Stockholm University

Univalent Constructive Algebraic Geometry

Foundations and Formalizations

Max Zeuner

Doctoral Thesis in Computational Mathematics at Stockholm University, Sweden 2024

Univalent Constructive Algebraic Geometry

Foundations and Formalizations

Max Zeuner

Academic dissertation for the Degree of Doctor of Philosophy in Computational Mathematics at Stockholm University to be publicly defended on Friday 4 October 2024 at 13.00 in lärosal 5, hus 1, Albano, Albanovägen 28.

Abstract

This thesis contains four papers concerned with Homotopy Type Theory and Univalent Foundations (HoTT/UF) and its use as a foundation to study constructive approaches to the theory of schemes, a fundamental notion in algebraic geometry. The main results of this project are presented in the first paper. The other three papers present formalizations of several of these results in Cubical Agda, an extension of the Agda proof assistant with constructive support for univalent foundations. The individual contributions of the papers are as follows.

The first paper, "Univalent Foundations of Constructive Algebraic Geometry", investigates two constructive approaches to defining quasi-compact and quasi-separated schemes (qcqs-schemes) in HoTT/UF, namely qcqs-schemes as locally ringed lattices and as functors from rings to sets. The main result is a constructive and univalent proof that the two definitions coincide, giving an equivalence between the respective categories of qcqs-schemes.

The second paper, "Internalizing Representation Independence with Univalence", discusses an implementation of the so-called structure identity principle (SIP), an important consequence of univalence employed in the other papers, in Cubical Agda and studies its applications in computer science. It is shown that the SIP guarantees representation independence internally for isomorphic implementations of common data structures. The SIP is then generalized to a relational version that can account for wider classes of implementations.

The third paper, "A Univalent Formalization of Constructive Affine Schemes", presents a formalization of affine schemes as ringed lattices in Cubical Agda. Standard textbook presentations of the structure sheaf of an affine scheme often gloss over certain details that turn out to be a serious obstacle when formalizing this construction. The main result of this paper is that with the help of univalence, or rather the SIP for rings and algebras, one can formalize affine schemes in a way that resembles the standard textbook approach more closely than previous formalizations of (affine) schemes in other proof assistants.

The fourth paper, "The Functor of Points Approach to Schemes in Cubical Agda", presents a formalization of qcqs-schemes in Cubical Agda, using Grothendieck's functor of points approach. This is the first formalization following this alternative approach and the first constructive formalization that manages to get beyond affine schemes. The main result is a streamlined definition of functorial qcqs-schemes allowing for a fully formal and constructive proof that compact open subfunctors of affine schemes are qcqs-schemes.

Keywords: *Homotopy Type Theory and Univalent Foundations, Agda, Cubical Agda, Constructive Mathematics, Schemes, Algebraic Geometry.*

Stockholm 2024
http://urn.kb.se/resolve?urn=urn:nbn:se:su:diva-231878

ISBN 978-91-8014-863-4
ISBN 978-91-8014-864-1

Stockholm University

Department of Mathematics

Stockholm University, 106 91 Stockholm

UNIVALENT CONSTRUCTIVE ALGEBRAIC GEOMETRY

Max Zeuner

Stockholm
University

Univalent Constructive Algebraic Geometry

Foundations and Formalizations

Max Zeuner

ISBN print 978-91-8014-863-4
ISBN PDF 978-91-8014-864-1

Printed in Sweden by Universitetsservice US-AB, Stockholm 2024

Abstract

This thesis contains four papers concerned with Homotopy Type Theory and Univalent Foundations (HoTT/UF) and its use as a foundation to study constructive approaches to the theory of schemes, a fundamental notion in algebraic geometry. The main results of this project are presented in the first paper. The other three papers present formalizations of several of these results in `Cubical Agda`, an extension of the `Agda` proof assistant with constructive support for univalent foundations. The individual contributions of the papers are as follows.

The first paper, “Univalent Foundations of Constructive Algebraic Geometry”, investigates two constructive approaches to defining quasi-compact and quasi-separated schemes (qcqs-schemes) in HoTT/UF, namely qcqs-schemes as locally ringed lattices and as functors from rings to sets. The main result is a constructive and univalent proof that the two definitions coincide, giving an equivalence between the respective categories of qcqs-schemes.

The second paper, “Internalizing Representation Independence with Univalence”, discusses an implementation of the so-called structure identity principle (SIP), an important consequence of univalence employed in the other papers, in `Cubical Agda` and studies its applications in computer science. It is shown that the SIP guarantees representation independence internally for isomorphic implementations of common data structures. The SIP is then generalized to a relational version that can account for wider classes of implementations.

The third paper, “A Univalent Formalization of Constructive Affine Schemes”, presents a formalization of affine schemes as ringed lattices in `Cubical Agda`. Standard textbook presentations of the structure sheaf of an affine scheme often gloss over certain details that turn out to be a serious obstacle when formalizing this construction. The main result of this paper is that with the help of univalence, or rather the SIP for rings and algebras, one can formalize affine schemes in a way that resembles the standard textbook approach more closely than previous formalizations of (affine) schemes in other proof assistants.

The fourth paper, “The Functor of Points Approach to Schemes in Cubical Agda”, presents a formalization of qcqs-schemes in `Cubical Agda`, using Grothendieck’s functor of points approach. This is the first formalization following this alternative approach and the first constructive formalization that manages to get beyond affine schemes. The main result is a streamlined definition of functorial qcqs-schemes allowing for a fully formal and constructive proof that compact open subfunctors of affine schemes are qcqs-schemes.

Sammanfattning

Denna avhandling består av fyra artiklar om homotopi-typteori och univalenta grundvalar (HoTT/UF) och dess användning som grund för konstruktiva metoder för teorin om scheman, ett grundläggande begrepp inom algebraisk geometri. Huvudresultaten presenteras i den första artikeln. De tre följande artiklarna presenterar formaliseringar av flera av dessa resultat i `Cubical Agda`, en utökning av bevisassistenten `Agda` med konstruktivt stöd för univalens. Artiklarnas bidrag är följande.

Den första artikeln, ”Univalent Foundations of Constructive Algebraic Geometry”, undersöker två konstruktiva tillvägagångssätt för att definiera kvasikompakta och kvasiseparerade scheman (qcqs-scheman) i HoTT/UF, nämligen som lokalt ringade gitter och som funktorer från ringar till mängder. Huvudresultatet är ett konstruktivt och univalent bevis att de två definitionerna sammanfaller, vilket i sin tur ger en ekvivalens mellan de respektive kategorierna av qcqs-scheman.

Den andra artikeln, ”Internalizing Representation Independence with Univalence”, diskuterar en implementation av den så kallade strukturidentitetsprincipen (SIP), en viktig konsekvens av univalensaxiomet som används i de andra artiklarna, i `Cubical Agda` och studerar dess tillämpningar inom datavetenskap. Det visas att SIP garanterar representationsoberoende internt för isomorfa implementationer av datastrukturer. SIP generaliseras sedan till en version för relationer så att bredare klasser av implementationer kan hanteras.

Den tredje artikeln, ”A Univalent Formalization of Constructive Affine Schemes”, presenterar en formalisering av affina scheman som ringade gitter i `Cubical Agda`. Klassiska presentationer av standardmetoden för att definiera strukturkärven av ett affint schema hoppar ofta över vissa detaljer, vilket visar sig utgöra ett allvarligt hinder när man formaliserar denna konstruktion. Det viktigaste resultatet i artikeln är att univalens, eller snarare SIP för ringar och algebror, möjliggör formalisering av affina scheman på ett sätt som liknar standardmetoden mer än tidigare formaliseringar av (affina) scheman i andra bevisassistenter.

Den fjärde artikeln, ”The Functor of Points Approach to Schemes in Cubical Agda”, presenterar en formalisering av qcqs-scheman i `Cubical Agda`, som bygger på Grothendiecks ”punktfunktor”-metod. Detta är den första formaliseringen som följer denna metod och den första konstruktiva formaliseringen som tar sig bortom affina scheman. Huvudresultat är en elegant definition av funktoriella qcqs-scheman, vilket möjliggör ett fullständigt formellt och konstruktivt bevis av att kompakta öppna subfunktorer av affina scheman är qcqs-scheman.

Acknowledgements

First and foremost, I would like to thank my supervisor Anders Mörtberg, who introduced me to the wonderful world of computer formalization, who would always have an open ear for my ideas and who taught me so much.

In the same vein, my thanks go to my many other collaborators and colleagues that helped me develop scientifically. Stockholm university, and the logic group in particular, were an immensely stimulating research environment.

Receiving the education and the support that eventually allowed me to pursue this PhD is an incredible privilege. For this, I will be forever grateful to my parents. Finally, I want to thank my friends, here in Stockholm and elsewhere, that made this phase of my life so special.

List of papers

The following papers, henceforth referred to by their Roman numerals, are included in this thesis.

Paper I: **Univalent foundations of constructive algebraic geometry**
Max Zeuner
Preprint, 2024

Paper II: **Internalizing representation independence with univalence**
Carlo Angiuli, Evan Cavallo, Anders Mörtberg and Max Zeuner
In *Proceedings of the ACM on Programming Languages*, Volume 5, Issue *POPL*, 2021

Paper III: **A univalent formalization of constructive affine schemes**
Max Zeuner and Anders Mörtberg
In *28th International Conference on Types for Proofs and Programs (TYPES 2022)*, Volume 269 of *Leibniz International Proceedings in Informatics (LIPIcs)*, 2023

Paper IV: **The functor of points approach to schemes in Cubical Agda**
Max Zeuner and Matthias Hutzler
To appear in *15th International Conference on Interactive Theorem Proving (ITP 2024)*, Volume 309 of *Leibniz International Proceedings in Informatics (LIPIcs)*, 2024

The following related paper [PZ23], which is partly based on the author's master's thesis [Zeu22a], is not included in this thesis.

Pre-measure spaces and pre-integration spaces in predicative Bishop–Cheng measure theory
Iosif Petrakis and Max Zeuner
Preprint, accepted for publication in *Logical Methods in Computer Science (LMCS)*, 2022

Papers II–IV have appeared in peer-reviewed conference proceedings. Paper II and an earlier version of Paper III have also appeared in the author's licentiate thesis [Zeu22b], defended January 19, 2023. The author's contributions to the four papers are as follows.

Paper I The paper was entirely written by the author. The topic of the paper and the definition of morphism of qcqs-schemes as locally ringed lattices were first suggested to the author by Thierry Coquand. All other novel mathematical contributions in the paper are the author's own.

Paper II Writing the paper was a collaborative effort. The author contributed substantially to all sections with the exception of section 5 on the relational SIP, where the author only worked on the finite multiset example. For the formalization, the author re-implemented the code of Escardó's SIP in `Cubical Agda` and was involved in formulating the cubical SIP. The author also wrote the code for applying the SIP to the queue example in the paper and the code for an earlier version of the finite multiset example.

Paper III The author took the lead on writing all parts of the paper, except for the background section on `Cubical Agda`. The formalization used results from `Cubical Agda`'s library. However, all code that is specific to the formalization project was written by the author. The author did also take the lead on developing the mathematical contributions of the paper.

Paper IV The paper was entirely written by the author. The author took the lead on the formalization, being responsible for most of the written code, except for the coauthor's code on coverages. The mathematical contributions are based on an idea of the author regarding the representation of compact open subfunctors. This idea was then elaborated and applied collaboratively.

Contents

General introduction

Sections 2.2–2.5, together with Tables 1 and 2, of this introduction are based on the general introduction included in the author's licentiate thesis [Zeu22b], defended January 19, 2023. They appear here in a restructured but otherwise only marginally altered version. All other sections are new.

1 What is this thesis about?

This introduction aims to set the stage for the four papers contained in this thesis. It discusses the core concepts required to understand the context in which the papers are set, the questions that the papers try to answer and the motivations for studying these questions. This section gives a very brief overview of these core notions with forward references to where they are discussed in this general introduction.

At its core, this thesis is concerned with the very foundations of mathematics, in particular non-classical foundations. It uses *type theory* to study *constructive* and *univalent* mathematics. The main Section 2 is devoted to describing what this means in practice. Section 2.1 also gives some historical context, motivating the study of these non-classical approaches. Constructive and univalent mathematics lend themselves to being developed formally. This means that proofs are written with such a level of rigor that their correctness can be established solely by algorithmic means. Formal proofs are usually written with the help of proof assistants, computer systems that allow the user to develop formal proofs interactively, while automatically checking correctness at every step along the way. The results in three of the four papers in this thesis are formalized and use the `Cubical Agda` system. How this works, is discussed in Sections 2.2–2.5.

The subject matter which is studied from the constructive and univalent point of view is algebraic geometry, or rather the basic theory of schemes, a central area of research in modern mathematics. So far, the literature on constructive scheme theory is somewhat sparse leaving many open questions. Schemes are also of interest from a univalent standpoint. Univalence gives a fundamentally different treatment of equality that seems more in line with the way equality of mathematical objects is handled in scheme theory. A very brief introduction to schemes, highlighting the relevant issues of constructivity and equality, can be found in Section 3. One of the main questions that the papers in this thesis try to answer is: What could or should a constructive and univalent theory of schemes that is amenable to formalization in a proof assistant like `Cubical Agda` look like?

This question will be revisited towards the end of this introduction in Section 4, after having discussed the necessary background.

2 Foundations of mathematics

Mathematical truths are widely held to be universal and indisputable. However, with the emergence of the field of mathematical logic in the late 19th century, it became clear that the task of giving a firm logical foundation to the notions of mathematical truth and proof is a rather delicate one. Over the course of the last century, various foundations and approaches to foundations have been put forward. Today, there exists a default foundation for working mathematicians in the form of axiomatic set theory. Nevertheless, exploring alternative foundations remains a fruitful research subject. This section gives some background and context on constructive mathematics, in particular Martin-Löf's constructive type theory, and univalent mathematics. It starts with a brief historical overview of some of the relevant developments, which by no means attempts to give a complete or even fully accurate description. The aim is rather to highlight some key ideas necessary to understand the motivations and contributions of the four papers contained in this thesis.

2.1 Origins of type theory and constructive mathematics

The idea that mathematics should be built on only a handful of "self-evident" axioms, from which all other theorems are deduced, is almost as old as the discipline of mathematics itself, famously serving as the basis of Euclid's "Elements". For almost two millennia, Euclid's work set the gold standard for mathematical rigor, but this view became increasingly challenged in the course of the 19th century. Promising work toward a rigorous foundation for all of mathematics was produced at the end of the 19th century by the likes of Frege, Peano and Cantor.[1] Hilbert's 1900 list of 23 important mathematical problems puts questions on set theory and mathematical logic at the very top of the list.[2]

Only one year into the 20th century, this project met a major obstacle in the form of Russell's paradox. Bertrand Russell realized that both Frege's logical system as well as Cantor's set theory allowed for self-reference within the theory, rendering them inconsistent.[3] The "foundational crisis" ensued, sparking a (in parts heated) debate among prominent figures in mathematics on what a mathematical proof can and should be and the meaning of mathematical truth. Russell

[1] This is of course a rather incomplete list of important contributors. A selection of particularly influential works by the above and other mathematicians can be found in [Hei67].

[2] A comprehensive historical account of the legacy of Hilbert's problems and their influence on the foundations of mathematics and other mathematical fields can be found in [Yan01], which also reprints an English translation of Hilbert's original presentation [Hil02].

[3] Russell dates his discovery to May 1901 [Rus99]. Similar paradoxical results for Cantor's set theory, such as the Burali-Forti paradox [Bur97], were known at the time but not considered as damning as Russell's paradox.

himself proposed a foundation that avoided paradoxes of self-reference together with Alfred North Whitehead in their (roughly 2,000 pages long) magnum opus "Principia Mathematica", published in three volumes between 1910 and 1913 [WR13]. The core idea behind the purported safeguard against paradoxes came to be called type theory.

In classical set theory every mathematical object gets encoded as a set, a mere collection of things, which will ultimately have to be sets themselves. Type theory introduces a strict distinction between objects and their type. The type of an object is uniquely determined and there are strict rules on how the objects of different types can interact with each other. To say that "x is an object of type A", one usually writes $x : A$ to distinguish it from the set-theoretic $x \in A$. While $x \in x$ is a well-formed statement in set theory that can be proved or disproved (hopefully the latter, if one wants to avoid paradoxes), the type-theoretic analogue $x : x$ is nonsensical. Statements like $n : \mathbb{N}$, meaning that n is a natural number, are more of a stipulation, like introducing a new variable, as n can only ever be a natural number nothing else. Nevertheless, replacing ":" by "$\in$", when reading a type-theoretic statement, is often a good enough approximation for readers not familiar with type theory.

The type-theoretic solution did not prevail in the mathematical community. Instead, alternative approaches that kept set theory at the basis of the foundation, but restricted the axioms so that sets could not reference themselves, found widespread acceptance. The so-called Zermelo–Fraenkel set theory with the axiom of choice (ZFC) is often taken to be *the* foundation on which modern mathematics is built.[4] Nevertheless, type theory continued to be studied and made its mark in mathematical logic and computer science. Variations range from Church's "simple theory of types" [Chu40] to Martin-Löf's "intuitionistic type theory" (MLTT) [MS84]. The latter is of particular importance in this thesis and will be described in more detail below. For the remainder of this section, we want to focus on an important inspiration for Martin-Löf's work: intuitionistic or constructive mathematics.

Intuitionism was conceived by L.E.J. Brouwer and obtained a somewhat infamous status as a particularly radical solution to the problems of the foundational crisis. It was heavily informed by Brouwer's anti-realist philosophical convictions about mathematics and mathematical objects [Bro83]. Although it never found widespread acceptance, Brouwer's approach got refined over the course of the last century, revealing its purely logical content. This is what is called intuitionistic or constructive logic, which is perhaps best understood using the so-called Brouwer–Heyting–Kolmogoroff (BHK) interpretation of logic.[5] We will discuss this interpretation in more detail in Section 2.3. Roughly speaking, the BHK

[4] The way in which ZFC is most commonly presented today, as a first order theory with only one relational symbol $\in$ and axiom schemes, is based on [Zer08; Zer30] and [Fra22], but saw contributions from other prominent figures such as Skolem and von Neumann, see [Bag23].

[5] For an overview of the history of the BHK-interpretation, how it became the default for understanding intuitionistic logic and the history of intuitionism/constructivism in general see [Tro11].

interpretation is not so much concerned with the truth of a statement, but rather with *how* we can prove it.

Using constructive logic restricts admissible proof strategies. For example proofs by contradiction that use the so-called *law of excluded middle* or the axiom of choice are no longer allowed. As a result constructive proofs always carry computational content or algorithmic meaning in a very broad sense. The following standard example illustrates this well: Consider the statement

There are irrational numbers a, b such that a^b is rational.

This statement admits a neat non-constructive proof: The number $\sqrt{2}^{\sqrt{2}}$ is either rational or irrational (this is already where the proof becomes non-constructive by using excluded middle). If it is rational, the statement follows by taking $a = b = \sqrt{2}$. If it is irrational, take $a = \sqrt{2}^{\sqrt{2}}$ and $b = \sqrt{2}$ and observe that $\left(\sqrt{2}^{\sqrt{2}}\right)^{\sqrt{2}} = \sqrt{2}^{\sqrt{2}\sqrt{2}} = 2$ is rational. Indeed, this proof lacks computational content, as it does not let one "compute" a power of irrationals as an integer fraction $a^b = {}^p/_q$, at least not without first deciding whether $\sqrt{2}^{\sqrt{2}}$ is rational or not.[6]

Constructive mathematics is often dismissed as too restrictive to be useful for modern mathematics on the whole. A view that was later challenged, in particular through the work of Errett Bishop. His book "Foundations of Constructive Analysis" [Bis67] showed that a large amount of contemporary analysis could be done in a generally constructive setting. Bishop stayed purposefully agnostic on certain foundational matters, as he primarily wanted to demonstrate that a lot of familiar looking mathematics can be developed without invoking proofs by contradiction or the axiom of choice. Martin-Löf, among others, sought to introduce a foundational system in line with the intuitionistic/constructive view on logic, but strong enough to formalize Bishop's work [Mar75]. His MLTT achieves this by introducing dependent types. We will see this in more detail below.

2.2 Formal mathematics

Before moving on to describe how Martin-Löf's ideas gave rise to Homotopy Type Theory and Univalent Foundations, we want to briefly discuss another achievement of type theory that plays a pivotal role in this thesis: The creation of proof assistants. These are software tools that allow the user to write correct formal proofs in a specified foundation. The first fully formal development of a large part of mathematics is perhaps the Principia Mathematica. However, having to maintain the required level of rigor using only pen and paper brought its authors to exhaustion. Russell would later state in his autobiography [Rus99] that his "intellect never quite recovered from the strain" of writing the Principia Mathematica. This illustrates well that such an endeavor without a fair bit of

[6]By the Gelfond-Schneider theorem, $\sqrt{2}^{\sqrt{2}}$ is not only irrational but in fact transcendental.

computer automation, something that was of course not available to Russell and his contemporaries, is ultimately not viable.

Over the last decades proof assistants have come a long way in providing trustworthy verified software and libraries of computer-checked formal mathematics, while becoming evermore user-friendly. Many of the most widely used proof assistants today, are based on type theory. Some are based on the simple theory of types, like `Isabelle/HOL` [NWP02], while others are based on constructive dependent type theory. The latter include `Coq` and `Lean`, both of which are based on the calculus of (inductive) constructions [CH86; CP90], and `Agda`, which is based on a version of MLTT [Agda; Nor07]. MLTT simultaneously serves as a powerful programming language and a formal language for the foundations of (constructive) mathematics [Mar82]. Arguably, it is this dual purpose that allowed for effective proof checkers and usable proof assistants, in which modern mathematics can be formalized.

Notable success stories of formalization projects tend to fall into two categories. The formal proof of the four-color theorem [Gon+08], or the formal proof of the Kepler conjecture by the FlySpeck project [Hal+17], provide formalizations of results that crucially rely on computer programs, whose correctness can hardly be checked by a human. On the other hand, the formal proof of the odd order theorem [Gon+13], a milestone in the classification of finite simple groups, and more recently the Liquid Tensor Experiment [Lea], a formalization of a large part of so-called condensed mathematics, are concerned with very abstract, technical results. Already when written down informally, these results can fill up to several hundred pages, making the peer-review process arduous and increasing the probability of mistakes passing unnoticed. It is this last point that got Fields medalist Vladimir Voevodsky interested in type theory and formal mathematics. After discovering that serious errors in some of his most influential papers had passed unnoticed for several years, he began to look into proof assistants. This resulted in Univalent Foundations, combining ideas from Voevodsky's "higher dimensional" mathematics with intuitionistic type theory.[7] To understand this novel approach, however, we first need to look into how MLTT works as a foundational system.

2.3 Propositions as Types

One of the most striking features of MLTT as a formal language for the foundations of mathematics is that, unlike formal systems for set theory, it is not a logical calculus with mathematical axioms added to it. Instead, it is only concerned with derivation rules for so-called *judgments*. Logic enters the picture via the *propositions-as-types* paradigm [MS84]. The Curry–Howard correspondence states that formulas of intuitionistic propositional logic can be interpreted as types in the simply typed lambda-calculus. Terms of these types represent proofs of the corresponding statement in intuitionistic propositional logic

[7] Voevodsky himself recounts his journey in a very accessible article [Voe14].

[How80]. With dependent types this can be extended to first-order intuitionistic logic.

The first two columns of Table 1 show how logic is interpreted in constructive mathematics according to the Brouwer–Heyting–Kolmogoroff (BHK) interpretation of logic. The BHK interpretation specifies the form a proof should take for a given statement. For example a proof of a conjunction is a pair of proofs and a proof of an implication is a function on proofs. More generally, if propositions are assigned to types and their proofs correspond to terms of the assigned type, the logical connectives should correspond to *type constructors*. This is shown in the last column of Table 1. For the connectives of propositional logic the corresponding type constructors exist in a functional programming language like `Haskell`. Conjuction becomes taking pair types (written $\times$ in `Agda`), disjunction becomes taking sum types (written $\uplus$ in `Agda`) and implication becomes taking function types (written $\rightarrow$ in `Agda`). This is at the heart of how interactive theorem proving with `Agda` works. We translate the statement we want to prove to a type that can be written down in `Agda`'s syntax. A proof formalized in `Agda` is a term of the corresponding type. If that statement is e.g. an implication, then a proof is a lambda-term of the corresponding function type, just like in any other functional programming language.

For dealing with quantification however, simple types are not enough. In order to extend the propositions-as-types paradigm to (constructive) first-order logic, one needs a way of interpreting predicates in the type theory. A predicate P can be seen as a family of propositions indexed by some domain and, analogously, a *dependent type* is a family of types indexed by the terms of some other type. In `Agda` this is written as a sort of function $P : S \rightarrow \mathsf{Type}$. Given any $x : S$, $P(x)$ is a type and might vary for different x. Going one step further one can now look at *dependent functions* (written $(x : S) \rightarrow P(x)$ in `Agda`) that vary along P in their codomain. Dependent functions are also called Π-types and can be used to interpret universal quantification, as shown in Table 1. The other important type constructor are *dependent pairs* or Σ-types. These are written $\Sigma[x \in S]\ P(x)$ in `Agda`. The "$\in$" here should be read as ":", as `Agda`'s syntax does not allow for the use of ":" inside Σ-types. Terms of this type are pairs where the type of the second component depends on the first component. Table 1 explains why these dependent type constructors correspond to the quantifiers in constructive mathematics.

There are some issues with this approach, the most obvious being that not all types are supposed to be propositions. The type of natural numbers $\mathbb{N}$ should not be considered a proposition, and neither should (dependent) functions involving $\mathbb{N}$ be automatically interpreted as an implication or universal quantification. What should be read as a proposition and what as a program is thus an external case by case decision. Variants of intuitionistic type theory, like the calculus of constructions [CH86], the system underlying the `Coq` proof assistant, make this distinction precise by making propositions a different class of types.

Connectives	BHK	Types
$\wedge$ **(and):**	to prove $P \wedge Q$ we must have both, a proof of P and a proof of Q.	$P \times Q$
$\vee$ **(or):**	to prove $P \vee Q$ we must have either a proof of P or have a proof of Q.	$P \uplus Q$
$\rightarrow$ **(implies):**	to prove $P \rightarrow Q$ we must have an algorithm that converts any proof of P into a proof of Q.	$P \rightarrow Q$
$\neg$ **(not):**	to prove $\neg P$ we must show that P implies a contradiction.	$P \rightarrow \bot$
$\forall$ **(for all):**	to prove $\forall(x \in S)\ P(x)$ we must have an algorithm that, applied to any object $x \in S$, provides a proof of $P(x)$.	$(x : S) \rightarrow P(x)$
$\exists$ **(exists):**	to prove $\exists(x \in S)\ P(x)$ we must construct an object $x \in S$ and prove that $P(x)$ holds.	$\Sigma[x \in S]\ P(x)$

Table 1: The BHK interpretation of logic following [BPI22] with the corresponding types under the propositions-as-types paradigm in `Agda` notation.

A more pressing issue is that of equality. MLTT has a notion of judgmental or definitional equality, which in the context of a computer implementation basically tells the computer when syntactic representations of types and terms can be regarded as equal (modulo unfolding definitions, renaming variables, etc.). However, for proving equalities in MLLT we need an *identity type*. For a type A with terms $x, y : A$, the identity type $x \equiv y$ should correspond to the type of proofs that x and y are equal.[8] For example, proving the commutativity of addition of natural numbers in MLTT amounts to providing a dependent function, called $+$-comm, of type

$$+\text{-comm} \quad : \quad (n\ m : \mathbb{N}) \rightarrow n + m \equiv m + n.$$

Going back to Table 1, one might notice that it is missing a row for equality. This is because the way equality is treated in constructive mathematics can not really be emulated by the identity type itself. This is essentially due to the fact that MLTT does not have a satisfactory account of quotients by equivalence relations. The standard solution when formalizing constructive mathematics is to use so-called *setoids*, which can be rather unwieldy in practice.

[8] `Agda`'s use of $\equiv$ for propositional equality and $=$ for judgmental equality can be a source of confusion, as the notation is often used exactly the other way around in the literature.

2.4 Homotopy Type Theory and Univalent Foundations

The nature of the identity type puzzled type theorists for a long time. It seems possible to have several proofs of equality $p, q : x \equiv y$ inhabiting the corresponding identity type and the same could hold for their identity type $p \equiv q$, and so on. Ideally, any two proofs of equality should be equal. However this principle, the *uniqueness of identity proofs* (UIP), was shown independent of MLTT by Hofmann and Streicher [HS98]. Proof assistants like `Coq` or `Agda` have some form of UIP baked into their system by default. A recent, fruitful research program, which takes the diametrically opposed option, is the aforementioned extension of MLTT called Homotopy Type Theory and Univalent Foundations (HoTT/UF). Homotopy type theory observes that in the absence of any form of UIP, identity types imbue types with a very interesting structure, namely that of an ∞-groupoid [AW09]. In particular, the identity types of a type behave like path spaces of a topological space. This results in a new foundation of mathematics, one that is qualitatively different from set-theoretic foundations [Uni13].

The first part of Table 2 shows how the type constructors of MLTT (roughly) correspond to homotopical constructions on spaces. Homotopy type theory also adds new features to MLTT, in particular the *univalence axiom* and so-called *higher inductive types* (HITs). This extended system now allows for a new synthetic approach to homotopy theory and much like spaces have a homotopy type, types have a homotopy level or just h-level. Although the UIP is generally false in HoTT/UF, it is provable for certain types, like $\mathbb{N}$, and these are the exactly the 0-types, i.e. the types of h-level zero.[9] They are also called homotopy sets or h-sets. By restricting to those types, one can recover a new foundation for ordinary mathematics inside HoTT/UF.

This works because there are actually (-1)-types and these are called h-propositions. By restricting the propositions-as-types paradigm to h-propositions we can reason about h-sets in HoTT, interpreting equality of elements of an h-set as the corresponding identity type. In more precise terms, a type P is an h-proposition if we have a dependent function of type

$$\mathsf{isProp}(P) \ := \ (x\ y : P) \to x \equiv y.$$

Similarly a type S is an h-set if we have a dependent function of type

$$\mathsf{isSet}(S) \ := \ (x\ y : S)(p\ q : x \equiv y) \to p \equiv q$$

that is if we have $\mathsf{isProp}(x \equiv y)$ for every $x, y : S$. Unfortunately, not all type constructors used to interpret the logical connectives preserve being a proposition. This holds in particular for sum types interpreting disjunction and dependent pair types interpreting existential quantification. This is remedied by using *propositional truncation*, a HIT that turns any type into an h-proposition.

[9] Sometimes 0-types are said to be of h-level 2 in order to avoid counting from -2 in the hierarchy of n-types/h-levels.

The refined "propositions-as-h-propositions" paradigm can be seen in the second half of Table 2. It is really a refinement of propositions-as-types and the key to understanding the formalizations presented in the papers of this thesis.

Types	Homotopy	Connectives	h-Props
$A \times B$	product space	$\wedge$ (and):	$P \times Q$
$A \uplus B$	coproduct	$\vee$ (or):	$\parallel P \uplus Q \parallel$
$A \to B$	function space	$\to$ (implies):	$P \to Q$
$(x : A) \to B(x)$	space of sections	$\forall$ (for all):	$(x : S) \to P(x)$
$\Sigma[x \in A]\ B(x)$	total space	$\exists$ (exists):	$\parallel \Sigma[x \in S]\ P(x) \parallel$
$x \equiv y$	path space x to y	$=$ (equal):	$x \equiv y$ (in an h-set)

Table 2: Types as spaces and propositions as h-propositions following [Uni13].

The other major addition of HoTT/UF is the univalence principle, which allows one to characterize equalities of types. It states that for two types A, B the type of paths $A \equiv B$ is equivalent to the type of *equivalences* $A \simeq B$. This notion of equivalence of types is somewhat technical but for h-sets A and B it amounts to a classical bijection (a function with an inverse). For h-propositions equivalence amounts to logical equivalence or biimplication. Considering now h-sets equipped with e.g. a group structure, one can show that equivalence in this case amounts to an isomorphism of groups. Hence by univalence, giving an equality of groups is equivalent to giving an isomorphism of groups. As a consequence any definable property of groups in univalent foundations has to be invariant under isomorphism. Generalizing this to broad classes of structures gives rise to the structure identity principle (SIP) and this is the form in which univalence will be used in the papers of this thesis.

2.5 A short note on cubical type theory and `Cubical Agda`

We conclude this background section on type theory with a short discussion of the proof assistant of choice for the formalization projects presented in this thesis: `Cubical Agda`, an extension of the MLTT-based `Agda` proof assistant with "native support for univalence and higher inductive types" [VMA21]. It is not directly built on HoTT/UF, but rather a variant of *cubical type theory*, which is described in [CCHM18] and [CHM18]. Even though the main features of HoTT/UF are still supported, it is worth noticing a few of its peculiarities that set it apart from standard HoTT/UF. Details on how cubical type theory is implemented in `Cubical Agda` can be found in [VMA21].

The most apparent difference is that the intuition of interpreting identity as path spaces is taken very literal in cubical type theory. `Cubical Agda` has a special interval $\mathbb{I}$ with endpoints 0 and 1. For $x, y : A$, the equality, or path type, is written $x \equiv y$. A path is like a function $p : \mathbb{I} \to A$ with $p\,(0) = x$ and $p\,(1) = y$.

Moreover, this generalizes to *dependent paths* over "lines", i.e. $\mathbb{I}$-indexed families of types $B : \mathbb{I} \to \mathsf{Type}$. This complicates some things such as the transitivity of equality, which becomes a problem of "path composition", but it gives a good notion of transport or substitution, even for dependent paths. As a result it is possible to conveniently transfer properties even between complex objects if they are equal up to a path.

In cubical type theory it is possible to prove univalence, which is of course the corner stone of univalent mathematics. As a consequence, `Cubical Agda` is capable of offering constructive support for univalence. Moreover, `Agda`'s pattern-matching for inductive data types extends to higher inductive types in `Cubical Agda`. Above, we have seen that we need one HIT, namely propositional truncation to properly interpret logic via h-propositions. Furthermore, set-quotients, i.e. quotients that are always an h-set, can be defined as a HIT in a well-behaved way in cubical type theory. These two HITs are easily definable and rather usable in `Cubical Agda`. These facts taken together are perhaps the biggest selling point for `Cubical Agda` as a proof assistant for the formalization of constructive, univalent, set-level mathematics. The reader will hopefully find this claim supported by the content of Papers II–IV in this thesis.[10]

3 Algebraic geometry, classical and constructive

Having outlined the particularities and perks of constructive type theory and HoTT/UF as a foundation for mathematics, let us turn to the field of mathematics studied from the univalent and constructive view in this thesis: Algebraic geometry. As a research field, algebraic geometry has a long and proud history. Today, it is mostly taught using the language of *schemes*, which were introduced by Grothendieck in the latter half of the 20th century. Schemes figure as a powerful generalization of previously existing notions and ideas and led to many important mathematical achievements. With the transition to schemes, the fundamental notions used in algebraic geometry became more abstract and (at least on the surface) more entangled with heavy set-theoretic, i.e. non-constructive, machinery.

Algebraic geometry emerged as the study of solution sets of (finite) systems of polynomial equations. Many geometric shapes can be described this way. For example, considering the unit circle embedded into the plane $\mathbb{R}^2$ around the origin, one realizes that the points lying on the circle are precisely the points $(x, y) \in \mathbb{R}^2$ satisfying the equation $x^2 + y^2 - 1 = 0$.[11] Usually, one wants to work over an algebraically closed field like $\mathbb{C}$, in which case the subspaces of $\mathbb{C}^n$ defined

[10] The code for the papers is integrated in the `agda/cubical`-library with a special summary file for each paper. There is even a rendered version that is clickable in a browser and allows one to explore the library without installing `Agda`, see
`https://github.com/agda/cubical`
`https://agda.github.io/cubical/Cubical.README.html`

[11] To see this, let α be the angle that the point (x, y) defines with respect to the x-axis. Then $x = \sin(\alpha)$ and $y = \cos(\alpha)$ and the identity $\sin(\alpha)^2 + \cos(\alpha)^2 = 1$ is always satisfied.

by the solutions of polynomials with complex coefficients are called *affine complex varieties*. By Hilbert's Nullstellensatz, the maximal ideals of the polynomial ring $\mathbb{C}[x_1, ..., x_n]$ are in bijection with the points of $\mathbb{C}^n$. The point $(a_1, ..., a_n) \in \mathbb{C}^n$ corresponds here to $\langle x_1 - a_1, ..., x_n - a_n \rangle$, the maximal ideal generated by monic linear polynomials. Schemes generalize varieties by (in a certain sense) moving "from maximal ideals to prime ideals" (among other things).

For introducing schemes, textbooks like Hartshorne's classic "Algebraic Geometry" [Har13] usually begin by defining the *spectrum* of a ring R:

$$\mathsf{Spec}(R) = \{\mathfrak{p} \subseteq R \text{ prime ideal }\}$$

This is the set of prime ideals of R equipped with the Zariski topology that is generated by basic opens

$$D(f) = \{\mathfrak{p} \mid f \notin \mathfrak{p}\}$$

where $f \in R$. The Zariski closed sets are given by (radical) ideals $\mathfrak{a} \subseteq R$ and are of the form

$$Z(\mathfrak{a}) = \{\mathfrak{p} \mid \mathfrak{a} \subseteq \mathfrak{p}\}$$

In the case where $R = \mathbb{C}[x_1, ..., x_n]$, the closed subsets are in correspondence to affine complex varieties. Finally, one obtains *affine schemes* by equipping $\mathsf{Spec}(R)$ with a sheaf of rings, the so-called *structure sheaf* $\mathcal{O}_R$, which assigns a ring to every Zariski open set, satisfying certain compatibility conditions. It turns out these compatibility conditions allow one to only specify the structure sheaf on basic opens, where one can set

$$\mathcal{O}_R\big(D(f)\big) = R[{}^1\!/\!_f]$$

Here, $R[{}^1\!/\!_f]$ is the ring of fractions ${}^r\!/\!_{f^n}$ where $r \in R$ and the denominator is a power of f. Again, if $R = \mathbb{C}[x_1, ..., x_n]$, $\mathcal{O}_R\big(D(f)\big)$ is the ring of "rational functions" that do not vanish at the roots of the polynomial f.

Already at this stage, before even arriving at general schemes, this construction is very interesting from a foundational point of view. The innocuous looking definition of the structure sheaf is in fact a prime example of the "structuralist" approach of the Grothendieck school, which is at odds with set-theoretic foundations: $\mathcal{O}_R\big(D(f)\big)$ should only depend on $D(f)$, but for $R[{}^1\!/\!_f]$ this only true up to (canonical) isomorphism.

From a constructive point of view, things immediately become problematic when defining the spectrum of a ring. Neither the notion of topological space nor the notion of prime ideal are well-behaved constructively. This is perhaps best illustrated by the fact that without choice principles or Zorn's lemma it is not generally possible to prove that an arbitrary ring possesses a prime or maximal ideal. Furthermore, it is not sufficient to study prime ideals, one also needs to consider *prime filters* of a ring. These are used for localizing at a prime, and classically they are just the set-theoretic complements of prime ideals.

Constructively, the relation between those two notions is rather complicated, at least for rings without decidable equality. Schuster's habilitation thesis [Sch03] contains an insightful and in-depth discussion on constructivity issues regarding the prime spectrum of a ring, which we refer the interested reader to.

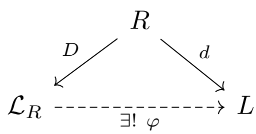
The solution is then to replace the prime spectrum by a point-free counterpart that still contains all the relevant information. A first step in this direction was Joyal's observation [Joy76] that the bounded distributive lattice of (quasi-) compact open sets of the Zariski topology satisfies a certain universal property: For a commutative ring R and a bounded distributive lattice L, call a map $d : L \to R$ a *support* if it satisfies the following conditions:

$$d(1) = \top \quad \text{and} \quad d(0) = \bot \tag{3.1}$$

$$\forall (f\ g \in R) : \ d(fg) = d(f) \wedge d(g) \tag{3.2}$$

$$\forall (f\ g \in R) : \ d(f + g) \leq d(f) \vee d(g) \tag{3.3}$$

The lattice of Zariski compact opens is called the Zariski lattice, denoted $\mathcal{L}_R$. The map $D : R \to \mathcal{L}_R$, sending $f : R$ to the basic open $D(f)$, satisfies conditions (3.1)–(3.3) and it is a universal support in the sense for any other support $d : R \to L$ there is a unique lattice homomorphism $\varphi : \mathcal{L}_R \to L$ such that the following commutes



This means that we can describe the Zariski lattice as the distributive lattice generated by "formal" elements $D(f)$ subject to the relations given by the support conditions.[12] Defining the spectrum of a ring from the point-free Zariski lattice is the approach that is for example taken in Johnstone's "Stone Spaces" [Joh82]. There prime ideals do, however, reenter the picture, in order to obtain the right notion of a morphism for affine schemes as a morphism of locally ringed spaces.

For this, the structure sheaf needs be presented as a sheaf of local rings, which means that its stalks are local rings. In the case of the classical spectrum, these stalks are induced by prime ideals. Building on a body of work starting with the thesis of Hakim [Hak72], Schuster [Sch08] gives a solution to presenting sheaves of local rings and their morphisms without using stalks, by using *supports* instead. Notably, this approach does not use the Zariski lattice but *formal topologies*. The result is a fully constructive account of affine schemes. Unfortunately however, this approach cannot easily be extended to more general schemes. Subsequent work on point-free constructive approaches to schemes, culminating in the work of Coquand, Lombardi and Schuster [CLS09], manages to get past affine schemes by using ringed lattices, but loses the ability to incorporate locality conditions. Instead morphisms of schemes are defined in a somewhat ad-hoc fashion.

[12] This construction can for example be carried out internally in any topos.

In the light of the above issues, it is worth noting that Grothendieck himself preferred the so-called "functor of points approach", which does not require the notion of prime spectrum [GG74]. Here, schemes are taken to be well-behaved functors from rings to sets, i.e. presheaves on the opposite category of commutative rings. Affine schemes are then just the representable presheaves, from which more general schemes can be "glued" together. From a constructive point of view this approach is particularly interesting because it allows one to develop algebraic geometry *synthetically*. The word synthetic means "working in the internal language of a suitable topos", with the internal logic of a topos generally being constructive. The thesis of Blechschmidt [Ble21] contains an excellent introduction to synthetic algebraic geometry for interested readers familiar with classical algebraic geometry.

The biggest drawback of the functor of points approach is perhaps that, from a foundational point of view, it quickly leads to size issues. The standard textbook by Demazure and Gabriel [DG80] assumes *two* Grothendieck universes, while Blechschmidt suggests a "parsimonious" approach, restricting from commutative rings to the small category of finitely presented algebras, allowing one to define schemes of finite presentation [Ble21, III.15]. These concerns do not play a role, when defining schemes a ringed lattices and one would hope that the "big" functorial schemes could still be proved constructively equivalent to lattice schemes. In the absence of a notion of point-free sheaves of local rings on lattices, this problem remained open. It is resolved in the first paper of this thesis by introducing locally ringed lattices.

4 So what is this thesis really about?

Let us return to the overarching question addressed in this thesis: Can we develop the basic theory of schemes constructively, with the help of univalence and ideally accompanied by a formalization in a proof assistant like `Cubical Agda`? The papers in this thesis focus on different aspects of this question. It is also important to observe that the order in which the four papers appear in this thesis does not necessarily reflect the order in which the main ideas of this thesis were developed. Paper I, gathers and presents the main mathematical results informally, while the other papers describe the `Cubical Agda` formalization of some of these results.

Paper II was actually the first paper to be written and it is concerned with applications of univalence to data structures and problems that arise in computer science. However, the formalization of the structure identity principle (SIP) in `Cubical Agda` does have interesting applications in formal mathematics as well. Formalizing schemes as a possible application was popularized by Kevin Buzzard in a series of talks on the equality related problems that arose when formalizing schemes in the `Lean` proof assistant [Buz+21]. As sketched above, the construction of the structure sheaf tacitly treats isomorphic rings as equal, at least in many of the most influential textbooks in algebraic geometry starting with Grothendieck's authoritative "EGA 1" [DG71].[13] It was expected by many in the HoTT/UF community that this problem could be solved with the help of univalence. Indeed, Paper III employs the SIP to give a construction of the structure sheaf that manages to stay relatively faithful to the informal textbook approach. The main result of the paper is thus of a univalent rather than a constructive nature. Since the results were formalized in `Cubical Agda`, using the Zariski lattice instead of the prime ideal spectrum of a ring in the formalization appears to be the natural choice, but it ultimately does not influence the way univalence or the SIP are applied.

Even with constructive affine schemes as the Zariski lattice together with its structure sheaf formalized, a fully constructive formalization of general qcqs-schemes in `Cubical Agda` remained an interesting challenge. Much of the tools presented in Paper I, were developed in order to streamline definitions and make them more amenable for formalization. Thierry Coquand suggested that qcqs-schemes and their morphisms might best be studied using supports and conjectured that this could also help establishing a connection with the functor of points approach. The realization that in view of this connection compact open subschemes of qcqs-schemes can be classified by the Zariski lattice gave rise to the formalization in Paper IV. This formalization is thus not primarily concerned with applications of univalence, albeit written in `Cubical Agda`. It rather demonstrates certain advantages of formalizing the functor of points approach, especially when working constructively. Paper I, which proves the constructive correspondence between qcqs-schemes as (locally) ringed lattices and as functors, adapts the univalent point of view and the reasoning style of the HoTT book [Uni13] in order to stay close to the formalization Papers III and IV. It should however be possible to extrapolate many of its ideas to e.g. Bishop-style constructive mathematics. See the summaries of the included papers below for more details.

5 Loose ends and future directions

This thesis aims to lay the groundwork for univalent constructive algebraic geometry in a manner that is suitable for formalization in a proof assistant like `Cubical Agda`. Paper I describes the necessary mathematics for this program, while Papers II–IV formalize important parts of it in `Cubical Agda`. The question that then immediately arises is: *How feasible is it to formalize the entirety of Paper I*? As discussed in Paper I, the bottleneck should not be extending the formalization of functorial schemes started in Paper III, but the formalization of lattice-theoretic schemes. When describing affine schemes as locally ringed lattices, one might hope that univalence (used in the right way) could be of help, just as it was in Paper III. At this point, however, this remains speculation and an interesting open question. Even then, formalizing the complete proof of the

[13] Buzzard's analysis of this problem can be found in [Buz24].

univalent, constructive comparison theorem, is expected to be a major effort. All in all, a complete formalization of Paper I should be regarded as an extensive research project in univalent formal mathematics.

Besides this natural, formal continuation of the work presented in this thesis, there remain other potentially fruitful questions that are, however, of a more vague nature or rather figure as an open ended program. Having the constructive notions of (qcqs-) schemes pinned down, one might wonder how much of algebraic geometry can be recovered constructively. This work does not necessarily have to be carried out in HoTT/UF, but could also be done in a more foundations-agnostic way by using e.g. Bishop-style constructive mathematics. Entirely speculative, yet potentially interesting even outside of constructive mathematics, is the question whether the additional "computational content" provided by a constructive theory of schemes can be leveraged to allow hands-on computations using otherwise incredibly abstract objects such as schemes.

Another recent fruitful line of research that was already mentioned is synthetic algebraic geometry (SAG), which is currently being studied using HoTT/UF by Cherubini, Coquand and Hutzler [CCH23]. Of particular interest in this context is the problem of constructing models of this "higher" SAG that allow one to actually connect internal definitions to their purported external counterparts. This problem is somewhat similar to the problem of constructing models of HoTT/UF that are in some sense equivalent to spaces. Having a constructive external approach that can conveniently be formalized in a univalent type theory, like `Cubical Agda`'s, could help with understanding and justifying internal notions.

Summaries of included papers

Since Paper II and an earlier version of Paper III are included in the author's licentiate thesis, their summaries are slightly updated versions of the respective summaries in [Zeu22b].

Paper I

The paper "Univalent Foundations of Constructive Algebraic Geometry" presents the foundations of a constructive theory of schemes in Homotopy Type Theory and Univalent Foundations. The paper gives two definitions of quasi-compact and quasi-separated schemes (qcqs-schemes) and proves them equivalent. This result can be seen as a constructivization of the "comparison theorem" of Demazure and Gabriel [DG80]. The paper is written in the informal style of the "HoTT-book" [Uni13] and makes use of univalence and higher inductive types.

The first definition of qcqs-scheme introduced in the paper, builds on the notion of "spectral scheme" introduced by Coquand, Lombardi and Schuster [CLS09]. Spectral schemes are defined as ringed lattices with an affine cover and morphisms of spectral schemes are locally affine morphisms of ringed lattices. The paper introduces the notion of *locally* ringed lattices and their morphisms using ideas of Schuster [Sch08] and of Coquand. One can then define qcqs-schemes as a full subcategory of locally ringed lattices.

The second definition follows Grothendieck's functor of points approach, regarding schemes as well-behaved $\mathbb{Z}$-functors, i.e. functors from rings to sets. The key notion for the functorial definition of qcqs-schemes is that of a compact open subfunctor, which has been studied in the context of synthetic algebraic geometry, in particular in the thesis of Blechschmidt [Ble21]. The classifier for these compact open subobjects was already used for the formalized definition of functorial qcqs-schemes in Paper IV. Building on that definition, this paper gives a proof that compact open subobjects of qcqs-schemes are qcqs-schemes.

The paper gives an adjunction, or rather a relative adjunction (due to size issues), between locally ringed lattices and $\mathbb{Z}$-functors. As its main result, the paper then shows with the help of univalence that when restricted to the respective full subcategories of qcqs-schemes, this becomes an adjoint equivalence.

Paper II

The paper "Internalizing Representation Independence with Univalence" builds on a cubical version of Escardó's *structure identity principle* (SIP) [Esc22]. Although being originally designed for algebraic structures, it is shown how the SIP can be used in computer science to transfer results between isomorphic implementations of common data structures such as queues and finite multisets. This means that in certain cases the SIP can be used to ensure *representation independence*, an important problem in the theory of programming languages [Mit86]: Given two implementations of an abstract data structure, how can one make sure that a program using the abstract data structure will return the same results when run on either implementation?

Implementations of data structures are often not isomorphic. To accommodate for this, the paper introduces a *relational* SIP that allows one to pass from related implementations to isomorphic *quotiented* implementations. The regular SIP can then be applied to the quotiented implementations, ensuring representation independence. In the paper, the relational machinery is applied to finite multisets, to illustrate how representation independence can be obtained for implementations that are originally in a *one-to-many* or *many-to-many* correspondence.

This strategy allows one to get around using parametricity results, which are the usual tool for showing interchangeability of suitably related implementations of a data structure. The advantage of this is that the entire process can be internalized in a sufficiently expressive language such as the cubical type theory underlying `Cubical Agda`. All of the results are formalized and part of the `agda/cubical`-library.

Paper III

The paper "A Univalent Formalization of Constructive Affine Schemes" presents a formalization of affine schemes in `Cubical Agda` that makes crucial use of univalence. Its approach is fully constructive and predicative, generally following Coquand, Lombardi and Schuster [CLS09], while at same time closely following the classic textbook approach of defining the so-called structure sheaf on basis elements by using univalence. As both univalence and HITs have computational meaning in `Cubical Agda`, formalizing constructive mathematics is a rather natural choice. This formal approach to affine schemes can even be seen as a direct continuation of Voevodsky's program [Voe15] of formalizing constructive algebra in Univalent Foundations.

Localizations of commutative rings are treated very similar to [Voe15] and most of the (constructive) commutative algebra needed in this formalization is rather standard. Working constructively and without resizing, the paper uses the so-called *Zariski lattice* associated to a commutative ring that corresponds to the compact open sets of the Zariski topology. The paper demonstrates how this lattice can be constructed in a point-free way that avoids any size issues.

The main part of defining affine schemes is to construct the *structure sheaf* and this works the same way for the Zariski lattice and the classical prime spectrum over some fixed commutative ring R, as both are generated by base elements $D(f)$, where $f : R$. Defining the underlying pre-sheaf and verifying the sheaf property for the structure sheaf only has to be done for these base elements. One wants to map $D(f)$ to $R[1/f]$, the ring of fractions over R whose denominator is a power of f. Proving formally that this gives a well-defined sheaf is however quite difficult for subtle reasons.

Informally, these issues can be ignored by observing that $D(f) \leq D(g)$ gives a *canonical* morphism from $R[1/g]$ to $R[1/f]$. The main result is that by instead observing that $D(f) \leq D(g)$ implies that the type of R-algebra morphisms from $R[1/g]$ to $R[1/f]$ is *contractible*, and combining this with the SIP for R-algebras and cubical transport we can construct the structure sheaf proceeding by the standard textbook strategy.

Paper IV

The paper "The Functor of Points Approach to Schemes in Cubical Agda" presents a formalization of quasi-compact and quasi-separated schemes (qcqs-schemes) in `Cubical Agda`. Unlike previously existing formalizations, it follows Grothendieck's functor of points approach, which does not require a formalization of the notion of locally ringed space or a point-free variant thereof. The standard textbook reference for the functor of points approach is "Introduction to Algebraic Geometry and Algebraic Groups" by Demazure and Gabriel [DG80], which the paper follows in its basic outline. However, due to the fact that the formalization stays within the bounds of `Cubical Agda`'s predicative and constructive type theory, several changes are made.

The paper defines qcqs-schemes as a full subcategory of $\mathbb{Z}$-functors, i.e. as functors from rings in a given type-theoretic universe to sets in the same universe. A key role for the definition of qcqs-scheme is taken by the notion of a compact open subfunctor. In [DG80] the corresponding notion of open subfunctor is defined as a property of subobjects $U \hookrightarrow X$ of a $\mathbb{Z}$-functor X. However, the paper uses a slightly different strategy.

Building on the predicative definition of the Zariski lattice given in Paper III, the paper constructs a *classifier* for compact opens and defines compact open subojects as those induced by natural transformations into the classifier. The restriction to *compact* opens is due to fact that in `Cubical Agda` one cannot define the type of Zariski opens as a small type without resizing assumptions. Using this definition of compact opens associated to a $\mathbb{Z}$-functor, the paper manages to give a streamlined and completely constructive definition of qcqs-schemes, together with a formalized proof that the subfunctor induced by the compact open of an affine scheme is a qcqs-scheme.

Bibliography

[AW09] S. Awodey and M. A. Warren. “Homotopy theoretic models of identity types”. In: *Mathematical proceedings of the cambridge philosophical society*. Vol. 146. 1. Cambridge University Press. 2009, pp. 45–55. DOI: 10.1017/S0305004108001783.

[Bag23] J. Bagaria. “Set Theory”. In: *The Stanford Encyclopedia of Philosophy*. Ed. by E. N. Zalta and U. Nodelman. Spring 2023. Metaphysics Research Lab, Stanford University, 2023. URL: https://plato.stanford.edu/archives/spr2023/entries/set-theory/.

[Bis67] E. Bishop. *Foundations of Constructive Analysis*. McGraw-Hill series in higher mathematics. McGraw-Hill, 1967.

[Ble21] I. Blechschmidt. *Using the internal language of toposes in algebraic geometry*. 2021. arXiv: 2111.03685.

[BPI22] D. Bridges, E. Palmgren, and H. Ishihara. “Constructive Mathematics”. In: *The Stanford Encyclopedia of Philosophy*. Ed. by E. N. Zalta and U. Nodelman. Fall 2022. Metaphysics Research Lab, Stanford University, 2022. URL: https://plato.stanford.edu/archives/fall2022/entries/mathematics-constructive/.

[Bro83] L. E. J. Brouwer. “Intuitionism and Formalism”. In: *Philosophy of Mathematics*. Ed. by P. Benacerraf and H. Putnam. Cambridge University Press, 1983, pp. 77–89.

[Bur97] C. Burali-Forti. “Una questione sui numeri transfiniti”. *Rendiconti del Circolo Matematico di Palermo (1884-1940)* 11.1 (1897), pp. 154–164. Translated by J. van Heijenoort in [Hei67].

[Buz24] K. Buzzard. *Grothendieck’s use of equality*. 2024. arXiv: 2405.10387.

[Buz+21] K. Buzzard et al. “Schemes in Lean”. *Experimental Mathematics* 31.2 (2021), pp. 355–363. DOI: 10.1080/10586458.2021.1983489.

[CCH23] F. Cherubini, T. Coquand, and M. Hutzler. *A Foundation for Synthetic Algebraic Geometry*. 2023. arXiv: 2307.00073.

[Chu40] A. Church. “A Formulation of the Simple Theory of Types”. *The journal of symbolic logic* 5.2 (1940), pp. 56–68.

[CCHM18] C. Cohen et al. “Cubical Type Theory: A Constructive Interpretation of the Univalence Axiom”. In: *21st International Conference on Types for Proofs and Programs (TYPES 2015)*. Ed. by T. Uustalu. Vol. 69. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2018, 5:1–5:34. ISBN: 978-3-95977-030-9. DOI: `10.4230/LIPIcs.TYPES.2015.5`.

[CP90] T. Coquand and C. Paulin. “Inductively defined types”. In: *Proceedings of the International Conference on Computer Logic*. COLOG-88. Tallinn, USSR: Springer-Verlag, 1990, pp. 50–66. URL: `https://dl.acm.org/doi/10.5555/88278.88291`.

[CHM18] T. Coquand, S. Huber, and A. Mörtberg. “On Higher Inductive Types in Cubical Type Theory”. In: *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science*. LICS 2018. Oxford, United Kingdom: ACM, 2018, pp. 255–264. ISBN: 978-1-4503-5583-4. DOI: `10.1145/3209108.3209197`.

[CH86] T. Coquand and G. Huet. *The calculus of constructions*. INRIA, 1986. URL: `https://hal.inria.fr/inria-00076024`.

[CLS09] T. Coquand, H. Lombardi, and P. Schuster. “Spectral Schemes as Ringed Lattices”. *Annals of Mathematics and Artificial Intelligence* 56.3 (2009), pp. 339–360. DOI: `10.1007/s10472-009-9160-7`.

[DG80] M. Demazure and P. Gabriel. *Introduction to algebraic geometry and algebraic groups*. Elsevier, 1980.

[DG71] J. Dieudonné and A. Grothendieck. *Éléments de géométrie algébrique*. Vol. 166. Springer Berlin Heidelberg New York, 1971.

[Esc22] M. H. Escardó. *Introduction to Univalent Foundations of Mathematics with Agda*. 2022. arXiv: `1911.00580`.

[Fra22] A. Fraenkel. “Zu den Grundlagen der Cantor-Zermeloschen Mengenlehre”. *Mathematische Annalen* 86.3 (1922), pp. 230–237. DOI: `10.1007/BF01457986`.

[Gon+08] G. Gonthier et al. “Formal Proof – The Four-Color Theorem”. *Notices of the AMS* 55.11 (2008), pp. 1382–1393. URL: `https://www.ams.org/notices/200811/tx081101382p.pdf`.

[Gon+13] G. Gonthier et al. “A Machine-Checked Proof of the Odd Order Theorem”. In: *Interactive Theorem Proving*. Ed. by S. Blazy, C. Paulin-Mohring, and D. Pichardie. Springer Berlin Heidelberg, 2013, pp. 163–179. ISBN: 978-3-642-39634-2. DOI: `10.1007/978-3-642-39634-2_14`.

[GG74] A. Grothendieck and F. Gaeta. *Introduction to Functorial Algebraic Geometry: After a Summer Course*. State University of New York at Buffalo, Department of Mathematics, 1974. `https://ncatlab.org/nlab/files/GrothendieckIntrodFunctorialGeometryI1973.pdf`.

[Hak72] M. Hakim. *Topos annelés et schémas relatifs*. Berlin: Springer-Verlag, 1972.

[Hal+17] T. Hales et al. "A formal proof of the Kepler conjecture". *Forum of Mathematics, Pi* 5 (2017), e2. DOI: `10.1017/fmp.2017.1`.

[Har13] R. Hartshorne. *Algebraic geometry*. Vol. 52. Springer Science & Business Media, 2013.

[Hei67] J. van Heijenoort. *From Frege to Gödel: A source book in mathematical logic, 1879–1931*. Vol. 9. Harvard University Press, 1967.

[Hil02] D. Hilbert. "Mathematical Problems – Lecture delivered before the International Congress of Mathematicians at Paris in 1900". *Bulletin of the American Mathematical Society* 8 (1902), pp. 437–479. Translated, with the author's permission, by Mary Winston Newsom.

[HS98] M. Hofmann and T. Streicher. "The groupoid interpretation of type theory". *Twenty-five Years of Constructive Type Theory (Venice, 1995)* 36 (1998), pp. 83–111.

[How80] W. A. Howard. "The formulae-as-types notion of construction". *To HB Curry: essays on combinatory logic, lambda calculus and formalism* 44 (1980), pp. 479–490.

[Joh82] P. T. Johnstone. *Stone spaces*. Vol. 3. Cambridge university press, 1982.

[Joy76] A. Joyal. "Les théoremes de Chevalley-Tarski et remarques sur lalgèbre constructive". *Cahiers Topologie Géom. Différentielle* 16 (1976), pp. 256–258.

[Lea] Leanprover-community. *The liquid tensor experiment*. URL: `https://github.com/leanprover-community/lean-liquid`. Accessed July 2024.

[Mar75] P. Martin-Löf. "An Intuitionistic Theory of Types: Predicative Part". In: *Logic Colloquium '73*. Ed. by H. E. Rose and J. C. Shepherdson. Vol. 80. Studies in Logic and the Foundations of Mathematics. North-Holland, 1975, pp. 73–118. DOI: `10.1016/S0049-237X(08)71945-1`.

[Mar82] P. Martin-Löf. “Constructive Mathematics and Computer Programming”. In: *Logic, Methodology and Philosophy of Science VI*. Ed. by L. J. Cohen et al. Vol. 104. Studies in Logic and the Foundations of Mathematics. Elsevier, 1982, pp. 153–175. DOI: `https://doi.org/10.1016/S0049-237X(09)70189-2`.

[MS84] P. Martin-Löf and G. Sambin. *Intuitionistic Type Theory*. Vol. 9. Bibliopolis Naples, 1984.

[Mit86] J. C. Mitchell. “Representation Independence and Data Abstraction”. In: *Proceedings of the 13th ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages*. POPL ’86. St. Petersburg Beach, Florida: Association for Computing Machinery, 1986, pp. 263–276. ISBN: 9781450373470. DOI: `10.1145/512644.512669`.

[NWP02] T. Nipkow, M. Wenzel, and L. C. Paulson. *Isabelle/HOL: A Proof Assistant for Higher-Order Logic*. Springer, 2002. ISBN: 978-3-540-45949-1. DOI: `10.1007/3-540-45949-9`.

[Nor07] U. Norell. “Towards a practical programming language based on dependent type theory”. PhD thesis. SE-412 96 Göteborg, Sweden: Department of Computer Science and Engineering, Chalmers University of Technology, Sept. 2007.

[PZ23] I. Petrakis and M. Zeuner. *Pre-measure spaces and pre-integration spaces in predicative Bishop-Cheng measure theory*. 2023. arXiv: `2207.08684`.

[Rus99] B. Russell. *The Autobiography of Bertrand Russell (2nd ed.)* Routledge, 1999. DOI: `10.4324/9781315824499`.

[Sch08] P. Schuster. “The Zariski spectrum as a formal geometry”. *Theoretical Computer Science* 405.1 (2008), pp. 101–115. DOI: `10.1016/j.tcs.2008.06.030`.

[Sch03] P. M. Schuster. *Spectra and sheaves in formal topology*. 2003. Habilitationsschrift, Universität München.

[Agda] The Agda Development Team. *The Agda Programming Language*. URL: `https://wiki.portal.chalmers.se/agda/pmwiki.php`. Accessed July 2024.

[Tro11] A. S. Troelstra. “History of constructivism in the 20th century”. In: *Set Theory, Arithmetic, and Foundations of Mathematics: Theorems, Philosophies*. Ed. by J. Kennedy and R. Kossak. Lecture Notes in Logic. Cambridge University Press, 2011, pp. 150–179.

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

[VMA21] A. Vezzosi, A. Mörtberg, and A. Abel. "Cubical Agda: A Dependently Typed Programming Language with Univalence and Higher Inductive Types". *Journal of Functional Programming* 31 (2021), e8. DOI: `10.1017/S0956796821000034`.

[Voe15] V. Voevodsky. "An experimental library of formalized Mathematics based on the univalent foundations". *Mathematical Structures in Computer Science* 25.5 (2015), pp. 1278–1294. DOI: `10.1017/S0960129514000577`.

[Voe14] V. Voevodsky. *The Origins and Motivations of Univalent Foundations - A Personal Mission to Develop Computer Proof Verification to Avoid Mathematical Mistakes*. 2014. URL: `https://www.ias.edu/ideas/2014/voevodsky-origins`. Accessed June 2024.

[WR13] A. N. Whitehead and B. Russell. *Principia Mathematica. Vol. I–III.* Cambridge: University Press, 1910, 1912, 1913.

[Yan01] B. Yandell. *The honors class: Hilbert's problems and their solvers.* AK Peters/CRC Press, 2001.

[Zer30] E. Zermelo. "Über Grenzzahlen und Mengenbereiche: Neue Untersuchungen über die Grundlagen der Mengenlehre". *Fundamenta Mathematicae* 16 (1930), pp. 29–47.

[Zer08] E. Zermelo. "Untersuchungen über die Grundlagen der Mengenlehre. I". *Mathematische Annalen* 65.2 (1908), pp. 261–281. DOI: `10.1007/BF01449999`.

[Zeu22a] M. Zeuner. *Families of Sets in Constructive Measure Theory*. Master's thesis LMU (arXiv version). 2022. arXiv: `2207.04000`.

[Zeu22b] M. Zeuner. *Formalizing Univalent Set-Level Structures in Cubical Agda*. Licentiate thesis in computational mathematics, Stockholm University, Department of Mathematics. 2022. URL: `https://urn.kb.se/resolve?urn=urn:nbn:se:su:diva-213221`.