Calculating the Fundamental Group of the Circle in Homotopy Type Theory

Formalized in the Coq Unimath library

Jeroen van Heiningen[1]

Supervisor(s): Benedikt Ahrens[1]**, Kobe Wullaert**[1]

[1]EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 25, 2023

Name of the student: Jeroen van Heiningen
Final project course: CSE3000 Research Project
Thesis committee: Benedikt Ahrens, Kobe Wullaert, Neil Yorke-Smith

Abstract

The goal of this paper is to formalize the Fundamental Group of the Circle within Coq and the Unimath library, as described in the paper by Mr Licata and Mr Shulman.,[1] and show it is isomorphic to $\mathbb{Z}$. Fundamental groups are a powerful algebraic invariant for studying Homotopy theory, and provide deep, yet concise insights into the fundamental properties of a space.

1 Introduction

Homotopy type theory (HoTT) is a new field within mathematics and computer science (arising around 2006 with the work of Voevodsky and the researchers Awodey and Warren),[2] that leverages a recently discovered correspondence between Martin-Löf dependent type theory and the mathematical disciplines of category theory and homotopy theory.[3] Within this framework types can be regarded as spaces in homotopy theory, where each element of a type corresponds to a point in space. We regard witnesses of a $=_A$ b of some arbitrary type A as paths between a and b in the space A. Paths are said to be homotopic if these paths can be continuously deformed into one another. Continuous paths within a space can be separated into distinct equivalence classes under the relation of homotopy. This combined with the Univalence axiom which in a somewhat oversimplified manner states that homotopic equivalence (an isomorphism between types) can be regarded as being equivalent to identity,[4] allows for a much deeper notion of what constitutes equality compared to traditional type systems, or indeed mathematics more generally.

The subject of this paper is to report on the research conducted to formalize the fundamental group of the circle, as described in the research paper of Mr Licata and Shulman,[5] within the Coq Unimath library. The Fundamental group of the circle can be regarded as the homotopy-equivalence classes of loops around a circle, and is shown to be isomorphic to the set of all integers $\mathbb{Z}$. The central research question to be answered is as follows:

- Does computer-checking the contents of the published paper: "Calculating the Fundamental Group of the Circle in Homotopy Type Theory"[6] within the Unimath library of Coq, confirm the results discussed therein?

The remainder of this paper is structured as follows. In section 2 the basic mathematical definitions and concepts required to understand the subject of this paper are introduced, for explanatory reasons these concepts are given in to topological or classical homotopical terms, rather then those of HoTT. In section 3, the (type-theoretic) meaning of the fundamental group of the circle is elaborated upon, and in section 4, the proofs performed utilizing the Unimath library are explained and motivated upon.

2 Background

The goal of this paper is to demonstrate how to prove that the fundamental group of the circle (FGoC) is isomorphic to $\mathbb{Z}$ using Coq and the Unimath libary. To do this, an intuition of these concepts is required, therefore within this section some of the mathematical theory required is elaborated upon.

2.1 Mathematical Theory

Definition 2.1.[7] Let X be a topological space with a, b $\in$ X. A **path** in X from a to b is a continuous function f : $[0,1] \subset \mathbb{R} \to$ X such that f(0) = a and f(1) = b. The points a and b are called the endpoints.

Here the domain [0,1] can be thought of as representing moments in time, with 0 representing the starting point of f, and f(1) the end point.[8][9]

Definition 2.2.[10] A **loop** in a topological space X is a path f such that f(0) = x_0 = f(1) for some $x_0 \in$ X. The starting and ending point, $x_0 \in$ X, is called the **basepoint**.

Definition 2.3.[11] Let X be a topological space with two paths f_0 and f_1 that have endpoints $x_0, x_1 \in$ X. A **homotopy** from f_0 to f_1 is a family of paths f_t : $[0,1] \to$ X such that for all t $\in$ [0,1], f_t satisfies the following:

1. $f_t(0)$ = a and $f_t(1)$ = b.
2. The map F : $[0,1] \times [0,1] \to$ X defined by F(s,t) = $f_t(s)$ is continuous.

The paramater t of F(s,t) can be though of as determining which of the paths between f_0 and f_1 is taken, and the parameter s specifying the moment in time. The horizontal bars shown in the domain $I \times I$ in figure 1 correspond to different paths in the co-domain X, thus:

- F(s,0) = f_0(s)
- F(s,1) = f_1(s)
- F(0,t) = x_0
- F(1,t) = x_1

If there exists a homotopy between two paths f_0 and f_1, these two paths are called **homotopic** with the notation $f_0 \simeq f_1$,. The homotopy class of f, written as [f], is the equivalence

1. Michael Shulman Daniel R. Licata, "Calculating the Fundamental Group of the Circle in Homotopy Type Theory," 2013, https://arxiv.org/abs/1301.3443.

2. The Univalent Foundations Program, *Homotopy Type Theory: Univalent Foundations of Mathematics* (Institute for Advanced Study: https://homotopytypetheory.org/book, 2013).

3. Daniel R. Licata, "Calculating the Fundamental Group of the Circle in Homotopy Type Theory."

4. Univalent Foundations Program, *Homotopy Type Theory: Univalent Foundations of Mathematics*.

5. Daniel R. Licata, "Calculating the Fundamental Group of the Circle in Homotopy Type Theory."

6. Daniel R. Licata.

7. Samuel Dooley, "Basic algebraic topology: the fundamental group of the circle," August 2011, Definition 1.1. https://www.math.uchicago.edu/~may/VIGRE/VIGRE2011/REUPapers/Dooley.pdf.

8. David Ran, "An introduction to the funamental group," September 2015, Definition 3.1. http://math.uchicago.edu/~may/REU2015/REUPapers/Ran.pdf.

9. Univalent Foundations Program, *Homotopy Type Theory: Univalent Foundations of Mathematics*, p. 55.

10. Dooley, "Basic algebraic topology: the fundamental group of the circle," Definition 1.6.

11. Dooley, Definition 1.3.

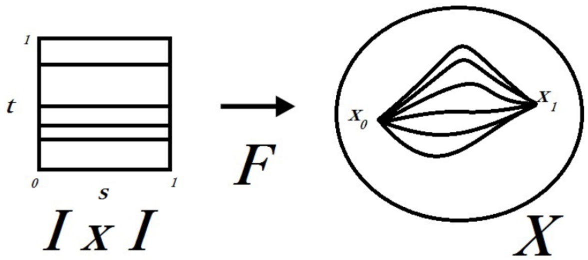

Figure 1: A path homotopy.[15] The paths shown in the codomain can be continuously deformed into one another and thus belong to the same equivalence class. If the space of X were to contain a hole (for example in the center), such that deforming the paths unto each other would cause their continuity property to be violated, the paths would belong to at least two distinct equivalence classes.

class of a path f under the homotopy equivalence relation .[12] Recall that an equivalence relation is a binary relation that is reflexive, symmetric and transitive, it is a means to partition an underlying set into disjoint equivalence classes. The elements of these classes are related under the binary relation given, and do not share this property with elements of other equivalence classes.[13]

Definition 2.4.[14][15]A group is a set G combined with a binary operator $\circ : G \times G \to G$ satisfying:

- (closure) for any two elements a and b in G, the result of their operation $a \circ b$, is also in G.
- (associativity) for all $a, b, c \in G$,
$$(a \circ b) \circ c = a \circ (b \circ c).$$
- (existence of identity element) there exists an element $e \in G$ such that,
$$a \circ e = a = e \circ a$$
- (existence of inverses) for each $a \in G$, there exists an $a' \in G$ such that,
$$a \circ a' = e = a' \circ a.$$

The fundamental group of a space is the group of equivalence classes under the relation of homotopy of the loops contained in the space relative to some basepoint x_0 with path composition as the as the group operation, denoted as $\pi_1(X, x_0)$.[16][17] Thus the Group laws as defined in definition 2.4 are fulfilled by path composition as follows:

- Closure - Given two loops in a space f, g: [0, 1] $\to$ X such that f(1) = g(0), their concatenation f $\cdot$ g can be defined by the formula:[18]
$$f \cdot g(s) = \begin{cases} f(2s), & 0 \leq s \leq \frac{1}{2}, \\ g(2s-1), & \frac{1}{2} \leq s \leq 1 \end{cases}$$
- Associativity - Path concatenation is associative, $(\alpha \circ \beta) \circ \gamma = \alpha \circ (\beta \circ \gamma)$.
- Identity element - The identity element is the constant loop (also called the trivial path), i.e. given $f : [0, 1] \to X$ and f(0) = x_0 = f(1) then for it to be a constant loop f is required stay fixed at x_0 during the entire [0,1] $\subset \mathbb{R}$ interval)
- Inverses - Every loop a has an inverse loop a^{-1} that can be concatenated, and simply represents the same loop in the opposite direction.[19]

The fundamental group of a space records information about that space, and can be used to tell spaces apart as well as give information about the basic shape, or holes, of the space.[20]

Definition 2.5.[21] Let $(G, *)$ and $(H, \circ)$ be groups, then:

- A **group homomorphism** f : G $\to$ H is a function such that for all $x, y \in G$ we have:
$$f(x * y) = f(x) \circ f(y)$$
- A **group isomorphism** is a group homomorphism which is a bijection.

An isomorphism can be regarded as an invertible homomorphism, because of this the mapping preserves information, i.e. you can revert the map and go back (a homomorphism however may lose information).

Definition 2.6. A topological space X is path-connected if for every x, y $\in$ X, there exists a continuous path f such that f(0) = x and f(1) = y.

Theorem 2.1. Let X be a path-connected topological space and consider some $x_0, x_1 \in X$. Then there exists an isomorphism between $\pi_1(X, x_1)$ and $\pi_1(X, x_0)$.

For reasons of brevity, the proof for this theorem has been omitted, however it may be found at the following reference.[22] Because of the preceding theorem, the fundamental group $\pi_1(X, x_0)$ is often simply denoted as $\pi_1(X)$, provided the space is path-connected.

Definition 2.7. A topological space X is *simply-connected* if it is path-connected and has a trivial fundamental group.

12. Dooley, "Basic algebraic topology: the fundamental group of the circle," Definition 1.3.
13. Mary Radcliffe, "Math 127: Equivalence Relations": p. 2, https://www.math.cmu.edu/~mradclif/teaching/127S19/Notes/EquivalenceRelations.pdf.
14. "Group Fundamentals," https://faculty.math.illinois.edu/~r-ash/Algebra/Chapter1.pdf.
15. Ran, "An introduction to the funamental group," P. 6.
16. Dooley, "Basic algebraic topology: the fundamental group of the circle," Theorem 1.9.
17. Daniel R. Licata, "Calculating the Fundamental Group of the Circle in Homotopy Type Theory," p. 6.
18. Dooley, "Basic algebraic topology: the fundamental group of the circle," Definition 1.7.
19. "Homotopy Theory in Homotopy Type Theory: Introduction," 2013, https://homotopytypetheory.org/2013/03/08/homotopy-theory-in-homotopy-type-theory-introduction/.
20. Univalent Foundations Program, *Homotopy Type Theory: Univalent Foundations of Mathematics*, p. 242.
21. "4.8 Homomorphisms and isomorphisms," https://www.ucl.ac.uk/~ucahmto/0007/_book/4-8-homomorphisms-and-isomorphisms.html.
22. Dooley, "Basic algebraic topology: the fundamental group of the circle," Theorem 1.11.

This means that, provided X is path-connected, then for some $x_0 \in X, \pi_1(X, x_0)$ there is only one equivalence class of loops (under the relation of homotopy), and thus the fundamental group is the trivial group with one element $[e_{x_0}]$. This occurs when a space has no *holes* that would prevent some loops from being deformed continuously into a single point. A simply-connected space has a strong connection to that of a contractible in HoTT, with it's fundamental group similarly being the trivial group.

Definition 2.8.[23]The n-dimensional sphere, or simply n-sphere, is the toplogical space given by the subset of the (n+1)-dimensional cartesian space $\mathbb{R}^{n+1}$ consisting of of all points x of radius r from the distance

$$S_r^n = \{x : \mathbb{R}^{n+1} \mid \ ||x|| = r\}.$$

For the purposes of this paper the radius of the of the n-sphere is irrelevant, so long as the "center" of the n-sphere contains a hole that violates continuity, as such it's notation is omitted within this paper. As an example, the n-sphere S^1 can be viewed as a circle in two-dimensional space, and S^2 as a sphere in three dimensional space, and so on.

2.2 Higher homotopy groups

The fundamental group of a space $\pi_1(X, x_0)$, is the first in a series of *homotopy groups* that provide additional information about a space. For, example $\pi_2(X, x_0)$ provides information about the two-dimensional structure of a space, and $\pi_3(X, x_0)$ about the 3-dimensional structure and so on.[24][25] One way to visualize this is as follows: if we take $\pi_k(X, x_0)$ with k=1 and some two-dimensional space X, you can regard this as one-dimensional loops on a 2D grid, where the loops can be deformed in any direction provided continuity is not violated. Similarly, k=2 with X being a 3-dimensional space, captures the 2-dimensional loops or spheres in X with basepoint x_0, provided it is understood that the "spheres" in this context are not spheres in the geometric sense, but rather can be deformed into any number of arbitrary shapes so long as they remain continuous.

In figure 2, it is shown that the homotopy groups of n-spheres are, in certain instances, isomorphic to the additive group of integers $\mathbb{Z}$. As an example, of an instance where this is not the case is for $\pi_1(S^2)$, visually this can be regarded as follows: if you take a 1-dimensional loop fixed at some basepoint x_0 you can deform the loop "around the sphere S^2" and contract it back to the basepoint, thus there is only one equivalence class of loops and the fundamental group is the trivial group of $[e_{x_0}]$.[26]

In the remainder of this paper, the results of the research project to show, utilizing the Unimath library, that the fundamental group of the *circle* $\pi_1(S^1)$ is isomorphic to $\mathbb{Z}$ is given.

	S^0	S^1	S^2	S^3	S^4	S^5	S^6	S^7	S^8
π_1	0	$\mathbb{Z}$	0	0	0	0	0	0	0
π_2	0	0	$\mathbb{Z}$	0	0	0	0	0	0
π_3	0	0	$\mathbb{Z}$	$\mathbb{Z}$	0	0	0	0	0
π_4	0	0	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\mathbb{Z}$	0	0	0	0
π_5	0	0	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\mathbb{Z}$	0	0	0
π_6	0	0	$\mathbb{Z}_{12}$	$\mathbb{Z}_{12}$	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\mathbb{Z}$	0	0
π_7	0	0	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\mathbb{Z}\times\mathbb{Z}_{12}$	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\mathbb{Z}$	0
π_8	0	0	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\mathbb{Z}_2^2$	$\mathbb{Z}_{24}$	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\mathbb{Z}$
π_9	0	0	$\mathbb{Z}_3$	$\mathbb{Z}_3$	$\mathbb{Z}_2^2$	$\mathbb{Z}_2$	$\mathbb{Z}_{24}$	$\mathbb{Z}_2$	$\mathbb{Z}_2$
π_{10}	0	0	$\mathbb{Z}_{15}$	$\mathbb{Z}_{15}$	$\mathbb{Z}_{24}\times\mathbb{Z}_3$	$\mathbb{Z}_2$	0	$\mathbb{Z}_{24}$	$\mathbb{Z}_2$
π_{11}	0	0	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\mathbb{Z}_{15}$	$\mathbb{Z}_2$	$\mathbb{Z}$	0	$\mathbb{Z}_{24}$
π_{12}	0	0	$\mathbb{Z}_2^2$	$\mathbb{Z}_2^2$	$\mathbb{Z}_2$	$\mathbb{Z}_{30}$	$\mathbb{Z}_2$	0	0
π_{13}	0	0	$\mathbb{Z}_{12}\times\mathbb{Z}_2$	$\mathbb{Z}_{12}\times\mathbb{Z}_2$	$\mathbb{Z}_2^3$	$\mathbb{Z}_2$	$\mathbb{Z}_{60}$	$\mathbb{Z}_2$	0

Figure 2: "Homotopy groups of spheres. The k^{th} homotopy group π_k of the n-dimensional sphere S^n is isomorphic to the group listed in each entry, where $\mathbb{Z}$ is the additive group of integers, and $\mathbb{Z}_m$ is the cyclic group of order m".[28]

3 The Fundamental Group of the Circle

The mathematical theory given in the preceding section mostly originated from Topology and classical homotopy theory, however in the following sections the theory, code and terminology will be derived from HoTT and the lexicon of the Unimath library and Coq programming language.

In HoTT, types are interpreted as a space, an element of some type as a point within that space, and an identity proof is regarded as a path between the corresponding points.[27] Therefore a Path is defined inductively with one constructor, an element of this type is regarded as a proof (or witness in Type theory) of propositional equality.[28] A path between two elements of a type can be constructed by writing either (a = b) or (paths a b) with a and b belonging to some arbitrary type.

As was stated previously, the fundamental group of a space X with some basepoint x_0, denoted as $\pi_1(X, x_0)$, is the group of equivalence classes under the relation of homotopy of loops from x_0 to itself, with path composition as the group operator.[29] In other words the elements of the group are the equivalence classes, with each equivalence class representing a distinct homotopy class of loops. "In type theory, this corresponds to the type ((paths x_0 x_0) : X), except for one caveat: for $\pi_1(X)$, the group has a *set* of elements, which are paths *quotiented by homotopy*. This means that any two paths that are homotopic are equal, but any non-trivial *structure* of paths between paths has been collapsed by quotienting."[30] However, in HoTT there may be paths between paths, i.e. (paths a b: (paths x_0 x_0)). Therefore, the type theoretic version of a loop, denoded by (paths x_0 x_0), has a closer cor-

23. "Sphere," https://ncatlab.org/nlab/show/sphere.
24. Univalent Foundations Program, *Homotopy Type Theory: Univalent Foundations of Mathematics*, ch 8.
25. "Homotopy Theory in Homotopy Type Theory: Introduction."
26. Univalent Foundations Program, *Homotopy Type Theory: Univalent Foundations of Mathematics*, p. 242.

27. "Just Kidding: Understanding Identity Elimination in Homotopy Type Theory," https://homotopytypetheory.org/2011/04/10/just-kidding-understanding-identity-elimination-in-homotopy-type-theory/.
28. Univalent Foundations Program, *Homotopy Type Theory: Univalent Foundations of Mathematics*.
29. Daniel R. Licata, "Calculating the Fundamental Group of the Circle in Homotopy Type Theory," p. 6.
30. Daniel R. Licata, p. 6.

respondence to the loop space $\Omega_1(X, x_0)$, which is the space of loops of X with basepoint x_0. In other words, when talking of a loop space $\Omega_1(X)$ of some arbitrary space X, loops can be interpreted as points. The loop space $\Omega_1(X)$ of some arbitrary space X, can be reduced to the fundamental group of X through a process called truncation. Therefore a proof that $\Omega_1(X)$ is isomorphic to $\mathbb{Z}$, automatically also shows that $\pi_1(X)$ is $\mathbb{Z}$.[31]

Thus there exists an isomorphism between the equivalence classes of $\Omega_1(X)$ and the integers. Visually, this can be best represented using the universal cover of the circle which is depicted as a helix above the circle, see figure 3. In this representation the winding number of loops, referring to the number of times a loop is taken in a particular direction, corresponds to a level on the helix above it. A negative integer corresponds to a clockwise winding and a positive integer with a counterclockwise winding. This correspondence can be shown to be an isomorphic mapping with the group operation in $\Omega_1(S^1)$ being concatenation of loops and the group operation in $\mathbb{Z}$ being addition of integers.[32][33]

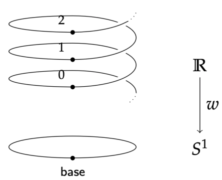

Figure 3: Universal cover of the circle can be represented as a helix projecting down to the circle. The map $w : \mathbb{R} \to \mathbb{S}^1$ sends every point on the helix to the point of the circle that is above it.[33] Note, that in the type-theoretic proof done in this paper the circle is mapped to the helix and not the other way around.

In homotopy theory, there is a standard proof of $\Omega_1(S^1)$ being isomorphic to $\mathbb{Z}$ using universal covering spaces as shown above, and the proof this paper follows can be seen as the type-theoretic version of this, using fibrations to represent the path lifting that is done in the classical proof.[34] Figure 3, shows the map w that maps the helix to the circle; the map w is a fibration, and the fiber over each point is isomorphic to the integers. This fibration is the *universal cover* of the circle[35]

The type-theoretic goal that the proofs given in the following sections built towards is to show that `paths base = base` is equivalent to `Cover base`, which as will be shown later is by definition Int.

4 Computer-checked proofs

In the following sections the formalization, utilizing the Coq Unimath library, of Mr Licata and Mr Shulman's paper[36] that was performed as part of the research project is given.

4.1 Defining the circle

It would be tempting to use Coq's regular inductive types to formulate the circle, however this is not possible. The reason for this is due to how the circle is defined[37] in homotopy theory, which in (erroneous) Coq notation would be as follows:

```
Inductive S1 : Type :=
    | base : S1
    | loop : base = base.
```

In other words a circle in Homotopy theory is defined as some point base, and a path from base to base (a loop).[38][39] Notice how the loop constructor has type 'base = base' which is an element of the identity type, however in Coq constructors of an inductive type should always output terms of that type. This is how Martin-Lof Type Theory (MLTT) is typically defined and is also how Coq (and most other dependent programming languages) are constructed. To circumvent this problem a more expansive version of inductive types are needed called Higher Inductive Types (HIT). In HoTT elements of a type can be thought of as points in a space and are what normal inductive types generate; however now we would like to also be able to construct a path (identity) and paths between paths within that space, which is what HIT allows us to do.[40] Unfortunately, Coq does not support the creation of HIT and thus a workaround is needed, as follows:

```
Private Inductive S1 : Type :=
    | base : S1.

Axiom loop : base = base.
```

Here, loop is given as an axiom and not defined as a constructor of S1. Axioms are statements that are assumed to be true without any accompanying proof and can lead to inconsistencies or contradictory statements if used improperly, but are a necessity here due to Coq's limitations. This also requires us to define our own elimination rules and take care to include their behaviour for both base and loop. However before these can be defined, we need to first define the functions maponpaths and transportf which state that functions and dependent functions respect equality (or more accurately paths retain their continuous property when (dependent) functions are applied to them).

4.2 Functions act functorially

Within HoTT functions act functorially on paths, meaning that applying functions to paths (identities) preserves their continuous property.[41] This is captured by the maponpaths definition located in Foundations.PartA.v:

31. Daniel R. Licata, "Calculating the Fundamental Group of the Circle in Homotopy Type Theory," p. 6.
32. Daniel R. Licata, p. 7.
33. Univalent Foundations Program, *Homotopy Type Theory: Univalent Foundations of Mathematics*, P. 243.
34. Daniel R. Licata, "Calculating the Fundamental Group of the Circle in Homotopy Type Theory," p. 7.
35. Daniel R. Licata, p. 7.
36. Daniel R. Licata.
37. Univalent Foundations Program, *Homotopy Type Theory: Univalent Foundations of Mathematics*, p. 174.
38. Univalent Foundations Program, p. 174.
39. Kajetan Sohnen, "Higher Inductive types in Homotopy Type Theory," July 2018, p.41, https://www.math.lmu.de/~petrakis/Soehnen.pdf.
40. Univalent Foundations Program, *Homotopy Type Theory: Univalent Foundations of Mathematics*, p. 167.
41. Univalent Foundations Program, p. 66.

```
Definition maponpaths {T1 T2 : UU} (f : T1 -> T2)
{t1 t2 : T1} (e: t1 = t2) : f t1 = f t2.
```

Which state that given a function f between two types (spaces) T1 and T2 and a path (identity) between two points, then applying f to these points causes their identity to be retained.

4.3 Dependent types as Fibrations

We still need to show that identity is preserved when the elements of this identity are applied to a dependent function. This is captured by the transportf definition in the Unimath library (variable names were changed for explanatory reasons).

```
Definition transportf {A : Type} (P : A -> Type)
{x y : A} (e : x = y) : P x -> P y
```

This function states that given some dependent function P : A $\to$ Type and two elements of A with a path between them, if the predicate P holds for one element then it also holds for the other, in fact it can be shown that P(x) and P(y) are equivalent.[42]
Topologically, the transportf function captures the notion of a "path lifting" operation in a fibration. With the example given above, the dependent function P : A $\to$ Type is a *fibration* with base space A, P(x) is the fiber over x, and $\sum_{x:A} P(x)$ is the *total space* of the fibration, as shown in figure 4.[43]

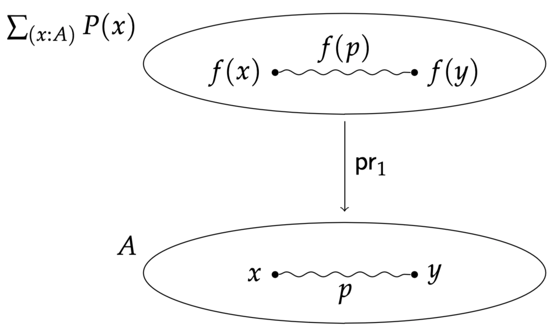

Figure 4: A fibration

In short, when a dependent function is applied to a path, this path is "lifted" to a different space P(x) and the continuity property of the paths is retained. In this paper the use of fibrations are typically referred to as transporting.

4.4 Elimination rules of the circle

We are now ready to define the elimination rules of the circle, for the terms base and loop of S1 as was defined in 4.1. The elimination rules for the circle express that it is the *initial* type with a point called base and a path from base to base (a loop), and given any other type with a point and a loop, the circle can be mapped onto it.[44] Recall, that in MLTT the recursor is just the non-dependent eliminator, and induction the dependent eliminator.[45]

```
Definition S1_recursion (A : Type) (b : A)
(l : b = b) : S1 -> A.
```

The recursor for S1 states that given:

1. A type A that values of S1 are to be mapped to.
2. An element 'b' of type 'A' to which base is to be mapped to.
3. A proof 'l' of a path from b to b, i.e. a witness that a loop exists within the space A.

then the circle can be mapped to this other arbitrary type A.

The induction rule for S1 is defined follows:

```
Definition S1_ind (P : S1 -> Type)
(b : P base) (l : (transportf P loop b) = b)
: forall x : S1, P x.
```

The induction principle for the natural numbers states that to prove a property (some predicate) of natural numbers, you need to show that it holds for zero (base case) and is preserved by the successor of an arbitrary natural number (inductive case). Similarly, to prove a property of points for S1, it suffices to prove that it holds for base and is maintained by going around the loop.[46] Thus S1_ind can be interpreted as follows, given:

1. A dependent type 'P' (a predicate) that represents the property to prove.
2. A proof that P holds for base.
3. A proof 'l' that asserts that the transport or lifting of S1 along the path 'loop' results in (b : P base).

then the circle can be mapped to a given arbitrary type (space) given by the fibration P.

4.5 Computation rules

The β-reduction rules, more commonly referred to as the computation rules, state how an eliminator acts on a constructor, and generally replaces more complicated expressions with simpler ones.[47][48] The β-reduction rules, as described in the paper by Mr Licata and Shulman[49], are given as axioms and are similarly implemented in this paper.

```
Axiom S1_rec_beta_loop : forall (A : Type)
(b : A) (l : b = b),
maponpaths (S1_rec A b l) loop = l.
```

The β-reduction rule of the recursor states that given some type A and a point 'b' and loop 'l' with basepoint 'b' within

42. Univalent Foundations Program, *Homotopy Type Theory: Univalent Foundations of Mathematics*, p. 67.
43. Univalent Foundations Program, p. 67.
44. "HoTTEST Summer School 2022: Agda Lecture 4," 2022, 55:00, https://www.youtube.com/watch?v=5JT3rhI_Pv0.
45. Univalent Foundations Program, *Homotopy Type Theory: Univalent Foundations of Mathematics*, p. 29.
46. Daniel R. Licata, "Calculating the Fundamental Group of the Circle in Homotopy Type Theory," p. 6.
47. Univalent Foundations Program, *Homotopy Type Theory: Univalent Foundations of Mathematics*, p. 26.
48. "nLab beta-reduction," 2016, https://ncatlab.org/nlab/show/beta-reduction.
49. Daniel R. Licata, "Calculating the Fundamental Group of the Circle in Homotopy Type Theory," p. 6.

this space, then mapping loop (of the circle) to the recursor of S1 is equal to the loop in the arbitrary type A.

Analogously, the β-reduction rule for the dependent eliminator as given below:

```
Axiom S1_ind_beta_loop : forall (P : S1 -> Type)
(b : P base) (l : transportf P loop b = b),
transport_section (S1_ind P b l) loop = l.
```

states that given:

- Some dependent function P, which assigns a type to each element of S1.
- A proof 'b' which states that the predicate 'P' holds for the basepoint of S1.
- An equality proof 'l' which states that the lifting of the loop of S1, with the fibration given by P, preserves the proof (b : P base).

then lifting loop (of the circle) with the dependent eliminator of S1 is equal to 'l'. the function transport_section used above is defined as follows:

```
Definition transport_section {X : UU} {P:X -> UU}
(f : forall x, P x) {x : X} {y : X} (e : x = y):
transportf P e (f x) = f y.
```

and states that given a proof that P holds for all elements x, then tranporting a path 'e' in X, along fibration P retains their continuity (their identity).

4.6 Homotopy Equivalence

Up to now paths existed within a space, however within HoTT there is also a notion of equality between different types if it can be proven that the types have the same fundamental structure; allowing for the creation of a path between types. This is captured by Voevodsky's univalence axiom:

Univalence Axiom: $(A = B) \simeq (A \simeq B)$[50]

which in a somewhat oversimplified manner says that isomorphic types are equal, with equal meaning a path, and isomorphic a homotopy equivalence.[51] A homotopy equivalence can be defined as follows: a function f : $A \to B$ is an isomorphism if there is a function g : $B \to A$ such that both composites f ∘ g and g ∘ f are pointwise equal to the identitiy, i.e. such that $f \circ g \sim id_B$ and $g \circ f \sim id_A$.[52]

In the proofs that are to follow the `encode` function corresponds to 'f', `decode` to 'g', `decode_encode` to $f \circ g$ and `encode_intToLoop` to $g \circ f$ establishing that the loop space of the circle is equivalent to Int, the functions mentioned are given in sections: 4.8, 4.9, 4.10, 4.11 respectively.

50. Univalent Foundations Program, *Homotopy Type Theory: Univalent Foundations of Mathematics*, p. 4.
51. Daniel R. Licata, "Calculating the Fundamental Group of the Circle in Homotopy Type Theory," p. 3.
52. Univalent Foundations Program, *Homotopy Type Theory: Univalent Foundations of Mathematics*, p. 72.

4.6.1 Homotopic equivalence integers

For the proof, S1 needs to be mapped to the integers, this requires there to be a basepoint x_0 and a loop with x_0 as it's starting and ending point. To do this a homotopy equivalence between int and itself is required, which can be shown by adding and subtracting 1 and showing that these operations are the mutual inverse of each other[53][54], which is captured by the following proofs:

```
Lemma succ_pred (z : Int): (succ (pred z)) = z.

Lemma pred_succ (z : Int): (pred (succ z)) = z.
```

Which can be proved by simple induction.

4.7 Universal Cover of the Circle

The lifting of the circle to the universal cover, see figure 3, is performed using the S1_rec eliminator, using the Cover function as given below:

```
Definition Cover : S1 -> Type :=
  fun x => S1_rec Type Int (weqtopaths (make_weq
  succ (isweq_iso succ pred pred_succ succ_pred)))
  x.
```

To map the circle to some other space by circle recursion, a point and a loop within that space has to be found. In this instance the 'space' is Type, the point is Int, and the loop is the successor/predecessor isomorphism on Int, which corresponds to a path from Int to Int.[55] The successor/predecessor isomorphism, utilizing the two lemmas proven in section 4.6.1, is implemented using the isweq_iso and make_weq functions, with successor indicating the forward direction of the (weak) equivalence that is created. The weqtopaths function constructs an actual path between spaces from the equivalence. It should be clear that `Cover base` is by definition Int.

To show that: tranporting along the cover one way corresponds to the successor operation (ascending a level in the helix); and transporting the other way, using the inverse loop, represents the predecessor operation (descending a level in the helix), the following proofs are performed:

```
Definition succEquivLoop (z: Int):
(transportf Cover loop) z = succ z.

Definition predEquivInvLoop (z: Int):
(transportf Cover (! loop)) z = pred z.
```

Both definitions are proved utilizing the functtransportf lemma:

```
Lemma functtransportf {X Y : UU} (f : X -> Y)
(P : Y -> UU) {x x' : X}
(e : x = x') (p : P (f x)):
transportf (λ x, P (f x)) e p = transportf P
(maponpaths f e) p.
```

53. Daniel R. Licata, "Calculating the Fundamental Group of the Circle in Homotopy Type Theory," p. 4.
54. Daniel R. Licata, p. 7.
55. Daniel R. Licata, p. 7.

taking Cover, the identity function, loop and 'z' as the corresponding arguments and showing that the resulting path can be concatenated to succ z or pred z. This is proven utilizing the S1_rec_beta_loop β-reduction rule showing that: `maponpaths (λ x:S1, S1_rec Type Int (weqtopaths (Int ≃ Int)) x) loop = (weqtopaths (Int ≃ Int))`, i.e. that loop can be mapped unto the integers. Then finally, that transporting with the identity function and a path from Int to Int applies the forward direction of the equivalence. In the case of the pred_succ Lemma, some basic additional reasoning about the inverses of equivalences is needed.

Next, the "encode" and "decode" functions are defined, which show that there is a correspondence between loops on the circle and the corresponding winding paths on the helix.

4.8 Encoding

The encode function transports some concatenation of loops on the circle to a level on the helix that corresponds to the loop composition that was supplied, e.g. loop ∘ (! loop) ∘ loop will be transported to the second level on the helix. The starting point needs to be an element of Cover base, for which Zero is a natural candidate.

```
Definition encode {x : S1} :
(base = x) -> Cover x :=
  (fun a => transportf Cover a Zero).
```

4.9 Decoding

Decoding does the reverse of the encoding step of the previous section, i.e. it computes the concatenation of loops that corresponds to a level on the helix. However, before the decode function can be defined, a function that returns the composition of loop concatenations for a particular Integer is required as follows:

```
Fixpoint posToLoopConcat (p : base = base)
(n : Positive) : (base = base) :=
    match n with
          | One => p
          | S n => posToLoopConcat p n @ p
        end.

Definition intToLoop (z : Int) : (base = base)
:= match z with
       | Neg n => posToLoopConcat (! loop) n
       | Zero => idpath base
       | Pos n => posToLoopConcat (loop) n
     end.
```

In addition, the following Lemma's are needed:

```
Definition transportf_arrow {A : Type}
{B C : A -> Type} {a a' : A} (p : a = a')
(f : B a -> C a) (y : B a'):
(transportf (fun x => B x -> C x) p f) y  =
transportf C p (f (transportf B (! p) y)).

transportf_id1 : forall (A : UU) (a x1 x2 : A)
(p : x1 = x2) (q : a = x1),
transportf (λ x : A, a = x) p q = q @ p
```

The former states that transporting with the family of `(fun x => B x -> C x)` is equivalent to pre-composing with transport at B (with the inverse path) and post-composing with transport at C. The latter states that transporting with the family of (paths M _) along the path p is equivalent to the composition of q @ p.[56] Both of which are easily proved using simple induction on the path hypotheses.

Finally, the actual decode function can be defined, which is the inverse of the encode function defined in section 4.8:

```
Definition decode {x : S1} : Cover x -> base = x.
```

To prove this, we proceed by circle induction taking `S1_ind (fun x' => Cover x' -> base = x') intToLoop _ x` as the arguments, which state if we want to prove that the property `Cover x' -> base = x'` holds for all x' element of S1, we take intToLoop as the base case and a proof that `transportf (fun (x' : S1) => Cover x' → base = x') loop intToLoop = intToLoop` holds. To prove the latter we use function extensionality to retrieve a third argument and applying the transportf_arrow the transportf_id1 lemma's, as well as predEqualsInvLoop defined in section 4.7. Reducing the goal term to `intToLoop (pred n) @ loop = intToLoop n` which can be proved by simple induction on n, and using the associativity, inverse and unit groupoid laws of paths.

In the next two sections, we establish that encode and decode are the mutual inverse of each other.

4.10 Encoding after Decoding

To show encode ∘ intToLoop can be done with the following definition:

```
Definition encode_intToLoop (z : Cover base):
encode (intToLoop z) = z.
```

which can be proved using simple induction on 'z', using the `succEquivLoop` lemma and the functoriality of transport captured by the following identity for the positive case of 'z' (the negative case is analogous):

```
transportf Cover (intToLoop (Pos p) @ loop)
Zero) =
(transportf Cover loop
(transportf Cover (intToLoop (Pos p)) Zero)))
```

4.11 Decoding after encoding

The (decode ∘ encode) composition can be shown as follows:

```
Definition decode_encode {x : S1} (a : base = x):
(decode (encode a)) = a.
```

and is a comparatively very simple proof relying on path induction, and a fuction that states that transporting a value along the trivial path retains it's reflexivity, i.e. `(P : X → UU) (x : X) (p : P x), transportf P (idpath x) p = p`.

56. Daniel R. Licata, "Calculating the Fundamental Group of the Circle in Homotopy Type Theory," p. 3.

4.12 Establishing the equivalence

Putting is all together, the proofs performed are all that is needed to construct a homotopy equivalence between `paths base base` and `Int`, establishing that $\Omega_1(S^1)$ is equivalent to Int with the following definition:

```
Definition S1_equivalentToInt : (paths base
base) ≃ Int := (make_weq encode (isweq_iso
encode decode decode_encode encode_intToLoop)).
```

5 Future work

Although, the proofs performed are sufficient to establish the equivalence of $\Omega_1(S^1)$ and Int,[57] two proofs given in the paper of Mr Licata and Mr Shulman remain unformalized due to time constraints encountered during the project. Namely, the `encode-decode` and `preserves-composition` proofs are uncompleted. The former establishes an equivalence between (paths base x) and (Cover x) for general (x : S1), and the latter establishes the fundamental group of S1 with Int as a *group*, and shows that the equivalence is a group homomorphism taking concatenation to addition as the operators.[58]

6 Responsible Research

Although, the topic of my research has little ethical or social considerations as such, given that it is a known theorem that the fundamental group of the circle is isomorphic to $\mathbb{Z}$, and formalized in proof-assistants multiple times.[59][60] It is still imperative to ensure the reproducibility of the results given in this paper. To that end the code base that was described within this paper will be published on a publicly available Unimath repository for inspection by third-parties. Peer-review of the code is also made easier by the fact that it builds upon a professionally constructed library, and a widely used proof-assistant. The truly critical pieces of code that are to be inspected are the 3 axioms that are included, which only represent three lines of code, and the type signatures of the proofs made.

7 Conclusions

The primary goal of the research paper is to formalize the fundamental group of the circle and show it is isomorphic to $\mathbb{Z}$, within the Coq programming language and Unimath library. The fact that the fundamental group of the cirlce is isomorphic to $\mathbb{Z}$ is a proven theorem in algebraic topology, and has been formalized within multiple proof assistants.[61][62] The steps taken within this research project lay the foundation for the formal verification of the above mentioned theorem utilizing the Unimath library, however a complete formalization was not completed within the duration of this research project, as the `encode_decode` and `preservers_composition` proofs are still to be completed.

Secondly, due to various known limitations of the Coq programming languages, certain steps had to be added as axioms, mainly due to Coq's lack of support of Higher Inductive Types.

The research and proofs performed during the research project can be expanded upon in numerous ways, most obviously by including homotopy groups of higher dimensions. In addition, the utilization of covering spaces as a means to prove the FGoC isomorphism to $\mathbb{Z}$ is one of many approaches, and different means of proving this may be worthwhile.

57. Daniel R. Licata, "Calculating the Fundamental Group of the Circle in Homotopy Type Theory," p. 9.
58. Daniel R. Licata, p. 9.
59. Univalent Foundations Program, *Homotopy Type Theory: Univalent Foundations of Mathematics*, P. 304.
60. Univalent Foundations Program, P. 283.
61. Mike Shulman, "A formal proof that π(S¹)=Z," 2011, https://homotopytypetheory.org/2011/04/29/a-formal-proof-that-pi1s1-is-z/.
62. Dan Licata, "A Simpler Proof that π(S¹) is Z," 2012, https://homotopytypetheory.org/2012/06/07/a-simpler-proof-that-%cf%80%e2%82%81s%c2%b9-is-z/.

References

"4.8 Homomorphisms and isomorphisms." https://www.ucl.ac.uk/~ucahmto/0007/_book/4-8-homomorphisms-and-isomorphisms.html.

Daniel R. Licata, Michael Shulman. "Calculating the Fundamental Group of the Circle in Homotopy Type Theory," 2013. https://arxiv.org/abs/1301.3443.

Dooley, Samuel. "Basic algebraic topology: the fundamental group of the circle," August 2011. https://www.math.uchicago.edu/~may/VIGRE/VIGRE2011/REUPapers/Dooley.pdf.

"Group Fundamentals." https://faculty.math.illinois.edu/~r-ash/Algebra/Chapter1.pdf.

"Homotopy Theory in Homotopy Type Theory: Introduction," 2013. https://homotopytypetheory.org/2013/03/08/homotopy-theory-in-homotopy-type-theory-introduction/.

"HoTTEST Summer School 2022: Agda Lecture 4," 2022. https://www.youtube.com/watch?v=5JT3rhI_Pv0.

"Just Kidding: Understanding Identity Elimination in Homotopy Type Theory." https://homotopytypetheory.org/2011/04/10/just-kidding-understanding-identity-elimination-in-homotopy-type-theory/.

Licata, Dan. "A Simpler Proof that π(S¹) is Z," 2012. https://homotopytypetheory.org/2012/06/07/a-simpler-proof-that-%cf%80%e2%82%81s%c2%b9-is-z/.

"nLab beta-reduction," 2016. https://ncatlab.org/nlab/show/beta-reduction.

Radcliffe, Mary. "Math 127: Equivalence Relations." https://www.math.cmu.edu/~mradclif/teaching/127S19/Notes/EquivalenceRelations.pdf.

Ran, David. "An introduction to the funamental group," September 2015. http://math.uchicago.edu/~may/REU2015/REUPapers/Ran.pdf.

Shulman, Mike. “A formal proof that $\pi(S^1)$=Z,” 2011. https://homotopytypetheory.org/2011/04/29/a-formal-proof-that-pi1s1-is-z/.

Sohnen, Kajetan. “Higher Inductive types in Homotopy Type Theory,” July 2018. https://www.math.lmu.de/~petrakis/Soehnen.pdf.

“Sphere.” https://ncatlab.org/nlab/show/sphere.

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