All Concepts are Essentially Algebraic

C.B. Aberlé (she/her)

July 30, 2024

"Algebra is the offer made by the devil to the mathematician. The devil says: I will give you this powerful machine, it will answer any question you like. All you need to do is give me your soul: give up geometry and you will have this marvellous machine." — Michael Francis

Algebraic Theories – Introduction, pt. 1

Many of the most common structures in mathematics admit descriptions entirely in terms of operations and equations governing them.

Algebraic Theories – Introduction, pt. 1

Many of the most common structures in mathematics admit descriptions entirely in terms of operations and equations governing them. Consider, e.g., the following compact description of a group:

$$\mathsf{G}\ \mathsf{Set} \qquad e \in \mathrm{G} \qquad \otimes : \mathrm{G} \times \mathrm{G} \to \mathrm{G} \qquad (-)^{-1} : \mathrm{G} \to \mathrm{G}$$

$$e \otimes x = x = x \otimes e \qquad (x \otimes y) \otimes z = x \otimes (y \otimes z) \qquad x \otimes x^{-1} = e = x^{-1} \otimes x$$

Algebraic Theories – Introduction, pt. 1

Many of the most common structures in mathematics admit descriptions entirely in terms of operations and equations governing them. Consider, e.g., the following compact description of a group:

$$\mathrm{G}\ \mathsf{Set} \qquad e \in \mathrm{G} \qquad \otimes : \mathrm{G} \times \mathrm{G} \to \mathrm{G} \qquad (-)^{-1} : \mathrm{G} \to \mathrm{G}$$

$$e \otimes x = x = x \otimes e \qquad (x \otimes y) \otimes z = x \otimes (y \otimes z) \qquad x \otimes x^{-1} = e = x^{-1} \otimes x$$

Such an algebraic description shows how the essential properties of a structure can be captured in syntactic manipulation of equations built from (typed) variables and function symbols. This in turn reveals many intricate properties and subtle dualities enjoyed by all structures that admit such an algebraic description.

Algebraic Theories – Introduction, pt. 2

Viewed in this light, the “machine” of algebra is nothing more or less than the syntax and semantics of (a form of) type theory.

Algebraic Theories – Introduction, pt. 2

Viewed in this light, the "machine" of algebra is nothing more or less than the syntax and semantics of (a form of) type theory. We could just as easily have written the above in a more type-theoretically-minded way as follows:

$$\mathsf{G}\ \mathsf{Type} \qquad \vdash e : \mathrm{G} \qquad x, y : \mathrm{G} \vdash x \otimes y : \mathrm{G} \qquad x : \mathrm{G} \vdash x^{-1} : \mathrm{G}$$

$$x : \mathrm{G} \vdash e \otimes x \equiv x : \mathrm{G} \qquad x : \mathrm{G} \vdash x \otimes e \equiv x : \mathrm{G}$$

$$x, y, z : \mathrm{G} \vdash (x \otimes y) \otimes z \equiv x \otimes (y \otimes z) : \mathrm{G}$$

$$x : \mathrm{G} \vdash x \otimes x^{-1} \equiv e : \mathrm{G} \qquad x : \mathrm{G} \vdash x^{-1} \otimes x \equiv e : \mathrm{G}$$

Algebraic Theories – Introduction, pt. 2

Viewed in this light, the "machine" of algebra is nothing more or less than the syntax and semantics of (a form of) type theory. We could just as easily have written the above in a more type-theoretically-minded way as follows:

$$\mathsf{G}\ \mathsf{Type} \qquad \vdash e : \mathrm{G} \qquad x, y : \mathrm{G} \vdash x \otimes y : \mathrm{G} \qquad x : \mathrm{G} \vdash x^{-1} : \mathrm{G}$$

$$x : \mathrm{G} \vdash e \otimes x \equiv x : \mathrm{G} \qquad x : \mathrm{G} \vdash x \otimes e \equiv x : \mathrm{G}$$

$$x, y, z : \mathrm{G} \vdash (x \otimes y) \otimes z \equiv x \otimes (y \otimes z) : \mathrm{G}$$

$$x : \mathrm{G} \vdash x \otimes x^{-1} \equiv e : \mathrm{G} \qquad x : \mathrm{G} \vdash x^{-1} \otimes x \equiv e : \mathrm{G}$$

The question, then, is what limitations – if any – such type-theoretic / algebraic syntax for mathematical structures may be subject to, and whether it can encompass all of mathematics, or whether the infernal machine requires a sacrifice of some part of mathematics.

Algebraic Theories – Syntax

We begin with a recap of the theory of ordinary algebraic theories in the manner of Lawvere, recasting their syntax in the type-theoretic mould.

Algebraic Theories – Syntax

We begin with a recap of the theory of ordinary algebraic theories in the manner of Lawvere, recasting their syntax in the type-theoretic mould.

The type theory of algebraic theories has 7 essential judgment forms:

$$\mathcal{S}\ \mathsf{Sig} \qquad \Gamma\ \mathsf{Ctx}_{\mathcal{S}} \qquad \mathrm{A}\ \mathsf{Type}_{\mathcal{S}} \qquad \Gamma \vdash_{\mathcal{S}} a : \mathrm{A}$$

$$\sigma : \Gamma \to_{\mathcal{S}} \Delta \qquad \Gamma \vdash_{\mathcal{S}} a \equiv a' : \mathrm{A} \qquad \sigma \equiv \sigma' : \Gamma \to_{\mathcal{S}} \Delta$$

expressing, respectively, that

- $\mathcal{S}$ is a well-formed signature of some algebraic theory,
- Γ is a well-formed context in the theory of $\mathcal{S}$,
- A is a type in the theory of $\mathcal{S}$,
- a is an element of type A in context Γ (modulo $\mathcal{S}$),
- σ is a valid substitution of Γ-terms for variables in Δ (modulo $\mathcal{S}$),
- a and a' are equal as elements of A in Γ (modulo $\mathcal{S}$),
- and σ and σ' are equal substitutions, modulo $\mathcal{S}$.

Algebraic Theories – Signatures, Contexts & Types

A signature is a description of an (algebraic) theory as a set of axioms:

$$\frac{}{\varnothing\ \mathsf{Sig}} \qquad \frac{\mathcal{S}\ \mathsf{Sig}}{\mathcal{S} \cup \{\mathrm{C}\ \mathsf{Type}\}\ \mathsf{Sig}} \qquad \frac{\mathcal{S}\ \mathsf{Sig} \qquad \Gamma\ \mathsf{Ctx}_{\mathcal{S}} \qquad \mathrm{A}\ \mathsf{Type}_{\mathcal{S}}}{\mathcal{S} \cup \{\Gamma \vdash c : \mathrm{A}\}\ \mathsf{Sig}}$$

$$\frac{\mathcal{S}\ \mathsf{Sig} \qquad \Gamma\ \mathsf{Ctx}_{\mathcal{S}} \qquad \mathrm{A}\ \mathsf{Type}_{\mathcal{S}} \qquad \Gamma \vdash_{\mathcal{S}} a : \mathrm{A} \qquad \Gamma \vdash_{\mathcal{S}} a' : \mathrm{A}}{\mathcal{S} \cup \{\Gamma \vdash a \equiv a' : \mathrm{A}\}\ \mathsf{Sig}}$$

Algebraic Theories – Signatures, Contexts & Types

A signature is a description of an (algebraic) theory as a set of axioms:

$$\frac{}{\varnothing\ \mathsf{Sig}} \qquad \frac{\mathcal{S}\ \mathsf{Sig}}{\mathcal{S} \cup \{\mathrm{C}\ \mathsf{Type}\}\ \mathsf{Sig}} \qquad \frac{\mathcal{S}\ \mathsf{Sig} \qquad \Gamma\ \mathsf{Ctx}_{\mathcal{S}} \qquad \mathrm{A}\ \mathsf{Type}_{\mathcal{S}}}{\mathcal{S} \cup \{\Gamma \vdash c : \mathrm{A}\}\ \mathsf{Sig}}$$

$$\frac{\mathcal{S}\ \mathsf{Sig} \qquad \Gamma\ \mathsf{Ctx}_{\mathcal{S}} \qquad \mathrm{A}\ \mathsf{Type}_{\mathcal{S}} \qquad \Gamma \vdash_{\mathcal{S}} a : \mathrm{A} \qquad \Gamma \vdash_{\mathcal{S}} a' : \mathrm{A}}{\mathcal{S} \cup \{\Gamma \vdash a \equiv a' : \mathrm{A}\}\ \mathsf{Sig}}$$

A context is then a list of variables annotated with types, which may be instantiated as terms:

$$\frac{}{\epsilon\ \mathsf{Ctx}_{\mathcal{S}}} \qquad \frac{\Gamma\ \mathsf{Ctx}_{\mathcal{S}} \qquad \mathrm{A}\ \mathsf{Type}_{\mathcal{S}}}{\Gamma, x : \mathrm{A}\ \mathsf{Ctx}_{\mathcal{S}}} \qquad \frac{}{\Gamma, x : \mathrm{A} \vdash x : \mathrm{A}} \qquad \frac{\Gamma \vdash x : \mathrm{A}}{\Gamma, y : \mathrm{B} \vdash x : \mathrm{A}}$$

Algebraic Theories – Signatures, Contexts & Types

A signature is a description of an (algebraic) theory as a set of axioms:

$$\frac{}{\varnothing\ \mathsf{Sig}} \qquad \frac{\mathcal{S}\ \mathsf{Sig}}{\mathcal{S} \cup \{\mathrm{C}\ \mathsf{Type}\}\ \mathsf{Sig}} \qquad \frac{\mathcal{S}\ \mathsf{Sig} \qquad \Gamma\ \mathsf{Ctx}_{\mathcal{S}} \qquad \mathrm{A}\ \mathsf{Type}_{\mathcal{S}}}{\mathcal{S} \cup \{\Gamma \vdash c : \mathrm{A}\}\ \mathsf{Sig}}$$

$$\frac{\mathcal{S}\ \mathsf{Sig} \qquad \Gamma\ \mathsf{Ctx}_{\mathcal{S}} \qquad \mathrm{A}\ \mathsf{Type}_{\mathcal{S}} \qquad \Gamma \vdash_{\mathcal{S}} a : \mathrm{A} \qquad \Gamma \vdash_{\mathcal{S}} a' : \mathrm{A}}{\mathcal{S} \cup \{\Gamma \vdash a \equiv a' : \mathrm{A}\}\ \mathsf{Sig}}$$

A context is then a list of variables annotated with types, which may be instantiated as terms:

$$\frac{}{\epsilon\ \mathsf{Ctx}_{\mathcal{S}}} \qquad \frac{\Gamma\ \mathsf{Ctx}_{\mathcal{S}} \qquad \mathrm{A}\ \mathsf{Type}_{\mathcal{S}}}{\Gamma, x : \mathrm{A}\ \mathsf{Ctx}_{\mathcal{S}}} \qquad \frac{}{\Gamma, x : \mathrm{A} \vdash x : \mathrm{A}} \qquad \frac{\Gamma \vdash x : \mathrm{A}}{\Gamma, y : \mathrm{B} \vdash x : \mathrm{A}}$$

and the rules for forming types are as follows:

$$\frac{\mathrm{C}\ \mathsf{Type} \in \mathcal{S}}{\mathrm{C}\ \mathsf{Type}_{\mathcal{S}}} \qquad \frac{}{1\ \mathsf{Type}_{\mathcal{S}}} \qquad \frac{\mathrm{A}\ \mathsf{Type} \qquad \mathrm{B}\ \mathsf{Type}}{\mathrm{A} \times \mathrm{B}\ \mathsf{Type}}$$

Algebraic Theories – Terms & Substitutions

To build a term from a constant declared in the signature in an arbitrary context, we may need to *substitute* variables in the context in which that constant is declared for terms in the ambient context:

$$\frac{\Delta \vdash c : A \in \mathcal{S} \qquad \sigma : \Gamma \to_{\mathcal{S}} \Delta}{\Gamma \vdash_{\mathcal{S}} c[\sigma] : A}$$

Algebraic Theories – Terms & Substitutions

To build a term from a constant declared in the signature in an arbitrary context, we may need to *substitute* variables in the context in which that constant is declared for terms in the ambient context:

$$\frac{\Delta \vdash c : \mathrm{A} \in \mathcal{S} \qquad \sigma : \Gamma \to_{\mathcal{S}} \Delta}{\Gamma \vdash_{\mathcal{S}} c[\sigma] : \mathrm{A}}$$

The rules for forming terms of compound types are then as follows:

$$\frac{}{\Gamma \vdash_{\mathcal{S}} \langle\rangle : 1} \qquad \frac{\Gamma \vdash_{\mathcal{S}} a : \mathrm{A} \qquad \Gamma \vdash_{\mathcal{S}} b : \mathrm{B}}{\Gamma \vdash_{\mathcal{S}} \langle a, b\rangle : \mathrm{A} \times \mathrm{B}} \qquad \frac{\Gamma \vdash_{\mathcal{S}} p : \mathrm{A} \times \mathrm{B}}{\Gamma \vdash_{\mathcal{S}} \pi_1(p) : \mathrm{A}} \qquad \frac{\Gamma \vdash_{\mathcal{S}} p : \mathrm{A} \times \mathrm{B}}{\Gamma \vdash_{\mathcal{S}} \pi_2(p) : \mathrm{B}}$$

Algebraic Theories – Terms & Substitutions

To build a term from a constant declared in the signature in an arbitrary context, we may need to *substitute* variables in the context in which that constant is declared for terms in the ambient context:

$$\frac{\Delta \vdash c : \mathrm{A} \in \mathcal{S} \qquad \sigma : \Gamma \to_{\mathcal{S}} \Delta}{\Gamma \vdash_{\mathcal{S}} c[\sigma] : \mathrm{A}}$$

The rules for forming terms of compound types are then as follows:

$$\frac{}{\Gamma \vdash_{\mathcal{S}} \langle\rangle : 1} \qquad \frac{\Gamma \vdash_{\mathcal{S}} a : \mathrm{A} \qquad \Gamma \vdash_{\mathcal{S}} b : \mathrm{B}}{\Gamma \vdash_{\mathcal{S}} \langle a, b\rangle : \mathrm{A} \times \mathrm{B}} \qquad \frac{\Gamma \vdash_{\mathcal{S}} p : \mathrm{A} \times \mathrm{B}}{\Gamma \vdash_{\mathcal{S}} \pi_1(p) : \mathrm{A}} \qquad \frac{\Gamma \vdash_{\mathcal{S}} p : \mathrm{A} \times \mathrm{B}}{\Gamma \vdash_{\mathcal{S}} \pi_2(p) : \mathrm{B}}$$

A substitution is then a list of terms in the domain context whose length and types of constituent terms match that of the codomain context.

$$\frac{}{\bullet : \Gamma \to_{\mathcal{S}} \epsilon} \qquad \frac{\sigma : \Gamma \to_{\mathcal{S}} \Delta \qquad \Gamma \vdash_{\mathcal{S}} a : \mathrm{A}}{\sigma, a : \Gamma \to_{\mathcal{S}} \Delta, x : \mathrm{A}}$$

Algebraic Theories – Terms & Substitutions

To build a term from a constant declared in the signature in an arbitrary context, we may need to *substitute* variables in the context in which that constant is declared for terms in the ambient context:

$$\frac{\Delta \vdash c : \mathrm{A} \in \mathcal{S} \qquad \sigma : \Gamma \to_{\mathcal{S}} \Delta}{\Gamma \vdash_{\mathcal{S}} c[\sigma] : \mathrm{A}}$$

The rules for forming terms of compound types are then as follows:

$$\frac{}{\Gamma \vdash_{\mathcal{S}} \langle\rangle : 1} \qquad \frac{\Gamma \vdash_{\mathcal{S}} a : \mathrm{A} \qquad \Gamma \vdash_{\mathcal{S}} b : \mathrm{B}}{\Gamma \vdash_{\mathcal{S}} \langle a, b\rangle : \mathrm{A} \times \mathrm{B}} \qquad \frac{\Gamma \vdash_{\mathcal{S}} p : \mathrm{A} \times \mathrm{B}}{\Gamma \vdash_{\mathcal{S}} \pi_1(p) : \mathrm{A}} \qquad \frac{\Gamma \vdash_{\mathcal{S}} p : \mathrm{A} \times \mathrm{B}}{\Gamma \vdash_{\mathcal{S}} \pi_2(p) : \mathrm{B}}$$

A substitution is then a list of terms in the domain context whose length and types of constituent terms match that of the codomain context.

$$\frac{}{\bullet : \Gamma \to_{\mathcal{S}} \epsilon} \qquad \frac{\sigma : \Gamma \to_{\mathcal{S}} \Delta \qquad \Gamma \vdash_{\mathcal{S}} a : \mathrm{A}}{\sigma, a : \Gamma \to_{\mathcal{S}} \Delta, x : \mathrm{A}}$$

substitutions may also be composed:

$$\frac{}{\mathsf{id} : \Gamma \to_{\mathcal{S}} \Gamma} \qquad \frac{\sigma : \Gamma \to_{\mathcal{S}} \Delta \qquad \tau : \Delta \to_{\mathcal{S}} \Theta}{\tau \circ \sigma : \Gamma \to_{\mathcal{S}} \Theta}$$

Algebraic Theories – Equations

The relation of equality on terms is required to be an equivalence relation, and a congruence for all constant symbols in the signature:

$$\frac{}{\Gamma \vdash_{\mathcal{S}} a \equiv a : \mathrm{A}} \qquad \frac{\Gamma \vdash_{\mathcal{S}} a' \equiv a : \mathrm{A}}{\Gamma \vdash_{\mathcal{S}} a \equiv a' : \mathrm{A}}$$

$$\frac{\Gamma \vdash_{\mathcal{S}} a \equiv a' : \mathrm{A} \qquad \Gamma \vdash_{\mathcal{S}} a' \equiv a'' : \mathrm{A}}{\Gamma \vdash_{\mathcal{S}} a \equiv a'' : \mathrm{A}} \qquad \frac{\sigma \equiv \sigma' : \Gamma \to_{\mathcal{S}} \Delta}{\Gamma \vdash_{\mathcal{S}} c[\sigma] \equiv c[\sigma'] : \mathrm{A}}$$

Algebraic Theories – Equations

The relation of equality on terms is required to be an equivalence relation, and a congruence for all constant symbols in the signature:

$$\frac{}{\Gamma \vdash_{\mathcal{S}} a \equiv a : \mathrm{A}} \qquad \frac{\Gamma \vdash_{\mathcal{S}} a' \equiv a : \mathrm{A}}{\Gamma \vdash_{\mathcal{S}} a \equiv a' : \mathrm{A}}$$

$$\frac{\Gamma \vdash_{\mathcal{S}} a \equiv a' : \mathrm{A} \qquad \Gamma \vdash_{\mathcal{S}} a' \equiv a'' : \mathrm{A}}{\Gamma \vdash_{\mathcal{S}} a \equiv a'' : \mathrm{A}} \qquad \frac{\sigma \equiv \sigma' : \Gamma \to_{\mathcal{S}} \Delta}{\Gamma \vdash_{\mathcal{S}} c[\sigma] \equiv c[\sigma'] : \mathrm{A}}$$

and satisfies the following laws for tuples

$$\frac{\Gamma \vdash u : 1}{\Gamma \vdash u \equiv \langle\rangle : 1} \qquad \frac{\Gamma \vdash p : \mathrm{A} \times \mathrm{B}}{\Gamma \vdash p \equiv \langle \pi_1(p), \pi_2(p) \rangle : \mathrm{A} \times \mathrm{B}} \qquad \frac{}{\Gamma \vdash \pi_1(\langle a, b \rangle) \equiv a : \mathrm{A}} \qquad \frac{}{\Gamma \vdash \pi_2(\langle a, b \rangle) : \mathrm{B}}$$

Algebraic Theories – Equations

The relation of equality on terms is required to be an equivalence relation, and a congruence for all constant symbols in the signature:

$$\frac{}{\Gamma \vdash_{\mathcal{S}} a \equiv a : \mathrm{A}} \qquad \frac{\Gamma \vdash_{\mathcal{S}} a' \equiv a : \mathrm{A}}{\Gamma \vdash_{\mathcal{S}} a \equiv a' : \mathrm{A}}$$

$$\frac{\Gamma \vdash_{\mathcal{S}} a \equiv a' : \mathrm{A} \qquad \Gamma \vdash_{\mathcal{S}} a' \equiv a'' : \mathrm{A}}{\Gamma \vdash_{\mathcal{S}} a \equiv a'' : \mathrm{A}} \qquad \frac{\sigma \equiv \sigma' : \Gamma \to_{\mathcal{S}} \Delta}{\Gamma \vdash_{\mathcal{S}} c[\sigma] \equiv c[\sigma'] : \mathrm{A}}$$

and satisfies the following laws for tuples

$$\frac{\Gamma \vdash u : 1}{\Gamma \vdash u \equiv \langle\rangle : 1} \qquad \frac{\Gamma \vdash p : \mathrm{A} \times \mathrm{B}}{\Gamma \vdash p \equiv \langle \pi_1(p), \pi_2(p) \rangle : \mathrm{A} \times \mathrm{B}} \qquad \frac{}{\Gamma \vdash \pi_1(\langle a, b \rangle) \equiv a : \mathrm{A}} \qquad \frac{}{\Gamma \vdash \pi_2(\langle a, b \rangle) : \mathrm{B}}$$

with the relation of equality on substitutions being straightforwardly induced from that on terms:

$$\frac{}{\bullet \equiv \bullet : \Gamma \to_{\mathcal{S}} \epsilon} \qquad \frac{\sigma \equiv \sigma' : \Gamma \to_{\mathcal{S}} \Delta \qquad \Gamma \vdash_{\mathcal{S}} a \equiv a' : \mathrm{A}}{\sigma, a \equiv \sigma', a' : \Gamma \to_{\mathcal{S}} \Delta, x : \mathrm{A}} \qquad \frac{}{(\tau, a) \circ \sigma \equiv (\tau \circ \sigma, a[\sigma]) : \Gamma \to_{\mathcal{S}} \Theta}$$

Algebraic Theories – Example

We have already seen an example of an algebraic theory cast in this type-theoretic form: the theory of groups.

Algebraic Theories – Example

We have already seen an example of an algebraic theory cast in this type-theoretic form: the theory of groups. We can then use the type-theoretic apparatus to prove facts about groups.

For instance, if we add the following axioms to our signature for groups, postulating the existence of an additional identity element

$$\vdash e' : \mathrm{G} \qquad x : \mathrm{G} \vdash x \otimes e' \equiv x \qquad x : \mathrm{G} \vdash e' \otimes x \equiv x : \mathrm{G}$$

Algebraic Theories – Example

We have already seen an example of an algebraic theory cast in this type-theoretic form: the theory of groups. We can then use the type-theoretic apparatus to prove facts about groups.

For instance, if we add the following axioms to our signature for groups, postulating the existence of an additional identity element

$$\vdash e' : \mathrm{G} \qquad x : \mathrm{G} \vdash x \otimes e' \equiv x \qquad x : \mathrm{G} \vdash e' \otimes x \equiv x : \mathrm{G}$$

we can straightforwardly derive the uniqueness of identity elements by building a derivation tree according to the rules given above:

$$\dfrac{\dfrac{\dfrac{}{\vdash e \otimes e' \equiv e}}{\vdash e \equiv e \otimes e' : \mathrm{G}} \qquad \dfrac{}{\vdash e \otimes e' \equiv e'}}{\vdash e \equiv e' : \mathrm{G}}$$

Algebraic Theories – Models, pt. 1

Following Lawvere, the syntax of algebraic theories can be interpreted in any category with finite products.

Algebraic Theories – Models, pt. 1

Following Lawvere, the syntax of algebraic theories can be interpreted in any category with finite products.

Let $\mathbb{C}$ be such a category, with terminal object $1 \in \mathrm{Ob}(\mathbb{C})$ and binary product operation $\times : \mathbb{C} \times \mathbb{C} \to \mathbb{C}$, and let $\mathcal{S}$ be the signature of some algebraic theory.

Algebraic Theories – Models, pt. 1

Following Lawvere, the syntax of algebraic theories can be interpreted in any category with finite products.

Let $\mathbb{C}$ be such a category, with terminal object $1 \in \mathrm{Ob}(\mathbb{C})$ and binary product operation $\times : \mathbb{C} \times \mathbb{C} \to \mathbb{C}$, and let $\mathcal{S}$ be the signature of some algebraic theory.

Then a *model* $\mathcal{M}$ of $\mathcal{S}$ in $\mathbb{C}$ consists of:

Algebraic Theories – Models, pt. 1

Following Lawvere, the syntax of algebraic theories can be interpreted in any category with finite products.

Let $\mathbb{C}$ be such a category, with terminal object $1 \in \mathrm{Ob}(\mathbb{C})$ and binary product operation $\times : \mathbb{C} \times \mathbb{C} \to \mathbb{C}$, and let $\mathcal{S}$ be the signature of some algebraic theory.

Then a *model* $\mathcal{M}$ of $\mathcal{S}$ in $\mathbb{C}$ consists of:

1. For each $C\ \mathsf{Type} \in \mathcal{S}$, an object $[\![C]\!]_{\mathcal{M}} \in \mathrm{Ob}(\mathbb{C})$.

Algebraic Theories – Models, pt. 1

Following Lawvere, the syntax of algebraic theories can be interpreted in any category with finite products.

Let $\mathbb{C}$ be such a category, with terminal object $1 \in \mathrm{Ob}(\mathbb{C})$ and binary product operation $\times : \mathbb{C} \times \mathbb{C} \to \mathbb{C}$, and let $\mathcal{S}$ be the signature of some algebraic theory.

Then a *model* $\mathcal{M}$ of $\mathcal{S}$ in $\mathbb{C}$ consists of:

1. For each $\mathrm{C}\ \mathsf{Type} \in \mathcal{S}$, an object $[\![\mathrm{C}]\!]_{\mathcal{M}} \in \mathrm{Ob}(\mathbb{C})$.
    - We then lift this to an assignment of objects in $\mathbb{C}$ to all types, and then all contexts in the theory of $\mathcal{S}$, by setting $[\![1]\!]_{\mathcal{M}} = [\![\epsilon]\!]_{\mathcal{M}} = 1$ and $[\![\mathrm{A} \times \mathrm{B}]\!]_{\mathcal{M}} = [\![\mathrm{A}]\!]_{\mathcal{M}} \times [\![\mathrm{B}]\!]_{\mathcal{M}}$ and $[\![\Gamma, x : \mathrm{A}]\!]_{\mathcal{M}} = [\![\Gamma]\!]_{\mathcal{M}} \times [\![\mathrm{A}]\!]_{\mathcal{M}}$.

Algebraic Theories – Models, pt. 1

Following Lawvere, the syntax of algebraic theories can be interpreted in any category with finite products.

Let $\mathbb{C}$ be such a category, with terminal object $1 \in \mathrm{Ob}(\mathbb{C})$ and binary product operation $\times : \mathbb{C} \times \mathbb{C} \to \mathbb{C}$, and let $\mathcal{S}$ be the signature of some algebraic theory.

Then a *model* $\mathcal{M}$ of $\mathcal{S}$ in $\mathbb{C}$ consists of:

1. For each $\mathsf{C}\ \mathsf{Type} \in \mathcal{S}$, an object $[\![\mathrm{C}]\!]_{\mathcal{M}} \in \mathrm{Ob}(\mathbb{C})$.
   - We then lift this to an assignment of objects in $\mathbb{C}$ to all types, and then all contexts in the theory of $\mathcal{S}$, by setting $[\![1]\!]_{\mathcal{M}} = [\![\epsilon]\!]_{\mathcal{M}} = 1$ and $[\![\mathrm{A} \times \mathrm{B}]\!]_{\mathcal{M}} = [\![\mathrm{A}]\!]_{\mathcal{M}} \times [\![\mathrm{B}]\!]_{\mathcal{M}}$ and $[\![\Gamma, x : \mathrm{A}]\!]_{\mathcal{M}} = [\![\Gamma]\!]_{\mathcal{M}} \times [\![\mathrm{A}]\!]_{\mathcal{M}}$.
2. For each $\Gamma \vdash c : \mathrm{A} \in \mathcal{S}$, a morphism

$$[\![c]\!]^{\Gamma}_{\mathcal{M}} : [\![\Gamma]\!] \to [\![\mathrm{A}]\!] \in \mathbb{C}$$

   We then lift this to an assignment of morphisms in $\mathbb{C}$ to all terms as in the following diagrams:

Algebraic Theories – Models, pt. 2

$$
\begin{tikzcd}
\llbracket \Gamma \rrbracket \arrow[r, "\sigma"] \arrow[dr, dashed, swap, "\llbracket c[\sigma] \rrbracket^{\Gamma}_{A}"] & \llbracket \Delta \rrbracket \arrow[d, "\llbracket c \rrbracket^{\Delta}_{A}"] \\
& \llbracket A \rrbracket
\end{tikzcd}
\qquad
\begin{tikzcd}
\llbracket \Gamma \rrbracket \arrow[r, dashed, "\llbracket \langle\rangle \rrbracket^{\Gamma}_{1}"] & \top
\end{tikzcd}
$$

$$
\begin{tikzcd}
\llbracket \Gamma \rrbracket \arrow[dr, dashed, "\llbracket \langle a,b \rangle \rrbracket^{\Gamma}_{A \times B}"] \arrow[drr, bend left, "\llbracket b \rrbracket^{\Gamma}_{B}"] \arrow[ddr, bend right, swap, "\llbracket a \rrbracket^{\Gamma}_{A}"] & & \\
& \llbracket A \rrbracket \times \llbracket B \rrbracket \arrow[r, "\pi_2"] \arrow[d, "\pi_1"] & \llbracket B \rrbracket \\
& \llbracket A \rrbracket &
\end{tikzcd}
\qquad
\begin{tikzcd}
\llbracket \Gamma \rrbracket \arrow[dr, dashed, "\llbracket p \rrbracket^{\Gamma}_{A \times B}"] \arrow[drr, dashed, bend left, "\llbracket \pi_2(p) \rrbracket^{\Gamma}_{B}"] \arrow[ddr, dashed, bend right, swap, "\llbracket \pi_1(p) \rrbracket^{\Gamma}_{A}"] & & \\
& \llbracket A \rrbracket \times \llbracket B \rrbracket \arrow[r, "\pi_2"] \arrow[d, "\pi_1"] & \llbracket B \rrbracket \\
& \llbracket A \rrbracket &
\end{tikzcd}
$$

and similarly for substitutions.

Algebraic Theories – Models, pt. 2

$$\begin{tikzcd}
\llbracket \Gamma \rrbracket \arrow[r, "\sigma"] \arrow[dr, dashed, "\llbracket c[\sigma] \rrbracket^{\Gamma}_{A}"'] & \llbracket \Delta \rrbracket \arrow[d, "\llbracket c \rrbracket^{\Delta}_{A}"] \\
& \llbracket A \rrbracket
\end{tikzcd}
\qquad
\begin{tikzcd}
\llbracket \Gamma \rrbracket \arrow[r, dashed, "\llbracket \langle\rangle \rrbracket^{\Gamma}_{1}"] & \top
\end{tikzcd}$$

$$\begin{tikzcd}
\llbracket \Gamma \rrbracket \arrow[dr, dashed, "\llbracket \langle a,b \rangle \rrbracket^{\Gamma}_{A\times B}"] \arrow[drr, bend left, "\llbracket b \rrbracket^{\Gamma}_{B}"] \arrow[ddr, bend right, "\llbracket a \rrbracket^{\Gamma}_{A}"'] & & \\
& \llbracket A \rrbracket \times \llbracket B \rrbracket \arrow[r, "\pi_2"] \arrow[d, "\pi_1"'] & \llbracket B \rrbracket \\
& \llbracket A \rrbracket &
\end{tikzcd}
\qquad
\begin{tikzcd}
\llbracket \Gamma \rrbracket \arrow[dr, dashed, "\llbracket p \rrbracket^{\Gamma}_{A\times B}"] \arrow[drr, dashed, bend left, "\llbracket \pi_2(p) \rrbracket^{\Gamma}_{B}"] \arrow[ddr, dashed, bend right, "\llbracket \pi_1(p) \rrbracket^{\Gamma}_{A}"'] & & \\
& \llbracket A \rrbracket \times \llbracket B \rrbracket \arrow[r, "\pi_2"] \arrow[d, "\pi_1"'] & \llbracket B \rrbracket \\
& \llbracket A \rrbracket &
\end{tikzcd}$$

and similarly for substitutions.

3. Such that, for all equations $\Gamma \vdash a \equiv a' : A \in \mathcal{S}$, we have $\llbracket a \rrbracket = \llbracket a' \rrbracket$.

Algebraic Theories – Models, pt. 3

Given models $\mathcal{M}, \mathcal{N}$ of a signature $\mathcal{S}$ in a category $\mathbb{C}$ with finite products, a *homomorphism* from $\mathcal{M}$ to $\mathcal{N}$ consists of:

Algebraic Theories – Models, pt. 3

Given models $\mathcal{M}, \mathcal{N}$ of a signature $\mathcal{S}$ in a category $\mathbb{C}$ with finite products, a *homomorphism* from $\mathcal{M}$ to $\mathcal{N}$ consists of:

1. For each $\mathrm{C}\ \mathsf{Type} \in \mathcal{S}$, a morphism $f_{\mathrm{C}} : [\![\mathrm{C}]\!]_{\mathcal{M}} \to [\![\mathrm{C}]\!]_{\mathcal{N}} \in \mathbb{C}$

Algebraic Theories – Models, pt. 3

Given models $\mathcal{M}, \mathcal{N}$ of a signature $\mathcal{S}$ in a category $\mathbb{C}$ with finite products, a *homomorphism* from $\mathcal{M}$ to $\mathcal{N}$ consists of:

1. For each $\mathrm{C}\ \mathsf{Type} \in \mathcal{S}$, a morphism $f_{\mathrm{C}} : [\![\mathrm{C}]\!]_{\mathcal{M}} \to [\![\mathrm{C}]\!]_{\mathcal{N}} \in \mathbb{C}$
   - We then lift these to morphisms $f_{\mathrm{A}} : [\![\mathrm{A}]\!]_{\mathcal{M}} \to [\![\mathrm{A}]\!]_{\mathcal{N}}$ and $f_{\Gamma} : [\![\Gamma]\!]_{\mathcal{M}} \to [\![\Gamma]\!]_{\mathcal{N}}$, for all types A and contexts Γ in the theory of $\mathcal{S}$, via functoriality of products.

Algebraic Theories – Models, pt. 3

Given models $\mathcal{M}, \mathcal{N}$ of a signature $\mathcal{S}$ in a category $\mathbb{C}$ with finite products, a *homomorphism* from $\mathcal{M}$ to $\mathcal{N}$ consists of:

1. For each $\mathrm{C}\ \mathsf{Type} \in \mathcal{S}$, a morphism $f_{\mathrm{C}} : [\![\mathrm{C}]\!]_{\mathcal{M}} \to [\![\mathrm{C}]\!]_{\mathcal{N}} \in \mathbb{C}$
   - We then lift these to morphisms $f_{\mathrm{A}} : [\![\mathrm{A}]\!]_{\mathcal{M}} \to [\![\mathrm{A}]\!]_{\mathcal{N}}$ and $f_{\Gamma} : [\![\Gamma]\!]_{\mathcal{M}} \to [\![\Gamma]\!]_{\mathcal{N}}$, for all types A and contexts Γ in the theory of $\mathcal{S}$, via functoriality of products.
2. Such that for all $\Gamma \vdash c : \mathrm{A} \in \mathcal{S}$, the following square commutes:

$$\begin{tikzcd}
[\![\Gamma]\!]_{\mathcal{M}} \arrow[r, "{[\![c]\!]_{\mathcal{M}}}"] \arrow[d, "f_{\Gamma}"'] & [\![\mathrm{A}]\!]_{\mathcal{M}} \arrow[d, "f_{\mathrm{A}}"] \\
[\![\Gamma]\!]_{\mathcal{N}} \arrow[r, "{[\![c]\!]_{\mathcal{N}}}"'] & [\![\mathrm{A}]\!]_{\mathcal{N}}
\end{tikzcd}$$

Algebraic Theories – Models, pt. 3

Given models $\mathcal{M}, \mathcal{N}$ of a signature $\mathcal{S}$ in a category $\mathbb{C}$ with finite products, a *homomorphism* from $\mathcal{M}$ to $\mathcal{N}$ consists of:

1. For each $\mathrm{C}\ \mathsf{Type} \in \mathcal{S}$, a morphism $f_{\mathrm{C}} : [\![\mathrm{C}]\!]_{\mathcal{M}} \to [\![\mathrm{C}]\!]_{\mathcal{N}} \in \mathbb{C}$
   - We then lift these to morphisms $f_{\mathrm{A}} : [\![\mathrm{A}]\!]_{\mathcal{M}} \to [\![\mathrm{A}]\!]_{\mathcal{N}}$ and $f_{\Gamma} : [\![\Gamma]\!]_{\mathcal{M}} \to [\![\Gamma]\!]_{\mathcal{N}}$, for all types A and contexts Γ in the theory of $\mathcal{S}$, via functoriality of products.
2. Such that for all $\Gamma \vdash c : \mathrm{A} \in \mathcal{S}$, the following square commutes:

$$\begin{tikzcd}
[\![\Gamma]\!]_{\mathcal{M}} \arrow[r, "[\![c]\!]_{\mathcal{M}}"] \arrow[d, "f_{\Gamma}"'] & [\![\mathrm{A}]\!]_{\mathcal{M}} \arrow[d, "f_{\mathrm{A}}"] \\
[\![\Gamma]\!]_{\mathcal{N}} \arrow[r, "[\![c]\!]_{\mathcal{N}}"'] & [\![\mathrm{A}]\!]_{\mathcal{N}}
\end{tikzcd}$$

Hence for each signature $\mathcal{S}$ and category $\mathbb{C}$ with finite products, we obtain a category $\mathbf{Mod}_{\mathcal{S}}(\mathbb{C})$ of $\mathcal{S}$-models in $\mathbb{C}$ with homomorphisms as morphisms.

Algebraic Theories – Example, cont'd

A model of the theory of groups in a category $\mathbb{C}$ with finite products is precisely a *group object* in $\mathbb{C}$ (in particular, a model of this theory in **Set** is just an ordinary group), consisting of:

Algebraic Theories – Example, cont'd

A model of the theory of groups in a category $\mathbb{C}$ with finite products is precisely a *group object* in $\mathbb{C}$ (in particular, a model of this theory in **Set** is just an ordinary group), consisting of:

$$[\![G]\!] \in \mathrm{Ob}(\mathbb{C}) \qquad [\![e]\!] : 1 \to [\![G]\!] \in \mathbb{C}$$

$$[\![\otimes]\!] : [\![G]\!] \times [\![G]\!] \to [\![G]\!] \in \mathbb{C} \qquad [\![(-)^{-1}]\!] : [\![G]\!] \to [\![G]\!] \in \mathbb{C}$$

Algebraic Theories – Example, cont'd

A model of the theory of groups in a category $\mathbb{C}$ with finite products is precisely a *group object* in $\mathbb{C}$ (in particular, a model of this theory in **Set** is just an ordinary group), consisting of:

$$[\![G]\!] \in \mathrm{Ob}(\mathbb{C}) \qquad [\![e]\!] : 1 \to [\![G]\!] \in \mathbb{C}$$

$$[\![\otimes]\!] : [\![G]\!] \times [\![G]\!] \to [\![G]\!] \in \mathbb{C} \qquad [\![(-)^{-1}]\!] : [\![G]\!] \to [\![G]\!] \in \mathbb{C}$$

such that the following diagrams commute:

$$\begin{tikzcd}
{[\![G]\!]} \arrow[r, "{[\![e]\!]\times[\![G]\!]}"] \arrow[dr, equal] & {[\![G]\!]\times[\![G]\!]} \arrow[d, "{[\![\otimes]\!]}"] & {[\![G]\!]} \arrow[l, "{[\![G]\!]\times[\![e]\!]}"'] \arrow[dl, equal] \\
& {[\![G]\!]} &
\end{tikzcd}
\qquad
\begin{tikzcd}
{[\![G]\!]\times[\![G]\!]\times[\![G]\!]} \arrow[r, "{[\![G]\!]\times[\![\otimes]\!]}"] \arrow[d, "{[\![\otimes]\!]\times[\![G]\!]}"] & {[\![G]\!]\times[\![G]\!]} \arrow[d, "{[\![\otimes]\!]}"] \\
{[\![G]\!]\times[\![G]\!]} \arrow[r, "{[\![\otimes]\!]}"'] & {[\![G]\!]}
\end{tikzcd}$$

$$\begin{tikzcd}
{[\![G]\!]} \arrow[r, "{\langle -,[\![(-)^{-1}]\!]\rangle}"] \arrow[d] & {[\![G]\!]\times[\![G]\!]} \arrow[d, "{[\![\otimes]\!]}"] & {[\![G]\!]} \arrow[l, "{\langle [\![(-)^{-1}]\!],-\rangle}"'] \arrow[d] \\
1 \arrow[r, "{[\![e]\!]}"'] & {[\![G]\!]} & 1 \arrow[l, "{[\![e]\!]}"]
\end{tikzcd}$$

Algebraic Theories – Example, cont'd

A model of the theory of groups in a category $\mathbb{C}$ with finite products is precisely a *group object* in $\mathbb{C}$ (in particular, a model of this theory in **Set** is just an ordinary group), consisting of:

$$[\![G]\!] \in \mathrm{Ob}(\mathbb{C}) \qquad [\![e]\!] : 1 \to [\![G]\!] \in \mathbb{C}$$

$$[\![\otimes]\!] : [\![G]\!] \times [\![G]\!] \to [\![G]\!] \in \mathbb{C} \qquad [\![(-)^{-1}]\!] : [\![G]\!] \to [\![G]\!] \in \mathbb{C}$$

such that the following diagrams commute:

$$\begin{tikzcd}
{[\![G]\!]} \arrow[r, "{[\![e]\!] \times [\![G]\!]}"] \arrow[dr, equal] & {[\![G]\!] \times [\![G]\!]} \arrow[d, dashed, "{[\![\otimes]\!]}"] & {[\![G]\!]} \arrow[l, "{[\![G]\!] \times [\![e]\!]}"'] \arrow[dl, equal] \\
& {[\![G]\!]} &
\end{tikzcd}
\qquad
\begin{tikzcd}
{[\![G]\!] \times [\![G]\!] \times [\![G]\!]} \arrow[r, "{[\![G]\!] \times [\![\otimes]\!]}"] \arrow[d, dashed, "{[\![\otimes]\!] \times [\![G]\!]}"'] & {[\![G]\!] \times [\![G]\!]} \arrow[d, dashed, "{[\![\otimes]\!]}"] \\
{[\![G]\!] \times [\![G]\!]} \arrow[r, "{[\![\otimes]\!]}"'] & {[\![G]\!]}
\end{tikzcd}$$

$$\begin{tikzcd}
{[\![G]\!]} \arrow[r, "{\langle -, [\![(-)^{-1}]\!] \rangle}"] \arrow[d] & {[\![G]\!] \times [\![G]\!]} \arrow[d, dashed, "{[\![\otimes]\!]}"] & {[\![G]\!]} \arrow[l, "{\langle [\![(-)^{-1}]\!], - \rangle}"'] \arrow[d] \\
1 \arrow[r, "{[\![e]\!]}"'] & {[\![G]\!]} & 1 \arrow[l, "{[\![e]\!]}"]
\end{tikzcd}$$

Moreover, a homomorphism of group objects is precisely a group homomorphism.

Algebraic Theories – Representability

Let $\mathbf{Cat}^{\times}$ be the 2-category of categories with finite products, finite product preserving functors between these categories, and natural transformations between these functors. We then have the following:

Algebraic Theories – Representability

Let $\mathbf{Cat}^\times$ be the 2-category of categories with finite products, finite product preserving functors between these categories, and natural transformations between these functors. We then have the following:

Theorem: for any $\mathcal{S}$, the functor $\mathbf{Mod}_{\mathcal{S}}(-) : \mathbf{Cat}^\times \to \mathbf{Cat}$ is *representable*, in that there exists a category $\mathbb{C}_{\mathcal{S}}$ with finite products, such that there is a natural equivalence

$$\mathrm{Hom}_{\mathbf{Cat}^\times}(\mathbb{C}_{\mathcal{S}}, \mathbb{D}) \simeq \mathbf{Mod}_{\mathcal{S}}(\mathbb{D})$$

Algebraic Theories – Representability

Let $\mathbf{Cat}^{\times}$ be the 2-category of categories with finite products, finite product preserving functors between these categories, and natural transformations between these functors. We then have the following:

Theorem: for any $\mathcal{S}$, the functor $\mathbf{Mod}_{\mathcal{S}}(-) : \mathbf{Cat}^{\times} \to \mathbf{Cat}$ is *representable*, in that there exists a category $\mathbb{C}_{\mathcal{S}}$ with finite products, such that there is a natural equivalence

$$\mathrm{Hom}_{\mathbf{Cat}^{\times}}(\mathbb{C}_{\mathcal{S}}, \mathbb{D}) \simeq \mathbf{Mod}_{\mathcal{S}}(\mathbb{D})$$

Proof sketch: we construct $\mathbb{C}_{\mathcal{S}}$ as the category whose objects are *types* in the theory of $\mathcal{S}$, with morphisms given by terms quotiented up to $\mathcal{S}$-provable equality.

Algebraic Theories – Representability

Let $\mathbf{Cat}^{\times}$ be the 2-category of categories with finite products, finite product preserving functors between these categories, and natural transformations between these functors. We then have the following:

Theorem: for any $\mathcal{S}$, the functor $\mathbf{Mod}_{\mathcal{S}}(-) : \mathbf{Cat}^{\times} \to \mathbf{Cat}$ is *representable*, in that there exists a category $\mathbb{C}_{\mathcal{S}}$ with finite products, such that there is a natural equivalence

$$\mathrm{Hom}_{\mathbf{Cat}^{\times}}(\mathbb{C}_{\mathcal{S}}, \mathbb{D}) \simeq \mathbf{Mod}_{\mathcal{S}}(\mathbb{D})$$

Proof sketch: we construct $\mathbb{C}_{\mathcal{S}}$ as the category whose objects are *types* in the theory of $\mathcal{S}$, with morphisms given by terms quotiented up to $\mathcal{S}$-provable equality.

By construction, $\mathbb{C}_{\mathcal{S}}$ has all finite products, and is such that finite product preserving functors $\mathbb{C}_{\mathcal{S}} \to \mathbb{D}$ correspond to models of $\mathcal{S}$ in $\mathbb{D}$, and natural transformations between these correspond to homomorphisms.

Algebraic Theories – Duality

- **Fact 1:** for any context $\Gamma \in \mathrm{Ob}(\mathbb{C}_{\mathcal{S}})$, the contravariant Yoneda embedding $\mathrm{Hom}_{\mathbb{C}_{\mathcal{S}}}(-,=) : \mathbb{C}_{\mathcal{S}}^{op} \to \mathbf{Set}^{\mathbb{C}_{\mathcal{S}}}$ yields, for each context $\Gamma \in \mathrm{Ob}(\mathbb{C}_{\mathcal{S}}^{op})$, a $\mathcal{S}$-model in **Set** given by

$$C \in \mathcal{T}_{\mathcal{S}} \mapsto \mathrm{Hom}_{\mathbb{C}_{\mathcal{S}}}(\Gamma, C)$$

The $\mathcal{S}$-models of this form are precisely the finitely generated free (f.g.f.) models of $\mathcal{S}$. Let $\mathsf{mod}_{\mathcal{S}}$ be the full subcategory of $\mathbf{Mod}_{\mathcal{S}}(\mathbf{Set})$ spanned by these models.

Algebraic Theories – Duality

- **Fact 1:** for any context $\Gamma \in \mathrm{Ob}(\mathbb{C}_{\mathcal{S}})$, the contravariant Yoneda embedding $\mathrm{Hom}_{\mathbb{C}_{\mathcal{S}}}(-,=) : \mathbb{C}_{\mathcal{S}}^{op} \to \mathbf{Set}^{\mathbb{C}_{\mathcal{S}}}$ yields, for each context $\Gamma \in \mathrm{Ob}(\mathbb{C}_{\mathcal{S}}^{op})$, a $\mathcal{S}$-model in **Set** given by

$$C \in \mathcal{T}_{\mathcal{S}} \mapsto \mathrm{Hom}_{\mathbb{C}_{\mathcal{S}}}(\Gamma, C)$$

  The $\mathcal{S}$-models of this form are precisely the finitely generated free (f.g.f.) models of $\mathcal{S}$. Let $\mathsf{mod}_{\mathcal{S}}$ be the full subcategory of $\mathbf{Mod}_{\mathcal{S}}(\mathbf{Set})$ spanned by these models.
- **Fact 2:** The functor $\mathrm{Hom}_{\mathbb{C}_{\mathcal{S}}}(-,=) : \mathbb{C}_{\mathcal{S}}^{op} \to \mathsf{mod}_{\mathcal{S}}$ is an equivalence of categories

$$\mathbb{C}_{\mathcal{S}}^{op} \simeq \mathsf{mod}_{\mathcal{S}}$$

Algebraic Theories – Duality

- **Fact 1:** for any context $\Gamma \in \mathrm{Ob}(\mathbb{C}_{\mathcal{S}})$, the contravariant Yoneda embedding $\mathrm{Hom}_{\mathbb{C}_{\mathcal{S}}}(-,=) : \mathbb{C}_{\mathcal{S}}^{op} \to \mathbf{Set}^{\mathbb{C}_{\mathcal{S}}}$ yields, for each context $\Gamma \in \mathrm{Ob}(\mathbb{C}_{\mathcal{S}}^{op})$, a $\mathcal{S}$-model in **Set** given by

$$C \in \mathcal{T}_{\mathcal{S}} \mapsto \mathrm{Hom}_{\mathbb{C}_{\mathcal{S}}}(\Gamma, C)$$

  The $\mathcal{S}$-models of this form are precisely the finitely generated free (f.g.f.) models of $\mathcal{S}$. Let $\mathsf{mod}_{\mathcal{S}}$ be the full subcategory of $\mathbf{Mod}_{\mathcal{S}}(\mathbf{Set})$ spanned by these models.

- **Fact 2:** The functor $\mathrm{Hom}_{\mathbb{C}_{\mathcal{S}}}(-,=) : \mathbb{C}_{\mathcal{S}}^{op} \to \mathsf{mod}_{\mathcal{S}}$ is an equivalence of categories

$$\mathbb{C}_{\mathcal{S}}^{op} \simeq \mathsf{mod}_{\mathcal{S}}$$

- **Fact 3:** The category $\mathbf{Mod}_{\mathcal{S}}(\mathbf{Set})$ is the free cocompletion of $\mathsf{mod}_{\mathcal{S}}$ under sifted colimits.

Algebraic Theories – Duality

- **Fact 1:** for any context $\Gamma \in \mathrm{Ob}(\mathbb{C}_{\mathcal{S}})$, the contravariant Yoneda embedding $\mathrm{Hom}_{\mathbb{C}_{\mathcal{S}}}(-,=) : \mathbb{C}_{\mathcal{S}}^{op} \to \mathbf{Set}^{\mathbb{C}_{\mathcal{S}}}$ yields, for each context $\Gamma \in \mathrm{Ob}(\mathbb{C}_{\mathcal{S}}^{op})$, a $\mathcal{S}$-model in **Set** given by

$$\mathrm{C} \in \mathcal{T}_{\mathcal{S}} \mapsto \mathrm{Hom}_{\mathbb{C}_{\mathcal{S}}}(\Gamma, \mathrm{C})$$

  The $\mathcal{S}$-models of this form are precisely the finitely generated free (f.g.f.) models of $\mathcal{S}$. Let $\mathsf{mod}_{\mathcal{S}}$ be the full subcategory of $\mathbf{Mod}_{\mathcal{S}}(\mathbf{Set})$ spanned by these models.

- **Fact 2:** The functor $\mathrm{Hom}_{\mathbb{C}_{\mathcal{S}}}(-,=) : \mathbb{C}_{\mathcal{S}}^{op} \to \mathsf{mod}_{\mathcal{S}}$ is an equivalence of categories

$$\mathbb{C}_{\mathcal{S}}^{op} \simeq \mathsf{mod}_{\mathcal{S}}$$

- **Fact 3:** The category $\mathbf{Mod}_{\mathcal{S}}(\mathbf{Set})$ is the free cocompletion of $\mathsf{mod}_{\mathcal{S}}$ under sifted colimits.

Hence types in the theory of $\mathcal{S}$ correspond (contravariantly) to f.g.f. models of $\mathcal{S}$, which are sufficient to generate all other models in **Set**.

Algebraic Theories – Non-Examples

Given a signature $\mathcal{S}$, let $|S| = \{C \mid C\ \mathsf{Type} \in \mathcal{S}\}$ be its set of base types.

Algebraic Theories – Non-Examples

Given a signature $\mathcal{S}$, let $|S| = \{C \mid C\ \mathsf{Type} \in \mathcal{S}\}$ be its set of base types.

Fact 4: the forgetful functor $\mathbf{Mod}_{\mathcal{S}}(\mathbf{Set}) \to \mathbf{Set}^{|S|}$ is

Algebraic Theories – Non-Examples

Given a signature $\mathcal{S}$, let $|S| = \{C \mid C\ \mathsf{Type} \in \mathcal{S}\}$ be its set of base types.

Fact 4: the forgetful functor $\mathbf{Mod}_{\mathcal{S}}(\mathbf{Set}) \to \mathbf{Set}^{|S|}$ is

1. monadic (i.e. has a left adjoint exhibiting $\mathbf{Mod}_{\mathcal{S}}(\mathbf{Set})$ as the category of algebras for the induced monad on $\mathbf{Set}^{|S|}$)

Algebraic Theories – Non-Examples

Given a signature $\mathcal{S}$, let $|S| = \{C \mid C\ \mathsf{Type} \in \mathcal{S}\}$ be its set of base types.

Fact 4: the forgetful functor $\mathbf{Mod}_{\mathcal{S}}(\mathbf{Set}) \to \mathbf{Set}^{|S|}$ is

1. monadic (i.e. has a left adjoint exhibiting $\mathbf{Mod}_{\mathcal{S}}(\mathbf{Set})$ as the category of algebras for the induced monad on $\mathbf{Set}^{|S|}$)
2. finitary (i.e. preserves filtered colimits)

Algebraic Theories – Non-Examples

Given a signature $\mathcal{S}$, let $|S| = \{C \mid C\ \mathsf{Type} \in \mathcal{S}\}$ be its set of base types.

Fact 4: the forgetful functor $\mathbf{Mod}_{\mathcal{S}}(\mathbf{Set}) \to \mathbf{Set}^{|S|}$ is

1. monadic (i.e. has a left adjoint exhibiting $\mathbf{Mod}_{\mathcal{S}}(\mathbf{Set})$ as the category of algebras for the induced monad on $\mathbf{Set}^{|S|}$)
2. finitary (i.e. preserves filtered colimits)

Hence from each signature $\mathcal{S}$ we obtain a finitary monad on $\mathbf{Set}^{|S|}$ whose algebras are models of $\mathcal{S}$. Conversely, given such a monad M, we can recover $\mathbb{C}_{\mathcal{S}}$ as the opposite of the category of finitely generated free algebras of M.

Algebraic Theories – Non-Examples

Given a signature $\mathcal{S}$, let $|\mathrm{S}| = \{\mathrm{C} \mid \mathrm{C}\ \mathsf{Type} \in \mathcal{S}\}$ be its set of base types.

Fact 4: the forgetful functor $\mathbf{Mod}_{\mathcal{S}}(\mathbf{Set}) \to \mathbf{Set}^{|\mathrm{S}|}$ is

1. monadic (i.e. has a left adjoint exhibiting $\mathbf{Mod}_{\mathcal{S}}(\mathbf{Set})$ as the category of algebras for the induced monad on $\mathbf{Set}^{|\mathrm{S}|}$)
2. finitary (i.e. preserves filtered colimits)

Hence from each signature $\mathcal{S}$ we obtain a finitary monad on $\mathbf{Set}^{|\mathrm{S}|}$ whose algebras are models of $\mathcal{S}$. Conversely, given such a monad M, we can recover $\mathbb{C}_{\mathcal{S}}$ as the opposite of the category of finitely generated free algebras of M.

It follows that a category of structures is *not* the category of models for an algebraic theory if it is not finitary monadic over $\mathbf{Set}^{|\mathrm{S}|}$ for any $|\mathrm{S}|$. Some examples of categories that can be shown not to be algebraic in this way are:

Algebraic Theories – Non-Examples

Given a signature $\mathcal{S}$, let $|S| = \{C \mid C\ \mathsf{Type} \in \mathcal{S}\}$ be its set of base types.

Fact 4: the forgetful functor $\mathbf{Mod}_{\mathcal{S}}(\mathbf{Set}) \to \mathbf{Set}^{|S|}$ is

1. monadic (i.e. has a left adjoint exhibiting $\mathbf{Mod}_{\mathcal{S}}(\mathbf{Set})$ as the category of algebras for the induced monad on $\mathbf{Set}^{|S|}$)
2. finitary (i.e. preserves filtered colimits)

Hence from each signature $\mathcal{S}$ we obtain a finitary monad on $\mathbf{Set}^{|S|}$ whose algebras are models of $\mathcal{S}$. Conversely, given such a monad M, we can recover $\mathbb{C}_{\mathcal{S}}$ as the opposite of the category of finitely generated free algebras of M.

It follows that a category of structures is *not* the category of models for an algebraic theory if it is not finitary monadic over $\mathbf{Set}^{|S|}$ for any $|S|$. Some examples of categories that can be shown not to be algebraic in this way are:

- The category **Top** of topological spaces

Algebraic Theories – Non-Examples

Given a signature $\mathcal{S}$, let $|S| = \{C \mid C\ \mathsf{Type} \in \mathcal{S}\}$ be its set of base types.

Fact 4: the forgetful functor $\mathbf{Mod}_{\mathcal{S}}(\mathbf{Set}) \to \mathbf{Set}^{|S|}$ is

1. monadic (i.e. has a left adjoint exhibiting $\mathbf{Mod}_{\mathcal{S}}(\mathbf{Set})$ as the category of algebras for the induced monad on $\mathbf{Set}^{|S|}$)
2. finitary (i.e. preserves filtered colimits)

Hence from each signature $\mathcal{S}$ we obtain a finitary monad on $\mathbf{Set}^{|S|}$ whose algebras are models of $\mathcal{S}$. Conversely, given such a monad M, we can recover $\mathbb{C}_{\mathcal{S}}$ as the opposite of the category of finitely generated free algebras of M.

It follows that a category of structures is *not* the category of models for an algebraic theory if it is not finitary monadic over $\mathbf{Set}^{|S|}$ for any $|S|$. Some examples of categories that can be shown not to be algebraic in this way are:

- The category **Top** of topological spaces
- The category **Cat** of small categories

Dependent Type Theory – Introduction

The definition of a category *feels* much more algebraic than that of a topological space. After all, the equational laws of composition for homsets in a category are just the same as those in the theory of a monoid – which is algebraic – the only difference being that the homsets of a category are *indexed by* or *dependent upon* the set of objects. We can straightforwardly accommodate such dependency by a mere shift in setting from *simple type theory* to *dependent type theory*.

Dependent Type Theory – Introduction

The definition of a category *feels* much more algebraic than that of a topological space. After all, the equational laws of composition for homsets in a category are just the same as those in the theory of a monoid – which is algebraic – the only difference being that the homsets of a category are *indexed by* or *dependent upon* the set of objects. We can straightforwardly accommodate such dependency by a mere shift in setting from *simple type theory* to *dependent type theory*.

- In dependent type theory, we modify the type formation judgment so that it may depend upon a context of variables, and also include a judgment of equality for types, such that equal types must have the same elements:

$$\Gamma \vdash_{\mathcal{S}} \mathrm{A}\ \mathsf{Type} \qquad \Gamma \vdash_{\mathcal{S}} \mathrm{A} \equiv \mathrm{A}'\ \mathsf{Type} \qquad \frac{\Gamma \vdash_{\mathcal{S}} \mathrm{A} \equiv \mathrm{A}'\ \mathsf{Type} \qquad \Gamma \vdash_{\mathcal{S}} a : \mathrm{A}'}{\Gamma \vdash_{\mathcal{S}} a : \mathrm{A}}$$

Dependent Type Theory – Introduction

The definition of a category *feels* much more algebraic than that of a topological space. After all, the equational laws of composition for homsets in a category are just the same as those in the theory of a monoid – which is algebraic – the only difference being that the homsets of a category are *indexed by* or *dependent upon* the set of objects. We can straightforwardly accommodate such dependency by a mere shift in setting from *simple type theory* to *dependent type theory*.

- In dependent type theory, we modify the type formation judgment so that it may depend upon a context of variables, and also include a judgment of equality for types, such that equal types must have the same elements:

$$\Gamma \vdash_{\mathcal{S}} \mathrm{A}\ \mathsf{Type} \qquad \Gamma \vdash_{\mathcal{S}} \mathrm{A} \equiv \mathrm{A}'\ \mathsf{Type} \qquad \frac{\Gamma \vdash_{\mathcal{S}} \mathrm{A} \equiv \mathrm{A}'\ \mathsf{Type} \qquad \Gamma \vdash_{\mathcal{S}} a : \mathrm{A}'}{\Gamma \vdash_{\mathcal{S}} a : \mathrm{A}}$$

- Likewise, declarations of basic types in a signature must now include a context, and using such a type requires a subsitution:

$$\frac{\mathcal{S}\ \mathsf{Sig} \qquad \Gamma\ \mathsf{Ctx}_{\mathcal{S}}}{\mathcal{S} \cup \{\Gamma \vdash \mathrm{C}\ \mathsf{Type}\}\ \mathsf{Sig}} \qquad \frac{\Delta \vdash \mathrm{C}\ \mathsf{Type} \in \mathcal{S} \qquad \sigma : \Gamma \to_{\mathcal{S}} \Delta}{\Gamma \vdash_{\mathcal{S}} \mathrm{C}[\sigma]\ \mathsf{Type}}$$

Dependently Typed Theories – Types

Instead of ordinary pair types $\mathrm{A} \times \mathrm{B}$, we now have *dependent pair types* $\Sigma x : \mathrm{A}.\mathrm{B}$, where B may depend upon $x : \mathrm{A}$

$$\frac{\Gamma \vdash_{\mathcal{S}} \mathrm{A}\ \mathsf{Type} \qquad \Gamma, x : \mathrm{S} \vdash_{\mathcal{S}} \mathrm{B}\ \mathsf{Type}}{\Gamma \vdash_{\mathcal{S}} \Sigma x : \mathrm{A}.\mathrm{B}\ \mathsf{Type}}$$

$$\frac{\Gamma \vdash_{\mathcal{S}} a : \mathrm{A} \qquad \Gamma \vdash_{\mathcal{S}} b : \mathrm{B}[a/x]}{\Gamma \vdash_{\mathcal{S}} \langle a, b \rangle : \Sigma x : \mathrm{A}.\mathrm{B}} \qquad \frac{\Gamma \vdash_{\mathcal{S}} p : \Sigma x : \mathrm{A}.\mathrm{B}}{\Gamma \vdash_{\mathcal{S}} \pi_1(p) : \mathrm{A}} \qquad \frac{\Gamma \vdash_{\mathcal{S}} p : \Sigma x : \mathrm{A}.\mathrm{B}}{\Gamma \vdash_{\mathcal{S}} \pi_2(p) : \mathrm{B}[\pi_1(p)/x]}$$

Dependently Typed Theories – Types

Instead of ordinary pair types $A \times B$, we now have *dependent pair types* $\Sigma x : A.B$, where B may depend upon $x : A$

$$\frac{\Gamma \vdash_{\mathcal{S}} A \ \mathsf{Type} \qquad \Gamma, x : S \vdash_{\mathcal{S}} B \ \mathsf{Type}}{\Gamma \vdash_{\mathcal{S}} \Sigma x : A.B \ \mathsf{Type}}$$

$$\frac{\Gamma \vdash_{\mathcal{S}} a : A \qquad \Gamma \vdash_{\mathcal{S}} b : B[a/x]}{\Gamma \vdash_{\mathcal{S}} \langle a, b \rangle : \Sigma x : A.B} \qquad \frac{\Gamma \vdash_{\mathcal{S}} p : \Sigma x : A.B}{\Gamma \vdash_{\mathcal{S}} \pi_1(p) : A} \qquad \frac{\Gamma \vdash_{\mathcal{S}} p : \Sigma x : A.B}{\Gamma \vdash_{\mathcal{S}} \pi_2(p) : B[\pi_1(p)/x]}$$

Additionally, we now have (extensional) *identity types*, which internalize the judgment of equality:

$$\frac{\Gamma \vdash_{\mathcal{S}} A \ \mathsf{Type} \qquad \Gamma \vdash_{\mathcal{S}} a : A \qquad \Gamma \vdash_{\mathcal{S}} a' : A}{\Gamma \vdash_{\mathcal{S}} a =_A a' \ \mathsf{Type}}$$

$$\frac{\Gamma \vdash_{\mathcal{S}} a \equiv a' : A}{\Gamma \vdash_{\mathcal{S}} \mathsf{refl} : a =_A a'} \qquad \frac{\Gamma \vdash_{\mathcal{S}} e : a =_A a'}{\Gamma \vdash_{\mathcal{S}} a \equiv a' : A} \qquad \frac{\Gamma \vdash_{\mathcal{S}} e : a =_A a'}{\Gamma \vdash_{\mathcal{S}} e \equiv \mathsf{refl} : a =_A a'}$$

Dependently Typed Theories – Semantics, pt. 1

Just as the syntax of algebraic theories could be interpreted in any category with finite products, so the syntax of dependently typed theories can be interpreted in any category $\mathbb{C}$ with *finite limits*.

Dependently Typed Theories – Semantics, pt. 1

Just as the syntax of algebraic theories could be interpreted in any category with finite products, so the syntax of dependently typed theories can be interpreted in any category $\mathbb{C}$ with *finite limits*.

- a Context Γ is interpreted as an object $[\![\Gamma]\!]$ of $\mathbb{C}$

Dependently Typed Theories – Semantics, pt. 1

Just as the syntax of algebraic theories could be interpreted in any category with finite products, so the syntax of dependently typed theories can be interpreted in any category $\mathbb{C}$ with *finite limits*.

- a Context Γ is interpreted as an object $[\![\Gamma]\!]$ of $\mathbb{C}$
- a type A in context Γ is interpreted as a morphism $[\![A]\!] : [\![\Gamma, x : A]\!] \to [\![\Gamma]\!]$

Dependently Typed Theories – Semantics, pt. 1

Just as the syntax of algebraic theories could be interpreted in any category with finite products, so the syntax of dependently typed theories can be interpreted in any category $\mathbb{C}$ with *finite limits*.

- a Context Γ is interpreted as an object $[\![\Gamma]\!]$ of $\mathbb{C}$
- a type A in context Γ is interpreted as a morphism $[\![A]\!] : [\![\Gamma, x : A]\!] \to [\![\Gamma]\!]$
- a term $\Gamma \vdash a : A$ is interpreted as a *section* of $[\![A]\!]$ as in the following diagram:

$$\begin{tikzcd} {[\![\Gamma]\!]} \arrow[r, "{[\![a]\!]}"] \arrow[dr, equal] & {[\![\Gamma, x : A]\!]} \arrow[d, "{[\![A]\!]}"] \\ & {[\![\Gamma]\!]} \end{tikzcd}$$

Dependently Typed Theories – Semantics, pt. 2

- A substitution $\sigma : \Gamma \to \Delta$ is interpreted as a morphism $[\![\Gamma]\!] \to [\![\Delta]\!]$, and the application of σ to a type A is interpreted as the pullback

$$\begin{tikzcd}
[\![\Gamma, x : A[\sigma]]\!] \arrow[r] \arrow[d, "{[\![A[\sigma]]\!]}"'] \arrow[dr, phantom, "\lrcorner", very near start] & [\![\Delta, x : A]\!] \arrow[d, "{[\![A]\!]}"] \\
[\![\Gamma]\!] \arrow[r, "{[\![\sigma]\!]}"'] & [\![\Delta]\!]
\end{tikzcd}$$

Dependently Typed Theories – Semantics, pt. 2

- A substitution $\sigma : \Gamma \to \Delta$ is interpreted as a morphism $[\![\Gamma]\!] \to [\![\Delta]\!]$, and the application of σ to a type A is interpreted as the pullback

$$\begin{tikzcd}
[\![\Gamma, x : A[\sigma]]\!] \arrow[r] \arrow[d, "{[\![A[\sigma]]\!]}"'] \arrow[dr, phantom, "\lrcorner", very near start] & [\![\Delta, x : A]\!] \arrow[d, "{[\![A]\!]}"] \\
[\![\Gamma]\!] \arrow[r, "{[\![\sigma]\!]}"'] & [\![\Delta]\!]
\end{tikzcd}$$

- The type $\Sigma x : A.B$ is interpreted as the composition of $[\![A]\!]$ and $[\![B]\!]$

$$\begin{tikzcd}
[\![\Gamma, x : A, y : B]\!] \arrow[r, "{[\![B]\!]}"] \arrow[rr, bend right, "{[\![\Sigma x{:}A.B]\!]}"'] & [\![\Gamma, x : A]\!] \arrow[r, "{[\![A]\!]}"] & [\![\Gamma]\!]
\end{tikzcd}$$

Dependently Typed Theories – Semantics, pt. 2

- A substitution $\sigma : \Gamma \to \Delta$ is interpreted as a morphism $[\![\Gamma]\!] \to [\![\Delta]\!]$, and the application of σ to a type A is interpreted as the pullback

$$\begin{tikzcd}
[\![\Gamma, x : A[\sigma]]\!] \arrow[r] \arrow[d, "{[\![A[\sigma]]\!]}"'] \arrow[dr, phantom, "\lrcorner", very near start] & [\![\Delta, x : A]\!] \arrow[d, "{[\![A]\!]}"] \\
[\![\Gamma]\!] \arrow[r, "{[\![\sigma]\!]}"'] & [\![\Delta]\!]
\end{tikzcd}$$

- The type $\Sigma x : A.B$ is interpreted as the composition of $[\![A]\!]$ and $[\![B]\!]$

$$\begin{tikzcd}
[\![\Gamma, x : A, y : B]\!] \arrow[r, "{[\![B]\!]}"] \arrow[rr, bend right, "{[\![\Sigma x{:}A.B]\!]}"'] & [\![\Gamma, x : A]\!] \arrow[r, "{[\![A]\!]}"] & [\![\Gamma]\!]
\end{tikzcd}$$

- And the type $a =_A a'$ is interpreted as the equalizer of $[\![a]\!]$ and $[\![a']\!]$

$$\begin{tikzcd}
[\![\Gamma, a =_A a']\!] \arrow[r, "{[\![a =_A a']\!]}"] & [\![\Gamma]\!] \arrow[r, shift left, "{[\![a]\!]}"] \arrow[r, shift right, "{[\![a']\!]}"'] \arrow[dr, equal, bend right] & [\![\Gamma, x : A]\!] \arrow[d, "{[\![A]\!]}"] \\
& & [\![\Gamma]\!]
\end{tikzcd}$$

Dependently Typed Theories – Models

Given a signature $\mathcal{S}$, a *model* $\mathcal{M}$ of $\mathcal{S}$ in $\mathbb{C}$ is a consistent assignment of morphisms $[\![C]\!] : [\![\Gamma, x : C]\!] \to [\![\Gamma]\!]$ to each type $\Gamma \vdash C\ \mathsf{Type} \in \mathcal{S}$, and sections $[\![c]\!] : [\![\Gamma]\!] \to [\![\Gamma, x : A]\!]$ to each term $\Gamma \vdash c : A \in \mathcal{S}$.

Dependently Typed Theories – Models

Given a signature $\mathcal{S}$, a *model* $\mathcal{M}$ of $\mathcal{S}$ in $\mathbb{C}$ is a consistent assignment of morphisms $[\![\mathrm{C}]\!] : [\![\Gamma, x : \mathrm{C}]\!] \to [\![\Gamma]\!]$ to each type $\Gamma \vdash \mathrm{C}\ \mathsf{Type} \in \mathcal{S}$, and sections $[\![c]\!] : [\![\Gamma]\!] \to [\![\Gamma, x : \mathrm{A}]\!]$ to each term $\Gamma \vdash c : \mathrm{A} \in \mathcal{S}$.

A homomorphism of two such models $\mathcal{M}, \mathcal{N}$ consists of:

Dependently Typed Theories – Models

Given a signature $\mathcal{S}$, a *model* $\mathcal{M}$ of $\mathcal{S}$ in $\mathbb{C}$ is a consistent assignment of morphisms $[\![C]\!] : [\![\Gamma, x : C]\!] \to [\![\Gamma]\!]$ to each type $\Gamma \vdash C\ \mathsf{Type} \in \mathcal{S}$, and sections $[\![c]\!] : [\![\Gamma]\!] \to [\![\Gamma, x : A]\!]$ to each term $\Gamma \vdash c : A \in \mathcal{S}$.

A homomorphism of two such models $\mathcal{M}, \mathcal{N}$ consists of:

- for each $\Gamma \vdash C\ \mathsf{Type} \in \mathcal{S}$, a morphism $f_C : [\![\Gamma, x : C]\!]_{\mathcal{M}} \to [\![\Gamma, x : C]\!]_{\mathcal{N}}$ such that the following commutes:

$$\begin{tikzcd}
[\![\Gamma, x : C]\!]_{\mathcal{M}} \arrow[r, "f_C"] \arrow[d, "{[\![C]\!]_{\mathcal{M}}}"'] & [\![\Gamma, x : C]\!]_{\mathcal{N}} \arrow[d, "{[\![C]\!]_{\mathcal{N}}}"] \\
[\![\Gamma]\!]_{\mathcal{M}} \arrow[r, "f_\Gamma"'] & [\![\Gamma]\!]_{\mathcal{N}}
\end{tikzcd}$$

Dependently Typed Theories – Models

Given a signature $\mathcal{S}$, a *model* $\mathcal{M}$ of $\mathcal{S}$ in $\mathbb{C}$ is a consistent assignment of morphisms $[\![C]\!] : [\![\Gamma, x : C]\!] \to [\![\Gamma]\!]$ to each type $\Gamma \vdash C\ \mathsf{Type} \in \mathcal{S}$, and sections $[\![c]\!] : [\![\Gamma]\!] \to [\![\Gamma, x : A]\!]$ to each term $\Gamma \vdash c : A \in \mathcal{S}$.

A homomorphism of two such models $\mathcal{M}, \mathcal{N}$ consists of:

- for each $\Gamma \vdash C\ \mathsf{Type} \in \mathcal{S}$, a morphism $f_C : [\![\Gamma, x : C]\!]_{\mathcal{M}} \to [\![\Gamma, x : C]\!]_{\mathcal{N}}$ such that the following commutes:

$$\begin{tikzcd}
[\![\Gamma, x : C]\!]_{\mathcal{M}} \arrow[r, "f_C"] \arrow[d, "[\![C]\!]_{\mathcal{M}}"'] & [\![\Gamma, x : C]\!]_{\mathcal{N}} \arrow[d, "[\![C]\!]_{\mathcal{N}}"] \\
[\![\Gamma]\!]_{\mathcal{M}} \arrow[r, "f_\Gamma"'] & [\![\Gamma]\!]_{\mathcal{N}}
\end{tikzcd}$$

- and such that for each $\Gamma \vdash c : A \in \mathcal{S}$, the sections $[\![c]\!]_{\mathcal{M}}$ and $[\![c]\!]_{\mathcal{N}}$ commute with the induced morphisms f_A as follows:

$$\begin{tikzcd}
[\![\Gamma]\!]_{\mathcal{M}} \arrow[r, "f_\Gamma"] \arrow[d, dashed, "[\![c]\!]_{\mathcal{M}}"] & [\![\Gamma]\!]_{\mathcal{N}} \arrow[d, dashed, "[\![c]\!]_{\mathcal{N}}"] \\
[\![\Gamma, x : A]\!]_{\mathcal{M}} \arrow[r, "f_A"] & [\![\Gamma, x : A]\!]_{\mathcal{N}}
\end{tikzcd}$$

Dependently Typed Theories – Example, pt. 1

We can now striaghtforwardly write down the axioms for a category in this framework:

Dependently Typed Theories – Example, pt. 1

We can now striaghtforwardly write down the axioms for a category in this framework:

$$\vdash \mathbf{Ob}\ \mathsf{Type} \qquad x, y : \mathbf{Ob} \vdash \mathbf{Hom}(x, y)\ \mathsf{Type} \qquad x : \mathbf{Ob} \vdash \mathrm{id}_x : \mathbf{Hom}(x, x)$$

$$x, y, z : \mathbf{Ob},\ f : \mathbf{Hom}(x, y),\ g : \mathbf{Hom}(y, z) \vdash g \circ f : \mathbf{Hom}(x, z)$$

$$x, y : \mathbf{Ob}, f : \mathbf{Hom}(x, y) \vdash f \circ \mathrm{id}_x \equiv f : \mathbf{Hom}(x, y)$$

$$x, y : \mathbf{Ob}, f : \mathbf{Hom}(x, y) \vdash \mathrm{id}_y \circ f \equiv f : \mathbf{Hom}(x, y)$$

$$w, x, y, z : \mathbf{Ob},\ f : \mathbf{Hom}(w, x),\ g : \mathbf{Hom}(x, y),\ h : \mathbf{Hom}(y, z) \\ \vdash (h \circ g) \circ f \equiv h \circ (g \circ f) : \mathbf{Hom}(w, z)$$

Dependently Typed Theories – Example, pt. 2

A model of this signature in a finitely complete category $\mathbb{C}$ is precisely a *category object* in $\mathbb{C}$, consisting of a span

$$[\![\mathbf{Ob}]\!] \xleftarrow[s]{} [\![\mathbf{Hom}]\!] \xrightarrow[t]{} [\![\mathbf{Ob}]\!]$$

Dependently Typed Theories – Example, pt. 2

A model of this signature in a finitely complete category $\mathbb{C}$ is precisely a *category object* in $\mathbb{C}$, consisting of a span

$$[\![\mathbf{Ob}]\!] \xleftarrow{s} [\![\mathbf{Hom}]\!] \xrightarrow{t} [\![\mathbf{Ob}]\!]$$

together with identity and composition morphisms as in the following diagrams

$$\begin{tikzcd}
 & {[\![\mathbf{Ob}]\!]} \arrow[d, "{[\![\mathrm{id}]\!]}"'] \arrow[dl, equal, bend right] \arrow[dr, equal, bend left] & \\
 & {[\![\mathbf{Hom}]\!]} \arrow[dl, "s"] \arrow[dr, "t"'] & \\
{[\![\mathbf{Ob}]\!]} & & {[\![\mathbf{Ob}]\!]}
\end{tikzcd}
\qquad
\begin{tikzcd}
 & & {[\![\mathbf{Hom}]\!] \times_{t,s} [\![\mathbf{Hom}]\!]} \arrow[dll] \arrow[d, dotted, "{[\![\circ]\!]}"] \arrow[drr] & & \\
{[\![\mathbf{Hom}]\!]} \arrow[d, "s"'] \arrow[drr, "t"] & & {[\![\mathbf{Hom}]\!]} \arrow[dll, "s"'] \arrow[drr, "t"] & & {[\![\mathbf{Hom}]\!]} \arrow[dll, "s"'] \arrow[d, "t"] \\
{[\![\mathbf{Ob}]\!]} & & {[\![\mathbf{Ob}]\!]} & & {[\![\mathbf{Ob}]\!]}
\end{tikzcd}$$

That satisfy the usual associativity and unitality laws.

Dependently Typed Theories – Example, pt. 2

A model of this signature in a finitely complete category $\mathbb{C}$ is precisely a *category object* in $\mathbb{C}$, consisting of a span

$$\llbracket \mathbf{Ob} \rrbracket \xleftarrow[s]{} \llbracket \mathbf{Hom} \rrbracket \xrightarrow[t]{} \llbracket \mathbf{Ob} \rrbracket$$

together with identity and composition morphisms as in the following diagrams

$$\begin{tikzcd}
& \llbracket \mathbf{Ob} \rrbracket \arrow[d, "\llbracket \mathrm{id} \rrbracket"'] \arrow[ddl, bend right, equal] \arrow[ddr, bend left, equal] & \\
& \llbracket \mathbf{Hom} \rrbracket \arrow[dl, "s"] \arrow[dr, "t"'] & \\
\llbracket \mathbf{Ob} \rrbracket & & \llbracket \mathbf{Ob} \rrbracket
\end{tikzcd}
\qquad
\begin{tikzcd}
& & \llbracket \mathbf{Hom} \rrbracket \times_{t,s} \llbracket \mathbf{Hom} \rrbracket \arrow[dl] \arrow[dr] \arrow[d, dotted, "\llbracket \circ \rrbracket"] & & \\
& \llbracket \mathbf{Hom} \rrbracket \arrow[dl, "s"] \arrow[dr, "t"] & \llbracket \mathbf{Hom} \rrbracket \arrow[dll, "s"] \arrow[drr, "t"] & \llbracket \mathbf{Hom} \rrbracket \arrow[dl, "s"] \arrow[dr, "t"] & \\
\llbracket \mathbf{Ob} \rrbracket & & \llbracket \mathbf{Ob} \rrbracket & & \llbracket \mathbf{Ob} \rrbracket
\end{tikzcd}$$

That satisfy the usual associativity and unitality laws.

Moreover, a homomorphism of category objects is precisely a functor between them, as typically defined.

Dependently Typed Theories – Representability & Duality

As before, we can construct $\mathbb{C}_{\mathcal{S}}$ as the category whose objects are equivalence classes of *closed types* in the theory of $\mathcal{S}$, i.e. types $\vdash_{\mathcal{S}} A\ \mathsf{Type}$, up to $\mathcal{S}$-provable equality, and whose morphisms are given by equivalence classes of terms up to $\mathcal{S}$-provable equality.

Dependently Typed Theories – Representability & Duality

As before, we can construct $\mathbb{C}_{\mathcal{S}}$ as the category whose objects are equivalence classes of *closed types* in the theory of $\mathcal{S}$, i.e. types $\vdash_{\mathcal{S}} A\ \mathsf{Type}$, up to $\mathcal{S}$-provable equality, and whose morphisms are given by equivalence classes of terms up to $\mathcal{S}$-provable equality.

By construction, this is a finitely complete category such that finite limit preserving functors out of $\mathbb{C}_{\mathcal{S}}$ are equivalent to models of $\mathcal{S}$, i.e. there is a natural equivalence

$$\mathrm{Hom}_{\mathbf{Lex}}(\mathbb{C}_{\mathcal{S}}, \mathbb{D}) \simeq \mathbf{Mod}_{\mathcal{S}}(\mathbb{D})$$

Dependently Typed Theories – Representability & Duality

As before, we can construct $\mathbb{C}_{\mathcal{S}}$ as the category whose objects are equivalence classes of *closed types* in the theory of $\mathcal{S}$, i.e. types $\vdash_{\mathcal{S}} A\ \mathsf{Type}$, up to $\mathcal{S}$-provable equality, and whose morphisms are given by equivalence classes of terms up to $\mathcal{S}$-provable equality.

By construction, this is a finitely complete category such that finite limit preserving functors out of $\mathbb{C}_{\mathcal{S}}$ are equivalent to models of $\mathcal{S}$, i.e. there is a natural equivalence

$$\mathrm{Hom}_{\mathbf{Lex}}(\mathbb{C}_{\mathcal{S}}, \mathbb{D}) \simeq \mathbf{Mod}_{\mathcal{S}}(\mathbb{D})$$

- **Fact 1:** the contravariant Yoneda embedding $\mathrm{Hom}_{\mathbb{C}_{\mathcal{S}}}(-,=) : \mathbb{C}_{\mathcal{S}}^{op} \to \mathbf{Set}^{\mathbb{C}_{\mathcal{S}}}$ yields, for each closed type $A \in \mathrm{Ob}(\mathbb{C}_{\mathcal{S}}^{op})$, an $\mathcal{S}$-model in **Set**. The $\mathcal{S}$-models of this form are precisely the *finitely presentable* models of $\mathcal{S}$. Let $\mathsf{mod}_{\mathcal{S}}$ be the full subcategory of $\mathbf{Mod}_{\mathcal{S}}(\mathbf{Set})$ spanned by these models.

Dependently Typed Theories – Representability & Duality

As before, we can construct $\mathbb{C}_{\mathcal{S}}$ as the category whose objects are equivalence classes of *closed types* in the theory of $\mathcal{S}$, i.e. types $\vdash_{\mathcal{S}} \mathrm{A}\ \mathsf{Type}$, up to $\mathcal{S}$-provable equality, and whose morphisms are given by equivalence classes of terms up to $\mathcal{S}$-provable equality.

By construction, this is a finitely complete category such that finite limit preserving functors out of $\mathbb{C}_{\mathcal{S}}$ are equivalent to models of $\mathcal{S}$, i.e. there is a natural equivalence

$$\mathrm{Hom}_{\mathbf{Lex}}(\mathbb{C}_{\mathcal{S}}, \mathbb{D}) \simeq \mathbf{Mod}_{\mathcal{S}}(\mathbb{D})$$

- **Fact 1:** the contravariant Yoneda embedding $\mathrm{Hom}_{\mathbb{C}_{\mathcal{S}}}(-,=) : \mathbb{C}_{\mathcal{S}}^{op} \to \mathbf{Set}^{\mathbb{C}_{\mathcal{S}}}$ yields, for each closed type $\mathrm{A} \in \mathrm{Ob}(\mathbb{C}_{\mathcal{S}}^{op})$, an $\mathcal{S}$-model in **Set**. The $\mathcal{S}$-models of this form are precisely the *finitely presentable* models of $\mathcal{S}$. Let $\mathsf{mod}_{\mathcal{S}}$ be the full subcategory of $\mathbf{Mod}_{\mathcal{S}}(\mathbf{Set})$ spanned by these models.
- **Fact 2:** $\mathrm{Hom}_{\mathbb{C}_{\mathcal{S}}}(-,=) : \mathbb{C}_{\mathcal{S}}^{op} \to \mathsf{mod}_{\mathcal{S}}$ is an equivalence of categories.

Dependently Typed Theories – Representability & Duality

As before, we can construct $\mathbb{C}_{\mathcal{S}}$ as the category whose objects are equivalence classes of *closed types* in the theory of $\mathcal{S}$, i.e. types $\vdash_{\mathcal{S}} A\ \mathsf{Type}$, up to $\mathcal{S}$-provable equality, and whose morphisms are given by equivalence classes of terms up to $\mathcal{S}$-provable equality.

By construction, this is a finitely complete category such that finite limit preserving functors out of $\mathbb{C}_{\mathcal{S}}$ are equivalent to models of $\mathcal{S}$, i.e. there is a natural equivalence

$$\mathrm{Hom}_{\mathbf{Lex}}(\mathbb{C}_{\mathcal{S}}, \mathbb{D}) \simeq \mathbf{Mod}_{\mathcal{S}}(\mathbb{D})$$

- **Fact 1:** the contravariant Yoneda embedding $\mathrm{Hom}_{\mathbb{C}_{\mathcal{S}}}(-,=) : \mathbb{C}_{\mathcal{S}}^{op} \to \mathbf{Set}^{\mathbb{C}_{\mathcal{S}}}$ yields, for each closed type $A \in \mathrm{Ob}(\mathbb{C}_{\mathcal{S}}^{op})$, an $\mathcal{S}$-model in **Set**. The $\mathcal{S}$-models of this form are precisely the *finitely presentable* models of $\mathcal{S}$. Let $\mathsf{mod}_{\mathcal{S}}$ be the full subcategory of $\mathbf{Mod}_{\mathcal{S}}(\mathbf{Set})$ spanned by these models.
- **Fact 2:** $\mathrm{Hom}_{\mathbb{C}_{\mathcal{S}}}(-,=) : \mathbb{C}_{\mathcal{S}}^{op} \to \mathsf{mod}_{\mathcal{S}}$ is an equivalence of categories.
- **Fact 3:** The category $\mathbf{Mod}_{\mathcal{S}}(\mathbf{Set})$ is generated from $\mathsf{mod}_{\mathcal{S}}$ under filtered colimits, i.e. it is *locally finitely presentable* (lfp).

All Concepts are Essentially Algebraic?

- It follows that a category of structures is *not* the category of models for a dependently typed theory if that category is not lfp.

All Concepts are Essentially Algebraic?

- It follows that a category of structures is *not* the category of models for a dependently typed theory if that category is not lfp.
- We have already seen that **Cat** is the category of models for a dependently typed theory.

All Concepts are Essentially Algebraic?

- It follows that a category of structures is *not* the category of models for a dependently typed theory if that category is not lfp.
- We have already seen that **Cat** is the category of models for a dependently typed theory.
  - In particular, the standard n-simplices

    $$\{\bullet\} \qquad \{\bullet \to \bullet\} \qquad \{\bullet \to \bullet \to \bullet\} \qquad \ldots$$

    are a generating set for finitely presentable categories under finite colimits, hence for *all* categories under general colimits.

All Concepts are Essentially Algebraic?

- It follows that a category of structures is *not* the category of models for a dependently typed theory if that category is not lfp.
- We have already seen that **Cat** is the category of models for a dependently typed theory.
    - In particular, the standard n-simplices

      $$\{\bullet\} \qquad \{\bullet \to \bullet\} \qquad \{\bullet \to \bullet \to \bullet\} \quad \dots$$

      are a generating set for finitely presentable categories under finite colimits, hence for *all* categories under general colimits.
- However, the category **Top** of topological spaces is *not* lfp.

All Concepts are Essentially Algebraic?

- It follows that a category of structures is *not* the category of models for a dependently typed theory if that category is not lfp.
- We have already seen that **Cat** is the category of models for a dependently typed theory.
  - In particular, the standard n-simplices

    $$\{\bullet\} \qquad \{\bullet \to \bullet\} \qquad \{\bullet \to \bullet \to \bullet\} \quad \ldots$$

    are a generating set for finitely presentable categories under finite colimits, hence for *all* categories under general colimits.
- However, the category **Top** of topological spaces is *not* lfp.
- Has the devil come to claim his due? Must we give up topology for the sake of algebra?

All Concepts are Essentially Algebraic?

- It follows that a category of structures is *not* the category of models for a dependently typed theory if that category is not lfp.
- We have already seen that **Cat** is the category of models for a dependently typed theory.
  - In particular, the standard n-simplices

    $$\{\bullet\} \qquad \{\bullet \to \bullet\} \qquad \{\bullet \to \bullet \to \bullet\} \quad \ldots$$

    are a generating set for finitely presentable categories under finite colimits, hence for *all* categories under general colimits.
- However, the category **Top** of topological spaces is *not* lfp.
- Has the devil come to claim his due? Must we give up topology for the sake of algebra?
- **Thesis:** the failure of a category of structures to be lfp is a sign that we haven't given enough information about *where* those structures live. If we "zoom out" to include not only these structures, but the *mathematical universes* they inhabit, then the resulting theories are essentially algebraic.

The Theory of Finite Limits

We can obtain a theory of categories with finite limits by adding the following axioms to the theory of a category given above:

$$\vdash \top : \mathbf{Ob} \qquad x : \mathbf{Ob} \vdash !_x : \mathbf{Hom}(x, \top) \qquad x : \mathbf{Ob},\ f : \mathbf{Hom}(x, \top) \vdash f \equiv !_x : \mathbf{Hom}(x, \top)$$

$$x, y, z : \mathbf{Ob},\ f : \mathbf{Hom}(x, z),\ g : \mathbf{Hom}(y, z) \vdash x \times_{f,g} y : \mathbf{Ob}$$

$$x, y, z : \mathbf{Ob},\ f : \mathbf{Hom}(x, z),\ g : \mathbf{Hom}(y, z) \vdash \pi_1 : \mathbf{Hom}(x \times_{f,g} y, x)$$

$$x, y, z : \mathbf{Ob},\ f : \mathbf{Hom}(x, z),\ g : \mathbf{Hom}(y, z) \vdash \pi_2 : \mathbf{Hom}(x \times_{f,g} y, y)$$

$$x, y, z : \mathbf{Ob},\ f : \mathbf{Hom}(x, z),\ g : \mathbf{Hom}(y, z) \vdash f \circ \pi_1 \equiv g \circ \pi_2 : \mathbf{Hom}(x \times_{f,g} y, z)$$

$$\begin{aligned} & x, y, z, w : \mathbf{Ob},\ f : \mathbf{Hom}(x, z),\ g : \mathbf{Hom}(y, z), \\ & h : \mathbf{Hom}(w, x),\ k : \mathbf{Hom}(w, y),\ f \circ h =_{\mathbf{Hom}(w,z)} g \circ k \\ & \vdash \langle h, k \rangle : \mathbf{Hom}(w, x \times_{f,g} y) \end{aligned}$$

$$\begin{aligned} & x, y, z, w : \mathbf{Ob},\ f : \mathbf{Hom}(x, z),\ g : \mathbf{Hom}(y, z), \\ & h : \mathbf{Hom}(w, x),\ k : \mathbf{Hom}(w, y),\ f \circ h =_{\mathbf{Hom}(w,z)} g \circ k \\ & \vdash \pi_1 \circ \langle h, k \rangle \equiv h : \mathbf{Hom}(w, x) \end{aligned}$$

$$\begin{aligned} & x, y, z, w : \mathbf{Ob},\ f : \mathbf{Hom}(x, z),\ g : \mathbf{Hom}(y, z), \\ & h : \mathbf{Hom}(w, x),\ k : \mathbf{Hom}(w, y),\ f \circ h =_{\mathbf{Hom}(w,z)} g \circ k \\ & \vdash \pi_2 \circ \langle h, k \rangle \equiv k : \mathbf{Hom}(w, y) \end{aligned}$$

$$\begin{aligned} & x, y, z : \mathbf{Ob},\ f : \mathbf{Hom}(x, z),\ g : \mathbf{Hom}(y, z) \\ & \vdash \langle \pi_1, \pi_2 \rangle \equiv \mathrm{id}_{x \times_{f,g} y} : \mathbf{Hom}(x \times_{f,g} y,\ x \times_{f,g} y) \end{aligned}$$

Isos & Monos

- Given $f : \mathbf{Hom}(x, y)$, we can straightforwardly encode the condition that f is an isomorphism as a type in the above theory

$$\mathsf{isIso}(f) := \Sigma g : \mathbf{Hom}(y, x).(g \circ f = \mathrm{id}_x \times f \circ g = \mathrm{id}_y)$$

Isos & Monos

- Given $f : \mathbf{Hom}(x, y)$, we can straightforwardly encode the condition that f is an isomorphism as a type in the above theory

$$\mathsf{isIso}(f) := \Sigma g : \mathbf{Hom}(y, x).(g \circ f = \mathrm{id}_x \times f \circ g = \mathrm{id}_y)$$

- Expressing the condition that f is a monomorphism is trickier, but we can make use of the fact that f is mono iff the following square

$$\begin{tikzcd} x \arrow[r, equal] \arrow[d, equal] & x \arrow[d, "f"] \\ x \arrow[r, "f"'] & y \end{tikzcd}$$

is a pullback

$$\mathsf{isMono}(f) := \mathsf{isIso}(\langle f, f \rangle)$$

The Theory of a Topos

Recall that an elementary topos is equivalently a finitely complete category with power objects. We can therefore obtain a theory of elementary topoi by adding axioms for power objects to the above theory.

$$x : \mathbf{Ob} \vdash \mathcal{P}(x) : \mathbf{Ob} \qquad x : \mathbf{Ob} \vdash \in_x : \mathbf{Ob}$$

$$x : \mathbf{Ob} \vdash \pitchfork_x : \mathbf{Hom}(\in_x, x \times \mathcal{P}(x)) \qquad x : \mathbf{Ob} \vdash \mathsf{isMono}(\pitchfork_x)$$

$$x, y, z : \mathbf{Ob},\ r : \mathbf{Hom}(z, x \times y),\ \mathsf{isMono}(r) \vdash \chi_r : \mathbf{Hom}(y, \mathcal{P}(x))$$

$$x, y, z : \mathbf{Ob},\ r : \mathbf{Hom}(z, x \times y),\ \mathsf{isMono}(r) \vdash \gamma_r : \mathbf{Hom}(z, \in_x)$$

$$\begin{array}{l} x, y, z : \mathbf{Ob},\ r : \mathbf{Hom}(z, x \times y),\ \mathsf{isMono}(r) \\ \vdash \langle \pi_1, \chi_r \circ \pi_2 \rangle \circ r \equiv \pitchfork_x \circ \gamma_r : \mathbf{Hom}(z, x \times \mathcal{P}(x)) \end{array}$$

$$\begin{array}{l} x, y, z : \mathbf{Ob},\ r : \mathbf{Hom}(z, x \times y),\ \mathsf{isMono}(r) \\ \vdash \mathsf{isIso}(\langle r, \gamma_r \rangle) : \mathbf{Hom}(z, (x \times y) \times_{\langle \pi_1, \chi_r \circ \pi_2 \rangle, \pitchfork_x} \in_x) \end{array}$$

$$\begin{array}{l} x, y, z : \mathbf{Ob},\ r : \mathbf{Hom}(z, x \times y),\ \mathsf{isMono}(r) \\ f : \mathbf{Hom}(y, \mathcal{P}(x)),\ g : \mathbf{Hom}(z, \in_x), \\ \langle \pi_1, f \circ \pi_2 \rangle \circ r =_{z, x \times \mathcal{P}(x)} \pitchfork_x \circ g,\ \mathsf{isIso}(\langle r, g \rangle) \\ \vdash f \equiv \chi_r : \mathbf{Hom}(y, \mathcal{P}(x)) \end{array}$$

The Theory of a Topological Spaces (in a Topos)

Using elementary reasoning in the theory of a topos, we may then define maps

$$\mathcal{P}(\mathcal{P}(x)) \xrightarrow{\bigcup} \mathcal{P}(x) \quad \text{and} \quad \mathcal{P}(x) \times \mathcal{P}(x) \xrightarrow{\cap} \mathcal{P}(x)$$

The Theory of a Topological Spaces (in a Topos)

Using elementary reasoning in the theory of a topos, we may then define maps

$$\mathcal{P}(\mathcal{P}(x)) \xrightarrow{\bigcup} \mathcal{P}(x) \quad \text{and} \quad \mathcal{P}(x) \times \mathcal{P}(x) \xrightarrow{\cap} \mathcal{P}(x)$$

A topological space is then an object x equipped with a monomorphism $u : \mathrm{U} \hookrightarrow \mathcal{P}(x)$ such that there are maps completing the following triangles:

$$\begin{tikzcd}
 & & \mathrm{U} \arrow[d, hook] \\
\mathcal{P}(\mathrm{U}) \arrow[r, hook] \arrow[urr, dashed] & \mathcal{P}(\mathcal{P}(x)) \arrow[r, "\bigcup"] & \mathcal{P}(x)
\end{tikzcd}
\qquad
\begin{tikzcd}
 & & \mathrm{U} \arrow[d, hook] \\
\mathrm{U} \times \mathrm{U} \arrow[r, hook] \arrow[urr, dashed] & \mathcal{P}(x) \times \mathcal{P}(x) \arrow[r, "\cap"] & \mathcal{P}(x)
\end{tikzcd}$$

The Theory of a Topological Spaces (in a Topos)

Using elementary reasoning in the theory of a topos, we may then define maps

$$\mathcal{P}(\mathcal{P}(x)) \xrightarrow{\cup} \mathcal{P}(x) \quad \text{and} \quad \mathcal{P}(x) \times \mathcal{P}(x) \xrightarrow{\cap} \mathcal{P}(x)$$

A topological space is then an object x equipped with a monomorphism $u : \mathrm{U} \hookrightarrow \mathcal{P}(x)$ such that there are maps completing the following triangles:

$$\begin{tikzcd}
 & & \mathrm{U} \arrow[d, hook] \\
\mathcal{P}(\mathrm{U}) \arrow[urr, dashed] \arrow[r, hook] & \mathcal{P}(\mathcal{P}(x)) \arrow[r, "\cup"] & \mathcal{P}(x)
\end{tikzcd}
\qquad
\begin{tikzcd}
 & & \mathrm{U} \arrow[d, hook] \\
\mathrm{U} \times \mathrm{U} \arrow[urr, dashed] \arrow[r, hook] & \mathcal{P}(x) \times \mathcal{P}(x) \arrow[r, "\cap"] & \mathcal{P}(x)
\end{tikzcd}$$

Hence we can in fact define topological spaces using essentially algebraic reasoning, by passing through the theory of an elementary topos, which can also be defined essentially algebraically!

The Theory of Theories

A notable thing about the foregoing argument is that we were able to define finitely complete categories – which form the semantics of dependently typed theories – as a dependently typed theory itself.

The Theory of Theories

A notable thing about the foregoing argument is that we were able to define finitely complete categories – which form the semantics of dependently typed theories – as a dependently typed theory itself.In fact, we can go even further, and express the judgments and rules of our type theory within itself as follows:

$$\vdash \mathbf{Ctx}\ \mathsf{Type} \qquad \Gamma : \mathbf{Ctx} \vdash \mathbf{Ty}[\Gamma]\ \mathsf{Type} \qquad \Gamma : \mathbf{Ctx},\ A : \mathbf{Ty}[\Gamma] \vdash \mathbf{Tm}[\Gamma, A]\ \mathsf{Type}$$

$$\Gamma, \Delta : \mathbf{Ctx} \vdash \mathbf{Subst}[\Gamma, \Delta]\ \mathsf{Type} \qquad \vdash \epsilon : \mathbf{Ctx} \qquad \Gamma : \mathbf{Ctx},\ A : \mathbf{Ty}[\Gamma] \vdash \mathsf{ext}(\Gamma, A) : \mathbf{Ctx}$$

$$\begin{array}{l} \Gamma, \Delta : \mathbf{Ctx},\ A : \mathbf{Ty}[\Delta],\ \sigma : \mathbf{Subst}[\Gamma, \Delta] \\ \vdash A[\sigma] : \mathbf{Ty}[\Gamma] \end{array} \qquad \begin{array}{l} \Gamma, \Delta : \mathbf{Ctx},\ A : \mathbf{Ty}[\Delta],\ a : \mathbf{Tm}[\Delta, a],\ \sigma : \mathbf{Subst}[\Gamma, \Delta] \\ \vdash a[\sigma] : \mathbf{Tm}[\Gamma, A[\sigma]] \end{array}$$

$$\Gamma : \mathbf{Ctx} \vdash \bullet : \mathbf{Subst}[\Gamma, \epsilon] \qquad \begin{array}{l} \Gamma, \Delta : \mathbf{Ctx},\ A : \mathbf{Ty}[\Gamma], \\ \ \sigma : \mathbf{Subst}[\Gamma, \Delta],\ a : \mathbf{Tm}[\Delta, A] \\ \vdash \sigma, a : \mathbf{Subst}[\Gamma, \mathsf{ext}(\Delta, A)] \end{array} \qquad \begin{array}{l} \Gamma : \mathbf{Ctx},\ A : \mathbf{Ty}[\Gamma] \\ \vdash \omega : \mathbf{Subst}(\mathsf{ext}(\Gamma, A), \Gamma) \end{array}$$

$$\begin{array}{l} \Gamma : \mathbf{Ctx}, A : \mathbf{Ty}[\Gamma] \\ \vdash 0 : \mathbf{Tm}[\mathsf{ext}(\Gamma, A), A[\omega]] \end{array}$$

etc.

The Theory of Theories

A notable thing about the foregoing argument is that we were able to define finitely complete categories – which form the semantics of dependently typed theories – as a dependently typed theory itself.In fact, we can go even further, and express the judgments and rules of our type theory within itself as follows:

$$\vdash \mathbf{Ctx}\ \mathsf{Type} \qquad \Gamma : \mathbf{Ctx} \vdash \mathbf{Ty}[\Gamma]\ \mathsf{Type} \qquad \Gamma : \mathbf{Ctx},\ A : \mathbf{Ty}[\Gamma] \vdash \mathbf{Tm}[\Gamma, A]\ \mathsf{Type}$$

$$\Gamma, \Delta : \mathbf{Ctx} \vdash \mathbf{Subst}[\Gamma, \Delta]\ \mathsf{Type} \qquad \vdash \epsilon : \mathbf{Ctx} \qquad \Gamma : \mathbf{Ctx},\ A : \mathbf{Ty}[\Gamma] \vdash \mathsf{ext}(\Gamma, A) : \mathbf{Ctx}$$

$$\begin{array}{l} \Gamma, \Delta : \mathbf{Ctx},\ A : \mathbf{Ty}[\Delta],\ \sigma : \mathbf{Subst}[\Gamma, \Delta] \\ \vdash A[\sigma] : \mathbf{Ty}[\Gamma] \end{array} \qquad \begin{array}{l} \Gamma, \Delta : \mathbf{Ctx},\ A : \mathbf{Ty}[\Delta],\ a : \mathbf{Tm}[\Delta, a],\ \sigma : \mathbf{Subst}[\Gamma, \Delta] \\ \vdash a[\sigma] : \mathbf{Tm}[\Gamma, A[\sigma]] \end{array}$$

$$\Gamma : \mathbf{Ctx} \vdash \bullet : \mathbf{Subst}[\Gamma, \epsilon] \qquad \begin{array}{l} \Gamma, \Delta : \mathbf{Ctx},\ A : \mathbf{Ty}[\Gamma], \\ \ \sigma : \mathbf{Subst}[\Gamma, \Delta],\ a : \mathbf{Tm}[\Delta, A] \\ \vdash \sigma, a : \mathbf{Subst}[\Gamma, \mathsf{ext}(\Delta, A)] \end{array} \qquad \begin{array}{l} \Gamma : \mathbf{Ctx},\ A : \mathbf{Ty}[\Gamma] \\ \vdash \omega : \mathbf{Subst}(\mathsf{ext}(\Gamma, A), \Gamma) \end{array}$$

$$\begin{array}{l} \Gamma : \mathbf{Ctx}, A : \mathbf{Ty}[\Gamma] \\ \vdash 0 : \mathbf{Tm}[\mathsf{ext}(\Gamma, A), A[\omega]] \end{array}$$

etc. The (meta-theoretic) types we may build in this theory correspond to lists of axioms stipulating the existence of types, terms, and equalities between these. In other words, they are essentially the *signatures* of dependently-typed theories. Hence working in this theory amounts to working in the internal language of (the opposite of) the category of finitely-presentable theories.

Foundations without Foundations

All presentations of various type theories in terms of (finitely many) judgments and rules live inside this theory. Here, for instance, is the signature of dependent product types (as a type dependent upon the above signature):

$$\Gamma : \mathbf{Ctx},\ \mathrm{A} : \mathbf{Ty}[\Gamma],\ \mathrm{B} : \mathbf{Ty}[\mathsf{ext}(\Gamma, \mathrm{A})] \vdash \Pi \mathrm{A.B} : \mathbf{Ty}[\Gamma]$$

$$\begin{array}{l} \Gamma : \mathbf{Ctx},\ \mathrm{A} : \mathbf{Ty}[\Gamma],\ \mathrm{B} : \mathbf{Ty}[\mathsf{ext}(\Gamma, \mathrm{A})], \\ f : \mathbf{Tm}[\mathsf{Ext}(\Gamma, \mathrm{A}), \mathrm{B}] \vdash \lambda(f) : \mathbf{Tm}[\Gamma, \Pi \mathrm{A.B}] \end{array}$$

$$\begin{array}{l} \Gamma : \mathbf{Ctx},\ \mathrm{A} : \mathbf{Ty}[\Gamma],\ \mathrm{B} : \mathbf{Ty}[\mathsf{ext}(\Gamma, \mathrm{A})], \\ f : \mathbf{Tm}[\Gamma, \Pi \mathrm{A.B}],\ a : \mathbf{Tm}[\Gamma, \mathrm{A}] \\ \vdash \mathsf{app}(f, a) : \mathbf{Tm}[\Gamma, \mathrm{B}[\mathsf{id}_\Gamma, a]] \end{array}$$

$$\begin{array}{l} \Gamma : \mathbf{Ctx},\ \mathrm{A} : \mathbf{Ty}[\Gamma],\ \mathrm{B} : \mathbf{Ty}[\mathsf{ext}(\Gamma, \mathrm{A})], \\ f : \mathbf{Tm}[\mathsf{Ext}(\Gamma, \mathrm{A}), \mathrm{B}], a : \mathbf{Tm}[\Gamma, \mathrm{A}] \\ \vdash \mathsf{app}(\lambda(f), a) \equiv f[\mathsf{id}_\Gamma, a] : \mathbf{Tm}[\Gamma, \mathrm{B}[\mathsf{id}_\Gamma, a]] \end{array}$$

$$\begin{array}{l} \Gamma : \mathbf{Ctx},\ \mathrm{A} : \mathbf{Ty}[\Gamma],\ \mathrm{B} : \mathbf{Ty}[\mathsf{ext}(\Gamma, \mathrm{A})], \\ f : \mathbf{Tm}[\Gamma, \Pi \mathrm{A.B}] \\ \vdash f \equiv \lambda(\mathsf{app}(f, 0)) : \mathbf{Tm}[\Gamma, \Pi \mathrm{A.B}] \end{array}$$

Foundations without Foundations

All presentations of various type theories in terms of (finitely many) judgments and rules live inside this theory. Here, for instance, is the signature of dependent product types (as a type dependent upon the above signature):

$$\Gamma : \mathbf{Ctx},\ A : \mathbf{Ty}[\Gamma],\ B : \mathbf{Ty}[\mathsf{ext}(\Gamma, A)] \vdash \Pi A.B : \mathbf{Ty}[\Gamma]$$

$$\begin{array}{l} \Gamma : \mathbf{Ctx},\ A : \mathbf{Ty}[\Gamma],\ B : \mathbf{Ty}[\mathsf{ext}(\Gamma, A)], \\ f : \mathbf{Tm}[\mathsf{Ext}(\Gamma, A), B] \vdash \lambda(f) : \mathbf{Tm}[\Gamma, \Pi A.B] \end{array}$$

$$\begin{array}{l} \Gamma : \mathbf{Ctx},\ A : \mathbf{Ty}[\Gamma],\ B : \mathbf{Ty}[\mathsf{ext}(\Gamma, A)], \\ f : \mathbf{Tm}[\Gamma, \Pi A.B],\ a : \mathbf{Tm}[\Gamma, A] \\ \vdash \mathsf{app}(f, a) : \mathbf{Tm}[\Gamma, B[\mathsf{id}_\Gamma, a]] \end{array}$$

$$\begin{array}{l} \Gamma : \mathbf{Ctx},\ A : \mathbf{Ty}[\Gamma],\ B : \mathbf{Ty}[\mathsf{ext}(\Gamma, A)], \\ f : \mathbf{Tm}[\mathsf{Ext}(\Gamma, A), B], a : \mathbf{Tm}[\Gamma, A] \\ \vdash \mathsf{app}(\lambda(f), a) \equiv f[\mathsf{id}_\Gamma, a] : \mathbf{Tm}[\Gamma, B[\mathsf{id}_\Gamma, a]] \end{array}$$

$$\begin{array}{l} \Gamma : \mathbf{Ctx},\ A : \mathbf{Ty}[\Gamma],\ B : \mathbf{Ty}[\mathsf{ext}(\Gamma, A)], \\ f : \mathbf{Tm}[\Gamma, \Pi A.B] \\ \vdash f \equiv \lambda(\mathsf{app}(f, 0)) : \mathbf{Tm}[\Gamma, \Pi A.B] \end{array}$$

Other theories that live inside the theory of theories are:

Foundations without Foundations

All presentations of various type theories in terms of (finitely many) judgments and rules live inside this theory. Here, for instance, is the signature of dependent product types (as a type dependent upon the above signature):

$$\Gamma : \mathbf{Ctx},\ A : \mathbf{Ty}[\Gamma],\ B : \mathbf{Ty}[\mathsf{ext}(\Gamma, A)] \vdash \Pi A.B : \mathbf{Ty}[\Gamma]$$

$$\Gamma : \mathbf{Ctx},\ A : \mathbf{Ty}[\Gamma],\ B : \mathbf{Ty}[\mathsf{ext}(\Gamma, A)],\\ f : \mathbf{Tm}[\mathsf{Ext}(\Gamma, A), B] \vdash \lambda(f) : \mathbf{Tm}[\Gamma, \Pi A.B]$$

$$\Gamma : \mathbf{Ctx},\ A : \mathbf{Ty}[\Gamma],\ B : \mathbf{Ty}[\mathsf{ext}(\Gamma, A)],\\ f : \mathbf{Tm}[\Gamma, \Pi A.B],\ a : \mathbf{Tm}[\Gamma, A]\\ \vdash \mathsf{app}(f, a) : \mathbf{Tm}[\Gamma, B[\mathsf{id}_\Gamma, a]]$$

$$\Gamma : \mathbf{Ctx},\ A : \mathbf{Ty}[\Gamma],\ B : \mathbf{Ty}[\mathsf{ext}(\Gamma, A)],\\ f : \mathbf{Tm}[\mathsf{Ext}(\Gamma, A), B], a : \mathbf{Tm}[\Gamma, A]\\ \vdash \mathsf{app}(\lambda(f), a) \equiv f[\mathsf{id}_\Gamma, a] : \mathbf{Tm}[\Gamma, B[\mathsf{id}_\Gamma, a]]$$

$$\Gamma : \mathbf{Ctx},\ A : \mathbf{Ty}[\Gamma],\ B : \mathbf{Ty}[\mathsf{ext}(\Gamma, A)],\\ f : \mathbf{Tm}[\Gamma, \Pi A.B]\\ \vdash f \equiv \lambda(\mathsf{app}(f, 0)) : \mathbf{Tm}[\Gamma, \Pi A.B]$$

Other theories that live inside the theory of theories are:

- Martin-Löf Type Theory

Foundations without Foundations

All presentations of various type theories in terms of (finitely many) judgments and rules live inside this theory. Here, for instance, is the signature of dependent product types (as a type dependent upon the above signature):

$$\Gamma : \mathbf{Ctx},\ A : \mathbf{Ty}[\Gamma],\ B : \mathbf{Ty}[\mathsf{ext}(\Gamma, A)] \vdash \Pi A.B : \mathbf{Ty}[\Gamma]$$

$$\begin{array}{l}\Gamma : \mathbf{Ctx},\ A : \mathbf{Ty}[\Gamma],\ B : \mathbf{Ty}[\mathsf{ext}(\Gamma, A)],\\ f : \mathbf{Tm}[\mathsf{Ext}(\Gamma, A), B] \vdash \lambda(f) : \mathbf{Tm}[\Gamma, \Pi A.B]\end{array}$$

$$\begin{array}{l}\Gamma : \mathbf{Ctx},\ A : \mathbf{Ty}[\Gamma],\ B : \mathbf{Ty}[\mathsf{ext}(\Gamma, A)],\\ f : \mathbf{Tm}[\Gamma, \Pi A.B],\ a : \mathbf{Tm}[\Gamma, A]\\ \vdash \mathsf{app}(f, a) : \mathbf{Tm}[\Gamma, B[\mathsf{id}_\Gamma, a]]\end{array}$$

$$\begin{array}{l}\Gamma : \mathbf{Ctx},\ A : \mathbf{Ty}[\Gamma],\ B : \mathbf{Ty}[\mathsf{ext}(\Gamma, A)],\\ f : \mathbf{Tm}[\mathsf{Ext}(\Gamma, A), B], a : \mathbf{Tm}[\Gamma, A]\\ \vdash \mathsf{app}(\lambda(f), a) \equiv f[\mathsf{id}_\Gamma, a] : \mathbf{Tm}[\Gamma, B[\mathsf{id}_\Gamma, a]]\end{array}$$

$$\begin{array}{l}\Gamma : \mathbf{Ctx},\ A : \mathbf{Ty}[\Gamma],\ B : \mathbf{Ty}[\mathsf{ext}(\Gamma, A)],\\ f : \mathbf{Tm}[\Gamma, \Pi A.B]\\ \vdash f \equiv \lambda(\mathsf{app}(f, 0)) : \mathbf{Tm}[\Gamma, \Pi A.B]\end{array}$$

Other theories that live inside the theory of theories are:

- Martin-Löf Type Theory
- First-order logic with the ZFC axioms

Foundations without Foundations

All presentations of various type theories in terms of (finitely many) judgments and rules live inside this theory. Here, for instance, is the signature of dependent product types (as a type dependent upon the above signature):

$$\Gamma : \mathbf{Ctx},\ \mathrm{A} : \mathbf{Ty}[\Gamma],\ \mathrm{B} : \mathbf{Ty}[\mathsf{ext}(\Gamma, \mathrm{A})] \vdash \Pi \mathrm{A}.\mathrm{B} : \mathbf{Ty}[\Gamma]$$

$$\Gamma : \mathbf{Ctx},\ \mathrm{A} : \mathbf{Ty}[\Gamma],\ \mathrm{B} : \mathbf{Ty}[\mathsf{ext}(\Gamma, \mathrm{A})], \\ f : \mathbf{Tm}[\mathsf{Ext}(\Gamma, \mathrm{A}), \mathrm{B}] \vdash \lambda(f) : \mathbf{Tm}[\Gamma, \Pi \mathrm{A}.\mathrm{B}]$$

$$\Gamma : \mathbf{Ctx},\ \mathrm{A} : \mathbf{Ty}[\Gamma],\ \mathrm{B} : \mathbf{Ty}[\mathsf{ext}(\Gamma, \mathrm{A})], \\ f : \mathbf{Tm}[\Gamma, \Pi \mathrm{A}.\mathrm{B}],\ a : \mathbf{Tm}[\Gamma, \mathrm{A}] \\ \vdash \mathsf{app}(f, a) : \mathbf{Tm}[\Gamma, \mathrm{B}[\mathsf{id}_\Gamma, a]]$$

$$\Gamma : \mathbf{Ctx},\ \mathrm{A} : \mathbf{Ty}[\Gamma],\ \mathrm{B} : \mathbf{Ty}[\mathsf{ext}(\Gamma, \mathrm{A})], \\ f : \mathbf{Tm}[\mathsf{Ext}(\Gamma, \mathrm{A}), \mathrm{B}], a : \mathbf{Tm}[\Gamma, \mathrm{A}] \\ \vdash \mathsf{app}(\lambda(f), a) \equiv f[\mathsf{id}_\Gamma, a] : \mathbf{Tm}[\Gamma, \mathrm{B}[\mathsf{id}_\Gamma, a]]$$

$$\Gamma : \mathbf{Ctx},\ \mathrm{A} : \mathbf{Ty}[\Gamma],\ \mathrm{B} : \mathbf{Ty}[\mathsf{ext}(\Gamma, \mathrm{A})], \\ f : \mathbf{Tm}[\Gamma, \Pi \mathrm{A}.\mathrm{B}] \\ \vdash f \equiv \lambda(\mathsf{app}(f, 0)) : \mathbf{Tm}[\Gamma, \Pi \mathrm{A}.\mathrm{B}]$$

Other theories that live inside the theory of theories are:

- Martin-Löf Type Theory
- First-order logic with the ZFC axioms
- Homotopy Type Theory

Foundations without Foundations

All presentations of various type theories in terms of (finitely many) judgments and rules live inside this theory. Here, for instance, is the signature of dependent product types (as a type dependent upon the above signature):

$$\Gamma : \mathbf{Ctx},\ A : \mathbf{Ty}[\Gamma],\ B : \mathbf{Ty}[\mathsf{ext}(\Gamma, A)] \vdash \Pi A.B : \mathbf{Ty}[\Gamma]$$

$$\Gamma : \mathbf{Ctx},\ A : \mathbf{Ty}[\Gamma],\ B : \mathbf{Ty}[\mathsf{ext}(\Gamma, A)], \\ f : \mathbf{Tm}[\mathsf{Ext}(\Gamma, A), B] \vdash \lambda(f) : \mathbf{Tm}[\Gamma, \Pi A.B]$$

$$\Gamma : \mathbf{Ctx},\ A : \mathbf{Ty}[\Gamma],\ B : \mathbf{Ty}[\mathsf{ext}(\Gamma, A)], \\ f : \mathbf{Tm}[\Gamma, \Pi A.B],\ a : \mathbf{Tm}[\Gamma, A] \\ \vdash \mathsf{app}(f, a) : \mathbf{Tm}[\Gamma, B[\mathsf{id}_\Gamma, a]]$$

$$\Gamma : \mathbf{Ctx},\ A : \mathbf{Ty}[\Gamma],\ B : \mathbf{Ty}[\mathsf{ext}(\Gamma, A)], \\ f : \mathbf{Tm}[\mathsf{Ext}(\Gamma, A), B], a : \mathbf{Tm}[\Gamma, A] \\ \vdash \mathsf{app}(\lambda(f), a) \equiv f[\mathsf{id}_\Gamma, a] : \mathbf{Tm}[\Gamma, B[\mathsf{id}_\Gamma, a]]$$

$$\Gamma : \mathbf{Ctx},\ A : \mathbf{Ty}[\Gamma],\ B : \mathbf{Ty}[\mathsf{ext}(\Gamma, A)], \\ f : \mathbf{Tm}[\Gamma, \Pi A.B] \\ \vdash f \equiv \lambda(\mathsf{app}(f, 0)) : \mathbf{Tm}[\Gamma, \Pi A.B]$$

Other theories that live inside the theory of theories are:

- Martin-Löf Type Theory
- First-order logic with the ZFC axioms
- Homotopy Type Theory

We can thus use the theory of theories to mediate between these and other foundational systems. In particular, by duality, a function $f : S \to T$ in this theory corresponds to a structure-preserving map from the theory T to S, i.e. a model of the theory T inside the theory S.

Further Questions

- We have not touched on the issue of *strictness* in interpreting dependent type theory in categorical models. There is thus evidently more of this story to be told, having to do with things like natural models, etc.

Further Questions

- We have not touched on the issue of *strictness* in interpreting dependent type theory in categorical models. There is thus evidently more of this story to be told, having to do with things like natural models, etc.
- Type-checking for the type theory in this talk is *very* undecidable, because of the use of extensional equality. Can we approximate extensional type theory with a more well-behaved type theory?

Further Questions

- We have not touched on the issue of *strictness* in interpreting dependent type theory in categorical models. There is thus evidently more of this story to be told, having to do with things like natural models, etc.
- Type-checking for the type theory in this talk is *very* undecidable, because of the use of extensional equality. Can we approximate extensional type theory with a more well-behaved type theory?
    - E.g. Agda allows promoting equality proofs of an appropriate form to rewrite rules, which can simulate much of the extensional type theory given in this talk. Can we find a strong theoretical basis for such "simulation"?

Further Questions

- We have not touched on the issue of *strictness* in interpreting dependent type theory in categorical models. There is thus evidently more of this story to be told, having to do with things like natural models, etc.
- Type-checking for the type theory in this talk is *very* undecidable, because of the use of extensional equality. Can we approximate extensional type theory with a more well-behaved type theory?
    - E.g. Agda allows promoting equality proofs of an appropriate form to rewrite rules, which can simulate much of the extensional type theory given in this talk. Can we find a strong theoretical basis for such "simulation"?
- Not all presentations of theories are created equal when it comes to computation, elaboration, etc. E.g. there are different presentations of categories, some of which are more well-suited to the discipline of *bidirectional type-checking* (c.f. Conor McBride's 2021 Topos Colloquium talk). What is the theoretical basis for these sorts of distinctions?

Further Questions

- We have not touched on the issue of *strictness* in interpreting dependent type theory in categorical models. There is thus evidently more of this story to be told, having to do with things like natural models, etc.
- Type-checking for the type theory in this talk is *very* undecidable, because of the use of extensional equality. Can we approximate extensional type theory with a more well-behaved type theory?
  - E.g. Agda allows promoting equality proofs of an appropriate form to rewrite rules, which can simulate much of the extensional type theory given in this talk. Can we find a strong theoretical basis for such "simulation"?
- Not all presentations of theories are created equal when it comes to computation, elaboration, etc. E.g. there are different presentations of categories, some of which are more well-suited to the discipline of *bidirectional type-checking* (c.f. Conor McBride's 2021 Topos Colloquium talk). What is the theoretical basis for these sorts of distinctions?
- The foundational frameworks that have generally been adopted for mathematics mostly look like the internal languages of various topoi/∞-topoi. What optimizations/enhancements can be made to this framework if this is where we know we are headed anyway?

Thank you!