The Leibniz adjunction in homotopy type theory, with an application to simplicial type theory

Tom de Jong, Nicolai Kraus, and Axel Ljungström

University of Nottingham, Nottingham, UK
{`tom.dejong`, `nicolai.kraus`, `axel.ljungstrom`}`@nottingham.ac.uk`

Simplicial type theory extends homotopy type theory and equips types with a notion of directed morphisms. A *Segal type* is defined to be a type in which these directed morphisms can be composed. We show that all higher coherences can be derived. In technical terms, this means that if a type has unique fillers for Λ^2_1-horns, it has unique fillers for all inner Λ^n_k-horns (Theorem 8). This generalizes Riehl and Shulman's result [5, Proposition 5.21] for the case $n = 3$, $k \in \{1, 2\}$.

Our main technical tool (Theorem 1) is the Leibniz adjunction: the pushout-product is left adjoint to the pullback-hom in the wild category of types. While this adjunction is well known for 1-categories (e.g. [6, Lemma 4.10]), it is more involved for higher categories and we make profitable use of the fact that the wild category of maps is equivalent to that of families.

We have formalized the results in Cubical Agda; a full paper is available as arXiv:2601.21843.

The Leibniz adjunction for types A naive translation of the definition of a (1-)category into homotopy type theory results in the notion of a wild category. A *wild category* consists of a type of objects and a type family of arrows with identities and an associative composition operation for which the identities are left and right neutral, as usual.[1] This notion is given the adjective "wild" because the types of arrows are not restricted to be sets, nor do we require higher coherences on the identities and composition [1]. The paradigmatic examples is the *wild category of types* which has types in a universe $\mathcal{U}$ as objects and functions between such types as arrows. The required identifications hold definitionally. Similarly, we can naively translate the notion of a functor to arrive at the definition of a *wild functor* between two wild categories.[2]

Another example is the *wild category* Map *of maps.* Its objects are maps between types and an arrow between objects $f : A \to B$ and $g : X \to Y$ is a commutative square, i.e. two maps $u : A \to X$ and $v : B \to Y$ with a witness γ that $v \circ f$ equals $g \circ u$. Our first main contribution, Theorem 1, concerns the interplay of two binary operations on Map: *pushout-products* and *pullback-homs.* Given two maps $f : A \to B$ and $g : X \to Y$, their pushout-product $f \mathbin{\widehat{\times}} g$ and pullback-hom $f \pitchfork g$ are respectively given by the dashed maps in the following diagrams.

$$\begin{tikzcd}
A \times X \arrow[r, "f \times \mathrm{id}_X"] \arrow[d, "\mathrm{id}_A \times g"] & B \times X \arrow[d] \arrow[ddr, bend left, "\mathrm{id}_B \times g"] & \\
A \times Y \arrow[r] \arrow[ur, phantom, "\ulcorner"] \arrow[drr, bend right, "f \times \mathrm{id}_Y"'] & (B \times X) +_{A \times X} (A \times Y) \arrow[dr, dashed, "f \mathbin{\widehat{\times}} g"] & \\
 & & B \times Y
\end{tikzcd}
\qquad
\begin{tikzcd}
X^B \arrow[dr, dashed, "f \pitchfork g"] \arrow[drr, bend left, "X^f"] \arrow[ddr, bend right, "g^B"'] & & \\
 & X^A \times_{B^A} Y^B \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner"] & X^A \arrow[d, "g^A"] \\
 & Y^B \arrow[r, "Y^f"] & Y^A
\end{tikzcd}$$

---

[1] For the present work, the notion of a wild category is not strictly necessary and simply acts as a bookkeeping device. The examples that we consider here have much more structure than what the concept of a wild category prescribes and we do not expect our results to apply to arbitrary wild categories.

[2] Note that there is a little bit of ambiguity as to what a wild functor should be. If we start from 1-category theory and ignore conditions on truncation levels, we arrive at the definition of a wild functor that preserves identities and compositions. However, if we start from $(\infty, 1)$-category theory and forget all coherence conditions above a certain level, then a wild functor also preserves the identifications witnessing associativity and identity equations. While the latter is arguably the conceptually more convincing approach, we adopt the former one.

Theorem 1 (Leibniz adjunction)**.** The pushout-product and pullback-hom extend to wild functors on Map that are adjoints, i.e. we have natural equivalences $\mathsf{Map}(f \mathbin{\widehat{\times}} g, h) \simeq \mathsf{Map}(f, g \pitchfork h)$. In particular, given objects i, j, and f in Map, the objects $(i \mathbin{\widehat{\times}} j) \pitchfork f$ and $i \pitchfork (j \pitchfork f)$ are equal.

A naive approach to proving Theorem 1 quickly becomes intractable due to the involved bookkeeping of coherences. To get around this, we turn to the wild category Fam. Its objects are pairs $(A : \mathcal{U}, B : A \to \mathcal{U})$ of an index type and a type family over it. An arrow in $\mathsf{Fam}((A,B);(X,Y))$ is given by a map $m : A \to X$ and a dependent map $\mu : \prod(a : A)\,(B(a) \to Y(m\,a))$. Identities and composition are as expected, and we note that, unlike for Map, the identifications for a wild category hold definitionally for Fam.

It is a well known consequence of the univalence axiom (see e.g. [9, Theorem 4.8.3]) that the objects of Map and Fam are equivalent. Specifically, a map corresponds to the type family of its fibers, and a type family corresponds to the projection map from its total space.

Proposition 2. The wild categories Map and Fam are equivalent in the sense that we have a wild functor χ between them that is an equivalence on objects and on arrows.

Tracing the equivalence of wild categories we consider the following constructions on Fam.

Definition 3. Given two types families $B : A \to \mathcal{U}$ and $Y : X \to \mathcal{U}$, their pushout-product and pullback-hom are respectively defined to be

$$(A,B) \mathbin{\widehat{\times}} (X,Y) :\equiv (A \times X, (a,x) \mapsto B(a) * Y(x)),$$

$$(A,B) \pitchfork (X,Y) :\equiv \Big(\mathsf{Fam}((A,B);(X,Y)), (m,\mu) \mapsto \prod(a : A)\,\mathrm{const}(\mu\,a)\Big).$$

Here, $*$ denotes the join [9, p. 198] of two types (i.e. the pushout of the product projections) and for a map $g : C \to D$, we put $\mathrm{const}\,g :\equiv \sum(d : D)\prod(c : C)\,g\,c = d$.

By construction, the wild functor χ from Proposition 2 and its inverse preserve pushout-products and pullback-homs. In fact, for the pushout-product, this follows directly from the observation that it is the fiberwise join [7, Theorem 2.2]. For Fam is relatively straightforward to prove that the pushout-product and pullback-hom are adjoints. Appealing to Proposition 2 and the preservation of pushout-products and pullback-homs, we obtain Theorem 1.

Application: orthogonality and simplicial type theory Important concepts in the classical development of higher category theory, and the study of spaces more generally, are the concepts of *lifting conditions* and *orthogonality.* Given the solid commutative square (1), we can consider the type of diagonal lifts, i.e. the type of functions $d : B \to Y$ that make the triangles commute. Because we are in the wild categorical setting of untruncated types, we have to take the higher cells into account, i.e. we require witnesses α and β of the commutativity of the upper and lower triangles, together with a suitable coherence that α and β compose to the given witness γ of the commutativity of the square.

$$\begin{tikzcd} A \arrow[r, "u"] \arrow[d, "i"'] & Y \arrow[d, "f"] \\ B \arrow[r, "v"'] \arrow[ur, dashed, "d"] & X \end{tikzcd} \tag{1}$$

Definition 4 (Diagonal fillers of a square; [8, Def. 1.43])**.** A *commutative square* is an arrow in Map. Given a commutative square Γ, its *type of diagonal fillers* consists of a function, proofs of commutativity of the triangles, and a proof that the triangles compose to the full square.

Definition 5 (Orthogonality, $i \perp f$)**.** Let $i : A \to B$ and $f : Y \to X$ be functions. We say that i is *left orthogonal* to f (and f is *right orthogonal* to i), written $i \perp f$, if every commutative square as in (1) has a contractible type of diagonal fillers.

Lemma 6.

(i) For all maps i and f, we have $i \perp f$ if and only if $i \pitchfork f$ is an equivalence.
(ii) Equivalences are left- and right-orthogonal to any map.
(iii) Left orthogonal maps are closed under pushout-products with arbitrary functions.
(iv) Left orthogonal maps are closed under retracts.

We note that the proof of (iii) makes use of Theorem 1.

Simplicial type theory For simplicial type theory in general, see [5]; we follow the "internal" approach of Gratzer, Weinberger, and Buchholtz [2, 3]. We assume that we are given a set I (the "interval") together with $0, 1 : I$ and binary operations $\wedge, \vee$ that turn I into a bounded distributive lattice. As is standard, we write $x \leq y$ (and $y \geq x$) for $x \wedge y = y$. This turns $\leq$ into a reflexive and transitive binary order on I, with 0 and 1 as minimal and maximal elements. In contrast to [2, 3], we do not require linearity of the order $\leq$.

Using I, one can define the standard simplices Δ^n and horns Λ^n_k, together with canonical horn inclusions $\lambda^n_k : \Lambda^n_k \hookrightarrow \Delta^n$.

Definition 7 (Following [5])**.** A type X is a *Segal type* if the map $X \to 1$ is right orthogonal to the horn inclusion λ^2_1, and $i : A \to B$ is *inner anodyne* if it is left orthogonal to every Segal type.

By a result of Riehl and Shulman [5, Proposition 5.21], the maps λ^3_1 and λ^3_2 are inner anodyne. In our second main result, we generalize their result to all inner horn inclusions, i.e. all horn inclusions λ^n_k where k is strictly between 0 and n. This is a version of Lurie's [4, Corollary 2.3.2.2] in simplicial type theory.

Theorem 8. For $n : \mathbb{N}$ and k with $0 < k < n$, the inner horn inclusion λ^n_k is inner anodyne.

The proof of Theorem 8 relies on Lemma 6 and proceeds by showing that λ^n_k is a retract of $\lambda^n_k \mathbin{\widehat{\times}} \lambda^2_1$. It further relies on a version of Joyal's lemma [4, Proposition 2.3.2.1]. This lemma is based on pushouts and retracts in the category of sets, while all our results are about the corresponding notions in the wild category of types. The canonical inclusion of the univalent 1-category of sets into the wild category of types does not in general preserve pushouts (i.e. not all set-pushout squares are homotopy pushouts); fortunately, pushouts along embeddings (such as the horn inclusions) are preserved, which explains why no mismatch between the different notions of pushout occurs.

Acknowledgements We thank Christian Sattler for an interesting discussion on the Leibniz adjunction. This work was supported by the Royal Society [URF\R1\191055, URF\R\241007], the Engineering and Physical Sciences Research Council [EP/Z000602/1], and the Knut and Alice Wallenberg Foundation's program for mathematics.

References

[1] Paolo Capriotti and Nicolai Kraus. "Univalent higher categories via complete semi-Segal types". In: *Proceedings of the ACM on Programming Languages*. Vol. 2. POPL. Association for Computing Machinery, 2017, pp. 1–29. DOI: 10.1145/3158132.

[2] Daniel Gratzer, Jonathan Weinberger, and Ulrik Buchholtz. "Directed univalence in simplicial homotopy type theory". 2024. arXiv: 2407.09146 [cs.LO].

[3] Daniel Gratzer, Jonathan Weinberger, and Ulrik Buchholtz. "The Yoneda embedding in simplicial homotopy type theory". In: *2025 40th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*. IEEE, 2025, pp. 127–142. DOI: 10.1109/LICS65433.2025.00017. arXiv: 2501.13229 [cs.LO].

[4] Jacob Lurie. *Higher Topos Theory*. Vol. 170. Annals of Mathematics Studies. Princeton University Press, 2009. DOI: `10.1515/9781400830558`. arXiv: `math/0608040`.

[5] Emily Riehl and Michael Shulman. "A type theory for synthetic ∞-categories". In: *Higher Structures* 1.1 (2017), pp. 147–224. DOI: `10.21136/hs.2017.06`.

[6] Emily Riehl and Dominic Verity. "The theory and practice of Reedy categories". In: *Theory and Applications of Categories* 29.9 (2013), pp. 256–301. DOI: `10.70930/tac/2q3vx1tg`.

[7] Egbert Rijke. "The join construction". 2017. arXiv: `1701.07538 [math.CT]`.

[8] Egbert Rijke, Michael Shulman, and Bas Spitters. "Modalities in homotopy type theory". In: *Logical Methods in Computer Science* 16 (2020). DOI: `10.23638/LMCS-16(1:2)2020`.

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