

Notes on Algebraic Type Theory

Steve Awodey

December 9, 2023

1 Natural models of type theory

We write $\hat{\mathbb{C}} = [\mathbb{C}^{\text{op}}, \text{Set}]$ for the category of presheaves on a small category \mathbb{C} . In (??), a *natural model* of type theory is defined to be a representable natural tranformation $t : \dot{T} \rightarrow T$ of presheaves on a small category \mathbb{C} . Theorem XX shows that such a map is equivalent to the notion of a *category with families* in the sense of [?].

A natural model $t : \dot{T} \rightarrow T$ will be said to *model* the type-constructors $1, \Sigma, \Pi$ if there are pullback squares in \mathbb{C} of the following form,

$$\begin{array}{ccc} 1 & \xrightarrow{*} & \dot{T} \\ \downarrow ! & \lrcorner & \downarrow t \\ 1 & \xrightarrow{1} & T \end{array} \quad \begin{array}{ccc} \dot{T}_2 & \xrightarrow{\sigma} & \dot{T} \\ \downarrow t \cdot t & \lrcorner & \downarrow t \\ T_2 & \xrightarrow{\Sigma} & T \end{array} \quad \begin{array}{ccc} P_t(\dot{T}) & \xrightarrow{\lambda} & \dot{T} \\ P_t(t) \downarrow & \lrcorner & \downarrow t \\ P_t(T) & \xrightarrow{\Pi} & T \end{array} \quad (1)$$

where

1. $P_t = T_! \circ t_* \circ \dot{T}^* : \hat{\mathbb{C}} \rightarrow \hat{\mathbb{C}}$ is the polynomial endofunctor determined uniquely by $t : \dot{T} \rightarrow T$ (its *signature*, cf. [?]),
2. $t \cdot t : \dot{T}_2 \rightarrow T_2$ is defined by $P_{t \cdot t} = P_t \circ P_t$.

The terminology is justified by the following result, also from [?].

Theorem 1. *Let $t : \dot{T} \rightarrow T$ be a natural model. The associated category with families satisfies the usual rules for the type-constructors $1, \Sigma, \Pi$ just if $t : \dot{T} \rightarrow T$ models them, in the sense of the diagrams (1).*

1.1 Display maps and clans

[define \mathcal{D}_t and show that it is a display map category with ...]

[given a display map category \mathcal{D} with ... define \mathbf{d} and show that it is ...]

[the constructions are not mutually inverse but rather adjoint inverse ...]

In these terms, we have the following description of the type-constructors $\Pi, \Sigma, 1$: Given a natural model

Theorem 2 ([?], XX). *Let $t : \dot{T} \rightarrow T$ be a natural model of type theory. Then*

2. Martin-Löf algebras

Now let \mathcal{E} be a locally cartesian closed category and $t : \dot{T} \rightarrow T$ any map in \mathcal{E} . As in the representable case, t gives rise to a polynomial endofunctor $P_t : \mathcal{E} \rightarrow \mathcal{E}$, in terms of which we define the following abstraction of the notion of a natural model.

Definition 3. A *Martin-Löf algebra* in \mathcal{E} is a map $t : \dot{T} \rightarrow T$ equipped with structure maps $(*, 1, \sigma, \Sigma, \lambda, \Pi)$ making pullback squares

$$\begin{array}{ccccc} 1 & \xrightarrow{*} & \dot{T} & & \dot{T}^2 & \xrightarrow{\sigma} & \dot{T} & & \dot{T}' & \xrightarrow{\lambda} & \dot{T} \\ \downarrow ! & \lrcorner & \downarrow t & & \downarrow t^2 & \lrcorner & \downarrow t & & \downarrow t' & \lrcorner & \downarrow t \\ 1 & \xrightarrow{1} & T & & T^2 & \xrightarrow{\Sigma} & T & & T' & \xrightarrow{\Pi} & T \end{array} \quad (2)$$

where the maps t^2 and t' are defined in terms of the polynomial endofunctor

$$P_t = T_! \circ t_* \circ \dot{T}^* : \mathcal{E} \longrightarrow \mathcal{E}$$

as in the foregoing section.

In place of representability, we may require that $t : \dot{T} \rightarrow T$ be *tiny*.

Definition 4. A map $f : A \rightarrow B$ in a locally cartesian closed category \mathcal{E} is *tiny* if it is so as an object in $\mathcal{E}/_B$, in the sense that exponentiation by f has a right adjoint $(-)^f \dashv (-)_f$.

Lemma 5. *A map $f : A \rightarrow B$ is tiny just if the pushforward $f_* : \mathcal{E}/_A \rightarrow \mathcal{E}/_B$ has a right adjoint,*

$$f_* \dashv f^! : \mathcal{E}/_B \longrightarrow \mathcal{E}/_A.$$

Proposition 6. *Any representable natural transformation $f : A \rightarrow B$ between presheaves on a small category \mathbb{C} is a tiny map in $\hat{\mathbb{C}}$.*

Proof. See [?]. Briefly, ...

□