A Type Theory for Comprehension Categories

Niyousha Najmaei[1], Niels van der Weide[2], Benedikt Ahrens[3], Paige Randall North[4]

[1] École Polytechnique, Palaiseau, France
[2] Radboud University Nijmegen, The Netherlands
[3] Delft University of Technology, The Netherlands
[4] Utrecht University, The Netherlands

Recent models of intensional type theory have been constructed in algebraic weak factorization systems (AWFSs). AWFSs give rise to comprehension categories that feature non-trivial morphisms between types; these morphisms are not used in the standard interpretation of Martin-Löf type theory in comprehension categories.

We develop a type theory that internalizes morphisms between types, reflecting this semantic feature back into syntax. Our type theory comes with Π-, Σ-, and identity types. We discuss how it can be viewed as an extension of Martin-Löf type theory with coercive subtyping, as sketched by Coraglia and Emmenegger [5], and we recognize this structure in many intensional models of Martin-Löf type theory.

This talk is based on a recently published paper [11] and its ongoing continuation.

Interpretation of Type Theories in Comprehension Categories

We sketch the interpretation of type theories in categorical structures, contrasting that of MLTT with that of other type theories.

Interpretation of MLTT Martin-Löf type theory (MLTT), and variations thereof, are typically given semantics in categories with structure, such as categories with families [7], contextual categories [2, 3], and type categories [12]. All of these categorical structures can be viewed as a *full (or discrete) comprehension category* [1].

A comprehension category consists (among other things) of two categories: one whose objects interpret the contexts of the type theory and one which interprets its types. Thus, at first glance, a comprehension category can express both morphisms between contexts and morphisms between types. In MLTT, however, there is only one such notion in the sense that the morphisms between both contexts and types are generated by the terms of the theory. Hence, morphisms between types can in turn be recovered from the context morphisms. The requirements that a comprehension category is discrete or that it is full then represent the two universal ways of deriving the type morphisms from the context morphisms: discreteness assumes that there are as few type morphisms as possible, whereas fullness assumes there are as many type morphisms as possible. Thus, both the requirements of discreteness and fullness are used to 'kill off' this 'extra dimension' of morphisms.

Interpretation of Intensional Type Theories To give semantics to intensional type theories, such as homotopy type theory [13] and cubical type theories [4], one generally uses more refined ideas, in particular from higher category theory and homotopy theory. Specifically, such type theories are often given semantics in an *algebraic weak factorization system* (or *AWFS* for short) [8, 9]. These AWFSs are usually interesting in their own right, and they endow the category with higher dimensional structure that we can understand synthetically via the identity type. Just like other categorical structures such as categories with families, AWFSs give rise to comprehension categories. However, these comprehension categories are typically neither full nor discrete, nor are they split. In other words, these comprehension categories have morphisms between types that do *not* arise from context morphisms, and their substitution structure is more intricate than that of Martin-Löf type theory itself. We argue that this extra structure in the semantics should not be ignored, and it is captured by the type theory presented here.

Reflecting Semantic Features Into the Syntax

We develop a type theory that allows for synthetic reasoning about the comprehension categories arising from AWFSs— thus, in particular, about morphisms of types and about non-split substitution. To this end, we develop a new type theory, which we call *comprehension category type theory* or CCTT for short. This type theory is obtained by "reflecting" some semantic features back into traditional type theory.

We are not the first to develop type theory by reflecting semantic features back into the syntax. For instance, a type theory for non-split substitution has been developed by [6], who developed a type theory with explicit substitutions to reflect the non-split substitution structure of comprehension categories into type theory. Similarly, [5] studied generalized categories with families for the semantics of type theory; they discovered that, syntactically, such generalized structures give rise to a notion of coercive subtyping. In our work, we encounter both of these syntactic features, and, in particular, give a systematic and extensive account of the syntax and semantics of subtyping sketched by [5].

Specifically, by not imposing the restrictions of discreteness or fullness, our syntax and semantics can capture coercive subtyping [5]. As argued there, every comprehension category is equipped with a notion of subtyping, which is given by morphisms between types. Fullness expresses that coercions and terms are the same, making the subtyping, in one sense, trivial. Discreteness expresses that all coercions arise from identities of types, making the subtyping, in another sense, trivial. To faithfully model coercive subtyping, one thus need to use comprehension categories that are neither full nor discrete.

Moreover, from a practical point of view, having both terms and type morphisms in a type theory allows for tight control over definitional equalities. Specifically, when considering semantics of type theory in AWFSs, types and type morphisms are interpreted as algebras for some monad, where the algebra structure models the transport, $(a = a') \to B(a) \to B(a')$, of structure along a term of an identity type, and morphisms of algebras preserve such transport *strictly*, up to definitional equality. Semantics in AWFSs justify adding syntactic rules expressing such definitional equalities, which only hold up to *propositional* identity in MLTT.

We develop a type theory that we show is an internal language for comprehension categories. Usually, the semantics of Martin-Löf type theory (MLTT) is given in discrete or full comprehension categories. Requiring a comprehension category to be full or discrete can be understood as removing one 'dimension' of morphisms. In our syntax, we recover this extra dimension. We show that this extra dimension can be used naturally to encode subtyping as sketched by Coraglia and Emmenegger [5]. We then extend our type theory with Π-, Σ- and Id-types and discuss how to add subtyping for these type formers to both the syntax and semantics.

In [10], we studied non-full comprehension categories, partly motivated by the fact that they arise from AWFSs. In our ongoing work, we study more closely the (non-full) comprehension categories that arise from AWFSs. In this talk, we will discuss the type theory developed in [10] and the ongoing work on the relationship between comprehension categories and AWFSs.

References

[1] Benedikt Ahrens, Peter LeFanu Lumsdaine, and Paige Randall North. Comparing semantic frameworks for dependently-sorted algebraic theories. In Oleg Kiselyov, editor, *Programming Languages and Systems - 22nd Asian Symposium, APLAS 2024, Kyoto, Japan, October 22-24, 2024, Proceedings*, volume 15194 of *Lecture Notes in Computer Science*, pages 3–22. Springer, 2024.

[2] John Cartmell. *Generalised Algebraic Theories and Contextual Categories*. PhD thesis, Oxford University, UK, 1978.

[3] John Cartmell. Generalised algebraic theories and contextual categories. *Ann. Pure Appl. Log.*, 32:209–243, 1986.

[4] Cyril Cohen, Thierry Coquand, Simon Huber, and Anders Mörtberg. Cubical Type Theory: A Constructive Interpretation of the Univalence Axiom. In Tarmo Uustalu, editor, *21st International Conference on Types for Proofs and Programs (TYPES 2015)*, volume 69 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 5:1–5:34, Dagstuhl, Germany, 2018. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[5] Greta Coraglia and Jacopo Emmenegger. Categorical Models of Subtyping. In Delia Kesner, Eduardo Hermo Reyes, and Benno van den Berg, editors, *29th International Conference on Types for Proofs and Programs (TYPES 2023)*, volume 303 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 3:1–3:19, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[6] Pierre-Louis Curien, Richard Garner, and Martin Hofmann. Revisiting the categorical interpretation of dependent type theory. *Theor. Comput. Sci.*, 546:99–119, 2014.

[7] Peter Dybjer. Internal type theory. In Stefano Berardi and Mario Coppo, editors, *Types for Proofs and Programs, International Workshop TYPES'95, Torino, Italy, June 5-8, 1995, Selected Papers*, volume 1158 of *Lecture Notes in Computer Science*, pages 120–134. Springer, 1996.

[8] Nicola Gambino and Marco Federico Larrea. Models of Martin-Löf type theory from algebraic weak factorisation systems. *The Journal of Symbolic Logic*, 88(1):242–289, 2023.

[9] Martin Hofmann and Thomas Streicher. The groupoid interpretation of type theory. In *Twenty-Five Years of Constructive Type Theory (Venice, 1995)*, volume 36 of *Oxford Logic Guides*, pages 83–111. Oxford Univ. Press, New York, 1998.

[10] Niyousha Najmaei, Niels van der Weide, Benedikt Ahrens, and Paige Randall North. From semantics to syntax: A type theory for comprehension categories, 2025.

[11] Niyousha Najmaei, Niels van der Weide, Benedikt Ahrens, and Paige Randall North. From semantics to syntax: A type theory for comprehension categories. *Proc. ACM Program. Lang.*, 10(POPL):2409–2438, 2026.

[12] Andrew M. Pitts. Categorical logic. In *Handbook of Logic in Computer Science, Vol. 5*, volume 5 of *Handb. Log. Comput. Sci.*, pages 39–128. Oxford Univ. Press, New York, 2000.

[13] The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics*. https://homotopytypetheory.org/book, Institute for Advanced Study, 2013.