Kripke-Joyal Semantics for Dependent Type Theory\textsuperscript{1}

\textit{Pittsburgh’s HoTT Seminar}

Sina Hazratpour

December 2020

\textsuperscript{1} joint work with Steve Awodey (CMU) and Nicola Gambino (Leeds)
Outline

1. Review of classical Kripke-Joyal semantics for toposes

2. Kripke–Joyal semantics for dependent type theories
   - Definition
   - Forcing dependent sum types
   - Forcing extensional equality types
   - Forcing dependent product types
   - Forcing a universe
   - Forcing the impredicative universe of propositions
   - Forcing the universe of cofibrant propositions
   - Forcing the type of partial elements
   - Applications
Review of classical Kripke-Joyal semantics for toposes
The Kripke–Joyal semantics of a topos $\mathcal{E}$ gives an interpretation to formulas written in its higher order intuitionistic internal language $\text{HoL}(\Sigma_{\mathcal{E}})$. 
The Kripke–Joyal semantics of a topos $\mathcal{E}$ gives an interpretation to formulas written in its higher order intuitionistic internal language $HoL(\Sigma_{\mathcal{E}})$.

The Kripke–Joyal semantics is in fact a higher order generalization of the well-known Kripke semantic for intuitionistic propositional logic.
Definition

Let $\mathcal{E}$ be an elementary topos. Given a formula $\varphi(x)$ with a free variable $x$ of sort $A$ in $HoL(\Sigma_{\mathcal{E}})$, and a generalized element $\alpha: U \to A$ in $\mathcal{E}$, we define

$$U \models \varphi(\alpha) \iff \alpha \text{ factors through the subobject } [\varphi] \to A.$$
Definition

Let $\mathcal{E}$ be an elementary topos. Given a formula $\varphi(x)$ with a free variable $x$ of sort $A$ in $\text{HoL}(\Sigma_{\mathcal{E}})$, and a generalized element $\alpha: U \to A$ in $\mathcal{E}$, we define

$$U \models \varphi(\alpha) \iff \alpha \text{ factors through the subobject } [\varphi] \hookrightarrow A.$$ 

- Call $(U, \alpha)$ the **stage** of forcing.
- Write $\mathcal{E} \models \varphi$ if at every stage $(U, \alpha)$, we have $U \models \varphi(\alpha)$. 

![Diagram](diagram.png)
One can then show:

- \( U \models \top(\alpha) \).
- \( U \models \bot(\alpha) \) iff \( U \) in the initial object of \( \mathcal{E} \).
- \( U \models (x = x')(\langle \alpha, \alpha' \rangle) \) iff \( \alpha : U \to X \) and \( \alpha' : U \to X \) are the same maps in \( \mathcal{E} \).
- \( U \models (\varphi \land \psi)(\alpha) \) iff \( U \models \varphi(\alpha) \) and \( U \models \psi(\alpha) \).
- \( U \models (\varphi \lor \psi)(\alpha) \) iff there are jointly epimorphic arrows \( p : V \to U \) and \( q : W \to U \) such that \( V \models \varphi(\alpha \circ p) \) and \( W \models \varphi(\alpha \circ q) \).
- \( U \models (\varphi \Rightarrow \psi)(\alpha) \) iff for any arrow \( f : V \to U \) such that \( V \models \varphi(\alpha \circ f) \) then \( V \models \psi(\alpha \circ f) \).
- \( c \models \neg \varphi(\alpha) \) iff for all maps \( f : V \to U \) in \( \mathcal{E} \), \( V \not\models \varphi(\alpha.f) \).
Henceforth, $\mathcal{E} = \mathcal{P}Shv(C) = \mathcal{S}et^{C^{\text{op}}}$. 
Henceforth, $\mathcal{E} = \mathcal{P}\text{Shv}(C) = \text{Set}^{C^{\text{op}}}$. 

In the presheaf toposes, every presheaf is a colimit of representables. 

So it is enough to consider forcing statements $U \models \varphi(\alpha)$ for representables $U = yc$. 

Classical Kripke-Joyal semantics for presheaf toposes (review) (III)
Henceforth, $E = PShv(C) = \text{Set}^{\text{C}^{\text{op}}}$. 

In the presheaf toposes, every presheaf is a colimit of representables. 

So it is enough to consider forcing statements $U \models \varphi(\alpha)$ for representables $U = yc$. 

$c \models (\varphi \lor \psi)(\alpha) \iff c \models \varphi(\alpha) \text{ or } c \models \psi(\alpha)$

Recall $yc$ is projective & indecomposable.
Limitations of classical Kripke–Joyal semantics

- Bounded quantification. We shall overcome this by generalizing Kripke–Joyal semantics to dependent type theory with universes.
- Equality of terms is extensional and not “up to homotopy”. We will also generalize to homotopy type theory.
Kripke–Joyal semantics for dependent type theory

- Provability versus proof relevance: the classical Kripke–Joyal semantics is only concerned with the provability of a proposition; it is proof irrelevant. Kripke–Joyal semantics for dependent type theory includes the terms (proofs) in the forcing statements.
Provability versus proof relevance: the classical Kripke–Joyal semantics is only concerned with the provability of a proposition; it is proof irrelevant. Kripke–Joyal semantics for dependent type theory includes the terms (proofs) in the forcing statements.

We want a sound, formal and (quasi-) mechanical process to relate internal developments (Cohen et al., 2018), (Orton and Pitts, 2018), etc. with the diagrammatic developments (Gambino and Sattler, 2017), (Sattler, 2017), (Awodey, 2018), etc. found in the models of HoTT literature.
Kripke–Joyal semantics for dependent type theories
Definition (Dependent Kripke–Joyal semantics– forcing terms)

For a context $\Gamma$, 

$\Gamma$
Definition (Dependent Kripke–Joyal semantics– forcing terms)

For a context $\Gamma$, a type $\Gamma \vdash A \text{ Type}$,
Definition (Dependent Kripke–Joyal semantics– forcing terms)
For a context $\Gamma$, a type $\Gamma \vdash A$ Type, an object $c$ of $C$, 
$\Gamma.A \quad \downarrow \quad p_A \quad \Gamma \quad \downarrow \quad \Gamma$
Definition (Dependent Kripke–Joyal semantics– forcing terms)

For a context $\Gamma$, a type $\Gamma \vdash A \text{ Type}$, an object $c$ of $\mathcal{C}$, and a morphism $\gamma : yc \to \Gamma$, $\Gamma.A \xrightarrow{p_A} \Gamma \xrightarrow{\gamma} yc$.
Definition (Dependent Kripke–Joyal semantics– forcing terms)
For a context $\Gamma$, a type $\Gamma \vdash A \text{ Type}$, an object $c$ of $\mathcal{C}$, and a morphism $\gamma : yc \to \Gamma$, we say $c$ forces $a : A$ at stage $\gamma$,

$$c \vdash [a : A](\gamma) \iff \text{there is a lift } \langle \gamma, a \rangle \text{ of } \gamma \text{ against } p_A : \Gamma.A \to \Gamma.$$
Definition (Dependent Kripke–Joyal semantics– forcing terms)
For a context $\Gamma$, a type $\Gamma \vdash A$ Type, an object $c$ of $C$, and a morphism $\gamma : yc \to \Gamma$, we say $c$ forces $a : A$ at stage $\gamma$,

$$c \vdash [a : A](\gamma) \iff \text{there is a lift } \langle \gamma, a \rangle \text{ of } \gamma \text{ against } p_A : \Gamma.A \to \Gamma.$$ 

Proposition
$\Gamma \vdash a : A \iff$ There is a family $(a_\gamma \mid c : \text{an object of } C, \gamma : yc \to \Gamma)$ satisfying

$$c \vdash [a_\gamma : A](\gamma)$$

and for every morphism $f : d \to c$ of $C$,

$$a_\gamma \cdot f = a_\gamma \cdot f$$

Proof.
By Yoneda Lemma.
Forcing dependent sum types

Proposition

*Given a context $\Gamma$, \*
Forcing dependent sum types

Proposition
Given a context $\Gamma$, a type $\Gamma \vdash A \ Type$, $\Gamma.A$ $\vdash \Gamma$. 

$\Gamma.A$ 
$\Gamma$ $p_A$
Forcing dependent sum types

Proposition
Given a context \( \Gamma \), a type \( \Gamma \vdash A \ Type \), a type \( \Gamma, x : A \vdash B \ Type \),

\[
\begin{align*}
& \Gamma.A.B \\
& \Gamma \vdash \sum_{a:A} B(a) \\
& \Gamma.A \\
& \Gamma
\end{align*}
\]
Forcing dependent sum types

Proposition

Given a context $\Gamma$, a type $\Gamma \vdash A$ Type, a type $\Gamma, x : A \vdash B$ Type, an object $c$ of $C$,
Forcing dependent sum types

Proposition

Given a context $\Gamma$, a type $\Gamma \vdash A$ Type, a type $\Gamma$, $x : A \vdash B$ Type, an object $c$ of $C$, and a morphism $\gamma : yc \to \Gamma$,
Forcing dependent sum types

Proposition

Given a context $\Gamma$, a type $\Gamma \vdash A \text{ Type}$, a type $\Gamma, x : A \vdash B \text{ Type}$, an object $c$ of $C$, and a morphism $\gamma : yc \to \Gamma$,

\[ c \vdash [d : \sum_{a : A} B(a)](\gamma) \]

iff

\[ d = (d_0, d_1) \]

\[ c \vdash [d_0 : A](\gamma) \]

\[ c \vdash [d_1 : B](\langle \gamma, d_0 \rangle) . \]
Forcing extensional equality types

Proposition

*Given a context* \( \Gamma \),

\[ \Gamma \]
Forcing extensional equality types

Proposition

*Given a context $\Gamma$, a type $\Gamma \vdash A \text{ Type}$,*
Forcing extensional equality types

**Proposition**

*Given a context \( \Gamma \), a type \( \Gamma \vdash A \) \( \text{Type} \), an object \( c \) of \( C \),*
Proposition

Given a context $\Gamma$, a type $\Gamma \vdash A \text{ Type}$, an object $c$ of $C$, a morphism $\gamma : yc \to \Gamma$,
Forcing extensional equality types

Proposition

Given a context $\Gamma$, a type $\Gamma \vdash A$ Type, an object $c$ of $C$, a morphism $\gamma : yc \to \Gamma$, $c \vdash [(a, a') : A \times A](\gamma)$
Forcing extensional equality types

Proposition

Given a context $\Gamma$, a type $\Gamma \vdash A$ Type, an object $c$ of $C$, a morphism $\gamma : yc \to \Gamma$, $c \vdash [(a, a') : A \times A](\gamma)$ we have

$$c \vdash [e : \text{Eq}_A](\langle \gamma, (a, a') \rangle) \iff a, a' \text{ are equal as morphisms in } \mathcal{C} \iff a, a' \text{ are equal elements of } A(c).$$

Type $\text{Eq}_A$ is interpreted by the diagonal morphism

$$\delta : A \to A \times_\Gamma A$$

over $\Gamma$. 
Forcing dependent product types

**Proposition**

*Given a context* $\Gamma$, 

\[ \langle \gamma, f, a \rangle (\langle \gamma, f, a \rangle, b f (a)) y (f) p A \langle \gamma, b \rangle \gamma \]
Forcing dependent product types

Proposition

Given a context $\Gamma$, a type $\Gamma \vdash A : Type$, 

\[
\Gamma.A \\
\text{PA} \\
\Gamma
\]
Forcing dependent product types

Proposition

Given a context $\Gamma$, a type $\Gamma \vdash A$ Type, a type $\Gamma, x : A \vdash B$ Type,
Forcing dependent product types

Proposition

*Given a context $\Gamma$, a type $\Gamma \vdash A \, Type$, a type $\Gamma, x : A \vdash B \, Type$, an object $c$ of $C$,.*
Forcing dependent product types

Proposition

Given a context $\Gamma$, a type $\Gamma \vdash A \text{ Type}$, a type $\Gamma, x : A \vdash B \text{ Type}$, an object $c$ of $C$, and a morphism $\gamma : yc \rightarrow \Gamma$, 

$$
\begin{align*}
&\Gamma.A.B \\
&\downarrow p_B \\
&\Gamma.A \\
&\downarrow p_A \\
&yc \\
&\downarrow \gamma \\
&\Gamma
\end{align*}
$$
Forcing dependent product types

Proposition

Given a context $\Gamma$, a type $\Gamma \vdash A \text{ Type}$, a type $\Gamma, x : A \vdash B \text{ Type}$, an object $c$ of $C$, and a morphism $\gamma : yc \to \Gamma$,

$$c \vdash [b : \prod x : A B](\gamma)$$

iff for every morphism $f : d \to c$ in $C$,

$$d \vdash [a : A](\gamma.f)$$

returns

$$d \vdash [b_f(a) : B](\langle \gamma.f, a \rangle)$$

such that for every morphism $g : d' \to d$, we have

$$b_f(a).g = b_{f \circ g}(a.g).$$
Definition (Dependent Kripke–Joyal semantics– forcing types)

For a type $\Gamma \vdash A \text{ Type}$, an object $c$ of $\mathcal{C}$, and a morphism $\gamma : yc \to \Gamma$, we say $c$ forces $A \text{ Type}$ at stage $\gamma$, and we write $c \models [A \text{ Type}](\gamma)$, whenever there is a presheaf $\tilde{A}_\gamma$ and a map $p_\gamma : \tilde{A}_\gamma \to yc$ such that for every morphism $f : d \to c$ in $\mathcal{C}$ there is a presheaf $\tilde{A}_{\gamma,f}$ and a choice of map $\tilde{A}_{\gamma,f} \to \tilde{A}_\gamma$, making a pullback square

$$
\begin{array}{c}
\tilde{A}_{\gamma,f} \\
\downarrow p_{\gamma,f}
\end{array} 
\begin{array}{c}
\rightarrow \\
\tilde{A}_\gamma \\
\downarrow p_\gamma
\end{array}

\begin{array}{c}
yd \\
yf
\end{array} 
\begin{array}{c}
\rightarrow \\
yc
\end{array}

(1)
Forcing a universe $\mathcal{V}$ of small types

- Let $\kappa$ be an inaccessible cardinal number.
- Call the sets of size strictly less than $\kappa$ small.
- Write $\text{Set}_\kappa$ for the category of small sets and $\hat{C}_\kappa \triangleq [C^{\text{op}}, \text{Set}_\kappa]$.
- Call a family $p : E \to \Gamma$ small whenever all the fibres $E(c) \to \Gamma(c)$ are small.
Recall that the \((\kappa-)\text{universe}\) \(p_V : \mathcal{V} \to \mathcal{V}\) in \(\mathcal{P}\text{Shv}(\mathcal{C})\) is defined as follows:

1. \(\mathcal{V}c \triangleq \text{Ob}[(\mathcal{C}/c)^{\text{op}}, \text{Set}_\kappa]\)

2. \(\mathcal{V}_c \triangleq \text{Ob}[(\mathcal{C}/c)^{\text{op}}, (\text{Set}_\kappa)_c]\)

3. There is a forgetful map \(p_V : \mathcal{V} \to \mathcal{V}\) which takes \((A, a)\) to \(A\).
Proposition

For an object $c$ of $\mathcal{C}$,

$$c \models [a : \mathcal{V}] \iff c \models [\text{El}(a) \text{Type}],$$

$$\text{El}(a.f) \equiv \text{El}(a).f \text{ for every } f : d \to c , \text{ and}$$

$$\text{El}(a) \to yc \text{ and } \text{El}(a.f) \to yd \text{ (for all } f : d \to c \text{) are small.}$$
Forcing a universe

Proposition

For an object $c$ of $\mathcal{C}$,

$$c \vdash [a : \mathcal{V}] \iff c \vdash [\text{El}(a)\text{Type}],$$

$$\text{El}(a.f) \equiv \text{El}(a).f \text{ for every } f : d \to c, \text{ and }$$

$$\text{El}(a) \to yc \text{ and } \text{El}(a.f) \to yd \text{ (for all } f : d \to c \text{) are small.}$$

Proposition

For an object $c$ of $\mathcal{C}$,

$$c \vdash [a_\bullet : \mathcal{V}_\bullet] \iff a_\bullet = (a, b) \text{ such that } c \vdash [a : \mathcal{V}],$$

$$\text{and } c \vdash [b : \text{El}(a)](\text{id}_c)$$
Forcing universe

**Proposition**

*For a small family* \( p_E : E \rightarrow y c \), *we have* \( c \Vdash [\text{code}(E) : \mathcal{V}] \) *for a canonical* \( \text{code}(E) \).
Remark

- The “classifying” operation

\[
\text{code: } \mathcal{E}_\kappa/\Gamma \to \mathcal{E}(\Gamma, \mathcal{V})
\]

has a left (quasi-)inverse, namely the evident “pullback of \( p_\mathcal{V}: \mathcal{V}_\bullet \to \mathcal{V} \)” operation

\[
\text{El: } \mathcal{E}(\Gamma, \mathcal{V}) \to \mathcal{E}_\kappa/\Gamma
\]

pseudo-naturally in \( \Gamma \).

- But there is no corresponding uniqueness of classifying maps, relating \( a: \Gamma \to \mathcal{V} \) and code \( \text{El}(a): \Gamma \to \mathcal{V} \).

- We do get the uniqueness of classifying maps by restricting to a smaller universe \( \Omega \) consisting of only “propositions”.

Forcing universe (IV)
Forcing the impredicative universe $\Omega$ of propositions

- As usual, a impredicative universe $\Omega$ of (small) propositions in $\mathcal{E}$ is defined object-wise by

$$\Omega(c) \triangleq \text{Ob } \mathcal{F}un((C/c)\text{op}, \mathcal{2}),$$

where $\mathcal{2}$ is the category with two objects, say $\bot$, $\top$, and one non-identity arrow $\bot \rightarrow \top$.

- $\Omega(c)$ is isomorphic to the set of sieves on object $c$, or equivalently, the set of subobjects of $y c$. 

Forcing the impredicative universe $\Omega$ of propositions

$$\Omega(c) \triangleq \text{Ob } \mathcal{F} \text{un}((\mathcal{C}/c)^{\text{op}}, \mathcal{2}),$$

**Proposition**

*Given an object $c$ of $\mathcal{C}$, the following statements are equivalent:*

1. $c \vDash [\varphi : \Omega].$
2. $c \vDash [[\varphi] \text{ Type}]$ such that the maps $p_c : [\varphi] \rightarrow yc$ and $p_f : [\varphi.f] \rightarrow yd$ (for all $f : d \rightarrow c$) are monomorphisms.
Forcing the impredicative universe $\Omega$ of propositions

Proposition

Suppose $c \models [\varphi : \Omega]$. The following statements are equivalent:
Forcing the impredicative universe $\Omega$ of propositions

**Proposition**

*Suppose $c \models [\varphi : \Omega]$. The following statements are equivalent:*

1. $c \models [x : [\varphi]]$.
2. $[\varphi] = yc$.
3. $c \models [e : Eq_{\Omega}(\varphi, \text{true})]$ for some $e$. 

![Diagram](image_url)
Propositions as types

There is a canonical map $\iota : \Omega \to \mathcal{V}$ which fits into a cartesian square

$$
\begin{array}{ccc}
1 & \xrightarrow{\tilde{*}} & \mathcal{V} \\
\downarrow \text{true} & & \downarrow p \mathcal{V} \\
\Omega & \xrightarrow{\iota} & \mathcal{V}
\end{array}
$$

where $\tilde{*} = (\iota \text{true}, *)$, and $*$ is the unique term of $\text{El}(\iota \text{true}) \cong \text{[true]} \cong 1$. 
Proposition

Suppose $\Gamma. A \vdash \varphi : \Omega$. We have

1. $\nu(\forall x : A. \varphi(x)) \equiv \Pi(a, \nu \varphi)$.

2. $\Sigma(a, \nu \varphi) \to \nu(\exists x : A. \varphi(x))$ is inhabited.
As in (Orton and Pitts, 2018), we consider a modality $\text{cof}: \Omega \to \Omega$ satisfying:

1. $\text{cof} \circ \text{true} = \text{true}$
2. $\text{cof} \circ \text{false} = \text{true}$
3. $\forall (\phi, \psi: \Omega). \text{cof} \phi \Rightarrow (\phi \Rightarrow \text{cof} \psi) \Rightarrow \text{cof} (\phi \land \psi)$.

The last axiom is called the principle of dominance.
As in (Orton and Pitts, 2018), we consider a modality $\text{cof} : \Omega \to \Omega$ satisfying:

(i) $\text{cof} \circ \text{true} = \text{true}$,
As in (Orton and Pitts, 2018), we consider a modality $\text{cof} : \Omega \to \Omega$ satisfying:

(i) $\text{cof} \circ \text{true} = \text{true}$,

(ii) $\text{cof} \circ \text{false} = \text{true}$,
As in (Orton and Pitts, 2018), we consider a modality $\text{cof} : \Omega \to \Omega$ satisfying:

1. $\text{cof} \circ \text{true} = \text{true}$,
2. $\text{cof} \circ \text{false} = \text{true}$,
3. $\forall (\varphi, \psi : \Omega). \ \text{cof} \varphi \Rightarrow (\varphi \Rightarrow \text{cof} \psi) \Rightarrow \text{cof}(\varphi \land \psi)$.
As in (Orton and Pitts, 2018), we consider a modality $\text{cof} : \Omega \to \Omega$ satisfying:

(i) $\text{cof} \circ \text{true} = \text{true}$,
(ii) $\text{cof} \circ \text{false} = \text{true}$,
(iii) $\forall (\phi, \psi : \Omega). \text{cof} \phi \Rightarrow (\phi \Rightarrow \text{cof} \psi) \Rightarrow \text{cof}(\phi \land \psi)$.

The last axiom is called the principle of dominance.
Forcing cofibrant propositions

- Obtain $m_{Cof}: Cof \hookrightarrow \Omega$ as the comprehension subtype; in the internal language

\[
\text{Cof} \overset{\Delta}{=} \{ \varphi \in \Omega \mid \text{cof} \varphi \}
\]
Obtain $m_{Cof} : \text{Cof} \to \Omega$ as the comprehension subtype; in the internal language

$$\text{Cof} \triangleq \{ \varphi \in \Omega \mid \text{cof} \, \varphi \}$$

cof (true) = true implies that true = $m_{Cof} \circ t$ for a monomorphism $t : 1 \to \text{Cof}$. 
Forcing cofibrant propositions

- Obtain $m_{\text{Cof}} : \text{Cof} \rightarrow \Omega$ as the comprehension subtype; in the internal language

\[ \text{Cof} \triangleq \{ \varphi \in \Omega \mid \text{cof} \varphi \} \]

- $\text{cof}(\text{true}) = \text{true}$ implies that $\text{true} = m_{\text{Cof}} \circ t$ for a monomorphism $t : 1 \rightarrow \text{Cof}$.

- Call $t$ the generic cofibrant proposition.
Cofibrations

A monomorphism $m: C \rightarrow Z$ is a **cofibration** if its classifying map $\chi_m: Z \rightarrow \Omega$ factors through $m_{\text{cof}}: \text{Cof} \rightarrow \Omega$.

\[
\begin{array}{ccc}
C & \to & 1 \\
\downarrow m & & \downarrow t \\
Z & \longrightarrow & \text{Cof} \end{array}
\]

$\chi_m$ factors through $m_{\text{cof}}$.
Cofibrations

- A monomorphism \( m: C \hookrightarrow Z \) is a **cofibration** if its classifying map \( \chi_m: Z \rightarrow \Omega \) factors through \( m_{\text{cof}}: \text{Cof} \hookrightarrow \Omega \).

\[
\begin{array}{ccc}
C & \rightarrow & 1 \\
m \downarrow & & \downarrow t \\
Z & \rightarrow & \text{Cof} \\
\end{array}
\]

\[
\begin{array}{ccc}
1 & \rightarrow & 1 \\
\rightarrow & & \rightarrow \\
\text{Cof} & \rightarrow & \Omega \\
m_{\text{cof}} \downarrow & & \downarrow \text{true} \\
\chi_m \downarrow & & \\
Z & \rightarrow & \Omega
\end{array}
\]

- Therefore, a monomorphism \( m: C \hookrightarrow Z \) is a cofibration iff it is a pullback of the generic cofibration \( t: 1 \hookrightarrow \text{Cof} \).
Cofibrations

- A monomorphism \( m: C \rightarrow Z \) is a \textbf{cofibration} if its classifying map \( \chi_m: Z \rightarrow \Omega \) factors through \( m_{\text{cof}}: \text{Cof} \rightarrow \Omega \).

\[ \begin{array}{ccc}
C & \rightarrow & 1 & \rightarrow & 1 \\
\downarrow m & & \downarrow t & & \downarrow \text{true} \\
Z & \rightarrow & \text{Cof} & \rightarrow & \Omega \\
\downarrow \chi_m & & \nearrow m_{\text{cof}} & & \\
& & \end{array} \]

- Therefore, a monomorphism \( m: C \rightarrow Z \) is a cofibration iff it is a pullback of the generic cofibration \( t: 1 \rightarrow \text{Cof} \).

**Proposition**

\( m: C \rightarrow Z \) is a cofibration \( \iff \mathcal{E} \models \forall z: Z. \text{cof}(\exists c: C. m(c) = z) \).
Forcing dominance

Consider the following polynomials

\[ 1 \xrightarrow{t} \text{Cof} \]

\[ 1 \xrightarrow{\text{true}} \Omega \]

\[ \forall \cdot \xrightarrow{p_\forall} \forall \]

where

\[ P_t(A) = \sum_{\varphi : \text{Cof}} A[\varphi] \]

\[ P_{\text{true}}(A) = \sum_{\varphi : \Omega} A[\varphi] \]

\[ P_{p_\forall}(A) = \sum_{a : \forall} A^{\text{El}(a)} \]
Forcing dominance

Because the square

\[
\begin{array}{c}
1 \\
\downarrow \scriptstyle t
\end{array}
\begin{array}{c}
1 \\
\downarrow \scriptstyle \text{true}
\end{array}
\begin{array}{c}
\text{Cof} \\
\downarrow \scriptstyle m_{\text{cof}}
\end{array}
\begin{array}{c}
\Omega
\end{array}
\]

is cartesian, we obtain a cartesian square:

\[
\begin{array}{ccc}
P_t(\text{Cof}) & \longrightarrow & P_{\text{true}}(\text{Cof}) \\
\downarrow \scriptstyle P_t(m_{\text{cof}}) & & \downarrow \scriptstyle P_{\text{true}}(m_{\text{cof}}) \\
P_t(\Omega) & \longrightarrow & P_{\text{true}}(\Omega)
\end{array}
= \sum_{\phi : \text{Cof}} \phi^{[\phi]} \longrightarrow \sum_{\phi : \Omega} \phi^{[\phi]}
\]

\[
\begin{array}{ccc}
\sum_{\phi : \text{Cof}} \phi^{[\phi]} & \longrightarrow & \sum_{\phi : \Omega} \phi^{[\phi]} \\
\downarrow & & \downarrow \\
\sum_{\phi : \text{Cof}} \phi^{[\phi]} & \longrightarrow & \sum_{\phi : \Omega} \phi^{[\phi]}
\end{array}
\]
And because

\[
1 \xrightarrow{\sim} \mathcal{V} \\
\text{true} \downarrow \quad \downarrow p_{\mathcal{V}} \Rightarrow \Omega \xrightarrow{\iota} \mathcal{V}
\]

is cartesian, we obtain a cartesian square:

\[
\begin{array}{c}
\xrightarrow{P_{\text{true}}(\Omega)} \quad \xrightarrow{P_{p_{\mathcal{V}}}(\Omega)} \\
\xrightarrow{P_{\text{true}}(\iota)} \quad \xrightarrow{P_{p_{\mathcal{V}}}(\iota)} \\
\xrightarrow{P_{\text{true}}(\mathcal{V})} \quad \xrightarrow{P_{p_{\mathcal{V}}}(\mathcal{V})}
\end{array}
\]

\[
\sum_{\varphi: \Omega} \Omega^{[\varphi]} \xrightarrow{\bigg\rightharpoonup} \sum_{a: \mathcal{V}} \Omega^{\text{El}(a)}
\]

\[
\sum_{\varphi: \Omega} \mathcal{V}^{[\varphi]} \xrightarrow{\bigg\rightharpoonup} \sum_{a: \mathcal{V}} \mathcal{V}^{\text{El}(a)}
\]
Therefore, there is a composite map $P_t(\text{Cof}) \hookrightarrow P_{\text{true}}(\text{Cof}) \rightarrow P_{\text{true}}(\Omega) \rightarrow P_{p_{\mathcal{V}}}(\Omega)$ which takes $(\varphi, \psi)$ to $(\nu \varphi, \nu \psi)$. 

\begin{tikzcd}
\text{P}_t(\text{Cof}) & \text{P}_{\text{true}}(\text{Cof}) \\
\downarrow & \downarrow \\
\text{P}_t(\Omega) & \text{P}_{\text{true}}(\Omega) & \text{P}_{p_{\mathcal{V}}}(\Omega) \\
\downarrow & \downarrow & \downarrow \\
\text{P}_{\text{true}}(\mathcal{V}) & \text{P}_{p_{\mathcal{V}}}(\mathcal{V}) \\
\end{tikzcd}
Proposition

\( \mathcal{E} \models [\text{dom}: \forall (\varphi, \psi: \Omega). \text{cof } \varphi \Rightarrow (\varphi \Rightarrow \text{cof } \psi) \Rightarrow \text{cof}(\varphi \land \psi)] \iff \text{there is a lift } \text{dom} \text{ of } \Sigma \text{ making the square commute.} \)

\[
P_t(\text{Cof}) \xrightarrow{\text{dom}} \text{Cof} \\
\downarrow \\
\Sigma \\
P_{P_\mathcal{V}}(\mathcal{V}) \xrightarrow{\Sigma} \mathcal{V}
\]

- Note – \( \Sigma: P_{P_\mathcal{V}}(\mathcal{V}) \rightarrow (\mathcal{V}) \) in above is the Natural Model (resp. CwF) interpretation of the \( \Sigma \) type-former following (Awodey, 2018).
Forcing dominance (IV)

Proposition

For \( \varphi : \text{Cof} \) and \( \psi : [\varphi] \rightarrow \text{Cof} \), the following statements hold:

(i) \( \text{dom}(t, \varphi) = \varphi = \text{dom}(\varphi, t) \).

(ii) \( \text{dom}(\text{dom}(\varphi, \psi), \theta) = \text{dom}(\varphi, \text{dom}(\psi, \theta)) \).

(iii) \( [\text{dom}(\varphi, \psi)] \equiv \sum_{x : [\varphi]} [\psi(x)] \).

Proof.

For (i), note that \( \iota(t) = \text{code}(1) \) where 1 is the terminal type. Since \( \sum_{* : 1} \varphi(*) = \iota \varphi \) and \( \iota \) is monic, \( \text{dom}(t, \varphi) = \varphi \).

For (ii), since \( \sum_{x : \iota \varphi} t \cong \text{code}(1) \) and the "Frobenius theorem" for the sum types.

For (iii), observe that

\[
[\text{dom}(\varphi, \psi)] \equiv \text{El}(\text{dom}(\varphi, \psi)) \equiv \text{El}(\sum(\iota \varphi, \iota \psi)) \equiv \sum_{x : [\varphi]} [\psi(x)] .
\]
Forcing dominance

Proposition

Cofibrations are closed under composition.
Forcing dominance

Proposition

Cofibrations are closed under composition.

Proof.

It suffices to prove that if \( m_\varphi : [\varphi] \hookrightarrow yc \) and \( m_\psi : [\psi] \hookrightarrow [\varphi] \) are cofibrations then so is their composite.
Proposition

*Cofibrations are closed under composition.*

Proof.

It suffices to prove that if \( m_\varphi: [\varphi] \hookrightarrow yc \) and \( m_\psi: [\psi] \hookrightarrow [\varphi] \) are cofibrations then so is their composite.

\[ c \Vdash [\varphi : \text{Cof}] \text{ and } c \Vdash [\psi : [\varphi] \rightarrow \text{Cof}] \text{ imply } c \Vdash [\text{dom}(\varphi, \psi) : \text{Cof}]: \]
Proposition

Cofibrations are closed under composition.

Proof.

It suffices to prove that if $m_\varphi : [\varphi] \hookrightarrow yc$ and $m_\psi : [\psi] \hookrightarrow [\varphi]$ are cofibrations then so is their composite.

$c \vdash [\varphi : \text{Cof}]$ and $c \vdash [\psi : [\varphi] \to \text{Cof}]$ imply $c \vdash [\text{dom}(\varphi, \psi) : \text{Cof}]$:

dom$(\varphi, \psi)$ classifies $m_\varphi \circ m_\psi$ since (i) $[\text{dom}(\varphi, \psi)] \equiv \sum_x : [\varphi] [\psi(x)]$, and (ii) $m_\varphi \circ m_\psi$ is the display map of the sum type $\sum_x : [\varphi] [\psi(x)]$.
The type of partial elements of a type $A$ is given by the polynomial functor

$$P_{\text{true}}(A) = \sum_{\varphi : \Omega} [\varphi] \to A.$$
The **type of partial elements** of a type $A$ is given by the polynomial functor

$$P_{\text{true}}(A) = \sum_{\varphi : \Omega} [\varphi] \to A.$$  

The type of **cofibrant partial elements** of a type $A$ is given by the polynomial functor

$$A^+ = P_t(A) = \sum_{\varphi : \text{Cof}} [\varphi] \to A.$$
Forcing for partial elements

There is a natural map

\[ \eta: A \rightarrow A^+ \]

\[ a \mapsto (\text{true, } \lambda \ast \cdot a: 1 \rightarrow A) \]
Forcing for partial elements

There is a natural map

\[ \eta: A \to A^+ \]

\[ a \mapsto (\text{true}, \lambda \ast \cdot a : 1 \to A) \]

which fits into the pullback square

\[
\begin{array}{ccc}
A & \xrightarrow{\eta} & A^+ \\
\downarrow & & \downarrow \\
1 & \xrightarrow{t} & \text{Cof}
\end{array}
\]

Proposition ((Awodey, 2018))

The map \( \eta_A : A \to A^+ \) is a cofibration and it classifies partial maps with cofibrant domain.
In fact, $\eta : \text{Id} \Rightarrow +$ is cartesian:

```
\[
\begin{array}{ccc}
A & \xrightarrow{f} & B & \xrightarrow{t} & 1 \\
\downarrow{\eta_A} & \searrow{\eta} & \downarrow{\eta_B} & & \\
A^+ & \xrightarrow{f^+} & B^+ & \rightarrow & \text{Cof}
\end{array}
\]
```

The right square & the outer rectangle are cartesian $\Rightarrow$ The left square is cartesian.
Forcing for partial elements

Proposition

\[ c \vDash [(\varphi, u) : A^+](\gamma) \iff \]
\[ c \vDash [\varphi : \text{Cof}](\gamma) \text{ and for all } f : d \to c, \text{ if } d \vDash [x : \varphi.f](\gamma.f) \text{ then } d \vDash [u_f(x) : A](\gamma.f), \]
\[ \text{where } u_f(x).g = u_{fg}(x), \text{ for all } g : d' \to d. \]
Forcing for partial elements

Proposition

\(c \vdash [(\varphi, u) : A^+](\gamma) \iff c \vdash [\varphi : \text{Cof}](\gamma) \) and for all \(f : d \to c\), if \(d \vdash [x : \varphi.f](\gamma.f)\) then \(d \vdash [u_f(x) : A](\gamma.f)\), where \(u_f(x).g = u_{fg}(x)\), for all \(g : d' \to d\).
Forcing for partial elements

The above gets simplified when $\Gamma = 1$.

\[
\begin{array}{c}
\text{Cof} \leftarrow \phi \quad yc \\
\downarrow \quad \phi \\
1 \leftarrow [\varphi, u] : A^+ \quad u \rightarrow A
\end{array}
\]
Monad structure from dominance

Proposition ((Awodey, 2018))

\[ + : \mathcal{E} \to \mathcal{E} \text{ is a (fibred) monad.} \]
Monad structure from dominance

Proposition ((Awodey, 2018))

\[ + : \mathcal{E} \to \mathcal{E} \text{ is a (fibred) monad.} \]

First, we give a category-theoretic proof.

1st Proof.

\( \eta_A, \eta_A^+ : \text{cofibrations} \Rightarrow \eta_A^+ \circ \eta_A : \text{cofibration by dominance.} \)

\( \eta_A : \text{cofibrant partial map classifier} \Rightarrow \text{there is a unique morphism } \mu_A \text{ classifying the partial map } (\eta_A^+ \circ \eta_A, \text{id}_A). \)

\[
\begin{array}{ccc}
A & \xrightarrow{\eta_A} & A \\
\downarrow & & \downarrow \eta \\
A^+ & \xrightarrow{\eta_A^+} & A^+ \\
\downarrow & \mu & \downarrow \\
A^{++} & \xrightarrow{\text{id}} & A^+
\end{array}
\]
Monad structure from dominance

(1st Proof cont’d.)

\(\mu_A\) thus obtained is natural in \(A\):

By classifying property of \(\eta_B\) the bottom square commutes since
(i) all vertical squares are pullbacks (I and II because \(\eta\) is cartesian),
(ii) the top square commutes,
(iii) \(\eta_{A+} \circ \eta_A\): cofibration by dominance.
Monad structure from dominance

(1st Proof cont’d.)

To see that $\mu \circ \eta_A^+ = \text{id}_{A^+}$, observe that the following is a pullback by an easy diagram chase using the previous diagram and the fact that $\eta$ is always monic.

\[
\begin{array}{ccc}
A & \xrightarrow{=} & A \\
\downarrow_{\eta_A} & & \downarrow_{\eta_A} \\
A^+ & \xrightarrow{\eta_A^+} & A^{++} & \xrightarrow{\mu} & A^+
\end{array}
\]

By the uniqueness of the classifying map of $(\eta_A, \text{id}_A)$, we have $\mu_A \circ \eta_A^+ = \text{id}_{A^+}$. By naturality of $\eta$,

$$\eta_A^+ \circ \eta_A = (\eta_A)^+ \circ \eta_A$$

The same argument above shows

$$\mu_A \circ \eta_A^+ = \text{id}_{A^+}.$$
Proposition ((Awodey, 2018))

\( + : \mathcal{E} \to \mathcal{E} \) is a (fibred) monad.

Now, we give a proof using Kripke–Joyal semantics.

2nd Proof.
Proposition ((Awodey, 2018))

\[ + : \mathcal{E} \to \mathcal{E} \text{ is a (fibred) monad.} \]

Now, we give a proof using Kripke–Joyal semantics.

2nd Proof. Write \( A^{++} = (A^+)^+ \).

\[ c \models [(\varphi, u) : A^{++}] \]

\[ \iff u = (\psi, u'), c \models [\varphi : \text{Cof}], \text{ and for every } f : c' \to c, \text{ if } c' \models [x : \varphi.f] \text{ then } \\
\]

\[ c' \models [\psi_f(x) : \text{Cof}], \text{ and for every } g : d \to c', \text{ if } d \models [y : \psi.g] \text{ then } d \models [u'_g(y) : A] \text{ and } u' \]

is uniform.
Proposition ((Awodey, 2018))

\[ + : \mathcal{E} \to \mathcal{E} \text{ is a (fibred) monad.} \]

Now, we give a proof using Kripke–Joyal semantics.

2nd Proof. Write \( A^{++} = (A^+)^+ \).

\[ c \models [((\varphi, u) : A^{++})] \]
\[ \iff u = (\psi, u'), \ c \models [\varphi : \text{Cof}], \text{ and for every } f : c' \to c, \text{ if } c' \models [x : \varphi.f] \text{ then } \]
\[ c' \models [\psi_f(x) : \text{Cof}], \text{ and for every } g : d \to c', \text{ if } d \models [y : \psi.g] \text{ then } d \models [u'_g(y) : A] \text{ and } u' \]
\[ \text{is uniform.} \]

Now, set \( f = \text{id}_c \).

The statement above (after \( \iff \)) becomes \( u = (\psi, u') \) and \( c \models [\varphi : \text{Cof}], \)
\[ c \models [\psi : [\varphi] \to \text{Cof}], \ c \models [u' : \sum_x [\varphi] [\psi(x)] \to A] \]
Monad structure from dominance

The latter implies

\[ c \vDash [\text{dom}(\varphi, \psi) : \text{Cof}] \quad \text{and} \quad c \vDash [u' : [\text{dom}(\varphi, \psi)] \to A]. \]
The latter implies

\[ c \vdash [\text{dom}(\varphi, \psi) : \text{Cof}] \text{ and } c \vdash [u' : \text{dom}(\varphi, \psi) \to A]. \]

Hence

\[ c \vdash [(\text{dom}(\varphi, \psi), u') : A^+] . \]
The latter implies

\[ c \vdash \text{dom}(\varphi, \psi) : \text{Cof} \text{ and } c \vdash \left[ u' : [\text{dom}(\varphi, \psi)] \to A \right]. \]

Hence

\[ c \vdash \left( \text{dom}(\varphi, \psi), u' \right) : A^+. \]

Uniformity of \( u' \) implies \( E \vdash \left[ \mu : A^{++} \to A^+ \right]. \)

By Yoneda, we get \( \mu : A^{++} \to A^+. \)
Monad structure from dominance

The latter implies

\[ c ⊩ [\text{dom}(\varphi, \psi) : \text{Cof}] \text{ and } c ⊩ [u' : [\text{dom}(\varphi, \psi)] \to A]. \]

Hence

\[ c \vdash [(\text{dom}(\varphi, \psi), u') : A^+]. \]

Uniformity of \( u' \) implies \( E \vdash [\mu : A^{++} \to A^+] \).

By Yoneda, we get \( \mu : A^{++} \to A^+ \).

Also, \( \mu \circ \eta_{A^+} = \text{id} = \mu \circ + (\eta_A) \) because \( \text{dom}(\varphi, t) = \varphi \) and \( \text{dom}(t, \psi) = \psi \).

\( \mu \circ \mu_{A^+} = \mu \circ + (\mu_A) \) because \( \text{dom}(\text{dom}(\varphi, \psi), \theta) = \text{dom}(\varphi, \text{dom}(\psi, \theta)) \).
For any type $A$ define

$$\text{TFib}(A) := \prod_{\varphi:\text{Cof}} \prod_{u:[\varphi]\to A} \sum_{a:A} u =_{\varphi} a,$$

where the type $u =_{\varphi} a$ (written $(\varphi, u) \mapsto a$ in Orton and Pitts 2018) is defined

$$(u =_{\varphi} a) := \prod_{\rho:[\varphi]} \text{Eq}_A(up, a).$$
For any type $A$ define

$$\text{TFib}(A) := \prod_{\varphi : \text{Cof}} \prod_{u : \varphi \to A} \sum_{a : A} u =_{\varphi} a,$$

where the type $u =_{\varphi} a$ (written $(\varphi, u) \looparrowright a$ in Orton and Pitts 2018) is defined

$$(u =_{\varphi} a) := \prod_{p : \varphi} \text{Eq}_A(up, a).$$

Proposition

The map $p_A : \Gamma.A \to \Gamma$ is a uniform trivial fibration $\iff$ there is a term $\Gamma \vdash \alpha : \text{TFib}(A)$. 
Proposition

The map \( p_A : \Gamma. A \to \Gamma \) is a uniform trivial fibration \( \iff \) there is a term \( \Gamma \vdash \alpha : \text{TFib}(A) \).

Recall that \( p_A \) being a uniform trivial fibration means that for \( p_A : \Gamma. A \to \Gamma \) means that

\[
\begin{array}{c}
\Gamma. A \\
\downarrow p_A \\
\Gamma
\end{array}
\]
An application of Kripke–Joyal semantics: Uniform trivial fibration

Proposition

The map $p_A : \Gamma.A \to \Gamma$ is a uniform trivial fibration $\iff$ there is a term $\Gamma \vdash \alpha : \text{TFib}(A)$.

Recall that $p_A$ being a uniform trivial fibration means that for $p_A : \Gamma.A \to \Gamma$ means that for every cofibration $C \hookrightarrow Z$

\[
\begin{array}{ccc}
C & \xrightarrow{\gamma} & \Gamma.A \\
\downarrow & & \downarrow p_A \\
Z & & \Gamma
\end{array}
\]
Proposition

The map $p_A : \Gamma. A \to \Gamma$ is a uniform trivial fibration $\iff$ there is a term $\Gamma \vdash \alpha : \text{Tfib}(A)$.

Recall that $p_A$ being a **uniform trivial fibration** means that for $p_A : \Gamma. A \to \Gamma$ means that for every cofibration $C \hookrightarrow Z$ and commutative square:

\[
\begin{array}{ccc}
C & \overset{a}{\to} & \Gamma.A \\
\downarrow & & \downarrow \text{p}_A \\
Z & \overset{z}{\to} & \Gamma
\end{array}
\]
An application of Kripke–Joyal semantics: Uniform trivial fibration

**Proposition**

*The map \( p_A : \Gamma . A \to \Gamma \) is a uniform trivial fibration \( \iff \) there is a term \( \Gamma \vdash \alpha : \text{TFib}(A) \).*

Recall that \( p_A \) being a **uniform trivial fibration** means that for \( p_A : \Gamma . A \to \Gamma \) means that for every cofibration \( C \hookrightarrow Z \) and commutative square there is a diagonal filler \( j_C(z, a) : Z \to \Gamma . A \) making both triangles commute,
Recall that $p_A$ being a **uniform trivial fibration** means that for $p_A : \Gamma.A \to \Gamma$ means that for every cofibration $C \to Z$ and commutative square there is a diagonal filler $j_C(z, a) : Z \to \Gamma.A$ making both triangles commute, and for any map $f : Z' \to Z$,
Recall that $p_A$ being a **uniform trivial fibration** means that for $p_A : \Gamma.A \to \Gamma$ means that for every cofibration $C \hookrightarrow Z$ and commutative square there is a diagonal filler $j_C(z, a) : Z \to \Gamma.A$ making both triangles commute, and for any map $f : Z' \to Z$, these maps $j$ satisfy the indicated condition,

$$j_{C'}(zf, af') = j_C(z, a) \circ f.$$
Lemma

For $\Gamma \vdash A \text{ Type}$, $\gamma : yc \to \Gamma$ such that

\[
\begin{align*}
    c &\vdash [a : A](\gamma) \\
    c &\vdash [\varphi : \text{Cof}](\gamma) \\
    c &\vdash [u : [\varphi] \to A](\gamma).
\end{align*}
\]

then we also have

\[
\begin{align*}
    c &\vdash [e : u =_{\varphi} a](\gamma) \iff [\varphi] \xrightarrow{u} \Gamma.A \xrightarrow{p_A} \Gamma \\
    yc &\xrightarrow{\gamma} \Gamma
\end{align*}
\]

commutes,

where

\[
(u =_{\varphi} a) := \prod_{x : [\varphi]} \text{Eq}_A(ux, a).
\]
Proof of Lemma.

\[
\begin{array}{c}
\varphi \\
\downarrow \\
y \in C \\
\end{array} \quad \xrightarrow{u} \quad \begin{array}{c}
\Gamma.A \\
\downarrow \\
p_A \\
\end{array}
\]

\[
\begin{array}{c}
\langle \gamma, a \rangle \\
\gamma \\
\end{array} \quad \rightarrow \quad \begin{array}{c}
\Gamma \\
\end{array}
\]

\(c \models [a : A](\gamma) \iff\) the lower triangle commutes.
Proof of Lemma.

\[
\begin{align*}
\varphi & \rightarrow_{u} \Gamma.A \\
\gamma & \rightarrow_{\langle \gamma, a \rangle} \Gamma.A \\
\gamma & \rightarrow_{\gamma} \Gamma
\end{align*}
\]

\(c \models [a : A](\gamma) \iff \text{the lower triangle commutes.}\)

\(c \models [\varphi : \text{Cof}](\gamma) \) and \(c \models [u : [\varphi] \rightarrow A](\gamma) \iff \text{the outer square commutes.}\)
Proof of Lemma.

\[
\begin{array}{ccc}
[\varphi] & \xrightarrow{u} & \Gamma.A \\
\gamma \downarrow & & \downarrow p_A \\
\gamma yc & \xrightarrow{\langle \gamma, a \rangle} & \Gamma
\end{array}
\]

\[c \models [a : A](\gamma) \iff \text{the lower triangle commutes.}\]

\[c \models [\varphi : \text{Cof}](\gamma) \text{ and } c \models [u : [\varphi] \to A](\gamma) \iff \text{the outer square commutes.}\]

\[c \models [e : u =_{\varphi} a](\gamma) \]
\[\iff c \models [e : \prod_{x : [\varphi]} \text{Eq}_A(ux, a)](\gamma)\]
\[\iff \text{for all } f : d \to c \text{ in } C, \ d \models [x : [\varphi]](\gamma.f) \text{ returns}\]
\[d \models [e_f(x) : \text{Eq}_A(ux, a)](\gamma.f, u[\gamma.f]x, a.f)\]
\[\iff \text{the top triangle commutes.} \quad \text{QED.}\]
Proof of Theorem.
Suppose $\Gamma \vdash \alpha : \text{TFib}(A)$.
Thus for all $\gamma : yc \to \Gamma$, we have $c \models [\alpha_\gamma : \text{TFib}(A)](\gamma)$, coherently in $\gamma$. 
Proof of Theorem.

Suppose $\Gamma \vdash \alpha : \text{TFib}(A)$.

Thus for all $\gamma : yc \to \Gamma$, we have $c \vdash [\alpha_\gamma : \text{TFib}(A)](\gamma)$, coherently in $\gamma$.

Note that

$$\text{TFib}(A) = \prod_{\varphi : \text{Cof}} \prod_{u : [\varphi] \to A} \sum_a \prod_{x : [\varphi]} \text{Eq}_A(ux, a)$$

$$= \prod_{(\varphi, u) : A^+} \sum_{a : A} u =_{\varphi} a$$

We thus obtain

$$c \vdash \left[\alpha_\gamma : \prod_{(\varphi, u) : A^+} \sum_{a : A} u =_{\varphi} a\right](\gamma) .$$
Proof of Theorem (cont’d).

By Kripke–Joyal semantics of $\prod$ and $\sum$, we have for every $f : d \to c$ in $\mathcal{C}$, if

$$d \models [(\varphi, u) : A^+] (\gamma.f)$$  \hspace{1cm} (2)

then

$$d \models [\alpha_{\gamma,f}(\varphi, u)^0 : A] (\gamma.f)$$ \hspace{1cm} (3)

and

$$d \models [\alpha_{\gamma,f}(\varphi, u)^1 : (u =_{\varphi} \alpha_{\gamma,f}(\varphi, u)^0)] (\gamma.f)$$ \hspace{1cm} (4)

and, for any $g : d' \to d$,

$$\alpha_{\gamma,f}(\varphi, u).g = \alpha_{(\gamma.fg)}(\varphi[g], u[g])$$ \hspace{1cm} (5)
Unfolding the condition (2) yields the following commutative diagram.

\[
\begin{array}{c}
\varphi.f \\
\downarrow \\
yd \\
\end{array} \quad \begin{array}{c}
\langle \gamma.f, u_f \rangle \quad \Gamma.A \\
\downarrow p_A \\
\end{array}
\begin{array}{c}
\gamma.f \quad \Gamma \\
\end{array}
\]
Unfolding the condition (2) yields the following commutative diagram.

\[
\begin{array}{ccc}
[\varphi.f] & \xrightarrow{\langle \gamma.f, u_f \rangle} & \Gamma.A \\
\downarrow & & \downarrow p_A \\
yd & \xrightarrow{\gamma.f} & \Gamma
\end{array}
\]

Lemma applied to (3) and (4) yields the following commuting diagram.

\[
\begin{array}{ccc}
[\varphi.f] & \xrightarrow{\langle \gamma.f, u_f \rangle} & \Gamma.A \\
\downarrow & \xleftarrow{\alpha_{\gamma.f}(\varphi,u)^0} & \downarrow p_A \\
yd & \xrightarrow{\gamma.f} & \Gamma
\end{array}
\]
Thus forcing $\text{TFib}(A)$ produces diagonal fillers

$$j_\varphi(\gamma, u) \triangleq \alpha_{\gamma,f}(\varphi, u)^0$$

for each lifting problem as in the right hand square below:
Proof of Theorem (cont’d) – converse argument

If $p_A : \Gamma.A \to \Gamma$ is a uniform trivial fibration then in particular for every basic cofibration $[\varphi] \hookrightarrow yc$ and square as on the right below, there is a diagonal filler $j_{\varphi}(\gamma, u)$ as indicated.

\[
\begin{array}{c}
[\varphi.f] \\
\downarrow \\
yc'
\end{array}
\to
\begin{array}{c}
[\varphi] \\
\downarrow \\
yc
\end{array}
\to
\begin{array}{c}
\Gamma.A \\
\downarrow p_A
\end{array}
\]

\[
\begin{array}{c}
\downarrow j_{\varphi.f}(\gamma.f,u.f) \\
yf
\end{array}
\]

\[
\begin{array}{c}
j_{\varphi}(\gamma,u) \\
\gamma
\end{array}
\]
Proof of Theorem (cont’d) – converse argument

If $p_A : \Gamma.A \to \Gamma$ is a uniform trivial fibration then in particular for every basic cofibration $[\varphi] \to yc$ and square as on the right below, there is a diagonal filler $j_\varphi(\gamma, u)$ as indicated.

\[
\begin{array}{cccccc}
[\varphi.f] & \to & [\varphi] & \mapsto & u & \to & \Gamma.A \\
\downarrow & & \downarrow & & \downarrow & & \downarrow \\
yc' & \to & yc & \to & \gamma & \to & \Gamma \\
\gamma \downarrow & & \gamma \downarrow & & \gamma \downarrow & & \gamma \downarrow \\
yf & & \gamma f & & \gamma f & & \gamma f \\
\end{array}
\]

By the lemma, this corresponds to an element $\alpha_\gamma : yc \to TFib(A)$ over $\gamma : yc \to \Gamma$,
Proof of Theorem (cont’d) – converse argument

The uniformity condition says exactly that for all $f : c' \to c$, the elements $\alpha_\gamma$ cohere,

$$\alpha(\gamma.yf) = \alpha_\gamma \circ f.$$
Proof of Theorem (cont’d) – converse argument

The uniformity condition says exactly that for all \( f : c' \to c \), the elements \( \alpha_\gamma \) cohere,

\[
\alpha(\gamma \cdot yf) = \alpha_\gamma \circ f .
\]

By Yoneda for the slice category \( \mathcal{E}/\Gamma \) that there is a term \( \Gamma \vdash \alpha : \text{Tfib}(A) \). QED.
Further use of Kripke–Joyal semantics for dependent type theory in

- Extending to uniform **fibrations** using an interval $\mathbb{I}$.
- Showing the fibrancy of path types.
- Showing the universe of fibrations is itself fibrant.
- Showing Frobenius property of fibrations.
References I


Ian Orton and Andrew M. Pitts. “Axioms for Modelling Cubical Type Theory in a Topos”. In: Logical Methods in Computer Science 14 (4 2018).
The End

Thanks for your attention!