Algebraic models of dependent type theory

Clive Newstead

HoTT/UF Workshop 2018
Oxford, UK
(in absentia)

Saturday 7th July 2018

These slides: https://goo.gl/Ttacdq
1 Natural models

2 Connection with polynomial functors

3 Natural model semantics

4 Concluding remarks
1. Natural models

2. Connection with polynomial functors

3. Natural model semantics

4. Concluding remarks
Representable natural transformations

A map \( f : Y \to X \) in \( \widehat{C} := [C^{op}, \text{Set}] \) is representable if

\[
\begin{tikzcd}
Y \\
\downarrow f \\
X
\end{tikzcd}
\]
Representable natural transformations

A map \( f : Y \to X \) in \( \hat{C} := [C^{op}, \text{Set}] \) is **representable** if for all \( A \in C \) and \( x \in X(A) \) there exist \( g : B \to A \) in \( C \) and \( y \in Y(B) \) such that the following square is a pullback:

\[
\begin{array}{ccc}
Y & \xrightarrow{y} & X \\
\downarrow & & \downarrow \\
y(A) & \xrightarrow{x} & X
\end{array}
\]
A map $f : Y \to X$ in $\widehat{\mathcal{C}} := [\mathcal{C}^{\text{op}}, \text{Set}]$ is **representable** if for all $A \in \mathcal{C}$ and $x \in X(A)$ there exist $g : B \to A$ in $\mathcal{C}$ and $y \in Y(B)$.
A map \( f : Y \to X \) in \( \hat{\mathcal{C}} := [\mathcal{C}^{\text{op}}, \text{Set}] \) is \textbf{representable} if for all \( A \in \mathcal{C} \) and \( x \in X(A) \) there exist \( g : B \to A \) in \( \mathcal{C} \) and \( y \in Y(B) \) such that the following square is a pullback:

\[
\begin{array}{ccc}
 y(B) & \xrightarrow{y} & Y \\
 \downarrow & & \downarrow \\
 y(A) & \xrightarrow{x} & X
\end{array}
\]
A **natural model** is a representable natural transformation.
Natural models

A **natural model** is a representable natural transformation.
A natural model consists of:

- A base category $C$ (of 'contexts' and 'substitutions')
- A chosen terminal object $\diamond$ (the 'empty context')
- Presheaves $U$ and $\cdot U$ (of 'types-in-context' and 'terms-in-context')
- A map of presheaves $p: \cdot U \to U$ (term $\mapsto$ its type)
- Data witnessing representability of $p$:
  \[
  \forall \Gamma, A \exists (\text{chosen}) \Gamma \cdot U, p A, q A y (\Gamma \cdot A) q A q A y (p A) y (p A)
  \]
A **natural model** consists of:

- A base category \( \mathbb{C} \) (of ‘contexts’ and ‘substitutions’);
A **natural model** consists of:

- A base category $\mathcal{C}$ (of ‘contexts’ and ‘substitutions’);
- ... with chosen terminal object $\diamond$ (the ‘empty context’);
Natural models

A natural model consists of:

- A base category $\mathcal{C}$ (of ‘contexts’ and ‘substitutions’);
- ... with chosen terminal object $\Diamond$ (the ‘empty context’);
- Presheaves $\mathcal{U}$ and $\mathcal{U}$ (of ‘types-in-context’ and ‘terms-in-context’);
Natural models

A natural model consists of:

- A base category $\mathbb{C}$ (of ‘contexts’ and ‘substitutions’);
- ... with chosen terminal object $\diamond$ (the ‘empty context’);
- Presheaves $\mathcal{U}$ and $\mathcal{U}$ (of ‘types-in-context’ and ‘terms-in-context’);
- A map of presheaves $p : \mathcal{U} \to \mathcal{U}$ (term $\mapsto$ its type);
A **natural model** consists of:

- A base category $\mathbb{C}$ (of ‘contexts’ and ‘substitutions’);
- ... with chosen terminal object $\diamond$ (the ‘empty context’);
- Presheaves $\mathcal{U}$ and $\mathcal{U}'$ (of ‘types-in-context’ and ‘terms-in-context’);
- A map of presheaves $p : \mathcal{U}' \to \mathcal{U}$ (term $\mapsto$ its type);
- + data witnessing representability of $p$:

\[
\begin{array}{c}
\mathcal{U}' \\
\downarrow p \\
\mathcal{U}
\end{array}
\]
Natural models

A **natural model** consists of:

- A base category $\mathbb{C}$ (of ‘contexts’ and ‘substitutions’);
- . . . with chosen terminal object $\diamond$ (the ‘empty context’);
- Presheaves $\mathcal{U}$ and $\mathcal{U}$ (of ‘types-in-context’ and ‘terms-in-context’);
- A map of presheaves $p : \mathcal{U} \rightarrow \mathcal{U}$ (term $\mapsto$ its type);
- + data witnessing representability of $p:\n\forall \Gamma, A

\[
\begin{array}{c}
\mathcal{U} \\
p \\
\downarrow \\
y(\Gamma) \xrightarrow{A} \mathcal{U}
\end{array}
\]
Natural models

A **natural model** consists of:

- A base category $\mathbf{C}$ (of ‘contexts’ and ‘substitutions’);
- . . . with chosen terminal object $\mathbf{⋄}$ (the ‘empty context’);
- Presheaves $\mathbf{U}$ and $\dot{\mathbf{U}}$ (of ‘types-in-context’ and ‘terms-in-context’);
- A map of presheaves $p : \dot{\mathbf{U}} \to \mathbf{U}$ (term $\mapsto$ its type);
- + data witnessing representability of $p$:
  \[
  \forall \Gamma, A \exists \text{(chosen)} \quad \Gamma \cdot A, p_A, q_A
  \]

\[
\begin{array}{ccc}
  y(\Gamma \cdot A) & \xrightarrow{q_A} & \dot{\mathbf{U}} \\
  y(p_A) & \downarrow & \downarrow p \\
  y(\Gamma) & \xrightarrow{A} & \mathbf{U}
\end{array}
\]
Examples of natural models at work

Type theory:

\[ \Gamma \vdash a : A \]
Examples of natural models at work

Type theory:

\[ \Gamma \vdash a : A \]

Natural model:

\[ \begin{array}{c}
\text{y(\(\Gamma\))} \\
\downarrow \text{A} \\
\text{U}
\end{array} \xrightarrow{a} \begin{array}{c}
\text{U} \\
\downarrow \text{p}
\end{array} \xrightarrow{\dot{\mathcal{U}}} \dot{\mathcal{U}} \]
Examples of natural models at work

Type theory:

\[
\frac{\Gamma \vdash A \text{ type} \quad \Gamma, x : A \vdash B(x) \text{ type}}{\Gamma \vdash T(A, B) \text{ type} \quad (T\text{-FORM})}
\]
Examples of natural models at work

Type theory:

\[
\frac{\Gamma \vdash A \text{ type} \quad \Gamma, x : A \vdash B(x) \text{ type}}{\Gamma \vdash T(A, B) \text{ type}} \quad (T\text{-FORM})
\]

Natural model:

\[
T : \sum_{A : U} U\dot{U}_A \rightarrow U
\]
Examples of natural models at work

Type theory:

\[
\frac{\Gamma \vdash A \text{ type} \quad \Gamma, x : A \vdash B(x) \text{ type}}{\Gamma \vdash T(A, B) \text{ type}} \quad (T\text{-FORM})
\]

Natural model:

\[
T : \sum_{A : \mathcal{U}} \mathcal{U}^{[A]} \rightarrow \mathcal{U}
\]
1. Natural models

2. Connection with polynomial functors

3. Natural model semantics

4. Concluding remarks
Polynomials and polynomial functors

Fix a locally cartesian closed category $\mathcal{E}$. 
Polynomials and polynomial functors

Fix a locally cartesian closed category $\mathcal{E}$.

$f : B \to A \rightsquigarrow P_f : \mathcal{E} \to \mathcal{E}$
Polynomials and polynomial functors

Fix a locally cartesian closed category $\mathcal{E}$.

\[ f : B \to A \quad \rightsquigarrow \quad P_f : \mathcal{E} \to \mathcal{E} \]

\[ X \mapsto \sum_{a : A} X^{B_a} \]
Polynomials and polynomial functors

Fix a locally cartesian closed category $\mathcal{E}$.

$$f : B \rightarrow A \rightsquigarrow P_f : \mathcal{E} \rightarrow \mathcal{E}$$

$$X \mapsto \sum_{a : A} X^{B_a}$$

Call $P_f$ a **polynomial endofunctor** and $f$ a **polynomial**.
Polynomials and polynomial functors

Fix a locally cartesian closed category $\mathcal{E}$. 

\[
f : B \to A \quad \leadsto \quad P_f : \mathcal{E} \to \mathcal{E} \quad X \mapsto \sum_{a : A} X^{B_a}
\]

Call $P_f$ a **polynomial endofunctor** and $f$ a **polynomial**.

Officially, $P_f$ is the composite

\[
\mathcal{E} \xrightarrow{\Delta_B \to 1} \mathcal{E}/B \xrightarrow{\Pi_f} \mathcal{E}/A \xrightarrow{\Sigma_{A \to 1}} \mathcal{E}
\]

where $\Delta_f$ is pullback along $f$ and $\Sigma_f \dashv \Delta_f \dashv \Pi_f$. 
Cartesian morphisms of polynomials

\[ \varphi : P_f \Rightarrow P_g \]

\[ \leadsto \text{cartesian natural transformation} \]
Cartesian morphisms of polynomials

\[
\begin{align*}
B \xrightarrow{\varphi_1} D \\
\downarrow f \quad \quad \downarrow g \\
A \xrightarrow{\varphi_0} C
\end{align*}
\]

\[\varphi : P_f \Rightarrow P_g\]
cartesian natural transformation
Cartesian morphisms of polynomials

\[
\begin{array}{ccc}
B & \xrightarrow{\varphi_1} & D \\
\downarrow f & & \downarrow g \\
A & \xrightarrow{\varphi_0} & C
\end{array}
\]

\(\varphi : P_f \Rightarrow P_g\)
cartesian natural transformation

Theorem (Gambino & Kock)

- Polynomials and cartesian morphisms are the 1- and 2-cells of a bicategory;
Cartesian morphisms of polynomials

\[
\begin{array}{ccc}
B & \xrightarrow{\varphi_1} & D \\
\downarrow f & & \downarrow g \\
A & \xrightarrow{\varphi_0} & C \\
\end{array}
\]

\[\varphi : P_f \Rightarrow P_g\]
cartesian natural transformation

**Theorem (Gambino & Kock)**

- **Polynomials and cartesian morphisms are the 1- and 2-cells of a bicategory;**
- **Polynomial functors and cartesian natural transformations are the 1- and 2-cells of a 2-category;**
Cartesian morphisms of polynomials

\[ \begin{array}{ccc}
B & \overset{\varphi_1}{\longrightarrow} & D \\
f \downarrow & & \downarrow g \\
A & \overset{\varphi_0}{\longrightarrow} & C
\end{array} \quad \varphi : P_f \Rightarrow P_g 
\]
cartesian natural transformation

Theorem (Gambino & Kock)

- Polynomials and cartesian morphisms are the 1- and 2-cells of a bicategory;
- Polynomial functors and cartesian natural transformations are the 1- and 2-cells of a 2-category;
- These are biequivalent.
Admitting a unit type

Theorem (Awodey)

A natural model admits a unit type \( \iff \) there exist \( \hat{1}, \hat{\star} \) as in:

\[
\begin{array}{c}
\hat{U} \\
\downarrow p \\
\hat{U}
\end{array}
\]
Admitting a unit type

**Theorem (Awodey)**

*A natural model admits a unit type ⇔ there exist \( \hat{1}, \hat{*} \) as in:*

\[
\begin{align*}
y(\Diamond) & \xrightarrow{\hat{*}} \mathcal{U} \\
y(\Diamond) & \xrightarrow{\hat{1}} \mathcal{U}
\end{align*}
\]
Admitting a unit type

**Theorem (Awodey)**

*A natural model admits a unit type ⇔ there exist \( \hat{1}, \hat{\star} \) as in:*

\[
\begin{array}{ccc}
y(\diamond) & \xrightarrow{\hat{\star}} & \mathcal{U} \\
\downarrow & & \downarrow p \\
y(\diamond) & \xrightarrow{\hat{1}} & \mathcal{U}
\end{array}
\]
Admitting a unit type

**Theorem (Awodey)**

*A natural model admits a unit type $\iff$ there exist $\hat{1}, \hat{\ast}$ as in:*

\[
\begin{align*}
\begin{array}{c}
y(\diamond) \quad \hat{\ast} \\
\_ \quad \\
y(\diamond) \quad \hat{1}
\end{array}
\end{align*}
\begin{array}{c}
\quad \quad U \\
\downarrow p
\end{array}
\]

**Corollary**

*interpretations of unit types $\leftrightarrow$ cartesian morphisms of polynomials*
Admitting a unit type

**Theorem (Awodey)**

A natural model admits a unit type ⇔ there exist $\hat{1}$, $\hat{\ast}$ as in:

$$
\begin{array}{c}
y(\Diamond) \quad \xrightarrow{\hat{\ast}} \quad U \\
\parallel \quad \downarrow \quad \downarrow p \\
y(\Diamond) \quad \xrightarrow{\hat{1}} \quad U
\end{array}
$$

**Corollary**

interpretations of unit types ⇔ cartesian morphisms of polynomials $1 \Rightarrow p$
Admitting dependent sum types

**Theorem (Awodey)**

*A natural model admits a $\Sigma$-types $\iff$ there exist $\hat{\Sigma}$, pair as in:*

\[
\begin{array}{c}
\hat{\Sigma} \\
p \\
\downarrow \\
\mathcal{U}
\end{array}
\]

Note also that $P \pi = P p \circ P p$. 

**Corollary**

interpretations of $\Sigma$-types $\leftrightarrow$ cartesian morphisms of polynomials $p \cdot p \Rightarrow p$. 

Clive Newstead (cnewstead@cmu.edu) Carnegie Mellon University

Algebraic models of dependent type theory

These slides: https://goo.gl/Ttacdq
Admitting dependent sum types

Theorem (Awodey)

A natural model admits a Σ-types ⇔ there exist \( \hat{\Sigma} \), \( \hat{\text{pair}} \) as in:

\[
\sum_{A:U} \sum_{B:U[A]} \sum_{a:A} [B(a)] \xrightarrow{\text{pair}} \hat{\Sigma} \xrightarrow{\hat{\text{pair}}} \hat{\mathcal{U}}
\]

\[
\sum_{A:U} U[A] \xrightarrow{\hat{\Sigma}} \mathcal{U}
\]

Note also that \( P^\pi = P \circ P \).
Admitting dependent sum types

**Theorem (Awodey)**

*A natural model admits a \( \Sigma \)-types <\( \iff \) there exist \( \hat{\Sigma}, \hat{\text{pair}} \) as in:*

\[
\sum_{A:U} \sum_{B:U[A]} \sum_{a:[A]} [B(a)] \xrightarrow{\text{pair}} \hat{\Sigma} \\
\pi \downarrow \quad \quad \quad \quad \quad \quad \quad p \\
\sum_{A:U} U[A] \xrightarrow{\hat{\text{pair}}} U
\]
Admitting dependent sum types

Theorem (Awodey)

A natural model admits a \( \Sigma \)-types \( \iff \) there exist \( \hat{\Sigma}, \hat{\text{pair}} \) as in:

\[
\sum_{A:U} \sum_{B:U[A]} \sum_{a:[A]} [B(a)] \xrightarrow{\text{pair}} U
\]

\[
\sum_{A:U} U[A] \xrightarrow{\hat{\Sigma}} U
\]

\[
\pi \downarrow \quad \downarrow \quad \downarrow p
\]

Note also that \( P \pi = P p \circ P p \).

Corollary interpretations of \( \Sigma \)-types \( \leftrightarrow \) cartesian morphisms of polynomials \( p \cdot p \Rightarrow p \).
Admitting dependent sum types

Theorem (Awodey)

A natural model admits a $\Sigma$-types $\iff$ there exist $\hat{\Sigma}$, $\hat{\text{pair}}$ as in:

\[
\begin{array}{c}
\sum_{A:U} \sum_{B:[A]} \sum_{a:[A]} [B(a)] \\
\sum \left[ B(\pi(a)) \right] \\
\sum_{A:U} U^A \\
\sum \left[ U(\pi(a)) \right]
\end{array}
\xrightarrow{\text{pair}}
U
\xrightarrow{\hat{\text{pair}}}
\hat{U}
\xrightarrow{\hat{\pi}}
\hat{U}
\xrightarrow{p}
P
\]

Note also that $P_\pi = P_p \circ P_p$. 
Admitting dependent sum types

Theorem (Awodey)

A natural model admits a \( \Sigma \)-types \( \iff \) there exist \( \hat{\Sigma}, \hat{\text{pair}} \) as in:

\[
\sum_{A:U} \sum_{B:[A]} \sum_{a:[A]} [B(a)] \xrightarrow{\text{pair}} U
\]

\[
\sum_{A:U} U^[[A]] \xrightarrow{\hat{\Sigma}} U
\]

Note also that \( P_\pi = P_p \circ P_p \).

Corollary

interpretations of \( \Sigma \)-types \( \iff \) cartesian morphisms of polynomials \( p \cdot p \Rightarrow p \)
Admitting dependent product types

**Theorem (Awodey)**

_A natural model admits a \( \Pi \)-types \( \iff \) there exist \( \hat{\Pi}, \hat{\lambda} \) as in:

\[
\begin{array}{c}
\hat{\Pi} \\
\downarrow \\
U
\end{array}
\]

\[
\begin{array}{c}
\hat{\lambda} \\
\downarrow \\
U
\end{array}
\]

\[
\sum_{A:U} U[A] \sum_{A:U} U[A] \sum_{A:U} p[A] \sum_{A:U} p[A] \downarrow\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]

\[
\downarrow
\]
Admitting dependent product types

Theorem (Awodey)

*A natural model admits a $\Pi$-types $\iff$ there exist $\hat{\Pi}$, $\hat{\lambda}$ as in:*

\[
\sum_{A:U} U^[[A]] \xrightarrow{\hat{\lambda}} U \\
\sum_{A:U} U^[[A]] \xrightarrow{\hat{\Pi}} U
\]
Admitting dependent product types

Theorem (Awodey)

A natural model admits a \( \Pi \)-types \( \iff \) there exist \( \hat{\Pi}, \hat{\lambda} \) as in:

\[
\begin{align*}
\sum_{A:U} \mathcal{U}^{[A]} & \xrightarrow{\lambda} \mathcal{U} \\
\sum_{A:U} p^{[A]} & \downarrow \\
\sum_{A:U} \mathcal{U}^{[A]} & \xrightarrow{\hat{\Pi}} \mathcal{U}
\end{align*}
\]
Admitting dependent product types

**Theorem (Awodey)**

*A natural model admits a $\Pi$-types $\iff$ there exist $\hat{\Pi}$, $\hat{\lambda}$ as in:*

\[
\begin{align*}
\sum_{A:U} U^[[A]] & \rightarrow^\hat{\lambda} U \\
\sum_{A:U} p^[[A]] & \downarrow \quad \\
\sum_{A:U} U^[[A]] & \rightarrow^\hat{\Pi} U
\end{align*}
\]
Admitting dependent product types

Theorem (Awodey)

A natural model admits a \( \Pi \)-types \( \iff \) there exist \( \hat{\Pi}, \hat{\lambda} \) as in:

\[
\begin{align*}
\sum_{A:U} U^[[A]] & \xrightarrow{\hat{\lambda}} U \\
\sum_{A:U} p^[[A]] & \Downarrow \quad \Downarrow p \\
\sum_{A:U} U^[[A]] & \xrightarrow{\hat{\Pi}} U
\end{align*}
\]

Corollary

interpretations of \( \Pi \)-types \( \iff \) cartesian morphisms of polynomials \( P_p(p) \Rightarrow p \)
Monad and algebra?

In summary:

\[ n.m. \text{ admits} \ldots \iff \exists \text{ cartesian} \ldots \]
Monad and algebra?

In summary:

\[
\begin{array}{c}
n.m. \text{ admits} \ldots \\
\equiv \\
\exists \text{ cartesian} \ldots \\
\end{array}
\]

\[
\begin{array}{c}
1 \\
\Rightarrow \\
p \\
\end{array}
\]

This is almost a monad and an algebra.

Goal. Find the appropriate notion of 3-cell (morphism of morphisms of polynomials) allowing us to make this more precise.
Monad and algebra?

In summary:

\[
\begin{align*}
\text{n.m. admits...} & \iff \exists \text{ cartesian...} \\
1 & \Rightarrow p \\
\Sigma & \Rightarrow p \\
p \cdot p & \Rightarrow p
\end{align*}
\]
Monad and algebra?

In summary:

\[
\begin{align*}
n.m. \text{ admits } \ldots & \iff \exists \text{ cartesian } \ldots \\
1 & \Rightarrow p \\
\Sigma & \Rightarrow p \\
\Pi & \Rightarrow p
\end{align*}
\]
Monad and algebra?

In summary:

\[
\begin{align*}
n.m. \text{ admits} & \iff \exists \text{ cartesian} \\
1 & \Rightarrow p \\
\Sigma & \Rightarrow p \\
\Pi & \Rightarrow p
\end{align*}
\]

This is a monad and an algebra.
Monad and algebra?

In summary:

\[
\begin{array}{c|c}
\text{n.m. admits...} & \exists \text{ cartesian...} \\
\hline
1 & 1 \Rightarrow p \\
\Sigma & p \cdot p \Rightarrow p \\
\Pi & P(p) \Rightarrow p \\
\end{array}
\]

This is almost a monad and an algebra.
### Monad and algebra?

In summary:

<table>
<thead>
<tr>
<th>n.m. admits...</th>
<th>⇔</th>
<th>∃ cartesian...</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1  ⇒  p</td>
<td></td>
</tr>
<tr>
<td>Σ</td>
<td>p · p ⇒ p</td>
<td></td>
</tr>
<tr>
<td>Π</td>
<td>P(p) ⇒ p</td>
<td></td>
</tr>
</tbody>
</table>

This is almost a monad and an algebra.

**Goal.** Find the appropriate notion of 3-cell (morphism of morphisms of polynomials) allowing us to make this more precise.
Full internal subcategories

Given any morphism \( f : B \to A \) in a locally cartesian closed category \( \mathcal{E} \), we can form the **full internal subcategory** \( \mathcal{S}(f) \in \text{Cat}(\mathcal{E}) \).
Given any morphism $f : B \to A$ in a locally cartesian closed category $\mathcal{E}$, we can form the **full internal subcategory** $\mathcal{S}(f) \in \text{Cat}(\mathcal{E})$.

- Object of objects $= A$;
Full internal subcategories

Given any morphism $f : B \to A$ in a locally cartesian closed category $\mathcal{E}$, we can form the **full internal subcategory** $S(f) \in \text{Cat}(\mathcal{E})$.

- Object of objects $= A$;
- Object of morphisms $= \sum_{a, a' \in A} B_{a'}^{Ba}$
Full internal subcategories

Given any morphism $f : B \to A$ in a locally cartesian closed category $\mathcal{E}$, we can form the **full internal subcategory** $S(f) \in \mathbf{Cat}(\mathcal{E})$.

- Object of objects = $A$;
- Object of morphisms = $\sum_{a, a' \in A} B_{a'}^{B_a} = (\pi_2^* f) \pi_1^* f$ over $A \times A$. 

Clive Newstead (cnewstead@cmu.edu)
Algebraic models of dependent type theory

These slides: https://goo.gl/Ttacdq
Full internal subcategories

Given any morphism \( f : B \to A \) in a locally cartesian closed category \( \mathcal{E} \), we can form the **full internal subcategory** \( \mathcal{S}(f) \in \text{Cat}(\mathcal{E}) \).

- Object of objects = \( A \);
- Object of morphisms = \( \sum_{a, a' \in A} B_{a'}^{B_a} = (\pi_2^* f) \pi_1^* f \) over \( A \times A \).

Cartesian morphisms of polynomials \( \varphi : f \Rightarrow g \) induce full and faithful functors \( \mathcal{S}(\varphi) : \mathcal{S}(f) \to \mathcal{S}(g) \).
Full internal subcategories

Given any morphism $f : B \to A$ in a locally cartesian closed category $\mathcal{E}$, we can form the **full internal subcategory** $S(f) \in \text{Cat}(\mathcal{E})$.

- Object of objects $= A$;
- Object of morphisms $= \sum_{a,a' \in A} B_{a,a'}^B = (\pi_2^* f) \pi_1^* f$ over $A \times A$.

Cartesian morphisms of polynomials $\varphi : f \Rightarrow g$ induce full and faithful functors $S(\varphi) : S(f) \to S(g)$.

**Idea:** Given cartesian morphisms $\varphi, \psi : f \Rightarrow g$, take internal natural transformations $S(\varphi) \Rightarrow S(\psi)$ to be our 3-cells.
Full internal subcategories

**Theorem**

*With respect to this notion of 3-cell:*
Full internal subcategories

Theorem

*With respect to this notion of 3-cell:*

- *p admits 1, Σ ⇐⇒ p is a pseudomonad*
Full internal subcategories

Theorem

With respect to this notion of 3-cell:

- $p$ admits $1, \Sigma \iff p$ is a pseudomonad
- $p$ also admits $\Pi \iff p$ is a $p$-pseudoalgebra

Aside:
For a natural model $p$:

- $U \to U$, let $U = S(p) \in \text{Cat}(\hat{C})$.

Object of objects = $U$.
Object of morphisms = $\sum_{A, B : U}[B][A]$.

Considered as an indexed category $C^{op} \to \text{Cat}$, $U$ is equivalent to the 'context-indexed category of types' of Clairambault & Dybjer (2011).

Bonus:
If $(C, p)$ admits $1, \Sigma, \Pi$, then $U$ is cartesian closed.
Full internal subcategories

**Theorem**

*With respect to this notion of 3-cell:*

- $p$ admits $1, \Sigma \iff p$ is a pseudomonad
- $p$ also admits $\Pi \iff p$ is a $p$-pseudoalgebra

**Aside:** For a natural model $p : \mathcal{U} \to \mathcal{U}$, let $\mathcal{U} = S(p) \in \text{Cat}(\hat{\mathcal{C}})$. 
Full internal subcategories

**Theorem**

*With respect to this notion of 3-cell:*

- $p$ admits $1, \Sigma \iff p$ is a pseudomonad
- $p$ also admits $\Pi \iff p$ is a $p$-pseudoalgebra

**Aside:** For a natural model $p : \mathcal{U} \to \mathcal{U}$, let $\mathcal{U} = S(p) \in \textbf{Cat}(\hat{\mathcal{C}})$.

- Object of objects $= \mathcal{U}$. 
Full internal subcategories

Theorem

*With respect to this notion of 3-cell:*

- $p$ admits $1, \Sigma \iff p$ is a *pseudomonad*
- $p$ also admits $\Pi \iff p$ is a *$p$-pseudoalgebra*

**Aside:** For a natural model $p : \mathcal{U} \to \mathcal{U}$, let $\mathbb{U} = S(p) \in \text{Cat}(\hat{\mathcal{C}})$.

- Object of objects $= \mathcal{U}$.
- Object of morphisms $= \sum_{A,B: \mathcal{U}} [B]^A$. 
Full internal subcategories

Theorem

With respect to this notion of 3-cell:

- $p$ admits $1, \Sigma \iff p$ is a pseudomonad
- $p$ also admits $\Pi \iff p$ is a $p$-pseudoalgebra

Aside: For a natural model $p : \mathcal{U} \to \mathcal{U}$, let $\mathcal{U} = S(p) \in \text{Cat}(\hat{C})$.

- Object of objects = $\mathcal{U}$.
- Object of morphisms = $\sum_{A,B:U}[B][A]$.

Considered as an indexed category $\mathcal{C}^{\text{op}} \to \text{Cat}$, $\mathcal{U}$ is equivalent to the ‘context-indexed category of types’ of Clairambault & Dybjer (2011).
Full internal subcategories

**Theorem**

*With respect to this notion of 3-cell:*

- $p$ admits $1, \Sigma \iff p$ is a **pseudomonad**
- $p$ also admits $\Pi \iff p$ is a $p$-**pseudoalgebra**

**Aside:** For a natural model $p : \mathcal{U} \to \mathcal{U}$, let $\mathbb{U} = S(p) \in \mathbf{Cat}(\hat{\mathcal{C}})$.

- Object of objects = $\mathcal{U}$.
- Object of morphisms = $\sum_{A,B : \mathcal{U}} [B]^{[A]}$.

Considered as an indexed category $\mathcal{C}^{\mathsf{op}} \to \mathbf{Cat}$, $\mathbb{U}$ is equivalent to the ‘context-indexed category of types’ of Clairambault & Dybjer (2011).

**Bonus:** If $(\mathcal{C}, p)$ admits $1, \Sigma, \Pi$, then $\mathbb{U}$ is cartesian closed.
1. Natural models

2. Connection with polynomial functors

3. Natural model semantics

4. Concluding remarks
Initiality of the syntax

Idea (Initiality ‘conjecture’)

*The syntax of a dependent type theory $\mathbb{T}$ should itself have the structure of a natural model, which is initial amongst all natural models interpreting $\mathbb{T}$.*
Initiality of the syntax

Idea (Initiality ‘conjecture’)

*The syntax of a dependent type theory $\mathcal{T}$ should itself have the structure of a natural model, which is initial amongst all natural models interpreting $\mathcal{T}$.*

Goals:
Initiality of the syntax

Idea (Initiality ‘conjecture’)

The syntax of a dependent type theory $T$ should itself have the structure of a natural model, which is initial amongst all natural models interpreting $T$.

Goals:

- Build the syntactic natural models for some basic type theories and prove that they satisfy an appropriate universal property;
Initiability of the syntax

Idea (Initiability ‘conjecture’)

The syntax of a dependent type theory $\mathcal{T}$ should itself have the structure of a natural model, which is initial amongst all natural models interpreting $\mathcal{T}$.

Goals:

- Build the syntactic natural models for some basic type theories and prove that they satisfy an appropriate universal property;
- Expand to more complicated type theories by (algebraically) freely adding type theoretic structure.
Example #1: set of basic types

We’ll construct the free natural model on the theory with an $I$-indexed family of basic types.
Example #1: set of basic types

We’ll construct the free natural model on the theory with an \( I \)-indexed family of basic types.

**Definition**

Define \( (\mathbb{C}_I, p_I : \mathcal{U}_I \to \mathcal{U}_I) \) as follows:
Example #1: set of basic types

We’ll construct the free natural model on the theory with an $I$-indexed family of basic types.

Definition

Define $(C, p : \mathcal{U}_I \to \mathcal{U}_I)$ as follows:

- Category of contexts: $C_I = (\text{Fin}/I)^\text{op}$
Example #1: set of basic types

We’ll construct the free natural model on the theory with an $I$-indexed family of basic types.

**Definition**
Define $(C_I, p_i : U_I \to U_I)$ as follows:

- Category of contexts: $C_I = (\text{Fin}/I)^{\text{op}}$
- Presheaf of types: $U_I = \text{cod} : \text{Fin}/I \to \text{Set}$  \( (A \xrightarrow{u} I) \mapsto I \)
Example #1: set of basic types

We’ll construct the free natural model on the theory with an \( I \)-indexed family of basic types.

**Definition**

Define \((\mathcal{C}_I, p_I : \mathcal{U}_I \rightarrow \mathcal{U}_I)\) as follows:

- **Category of contexts:** \( \mathcal{C}_I = (\text{Fin}/I)^{\text{op}} \)
- **Presheaf of types:** \( \mathcal{U}_I = \text{cod} : \text{Fin}/I \rightarrow \text{Set} \)
- **Presheaf of terms:** \( \mathcal{U}_I = \text{dom} : \text{Fin}/I \rightarrow \text{Set} \)

Typing map:

\[(p_I)_A : \mathcal{U}_I \mapsto A \]

Representability data: given \( A \) and \( i \in \mathcal{U}_I \), let \( (\mathcal{U}_I) 
\quad p_i : A \hookrightarrow A^{i+1} \]
Example #1: set of basic types

We’ll construct the free natural model on the theory with an \( I \)-indexed family of basic types.

**Definition**

Define \((C_I, p_I : U_I \to U_I)\) as follows:

- Category of contexts: \( C_I = (\text{Fin}/I)^{\text{op}} \)
- Presheaf of types: \( U_I = \text{cod} : \text{Fin}/I \to \text{Set} \quad (A \xrightarrow{u} I) \mapsto I \)
- Presheaf of terms: \( U_I = \text{dom} : \text{Fin}/I \to \text{Set} \quad (A \xrightarrow{u} I) \mapsto A \)
- Typing map: \((p_I)_{A \xrightarrow{u} I} = u : A \to I\)
Example #1: set of basic types

We’ll construct the free natural model on the theory with an $I$-indexed family of basic types.

Definition

Define $(\mathbb{C}_I, p_I : \mathcal{U}_I \rightarrow \mathcal{U}_I)$ as follows:

- Category of contexts: $\mathbb{C}_I = (\text{Fin}/I)^{\text{op}}$
- Presheaf of types: $\mathcal{U}_I = \text{cod} : \text{Fin}/I \rightarrow \text{Set} \quad (A \xrightarrow{u} I) \mapsto I$
- Presheaf of terms: $\mathcal{U}_I = \text{dom} : \text{Fin}/I \rightarrow \text{Set} \quad (A \xrightarrow{u} I) \mapsto A$
- Typing map: $(p_I)_{A \xrightarrow{u} I} = u : A \rightarrow I$
- Representability data: given $A \xrightarrow{u} I$ and $i \in \mathcal{U}_I(u) = I$, let

$$ (A \xrightarrow{u} I) \cdot i = (A + 1 \xrightarrow{[u,i]} I) $$

Clive Newstead (cnewstead@cmu.edu)
Algebraic models of dependent type theory

These slides: https://goo.gl/Ttacdq
**Example #1: set of basic types**

We’ll construct the free natural model on the theory with an \( I \)-indexed family of basic types.

**Definition**

Define \((\mathbb{C}_I, \rho_I : \dot{\mathbb{U}}_I \to \mathbb{U}_I)\) as follows:

- **Category of contexts:** \( \mathbb{C}_I = (\text{Fin}/I)^{\text{op}} \)
- **Presheaf of types:** \( \mathbb{U}_I = \text{cod} : \text{Fin}/I \to \text{Set} \quad (A \xrightarrow{u} I) \mapsto I \)
- **Presheaf of terms:** \( \dot{\mathbb{U}}_I = \text{dom} : \text{Fin}/I \to \text{Set} \quad (A \xrightarrow{u} I) \mapsto A \)
- **Typing map:** \( (\rho_I)_{A \xrightarrow{u} I} = u : A \to I \)
- **Representability data:** given \( A \xrightarrow{u} I \) and \( i \in \mathbb{U}_I(u) = I \), let

\[
(A \xrightarrow{u} I) \cdot i = (A + 1 \xrightarrow{[u,i]} I) \quad \rho_i : A \hookrightarrow A + 1
\]
Example #1: set of basic types

We’ll construct the free natural model on the theory with an \( I \)-indexed family of basic types.

**Definition**

Define \( (\mathbb{C}_I, p_I : \mathcal{U}_I \to \mathcal{U}_I) \) as follows:

- **Category of contexts:** \( \mathbb{C}_I = (\text{Fin}/I)^{\text{op}} \)
- **Presheaf of types:** \( \mathcal{U}_I = \text{cod} : \text{Fin}/I \to \text{Set} \) \( (A \xrightarrow{u} I) \mapsto I \)
- **Presheaf of terms:** \( \mathcal{U}_I = \text{dom} : \text{Fin}/I \to \text{Set} \) \( (A \xrightarrow{u} I) \mapsto A \)
- **Typing map:** \( (p_I)_{A \xrightarrow{u} I} = u : A \to I \)
- **Representability data:** given \( A \xrightarrow{u} I \) and \( i \in \mathcal{U}_I(u) = I \), let

\[
(A \xrightarrow{u} I) \cdot i = (A + 1 \xrightarrow{[u,i]} I) \quad p_i : A \hookrightarrow A + 1 \quad q_i = \ast
\]
Example #1: set of basic types

**Theorem**

$(C_I, p_I)$ is a natural model
Example #1: set of basic types

**Theorem**

$(C_I, p_I)$ is a natural model, and for all natural models $(C, p: \mathcal{U} \to \mathcal{U})$ and all $I$-indexed families $\{O_i\}_{i \in I} \subseteq \mathcal{U}(\diamond)$,
Example #1: set of basic types

Theorem

\((C_I, p_I)\) is a natural model, and for all natural models \((C, p : \mathcal{U} \to \mathcal{U})\) and all \(I\)-indexed families \(\{O_i\}_{i \in I} \subseteq \mathcal{U}(\diamond)\), there is a unique \(F : (C_I, p_I) \to (C, p)\) with \(F(i) = O_i\) for all \(i \in I\).
Example #2: freely admitting $\Sigma$-types

**Goal:** Given a natural model $(\mathcal{C}, p)$, construct the ‘smallest’ natural model $(\mathcal{C}_\Sigma, p_\Sigma)$ which extends $(\mathcal{C}, p)$ and admits $\Sigma$-types.
Example #2: freely admitting $\Sigma$-types

We can represent (iterated) $\Sigma$-types by binary trees.
Example #2: freely admitting $\Sigma$-types

We can represent (iterated) $\Sigma$-types by binary trees.
Example #2: freely admitting $\Sigma$-types

We can represent (iterated) $\Sigma$-types by binary trees.

$$
\sum_{\langle\langle x,y\rangle,z\rangle} C(x,y) \left( \sum_{w:D(x,y,z)} E(x,y,z,w) \right)
$$
Example #2: freely admitting $\Sigma$-types

We can represent (iterated) $\Sigma$-types by binary trees.

$$\sum_{\langle \langle x, y \rangle \rangle, z} C(x, y) \left( \sum_{w : D(x, y, z)} E(x, y, z, w) \right)$$
Example #2: freely admitting $\Sigma$-types

We can represent (iterated) $\Sigma$-types by binary trees.

$$\sum_{\langle\langle x,y\rangle,z\rangle} \left( \sum_{x:A} B(x) C(x,y) \sum_{w:D(x,y,z)} E(x,y,z,w) \right)$$
Example #2: freely admitting $\Sigma$-types

We can represent (iterated) $\Sigma$-types by binary trees.

$$\sum_{\langle\langle x, y \rangle, z \rangle} C(x, y) \left( \sum_{w : D(x, y, z)} E(x, y, z, w) \right)$$
Example #2: freely admitting $\Sigma$-types

We can represent (iterated) $\Sigma$-types by binary trees.

$$\sum_{\langle\langle x, y\rangle, z\rangle} \left( \sum_{\langle x, y\rangle} C(x, y) \sum_{w : D(x, y, z)} E(x, y, z, w) \right)$$
Example #2: freely admitting $\Sigma$-types

**Definition**
Define $(C_\Sigma, p_\Sigma : U_\Sigma \to U_\Sigma)$ as follows:
Example #2: freely admitting $\Sigma$-types

Definition

Define $(C_{\Sigma}, \rho_{\Sigma} : \mathcal{U}_{\Sigma} \to \mathcal{U}_{\Sigma})$ as follows:

- $C_{\Sigma}$: Objects (contexts) are the objects of $\mathcal{C}$ ‘formally extended’ by trees of dependent types;

- $\rho_{\Sigma}$: A morphism which sends types and terms to trivial trees (one vertex, no edges).
Example #2: freely admitting $\Sigma$-types

**Definition**

Define $(\mathcal{C}_\Sigma, \rho_\Sigma : \mathcal{U}_\Sigma \to \mathcal{U}_\Sigma)$ as follows:

- $\mathcal{C}_\Sigma$: Objects (contexts) are the objects of $\mathcal{C}$ ‘formally extended’ by trees of dependent types;
- $\mathcal{U}_\Sigma$ is the presheaf of *type trees*;
Example #2: freely admitting $\Sigma$-types

Definition

Define $(C_\Sigma, p_\Sigma : \mathcal{U}_\Sigma \to \mathcal{U}_\Sigma)$ as follows:

- $C_\Sigma$: Objects (contexts) are the objects of $C$ ‘formally extended’ by trees of dependent types;
- $\mathcal{U}_\Sigma$ is the presheaf of *type trees*;
- $\mathcal{U}_\Sigma$ is the presheaf of *term trees*;
Example #2: freely admitting $\Sigma$-types

**Definition**

Define $(C_\Sigma, p_\Sigma : \mathcal{U}_\Sigma \to \mathcal{U}_\Sigma)$ as follows:

- $C_\Sigma$: Objects (contexts) are the objects of $C$ ‘formally extended’ by trees of dependent types;
- $\mathcal{U}_\Sigma$ is the presheaf of *type trees*;
- $\mathcal{U}_\Sigma$ is the presheaf of *term trees*;
- $p_\Sigma : (\text{tree of terms}) \mapsto (\text{tree of their types})$. 

Clive Newstead (cnewstead@cmu.edu) Carnegie Mellon University

Algebraic models of dependent type theory

These slides: https://goo.gl/Ttaoq
Example #2: freely admitting $\Sigma$-types

Definition

Define $(\mathbb{C}_\Sigma, p_\Sigma : \mathcal{U}_\Sigma \to \mathcal{U}_\Sigma)$ as follows:

- $\mathbb{C}_\Sigma$: Objects (contexts) are the objects of $\mathbb{C}$ ‘formally extended’ by trees of dependent types;
- $\mathcal{U}_\Sigma$ is the presheaf of type trees;
- $\mathcal{U}_\Sigma$ is the presheaf of term trees;
- $p_\Sigma : (\text{tree of terms}) \mapsto (\text{tree of their types})$.

There is a morphism $I : (\mathbb{C}, p) \to (\mathbb{C}_\Sigma, p_\Sigma)$, which sends types and terms to trivial trees (one vertex, no edges).
Example #2: freely admitting $\Sigma$-types

Theorem

$(C_\Sigma, p_\Sigma)$ is a natural model admitting $\Sigma$-types
Example #2: freely admitting $\Sigma$-types

**Theorem**

$(\mathbb{C}_\Sigma, p_\Sigma)$ is a natural model admitting $\Sigma$-types, and for all $F : (\mathbb{C}, p) \to (\mathbb{D}, q)$ with $(\mathbb{D}, q)$ admitting $\Sigma$-types,

$$
\begin{array}{c}
(\mathbb{C}, p) \xrightarrow{F} (\mathbb{D}, q) \\
\downarrow l \\
(\mathbb{C}_\Sigma, p_\Sigma)
\end{array}
$$
Example #2: freely admitting $\Sigma$-types

**Theorem**

$(\mathbb{C}_{\Sigma}, p_{\Sigma})$ is a natural model admitting $\Sigma$-types, and for all $F : (\mathbb{C}, p) \to (\mathbb{D}, q)$ with $(\mathbb{D}, q)$ admitting $\Sigma$-types, there is a unique $\Sigma$-type-preserving $F^\# : (\mathbb{C}_{\Sigma}, p_{\Sigma}) \to (\mathbb{D}, q)$ extending $F$ along $I$.

\[
\begin{array}{c}
(\mathbb{C}, p) \\
\downarrow I \\
(\mathbb{C}_{\Sigma}, p_{\Sigma})
\end{array} \xrightarrow{F} \begin{array}{c}
(\mathbb{D}, q) \\
\downarrow F^\
\end{array}
\]
Example #2: freely admitting $\Sigma$-types

We can characterise freely admitting $\Sigma$-types functorially.
Example #2: freely admitting $\Sigma$-types

We can characterise freely admitting $\Sigma$-types functorially.

**Inspiration:** Given a set $S$, the set of finite rooted binary trees with leaves labelled by elements of $S$ is an initial algebra for the polynomial functor $X \mapsto S + X \times X$. 
Example #2: freely admitting $\Sigma$-types

We can characterise freely admitting $\Sigma$-types functorially.

**Inspiration:** Given a set $S$, the set of finite rooted binary trees with leaves labelled by elements of $S$ is an initial algebra for the polynomial functor $X \mapsto S + X \times X$.

**Theorem**

$p_\Sigma$ is an initial algebra for the endofunctor $f \mapsto p + f \cdot f$. 
Example #3: freely adjoining a term

Let $(C, p)$ be a natural model and $O \in \mathcal{U}(\diamond)$.
Example #3: freely adjoining a term

Let $(\mathbb{C}, \rho)$ be a natural model and $O \in \mathcal{U}(\diamond)$.

**Idea:** Freely adjoin new term $x : O$ by slicing by $O (= \diamond \cdot O)$. 
Example #3: freely adjoining a term

Let \((\mathcal{C}, p)\) be a natural model and \(O \in \mathcal{U}(\diamond)\).

**Idea:** Freely adjoin new term \(x : O\) by slicing by \(O (\equiv \diamond \cdot O)\).
Example #3: freely adjoining a term

Let \((C, p)\) be a natural model and \(O \in \mathcal{U}(\diamond)\).

**Idea:** Freely adjoin new term \(x : O\) by slicing by \(O \bowtie \diamond \cdot O\).

\[
\begin{array}{c}
\mathcal{C} \\
\downarrow \Delta_O \\
\mathcal{C}_x:O \xrightarrow{\simeq} \mathcal{C}/O
\end{array}
\]
Example #3: freely adjoining a term

Let \((\mathbb{C}, p)\) be a natural model and \(O \in U(\diamond)\).

**Idea:** Freely adjoin new term \(x : O\) by slicing by \(O (\cong \diamond \cdot O)\).
Example #3: freely adjoining a term

Let \((\mathcal{C}, p)\) be a natural model and \(O \in \mathcal{U}(\diamond)\).

**Idea:** Freely adjoin new term \(x : O\) by slicing by \(O (= \diamond \cdot O)\).

\[
\begin{array}{ccc}
\mathbb{C} & \xymatrix{\mathbb{C}/O} \\
\ar @{} |lll| [r] \Delta_O & \ar @{} |l| [l] \leftarrow \mathcal{C}_{x:O} & \ar @{} |r| [r] \\
\end{array}
\]
Example #3: freely adjoining a term

Let \((C, p)\) be a natural model and \(O \in \mathcal{U}(\Diamond)\).

Idea: Freely adjoin new term \(x : O\) by slicing by \(O (= \Diamond \cdot O)\).
Example #3: freely adjoining a term

Let \((\mathbb{C}, p)\) be a natural model and \(O \in \mathcal{U}(\diamond)\).

**Idea:** Freely adjoin new term \(x : O\) by slicing by \(O \ (= \diamond \cdot O)\).
Example #3: freely adjoining a term

Let $(\mathbb{C}, p)$ be a natural model and $O \in \mathcal{U}(\otimes)$.

**Idea:** Freely adjoin new term $x : O$ by slicing by $O (= \otimes \cdot O)$.

\[
\begin{array}{c}
\begin{array}{c}
\Delta_{\otimes} \\
\downarrow \\
\mathbb{C} \\
\uparrow I \\
\mathbb{C}_{x : O} \\
\sim \\
\mathbb{C}/O
\end{array}
\end{array}
\]

**Note:** The objects of $\mathbb{C}_{x : O}$ look like

\[
\Gamma \cdot O \cdot A_1 \ldots \cdot A_n \xrightarrow{\text{projections}} \Gamma \cdot O \xrightarrow{! \cdot O} O
\]

\[
\begin{array}{c}
\begin{array}{c}
\downarrow p_0 \\
\Gamma \\
\downarrow I \\
! \\
\downarrow p_0 \\
\mathbb{C}_{x : O} \\
\sim \\
\mathbb{C}/O
\end{array}
\end{array}
\]
Example #3: freely adjoining a term

Theorem

$\left(\mathbb{C}_{x:O}, p_{x:O}\right)$ is a natural model
Example #3: freely adjoining a term

Theorem

\((\mathbb{C}, p)\) is a natural model, and for all \(F : (\mathbb{C}, p) \to (\mathbb{D}, q)\) and all \(a : FO\) in \(\mathbb{D}\),

\[
\begin{array}{ccc}
(\mathbb{C}, p) & \xrightarrow{F} & (\mathbb{D}, q) \\
\downarrow & & \downarrow \\
(\mathbb{C}, p) & \xrightarrow{I} & (\mathbb{C}, p)
\end{array}
\]
Example #3: freely adjoining a term

Theorem

\((\mathbb{C}_x:O, p_x:O)\) is a natural model, and for all \(F : (\mathbb{C}, p) \rightarrow (\mathbb{D}, q)\) and all \(a : FO\) in \(\mathbb{D}\), there is a unique \(F^\# : (\mathbb{C}_x:O, p_x:O) \rightarrow (\mathbb{D}, q)\) extending \(F\), such that \(F^\#(x) = a\).

\[
\begin{array}{ccc}
(\mathbb{C}, p) & \xrightarrow{F} & (\mathbb{D}, q) \\
\downarrow l & & \downarrow \exists a : FO \\
(\mathbb{C}_x:O, p_x:O) & \xrightarrow{F^\#} & \exists x : O
\end{array}
\]
Example #3: freely adjoining a term

**Theorem**

\((\mathbb{C}_{x:o}, p_{x:o})\) is a natural model, and for all \(F : (\mathbb{C}, p) \to (\mathbb{D}, q)\) and all \(a : FO\) in \(\mathbb{D}\), there is a unique \(F^\# : (\mathbb{C}_{x:o}, p_{x:o}) \to (\mathbb{D}, q)\) extending \(F\), such that \(F^\#(x) = a\).

\[
\begin{align*}
(\mathbb{C}, p) & \xrightarrow{F} (\mathbb{D}, q) & \exists a : FO \\
\downarrow I & & \downarrow \exists x : O \\
(\mathbb{C}_{x:o}, p_{x:o}) & \xrightarrow{F^\#} (\mathbb{D}, q) & \exists x : O
\end{align*}
\]

**Note:** \(x\) is given by the diagonal map \(O \to O \times O (= \diamond \cdot O \cdot O[p_O])\) in \(\mathbb{C}/O\).
1. Natural models

2. Connection with polynomial functors

3. Natural model semantics

4. Concluding remarks
Future directions

Some areas of interest for the future:
Future directions

Some areas of interest for the future:

- Develop a **formal theory of natural models** in an arbitrary (suitably structured) category $\mathcal{E}$, not just a presheaf topos;
Future directions

Some areas of interest for the future:

- Develop a **formal theory of natural models** in an arbitrary (suitably structured) category $\mathcal{E}$, not just a presheaf topos;
- Further investigate properties of the full internal subcategory associated with a natural model;
Future directions

Some areas of interest for the future:

- Develop a **formal theory of natural models** in an arbitrary (suitably structured) category $\mathcal{E}$, not just a presheaf topos;
- Further investigate properties of the full internal subcategory associated with a natural model;
- Translate connections between polynomial monads and **operads** to this setting;
Future directions

Some areas of interest for the future:

- Develop a **formal theory of natural models** in an arbitrary (suitably structured) category $\mathcal{E}$, not just a presheaf topos;
- Further investigate properties of the full internal subcategory associated with a natural model;
- Translate connections between polynomial monads and **operads** to this setting;
- Formalise natural models in HoTT.
Thanks for listening!
References

Natural models

- Fiore (2012) *Discrete generalised polynomial functors*, slides from talk at ICALP 2012

Polynomials


Related work with CwFs


Clive Newstead (cnewstead@cmu.edu)
Algebraic models of dependent type theory

Carnegie Mellon University

These slides: https://goo.gl/Ttacdq