An overlooked coherence construction for dependent type theory

Peter LeFanu Lumsdaine
(joint work with Michael Warren)

Institute for Advanced Study
Princeton, New Jersey

CT2013, Sydney

Background

- Dependent type theory: powerful, expressive, natural class of logical systems (e.g. homotopy type theory).
- Models of DTT: well-developed categorical theory, most aspects satisfactory.
- However: coherence issues still present obstructions, not fully understood.
- Existing theorems bridge the gap for specific type theories: Hofmann, van den Berg–Garner, … But: general theorems lacking, esp. for intensional type theory.

Present result

Theorem (Lumsdaine, Warren)

Let $(\mathbf{C}, \mathcal{T})$ *be a comprehension category, such that* $\mathbf{C}$ *has finite products and display maps are exonentiable in* $\mathbf{C}$*.*

Then there is an associated split comprehension category $(\mathbf{C}, \mathcal{T}_!)$*, with* $\mathcal{T} \simeq \mathcal{T}_!$ *as fibrations over* $\mathbf{C}$*; and if* $(\mathbf{C}, \mathcal{T})$ *has weakly stable* Π*-types (resp.* Σ*-types,* Id*-types,* W*-types, inductive types, higher inductive types, ...), then* $(\mathbf{C}, \mathcal{T}_!)$ *may be equipped with strictly coherent* Π*-types (resp.* Σ*-types,* Id*-types, etc.)*

Present result

Theorem (Lumsdaine, Warren)

Let $(\mathbf{C}, \mathcal{T})$ be a comprehension category, such that $\mathbf{C}$ has finite products and display maps are exonentiable in $\mathbf{C}$.

Then there is an associated split comprehension category $(\mathbf{C}, \mathcal{T}_!)$, with $\mathcal{T} \simeq \mathcal{T}_!$ as fibrations over $\mathbf{C}$; and if $(\mathbf{C}, \mathcal{T})$ has weakly stable Π-types (resp. Σ-types, Id-types, W-types, inductive types, higher inductive types, ...), then $(\mathbf{C}, \mathcal{T}_!)$ may be equipped with strictly coherent Π-types (resp. Σ-types, Id-types, etc.)

- Weakly stable satisfied in natural categorical settings; split + strictly coherent allows direct interpretation of syntax.

Present result

Theorem (Lumsdaine, Warren)

Let $(\mathbf{C}, \mathcal{T})$ be a comprehension category, such that $\mathbf{C}$ has finite products and display maps are exonentiable in $\mathbf{C}$.

Then there is an associated split comprehension category $(\mathbf{C}, \mathcal{T}_!)$, with $\mathcal{T} \simeq \mathcal{T}_!$ as fibrations over $\mathbf{C}$; and if $(\mathbf{C}, \mathcal{T})$ has weakly stable Π-types (resp. Σ-types, Id-types, W-types, inductive types, higher inductive types, ...), then $(\mathbf{C}, \mathcal{T}_!)$ may be equipped with strictly coherent Π-types (resp. Σ-types, Id-types, etc.)

- Weakly stable satisfied in natural categorical settings; split + strictly coherent allows direct interpretation of syntax.
- Main hypothesis: the exponentiability.

Present result

Theorem (Lumsdaine, Warren)

Let $(\mathbf{C}, \mathcal{T})$ be a comprehension category, such that $\mathbf{C}$ has finite products and display maps are exonentiable in $\mathbf{C}$.

Then there is an associated split comprehension category $(\mathbf{C}, \mathcal{T}_!)$, with $\mathcal{T} \simeq \mathcal{T}_!$ as fibrations over $\mathbf{C}$; and if $(\mathbf{C}, \mathcal{T})$ has weakly stable Π-types (resp. Σ-types, Id-types, W-types, inductive types, higher inductive types, ...), then $(\mathbf{C}, \mathcal{T}_!)$ may be equipped with strictly coherent Π-types (resp. Σ-types, Id-types, etc.)

- Weakly stable satisfied in natural categorical settings; split + strictly coherent allows direct interpretation of syntax.
- Main hypothesis: the exponentiability.
- Payoff: no restriction on type theory; result is uniform for all type-constructors (even for individual rules).

Comprehension Categories

Definition (Jacobs)

A comprehension category $(\mathbf{C}, \mathcal{T})$ is a (Grothendieck) fibration $p\colon \mathcal{T} \to \mathbf{C}$, together with a functor $\chi\colon \mathcal{T} \to \mathbf{C}^{\to}$, such that $\mathrm{cod} \circ \chi = p$, and χ sends cartesian arrows to pullback squares.

$(\mathbf{C}, \mathcal{T})$ is full if χ is full, and split (resp. cloven) if p is split (resp. cloven).

- Idea: see objects of $\mathbf{C}$ as contexts Γ, objects of $\mathcal{T}(\Gamma)$ as types A in context Γ, and χ as providing context extension $\Gamma.A$.

Comprehension Categories

Definition (Jacobs)

A comprehension category $(\mathbf{C}, \mathcal{T})$ is a (Grothendieck) fibration $p \colon \mathcal{T} \to \mathbf{C}$, together with a functor $\chi \colon \mathcal{T} \to \mathbf{C}^{\to}$, such that $\mathrm{cod} \circ \chi = p$, and χ sends cartesian arrows to pullback squares.

$(\mathbf{C}, \mathcal{T})$ is full if χ is full, and split (resp. cloven) if p is split (resp. cloven).

- Idea: see objects of $\mathbf{C}$ as contexts Γ, objects of $\mathcal{T}(\Gamma)$ as types A in context Γ, and χ as providing context extension $\Gamma.A$.
- Comprehension categories abound in nature.

Comprehension Categories

Definition (Jacobs)

A comprehension category $(\mathbf{C}, \mathcal{T})$ is a (Grothendieck) fibration $p \colon \mathcal{T} \to \mathbf{C}$, together with a functor $\chi \colon \mathcal{T} \to \mathbf{C}^{\to}$, such that $\mathrm{cod} \circ \chi = p$, and χ sends cartesian arrows to pullback squares.

$(\mathbf{C}, \mathcal{T})$ is full if χ is full, and split (resp. cloven) if p is split (resp. cloven).

- Idea: see objects of $\mathbf{C}$ as contexts Γ, objects of $\mathcal{T}(\Gamma)$ as types A in context Γ, and χ as providing context extension $\Gamma.A$.
- Comprehension categories abound in nature.
- Split comprehension categories model the structural core of DTT.

Comprehension Categories

Definition (Jacobs)

A comprehension category $(\mathbf{C}, \mathcal{T})$ is a (Grothendieck) fibration $p \colon \mathcal{T} \to \mathbf{C}$, together with a functor $\chi \colon \mathcal{T} \to \mathbf{C}^{\to}$, such that $\mathrm{cod} \circ \chi = p$, and χ sends cartesian arrows to pullback squares.

$(\mathbf{C}, \mathcal{T})$ is full if χ is full, and split (resp. cloven) if p is split (resp. cloven).

- Idea: see objects of $\mathbf{C}$ as contexts Γ, objects of $\mathcal{T}(\Gamma)$ as types A in context Γ, and χ as providing context extension $\Gamma.A$.
- Comprehension categories abound in nature.
- Split comprehension categories model the structural core of DTT.
- (Alternatives: contextual categories; categories with attributes/families; type-categories; etc.)

Comprehension Categories: example

Example

Sets, with the codomain fibration $(\mathbf{Sets}^{\to})_X = \mathbf{Sets}/X$, is a comprehension category. (Call this just **Sets**.)

Comprehension Categories: example

Example

Sets, with the codomain fibration $(\mathbf{Sets}^{\to})_X = \mathbf{Sets}/X$, is a comprehension category. (Call this just **Sets**.)

Example

Fix some class of sets V (e.g. a Grothendieck universe; or all sets). Define split comprehension category $\mathbf{Sets}_V = (\mathbf{Sets}, \mathcal{T}_V)$ by:

- $\mathcal{T}(X) := [X, V]$;
- re-indexing is precomposition;
- χ is disjoint union.

If V is the class of all sets, then $\mathbf{Sets}_V \simeq \mathbf{Sets}$.

Comprehension Categories: example

Example

Sets, with the codomain fibration $(\mathbf{Sets}^{\rightarrow})_X = \mathbf{Sets}/X$, is a comprehension category. (Call this just **Sets**.)

Example

Fix some class of sets V (e.g. a Grothendieck universe; or all sets). Define split comprehension category $\mathbf{Sets}_V = (\mathbf{Sets}, \mathcal{T}_V)$ by:

- $\mathcal{T}(X) := [X, V]$;
- re-indexing is precomposition;
- χ is disjoint union.

If V is the class of all sets, then $\mathbf{Sets}_V \simeq \mathbf{Sets}$.

Moral: universes make things stricter, when available.

Logical structure

Definition

$(\mathbf{C}, \mathcal{T})$ has (+)-types if for each $\Gamma \in \mathbf{C}$ and $A, B \in \mathcal{T}(\Gamma)$, there is an object $A + B \in \mathcal{T}(\Gamma)$, and maps ν_A, ν_B, such that …

Logical structure

Definition

$(\mathbf{C}, \mathcal{T})$ has $(+)$-types if for each $\Gamma \in \mathbf{C}$ and $A, B \in \mathcal{T}(\Gamma)$, there is an object $A + B \in \mathcal{T}(\Gamma)$, and maps ν_A, ν_B, such that …

$(\mathbf{C}, \mathcal{T})$ has (type-theoretic) Π-types if for each $\Gamma \in \mathbf{C}$, $A \in \mathcal{T}(\Gamma)$, and $B \in \mathcal{T}(\Gamma.B)$, there is an object $\Pi_A B \in \mathcal{T}(\Gamma)$, and maps λ, ap, such that …

Logical structure

Definition

$(\mathbf{C}, \mathcal{T})$ has $(+)$-types if for each $\Gamma \in \mathbf{C}$ and $A, B \in \mathcal{T}(\Gamma)$, there is an object $A + B \in \mathcal{T}(\Gamma)$, and maps ν_A, ν_B, such that …

$(\mathbf{C}, \mathcal{T})$ has (type-theoretic) Π-types if for each $\Gamma \in \mathbf{C}$, $A \in \mathcal{T}(\Gamma)$, and $B \in \mathcal{T}(\Gamma.B)$, there is an object $\Pi_A B \in \mathcal{T}(\Gamma)$, and maps λ, ap, such that …

A split comp. cat. $(\mathbf{C}, \mathcal{T})$ has strictly coherent $(+)$-types (resp. Π-types) if it is equipped with choices of the above data, commuting with the splitting (i.e. with substitution in the ambient context Γ).

Logical structure

Definition

$(\mathbf{C}, \mathcal{T})$ has $(+)$-types if for each $\Gamma \in \mathbf{C}$ and $A, B \in \mathcal{T}(\Gamma)$, there is an object $A + B \in \mathcal{T}(\Gamma)$, and maps ν_A, ν_B, such that …

$(\mathbf{C}, \mathcal{T})$ has (type-theoretic) Π-types if for each $\Gamma \in \mathbf{C}$, $A \in \mathcal{T}(\Gamma)$, and $B \in \mathcal{T}(\Gamma.B)$, there is an object $\Pi_A B \in \mathcal{T}(\Gamma)$, and maps λ, ap, such that …

A split comp. cat. $(\mathbf{C}, \mathcal{T})$ has strictly coherent $(+)$-types (resp. Π-types) if it is equipped with choices of the above data, commuting with the splitting (i.e. with substitution in the ambient context Γ).

- The strictly coherent structure is exactly what's required to model syntactic $(+)$-types, Π-types, etc.

Logical structure

Definition

$(\mathbf{C}, \mathcal{T})$ has (+)-types if for each $\Gamma \in \mathbf{C}$ and $A, B \in \mathcal{T}(\Gamma)$, there is an object $A + B \in \mathcal{T}(\Gamma)$, and maps ν_A, ν_B, such that …

$(\mathbf{C}, \mathcal{T})$ has (type-theoretic) Π-types if for each $\Gamma \in \mathbf{C}$, $A \in \mathcal{T}(\Gamma)$, and $B \in \mathcal{T}(\Gamma.B)$, there is an object $\Pi_A B \in \mathcal{T}(\Gamma)$, and maps λ, ap, such that …

A split comp. cat. $(\mathbf{C}, \mathcal{T})$ has strictly coherent (+)-types (resp. Π-types) if it is equipped with choices of the above data, commuting with the splitting (i.e. with substitution in the ambient context Γ).

- The strictly coherent structure is exactly what's required to model syntactic (+)-types, Π-types, etc.
- **Sets** has (+)-types and Π-types. For V suitably closed, $\mathbf{Sets}_V$ has strictly coherent (+)-types and Π-types.

Coherence problem

Splitness and strict coherence of logical structure are not categorical: involve equality on objects.

Most intended models are categorical in nature; don't naturally satisfy these.

Coherence problem

Splitness and strict coherence of logical structure are not categorical: involve equality on objects.

Most intended models are categorical in nature; don't naturally satisfy these.

Problem (Coherence problem for type theory)

Given a comp. cat. with some weak logical structure, when can one construct a related (equivalent?) split one, with strict logical structure?

Expect some kind of stability condition to be needed on the logical structure.

Coherence for fibrations

First: splitness of the fibration.

Coherence for fibrations

First: splitness of the fibration.

Proposition (Giraud, Grothendieck)

For any $\mathbf{C}$, *there is a triple adjunction*

$$\mathbf{Fib}(\mathbf{C}) \underset{(-)_*}{\overset{(-)_!}{\rightleftarrows}} \mathbf{Fib}_{spl}(\mathbf{C}), \qquad (-)_! \dashv (-)^* \dashv (-)_*, \quad (-)^* : \mathbf{Fib}_{spl}(\mathbf{C}) \to \mathbf{Fib}(\mathbf{C})$$

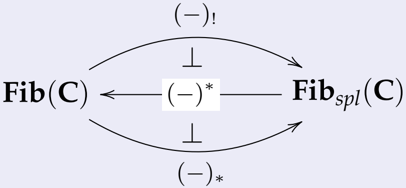
and for any $\mathcal{T} \in \mathbf{Fib}(\mathbf{C})$, *both* $\mathcal{T}_*$ *and* $\mathcal{T}_!$ *are equivalent to* $\mathcal{T}$ *as (non-split) fibrations over* $\mathbf{C}$.

Coherence for fibrations

First: splitness of the fibration.

Proposition (Giraud, Grothendieck)

For any $\mathbf{C}$*, there is a triple adjunction*



and for any $\mathcal{T} \in \mathbf{Fib}(\mathbf{C})$*, both* $\mathcal{T}_*$ *and* $\mathcal{T}_!$ *are equivalent to* $\mathcal{T}$ *as (non-split) fibrations over* $\mathbf{C}$*.*

Right adjoint $\mathcal{T}_*$: "choose re-indexings in advance".

Left adjoint $\mathcal{T}_!$: "put off re-indexing until later".

Coherence using right adjoint

Proposition (Hofmann, 1995)

Suppose $(\mathbf{C}, \mathcal{T})$ *is a comprehension category with identity types, satisfying the reflection rule (of extensional type theory).*

Then $(\mathbf{C}, \mathcal{T}_*)$ *is again a comprehension category; and if* $(\mathbf{C}, \mathcal{T})$ *has* Π*-types (resp.* Σ*-types,* W*-types, etc.) commuting up to isomorphism with reindexing, then* $(\mathbf{C}, \mathcal{T}_*)$ *has strictly coherent* Π*-types (*Σ*-types. etc.).*

Coherence using right adjoint

Proposition (Hofmann, 1995)

Suppose $(\mathbf{C}, \mathcal{T})$ is a comprehension category with identity types, satisfying the reflection rule (of extensional type theory).

Then $(\mathbf{C}, \mathcal{T}_*)$ is again a comprehension category; and if $(\mathbf{C}, \mathcal{T})$ has Π-types (resp. Σ-types, W-types, etc.) commuting up to isomorphism with reindexing, then $(\mathbf{C}, \mathcal{T}_*)$ has strictly coherent Π-types (Σ-types. etc.).

Generally: if a type-constructor is forced by its definition to be unique up to canonical isomorphism, or if a term-constructor is forced to be unique, then the constructor will lift to $(\mathbf{C}, \mathcal{T}_*)$. Extensional type theory is happy.

Coherence using right adjoint

Proposition (Hofmann, 1995)

Suppose $(\mathbf{C}, \mathcal{T})$ *is a comprehension category with identity types, satisfying the reflection rule (of extensional type theory).*

Then $(\mathbf{C}, \mathcal{T}_*)$ *is again a comprehension category; and if* $(\mathbf{C}, \mathcal{T})$ *has* Π*-types (resp.* Σ*-types,* W*-types, etc.) commuting up to isomorphism with reindexing, then* $(\mathbf{C}, \mathcal{T}_*)$ *has strictly coherent* Π*-types (*Σ*-types. etc.).*

Generally: if a type-constructor is forced by its definition to be unique up to canonical isomorphism, or if a term-constructor is forced to be unique, then the constructor will lift to $(\mathbf{C}, \mathcal{T}_*)$. Extensional type theory is happy.

This kills the homotopy theory!

Coherence using right adjoint

Proposition (Hofmann, 1995)

Suppose $(\mathbf{C}, \mathcal{T})$ *is a comprehension category with identity types, satisfying the reflection rule (of extensional type theory).*

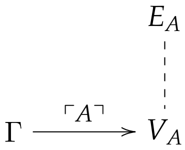
Then $(\mathbf{C}, \mathcal{T}_*)$ *is again a comprehension category; and if* $(\mathbf{C}, \mathcal{T})$ *has* Π*-types (resp.* Σ*-types,* W*-types, etc.) commuting up to isomorphism with reindexing, then* $(\mathbf{C}, \mathcal{T}_*)$ *has strictly coherent* Π*-types (*Σ*-types. etc.).*

Generally: if a type-constructor is forced by its definition to be unique up to canonical isomorphism, or if a term-constructor is forced to be unique, then the constructor will lift to $(\mathbf{C}, \mathcal{T}_*)$. Extensional type theory is happy.

This kills the homotopy theory!

Coherence using left adjoint?

An object A of $\mathcal{T}_!$ over $\Gamma \in \mathbf{C}$ consists of objects $V_A \in \mathbf{C}$, $E_A \in \mathcal{T}(V_A)$, and a map $\ulcorner A \urcorner : \Gamma \to V_A$.



Reindexing: pre-composition with $\ulcorner A \urcorner$.

Intuition

- A is a stand-in for $\ulcorner A \urcorner^* E_A \in \mathcal{T}(\Gamma)$;
- (V_A, E_A) as local universe or space of names;
- $\ulcorner A \urcorner$ as delayed substitution.

Comprehension structure

If $(\mathbf{C}, \mathcal{T})$ is a comprehension category, define comprehension on $(\mathbf{C}, \mathcal{T}_!)$ by re-indexing followed by comprehension in $(\mathbf{C}, \mathcal{T})$:

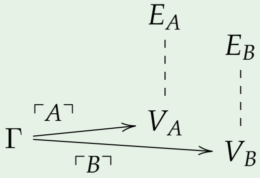
$$\Gamma.A := \Gamma.(\ulcorner A \urcorner^* E_A)$$

So: $(\mathbf{C}, \mathcal{T}_!)$ a split comprehension category, equivalent to $\mathcal{T}$.

What about logical structure?

Example

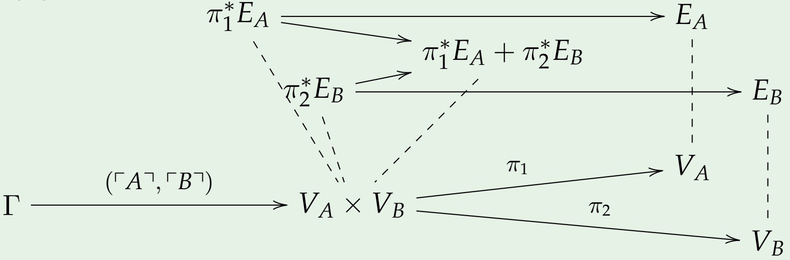
Suppose $\mathbf{C}$ has $(+)$-types. Given $A, B \in \mathcal{T}_!(\Gamma)$, how to form $A + B$?



Manipulating universes

Example

Answer: change the universe. Re-index to $V_A \times V_B$; take sum there.



Idea: $V_A \times V_B$ parametrises "sums of a type from V_A and a type from V_B".

More precisely: $V_A \times V_B$ represents the data for "$(+)$-formation with types from V_A and V_B".

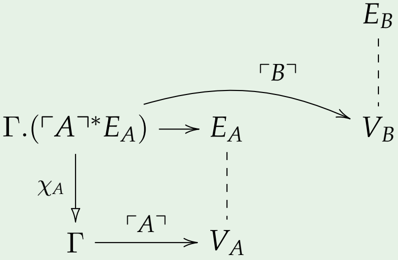
Commutes with re-indexing, since there's no interaction with Γ.

Manipulating universes, 2

Example

More difficult example: Π-types.

Data for Π-type formation: a type $A \in \mathcal{T}_!(\Gamma)$, and a further dependent type $B \in T_!(\Gamma.A)$.



What universe do we reindex to?

What represents data like $(\ulcorner A \urcorner, \ulcorner B \urcorner)$, i.e. "Π-formation data with types from V_A, V_B"?

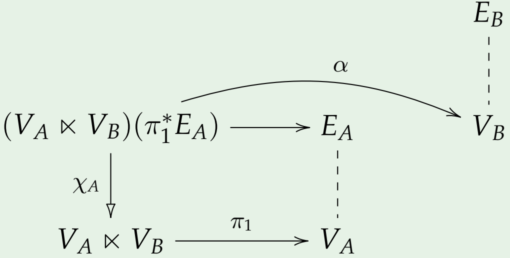
Manipulating universes, 2

Example

Write $V_A \ltimes V_B$ for exponential in $\mathbf{C}/V_A$ of $V_A \times V_B$ by $V_A.E_A$.

In internal language: $V_A \ltimes V_B := [a : V_A,\, b : V_B^{E_A(a)}]$.

This represents "Π-formation data with types from V_A, V_B", and carries universal such data (π_1, α):



Manipulating universes, 2

Example

Now can define the Π-type $\Pi_A B$ in $\mathcal{T}_!(\Gamma)$:

Manipulating universes, 2

Example

Now can define the Π-type $\Pi_A B$ in $\mathcal{T}_!(\Gamma)$:

- take $V_{\Pi_A B}$ to be $V_A \ltimes V_B$;

Manipulating universes, 2

Example

Now can define the Π-type $\Pi_A B$ in $\mathcal{T}_!(\Gamma)$:

- take $V_{\Pi_A B}$ to be $V_A \ltimes V_B$;
- take $E_{\Pi_A B}$ to be the Π-type from $\mathcal{T}$ of the universal family over $V_{\Pi_A B}$, i.e. $\Pi_{\pi_1^* E_A} \alpha^* E_B$;

Manipulating universes, 2

Example

Now can define the Π-type $\Pi_A B$ in $\mathcal{T}_!(\Gamma)$:

- take $V_{\Pi_A B}$ to be $V_A \ltimes V_B$;
- take $E_{\Pi_A B}$ to be the Π-type from $\mathcal{T}$ of the universal family over $V_{\Pi_A B}$, i.e. $\Pi_{\pi_1^* E_A} \alpha^* E_B$;
- take $\ulcorner \Pi_A B \urcorner$ to be the map $\Gamma \to V_A \ltimes V_B$ induced by $(\ulcorner A \urcorner, \ulcorner B \urcorner)$.

Manipulating universes, 2

Example

Now can define the Π-type $\Pi_A B$ in $\mathcal{T}_!(\Gamma)$:

- take $V_{\Pi_A B}$ to be $V_A \ltimes V_B$;
- take $E_{\Pi_A B}$ to be the Π-type from $\mathcal{T}$ of the universal family over $V_{\Pi_A B}$, i.e. $\Pi_{\pi_1^* E_A} \alpha^* E_B$;
- take $\ulcorner \Pi_A B \urcorner$ to be the map $\Gamma \to V_A \ltimes V_B$ induced by $(\ulcorner A \urcorner, \ulcorner B \urcorner)$.

Again, commutes strictly with re-indexing, since the Π-type taken in $\mathcal{T}$ depended only on the universes $(V_A, E_A), (V_B, E_B)$. No interaction with Γ.

Manipulating universes, 3

Also need to construct in $\mathcal{T}_!$ the term-constructors going along with these type-constructors. For each piece of structure, use the same approach as above:

Manipulating universes, 3

Also need to construct in $\mathcal{T}_!$ the term-constructors going along with these type-constructors. For each piece of structure, use the same approach as above:

1. construct a new universe, representing the input data of the operation, e.g. representing "$(+)$-elimination data, over a sum of types from V_A and V_B, into a target type from V_C";

Manipulating universes, 3

Also need to construct in $\mathcal{T}_!$ the term-constructors going along with these type-constructors. For each piece of structure, use the same approach as above:

1. construct a new universe, representing the input data of the operation, e.g. representing "$(+)$-elimination data, over a sum of types from V_A and V_B, into a target type from V_C";
2. use the assumed structure in $(\mathbf{C}, \mathcal{T})$, over this universe, taking e.g. "the universal $(+)$-elimination with types from V_A, V_B, V_C";

Manipulating universes, 3

Also need to construct in $\mathcal{T}_!$ the term-constructors going along with these type-constructors. For each piece of structure, use the same approach as above:

1. construct a new universe, representing the input data of the operation, e.g. representing "$(+)$-elimination data, over a sum of types from V_A and V_B, into a target type from V_C";
2. use the assumed structure in $(\mathbf{C}, \mathcal{T})$, over this universe, taking e.g. "the universal $(+)$-elimination with types from V_A, V_B, V_C";
3. take the new "name" map from Γ to be the map corresponding to the original supplied data over Γ, induced by the universal property of the new universe.

Manipulating universes, 3

Also need to construct in $\mathcal{T}_!$ the term-constructors going along with these type-constructors. For each piece of structure, use the same approach as above:

1. construct a new universe, representing the input data of the operation, e.g. representing "$(+)$-elimination data, over a sum of types from V_A and V_B, into a target type from V_C";
2. use the assumed structure in $(\mathbf{C}, \mathcal{T})$, over this universe, taking e.g. "the universal $(+)$-elimination with types from V_A, V_B, V_C";
3. take the new "name" map from Γ to be the map corresponding to the original supplied data over Γ, induced by the universal property of the new universe.

However: when we use e.g. the $(+)$-elimination of $\mathbf{C}$, we have re-indexed from $V_A \times V_B$ to the new universe.

So: need some kind of stability/Beck-Chevalley condition for $(+)$-types of $\mathbf{C}$.

Beck-Chevalley conditions

Traditional Beck-Chevalley condition for e.g. coproducts:

"The canonical map $f^*A + f^*B \to f^*(A+B)$ is an isomorphism".

Beck-Chevalley conditions

Traditional Beck-Chevalley condition for e.g. coproducts:

"The canonical map $f^*A + f^*B \to f^*(A + B)$ is an isomorphism".

Alternative rephrasing:

"$f^*(A + B)$, with $f^*\nu_1, f^*\nu_2$, is a coproduct for f^*A and f^*B".

Beck-Chevalley conditions

Traditional Beck-Chevalley condition for e.g. coproducts:

"The canonical map $f^*A + f^*B \to f^*(A + B)$ is an isomorphism".

Alternative rephrasing:

"$f^*(A + B)$, with $f^*\nu_1, f^*\nu_2$, is a coproduct for f^*A and f^*B".

Equivalent for coproducts, since those are unique up to canonical isomorphism, by their universal property.

For constructions with weaker universal properties: second phrasing is what we need.

Weakly stable constructors

Definition

$(\mathbf{C}, \mathcal{T})$ has weakly stable $(+)$-types if for every $A, B \in \mathcal{T}(\Gamma)$, there are $A + B$, ν_1, ν_2, such that for each $f \colon \Gamma' \to \Gamma$, the re-indexings $f^*(A+B), f^*\nu_1, f^*\nu_2$ form a $(+)$-type for f^*A, f^*B.

Similarly, define weakly stable Π-types, Id-types, etc.: a weakly stable widget for some input data is a widget, all of whose re-indexings are again widgets for the re-indexed data.

Weakly stable constructors

Definition

$(\mathbf{C}, \mathcal{T})$ has weakly stable $(+)$-types if for every $A, B \in \mathcal{T}(\Gamma)$, there are $A + B$, ν_1, ν_2, such that for each $f : \Gamma' \to \Gamma$, the re-indexings $f^*(A+B), f^*\nu_1, f^*\nu_2$ form a $(+)$-type for f^*A, f^*B.

Similarly, define weakly stable Π-types, Id-types, etc.: a weakly stable widget for some input data is a widget, all of whose re-indexings are again widgets for the re-indexed data.

- Pure existence condition: no chosen structure involved.

Weakly stable constructors

Definition

$(\mathbf{C}, \mathcal{T})$ has weakly stable $(+)$-types if for every $A, B \in \mathcal{T}(\Gamma)$, there are $A + B$, ν_1, ν_2, such that for each $f : \Gamma' \to \Gamma$, the re-indexings $f^*(A + B), f^*\nu_1, f^*\nu_2$ form a $(+)$-type for f^*A, f^*B.

Similarly, define weakly stable Π-types, Id-types, etc.: a weakly stable widget for some input data is a widget, all of whose re-indexings are again widgets for the re-indexed data.

- Pure existence condition: no chosen structure involved.
- Independent of choice of re-indexings in $\mathcal{T}$.

Weakly stable constructors

Definition

$(\mathbf{C}, \mathcal{T})$ has weakly stable $(+)$-types if for every $A, B \in \mathcal{T}(\Gamma)$, there are $A + B$, ν_1, ν_2, such that for each $f : \Gamma' \to \Gamma$, the re-indexings $f^*(A + B), f^*\nu_1, f^*\nu_2$ form a $(+)$-type for f^*A, f^*B.

Similarly, define weakly stable Π-types, Id-types, etc.: a weakly stable widget for some input data is a widget, all of whose re-indexings are again widgets for the re-indexed data.

- Pure existence condition: no chosen structure involved.
- Independent of choice of re-indexings in $\mathcal{T}$.
- Categorically natural: no equality on objects involved.

Weakly stable constructors

Definition

$(\mathbf{C}, \mathcal{T})$ has weakly stable $(+)$-types if for every $A, B \in \mathcal{T}(\Gamma)$, there are $A + B$, ν_1, ν_2, such that for each $f : \Gamma' \to \Gamma$, the re-indexings $f^*(A+B), f^*\nu_1, f^*\nu_2$ form a $(+)$-type for f^*A, f^*B.

Similarly, define weakly stable Π-types, Id-types, etc.: a weakly stable widget for some input data is a widget, all of whose re-indexings are again widgets for the re-indexed data.

- Pure existence condition: no chosen structure involved.
- Independent of choice of re-indexings in $\mathcal{T}$.
- Categorically natural: no equality on objects involved.
- Implied by most (all?) other Beck-Chevalley-type conditions; holds in most (all?) reasonable examples.

Weakly stable constructors

Definition

$(\mathbf{C}, \mathcal{T})$ has weakly stable $(+)$-types if for every $A, B \in \mathcal{T}(\Gamma)$, there are $A + B$, ν_1, ν_2, such that for each $f \colon \Gamma' \to \Gamma$, the re-indexings $f^*(A + B), f^*\nu_1, f^*\nu_2$ form a $(+)$-type for f^*A, f^*B.

Similarly, define weakly stable Π-types, Id-types, etc.: a weakly stable widget for some input data is a widget, all of whose re-indexings are again widgets for the re-indexed data.

- Pure existence condition: no chosen structure involved.
- Independent of choice of re-indexings in $\mathcal{T}$.
- Categorically natural: no equality on objects involved.
- Implied by most (all?) other Beck-Chevalley-type conditions; holds in most (all?) reasonable examples.

Coherence using left adjoint

Theorem (Lumsdaine, Warren)

Let $(\mathbf{C}, \mathcal{T})$ *be a comprehension category, such that* $\mathbf{C}$ *has finite products and display maps are exonentiable in* $\mathbf{C}$*.*

Then $(\mathbf{C}, \mathcal{T}_!)$ *is a split comprehension category, with* $\mathcal{T} \simeq \mathcal{T}_!$ *as fibrations over* $\mathbf{C}$*; and if* $(\mathbf{C}, \mathcal{T})$ *has weakly stable* Π*-types (resp.* Σ*-types,* Id*-types,* W*-types, inductive types, higher inductive types, ...), then* $(\mathbf{C}, \mathcal{T}_!)$ *may be equipped with strictly coherent* Π*-types (resp.* Σ*-types,* Id*-types, etc.)*

Coherence using left adjoint

Theorem (Lumsdaine, Warren)

Let $(\mathbf{C}, \mathcal{T})$ *be a comprehension category, such that* $\mathbf{C}$ *has finite products and display maps are exonentiable in* $\mathbf{C}$*.*

Then $(\mathbf{C}, \mathcal{T}_!)$ *is a split comprehension category, with* $\mathcal{T} \simeq \mathcal{T}_!$ *as fibrations over* $\mathbf{C}$*; and if* $(\mathbf{C}, \mathcal{T})$ *has weakly stable* Π*-types (resp.* Σ*-types,* Id*-types,* W*-types, inductive types, higher inductive types, ...), then* $(\mathbf{C}, \mathcal{T}_!)$ *may be equipped with strictly coherent* Π*-types (resp.* Σ*-types,* Id*-types, etc.)*

- Hypotheses all categorically natural.

Coherence using left adjoint

Theorem (Lumsdaine, Warren)

Let $(\mathbf{C}, \mathcal{T})$ *be a comprehension category, such that* $\mathbf{C}$ *has finite products and display maps are exonentiable in* $\mathbf{C}$*.*

Then $(\mathbf{C}, \mathcal{T}_!)$ *is a split comprehension category, with* $\mathcal{T} \simeq \mathcal{T}_!$ *as fibrations over* $\mathbf{C}$*; and if* $(\mathbf{C}, \mathcal{T})$ *has weakly stable* Π*-types (resp.* Σ*-types,* Id*-types,* W*-types, inductive types, higher inductive types, ...), then* $(\mathbf{C}, \mathcal{T}_!)$ *may be equipped with strictly coherent* Π*-types (resp.* Σ*-types,* Id*-types, etc.)*

- Hypotheses all categorically natural.
- Uniform in all type constructors; indeed, in all standard rules individually.

Coherence using left adjoint

Theorem (Lumsdaine, Warren)

Let $(\mathbf{C}, \mathcal{T})$ *be a comprehension category, such that* $\mathbf{C}$ *has finite products and display maps are exonentiable in* $\mathbf{C}$*.*

Then $(\mathbf{C}, \mathcal{T}_!)$ *is a split comprehension category, with* $\mathcal{T} \simeq \mathcal{T}_!$ *as fibrations over* $\mathbf{C}$*; and if* $(\mathbf{C}, \mathcal{T})$ *has weakly stable* Π*-types (resp.* Σ*-types,* Id*-types,* W*-types, inductive types, higher inductive types, ...), then* $(\mathbf{C}, \mathcal{T}_!)$ *may be equipped with strictly coherent* Π*-types (resp.* Σ*-types,* Id*-types, etc.)*

- Hypotheses all categorically natural.
- Uniform in all type constructors; indeed, in all standard rules individually.
- Applies to most natural models of intensional type theory (exception: **Top**).

Coherence using left adjoint

Theorem (Lumsdaine, Warren)

Let $(\mathbf{C}, \mathcal{T})$ *be a comprehension category, such that* $\mathbf{C}$ *has finite products and display maps are exonentiable in* $\mathbf{C}$*.*

Then $(\mathbf{C}, \mathcal{T}_!)$ *is a split comprehension category, with* $\mathcal{T} \simeq \mathcal{T}_!$ *as fibrations over* $\mathbf{C}$*; and if* $(\mathbf{C}, \mathcal{T})$ *has weakly stable* Π*-types (resp.* Σ*-types,* Id*-types,* W*-types, inductive types, higher inductive types, …), then* $(\mathbf{C}, \mathcal{T}_!)$ *may be equipped with strictly coherent* Π*-types (resp.* Σ*-types,* Id*-types, etc.)*

- Hypotheses all categorically natural.
- Uniform in all type constructors; indeed, in all standard rules individually.
- Applies to most natural models of intensional type theory (exception: **Top**).
- Only strong hypothesis: the exponentiability.