Covering Spaces in Homotopy Type Theory*

Kuen-Bang Hou (Favonia)[1] and Robert Harper[2]

1 **Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA**
`favonia@cs.cmu.edu`

2 **Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA**
`rwh@cs.cmu.edu`

Abstract

Broadly speaking, algebraic topology consists of associating algebraic structures to topological spaces that give information about their structure. An elementary, but fundamental, example is provided by the theory of covering spaces, which associate groups to covering spaces in such a way that the universal cover corresponds to the fundamental group of the space. One natural question to ask is whether these connections can be stated in homotopy type theory, a new area linking type theory to homotopy theory. In this paper, we give an affirmative answer with a surprisingly concise definition of covering spaces in type theory; we are able to prove various expected properties about the newly defined covering spaces, including the connections with fundamental groups. An additional merit is that our work has been fully mechanized in the proof assistant Agda.

1998 ACM Subject Classification F.4.1 Mathematical Logic

Keywords and phrases homotopy type theory, covering space, fundamental group, mechanized reasoning

Digital Object Identifier 10.4230/LIPIcs.CVIT.2016.23

1 Introduction

Homotopy type theory [33] is a new area arising from surprising connections between type theory, homotopy theory and category theory. Using a variant of Martin-Löf type theory [29, 33, 36] extended with Vladimir Voevodsky's univalence axiom [19] and higher inductive types [27, 33], homotopy-theoretic concepts can be expressed in type theory in a direct and intuitive way, as we will see in the case of covering spaces.

The connection between this variant of Martin-Löf type theory and homotopy theory is through the *identification-as-path*[1] interpretation [3, 12, 19, 28, 33–35, 38]. According to this interpretation, types may be treated as spaces,[2] elements of a type as points in a space, functions as continuous mappings, families of types as fibrations, and of course identifications

* This research was sponsored by the National Science Foundation under grant numbers CCF-1116703 and Air Force Office of Scientific Research under grant numbers FA-95501210370 and FA-95501510053. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution, the U.S. government or any other entity.

[1] Identification types are also called *identity types* in the literature.

[2] More precisely, *simplicial sets*, and the interpretation was given in [19]. *Spaces* in this paper really mean *simplicial sets* unless we are explicitly discussing point-set topology. For clarity, spaces in point-set topology will be denoted as *topological* spaces.

42nd Conference on Very Important Topics (CVIT 2016).
Editors: John Q. Open and Joan R. Acces; Article No. 23; pp. 23:1–23:16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

as paths;[3] the higher-dimensional structures induced by iterated identification make every type an ∞-groupoid.[4] With this connection, we can use type-theoretic approaches to state, prove and even mechanize theorems from classical homotopy theory, making the type theory a framework of *synthetic homotopy theory*; proofs are dependently typed functional programs, except that they do not run due to incomplete support of computation of the univalence axiom and higher inductive types in current mature proof assistants and the type theory we use in this paper.

A wide range of homotopy-theoretic results have been developed and mechanized in proof assistants such as Agda [6,30], Coq [1,4] and Lean [9,11], for example homotopy groups of spheres [5,23,26,33], the Seifert–van Kampen theorem [17], the Blakers–Massey connectivity theorem [16], the Eilenberg–Mac Lane spaces [25], the Mayer–Vietoris sequences [10], the Cayley–Dickson construction [8], the double groupoids [37] and many more [24,32,33]. Proofs done in homotopy type theory have the advantage that they admit many models other than the homotopy theory of topological spaces; some even stimulated new research in mathematics [2,31]. As a side note, many theorems were actually first mechanized in proof assistants and then "unmechanized" to engage wider audience, which is only possible through a powerful, high-level framework such as homotopy type theory.

Covering spaces are one of the important constructs in homotopy theory, and given the connection between type theory and homotopy theory, a natural question to ask is whether such a notion can be stated in type theory as well. It turns out that we can express covering spaces concisely as follows.

▶ **Definition 1.** A *covering space* of a type (space) A is a family of sets indexed by A.

That is, the type of covering spaces of A is simply $A \to \texttt{Set}$ where `Set` is the type of all sets, the universe of all types that have at most one identification between any two points. Several examples are shown in Figure 1.

How do we know this definition really defines covering spaces? A characteristic feature of covering spaces of a connected space A in the classical theory is that they are represented by sets with a group action of the fundamental group of A (the set of loops at some point in A). Therefore, we may justify our definitions by proving this theorem, as we will in Section 4. See Figure 1 which also lists such sets corresponding to the covering spaces in the figure. Moreover, considering the category[5] of pointed covering spaces where morphisms are fiberwise functions, we also know there should be an initial covering space (named the *universal* covering space) and it should be represented by the fundamental group itself through the representation theorem stated above.[6] We also managed to show these results as demonstrated in Section 5. Before transitioning to these main theorems, in Section 3 we will also discuss briefly about the discrepancies between our formulation and the classical definition. More discussions and future research directions can be found in Section 6.

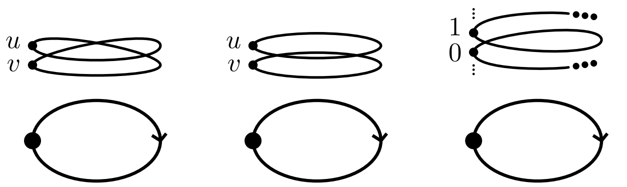
All the results mentioned in this paper have been mechanized in the proof assistant Agda [6]. The representation theorem was briefly mentioned in the book without proofs [33]

[3] All topological terms in this paper should be understood "up-to-homotopy" in some appropriate sense, because every construct in the type theory will respect homotopy equivalence under the intended interpretation to simplicial sets.

[4] To avoid confusion, this result was a meta-theoretical result given in [34], and we believe it has not been internalized in type theory yet.

[5] The *category* here is the same as the *category* introduced in [33, §9.1]. The type of morphisms (fiberwise functions) between two covering spaces is a set because each fiber of a covering space is a set, and the notion of isomorphism in this category collides with identification.

[6] See Section 5 for a more precise statement.



$\begin{cases}\text{base} \mapsto \mathbb{2} \\ \text{loop} \mapsto \mathtt{ua}(\mathtt{not})\end{cases}$	$\begin{cases}\text{base} \mapsto \mathbb{2} \\ \text{loop} \mapsto \mathtt{ua}(\mathtt{id})\end{cases}$	$\begin{cases}\text{base} \mapsto \mathbb{Z} \\ \text{loop} \mapsto \mathtt{ua}(+1)\end{cases}$
set: $\{u, v\}$	set: $\{u, v\}$	set: $\mathbb{Z}$
action: $u \mapsto v$, $v \mapsto u$	action: $u \mapsto u$, $v \mapsto v$	action: $i \mapsto (i+1)$

Figure 1 Correspondence between covering spaces and sets equipped with a group action.

The top row is the visualization of the covering spaces. The middle row shows their type-theoretic formulations (as a function from the circle to the universe) specified by a fiber for the base generator and an identification from the fiber to itself for the loop generator of the circle. The last row is the corresponding sets and their actions, represented by their acts on the (only) loop generator.

and an extended abstract without peer review was posted before [15], but a full paper was never published.

2 Type-Theoretic Notation and Background

We assume readers are already familiar with basic concepts in homotopy type theory, including higher inductive types; interested readers are recommended to read the book [33] for introduction, especially its Chapter 2 describing how type-theoretic concepts may be understood homotopy-theoretically. This section is mainly a brief overview of the notation used in this paper with remarks on some subtle differences from the book [33] or the proof assistant Agda. Overall we are loosely following the style of the book [33] while keeping an Agda translation obvious.

Throughout this paper, $\equiv$ is judgmental equality and $:\equiv$ indicates a definition. The equal sign $=$ is reserved for identification as mentioned below.

2.1 Sums and Products

Let B be a family of types indexed by a type A. *Dependent sum types* are written $\sum_{x:A} B(x)$ with pairs $\langle a; b\rangle$ as elements and *dependent function types* $\prod_{x:A} B(x)$ with λ-functions. The type $\sum_{x:A} B(x)$ is also called the *total space* of B. If $B(x) \equiv B'$ actually does not depend on the index $x : A$, we have the binary product type $A \times B'$ meaning the *non-dependent* sum type $\sum_{\_:A} B'$ and the arrow type $A \to B'$ the *non-dependent* function type $\prod_{\_:A} B'$. Function compositions are written $f \circ g$.

Multi-argument application is written $f(x_1, x_2, \cdots, x_n)$ and nested sum types will be presented as records types with labels (like "`label`"). As a notational abuse, a label is also the projection function which projects out the corresponding component from a record.

2.2 Identification

Let a and b be two points in some type A. The *identification type* or the *path type* between a and b is written $a =_A b$, and A may be omitted if clear from the context. The reflexivity identification at a is written $\texttt{refl}_a$, the concatenation (in the diagram order) written $p \cdot q$, and the inverse identification written p^{-1}.

The induction principle of identification types intuitively states that, given a statement about identifications, one can just consider the `refl` case. The argument is that one may continuously grow a `refl` to arbitrary identifications, and because every function in the type theory is continuous, the conclusion remains valid. However, to make this "continuous-growing" argument work, the precise formulation of this principle is quite delicate and is discussed in more details in [33, §1.12]. For example, the statement about identifications must make sense for identifications between two possibly different points in order to allow the `refl` case to "grow".

As mentioned in the introduction, identification types may be iterated as $p =_{a =_A b} q$, $P =_{p =_{a =_A b} q} Q$ and so on. Throughout the paper the word *dimension* refers to the level of identification iteration; that is, the n-dimensional structures in type A refer to the nth iteration of identification starting from the type A.

2.3 Universes, Equivalence and Univalence

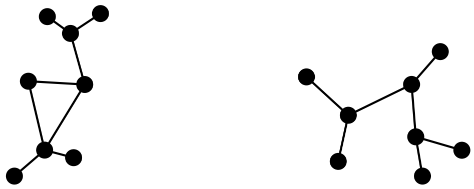
Both the type theory and the proof assistant Agda have a ramified hierarchy of cumulative universes to avoid Girard's paradox [13, 18], but in this paper we will suppress the universe level, pretending there is only one universe written $\mathcal{U}$. Universe levels are explicit and universe lifting is manual in the current Agda system, but they did not constitute an obstacle to mechanizing covering spaces.

The equivalence type $A \simeq B$ intuitively collects all the equivalences between types A and B. It is actually tricky to obtain a good definition for equivalences in homotopy type theory; interested readers are recommended to consult [33, Chap. 4] for a precise definition. For this paper it suffices to know that the following data are sufficient to build a good equivalence: a function from A to B, a function from B to A, and two proofs showing the two compositions are homotopic to the identity functions.

With a good definition of equivalences, we have the univalence axiom stating that the equivalence type between types A and B is itself equivalent to the identification type $A =_{\mathcal{U}} B$. The univalence axiom not only recognizes new identifications between types, but also has profound impact on the type theory; in particular, functional extensionality becomes provable, and is used throughout the paper for our covering spaces are defined as functions from the base type to the universe.

Two families of types indexed by the same type are equivalent if they are fiberwise equivalent, and a *fiberwise function* between two families is a family of functions between corresponding fibers. By the univalence axiom (and functional extensionality), fiberwise equivalence also implies the identification of two families of types.

For a family of types B indexed by a type A, an identification p in A from point a to point b will force an equivalence between corresponding fibers $B(a)$ and $B(b)$. The intuition is that a family of types indexed by B is a function from B to the universe $\mathcal{U}$; it will preserve identifications, and by the univalence axiom identifications in the universe are equivalences. The equivalence (as a function) is called *transport* and is written $\texttt{transport}^{x.B(x)}(p; a')$, meaning the transport of $a' : B(a)$ along $p : a =_A b$ across the family B to the fiber $B(b)$. It is also functorial in p in the sense that it sends reflexivity to identity equivalence and path



A 0-connected space. A 1-connected space.

Figure 2 Examples of connected spaces without structures above dimension 1.

Vertices represent the elements and edges represent the identification generators. The space on the left is not 1-connected because paths between points are not unique. Conversely, a 1-connected space is always 0-connected.

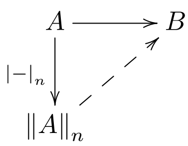
concatenation to equivalence composition.

2.4 Truncation and Connectivity

Truncation levels denote the dimension (iteration level of identification) *above which* a type is trivial: a type is at level -2 if it is *contractible*, which means it is equivalent to the unit type and is trivial at all dimensions, and a type is at level $(n+1)$ if its identification types lie at level n. It may seem odd that the level starts with -2, not 0, but it matches well with other theories such as groupoid theory; for example, there is a tight connection between types at level 1 and 1-groupoids.

A type at level -1 is called a *mere proposition*, where any two points are identified, and a type at level 0 is called a *set*, where any two parallel identifications are identified. Equivalences between sets are called *isomorphisms*.[7] It can be shown that the truncation levels form a cumulative hierarchy, in addition to the existing one based on their universe levels (which are suppressed in this paper).

An n-*type* [33, §7.1] is a type at truncation level n. The type `Set`, as mentioned above, is the type of all 0-types. The n-*truncation* of a type A is, intuitively, the *best* n-type approximation of the type A, written $\|A\|_n$, where the projection of $a : A$ into the truncation is written $|a|_n$. More precisely, $\|A\|_n$ is the n-type with the universal property that there is a unique extension of any function of type $A \to B$ to $\|A\|_n$ for any n-type B, as shown below. The n-truncation of an n-type is equivalent to the n-type itself.



for any n-type B.

Connectivity [33, §7.5] is the "dual" of truncation level in the sense that an n-connected type is trivial *below or at* the dimension n. See Figure 2 for a visualization of 0-connected and 1-connected spaces. In this work we critically rely on the fact that, for any two points in an n-connected type, there is an element in the $(n-1)$-truncation of the identification type between those two points. Technically, an n-connected type is defined to be a type whose n-truncation is contractible, meaning that it can only have non-trivial structures above dimension n.

[7] This follows the convention in [33, §2.4].

Because we will be working closely with many elements in the n-truncation of identification types, we may call such elements *n-truncated identifications* for short, or even *truncated identifications* if the truncation level is clear from the context.

Throughout this paper, some mere propositions are called *properties*, hinting they are mathematical properties whose witnesses are irrelevant (except ther existence), in contrast with mathematical *structures* which might carry non-trivial information.

2.5 Set Quotients

Let A be a type and $R : A \to A \to \mathcal{U}$ a family of types doubly indexed by A. We write A/R as the set quotient of A by R, $[a]$ as the equivalence class of $a : A$, and $\texttt{quot}(r)$ for $r : R(a, b)$ as a witness of $[a] =_{A/R} [b]$. The family R need not be an equivalence relation itself, but the set quotient in type theory effectively takes the reflexive, symmetric and transitive closure of R. Note that we did not require R to be a family of mere propositions as in the book [33] because in theory it made little difference and in practice it is convenient not to be concerned about truncation levels. Similarly, A is not required to be a set, even though the set quotient A/R always is.

2.6 Fundamental Groups and Truncated Identification

As mentioned earlier, iterated identification forms the structure of ∞-groupoids. The 0-truncation of identification thus behaves like ordinary groupoids, which reduce to groups if we only focus on some particular point [33]. More precisely, given a type A with a distinguished point a, the *fundamental group* of the type A at a, written $\pi_1(A, a)$, is the set $\|a = a\|_0$ along with concatenation as composition and (truncated) reflexivity as the unit. When the distinguished point a is clear from the context, we may omit the point and write $\pi_1(A)$ for short.

We will reuse the path concatenation and path inverse ($p \bullet q$ and p^{-1}) on 0-truncated identifications for a cleaner presentation; however, the distinction is still important and so we will mark every other use of truncation. In particular, the transport function on 0-truncated identifications is written with an additional subscript "0" as

$$\texttt{transport}_0^{x.B(x)}(p; a') : B(b)$$

where B is a family of 0-type indexed by A, p is a 0-truncated identification in $\|a = b\|_0$, and a' is a point in the fiber $B(a)$. It is important that the truncation level of p (which is 0 here) matches the truncation level of B so that we may apply the universal property of truncation.

2.7 Implicit Coercion

To further reduce notational clutter, we adopt implicit coercion when no confusion would occur. For example, a group may be implicitly coerced into its underlying set; an n-type, which in reality carries a proof of its truncation level, may drop the proof silently; and a pointed type may be coerced into its carrier. AGDA has limited support of coercion through instance arguments, but we did not use them in our mechanization except for numeric literals.

3 Comparison with Classical Definition

Our type-theoretic formulation appears quite different from the classical definition of covering spaces, and thus readers with the background in classical algebraic topology might wonder how this definition links to the classical one.

The situation is somewhat complicated because our construction lies in the type theory while the most common classical definition is expressed in point-set topology. We have an interpretation of the type theory into simplicial sets, and then geometric realization of simplicial sets into topological spaces, but not a direct interpretation into topological spaces yet to the best of our knowledge. A rigorous mathematical proof will involve interpreting our construction (Definition 1) into simplicial sets and then topological spaces, and is unfortunately beyond the scope of this paper. Instead, we will only give some intuition about the linkage in this section, and provide more internal evidence throughout the paper.

Here is a definition of covering spaces in terms of point-set topology [14, p. 29]:

▶ **Definition 2** (classical definition of covering space)**.** A *covering space* of a topological space A is a topological space C with a continuous surjective map $\pi : C \to A$ such that for each point $a \in A$ there is an open neighborhood U of a in A such that $p^{-1}(U)$ is a union of disjoint open sets, each mapped homeomorphically onto U by p.

The connection between the two kinds of covering spaces lies in several critical observations:

- Definition 1 defines a covering space as an A-indexed family of type F while Definition 2 focuses on a map p from C to A. To fit a covering space of the first kind, F, into the latter definition, one may choose the total space $\sum_{a:A} F(a)$ as C and the first projection as the map from C to A; the notion preimage $p^{-1}(a)$ is then replaced by the fiber $F(a)$. In general, type families and fibrations (for example p here) are equivalent and this connection is discussed in details in [33, §2.3]. We chose families over fibrations because it is easier to work with families of types inside the type theory.
- Next, the use of neighborhoods can be largely avoided because every space constructed by the standard geometric realization of a simplicial set is a CW complex and thus satisfies all local connectedness properties (for example local path-connectedness or semi-local simple connectedness). Moreover, every construct in type theory is continuous under this interpretation. Therefore, there is no need to mention local connectivity or continuity, because we cannot define any "bad" space in the type theory.
- Homeomorphism is weakened to homotopic equivalence because, again, it is impossible to distinguish homeomorphic but not homotopic objects inside the type theory.[8]

The real discrepancy is that the classical definition requires that the total space C (or $\sum_{a:A} F(a)$ from the first definition) to be non-empty and that p (or the first projection from $\sum_{a:A} F(a)$ to A) is surjective. This condition is needed for the universal covering to be *universal*, as we will discuss in Section 5; otherwise the empty space would be the universal covering space for any base type. However, without the non-emptiness or surjectivity requirement, the representation theorem (Theorem 4) does not have to rule out empty sets with actions; moreover, in a constructive setting there are many possible formulations of these conditions that are all classically equivalent but with different constructive content. Indeed, in Section 5 where we discuss universal covering spaces, we derive a pointedness condition that is constructively much stronger than (but classically equivalent to) mere non-emptiness. It is important to isolate the usage of non-emptiness or surjectivity to study their impact in constructive mathematics.

As a further justification, one can immediately prove the following lemma in the type theory when the base type A is the circle S^1:

[8] This does not take into account of the possibility of, for example, internalizing the entire set theory in the type theory and redoing the point-set topology. We assume a more direct interpretation into simplicial sets and then topological spaces.

▶ **Lemma 3.** *There is an equivalence between* $S^1 \to \texttt{Set}$ *and sets with an automorphism.*

Proof. (Omitted, but fully mechanized in the proof assistant AGDA as a separate lemma.) ◀

This lemma is a special case of the main theorem we will present in the next section.

4 Representation Theorem

The first main result of this paper is that covering spaces of a 0-connected, pointed space A are represented by *sets equipped with a group action of the fundamental group of* A, which is to say there is an equivalence between covering spaces and such sets. The intuition is that everything in homotopy type theory must respect identification, and the fact that the base type A is 0-connected indicates that there is a (-1)-truncated identification between any two points and thus a (-1)-truncated isomorphism between any two fibers. Therefore, it is represented by one copy of these isomorphic sets and a description of how they are isomorphic, encoded as an action of the fundamental group. See Figure 1 for examples of how a covering space is represented by a set with an action.

Formally, a set with a group action of G is called a *G-set*, a functor from the group G (treated as a category with one object and elements in G as morphisms) to the category of sets up to isomorphism; a *group set* is a G-set without the group G being specified. In type theory, a *G-set* is a record with the following components:

- $\texttt{El}$: a set.
- α: a (right) group action of type $\texttt{El} \to G \to \texttt{El}$.
- α-$\texttt{unit}$: a proof of the property that α preserves the group identity:

$$\prod_{x:\texttt{El}} \alpha(x, \texttt{unit}(G)) =_{\texttt{El}} x.$$

- α-$\texttt{comp}$: a proof of the property that α preserves the group composition:

$$\prod_{x:\texttt{El}} \prod_{g_1,g_2:G} \alpha(x, \texttt{comp}(G)(g_1, g_2)) =_{\texttt{El}} \alpha(\alpha(x, g_1), g_2).$$

The representation theorem is then about covering spaces being represented by $\pi_1(A, a)$-sets, which can be formally stated as follows:

▶ **Theorem 4** (representation by group sets)**.** *For any* 0*-connected type* A *with a point* a*, we have* $(A \to \texttt{Set}) \simeq \pi_1(A, a)\text{-}\texttt{Set}$.

Proof. The standard methodology to show equivalence in homotopy type theory is to establish two functions inverse to each other. That is, we want to establish two functions from covering spaces $A \to \texttt{Set}$ to group sets $\pi_1(A)\text{-}\texttt{Set}$ and vice versa, and show that the round-trips are the identity function.

The direction from covering spaces $A \to \texttt{Set}$ to group sets $\pi_1(A)\text{-}\texttt{Set}$ is relatively straightforward: the group set should capture a representative fiber with isomorphisms between fibers. Because the base type A is 0-connected, every fiber is equally qualified, and so we choose the one over the distinguished point a. Moreover, recall that the isomorphism forced by an identification in the base type, as discussed in Section 2, is the transport function. Putting these together, we can define a $\pi_1(A)$-set from a covering space $F : A \to \texttt{Set}$ by taking

$$\texttt{El} :\equiv F(a)$$

$$\alpha :\equiv \lambda x.\lambda g.\texttt{transport}_0^{x.F(x)}(g; x)$$

with properties α-`unit` and α-`comp` derived from functoriality of $\texttt{transport}_0$. The reason that we only have to record the automorphisms of $F(a)$ forced by loops at a (instead of all isomorphisms between all fibers) is because A is 0-connected; that is, every point in A is merely connected to a by a (-1)-truncated identification, and thus the automorphisms at a determine the isomorphisms between all fibers.

The other direction, from group sets $X : \pi_1(A)\text{-}\texttt{Set}$ to covering spaces, is more technically involved. A good guide is to focus on a group set generated from some covering space $F' : A \to \texttt{Set}$ through the above process; if the theorem is true, we should be able to recreate a covering space $F : A \to \texttt{Set}$ equivalent to F'. A key observation is that every point in any fiber of F' is a result of transporting some point in the fiber $F'(a)$ to that fiber, noting that X was defined to be $F'(a)$. Thus, one idea is to populate the new family F with *formal transports from X quotiented by the supposed functoriality of transports and the agreement with α*, in the hope to mimic the real $\texttt{transport}_0$ in F'. The formal definition is shown as follows; in the definition, the quotient relation $\sim_b$ can be seen as a succinct summary of the functoriality of transports and the agreement with α.

▶ **Definition 5** (reconstructed covering space)**.** Let A be a type with a point a and X be a $\pi_1(A, a)$-set with an action α. The *reconstructed covering space*, $F : A \to \texttt{Set}$, is defined as

$$F :\equiv \lambda b.(X \times \|a =_A b\|_0)/\!\sim_b$$

where the relation $\sim_b$ is defined as the least relation containing

$$\langle \alpha(x, \ell); p\rangle \sim_b \langle x; \ell \bullet p\rangle \text{ for any } x : X,\ \ell : \|a = a\|_0 \text{ and } p : \|a = b\|_0.$$

This completes the construction of the new covering space F.

The next step is to show that these two functions are indeed inverse to each other. However, in this paper we will only highlight the interesting part in proving the reconstructed covering space F is indeed equivalent to the original F'. Following the standard recipe of equivalence, two functions back and forth are needed for the equivalence between two covering spaces. The direction from F to F' is simply realizing the formal transports; that is, for any point $b : A$ and any representive $\langle x; p\rangle$ in the fiber $F(b)$ (defined as a set quotient), we have

$$\texttt{transport}_0^{x.F'(x)}(p; x) : F'(b)$$

because $x : X$, $p : \|a = b\|_0$ and $X :\equiv F'(a)$. One can then show this expression respects the quotient relation $\sim_b$ imposed on $F(b)$ in Definition 5. The other direction is somewhat unclear—given a point y in the fiber $F'(b)$, how shall we locate a point x in $F(a)$ and compute a truncated identification p such that y will be the result of transporting x along p?

Recall that the connectivity of A implies that there is a (-1)-truncated identification between any two points. That is, for any point $b : A$ we have a truncated identification $p : \|a =_A b\|_{-1}$. One attempt is then to transport y along the *inverse* of p to some point x in $F(a)$, for transporting x back along p should cancel the opposite transportation and recover y; the pair $\langle x; p\rangle$ in $F(b)$ then corresponds to y. The problem is that all the transportation and pairing demand 0-truncated identifications but p is a (-1)-truncated identification. In other words, there is a gap between the truncation level of the identifications from connectivity (-1) and that of the fibers of covering spaces (0), which prevents the application of the universal property of truncation.

Fortunately, such a truncation level gap can be filled by a constancy condition. We can show that different choices of identifications between a and b result in pairs related by the quotient relation imposed on $F(b)$, and then, by the following lemma, we can extend the above construction to (-1)-truncated identifications. The intuition is that if a function does not depend on the value of the input but only its existence, a (-1)-truncated input should suffice.

▶ **Lemma 6** (extension by weak constancy[9]). *Let A be a type and B a set. For any function $f : A \to B$ such that $\prod_{x,y:A} f(x) =_B f(y)$ there exists a function $g : \|A\|_{-1} \to B$ such that $f \equiv g \circ |-|_{-1}$.*

We will now carefully construct the function from F' to F sketched above, using this lemma. For any point $b : A$, we have a function $f_b : F'(b) \to (a =_A b) \to F(b)$ as

$$f_b :\equiv \lambda y.\lambda p.\Big[\Big\langle \texttt{transport}_0^{x.F'(x)}\Big(|p|_0{}^{-1}; y\Big); |p|_0\Big\rangle\Big],$$

which transports y to some point in $F(a)$. We want to show Lemma 6 applies to $f_b(y, -)$ for any $y : F'(b)$ so that a (-1)-truncated identification suffices. To satisfy the constancy condition in Lemma 6, it is sufficient to demonstrate that for any two identifications p, q of type $a =_A b$

$$\Big\langle \texttt{transport}_0^{x.F'(x)}\Big(|p|_0{}^{-1}; y\Big); |p|_0\Big\rangle \sim_b \Big\langle \texttt{transport}_0^{x.F'(x)}\Big(|q|_0{}^{-1}; y\Big); |q|_0\Big\rangle$$

where $\sim_b$ is the quotient relation of $F(b)$ and thus $f_b(y, p) =_{F(b)} f_b(y, q)$. This can be proved by the groupoid laws of identification and the definition of $\sim_b$; we have

$$\begin{aligned}
&\Big\langle \texttt{transport}_0^{x.F'(x)}\Big(|p|_0{}^{-1}; y\Big); |p|_0\Big\rangle \\
=\; &\Big\langle \texttt{transport}_0^{x.F'(x)}\Big(|p|_0{}^{-1}; y\Big); |p|_0 \cdot |q|_0{}^{-1} \cdot |q|_0\Big\rangle \\
\sim_b\; &\Big\langle \alpha\Big(\texttt{transport}_0^{x.F'(x)}\Big(|p|_0{}^{-1}; y\Big), |p|_0 \cdot |q|_0{}^{-1}\Big); |q|_0\Big\rangle \\
\equiv\; &\Big\langle \texttt{transport}_0^{x.F'(x)}\Big(|p|_0 \cdot |q|_0{}^{-1}; \texttt{transport}_0^{x.F'(x)}\Big(|p|_0{}^{-1}; y\Big)\Big); |q|_0\Big\rangle \quad \text{(by definition)} \\
=\; &\Big\langle \texttt{transport}_0^{x.F'(x)}\Big(|p|_0{}^{-1} \cdot |p|_0 \cdot |q|_0{}^{-1}; y\Big); |q|_0\Big\rangle \\
=\; &\Big\langle \texttt{transport}_0^{x.F'(x)}\Big(|q|_0{}^{-1}; y\Big); |q|_0\Big\rangle.
\end{aligned}$$

This means $f_b(y, -)$ is (pairwise) constant, and thus by Lemma 6 there exists an extension $g_{b,y} : \|a =_A b\|_{-1} \to F'(b)$ to the constant function $f_b(y, -)$. Putting these together, we have the following function of type $F'(b) \to F(b)$:

$$\lambda y.g_{b,y}(p(a, b))$$

where $p(x, y)$ is the (-1)-truncated identification between x and y derived from the connectivity of A. This concludes the two functions between $F'(b)$ and $F(b)$; the remaining parts of the equivalence proof are a routine calculation. ◀

The proof of Theorem 4 critically relies on Lemma 6, which provides a sufficient condition for establishing a function from types at lower truncation level to ones at higher level, which

---

[9] The word *weak* here indicates that we do not know the value in the codomain, which is weaker than other possible definitions of *constancy*. In particular, all functions from or to the empty type are weakly constant.

is usually impossible because of missing coherence conditions in codomains. The lemma asserts that constancy can fill in the gap so that there is a way to extend a function to truncated types. Nicolai Kraus *et al.* have significantly generalized the result and considered the cases from mere propositions to types at arbitrary levels; see [20–22]. The following is a proof of the special case (Lemma 6):

Proof of Lemma 6. Given a function f from A to B satisfying the constancy condition, construct the set quotient $A/\sim$ where

$$a \sim b :\equiv f(a) =_B f(b).$$

One can then show that the function f factors through $A/\sim$. Because $A/\sim$ is provably a mere proposition, the function f can be extended to the (-1)-truncation of A. The judgmental equality is derived from the computation rules of truncations and set quotients on points. ◀

There is also an alternative argument (provided by Steve Awodey) for Theorem 4 that proceeds as follows: In the context of $A \to \texttt{Set}$, because the codomain $\texttt{Set}$ is itself a 1-type (as the type of all n-types is an $(n+1)$-type [33, Theorem 7.1.11]), structures at dimension higher than 1 in the domain A are irrelevant, which means that $(A \to \texttt{Set}) \simeq (\|A\|_1 \to \texttt{Set})$. (This can also be argued from the universal property of the 1-truncation of A.) Moreover, the 1-truncation of a pointed, 0-connected type A can be represented by its fundamental group $\pi_1(A, a)$ where a is the point,[10] and so the type $\|A\|_1 \to \texttt{Set}$ is really the collection of functors from $\pi_1(A, a)$ (as a category) to $\texttt{Set}$, or simply $\pi_1(A, a)$-sets. However, this argument relies on several components that are still not available in the Agda development; in comparison our proof is more elementary.

5 Universal Covering Spaces

In addition to the representation theorem, we also mechanize several well-known properties about a special covering space, the *universal covering space*, which is intuitively the most general or the most "unfolded" covering space over a space. It has two equivalent definitions, one based on connectivity and one based on initiality (and hence the name *universal*). In addition to the two definitions, when the base type is 0-connected it is also represented by the fundamental group—which is itself a $\pi_1(A, a)$-set—through the representation theorem in Section 4; this argument was implicitly used in the calculation about the fundamental group of the circle in [26] and here we show a general result.

In this section the base type is fixed to be a type A with a distinguished point a.

▶ **Definition 7** (pointed covering space)**.** A *pointed covering space* is a covering space whose fiber over a is pointed.

▶ **Definition 8** (universal covering space)**.** A *universal covering space* is a pointed covering space whose total space is 1-connected.

The reason we stipulated a point in the specific fiber over the specific point is to make available a canonical choice among fiberwise equivalents. Considering the helix in Figure 3,

[10] The equivalence between $\|A\|_1$ and the Eilenberg-Mac Lane space $K(\pi_1(A, a), 1)$ was mechanized by Floris van Doorn in the library of LEAN [9, 11]. However, the authors are not aware of a published paper discussing this result in details.

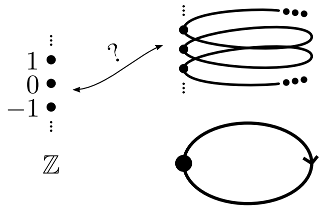


Figure 3 The lack of a canonical equivalence.

the universal covering space over the circle whose fundamental group is integers, there are multiple different equivalences between integers and any fiber of the helix, and there is no canonical choice—until we pin down a particular point in the helix and demand it be mapped to zero. To fit the definition of fiberwise equivalences, distinguished points of different covering spaces should be in the matching fibers, and thus we further demand the distinguished point lie in the fiber over the point a.

As hinted above, the following definition should be equivalent.

▶ **Definition 9** (alternative definition of universal covering space)**.** A *universal covering space* is a covering space which is initial in the category of pointed covering spaces with point-preserving fiberwise functions as morphisms.

The main observation to unify all these properties and simplify the proof is that the covering space consisting of 0-truncated identifications from the distinguished point

$$P :\equiv \lambda b.\|a =_A b\|_0$$

with its own distinguished point $|\texttt{refl}_a|_0$ in $P(a)$ *is* the universal covering space. This means that it suffices to show the covering space P is the one and only pointed covering space satisfying the two definitions of universal covering spaces, and that it is represented by the fundamental group. In fact, its correspondence to the fundamental group is trivial because its fiber over the distinguished point a is exactly (the underlying set of) the fundamental group, and it is not difficult to prove the group action is the concatenation. The rest of the section is dedicated to showing the equivalence of two definitions.

First, we will show P is the one and only 1-connected covering space.

▶ **Lemma 10.** *The total space of P is 1-connected.*

Proof. To show that the total space is 1-connected, by definition it suffices to show that the 1-truncation of $\sum_{b:A} P(b)$ is contractible, which means the 1-truncation is pointed and there is an identification to any point in that truncation. The truncated pair $|\langle a; |\texttt{refl}_a|_0\rangle|_1$ is clearly a point, and the identification between $|\langle a; |\texttt{refl}_a|_0\rangle|_1$ and some other truncated pair $|\langle b; p\rangle|_1$ can be established by applying truncation induction and identification induction on p, which states that it suffices to consider the case $p \equiv |\texttt{refl}_a|_0$ (and that $b \equiv a$). ◀

▶ **Lemma 11.** *Any pointed covering space whose total space is 1-connected is equivalent to P.*

Proof. Let F be a pointed covering space whose total space is 1-connected. Once again we will follow the recipe of equivalence by establishing two functions inverse to each other. The direction from P to F can be done fiberwise by transports; that is, for any $b : A$, we can define a function from $P(b)$ to $F(b)$ as evaluating the transport of the distinguished point in

$F(a)$ along the input in $P(b)$ (which is a truncated identification from a to b) to the fiber $F(b)$. Formally, it is

$$\lambda p.\mathtt{transport}_0^{x.F(x)}(p; a_F^*)$$

where a_F^* is the distinguished point of F over a. The other direction is to exploit the 1-connectivity: for any point y in the total space of F, there is a 0-truncated identification from the distinguished point $\langle a; a_F^* \rangle$ to y in the total space, which can then be "projected down" to the base type as a 0-truncated identification from the point a to the point over which y is. It can then be shown that these two functions are inverse to each other. ◀

Lemmas 10 and 11 tell us P is the only 1-connected universal covering space. Thus the remaining step is to prove that P is the initial object in the category up to homotopy. Note that we did not explicitly define the category but directly talked about its morphisms.

▶ **Lemma 12.** *For any pointed covering space F, there exists one and only one point-preserving fiberwise function from P to F.*

Proof. The existence is again by transporting the distinguished point of F along the points in P, which are themselves 0-truncated identifications. The uniqueness is by applying truncation induction and identification induction on points in the total space P, which suggests we only have to consider the case $|\mathtt{refl}_a|_0$, the distinguished point of P. However, a point-preserving function must send $|\mathtt{refl}_a|_0$ to the distinguished point of F, and thus all such functions must agree. ◀

Now we are ready to conclude this section with the following theorem:

▶ **Theorem 13.** *For any type A with a point a, the covering space $P :\equiv \lambda b.\|a =_A b\|_0$ of type A with $|\mathtt{refl}_a|_0$ as its distinguished point is* the *universal covering. It is also represented by $\pi_1(A, a)$ if A is 0-connected.*

Proof. The first statement directly follows Lemmas 10, 11 and 13. The second statement comes from the definition of P whose fiber over a is exactly the underlying set of $\pi_1(A, a)$. ◀

6 Discussion

In this paper we show that covering spaces, an important concept in homotopy theory, can be elegantly expressed in the new framework homotopy type theory, whose synthetic nature also makes possible Agda mechanization of length comparable to proofs on paper. The code is available at [6], and a snapshot that matches this paper is available at [7]. The development is broken into four files:

- `theorems/homotopy/CircleCover.agda`: Lemma 3.
- `theorems/homotopy/GroupSetsRepresentCovers.agda`: Theorem 4.
- `theorems/homotopy/AnyUniversalCoverIsPathSet.agda`: Lemmas 10 and 11.
- `theorems/homotopy/PathSetIsInitalCover.agda`: Lemma 12.

This paper is only the starting point of the study of covering spaces in homotopy type theory. There are still many properties unproven: for example, the representation theorem in classical theory is actually a correspondence between two categories, not just the objects. Also, the connectivity condition may be dropped if we replace fundamental groups by fundamental groupoids. Moreover, there are other possible generalizations such as n-covering spaces over a space as families of n-types indexed by that space (as a type), which to our knowledge do not immediately correspond to well-known structures in classical homotopy theory.

Acknowledgements

We are grateful to Carlo Angiuli, Steve Awodey, Spencer Breiner, Guillaume Brunerie, Ulrik Buchholtz, Daniel Grayson, Chris Kapulkin, Peter LeFanu Lumsdaine, Ed Morehouse and Jonathan Sterling for teaching the first author the classical theory and providing useful feedback on earlier presentations or drafts. We also thank anonymous reviewers for providing detailed and useful comments. The category-theoretic argument for the representation theorem was contributed by Steve Awodey.

References

1 The Coq proof assistant. URL: `https://coq.inria.fr/`.

2 Mathieu Anel, Georg Biedermann, Eric Finster, and Andre Joyal. The generalized Blakers–Massey theorem. In preparation, 2016.

3 Steve Awodey and Michael A. Warren. Homotopy theoretic models of identity types. *Mathematical Proceedings of the Cambridge Philosophical Society*, 146(1):45–55, 1 2009. doi:10.1017/S0305004108001783.

4 Andrej Bauer, Jason Gross, Peter LeFanu Lumsdaine, Michael Shulman, Bas Spitters, et al. The HoTT library. URL: `https://github.com/HoTT/HoTT`.

5 Guillaume Brunerie. *On the homotopy groups of spheres in homotopy type theory.* PhD thesis, Université Nice Sophia Antipolis, 2016. arXiv:1606.05916v1.

6 Guillaume Brunerie, Kuen-Bang Hou (Favonia), Evan Cavallo, Jesper Cockx, Christian Sattler, Chris Jeris, Michael Shulman, et al. Homotopy type theory in Agda. URL: `https://github.com/HoTT/HoTT-Agda`.

7 Guillaume Brunerie, Kuen-Bang Hou (Favonia), Evan Cavallo, Jesper Cockx, Christian Sattler, Chris Jeris, Michael Shulman, et al. Homotopy type theory in Agda. doi:10.6084/m9.figshare.5161546.

8 Ulrik Buchholtz and Egbert Rijke. The Cayley-Dickson construction in homotopy type theory. arXiv:1610.01134v1.

9 Ulrik Buchholtz, Floris van Doorn, and Jakob von Raumer. Homotopy type theory in Lean. To appear in the Proceedings of the 8th International Conference on Interactive Theorem Proving. arXiv:1704.06781v1.

10 Evan Cavallo. Synthetic cohomology in homotopy type theory. Master's thesis, 2015. URL: `http://www.cs.cmu.edu/~ecavallo/works/thesis.pdf`.

11 Leonardo de Moura, Soonho Kong, Jeremy Avigad, Floris van Doorn, and Jakob von Raumer. The Lean theorem prover (system description). In *Automated Deduction - CADE-25*, volume 9195 of *Lecture Notes in Computer Science*, pages 378–388, Switzerland, 2015. Springer International Publishing. doi:10.1007/978-3-319-21401-6_26.

12 Nicola Gambino and Richard Garner. The identity type weak factorisation system. *Theoretical Computer Science*, 409(1):94–109, 2008. doi:10.1016/j.tcs.2008.08.030.

13 Jean-Yves Girard. *Interprétation Fonctionnelle et élimination des Coupures de l'Arithmétique d'Ordre Supérieur*. PhD thesis, Université Paris 7, 1972.

14 Allen Hatcher. *Algebraic Topology.* Cambridge University Press, Cambridge, UK, 2002. URL: `http://www.math.cornell.edu/~hatcher/AT/ATpage.html`.

15 Kuen-Bang Hou (Favonia). Covering spaces in homotopy type theory. In Maria del Mar González, Paul C. Yang, Nicola Gambino, and Joachim Kock, editors, *Extended Abstracts Fall 2013: Geometrical Analysis; Type Theory, Homotopy Theory and Univalent Foundations*, pages 77–82, Cham, 2015. Birkhäuser. doi:10.1007/978-3-319-21284-5_15.

16 Kuen-Bang Hou (Favonia), Eric Finster, Daniel R. Licata, and Peter LeFanu Lumsdaine. A mechanization of the Blakers–Massey connectivity theorem in homotopy type theory.

In *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '16, pages 565–574, New York, NY, USA, 2016. ACM. doi:10.1145/2933575.2934545.

17 Kuen-Bang Hou (Favonia) and Michael Shulman. The Seifert-van Kampen theorem in homotopy type theory. In Jean-Marc Talbot and Laurent Regnier, editors, *25th EACSL Annual Conference on Computer Science Logic (CSL 2016)*, volume 62 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 22:1–22:16, Dagstuhl, Germany, 2016. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:http://dx.doi.org/10.4230/LIPIcs.CSL.2016.22.

18 Antonius J. C. Hurkens. A simplification of Girard's paradox. In *Typed Lambda Calculi and Applications*, volume 902 of *Lecture Notes in Computer Science*, pages 266–278. Springer, Berlin, Heidelberg, 1995. doi:10.1007/BFb0014058.

19 Krzysztof Kapulkin and Peter LeFanu Lumsdaine. The simplicial model of univalent foundations (after Voevodsky), 2012. arXiv:1211.2851v4.

20 Nicolai Kraus. The general universal property of the propositional truncation. arXiv:1411.2682v3.

21 Nicolai Kraus, Martín Hötzel Escardó, Thierry Coquand, and Thorsten Altenkirch. Notions of anonymous existence in Martin-Löf type theory. arXiv:1610.03346v1.

22 Nicolai Kraus, Martín Hötzel Escardó, Thierry Coquand, and Thorsten Altenkirch. Generalizations of Hedberg's theorem. In *Typed Lambda Calculi and Applications, 11th International Conference, TLCA 2013, Eindhoven, The Netherlands, June 26-28, 2013. Proceedings*, pages 173–188, 2013. doi:10.1007/978-3-642-38946-7_14.

23 Daniel R. Licata and Guillaume Brunerie. $\pi_n(s^n)$ in homotopy type theory. In Georges Gonthier and Michael Norrish, editors, *Certified Programs and Proofs: Third International Conference, CPP 2013*, pages 1–16, Cham, 2013. Springer International Publishing. doi:10.1007/978-3-319-03545-1_1.

24 Daniel R. Licata and Guillaume Brunerie. A cubical approach to synthetic homotopy theory. In *Proceedings of the 2015 30th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, LICS '15, pages 92–103, Washington, DC, USA, 2015. IEEE Computer Society. doi:10.1109/LICS.2015.19.

25 Daniel R. Licata and Eric Finster. Eilenberg-MacLane spaces in homotopy type theory. In *Proceedings of the Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 66:1–66:9, New York, NY, USA, 2014. ACM. doi:10.1145/2603088.2603153.

26 Daniel R. Licata and Michael Shulman. Calculating the fundamental group of the circle in homotopy type theory. In *Proceedings of the 2013 28th Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '13, pages 223–232, Washington, DC, USA, 2013. IEEE Computer Society. doi:10.1109/LICS.2013.28.

27 Peter LeFanu Lumsdaine. Higher inductive types: a tour of the menagerie. URL: https://homotopytypetheory.org/2011/04/24/higher-inductive-types-a-tour-of-the-menagerie/.

28 Peter LeFanu Lumsdaine. Weak ω-categories from intensional type theory. In *Typed Lambda Calculi and Applications*, volume 5608 of *Lecture Notes in Computer Science*, pages 172–187. Springer, 2009. doi:10.1007/978-3-642-02273-9_14.

29 Per Martin-Löf. An intuitionistic theory of types: Predicative part. In *Logic Colloquium '73, Proceedings of the Logic Colloquium*, volume 80 of *Studies in Logic and the Foundations of Mathematics*, pages 73–118. Elsevier, 1975. doi:10.1016/S0049-237X(08)71945-1.

30 Ulf Norell. *Towards a practical programming language based on dependent type theory.* PhD thesis, Chalmers University of Technology, 2007. URL: `http://www.cse.chalmers.se/~ulfn/papers/thesis.html`.

31 Charles Rezk. Proof of the Blakers-Massey theorem. Prepublished, 2015. URL: `http://www.math.uiuc.edu/~rezk/freudenthal-and-blakers-massey.pdf`.

32 Egbert Rijke. Homotopy type theory. Master's thesis, Utrecht University, 2012. URL: `http://hottheory.files.wordpress.com/2012/08/hott2.pdf`.

33 The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations for Mathematics.* Institute for Advanced Study, git commit hash g662cdd8 edition, 2013. URL: `http://homotopytypetheory.org/book`.

34 Benno van den Berg and Richard Garner. Types are weak ω-groupoids. *Proceedings of the London Mathematical Society*, 102(2):370–394, 2011. `doi:10.1112/plms/pdq026`.

35 Benno van den Berg and Richard Garner. Topological and simplicial models of identity types. *ACM Transactions on Computational Logic*, 13(1):3:1–3:44, 2012. `doi:10.1145/2071368.2071371`.

36 Vladimir Voevodsky. A very short note on homotopy λ-calculus, 09 2006. URL: `http://www.math.ias.edu/vladimir/files/2006_09_Hlambda.pdf`.

37 Jakob von Raumer. Formalization of non-abelian topology for homotopy type theory. Master's thesis, Karlsruhe Institute of Technology, 2015. URL: `http://von-raumer.de/msc-thesis.pdf`.

38 Michael A. Warren. *Homotopy theoretic aspects of constructive type theory.* PhD thesis, Carnegie Mellon University, 2008. URL: `http://mawarren.net/papers/phd.pdf`.