Categorical models of type theory

Michael Shulman

February 28, 2012

Theories and models

Example

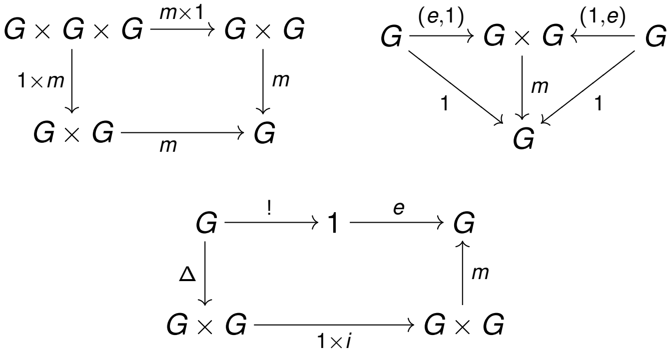
The theory of a group asserts an identity e, products $x \cdot y$ and inverses x^{-1} for any x, y, and equalities $x \cdot (y \cdot z) = (x \cdot y) \cdot z$ and $x \cdot e = x = e \cdot x$ and $x \cdot x^{-1} = e$.

- A model of this theory (in sets) is a *particular*particular group, like $\mathbb{Z}$ or S_3.
- A model in spaces is a *topological* group.
- A model in manifolds is a *Lie group*.
- …

Group objects in categories

Definition

A group object in a category with finite products is an object G with morphisms $e\colon 1 \to G$, $m\colon G \times G \to G$, and $i\colon G \to G$, such that the following diagrams commute.



Categorical semantics

Categorical semantics is a general procedure to go from

1. the theory of a group to
2. the notion of group object in a category.

A group object in a category is a model of the theory of a group.

Then, anything we can prove formally in the theory of a group will be valid for group objects in any category.

Doctrines

For each kind of type theory there is a corresponding kind of structured category in which we consider models.

Algebraic theory	$\longleftrightarrow$	Category with finite products
Simply typed λ-calculus	$\longleftrightarrow$	Cartesian closed category
Dependent type theory	$\longleftrightarrow$	Locally c.c. category
	$\vdots$	

A doctrine specifies

- A collection of type constructors (e.g. $\times$), and
- A categorical structure realizing those constructors as operations (e.g. cartesian products).

Theores and models

Once we have fixed a doctrine **D**, then

- A **D**-theory specifies "generating" or "axiomatic" types and terms.
- A **D**-category is one possessing the specified structure.
- A model of a **D**-theory **T** in a **D**-category **C** realizes the types and terms in **T** as objects and morphisms of **C**.

The doctrine of finite products

Definition

A finite-product theory is a type theory with unit and $\times$ as the only type constructors, plus any number of *axioms*.

Example

The theory of magmas has one axiomatic type M, and axiomatic terms

$$\vdash e : M \qquad \text{and} \qquad x : M, y : M \vdash (x \cdot y) : M$$

For monoids or groups, we need equality axioms (later).

Models of finite-product theories

T a finite-product theory, **C** a category with finite products.

Definition

A model of **T** in **C** assigns

1. To each type A in **T**, an object $[\![A]\!]$ in **C**
2. To each judgment derivable in **T**:

$$x_1 : A_1, \ldots, x_n : A_n \vdash b : B$$

   a morphism in **C**:

$$[\![A_1]\!] \times \cdots \times [\![A_n]\!] \xrightarrow{[\![b]\!]} [\![B]\!].$$

3. Such that $[\![A \times B]\!] = [\![A]\!] \times [\![B]\!]$, etc.

Models of finite-product theories

To define a model of **T** in **C**, it suffices to interpret the axioms.

Example

A model of the theory of magmas in **C** consists of

- An object $[\![M]\!]$.
- A morphism $1 \xrightarrow{[\![e]\!]} [\![M]\!]$.
- A morphism $[\![M]\!] \times [\![M]\!] \xrightarrow{[\![\cdot]\!]} [\![M]\!]$.

Given this, any other term like

$$x : M, y : M, z : M \vdash x \cdot (y \cdot z) : M$$

is *automatically* interpreted by the composite

$$[\![M]\!] \times [\![M]\!] \times [\![M]\!] \xrightarrow{1 \times [\![\cdot]\!]} [\![M]\!] \times [\![M]\!] \xrightarrow{[\![\cdot]\!]} [\![M]\!]$$

Complete theories

Definition

The complete theory Th(**C**) of a **D**-category **C** has

- As axiomatic types, all the objects of **C**.
- As axiomatic terms, all the morphisms of **C**.

Remarks

- The theory Th(**C**) has a tautological model in **C**.
- A model of **T** in **C** is equivalently a *translation* of **T** into Th(**C**).
- Reasoning in Th(**C**), or a subtheory of it, is a way to prove things specifically about **C**.

Syntactic categories

Definition

The syntactic category Syn(**T**) of a **D**-theory **T** has

- As objects, exactly the types of **T**.
- As morphisms, exactly the terms of **T**.

Remarks

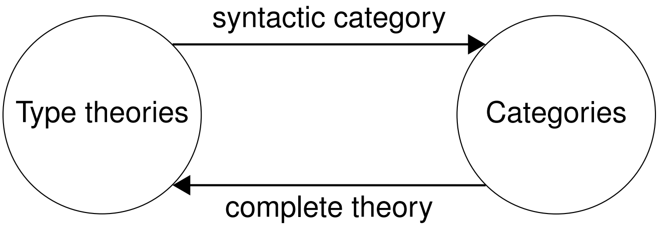
- The theory **T** has a tautological model in Syn(**T**).
- A model of **T** in **C** is equivalently a structure-preserving functor Syn(**T**) $\to$ **C**.
- That is, Syn(**T**) $\to$ **C** is the *free* **D***-category* generated by a model of **T**.
- Studying Syn(**T**) categorically can yield meta-theoretic information about **T**.

The syntax–semantics adjunction

There are bijections between:

1. Models of a theory **T** in a category **C**
2. Structure-preserving functors Syn(**T**) $\to$ **C**
3. Translations **T** $\to$ Th(**C**)

Hence Syn is left adjoint to Th.



Depending on how you set things up, you can make this adjunction an equivalence.

Why categorical semantics

- When we prove something in a particular type theory, like the theory of a group, it is then automatically valid for models of that theory in all different categories.
- We can use type theory to prove things about a particular category by working in its complete theory.
- We can use category theory to prove things about a type theory by working with its syntactic category.

A list of doctrines

unit	$\longleftrightarrow$	terminal object
$\emptyset$	$\longleftrightarrow$	initial object
product $A \times B$	$\longleftrightarrow$	categorical product
disjoint union $A + B$	$\longleftrightarrow$	categorical coproduct
function type $A \to B$	$\longleftrightarrow$	exponentials (cartesian closure)

To include a type constructor in a doctrine, we have to specify meanings for

1. the type constructor (an operation on objects)
2. its constructors, and
3. its eliminators.

Universal properties

The categorical versions of type constructors are generally characterized by *universal properties*.

Definition

A left universal property for an object X of a category is a way of describing $\hom(X, Z)$ up to isomorphism for every object Z, which is "natural in Z".

Examples

- $\hom(\emptyset, Z) \cong *$.
- $\hom(A + B, Z) \cong \hom(A, Z) \times \hom(B, Z)$.

Definition

A right universal property for an object X of a category is a way of describing $\hom(Z, X)$ up to isomorphism for every object Z, which is "natural in Z".

Uniqueness of universal properties

Theorem

If X and X' have the same universal property, then $X \cong X'$.

Example

Suppose $\hom(\emptyset, Z) \cong *$ and $\hom(\emptyset', Z) \cong *$ for all Z.

- Then $\hom(\emptyset, \emptyset') \cong *$ and $\hom(\emptyset', \emptyset) \cong *$, so we have morphisms $\emptyset \to \emptyset'$ and $\emptyset' \to \emptyset$.
- Also $\hom(\emptyset, \emptyset) \cong *$ and $\hom(\emptyset', \emptyset') \cong *$, so the composites $\emptyset \to \emptyset' \to \emptyset$ and $\emptyset' \to \emptyset \to \emptyset'$ must be identities.

Interpreting positive types

Positive type constructors are generally interpreted by objects with left universal properties.

- The constructors are given as data along with the objects.
- The eliminators are obtained from the universal property.

Example

An initial object has $\hom(\emptyset, Z) \cong *$.

- No extra data (no constructors).
- For every Z, we have a unique morphism $\emptyset \to Z$ (the eliminator "abort" or "match with end").

Interpreting positive types

Positive type constructors are generally interpreted by objects with left universal properties.

- The constructors are given as data along with the objects.
- The eliminators are obtained from the universal property.

Example

A coproduct of A, B has morphisms $\mathrm{inl}\colon A \to A + B$ and $\mathrm{inr}\colon B \to A + B$, such that composition with inl and inr:

$$\hom(A + B, Z) \to \hom(A, Z) \times \hom(B, Z)$$

is a bijection.

- Two data inl and inr (type constructors of a disjoint union).
- Given $A \to Z$ and $B \to Z$, we have a unique morphism $A + B \to Z$ (the eliminator, definition by cases).

Interpreting negative types

Negative type constructors are generally interpreted by objects with right universal properties.

- The eliminators are given as data along with the objects.
- The constructors are obtained from the universal property.

Example

An exponential of A, B has a morphism $\mathrm{ev}\colon B^A \times A \to B$, such that composition with ev:

$$\hom(Z, B^A) \to \hom(Z \times A, B)$$

is a bijection.

- One datum ev (eliminator of function types, application).
- Given a morphism $A \to B$, we have a unique element of B^A (the constructor, λ-abstraction).

Cartesian products are special

Definition

A product of A, B has morphisms $\mathrm{pr}_1 \colon A \times B \to A$ and $\mathrm{pr}_2 \colon A \times B \to B$, such that composition with pr_1 and pr_2:

$$\hom(Z, A \times B) \to \hom(Z, A) \times \hom(Z, B)$$

is a bijection.

- This is a right universal property... but we said products were a positive type!
- Also: we already used products $\times$ in other places!

How to deal with products

Backing up: how do we interpret terms

$$x : A,\ y : B \vdash\ c : C$$

if we don't have the type constructor $\times$?
(i.e. if our category of types doesn't have products?)

1. Work in a cartesian multicategory: in addition to morphisms $A \to C$ we have "multimorphisms" $A, B \to C$.
2. OR: associate objects to contexts rather than types.

These are basically equivalent. The first is arguably better; the second is simpler to describe and generalize.

Display object categories

Definition

A display object category is a category with

- A terminal object.
- A subclass of its objects called the display objects.
- The product of any object by a display object exists.

Idea

- The objects represent *contexts*.
- The display objects represent singleton contexts $x : A$, which are equivalent to *types*.
- Think of non-display objects as “formal products” of display objects.

Examples of d.o. categories

Example

Any category having products and a terminal object (e.g. sets), with *all* objects being display.

Example

To define Syn(**T**) when the doctrine lacks products:

- objects = contexts
- morphisms = tuples of terms
- display objects = singleton contexts

Contexts in d.o. categories

Now we interpret types by display objects, and a term

$$x\colon A,\, y\colon B \vdash\ c\colon C$$

by a morphism

$$[\![A]\!] \times [\![B]\!] \to [\![C]\!]$$

where $[\![A]\!] \times [\![B]\!]$ interprets the context $x\colon A,\, y\colon B$, and need not be a display object itself.

Similarly, a term $\vdash c\colon C$ in the empty context gives a morphism $1 \to [\![C]\!]$ out of the terminal object 1, which may not be display.

Products in d.o. categories

The left universal property for the positive product type:

$$\frac{x : A, y : B \vdash z : Z}{p : A \times B \vdash \mathrm{match}(\dots) : Z}$$

Definition

Given display objects A and B, a display product is a display object P with a morphism $A \times B \to P$, such that composition with it:

$$\hom(P, Z) \to \hom(A \times B, Z)$$

is a bijection.

It follows that $A \times B \to P$ is an isomorphism, so we are really just saying that display objects are closed under products.

Other types in d.o. categories

- Products: Display objects are closed under products.
- Disjoint unions: any two display objects have a coproduct which is also a display object, and products distribute over coproducts.
- $\emptyset$: there is an initial object that is a display object.
- unit: The terminal object is a display object.
- Function types: any two display objects have an exponential which is also a display object.

Dependent contexts

Question

If $B\colon A \to \mathrm{Type}$, how do we interpret a judgment

$$x\colon A,\, y\colon B(x) \;\vdash\; c\colon C \quad ?$$

Partial Answer

If we associate objects to contexts as in a display object category, this will just be a morphism

$$[\![x\colon A,\, y\colon B(x)]\!] \to [\![C]\!]$$

but what is the object on the left, and how is it related to $[\![A]\!]$ and $B\colon A \to \mathrm{Type}$?

Well: there should be a projection $[\![x\colon A,\, y\colon B(x)]\!] \to [\![A]\!]$.

Display map categories

Definition

A display map category is a category with

- A terminal object.
- A subclass of its morphisms called the display maps, denoted $B \twoheadrightarrow A$ or $B \rightarrowtriangle A$.
- Any pullback of a display map exists and is a display map.

Remarks

- The objects represent contexts.
- A display map represents a projection $[\![\Gamma, y : B]\!] \twoheadrightarrow [\![\Gamma]\!]$ (the type B may depend on Γ).
- The fiber of this projection over $x : \Gamma$ is the type $B(x)$.
- The display objects are those with $A \twoheadrightarrow 1$ a display map.

Pullbacks and substitution

The pullback of a display map represents substitution into a dependent type. Given $f\colon A \to B$ and a dependent type $y\colon B \vdash C\colon \mathrm{Type}$, we have $x\colon A \vdash C[f(x)/y]\colon \mathrm{Type}$.

$$\begin{array}{ccc} [\![C[f(x)/y]]\!] & \longrightarrow & [\![C]\!] \\ \downarrow\!\!\!\downarrow & & \downarrow\!\!\!\downarrow \\ [\![A]\!] & \underset{f}{\longrightarrow} & [\![B]\!] \end{array}$$

In particular, for two types A and B in the empty context:

$$\begin{array}{ccc} [\![A]\!] \times [\![B]\!] & \longrightarrow & [\![B]\!] \\ \downarrow\!\!\!\downarrow & & \downarrow\!\!\!\downarrow \\ [\![A]\!] & \twoheadrightarrow & 1 \end{array}$$

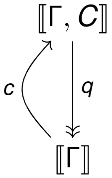
represents the context $x\colon A, y\colon B$, as in a d.o. category.

Dependent terms

Given $\Gamma \vdash C\colon \mathrm{Type}$ represented by $q\colon [\![\Gamma, C]\!] \twoheadrightarrow [\![\Gamma]\!]$, a term

$$\Gamma \;\vdash\; c\colon C$$

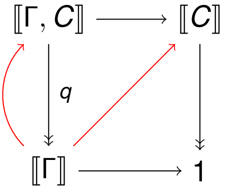
is represented by a section



(i.e. $qc = 1_{[\![\Gamma]\!]}$)

Non-dependent terms

If C is independent of Γ, then $q\colon [\![\Gamma, C]\!] \twoheadrightarrow [\![\Gamma]\!]$ is the pullback



and sections of it are the same as maps $[\![\Gamma]\!] \to [\![C]\!]$, as before.

Dependent sums in d.m. categories

Definition

Given a display object $A \twoheadrightarrow 1$ and a display map $B \twoheadrightarrow A$, a dependent sum is a display object $P \twoheadrightarrow 1$ with a map $B \to P$, such that composition with it

$$\hom(P, Z) \to \hom(B, Z)$$

is a bijection.

Note: if $B \twoheadrightarrow A$ is the pullback of some $C \twoheadrightarrow 1$, then $B = A \times C$ and this is just a product.

As there, it follows that $B \to P$ is an isomorphism, so we are really saying that display maps are closed under composition.

Dependent products in d.m. categories

Definition

Given $A \twoheadrightarrow 1$ and $B \twoheadrightarrow A$, a dependent product is a display object $P \twoheadrightarrow 1$ with a map $P \times A \to B$ over A, such that composition with it

$$\hom(Z, P) \to \hom_A(Z \times A, B)$$

is a bijection.

(Really, we replace 1 by an arbitrary context Γ everywhere.)

If the category is locally cartesian closed, this means display maps are closed under Π-functors.

Universes and dependent types

But if types are just terms of type Type . . .

$$\text{type of types "Type"} \quad \longleftrightarrow \quad \text{universe object } U$$

Examples

- In sets, U = a Grothendieck universe of "small sets"
- In ∞-groupoids, U = the ∞-groupoid of small ∞-groupoids

Then. . .

$$\begin{aligned} \text{dependent type } A \to \text{Type} \quad &\longleftrightarrow \quad \text{morphism } A \to U \\ &\overset{?}{\longleftrightarrow} \quad \text{display map } B \twoheadrightarrow A \end{aligned}$$

The universal dependent context

A universe object U has to come with a display map

$$\widetilde{U} \twoheadrightarrow U$$

representing the universal dependent context

$$A\colon \mathrm{Type}, x\colon A.$$

A display map $B \twoheadrightarrow A$ represents a context extension by a type in U (a "small type") just when it is a pullback:

$$\begin{array}{ccc} B & \longrightarrow & \widetilde{U} \\ \downarrow\!\!\downarrow & \lrcorner & \downarrow\!\!\downarrow \\ A & \longrightarrow & U \end{array}$$

Coherence

There are issues with coherence.

$$\begin{array}{ccccc} g^*(f^*B) & \longrightarrow & f^*B & \longrightarrow & B \\ \big\downarrow & \lrcorner & \big\downarrow & \lrcorner & \big\downarrow \\ A_3 & \xrightarrow[g]{} & A_2 & \xrightarrow[f]{} & A_1 \end{array} \quad \neq \quad \begin{array}{ccc} (fg)^*B & \longrightarrow & B \\ \big\downarrow & \lrcorner & \big\downarrow \\ A_3 & \xrightarrow[fg]{} & A_1 \end{array}$$

but substitutions in type theory

$$B(z) \mapsto B(f(y)) \mapsto B(f(g(x)))$$
$$B(z) \mapsto B((f \circ g)(x)) = B(f(g(x)))$$

are the same.

Coherence via universes

One solution (Voevodsky)

Interpret dependent types $B\colon A \to \text{Type}$ by morphisms $[\![A]\!] \to U$, obtaining the corresponding display map by pullback when necessary. Then substitution is by composition:

$$A_3 \xrightarrow{g} (A_2 \xrightarrow{f} A_1 \xrightarrow{B} U)$$

$$(A_3 \xrightarrow{g} A_2 \xrightarrow{f} A_1) \xrightarrow{B} U$$

and thus strictly associative.

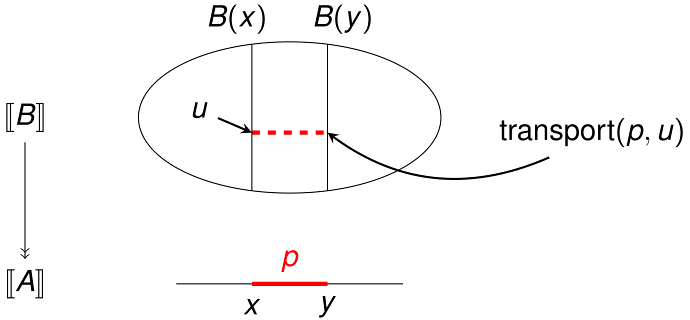
There are other solutions too.

Display maps in homotopy theory

Question

Which maps can be display maps?

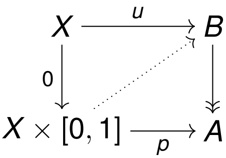
Recall: given $B\colon A \to \mathrm{Type}$, $x, y\colon A$, and $p\colon (x = y)$, we have the operation of transporting along p:



Fibrations

Definition

A map $B \to A$ of spaces (or ∞-groupoids) is a fibration if for any any path $p\colon x \rightsquigarrow y$ in A and any point u in the fiber over x, there is a path $u \rightsquigarrow v$ lying over p. . . and such a path can be chosen to vary continuously in its inputs.



In homotopy type theory, display maps must be fibrations.

Transport in fibrations

If $B \twoheadrightarrow A$ is a fibration, then paths in A act on its fibers by transporting along lifted paths.

Example

The infinite helix $\mathbb{R} \to S^1$.

- Each fiber is $\mathbb{Z}$.
- Transporting around a loop acts on $\mathbb{Z}$ by "$+1$".

Example

The inclusion of a point $* \to S^1$ is not a fibration.

- No way to transport the point $*$ in one fiber any other (empty) fiber.
- Note: $\mathbb{R}$ is homotopy equivalent to $*$, as a space!