A Type Theory for Comprehension Categories

Niyousha Najmaei, Niels van der Weide, Benedikt Ahrens, Paige Randall North

MSFP Workshop — 18 July 2026

Overview

- *From Semantics to Syntax: A Type Theory for Comprehension Categories* [Naj+26], POPL 2026
- Ongoing work in its continuation

Setting

- Martin-Löf type theory (MLTT) serves as a foundation for proof assistants and programming languages
- Several well-established categorical semantics: contextual categories, categories with families, display map categories, natural models, ...
- Comprehension categories as a general framework to organize these notions [ALN24]:

  > *"We take comprehension categories as a unifying language and show how almost all established notions of model embed as sub-2-categories (usually full) of the 2-category of comprehension categories."*

Motivation

Interpretation of MLTT in comprehension categories:

$$
\text{MLTT} \xrightarrow{\text{semantics}}
\begin{tikzcd}
\mathcal{T} \arrow[rr, "\chi"] \arrow[dr, "p"'] & & \mathcal{C}^{\to} \arrow[dl, "\mathrm{cod}"] \\
& \mathcal{C} &
\end{tikzcd}
$$

Semantics does not use all the features of
$$
\begin{tikzcd}
\mathcal{T} \arrow[rr, "\chi"] \arrow[dr, "p"'] & & \mathcal{C}^{\to} \arrow[dl, "\mathrm{cod}"] \\
& \mathcal{C} &
\end{tikzcd}
$$

Motivation

Interpretation of MLTT in comprehension categories:

$$\text{MLTT} \xrightarrow{\text{semantics}} \begin{tikzcd} \mathcal{T} \arrow[rr, "\chi"] \arrow[dr, "p"'] & & \mathcal{C}^{\to} \arrow[dl, "\mathsf{cod}"] \\ & \mathcal{C} & \end{tikzcd}$$

Semantics does not use all the features of $\begin{tikzcd} \mathcal{T} \arrow[rr, "\chi"] \arrow[dr, "p"'] & & \mathcal{C}^{\to} \arrow[dl, "\mathsf{cod}"] \\ & \mathcal{C} & \end{tikzcd}$

Two options:

1. Restrict the comprehension categories to 'simple' ones (fully faithful or discrete)

Motivation

Interpretation of MLTT in comprehension categories:

$$\text{MLTT} \xrightarrow{\text{semantics}} \begin{tikzcd} \mathcal{T} \arrow[rr, "\chi"] \arrow[dr, "p"'] & & \mathcal{C}^{\rightarrow} \arrow[dl, "\mathsf{cod}"] \\ & \mathcal{C} & \end{tikzcd} \xrightarrow{\text{language}} \text{CCTT}$$

Semantics does not use all the features of
$$\begin{tikzcd} \mathcal{T} \arrow[rr, "\chi"] \arrow[dr, "p"'] & & \mathcal{C}^{\rightarrow} \arrow[dl, "\mathsf{cod}"] \\ & \mathcal{C} & \end{tikzcd}$$

Two options:

1. Restrict the comprehension categories to 'simple' ones (fully faithful or discrete)
2. **Make the type theory more expressive: CCTT**

Why not Restrict the Models

- Are there interesting examples we would miss?
- Are there interesting features that we would lose?

More on this after some preliminaries

Outline

1. Review: Comprehension Categories
2. Back to Our Motivation
3. Our Work: Core Syntax CCTT
4. CCTT Captures Subtyping
5. Extending CCTT with Type Formers
6. Conclusion and Current Work

Outline

Comprehension Category [Jac93]

1. a category $\mathcal{C}$,
2. a (cloven) fibration $p : \mathcal{T} \to \mathcal{C}$,
3. a functor $\chi : \mathcal{T} \to \mathcal{C}^{\to}$ preserving cartesian arrows,

such that the following diagram commutes:

$$\begin{tikzcd} \mathcal{T} \arrow[rr, "\chi"] \arrow[dr, "p"'] & & \mathcal{C}^{\to} \arrow[dl, "\mathrm{cod}"] \\ & \mathcal{C} & \end{tikzcd}$$

Comprehension Category [Jac93]

1. a category $\mathcal{C}$,
2. a (cloven) fibration $p : \mathcal{T} \to \mathcal{C}$,
3. a functor $\chi : \mathcal{T} \to \mathcal{C}^{\to}$ preserving cartesian arrows,

such that the following diagram commutes:

$$\begin{tikzcd} \mathcal{T} \arrow[rr, "\chi"] \arrow[dr, "p"'] & & \mathcal{C}^{\to} \arrow[dl, "\mathrm{cod}"] \\ & \mathcal{C} & \end{tikzcd}$$

A comprehension category is

full if χ is full and faithful;
split if p is a split fibration.

Interpreting MLTT in a Comprehension Category

$$\begin{tikzcd}
\mathcal{T} \arrow[rr, "\chi"] \arrow[dr, "p"'] & & \mathcal{C}^{\rightarrow} \arrow[dl, "\mathsf{cod}"] \\
& \mathcal{C} &
\end{tikzcd}$$

Category $\mathcal{C}$ models contexts

Fibre $\mathcal{T}_\Gamma$ models types in context Γ

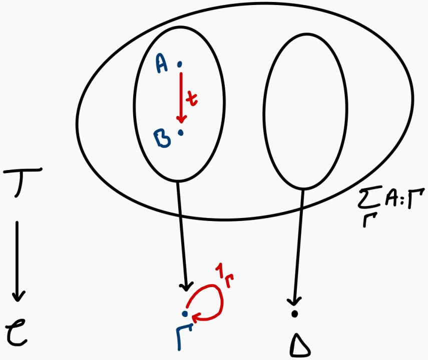
Reindexing models substitution

Comprehension χ models context extension $(\Gamma, A) \mapsto (\Gamma.A \xrightarrow{\chi(A)} \Gamma)$

Sections of $\chi(A)$ model terms $\Gamma \vdash t : A$

Vertical Morphisms

What about **morphisms in a fibre** $\mathcal{T}_\Gamma$?



Outline

Back to Our Motivation

Before we wondered:

- Are there interesting examples we would miss?
- Are there interesting features that we would lose?

Syntactical Features From Non-Full Comprehension Categores

$$\begin{tikzcd} \mathcal{T} \arrow[rr, "\chi"] \arrow[dr, "p"'] & & \mathcal{C}^{\to} \arrow[dl, "\mathsf{cod}"] \\ & \mathcal{C} & \end{tikzcd}$$

- A comprehension category can express both
  1. morphisms between contexts and
  2. morphisms between types.
- Interpreting MLTT does not make use of morphisms of types, hence these are often taken to be trivial ($\mathcal{T}$ discrete) or coming from $\mathcal{C}$ (χ fully faithful)
- Restriction 'kills off' this 'extra dimension' of morphisms.

Later we will see that this extra dimension captures **coercive subtyping**.

Examples of Non-Full Comprehension Categories I

- Intensional type theories are often given semantics in an algebraic weak factorisation system (AWFS)
- AWFSs[1] give rise to **non-full** comprehension categories

$$
\begin{tikzcd}
\mathsf{EM}(R) \arrow[rr, "U"] \arrow[dr] & & \mathcal{C}^{\to} \arrow[dl, "\mathsf{cod}"] \\
& \mathcal{C} &
\end{tikzcd}
$$

We capture this extra semantic structure in CCTT.

[1] with $\mathcal{C}$ that has finite limits

Examples of Non-Full Comprehension Categories II

Example (Groupoids)

$$\begin{tikzcd} \mathsf{SplitFib} \arrow[rr, "U"] \arrow[dr, "\mathsf{cod}"'] & & \mathsf{Grp}^{\to} \arrow[dl, "\mathsf{cod}"] \\ & \mathsf{Grp} & \end{tikzcd}$$

In [Naj+26] ...

1. We design rules of a type theory that reflect the structure of comprehension categories: CCTT
2. We show how some rules of CCTT can be seen as rules for coercive subtyping, extending work by Coraglia and Emmenegger [CE24]
3. Extend CCTT with Π-, Σ- and Id-types and their compatibility with subtyping

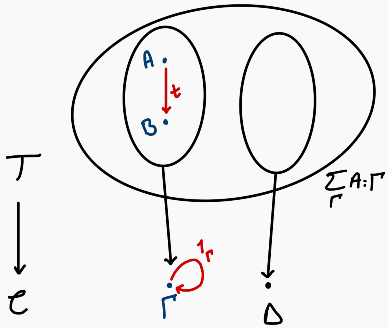
Outline

CCTT: Judgements

1. Γ ctx
2. $\Gamma \vdash s : \Delta$
3. $\Gamma \vdash s \equiv s' : \Delta$
4. $\Gamma \vdash A$ type
5. $\Gamma | A \vdash t : B$
6. $\Gamma | A \vdash t \equiv t' : B$

CCTT: Judgements

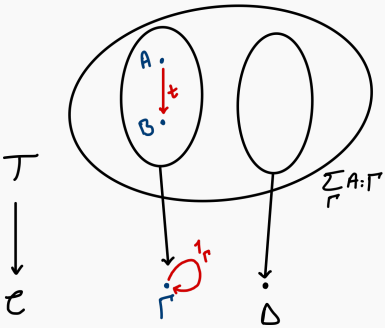
1. Γ ctx
2. $\Gamma \vdash s : \Delta$
3. $\Gamma \vdash s \equiv s' : \Delta$
4. $\Gamma \vdash A$ type
5. **$\Gamma | A \vdash t : B$**
6. $\Gamma | A \vdash t \equiv t' : B$



- Judgement 5: a morphism $[\![t]\!] : [\![A]\!] \to [\![B]\!]$ in the fibre $\mathcal{T}_{[\![\Gamma]\!]}$

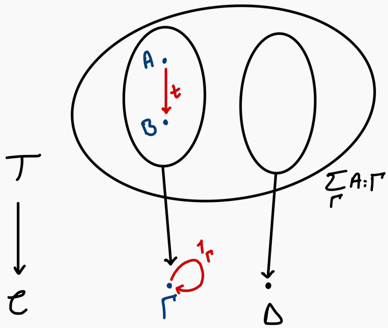
CCTT: Judgements

1. Γ ctx
2. $\Gamma \vdash s : \Delta$
3. $\Gamma \vdash s \equiv s' : \Delta$
4. $\Gamma \vdash A$ type
5. $\boldsymbol{\Gamma | A \vdash t : B}$
6. $\Gamma | A \vdash t \equiv t' : B$



- Judgement 5: a morphism $[\![t]\!] : [\![A]\!] \to [\![B]\!]$ in the fibre $\mathcal{T}_{[\![\Gamma]\!]}$
- Term judgement $\Gamma \vdash a : A$ as a macro for $\Gamma \vdash a : \Gamma.A$ and $\Gamma \vdash \pi_A \circ a \equiv 1$

1. Γ ctx
2. $\Gamma \vdash s : \Delta$
3. $\Gamma \vdash s \equiv s' : \Delta$
4. $\Gamma \vdash A$ type
5. $\Gamma|A \vdash t : B$
6. $\Gamma|A \vdash t \equiv t' : B$



- Judgement 5: a morphism $[\![t]\!] : [\![A]\!] \to [\![B]\!]$ in the fibre $\mathcal{T}_{[\![\Gamma]\!]}$
- Term judgement $\Gamma \vdash a : A$ as a macro for $\Gamma \vdash a : \Gamma.A$ and $\Gamma \vdash \pi_A \circ a \equiv 1$
- Judgement $\Gamma \mid A \overset{\sim}{\vdash} i : B$ is also a macro

CCTT: Structural Rules

Structural rules regarding the category of contexts:

$$\frac{\Gamma \ \mathsf{ctx}}{\Gamma \vdash 1_\Gamma : \Gamma}\ \textsf{ctx-mor-id}$$

$$\frac{\Gamma \vdash s : \Delta \qquad \Delta \vdash s' : \Theta}{\Gamma \vdash s' \circ s : \Theta}\ \textsf{ctx-mor-comp}$$

$$\frac{\Gamma \vdash s : \Delta}{\begin{array}{c}\Gamma \vdash s \circ 1_\Gamma \equiv s : \Delta \\ \Gamma \vdash 1_\Delta \circ s \equiv s : \Delta\end{array}}\ \textsf{ctx-id-unit}$$

$$\frac{\Gamma \vdash s : \Delta \qquad \Delta \vdash s' : \Theta \qquad \Theta \vdash s'' : \Phi}{\Gamma \vdash s'' \circ (s' \circ s) \equiv (s'' \circ s') \circ s : \Phi}\ \textsf{ctx-comp-assoc}$$

We have similar rules for the category of types.

CCTT: Structural Rules

See the paper for the rest of the structural rules: substitution, context extension, etc

Theorem (Soundness)

Every comprehension category models the rules of CCTT.

Next, we discuss some of the rules through the lens of subtyping.

Outline

1. Review: Comprehension Categories
2. Back to Our Motivation
3. Our Work: Core Syntax CCTT
4. **CCTT Captures Subtyping**
5. Extending CCTT with Type Formers
6. Conclusion and Current Work

Subtyping in CCTT

Coraglia and Emmenegger [CE24] observe that the vertical morphisms can be thought of as **witnesses for coercive subtyping**.

$$\Gamma | A \vdash t : B \qquad \rightsquigarrow \qquad \Gamma \vdash A \leq_t B$$

Also observed by: Adjedj, Lennon-Bertrand, Benjamin and Maillard [Adj+26].

Subtyping: Subsumption

Theorem (Subsumption)

From the rules of CCTT, we can derive the following rule.

$$\frac{\Gamma \vdash A, B \text{ type} \quad \Gamma \vdash A \leq_t B \quad \Gamma \vdash a : A}{\Gamma \vdash \Gamma.t \circ a : B}$$

$$\begin{tikzcd}
\Gamma.A \arrow[rr, "\Gamma.t"] & & \Gamma.B \\
& \Gamma \arrow[ul, "a"] \arrow[ur, "b"'] &
\end{tikzcd}$$

Γ.t is like a **coercion function for $A \leq_t B$.**

Subtyping: Subsumption

Theorem (Subsumption)

From the rules of CCTT, we can derive the following rule.

$$\frac{\Gamma \vdash A, B \text{ type} \quad \Gamma \vdash A \leq_t B \quad \Gamma \vdash a : A}{\Gamma \vdash \Gamma.t \circ a : B}$$

$$\begin{tikzcd}
\Gamma.A \arrow[rr, "\Gamma.t"] & & \Gamma.B \\
& \Gamma \arrow[ul, "a"] \arrow[ur, "b"'] &
\end{tikzcd}$$

Γ.t is like a **coercion function for $A \leq_t B$.**

We also have rules for weakening and substitution of the coercion witnesses.

Outline

Subtyping for Type Formers

Type Formers Are Functors

- Type formers make new types from old

$\rightsquigarrow$ Type formers should also act on **morphisms of types**

Steps Towards Type Formers

1. Extend CCTT with a type former (e.g. Σ-types) and show soundness.
2. Extend CCTT with subtyping for the type former and show soundness

We look at subtyping for Σ-types to keep things simple.

Rules for Subtyping for Σ-types

1. We want to have the following rule:

$$\frac{\Gamma \vdash A, A' \text{ type} \quad \Gamma.A \vdash B \text{ type} \quad \Gamma.A' \vdash B' \text{ type} \\ \Gamma \vdash A \leq_f A' \quad \Gamma.A \vdash B \leq_g B'[\Gamma.f]}{\Gamma \vdash \Sigma_A B \leq_{\Sigma(f,g)} \Sigma_{A'} B'}$$

   Σ acts covariantly on both arguments.

Rules for Subtyping for Σ-types

1. We want to have the following rule:

$$\frac{\Gamma \vdash A, A' \text{ type} \quad \Gamma.A \vdash B \text{ type} \quad \Gamma.A' \vdash B' \text{ type} \qquad \Gamma \vdash A \leq_f A' \quad \Gamma.A \vdash B \leq_g B'[\Gamma.f]}{\Gamma \vdash \Sigma_A B \leq_{\Sigma(f,g)} \Sigma_{A'} B'}$$

   Σ acts covariantly on both arguments.

2. The coercion function for $\Sigma_A B \leq_{\Sigma(f,g)} \Sigma_{A'} B'$ should act as follows:

$$\Gamma.\Sigma_A B \xrightarrow{\mathsf{proj}_{\Sigma_A B}} \Gamma.A.B \xrightarrow{\chi_0 g} \Gamma.A.B'[\chi_0 f] \xrightarrow{\chi_0 f.B'} \Gamma.A'.B' \xrightarrow{\mathsf{pair}_{\Sigma_{A'} B'}} \Gamma.\Sigma_{A'} B'$$

Rules for Subtyping for Σ-types

1. We want to have the following rule:

$$\frac{\Gamma \vdash A, A' \text{ type} \quad \Gamma.A \vdash B \text{ type} \quad \Gamma.A' \vdash B' \text{ type} \\ \Gamma \vdash A \leq_f A' \quad \Gamma.A \vdash B \leq_g B'[\Gamma.f]}{\Gamma \vdash \Sigma_A B \leq_{\Sigma(f,g)} \Sigma_{A'} B'}$$

   Σ acts covariantly on both arguments.

2. The coercion function for $\Sigma_A B \leq_{\Sigma(f,g)} \Sigma_{A'} B'$ should act as follows:

$$\Gamma.\Sigma_A B \xrightarrow{\mathsf{proj}_{\Sigma_A B}} \Gamma.A.B \xrightarrow{\chi_0 g} \Gamma.A.B'[\chi_0 f] \xrightarrow{\chi_0 f.B'} \Gamma.A'.B' \xrightarrow{\mathsf{pair}_{\Sigma_{A'} B'}} \Gamma.\Sigma_{A'} B'$$

3. Rules for functoriality for $\Sigma(-,-)$

Semantic Structure for Subtyping for Σ–Types

Definition

$(\mathcal{C}, \mathcal{T}, p, \chi)$ **has subtyping for Σ-types** if it has

1. dependent sums and
2. for each $f : A \to A'$ in $\mathcal{T}_\Gamma$ and $g : B \to B'[\chi_0 f]$ in $\mathcal{T}_{\Gamma.A}$, a morphism in $\mathcal{T}_\Gamma$
$$\Sigma_f g : \Sigma_A B \to \Sigma_{A'} B'$$
3. $\chi_0(\Sigma_f g)$ is the following composite
$$\Gamma.\Sigma_A B \xrightarrow{\mathsf{proj}_{\Sigma_A B}} \Gamma.A.B \xrightarrow{\chi_0 g} \Gamma.A'.B'[\chi_0 f] \xrightarrow{\chi_0 f.B'} \Gamma.A'.B' \xrightarrow{\mathsf{pair}_{\Sigma_{A'} B'}} \Gamma.\Sigma_{A'} B'$$
4. $\Sigma_{(-)}(-)$ preserves identities and composition

Interpretation and Sanity Check

Theorem

Any comprehension category with subtyping for Σ-types models CCTT extended with subtyping for Σ-types.

Sanity Check

When χ is fully faithful,

- our Σ-structures are equivalent to Lumsdaine and Warren's [LW15]
- Jacobs' structure for Σ–types [Jac93] gives rise to ours

Outline

1. Review: Comprehension Categories
2. Back to Our Motivation
3. Our Work: Core Syntax CCTT
4. CCTT Captures Subtyping
5. Extending CCTT with Type Formers
6. **Conclusion and Current Work**

Conclusion and Current Work

“From Semantics to Syntax: A Type Theory For Comprehension Categories”

- CCTT reflects the structure of a comprehension category.
- Gain back the ‘extra dimension’ of type morphisms; captures coercive subtyping.

Current Work

- AWFSs give rise to comprehension categories
- We aim to characterise those comprehension categories (with type constructions) that arise from AWFSs

Link between AWFSs and Comprehension Categories

Our Goal

- Characterise comprehension categories that arise from AWFSs
- Type theory for AWFS; reflect interesting features of AWFS to syntax

Intermediate Goals

- Understand better the link between comprehension categories and AWFSs
- An equivalence of categories:

$$\mathsf{AWFS}^{\mathsf{lax}} + \text{some structure} \xleftrightarrow{\simeq} \mathsf{CompCat}^{\mathsf{strict}} + \text{some structure}$$

Thank you for your attention!

References

[Adj+26] Arthur Adjedj et al. **"AdapTT: Functoriality for Dependent Type Casts".** In: *Proc. ACM Program. Lang.* 10.POPL (2026), pp. 628–658. DOI: 10.1145/3776664. URL: https://doi.org/10.1145/3776664.

[ALN24] Benedikt Ahrens, Peter LeFanu Lumsdaine, and Paige Randall North. **"Comparing Semantic Frameworks for Dependently-Sorted Algebraic Theories".** In: *Programming Languages and Systems: 22nd Asian Symposium, APLAS 2024, Kyoto, Japan, October 22-24, 2024, Proceedings*. Kyoto, Japan: Springer-Verlag, 2024, pp. 3–22. ISBN: 978-981-97-8942-9. DOI: 10.1007/978-981-97-8943-6_1. URL: https://doi.org/10.1007/978-981-97-8943-6_1.

[CE24] Greta Coraglia and Jacopo Emmenegger. **"Categorical Models of Subtyping".** In: *29th International Conference on Types for Proofs and Programs (TYPES 2023)*. Ed. by Delia Kesner, Eduardo Hermo Reyes, and Benno van den Berg. Vol. 303. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024, 3:1–3:19. ISBN: 978-3-95977-332-4. DOI: 10.4230/LIPIcs.TYPES.2023.3. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.TYPES.2023.3.

[Jac93] Bart Jacobs. **"Comprehension Categories and the Semantics of Type Dependency".** In: *Theor. Comput. Sci.* 107.2 (1993), pp. 169–207. DOI: 10.1016/0304-3975(93)90169-T. URL: https://doi.org/10.1016/0304-3975(93)90169-T.

[LW15] Peter Lefanu Lumsdaine and Michael A. Warren. **"The Local Universes Model: An Overlooked Coherence Construction for Dependent Type Theories".** In: *ACM Transactions on Computational Logic* 16.3 (July 2015), pp. 1–31. ISSN: 1557-945X. DOI: 10.1145/2754931. URL: http://dx.doi.org/10.1145/2754931.

[Naj+26] Niyousha Najmaei et al. **"From Semantics to Syntax: A Type Theory for Comprehension Categories".** In: *Proc. ACM Program. Lang.* 10.POPL (Jan. 2026). DOI: 10.1145/3776725. URL: https://doi.org/10.1145/3776725.