}(\mathscr{C})$,
  - (Pseudo)functorial action: $F \mapsto F_!$,
  - $\mu = \mathbf{y}^* : \mathrm{Psh}(\mathrm{Psh}(\mathscr{C})) \to \mathrm{Psh}(\mathscr{C})$
- It's a **lax-idempotent** 2-monad: $\mathrm{Psh}\,\eta \dashv \mu \dashv \eta\,\mathrm{Psh}$



Why Psh : Cat → Cat is a Monad

- In $\mathrm{Psh}(\mathrm{Set})$, Yoneda-embeddings (representables) are **codiscrete**.
- Presheaves are colimits of codiscrete presheaves.
- $\mathrm{Psh}(\mathscr{C})$ is **freely** generated by $\mathbf{y} : \mathscr{C} \to \mathrm{Psh}(\mathscr{C})$ and **colimits**. (Free cocompletion.)
  - Like List X is freely generated by $[-] : X \to \mathrm{List}\,X$ and $[]$ and $++$.

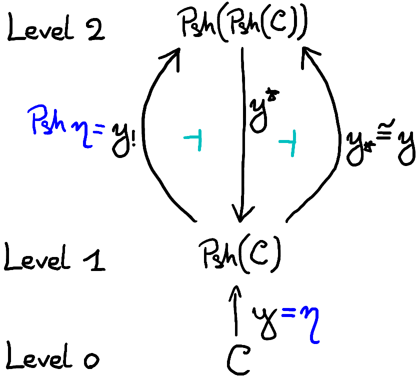
  It's a (2-)monad!
  - Unit $\eta = \mathbf{y} : \mathscr{C} \to \mathrm{Psh}(\mathscr{C})$,
  - (Pseudo)functorial action: $F \mapsto F_!$,
  - $\mu = \mathbf{y}^* : \mathrm{Psh}(\mathrm{Psh}(\mathscr{C})) \to \mathrm{Psh}(\mathscr{C})$
- It's a **lax-idempotent** 2-monad: $\mathrm{Psh}\,\eta \dashv \mu \dashv \eta\,\mathrm{Psh}$



Why Psh : Cat → Cat is a Monad

- In $\mathrm{Psh}(\mathrm{Set})$, Yoneda-embeddings (representables) are **codiscrete**.
- Presheaves are colimits of codiscrete presheaves.
- $\mathrm{Psh}(\mathscr{C})$ is **freely** generated by $\mathbf{y} : \mathscr{C} \to \mathrm{Psh}(\mathscr{C})$ and **colimits**. (Free cocompletion.)
  - Like List X is freely generated by $[-] : X \to \mathrm{List}\, X$ and $[]$ and $++$.

  It's a (2-)monad!
  - Unit $\eta = \mathbf{y} : \mathscr{C} \to \mathrm{Psh}(\mathscr{C})$,
  - (Pseudo)functorial action: $F \mapsto F_!$,
  - $\mu = \mathbf{y}^* : \mathrm{Psh}(\mathrm{Psh}(\mathscr{C})) \to \mathrm{Psh}(\mathscr{C})$
- It's a **lax-idempotent** 2-monad: $\mathrm{Psh}\,\eta \dashv \mu \dashv \eta\,\mathrm{Psh}$

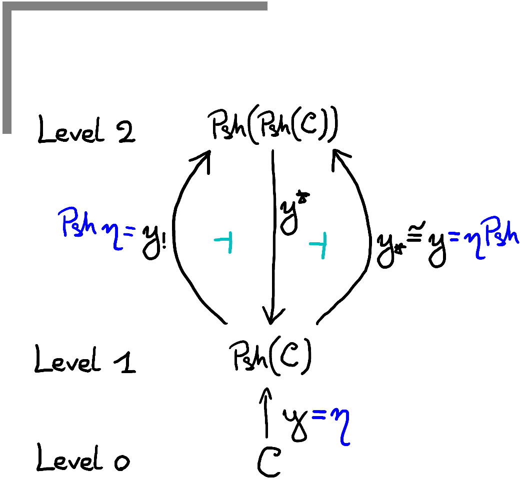


- In $\mathrm{Psh}(\mathrm{Set})$, Yoneda-embeddings (representables) are **codiscrete**.
- Presheaves are colimits of codiscrete presheaves.
- $\mathrm{Psh}(\mathscr{C})$ is **freely** generated by $\mathbf{y} : \mathscr{C} \to \mathrm{Psh}(\mathscr{C})$ and **colimits**. (Free cocompletion.)
    - Like List X is freely generated by $[-] : X \to \mathrm{List}\, X$ and $[]$ and $++$.

It's a (2-)monad!

- Unit $\eta = \mathbf{y} : \mathscr{C} \to \mathrm{Psh}(\mathscr{C})$,
- (Pseudo)functorial action: $F \mapsto F_!$,
- $\mu = \mathbf{y}^* : \mathrm{Psh}(\mathrm{Psh}(\mathscr{C})) \to \mathrm{Psh}(\mathscr{C})$

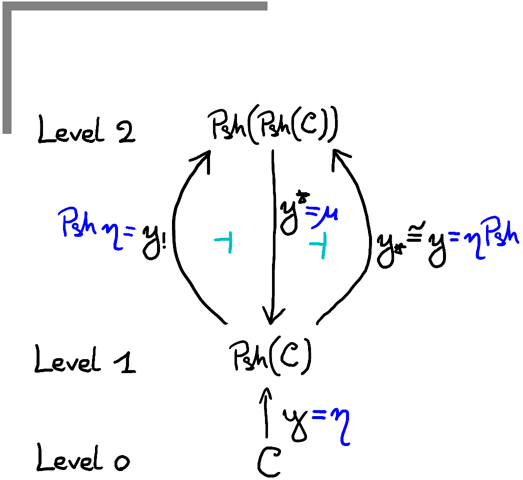
- It's a **lax-idempotent** 2-monad: $\mathrm{Psh}\,\eta \dashv \mu \dashv \eta\,\mathrm{Psh}$



Why Psh : Cat → Cat is a Monad

- In $\mathrm{Psh}(\mathrm{Set})$, Yoneda-embeddings (representables) are **codiscrete**.
- Presheaves are colimits of codiscrete presheaves.
- $\mathrm{Psh}(\mathscr{C})$ is **freely** generated by $\mathbf{y} : \mathscr{C} \to \mathrm{Psh}(\mathscr{C})$ and **colimits**. (Free cocompletion.)
  - Like List X is freely generated by $[-] : X \to \mathrm{List}\,X$ and [] and $++$.

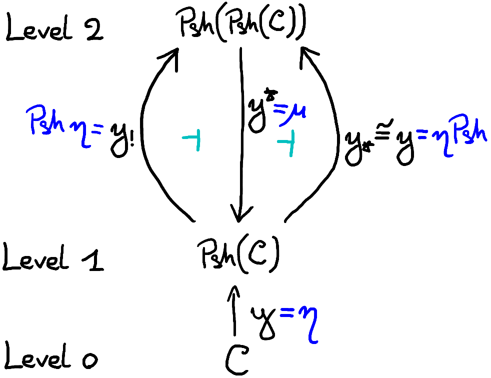
  It's a (2-)monad!
  - Unit $\eta = \mathbf{y} : \mathscr{C} \to \mathrm{Psh}(\mathscr{C})$,
  - (Pseudo)functorial action: $F \mapsto F_!$,
  - $\mu = \mathbf{y}^* : \mathrm{Psh}(\mathrm{Psh}(\mathscr{C})) \to \mathrm{Psh}(\mathscr{C})$
- It's a **lax-idempotent** 2-monad: $\mathrm{Psh}\,\eta \dashv \mu \dashv \eta\,\mathrm{Psh}$



Why Psh : Cat → Cat is a Monad

- In $\mathrm{Psh}(\mathrm{Set})$, Yoneda-embeddings (representables) are **codiscrete**.
- Presheaves are colimits of codiscrete presheaves.
- $\mathrm{Psh}(\mathscr{C})$ is **freely** generated by $\mathbf{y} : \mathscr{C} \to \mathrm{Psh}(\mathscr{C})$ and **colimits**. (Free cocompletion.)
  - Like List X is freely generated by $[-] : X \to \mathrm{List}\,X$ and $[]$ and $++$.

  It's a (2-)monad!
  - Unit $\eta = \mathbf{y} : \mathscr{C} \to \mathrm{Psh}(\mathscr{C})$,
  - (Pseudo)functorial action: $F \mapsto F_!$,
  - $\mu = \mathbf{y}^* : \mathrm{Psh}(\mathrm{Psh}(\mathscr{C})) \to \mathrm{Psh}(\mathscr{C})$
- It's a **lax-idempotent** 2-monad: $\mathrm{Psh}\,\eta \dashv \mu \dashv \eta\,\mathrm{Psh}$



Definition

Define 2-category DoR:

- Objects are $p \in \mathbb{Z}_{\geq -1}$
- Morphisms $\mu : p \to q$ are functions $\{0 \leq \ldots \leq q\} \to \{(=) \leq 0 \leq \ldots \leq p \leq \top\} : i \mapsto i \cdot \mu$
- 2-cells are degree-wise inequalities.
- Freely add $\bot =: -2$.

Definition

Let LIM be the 2-category freely generated by:

- $\mathbf{C} \in \mathrm{Obj}(\mathrm{LIM})$,
- Lax-idempotent 2-monad $\mathbf{M} : \mathrm{LIM} \to \mathrm{LIM}$

Clearly, we get $[\![-]\!]_{\mathscr{C}} : \mathrm{LIM} \to \mathrm{Cat}$ for any category $\mathscr{C}$:

- $\mathbf{C} \mapsto \mathscr{C}$,
- $\mathbf{M} \mapsto \mathrm{Psh}$.

Main theorem (formal proof WIP)

$$\begin{array}{ccccc} \mathrm{DoR} & \cong & \mathrm{LIM} & \to & \mathrm{Cat} \\ p & \mapsto & \mathbf{M}^{p+2}(\mathbf{C}) & \mapsto & \mathrm{Psh}^{p+2}(\mathscr{C}) \end{array}$$

Definition

Define 2-category DoR:

- Objects are $p \in \mathbb{Z}_{\geq -1}$
- Morphisms $\mu : p \to q$ are functions $\{0 \leq \ldots \leq q\} \to \{(=) \leq 0 \leq \ldots \leq p \leq \top\} : i \mapsto i \cdot \mu$
- 2-cells are degree-wise inequalities.
- Freely add $\bot =: -2$.

Definition

Let LIM be the 2-category freely generated by:

- $\mathbf{C} \in \mathrm{Obj}(\mathrm{LIM})$,
- Lax-idempotent 2-monad $\mathbf{M} : \mathrm{LIM} \to \mathrm{LIM}$

Clearly, we get $[\![-]\!]_{\mathscr{C}} : \mathrm{LIM} \to \mathrm{Cat}$ for any category $\mathscr{C}$:

- $\mathbf{C} \mapsto \mathscr{C}$,
- $\mathbf{M} \mapsto \mathrm{Psh}$.

Main theorem (formal proof WIP)

$$\begin{array}{ccccc} \mathrm{DoR} & \cong & \mathrm{LIM} & \to & \mathrm{Cat} \\ p & \mapsto & \mathbf{M}^{p+2}(\mathbf{C}) & \mapsto & \mathrm{Psh}^{p+2}(\mathscr{C}) \end{array}$$

Definition

Define 2-category DoR:

- Objects are $p \in \mathbb{Z}_{\geq -1}$
- Morphisms $\mu : p \to q$ are functions $\{0 \leq \ldots \leq q\} \to \{(=) \leq 0 \leq \ldots \leq p \leq \top\} : i \mapsto i \cdot \mu$
- 2-cells are degree-wise inequalities.
- Freely add $\bot =: -2$.

Definition

Let LIM be the 2-category freely generated by:

- $\mathbf{C} \in \mathrm{Obj}(\mathrm{LIM})$,
- Lax-idempotent 2-monad $\mathbf{M} : \mathrm{LIM} \to \mathrm{LIM}$

Clearly, we get $[\![-]\!]_{\mathscr{C}} : \mathrm{LIM} \to \mathrm{Cat}$ for any category $\mathscr{C}$:

- $\mathbf{C} \mapsto \mathscr{C}$,
- $\mathbf{M} \mapsto \mathrm{Psh}$.

Main theorem (formal proof WIP)

$$\begin{array}{ccccc} \mathrm{DoR} & \cong & \mathrm{LIM} & \to & \mathrm{Cat} \\ p & \mapsto & \mathbf{M}^{p+2}(\mathbf{C}) & \mapsto & \mathrm{Psh}^{p+2}(\mathscr{C}) \end{array}$$

Definition

Define 2-category DoR:

- Objects are $p \in \mathbb{Z}_{\geq -1}$
- Morphisms $\mu : p \to q$ are functions $\{0 \leq \ldots \leq q\} \to \{(=) \leq 0 \leq \ldots \leq p \leq \top\} : i \mapsto i \cdot \mu$
- 2-cells are degree-wise inequalities.
- Freely add $\perp =: -2$.

Definition

Let LIM be the 2-category freely generated by:

- $\mathbf{C} \in \mathrm{Obj}(\mathrm{LIM})$,
- Lax-idempotent 2-monad $\mathbf{M} : \mathrm{LIM} \to \mathrm{LIM}$

Clearly, we get $[\![-]\!]_{\mathscr{C}} : \mathrm{LIM} \to \mathrm{Cat}$ for any category $\mathscr{C}$:

- $\mathbf{C} \mapsto \mathscr{C}$,
- $\mathbf{M} \mapsto \mathrm{Psh}$.

Main theorem (formal proof WIP)

$$\begin{array}{ccccc} \mathrm{DoR} & \cong & \mathrm{LIM} & \to & \mathrm{Cat} \\ p & \mapsto & \mathbf{M}^{p+2}(\mathbf{C}) & \mapsto & \mathrm{Psh}^{p+2}(\mathscr{C}) \end{array}$$

Definition

Define 2-category DoR:

- Objects are $p \in \mathbb{Z}_{\geq -1}$
- Morphisms $\mu : p \to q$ are functions $\{0 \leq \ldots \leq q\} \to \{(=) \leq 0 \leq \ldots \leq p \leq \top\} : i \mapsto i \cdot \mu$
- 2-cells are degree-wise inequalities.
- Freely add $\bot =: -2$.

Definition

Let LIM be the 2-category freely generated by:

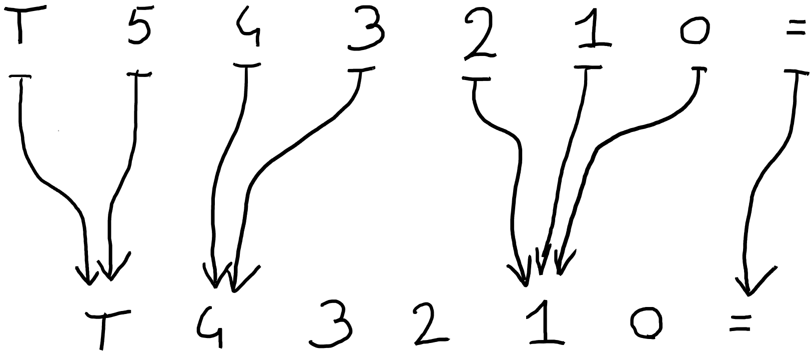
- $\mathbf{C} \in \mathrm{Obj}(\mathrm{LIM})$,
- Lax-idempotent 2-monad $\mathbf{M} : \mathrm{LIM} \to \mathrm{LIM}$

Clearly, we get $[\![-]\!]_{\mathscr{C}} : \mathrm{LIM} \to \mathrm{Cat}$ for any category $\mathscr{C}$:

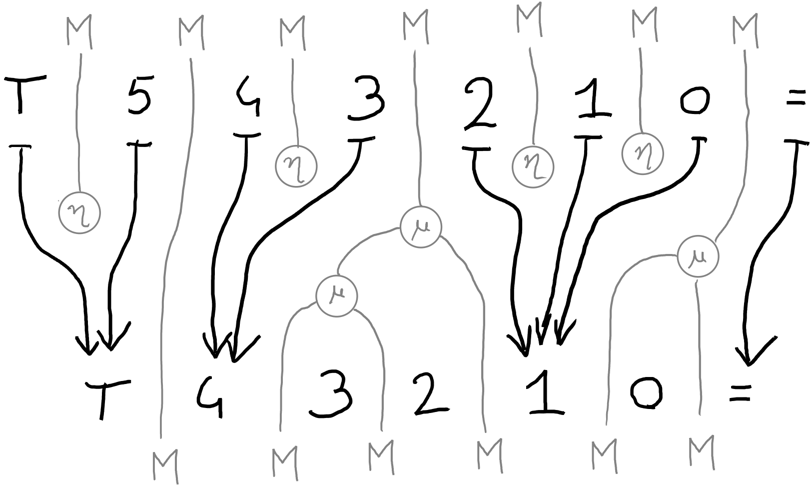
- $\mathbf{C} \mapsto \mathscr{C}$,
- $\mathbf{M} \mapsto \mathrm{Psh}$.

Main theorem (formal proof WIP)

$$\begin{array}{ccccc} \mathrm{DoR} & \cong & \mathrm{LIM} & \to & \mathrm{Cat} \\ p & \mapsto & \mathbf{M}^{p+2}(\mathbf{C}) & \mapsto & \mathrm{Psh}^{p+2}(\mathscr{C}) \end{array}$$


T 5 4 3 2 1 0 =
T 4 3 2 1 0 =

Sketch of Proof



Conclusion

Implications of main theorem:

- **Degrees of Relatedness*** can serve as an **internal language** for **multilevel type theory**,
- A model for **parametricity, irrelevance, . . .** found **in the wild**.

To do:

- Formalize proof.
- Study **discreteness** and **internal parametricity** in this setting.
- Directify :-)

*Or a reasonable adaptation.

Conclusion

Implications of main theorem:

- **Degrees of Relatedness*** can serve as an **internal language** for **multilevel type theory**,
- A model for **parametricity, irrelevance, . . .** found **in the wild**.

To do:

- Formalize proof.
- Study **discreteness** and **internal parametricity** in this setting.
- Directify :-)

*Or a reasonable adaptation.

Conclusion

Implications of main theorem:

- **Degrees of Relatedness*** can serve as an **internal language** for **multilevel type theory**,
- A model for **parametricity, irrelevance, . . .** found **in the wild**.

To do:

- Formalize proof.
- Study **discreteness** and **internal parametricity** in this setting.
- Directify :-)

*Or a reasonable adaptation.

Conclusion

Implications of main theorem:

- **Degrees of Relatedness*** can serve as an **internal language** for **multilevel type theory**,
- A model for **parametricity, irrelevance, . . .** found **in the wild**.

To do:

- Formalize proof.
- Study **discreteness** and **internal parametricity** in this setting.
- Directify :-)

*Or a reasonable adaptation.

Conclusion

Implications of main theorem:

- **Degrees of Relatedness*** can serve as an **internal language** for **multilevel type theory**,
- A model for **parametricity, irrelevance, . . .** found **in the wild**.

To do:

- Formalize proof.
- Study **discreteness** and **internal parametricity** in this setting.
- Directify :-)

*Or a reasonable adaptation.

Thanks!

Questions?