Homotopy Groups of Spheres in Homotopy Type Theory

Jinghui Yang
St Catherine's College
University of Oxford

A thesis submitted in partial fulfillment of the MSc in
Mathematical Sciences

Trinity 2022

Acknowledgements

It is a great pleasure to thank my supervisor, Professor Yakov Kremnitzer, for his guidance, helps, and suggestions in the thesis. His insightful comments really gave me much motivation, information, and confidence in this challenging area. I would also like to thank my friends Wenya Zhang, Yifan Wang, Xizheng Fang, and Juntao Ke for their support when I was at my low point. Finally, I would like to express my sincere gratitude to my family for their support in my life. Without their tremendous understanding and encouragement, it would be impossible for me to complete my study.

Contents

Chapter 1

Introduction

This dissertation aims to present the homotopy theory in homotopy type theory. In particular, we will compute several homotopy groups of spheres in this framework and develop the cohomological Serre spectral sequence originally proposed by Floris van Doorn [vD18] to discover more approaches to computing harder homotopy groups of spheres.

Invented by Bertrand Russell [Rus08], type theory served originally as a device to avoid the logical paradoxes like the Russell paradox. With the works by many people, especially Church [Chu40, Chu41] and Per Martin-Löf [ML75], type theory was developed into a formal system in mathematics based around λ-calculus. One influential system is Per Martin-Löf's type theory (also known as intuitionistic type theory), which was proposed as an alternative to set theory, as a foundation for constructive mathematics.

The philosophy of constructive mathematics is "proof by construction". Explicitly, to prove a proposition P is the same as finding a method to construct a shred of evidence ensuring P. Therefore, an algorithm can always be extracted from the proof. Theoretically, this enables one to use the computer to check proofs written in homotopy type theory. Various works on these "proof assistants" (computerized proof-writing programs) have been published as a new-rising field. For instance, see [Coq] for Coq, [Lea] for Lean, and [Agd] for Agda.

Homotopy type theory (HoTT) is a new branch of mathematics that interprets Martin-Löf type theory from a homotopical perspective. It was developed around 2006 by Awodey and Warren [AW09] and Voevodsky [Voe12]. Types in HoTT are regarded as *spaces*, and elements of types are regarded as *points* of spaces. Functions between types can be seen as continuous maps between spaces. The key idea of such homotopical interpretation is that identity $a = b$ of two elements in a type A can be regarded as a *path* from point a to point b in the space A. The advantage of these correspondences is that we can study the spaces and algebraic structures without establishing any point-set topology. It allows us to perform similar manipulations to homotopy theory in HoTT, like homotopies, loop spaces, path concatenations, etc., as long as they are homotopy invariant. Thus, a ∞-groupoid structure on types can be formalized, yielding that further homotopy theory can be established in HoTT. On the other hand, it provides a shortcut for proving propositions for all ∞-toposes. See [KL16] and [Shu19].

Homotopy Theory

Originated in algebraic topology, homotopy theory is a study of problems in which homotopies between maps get involved. Nowadays, it has become an independent discipline because the objects studied in homotopy theory can be defined independently of the underlying topology.

One fundamental problem in homotopy theory is to tell to what extent two spaces are "different" up to *homotopy equivalence*. The measurement of such "difference" is usually some algebraic invariants associated with the spaces, such as *homotopy groups*. If we can prove that homotopy groups of two spaces are different, we can say these two are distinct. So the algebraic invariants like homotopy groups provide much information about spaces, making them easier to distinguish efficiently.

A natural question is to ask what the homotopy groups of a space X look like, which we write as $\pi_k(X)$. Furthermore, since spaces can be built from spheres, calculating their homotopy groups becomes a central problem in classical homotopy theory. Unfortunately, it turns out to be a very hard problem. Even for the simplest case when $X = S^n$, results are barely known. But there are still numerous tools to tackle the problem, apart from some basic conclusions that can be derived directly. One of great interest is the spectral sequences. They are the generalizations of exact sequences and one of the most powerful computational tools in homotopy theory. A list of computations can be found in [WX10].

As such a powerful weapon in classical homotopy theory, one is curious if it can be formalized in HoTT. The hint is to consider Brown's representability in classical homotopy. It is homotopy invariant so that we can realize a similar construction in HoTT. In particular, we can define cohomology through this method in HoTT. This implies that under the suitable amendment, it is possible to develop the theory of spectral sequences in HoTT. Indeed, Floris van Doorn [vD18] implemented this idea in his doctoral thesis. Moreover, in light of the language of spectra, spectral sequences in HoTT are more fruitful in potential in the HoTT version of stable homotopy theory.

Content of the Thesis

The first two chapters, following the introduction, review the basic homotopy type theory and relevant results on homotopy groups of spheres in HoTT. The reference for these parts is [Uni13].

In Chapter 2, we start with the introduction to Martin-Löf type theory. We present several types of particular importance. Then we introduce some basic concepts in homotopy type theory, including the univalence axiom that will be presumed throughout the rest of the paper. After that, we briefly introduce the set theory in HoTT. We present the notion of contractible spaces and several propositions associated with them. Finally, we talk about the inductive types and higher inductive types. The natural numbers type $\mathbb{N}$ is an essential example of inductive types. The higher inductive types correspond to the spaces with cell structures in classical homotopy theory. For instance, the n-truncation is an operation to turn a general type into an n-truncated space. All these manipulations are standard in HoTT.

In Chapter 3, we define the homotopy groups of the space X as the 0-truncation of n-th loop space of X. We then sketch the proof of $\pi_1(S^1) = \mathbb{Z}$ and introduce two propositions on n-connected spaces to calculate $\pi_k(S^n)$ for $k < n$. Afterward, we discuss the fiber sequences and the associated long exact sequences of homotopy groups. A crucial example is the Hopf fibration $S^1 \to S^3 \to S^2$. The long exact sequence of homotopy groups associated to it induces the relations of $\pi_k(S^3)$ and $\pi_k(S^2)$, yielding $\pi_k(S^3) = \pi_k(S^2)$ for all $k \geq 3$. At last, we talk about the Freudenthal suspension theorem and deduce $\pi_3(S^2) = \mathbb{Z}$ from it.

In chapter 4, we first introduce the notion of Eilenberg-MacLane spaces $K(G, n)$ for an abelian group G, and define the ordinary cohomology of type X by the 0-truncation of the mapping space from X to $K(G, n)$. We show the cohomology of spheres has the desired results. Next, we introduce the language of spectra and define the generalized cohomology theory in this new language. Then we discuss the exact couples and spectral sequences in HoTT. As another crucial example, we introduce the generalized Atiyah-Hirzebruch spectral sequences and use them to prove the Serre spectral sequences. In the end, we calculate the cohomology of $K(\mathbb{Z}, 2)$ and ΩS^n. Inspired by EKMM [EKMM97], we present a conjecture on the form of the universal coefficient theorem in HoTT, and state how to use it to get $\pi_{n+1}(S^n) = \mathbb{Z}$, combining the cohomology of $K(\mathbb{Z}, 2)$ and ΩS^n.

Chapter 2

Backgrounds in Homotopy Type Theory

This chapter begins with the Martin-Löf (intuitionistic) type theory and its homotopy interpretation. This chapter serves as an introduction to homotopy type theory for those unfamiliar with the topic. For a detailed discussion, the readers are referred to the great book [Uni13].

2.1 Martin-Löf (Intuitionistic) Type Theory

Developed by Martin-Löf et al. [ML75], the Martin-Löf (intuitionistic) type theory is originally a modification of Alonzo Church's type system [Chu40] to formalize the constructive mathematics. Unlike the Zermelo-Fraenkel set theory, which has the deductive system of first-order logic with axioms, the Martin-Löf type theory has its own deductive system: it does not rely on any superstructures like first-order logic. Informally, a *deductive system* is a collection of *rules* for deriving things called *judgments*. There is only one kind of judgment in the deductive system of first-order logic (where the set theory is formulated): a given proposition has a proof. A rule of this logic is actually a process of "construction of proof". Thus, the sentence "judgment P has a proof" significantly differs from the proposition P itself. However, the deductive system on which the type theory is based has only one notion, the *types*. There are two kinds of judgments in type theory. One of them is $a : A$, pronounced as "a is an element of A". In this system, propositions are defined by particular types. Proving a proposition P is the same as constructing an element p (called a *witness*) in the proposition type P. This is the reason why the logic that type theory follows is called *constructive*. One should be warned that if a type A can be treated as a set, then "$a : A$" is a judgment, but $a \in A$ is a proposition. Although they share the same meaning, they exist at two different levels in logic.

The treatment of equality in type theory, which will bring much juice later, is another difference from set theory. For elements a, b in type A, we can define a new type "$a =_A b$". The subscript of "$=$" can be dropped if the underlying type is specified. When "$a = b$" is inhabited, we say a and b are *propositionally equal*. In other word, "$a = b$" is a proposition, but not a statement of judgment. To give the "equality judgment", we write "$a \equiv b : A$" (or simply "$a \equiv b$") to mean the *definitional equality*. Unlike the previous case, "$a \equiv b$" now represents "equality by definition" in our common sense.

Remark 2.1.1. To summarize, type theory is based on the deductive system with two kinds of judgments:

1. $a : A$, "a is an element of type A";

2. $a \equiv b : A$, "a and b are definitionally equal elements of type A".

In type theory, judgments may depend on *assumptions* of the form $x : A$, where x is a *variable* and A is a type. We can assume a propositional equality, like $p : x = y$, but we cannot assume a judgmental equality $x \equiv y$ since it is not a type. Logically, a judgmental equality is not a "proposition", so we cannot "prove" it. A judgment given at the outset is called an **axiom**.

The rules of the deductive system on which type theory is based is another thing of interest. Like the ones in the deductive system of first-order logic, the rules in our settings are what allow us to conclude one judgment from a collection of others. When we specify a type (a judgment), what we really do is specify the following rules:

1. **Formation Rules**: How to form new types of this kind.

2. **Introduction Rules (Constructor)**: How to construct elements of that type.

3. **Elimination Rules (Eliminator)**: How to use elements of that type. This is equivalent to how to define functions out of that type.

4. **Computational Rules**: How an eliminator acts on a constructor.

5. **Uniqueness Rules**: Express uniqueness of maps into or out of that type.

   Note. Uniqueness rules are often propositional. In this paper, we will only consider them if necessary.

The formation, introduction, and computational rules usually come with definitions. To gain an intuitive understanding of type theory, we will provide several examples that are essential for the future with extra explanations of their elimination rules. In our settings, the following fact is always presumed:

Fact 2.1.2. For any type A with $a : A$, there is a **reflexivity of** a given by $\mathsf{refl}_a : a =_A a$.

2.1.1 Non-dependent Function Types

Definition 2.1.3. Let A, B be types. Write $A \to B$ to denote the **non-dependent type of functions with domain** A **and codomain** B. Let a be an element of A. We can *apply* an element f in type $A \to B$ (called a **function** or **map**) on a to get the *value* of f at a. Denote it by $f(a)$. $f(a)$ is then an element in B. The elimination rules for non-dependent function types are function applications.

λ-abstraction is another way to state the introduction rules without introducing a name for the function. Suppose $f : A \to B$ is given by $f(x) \equiv \Phi$, where Φ is an expression that uses x. Then $\Phi : B$ is an element dependent on $x : A$. Write $\lambda(x : A).\Phi$ to indicate the same function f, i.e.

$$f(x) \equiv \lambda(x : A).\Phi : A \to B.$$

If a function f has two inputs $a : A$, $b : B$, then we can take one variable at a time to avoid using product types in §2.1.2. That is, we choose $\Phi \equiv f(a, b) : C$ for $f : A \to B \to C$. Rewritten in λ-abstraction,

$$f \equiv \lambda a.\lambda b.\Phi.$$

This is called **currying**.

2.1.2 Product Types

Types can be seen as elements of a "super-collection". In particular, we can regard all types are elements of a super-collection called the **universe** $\mathcal{U}$. Proceed with this process, we obtain a hierarchy of universes $\mathcal{U}_0 : \mathcal{U}_1 : \mathcal{U}_2 : \cdots$, where each $\mathcal{U}_i$ is an element of $\mathcal{U}_{i+1}$. We assume that the universes are **cumulative**; that is, if $A : \mathcal{U}_i$ for some i, then $A : \mathcal{U}_j$ for all $j \geq i$. For convenience, we write $A : \mathcal{U}$ to avoid mentioning the level i explicitly.

Definition 2.1.4. Let $A, B : \mathcal{U}$ be types. The **product type (or Cartesian product) of** A **and** B is $A \times B : \mathcal{U}$. The elements in $A \times B$ are $(a, b) : A \times B$ for $a : A$ and $b : B$.

Definition 2.1.5. The **unit type** $\mathbb{1} : \mathcal{U}$ is a nullary product type with only one unique element $* : \mathbb{1}$.

The elimination rules for the product type can be described via $\prod$-types (see §2.1.3). To be explicit, we define the **recursor** for the product type by

$$\mathrm{rec}_{A\times B} : \prod_{C:\mathcal{U}} (A \to B \to C) \to A \times B \to C,$$

with defining function $\mathrm{rec}_{A\times B}(C, g, (a, b)) \equiv g(a)(b)$, where $g : A \to B \to C$. So for every function $f : A \times B \to C$, we can represent it via the recursor

$$f((x, y)) \equiv \mathrm{rec}_{A\times B}(C, \lambda x.\lambda y.f).$$

If the codomain of f is allowed to vary based on the choice of $(x, y) : A \times B$, i.e. $f : \prod_{(z:A\times B)} C(z)$, then the recursor, which we renamed by **inductor** in this case, is

$$\mathrm{ind}_{A\times B} : \prod_{C:A\times B\to\mathcal{U}} \left(\prod_{x:A}\prod_{y:B} C(x, y)\right) \to \prod_{z:A\times B} C(z)$$

with defining function $\mathrm{ind}_{A\times B}(C, f, (a, b)) \equiv f(a)(b)$. Note that when C is constant, the inductor coincides with the recursor. Unless otherwise stated, we will only introduce the inductors for simplicity.

Example 2.1.6. Projection functions

$$\mathrm{pr}_1 : A \times B \to A, \quad \mathrm{pr}_1((a, b)) \equiv a$$
$$\mathrm{pr}_2 : A \times B \to B, \quad \mathrm{pr}_1((a, b)) \equiv b$$

can be represented by

$$\mathrm{pr}_1 \equiv \mathrm{rec}_{A\times B}(A, \lambda a.\lambda b.a), \quad \mathrm{pr}_2 \equiv \mathrm{rec}_{A\times B}(B, \lambda a.\lambda b.b).$$

2.1.3 $\prod$-types and $\sum$-types

Definition 2.1.7. $\prod$-types are also called **dependent function types**. Elements of them are functions whose codomain types can vary depending on elements of the domain to which function is applied (called **dependent functions**). We write $\prod_{(x:A)} B(x)$ to denote the $\prod$-type with domain A and codomains $B(x)$ dependent on $x : A$. Again, the elimination rule for $\prod$-types is function applications.

Example 2.1.8. Let $\prod_{(x:A)} B(x)$ be a $\prod$-type. If B is constant, then $\prod_{(x:A)} B(x) \equiv A \to B$, the non-dependent function type.

Definition 2.1.9. $\sum$-types are also called **dependent pair types**. It is a generalization of product type such that it allows the second component to vary depending on the choice of the first component. We write $\sum_{(x:A)} B(x)$ to denote the $\sum$-type with the first component A and the second component $B(x)$ dependent on $x : A$. The elements of $\sum_{(x:A)} B(x)$ are of the forms (a, b), where $a : A$ and $b : B(a)$.

Example 2.1.10. Let $\sum_{(x:A)} B(x)$ be a $\sum$-type. If B is constant, then $\sum_{(x:A)} B(x) \equiv A \times B$, the product type.

Like in §2.1.2, the elimination rule for $\sum$-types can be described similarly by inductor (recursor as a special case when codomain is constant):

$$\mathrm{ind}_{\sum_{(x:A)} B(x)} : \prod_{C:\sum_{(x:A)} B(x) \to \mathcal{U}} \left(\prod_{x:A} \prod_{y:B(x)} C(x, y) \right) \to \prod_{z:\sum_{(x:A)} B(x)} C(z)$$

with

$$\mathrm{ind}_{\sum_{(x:A)} B(x)}(C, f, (a, b)) \equiv f(a)(b).$$

Remark 2.1.11. The elimination rules for non-dependent cases are also called **recursion principle**, while for dependent cases they are called **induction principle**.

2.1.4 Coproduct Types

Definition 2.1.12. Let $A, B : \mathcal{U}$. The **coproduct type of A and B** is $A + B : \mathcal{U}$. Elements in $A + B$ are constructed by

- Left injection: $\mathrm{inl}(a) : A + B$ for $a : A$;
- Right injection: $\mathrm{inr}(b) : A + B$ for $b : B$.

Definition 2.1.13. The **empty type** $\mathbb{0} : \mathcal{U}$ is a nullary coproduct type with no inhabitants.

Induction principle for coproduct types can be stated via

$$\mathrm{ind}_{A+B} : \prod_{C:A+B\to\mathcal{U}} \left(\prod_{a:A} C(\mathrm{inl}(a)) \to \prod_{b:B} \mathrm{inr}(a) \right) \to \prod_{x:A+B} C(x)$$

with defining equations

$$\begin{aligned} \mathrm{ind}_{A+B}(C, f_0, f_1, \mathrm{inl}(a)) &\equiv f_0(a), \\ \mathrm{ind}_{A+B}(C, f_0, f_1, \mathrm{inr}(b)) &\equiv f_1(a), \end{aligned}$$

where $f_0 : A \to C$ and $f_1 : B \to C$ are functions chosen such that

$$f(\mathrm{inl}(a)) \equiv f_0(a), \quad f(\mathrm{inr}(b)) \equiv f_1(b),$$

for every function f we desire to construct out of $A + B$.

Example 2.1.14 (Boolean Type)**.** The Boolean type $\mathbb{2}$, also known as **0-dimensional sphere**, has exactly two elements $0, 1 : \mathbb{2}$. It can be constructed by coproduct of two copies of $\mathbb{1}$, i.e. $\mathbb{2} \equiv \mathbb{1} + \mathbb{1}$. The induction principle for $\mathbb{2}$ is a special case of the one for the coproduct type.

2.1.5 Identity Types

At the beginning of this chapter, we mentioned that the propositional equality of two elements $a, b : A$ is in fact a new type $a =_A b$, called an **identity type**. If we have an element p in this new type, we call it a **path between** a **and** b **in the space** A. If $a \equiv b : A$, then by Fact 2.1.2, there is an element $\mathrm{refl}_a : a = b$, called the **constant path at** a.

The induction principle for the identity types are also known as **path induction**. We will use this name for the rest of paper. To construct a dependent function f out of $a =_A b$, we choose a function $c : \prod_{(a:A)} C(a, a, \mathrm{refl}_a)$ for the family $C : \prod_{(a,b:A)}(a =_A b) \to \mathcal{U}$. Then $f : \prod_{(a,b:A)} \prod_{(p:a=_A b)} C(a, b, p)$ can be represented by

$$f(a, a, \mathrm{refl}_a) \equiv c(a).$$

In order words, given dependent functions

$$\begin{aligned} C &: \prod_{x,y:A} (x = y) \to \mathcal{U} \\ c &: \prod_{a:A} C(a, a, \mathrm{refl}_a), \end{aligned}$$

there exists a dependent function $\mathrm{ind}_{=_A}(C, c) : \prod_{(x,y:A)} \prod_{p:x=y} C(x, y, p)$ such that $\mathrm{ind}_{=_A}(C, c)(a, a, \mathrm{refl}_a) \equiv c(a)$ for every $a : A$.

2.2 Homotopy Type Theory

The central idea of homotopy type theory (HoTT) is that types can be treated as *spaces*, propositional equalities between elements of a type can be treated as *paths* (or *0-homotopies*), and elements of propositional equalities (as new types) can be treated as *(1-)homotopies*, and so on. So each type can be seen to have the structure of an ∞-groupoid.

All of basic constructions and axioms to derive a structure of the ∞-groupoid can be achieved by path induction in §2.1.5. For example, for every $x, y : A$ in a fixed type $A : \mathcal{U}$, there is a function $(x = y) \to (y = x)$ denoted $p \mapsto p^{-1}$, such that $\mathrm{refl}_x^{-1} \equiv \mathrm{refl}_x$. Call p^{-1} the **inverse** of p. This can be constructed by finding an element in the type $\prod_{(A:\mathcal{U})} \prod_{(x,y:A)} (x = y) \to (y = x)$. Let $D : \prod_{(x,y:A)} (x = y) \to \mathcal{U}$ be a family of types with $D(x, y, p) \equiv (y = x)$. By Fact 2.1.2, we have an element $d \equiv \lambda x.\mathrm{refl}_x : \prod_{(x:A)} D(x, x, \mathrm{refl}_x)$. Then by path induction, there exists an element $p^{-1} \equiv \mathrm{ind}_{=A}(D, d, x, y, p) : (y = x)$ for each $p : (x = y)$. Using the same method, we can get the **concatenation** of p and q, denoted $p \cdot q$, via the function

$$(x = y) \to (y = z) \to (x = z),$$

where $p : x = y$ and $q : y = z$ for $x, y, z : A$. The readers are encouraged to work out the details.

Lemma 2.2.1 ([Uni13], Lemma 2.1.4)**.** *The concatenation is associative. That is, for $x, y, z, w : A$ and $p : x = y$, $q : y = z$, $r : z = w$, we have $p \cdot (q \cdot r) = (p \cdot q) \cdot r$.*

We write $\Omega(A, a)$ to denote the type $a = a$ for $a : A$. It is called a **loop space of A at a**. Sometimes we include the basepoint of the loop space and write $\Omega(A, a) = ((a = a), \mathrm{refl}_a)$. In general, we can define the **n-fold iterated loop space** $\Omega^n(A, a)$ for $n \geq 0$ recursively by

$$\Omega^0(A, a) \equiv (A, a),$$
$$\Omega^{n+1}(A, a) \equiv \Omega^n(\Omega(A, a)).$$

Theorem 2.2.2 (Eckmann-Hilton, [Uni13], Theorem 2.1.6)**.** *The composition operation on the second loop space is commutative, i.e. $\alpha \cdot \beta = \beta \cdot \alpha$ for each $\alpha, \beta : \Omega^2(A, a)$.*

2.2.1 Fibrations

Let $f : A \to B$ be a function between two types. For every $x, y : A$, the action of f on paths between x and y is

$$\mathrm{ap}_f : (x =_A y) \to (f(x) =_B f(y)).$$

Moreover, $\mathrm{ap}_f(\mathrm{refl}_x) \equiv \mathrm{refl}_{f(x)}$. The existence of such function is ensured by Lemma 2.2.1, [Uni13]. In the dependent case when $f : \prod_{(x:A)} B(x)$, we need the following transport lemma to avoid the messy on distinct types.

Lemma 2.2.3 ([Uni13], Lemma 2.3.1)**.** *Let $P : A \to \mathcal{U}$ be a type family and $p : x =_A y$. Then there exists a function $p_* : P(x) \to P(y)$.*

Write $\mathrm{transport}^P(p, x)$ to denote the function p_* for $p : x =_A y$ starting at $x : A$ for the type family $P : A \to \mathcal{U}$. We may call P a **fibration** with **base space** A, $P(x)$ a **fiber** for each $x : A$, and $\sum_{(x:A)} P(x)$ the **total space**. So for $u : P(x)$, $\mathrm{transport}^P(p, x)(u)$ is the endpoint of the lifted path of p with starting point u.

Lemma 2.2.4 ([Uni13], Lemma 2.3.8)**.** *Let $f : \prod_{(x:A)} P(x)$ be a dependent function on type family $P : A \to U$. Then we have a map*

$$apd_f : \prod_{p:x=y} transport^P(p, x)(f(x)) =_{P(y)} f(y).$$

Lemma 2.2.5. *For a function $f : A \to B$ and a type family $P : B \to \mathcal{U}$, and any $p : x =_A y$ and $u : P(f(x))$, we have*

$$transport^{P\circ f}(p, u) = transport^P(ap_f(p), u).$$

Proof. By path induction, it suffices to assume y is x and p is refl_x. In this case, we have to prove $\mathrm{transport}^{P\circ f}(\mathrm{refl}_x, u) = u = \mathrm{transport}^P(\mathrm{ap}_f(\mathrm{refl}_x), u)$, which is true judgmentally by definition. □

2.2.2 Equivalences

Like the equivalence of elements in a type, we have a notion for the equivalence of functions and types in a universe.

Definition 2.2.6. Let $f, g : \prod_{(x:A)} P(x)$ be two dependent functions on a type family $P : A \to \mathcal{U}$. A **homotopy** from f to g is a dependent function of type

$$f \simeq g \equiv \prod_{x:A} (f(x) = g(x)).$$

One can check that homotopy is an equivalence relation on each dependent function type $\prod_{(x:A)} P(x)$. See Lemma 2.4.2 in [Uni13].

Definition 2.2.7. For a function $f : A \to B$, a **quasi-inverse** of f is a triple (g, α, β) consisting of a function $g : B \to A$ and two homotopies $\alpha : f \circ g \simeq \mathrm{id}_B$ and $\beta : g \circ f \simeq \mathrm{id}_A$. We denote the quasi-inverse of f by $\mathrm{qinv}(f)$.

Definition 2.2.8. The **equivalence of** f, denoted by $\mathrm{isequiv}(f)$, is a type that satisfies the following properties:

1. For each $f : A \to B$, there exists a function $\mathrm{qinv}(f) \to \mathrm{isequiv}(f)$.
2. For each $f : A \to B$, there exists a function $\mathrm{isequiv}(f) \to \mathrm{qinv}(f)$.
3. For every two elements $e_1, e_2 : \mathrm{isequiv}(f)$, $e_1 = e_2$.

One easy example to ensure the existence of $\mathrm{isequiv}(f)$ is given by

$$\mathrm{isequiv}(f) \equiv \left(\sum_{g:B\to A} (f \circ g \simeq \mathrm{id}_B) \right) \times \left(\sum_{h:B\to A} (h \circ f \simeq \mathrm{id}_A) \right).$$

Definition 2.2.9. An equivalence between types A, B is defined to be a function $f : A \to B$ with an inhabitant of $\mathrm{isequiv}(f)$. Write $A \simeq B$ for the type of equivalence from A to B. So $A \simeq B \equiv \sum_{(f:A\to B)} \mathrm{isequiv}(f)$.

Type equivalence is an equivalence relation on $\mathcal{U}$. The reader are referred to Lemma 2.4.12 in [Uni13] for details.

We can also define the equivalence fiberwise: for any two type families $P, Q : A \to \mathcal{U}$, the **fiberwise map** is $f : \prod_{(x:A)}(P(x) \to Q(x))$. f is a **fiberwise equivalence** if $f(x)$ is an equivalence for all $x : A$. We have the following theorem:

Theorem 2.2.10 ([Uni13], Theorem 4.7.7)**.** *$f : \prod_{(x:A)}(P(x) \to Q(x))$, defined as above, is a fiberwise equivalence iff $g \equiv \lambda w.(pr_1(w), f(pr_1(w), pr_2(w))) : \sum_{(x:A)} P(x) \to \sum_{(x:A)} Q(x)$ is an equivalence.*

2.2.3 Univalence Axiom

Univalence was introduced by Voevodsky [Voe12] to fill in the gaps of many desired theorems that were insufficient to prove simply using the Martin-Löf type theory.

Axiom 2.2.11 (Univalence)**.** For any types $A, B : \mathcal{U}$, there is an equivalence

$$(A =_{\mathcal{U}} B) \simeq (A \simeq B).$$

We always assume the universe $\mathcal{U}$ satisfying the univalence axiom. Write ua for an inhabitant of the type of quasi-inverse functions $(A \simeq B) \to (A =_{\mathcal{U}} B)$.

2.3 Sets

Types can be regarded as sets if they satisfy certain properties. Being a set makes a type easier to analyze.

Definition 2.3.1. A type P is a **mere proposition** if for every $x, y : A$, $x = y$.

Definition 2.3.2. A type P is a **set** if for every $x, y : A$, $p, q : x = y$, we have $p = q$. If A is a mere proposition, then it is a set.

Sets are also known as **0-types**, and mere propositions as (-1)**-types**. This definition can be generalized to n-type by adding propositional equalities on each pair of k-paths depending on $(k-1)$-paths for $k \leq n+1$. There is a way to decide whether a type is a set:

Theorem 2.3.3 (Hedberg's Theorem, [Uni13], Theorem 7.2.5)**.** *If a type P is such that for every $x, y : P$, either $x = y$ or $\neg(x = y)$ holds, then P is a set. Here $\neg(x = y) \equiv (x = y) \to \mathbb{0}$.*

One can apply the theorem to show $\mathbb{2}$, the natural numbers $\mathbb{N}$ (see §2.4.1), and the integers $\mathbb{Z}$ (see §2.4.2.4) are sets.

Lemma 2.3.4 ([Uni13], Lemma 3.3.2)**.** *Let P be a mere proposition, $x_0 : P$, then $P \simeq \mathbb{1}$. Thus any two mere propositions are equivalent.*

Definition 2.3.5. A type A is called **contractible** if there is an element $e : A$ (called the **center of contraction**) such that $a = x$ for every $x : A$.

Remark 2.3.6. Define the predicate

$$\mathrm{isContr}(A) \equiv \sum_{a:A} \prod_{x:A} (a = x).$$

Topologically, this type can be pronounced as path-connected space since, by definition, it can be translated into the phrase "A contains an element such that every other element of A equals to that one". Hence, it is easy to see that isContr(A) is a mere proposition for any type A.

The contractible types behave well like contractible spaces as expected.

Lemma 2.3.7 ([Uni13], Lemma 3.11.3)**.** *Let A be a type. The following are equivalent:*

1. *A is contractible.*

2. *A is equivalent to $\mathbb{1}$.*

3. *A is a mere proposition with an element $a : A$.*

Corollary 2.3.8. isContr(A) *is contractible iff A is contractible.*

The following lemmas are important in the proof of fact $\pi_1(S^1) = \mathbb{Z}$ (see §3.1).

Lemma 2.3.9 ([Uni13], Lemma 3.11.6, 3.11.8, 3.11.9)**.** *Let $P : A \to \mathcal{U}$ be a type family.*

1. *If $P(a)$ is contractible for all $a : A$, then $\prod_{(x:A)} P(x)$ is contractible.*

2. *If $P(x)$ is contractible for all x, then $\sum_{(x:A)} P(x) \simeq A$.*

3. *If A is contractible with the center a, then $\sum_{(x:A)} P(x) \simeq P(a)$.*

4. *For every type A and an element $a : A$, the type $\sum_{(x:A)}(a = x)$ is contractible.*

2.4 Inductive Types

Type X can be generated by some functions with codomain X. What is more, these functions can generate points of the new type, as well as paths and higher paths (e.g., n-homotopies). New types generated this way are called **inductive types**. We should note that inductive types are freely generated by a certain collection of functions; that is, elements in the inductive types are only obtained by repeatedly applying the functions in the collection. For example, $\mathbb{2}$ in Example 2.1.14 is generated by 0 and 1 (viewed as identity functions).

2.4.1 The Natural Numbers

The most important example of inductive types is the type of natural numbers, denoted by $\mathbb{N}$. It is generated by a base element $0 : \mathbb{N}$ and a successor function $\mathrm{succ} : \mathbb{N} \to \mathbb{N}$. If we adapt the usual notion, then $1 \equiv \mathrm{succ}(0)$, $2 \equiv \mathrm{succ}(2)$, $3 \equiv \mathrm{succ}(2)$, and so on.

The inductor for $\mathbb{N}$ is given by

$$\mathrm{ind}_{\mathbb{N}} : \prod_{C:\mathbb{N}\to\mathcal{U}} C(0) \to \left(\prod_{x:\mathbb{N}} C(x) \to C(\mathrm{succ}(x))\right) \to \prod_{x:\mathbb{N}} C(x).$$

So any function $f : \prod_{(x:\mathbb{N})} C(x)$ can be represented by

$$f(0) \equiv \mathrm{ind}_{\mathbb{N}}(C, f(0), f(\mathrm{succ}(n)), 0),$$
$$f(\mathrm{succ}(n)) \equiv \mathrm{ind}_{\mathbb{N}}(C, f(0), f(\mathrm{succ}(n)), \mathrm{succ}(n)).$$

We will adapt the usual notations $\leq, \geq, =$, etc., in $\mathbb{N}$ for the rest of the paper. The well-definedness of these notations is referred to §1.9, [Uni13].

2.4.2 Higher Inductive Types

Higher inductive types are a general schema of defining new types inductively. Compared to inductive types, the collection of generating functions of higher inductive types can generate not only points of the new type, but also paths and higher paths of it. We will refer to functions in the collection generating points in the new type as **point constructors**, and to paths and higher paths as **path constructors** and **higher path constructors**. One should be warned that applying functions on the path and higher path constructors is usually propositional, but not definitional.

2.4.2.1 The Intervals

A basic example of higher inductive types is the **interval** I. I is generated by

- two points $0_I, 1_I : I$, and
- a path $\mathrm{seg} : 0_I = 1_I$.

The induction principle of I is the following: given a type family $P : I \to \mathcal{U}$ with two points $b_0 : P(0_I)$, $b_1 : P(1_I)$, and a path $s : b_0 =^P_{\mathrm{seg}} b_1$, there exists a function $f : \prod_{(x:I)} P(x)$ such that $f(0_I) \equiv b_0$, $f(1_I) \equiv b_1$, and $\mathrm{apd}_f(\mathrm{seg}) = s$.

As we would expect, I behaves like the unit interval in classical topology.

Lemma 2.4.1 ([Uni13], Lemma 6.3.1)**.** *I is contractible.*

Lemma 2.4.2 ([Uni13], Lemma 6.3.2)**.** *Let $f, g : A \to B$ be two functions such that $f(x) = g(x)$ for every $x : A$. Then $f = g$.*

Corollary 2.4.3. *Let $X : \mathcal{U}$ with $x : X$. The type $\sum_{(f:I\to X)}(f(0_I) = x)$ is contractible.*

Proof. I is contractible, so $a = 0_I$ for all $a : I$. Hence, $f(a) = f(0_I)$ is inhabited by $\mathrm{ap}_f(\mathrm{seg})$ for every function $f : I \to X$. By Lemma 2.4.2, if f, g are two functions in $\sum_{(f:I\to X)}(f(0_I) = x)$, then $f = g$. Therefore, $\sum_{(f:I\to X)}(f(0_I) = x)$ is contractible. A center of contraction can be obtained by the constant function. □

2.4.2.2 Circles and Spheres

Another examples of higher inductive types that we will consider throughout the paper are the **sphere types**, denoted S^n. When $n = 1$, it is called a **circle type**. S^1 is generated by

- a point base : S^1 and
- a path loop : base $=_{S^1}$ base.

Note that loop is not equal to $\mathrm{refl}_{\mathrm{base}}$ (see Lemma 6.4.1, [Uni13]). Write $u =^P_p v$ to denote $\mathrm{transport}^P(p, u) = v$. Given a type family $P : S^1 \to \mathcal{U}$ such that it has an element $b : P(\mathrm{base})$ and a path $\ell : b =^P_{\mathrm{loop}} b$, there exists a function $f : \prod_{(x:S^1)} P(x)$ such that $f(\mathrm{base}) \equiv b$ and $\mathrm{apd}_f(\mathrm{loop}) = \ell$. This is the induction principle for S^1.

For n-sphere S^n, the generators are

- a point base : S^n and
- an n-loop $\mathrm{loop}_n : \Omega^n(S^n, \mathrm{base})$.

However, in general, it is hard to describe the induction principle using this definition because it involves higher paths. Thence we prefer the suspension in the next section.

2.4.2.3 Suspensions

Definition 2.4.4. The **suspension** of a type A is a new type ΣA generated by

- two points $N, S : \Sigma A$ and
- a function merid : $A \to (N =_{\Sigma A} S)$.

The induction principle for suspensions can be described in the same way as for S^1. Given type family $P : \Sigma A \to U$ with two points $n : P(N)$, $s : P(S)$ and a path $m(a) : n =^P_{\mathrm{merid}(a)} s$ for each $a : A$, there exists a function $f : \prod_{(x:\Sigma A)} P(x)$ such that $f(N) \equiv n$, $f(S) \equiv s$ and, $f(\mathrm{merid}(a)) = m(a)$. The advantage of this notion is the following:

Lemma 2.4.5 ([Uni13], Lemma 6.5.1)**.** $\Sigma \mathbf{2} \simeq S^1$.

Denote $S^0 \equiv \mathbf{2}$. Lemma 2.4.5 shows $\Sigma S^0 \simeq S^1$. This pattern continues: $\Sigma S^{n-1} \equiv S^n$. If we make the convention $S^{-1} \equiv \mathbf{0}$, then $\Sigma \mathbf{0} \simeq \mathbf{2}$. To see this definition agrees with previous definition in §2.4.2.2, one is referred to Lemma 6.5.3, [Uni13].

Like in classical homotopy theory, we have an adjoint pair (Ω, Σ).

Theorem 2.4.6 ([Uni13], Lemma 6.5.4)**.** *Let* (A, a_0) *and* (B, b_0) *be* ***pointed types****, i.e. types with points specified. Then*

$$Map_*(\Sigma A, B) \simeq Map_*(A, \Omega B),$$

where $Map_*(A, B)$ *is the types of* ***pointed function****s (i.e. functions preserving the basepoints) from* (A, a_0) *to* (B, b_0)*,*

$$Map_*(A, B) \equiv \sum_{f:A\to B} (f(a_0) = b_0).$$

Corollary 2.4.7. *For any type* B*, we have the following chain of equivalence:*

$$Map_*(S^n, B) \simeq Map_*(S^{n-1}, \Omega B) \simeq \cdots \simeq Map_*(S^1, \Omega^{n-1} B) \simeq Map_*(\mathbb{2}, \Omega^n B) \simeq \Omega^n B.$$

2.4.2.4 Quotients

Definition 2.4.8. Let A be a set, $R : A \times A \to \mathcal{U}$ be a family of mere propositions. The **set-quotient** (or simply **quotient**) of A by R, denoted A/R, is the higher inductive type generated by a function $q : A \to A/R$ satisfying

- for every $a, b : A$ with $R(a, b)$, $q(a) = q(b)$;
- for every $x, y : A/R$, $r, s : x = y$, we have $r = s$.

Usually, we demand R to have the equivalence relation. That is, the types $\prod_{(a:A)} R(a, a)$, $\prod_{(a,b:A)} R(a, b) \to R(b, a)$, $\prod_{(a,b,c:A)} R(a, b) \times R(b, c) \to R(a, c)$ are inhabited. The set-quotients behaves like quotient spaces in topology. For example, one can check q is surjective ([Uni13], §6.10).

Example 2.4.9 (Integers)**.** The type of integers $\mathbb{Z}$ can be defined as a set-quotient

$$\mathbb{Z} \equiv \mathbb{N} \times \mathbb{N}/\sim,$$

where $\sim$ is the family of mere propositions $\sim: \mathbb{N} \times \mathbb{N} \to \mathcal{U}$ with $\sim(m, n) \equiv m - n$. One can check integers defined in this way hold the usual properties as expected.

2.4.2.5 Truncations

Definition 2.4.10. Let $A : \mathcal{U}$. The **0-truncation** of A, denoted $\|A\|_0$, is the higher inductive type generated by a function $\| - \|_0 : A \to \|A\|_0$, such that $p = q$ for every $x, y : \|A\|_0$, $p, q : x = y$. So by definition, $\|A\|_0$ is a set for every type A.

Lemma 2.4.11 ([Uni13], Lemma 6.9.2)**.** *For any set B and type A, we have the equivalence*

$$(\|A\|_0 \to B) \simeq (A \to B).$$

The definition of 0-truncation can be generalized to n-truncation in a way similar to n-types (see §2.3). However, it is unwise to use this generalization since arguments on higher paths increase as n does. Instead, we prefer the following construction.

Definition 2.4.12. Let $A : \mathcal{U}$. For $n \geq -1$, the **n-truncation** of A, denoted $\|A\|_n$, is the higher inductive type generated by

- a function $\| - \|_n : A \to \|A\|_n$,
- a point (called *hub*) $h(r) : \|A\|_n$ for each $r : S^{n+1} \to \|A\|_n$, and
- a path (called *spoke*) $s_r(x) : r(x) = h(r)$ for each $r : S^{n+1} \to \|A\|_n$ and each $x : S^{n+1}$.

Lemma 2.4.13 (Cumulativity, [Uni13], Theorem 7.3.15)**.** *Let $A : \mathcal{U}$ be a type. For all $n, k \geq -1$, $k \leq n$, $\| (\|A\|_n) \|_k = \|A\|_k$.*

Truncations commute with the loop operator Ω.

Lemma 2.4.14 ([Uni13], Corollary 7.3.14)**.** *Let (A, a) be a pointed type, $n \geq -1$, $k \geq 0$. Then*

$$\|\Omega^k(A, a)\|_n = \Omega^k \|(A, a)\|_{n+k}.$$

The contractibility of n-truncation plays the role of n-connectedness in classical homotopy theory. In particular,

Definition 2.4.15. The type A is **n-connected** if $\|A\|_n$ is contractible.

Functions between n-connected types are defined via homotopy fibers.

Definition 2.4.16. Let $f : X \to Y$ with $y_0 : Y$. The **(homotopy) fiber** of f at y_0 is

$$\mathrm{fib}_f(y_0) \equiv \left(\sum_{x:A} f(x) = y_0 \right).$$

Example 2.4.17. Let $f : A \to \mathbb{1}$ be the constant map. Then $\mathrm{fib}_f(*) \simeq A$.

Definition 2.4.18. $f : A \to B$ is **n-connected** if for all $b : B$, the type $\|\mathrm{fib}_f(b)\|_n$ is contractible. So "type A is n-connected" is equivalent to "the function $A \to \mathbb{1}$ is n-connected".

Lemma 2.4.19 ([Uni13], Lemma 7.5.14)**.** *If $f : A \to B$ is n-connected, then it induces an equivalence $\|A\|_n \simeq \|B\|_n$.*

Chapter 3

Basic Results in Homotopy Groups of Spheres

As the name suggests, homotopy type theory has a natural connection with homotopy theory. The approach toward homotopy theory via notions in the last chapter is called *synthetic*. The algebraic structures on paths and higher paths are ensured by the ∞-groupoid structure on types. Especially, we can define the *monoids* and *groups* by

Definition 3.0.1. A **monoid** is a set (see §2.1.3) G with

- a multiplication function $m : G \times G \to G$, written $(x, y) \mapsto x \cdot y$, and
- a unit element $e : G$ such that
  1. for every $x : G$, $x \cdot e = x$, $e \cdot x = x$, and
  2. for every $x, y, z : G$, $x \cdot (y \cdot z) = (x \cdot y) \cdot z$.

G is a **group** if G is a monoid with

- an inversion function $i : G \to G$, written $x \mapsto x^{-1}$, such that for every $x : G$, $x \cdot x^{-1} = e$, $x^{-1} \cdot x = e$.

Two examples of groups are $(\mathbb{N}, +, 0)$ and $(\mathbb{N}, \times, 1)$. For any pointed type (A, a), its loop spaces type has the desired form of being homotopy groups even though it need not be a set. After 0-truncation, we can make (A, a) a well-defined group:

Definition 3.0.2. The ***n*-th homotopy group** of pointed type (A, a) is $\pi_n(A, a) \equiv \|\Omega^n(A, a)\|_0$. When $n = 1$, $\pi_1(A, a) \equiv \|\Omega(A, a)\|_0$ is called the **fundamental group** of (A, a).

The group structure on π_1 is defined by the concatenation of paths as multiplication. Note $\pi_n(A, a) = \pi_1(\Omega^{n-1}(A, a))$ for $n \geq 1$. So π_n has a group structure inherited from π. Moreover, by Eckmann-Hilton theorem 2.2.2, $\pi_n(A, a)$ is abelian when $n \geq 2$.

Our goal in this chapter is to study the algebraic properties of these homotopy groups of spheres, as one of the central tasks of synthetic homotopy theory.

3.1 $\pi_1(S^1)$

The study of the fundamental group of S^1 is the first step.

Theorem 3.1.1 (Homotopy Groups of S^1)**.**

$$\pi_n(S^1) = \begin{cases} \mathbb{Z} & , n = 1; \\ \mathbb{1} & , n > 1. \end{cases}$$

$n > 1$ is easy. After we show $\Omega(S^1) = \mathbb{Z}$, it is a set. So $\|\Omega^n(S^1)\|_0 = \|\Omega^{n-1}(\Omega(S^1))\|_0 = \|\Omega^{n-1}(\mathbb{Z})\|_0$ is contractible, implying the result. It suffices to show $\Omega(S^1) = \mathbb{Z}$. Recall that in classical homotopy theory, our strategy is to lift any loop in the circle S^1 to its universal cover $\mathbb{R}$. We can adapt this idea to a type-theoretic version as follows.

Step I: We define the "universal cover" of type S^1 through the map code : $S^1 \to \mathcal{U}$ with

- code(base) $\equiv \mathbb{Z}$, and
- $\mathrm{ap}_{\mathrm{code}}(\mathrm{loop}) = \mathrm{ua}(\mathrm{succ})$, where ua is defined in §2.2.3.

It is simple to see transporting with code takes loop to the successor function. That is,

Lemma 3.1.2. *$\mathrm{transport}^{\mathrm{code}}(\mathrm{loop}, x) = x + 1$, and $\mathrm{transport}^{\mathrm{code}}(\mathrm{loop}^{-1}, x) = x - 1$.*

Proof. By Lemma 2.2.5,

$$\begin{aligned} \mathrm{transport}^{\mathrm{code}}(\mathrm{loop}, x) &= \mathrm{transport}^{\mathrm{id}}(\mathrm{code}(\mathrm{loop}), x) \\ &= \mathrm{transport}^{\mathrm{id}}(\mathrm{ua}(\mathrm{succ}), x) \\ &= x + 1. \end{aligned}$$

The other equality follows in the same way. □

To describe the lifting, we introduce the notion

$$\text{encode} : \prod_{x:S^1} (\text{base} = x) \to \text{code}(x)$$

with $\text{encode}(p) \equiv \text{transport}^{\text{code}}(p, 0)$ for $0 : \mathbb{N}$ and a path p. So "encode" lifts a path to the universal cover.

Step II: By Lemma 8.1.12 in [Uni13], the type $\sum_{(x:S^1)} \text{code}(x)$ is contractible. Also by Lemma 2.3.9 (4), $\sum_{(x:S^1)}(\text{base} = x)$ is contractible. So Lemma 2.3.7 implies that $\sum_{(x:S^1)} \text{code}(x) \simeq \sum_{(x:S^1)}(\text{base} = x)$. Apply Theorem 2.2.10 to "encode" fiberwise, we show that $\Omega(S^1, \text{base}) = \mathbb{Z}$.

Thus, from the fact $\Omega(S^1, \text{base}) = \mathbb{Z}$ we conclude our proof for Theorem 3.1.1.

3.2 $\pi_{k<n}(S^n)$

The suspension operation increases connectedness as expected.

Lemma 3.2.1 ([Uni13], Theorem 8.2.1)**.** *If A is n-connected, then ΣA is $(n+1)$-connected.*

As a corollary, S^n is $(n-1)$-connected for all $n : \mathbb{N}$. This can be proved by induction, together with Theorem 3.1.1 and Lemma 3.2.1. Therefore, we have the following consequences:

Theorem 3.2.2 ([Uni13], Lemma 8.3.2)**.** *If A is n-connected, and $a : A$, then $\pi_k(A, a) = \mathbb{1}$ for all $k \leq n$.*

Combined with the fact that S^n is $(n-1)$-connected, we deduce

Corollary 3.2.3. *For all natural numbers $k < n$, $\pi_k(S^n) = \mathbb{1}$.*

3.3 Long Exact Sequences

Like in classical homotopy theory, we can define the long exact sequences of homotopy groups in homotopy type theory.

Definition 3.3.1. Let A and B be sets and $f : A \to B$ be a function. The **image** of f, denoted $\operatorname{im} f$, is the subset of B:

$$\operatorname{im} f \equiv \{b : B \mid f(a) = b \text{ for some } a : A\}.$$

If A, B are pointed with basepoints a_0, b_0, and f is pointed function, then the **kernel** of f, denoted $\ker f$, is the subset of A:

$$\ker f \equiv \{x : A \mid f(x) = b_0\}.$$

Note that any group is a pointed set with its unit element as a basepoint. So it is reasonable to call the pointed function between groups **homomorphism**.

Definition 3.3.2. An **exact sequence** of groups is a sequence of groups and homomorphisms

$$\cdots \to A_{n+1} \xrightarrow{d_n} A_n \xrightarrow{d_{n-1}} A_{n-1} \to \cdots$$

such that for every n, $\operatorname{im} d_n = \ker d_{n-1}$ as subsets.

As an example analogue to classical homotopy theory, the fiber sequence associated with $f : X \to Y$ induces a long exact sequence in homotopy groups. Before we continue, we introduce the notion of fiber sequences.

Definition 3.3.3. The **fiber sequence** of pointed map $f : X \to Y$ is the infinite sequence of pointed types and pointed maps

$$\cdots \xrightarrow{f_{n+1}} X_{n+1} \xrightarrow{f_n} X_n \xrightarrow{f_{n-1}} \cdots \xrightarrow{f_1} X_1 \xrightarrow{f_0} X_0,$$

where each term is defined recursively by

$$X_0 \equiv Y, \quad X_1 \equiv X, \quad f_0 \equiv f,$$

and

$$X_{n+1} \equiv \mathrm{fib}_{f_{n-1}}(a_{n-1}), \quad f_n \equiv \mathrm{pr}_1 : X_{n+1} \to X_n,$$

where a_n denoted the basepoint of X_n.

It is not hard to see $f_{n-1} \circ f_n = 0$ because each f_i is the projection onto the first factor. If we use the loop space operator Ω, and define $\Omega f : \Omega X \to \Omega Y$ on pointed types $(X, x_0), (Y, y_0)$ to be

$$(\Omega f)(p) \simeq f_0^{-1} \cdot f(p) \cdot f_0,$$

where $f_0 : f(x_0) = y_0$, subject to $\mathrm{refl}_{y_0} = f_0^{-1} \cdot f(\mathrm{refl}_{x_0} \cdot f_0) = (\Omega f)(\mathrm{refl}_{x_0})$, then we are able to show

Lemma 3.3.4 ([Uni13], Lemma 8.4.4)**.** *In Definition 3.3.3, the fiber sequence can be written as*

$$\cdots \xrightarrow{\Omega^2 f} \Omega^2 Y \xrightarrow{-\Omega f_2} \Omega F \xrightarrow{-\Omega f_1} \Omega X \xrightarrow{-\Omega f} \Omega Y \xrightarrow{f_2} F \xrightarrow{f_1} X \xrightarrow{f} Y,$$

where $F \equiv \mathit{fib}_f(y_0)$*, and negative signs mean composition with path inversion* $(-)^{-1}$*.*

Theorem 3.3.5 ([Uni13], Theorem 8.4.6)**.** *Let* $F \to X \xrightarrow{f} Y$ *with* $y_0 : Y$*,* $F \equiv \mathit{fib}_f(y_0)$ *be a slice of fiber sequence. It induces a long exact sequence of homotopy groups:*

$$\begin{aligned} \cdots \to \pi_k(F) \to \pi_k(X) \to \pi_k(Y) \to \cdots \\ \to \pi_1(F) \to \pi_1(X) \to \pi_1(Y) \to \pi_0(F) \to \pi_0(X) \to \pi_0(Y). \end{aligned}$$

One should be warned that the last three terms are not groups, and the groups are abelian only when $k \geq 2$. An application of the long exact sequence is the **Hopf fibration**.

Theorem 3.3.6 (Hopf fibration, [Uni13], Theorem 8.5.1)**.** $\mathrm{fib}_f(\mathit{base}) = S^1$ *for* $f : S^3 \to S^2$ *with* $\mathit{base} : S^2$*.*

By Theorem 3.3.5, it induces a long exact sequence

$$\begin{aligned} &\cdots \to \pi_k(S^1) \to \pi_k(S^3) \to \pi_k(S^2) \to \cdots \\ &\to \pi_1(S^1) \to \pi_1(S^3) \to \pi_1(S^2) \\ &\to \pi_0(S^1) \to \pi_0(S^3) \to \pi_0(S^2). \end{aligned} \tag{1}$$

Since, by Corollary 3.2.3 and Theorem 3.1.1, $\pi_k(S^3)$ is trivial for $k < 3$, $\pi_k(S^2)$ is trivial for $k < 2$, $\pi_k(S^1)$ is trivial for all k except for $k = 1$. We can simplify the long exact sequence (1) to

$$\begin{aligned} &\cdots \to \mathbb{1} \to \pi_k(S^3) \to \pi_k(S^2) \to \cdots \\ &\to \mathbb{1} \to \pi_3(S^3) \to \pi_3(S^2) \\ &\to \mathbb{1} \to \mathbb{1} \to \pi_2(S^2) \\ &\to \mathbb{Z} \to \mathbb{1} \to \mathbb{1} \to \mathbb{1}. \end{aligned} \tag{2}$$

Hence, we conclude

Corollary 3.3.7. $\pi_k(S^3) = \pi_k(S^2)$ *for all* $k \geq 3$*, and* $\pi_2(S^2) = \mathbb{Z}$*.*

3.4 The Freudenthal Suspension Theorem

Theorem 3.4.1 (The Freudenthal suspension theorem, [Uni13], Theorem 8.6.4)**.** *Let X be an n-connected and pointed type, $n \geq 0$. Then the map $\sigma : X \to \Omega\Sigma X$ is $2n$-connected.*

The proof relies on the contractibility of a certain type family in the form similar to "code" defined in §3.1. Applying Lemma 2.4.19, we obtain an equivalence:

Corollary 3.4.2. *Let X be defined as in Theorem 3.4.1. Then $\|X\|_{2n} = \|\Omega\Sigma X\|_{2n}$.*

The importance of this theorem lies in proving the stability of the homotopy group of spheres:

Theorem 3.4.3. *Let $k \leq 2n-2$, then $\pi_{k+1}(S^{n+1}) = \pi_k(S^n)$.*

Proof. By corollary 3.4.2 and the fact that S^n is $(n-1)$-connected, $\|\Omega\Sigma(S^n)\|_{2n-2} = \|S^n\|_{2n-2}$. From Lemma 2.4.13, $\|\Omega\Sigma(S^n)\|_k = \|S^n\|_k$ for all $k \leq 2n-2$. We can now compute

$$\begin{aligned}
\pi_{k+1}(S^{n+1}) &= \|\Omega^{k+1}(S^{n+1})\|_0 = \|\Omega^k\left(\Omega(S^{n+1})\right)\|_0 \\
&= \|\Omega^k\left(\Omega\Sigma(S^n)\right)\|_0 \\
&= \Omega^k\left(\|\Omega\Sigma(S^n)\|_k\right) && \text{(by Lemma 2.4.14)} \\
&= \Omega^k\left(\|S^n\|_k\right) \\
&= \|\Omega^k(S^n)\|_0 = \pi_k(S^n).
\end{aligned}$$

□

As the first corollary, we are able to show by applying Theorem 3.4.3 to Corollary 3.3.7,

Theorem 3.4.4. $\pi_n(S^n) = \pi_2(S^2) = \mathbb{Z}$ *for all* $n \geq 1$.

Thus, again by Corollary 3.3.7, we compute

Corollary 3.4.5. $\pi_3(S^2) = \mathbb{Z}$.

Chapter 4

Spectral Sequences in Homotopy Type Theory

The computational methods in Chapter 3 are crucial in homotopy theory, but they are insufficient in many situations, for example, $\pi_4(S^3)$. Guillaume [Bru16] showed in his doctoral thesis that $\pi_4(S^3) = \mathbb{Z}/2$ by applying the Gysin sequence to the fibration of $\mathbb{CP}^2$ defined by Hopf construction and then finding the Hopf invariant of the Hopf map $S^3 \to S^2$. In general, it is very hard to decide on a homotopy group of spheres. More handy tools are needed, and spectral sequences are one of them.

Spectral sequences are a generalization of long exact sequences, and become a powerful tool in the computation of homotopy groups. Relating cohomologies and homologies, they provide a much clearer schema of calculation that we can get a hand on. This chapter will give a basic construction of the Serre spectral sequences for cohomology. The main references for this chapter are Floris van Doorn in his doctoral thesis [vD18] and Mike Shulman in his post [Shu13].

The first task is to set up the cohomology theories. However, unlike in classical homotopy theory, where cohomology theories can be acquired via singular cohomologies, we cannot construct similar objects in HoTT. This is because taking the singular cochains requires the triangulation of the underlying topological space, which is not invariant under homotopy equivalence. Instead, we adopt the idea of Brown representability to define the (reduced) cohomology theories directly.

4.1 Cohomology

Definition 4.1.1. Define the predicate is-n-type : $\mathcal{U} \to \mathcal{U}$ for $n \geq -2$ by recursion as follows:

$$\text{is-}n\text{-type}(X) \equiv \begin{cases} \text{isContr}(X) & , n = -2 \\ \text{is-}n'\text{-type}(X) & , n = n' + 1 \end{cases}$$

If is-n-type(X) is inhabited, then We say X is an n**-type**, or X is n**-truncated**.

Lemma 4.1.2 ([Uni13], Corollary 7.1.5)**.** *If $X \simeq Y$ and X is an n-type, then so is Y.*

Lemma 4.1.3 ([Uni13], Theorem 7.1.7)**.** *If X is n-truncated, then it is $(n+1)$-truncated for each $n \geq -2$.*

The core of defining a cohomology theory is the **Eilenberg-MacLane spaces** $K(G, n)$, where G is a group and $n : \mathbb{N}$. The concrete construction of this higher inductive type can be found in §3 and §5 in [LF14]. We only focus on the following facts.

Fact 4.1.4 ([LF14], §5)**.** Let $K(G, n)$ be a Eilenberg-MacLane space, where G is a group and $n : \mathbb{N}$. Then

1. $K(G, n) \equiv \|\Sigma K(G, n)\|_{n+1}$ for $n \geq 1$ and G abelian;
2. $K(G, n)$ is the unique n-truncated pointed type X with $\pi_n(X) = G$ and $\pi_k(X) = 0$ for $k \neq n$;
3. $K(G, n) = \Omega K(G, n+1)$;
4. $K(\mathbb{Z}, 1) = S^1$.

Definition 4.1.5. The n**-th (unreduced) cohomology of a type** X **with coefficients in an abelian group** G is the type

$$H^n(X; G) \equiv \|X \to K(G, n)\|_0.$$

If X is pointed, then n**-th reduced cohomology of a type** X **with coefficients in an abelian group** G is the type

$$\tilde{H}^n(X; G) \equiv \|X \to_* K(G, n)\|_0,$$

where $\to_*$ denotes the pointed function.

Remark 4.1.6. A map $X \to K(G, n)$ can be seen as a pointed map $X + \mathbb{1} \to K(G, n)$, where $X + \mathbb{1}$ is the coproduct of X and the unit type $\mathbb{1}$, pointed by $\mathrm{inr}(*)$ for $* : \mathbb{1}$. Therefore, $H^n(X; G) \simeq \tilde{H}^n(X + \mathbb{1}; G)$. Elements of the latter one can be regarded as elements of the former one by forgetting the basepoint.

Theorem 4.1.7 (Cohomology of the Unit Type)**.** *$\tilde{H}^k(\mathbb{1}; \mathbb{Z}) = \mathbb{1}$ for all $k \geq 0$.*

Proof. We have

$$\tilde{H}^k(\mathbb{1}; \mathbb{Z}) \equiv \|\mathbb{1} \to_* K(\mathbb{Z}, k)\|_0.$$

From Corollary 2.4.3, the right hand side is contractible. □

Theorem 4.1.8 (Cohomology of Spheres)**.** *Let $n \geq 1$. We have $\tilde{H}^k(S^n; \mathbb{Z}) = \mathbb{Z}$ for $k \equiv n$, and trivial otherwise.*

Proof. From Corollary 2.4.7, we deduce that

$$\pi_n(X) \equiv \|\Omega^n(X)\|_0 \simeq \|S^n \to_* X\|_0.$$

So by definition, we have

$$\tilde{H}^k(S^n; \mathbb{Z}) \equiv \|S^n \to_* K(\mathbb{Z}, k)\|_0 = \pi_n(K(\mathbb{Z}, k)).$$

From Fact 4.1.4, this equals $\mathbb{Z}$ only if $n = k$, and 0 otherwise. □

Corollary 4.1.9. *Let $n \geq 1$. We have*

$$H^k(S^n; \mathbb{Z}) = \begin{cases} \mathbb{Z} & , k = 0, n; \\ \mathbb{1} & , \textit{otherwise}. \end{cases}$$

Specially, $H^k(S^n; \mathbb{Z}) = \mathbb{Z}$ only when $k = 0$, and trivial otherwise.

The group and the ring structures of $H^n(X; G)$ come from the respective structures on $K(G, n)$. The readers are referred to Definition 5.1.5 and 5.1.6 in [Bru16]. We always assume the group and the ring structures on cohomology are established from now on. For simplicity, we write $H^n(X)$ for $H^n(X; G)$ when $G \equiv \mathbb{Z}$, and similarly for reduced cohomology. The cohomology theory can be generalized to a functor with the target being the spectra, see §5.3 in [vD18]. However, we will only focus on the special case when the spectrum is the Eilenberg-MacLane spectrum.

As a last reminder, the cohomology in Definition 4.1.5 satisfies the Eilenberg-Steenrod axiom for cohomology. This was proved by Evan Cavallo in his Master's thesis. See [Cav15].

4.2 Spectra

The coefficients of cohomology in abelian groups can be generalized to be in spectra.

Definition 4.2.1. A **prespectrum** (Y, s) is a pair consisting of a pointed type family $Y : \mathbb{Z} \to \mathcal{U}^*$ and a family of **pointed structure maps** $s : \prod_{(n:\mathbb{Z})} Y_n \to_* \Omega Y_{n+1}$. If $s(n)$ (sometimes written as s_n) is a pointed equivalence for all $n : \mathbb{Z}$, we call the prespectrum (Y, s) an **spectrum**. We denote the (pre)spectrum simply by Y if the structure maps are clear from the context.

Equivalently, by Theorem 2.4.6, we can define the pointed structure maps as $s : \prod_{(n:\mathbb{Z})} \Sigma Y_n \to_* Y_{n+1}$.

Definition 4.2.2. A **map between (pre)spectra** $(Y, s) \to (Y', s')$ is a pair consisting of $f : \prod_{(n:\mathbb{Z})} Y_n \to_* Y'_n$ and $p : \prod_{(n:\mathbb{Z})} \Omega s'(n) \circ f(n) \to_* \Omega f(n+1) \circ s(n)$.

From now on, we will assume any (pre)spectrum we take is always a spectrum.

Example 4.2.3. The most commonly used spectrum is the **Eilenberg-MacLane spectrum**. Let A be an abelian group. The Eilenberg-MacLane spectrum HA is defined by $(HA)_n \equiv K(A, n)$ with pointed structure maps $s : \prod_{(n:\mathbb{Z})} (HA)_n \to_* \Omega(HA)_{n+1}$ for $n \geq 0$. In particular, we let $(HA)_0 \equiv K(A, 0) \equiv A$ pointed at the unit $0 : A$. For $n < 0$, we define $(HA)_n \equiv \mathbb{1}$.

Example 4.2.4 (Function Spectra)**.** Let $(X, s), (Y, r)$ be two spectra. The **function spectrum** of (X, s) and (Y, r), denoted by $F(X, Y)$, is the type $\mathrm{Map}_*(X, Y)$, with $F(X, Y)_n \equiv (X_n \to Y_n)$. The structure maps are chosen to be compatible with the structure maps s_n and r_n.

Example 4.2.5 (Dependent Spectra)**.** Let X be a pointed type and Y be a function on X sending each $x : X$ to a spectrum Yx. We define the spectrum $\prod_{(x:X)}(x \to_* Yx)$ by $\left(\prod_{(x:X)}(x \to_* Yx)\right)_n \equiv \prod_{(x:X)}(x \to_* (Yx)_n)$. If Y does not depend on X, we write $X \to_* Y$. These spectra are well-defined, since we have

$$\Omega \prod_{x:X}(x \to_* Yx) \simeq \prod_{x:X}^{*} (x \to_* \Omega(Yx)) \,.$$

If X is not pointed, we can similarly define the dependent spectrum $\prod_{(x:X)}(x \to Yx)$ by $\left(\prod_{(x:X)}(x \to Yx)\right)_n \equiv \prod_{(x:X)} x \to (Yx)_n$. If Y does not depend on X, we write $X \to Y$. Again, by similar reason we can show that these spectra are well-defined.

Let Y be a spectrum. The **n-th homotopy group** of Y is defined to be

$$\pi_n(Y) \equiv \pi_2(Y_{2-n}).$$

Definition 4.2.6. Let $X : \mathcal{U}^*$ be a pointed type and Y be a dependent spectrum. Then the **n-th generalized reduced cohomology of X with coefficients in Y** is the type

$$\tilde{H}^n(X; \lambda x.Yx) \equiv \pi_{-n}\left(\prod_{x:X}(x \to_* Yx)\right) \simeq \left\|\prod_{x:X}(x \to_* (Yx)_n)\right\|_0 .$$

If Y does not depend on X, then

$$\tilde{H}^n(X; Y) \equiv \pi_{-n}(X \to_* Y) \simeq \|X \to_* Y_n\|_0.$$

If X is not pointed, the **n-th generalized (unreduced) cohomology of X with coefficients in Y** is the type

$$H^n(X; \lambda x.Yx) \equiv \pi_{-n}\left(\prod_{x:X}(x \to Yx)\right) \simeq \left\|\prod_{x:X}(x \to (Yx)_n)\right\|_0 .$$

Similarly, when Y does not depend on X,

$$H^n(X; Y) \simeq \|X \to Y_n\|_0$$

In particular, taking $Y \equiv H\mathbb{Z}$, we obtain the usual cohomology in Definition 4.1.5.

Remark 4.2.7. The generalized reduced and unreduced cohomology can be connected similar to Remark 4.1.6. Namely,

$$H^n(X; \lambda x.Yx) \simeq \tilde{H}^n(X + \mathbb{1}; \lambda x.Y_+x),$$

where Y_+ is Y with domain replaced by $X + \mathbb{1}$. In particular, $Y_+(\mathrm{inl}(x)) \equiv Yx$ and $Y_+(\mathrm{inr}(*)) \equiv \mathbb{1}$, where $* : \mathbb{1}$.

Another motivation for spectra is defining homology. While defining cohomology can be relatively straightforward, defining homology, on the other hand, needs to take extra effort since we do not have an accessible version of homological Brown representability. However, by the language of spectra, we can easily define homology groups of the type X with coefficients in the (pre)spectrum Y as the stable homotopy groups of $X \wedge Y$, the smash product of X and Y (see [CS20]). We will not discuss them in this paper.

4.3 Exact Couples

A spectral sequence can be thought of as a book in that each page is a two-dimensional array of abelian groups. On each page, there are collections of (co)chain complexes and maps between them. The maps forming (co)chain complexes are called the **differentials**. The (co)homology of the (co)chain complexes formed by the differentials determine the groups on the next pages.

Formally, each page of a spectral sequence is known as a bigraded abelian group. We will follow Floris' method [vD18] to give a nonstandard definition that is equivalent to the standard one.

Definition 4.3.1. Let G be an abelian group with unit 0. A G**-graded abelian group** M is a family of abelian groups $\{M_x\}$ indexed over G. Let M, M' be two G-graded abelian groups. The type of **(graded abelian group) homomorphism** from M to M' is a triple (e, p, q), where $e : G \simeq G$ is an equivalence of types called a **degree**, $p : \sum_{(g:G)} (e(g) = g + e(0))$, and $q : \sum_{(x,y:G)} \prod_{(p:e(x)=y)} (M_x \to M_y{}')$. We denote the type of homomorphisms from M to M' as $M \to M'$, and write $\deg_\phi \equiv \mathrm{pr}_1(e, p, q)$ for $\phi \equiv (e, p, q) : M \to M'$. Call $\deg_\phi(0)$ the **degree of** ϕ. For $x : I$, where I is an arbitrary subset of G, we write

$$\phi_x \equiv \phi_{\mathrm{refl}_x} : M_x \to M'_{\deg_\phi(x)},$$

and

$$\phi_{[x]} \equiv \phi_{p_x} : M_{\deg_\phi^{-1}(x)} \to M'_x.$$

The maps ϕ_x and $\phi_{[x]}$ are used to describe the maps at each index. The advantage of this definition can be seen in the composition of two graded homomorphisms $\phi : M \to M'$ and $\psi : M' \to M''$. Traditionally, if we want to make the pointwise composite $\lambda(g : G).\lambda(m : M_g).\psi_{g+h}(\phi_g(m))$, we need to transport along the equality $(g + h) + k = g + (h + k)$ to get a graded homomorphism of degree $h + k$. In our setting, we can avoid any transport by directly taking the composite of two degrees $\deg_\psi \circ \deg_\phi$. Note that we have $\deg_{\psi\circ\phi} = \deg_\psi \circ \deg_\phi$ following from the property of projection maps. However, we must point out that sometimes the transport is inevitable, such as considering the homology (see Remark 5.1.2, [vD18]). But we can be relieved since it will not occur in this paper.

Definition 4.3.2. A **spectral sequence** consists of the following data:

1. A sequence E_r of abelian groups graded over $\mathbb{Z} \times \mathbb{Z}$ for $r \geq 2$. E_r is called the r**-page** of the spectral sequence;

2. **differentials**, which are graded homomorphisms $d_r : E_r \to E_r$ such that $d_r \circ d_r = 0$. At $(p,q) : \mathbb{Z} \times \mathbb{Z}$, $d_r^{p,q} : E_r^{p,q} \to E_r^{p+r,q-r+1}$;

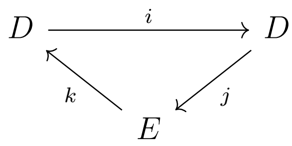
3. isomorphisms $\alpha_r^{p,q} : H^{p,q}(E_r) \simeq E_{r+1}^{p,q}$, where $H^{p,q}(E_r) = \ker(d_r^{p,q})/\mathrm{im}\,(d_r^{p-r,q+r-1})$ is the cohomology of the cochain complex determined by d_r.

The **degree** of the differential d_r is defined to be $(r, 1-r)$.

We start counting the pages at 2 because the first page of the Serre spectral sequence that we will introduce in §4.4 will not be homotopy invariant.

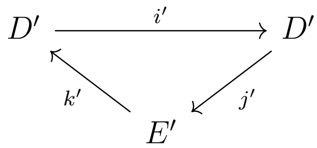
Like in classical homotopy theory, the term (E_r, d_r) in the spectral sequence will determines E_{r+1}, but not d_{r+1}. One way to iterate this construction is through the (derived) exact couples.

Definition 4.3.3. An **exact couple** is a pair (D, E) of $\mathbb{Z} \times \mathbb{Z}$-graded abelian groups with graded homomorphisms i, j, k that is exact in all three vertices:



That is, for all $p : \deg_j(x) = y$ and $q : \deg_k(y) = z$, $\ker(k_q) = \mathrm{im}\,(j_p)$. Similar for the other two pairs of maps. Write $\imath \equiv \deg_i$, $\jmath \equiv \deg_j$, and $\kappa \equiv \deg_k$ for the degrees.

Lemma 4.3.4 (Lemma 5.2.2, [vD18])**.** *Let (D, E, i, j, k) be an exact couple. Then we can define a new exact couple, called **derived exact couple** (D', E', i', j', k'), from (D, E, i, j, k):*



where E' is the homology of $d \equiv j \circ k : E \to E$. The degrees of the derived maps satisfy $\deg_{i'} \equiv \imath$, $\deg_{j'} \equiv \jmath \circ \imath^{-1}$, and $\deg_{k'} \equiv \kappa$.

Repeating the process in Lemma 4.3.4, we obtain a sequence of exact couples $(D_r, E_r, i_r, j_r, k_r)$. They constitute into a spectral sequence (E_r, d_r), where $d_r \equiv j_r \circ k_r$. Note that we have $\deg_{i_r} \equiv \imath$, $\deg_{j_r} \equiv \jmath \circ \imath^{-r}$, $\deg_{k_r} \equiv \kappa$, and $\deg_{d_r} = \deg_{j_r} \circ \deg_{k_r} \equiv \jmath \circ \imath^{-r} \circ \kappa$.

It is easy to see that if $E_2^{p,q}$ in the spectral sequence (E_r, d_r) is trivial, then $E_r^{p,q}$ is trivial for all $r \geq 2$. This is a special case when the spectral sequence converges. Formally, for a fixed pair $(p,q) : \mathbb{Z} \times \mathbb{Z}$, a spectral sequence (E_r, d_r) **converges** if

$E_r^{p,q}$ will be constant for r large enough. In this case, we write $E_\infty^{p,q}$ for the stable (or eventual) value of $E_r^{p,q}$. The first number n such that $E_n^{p,q} = E_\infty^{p,q}$ is called the **stable term** of (E_r, d_r).

The spectral sequence obtained from exact couples with extra conditions can be convergent. To see how, we need the following notation.

Definition 4.3.5. An exact couple is **bounded** if for every $x : \mathbb{Z} \times \mathbb{Z}$, there is a bound $B_x : \mathbb{N}$ such that for all $s \geq B_x$, we have $E^{\imath^{-s}(x)}$ and $D^{\imath^{s}(x)} = \mathbb{1}$. We call x a **stable index** whenever $i^{\imath^{-s}(x)}$ is surjective for all $s \geq 0$.

Lemma 4.3.6 ([vD18], Lemma 5.2.5)**.** *For a bounded exact couple (D, E, i, j, k), we have for all sufficiently large r that $D^x_{r+1} = D^x_r$ and $E^x_{r+1} = E^x_r$, where $x : \mathbb{Z} \times \mathbb{Z}$ is a stable index.*

We end this section by introducing a notation used in classical homotopy theory.

Definition 4.3.7. Let D be an abelian group, and $\{E^n\}$ be a finite sequence of abelian group. We say D is **built from** $\{E^n\}$ if there is a sequence of abelian groups $\{D^n\}$ and a short exact sequence for each $k \geq 1$:

$$E^k \to D^k \to D^{k+1},$$

with $D^0 \equiv D$ and $D^{m+1} \equiv \mathbb{1}$ for some $m : \mathbb{N}$. The sequence $\{D^n\}$ is called a **cofiltration** of D.

Definition 4.3.8. Let D^n be a graded abelian group and $C^{p,q}$ be a bigraded abelian group. We write

$$E_2^{p,q} = C^{p,q} \Rightarrow D^{p+q}$$

if there is a spectral sequence E such that

- $E_2^{p,q} = C^{p,q}$;
- E converges to E_∞;
- D^n is built from $E_\infty^{p,q}$ for $p + q = n$.

4.4 Serre Spectral Sequences for Cohomology

A famous example of spectral sequences is the **Serre spectral sequences** for cohomology. It is a special case of the Atiyah-Hirzebruch spectral sequence. We will mention the latter and prove the Serre spectral sequences with it. For detail, We refer the readers to §5.4 in [vD18].

Definition 4.4.1. Let Y be a spectrum and $k : \mathbb{Z}$, $k \geq -2$. We say Y is k-**truncated** if Y_n is $(k+n)$-truncated for all $n : \mathbb{Z}$.

For instance, the Eilenberg-MacLane spectrum HA for A an abelian group is always k-truncated for $k \geq -2$. We are now ready to state the Atiyah-Hirzebruch spectral sequence.

Theorem 4.4.2 (Atiyah-Hirzebruch Spectral Sequence for Reduced Cohomology)**.** *Let $X : \mathcal{U}^*$ be a pointed type and Y be a function sending each point $x : X$ to spectra Yx. Suppose each spectrum Yx is k-truncated, then we get a spectral sequence with*

$$E_2^{p,q} = \tilde{H}^p(X; \lambda x.\pi_{-q}(Yx)) \Rightarrow \tilde{H}^{p+q}(X; \lambda x.Yx).$$

Proof. See Theorem 5.4.10, [vD18]. □

The unreduced version of the Atiyah-Hirzebruch Spectral Sequence can be obtained by applying the preceding theorem to the function Y_+ in Remark 4.2.7.

Corollary 4.4.3 (Atiyah-Hirzebruch Spectral Sequence for Cohomology)**.** *Same settings as in Theorem 4.4.2. We get a spectral sequence with*

$$E_2^{p,q} = H^p(X; \lambda x.\pi_{-q}(Yx)) \Rightarrow H^{p+q}(X; \lambda x.Yx).$$

Theorem 4.4.4 (Generalized Serre Spectral Sequence for Cohomology)**.** *Let $B : \mathcal{U}$ be a type and $F : B \to \mathcal{U}$ be a type family. Suppose Y is a k-truncated spectrum. Then*

$$E_2^{p,q} = H^p(B; \lambda b.H^q(Fb; Y)) \Rightarrow H^{p+q}\left(\prod_{b:B}(b \times Fb); Y\right).$$

Proof. Apply the Atiyah-Hirzebruch spectral sequence for cohomology to the type B and the function $\lambda b.Fb \to Y$, whose codomains are k-truncated by assumption. We obtain a spectral sequence with E_2-page, by definition, being

$$E_2^{p,q} = H^p(B; \lambda b.\pi_{-q}(Fb \to Y)) = H^p(B; \lambda b.H^q(Fb; Y)),$$

which converges to

$$\begin{aligned} H^{p+q}(B; \lambda b.Fb \to Y) &= \pi_{-(p+q)}\left(\prod_{b:B} b \to Fb \to Y\right) \\ &= \pi_{-(p+q)}\left(\prod_{b:B} (b \times Fb) \to Y\right) \\ &= H^{p+q}\left(\prod_{b:B} (b \times Fb); Y\right). \end{aligned}$$

□

Take $Y \equiv H\mathbb{Z}$. If we let B be a simply-connected space, i.e. $\|B\|_1$ is contractible, then $\lambda b.H^q(Fb; Y) \equiv H^q(Fb_0; Y)$ for some $b_0 : B$ since $F : B \to \mathcal{U}$ is constant by Corollary 2.4.3. Let $F \equiv \mathrm{fib}_f(b_0)$ be the fiber of f at b_0, where $f : X \to B$ with $f(x) = b_0$ for some pointed type (X, x). Geometrically, f can be regarded as a bundle with total space X, base space B, and fiber F. It is not hard to see that $X \simeq \prod_{(b:B)}(b \times Fb)$. We now get the classical Serre spectral sequence for cohomology as follows:

Theorem 4.4.5 (Serre Spectral Sequence for Cohomology)**.** *Let $X, B : \mathcal{U}^*$ be pointed types with $b_0 : B$ and $x : X$, and $F \equiv \mathrm{fib}_f(b_0)$ be the fiber of f at b_0, where $f : X \to B$ with $f(x) = b_0$. Suppose B is simply-connected. Then we have a spectral sequence with*

$$E_2^{p,q} = H^p(B; H^q(F)) \Rightarrow H^{p+q}(X).$$

4.5 Applications of Serre Spectral Sequences

4.5.1 Cohomology of $K(\mathbb{Z}, 2)$

The first application of Serre spectral sequence 4.4.5 is the computation of $H^*(K(\mathbb{Z}, 2))$ and $H^*(K(\mathbb{Z}, 3))$. Consider the **path fibration** over a pointed type (X, x),

$$\Omega X \to PX \to X,$$

where $PX \equiv \mathrm{Map}_*(I, X)$ is the **path space** of X, and $(I, 0_I)$ is the pointed identity type. By Corollary 2.4.3, PX is contractible. We can substitute PX by $\mathbb{1}$ in the path fibration. Now apply Theorem 4.4.5 to the path fibration, we get

$$E_2^{p,q} = H^p(X; H^q(\Omega X)) \Rightarrow H^{p+q}(\mathbb{1}).$$

By Corollary 4.1.9, $H^{p+q}(\mathbb{1}) = \mathbb{Z}$ when $p+q=0$, and is trivial otherwise. Now let $X \equiv K(\mathbb{Z},2)$. Then $\Omega X \equiv \Omega K(\mathbb{Z},2) = K(\mathbb{Z},1) = S^1$. By Corollary 4.1.9, $H^q(\Omega X) = H^q(S^1) = \mathbb{Z}$ when $q=0$ or 1, and is trivial otherwise. The E_2-page of the corresponding spectral sequence is

Figure 4.5.1: E_2-page of $K(\mathbb{Z},2)$.

1	$H^0(X)$	$H^1(X)$	$H^2(X)$	$H^3(X)$	$H^4(X)$
0	$H^0(X)$	$H^1(X)$	$H^2(X)$	$H^3(X)$	$H^4(X)$
	0	1	2	3	4

The functions coming out of $H^3(X)$, $H^4(X)$, and so on are omitted in the figure. The differentials in the subsequent pages are obviously trivial. So $E_3 = E_4 = \cdots = E_\infty$. Since the E_∞-page have only one $\mathbb{Z}$ at the entry $(0,0)$, namely

Figure 4.5.2: E_∞-page of $K(\mathbb{Z},2)$.

1	$\mathbb{1}$	$\mathbb{1}$	$\mathbb{1}$	$\mathbb{1}$	$\mathbb{1}$
0	$\mathbb{Z}$	$\mathbb{1}$	$\mathbb{1}$	$\mathbb{1}$	$\mathbb{1}$
	0	1	2	3	4

Note that there is no function with source or target being $H^0(X)$ or H^1X in the E_2-page. So both $E_2^{0,0}$ and $E_2^{1,0}$ survive, while other terms are killed by the differentials d_2. Thence, $E_\infty^{0,0} = \mathbb{Z} = H^0(X)$ and $E_\infty^{1,0} = \mathbb{1} = H^1(X)$. Going back to Figure 4.5.1, $H^2(X)$ is killed by $H^0(X) \to H^2(X)$, implying $H^2(X) = H^0(X) = \mathbb{Z}$. Similarly, $H^1(X) = H^3(X) = \mathbb{1}$. This yields

$$H^k(X) = \begin{cases} \mathbb{Z} & \text{, when } k \text{ is even;} \\ \mathbb{1} & \text{, when } k \text{ is odd.} \end{cases} \tag{$*$}$$

This is the desired result of the cohomology of $K(\mathbb{Z},2)$.

4.5.2 Cohomology of ΩS^n

Another calculation that can be useful to prove $\pi_k(S^{n-1}) = \pi_{k+1}(S^n)$ is the cohomology of ΩS^n for $n > 1$.

Theorem 4.5.1. *Let $n > 1$. Then*

$$H^k(\Omega S^n) = \begin{cases} \mathbb{Z} & , k = 0 \text{ or } (n-1) \mid k; \\ \mathbb{1} & , \text{otherwise.} \end{cases}$$

Proof. Applying the Serre spectral sequence to the path fibration

$$\Omega S^n \to \mathbb{1} \to S^n,$$

we obtain

$$E_2^{p,q} = H^p(S^n; H^q(\Omega S^n)) \Rightarrow H^{p+q}(\mathbb{1}).$$

By Corollary 4.1.9, $H^{p+q}(\mathbb{1}) = \mathbb{Z}$ when $p + q = 0$, and is trivial otherwise. The E_∞-page is basically similar to the one shown in Figure 4.5.1. For E_2-page, the term $H^p(S^n; H^q(\Omega S^n))$ can only be nontrivial when $p = 0$ or n by Theorem 4.1.8. In that case, $H^p(S^n; H^q(\Omega S^n)) = H^q(\Omega S^n)$. So the E_2-page looks like

Figure 4.5.3: E_2-page of ΩS^n.

3	$H^3(\Omega S^n)$	$H^3(\Omega S^n)$
2	$H^2(\Omega S^n)$	$H^2(\Omega S^n)$
1	$H^1(\Omega S^n)$	$H^1(\Omega S^n)$
0	$H^0(\Omega S^n)$	$H^0(\Omega S^n)$
	0	n

The entries (p, q) for $0 < p < n$ are all trivial groups. There is no nontrivial differential existing on the E_2-page since the target of d_2 must be trivial. Thus, every term survives to the next page. On the E_3-page, things are similar: no nontrivial differential exists, and every term survives to the next page. Repeat this process, we find that nontrivial differentials exist on the E_n-page, and $E_2 = E_3 = \cdots = E_n$. Now we have

Figure 4.5.4: E_n-page of ΩS^n.

$2n-1$	$H^{2n-1}(\Omega S^n)$	
$\vdots$	$\vdots$ $\searrow$	$\vdots$
n	$H^n(\Omega S^n)$	$H^n(\Omega S^n)$
$n-1$	$H^{n-1}(\Omega S^n)$	$H^{n-1}(\Omega S^n)$
$\vdots$	$\vdots$ $\searrow$	$\vdots$
0	$H^0(\Omega S^n)$	$H^0(\Omega S^n)$
	0	n

From the results on E_∞-page, $H^0(\Omega S^n) = \mathbb{Z}$, and $H^k(\Omega S^n) = \mathbb{1}$ for $1 < k < n-1$. $H^0(\Omega S^n)$ at $(n, 0)$-entry of E_n-page will die on E_∞-page, so it must be killed by the differential $d_n^{0,n} : H^{n-1}(\Omega S^n) \to H^0(\Omega S^n)$, implying $H^{n-1}(\Omega S^n) = H^0(\Omega S^n) = \mathbb{Z}$. Similarly, we show $H^{k+n-1}(\Omega S^n) = H^k(\Omega S^n)$ for all $k \geq 0$. Therefore, we prove the desired result. □

4.6 Comments on $\pi_{n+1}(S^n)$

The cohomology of $K(\mathbb{Z}, 2)$ and ΩS^n are two important ingredients in the proof of the following theorem.

Theorem 4.6.1. *$\pi_4(S^3) = \pi_5(S^4) = \cdots = \pi_{n+1}(S^n) = \mathbb{Z}/2\mathbb{Z}$.*

This theorem can be divided into two parts:

Lemma 4.6.2. *$\pi_{k+1}(S^n) = \pi_k(S^{n-1})$ for $n \geq 3$ and $k \leq 2n-4$.*

Lemma 4.6.3. *$\pi_4(S^3) = \mathbb{Z}/2\mathbb{Z}$.*

The sketch of proof of Lemma 4.6.3 goes as follows: consider the fibration $K(\mathbb{Z}, 2) \to \mathbb{1} \to K(\mathbb{Z}, 3)$. Note by Theorem 4.1.8 $\mathbb{Z} = H^3(S^3) = \|S^3 \to K(\mathbb{Z}, 3)\|_0$. We can choose $f : \|S^3 \to K(\mathbb{Z}, 3)\|_0$, which is again a function from S^3 to $K(\mathbb{Z}, 3)$. Pull back [1] the fibration via f to get a new fibration

$$K(\mathbb{Z}, 2) \to X \to S^3,$$

[1] See §4, [AKL15]

where X is an unknown space. Apply Theorem 3.3.5 to the new fibration, we get

$$\begin{aligned}\cdots \to \mathbb{1} \to \pi_k(X) \to \pi_k(S^3) \to \cdots\\ \to \mathbb{1} \to \pi_3(X) \to \pi_3(S^3)\\ \to \pi_2(K(\mathbb{Z},2)) \to \pi_2(X) \to \mathbb{1}\end{aligned}$$

Since $\pi_3(S^3) = \pi_2(K(\mathbb{Z},2)) = \mathbb{Z}$, $\pi_3(X) = \mathbb{1}$. We obtain

$$\pi_k(X) = \begin{cases} \pi_k(S^3) & , k > 3; \\ \mathbb{1} & , k \leq 3. \end{cases}$$

It suffices to calculate $\pi_4(X)$. While we have fewer tools in homotopy, we are familiar with cohomology. The first step is to turn the homotopy into homology. J. Daniel Christensen and Luis Scoccola [CS20] have developed the Hurewicz theorem in homotopy type theorem:

Theorem 4.6.4 (Christensen & Scoccola, 2020)**.** *For $n \geq 1$, X a pointed, $(n-1)$-connected type. and A an abelian group, there is a natural isomorphism*

$$\pi_n(X)^{ab} \otimes A \simeq \tilde{H}_n(X; A),$$

where $\pi_n(X)^{ab}$ is the abelianization of $\pi_n(X)$.

By Hurewicz theorem, $\pi_4(X) = H_4(X)$, where $H_k(X)$ is the "homology" of X. Now applying the Serre spectral sequence to the fibration $K(\mathbb{Z},2) \to X \to S^3$, we can easily find out the cohomology of X. Finally, by a homotopy type theoretical "universal coefficient theorem", we can relate the cohomology of X to the homology of X and then get the desired result. To generalize, we use the "universal coefficient theorem" again to get the homology of ΩS^n from the calculation of $H^n(\Omega S^n)$. Then follow the proof of Theorem 6.2 in [Max18] to get Lemma 4.6.2. Theorem 4.6.1 follows immediately from Lemma 4.6.2 and 4.6.3.

There are two severe problems with this approach:

- Definition of homology.

  Same as cohomology, we cannot use the complexes to define the homology since it is not homotopy invariant. Luckily, like the generalized cohomology theory, we can define homology in the language of spectra. Given a pointed type X and a (pre)spectrum (Y, s), we can form a new (pre)spectrum $X \wedge Y$, called the **smash product** of X and Y. It is given by $(X \wedge Y)_n \equiv X \wedge Y_n$, with structure maps

$$\overline{s_n} : \Sigma(X \wedge Y)_n \equiv X \wedge \Sigma Y_n \xrightarrow{\mathrm{id} \wedge s_n} X \wedge Y_{n+1} \equiv (X \wedge Y)_{n+1}.$$

The homology of X with coefficients in Y is then defined to be $H(X;Y) \equiv \pi_n^s(X \wedge Y)$, where $\pi_n^s(Y)$ is the n-th stable homotopy groups of Y. Christensen and Scoccola have provided a concrete construction of this object. See [CS20]. The definitions and properties of smash products can be found in §4.3, [vD18]. Another reference is §4, [Bru16].

- Universal coefficient theorem (UCT).

  In classical cohomology theory, we have the universal coefficient theorem measuring the isomorphism between cohomology and homology with the error term described by Ext group. However, there is no analog in homotopy type theory for now. The homotopy type theoretical UCT has not even been formularized. The difficulty lies in the definition and algebraic property of homology. Nevertheless, there is still a brunch of hints on this issue. One is the way in stable homotopy theory. Suggested by Peter May's EKMM [EKMM97], we can define the n-th Ext group of spectra M, N to be

  $$\mathrm{Ext}^n_{\mathbb{S}}(M,N) \equiv \pi_{-n}\left(F_{\mathbb{S}}(M,N)\right),$$

  where $F_{\mathbb{S}}(M,N)$ is the function spectrum (see Example 4.2.4) with coefficients in $\mathbb{S}$, and $\mathbb{S}$ is the sphere spectrum with $\mathbb{S}_n = S^n$ and obvious structure maps. UCT can then be stated as follows:

  Theorem 4.6.5 (Universal Coefficient Theorem)**.**

  $$E_2^{p,q} = \mathrm{Ext}^{p,q}_{\mathbb{S}^*}(M^*, N^*) \Rightarrow \mathrm{Ext}^{p+q}_{\mathbb{S}}(M,N),$$

  where $(-)^*$ *denotes the dual of* $(-)$*, i.e.* $Map_*(-, \mathbb{S})$*.*

  This version of UCT can be proved by the generalized Atiyah-Hirzebruch spectral sequence, see Chapter IV.3, [EKMM97]. Substituting M, N by Eilenberg-MacLane spectra, we ought to obtain the UCT in the usual form.

Constructing the homology and proving the UCT in HoTT will be tricky, and getting usable results might take extra work in reformulating or weakening some notions. Yet, with these established, we can give the proof to Theorem 4.6.1 and calculate more homotopy groups in HoTT. Moreover, it is likely a HoTT version of stable homotopy theory can be formalized in the framework.

Chapter 5

Conclusion

In this dissertation, we reviewed the basics of HoTT, and used this constructive language to study the homotopy theory, notably the homotopy groups of spheres. We have shown the formalization of cohomology theory in HoTT, and constructed the Atiyah-Hirzebruch and the Serre spectral sequences via the language of spectra. As in classical homotopy theory, we would expect more spectral sequences formalized, like Adams spectral sequences. The formal language of stable homotopy theory can be established with these tools in hand. A computer-checkable theory of modern homotopy theory can then be put on the agenda. However, there are many obstacles along the way. First is the universal coefficient in HoTT, as discussed in §4.6. Since the usual constructions in cell structures of topological spaces are no longer available, it is likely that we need to start from the algebraic structures on spectra. So a symmetric monoidal functor may be needed in the type of spectra. This is where different models like orthogonal spectra, symmetric spectra, or S-modules got involved. An alternative is to connect the HoTT with ∞-category theory. All of these approaches take much effort. But with time and faith, it is believed that a modern homotopy theory can be built from HoTT. Afterward, with the development of computer science, we may expect a "proof assistant" to calculations in homotopy theory.

References

[Agd] *The HoTT-Agda Library*. `https://github.com/HoTT/HoTT-Agda`.

[AKL15] Jeremy Avigad, Krzysztof Kapulkin, and Peter Lefanu Lumsdaine. *Homotopy Limits in Type Theory*. Mathematical Structures in Computer Science, 25(5):1040–1070, jan 2015.

[AW09] Steve Awodey and Michael A. Warren. *Homotopy Theoretic Models of Identity Types*. Mathematical Proceedings of the Cambridge Philosophical Society, 146(1):45–55, jan 2009.

[Bru16] Guillaume Brunerie. *On the Homotopy Groups of Spheres in Homotopy Type Theory*, 2016. arXiv: 1606.05916.

[Cav15] Evan Cavallo. *Synthetic Cohomology in Homotopy Type Theory*. Master's thesis, Carnegie Mellon University, 2015.

[Chu40] Alonzo Church. *A Formulation of the Simple Theory of Types*. Journal of Symbolic Logic, 5(2):56–68, 1940.

[Chu41] Alonzo Church. *The Calculi of Lambda Conversion. (AM-6)*. Princeton University Press, 1941.

[Coq] *The Coq Proof Assistant*. `https://coq.inria.fr/`.

[CS20] J. Daniel Christensen and Luis Scoccola. *The Hurewicz Theorem in Homotopy Type Theory*, 2020. arXiv: 2007.05833.

[EKMM97] Anthony D. Elmendorf, Igor Kriz, Michael A. Mandell, and J. Peter May. *Rings, Modules, and Algebras in Stable Homotopy Theory*. American Mathematical Soc., 1997.

[KL16] Chris Kapulkin and Peter LeFanu Lumsdaine. *The Homotopy Theory of Type Theories*, 2016. arXiv: 1610.00037.

[Lea] *The HoTT-Lean Prover Library.* https://github.com/leanprover/lean2.

[LF14] Daniel R. Licata and Eric Finster. *Eilenberg-MacLane Spaces in Homotopy Type Theory.* CSL-LICS '14, New York, NY, USA, 2014. Association for Computing Machinery.

[Max18] Laurentţiu Maxim. *Math 754 Chapter II: Spectral Sequences and Applications*, 2018. Lecture notes. URL: https://people.math.wisc.edu/~maxim/spseq.pdf.

[ML75] Per Martin-Löf. *An Intuitionistic Theory of Types: Predicative Part.* In H.E. Rose and J.C. Shepherdson, editors, Logic Colloquium '73, volume 80 of Studies in Logic and the Foundations of Mathematics, pages 73–118. Elsevier, 1975.

[Rus08] Bertrand Russell. *Mathematical Logic as Based on the Theory of Types.* American Journal of Mathematics, 30(3):222–262, 1908.

[Shu13] Mike Shulman. *Spectral Sequences*, 2013. Blog posts. URL: https://homotopytypetheory.org/2013/08/08/spectral-sequences/.

[Shu19] Michael Shulman. *All* $(\infty, 1)$*-toposes Have Strict Univalent Universes*, 2019. arXiv: 1904.07004.

[Uni13] The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics.* https://homotopytypetheory.org/book, Institute for Advanced Study, 2013.

[vD18] Floris van Doorn. *On the Formalization of Higher Inductive Types and Synthetic Homotopy Theory*, 2018. arXiv: 1808.10690.

[Voe12] Vladimir Voevodsky. *A Universe Polymorphic Type System.* URL: https://www.math.ias.edu/Voevodsky/files/files-original/Dropbox/Unfinished_papers/Dynamic_logic/Stage_9_2012_09_01/Universe%20polymorphic%20type%20sytem.pdf, 2012.

[WX10] Guozhen Wang and Zhouli Xu. *A Survey of Computations of Homotopy Groups of Spheres and Cobordisms.* URL: https://ncatlab.org/nlab/files/WangXuHomotopyGroupsOfSpheres2010.pdf, 2010.