Tutorial on *Polynomial Functors and Type Theory* Part I

Steve Awodey

Workshop on Polynomial Functors
Topos Institute
March 2022

Outline

1. Polynomials

Let $\mathcal{E}$ be a locally cartesian closed category.
Thus for every map $f : B \to A$ we have adjoint functors on the slice categories,

$$\begin{array}{c} B \\ \downarrow f \\ A \end{array} \qquad\qquad \begin{array}{c} \mathcal{E}/B \\ \Sigma_f \downarrow \quad \uparrow f^* \quad \downarrow \Pi_f \\ \mathcal{E}/A \end{array}$$

When $A = 1$ we write

$$\Sigma_B \dashv B^* \dashv \Pi_B$$

for the corresponding functors determined by $B \to 1$.

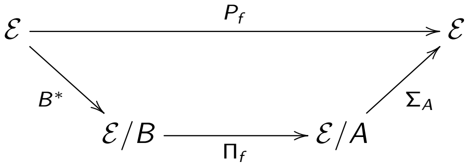
1. Polynomials

Definition

The *polynomial endofunctor* $P_f : \mathcal{E} \longrightarrow \mathcal{E}$ determined by a map

$$f : B \longrightarrow A$$

is the composite



which we may write in the internal language of $\mathcal{E}$ as

$$P_f X = \Sigma_{x:A} \Pi_f B^* X = \Sigma_{x:A} \Pi_f f^* A^* X$$
$$= \Sigma_{x:A} \Pi_f f^* A^* X = \Sigma_{x:A} (A^* X)^f = \Sigma_{x:A} X^{B(x)}.$$

1. Polynomials

$$
\begin{array}{ccccccc}
\mathcal{E} & & & \xrightarrow{\quad P_f \quad} & & & \mathcal{E} \\
& \searrow^{B^*} & & & & \nearrow_{\Sigma_A} & \\
& & \mathcal{E}/B & \xrightarrow[\Pi_f]{} & \mathcal{E}/A & &
\end{array}
$$

The construction of $P_f X$ can be visualized as follows:

$$
\begin{array}{ccccc}
X & \longleftarrow & X \times B & & P_f X \\
& & \downarrow & & \downarrow \\
& & B & \xrightarrow[f]{} & A
\end{array}
$$

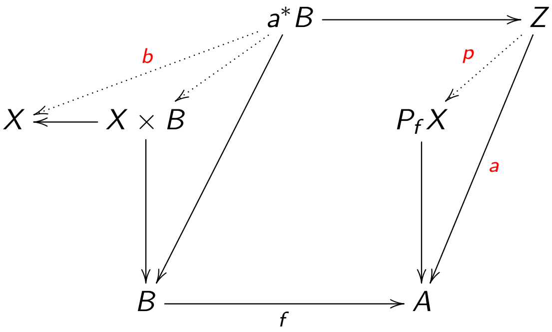
1. Polynomials

Lemma (UMP of $P_f X$)

Maps $p : Z \to P_f X$ correspond naturally to pairs (a, b) where

$$A : Z \to A \qquad b : a^* B \to X.$$

Proof.



□

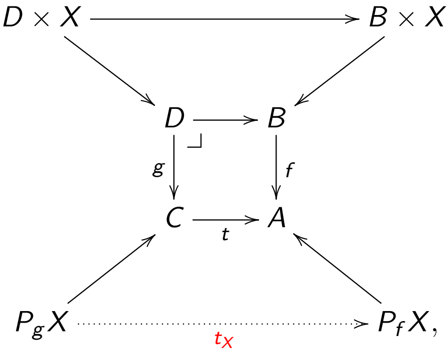
1. Polynomials

Now suppose we have a pullback square

$$\begin{array}{ccc} D & \longrightarrow & B \\ {\scriptstyle g}\downarrow & \lrcorner & \downarrow{\scriptstyle f} \\ C & \xrightarrow[t]{} & A. \end{array}$$

1. Polynomials

Then for each X we get a map $t_X : P_g X \to P_f X$ as follows:

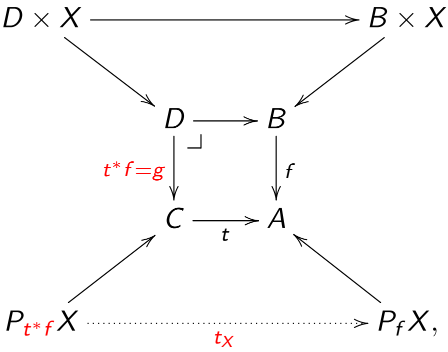


because the lower square is a pullback by Beck-Chavalley,

$$P_g X \cong t^* P_f X.$$

1. Polynomials

Then for each X we get map $t_X : P_g X \to P_f X$ as follows:



because the lower square is a pullback by Beck-Chavalley,

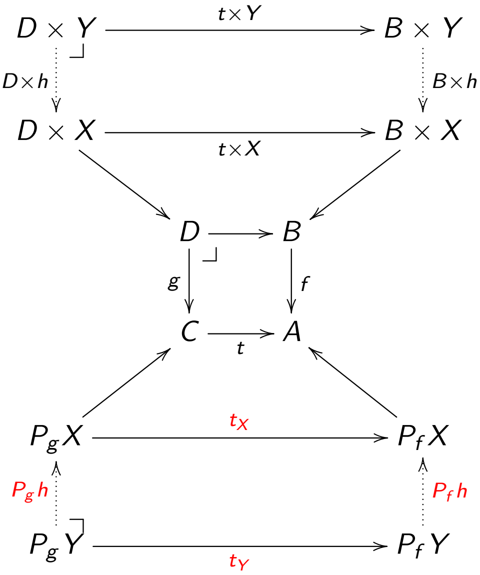
$$P_g X \cong t^* P_f X.$$

Indeed, since $g = t^* f$, we have

$$P_{t^* f} X \cong P_g X \cong t^* P_f X.$$

1. Polynomials

Then for each $h : Y \to X$ we have the pullback square below.



1. Polynomials

Proposition

Taking the polynomial functor $P_f : \mathcal{E} \to \mathcal{E}$ of a map $f : B \to A$ determines a functor

$$P : \mathcal{E}^{\to}_{\text{cart}} \longrightarrow \text{End}(\mathcal{E}).$$

The cartesian squares in $\mathcal{E}^{\to}$ are taken to cartesian natural transformations between endofunctors on $\mathcal{E}$. Moreover, the polynomials are closed under composition.

Proof.

It remains only to show that polynomial functors compose: given any $f : B \to A$ and $g : D \to C$, there is a map $h : F \to E$ such that

$$P_g \circ P_f = P_h : \mathcal{E} \longrightarrow \mathcal{E}.$$

See Spivak (2022) for the definition of $h = g \triangleleft f$. □

2. Dependent type theory

Types:

$$A, B, \ \ldots$$

Terms:

$$x : A, \ \ b : B, \ \ldots$$

Dependent Types ("indexed families of types")

$$x : A \vdash B(x)$$

$$x : A, y : B(x) \vdash C(x, y)$$

$$\ldots$$

Type Forming Operations:

$$\sum_{x:A} B(x), \quad \prod_{x:A} B(x), \quad \ldots$$

Term Forming Operations:

$$\langle a, b \rangle, \quad \lambda x.b(x), \ \ldots$$

Equations:

$$s = t : A$$

2. Dependent type theory: Rules

Contexts:

$$\frac{x : A \vdash B(x)}{x : A,\ y : B(x) \vdash}$$

Writing Γ for any context, we have:

$$\frac{\Gamma \vdash C}{\Gamma, z : C \vdash}$$

2. Dependent type theory: Rules

Sums:

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

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

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

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

2. Dependent type theory: Rules

Sums:

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

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

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

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

2. Dependent type theory: Rules

Sums:

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

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

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

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

2. Dependent type theory: Rules

Products:

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

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

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

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

2. Dependent type theory: Substitution

A tuple of terms in context $\sigma : \Delta \to \Gamma$ induces an operation

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

which preserves *everything*.

For example given $y : Y \vdash s : Z$ and $z{:}Z, x{:}A(z) \vdash B(z,x)$ we can do

$$\frac{y : Y \vdash s : Z \qquad \frac{z{:}Z,\, x{:}A(z) \vdash B(z,x)}{z{:}Z \vdash \prod_{x:A(z)} B(z,x)}}{y : Y \vdash \left(\prod_{x:A(z)} B(z,x)\right)[s/z]} \quad \text{or} \quad \frac{\frac{y{:}Y \vdash s{:}Z \qquad z{:}Z,\, x{:}A(z) \vdash B(z,x)}{y{:}Y,\, x{:}A(s) \vdash B(s,x)}}{y : Y \vdash \prod_{x:A(s)} B(s,x)}$$

and syntactically the results are *the same*,

$$\Big(\prod_{x:A(z)} B(z,x)\Big)[s/z] = \prod_{x:A(s)} B(s,x)\,.$$

This suggests a reformulation as an *indexed algebraic structure*.

3. Natural models

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 representable:

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

Proposition (A, Fiore)

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

3. Natural models

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 representable: for all $C \in \mathbb{C}$ and $x \in X(C)$ there is given $p : D \to C$ and $y \in Y(D)$ such that the following is a pullback:

$$\begin{array}{ccc} \mathrm{y}D & \xrightarrow{\;y\;} & Y \\ {\scriptstyle \mathrm{y}p}\downarrow & \lrcorner & \downarrow{\scriptstyle f} \\ \mathrm{y}C & \xrightarrow[\;x\;]{} & X \end{array}$$

Proposition (A, Fiore)

A representable natural transformation equipped with a choice of such pullbacks is the same thing as a **Category with Families** *in the sense of Dybjer.*

3. Natural models

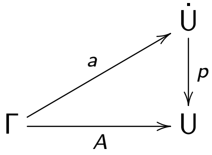
Write the objects and arrows of $\mathbb{C}$ as $\sigma : \Delta \to \Gamma$, thinking of a *category of contexts and substitutions*.

Let $p : \dot{\mathsf{U}} \to \mathsf{U}$ be a representable map of presheaves on $\mathbb{C}$.

Think of U as the *presheaf of types*, $\dot{\mathsf{U}}$ as the *presheaf of terms*, and then p gives the type of a term.

$$\Gamma \vdash A \quad \approx \quad A \in \mathsf{U}(\Gamma)$$
$$\Gamma \vdash a : A \quad \approx \quad a \in \dot{\mathsf{U}}(\Gamma)$$

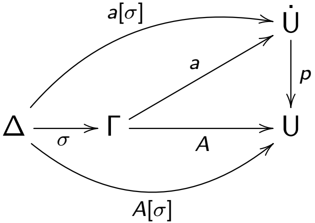
where $A = p \circ a$.



3. Natural models

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

$$\Gamma \vdash A \quad \Rightarrow \quad \Delta \vdash A[\sigma]$$
$$\Gamma \vdash a : A \quad \Rightarrow \quad \Delta \vdash a[\sigma] : A[\sigma]$$



3. Natural models

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

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

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

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

This is the basic structure of a CwF.

2. Natural models, context extension

The remaining operation of **context extension**

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

is modeled by the representability of $p : \dot{\mathrm{U}} \to \mathrm{U}$ as follows.

3. 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

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

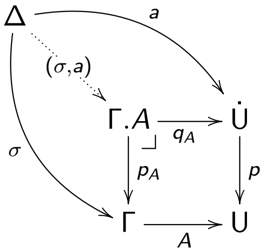
Let $p_A : \Gamma.A \to \Gamma$ be the pullback of p along A.

$$\begin{array}{ccc} \Gamma.A & \xrightarrow{q_A} & \dot{\mathrm{U}} \\ {\scriptstyle p_A}\downarrow & \lrcorner & \downarrow{\scriptstyle p} \\ \Gamma & \xrightarrow[A]{} & \mathrm{U} \end{array}$$

The map $q_A : \Gamma.A \to \dot{\mathrm{U}}$ gives the required term $\Gamma.A \vdash q_A : A[p_A]$. Syntactically, this is just the term

$$\Gamma, x\!:\!A \vdash x\!:\!A\,.$$

3. Natural models, context extension



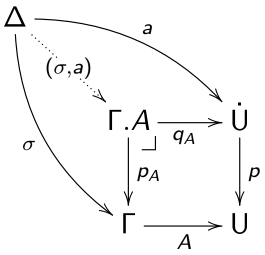
The pullback means that given any substitution $\sigma : \Delta \to \Gamma$ and term $\Delta \vdash a : A[\sigma]$ there is a map

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

satisfying

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

3. 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 *all* the laws for a CwF. □

3. Natural models, algebraic formulation

Natural models can be presented as an essentially algebraic theory, with several sorts, partial operations, and equations between terms.

We have four basic sorts:

$$C_0,\ C_1,\ A,\ B$$

and the following operations and equations:

category: the usual domain, codomain, identity and composition operations for the index category $\mathbb{C}$:

$$C_1 \times_{C_0} C_1 \xrightarrow{\circ} C_1 \;\underset{\text{dom}}{\overset{\text{cod}}{\rightrightarrows}}\; C_0 \quad (\text{id} : C_0 \to C_1),$$

together with the familiar equations for a category.

3. Natural models, algebraic formulation

presheaf: the indexing and action operations for the presheaves $A, B : \mathbb{C}^{\mathsf{op}} \to \mathsf{Set}$:

$$
\begin{array}{ccc}
C_1 \times_{C_0} A & \xrightarrow{\alpha} & A \\
 & & \downarrow{\scriptstyle p_A} \\
 & & C_0
\end{array}
\qquad\qquad
\begin{array}{ccc}
C_1 \times_{C_0} B & \xrightarrow{\beta} & B \\
 & & \downarrow{\scriptstyle p_B} \\
 & & C_0
\end{array}
$$

together with the equations making α an action:

$$
\begin{aligned}
p_A(\alpha(u, a)) &= \mathrm{dom}(u), \\
\alpha(u \circ v, a) &= \alpha(v, \alpha(u, a)), \\
\alpha(1_{p_A(a)}, a) &= a,
\end{aligned}
$$

and similarly for β.

3. Natural models, algebraic formulation

natural transformation: an operation

$$f : A \to B$$

satisfying the naturality equations:

$$p_B \circ f = p_A, \qquad f \circ \alpha = \beta \circ (C_1 \times_{C_0} f).$$

representable: a natural transformation $f : A \to B$ is representable just if the associated functor,

$$\textstyle\int_{\mathbb{C}} f : \int_{\mathbb{C}} A \to \int_{\mathbb{C}} B$$

on the categories of elements has a right adjoint

$$f^* : \textstyle\int_{\mathbb{C}} B \to \int_{\mathbb{C}} A$$

(an algebraic condition, see Newstead (2018)).

3. Natural models and initiality

- The notion of a natural model is thus *essentially algebraic*.
- The algebraic homomorphisms correspond exactly to syntactic translations.
- There is an *initial algebra* as well as a *free algebra* over any signature of basic types and terms.
- The rules of dependent type theory specify a procedure for generating a free algebra.

3. Natural models and tribes

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

The fibration

$$\int_{\mathbb{C}} \mathsf{U} \to \mathbb{C}$$

of all *display maps* $p_A : \Gamma.A \to \Gamma$, for all $A : \Gamma \to \mathsf{U}$, determines a *clan* in the sense of Joyal (2017).

Conversely, given a clan $\mathcal{D} \hookrightarrow \mathbb{C}^{\to}$, there is a natural model in $\hat{\mathbb{C}}$,

$$\coprod_{f \in \mathcal{D}} \mathsf{y}f : \coprod_{f \in \mathcal{D}} \mathsf{y}\mathrm{dom}(f) \longrightarrow \coprod_{f \in \mathcal{D}} \mathsf{y}\mathrm{cod}(f).$$

This natural model $p_{\mathcal{D}} : \dot{\mathsf{U}}_{\mathcal{D}} \to \mathsf{U}_{\mathcal{D}}$ determines a *splitting* of the associated fibration $\mathcal{D} \to \mathbb{C}$.

3. Natural models and tribes

Theorem (ish)

There is an adjunction between the categories of clans and of natural models, which specializes to a biequivalence between (certain) tribes and natural models with (certain) type-forming operations.

See A. (2017) for details.

References for Part I

1. Awodey, S. (2017) Natural models of homotopy type theory, MSCS 28(2). arXiv:1406.3219

2. Dybjer, P. (1995) Internal Type Theory. Types 1995.

3. Joyal, A. (2017) Notes on clans and tribes. arXiv:1710.10238

4. Spivak, D. (2022) A summary of categorical structures in Poly. arXiv:2202.00534