Algebraic Type Theory

Steve Awodey

Itaca Fest
September 2023

1. Strictifying Homotopical Models

- A **homotopical model** of (homotopy) type theory can be defined to be a Quillen model category $\mathcal{E}$, with the Frobenius property and a (fibrant, univalent) universe $\dot{\mathsf{U}} \twoheadrightarrow \mathsf{U}$.
- We can extract a **strict model** of (homotopy) type theory from a homotopical one using some ideas of Voevodsky and Lumsdaine-Warren.
- The resulting structure is a **category with families**, which is a quite strict notion of a model of dependent type theory. (A related construction gives a **contextual category**.)
- I will do the cases of Σ and Π types, but one can add the Id-types and a universe U.
- Time permitting, I will consider the resulting structure from another perspective – as a **polynomial monad**.

1. Dependent type theory

The system to be modelled has:

Basic types and terms: $A, B, \ldots, \; x\!:\!A, \; b\!:\!B, \; \ldots$

Dependent types and terms: $x\!:\!A \vdash b(x) : B(x), \; \ldots$

Contexts: $(x\!:\!A, y\!:\!B(x), \ldots), \Gamma, \; \Delta, \; \ldots$

Substitutions: $\sigma\!:\!\Delta \to \Gamma \; , \ldots$

Type forming operations: $\sum_{x:A} B(x), \; \prod_{x:A} B(x), \; \ldots$

Equations between terms: $\Gamma \vdash s = t : A$

1. Dependent type theory: Rules

Contexts:

$$\frac{x : A \vdash B(x)}{x : A,\; y : B(x) \vdash} \qquad \frac{\Gamma \vdash C}{\Gamma, z : C \vdash}$$

Sums:

$$\frac{x : A \vdash B(x)}{\sum_{x:A} B(x)} \qquad \frac{a : A \qquad b : B(a)}{\langle a, b\rangle : \sum_{x:A} B(x)}$$

$$\frac{c : \sum_{x:A} B(x)}{\mathsf{fst}\, c : A} \qquad \frac{c : \sum_{x:A} B(x)}{\mathsf{snd}\, c : B(\mathsf{fst}\, c)}$$

$$\mathsf{fst}\langle a, b\rangle = a : A \qquad \mathsf{snd}\langle a, b\rangle = b : B$$

$$\langle \mathsf{fst}\, c, \mathsf{snd}\, c\rangle = c : \sum_{x:A} B(x)$$

1. Dependent type theory: Rules

Products:

$$\frac{x : A \vdash B(x)}{\prod_{x:A} B(x)} \qquad \frac{x : A \vdash b : B(x)}{\lambda x.b : \prod_{x:A} B(x)}$$

$$\frac{a : A \qquad f : \prod_{x:A} B(x)}{fa : B(a)}$$

$$x : A \vdash (\lambda x.b)x = b : B(x)$$

$$\lambda x.fx = f : \prod_{x:A} B(x)$$

Substitution:

$$\frac{\sigma : \Delta \to \Gamma \qquad \Gamma \vdash a : A}{\Delta \vdash a[\sigma] : A[\sigma]}$$

2. Natural Models of Type Theory

Definition

A natural transformation $f : Y \to X$ of presheaves on a category $\mathbb{C}$ is called **representable** if its pullback along any $\mathrm{y}C \to X$ is represented:

$$\begin{array}{ccc} \mathrm{y}D & \longrightarrow & Y \\ \downarrow & \lrcorner & \downarrow{\scriptstyle f} \\ \mathrm{y}C & \longrightarrow & X \end{array}$$

Proposition

A representable natural transformation is the same thing as a **category with families** *in the sense of Dybjer.*

2. Natural Models as CwFs

Write the objects and arrows of $\mathbb{C}$ as $\sigma : \Delta \to \Gamma$, giving the **category of contexts and substitutions**.

A CwF is usually defined as a presheaf of **types in context**,

$$\mathsf{Ty} : \mathbb{C}^{\mathrm{op}} \to \mathsf{Set}\,,$$

together with a presheaf of **typed terms**,

$$\mathsf{Tm} : (\textstyle\int_{\mathbb{C}} \mathsf{Ty})^{\mathrm{op}} \to \mathsf{Set}\,.$$

But we will reformulate this notion using the equivalence

$$\mathsf{Set}^{(\int_{\mathbb{C}} \mathsf{Ty})^{\mathrm{op}}} \;\simeq\; \mathsf{Set}^{\mathbb{C}^{\mathrm{op}}}/\mathsf{Ty}\,.$$

So we will instead have a map $p : \mathsf{Tm} \to \mathsf{Ty}$.

2. Natural Models as CwFs

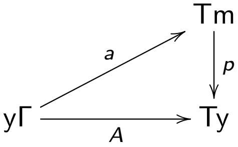
Let $p : \mathsf{Tm} \to \mathsf{Ty}$ be a **representable** map of presheaves on $\mathbb{C}$.

Then Ty is again the **presheaf of types in context**, and now Tm is the **presheaf of terms in context**, and p gives the **typing of terms**.

Formally, we **interpret**:

$$\begin{aligned} \Gamma \vdash A \quad &\approx \quad A \in \mathsf{Ty}(\Gamma) \\ \Gamma \vdash a : A \quad &\approx \quad a \in \mathsf{Tm}(\Gamma) \end{aligned}$$

where $A = p \circ a$.

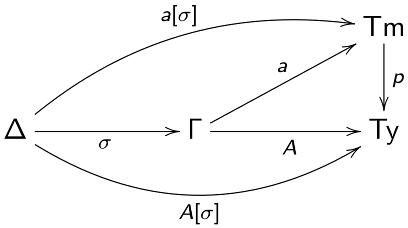


NB: we will now just write Γ rather than yΓ for the representables.

2. Natural Models as CwFs

Naturality of $p : \mathsf{Tm} \to \mathsf{Ty}$ means that for any **substitution** $\sigma : \Delta \to \Gamma$, we have the required action on types and terms:

$$\begin{array}{rcl} \Gamma \vdash A & \Rightarrow & \Delta \vdash A[\sigma] \\ \Gamma \vdash a : A & \Rightarrow & \Delta \vdash a[\sigma] : A[\sigma] \end{array}$$



2. Natural Models as CwFs

Given any further $\tau : \Delta' \to \Delta$ we clearly have

$$A[\sigma][\tau] = A[\sigma \circ \tau] \qquad\qquad a[\sigma][\tau] = a[\sigma \circ \tau]$$

and for the identity substitution $1 : \Gamma \to \Gamma$ we have

$$A[1] = A \qquad\qquad a[1] = a\,.$$

This is the **basic structure** of a CwF.

The remaining operation of **context extension**

$$\frac{\Gamma \vdash A}{\Gamma, x\!:\!A \vdash}$$

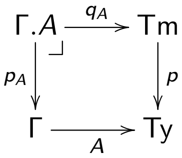
is given by the representability of $p : \mathsf{Tm} \to \mathsf{Ty}$ as follows.

2. Natural Models: Context Extension

Given $\Gamma \vdash A$ we need a new context $\Gamma.A$ together with a substitution $p_A : \Gamma.A \to A$ and a term

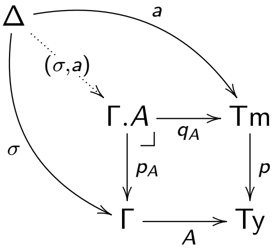
$$\Gamma.A \vdash q_A : A[p_A]\,.$$

Let $p_A : \Gamma.A \to \Gamma$ be the pullback of p along A.



The map $q_A : \Gamma.A \to \mathrm{Tm}$ gives the required term $\Gamma.A \vdash q_A : A[p_A]$.

2. Natural Models: Context Extension



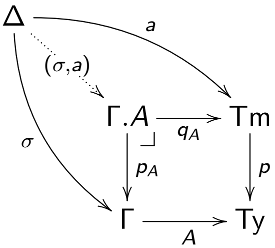
The pullback means that given any substitution $\sigma : \Delta \to \Gamma$ and term $\Delta \vdash a : A[\sigma]$ there is a map

$$(\sigma, a) : \Delta \to \Gamma.A$$

satisfying

$$p_A(\sigma, a) = \sigma$$
$$q_A[\sigma, a] = a.$$

2. Natural Models: Context Extension



By the uniqueness of (σ, a), we also have

$$(\sigma, a) \circ \tau \;=\; (\sigma \circ \tau, a[\tau]) \qquad \text{for any } \tau : \Delta' \to \Delta$$

and

$$(p_A, q_A) = 1.$$

These are precisely the **laws of a CwF**, under the equivalence

$$\mathsf{Set}^{(\int_{\mathbb{C}} \mathsf{Ty})^{\mathrm{op}}} \simeq \mathsf{Set}^{\mathbb{C}^{\mathrm{op}}}/\mathsf{Ty}$$

□

2. Natural Models and Clans

Let $p : \dot{\mathsf{U}} \to \mathsf{U}$ be a natural model.

The fibration $\mathcal{F}_p \to \mathbb{C}$ of all pullbacks

$$A^*p : \Gamma.A \to \Gamma \qquad\qquad \text{for all } A : \Gamma \to \mathsf{U}$$

form a **display map category** (=: **pre-clan**).

Conversely, given any pre-clan $(\mathbb{C}, \mathcal{F})$, there is a natural model $p_{\mathcal{F}} : \dot{\mathsf{U}}_{\mathcal{F}} \to \mathsf{U}_{\mathcal{F}}$ over $\mathbb{C}$,

$$p_{\mathcal{F}} = \coprod_{f \in \mathcal{F}} \mathsf{y}f : \coprod_{f \in \mathcal{F}} \mathsf{y}\mathrm{dom}(f) \to \coprod_{f \in \mathcal{F}} \mathsf{y}\mathrm{cod}(f)\,.$$

There is an adjunction $p \dashv \mathcal{F}$

$$\mathsf{PreClan} \underset{p}{\overset{\mathcal{F}}{\leftrightarrows}} \mathsf{NatMod}\,.$$

2. Natural Models as Algebras

- The notion of a natural model is **essentially algebraic** (generalized algebraic, dependently typed algebraic, clan algebraic, finite limit theory, ...).
- The algebraic homomorphisms correspond exactly to **syntactic translations**.
- There are **initial algebras**, as well as **free algebras** over basic types and terms.
- The rules of type theory can be seen as a procedure for **generating the free algebras**.

3. Modeling the Type Formers

A natural model $p : \dot{\mathsf{U}} \to \mathsf{U}$ determines a **polynomial endofunctor**

$$P : \mathsf{Set}^{\mathbb{C}^{\mathsf{op}}} \to \mathsf{Set}^{\mathbb{C}^{\mathsf{op}}},$$

defined for every $X : \mathbb{C}^{\mathrm{op}} \to \mathsf{Set}$ by

$$P(X) \ = \ \sum_{A\,:\,\mathsf{U}} X^{[A]}\,,$$

where $[A] = p^{-1}(A)$ is the fiber of $p : \dot{\mathsf{U}} \to \mathsf{U}$ at $A : \mathsf{U}$.

3. Modeling the Type Formers: Polynomials

The construction of $P(X)$ can be described as follows.

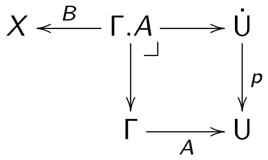
$$\begin{array}{ccc}
\mathsf{Set}^{\mathbb{C}^{op}} & \xrightarrow{\quad P \quad} & \mathsf{Set}^{\mathbb{C}^{op}} \\
{\scriptstyle \dot{\mathsf{U}}^*}\big\downarrow & & \big\uparrow{\scriptstyle \Sigma_{\mathsf{U}}} \\
\mathsf{Set}^{\mathbb{C}^{op}}/\dot{\mathsf{U}} & \xrightarrow[\quad \Pi_p \quad]{} & \mathsf{Set}^{\mathbb{C}^{op}}/U
\end{array}$$

$$\begin{array}{cccc}
X & \longleftarrow & X \times \dot{\mathsf{U}} & & P(X) \\
 & & \big\downarrow & & \big\downarrow \\
 & & \dot{\mathsf{U}} & \xrightarrow[\quad p \quad]{} & \mathsf{U}
\end{array}$$

3. Modeling the Type Formers

Lemma (UMP of a polynomial)

Maps $\Gamma \to P(X)$ *correspond naturally to pairs* (A, B) *where:*

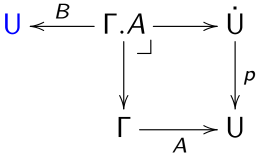


3. Modeling the Type Formers

Applying P to U itself therefore gives an object

$$P(\mathsf{U}) = \sum_{A:\mathsf{U}} \mathsf{U}^{[A]}$$

such that maps $\Gamma \to P(\mathsf{U})$ correspond naturally to **types in an extended context** $\Gamma.A \vdash B$



3. Modeling the Type Formers: Π

Proposition

A natural model $p : \dot{\mathsf{U}} \to \mathsf{U}$ *models the rules for* Π*-types just if there are maps* λ, Π *making the following a pullback.*

$$\begin{array}{ccc} P(\dot{\mathsf{U}}) & \xrightarrow{\lambda} & \dot{\mathsf{U}} \\ {\scriptstyle P(p)}\downarrow & & \downarrow{\scriptstyle p} \\ P(\mathsf{U}) & \xrightarrow[\Pi]{} & \mathsf{U} \end{array}$$

3. Modeling the Type Formers: Π

Proposition

A natural model $p : \dot{\mathsf{U}} \to \mathsf{U}$ *models the rules for* Π*-types just if there are maps* λ, Π *making the following a pullback.*

Proof:

$$\begin{array}{ccc} P(\dot{\mathsf{U}}) & \xrightarrow{\lambda} & \dot{\mathsf{U}} \\ \downarrow & & \downarrow p \\ P(\mathsf{U}) & \xrightarrow[\Pi]{} & \mathsf{U} \end{array}$$

$$\sum_{A:\mathsf{U}} \mathsf{U}^{[A]}$$

3. Modeling the Type Formers: Π

Proposition

A natural model $p : \dot{\mathsf{U}} \to \mathsf{U}$ models the rules for Π-types just if there are maps λ, Π making the following a pullback.

Proof:

$$
\begin{array}{ccc}
P(\dot{\mathsf{U}}) & \xrightarrow{\ \lambda\ } & \dot{\mathsf{U}} \\
\big\downarrow & & \big\downarrow p \\
P(\mathsf{U}) & \xrightarrow[\ \Pi\ ]{} & \mathsf{U}
\end{array}
$$

$$\sum_{A:\mathsf{U}} \mathsf{U}^{[A]}$$

$A \vdash B$ $\qquad\qquad$ $\Pi_A B$

3. Modeling the Type Formers: Π

Proposition

A natural model $p : \dot{\mathsf{U}} \to \mathsf{U}$ models the rules for Π-types just if there are maps λ, Π making the following a pullback.

Proof:

$A \vdash b : B$ $\qquad\qquad$ $\lambda_A b$

$$\begin{array}{ccccc}
\sum_{A:\mathsf{U}} \dot{\mathsf{U}}^{[A]} & & P(\dot{\mathsf{U}}) & \xrightarrow{\lambda} & \dot{\mathsf{U}} \\
 & & \downarrow & & \downarrow p \\
\sum_{A:\mathsf{U}} \mathsf{U}^{[A]} & & P(\mathsf{U}) & \xrightarrow[\Pi]{} & \mathsf{U}
\end{array}$$

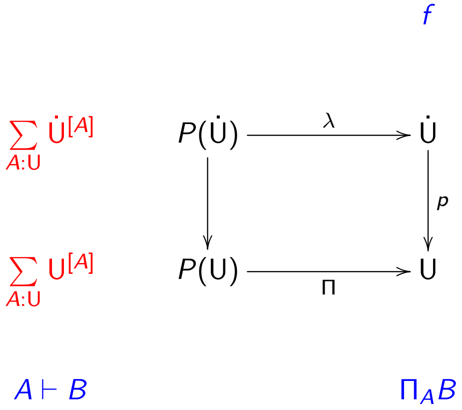
$A \vdash B$ $\qquad\qquad$ $\Pi_A B$

3. Modeling the Type Formers: Π

Proposition

A natural model $p : \dot{\mathsf{U}} \to \mathsf{U}$ models the rules for Π-types just if there are maps λ, Π making the following a pullback.

Proof:



3. Modeling the Type Formers: Π

Proposition

A natural model $p : \dot{\mathsf{U}} \to \mathsf{U}$ models the rules for Π-types just if there are maps λ, Π making the following a pullback.

Proof:

$A \vdash f(x) : B$ $\qquad\qquad$ $\lambda_A f(x) = f$

$$\begin{array}{ccccc}
\sum_{A:\mathsf{U}} \dot{\mathsf{U}}^{[A]} & & P(\dot{\mathsf{U}}) & \xrightarrow{\lambda} & \dot{\mathsf{U}} \\
 & & \downarrow & & \downarrow p \\
\sum_{A:\mathsf{U}} \mathsf{U}^{[A]} & & P(\mathsf{U}) & \xrightarrow[\Pi]{} & \mathsf{U}
\end{array}$$

$A \vdash B$ $\qquad\qquad$ $\Pi_A B$

3. Modeling the Type Formers: Σ

Proposition

A natural model $p : \dot{\mathsf{U}} \to \mathsf{U}$ *models the rules for* Σ*-types just if there are maps* (pair, Σ) *making the following a pullback*

$$\begin{array}{ccc} Q & \xrightarrow{\ \mathsf{pair}\ } & \dot{\mathsf{U}} \\ {\scriptstyle q}\downarrow & & \downarrow{\scriptstyle p} \\ P(\mathsf{U}) & \xrightarrow[\ \Sigma\ ]{} & \mathsf{U} \end{array}$$

where $q : Q \to P(\mathsf{U})$ *is the polynomial composition* $P_q = P \circ P$.

Explicitly:

$$Q \ = \ \sum_{A:\mathsf{U}} \sum_{B:\mathsf{U}^A} \sum_{x:A} B(x)$$

3. Modeling the Type Formers: Strictification

Theorem

Given any Π-**tribe** $(\mathbb{C}, \mathcal{F})$, *for example a Quillen model category with the Frobenius property, the associated natural model* $p_{\mathcal{F}} : \dot{\mathsf{U}}_{\mathcal{F}} \to \mathsf{U}_{\mathcal{F}}$ *under the adjunction*

$$\mathsf{PreClan} \underset{p}{\overset{\mathcal{F}}{\leftrightarrows}} \mathsf{NatMod}\,.$$

has Σ *and* Π *types (as well as* Id*-types).*

The natural model $p_{\mathcal{F}} : \dot{\mathsf{U}}_{\mathcal{F}} \to \mathsf{U}_{\mathcal{F}}$ is thus a **strictification** of the homotopical model $(\mathbb{C}, \mathcal{F})$.

4. A Polynomial Monad

Consider the rules for a terminal type T.

$$\frac{}{\vdash \mathsf{T}} \qquad \frac{}{\vdash * : \mathsf{T}} \qquad \frac{}{x : \mathsf{T} \vdash x = * : \mathsf{T}}$$

Proposition

A natural model $p : \dot{\mathsf{U}} \to \mathsf{U}$ models the rules for a terminal type just if there are maps $(*, \mathsf{T})$ making the following a pullback.

$$\begin{array}{ccc} 1 & \xrightarrow{\;*\;} & \dot{\mathsf{U}} \\ \downarrow & & \downarrow{\scriptstyle p} \\ 1 & \xrightarrow[\;\mathsf{T}\;]{} & \mathsf{U} \end{array}$$

4. A Polynomial Monad

Consider the pullback squares for T and Σ.

$$
\begin{array}{ccc}
1 & \xrightarrow{\;*\;} & \dot{\mathsf{U}} \\
\big\downarrow & & \big\downarrow{\scriptstyle p} \\
1 & \xrightarrow[\;\mathsf{T}\;]{} & \mathsf{U}
\end{array}
\qquad\qquad
\begin{array}{ccc}
Q & \xrightarrow{\;\mathsf{pair}\;} & \dot{\mathsf{U}} \\
{\scriptstyle q}\big\downarrow & & \big\downarrow{\scriptstyle p} \\
P(\mathsf{U}) & \xrightarrow[\;\Sigma\;]{} & \mathsf{U}
\end{array}
$$

These determine cartesian natural transformations between the corresponding polynomial endofunctors.

$$\tau : 1 \Rightarrow P \qquad\qquad \sigma : P \circ P \Rightarrow P$$

4. A Polynomial Monad

Theorem (A-Newstead)

A natural model $p : \dot{\mathsf{U}} \to \mathsf{U}$ *models* $\top$ *and* Σ *types just if the associated polynomial endofunctor* P *has the structure of a* **cartesian monad**.

$$\tau : 1 \Rightarrow P \qquad\qquad \sigma : P \circ P \Rightarrow P$$

4. A Polynomial Monad

The **monad laws** correspond to the following type isomorphisms.

$\sigma \circ P\sigma = \sigma \circ \sigma_P$	$\sum_{a:A} \sum_{b:B(a)} C(a,b) \;\cong\; \sum_{(a,b):\sum_{a:A} B(a)} C(a,b)$
$\sigma \circ P\tau = 1$	$\sum_{a:A} 1 \;\cong\; A$
$\sigma \circ \tau_P = 1$	$\sum_{x:1} A \;\cong\; A$

4. A Polynomial Monad

The pullback square for Π

$$\begin{array}{ccc} P(\dot{\mathsf{U}}) & \xrightarrow{\lambda} & \dot{\mathsf{U}} \\ {\scriptstyle P(p)}\downarrow & & \downarrow{\scriptstyle p} \\ P(\mathsf{U}) & \xrightarrow[\Pi]{} & \mathsf{U} \end{array}$$

determines a cartesian natural transformation

$$\pi : P^{\mathbb{2}}(p) \Rightarrow p$$

where $P^{\mathbb{2}} : \hat{\mathbb{C}}^{\mathbb{2}} \to \hat{\mathbb{C}}^{\mathbb{2}}$ is the lift of P to the arrow category.

4. A Polynomial Monad

So a natural model $p : \dot{\mathsf{U}} \to \mathsf{U}$ models Π types just if it has an **algebra structure** for the lifted endofunctor $P^{\mathbb{2}}$.

$$\pi : P^{\mathbb{2}}(p) \Rightarrow p$$

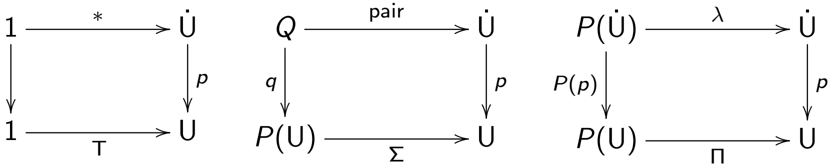
The **algebra laws** correspond to the following type isomorphisms.

$\pi \circ P\pi \ = \ \pi \circ \sigma$	$\prod_{a:A} \prod_{b:B(a)} C(a,b) \ \cong \ \prod_{(a,b):\sum_{a:A} B(a)} C(a,b)$
$\pi \circ \tau \ = \ 1$	$\prod_{x:1} A \ \cong \ A$

Summary: Martin-Löf Algebras

Definition

A **Martin-Löf Algebra** in a locally cartesian closed category $\mathcal{E}$ may be defined as a map $p : \dot{\mathsf{U}} \to \mathsf{U}$ together with pullback squares



Theorem

A homotopical model of (homotopy) type theory determines a natural model $p : \dot{\mathsf{U}} \to \mathsf{U}$ *which*

(i) is a Martin-Löf algebra,

(ii) models the $\top, \Sigma, \Pi$ *type formers, as well as* Id*-types.*

References

1. Awodey, S. (2023) Algebraic type theory, HoTT 2023.
2. Awodey, S. (2017) Natural models of homotopy type theory, MSCS 28(2). arXiv:1406.3219
3. Awodey, S. and C. Newstead (2018) Polynomial pseudomonads and dependent type theory. arXiv:1802.00997
4. Bezem, M., T. Coquand, P. Dybjer, and M. Escardó (2021). On generalized algebraic theories and categories with families. Mathematical Structures in Computer Science, 31(9).
5. Newstead, C. (2018) Algebraic Models of Dependent Type Theory, CMU PhD thesis. arXiv:2103.06155