# Models for Axiomatic Type Theory

Daniël Otten, University of Amsterdam
Matteo Spadetto, University of Leeds

**Overview.** We compare two semantics for type theory: *comprehension categories* [Jac93], which closely follow the syntax and intricacies of type theory, and *path categories* [vdBM18], which are relatively simple structures that take inspiration from homotopy theory. Both are only semantics in a weak way because they only specify substitutions up to isomorphism. However, it is known that we can turn comprehension categories into actual models by 'splitting' [LW15, Boc22]. We show that this can also be done for path categories by proving an equivalence between path categories and certain comprehension categories.

Path categories provide semantics for a minimal dependent type theory: one with only =-types, and moreover, where the $\beta$-rule is only an *axiom* (a propositional equality) and not a *reduction* (a definitional equality). This is the notion that appears in Cubical Type Theory [CCHM18] as well as in Axiomatic Type Theory (type theory without reductions). To make this precise, we show that the 2-category of path categories is equivalent to the 2-category of comprehension categories that have:

- contextuality (every context must be build out of a finite number of types),
- =-types with only a propositional $\beta$-rule,
- $\Sigma$-types with a definitional $\beta$-rule and a definitional $\eta$-rule,

in a weakly stable way. Here weak stability is a technical condition and precisely what we need to obtain a (genuine) model by applying the left-adjoint splitting functor. The reason why we obtain $\Sigma$-types is that path categories make no distinction between contexts and types: $\Gamma, x : A, y : B[x]$ doubles as $\Gamma, z : \Sigma(x : A)\, B[x]$. Because this is not always desirable, we also introduce a more fine-grained notion: that of a *display path category* where we do make this distinction. We show that display path categories are equivalent to comprehension categories that just have contextuality and propositional =-types in a weakly stable way, without any assumption regarding $\Sigma$-types. We obtain the following diagram:

$$\begin{array}{ccc} \mathrm{PathCat} & \xrightarrow{\sim} & \mathrm{ComprehensionCat}_{\mathrm{Contextual},=,\Sigma_{\beta\eta}} \\ {\scriptstyle U}\uparrow \dashv \downarrow{\scriptstyle C} & & {\scriptstyle F}\uparrow \dashv \downarrow{\scriptstyle U} \\ \mathrm{DisplayPathCat} & \xrightarrow[\sim]{} & \mathrm{ComprehensionCat}_{\mathrm{Contextual},=} \end{array}$$

where the arrows denoted with $U$ are forgetful 2-functors, $C$ is a cofree 2-functor that interprets every context extension as a type, and $F$ is a free 2-functor that adds $\Sigma$-types.

**Axiomatic Type Theory.** An *Axiomatic Type Theory (ATT)* is a dependent type theory without any reduction rules. So, (display) path categories provide semantics for a minimal ATT, and we can model more extensive versions of ATT by adding more structure. Removing the reductions from a type theory makes it easier to find models and reduces the complexity of type checking (from non-elementary to quadratic [vdBdB21]). Despite these simplifications, ATT does not lose much deductive strength: by adding only two principles to ATT—binder extensionality and uniqueness of identity proofs—we can prove the same as Extensional Type Theory [Win20], which has the maximal number of reductions. For more, see [Boc23].

**Path Categories.** A path category extends Brown's notion of a category of fibrant objects [Bro73] and Joyal's notion of a tribe [Joy17] as seen in Van den Berg [vdB18]. In this way, we can view it both as a framework for homotopy theory as well as a semantics for type theory. From the first perspective: it is a setting in which we can define homotopy, build factorisation systems, and prove lifting theorems. From the second perspective: it simplifies semantics by removing explicit elimination and computation rules.

To model the basic structure of dependent type theory, we have a category $\mathcal{C}$ and a collection of morphisms called *fibrations*. We view an object as a type or a context and a fibration $A \twoheadrightarrow \Gamma$ as a type or telescope $A$ in context $\Gamma$. To model propositional =-types, we require that every object $A$ has an object $PA$ of paths in $A$. The formation rule is modelled by the source and target maps $(s, t) : PA \twoheadrightarrow A \times A$ and the introduction rule is modelled by the constant path map $r : A \to PA$. Now, instead of adding elimination and computation rules, we require that we are given a collection of morphisms called *(weak) equivalences*, and that $r$ is in this collection. These morphisms will model the equivalences of the type theory. Lastly we require that fibrations, equivalences, and path objects satisfy some simple axioms.

**Display Path Categories.** In a display path category we use *display maps* as a primitive notion instead of fibrations. Intuitively, these are the fibrations $\Delta \twoheadrightarrow \Gamma$ that only extend $\Gamma$ with a single type. Now we define the fibrations as the maps that can be build as a composition of isomorphisms and display maps. In addition, we replace the path objects for contexts $\Gamma$ with path objects for display maps $A \twoheadrightarrow \Gamma$. This appears weaker but is sufficient, as they can be used to inductively construct path objects for contexts using a lifting theorem and a notion of transport. Hence, a display path category is in particular a path category.

**Equivalences.** To interpret a path category as a comprehension category we interpret the objects as contexts and the fibrations as types. We can recover the elimination and computation rules for propositional =-types using a lifting theorem for path categories; the details can be found in Van den Berg [vdB18]. We get contextuality because every map $A \to 1$ is a fibration, and $\Sigma$-types because fibrations are closed under composition. For the other direction, we interpret compositions of display maps, and isomorphism as our fibrations and the homotopy equivalences as our weak equivalences.

On the level of display path categories we can be a bit more precise because we have additional structure. When interpreting a display path category as a comprehension category we only interpret display maps as types instead of arbitrary fibration. This also means that we retain more structure when interpreting a suitable comprehension category as a display path category, namely which fibration are display maps.

**Additional Structure.** To extend this work, we are hoping to interpret addition propositional type constructors in (display) path categories. The idea is that $\Sigma$-types and $\Pi$-types should be weakenings of existing presentations: as left and right adjoints of the pullback functor. In this way, we hope to obtain similar simplifications: requiring that certain maps are equivalences and thereby being able to ommit computation rules.

# References

[Boc22] Rafaël Bocquet. Strictification of Weakly Stable Type-Theoretic Structures Using Generic Contexts. In Henning Basold, Jesper Cockx, and Silvia Ghilezan, editors, *27th International Conference on Types for Proofs and Programs (TYPES 2021)*, volume 239 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 3:1–3:23, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[Boc23] Rafaël Bocquet. Towards coherence theorems for equational extensions of type theories, 2023.

[Bro73] Kenneth S. Brown. Abstract homotopy theory and generalized sheaf cohomology. *Transactions of the American Mathematical Society*, 186:419–458, 1973.

[CCHM18] 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.

[Jac93] Bart Jacobs. Comprehension categories and the semantics of type dependency. *Theoretical Computer Science*, 107(2):169–207, 1993.

[Joy17] Andre Joyal. Notes on clans and tribes, 2017.

[LW15] Peter Lefanu Lumsdaine and Michael A. Warren. The local universes model: An overlooked coherence construction for dependent type theories. *ACM Trans. Comput. Logic*, 16(3), jul 2015.

[vdB18] Benno van den Berg. Path categories and propositional identity types. *ACM Trans. Comput. Logic*, 19(2), jun 2018.

[vdBdB21] Benno van den Berg and Martijn den Besten. Quadratic type checking for objective type theory, 2021.

[vdBM18] Benno van den Berg and Ieke Moerdijk. Exact completion of path categories and algebraic set theory. Part I: Exact completion of path categories. *Journal of Pure and Applied Algebra*, 222(10):3137–3181, 2018.

[Win20] Théo Winterhalter. *Formalisation and meta-theory of type theory*. PhD thesis, Université de Nantes, 2020.