A Type Theory for Comprehension Categories

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

CHoCoLa Seminar — 30 April 2026

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:

$$\text{MLTT} \xrightarrow{\text{semantics}} \begin{array}{ccc} \mathcal{T} & \xrightarrow{\chi} & \mathcal{C}^{\to} \\ & {\scriptstyle p}\searrow \quad \swarrow{\scriptstyle \text{cod}} & \\ & \mathcal{C} & \end{array}$$

Semantics does not use all the features of

$$\begin{array}{ccc} \mathcal{T} & \xrightarrow{\chi} & \mathcal{C}^{\to} \\ & {\scriptstyle p}\searrow \quad \swarrow{\scriptstyle \text{cod}} & \\ & \mathcal{C} & \end{array}$$

Motivation

Interpretation of MLTT in comprehension categories:

$$\text{MLTT} \xrightarrow{\text{semantics}} \begin{array}{ccc} \mathcal{T} & \xrightarrow{\chi} & \mathcal{C}^{\to} \\ & {\scriptstyle p}\searrow \quad \swarrow{\scriptstyle \mathrm{cod}} & \\ & \mathcal{C} & \end{array}$$

Semantics does not use all the features of $\begin{array}{ccc} \mathcal{T} & \xrightarrow{\chi} & \mathcal{C}^{\to} \\ & {\scriptstyle p}\searrow \quad \swarrow{\scriptstyle \mathrm{cod}} & \\ & \mathcal{C} & \end{array}$ Two options:

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

Motivation

Interpretation of MLTT in comprehension categories:

$$\mathrm{MLTT} \xrightarrow{\text{semantics}} \begin{array}{ccc} \mathcal{T} & \xrightarrow{\chi} & \mathcal{C}^{\to} \\ {\scriptstyle p}\searrow & & \swarrow{\scriptstyle \mathrm{cod}} \\ & \mathcal{C} & \end{array} \xrightarrow{\text{language}} \mathrm{CCTT}$$

Semantics does not use all the features of $\begin{array}{ccc} \mathcal{T} & \xrightarrow{\chi} & \mathcal{C}^{\to} \\ {\scriptstyle p}\searrow & & \swarrow{\scriptstyle \mathrm{cod}} \\ & \mathcal{C} & \end{array}$ 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

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

Fibrations

Definition

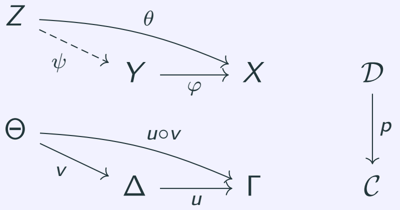
A functor $p : \mathcal{D} \to \mathcal{C}$ is a (cloven Grothendieck) **fibration** if for every $u : \Delta \to \Gamma$ in $\mathcal{C}$ and X over Γ, we have a chosen **"cartesian"** morphism $\varphi : Y \to X$ with $p(\varphi) = u$.

$$
\begin{array}{ccccc}
Y & \xrightarrow[\varphi]{} & X & \qquad & \mathcal{D} \\
 & & & & \big\downarrow p \\
\Delta & \xrightarrow[u]{} & \Gamma & \qquad & \mathcal{C}
\end{array}
$$

Fibrations

Definition

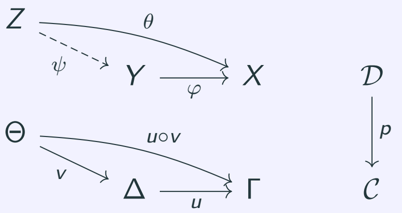
A functor $p : \mathcal{D} \to \mathcal{C}$ is a (cloven Grothendieck) **fibration** if for every $u : \Delta \to \Gamma$ in $\mathcal{C}$ and X over Γ, we have a chosen **"cartesian"** morphism $\varphi : Y \to X$ with $p(\varphi) = u$.



Fibrations

Definition

A functor $p : \mathcal{D} \to \mathcal{C}$ is a (cloven Grothendieck) **fibration** if for every $u : \Delta \to \Gamma$ in $\mathcal{C}$ and X over Γ, we have a chosen **"cartesian"** morphism $\varphi : Y \to X$ with $p(\varphi) = u$.



Fiber $\mathcal{D}_\Gamma$ subcategory over Γ and Id_Γ

Split
1. chosen lifts of identitites are identities
2. chosen lift of any composite is the composite of the individual lifts

Examples of Fibrations I

Example

The forgetful functor $\mathrm{dom} : \mathcal{C}^{\rightarrow} \rightarrow \mathcal{C}$.

$$\begin{array}{ccc} f : A \rightarrow B & & \mathcal{C}^{\rightarrow} \\ & & \big\downarrow \mathrm{dom} \\ A & & \mathcal{C} \end{array}$$

Examples of Fibrations II

Example

The forgetful functor $\mathrm{cod} : \mathcal{C}^{\rightarrow} \rightarrow \mathcal{C}$, if and only if $\mathcal{C}$ has chosen pullbacks.

$$
\begin{array}{ccc}
f : A \rightarrow B & & \mathcal{C}^{\rightarrow} \\
 & & \downarrow \mathrm{cod} \\
B & & \mathcal{C}
\end{array}
$$

Reindexing

Given a fibration $p : \mathcal{D} \to \mathcal{C}$ and $u : \Delta \to \Gamma$ in $\mathcal{C}$, we denote by u^* the **reindexing** functor

$$u^* : \mathcal{D}_\Gamma \to \mathcal{D}_\Delta$$

mapping $X \in \mathcal{D}_\Gamma$ to the following Y:

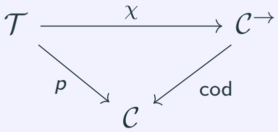
$$\begin{array}{ccccc} Y & \xrightarrow[\varphi]{} & X & \qquad & \mathcal{D} \\ & & & & \Big\downarrow{\scriptstyle p} \\ \Delta & \xrightarrow[u]{} & \Gamma & \qquad & \mathcal{C} \end{array}$$

Comprehension Categories

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,

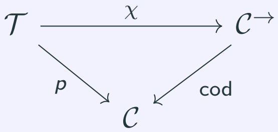
such that the following diagram commutes:



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:



A comprehension category is

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

Interpreting MLTT in a Comprehension Category

$$
\begin{array}{ccc}
\mathcal{T} & \xrightarrow{\chi} & \mathcal{C}^{\to} \\
 & {\scriptstyle p}\searrow \quad \swarrow{\scriptstyle \mathsf{cod}} & \\
 & \mathcal{C} &
\end{array}
$$

Category $\mathcal{C}$ models contexts

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

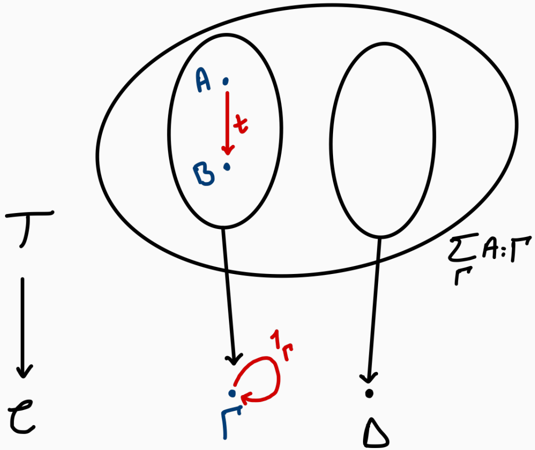
Reindexing models substitution

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

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

Back to Our Motivation

$$\begin{array}{ccc} \mathcal{T} & \xrightarrow{\quad \chi \quad} & \mathcal{C}^{\to} \\ & {\scriptstyle p}\searrow \qquad \swarrow{\scriptstyle \mathsf{cod}} & \\ & \mathcal{C} & \end{array}$$

- 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

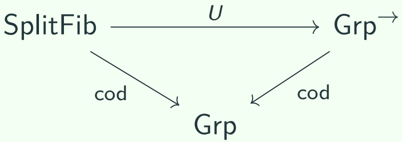
- Intensional type theories are often given semantics in an algebraic weak factorisation system (AWFS)
- AWFSs give rise to **non-full** comprehension categories

$$\begin{array}{ccc} \mathrm{EM}(R) & \xrightarrow{\;U\;} & \mathcal{C}^{\rightarrow} \\ & \searrow \quad \swarrow_{\mathrm{cod}} & \\ & \mathcal{C} & \end{array}$$

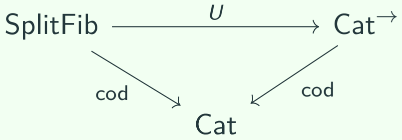
We capture this extra semantic structure in CCTT.

Examples of Non-Full Comprehension Categories II

Example (Groupoids)



Example (Categories)



Examples of Non-Full Comprehension Categories III

Example (Topos $\mathcal{E}$)

$$\begin{array}{ccc} \mathcal{E}/\Omega & \xrightarrow{\text{subobject}} & \mathcal{E}^{\to} \\ & \searrow \quad \swarrow_{\text{cod}} & \\ & \mathcal{E} & \end{array}$$

Example (Heyting algebra H)

$$\begin{array}{ccc} \mathsf{Set}/H & \xrightarrow{\text{subobject}} & \mathsf{Set}^{\to} \\ & \searrow \quad \swarrow_{\text{cod}} & \\ & \mathsf{Set} & \end{array}$$

In Our Work...

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

Based on *From Semantics to Syntax: A Type Theory for Comprehension Categories*

Outline

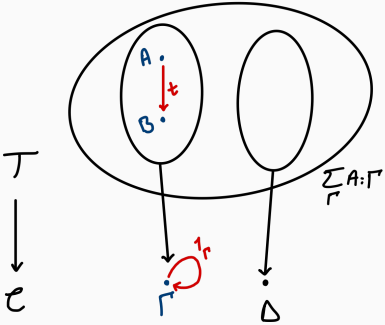
1. Review: Comprehension Categories
2. Back to Our Motivation
3. Our Work: Core Syntax CCTT
4. CCTT Captures Subtyping
5. Extending CCTT with Type Formers

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. $\Gamma | A \vdash t \equiv t' : B$

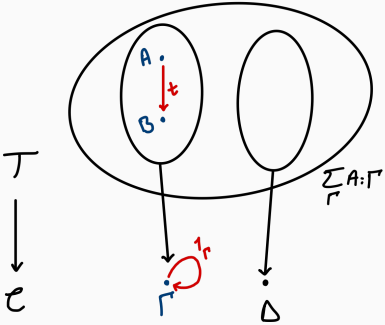
CCTT: Judgements

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



CCTT: Judgements

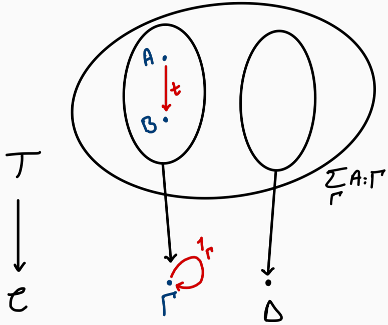
1. Γ ctx
2. $\Gamma \vdash s : \Delta$
3. $\Gamma \vdash s \equiv s' : \Delta$
4. $\Gamma \vdash A$ type
5. **$\Gamma | A \vdash t : B$**
6. $\Gamma | A \vdash t \equiv t' : B$



- 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. $\Gamma | A \vdash t : B$
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 $\Gamma \vdash \pi_a \circ a \equiv 1$

CCTT: Structural Rules

Structural rules regarding the category of contexts:

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

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

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

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

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

Subtyping in CCTT

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

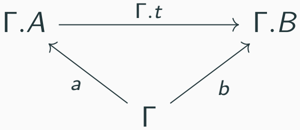
$$\Gamma | A \vdash t : B \quad \rightsquigarrow \quad \Gamma \vdash A \leq_t B$$

Subtyping: Subsumption

Theorem (Subsumption)

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

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



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

Outline

Subtyping for Type Formers

Type Formers Are Functors

- Type formers make new types from old
- Also have term formers; here: context morphism formers

$\rightsquigarrow$ 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 Σ-types to keep things simple.

Rules for Σ-types

Extend CCTT with Σ-types, e.g.:

$$\frac{\Gamma \vdash A \text{ type} \qquad \Gamma.A \vdash B \text{ type}}{\Gamma \vdash \Sigma_A B \text{ type}} \text{ sigma-form}$$

$$\frac{\Gamma \vdash A \text{ type} \qquad \Gamma.A \vdash B \text{ type}}{\Gamma.A.B \vdash \mathsf{pair}_{\Sigma_A B} : \Gamma.\Sigma_A B} \text{ sigma-intro}$$

$$\frac{\Gamma \vdash A \text{ type} \qquad \Gamma.A \vdash B \text{ type}}{\Gamma.\Sigma_A B \vdash \mathsf{proj}_{\Sigma_A B} : \Gamma.A.B} \text{ sigma-elim}$$

$$\frac{\Gamma \vdash A \text{ type} \qquad \Gamma.A \vdash B \text{ type}}{\begin{array}{c}\Gamma.A.B \vdash \mathsf{proj}_{\Sigma_A B} \circ \mathsf{pair}_{\Sigma_A B} \equiv 1_{\Gamma.A.B} : \Gamma.A.B \\ \Gamma.\Sigma_A B \vdash \mathsf{pair}_{\Sigma_A B} \circ \mathsf{proj}_{\Sigma_A B} \equiv 1_{\Gamma.\Sigma_A B} : \Gamma.\Sigma_A B\end{array}} \text{ sigma-beta-eta}$$

$$\frac{\Delta \vdash A \text{ type} \qquad \Delta.A \vdash B \text{ type} \qquad \Gamma \vdash s : \Delta}{\Gamma \mid \Sigma_{A[s]} B[s.A] \overset{\sim}{\vdash} i_{\Sigma_A B, s} : (\Sigma_A B)[s]} \text{ subst-sigma}$$

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:

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

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

Conclusion

Summary

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

Current: Link between AWFSs and Comprehension Categories

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

$\rightsquigarrow$ Hope to understand better the role of AWFSs when building (constructive) models of univalence (e.g., [BF22; Swa18; Awo18])

Thank you for your attention!

References

[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.

[Awo18] Steve Awodey. **"A cubical model of homotopy type theory".** In: *Annals of Pure and Applied Logic* 169.12 (2018), pp. 1270–1294. ISSN: 0168-0072. DOI: https://doi.org/10.1016/j.apal.2018.08.002. URL: https://www.sciencedirect.com/science/article/pii/S0168007218300861.

[BF22] Benno van den Berg and Eric Faber. ***Effective Kan fibrations in simplicial sets.*** Lecture Notes in Mathematics. Springer Cham, 2022. DOI: https://doi.org/10.1007/978-3-031-18900-5.

[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.

[Swa18] Andrew Swan. **"Identity types in algebraic model structures and cubical sets".** In: *arXiv:1808.00915* (2018).