Introduction

This is an introductory textbook to univalent mathematics and homotopy type theory, a mathematical foundation that takes advantage of the structural nature of mathematical definitions and constructions. It is common in mathematical practice to consider equivalent objects to be the same. For example, mathematicians often identify isomorphic groups. In set theory it is not possible to make this common practice formal. For example, there are as many distinct trivial groups in set theory as there are distinct singleton sets. Type theory, on the other hand, takes a more structural approach to the foundations of mathematics that accommodates the univalence axiom. This, however, requires us to rethink what it means for two objects to be equal.

The Origins of Homotopy Type Theory

Homotopy type theory emerged in the period from 2006 to 2012, following the discovery of the homotopy interpretation of Martin-Löf's dependent type theory by Awodey and Warren [3] and independently by Voevodsky [25], and Voevodsky's discovery of the univalence axiom [26]. Martin-Löf's dependent type theory [19] is a foundational language for mathematics which is used by many of today's computer proof assistants.

In dependent type theory there are primitive objects called types and primitive objects called elements. Martin-Löf's dependent type theory contains type formation rules for many operations that we are familiar with from traditional mathematics, such as products, sums, and inductive types such as the type of natural numbers. It is called *dependent* type theory because both types and elements may be parametrized by elements of other types.

One of the distinguishing features of Martin-Löf's dependent type theory is

https://doi.org/10.1017/9781108933568.001 Published online by Cambridge University Press

the *identity type*. The identity type

$$\mathsf{Id}_A(a, b)$$

is an example of a dependent type because it is parametrized by two elements $a, b : A$. The elements of the identity type are called identifications, and the type-theoretical way to establish that a and b are equal elements of type A is to define an element in the identity type $\mathsf{Id}_A(a, b)$. In other words, to prove in type theory that two elements a and b in a type A are equal, one has to define an identification $p : \mathsf{Id}_A(a, b)$. It is therefore common to write $a = b$ for the identity type $\mathsf{Id}_A(a, b)$, or if we want to be explicit about the ambient type we can write $a =_A b$.

The rules for the identity type postulate that it is inductively generated by one single element,

$$\mathsf{refl}_a : a =_A a$$

parametrized by $a : A$. This raises an important question. Since the identity type is a type, it could be possible that there are many identifications $p : a =_A b$ between any two elements $a, b : A$. On the other hand, the identity type is generated inductively by only one element, that is by reflexivity. Is it possible to prove, using the rules of the identity type, that there is indeed at most one identification between any two elements of a type?

The pioneers of type theory have already been aware that this seemed impossible, but it was not until Hofmann and Streicher constructed the groupoid model of Martin-Löf's dependent type theory [15] that the question was settled. In their model, types are groupoids and the type of identifications between two objects in a groupoid is the set of isomorphisms between them. Since there can be multiple isomorphisms between two objects in a groupoid, there can be multiple identifications between two objects. Furthermore, they showed that under this interpretation, the identity type indeed satisfies the rule that it is inductively generated by reflexivity. In other words, they soundly refuted the idea that identity types have at most one element. This is quite unlike ordinary mathematics, where two elements of a set are either equal or they aren't. At the end of their paper, they even wondered whether there could be a similar model of type theory using higher groupoids, but the theory of higher groupoids had still been underdeveloped at that point in time. Nevertheless, the stage was set for the homotopy interpretation of type theory to emerge.

In the homotopy interpretation of type theory, we think of types as spaces. Their elements are points in those spaces, and for any two points in a space there is a *space of paths* from one to the other. Analogously, for any two elements in a type there is a *type of identifications* from one to the other. This way of thinking

https://doi.org/10.1017/9781108933568.001 Published online by Cambridge University Press

about types turned out to be very fruitful, and it opened the door to rethinking the foundations of mathematics with a prominent role for homotopy theory. It is important, however, to step back and ask the question:

How is it possible that type-theoretic foundations for mathematics can be so different from the usual set-theoretic foundations for mathematics?

At first glance, types seem to be objects that contain stuff just like sets. There is a type $\mathbb{N}$ of natural numbers, a type $\mathbb{Z}$ of integers, standard finite types Fin_k, function types $A \to B$ and product types $A \times B$, and all of them are not very different from their set-theoretic counterparts. The type of natural numbers contains the natural numbers; the type $A \to B$ contains functions from A to B; the type $A \times B$ contains pairs (a, b) consisting of elements $a : A$ and $b : B$, and so on. A big difference between type theory and set theory, however, is that in type theory types and elements are separate entities, whereas in classical set theory there is a global elementhood relation: Everything in set theory is a set, and for any two sets x and y we can ask the question whether the proposition $x \in y$ holds. In type theory, on the other hand, there are things called types and separately there are things called elements. Furthermore, every element in type theory has a designated type. For example, true and false are specified to be elements of type bool; the numbers $0_{\mathbb{N}}$, $1_{\mathbb{N}}$, $2_{\mathbb{N}}$, and so on, are specified to be elements of type $\mathbb{N}$; the successor function $\mathsf{succ}_{\mathbb{N}}$ is an element of type $\mathbb{N} \to \mathbb{N}$, and the identification refl_a is an element of type $a =_A a$. In other words, types in Martin-Löf's dependent type theory don't share their elements. Whereas in set theory a set x is uniquely determined by its relation $y \in x$ with respect to all other sets y, types in type theory are constructed out of a small set of type-forming operations, and each type-forming operation comes with set of structural rules that postulate how to construct elements of that type and how elements of that type can be used. This simple change of setup has deep implications for how the foundational system works, and ultimately it opens the door to new ways of thinking about the foundations of mathematics, including the homotopy interpretation of type theory—the subject of this book.

First of all it turns out to be extremely useful for computers if we keep track of the types of elements. Most of the widely used computer proof assistants such as Agda, Coq, and Lean are based on a type theory. Given that every element comes equipped with a designated type, the computer can verify whether a function has been applied to elements of the correct type and outputs elements of the specified type. Such type-checking algorithms are at the heart of every proof assistant, and they can be used to verify the correctness of mathematical constructions as well as proofs.

https://doi.org/10.1017/9781108933568.001 Published online by Cambridge University Press

Furthermore, the identity type only compares elements in the same type. The question whether $\mathsf{true} = 1_{\mathbb{N}}$ simply doesn't make sense in type theory, because true is a boolean and $1_{\mathbb{N}}$ is a natural number. However, if the identity type can only compare two elements in the same type, how can we hope to prove that two types are the same? This is possible with universes, which are types of which the elements themselves encode types. Given two types A and B in the same universe $\mathcal{U}$, we can ask whether there is an identification $A =_{\mathcal{U}} B$ in the universe $\mathcal{U}$. The univalence axiom gives a characterization of this identity type. It asserts that an identification of types is equivalently described as an equivalence of types: There is an equivalence

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

for any two types A and B in the same universe $\mathcal{U}$. Roughly speaking, there are as many identifications between A and B in $\mathcal{U}$ as there are equivalences between them. For example, there are $k!$ equivalences from the standard finite type Fin_k to itself, so the univalence axiom implies that there must be $k!$ identifications from Fin_k to itself. Extending this example, the type $\mathsf{Set}_{\mathcal{U}}$ of all sets in a universe $\mathcal{U}$ should be thought of as the groupoid of all sets, because the identifications in $\mathsf{Set}_{\mathcal{U}}$ correspond to equivalences between sets. This directly violates the principles of Zermelo–Fraenkel set theory, because in set theory two sets are equal if and only if they contain exactly the same elements, whereas in type theory the identifications between two types are equivalently described by equivalences between them. For example, there are many distinct singleton sets in Zermelo–Fraenkel set theory, but they are all the same in univalent mathematics.

That raises the question: Is the univalence axiom consistent? The answer is a resounding yes. Voevodsky proposed a model of Martin-Löf's dependent type theory in which he interpreted types as Kan simplicial sets. Kan simplicial sets are the higher groupoids that Hofmann and Streicher alluded to at the end of their paper about the groupoid model of type theory. The simplicial model of type theory with the univalence axiom was later published in [16].

At this point it became clear that Martin-Löf's dependent type theory together with the univalence axiom could serve as a new foundational system for mathematics, which has homotopy theory built into its core. The famous HoTT book [24] was the first textbook exploring this exciting new subject. It was written during the special year 2012–2013 at the Institute for Advanced Study in Princeton as a collaborative effort by over 50 participants. The HoTT book opened up many new avenues of research, including general mathematics from a univalent point of view, (higher) group theory [4], synthetic homotopy theory [8], and modal homotopy type theory [10, 21].

https://doi.org/10.1017/9781108933568.001 Published online by Cambridge University Press

It has now been over a decade since the HoTT book was published. Since its publication, some important open problems have been solved. It was conjectured that homotopy type theory should be modeled by all higher toposes. Higher toposes are ∞-categories in which the objects resemble homotopy types of spaces. The simplest ∞-topos is the ∞-category of simplicial sets, namely Voevodsky's model of univalence. The question whether all ∞-toposes model Martin-Löf's dependent type theory with univalent universes was settled affirmatively by Michael Shulman in [23]. In other words, all theorems proven in homotopy type theory are valid in all ∞-toposes.

Another problem was whether it is possible to find a constructive model of univalence. In the simplicial model of type theory, Voevodsky used the axiom of choice to construct univalent universes. However, type theory is traditionally considered a foundation for constructive mathematics, so it was natural to ask whether it was possible to justify univalence constructively. This question was solved when Bezem, Coquand, and Huber found a model of dependent type theory and univalence in cubical sets [5, 6]. The cubical extension of the Agda proof assistant is based on this model. By the constructive interpretation of univalence in the cubical model it becomes possible to compute with the univalence axiom. Axel Ljungström has recently used cubical Agda to compute and formally verify that Brunerie's number [8], which is a number n such that $\pi_4(\mathbf{S}^3) \cong \mathbb{Z}/n$, is 2.

About This Book

Type theory can be challenging for beginners, particularly because traditional mathematical training often focuses on sets. The difference between set theory and type theory may appear to be rather subtle to the untrained eye. The book therefore starts by focusing in Part I on Martin-Löf's dependent type theory, without going into homotopy theory. We first introduce the system of type dependency, gradually introduce all the type formers with their rules, and show how to get some basic mathematics off the ground in type theory.

In the second part of the book, we build the univalent foundation of mathematics. The central concepts of univalent mathematics are the notion of equivalence, contractibility, the hierarchy of truncation levels which includes propositions and sets, and eventually the univalence axiom. It should be noted that the univalence axiom can technically be introduced as soon as equivalences are defined, but this tends to be confusing rather than enlightening. For a good understanding of the univalence axiom, the student should have a good working knowledge of type theory, and in order to use univalence effectively they should be familiar

https://doi.org/10.1017/9781108933568.001 Published online by Cambridge University Press

with some of the subtleties of introducing mathematical concepts in univalent mathematics. I have three particular examples in mind: The definitions of the image of a map, surjectivity of a map, and finiteness of a type all require some type-theoretical finesse. We cover those topics before we cover the univalence axiom, and they then also serve as a source of illustrative applications of the univalence axiom.

In the final part of the book we define the circle. The circle was the first example of a higher inductive type, and Shulman's proof using the univalence axiom of the fact that its fundamental group is $\mathbb{Z}$ is a pure gem [17]. It led to the realization that the methods of algebraic topology equally apply to univalent type theory, and perhaps it is because of this proof that our subject is called *homotopy type theory*.

Each part of the book is divided into chapters that are roughly the length of one lecture, and at the end of each chapter there is a set of exercises. There are a total of 216 exercises in this book. These exercises are an essential part of the material, and they will be referred to throughout the text. We encourage the reader to read through them, and make sure that they understand what the exercises are asking. When you see an exercise referred to in the text, we hope that you will feel encouraged to try it, or feel rewarded if you have already put in the hard work.

The more ambitious student may even try to formalize the solutions of some of the exercises in a computer proof assistant. Proof assistants provide an excellent way to become familiar with type theory, because they give instant feedback on your work. This book, including the solutions to most of its exercises, has also been formalized in the agda-unimath library [22]. For practice with formalization, all the exercises in Part I on Martin-Löf's dependent type theory are particularly suitable.

https://doi.org/10.1017/9781108933568.001 Published online by Cambridge University Press