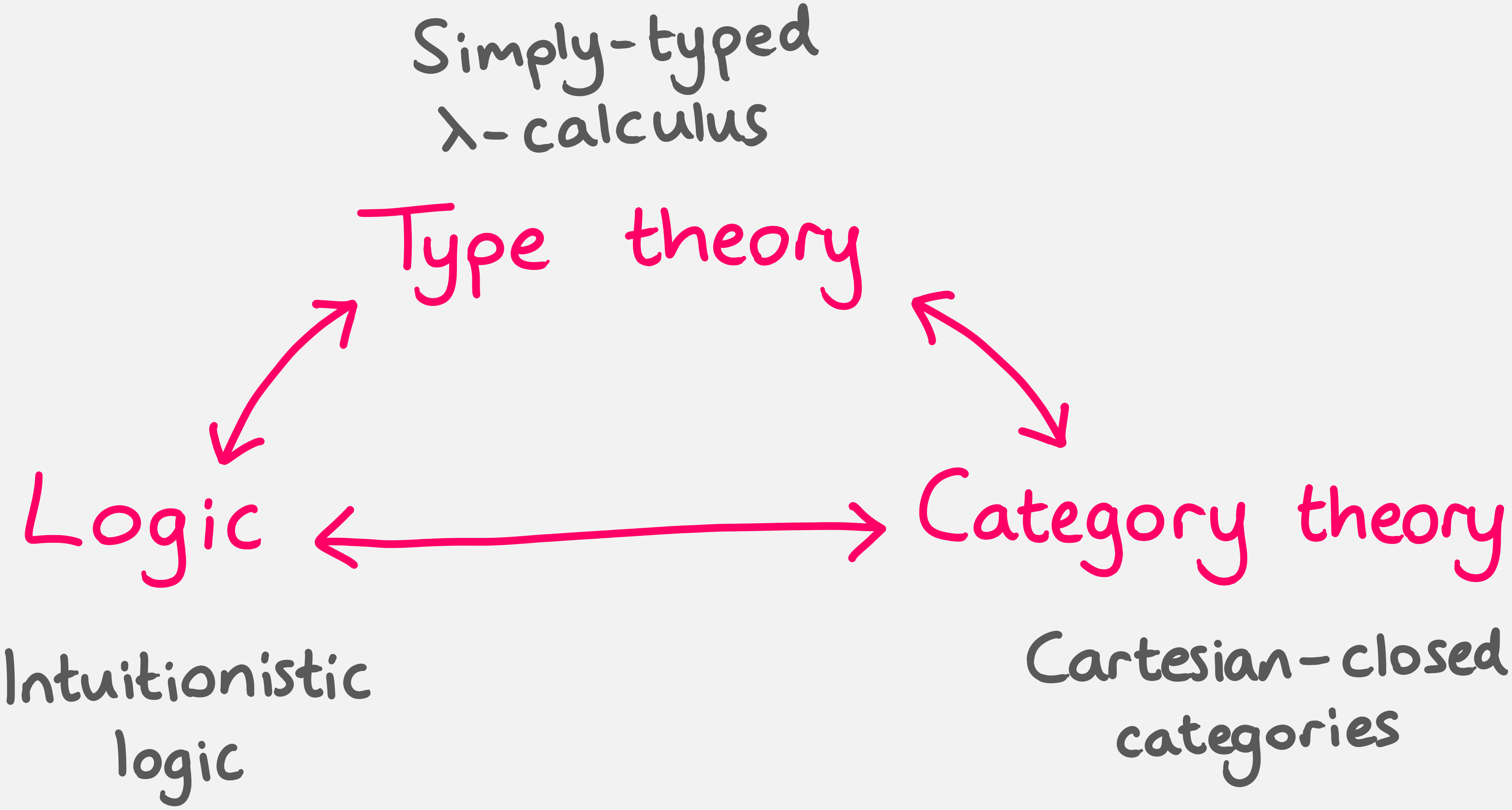
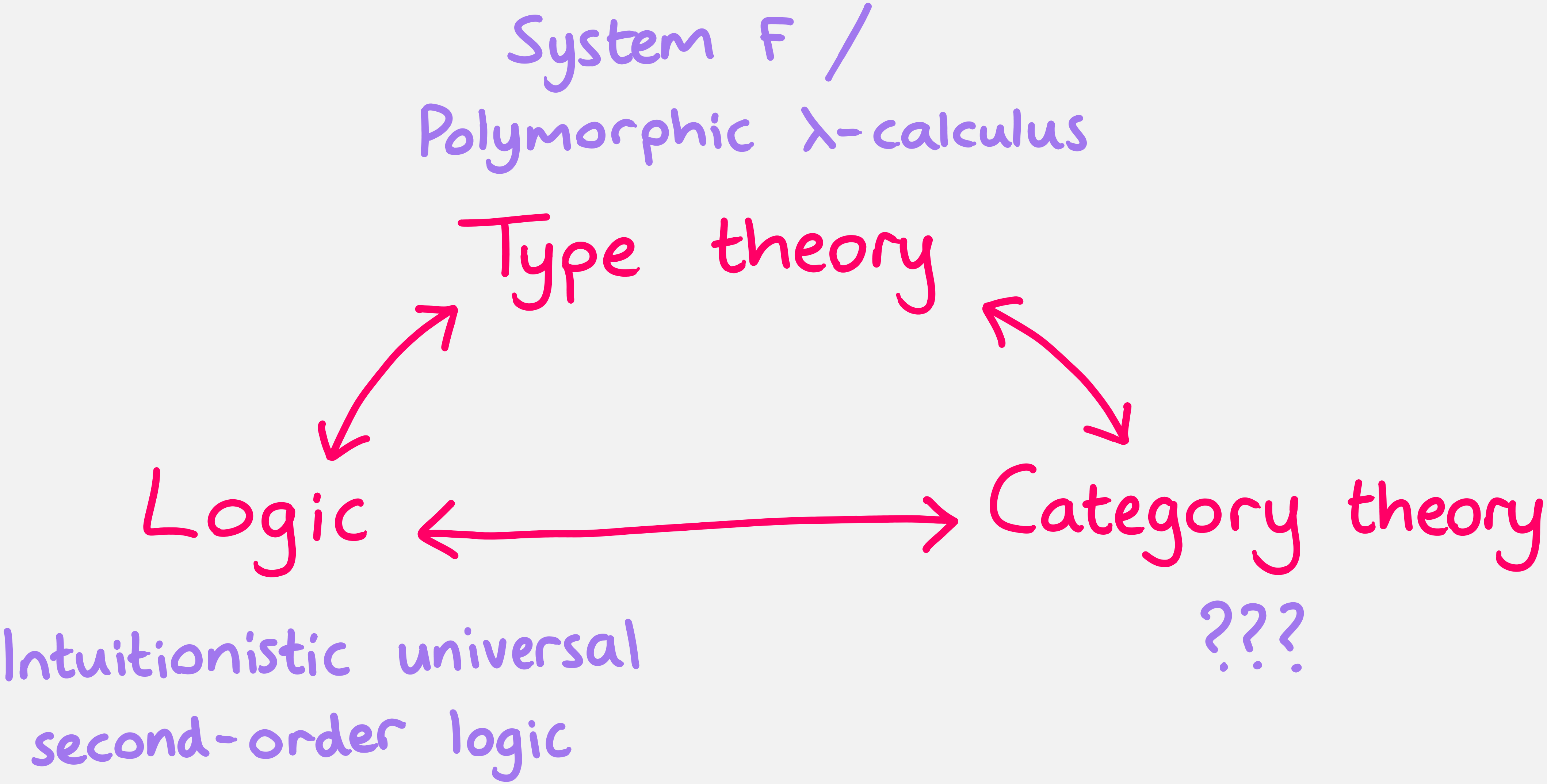
A 2-dimensional perspective on polymorphism

Nathanael Arkor (Masaryk University)

(ICE-TCS Seminar 11.2022)


Simply-typed
λ-calculus
Type theory
Logic
Category theory
Intuitionistic
logic
Cartesian-closed
categories


System F /
Polymorphic λ-calculus
Type theory
Logic
Category theory
Intuitionistic universal
second-order logic
???

Variable binding

In a type theory, terms may abstract over terms.

We say that a type theory
- is polymorphic if terms may abstract over types;
- is dependent if types may abstract over terms;
- admits (second-order) type constructors if types may abstract over types.

Variable binding

In a type theory, terms may abstract over terms.

We say that a type theory
- is polymorphic if terms may abstract over types;
- is dependent if types may abstract over terms;
- admits (second-order) type constructors if types may abstract over types.

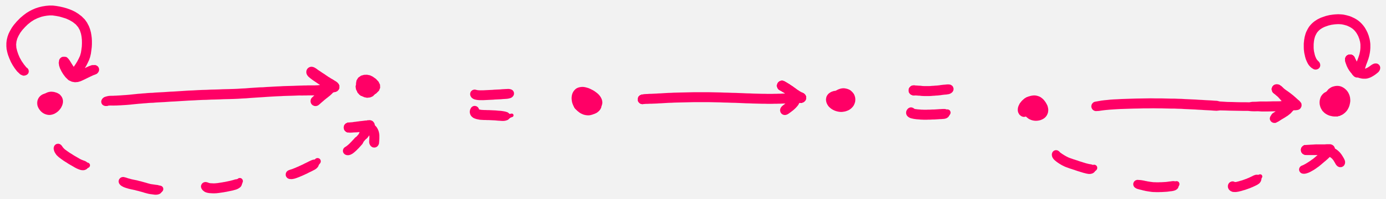
Recap

Multisorted universal algebra $\sim$ Cartesian categories

Categories

Informally, a category is a directed reflexive multigraph in which

subject to



Categories

A **small category** comprises

- a set $|\mathcal{C}|$ of objects;
- for each pair $X, Y \in |\mathcal{C}|$, a set $\mathcal{C}(X,Y)$ of morphisms;
- for each $X \in |\mathcal{C}|$, an identity morphism $1_X \in \mathcal{C}(X,X)$;
- for each $f \in \mathcal{C}(X,Y)$ and $g \in \mathcal{C}(Y,Z)$, a composite $f;g \in \mathcal{C}(X,Z)$;

such that composition is unital and associative.

Cartesian categories

A category $\mathscr{C}$ is **cartesian** when, for each pair $X, Y \in |\mathscr{C}|$, there is a specified object $X \times Y$ and morphisms

$$\begin{tikzcd} & X \times Y \arrow[dl, "\pi_1"'] \arrow[dr, "\pi_2"] & \\ X & & Y \end{tikzcd}$$

through which every span over (X, Y) factors uniquely; and an object 1 for which each object $X \in |\mathscr{C}|$ has a unique morphism $X \xrightarrow{\langle\rangle} 1$.

Categorical semantics

The idea relating type theory to category theory is to interpret **contexts** as **objects**, **types** as **singleton contexts**, and **terms** as morphisms (and vice versa).

Categorical semantics

The idea relating type theory to category theory is to interpret contexts as objects, types as singleton contexts, and terms as morphisms (and vice versa).

Different type theoretic structure then corresponds to analogous category theoretic structure.

Algebraic structure is cartesian structure

Fix a set S of **sorts**, and consider a cartesian category $\mathcal{C}$ whose set of objects is given by finite cartesian products of sorts.

We may interpret a morphism

$$A_1 \times \cdots \times A_n \longrightarrow B_1 \times \cdots \times B_m$$

as a tuple of terms

$$\{a_1 : A_1, \ldots, a_n : A_n \vdash t_i : B_i\}_{1 \leq i \leq m}$$

The identity morphism on $A_1 \times \cdots \times A_n$ is given by variable projection:

$$\{a_1 : A_1, \ldots, a_n : A_n \vdash a_i : A_i\}_{1 \leq i \leq n}$$

The composition of

$$A_1 \times \cdots \times A_n \xrightarrow{\{t_i\}_i} B_1 \times \cdots \times B_m \xrightarrow{\{s_j\}_j} C_1 \times \cdots \times C_l$$

is given by substitution:

$$\{a_1 : A_1, \ldots, a_n : A_n \vdash s_j[b_i \mapsto t_i]_{1 \leq i \leq m}\}_{1 \leq j \leq l}$$

We may view a term

$$a_1 : A_1, \ldots, a_n : A_n \vdash t : B$$

equivalently as an n-ary operator

$$t : (A_1, \ldots, A_n) \longrightarrow B$$

and in this way we can set up a correspondence

cartesian categories

$\Longleftrightarrow$

multisorted universal algebras

(AKA simply-typed pairing calculus)

An example

A **monoid** is expressed by a cartesian category whose objects are generated by a single sort M, and whose morphisms are generated by

$$M \times M \xrightarrow{\otimes} M$$

$$1 \xrightarrow{e} M$$

Subject to

$$\begin{tikzcd}
M \times M \times M \arrow[r, "\otimes \times 1_M"] \arrow[d, "1_M \times \otimes"'] & M \times M \arrow[d, "\otimes"] \\
M \times M \arrow[r, "\otimes"] & M
\end{tikzcd}
\qquad
\begin{tikzcd}
1 \times M \arrow[r, "e \times 1_M"] \arrow[dr, "\pi_2"'] & M \otimes M \arrow[d] & M \times 1 \arrow[l, "1_M \times e"'] \arrow[dl, "\pi_1"] \\
& M &
\end{tikzcd}$$

Algebra to type theory

To express variable-binding structure, such as the λ-abstraction operator of the λ-calculus, we require further structure. In particular, we observe that

$$\frac{\Gamma, x : A \vdash t : B \quad (1)}{\Gamma \vdash \lambda(x : A).\, t : \mathrm{Fn}(A, B) \quad (2)}$$

together with application sets up a bijection between terms of the form (1) and (2).

Cartesian closure

Categorically, this structure is expressed via **adjunction**, which is a relationship between two kinds of morphism.

In particular, a cartesian category is **closed** if $(-) \times A$ has a right adjoint $(-)^A$ for each object A, which means that

$$\mathcal{C}(\Gamma \times A, B) \cong \mathcal{C}(\Gamma, B^A)$$

natural in Γ and B.

What about algebraic structure on types?

In a simple type theory, each term has an associated **context** of **term variables**, and a **type**.

$$\Gamma \vdash t : B$$

We shall consider a setting in which each type has an associated **context** of **type variables**, and a **kind**.

$$\Xi \vdash A : \mathcal{U}$$

In this setting, kinds and types have the same structure that types and terms had: that is, they form a cartesian category.

For instance, in the simply-typed λ-calculus, there is a single (implicit) kind $\mathcal{U}$ and operations

$$\mathcal{U} \times \mathcal{U} \xrightarrow{\text{Prod}} \mathcal{U}$$

$$1 \xrightarrow{\text{Unit}} \mathcal{U}$$

$$\mathcal{U} \times \mathcal{U} \xrightarrow{\text{Fn}} \mathcal{U}$$

What about terms?

Since types are only well-formed in a context of type variables, terms must have an associated context of type variables and term variables.

$$\Xi \mid \Gamma \vdash t : B$$

Thus, for each type context Ξ, we have an associated cartesian category of types and terms in context Ξ.

How can we collect this data into a single structure that captures the relationship between these categories?

Hyperdoctrines

The traditional answer to this question is the notion of **hyperdoctrine**, which is a **contravariant functor** from a cartesian category of kinds and types to the category of cartesian categories.

$$\mathrm{Tm} : \mathrm{Ty}^{op} \longrightarrow \mathrm{CartCat}$$

Then $\mathrm{Tm}(\Xi)$ is the category of types and terms in type context Ξ.

However, this formalism is conceptually dissatisfying for several reasons.

- Term-in-term and type-in-type substitution are interpreted as composition, but type-in-term substitution is interpreted as functorial action.
- The definition as is only permits a single kind.
- The definition is not amenable to generalisation to linear kinds.

An alternative perspective

Start with a cartesian category $\mathcal{C}$ of kinds and types. We consider a notion of **2-cell** between morphisms of $\mathcal{C}$.

$$\begin{tikzcd}
\Xi \arrow[rr, bend left, "A"] \arrow[rr, bend right, "B"'] \arrow[rr, phantom, "\Downarrow t"] & & \mathcal{U}
\end{tikzcd}$$

This represents a term

$$\Xi \mid a : A \vdash t : B \quad \left(\text{where } \begin{array}{l} \Xi \vdash A : \mathcal{U} \\ \Xi \vdash B : \mathcal{U} \end{array}\right)$$

Type-in-term substitution

Type-in-type substitution is given by composition of morphisms (or 1-cells).

$$X \longrightarrow Y \longrightarrow Z \quad \longmapsto \quad X \longrightarrow Z$$

Term-in-term substitution is given by composition of 2-cells.

$$\begin{tikzcd} X \arrow[r, bend left=60, "s", ""{name=S, below}] \arrow[r, "t" description, ""{name=T}] \arrow[r, bend right=60, "u"', ""{name=U, above}] & Y \arrow[Rightarrow, from=S, to=T] \arrow[Rightarrow, from=T, to=U] \end{tikzcd} \quad \longmapsto \quad \begin{tikzcd} X \arrow[r, bend left=50, "s", ""{name=S2, below}] \arrow[r, bend right=50, "u"', ""{name=U2, above}] & Y \arrow[Rightarrow, from=S2, to=U2] \end{tikzcd}$$

What about type-in-term substitution?

In other words, suppose we have a term in a context with a type variable, and we wish to substitute a concrete type for the type variable.

$$\frac{\Xi, \alpha : \mathcal{U} \vdash B : \mathcal{U} \qquad \Xi, \alpha : \mathcal{U} \mid \Gamma \vdash t : B \qquad \Xi \vdash A : \mathcal{U}}{\Xi \mid \Gamma[\alpha \mapsto A] \vdash t[\alpha \mapsto A] : B[\alpha \mapsto A]}$$

How do we interpret this categorically?

Left-whiskering

Given

$$\begin{tikzcd} \Xi \arrow[r, "\langle 1, A \rangle"] & \Xi \times \mathcal{U} \arrow[r, bend left=50, "\Gamma"{name=U}] \arrow[r, bend right=50, "B"'{name=D}] & \mathcal{U} \arrow[from=U, to=D, phantom, "\Downarrow t"] \end{tikzcd}$$

we want a 2-cell

$$\begin{tikzcd} \Xi \arrow[r, bend left=50, "{\Gamma[\alpha \mapsto A]}"{name=U}] \arrow[r, bend right=50, "{B[\alpha \mapsto A]}"'{name=D}] & \mathcal{U} \arrow[from=U, to=D, phantom, "{\Downarrow t[\alpha \mapsto A]}"] \end{tikzcd}$$

Left-sesquicategories

A **left-sesquicategory** comprises

- a category $\mathcal{C}$;
- for each pair $X, Y \in |\mathcal{C}|$, a category structure on $\mathcal{C}(X,Y)$, whose morphisms we call 2-cells;
- for each $X \xrightarrow{f} Y$ and $\begin{tikzcd} Y \arrow[r, bend left, "g"] \arrow[r, bend right, "g'"'] \arrow[r, phantom, "\Downarrow\gamma"] & Z \end{tikzcd}$, a whiskering 2-cell $\begin{tikzcd} X \arrow[r, bend left, "f;g"] \arrow[r, bend right, "f;g'"'] \arrow[r, phantom, "\Downarrow f;\gamma"] & Z \end{tikzcd}$;

such that whiskering respects identities and composition.

Left-sesquicategories

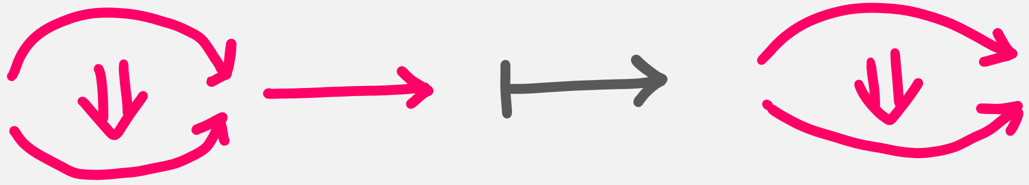
A **left-sesquicategory** comprises

- a category $\mathcal{C}$;
- for each pair $X, Y \in |\mathcal{C}|$, a category structure on $\mathcal{C}(X,Y)$, whose morphisms we call 2-cells;
- for each $X \xrightarrow{f} Y \underset{g'}{\overset{g}{\Downarrow \gamma}} Z$, a whiskering 2-cell
$$\begin{tikzcd} X \arrow[r, bend left, "f;g"] \arrow[r, bend right, "f;g'"'] \arrow[r, phantom, "\Downarrow f;\gamma"] & Z \end{tikzcd};$$

such that whiskering respects identities and composition.

Right-whiskering?

The definition of 2-category is the same as a left-sesquicategory with an additional right-whiskering operation (subject to a compatibility axiom between left- and right-whiskering).



What would this mean for a type theory?

Consider the following diagram:

$$
\begin{tikzcd}
\mathcal{U} \times \mathcal{U} \arrow[rr, bend left, "\mathrm{Prod}"] \arrow[rr, bend right, "\pi_1"'] \arrow[rr, phantom, "\Downarrow \mathrm{proj}_1"] & & \mathcal{U} \arrow[r, "T"] & \mathcal{U}
\end{tikzcd}
$$

Consider the following diagram:

$$\begin{tikzcd} \mathcal{U} \times \mathcal{U} \arrow[rr, bend left, "\mathrm{Prod}"] \arrow[rr, bend right, "\pi_1"'] \arrow[rr, phantom, "\Downarrow \mathrm{proj}_1"] & & \mathcal{U} \arrow[r, "T"] & \mathcal{U} \end{tikzcd}$$

$$\frac{}{\alpha : \mathcal{U},\ \beta : \mathcal{U} \vdash \mathrm{Prod}(\alpha, \beta) : \mathcal{U}}$$

Consider the following diagram:

$$\begin{tikzcd} \mathcal{U} \times \mathcal{U} \arrow[r, bend left, "\mathrm{Prod}"] \arrow[r, bend right, "\pi_1"'] \arrow[r, phantom, "\Downarrow \mathrm{proj}_1"] & \mathcal{U} \arrow[r, "T"] & \mathcal{U} \end{tikzcd}$$

$$\frac{}{\alpha : \mathcal{U}, \beta : \mathcal{U} \vdash \mathrm{Prod}(\alpha, \beta) : \mathcal{U}}$$

$$\frac{}{\alpha : \mathcal{U} \vdash T(\alpha) : \mathcal{U}}$$

Consider the following diagram:

$$\begin{tikzcd}
\mathcal{U} \times \mathcal{U} \arrow[r, bend left, "\mathrm{Prod}"] \arrow[r, bend right, "\pi_1"'] \arrow[r, phantom, "\Downarrow \mathrm{proj}_1"] & \mathcal{U} \arrow[r, "T"] & \mathcal{U}
\end{tikzcd}$$

$$\frac{}{\alpha : \mathcal{U},\ \beta : \mathcal{U} \vdash \mathrm{Prod}(\alpha, \beta) : \mathcal{U}}$$

$$\frac{}{\alpha : \mathcal{U} \vdash T(\alpha) : \mathcal{U}}$$

$$\frac{}{\alpha : \mathcal{U},\ \beta : \mathcal{U},\ p : \mathrm{Prod}(\alpha, \beta) \vdash \mathrm{proj}_1(p) : \alpha}$$

Right-whiskering for the following diagram

$$\begin{tikzcd} \mathcal{U} \times \mathcal{U} \arrow[r, bend left, "\mathrm{Prod}"] \arrow[r, bend right, "\pi_1"'] \arrow[r, phantom, "\Downarrow \mathrm{proj}_1"] & \mathcal{U} \arrow[r, "T"] & \mathcal{U} \end{tikzcd}$$

implies the existence of a 2-cell

$$\begin{tikzcd} \mathcal{U} \times \mathcal{U} \arrow[r, bend left, "T(\mathrm{Prod}(-,-))"] \arrow[r, bend right, "T(\pi_1)"'] \arrow[r, phantom, "\Downarrow"] & \mathcal{U} \end{tikzcd} \qquad \frac{}{\alpha : \mathcal{U},\ \beta : \mathcal{U},\ \tau : T(\mathrm{Prod}(\alpha, \beta)) \vdash \_\_\_\_ : T(\alpha)}$$

which is not necessarily the case.

Interlude: enriched categories

Often the morphisms of a category carry more structure than simply sets. In these instances, it is often worth considering the homs $\mathcal{C}(X,Y)$ of a category to be objects of some other category $\mathcal{V}$.

- When $\mathcal{V} = \mathbf{Set}$, we get ordinary categories.
- When $\mathcal{V} = \mathbb{R}_{\geq 0}$, we get metric spaces.
- When $\mathcal{V} = \mathbf{Cat}$, we get 2-categories.

Left-sesquicategories as enriched categories

The category **Cat** is equipped with ($\alpha\lambda$-normal) right-skew-monoidal structure given by

$$A \otimes B := |A| \times B \qquad J := \mathbb{1}$$

A left-sesquicategory is a $(\mathbf{Cat}, \otimes, J)$-enriched cat.

Left-sesquicategories as enriched categories

The category **Cat** is equipped with ($\alpha\lambda$-normal) right-skew-monoidal structure given by

$$A \otimes B := \underline{|A|} \times B \qquad J := \mathbb{1}$$

no right-whiskering

A left-sesquicategory is a $(\mathbf{Cat}, \otimes, J)$-enriched cat.

Left-sesquicategories as enriched categories

The category $\mathbf{Cat}$ is equipped with ($\alpha\lambda$-normal) right-skew-monoidal structure given by

$$A \otimes B := |A| \times B \qquad J := \mathbb{1}$$

(Abstractly, this structure is induced by the skew-warping given by the monoidal comonad

$$\begin{tikzcd} \mathbf{Set} \arrow[r, hook, shift left] \arrow[r, phantom, "\perp"] & \mathbf{Cat} \arrow[l, shift left, "|-|"] \end{tikzcd}$$

)

A left-sesquicategory is a $(\mathbf{Cat}, \otimes, J)$-enriched cat.

Cartesian structure

The structure of a left-sesquicategory alone is not quite sufficient: we require cartesian products of both kinds and types to be able to form contexts of type and term variables.

Cartesian structure: kinds

A left-sesquicategory is cartesian when equipped with enriched cartesian products (a kind of 2-dimensional limit).

Cartesian structure: kinds

A left-sesquicategory is cartesian when equipped with enriched cartesian products (a kind of 2-dimensional limit).

Given $\Gamma \underset{A_1}{\overset{\Gamma_1}{\rightrightarrows}} \mathcal{U}_1$ with $\Downarrow t_1$, and $\Gamma \underset{A_2}{\overset{\Gamma_2}{\rightrightarrows}} \mathcal{U}_2$ with $\Downarrow t_2$,

we require $\Gamma \underset{\langle A_1, A_2 \rangle}{\overset{\langle \Gamma_1, \Gamma_2 \rangle}{\rightrightarrows}} \mathcal{U}_1 \times \mathcal{U}_2$ with $\Downarrow \langle t_1, t_2 \rangle$ satisfying appropriate laws.

Cartesian structure: types

The skew-monoidal structure on **Cat** lifts to **CartCat**, via the monoidal adjunction

$$\begin{tikzcd} \mathbf{Cat} \arrow[r, shift left, "F"] \arrow[r, phantom, "\perp"] & \mathbf{CartCat} \arrow[l, shift left, "U"] \end{tikzcd}$$

A **homwise-cartesian left-sesquicategory** is a $(\mathbf{CartCat}, \otimes, J)$-enriched category.

Cartesian structure: types

The skew-monoidal structure on Cat lifts to CartCat, via the monoidal adjunction

$$\begin{tikzcd} \mathbf{Cat} \arrow[r, shift left, "F"] \arrow[r, phantom, "\perp"] & \mathbf{CartCat} \arrow[l, shift left, "U"] \end{tikzcd}$$

A homwise-cartesian left-sesquicategory is a $(\mathbf{CartCat}, \otimes, J)$-enriched category.

(Intuitively, each hom-category is cartesian, and this structure interacts nicely with left-whiskering.)

Universal algebra with kinds

Just as cartesian categories model (multisorted) universal algebra, so do cartesian homwise-cartesian left-sesquicategories model (multikinded) universal algebra with kinds (hierarchical algebra).

Universal algebra with kinds

Just as cartesian categories model (multisorted) universal algebra, so do cartesian homwise-cartesian left-sesquicategories model (multikinded) universal algebra with kinds (hierarchical algebra).

If we enrich instead over cartesian-closed categories, we obtain a model for simple type theories.

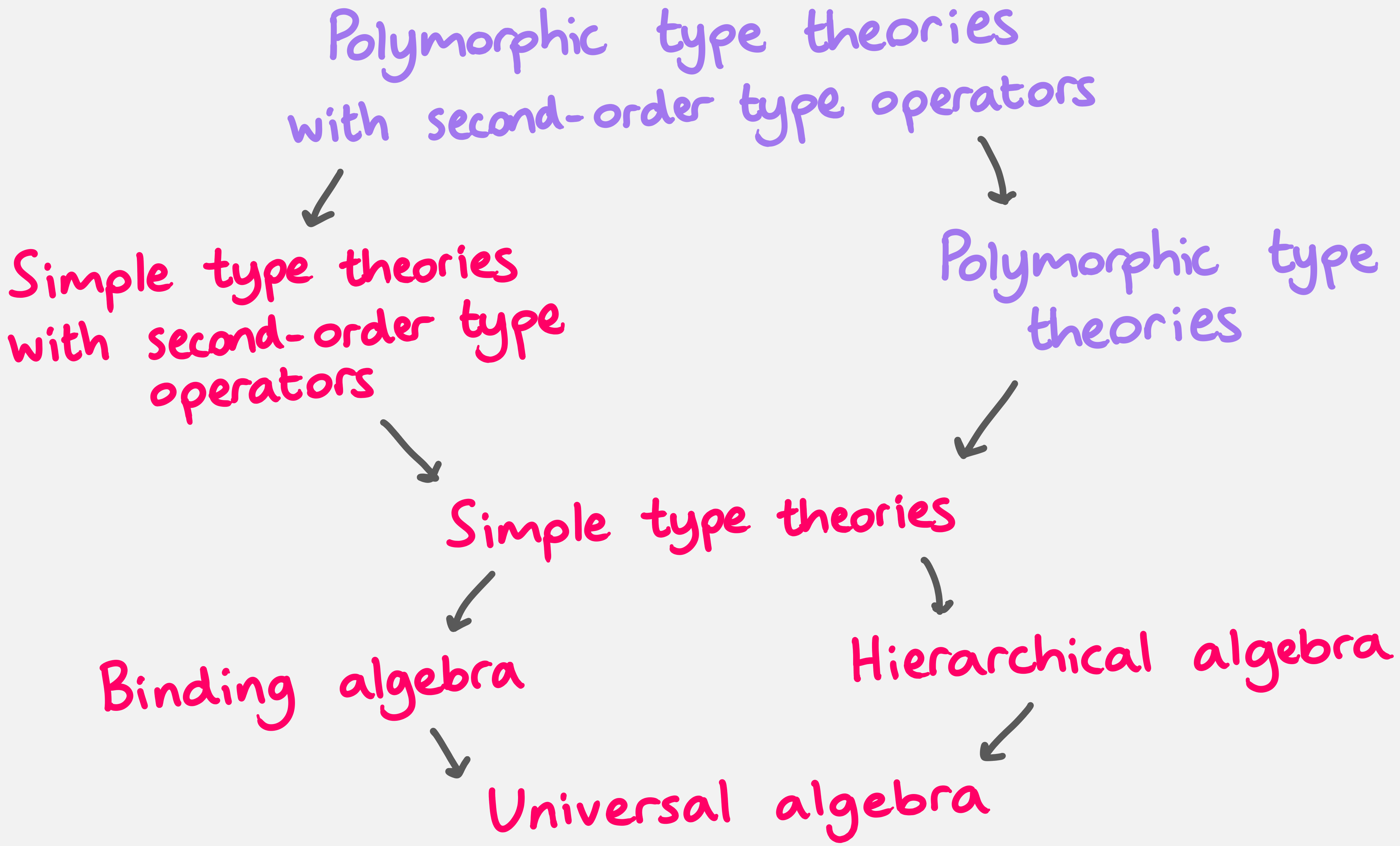
Universal algebra with kinds

Just as cartesian categories model (multisorted) universal algebra, so do cartesian homwise-cartesian left-sesquicategories model (multikinded) universal algebra with kinds (hierarchical algebra).

If we enrich instead over cartesian-closed categories, we obtain a model for simple type theories.

If we ask for enriched cartesian-closure, we obtain a model for second-order type operators.

$$\begin{tikzcd}
\text{Simple type theories with second-order type operators} \arrow[dr] & & \\
& \text{Simple type theories} \arrow[dl] \arrow[dr] & \\
\text{Binding algebra} \arrow[dr] & & \text{Hierarchical algebra} \arrow[dl] \\
& \text{Universal algebra} &
\end{tikzcd}$$


Polymorphic type theories with second-order type operators
Simple type theories with second-order type operators
Polymorphic type theories
Simple type theories
Binding algebra
Hierarchical algebra
Universal algebra

Polymorphism

To capture variable binding (either binding term variables in terms, or type variables in types), we saw that the appropriate categorical structure was that of **adjunction**.

To capture polymorphism, which is type variable binding in terms, we expect that we shall need something similar to adjunction, except simultaneously relating 1-cells and 2-cells.

Impredicative polymorphism

The formation and introduction rules for *impredicative* polymorphism are as follows.

$$\frac{\Xi, \alpha : \mathcal{U} \vdash B : \mathcal{U}}{\Xi \vdash \forall(\alpha : \mathcal{U}).\, B : \mathcal{U}} \; (\forall\text{-FORM})$$

$$\frac{\Xi, \alpha : \mathcal{U} \mid \Gamma \vdash t : B}{\Xi \mid \Gamma \vdash \Lambda(\alpha : \mathcal{U}).\, t : \forall(\alpha : \mathcal{U}).\, B} \; (\forall\text{-INTRO})$$

Impredicative polymorphism

The formation and introduction rules for impredicative polymorphism are as follows.

$$\frac{\Xi, \alpha : \mathcal{U} \vdash B : \mathcal{U}}{\Xi \vdash \forall(\alpha : \mathcal{U})\, B : \mathcal{U}} \;(\forall\text{-FORM})$$

quantifies over itself

$$\frac{\Xi, \alpha : \mathcal{U} \mid \Gamma \vdash t : B}{\Xi \mid \Gamma \vdash \Lambda(\alpha : \mathcal{U}).\, t : \forall(\alpha : \mathcal{U}).\, B} \;(\forall\text{-INTRO})$$

Categorically, given a diagram of this form:

$$\begin{tikzcd}
[\![-]\!] \times \mathcal{U} \arrow[r, bend left, "\Gamma"] \arrow[r, bend right, "B"'] \arrow[r, phantom, "\Downarrow t"] & \mathcal{U}
\end{tikzcd}$$

we want a diagram of this form:

$$\begin{tikzcd}
[\![-]\!] \arrow[r, bend left, "\Gamma"] \arrow[r, bend right, "\forall B"'] \arrow[r, phantom, "\Downarrow \Lambda t"] & \mathcal{U}
\end{tikzcd}$$

(and vice versa).

Local adjunctions

Let $\mathcal{A}$ and $\mathcal{B}$ be left-sesquicategories and

$$\mathcal{A} \underset{R}{\overset{L}{\rightleftarrows}} \mathcal{B}$$

be left-sesquifunctors. A **local adjunction** between L and R is a family of adjunctions

$$\mathcal{B}(LA, B) \underset{R_{A,B}}{\overset{L_{A,B}}{\underset{\perp}{\rightleftarrows}}} \mathcal{A}(A, RB)$$

natural in $A \in \mathcal{A}$, $B \in \mathcal{B}$.

Polymorphism via local adjunction

Let $\mathcal{C}$ be a cartesian left-sesquicategory. Fix an object $\mathcal{U} \in \mathcal{C}$. A local adjunction between

$$\mathcal{C} \underset{1_{\mathcal{C}}}{\overset{(-)\times\mathcal{U}}{\rightleftarrows}} \mathcal{C}$$

is a natural family

$$\mathcal{C}(\Xi \times \mathcal{U}, \Delta)(l_{\Xi,\Delta}\Gamma, B) \cong \mathcal{C}(\Xi, \Delta)(\Gamma, r_{\Xi,\Delta}B)$$

Polymorphism via local adjunction

Let $\mathcal{C}$ be a cartesian left-sesquicategory. Fix an object $\mathcal{U} \in \mathcal{C}$. A local adjunction between

$$\mathcal{C} \underset{1_{\mathcal{C}}}{\overset{(-)\times\mathcal{U}}{\rightleftarrows}} \mathcal{C}$$

is a natural family

$$\mathcal{C}(\Xi \times \mathcal{U}, u)(\Gamma, B) \cong \mathcal{C}(\Xi, u)(\Gamma, \forall_u B)$$

Take $l_{x,y} := (-)\circ \pi_1$, and denote r by $\forall_u$.

Taking $\Delta = \mathcal{U}$, this exactly expresses polymorphism.

Universally quantifiable objects

Let $\mathcal{C}$ be a cartesian left-sesquicategory.
We say that an object $\mathcal{U} \in |\mathcal{C}|$ is **universally quantifiable** when

$$\mathcal{C}(-,-) \xRightarrow{(-)\circ\pi_1} \mathcal{C}(- \times \mathcal{U}, -)$$

has a local right adjoint.

Universally quantifiable objects

Let $\mathcal{C}$ be a cartesian left-sesquicategory.
We say that an object $\mathcal{U} \in |\mathcal{C}|$ is **universally quantifiable** when

$$\mathcal{C}(-,-) \xRightarrow{(-)\circ\pi_1} \mathcal{C}(- \times \mathcal{U}, -)$$

has a local right adjoint.
((Cf. **exponentiable objects** in a cartesian category.)

Universally quantifiable objects

Let $\mathcal{C}$ be a cartesian left-sesquicategory.
We say that an object $\mathcal{U} \in |\mathcal{C}|$ is **universally quantifiable** when

$$\mathcal{C}(-,-) \xRightarrow{(-)\circ\pi_1} \mathcal{C}(- \times \mathcal{U}, -)$$

has a local right adjoint.
((Cf. **exponentiable objects** in a cartesian category.)

$\mathcal{C}$ **admits impredicative polymorphism** when every object is universally quantifiable.

Aside: the Chevalley condition

In the hyperdoctrine approach, to model polymorphism we require a family of adjoints $\forall_\Gamma$ satisfying a Chevalley condition imposing compatibility between polymorphic abstraction and type-in-term substitution.

This is precisely naturality of the local adjunction.

Cartesian homwise-cartesian-closed left-sesquicategories admitting polymorphism

model

System F

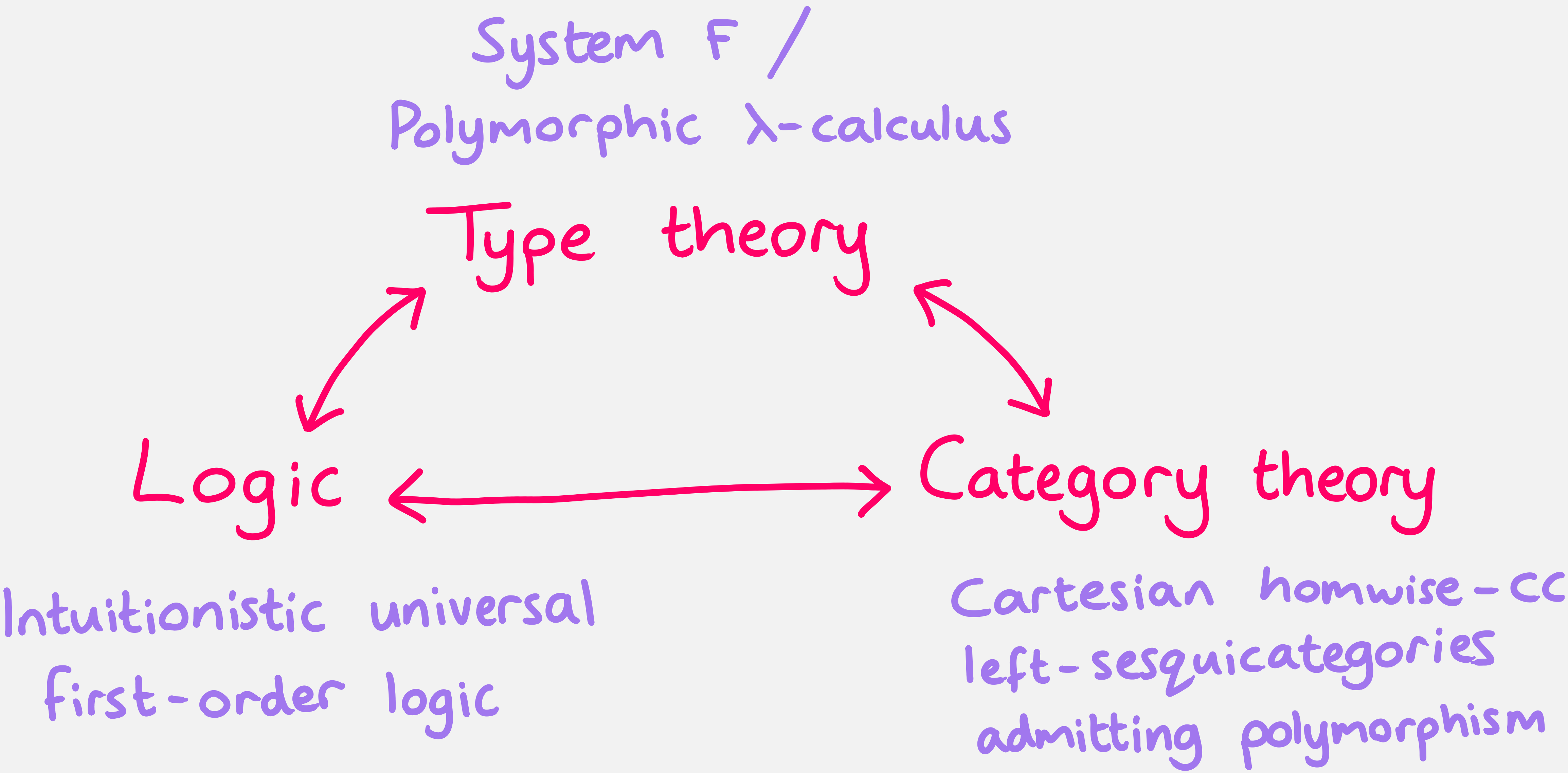
(which captures polymorphic type theories)

Cartesian-closed homwise-cartesian-closed left-sesquicategories admitting polymorphism

model

System $F\omega$

(which captures polymorphic type theories with second-order type operators)


System F /
Polymorphic λ-calculus
Type theory
Logic
Category theory
Intuitionistic universal
first-order logic
Cartesian homwise-CC
left-sesquicategories
admitting polymorphism

Predicativity

Whilst, historically, impredicative polymorphism has proven an important concept, in practice it is difficult to find models of System F (for a similar reason it is difficult to find models of the unityped λ-calculus).

More commonly, we are interested in theories with predicative polymorphism, where terms that abstract over small types necessarily have large types.

Predicative polymorphism

The formation and introduction rules for **predicative** polymorphism are as follows.

$$\frac{\Xi, \alpha : \mathcal{U}_i \vdash B : \mathcal{U}_i}{\Xi \vdash \forall(\alpha : \mathcal{U}_i).\, B : \mathcal{U}_{i+1}} \ (\forall\text{-FORM})$$

abstraction increases the size of a type

$$\frac{\Xi, \alpha : \mathcal{U}_i \mid \Gamma \vdash t : B}{\Xi \mid \Gamma \vdash \Lambda(\alpha : \mathcal{U}_i).\, t : \forall(\alpha : \mathcal{U}_i).\, B} \ (\forall\text{-INTRO})$$

Predicative polymorphism is expressed by **relative local adjunction**. We consider an $\mathbb{N}$-indexed chain of full sub-left-sesquicategories

$$\mathcal{C}_0 \xhookrightarrow{J_0} \mathcal{C}_1 \xhookrightarrow{J_1} \mathcal{C}_2 \xhookrightarrow{J_2} \cdots$$

and distinguished objects $\mathcal{U}_i \in |\mathcal{C}_i|$. We then consider a **relative** (and $\mathbb{N}$-indexed) analogue of a local adjunction.

$$\mathcal{C}_i(\Xi \times \mathcal{U}_i, \Delta)(a \circ \pi_1, b) \cong \mathcal{C}_{i+1}(\underline{J_i}\,\Xi, \underline{J_i}\,\Delta)(a, \forall b)$$

we may only abstract over i-small types

Predicative polymorphism is expressed by **relative local adjunction**. We consider an $\mathbb{N}$-indexed chain of full sub-left-sesquicategories

$$\mathcal{C}_0 \xhookrightarrow{J_0} \mathcal{C}_1 \xhookrightarrow{J_1} \mathcal{C}_2 \xhookrightarrow{J_2} \cdots$$

and distinguished objects $\mathcal{U}_i \in |\mathcal{C}_i|$. We then consider a **relative** (and $\mathbb{N}$-indexed) analogue of a local adjunction.

$$\mathcal{C}_i(\Xi \times \mathcal{U}_i, \Delta)(a \circ \pi_1, b) \cong \mathcal{C}_{i+1}(J_i \Xi, J_i \Delta)(a, \forall b)$$

This more closely reflects the type systems of functional programming languages like Standard ML.

Summary

- Polymorphic type theories may be modelled in a 2-dimensional/enriched categorical framework.
- This approach extends the CCC model of the STLC more naturally than prior models.

Future work

- Relation with dependent types: any dependent type theory with Π-types and a universe should give rise to a predicative polymorphic type theory.