WHAT IS HOMOTOPICAL ABOUT HOMOTOPY TYPE THEORY?

CHENTIAN WU

Abstract. In this paper I try to answer the question: *what makes Homotopy Type Theory (HoTT) genuinely homotopic?* By bridging Martin-Löf's dependent type theory and classical homotopy theory, we (1) show how Martin-Löftype theory's core components - universes, dependent types, and identity types - naturally encode geometric concepts when viewed through a homotopical lens, (2) demonstrate how HoTT internalizes five key homotopy-theoretic ideas: paths as equality proofs, fibrations as dependent types, equivalences, univalence, and higher inductive types, and (3) apply these to reconstruct two classical results: the computation of $\pi_1(S^1) \simeq \mathbb{Z}$ via path induction and transport, and the Seifert-van Kampen theorem through pushouts of higher inductive types. The analysis reveals how HoTT captures topological phenomena synthetically, replacing analytic machinery with type-theoretic primitives.

Contents

1. Why Homotopy inside Type Theory? 1
2. Martin-Löf's Dependent Type Theory: A Topological Lens 2
2.1. Universes and Type Families as Stratified Spaces 2
2.2. Dependent Types as Continuous Constructions 2
2.3. Identity Types as Path Spaces 3
2.4. Path Induction - Contracting the Interval 3
2.5. A First Glimpse of Homotopy 3
3. Key Homotopic Ingredients of HoTT 4
3.1. Paths as Homotopies and Covering Spaces 5
3.2. Π-Types as Sections of Fibrations 5
3.3. Equivalences and the Universal Cover 5
3.5. Lifting Properties and Transport 6
3.6. Free Groups and the Van Kampen Theorem 6
3.7. Fundamental Group Action via Deck Transformations 6
3.9. Synthesis: HoTT as Synthetic Homotopy Theory 7
4. Semantic Models at a Glance 7
5. Conclusion and Outlook 7
References 8

1. Why Homotopy inside Type Theory?

Homotopy Type Theory (HoTT) merges two seemingly disparate fields: the logical rigor of type theory and the geometric intuition of homotopy theory. To appreciate its significance, we begin by contextualizing its foundations.

Type Theory formalizes mathematics through a computational lens. In this framework, propositions are represented as *types*, and proofs as *terms* inhabiting those types. For instance, the statement "2 is even" corresponds to a type $\mathsf{Even}(2)$, and a proof of this statement is a term $t : \mathsf{Even}(2)$. *Dependent Type Theory* extends this idea by allowing types to depend on terms: if A is a type and $B(x)$ is a type for each $x : A$, the dependent product type $\prod_{x:A} B(x)$ encodes universal quantification ("for all $x : A$, $B(x)$ holds"). This mechanism underpins modern proof assistants like Coq [2] and Agda [1]. However, traditional type theory treats equality simplistically—terms a and b are either judgmentally equal ($a :\equiv b$) or not, discarding any higher-dimensional structure behind why they are equal.

Homotopy Theory, on the other hand, studies spaces through continuous deformations: paths between points, homotopies between paths, and so on. Vladimir Voevodsky's groundbreaking insight was to reinterpret types as spaces, terms as points, and equality proofs $p : a = b$ as paths from a to b. HoTT elevates equality to a first-class geometric notion, preserving not just whether two terms are equal but how they are equal. For example, two programs proven equal in HoTT may follow distinct computational paths, analogous to different routes connecting the same endpoints.

The fusion of these ideas crystallized with Hofmann and Streicher's 1994 groupoid model and Voevodsky's later contributions: the *univalence axiom* (equating equivalent types) and *higher inductive types* [5] (defining spaces by specifying points and paths). Together, they enable HoTT to internalize homotopy-theoretic concepts directly into type theory. For computer scientists, this enriches program verification with geometric reasoning; for mathematicians, it offers a language where proofs are both human-readable and machine-checkable.

2. Martin-Löf's Dependent Type Theory: A Topological Lens

Modern homotopy type theory rests on three structural features of Martin-Löf dependent type theory—*universes*, *dependent types*, and *identity types*. Interpreting each through a geometric lens reveals how the syntax of proofs encodes classical homotopy theoretic ideas.

2.1. Universes and Type Families as Stratified Spaces.

To quantify over types without reproducing Russell-style paradoxes Martin-Löf introduced an ascending hierarchy

$$\mathcal{U}_0 : \mathcal{U}_1 : \mathcal{U}_2 : \cdots,$$

where every level $\mathcal{U}_i$ itself lives in the next, a property called *cumulativity.* We can notice the same pattern when we are assembling a CW-complex: the n-skeleton depends only on cells attached in lower dimensions, yet remains a subspace of the completed object. Thus each universe operates like a skeleton that supports all types—and hence all spaces—constructed so far, while leaving room to adjoin further "cells" at the next stage.

Given a base type A a *type family* $B : A \to \mathcal{U}$ assigns to every point $x : A$ a fibre $B(x)$. Topologically one recognises the assignment as a fibration $p : \sum_{x:A} B(x) \to A$ whose total space is the Σ-type $\sum_{x:A} B(x)$ and whose fibre over x is literally $B(x)$. When B is the constant function $\lambda(x : A).\ C$ the fibration becomes trivial, $A \times C \xrightarrow{\pi_1} A$; when B varies non-trivially one obtains twisted bundles, covering spaces, or more exotic constructions such as local systems of higher groupoids.

2.2. Dependent Types as Continuous Constructions.

Definition 2.1 (Π-types as Sections)**.** Given a family $B : A \to \mathcal{U}$ the dependent function type $\prod_{(x:A)} B(x)$ consists of terms that pick a point $f(x) : B(x)$ *continuously* in x. Equivalently a term f determines a section $s : A \to \sum_{x:A} B(x)$ via $s(x) :\equiv (x, f(x))$, and the definitional equality $\pi \circ s = \mathsf{Id}_A$ reads "s is a genuine section" after projection.

From a logical standpoint Π-types internalize universal quantification: the judgment $t : \prod_{x:A} B(x)$ is a constructive proof of "for every x we have $B(x)$". From a homotopical standpoint sections detect whether a bundle admits global trivializing data, a theme that resurfaces when Π-types are used to model parallel transport and connection forms.

Definition 2.2 (Σ-types as Total Spaces)**.** Dually, the dependent pair type $\sum_{(x:A)} B(x)$ represents the space obtained by *gluing* all fibres together. A point $(a, b) : \sum_{x:A} B(x)$ simultaneously witnesses the existence of $a : A$ and a dependent element $b : B(a)$; it therefore embodies the constructive content of the existential quantifier. In topology the same ordered pair marks a geometric point lying over a in the total space of the fibration.

Because Σ and Π are adjoint (yes, by this we mean adjunctions in category theory) in an appropriate sense, one already glimpses a categorical formulation of fibre bundles: sections are right adjoints to projection, while total-space formation is left adjoint.

2.3. **Identity Types as Path Spaces.** The conceptual breakthrough of HoTT is to interpret propositional equality as a space of paths in the topological sense. For two terms $a, b : A$ the identity type $\mathsf{Id}_A(a, b)$ carries as inhabitants the paths from a to b; the canonical constructor $\texttt{refl}_a : \mathsf{Id}_A(a, a)$ is the constant path. Two radically different equalities coexist:

$a \equiv b : A$	$p : \mathsf{Id}_A(a, b)$
judgmental, enforced by definitional computation	propositional, a geometric path that may vary

Judgmental (or definitional) equality corresponds to syntactic normalization, whereas propositional equality admits many distinct witnesses; the latter is where homotopy becomes useful.

2.4. **Path Induction - Contracting the Interval.** Every identity type is governed by the *path induction* (or J) rule: to prove a property C of arbitrary paths it suffices to prove it for the trivial path. Formally, given a family $C : \prod_{x,y:A} \mathsf{Id}_A(x, y) \to \mathcal{U}$ and a term $d : \prod_{x:A} C(x, x, \texttt{refl}_x)$ there exists a function

$$\mathsf{J}(d) : \prod_{x,y:A} \prod_{p:\mathsf{Id}_A(x,y)} C(x, y, p),$$

unique up to definitional equality. From a geometric vantage point path induction says that the unit interval contracts onto a point, so any construction that is homotopy-invariant over the interval already follows from its value at the constant path.

2.5. **A First Glimpse of Homotopy.** Suppose $p, q : \mathsf{Id}_A(a, b)$ are two non-identical proofs of equality. They materialize as distinct paths between a and b. Applying identity types once more, $\mathsf{Id}_{\mathsf{Id}_A(a,b)}(p, q)$ is the type of homotopies $p \Rightarrow q$, and iterating the process generates a tower of higher paths whose globular structure equips every type with an ∞-groupoid of its points, paths, homotopies, and so on. Grothendieck envisioned such structures as the language in which algebraic topology should be carried out; in HoTT they arise automatically from the basic rules of type formation and path induction.

So far, we have now reinterpreted the building blocks of Martin-Löf theory — universes, dependent types, identity types, and path induction — through geometry. Table 1 summarizes the interpretation of Type Theory in Logic and Topology.

Type Theory	**Logic**	**Homotopy Theory**
$A : \mathcal{U}$	a proposition A	a topological space A
$a : A$	a proof a of proposition A	a point a in space A
$A \to B$	$A \Rightarrow B$ (implication)	a continuous map $A \to B$
$P : A \to \mathcal{U}$	a predicate $P(x)$ for each $x : A$	a fibration P over A with fiber $P(x)$ at each $x : A$
$\prod_{x:A} P(x)$	$\forall x : A.P(x)$ (universal quantification)	space of continuous sections of fibration P
$\sum_{x:A} P(x)$	$\exists x : A.P(x)$ (existential quantification)	total space of fibration P
$A \times B$	$A \wedge B$ (conjunction)	the product space $A \times B$
$A + B$	$A \vee B$ (disjunction)	the disjoint union $A \sqcup B$
$p : \mathsf{Id}_A(a, b)$	p is a proof of the equality of a and b	p is a path from a to b in space A
$\mathtt{refl}_a : \mathsf{Id}_A(a, a)$	the reflexivity of equality at a	the constant path at point a
$\mathbb{1}$	`true` (true proposition)	contractible space
$\mathbb{0}$	`false` (false proposition)	empty space
`Prop`	propositions (0-truncated types)	spaces with at most one path between any two points
`Set`	sets (1-truncated types)	spaces with contractible path spaces
n-truncated type	proposition up to level n	space with trivial homotopy groups above level n
function extensionality	functions equal if equal on all inputs	homotopy between functions
univalence	isomorphic types are equal	equivalent spaces are equal
higher inductive type	inductive definition with path constructors	space with specified paths and higher cells

TABLE 1. The HoTT translation dictionary (merged from [4, 6]).

3. Key Homotopic Ingredients of HoTT

In this section we are going to analyze five important *synthetic*[1] ingredients of homotopy in HoTT. While there are many more advanced aspects to this topic, to the best of my knowledge from MATH 552, these are the key concepts I've been able to properly understand (given time constraints).

[1]"Synthetic" (in HoTT) refers to reasoning about spaces and paths directly within type theory without explicitly constructing their underlying topological or set-theoretic models.

3.1. **Paths as Homotopies and Covering Spaces.** Given a type A and points $a, b : A$, the identity type $\mathsf{Paths}_A(a, b) :\equiv \mathsf{Id}_A(a, b)$ is interpreted homotopically as the space of paths connecting a to b. Repeating the identity-type construction endows every type with an entire tower of higher path spaces $\mathsf{Paths}^n_A(a, b)$, and the coherence laws of identity establish an ∞-groupoid structure. In classical language this means that reflexivity, symmetry, transitivity, and their higher analogues are not separate axioms but canonical operations internal to $\mathcal{U}$.

A basic but instructive illustration is the circle S^1. Presenting S^1 as a higher inductive type with a single point constructor base and a loop constructor $\mathsf{loop} : \mathsf{Paths}_{S^1}(\mathsf{base}, \mathsf{base})$ already forces the entire set of homotopy classes of maps out of the circle to satisfy the desired universal property; no appeal to piecewise linear charts or CW structures is necessary. In this way HoTT replaces analytic path concatenation by the purely syntactic operation of transitivity in Id-types.

A dependent type $P : A \to \mathcal{U}$ then behaves as a fibration whose total space is $\sum_{x:A} P(x)$. The canonical projection $\pi : \sum_{x:A} P(x) \to A$ mirrors a covering map in topology. The *path induction* principle asserts that to define data depending on an arbitrary path it suffices to specify the case of the trivial path $\mathtt{refl}_a$; this "contracts the interval" and yields the synthetic counterpart of the homotopy lifting property. The induced transport map

$$\mathsf{transport}^P_q : P(b_0) \longrightarrow P(b_1) \qquad (q : \mathsf{Paths}_B(b_0, b_1))$$

is formally unique and automatically functorial, thus reproducing the monodromy of a covering space without explicit reference to topological microstructure.

3.2. **Π-Types as Sections of Fibrations.** Where topology studies continuous sections of a fibration $\pi : E \to A$, type theory studies terms of the dependent function type $\prod_{x:A} P(x)$. A term $f : \prod_{x:A} P(x)$ canonically determines a section $s : A \to \sum_{x:A} P(x)$ by the rule $s(x) :\equiv (x, f(x))$, and the typing judgement $\mathsf{ap}_\pi(f(x)) = \mathtt{refl}_x$ is literally the equation $\pi \circ s = \mathsf{Id}_A$ transcribed into the internal language. Consequently the classical slogan "sections are homotopies" acquires a formal meaning: Π-types serve simultaneously as spaces of global elements and as witnesses of path-based coherence.

Definition 3.1. A dependent map $f : \prod_{x:A} P(x)$ is called *constant up to homotopy* if for all $x, y : A$ there exists a path $p : \mathsf{Paths}_{P(y)}(f(x = y), f(y))$ whose projection onto A is $\mathtt{refl}_y$. Constant maps in this sense correspond to π_1-invariant sections of a covering space in the classical setting.

This perspective clarifies why Π-types satisfy functional extensionality whenever univalence holds: two sections are equal just when they are pointwise connected by a path, precisely the condition under which they represent the same global section.

3.3. **Equivalences and the Universal Cover.** A map $f : A \to B$ is an *equivalence* if each of its fibres $\sum_{b:B} \mathsf{Paths}_B(f(a), b)$ is contractible. Univalence enhances this notion by stipulating that the type $(A \simeq B)$ of equivalences coincides with the identity type $(A = B)$. Equivalences are therefore paths in $\mathcal{U}$, so that "moving along" a family of equivalences simply is transport in $\mathcal{U}$.

Theorem 3.4. *The loop space of the circle satisfies $\Omega S^1 \simeq \mathbb{Z}$.*

Sketch. Define $\mathsf{code} : S^1 \to \mathcal{U}$ by $\mathsf{code}(\mathsf{base}) :\equiv \mathbb{Z}$ and $\mathsf{transport}^{\mathsf{code}}(\mathsf{loop}) :\equiv \mathsf{suc}$. The total space $\sum_{x:S^1} \mathsf{code}(x)$ carries a projection with homotopy-lifting behaviour identical to the exponential map $\exp : \mathbb{R} \to S^1$. Contractibility of every fibre implies that paths at base

are in bijective correspondence with integers, yielding the stated equivalence without leaving type theory. □

Because the argument never mentions open covers or local triviality, it demonstrates that HoTT faithfully reproduces covering-space phenomena using purely synthetic tools.

3.5. **Lifting Properties and Transport.** In classical algebraic topology the homotopy lifting property (HLP) says that a covering map $p : E \to B$ allows every homotopy $H : X \times [0,1] \to B$ to be lifted uniquely once its restriction to $X \times \{0\}$ is fixed. HoTT wraps the entire content of HLP into the functoriality and higher coherences of $\mathsf{transport}$.

Let $q : \mathsf{Paths}_B(b_0, b_1)$ and $e_0 : P(b_0)$. The dependent elimination rule for identity types produces the unique term $\mathsf{transport}^P_q(e_0) : P(b_1)$. Naturality of $\mathsf{transport}$ under path concatenation encodes the uniqueness clause of path lifting, while the computation rule for $\mathtt{refl}_{b_0}$ encodes the starting-point condition. Higher paths between paths in B lift to dependent paths between the corresponding transport maps, providing the synthetic substitute for homotopy lifting. Hence the entire monodromy representation $\pi_1(B) \to \mathsf{Aut}(P(b_0))$ can be recovered from first principles of dependent type theory.

3.6. **Free Groups and the Van Kampen Theorem.** The Seifert-Van Kampen theorem (SVK) computes the fundamental group of a union from the fundamental groups of its parts and their intersection,

$$\pi_1(U \cup V) \simeq \pi_1(U) *_{\pi_1(U \cap V)} \pi_1(V).$$

HoTT reconstructs SVK synthetically via pushouts of higher inductive types. Suppose X is the pushout of maps $i : W \to U$ and $j : W \to V$. Because pushouts in HoTT satisfy the same universal property as in homotopy theory, mapping out of X into any other type induces an equivalence between dependent function spaces mirroring the amalgamated free product above.

Theorem 1 (Wedge of two circles)**.** *Present $S^1 \vee S^1$ by two point constructors $\mathsf{base}_1, \mathsf{base}_2$ and two loop constructors $\mathsf{loop}_1, \mathsf{loop}_2$. Any map from $S^1 \vee S^1$ into a group-valued Eilenberg–MacLane space is completely specified by the images of loop_1 and loop_2, and the concatenation law for loops translates into the group law in the free product $\mathbb{Z}*\mathbb{Z}$. Path induction formalizes this reasoning, yielding a direct proof of $\pi_1(S^1 \vee S^1) \simeq \mathbb{Z} * \mathbb{Z}$ internal to HoTT.*

The synthetic route not only avoids piecewise-linear approximations but also produces computational content: the proof yields an explicit algorithm extracting normal forms of words in $\pi_1(S^1 \vee S^1)$.

3.7. **Fundamental Group Action via Deck Transformations.** Let $p : \tilde{X} \to X$ be a universal cover. The group $\mathsf{Deck}(\tilde{X})$ of deck transformations consists of those self-equivalences of $\tilde{X}$ that commute with p. In HoTT the covering is a dependent type $\mathsf{code} : X \to \mathcal{U}$; its total space $\sum_{x:X}\mathsf{code}(x)$ inherits an internal Deck-action from transport.

Theorem 3.8. *For $X = S^1$ and code as in Theorem 3.4, $\mathsf{Deck}\big(\sum_{x:S^1}\mathsf{code}(x)\big) \simeq \mathbb{Z}$.*

Idea. Any integer $n : \mathbb{Z}$ gives an equivalence $f_n : (x,k) \mapsto (x, k+n)$, and $\mathsf{transport}^{\mathsf{code}}(\mathsf{loop}^n) = \lambda k.\ k+n$ shows that these exhaust all equivalences commuting with the projection. Functoriality of $\mathsf{transport}$ yields a group homomorphism $\mathbb{Z} \to \mathsf{Deck}$ that is easily seen to be inverse to the map sending a deck transformation to its action on the distinguished fibre over base. □

Because deck transformations are equivalences in the sense of univalence, their classification by $\mathbb{Z}$ is witnessed by a path in $\mathcal{U}$, once again turning a geometric identification into an internal equality.

3.9. **Synthesis: HoTT as Synthetic Homotopy Theory.** Collecting the preceding themes, one may characterise HoTT as a framework where traditional homotopy theory is recovered from the interaction of four primitive notions:

(i) identity types model paths and higher paths; *(ii) dependent types* encode fibrations and covering spaces; *(iii) higher inductive types* allow defining spaces by specifying both points and paths (e.g., attaching cells or gluing components), which directly encodes theorems like Seifert-van Kampen within type theory. *(iv) univalence* unifies equivalence with identity, letting group actions and deck transformations be expressed by transport.

Because these ingredients are computationally meaningful, classical homotopical arguments admit direct formalization in proof assistants such as AGDA [1], COQ-HOTT [2], or LEAN [3]. The cost of analytic overhead is thus exchanged for the benefit of executable proofs whose correctness rests solely on the constructive core of type theory.

4. Semantic Models at a Glance

The consistency of HoTT rests on mathematical models that bridge its syntax with concrete geometric or computational structures. Two models stand out for their philosophical and practical implications.

The *Kan simplicial set model* interprets types as spaces built from simplices—higher-dimensional analogs of triangles and tetrahedrons. This model validates univalence by treating type equivalences as homotopy equivalences. However, it lacks computational rules for paths: while it guarantees the existence of a path $a = b$, it does not specify how to construct or compute it.

A more constructive approach is offered by *cubical type theory*, pioneered by Bezem, Coquand, and Huber (2014). This model introduces an *interval object* $\mathbb{I}$ (representing a continuum from 0 to 1) and *connection operations* to glue higher-dimensional cubes. Here, paths become computable functions over $\mathbb{I}$. For instance, the loop $\mathsf{loop} : S^1$ in Cubical Agda can be evaluated as a concrete sequence of interval manipulations. This model not only justifies univalence constructively but also enables algorithms for normalizing higher-dimensional paths, as demonstrated in our analysis of $\pi_1(S^1)$ in Section 3.

These models are not mere abstractions. They ground HoTT's synthetic reasoning in computation: when we assert $\pi_1(S^1) \simeq \mathbb{Z}$, the cubical model ensures this equivalence is not just symbolic—it reduces to executable code manipulating integers. This synergy between syntax and semantics positions HoTT as both a foundational theory and a practical tool for formal mathematics.

5. Conclusion and Outlook

Homotopy Type Theory redefines the foundations of mathematics by internalizing the geometric intuition of homotopy theory into type theory. We have demonstrated how paths replace equality proofs, enabling synthetic reasoning about spaces like the circle S^1. Dependent types model fibrations, recovering classical covering spaces and monodromy through transport, while univalence simplifies the treatment of equivalent structures. Higher inductive types allow defining spaces synthetically, bypassing cumbersome topological machinery.

These advances are not merely theoretical. The proof of $\pi_1(S^1) = \mathbb{Z}$, once a laborious theorem in algebraic topology, now fits within a few lines of Agda code. Looking ahead, HoTT promises to reshape mathematical practice. Synthetic homology could define homology groups directly via higher inductive types, avoiding simplicial complexes. Formalized algebraic geometry might encode schemes in type theory, enabling verified computations. Pedagogically, HoTT proof assistants could make advanced topology accessible to undergraduates, transforming abstract concepts like fibrations into interactive code.

For students and researchers alike, HoTT offers a unique bridge: it demands neither encyclopedic topology nor category theory, yet rewards geometric intuition with profound connections between logic and computation. As the field matures, it may well become the lingua franca of 21st-century mathematics—rigorous, computational, and deeply geometric.

References

1. Agda Developers, *Agda*.
2. Andrej Bauer, Jason Gross, Peter LeFanu Lumsdaine, Mike Shulman, Matthieu Sozeau, and Bas Spitters, *The hott library: A formalization of homotopy type theory in coq*, 2016.
3. Leonardo de Moura and Sebastian Ullrich, *The lean 4 theorem prover and programming language*, Automated Deduction – CADE 28: 28th International Conference on Automated Deduction, Virtual Event, July 12–15, 2021, Proceedings (Berlin, Heidelberg), Springer-Verlag, 2021, p. 625–635.
4. Egbert Rijke, *Introduction to homotopy type theory*, 2022.
5. The Univalent Foundations Program, *Homotopy type theory: Univalent foundations of mathematics*, `https://homotopytypetheory.org/book`, Institute for Advanced Study, 2013.
6. Yuhang Wei, *Synthetic homotopy theory*, 2024.